Contents

1 About this manual 2
  1.1 Goal of the manual 2
  1.2 Contents of the manual 2

2 Migration Guide 4
  2.1 Introduction 4
  2.2 Hardware revision 5

3 Imprint 5

4 Legal notice 6
  4.1 Introduction 6
  4.2 cJSON 6
    4.2.1 cJSON license 6
  4.3 Unity 6
    4.3.1 Unity license 6

5 Revisions 8

6 Graphic Symbols 23
  6.1 Notes, Warnings, Attentions 23
  6.2 Webcasts 23

7 Important information 24
  7.1 Important safety instructions 24
  7.2 Operating considerations 24
    7.2.1 Important safety notes 24
    7.2.2 Handling and cleaning 25
    7.2.3 Installing 25
    7.2.4 Optimizing performance and life time 25
    7.2.5 Connectors 25
    7.2.6 Cleaning 25
    7.2.7 Adjusting the C-mount 26
    7.2.8 Additional temperature and power consumption notes concerning mvBlueCOUGAR-XD and mvBlueCOUGAR-X2xx 27
  7.3 European Union Declaration of Conformity statement 27
  7.4 Legal notice 32
    7.4.1 For customers in the U.S.A. 32
    7.4.2 For customers in Canada 33
    7.4.3 Pour utilisateurs au Canada 33
10 Technical data

10.1 mvBlueCOUGAR-X / -POE
10.1.1 Dimensions
10.1.2 Connectors
10.1.3 Signal LED

10.2 mvBlueCOUGAR-X-POE-I / -IP67
10.2.1 Dimensions
10.2.2 Connectors

10.3 mvBlueCOUGAR-XD
10.3.1 Dimensions
10.3.2 Connectors
10.3.3 Signal LED

10.4 mvBlueCOUGAR-3X
10.4.1 Dimensions
10.4.2 Model specific features

10.5 Summary of components
10.6 Summary of advanced features

11 Sensor overview

11.1 Image data flow
11.2 CCD sensors
11.2.1 mvBlueCOUGAR-XD specific CCD sensors
11.2.2 Negative gain

11.3 mvBlueCOUGAR-X specific CMOS sensors
11.3.1 Sony Pregius
11.3.2 Aptina, CMOSIS, e2v
11.3.3 mvBlueCOUGAR-XD specific CMOS sensors

11.4 Output sequence of color sensors (RGB Bayer)
11.5 Bilinear interpolation of color sensors (RGB Bayer)
11.6 Supported image formats
14 GenICam and Advanced Features

14.1 Introduction .................................................. 167

14.2 Device Control ............................................... 168

14.3 Image Format Control ...................................... 169

14.4 Acquisition Control ....................................... 170

14.5 Counter And Timer Control .............................. 173

14.6 Analog Control ............................................... 174

14.7 mv Logic Gate Control .................................... 176

14.8 Color Transformation Control ......................... 177

14.9 mv Flat Field Correction Control .................... 178

14.10 Event Control ................................................ 178

14.11 Chunk Data Control ...................................... 179

14.12 File Access Control ....................................... 180

14.13 mv Serial Interface Control ............................ 180

14.14 Digital I/O Control ........................................ 181

14.15 Encoder Control ............................................ 182

14.16 LUT Control .................................................. 182

14.16.1 mvLUTType .............................................. 183

14.16.2 mvLUTInputData ...................................... 184

14.16.3 mvLUTMapping ......................................... 184

14.16.4 LUT support in MATRIX VISION cameras .... 184

14.17 Sequencer Control ......................................... 190

14.17.1 Sequencer overview .................................. 190

14.17.2 Configuration of a sequencer set ................. 190

14.18 mv Defective Pixel Correction Control ............ 194

14.19 Transport Layer Control ................................ 194

14.20 User Set Control ............................................ 195

14.21 mv Frame Average Control (only with specific models) .................................................. 196

14.22 mv High Dynamic Range Control (only with specific sensor models) ......................... 196

14.23 Action Control ................................................ 197
15 C developers

16 C++ developers

17 .NET developers

18 Python developers

18.1 Introduction

18.2 Building

18.2.1 Windows

18.2.2 Linux

18.3 Using

19 DirectShow Interface

19.1 Supported Interfaces

19.1.1 IAMCameraControl

19.1.2 IAMDroppedFrames

19.1.3 IAMStreamConfig

19.1.4 IAMVideoProcAmp

19.1.5 IKsPropertySet

19.1.6 ISpecifyPropertyPages

19.2 Logging

19.3 Registering and renaming devices for DirectShow usage

19.3.1 Registering devices

19.3.2 Renaming devices

19.3.3 Make silent registration
20 Troubleshooting

20.1 I can see the camera in mvIPConfigure, but I cannot access the camera

20.2 Image Data is lost (incomplete frames)

20.3 There are image error counts

20.4 I get an exception when setting offsetX or offsetY

20.5 Cognex VisionPro

20.5.1 I cannot use mvBlueCOUGAR-XD

20.5.2 I cannot initialize 4 Tap sensor mvBlueCOUGAR-XDs

20.6 I get an oscillating frame rate

20.7 I cannot focus on distant objects

20.8 Accessing log files

20.8.1 Windows

20.8.2 Linux

20.9 mvGigEConfigure freezes when trying to activate/deactivate the filter driver a Link Aggregation Group (LAG)

20.9.1 Symptoms

20.9.2 Cause

20.9.3 Resolution

20.10 mvBlueCOUGAR-X105G with frame rates < 5.75 fps

20.10.1 Cause

20.10.2 Resolution

20.11 Why does updating the device list take so long

20.12 Why is there no GenICam device or interface found on my Linux system

20.13 Error code list

21 Glossary
22 Use cases

22.1 GenICam to mvIMPACT Acquire code generator

22.1.1 Using the code generator

22.1.2 Using the result of the code generator in an application

22.2 Introducing acquisition / recording possibilities

22.2.1 Acquiring a number of images

22.2.2 Recording sequences in the camera

22.2.3 Recording sequences with pre-trigger

22.2.4 Creating acquisition sequences (Sequencer Control)

22.2.5 Generating very long exposure times

22.2.6 Working with multiple AOIs (mv Multi Area Mode)

22.2.7 Working with burst mode buffer

22.2.8 Using the SmartFrameRecall feature

22.2.9 Using VLC Media Player

22.2.10 Using the linescan mode

22.2.11 Working with Event Control

22.3 Improving the acquisition / image quality

22.3.1 Correcting image errors of a sensor

22.3.2 Optimizing the color fidelity of the camera

22.3.3 Reducing noise by frame averaging

22.3.4 Optimizing the bandwidth

22.3.5 Setting a flicker-free auto expose and auto gain

22.3.6 Working with binning

22.3.7 Working with Sony’s 4 Tap CCD sensors

22.3.8 Minimizing sensor pattern of mvBlueCOUGAR-X1010G

22.3.9 Working with the dual exposure feature ("mvMultiZone") of mvBlueCOUGAR-XD107

22.3.10 Working with the dual gain feature of mvBlueCOUGAR-XD107/XD1031 and mvBlueCOUGAR-GAR-X102m/X107

22.3.11 Working with the dual ADC feature of mvBlueCOUGAR-XD107

22.4 Working with triggers

22.4.1 Processing triggers from an incremental encoder
Appendix A.1 CCD specific camera/sensor data

23.1 mvBlueCOUGAR-Xx20a (0.3 Mpix [640 x 480])

23.1.1 Introduction
23.1.2 Details of operation
23.1.3 CCD Timing
23.1.4 Reprogramming CCD Timing
23.1.5 CCD Sensor Data
23.1.6 Device feature And Property List

23.2 mvBlueCOUGAR-Xx20b (0.3 Mpix [640 x 480])

23.2.1 Introduction
23.2.2 Details of operation
23.2.3 CCD Timing
23.2.4 Reprogramming CCD Timing
23.2.5 CCD Sensor Data
23.2.6 Device Feature And Property List

23.3 mvBlueCOUGAR-Xx20d (0.5 Mpix [776 x 580])

23.3.1 Introduction
23.3.2 Details of operation
23.3.3 CCD Timing
23.3.4 Reprogramming CCD Timing
23.3.5 CCD Sensor Data
23.3.6 Device Feature And Property List

23.4 mvBlueCOUGAR-Xx22 (1.2 Mpix [1280 x 960])

23.4.1 Introduction
23.4.2 Details of operation
23.4.3 CCD Timing
23.4.4 Reprogramming CCD Timing
23.4.5 CCD Sensor Data

23.5 mvBlueCOUGAR-Xx23 (1.4 Mpix [1360 x 1024])

23.5.1 Introduction
23.5.2 Details of operation ........................................... 465
23.5.3 CCD Timing ....................................................... 467
23.5.4 Reprogramming CCD Timing ................................. 469
23.5.5 CCD Sensor Data ............................................... 469
23.5.6 Device Feature And Property List ......................... 474

23.6 mvBlueCOUGAR-Xx24 (1.9 Mpix [1600 x 1200]) .............. 475
23.6.1 Introduction ..................................................... 475
23.6.2 Details of operation ........................................... 475
23.6.3 CCD Timing ....................................................... 478
23.6.4 Reprogramming CCD Timing ................................. 479
23.6.5 CCD Sensor Data ............................................... 480
23.6.6 Device Feature And Property List ......................... 484

23.7 mvBlueCOUGAR-XD124a (3 Mpix [1936 x 1460]) ............. 484
23.7.1 Introduction ..................................................... 484
23.7.2 Details of operation ........................................... 484
23.7.3 CCD Timing ....................................................... 487
23.7.4 Reprogramming CCD Timing ................................. 488
23.7.5 CCD Sensor Data ............................................... 489
23.7.6 Device Feature And Property List ......................... 493

23.8 mvBlueCOUGAR-X225 (5.1 Mpix [2448 x 2050]) .............. 494
23.8.1 Introduction ..................................................... 494
23.8.2 Details of operation ........................................... 494
23.8.3 CCD Timing ....................................................... 496
23.8.4 Reprogramming CCD Timing ................................. 498
23.8.5 CCD Sensor Data ............................................... 499
23.8.6 Device Feature And Property List ......................... 503

23.9 mvBlueCOUGAR-Xx25a (5.1 Mpix [2448 x 2050]) ............. 504
23.9.1 Introduction ..................................................... 504
23.9.2 Details of operation ........................................... 504
23.9.3 CCD Timing ....................................................... 506
24 Appendix A.2 Pregius CMOS specific camera / sensor data

24.1 mvBlueCOUGAR-X100f (0.4 Mpix [728 x 544]) .......................... 563
   24.1.1 Introduction ........................................... 563
   24.1.2 Details of operation .................................. 563
   24.1.3 Sensor Data ............................................. 565
   24.1.4 Device Feature And Property List .................. 566

24.2 mvBlueCOUGAR-X[D]102f (1.6 Mpix [1456 x 1088]) .................. 566
   24.2.1 Introduction ........................................... 566
   24.2.2 Details of operation .................................. 566
   24.2.3 Sensor Data ............................................. 568
   24.2.4 Device Feature And Property List .................. 569

24.3 mvBlueCOUGAR-X102k (1.8 Mpix [1608 x 1104]) ...................... 570
   24.3.1 Introduction ........................................... 570
   24.3.2 Details of operation .................................. 570
   24.3.3 Sensor Data ............................................. 572
   24.3.4 Device Feature And Property List .................. 573

24.4 mvBlueCOUGAR-X102m (1.8 Mpix [1608 x 1104]) ...................... 573
   24.4.1 Introduction ........................................... 573
   24.4.2 Details of operation .................................. 573
   24.4.3 Sensor Data ............................................. 575
   24.4.4 Device Feature And Property List .................. 576

24.5 mvBlueCOUGAR-X102n (2 Mpix [1632 x 1248]) ......................... 577
   24.5.1 Introduction ........................................... 577
   24.5.2 Details of operation .................................. 577
   24.5.3 Sensor Data ............................................. 579
   24.5.4 Device Feature And Property List .................. 580

24.6 mvBlueCOUGAR-XD104d (2.4 Mpix [1936 x 1216]) ................. 580
   24.6.1 Introduction ........................................... 580
   24.6.2 Details of operation .................................. 580
   24.6.3 Sensor Data ............................................. 582
Appendix A.4 Polarsens CMOS specific camera / sensor data

26.1 mvBlueCOUGAR-X105p (5.1 Mpix [2464 x 2056])
   26.1.1 Introduction
   26.1.2 Spectral Sensitivity
   26.1.3 Timings
   26.1.4 Device Feature And Property List
   26.1.5 Gray scale version
   26.1.6 Color version

Appendix A.5 CMOS specific camera / sensor data

27.1 mvBlueCOUGAR-Xx00w (0.4 Mpix [752 x 480])
   27.1.1 Introduction
   27.1.2 Details of operation
   27.1.3 Sensor Data
   27.1.4 Device Feature And Property List

27.2 mvBlueCOUGAR-Xx02b (1.2 Mpix [1280 x 960])
   27.2.1 Introduction
   27.2.2 Details of operation
   27.2.3 Sensor Data
   27.2.4 Device Feature And Property List

27.3 mvBlueCOUGAR-Xx02d (1.2 Mpix [1280 x 960])
   27.3.1 Introduction
   27.3.2 Details of operation
   27.3.3 Sensor Data
   27.3.4 Device Feature And Property List

27.4 mvBlueCOUGAR-Xx02e (1.3 Mpix [1280 x 1024])
   27.4.1 Introduction
   27.4.2 Details of operation
   27.4.3 Sensor Data
   27.4.4 Device Feature And Property List

27.5 mvBlueCOUGAR-Xx02eGE (1.3 Mpix [1280 x 1024])
1 About this manual

1.1 Goal of the manual

This manual gives you an overview of the mvBlueCOUGAR-X/-XD, MATRIX VISION’s compact Gigabit Ethernet industrial camera family compliant to GigE Vision, its technical data and basic operation of the mvBlueCOUGAR-X/-XD. Programming the device is detailed in a separate documentation, which will be available in an online format.

The mvBlueCOUGAR-X/-XD manual is based on a modular concept and contains several individual books. That means like in many object-oriented programming languages you have for each functionality your own “class”. Instead of classes, you have books. For example, if you want to know anything about the GUI based applications, then you have to go to the Application Usage (p. 115) book. If you want to know how images are acquired with the mvBlueCOUGAR-X/-XD, have a look in the respective programming language chapter.

1.2 Contents of the manual

At the beginning of the manual, you will get an introduction (p. 34) to the possible usages of the camera. The further chapters contain general information about the camera including:

- how to install the camera for the first time (p. 40) followed by
- Technical data (p. 68)
- Sensor overview (p. 96)
- Filters (p. 113)

The general information is followed by the description of the

- software tools for the camera (p. 115) including the tools
  - wxPropView (p. 115)
  - mvDeviceConfigure (p. 156)
  - mvIPConfigure (p. 152)
  - mvDeviceConfigure (p. 156)
- GenICam and Advanced Features (p. 167) introduces the GenICam and the Advanced features of the camera (the cameras are GenICam compliant devices).

- DirectShow developers (p. 205) documents MATRIX VISION’s DirectShow_acquire interface.

- Troubleshooting (p. 210) shows how to detect damages and other inconveniences.

- Use cases (p. 224) describes solutions for general tasks and

- A Glossary (p. 234) explains abbreviations and technical terms.

- Appendix A.1 CCD specific camera / sensor data (p. 424) contains all data of the Sony CCD sensors like timings, details of operation, etc.

- Appendix A.2 Pregius CMOS specific camera / sensor data (p. 563) contains all data of the Pregius CMOS sensors like timings, details of operation, etc.

- Appendix A.3 Starvis CMOS specific camera / sensor data (p. 621) contains all data of the Starvis CMOS sensors like timings, details of operation, etc.

- Appendix A.4 Polarsens CMOS specific camera / sensor data (p. 625) contains all data of the Polarsens CMOS sensors like timings, details of operation, etc.

- Appendix A.5 CMOS specific camera / sensor data (p. 629) contains all data of the other CMOS sensors like timings, details of operation, etc.
For C, C++, .NET developers, there are separate mvIMPACT Acquire manuals

- "mvIMPACT_Acquire_API_CPP_manual.chm",
- "mvIMPACT_Acquire_API_C_manual.chm", and
- "mvIMPACT_Acquire_API_NET_manual.chm"

available as downloads from our website [https://www.matrix-vision.com](https://www.matrix-vision.com). The manuals contain chapter about

- how to link and build applications using mvIMPACT Acquire,
- how the log output for "mvIMPACT Acquire" devices is configured and how it works in general,
- how to create your own installer packages for Windows and Linux, and
- the general mvIMPACT Acquire API documentation.
2 Migration Guide

2.1 Introduction

Since Firmware 2.27.0 for mvBlueCOUGAR-X(D) devices (released 9th of February 2018)

In general all firmware versions are meant to operate on every device. Whenever this is not the case this page will provide some guidelines on what to do or what to consider. Apart from that all known compatibility issues are listed here.

2.2 Hardware revision 5

Hardware revision 5 for mvBlueCOUGAR-X(D) devices are shipped with a different network interface chip since the chip used before is no longer available. As a consequence devices with this hardware revision cannot be operated with previous firmware releases:

<table>
<thead>
<tr>
<th>Device version / Hardware revision</th>
<th>Supported firmware versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 5.0</td>
<td>All</td>
</tr>
<tr>
<td>5.0</td>
<td>&gt;= 2.27.0 (2.28.0 and greater is recommended)</td>
</tr>
</tbody>
</table>

**Warning**

Downgrading to a firmware version smaller than the versions listed in the table above may result in device malfunction!

To verify the compatibility of your application with new firmware / mvIMPACT Acquire versions you can use cameras with smaller hardware revisions. Just update them to firmware version 2.28.0 or greater. If your applications runs without any problems afterwards you can use the new hardware revisions without any problems as well.

For a detailed migration guide for your application please refer to https://www.matrix-vision.com/manuals/SDK_CPP/MigrationGuide.html
This document assumes a general knowledge of PCs and programming.

Since the documentation is published electronically, an updated version may be available online. For this reason we recommend checking for updates on the MATRIX VISION website.

MATRIX VISION cannot guarantee that the data is free of errors or is accurate and complete and, therefore, assumes no liability for loss or damage of any kind incurred directly or indirectly through the use of the information of this document.

MATRIX VISION reserves the right to change technical data and design and specifications of the described products at any time without notice.

Copyright

MATRIX VISION GmbH. All rights reserved. The text, images and graphical content are protected by copyright and other laws which protect intellectual property. It is not permitted to copy or modify them for trade use or transfer. They may not be used on websites.

- Windows® XP, Windows® Vista, Windows® 7 are trademarks of Microsoft, Corp.
- Linux® is a trademark of Linus Torvalds.
- GenICam™ is a trademark of the GenICam™ standard group.
- GigE Vision™ and the distinctive logo are trademarks owned by the Automated Imaging Association and may only be used under license for compliant products registered with the AIA.

All other product and company names in this document may be the trademarks and tradenames of their respective owners and are hereby acknowledged.

The manual has been generated with Doxygen (Website: http://www.doxygen.org).

Parts of the log file creation and the log file display make use of Sarissa (Website: http://dev.abiss.gr/sarissa) which is distributed under the GNU GPL version 2 or higher, GNU LGPL version 2.1 or higher and Apache Software License 2.0 or higher. The Apache Software License 2.0 is part of this driver package.
4 Legal notice

4.1 Introduction

The firmware running on mvBlueCOUGAR-X, mvBlueCOUGAR-XD and mvBlueFOX3 devices make use of a couple of third party software packages that come with various licenses. This section is meant to list all these packages and to give credit to those whose code helped in the creation of this software.

4.2 cJSON

A slightly modified version of cJSON (http://sourceforge.net/projects/cjson/) is used inside some of the modules that eventually build up the firmware.

4.2.1 cJSON license

Copyright (c) 2009 Dave Gamble

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

4.3 Unity

A slightly modified version of Unity (https://github.com/ThrowTheSwitch/Unity) is used for unit testing various modules that eventually build up the firmware.

4.3.1 Unity license

Copyright (c) 2007–2010 Mike Karlesky, Mark VanderVoord, Greg Williams

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed for the Unity Project, by Mike Karlesky, Mark VanderVoord, and Greg Williams and other contributors", in the same place and form as other third-party acknowledgments. Alternately, this acknowledgment may appear in the software itself, in the same form and location as other such third-party acknowledgments.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
## Revisions

<table>
<thead>
<tr>
<th>Date</th>
<th>Rev.</th>
<th>Description</th>
<th>Driver / Firmware version</th>
</tr>
</thead>
<tbody>
<tr>
<td>10. April 2019</td>
<td>V4.60</td>
<td>Added user output in use case Creating acquisition sequences (Sequencer Control) (p.251).</td>
<td></td>
</tr>
<tr>
<td>05. March 2019</td>
<td>V4.58</td>
<td>Added description for Hi-res model with M42 mount (-5x1x) (p.62).</td>
<td></td>
</tr>
<tr>
<td>7. January 2019</td>
<td>V4.54</td>
<td>Added sensor mvBlueCOUGAR-X102n (2 Mpix [1632 x 1248]) (p.577).</td>
<td></td>
</tr>
<tr>
<td>18. December 2018</td>
<td>V4.53</td>
<td>Added sensor mvBlueCOUGAR-X1020 (20.5 Mpix [5544 x 3692]) (p.621) Update sensor data.</td>
<td></td>
</tr>
<tr>
<td>20. November 2018</td>
<td>V4.53</td>
<td>Added sensor mvBlueCOUGAR-XD1031 (31.5 Mpix [6480 x 4856]) (p.617) Update sensor data.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added mvBlueCOUGAR-X102m (p.573).</td>
<td></td>
</tr>
<tr>
<td>11. April 2018</td>
<td>V4.46</td>
<td>Updated Optimizing the bandwidth (p.315).</td>
<td>Firmware: 2.27.1231.0</td>
</tr>
<tr>
<td>05. April 2018</td>
<td>V4.45</td>
<td>Updated Model without housing (-xx9x) (on demand) (p.68).</td>
<td></td>
</tr>
<tr>
<td>08. March 2018</td>
<td>V4.44</td>
<td>Added Generating very long exposure times (p.264).</td>
<td></td>
</tr>
<tr>
<td>27. February 2018</td>
<td>V4.43</td>
<td>Added Working with sequence paths (p.262).</td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>Version</td>
<td>Description</td>
<td>Page(s)</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>29. January 2018</td>
<td>V4.42</td>
<td>Added <strong>Action Control</strong> (p.[157])</td>
<td></td>
</tr>
<tr>
<td>10. January 2018</td>
<td>V4.41</td>
<td>Updated <strong>Linux</strong> (p.[217])</td>
<td></td>
</tr>
<tr>
<td>13. December 2017</td>
<td>V4.40</td>
<td>Added troubleshooting solution for <em>Why does updating the device list</em> take so long (p.[219])</td>
<td></td>
</tr>
<tr>
<td>06. December 2017</td>
<td>V4.39</td>
<td>Added use case <strong>Working with the dual exposure feature</strong> (<em>&quot;mvMulti→ Zone&quot;</em>) of mvBlueCOUGAR-XD107 (p.[328])</td>
<td></td>
</tr>
<tr>
<td>30. November 2017</td>
<td>V4.38</td>
<td>Extended use case <strong>Processing triggers from an incremental encoder</strong> (p.[336]). Extended <strong>Synchronizing camera timestamps without IEEE 1588</strong> (p.[392]).</td>
<td></td>
</tr>
<tr>
<td>23. October 2017</td>
<td>V4.37</td>
<td>Added mvBlueCOUGAR-XD107 (7.1 Mpix [3216 x 2208]) (p.[502])</td>
<td></td>
</tr>
<tr>
<td>14. August 2017</td>
<td>V4.36</td>
<td>Updated sensors mvBlueCOUGAR-R-Xx02b (1.2 Mpix [1280 x 960]) (p.[633]) and mvBlueCOUGAR-Xx02d (1.2 Mpix [1280 x 960]) (p.[637])</td>
<td></td>
</tr>
<tr>
<td>17. July 2017</td>
<td>V4.35</td>
<td>Added <strong>Working with the serial interface</strong> (mv Serial Interface Control) (p.[397])</td>
<td></td>
</tr>
<tr>
<td>02. June 2017</td>
<td>V4.34</td>
<td>Updated <strong>mviPConfigure</strong> (p.[152])</td>
<td></td>
</tr>
<tr>
<td>29. May 2017</td>
<td>V4.33</td>
<td>Added sensors mvBlueCOUGAR-AR-X100f (0.4 Mpix [728 x 544]) (p.[563]) and mvBlueCOUGAR-X[D]102f (1.6 Mpix [1456 x 1088]) (p.[566])</td>
<td></td>
</tr>
<tr>
<td>15. May 2017</td>
<td>V4.32</td>
<td>Updated use case <strong>Working with burst mode buffer</strong> (p.[270])</td>
<td></td>
</tr>
<tr>
<td>03. May 2017</td>
<td>V4.31</td>
<td>Added use case <strong>Working with binning</strong> (p.[329])</td>
<td></td>
</tr>
<tr>
<td>20. March 2017</td>
<td>V4.30</td>
<td>Updated frame rates of E2V sensors. Added description of the wizard in <strong>Working with multiple AOIs</strong> (mv Multi Area Mode) (p.[266]).</td>
<td></td>
</tr>
<tr>
<td>14. February 2017</td>
<td>V4.29</td>
<td>Added <strong>Working with multiple AOIs</strong> (mv Multi Area Mode) (p.[266]).</td>
<td></td>
</tr>
<tr>
<td>13. February 2017</td>
<td>V4.28</td>
<td>Added <strong>Implementing a hardware-based binarization</strong> (p.[377]).</td>
<td></td>
</tr>
<tr>
<td>24. January 2017</td>
<td>V4.27</td>
<td>Added <strong>Error code list</strong> (p.[221]).</td>
<td></td>
</tr>
<tr>
<td>21. December 2016</td>
<td>V4.26</td>
<td>Updated <strong>Setting up multiple display support and/or work with several capture settings in parallel</strong> (p.[333]).</td>
<td></td>
</tr>
<tr>
<td>15. December 2016</td>
<td>V4.25</td>
<td>Added <strong>Micro-Manger in Driver concept</strong> (p.[40]).</td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>Version</td>
<td>Changes</td>
<td>Firmware</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------</td>
<td>-------------------------------------------------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>01. December 2016</td>
<td>V4.23</td>
<td>Added use case Using the SmartFrameRecall feature</td>
<td>2.15.544.0</td>
</tr>
<tr>
<td>23. August 2017</td>
<td>V4.21</td>
<td>Added mv Defective Pixel Correction Control</td>
<td>2.12.406.0</td>
</tr>
<tr>
<td>01. August 2016</td>
<td>V4.19</td>
<td>Added Triggered frame burst mode</td>
<td></td>
</tr>
<tr>
<td>24. June 2016</td>
<td>V4.18</td>
<td>Updated use case Creating acquisition sequences (Sequencer Control)</td>
<td></td>
</tr>
<tr>
<td>21. June 2016</td>
<td>V4.17</td>
<td>Added use case Triggering of an indefinite sequence with precise starting time</td>
<td></td>
</tr>
<tr>
<td>17. June 2016</td>
<td>V4.16</td>
<td>Added use case Synchronizing camera timestamps without IEEE 1588</td>
<td></td>
</tr>
<tr>
<td>05. April 2016</td>
<td>V4.15</td>
<td>Updated chapter Introducing multicasting</td>
<td></td>
</tr>
<tr>
<td>22. January 2016</td>
<td>V4.13</td>
<td>Added sensor mvBlueCOUGAR→Xx04i (3.2 Mpix [2064 x 1544]) . Added sensor mvBlueCOUGAR→Xx05b (5.1 Mpix [2464 x 2056]) .</td>
<td></td>
</tr>
<tr>
<td>19. January 2016</td>
<td>V4.12</td>
<td>Changed resolution of mvBlueCOUGAR-XD1212a (12.1 Mpix [4248 x 2836]) from 4250 x 2838 pixels to 4248 x 2836 pixels.</td>
<td></td>
</tr>
<tr>
<td>02. December 2015</td>
<td>V4.11</td>
<td>Updated CE declarations</td>
<td></td>
</tr>
<tr>
<td>30. November 2015</td>
<td>V4.10</td>
<td>Added Minimizing sensor pattern of mvBlueCOUGAR-X1010G</td>
<td>2.3.70.0</td>
</tr>
<tr>
<td>27. October 2015</td>
<td>V4.09</td>
<td>Added Command-line options</td>
<td></td>
</tr>
<tr>
<td>22. October 2015</td>
<td>V4.08</td>
<td>Added sensor IMX250: mvBlueCOUGAR-XD105a (5.1 Mpix [2464 x 2056]) .</td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>Version</td>
<td>Revision Details</td>
<td></td>
</tr>
<tr>
<td>-----------------</td>
<td>---------</td>
<td>-----------------</td>
<td></td>
</tr>
<tr>
<td>25. September 2015</td>
<td>V4.07</td>
<td>Added <strong>mvBlueCOUGAR-3X</strong> (p.59)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updated &quot;About this manual&quot;.</td>
<td></td>
</tr>
<tr>
<td>07. September 2015</td>
<td>V4.06</td>
<td>Changed sensor size of <strong>10 Mpix</strong>es resolution CMOS sensor (-x010) (p.101) from 1/2.35&quot; to 1/2.3&quot;.</td>
<td></td>
</tr>
<tr>
<td>011. August 2015</td>
<td>V4.05</td>
<td>Added color of <strong>KS-BCXD-HR12</strong> in <strong>Circular connector female</strong> (p.83).</td>
<td></td>
</tr>
<tr>
<td>04. August 2015</td>
<td>V4.04</td>
<td>Added Windows 10 support.</td>
<td></td>
</tr>
<tr>
<td>09. July 2015</td>
<td>V4.03</td>
<td>Added chapter I cannot focus on distant objects (p.218).</td>
<td></td>
</tr>
<tr>
<td>25. June 2015</td>
<td>V4.02</td>
<td>Added note in about using <strong>mvBlueCOUGAR-XD</strong> with two or one <strong>LAN (RJ 45)</strong> connections (p.59).</td>
<td></td>
</tr>
<tr>
<td>19. June 2015</td>
<td>V4.00</td>
<td>Restructured chapter <strong>Use cases</strong> (p.244).</td>
<td></td>
</tr>
<tr>
<td>08. June 2015</td>
<td>V3.54</td>
<td>Updated dimensional drawings in <strong>Technical data</strong> (p.58).</td>
<td></td>
</tr>
<tr>
<td>21. May 2015</td>
<td>V3.53</td>
<td>Updated pixel clock of <strong>mvBlueCOUGAR-XD104d</strong> (2.4 Mpix [1936 x 1216]) (p.580) and <strong>mvBlueCOUGAR-Xx04f</strong> (2.4 Mpix [1936 x 1216]) (p.584).</td>
<td></td>
</tr>
<tr>
<td>17. April 2015</td>
<td>V3.52</td>
<td>Adapted the sensor resolution of <strong>mvBlueCOUGAR-XD104d</strong> (2.4 Mpix [1936 x 1216]) (p.580) and <strong>mvBlueCOUGAR-Xx04f</strong> (2.4 Mpix [1936 x 1216]) (p.584).</td>
<td></td>
</tr>
<tr>
<td>16. April 2015</td>
<td>V3.51</td>
<td>Updated supported Windows versions. Extended the list of sensors which supports the <strong>Creating acquisition sequences (Sequencer Control)</strong> (p.251).</td>
<td></td>
</tr>
<tr>
<td>31. March 2015</td>
<td>V3.49</td>
<td>Added line scan frame rate calculator for the sensors <strong>mvBlueCOUGAR-Xx02e</strong> (1.3 Mpix [1280 x 1024]) (p.641), <strong>mvBlueCOUGAR-Xx02eGE</strong> (1.3 Mpix [1280 x 1024]) (p.647), and <strong>mvBlueCOUGAR-Xx04e</strong> (1.9 Mpix [1600 x 1200]) (p.655).</td>
<td></td>
</tr>
<tr>
<td>25. March 2015</td>
<td>V3.48</td>
<td>Upgraded frame rate calculator of <strong>mvBlueCOUGAR-XD104d</strong> (2.4 Mpix [1936 x 1216]) (p.580) and <strong>mvBlueCOUGAR-Xx04f</strong> (2.4 Mpix [1936 x 1216]) (p.584).</td>
<td></td>
</tr>
<tr>
<td>05. March 2015</td>
<td>V3.46</td>
<td>Added sensor <strong>mvBlueCOUGAR-Xx04f</strong> (2.4 Mpix [1936 x 1216]) (p.584).</td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>Version</td>
<td>Changes</td>
<td></td>
</tr>
<tr>
<td>-----------------</td>
<td>---------</td>
<td>---------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>27. February 2015</td>
<td>V3.44</td>
<td>Updated pixel clock for sensors mvBlueCOUGAR-XD129a (9.2 Mpix [3384 x 2712]) (p. 543) and mvBlueCOUGAR-XD126a (6 Mpix [2752 x 2208]) (p. 523).</td>
<td></td>
</tr>
<tr>
<td>26. February 2015</td>
<td>V3.43</td>
<td>Added note in I cannot use mvBlueCOUGAR-XD (p. 211) and I cannot initialize 4 Tap sensor mvBlueCOUGAR-XDs (p. 212). Moved Optimizing the bandwidth (p. 315) to Use Cases.</td>
<td></td>
</tr>
<tr>
<td>09. February 2015</td>
<td>V3.42</td>
<td>Updated pixel clock of mvBlueCOUGAR-XD104d (2.4 Mpix [1936 x 1216]) (p. 580).</td>
<td></td>
</tr>
<tr>
<td>02. February 2015</td>
<td>V3.41</td>
<td>Added pinning of KS-BCX-HR12 in Circular connector male (p. 69). Driver: 2.11.9; Firmware: 1.6.← 295.0.</td>
<td></td>
</tr>
<tr>
<td>07. January 2015</td>
<td>V3.39</td>
<td>Extended the description of <strong>Disabling the heartbeat</strong> (p. 389).</td>
<td></td>
</tr>
<tr>
<td>08. December 2014</td>
<td>V3.38</td>
<td>Added use case <strong>Adjusting sensor -x02e (-1013) / -x04e (-1020)</strong> (p. 369). Added further lens in <strong>Using motorized lenses with mvBlueCOUGAR-XD</strong> (p. 357).</td>
<td></td>
</tr>
<tr>
<td>02. December 2014</td>
<td>V3.36</td>
<td>Added <strong>Quick Setup Wizard</strong> (p. 116). Driver: 2.11.3; Firmware: 1.6.← 230.0.</td>
<td></td>
</tr>
<tr>
<td>21. October 2014</td>
<td>V3.34</td>
<td>Added description about the record mode in <strong>How to see the first image</strong> (p. 120).</td>
<td></td>
</tr>
<tr>
<td>25. September 2014</td>
<td>V3.32</td>
<td>Added some power consumption data for mvBlueCOUGAR-XD sensors in <strong>Standard model (-xx1x)</strong> (p. 68).</td>
<td></td>
</tr>
<tr>
<td>10. September 2014</td>
<td>V3.31</td>
<td>Updated <strong>Standard model (-xx1x)</strong> (p. 68).</td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>Version</td>
<td>Changes</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>---------</td>
<td>--------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>30. July 2014</td>
<td>V3.29</td>
<td>Added sensor -XD1212. Added sensor mvBlueCOUGAR-XD1212a (12.1 Mpix [4248 x 2836]) (p. 553)</td>
<td></td>
</tr>
<tr>
<td>17. July 2014</td>
<td>V3.28</td>
<td>Added use case Introducing LUTs (p. 371)</td>
<td></td>
</tr>
<tr>
<td>05. June 2014</td>
<td>V3.27</td>
<td>Update chapter Using the linescan mode (p. 230)</td>
<td></td>
</tr>
<tr>
<td>15. May 2014</td>
<td>V3.26</td>
<td>Updated data of mvBlueCOUGAR-AR-XD specific CCD sensors (p. 98) and added -XD126a (p. 523), -X-D129a (p. 543), and &quot;-XD1212&quot;.</td>
<td></td>
</tr>
<tr>
<td>25. March 2014</td>
<td>V3.25</td>
<td>Added use case Correcting image errors of a sensor (p. 286)</td>
<td></td>
</tr>
<tr>
<td>17. Mar. 2014</td>
<td>V3.24</td>
<td>Updated length of mvBlueCOUGAR-AR-XD in Standard model (-xx1x) (p. 81)</td>
<td></td>
</tr>
<tr>
<td>10. Mar. 2014</td>
<td>V3.23</td>
<td>mvDeviceConfigure (p. 156) extended</td>
<td></td>
</tr>
<tr>
<td>18. Feb. 2014</td>
<td>V3.21</td>
<td>Updated Characteristics (p. 640) of mvBlueCOUGAR-Xx02d (1.2 Mpix [1280 x 960]) (p. 637)</td>
<td></td>
</tr>
<tr>
<td>17 Feb. 2014</td>
<td>V3.20</td>
<td>Added use case Detecting overtriggering (p. 346)</td>
<td></td>
</tr>
<tr>
<td>10 Feb. 2014</td>
<td>V3.18</td>
<td>Updated length of mvBlueCOUGAR-XD Standard model (-xx1x) (p. 81).</td>
<td></td>
</tr>
<tr>
<td>30 Jan. 2014</td>
<td>V3.17</td>
<td>Added note to Video iris (p. 88) section. Added 2 troubleshooting sections: I cannot use mvBlueCOUGAR-XD (p. 211) and I cannot initialize 4 Tap sensor mvBlueCOUGAR-XDs (p. 212).</td>
<td></td>
</tr>
<tr>
<td>22 Jan. 2014</td>
<td>V3.15</td>
<td>New sensors mvBlueCOUGAR-X[D]x04 (2.2 Mpix [2048 x 1088]) (p. 581) and mvBlueCOUGAR-X[D]x04b (4.2 Mpix [2048 x 2048]) (p. 665) for mvBlueCOUGAR-X.</td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>Version</td>
<td>Changes</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
<td>-------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>20 Jan. 2014</td>
<td>V3.14</td>
<td>Added use cases Creating acquisition sequences (Sequencer Control) (p.251) and Working with LUTValueAll (p.375) (which was originally a part of LUT Control (p.182).</td>
<td></td>
</tr>
<tr>
<td>20 Dec. 2013</td>
<td>V3.11</td>
<td>Added Sample 2: Triggered line scan acquisition with a specified number of image blocks and pausing trigger signals (p.283) scan_Sample.</td>
<td></td>
</tr>
<tr>
<td>17 Dec. 2013</td>
<td>V3.10</td>
<td>Updated Summary of advanced features (p.54) and Summary of components (p.82).</td>
<td></td>
</tr>
<tr>
<td>06. December 2013</td>
<td>V3.07</td>
<td>Added information about Changing the view of the property grid to assist writing code that shall locate driver features (p.141).</td>
<td></td>
</tr>
<tr>
<td>22. November 2013</td>
<td>V3.06</td>
<td>Extended information in Adjusting sensor -x00w (p.353) and Adjusting sensor -x02d (-1012d) (p.366).</td>
<td></td>
</tr>
<tr>
<td>06. November 2013</td>
<td>V3.04</td>
<td>Unified power supply specification.</td>
<td></td>
</tr>
<tr>
<td>30. October 2013</td>
<td>V3.03</td>
<td>Added I get an exception when setting offsetX or offsetY (p.211)</td>
<td></td>
</tr>
<tr>
<td>22. October 2013</td>
<td>V3.02</td>
<td>Added declaration of conformity (p.27) of mvBlueCOUGAR-XD.</td>
<td></td>
</tr>
<tr>
<td>17. October 2013</td>
<td>V3.01</td>
<td>mvBlueCOUGAR-X without housing (p.69) is an order option (-xx9x).</td>
<td></td>
</tr>
<tr>
<td>15. October 2013</td>
<td>V3.00</td>
<td>Added Webcasts (p.23) links. Added chapter Bit-shifting an image (p.140).</td>
<td></td>
</tr>
<tr>
<td>14. October 2013</td>
<td>V2.22</td>
<td>“User Set Default Selector” is deprecated and will be replaced by “User Set Default”.</td>
<td></td>
</tr>
<tr>
<td>02. October 2013</td>
<td>V2.21</td>
<td>Added max current consumption of the digital inputs (p.70).</td>
<td></td>
</tr>
<tr>
<td>27. September 2013</td>
<td>V2.20</td>
<td>Added pinning of Circular connector male (POE-I Rev. 2.00) (p.77). Added pinning of M12 x-coded 100/1000 MBit Ethernet (option POE-I Rev. 2.00) (p.80). Updated Figure 10.2: DigIn mvBlueCOUGAR-X (PLC option) (p.70).</td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>Version</td>
<td>Changes</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
<td>-------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>23. September 2013</td>
<td>V2.19</td>
<td>Updated resolution of -XD126 and -XD129 (p.98).</td>
<td></td>
</tr>
<tr>
<td>17. September 2013</td>
<td>V2.18</td>
<td>Added note about the galvanic isolation of the Ethernet signals (p.75).</td>
<td></td>
</tr>
<tr>
<td>12. September 2013</td>
<td>V2.17</td>
<td>Added CMOSIS variants -204 and -204b in mvBlueCOUGAR-XD specific CMOS sensors (p.107). These models support 30 MHz pixel clock and 12 bit ADC on the basis of a firmware variant.</td>
<td></td>
</tr>
<tr>
<td>03. September 2013</td>
<td>V2.16</td>
<td>Enhanced sensor listing of Order code nomenclature (p.35). Added troubleshooting section I can see the camera in mvIPConfigure, but I cannot access the camera (p.210).</td>
<td></td>
</tr>
<tr>
<td>07. August 2013</td>
<td>V2.15</td>
<td>Changed resolution of sensor mvBlueCOUGAR-Xx22 (1.2 Mpix [1280 x 960]) from 1290x964 to 1290x960. (p.455).</td>
<td></td>
</tr>
<tr>
<td>19. July 2013</td>
<td>V2.14</td>
<td>Added sensor mvBlueCOUGAR-Xx04e (1.9 Mpix [1600 x 1200]) (p.655).</td>
<td></td>
</tr>
<tr>
<td>17. July 2013</td>
<td>V2.13</td>
<td>Added recommended network interface cards in Hardware (p.43).</td>
<td></td>
</tr>
<tr>
<td>11. July 2013</td>
<td>V2.12</td>
<td>Added Additional hardware requirements for mvBlueCOUGAR-XD (p.43).</td>
<td></td>
</tr>
<tr>
<td>25. June 2013</td>
<td>V2.11</td>
<td>Added more links to line scan mode (p.280).</td>
<td></td>
</tr>
<tr>
<td>20. June 2013</td>
<td>V2.10</td>
<td>Added Additional temperature and power consumption notes concerning mvBlueCOUGAR-XD and mvBlueCOUGAR-Xx2x (p.27).</td>
<td></td>
</tr>
<tr>
<td>04. June 2013</td>
<td>V2.08</td>
<td>Added power consumption in Sensor overview (p.98).</td>
<td></td>
</tr>
<tr>
<td>03. June 2013</td>
<td>V2.07</td>
<td>Added use case Using VLC Media Player (p.277).</td>
<td></td>
</tr>
<tr>
<td>13. May 2013</td>
<td>V2.06</td>
<td>Changed mvBlueCOUGAR-Xx10 to mvBlueCOUGAR-Xx010.</td>
<td></td>
</tr>
<tr>
<td>26. February 2013</td>
<td>V2.05</td>
<td>Added chapter Relationship between driver, firmware, FPGA file and user settings (p.64).</td>
<td></td>
</tr>
<tr>
<td>14. February 2013</td>
<td>V2.04</td>
<td>Corrected overlapping capability of mvBlueCOUGAR-Xx02e (1.3 Mpix [1280 x 1024]) and mvBlueCOUGAR-Xx02eGE (1.3 Mpix [1280 x 1024]) (p.647).</td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>Version</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---------------------</td>
<td>---------</td>
<td>---------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>28. January 2013</td>
<td>V2.03</td>
<td>Added description Working with the UserFile section (Flash memory) (p. 384).</td>
<td></td>
</tr>
<tr>
<td>24. January 2013</td>
<td>V2.02</td>
<td>Added information about image error counts and disabling CPU sleep states: Setting up the camera (p. 61). How to disable CPU sleep states a.k.a. C states (&lt; Windows 8) (p. 163), and There are image error counts (p. 210).</td>
<td></td>
</tr>
<tr>
<td>19. December 2012</td>
<td>V2.01</td>
<td>Added 2 sensors: mvBlueCOUGAR-Xx02eGE (1.3 Mpix [1280 x 1024]) (p. 547) and mvBlueCOUGAR-Xx010 (10 Mpix [3856 x 2764]) (p. 574).</td>
<td></td>
</tr>
<tr>
<td>14. December 2012</td>
<td>V2.00</td>
<td>New version of technical documentation.</td>
<td></td>
</tr>
<tr>
<td>07. December 2012</td>
<td>V1.27</td>
<td>All parts of the manual to do with programming are available as a separate manual now:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &quot;mvIMPACT_Acquire_API←_CPP_manual.chm&quot;,</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &quot;mvIMPACT_Acquire_API←_C_manual.chm&quot;, and</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &quot;mvIMPACT_Acquire_API←_NET_manual.chm&quot;. These manuals can be download from <a href="http://www.matrix-vision.com">http://www.matrix-vision.com</a>.</td>
<td></td>
</tr>
<tr>
<td>12. November 2012</td>
<td>V1.25</td>
<td>Added mvBlueCOUGAR-XD sensor mvBlueCOUGAR-XD129 (9.2 Mpix [3384 x 2712]) (p. 533).</td>
<td></td>
</tr>
<tr>
<td>30. October 2012</td>
<td>V1.24</td>
<td>Updated Characteristics of the digital inputs (p. 70).</td>
<td></td>
</tr>
<tr>
<td>22. October 2012</td>
<td>V1.23</td>
<td>Updated Legal notice (p. 82).</td>
<td></td>
</tr>
<tr>
<td>25. September 2012</td>
<td>V1.21</td>
<td>Added mvBlueCOUGAR-XD sensors mvBlueCOUGAR-XD124a (3 Mpix [1936 x 1460]) (p. 484) and mvBlueCOUGAR-XD126 (6 Mpix [2752 x 2208]) (p. 573).</td>
<td></td>
</tr>
<tr>
<td>20. September 2012</td>
<td>V1.20</td>
<td>Added chapter “Porting existing code written with versions earlier then 2.0.0 (C++)”.</td>
<td></td>
</tr>
<tr>
<td>06. September 2012</td>
<td>V1.19</td>
<td>Clarified view of connectors in Technical data (p. 83).</td>
<td></td>
</tr>
</tbody>
</table>
Added mvBlueCOUGAR-XD sensor mvBlueCOUGAR-X[D]x04b (4.2 Mpix [2048 x 2048]) (p. 58).

17. August 2012 V1.17 Added use case Adjusting sensor-x02d (-1012d) (p. 366).

18. July 2012 V1.16 Added use case Creating a de-bouncing filter at the inputs (p. 355).

12. July 2012 V1.15 Added use case Creating user set entries (p. 381).


06. July 2012 V1.13 Added description about the second (female) connector of mvBlueCOUGAR-XD (p. 83).

05. July 2012 V1.12 Changed manual structure.

02. July 2012 V1.11 Added uses cases Using the primary application switchover functionality (p. 419) and Disabling the heartbeat (p. 389).

25. June 2012 V1.10 Corrected frame rate calculator of mvBlueCOUGAR-Xx02e (1.3 Mpix [1280 x 1024]) (p. 241).
Completed spectral characteristics of (-x02b) (p. 634), (-x02e) (p. 642), (-x05) (p. 670).

19. June 2012 V1.9 Updated trigger modes summary of all CCD sensors (p. 96) and (-x00w) (p. 830), (-x02b) (p. 634), (-x02d) (p. 638), (-x02e) (p. 642), (-x05) (p. 670).

12. June 2012 V1.8 Updated digital input characteristics in chapter Circular connector male (p. 69).

29. May 2012 V1.7 Updated chapter Introducing multicasting (p. 400).

25. May 2012 V1.6 Added chapter Setting a flicker-free auto expose and auto gain (p. 317).

07. May 2012 V1.5 Added information about -PLC and -POE options in Order code nomenclature (p. 35), Technical data (p. 58) and Summary of components (p. 92).
<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>20. April 2012</td>
<td>V1.4</td>
<td>Added chapter <strong>Optimizing the color fidelity of the camera</strong> (p. 293)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Extended chapter <strong>Order code nomenclature</strong> (p. 35)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added chapter <strong>Cold mirror filter</strong> (p. 113)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added new sensor <strong>mvBlueCO</strong> → UGAR-X[D]x04 (2.2 Mpix [2048 x 1088]) (p. 35)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Changed length of <strong>mvBlueCO</strong> → GAR-XD (p. 81)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added description about the line scan mode of sensor <strong>mvBlueCO</strong> → OUGAR-Xx02e (1.3 Mpix [1280 x 1024]) (p. 341).</td>
</tr>
<tr>
<td>02. April 2012</td>
<td>V1.3</td>
<td>Enhanced chapter <strong>Output sequence of color sensors (RGB Bayer)</strong> (p. 110) and added chapter <strong>Bilinear interpolation of color sensors (RGB Bayer)</strong> (p. 110).</td>
</tr>
<tr>
<td>13. March 2012</td>
<td>V1.2</td>
<td>Removed note about inverse-polarity protection, because the <strong>digital inputs</strong> (p. 88) have inverse-polarity protection in the meantime.</td>
</tr>
<tr>
<td>12. March 2012</td>
<td>V1.1</td>
<td>Updated image of <strong>Circular connector male</strong> (p. 89)</td>
</tr>
<tr>
<td>07. March 2012</td>
<td>V1.0</td>
<td>Updated Linux installation section</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added use case <strong>Generating a pulse width modulation (PWM)</strong> (p. 339)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added use case <strong>Outputting a pulse at every other external trigger</strong> (p. 341)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added use case <strong>Reset timestamp by hardware</strong> (p. 391)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added chapter <strong>Troubleshooting</strong> (p. 210)</td>
</tr>
<tr>
<td>Date</td>
<td>Version</td>
<td>Changes</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| 06. March 2012   | V1.0b33 | Added chapter **GenICam and Advanced Features** (p. 167)  
Updated description: Creating synchronized acquisitions using timers (p. 414)  
Added use case: Acquiring a number of images (p. 248)  
Added use case: Controlling strobe or flash at the outputs (p. 353)  
Added use case: Working with burst mode buffer (p. 270)  
Added use case: Processing triggers from an incremental encoder (p. 336)  
Added use case: Working with the temperature sensors (p. 387)  
Added use case: Creating different exposure times for consecutive images (p. 343)  
Added note about the short circuit protection of the digital outputs in Circular connector male (p. 69) |
| 23. February 2012| V1.0b32 | Updated data of sensors mvBlueCOUGAR-Xx05 (5 Mpix [2592 x 1944]) (p. 669) |
| 17. February 2012| V1.0b31 | Renewed chapter **wxPropView** (p. 115) |
| 09. January 2012 | V1.0b30 | Added mvBlueCOUGAR-XD in chapter **Technical data** (p. 58) and **Introduction** (p. 34) |
| 06. December 2011| V1.0b29 | Added block diagrams of image data flow in chapter **Sensor overview** (p. 96) |
| 09. November 2011| V1.0b28 | Added chapter about **Settings behaviour during startup** (p. 62) in chapter **Quickstart** (p. 40) |
| 11. October 2011 | V1.0b27 | Added **GenICam** (p. 234) specific trigger setting names in sensor descriptions (p. 56)  
Added **Recording sequences with pre-trigger** (p. 249)  
Added **Reducing noise by frame averaging** (p. 313) |
| 05. October 2011 | V1.0b26 | Added accessory: mvBC-X I/O-BOX (p. 38)  
Added digital input and output delay: Circular connector male (p. 69) |
<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>22. September 2011</td>
<td>V1.0b25</td>
<td>Added sensor: mvBlueCOUGA-R-Xx02b (1.2 Mpix [1280 x 960]) (p. 633)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added sensor: mvBlueCOUGA-R-Xx02d (1.2 Mpix [1280 x 960]) (p. 637)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added sensor: mvBlueCOUGA-R-Xx02e (1.3 Mpix [1280 x 1024]) (p. 641)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added sensor: mvBlueCOUGAR-Xx05 (5 Mpix [2592 x 1944]) (p. 669)</td>
</tr>
<tr>
<td>26. July 2011</td>
<td>V1.0b24</td>
<td>Removed chapter “EventHandling (C++ developers). See Porting existing code written with versions earlier then 2.0.0 (--&gt; C++)”</td>
</tr>
<tr>
<td>11. July 2011</td>
<td>V1.0b23</td>
<td>Added chapter “Callback demo”</td>
</tr>
<tr>
<td>06. June 2011</td>
<td>V1.0b22</td>
<td>Added chapter “Porting existing code written with versions earlier then 2.0.0”</td>
</tr>
<tr>
<td>21. April 2011</td>
<td>V1.0b21</td>
<td>Added example circuits for the high-side switch: Circular connector male (p. 69)</td>
</tr>
<tr>
<td>14. April 2011</td>
<td>V1.0b20</td>
<td>Added sensor: mvBlueCOUGA-R-Xx20d (0.5 Mpix [776 x 580]) (p. 444)</td>
</tr>
<tr>
<td>24. March 2011</td>
<td>V1.0b19</td>
<td>Added chapter: Adjusting sensor - x00w (p. 363)</td>
</tr>
<tr>
<td>09. March 2011</td>
<td>V1.0b18</td>
<td>Update sensor area: Sensor overview (p. 96)</td>
</tr>
<tr>
<td>18. January 2011</td>
<td>V1.0b17</td>
<td>Added chapter “Porting applications written with the Generic interface layout to use the GenICam interface layout”</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Setting up multiple display support and/or work with several capture settings in parallel (p. 133)</td>
</tr>
<tr>
<td>29. November 2010</td>
<td>V1.0b16</td>
<td>Added ADC resolutions in Sensor overview (p. 96)</td>
</tr>
<tr>
<td>25. November 2010</td>
<td>V1.0b15</td>
<td>Added information about Summary of advanced features (p. 84)</td>
</tr>
<tr>
<td>22. November 2010</td>
<td>V1.0b14</td>
<td>Added chapter C: Notes about mvIMPACT Acquire version 1.11.32 or higher and/or using mvBlueCOUGAR-X with firmware 1.2.0 or higher</td>
</tr>
<tr>
<td>18. November 2010</td>
<td>V1.0b13</td>
<td>Added chapter C++: Notes about mvIMPACT Acquire version 1.11.32 or higher and/or using mvBlueCOUGAR-X with firmware 1.2.0 or higher</td>
</tr>
<tr>
<td>19. October 2010</td>
<td>V1.0b12</td>
<td>Added chapter “Chunk data format”</td>
</tr>
<tr>
<td>14. Sep. 2010</td>
<td>V1.0b11</td>
<td>Added chapter Creating synchronized acquisitions using timers (p. 414)</td>
</tr>
<tr>
<td>Date</td>
<td>Version</td>
<td>Changes</td>
</tr>
<tr>
<td>------------</td>
<td>---------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| 03. Sep. 2010 | V1.0b10 | Added IPv4 converter in **Introducing multicasting** *(p. 400)*  
Corrected integration time of **Wide-VGA resolution CMOS sensor** *(p. 101)* |
| 26. Aug. 2010 | V1.0b9  | Added chapter about **Creating user data entries** *(p. 379)*          |
| 04. Aug. 2010 | V1.0b8  | Added chapter **Import and Export images** *(p. 132)*  
Updated **sensor data** *(p. 95)*  
Added **Summary of advanced features** *(p. 94)* |
| 30. Jun. 2010 | V1.0b7  | Added chapter **GenICam to mvIMPACT Acquire code generator** *(p. 244)* |
| 21. Jun. 2010 | V1.0b5  | Included exposure modes in the frame rate calculator of the **Sensor overview** *(p. 96)*  
Added detailed information to **infoTimeStamp_us** of the "Request Class" |
| 15. Jun. 2010 | V1.0b4  | Added note about **Digital Input** *(p. 68)*  
and updated the Digital Input figure |
| 28. May 2010  | V1.0b3  | Added new example **SingleCaptureMasterSlave**                              |
| 27. Apr. 2010 | V1.0b2  | Added chapter **Saving user settings in the non-volatile flash memory** *(p. 148)*, which shows how to work with user sets |
| 27. Apr. 2010 | V1.0b1  | Added chapter "**Building, linking and running applications using mvIMPACT Acquire**, which shows how to run an application that has been built using the mvIMPACT Acquire framework |
| 19. Apr. 2010 | V1.0b   | Added new example **ContinuousCaptureDirectX**                              |
| 15. Apr. 2010 | V1.0b   | Updated mvBlueCOUGAR-X technical documentation                               |
| 24. Mar. 2010 |         | Updated **Sensor overview** *(p. 96)*  
(Frame rate calculator of -x20a, -x20b, -x22, -x23, -x00w) |
| 15. Mar. 2010 |         | Updated **What’s inside and accessories** *(p. 98)*  
Added dimensional drawing of **tripod adapter** *(p. 69)* |
| 28. Jan. 2010 |         | Added chapter **Copy grid data to the clipboard** *(p. 131)*             |
| 27. Jan. 2010 |         | Added chapter **Introducing multicasting** *(p. 400)*                      |

MATRIX VISION GmbH
<table>
<thead>
<tr>
<th>Date</th>
<th>Updates</th>
</tr>
</thead>
<tbody>
<tr>
<td>25. Jan. 2010</td>
<td>Updated digital in figure of <strong>Hirose 12-pin male connector</strong> (p.69)</td>
</tr>
<tr>
<td></td>
<td>Updated description of <strong>Hot mirror filter</strong> (p.113)</td>
</tr>
<tr>
<td>20. Jan. 2010</td>
<td>Updated pinning and digital out figure of <strong>Hirose 12-pin male connector</strong> (p.69)</td>
</tr>
<tr>
<td>12. Jan. 2010</td>
<td>Due to a software update, documentation of <strong>CMOS sensor (-x00w)</strong> (p.629) updated</td>
</tr>
<tr>
<td>November 2009</td>
<td>Initial version</td>
</tr>
</tbody>
</table>
6 Graphic Symbols

6.1 Notes, Warnings, Attentions

**Note**

A **note** indicates important information that helps you optimize usage of the products.

**Warning**

A **warning** indicates how to avoid either potential damage to hardware or loss of data.

**Attention**

An **attention** indicates a potential for property damage, personal injury, or death.

All due care and attention has been taken in preparing this manual. In view of our policy of continuous product improvement, however, we can accept no liability for completeness and correctness of the information contained in this manual. We make every effort to provide you with a flawless product.

In the context of the applicable statutory regulations, we shall accept no liability for direct damage, indirect damage or third-party damage resulting from the acquisition or operation of a MATRIX VISION product. Our liability for intent and gross negligence is unaffected. In any case, the extend of our liability shall be limited to the purchase price.

6.2 Webcasts

<table>
<thead>
<tr>
<th>![Webcast Icon]</th>
</tr>
</thead>
<tbody>
<tr>
<td>This icon indicates a webcast about an issue which is available on our website.</td>
</tr>
</tbody>
</table>
7 Important information

7.1 Important safety instructions

• We cannot and do not take any responsibility for the damage caused to you or to any other equipment connected to the mvBlueCOUGAR-X. Similarly, warranty will be void, if a damage is caused by not following the manual.

• Handle the mvBlueCOUGAR-X with care. Do not misuse the mvBlueCOUGAR-X. Avoid shaking, striking, etc. The mvBlueCOUGAR-X could be damaged by faulty handling or shortage.

• Do not use accessories not recommended by the product manufacturer as they may cause hazards.

• The product should be situated away from heat sources such as radiators, heat registers, stoves, or other products (including amplifiers) that produce heat.

• Using the board-level version:
  – Provide sufficient cooling because single components can reach high temperatures.
  – Handle with care and avoid damage of electrical components by electrostatic discharge (ESD):
    • Discharge body static (contact a grounded surface and maintain contact).
    • Avoid all plastic, vinyl, and Styrofoam (except anti-static versions) around printed circuit boards.
    • Do not touch components on the printed circuit board with your hands or with conductive devices.

7.2 Operating considerations

7.2.1 Important safety notes

• Use this camera with a 12V to 24V DC power supply (+- 1V).

• Using the Model without housing (-xx9x) (on demand) (p.66):
  – Provide sufficient cooling because single components can reach high temperatures. Inadequate and incorrect cooling invalidate the guarantee.

• For extended temperature (-ET) models:
  
  Warning

  – In general, the housing temperature of the ET models must not exceed 65°C!

• Observe that flammable objects, water or metal do not enter the camera interior. These may lead to failure or accident.

• Do not modify the camera or use the camera with external covers removed. These may cause failure, void any warranties and pose a safety hazard.

• Stop using the camera at the approach of electrical storm (thunder audible). Protect the camera from rain if using it outdoors.

• In the event of abnormal functioning, switch off the camera and disconnect the power cord. Contact MATRIX VISION.
7.2 Operating considerations

7.2.2 Handling and cleaning

- Do not attempt to disassemble camera.
- When installing or removing a lens, take care that water or dust does not enter the inside of the camera.

7.2.3 Installing

Avoid installing or storing the camera in the following environments:

- Environments exposed to direct sunlight, rain or snow.
- Environments where combustible or corrosive gas exists.
- Excessively warm or cold environment (Operating ambient temperature: 0 to 45 °C)
- Humid or dusty environment.
- Place subjected to excessive vibration or shock.
- Environment exposed to strong electric or magnetic field.
- It is recommended to mount the camera on a thermoconducting surface such as aluminum or other metals rather than plastic or wood.
- Please contact manufacturer or local distributor if you want to use additional enclosures for higher ingress protection.
- Do not aim the camera lens at the sun or other very strong light sources.

“CCD phenomenon: Smearing”: Vertical smearing may appear, in the image. However, this is not due to failure of the camera.

7.2.4 Optimizing performance and life time

When the camera is used continuously for long time under high ambient temperature, the inside electrical parts may deteriorate, resulting in shorter life span. Additional cooling by e.g. air convection is recommended.

7.2.5 Connectors

Confirm the power is off before connecting or disconnecting a signal cable. Grasp connectors by the body, not the attached wires.

7.2.6 Cleaning

- Use a blower or a lens brush to remove dust on the lens or the optical filter.
- Do not disassemble front flange.
- Clean case with dry soft cloth. Use neutral detergent liquid if needed; wipe the cover with dry cloth.
- Do not use benzene, thinner, alcohol, liquid cleaner or spray-type cleaner.
- If dust or other debris is located between the CCD and optical filter, consult dealer for cleaning by an optical technician.
7.2.7 Adjusting the C-mount

The mvBlueCOUGAR-X cameras allow a precise adjustment of the back focus of the C-mount by means of a back focus ring which is threaded into the C-mount and is secured by a lock nut ring which itself is secured by two screws. The mechanical adjustment of the imaging device is important in order to achieve a perfect alignment with the focal point of the lens. This adjustment is made before leaving the factory to conform to the standard of 17.526 mm (in air) and should normally not require adjustment in the field. However, if the back focal plane of your lens does not conform to the C-mount back focus specification or if you have e.g. removed the IR-CUT filter (p.113), renewed adjustment may be required.

How to proceed:

- Loosen screws (location as shown above by arrows) of the lock nut ring with an Allen key (0.9 x 50).
- Loosen the lock nut ring.
- With the lens set to infinity or a known focus distance, set the camera to view an object located at "infinity" or the known distance.
- Rotate the C-mount ring and lens forward or backwards on its thread until the object is in sharp focus.

Note

Be careful that the lens remains seated in the C-mount.

- Once focus is achieved, tighten the lock nut ring, then tighten the two locking screws of the lock ring without applying excessive torque.
7.2.8 Additional temperature and power consumption notes concerning mvBlueCOUGAR-XD and mvBlueCOUGAR-X2xx

mvBlueCOUGAR-XD and mvBlueCOUGAR-X2xx cameras have very different power consumptions depending on sensor technology and connection and operation modes. As a rule of thumb:

- CMOS sensor camera with single network interface controller (NIC) output consumes less around 4W
- CCD equipped cameras with quad tap use and dual NIC output and max. pixel clock may consume up to 9W
- CCD sensor dual tap mode consumes 1.7W less than quad tap
- For mvBlueCOUGAR-XD: Dual NIC mode (so-called link aggregation [LAG] or bonding (p. 237)) consumes 0.8W more than single NIC used

Data sheets of cameras commonly specify the allowed temperature range by indicating the min/max. ambient temperature of the air (e.g. in a room) where the camera is installed and operated. Whereas this is good for a first indication and comparison it presents a fuzziness for the temperature of the components inside because it is not defined how the camera is mounted and thus the overtemperature inside the camera is not clear.

And since each installation is different, MATRIX VISION cannot supply a strictly required procedure for proper heat dissipation. Instead, we provide the following general guidelines:

In all cases,

1. you should monitor the temperature of the camera by using the two built-in temperature sensors (separate for sensor board and mainboard) and
2. make sure that neither the sensor temperature nor the mainboard temperature exceed 75 °C.

Keep in mind that the camera will gradually heat up during the first hour of operation. After one hour, the housing temperature should have stabilized and increases no longer.

- If your camera is mounted on a substantial metal holder or component in your system, this may already provide sufficient heat dissipation because this metal component presents a low thermal resistance for the heat path.
- You may use additional heat conductors between the camera and the holder to bridge gaps.
- The use of a fan to provide air flow over the camera is an extremely efficient method of heat dissipation.

The fan can be driven directly by one of the robust camera outputs and controlled by the camera itself (temp. settings to switch fan on and off with hysteresis) as long as

- the switched voltage at the output is suitable for the used fan (12V up to 24V) and that
- the fan does not consume more than 700 mA which is the maximum one output of the camera can deliver.

Available CPU Socket 7 (370 or socket A) active cooler fit very well directly on any side of the camera by means of thermal glue. Ask for support if you need further assistance.

7.3 European Union Declaration of Conformity statement
The mvBlueCOUGAR-X is in conformity with all applicable essential requirements necessary for CE marking. It corresponds to the EU EMC guideline 2014/30/EU based on the following harmonized standards Electromagnetic compatibility (EMC):

- Interference emission EN 61000-6-3 / 2007
- Interference emission EN 61000-6-4 / 2007 for mvBlueCOUGAR-X PoE (p. 242) version
- Interference immunity EN 61000-6-2 / 2005

EN 61000-6-3 / 2007 requires an I/O cable with a retrofittable ferrite to be used such as:
- Company: Würth Elektronik Type: WE No. 742 711 31

MATRIX VISION corresponds to the EU guideline WEEE 2002/96/EG on waste electrical and electronic equipment and is registered under WEEE-Reg.-No. DE 25244305.

<table>
<thead>
<tr>
<th>RoHS</th>
<th>All units delivered are RoHS compliant.</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP30</td>
<td>mvBlueCOUGAR-X / mvBlueCOUGAR-XD / mvBlueCOUGAR-X -POE (p. 35)</td>
</tr>
<tr>
<td>IP50</td>
<td>mvBlueCOUGAR-X -POE-I (p. 35)</td>
</tr>
<tr>
<td>IP67</td>
<td>mvBlueCOUGAR-X -POE-IP67 (p. 35)</td>
</tr>
<tr>
<td>PoE 802.3at (p. 242)</td>
<td>The option -POE (p. 35) is compliant to IEEE 802.3af (IEEE 802.3at Type 1).</td>
</tr>
</tbody>
</table>
EG-Konformitätserklärung

Declaration of conformity

Der Hersteller
The Manufacturer
Matrix Vision GmbH
Talstraße 16
71570 Oppenweiler
Germany

erklärt hiermit, dass sein Produkt
herewith declares, that his product

Typbezeichnung: mvBlueCOUGAR-X
Type: mvBlueCOUGAR-X

mit den Bestimmungen folgender Europäischer Richtlinien übereinstimmt:
compiles with the provisions of the following European Directives:

2014/30/EU (EMV-Richtlinie) / 2014/35/EU (EMC Directive)
auf Grundlage folgender harmonisierter Normen:

based on the following harmonized standards:

Elektromagnetische Verträglichkeit (EMV) / Electromagnetic compatibility (EMC)
Störaussendung / Interference emission

Störfestigkeit / Interference immunity
EN 61000-6-2 / 2005

Oppenweiler, 24.11.2015

i.V. Uwe Hagmaier
Leiter Entwicklung / Head of Development

Bitte beachten: Um die Einhaltung obiger Normen sicherzustellen, ist die Verwendung
hochwertiger, vollständig geschirmer Anschlusskabel unbedingt erforderlich.
Please note: In order to fulfill the above standards, the use of high-quality, shielded connecting
cables is required.
EG-Konformitätserklärung

Declaration of conformity

Der Hersteller  
The Manufacturer  
Matrix Vision GmbH  
Talstraße 16  
71570 Oppenweiler  
Germany

erklärt hiermit, dass sein Produkt  
herewith declares, that his product

Typbezeichnung:  
Type:  
mvBlueCOUGAR-X-POE  
mvBlueCOUGAR-X-POE

mit den Bestimmungen folgender Europäischer Richtlinien übereinstimmt:  
complies with the provisions of the following European Directives:

2014/30/EU (EMV-Richtlinie) / 2014/35/EU (EMC Directive)

auf Grundlage folgender harmonisierter Normen:  
based on the following harmonized standards:

Elektromagnetische Verträglichkeit (EMV) / Electromagnetic compatibility (EMC)  
Störaussendung / interference emission  

Störfestigkeit / interference immunity  
EN 61000-6-2 / 2005

Oppenweiler, 24.11.2015

[Signature]

i.V. Uwe Hagmaier  
Leiter Entwicklung / Head of Development

Bitte beachten: Um die Einhaltung obiger Normen sicherzustellen, ist die Verwendung hochwertiger, vollständig geschirrter Anschlusskabel unbedingt erforderlich.  
Please note: In order to fulfill the above standards, the use of high-quality, shielded connecting cables is required.
EG-Konformitätserklärung

Declaration of conformity

Der Hersteller

The Manufacturer

Matrix Vision GmbH

Talstraße 16

71570 Oppenweiler

Germany

erklärt hiermit, dass sein Produkt
herewith declares, that his product

Typbezeichnung: mvBlueCOUGAR-X-POE-I/IP

Type: mvBlueCOUGAR-X-POE-I/IP

mit den Bestimmungen folgender Europäischer Richtlinien übereinstimmt:
complies with the provisions of the following European Directives:

2014/30/EU (EMV-Richtlinie) / 2014/35/EU (EMC Directive)
auf Grundlage folgender harmonisierter Normen:
based on the following harmonized standards:

Elektromagnetische Verträglichkeit (EMV) / Electromagnetic compatibility (EMC)
Störaussendung / Interference emission


Störfestigkeit / Interference immunity

EN 61000-6-2 / 2005

Oppenweiler, 24.11.2015

i.V. Uwe Hagmayer

Leiter Entwicklung / Head of Development

Bitte beachten: Um die Einhaltung obiger Normen sicherzustellen, ist die Verwendung
hochwertiger, vollständig geschirrter Anschlusskabel unbedingt erforderlich.
Please note: In order to fulfill the above standards, the use of high-quality, shielded connecting cables is required.
EG-Konformitätserklärung

Declaration of conformity

Der Hersteller
The Manufacturer
Matrix Vision GmbH
Talstraße 16
71570 Oppenweiler
Germany

erklärt hiermit, dass sein Produkt
herewith declares, that his product

Typbezeichnung: mvBlueCOUGAR-XD
Type: mvBlueCOUGAR-XD

mit den Bestimmungen folgender Europäischer Richtlinien übereinstimmt:
complies with the provisions of the following European Directives:

2014/30/EU (EMV-Richtlinie) / 2014/35/EU (EMC Directive)
auf Grundlage folgender harmonisierter Normen:
based on the following harmonized standards:
Elektromagnetische Verträglichkeit (EMV) / Electromagnetic compatibility (EMC)
Störaussendung / Interference emission
Störfestigkeit / Interference immunity
EN 61000-6-2 / 2005

Oppenweiler, 24.11.2015

i.V. Uwe Hagmaier
Leiter Entwicklung / Head of Development

Bitte beachten: Um die Einhaltung obiger Normen sicherzustellen, ist die Verwendung
hochwertiger, vollständig geschirmter Anschlusskabel unbedingt erforderlich.
Please note: In order to fulfill the above standards, the use of high-quality, shielded connecting
cables is required.

7.4 Legal notice

7.4.1 For customers in the U.S.A.
This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a residential environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. However there is no guarantee that interferences will not occur in a particular installation. If the equipment does cause harmful interference to radio or television reception, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the distance between the equipment and the receiver.
- Use a different line outlet for the receiver.
- Consult a radio or TV technician for help.

You are cautioned that any changes or modifications not expressly approved in this manual could void your authority to operate this equipment. The shielded interface cable recommended in this manual must be used with this equipment in order to comply with the limits for a computing device pursuant to Subpart B of Part 15 of FCC Rules.

- FCC Class B requires an I/O cable with an retrofittable ferrite to be used such as
  - Company: Würth Elektronik Type: WE No. 742 711 31

7.4.2 For customers in Canada

This apparatus complies with the Class B limits for radio noise emissions set out in the Radio Interference Regulations.

7.4.3 Pour utilisateurs au Canada

Cet appareil est conforme aux normes classe B pour bruits radioélectriques, spécifiées dans le Règlement sur le brouillage radioélectrique.
8 Introduction

The mvBlueCOUGAR is an extreme compact and flexible industrial Gigabit Ethernet camera family compliant to GigE Vision (p. 236) and GenICam (p. 234). The mvBlueCOUGAR offers

• a wide range of sensors,
• high frame rates over long distances using inexpensive standard network parts,
• I/Os suitable for industrial applications, and
• wide range of resolutions.

The image memory of the camera enables a high-speed-buffer mode (a.k.a. burst mode) which writes images faster in the camera's memory as they are transferred. With this mode, image losses are a thing of the past.

The mvBlueCOUGAR is ideally suited for all classical areas of machine vision and allows an easy integration in the environment and support the application with helpful features to deliver faster and better results. The hardware capabilities can be teamed with machine vision libraries which are compliant to GigE Vision.

The Dual-GigE version named mvBlueCOUGAR-XD is also available.

8.1 Software concept

The mvBlueCOUGAR is a GigE Vision (p. 236) compliant device, using a GenICam (p. 234) XML-file describing the device capabilities. Within this XML-file it uses the names, units, data types etc. recommended by the SFNC (p. 242) to describe the devices features. Whenever custom features have been added to the mvBlueCOUGAR firmware these can clearly be spotted by the leading ‘mv’ in the features name. The device can communicate with every third party compliant GigE Vision (p. 236) capture driver that can interpret GenICam (p. 234) XML-files.

The following figure shows the software concept of MATRIX VISION's camera devices:
8.2 Order code nomenclature

8.2.1 mvBlueCOUGAR-X

The mvBlueCOUGAR-X nomenclature scheme is as follows:

mvBlueCOUGAR-X - A B C - D - (1) (2) (3) (4) E

- A: Sensor model
  120a: 0.3 Mpix, 640 x 480, 1/3", CCD

As shown in figure 2, for the mvBlueCOUGAR the mvIMPACT_Acquire (p. 240) interface is stacked on the GigE Vision (p. 236) and Genicam (p. 234) layers. The mvIMPACT_Acquire (p. 240) interface internally uses the GenICam (p. 234) runtime libs, so that it can be considered as an user application written with the GenICam (p. 234) interface.

Figure 2: Software concept of mvBlueFOX3
<table>
<thead>
<tr>
<th>Camera Model</th>
<th>Resolution</th>
<th>Pixel Size</th>
<th>Sensor Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>120b: 0.3 Mpix</td>
<td>640 x 480</td>
<td>1/3&quot;</td>
<td>CCD</td>
</tr>
<tr>
<td>100f: 0.4 Mpix</td>
<td>728 x 544</td>
<td>1/2.9&quot;</td>
<td>CMOS</td>
</tr>
<tr>
<td>100w: 0.4 Mpix</td>
<td>752 x 480</td>
<td>1/3&quot;</td>
<td>CMOS</td>
</tr>
<tr>
<td>120d: 0.5 Mpix</td>
<td>776 x 580</td>
<td>1/2&quot;</td>
<td>CCD</td>
</tr>
<tr>
<td>102b: 1.2 Mpix</td>
<td>1280 x 960</td>
<td>1/3&quot;</td>
<td>CMOS</td>
</tr>
<tr>
<td>102d: 1.2 Mpix</td>
<td>1280 x 960</td>
<td>1/3&quot;</td>
<td>CMOS</td>
</tr>
<tr>
<td>122: 1.2 Mpix</td>
<td>1280 x 960</td>
<td>1/3&quot;</td>
<td>CCD</td>
</tr>
<tr>
<td>102e: 1.3 Mpix</td>
<td>1280 x 1024</td>
<td>1/1.8&quot;</td>
<td>CMOS</td>
</tr>
<tr>
<td>102f: 1.6 Mpix</td>
<td>1456 x 1088</td>
<td>1/2.9&quot;</td>
<td>CMOS</td>
</tr>
<tr>
<td>102m: 1.8 Mpix</td>
<td>1600 x 1104</td>
<td>1/1.8&quot;</td>
<td>CMOS</td>
</tr>
<tr>
<td>102n: 2.0 Mpix</td>
<td>1632 x 1248</td>
<td>1/1.2&quot;</td>
<td>CMOS</td>
</tr>
<tr>
<td>123: 1.4 Mpix</td>
<td>1360 x 1024</td>
<td>1/2&quot;</td>
<td>CCD</td>
</tr>
<tr>
<td>124: 1.9 Mpix</td>
<td>1600 x 1200</td>
<td>1/1.8&quot;</td>
<td>CCD</td>
</tr>
<tr>
<td>104e: 1.9 Mpix</td>
<td>1600 x 1200</td>
<td>1/1.8&quot;</td>
<td>CMOS</td>
</tr>
<tr>
<td>104: 2.2 Mpix</td>
<td>2048 x 1088</td>
<td>2/3&quot;</td>
<td>CMOS</td>
</tr>
<tr>
<td>104f: 2.4 Mpix</td>
<td>1936 x 1214</td>
<td>1/1.2&quot;</td>
<td>CMOS</td>
</tr>
<tr>
<td>104i: 3.2 Mpix</td>
<td>2048 x 1544</td>
<td>1/1.8&quot;</td>
<td>CMOS</td>
</tr>
<tr>
<td>104b: 4.2 Mpix</td>
<td>2048 x 2048</td>
<td>1&quot;</td>
<td>CMOS</td>
</tr>
<tr>
<td>104bUV: 4.2 Mpix</td>
<td>2048 x 2048</td>
<td>1&quot;</td>
<td>CMOS</td>
</tr>
<tr>
<td>105: 5.0 Mpix</td>
<td>2592 x 1944</td>
<td>1/2.5&quot;</td>
<td>CMOS</td>
</tr>
<tr>
<td>105b: 5.1 Mpix</td>
<td>2464 x 2056</td>
<td>2/3&quot;</td>
<td>CMOS</td>
</tr>
<tr>
<td>105p: 5.1 Mpix</td>
<td>2464 x 2056</td>
<td>2/3&quot;</td>
<td>CMOS</td>
</tr>
<tr>
<td>125a: 5.1 Mpix</td>
<td>2448 x 2050</td>
<td>2/3&quot;</td>
<td>CCD</td>
</tr>
<tr>
<td>125: 5.1 Mpix</td>
<td>2448 x 2050</td>
<td>2/3&quot;</td>
<td>CCD</td>
</tr>
<tr>
<td>107b: 7.1 Mpix</td>
<td>3216 x 2208</td>
<td>1.1&quot;</td>
<td>CMOS</td>
</tr>
<tr>
<td>109b: 8.9 Mpix</td>
<td>4112 x 2176</td>
<td>1&quot;</td>
<td>CMOS</td>
</tr>
<tr>
<td>1010: 10.7 Mpix</td>
<td>3856 x 2764</td>
<td>1/2.35&quot;</td>
<td>CMOS</td>
</tr>
<tr>
<td>1012b: 12.4 Mpix</td>
<td>4112 x 3008</td>
<td>1.1&quot;</td>
<td>CMOS</td>
</tr>
<tr>
<td>1020: 20.5 Mpix</td>
<td>5544 x 3692</td>
<td>1.1&quot;</td>
<td>CMOS</td>
</tr>
</tbody>
</table>

- **B**: Sensor color
- **G**: Gray scale version
- **C**: Color version
- **C**: Infrared enhanced (for -102eG)
- **E**: Infrared enhanced

- **D**: Options
  - POE: Power over Ethernet (only for -1xx models), PoE class 2
  - PLC: Inputs for PLC levels only
  - POE-IP67: Power over Ethernet (only for -1xx models), PoE class 2 with a modular M12 industrial connector and IP67 protection class

- **(1)**: Lensholder
  1: C-mount (standard)
  2: CS-mount
  3: C-mount without backfocus adjustment
  4: CS-mount without backfocus adjustment
  5: CS-mount with free option of rotation of object

- **(2)**: Filter
  1: IR-CUT (standard)
  2: Glass
  3: Daylight cut (only without backfocus adjustment)
  4: eIR-CUT (enhanced)
  5: UV transmissive
  9: None

- **(3)**: Case
  1: Color blue (standard)
  2: Color black, no logo, no label MATRIX VISION
  3: Color blue, no logo, no label MATRIX VISION
  4: IP67 long version (+10mm)
  9: None

- **(4)**: Misc
  1: None (standard)

- **E**: Extended Temperature
  ET: With extended temperature behavior
## 8.2.2 mvBlueCOUGAR-XD

The mvBlueCOUGAR-XD nomenclature scheme is as follows:

```
  mvBlueCOUGAR-XD - A B - (1) (2) (3) (4) E

  - A: Sensor model
    102f: 1.6 Mpix, 1456 x 1088, 1/2.9", CMOS
    104: 2.2 Mpix, 2048 x 1088, 2/3", CMOS
    204: 2.2 Mpix, 2048 x 1088, 2/3", CMOS
    104a12: 2.2 Mpix, 2048 x 1088, 2/3", CMOS
    104d: 2.4 Mpix, 1936 x 1214, 1/1.2", CMOS
    124a: 2.8 Mpix, 1936 x 1460, 2/3", CCD
    104h: 3.2 Mpix, 2064 x 1544, 1/1.8", CMOS
    104b: 4.2 Mpix, 2048 x 2048, 1", CMOS
    204b: 4.2 Mpix, 2048 x 2048, 1", CMOS
    105a: 5.1 Mpix, 2464 x 2056, 2/3", CMOS
    126: 6.1 Mpix, 2752 x 2208, 1", CCD
    126a: 6.1 Mpix, 2752 x 2208, 1", CCD
    107: 7.1 Mpix, 3216 x 2208, 1.1", CMOS
    107b: 7.1 Mpix, 3216 x 2208, 1.1", CMOS
    109b: 8.9 Mpix, 4112 x 2176, 1", CMOS
    1010: 10.7 Mpix, 3856 x 2764, 1/2.35", CMOS
    129: 9.2 Mpix, 3384 x 2712, 1", CCD
    129a: 9.2 Mpix, 3384 x 2712, 1", CCD
    1212: 12.1 Mpix, 4250 x 2838, 1", CCD
    1212a: 12.1 Mpix, 4250 x 2838, 1", CCD
    1012b: 12.4 Mpix, 4112 x 3008, 1.1", CMOS
    1031: 31.5 Mpix, 6480 x 4856, APS-C, CMOS

  - B: Sensor color
    G: Gray scale version
    C: Color version

  - (1): Lensholder
    1: C-mount (standard)
    2: CS-mount
    3: C-mount without backfocus adjustment
    4: CS-mount without backfocus adjustment
    5: M42 mount

  - (2): Filter
    1: IR-CUT (standard)
    2: Glass
    3: Daylight cut (only without backfocus adjustment)
    9: None

  - (3): Case
    1: Color blue (standard)
    2: Color black, no logo, no label MATRIX VISION
    3: Color blue, no logo, no label MATRIX VISION
    9: None

  - (4): Misc
    1: None (standard)
```

## 8.2.3 mvBlueCOUGAR-3X

The mvBlueCOUGAR-3X nomenclature scheme is as follows:

```
  mvBlueCOUGAR-3X - A B - D - (1) (2) (3) (4)

  - A: Sensor model
    120a: 0.3 Mpix, 640 x 480, 1/3", CCD
    123: 1.4 Mpix, 1360 x 1024, 1/2", CCD

  - B: Sensor color
```
8.2.4 Ordering code samples

<table>
<thead>
<tr>
<th>mvBlueCOUGAR-X120aC¹</th>
<th>640 x 480, CCD 1/3&quot;, color, Gigabit Ethernet, IR-CUT filter, C-mount, etc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>mvBlueCOUGAR-X122G-POE-1211</td>
<td>1280 x 964, CCD 1/3&quot;, gray scale, Gigabit Ethernet with Power over Ethernet, C-mount, glass filter, etc.</td>
</tr>
<tr>
<td>mvBlueCOUGAR-XD129C-2111</td>
<td>3384 x 2712, CCD 1&quot;, color, Dual-Gigabit Ethernet, CS-mount, IR-CUT filter, etc.</td>
</tr>
</tbody>
</table>

¹: -1111 is the standard delivery variant and for this reason it is not mentioned.

8.3 What's inside and accessories

Due to the varying fields of application the mvBlueCOUGAR is shipped without accessories. The package contents:

• mvBlueCOUGAR

For the first use of the mvBlueCOUGAR we recommend the following accessories to get the camera up and running:

• network cable and
• power supply via
  – MV-DC1201 BCX.

**Note**

To be able use the mvBlueCOUGAR-XD it is required that the GigE ports (the two GigE interfaces at the host; either PC or a switch) support the LAG (Link Aggregation Group) functionality. Windows 7 and Linux support this functionality by default. Although the mvBlueCOUGAR-XD is connected via two cables it will act like a single device.

Accessories for the mvBlueCOUGAR:
<table>
<thead>
<tr>
<th>Part code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MV-Tripod Adapter</td>
<td>Tripod adapter for mvBlueCOUGAR-X (p. 69)</td>
</tr>
<tr>
<td>MV-DC1201 BCS</td>
<td>Power supply 12V for mvBlueCOUGAR-X</td>
</tr>
<tr>
<td>MV-DC1201 BCSX IO REV2</td>
<td>Power supply 12V for mvBlueCOUGAR-X with digital I/O on separate cable length Power 3m, I/O: 0.5m</td>
</tr>
<tr>
<td>KS-RJ45-M12-8X</td>
<td>Connecting cable RJ45/M12 8 pin X-coded, CAT6a, length up to 20m</td>
</tr>
<tr>
<td>KS-GIGE LS SLK x.0</td>
<td>Gigabit Ethernet cable RJ45 to RJ45 with horizontal screw lock 20mm, M2x0.4, drag chain capable, CAT6, length up to 20m</td>
</tr>
<tr>
<td>KS-GIGE VLS SLK x.0</td>
<td>Gigabit Ethernet cable RJ45 to RJ45 with vertical screw lock 20mm for mvBlueCOUGAR-XD, M2x0.4, drag chain capable, CAT6, length up to 20m</td>
</tr>
<tr>
<td>KS-GIGE LS SLK AK90L x.0</td>
<td>Gigabit Ethernet cable RJ45 to RJ45 with horizontal screw lock 20mm, M2x0.4, drag chain capable, CAT6 Camera plug 90° angled left direction, length up to 20m</td>
</tr>
<tr>
<td>KS-GIGE LS SLK AK90R x.0</td>
<td>Gigabit Ethernet cable RJ45 to RJ45 with horizontal screw lock 20mm, M2x0.4, drag chain capable, CAT6 Camera plug 90° angled right direction, length up to 20m</td>
</tr>
<tr>
<td>KS-BCX-HR12</td>
<td>Hirose I/O cable up to 20m</td>
</tr>
<tr>
<td>KS-BCXD-HR12-MZ 00.50</td>
<td>Hirose I/O cable adapter for motor zoom (0.5m)</td>
</tr>
<tr>
<td>KS-M12-PWR-IO</td>
<td>Shielded connection, M12 male connector to prepared wires, UL/CSA, drag chain, length up to 20m</td>
</tr>
<tr>
<td>MV-X I/O-BOX</td>
<td>IO-Box to connect and test a mvBlueCOUGAR-X camera, to loop the camera into an existing system, and to mix both connection variants selectively</td>
</tr>
<tr>
<td>LAN KARTE 1000 PCIE REV2</td>
<td>PCI Express extension card with Gigabit Ethernet interface</td>
</tr>
<tr>
<td>LAN KARTE 1000 QUAD PCIE REV2</td>
<td>PCI Express extension card with 4 Gigabit Ethernet interfaces</td>
</tr>
</tbody>
</table>


9 Quickstart

9.1 Driver concept

The driver supplied with the MATRIX VISION product represents the port between the programmer and the hardware. The driver concept of MATRIX VISION provides a standardized programming interface to all image processing products made by MATRIX VISION GmbH.

The advantage of this concept for the programmer is that a developed application runs without the need for any major modifications to the various image processing products made by MATRIX VISION GmbH. You can also incorporate new driver versions, which are available for download free of charge on our website: http://www.matrix-vision.com.

The following diagram shows a schematic structure of the driver concept:

![Diagram of Driver Concept]

Figure 1: Driver concept
9.1 Driver concept

- 1 Part of any mvIMPACT Acquire driver installation package (Windows).
- 2 Separately available for 32 bit and 64 bit. Requires at least one installed driver package.
- 3 See 2, but requires an installed version of the mvBlueFOX driver.
- 4 Part of the NeuroCheck installer but requires at least one installed frame grabber driver.
- 5 Part of the mvIMPACT SDK installation. However, new designs should use the .NET libs that are now part of mvIMPACT Acquire ("mv.impact.acquire.dll"). The namespace "mv.impact.acquire" of "mv.impact.acquire.dll" provides a more natural and more efficient access to the same features as contained in the namespace "mvIMPACT_NET.acquire" of "mvIMPACT_NET.dll", which is why the latter one should only be used for backward compatibility but NOT when developing a new application.
- 6 Part of Micro-Manager.

9.1.1 NeuroCheck support

A couple of devices are supported by NeuroCheck. However between NeuroCheck 5.x and NeuroCheck 6.x there has been a breaking change in the internal interfaces. Therefore also the list of supported devices differs from one version to another and some additional libraries might be required.

For NeuroCheck 5.x the following devices are supported:

<table>
<thead>
<tr>
<th>Device</th>
<th>Additional software needed</th>
</tr>
</thead>
<tbody>
<tr>
<td>mvTITAN-G1</td>
<td>mvSDK driver for mvTITAN/mvGAMMA devices</td>
</tr>
<tr>
<td>mvTITAN-CL</td>
<td>mvSDK driver for mvTITAN/mvGAMMA devices</td>
</tr>
<tr>
<td>mvGAMMA-CL</td>
<td>mvSDK driver for mvTITAN/mvGAMMA devices</td>
</tr>
<tr>
<td>mvBlueFOX</td>
<td>mvIMPACT Acquire driver for mvBlueFOX devices, &quot;NCUSBmvBF.dll&quot;</td>
</tr>
</tbody>
</table>

For NeuroCheck 6.0 the following devices are supported:

<table>
<thead>
<tr>
<th>Device</th>
<th>Additional software needed</th>
</tr>
</thead>
<tbody>
<tr>
<td>mvTITAN-G1</td>
<td>mvIMPACT Acquire driver for mvTITAN/mvGAMMA devices</td>
</tr>
<tr>
<td>mvTITAN-CL</td>
<td>mvIMPACT Acquire driver for mvTITAN/mvGAMMA devices</td>
</tr>
<tr>
<td>mvGAMMA-CL</td>
<td>mvIMPACT Acquire driver for mvTITAN/mvGAMMA devices</td>
</tr>
<tr>
<td>mvHYPERION-CLb</td>
<td>mvIMPACT Acquire driver for mvHYPERION devices</td>
</tr>
</tbody>
</table>
| Every other mvIMPACT Acquire compliant device | mvIMPACT Acquire driver for the corresponding device family, "mv.impact.acquire.NeuroCheck6.←dll" (comes with the driver package, but the driver package must be installed AFTER installing NeuroCheck 6)

For NeuroCheck 6.1 the following devices are supported:

<table>
<thead>
<tr>
<th>Device</th>
<th>Additional software needed</th>
</tr>
</thead>
<tbody>
<tr>
<td>mvTITAN-G1</td>
<td>mvIMPACT Acquire driver for mvTITAN/mvGAMMA devices</td>
</tr>
<tr>
<td>mvTITAN-CL</td>
<td>mvIMPACT Acquire driver for mvTITAN/mvGAMMA devices</td>
</tr>
<tr>
<td>mvGAMMA-CL</td>
<td>mvIMPACT Acquire driver for mvTITAN/mvGAMMA devices</td>
</tr>
<tr>
<td>mvHYPERION-CLb</td>
<td>mvIMPACT Acquire driver for mvHYPERION devices</td>
</tr>
</tbody>
</table>
Every other mvIMPACT Acquire compliant device

<table>
<thead>
<tr>
<th>mvIMPACT Acquire driver for the corresponding device family, &quot;mv.impact.acquire.NeuroCheck6_etc1.dll&quot; comes with the driver package, but the driver package must be installed AFTER installing NeuroCheck 6.1</th>
</tr>
</thead>
</table>

9.1.2 VisionPro support

Every *mvIMPACT Acquire* driver package under Windows comes with an adapter to VisionPro from Cognex. The installation order does not matter. After the driver package and VisionPro has been installed, the next time VisionPro is started it will allow selecting the *mvIMPACT Acquire* device. No additional steps are needed.

MATRIX VISION devices that also comply with the GigE Vision or USB3 Vision standard don't need any software at all, but can also use VisionPro's built-in GigE Vision or USB3 Vision support.

9.1.3 HALCON support

HALCON comes with built-in support for *mvIMPACT Acquire* compliant devices, so once a device driver has been installed for the *mvIMPACT Acquire* device, it can also be operated from a HALCON environment using the corresponding acquisition interface. No additional steps are needed.

MATRIX VISION devices that also comply with the GigE Vision standard don't need any software at all, but can also use HALCON's built-in GigE Vision support.

As some *mvIMPACT Acquire* device driver packages also come with a GenTL compliant interface, these can also be operated through HALCON's built-in GenTL acquisition interface.

9.1.4 LabVIEW support

Every *mvIMPACT Acquire* compliant device can be operated under LabVIEW through an additional set of VIs which is shipped by MATRIX VISION as a separate installation ("mvLabVIEW Acquire").

MATRIX VISION devices that also comply with the GigE Vision or USB3 Vision standard don't need any additional software at all, but can also be operated through LabVIEW's GigE Vision or USB3 Vision driver packages.

9.1.5 DirectShow support

Every *mvIMPACT Acquire* compliant device driver package comes with an interface to DirectShow. In order to be usable from a DirectShow compliant application, devices must first be registered for DirectShow support. How to this is explained [here](p. 208).

9.1.6 Micro-Manager support

Every *mvIMPACT Acquire* compliant device can be operated under [https://micro-manager.org](https://micro-manager.org) when using mvIMPACT Acquire 2.18.0 or later and at least Micro-Manager 1.4.23 build **AFTER** 15.12.2016. The adapter needed is part of the Micro-Manager release. Additional information can be found [here](https://micro-manager.org/wiki/MatrixVision).
9.2 Windows

9.2.1 System requirements

Currently supported Windows versions are:

- Microsoft Windows 7 (32-bit, 64-bit)
- Microsoft Windows 8.1 (32-bit, 64-bit)
- Microsoft Windows 10 (32-bit, 64-bit)

Other Windows version can be used at the user's own risk.

Note

Since mvIMPACT Acquire version 2.8.0 it could be possible that you have to update your Windows installer at least using Windows XP. The necessary packages are available from Microsoft's website:


All necessary drivers are available from the MATRIX VISION website at www.matrix-vision.de, section "Products -> Cameras -> your interface -> your product -> Downloads".

9.2.1.1 Hardware

MATRIX VISION successfully tested several network interface cards (p. 240), which are summarized in the following document:


9.2.1.2 Additional hardware requirements for mvBlueCOUGAR-XD

The mvBlueCOUGAR-XD is a high-performance camera containing latest top-notch technology which requires

- a high data throughput on the host system,

for example, when processing large amount of image data and

- high CPU resources if processing color images on the host PC.

This means that you will need

- the latest PC architectures, preferably multi core Intel CPUs and chipsets
- sufficient RAM (4 GB in 32-bit OS; 8 GB in 64-bit OS), and
- appropriate and recommended network cards with PCI-E x1 or x4 interface and with the latest "Link Aggregate Group" (LAG) capable drivers.

Please ask your system vendor for further advice and consult our technical documentation.

Beside the host system requirements, you have to fine-tune

- the network settings,
- the camera settings, and
- the host system settings (e.g. disabling the sleep mode)

to achieve the specified frame rates possible from the camera's perspective.
9.2.2 Installing the mvGenTL-Acquire driver

By double clicking on

- `mvGenTL_Acquire-x86-n.n.n.msi` (for 32-bit systems) or

- `mvGenTL_Acquire-x86_64-n.n.n.msi` (for 64-bit systems), the installer will start automatically:

![Driver installation - start window](image)

Figure 2: Driver installation - start window

- Now, follow the instructions of installation program and use default settings:
Since

mvIMPACT Acquire 2.25.0

wxPropView (p.115) is able to check if there are new driver version available on a weekly basis. Deactivate the check box if wxPropView should not check for updates. You can activate the checking in wxPropView (p.115) via the help menu.
After confirmation, the installation will start and copy the data. At the end, MATRIX VISION's USB3 Vision (p. 242) / GigE Vision (p. 236) capture filter driver will be installed. A "Windows Security" dialog will appear.

- Click on **Install** (This however requires the system to be rebooted, thus the installer will ask the user to reboot the system).

After the system has been restarted, the filter driver installation will start automatically. If necessary, click "Ok" and ignore the compatibility testing of Windows by clicking on "Continue anyway" twice.
Now, the driver installation will be finished.

You will find all tools like

- wxPropView (p. [115]) and
- mvDeviceConfigure (p. [156])

either on desktop as single icons or in Windows menu under "MATRIX VISION -> mvIMPACT Acquire". Afterwards, you can use mvDeviceConfigure (p. [156]) to update the firmware. The latest firmware image is available on the web - please check for updates. The current firmware version can be read out using wxPropView (p. [115]) (device section on top).

9.2.3 Optimizing the network configuration

This chapter explains, how to achieve optimal performance by setting up the NIC (p. [240]) (network interface card) and the devices network controller in the best possible way with respect to the limitations imposed by the surrounding network. To check, if the camera is available via the network, you can use the tool mvIPConfigure (p. [152]).

Attention

Only network experts should change settings in this section.

The camera sets the MTU (p. [240]) to the maximum value automatically given by the NIC (p. [240]) or switch and supports a maximum MTU (p. [240]) of 16K. You can change the MTU (p. [240]) of the camera manually under "Setting -> Base -> Camera -> GenICam -> Transport Layer Control -> Gev Stream Channel Selector -> Gev SCPS Packet Size".

In the system settings of the camera, you can set up the overall network behaviour of the device in the "Transport Layer Control" section e.g. by assigning a persistent IP address, a persistent default gateway and a persistent subnet mask in "Transport Layer Control" section according to your needs. However IP configuration related settings should be done by using the tool mvIPConfigure (p. [152]) as this will provide a more convenient access to these parameters.

Network performance related parameters however can be controlled from the "Driver -> GenTL" section as well:
By default 1500 is used as MTU\footnote{[240]} ("Driver -> GenTL -> Interfaces -> Interface0 -> mv Gev Interface MTU"). You can change this according to your network. As a general rule of thumb it can be said, that the higher the MTU\footnote{[240]}, the better the overall performance, as less network packets are needed to transmit a full image, which results in less overhead that arises from the handling of each arriving network packet in the device driver etc. However every component involved in the data transmission (that includes every switch, router and other network component, that is installed in between the device and the system receiving the packets) must support the MTU\footnote{[240]} or packet size, as otherwise the first component not supporting the packet size will silently discard all packets, that are larger than the component can handle.

\begin{itemize}
  \item Note
  \begin{itemize}
  \item If you modify the interface MTU\footnote{[240]} of the device, it will take up to 5 seconds until these settings are saved in the non-volatile memory. During this time do not remove the power supply! Afterwards you have to reboot the device for the changed settings to take effect.
  \end{itemize}
\end{itemize}

9.2.4 Adjusting the network controller

If you have interplay problems between mvBlueCOUGAR and a network controller (NIC\footnote{[240]}) please check following settings in the driver settings:

\begin{itemize}
  \item Note
  \begin{itemize}
  \item There is no need to set the transfer packet size manually. Whenever the device is initialised, the driver will determine the maximum possible packet size for the current network settings automatically.
  \item The screenshots below are examples for how a dialog for a feature described here might look like. Not every NIC/NIC driver will support every feature discussed here and different drivers might use different names for the features presented. The hints are just meant to give an idea for what to look for.
  \end{itemize}
\end{itemize}

1. Is the NIC\footnote{[240]} a "1000MBit Full Duplex Controller" and is this mode activated (via properties of network settings)?

2. Is the driver of the NIC\footnote{[240]} up-to-date?
   Some manufacturers have optimized their drivers referring to higher data throughput.

3. Is the "GigE Vision Capture driver" installed as described in Install GigE Vision capture filter\footnote{[149]}?

4. Does the MTU\footnote{[240]} of the NIC\footnote{[240]} correspond the setting of the mvBlueCOUGAR? Using a peer-to-peer connection leads to a better performance and normally you can use values higher than 1500. Here, values between 4k and 12k make sense, whereas some network controllers only allow maximum of 9k which is enough for sensors up to 4 megapixels.
5. Is the NIC (p. 240) parameter "Interrupt Moderation" switched on?

6. Some multi-port network cards might also offer to configure the number of "RSS (Receive Side Scaling) Queues". When dealing with high data rates, increasing the number of queues might also improve the overall stability of the system in terms of the possibility of packet losses etc.
7. Is the amount of the "Receive Descriptors" (RxDesc) of the NIC (p. 240) set to the maximum (generally 2048)?

This amount depends on the MTU (p. 240). You can get a feeling about values with following formula:

\[
\text{RxDesc} \geq \frac{1.1 \times \text{PixelPerImg} \times \text{BytesPerPixel}}{\text{MTU}}
\]

*Note*

The formula is per camera and per network port.

"Example 1: 1500 at 1.3MPixel"

\[
\text{RxDesc} \geq \frac{1.1 \times 1.3\text{M} \times 1}{1500} = 950
\]

"Example 2: 8k at 1.3MPixel"

\[
\text{RxDesc} \geq \frac{1.1 \times 1.3\text{M} \times 1}{8192} = 175
\]

*Note*

Both examples are showing the RxDesc values based on packages per image per second. Now, if you increase the images per second you will see that you reach the amount of "Receive Descriptors" very soon. Usually, the default values of the system are between 64 and 256 and as a general rule too low. Also each use of a "Receive Descriptor" will result in a certain amount of CPU time needed, which is why larger packets result in a better overall performance.
If you are using small MTU (p. 240) values or if there is only a small value of "Receive Descriptors" possible, you have to enable "Interrupt Throttling". Interrupt throttling enables the NIC (p. 240) to combine packets thus lowering the amount of interrupts.

The "Receive Descriptors" a.k.a. "Receive Buffers" (Intel PRO/1000 network interface) can be found under "Performance Options":

![Performance Options]

Figure 10: "Performance Options"

![Receive Buffers]

Figure 11: Setting "Receive Buffers"

### 9.2.5 Additional network configuration for mvBlueCOUGAR-XD

The mvBlueCOUGAR-XD Dual-GigE camera needs a network interface card with two network interfaces which support so-called Link Aggregation, teaming, or "bonding". Normally, Windows installs a standard driver which does not support link aggregation. For this, you have to install and download a driver from the manufacturer’s website. When e.g. using a "Intel Ethernet Server Adapter I350-T2" network controller this will be:

See also


After installation you have to combine the two interfaces which are used by the mvBlueCOUGAR-XD. For this, open the driver settings of the network controller via the "Device Manager" of Windows:

9.2.5.1 Link Aggregation / Teaming with Windows 7

1. Open the tab "Teaming" and
2. create a new team.

![Create an aggregation group](image1.png)

**Figure 12: Create an aggregation group**

3. Then enter a name for this group e.g. "mvBlueCOUGAR-XD" and
4. click on "Next".

![Enter an aggregation group name](image2.png)

**Figure 13: Enter an aggregation group name**
5. Select the two interfaces which were used for the mvBlueCOUGAR-XD.

![Select the used interfaces](image1)

Figure 14: Select the used interfaces

6. Finally, select the connection type "**Static Link Aggregation**".

![Select the connection type](image2)

Figure 15: Select the connection type

After closing the group assistant, you can connect the camera using the two network interfaces. There is no special connection necessary between camera and network controller interfaces.

It is also possible to use the camera with one network interface. This of course will result in a reduced data rate. For this use the right (primary) network interface of the camera only (the label on the back can be read normally).

### 9.2.5.2 Link Aggregation / Teaming with Windows 10

1. Click Start (Windows Icon in the taskbar) and type in **Powershell**.
2. Right-click the Powershell icon and choose "**Run as Administrator**".
3. If you are prompted to allow the action by User Account Control, click **Yes**.
4. Enter the command (e.g. for Intel NIC)

```powershell
Import-Module -Name "C:\Program Files\Intel\Wired Networking\IntelNetCmdlets\IntelNetCmdlets"
```

5. Afterwards, execute the command:

```powershell
New-IntelNetTeam -TeamName "HDR Camera" -TeamMemberNames "Intel(R) Ethernet Server Adapter I350-T2","Intel(R) Ethernet Server Adapter I350-T2 #2" -TeamMode StaticLinkAggregation
```

Replace "HDR Camera" with your preferred teaming name and also "Intel(R) Ethernet Server Adapter I350-T2" with the name of your used NIC.

9.3 Linux

Please check our website for the availability of the Linux driver.

9.3.1 System requirements

- Kernel 3.5.x or greater

In case the target system runs an older Linux kernel, it is absolutely necessary to update the kernel to at least version 3.5.0. Please refer to the documentation of your Linux distribution for information on how to update your system's Linux kernel.

Following Kernels have been tested and verified by MATRIX VISION for seamless USB3 operation:

- Kernel 3.8.0
- Kernel 3.11.0
- Kernel 3.14.0

Before installation on a Linux system (e.g. Ubuntu 12.04), the system has to provide at least following packages:

- libwxbase3.0-0
- libwxbase3.0-dev
- libwxgtk3.0-0
- libwxgtk3.0-dev
- wx3.0-headers
- build-essential (meta package)
- libgtk2.0-dev
- gcc 4.1 environment or newer
9.3.1.1 Additional hardware requirements for mvBlueCOUGAR-XD

The mvBlueCOUGAR-XD is a high-performance camera containing latest top-notch technology which requires

- a high data throughput on the host system,

for example, when processing large amount of image data and

- high CPU resources if processing color images on the host PC.

This means that you will need

- the latest PC architectures, preferably multi core Intel CPUs and chipsets
- sufficient RAM (4 GB in 32-bit OS; 8 GB in 64-bit OS), and
- appropriate and recommended network cards with PCI-E x1 or x4 interface and with the latest "Link Aggregation Group" (LAG) capable drivers.

Please ask your system vendor for further advice and consult our technical documentation.

Beside the host system requirements, you have to fine-tune

- the network settings,
- the camera settings, and
- the host system settings (e.g. disabling the sleep mode)

to achieve the specified frame rates possible from the camera's perspective.

9.3.2 Installing the mvGenTL-Acquire driver

**Note**

If you are going to install the mvGenTL-Acquire driver on an embedded board, please read this section first.

To use the camera within Linux (grab images from it and change its settings), a driver is needed, consisting of several libraries and several configuration files. These files are required during run time.

To develop applications that can use the camera, a source tree is needed, containing header files, makefiles, samples, and a few libraries. These files are required at compile time.

Both file collections are distributed in a single package:

`mvGenTL_Acquire-x86-n.n.n.tgz`

- Please start a console and change into a directory e.g. `/home/username/workspace`:

  ```
  cd /home/username/workspace
  ```

- Copy the install script and the hardware driver to the workspace directory (e.g. from the website):

  ```
  ~/workspace$ cp /home/Downloads/install_mvGenTL_Acquire.sh /home/Downloads/mvGenTL_Acquire-x86-2.3.2.tgz
  ```

- Run the install script:

  ```
  ~/workspace$ ./install_mvGenTL_Acquire.sh
  ```
The install script has to be executable. So please check the rights of the file. During installation, the script will ask, if it should build all tools and samples.

You may need to enable the execute flag with

```
chmod a+x install_mvGenTL_Acquire.sh
```

The installation script checks the different packages and installs them with the respective standard packages manager (e.g. `apt-get`) if necessary.

The installation script (`install_mvGenTL_Acquire.sh`) and the archive (`mvGenTL_Acquire-x86-n.n.n.tgz`) must reside in the same directory. Nothing is written to this directory during script execution, so no write access to the directory is needed in order to execute the script.

You need Internet access in case one or more of the packages on which the GenICam (p. 234) libs depend are not yet installed on your system. In this case, the script will install these packages, and for that, Internet access is required.

The script takes two arguments, both of which are optional:

- target directory name
- version

The target directory name specifies where to place the driver. If the directory does not yet exist, it will be created. The path can be either absolute or relative; i.e. the name may but need not start with `/`.

This directory is only used for the files that are runtime required.

The files required at compile time are always installed in

```
$HOME/mvimpact-acquire-n.n.n
```

. The script also creates a convenient softlink to this directory:

```
mvimpact-acquire -> mvIMPACT_acquire-2.6.0
```

If this argument is not specified, the driver will be placed in the default directory `/opt`.

The version argument is entirely optional. If no version is specified, the most recent `mvGenTL_Acquire-x86-n.n.n.tgz` found in the current directory will be installed.
9.3.3 Optimizing the network configuration

This chapter explains how to achieve optimal performance by setting up the NIC (network interface card) and the devices network controller in the best possible way with respect to the limitations imposed by the surrounding network. To check if the camera is available via the network, you can use the tool mvIPConfigure.

**Attention**

Only network experts should change settings in this section.

The camera sets the MTU to the maximum value automatically given by the NIC or switch and supports a maximum MTU of 16K. You can change the MTU of the camera manually under "TransportLayer" -> "GevSCPSPacketSize".

In the system settings of the camera, you can set up the overall network behaviour of the device in the "TransportLayer" section e.g. by assigning a persistent IP address, a persistent default gateway and a persistent subnet mask in "TransportLayer" section according to your needs. However IP configuration related settings should be done by using the tool mvIPConfigure as this will provide a more convenient access to these parameters.

Network performance related parameters however can be controlled from the "TransportLayer" section as well:

![Figure 16: wxPropView - System settings](image)

By default 1500 is used as MTU ("DeviceInterfaceMTU"). You can change this according to your network. As a general rule of thumb it can be said, that the higher the MTU, the better the overall performance, as less network packets are needed to transmit a full image, which results in less overhead that arises from the handling of each arriving network packet in the device driver etc. However every component involved in the data transmission (that includes every switch, router and other network component, that is installed in between the device and the system receiving the packets) must support the MTU or packet size, as otherwise the first component not supporting the packet size will silently discard all packets, that are larger than the component can handle.

**Note**

If you modify the interface MTU of the device, it will take up to 5 seconds until these settings are saved in the non-volatile memory. During this time do not remove the power supply! Afterwards you have to reboot the device for the changed settings to take effect.
9.3.4 Adjusting the network controller

If you have interplay problems between mvBlueCOUGAR and a network controller (NIC (p.240)) please check following settings in the driver settings:

**Note**

There is no need to set the transfer packet size manually. Whenever the device is initialised, the driver will determine the maximum possible packet size for the current network settings automatically.

1. Is the amount of the "Receive Buffers" of the NIC (p.240) set to the maximum? If you want to set the "Receive Buffers", you have to

   (a) create a new file 62-buffers-performance.conf in "/etc/sysctl.d/" (/etc/sysctl.d/62-buffers-performance.conf) and
   (b) add the following lines to this file:

   ```
   net.core.wmem_max=16777216
   net.core.rmem_max=16777216
   net.core.netdev_max_backlog=10000
   ```
   (c) Afterwards, reboot the system or enter "sudo sysctl -p".

**See also**

For more information about Linux network tuning, please have a look at http://www.cyberciti.biz/faq/linux-tcp-tuning/.

9.3.5 Additional network configuration for mvBlueCOUGAR-XD

The mvBlueCOUGAR-XD Dual-GigE camera needs a network interface card with two network interfaces which support so-called "Link Aggregation" or "bonding".

To bond the network interface, the following description assumes that your system has three interfaces eth0, eth1, and eth2.

- eth0 is the standard system interface (e.g. connected to the Internet or intranet) and
- the mvBlueCOUGAR-XD is connected to eth1 and eth2 (these interfaces will be bonded).

Furthermore, it is assumed that the

- *Class C subnet* 192.168.10.1 is used for bonding, that the
- *receive buffers value was raised* (p.58), and
- Intel network interface controller (NIC) is used and the Intel e1000e Kernel module is available.
9.4 Connecting the camera

Note

If your system uses an active network manager, which assigns IP automatically, you should stop the network manager as follows:

```
sudo service network-manager stop
```

To bond the network interfaces, please follow these steps:

1. Install the package ifenslave:

```
sudo apt-get install ifenslave
```

2. Turn the network interfaces down, which you want to bond:

```
sudo ifconfig eth1 down
sudo ifconfig eth2 down
```

3. Afterwards, load the bonding Kernel module:

```
sudo modprobe bonding mode=0 miimon=100
```

"mode=0" means "round robin" mode which corresponds to the "static link aggregation" on Windows systems

"miimon=100" means that it will be checked every 100 ms if there are link failures in the bonding. 100 is the recommended value.

4. Now you have to define the parameters of the bonding:

```
sudo ifconfig bond0 192.168.10.1 netmask 255.255.255.0 mtu 8000
```

5. You have to link the physical network interface controllers to the bonding and turn them on:

```
sudo ifenslave bond0 eth1 eth2
sudo ifconfig eth1 up
sudo ifconfig eth2 up
```

6. Finally, you have to restart the Linux networking services:

```
sudo /etc/init.d/networking restart
```

Now, you can check if you can ping the mvBlueCOUGAR-XD via the bonding.

9.4 Connecting the camera

To start the device please accomplish following steps:

1. Connect the mvBlueCOUGAR with a

   • CAT5e/CAT6 Peer-to-Peer-cable to a network (e.g. a switch) or PC directly
With mvBlueCOUGAR-XD you can either connect both LAN connectors (RJ 45) using Link Aggregation (p. 51) or one LAN connector, which requires to use the connector LAN 1 (p. 81) only.

2. **Power the mvBlueCOUGAR via power supply.**
   The mvBlueCOUGAR will boot immediately.

   **Note**
   If you're using **Power over Ethernet (PoE)** (p. 242), connecting and powering are one step.
   If both power possibilities are connected (local power supply and **Power over Ethernet (PoE)** (p. 242)), the camera will use the local power supply.
   If you change from one supply to the other, the camera will reset and reboot.

   After energizing mvBlueCOUGAR via power supply or **Power over Ethernet (PoE)** (p. 242), the camera will start to boot.

   Depending on the configuration of the network or the mvBlueCOUGAR (more precisely: if **DHCP** (p. 234) is used or not), following IP addresses are assigned:

   - If the mvBlueCOUGAR is configured to use a static IP address, the mvBlueCOUGAR will use this one. In this case, the mvBlueCOUGAR will only be accessible, if the IP address is valid for the current network.
   - With **DHCP** (p. 234), the mvBlueCOUGAR will get an IP address from the **DHCP** (p. 234) server (default setting).
   - Without **DHCP** (p. 234), the mvBlueCOUGAR will start with a logical link address (LLA (p. 238)) / zero configuration class B IP address ("169.254.X.X", netmask 255.255.0.0).

   **Note**
   Please have a look at **LLA** (p. 238) in the **Glossary** (p. 234) when using Linux and **LLA** (p. 238).

   **Warning**
   If you are using a static IP address, please be sure that this IP address
   - does exist
   - is not used by anyone else in the same network
   - is a valid one

   **Note**
   If you have two **NICs** (p. 240) in your PC, please be sure that they are using different subnets, otherwise the systems does not know which Ethernet interface is used to exchange the data.

   **Working example:**
   192.168.0.1: PC Port1
   172.16.1.11: PC Port2
   192.168.0.3: Cam1
   172.16.1.4: Cam2
You can change settings like the static IP address to your needs (please have a look at: `mvIPConfigure` (p. 152)).

Further the mvBlueCOUGAR starts the `GigE Vision` (p. 236) control server making the device actually discoverable using standard `GigE Vision` (p. 236) mechanism.

**Figure 17: Boot sequence of mvBlueCOUGAR**

### 9.4.1 Communicating with the camera

You can communicate with the camera the following way

1. **Via `wxPropView` (p. 115):**
   - Since driver version 2.11.3, starting `wxPropView` (p. 115) the first time, the **Quick Setup Wizard** (p. 116), a tiny and powerful configuration tool, will be started.
   - If **Quick Setup Wizard** (p. 116) was disabled, `wxPropView` (p. 115) will start without wizard and you will see the serial number of the camera in the box "Serial No". After clicking "Use" the program connects to the camera. With "Acquire" you can see live images from the camera.

   **Note**
   
   The current IP address can be found in the "Device" properties under "DeviceIPAddress".

### 9.4.2 Setting up the camera

Besides `wxPropView` (p. 115), there are further tools to set up the device:

- `mvGigEConfigure` (p. 149).
- `mvIPConfigure` (p. 152) or
• **mvDeviceConfigure** (p. 156)

With **wxPropView** (p. 115) it is possible to change e.g. the

• image properties or

• IP address.

With **mvGigEConfigure** (p. 148) it is possible to install the

• **GigE** (p. 236) capture filter of the network controller.

With **mvIPConfigure** (p. 152) it is possible to configure the

• network behavior of **GigE** (p. 236) devices.

**mvDeviceConfigure** (p. 156) is the program e.g.

• to set the device ID or

• to update firmware or

• to upload a **GenICam** (p. 234) XML file or

• to **disable CPU sleep states** (p. 163).

For further information about the programs, please follow the link to the detailed description.

9.4.3 Settings behaviour during startup

**Settings** contain all the parameters that are needed to prepare and program the device for the image capture. Every image can be captured with completely different set of parameters. In almost every case, these parameters are accessible via a property offered by the device driver. A setting e.g. might contain

• The gain to be applied to the analog to digital conversion process for analog video sources or

• The AOI to be captured from the incoming image data.

So for the user a setting is the one an only place where all the necessary modifications can be applied to achieve the desired form of data acquisition. There is however an important difference in behaviour between different interface layouts. See **here** (p. 143) to find out how to modify the interface layout or check in the API documentation for the **interfaceLayout** property of the class **Device**. When working in **DeviceSpecific** interface layout, each frame will be captured with the settings as present when requesting the image. For **GenICam** interface layouts all device properties modified during a continuous acquisition will have immediate impact on the next frame transmitted by the device thus here when a precise moment to change settings is needed, continuous acquisition must be stopped and then restarted after modifying the features.

Now, whenever a device is opened, the driver will execute following procedure:
• Please note that each setting location step in the figure from above internally contains two search steps. First the framework will try to locate a setting with user scope and if this can't be located, the same setting will be searched with global (system-wide) scope. Under Windows this e.g. will access either the HKEY_CURRENT_USER or (in the second step) the HKEY_LOCAL_MACHINE branch in the Registry.

• Whenever storing a product specific setting, the device specific setting of the device used for storing will be deleted (if existing). E.g. you have a device "VD000001" which belongs to the product group "Virtual Device" with a setting exclusively for "VD000001". As soon as you store a product specific setting, the (device specific) setting for "VD000001" will be deleted. Otherwise a product specific setting would never be loaded as a device specific setting will always be found first.

• The very same thing will also happen when opening a device from any other application! wxPropView (p. 115) does not behave in a special way but only acts as an arbitrary user application.

• Whenever storing a device family specific setting, the device specific or product specific setting of the device used for storing will be deleted (if existing). See above to find out why.

• Under Windows the driver will not look for a matching XML file during start-up automatically as the native storage location for settings is the Windows Registry. This must be loaded explicitly by the user using the appropriate API function offered by the SDK. However, under Linux XML files are the only setting formats understood by the driver framework thus the driver will also look for them at start-up. The device specific setting will be an XML file with the serial number of the device as the file name, the product specific setting will be an XML file with the product string as the filename, the device family specific setting will be an XML file with the device family name as the file name. All other XML files containing settings will be ignored!

• Only the data contained in the lists displayed as
  - "Image Setting",
  - "Digital I/O", and
  - "Device Specific Data" under wxPropView (p. 115) will be stored in these settings!

Figure 18: wxPropView - Device setting start procedure
• Restoring of settings previously stored works in a similar way. After a device has been opened the settings will be loaded automatically as described above.

• A detailed description of the individual properties offered by a device will not be provided here but can be found in the C++ API reference, where descriptions for all properties relevant for the user (grouped together in classes sorted by topic) can be found. As wxPropView (p.115) doesn’t introduce new functionality but simply evaluates the list of features offered by the device driver and lists them any modification made using the GUI controls just calls the underlying function needed to write to the selected component. wxPropView (p.115) also doesn’t know about the type of component or e.g. the list of allowed values for a property. This again is information delivered by the driver and therefore can be queried by the user as well without the need to have special inside information. One version of the tool will always be delivered in source so it can be used as a reference to find out how to get the desired information from the device driver.

9.5 Relationship between driver, firmware, FPGA file and user settings

To operate a GenICam (p.234) based device like mvBlueFOX3 apart from the physical hardware itself 2 pieces of software are needed:

• A firmware running on the device. This firmware consists of
  – A GenICam (p.234) compliant XML file exposing the features in a generic and standard compliant way
  – A FPGA file
  – The actual micro-code making the device operational

• A device driver (this is the mvGenTLConsumer.dll and the mvGenTLProducer.cti on Windows and the libmvGenTLConsumer.so and the libmvGenTLProducer.so on Linux when using mvMPACT Acquire, but can be any other USB3 Vision (p.242) / GigE Vision (p.236) compliant driver package from a third party vendor) running of the host system (provides control over the device from an application running on the host system).

The physical GenICam (p.234) compliant device has a firmware programmed into the device’s non-volatile memory, thus allowing the device to boot to a fully functional state without the need of any additional software running on the host. The firmware version that will be used when operating the device does NOT depend on the driver version that is used to communicate with the device. This will allow any piece of compliant third party software to operate the device without the need to have special knowledge about the firmware structure. This shall be illustrated by the following figure:

Figure 19: The firmware is not a part of the device driver
As it can be seen in the image the firmware file is NOT part of the device driver but comes as a separate archive. It is important to notice that a firmware file that may be present on the host system will not be used automatically but only when the user or an application explicitly updates the firmware on the device and will only become active after power-cycling the device.

The name of the firmware update archive (+ in the figure above) is:

- **mvBlueCOUGAR-X_Update.mvu**

Only during a firmware update the firmware file that has been selected from the file system of the host system will be downloaded permanently into the device's non-volatile memory.

---

**Warning**

Each firmware archive might contain more than one specific firmware version per device thus in order to select the one that is appropriate for the device appropriate tools such as **mvDeviceConfigure** (p. 156) should be used.

So assume a device with a certain firmware version is connected to a host system.

During an explicit firmware update, the firmware file will be downloaded onto the device. In order to become active the device must be power-cycled:

![Diagram of firmware update process](image.png)

**Figure 20:** Firmware file will be downloaded during an firmware update...

This can either be done by unplugging the device and then by plugging it back in or (for devices supporting this feature) by resetting/rebooting the device by a certain software command (DeviceControl/DeviceReset). When using **mvDeviceConfigure** (p. 156) to update devices the latter mechanism will be used by the tool thus it is NOT necessary to unplug the device.

When the device has completed rebooting the new firmware version will become active:
The current firmware version of the device can be obtained either by using one of the applications which are part of the SDK such as `mvDeviceConfigure` or by reading the value of the property `Device/FirmwareVersion` or `DeviceControl/DeviceFirmwareVersion` using the API.

The current FPGA file version used by the device can be obtained by reading the value of the property `DeviceControl/mvDeviceFPGAVersion`.

Note: The FPGA file is a part of the firmware and cannot be updated independently; thus, reading its version just provides some additional information.

Using `wxPropView` the same information is available as indicated by the following figure:

Figure 22: `wxPropView` - FPGA and Firmware version numbers

Apart from the device driver and firmware relationship, there are certain places where a device configuration can be stored when dealing with GenICam compliant devices:

- There may be User Sets which are stored in the device's non-volatile memory. User Sets contain all the features, which affect the device's behaviour such as transfer pixel format, exposure time, etc. User Sets are bound to major GenICam XML file releases, thus these settings will be lost whenever a firmware contains a different major version of a device's GenICam XML file.
• mvIMPACT Acquire settings which contain the state of all the features also stored in a User Set as well as other features added by the device driver. These settings will be stored on the host system either as a XML file or (under Windows only) in the Registry.

Both methods can be used to pre-configure a device. Using the first method, the state of the features will travel with the physical device, using the mvIMPACT Acquire settings, feature states can be copied from host to host as a file.
10  Technical data

10.1  mvBlueCOUGAR-X / -POE

10.1.1  Dimensions

10.1.1.1  Standard model (-xx1x)

<table>
<thead>
<tr>
<th>front view</th>
<th>side view</th>
<th>back view</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="" /></td>
<td><img src="image2" alt="" /></td>
<td><img src="image3" alt="" /></td>
</tr>
</tbody>
</table>

**Figure 1: Connectors mvBlueCOUGAR-X**

<table>
<thead>
<tr>
<th>mvBlueCOUGAR-X</th>
<th>Size of body (w x h x l)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>39.8 x 39.8 x Y+5 mm</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Length</th>
<th>CMOSIS sensors</th>
<th>Remaining sensors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>32 mm</td>
<td>30 mm</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lens protrusion</th>
<th>C-Mount</th>
<th>CS-Mount</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>8 mm (9.5 mm with max. Ø 20 mm)</td>
<td>6 mm</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C-Mount</th>
<th>CS-Mount</th>
</tr>
</thead>
<tbody>
<tr>
<td>W</td>
<td>approx. 15 mm (C-mount)</td>
</tr>
<tr>
<td>Z</td>
<td>17.526 mm (in air)</td>
</tr>
</tbody>
</table>

10.1.1.2  Model without housing (-xx9x) (on demand)

<table>
<thead>
<tr>
<th>front view</th>
<th>side view</th>
<th>back view</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image4" alt="" /></td>
<td><img src="image5" alt="" /></td>
<td><img src="image6" alt="" /></td>
</tr>
</tbody>
</table>

**Figure 2: mvBlueCOUGAR-X without housing**
10.1 mvBlueCOUGAR-X / -POE

10.1.1.3 Tripod adapter

Figure 3: Dimensional drawing of tripod adapter

10.1.2 Connectors

10.1.2.1 Circular connector male

Figure 4: 12-pin (male; top view), digital I/O, power; orientation mvBlueCOUGAR-X

<table>
<thead>
<tr>
<th>Pin</th>
<th>Signal (Standard version)</th>
<th>Signal (-POE (p.242) option, not available for all products)</th>
<th>Digital I/O color code of power supply MV-DC1201 BCSX IO; Rev. 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GND</td>
<td>GND&lt;sup&gt;1&lt;/sup&gt;</td>
<td>gray-pink (not available with Rev. smaller than 2)</td>
</tr>
<tr>
<td>2</td>
<td>12V - 24V</td>
<td>12V - 24V&lt;sup&gt;1&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>DigOut3&lt;sup&gt; wxPropView&lt;/sup&gt; (p.115) numbering: line3</td>
<td>Reserved (Video iris to be supported on coming FW version←: Ask support of MATRIX VISON&lt;sup&gt; ON for more details.)&lt;/sup&gt;</td>
<td>white</td>
</tr>
<tr>
<td>4</td>
<td>Opto DigIn0 (line4)</td>
<td>Opto DigIn0 (line4)</td>
<td>brown</td>
</tr>
<tr>
<td>5</td>
<td>DigOut2 (line2)</td>
<td>N.C.</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>DigOut0 (line0)</td>
<td>Opto DigOut0 (line0)</td>
<td>yellow</td>
</tr>
<tr>
<td>7</td>
<td>Opto GND</td>
<td>Opto GND</td>
<td>gray</td>
</tr>
<tr>
<td>8</td>
<td>RS232_RX</td>
<td>RS232_RX</td>
<td>pink</td>
</tr>
<tr>
<td>9</td>
<td>RS232_TX</td>
<td></td>
<td>blue</td>
</tr>
<tr>
<td>10</td>
<td>+24V_HSD (12..24 V supply for the outputs)</td>
<td>+24V_OPT (12..24 V supply for the outputs)</td>
<td>red (using MV-DC1201 BCSX IO: you have to add supply for the outputs)</td>
</tr>
<tr>
<td>11</td>
<td>Opto DigIn1 (line5)</td>
<td>Opto DigIn1 (line5)</td>
<td>black</td>
</tr>
<tr>
<td>12</td>
<td>DigOut1 (line1)</td>
<td>Opto DigOut1 (line1)</td>
<td>violet</td>
</tr>
</tbody>
</table>
Using POE and an external power supply at the same time, the external power supply will be treated prioritized.

Connector (camera side): SAMWOO SNH-10-12 (RPCB) or equivalent
Plug (matching cable plug): Hirose HR10A-10P-12S (01) or equivalent

10.1.2.1.1 Pinning of cable KS-BCX-HR12

<table>
<thead>
<tr>
<th>Pin.</th>
<th>CON 1 mvBlueCOUGAR-X / -XD</th>
<th>Signal</th>
<th>CON 2 cutted cable</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GND (Power)</td>
<td></td>
<td></td>
<td>black</td>
</tr>
<tr>
<td>2</td>
<td>V_{IN} +12V to +24V</td>
<td></td>
<td></td>
<td>brown</td>
</tr>
<tr>
<td>3</td>
<td>DigOut3 12V to 24V (wxPropView (p. 115) numbering: line3)</td>
<td>red</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Opto DigIn0 3.3V to 24V (line4)</td>
<td>orange</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>DigOut2 (line2)</td>
<td></td>
<td></td>
<td>yellow</td>
</tr>
<tr>
<td>6</td>
<td>DigOut0 (line0)</td>
<td></td>
<td></td>
<td>green</td>
</tr>
<tr>
<td>7</td>
<td>Opto GND</td>
<td></td>
<td></td>
<td>blue</td>
</tr>
<tr>
<td>8</td>
<td>RS232_{RX}</td>
<td></td>
<td></td>
<td>violet</td>
</tr>
<tr>
<td>9</td>
<td>RS232_{TX}</td>
<td></td>
<td></td>
<td>gray</td>
</tr>
<tr>
<td>10</td>
<td>+24V_HSD (12..24 V supply for the outputs)</td>
<td>white</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Opto DigIn1 3.3V to 24V (line5)</td>
<td>white-black</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Opto DigOut1 12V to 24V (line1)</td>
<td>white-brown</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Main connector shield</td>
<td></td>
<td></td>
<td>main shield</td>
</tr>
</tbody>
</table>

Color assignment following international code for UL wiring.

10.1.2.2 Characteristics of the digital inputs

10.1.2.2.1 Delay

![Figure 5: Input switching times](image)

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Symbol</th>
<th>Test conditions</th>
<th>Typ.</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum trigger pulse width</td>
<td></td>
<td></td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Turn-On time</td>
<td>I_{ON}</td>
<td></td>
<td>2</td>
<td>μs</td>
</tr>
<tr>
<td>Storage time</td>
<td>t_s</td>
<td>R = 2 kOhm (Figure 7), internal output voltage 5V, I_e = 16 mA</td>
<td>2</td>
<td>μs</td>
</tr>
<tr>
<td>Turn-Off time</td>
<td>t_{OFF}</td>
<td></td>
<td>40</td>
<td></td>
</tr>
</tbody>
</table>
### 10.1.2.3 Characteristics of the digital outputs

#### Standard and -POE (p. 242) option

<table>
<thead>
<tr>
<th>Notes</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>High level</td>
<td>+3 to +24V</td>
</tr>
<tr>
<td>Low level</td>
<td>0 to +0.3V</td>
</tr>
<tr>
<td>Threshold (Low -&gt; High / High -&gt; Low)</td>
<td>2.5V +- 0.5V</td>
</tr>
<tr>
<td>$I_{\text{max}}$</td>
<td>5mA</td>
</tr>
</tbody>
</table>

![Figure 6: DigIn mvBlueCOUGAR-X (standard)](image)

#### PLC option (inputs with PLC levels only)

<table>
<thead>
<tr>
<th>Notes</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Not available in combination with -POE (p. 242) option. Inputs with highest tolerance against interferences.</td>
<td></td>
</tr>
<tr>
<td>High level</td>
<td>+11 to +24V</td>
</tr>
<tr>
<td>Low level</td>
<td>0 to +8V</td>
</tr>
<tr>
<td>Threshold (Low -&gt; High / High -&gt; Low)</td>
<td>10V +- 1V</td>
</tr>
<tr>
<td>$I_{\text{max}}$</td>
<td>16mA</td>
</tr>
</tbody>
</table>

![Figure 7: DigIn mvBlueCOUGAR-X (PLC option)](image)
Standard version

Delay

Figure 8: Output switching times (standard)

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Symbol</th>
<th>Typ.</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turn-On time</td>
<td>$t_{ON}$</td>
<td>20</td>
<td>us</td>
</tr>
<tr>
<td>Rise time</td>
<td>$t_{R}$</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Fall time</td>
<td>$t_{F}$</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Turn-Off time</td>
<td>$t_{OFF}$</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>High level</td>
<td></td>
<td>+12.0V to +24V</td>
<td></td>
</tr>
<tr>
<td>Low level</td>
<td></td>
<td>max. +1V</td>
<td></td>
</tr>
<tr>
<td>$I_{out}$</td>
<td></td>
<td>4x 0.7A max. short circuit</td>
<td></td>
</tr>
</tbody>
</table>

Notes
Each output has a short circuit protection between 1 A and 1.7 A (generally 1.3 A). So if you combine two outputs with one load, the short circuit protection can have an effect.

Figure 9: DigOut mvBlueCOUGAR-X

Features
- undervoltage detection
- current limitation
- overtemperature protection
- short circuit protection

-POE (p.242) option
Delay

![Figure 10: Output switching times (POE (p. 242) option)](image)

**Characteristics**

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Symbol</th>
<th>Test conditions</th>
<th>Typ.</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turn-On time</td>
<td>(t_{ON})</td>
<td>(R_L = 100) Ohm, (V_{CC} = 10V, I_C = 2mA)</td>
<td>3</td>
<td>(\text{us})</td>
</tr>
<tr>
<td>Storage time</td>
<td>(t_{S})</td>
<td>(R_L = 100) Ohm, (V_{CC} = 10V, I_C = 2mA)</td>
<td>3</td>
<td>(\text{us})</td>
</tr>
<tr>
<td>Turn-Off time</td>
<td>(t_{OFF})</td>
<td>(R_L = 1.9) kOhm, (V_{CC} = 5V, I_C = 16mA)</td>
<td>2</td>
<td>(\text{us})</td>
</tr>
<tr>
<td>Turn-On time</td>
<td>(t_{ON})</td>
<td>(R_L = 1.9) kOhm, (V_{CC} = 5V, I_C = 16mA)</td>
<td>40</td>
<td>(\text{us})</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Comment</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>(I_{on}) load current</td>
<td>15</td>
<td>mA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(I_{off}) leakage current</td>
<td>10</td>
<td>uA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(V_{on}) Sat. at 2.4mA</td>
<td>0 (0.2)</td>
<td>0.4</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>(V_{off})</td>
<td>30</td>
<td></td>
<td></td>
<td>V</td>
</tr>
</tbody>
</table>

**Notes**

![Figure 11: DigOut mvBlueCOUGAR-X with example circuit (POE (p. 242) option)](image)

**Features**

**Measurement conditions**: \(V_{CC} = 24\) V; \(-25 ^\circ\) C < \(T_J < 125 ^\circ\) C, \(R_L = 48\) Ohm

10.1.2.4 Example circuit 1: High-side switch uses power supply of the camera

In this case you have to bridge pin 2 (external power supply) with pin 10 (the supply of the outputs):
10.1.2.5 Example circuit 2: High-side switch uses external (higher) power supply

Following figure shows, how to connect, for example, high power flash LEDs in series at the direct drive output at pin 3:

A 24V power supply is connected at pin 10. To protect the LEDs a series resistor is needed, which can be calculated in this way:

Voltage:

\[ U_{HS} - (n \times U_F) = U_D \]

Series resistor:

\[ R = \frac{U_D}{0.7A} \]

Power:

\[ P = \text{duty cycle} \times U_D \times I \]

The duty cycle is a coefficient \(< 1\), which defines the power of the resistor. It is the ratio of the time period while current flows compared to the total time. If the current always flows, \(P = U \times I\) will be the determining factor. If the current only flows a hundredth of time (duty cycle 1%), a resistor with a hundreth of power will be enough.
10.1 mvBlueCOUGAR-X / -POE

**Note**

For the connected LEDs and the ground of the external power supply, you have to use the ground of the camera at Pin 1.

**Warning**

The series resistor has to handle the power, which depends on the duty cycle and the frequency.

10.1.2.6 Example circuit 3: Control motorized lens with mvBlueCOUGAR-X

It is possible to control a motorized lens using the mvBlueCOUGAR-X. However, an external micro relay is necessary which changes the polarity of the lens slider when triggered by the camera's digital output. The following figure shows an example circuit:

![Figure 14: External micro delay](image)

10.1.2.7 RJ45 100/1000 MBit Ethernet

With the -POE option, the camera is a class 2 compliant PoE device.

![Figure 15: RJ 45 pin numbering (female)](image)

<table>
<thead>
<tr>
<th>Pin</th>
<th>Signal</th>
<th>I/O</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>D1+</td>
<td>bi</td>
<td>data positive</td>
</tr>
<tr>
<td>2</td>
<td>D1-</td>
<td>bi</td>
<td>data negative</td>
</tr>
<tr>
<td>3</td>
<td>D2+</td>
<td>bi</td>
<td>data positive</td>
</tr>
<tr>
<td>4</td>
<td>D3+</td>
<td>bi</td>
<td>data positive</td>
</tr>
<tr>
<td>5</td>
<td>D3-</td>
<td>bi</td>
<td>data negative</td>
</tr>
<tr>
<td>6</td>
<td>D2-</td>
<td>bi</td>
<td>data negative</td>
</tr>
<tr>
<td>7</td>
<td>D4+</td>
<td>bi</td>
<td>data positive</td>
</tr>
</tbody>
</table>
The Ethernet signals are galvanically isolated from the camera electronics and camera housing.

### 10.1.2.8 RJ45 LED states

<table>
<thead>
<tr>
<th>States</th>
<th>LED1</th>
<th>LED2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power on</td>
<td>Green light on</td>
<td>Yellow light on</td>
</tr>
<tr>
<td>1Gb transmission</td>
<td>Green light on</td>
<td>Yellow blink on</td>
</tr>
<tr>
<td>100Mb transmission</td>
<td>Green light off</td>
<td>Yellow blink on</td>
</tr>
</tbody>
</table>

### 10.1.3 Signal LED

The mvBlueCOUGAR-X features a RGB LED. There are following states:

#### 10.1.3.1 Typical start sequence (LLA)

#### See also

**LLA (p. 59)**

<table>
<thead>
<tr>
<th>State</th>
<th>LED color</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPGA loaded</td>
<td>White on</td>
</tr>
<tr>
<td>Self-test running</td>
<td>Green on</td>
</tr>
<tr>
<td>Waiting for Ethernet connection</td>
<td>White blink</td>
</tr>
<tr>
<td>LLA (auto IP)</td>
<td>Green blink</td>
</tr>
<tr>
<td>LLA got</td>
<td>Green on</td>
</tr>
<tr>
<td>Waiting for client</td>
<td>Blue on</td>
</tr>
</tbody>
</table>

#### 10.1.3.2 Typical start sequence (DHCP)

#### See also

**DHCP (p. 59)**

<table>
<thead>
<tr>
<th>State</th>
<th>LED color</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPGA loaded</td>
<td>White on</td>
</tr>
<tr>
<td>Self-test running</td>
<td>Green on</td>
</tr>
<tr>
<td>Waiting for Ethernet connection</td>
<td>White blink</td>
</tr>
<tr>
<td>DHCP request</td>
<td>Blue blink</td>
</tr>
<tr>
<td>DHCP got</td>
<td>Bright Blue</td>
</tr>
<tr>
<td>Waiting for client</td>
<td>Blue on</td>
</tr>
</tbody>
</table>

#### 10.1.3.3 Typical start sequence (Fixed IP)

#### See also

**Fixed IP (p. 59)**
### 10.1.3.4 General behavior

<table>
<thead>
<tr>
<th>State</th>
<th>LED color</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting \textit{wxPropView} (p. 115)</td>
<td>Yellow on</td>
</tr>
<tr>
<td>Connected, streaming off</td>
<td>Yellow on</td>
</tr>
<tr>
<td>Streaming on</td>
<td>Green on</td>
</tr>
<tr>
<td>Error</td>
<td>Red on</td>
</tr>
</tbody>
</table>

### 10.2 mvBlueCOUGAR-X-POE-I / -IP67

#### 10.2.1 Dimensions

#### 10.2.1.1 Power over Ethernet model (-POE-I)

<table>
<thead>
<tr>
<th>front view</th>
<th>side view</th>
<th>back view</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Figure 16: mvBlueCOUGAR-X-POE-I" /></td>
<td><img src="image2.png" alt="Figure 16: mvBlueCOUGAR-X-POE-I" /></td>
<td><img src="image3.png" alt="Figure 16: mvBlueCOUGAR-X-POE-I" /></td>
</tr>
</tbody>
</table>

For sizes X and Y please have a look at: \textbf{Standard model (-xx1x) (p. 68)}

#### 10.2.2 Connectors

#### 10.2.2.1 Circular connector male (POE-I Rev. 2.00)

![Figure 17: M12 12-pin (male; top view), digital I/O, power](image4.png)
The power pinning is different to the **circular connector male** (p. 69) pinning due to the different cable color scheme.

<table>
<thead>
<tr>
<th>Pin</th>
<th>Pin Color</th>
<th>Signal</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>brown</td>
<td>PWR_IN+</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>blue</td>
<td>PWR_IN-/GND</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>white</td>
<td>DigOut3</td>
<td><strong>wxPropView</strong> (p. 115) numbering: line3, high side solid state relays</td>
</tr>
<tr>
<td>4</td>
<td>green</td>
<td>Opto DigIn0</td>
<td>line4, opto-coupler</td>
</tr>
<tr>
<td>5</td>
<td>pink</td>
<td>DigOut2</td>
<td>line2, high side solid state relays</td>
</tr>
<tr>
<td>6</td>
<td>yellow</td>
<td>DigOut0</td>
<td>line0, high side solid state relays</td>
</tr>
<tr>
<td>7</td>
<td>black</td>
<td>In GND</td>
<td>Ground for opto-coupler (In)</td>
</tr>
<tr>
<td>8</td>
<td>grey</td>
<td>Opto DigIn2</td>
<td>line6, opto-coupler</td>
</tr>
<tr>
<td>9</td>
<td>red</td>
<td>Opto DigIn3</td>
<td>line7, opto-coupler</td>
</tr>
<tr>
<td>10</td>
<td>purple</td>
<td>OUT_VCC</td>
<td>10..28 V supply for the high side outputs, ground via pin 2 &quot;PWR_IN-/GND&quot;</td>
</tr>
<tr>
<td>11</td>
<td>grey / pink</td>
<td>Opto DigIn1</td>
<td>line5, opto-coupler</td>
</tr>
<tr>
<td>12</td>
<td>red / blue</td>
<td>DigOut1</td>
<td>line1, high side solid state relays</td>
</tr>
</tbody>
</table>

1 Using POE and an external power supply at the same time, the external power supply will be treated prioritized.

Camera connector (male): Binder M12/12pol 09-3491-90-12 or equivalent

Matching cable and connector (female):

- Binder 79-3490-32-12 | M 12 | A-coded series 763, length 2m open end
- Binder 79-3490-35-12 | M 12 | A-coded series 763, length 5m open end

**Note**

- All inputs (DigIn1..DigIn4 and GND Opto-coupler) are galvanically isolated from the camera electronics.
  - The digital inputs have the same characteristics as the inputs of mvBlueCOUGAR-X with PLC option.
  - The digital outputs have the same characteristics as the outputs of the standard mvBlueCOUGAR-X.

**10.2.2.2 Characteristics of the digital inputs**

**10.2.2.2.1 Delay**

![Figure 18: Input switching times](image)

**Note**

- All inputs (DigIn1..DigIn4 and GND Opto-coupler) are galvanically isolated from the camera electronics.
  - The digital inputs have the same characteristics as the inputs of mvBlueCOUGAR-X with PLC option.
  - The digital outputs have the same characteristics as the outputs of the standard mvBlueCOUGAR-X.
### Characteristics

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Symbol</th>
<th>Test conditions</th>
<th>Typ.</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum trigger pulse width</td>
<td></td>
<td></td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Turn-On time</td>
<td>$t_{ON}$</td>
<td>$R = 2 \text{ kOhm (Figure 20), internal output voltage 5V, } I_F = 16\text{mA}$</td>
<td>2</td>
<td>us</td>
</tr>
<tr>
<td>Storage time</td>
<td>$t_S$</td>
<td>$R = 2 \text{ kOhm (Figure 20), internal output voltage 5V, } I_F = 16\text{mA}$</td>
<td>44</td>
<td></td>
</tr>
<tr>
<td>Turn-Off time</td>
<td>$t_{OFF}$</td>
<td></td>
<td>95</td>
<td></td>
</tr>
</tbody>
</table>

Notes: Inputs with highest tolerance against interferences.

- **High level**: +11 to +24V
- **Low level**: 0 to +8V
- **Threshold**
  - (Low --> High / High --> Low): 10V +/- 1V
- $I_{max}$: 16mA

---

#### 10.2.2.3 Characteristics of the digital outputs

![Diagram](image1)

**Figure 19: DigIn mvBlueCOUGAR-X-POE-I**

---

#### 10.2.2.3 Characteristics of the digital outputs

![Diagram](image2)

**Figure 20: Output switching times (standard)**
### Characteristics

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Typ.</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_{ON}$</td>
<td>20</td>
<td>(\mu)s</td>
</tr>
<tr>
<td>$t_{R}$</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>$t_{F}$</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>$t_{OFF}$</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>High level</td>
<td>+12.0V to +24V</td>
<td></td>
</tr>
<tr>
<td>Low level</td>
<td>max. +1V</td>
<td></td>
</tr>
<tr>
<td>$I_{out}$</td>
<td>4x 0.7A max. short circuit</td>
<td></td>
</tr>
</tbody>
</table>

**Notes**

Each output has a short circuit protection between 1 A and 1.7 A (generally 1.3 A). So if you combine two outputs with one load, the short circuit protection can have an effect.

### Features

- Undervoltage detection
- Current limitation
- Overtemperature protection
- Short circuit protection

### Measurement conditions

$V_{CC} = 24 \text{ V}; -25 \degree \text{C} < T_J < 125 \degree \text{C}, R_L = 48 \text{ Ohm}$

10.2.2.4 M12 x-coded 100/1000 MBit Ethernet (option POE-I Rev. 2.00)

![Diagram of DigOut mvBlueCOUGAR-X POE-I](image)

**Figure 21: DigOut mvBlueCOUGAR-X POE-I**

### Pinout

<table>
<thead>
<tr>
<th>Pin</th>
<th>Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MDX0+</td>
</tr>
<tr>
<td>2</td>
<td>MDX0-</td>
</tr>
<tr>
<td>3</td>
<td>MDX1+</td>
</tr>
</tbody>
</table>

![Diagram of M12 x-coded 8-pin Ethernet connector pin numbering (female)](image)

**Figure 22: M12 x-coded 8-pin Ethernet connector pin numbering (female)**
10.3 mvBlueCOUGAR-XD

10.3.1 Dimensions

10.3.1.1 Standard model (xx1x)

<table>
<thead>
<tr>
<th>Lens protrusion</th>
<th>C-Mount</th>
<th>CS-Mount</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>8 mm (9.5 mm with max. Ø 20 mm)</td>
<td>6 mm</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Length</th>
<th>CMOSIS sensors</th>
<th>CCD sensors</th>
<th>Other sensors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>7 mm</td>
<td>11 mm</td>
<td>5 mm</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Z</th>
<th>C-Mount</th>
<th>CS-Mount</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.526 mm (in air)</td>
<td>12.5 mm (in air)</td>
<td></td>
</tr>
</tbody>
</table>
10.3.2 Connectors

---

**Figure 24**: mvBlueCOUGAR-XD mounting holes

**Figure 25**: Connectors mvBl1031ueCOUGAR-XD1031 with M42 mount

**Figure 26**: mvBlueCOUGAR-XD1031 mounting holes
10.3.2.1 Circular connector male

Figure 27: 12-pin (male; top view), digital I/O, power; orientation mvBlueCOUGAR-XD

<table>
<thead>
<tr>
<th>Pin</th>
<th>Signal</th>
<th>Digital I/O color code of power supply MV-DC1201 BCSX IO; Rev. smaller than 2</th>
<th>Digital I/O color code of power supply MV-DC1201 BCSX IO; Rev. 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GND</td>
<td>GND</td>
<td>gray-pink</td>
</tr>
<tr>
<td>2</td>
<td>12V - 24V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>DigOut3 (wxPropView numbering: line3)</td>
<td>white</td>
<td>white</td>
</tr>
<tr>
<td>4</td>
<td>Opto DigIn0 (line4)</td>
<td>brown</td>
<td>brown</td>
</tr>
<tr>
<td>5</td>
<td>DigOut2 (line2)</td>
<td>green</td>
<td>green</td>
</tr>
<tr>
<td>6</td>
<td>DigOut0 (line0)</td>
<td>yellow</td>
<td>yellow</td>
</tr>
<tr>
<td>7</td>
<td>Opto GND</td>
<td>gray</td>
<td>gray</td>
</tr>
<tr>
<td>8</td>
<td>RS232_RX</td>
<td>pink</td>
<td>pink</td>
</tr>
<tr>
<td>9</td>
<td>RS232_TX</td>
<td>blue</td>
<td>blue</td>
</tr>
<tr>
<td>10</td>
<td>+24V_HSD (12..24 V supply for the outputs)</td>
<td>red</td>
<td>red</td>
</tr>
<tr>
<td>11</td>
<td>Opto DigIn1 (line5)</td>
<td>black</td>
<td>black</td>
</tr>
<tr>
<td>12</td>
<td>DigOut1 (line1)</td>
<td>violet</td>
<td>violet</td>
</tr>
</tbody>
</table>

Connector (camera side): SAMWOO SNH-10-12 (RPCB) or equivalent
Plug (matching cable plug): Hirose HR10A-10P-12S (01) or equivalent

We recommend to use the connection cable Pinning of cable KS-BCX-HR12 (p. 70).

10.3.2.2 Circular connector female

Figure 28: 12-pin (female; top view), digital I/O
<table>
<thead>
<tr>
<th>Pin.</th>
<th>Signal</th>
<th>Description</th>
<th>Color of cutted end of cable KS-BCXD-HR12</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Opto DigIn2 (line6)</td>
<td>Additional input with the same characteristics as the inputs of the male connector</td>
<td>blue</td>
</tr>
<tr>
<td>2</td>
<td>Opto GND</td>
<td>Ground of DigIn2 and DigIn3</td>
<td>red</td>
</tr>
<tr>
<td>3</td>
<td>Opto DigIn3 (line7)</td>
<td>Additional inputs with the same characteristics as the inputs of the male connector</td>
<td>pink</td>
</tr>
<tr>
<td>4</td>
<td>Focus+</td>
<td>Focus control for motorized lenses (12V PWM)</td>
<td>gray</td>
</tr>
<tr>
<td>5</td>
<td>Focus-</td>
<td></td>
<td>yellow</td>
</tr>
<tr>
<td>6</td>
<td>Zoom+</td>
<td>Zoom control for motorized lenses</td>
<td>green</td>
</tr>
<tr>
<td>7</td>
<td>Zoom-</td>
<td></td>
<td>brown</td>
</tr>
<tr>
<td>8</td>
<td>Iris+</td>
<td>Lens aperture control for lenses with three motors</td>
<td>white</td>
</tr>
<tr>
<td>9</td>
<td>Iris-</td>
<td></td>
<td>black</td>
</tr>
<tr>
<td>10</td>
<td>Channel4+</td>
<td>Optional fourth channel</td>
<td>light blue</td>
</tr>
<tr>
<td>11</td>
<td>Channel4-</td>
<td></td>
<td>orange</td>
</tr>
<tr>
<td>12</td>
<td>GND</td>
<td>Ground</td>
<td>violet</td>
</tr>
</tbody>
</table>

Connector (camera side): SAMWOO SN-10-12 (RPCB) or equivalent
Plug (matching cable plug): Hirose HR10A-10P-12P (01) or equivalent

10.3.2.3 Characteristics of the digital inputs

10.3.2.3.1 Delay

![Figure 29: Input switching times](image)

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Symbol</th>
<th>Test conditions</th>
<th>Typ.</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum trigger pulse width</td>
<td></td>
<td></td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Turn-On time</td>
<td>$t_{ON}$</td>
<td>$R = 2 \text{kOhm (Figure 31), internal output voltage } 5V, \text{ if } I_F = 16mA$</td>
<td>2</td>
<td>us</td>
</tr>
<tr>
<td>Storage time</td>
<td>$t_S$</td>
<td></td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Turn-Off time</td>
<td>$t_{OFF}$</td>
<td></td>
<td>40</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Notes</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>High level</td>
<td></td>
<td></td>
<td>+3 to +24V</td>
<td></td>
</tr>
</tbody>
</table>
10.3.2.4 Characteristics of the digital outputs

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Symbol</th>
<th>Typ.</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turn-On time</td>
<td>t\text{ON}</td>
<td>20</td>
<td>\text{us}</td>
</tr>
<tr>
<td>Rise time</td>
<td>t\text{R}</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Fall time</td>
<td>t\text{F}</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Turn-Off time</td>
<td>t\text{OFF}</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>High level</td>
<td></td>
<td>+12.0V to +24V</td>
<td></td>
</tr>
<tr>
<td>Low level</td>
<td></td>
<td>max. +1V</td>
<td></td>
</tr>
<tr>
<td>t\text{out}</td>
<td></td>
<td>4x 0.7A max. short circuit</td>
<td></td>
</tr>
</tbody>
</table>

Notes: Each output has a short circuit protection between 1 A and 1.7 A (generally 1.3 A). So if you combine two outputs with one load, the short circuit protection can have an effect.
Features
- undervoltage detection
- current limitation
- overtemperature protection
- short circuit protection

Measurement conditions: $V_{CC} = 24\,V; -25\, ^{\circ}\, C < T_J < 125\, ^{\circ}\, C, R_L = 48\, \text{Ohm}$

10.3.2.5 Example circuit 1: High-side switch uses power supply of the camera

In this case you have to bridge pin 2 (external power supply) with pin 10 (the supply of the outputs):

10.3.2.6 Example circuit 2: High-side switch uses external (higher) power supply

Following figure shows, how to connect, for example, high power flash LEDs in series at the direct drive output at Pin 3:
A 24V power supply is connected at Pin 10. To protect the LEDs a series resistor is needed, which can be calculated in this way:

**Voltage:**

\[ U_{HS} - (n \times U_F) = U_D \]

**Series resistor:**

\[ R = \frac{U_D}{0.7A} \]

**Power:**

\[ P = \text{dutycycle} \times U_D \times I \]

The duty cycle is a coefficient \(< 1\), which defines the power of the resistor. It is the ratio of the time period while current flows compared to the total time. If the current always flows, "\( P = U \times I \)" will be the determining factor. If the current only flows a hundredth of time (duty cycle 1%), a resistor with a hundredth of power will be enough.

**Note**

For the connected LEDs and the ground of the external power supply, you have to use the ground of the camera at Pin 1.

**Attention**

The series resistor has to handle the power, which depends on the duty cycle and the frequency.

10.3.2.7 RJ45 100/1000 MBit Ethernet
The Ethernet signals are galvanically isolated from the camera electronics and camera housing.

### 10.3.2.8 RJ45 LED states

<table>
<thead>
<tr>
<th>States</th>
<th>LED1</th>
<th>LED2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power on</td>
<td>Green light on</td>
<td>Yellow light on</td>
</tr>
<tr>
<td>1Gb transmission</td>
<td>Green light on</td>
<td>Yellow blink on</td>
</tr>
<tr>
<td>100Mb transmission</td>
<td>Green light off</td>
<td>Yellow blink on</td>
</tr>
</tbody>
</table>

### 10.3.2.9 Video iris

This pin configuration is compliant to the JEITA standard:

<table>
<thead>
<tr>
<th>Pin</th>
<th>Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12V iris power supply</td>
</tr>
<tr>
<td>2</td>
<td>NC</td>
</tr>
<tr>
<td>3</td>
<td>DC / video iris control</td>
</tr>
<tr>
<td>4</td>
<td>Ground</td>
</tr>
</tbody>
</table>

**Warning**

Please check the wiring of your video iris lenses, because there are brands which uses the standard 4-pin connector with non-standard wiring.
10.3.3 Signal LED

The mvBlueCOUGAR-XD features a RGB LED. There are following states:

10.3.3.1 Typical start sequence (LLA)

<table>
<thead>
<tr>
<th>State</th>
<th>LED color</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPGA loaded</td>
<td>White on</td>
</tr>
<tr>
<td>Self-test running</td>
<td>Green on</td>
</tr>
<tr>
<td>Waiting for Ethernet connection</td>
<td>White blink</td>
</tr>
<tr>
<td>LLA (auto IP)</td>
<td>Green blink</td>
</tr>
<tr>
<td>LLA got</td>
<td>Green on</td>
</tr>
<tr>
<td>Waiting for client</td>
<td>Blue on</td>
</tr>
</tbody>
</table>

See also [LLA](p. 59)

10.3.3.2 Typical start sequence (DHCP)

<table>
<thead>
<tr>
<th>State</th>
<th>LED color</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPGA loaded</td>
<td>White on</td>
</tr>
<tr>
<td>Self-test running</td>
<td>Green on</td>
</tr>
<tr>
<td>Waiting for Ethernet connection</td>
<td>White blink</td>
</tr>
<tr>
<td>DHCP request</td>
<td>Blue blink</td>
</tr>
<tr>
<td>DHCP got</td>
<td>Bright Blue</td>
</tr>
<tr>
<td>Waiting for client</td>
<td>Blue on</td>
</tr>
</tbody>
</table>

See also [DHCP](p. 59)

10.3.3.3 Typical start sequence (Fixed IP)

<table>
<thead>
<tr>
<th>State</th>
<th>LED color</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPGA loaded</td>
<td>White on</td>
</tr>
<tr>
<td>Self-test running</td>
<td>Green on</td>
</tr>
<tr>
<td>Waiting for Ethernet connection</td>
<td>White blink</td>
</tr>
<tr>
<td>Fixed IP</td>
<td>Bright Green</td>
</tr>
<tr>
<td>Waiting for client</td>
<td>Blue on</td>
</tr>
</tbody>
</table>

See also [Fixed IP](p. 59)

10.3.3.4 General behavior

<table>
<thead>
<tr>
<th>State</th>
<th>LED color</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting wxPropView</td>
<td></td>
</tr>
</tbody>
</table>

See also [wxPropView](p. 115)
10.4 mvBlueCOUGAR-3X

10.4.1 Dimensions

<table>
<thead>
<tr>
<th>front view</th>
<th>side view</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Front View" /></td>
<td><img src="image2" alt="Side View" /></td>
</tr>
</tbody>
</table>

Figure 37: Connectors mvBlueCOUGAR-3X

| Size of body (w x h x l) | 39.8 x 39.8 x 50.5 mm |

10.4.1.1 Connectors

Please have a look at chapter connectors of the standard model (p.69) for the characteristics of the standard connectors or at chapter connectors of the POE-I model (p.77) for the characteristics of the POE-I connectors.

The 3-head-model uses the same back panel as the standard mvBlueCOUGAR-X. Two version are available for the 3-head-model:

10.4.1.1.1 Standard

![Back View](image3)
10.4.1.2 Sensor heads

The three sensor heads of the mvBlueCOUGAR-3X are connected via a starrflex connection of 200 mm length. The sensor heads feature a S-mount lensholder. For both available sensors there are different sensor heads:

10.4.1.2.1 -120a

See also

-120a (p. 424)
10.4.2 Model specific features

In contrast to the standard mvBlueCOUGAR-X model:

<table>
<thead>
<tr>
<th>Features</th>
<th>mvBlueCOUGAR-3Xxxx -G/-C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Environment</td>
<td></td>
</tr>
<tr>
<td>Ambient Temperature</td>
<td></td>
</tr>
<tr>
<td>Operation</td>
<td>0..40 deg C / 30 to 80% RH</td>
</tr>
<tr>
<td>Power supply</td>
<td></td>
</tr>
<tr>
<td>$P_{\text{max}}$</td>
<td>3.6 W according to the used sensor</td>
</tr>
</tbody>
</table>

10.5 Summary of components

<table>
<thead>
<tr>
<th>Features</th>
<th>mvBlueCOUGAR-Xxxx -G/-C</th>
<th>mvBlueCOUGAR-AR-XDxxx -G/-C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard version</td>
<td>&quot;-PLC&quot; option (inputs with PLC levels only)</td>
<td>Power over Ethernet (p. 242) (not available for all products)</td>
</tr>
<tr>
<td>Image Memory</td>
<td>64 MByte</td>
<td>256 MByte</td>
</tr>
<tr>
<td>Proc. Memory</td>
<td>64 MByte</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>----------</td>
<td></td>
</tr>
<tr>
<td>ADC resolution</td>
<td>CCD: 14 bits (14/12/8 bit transfer)/CMOS: ADC on chip</td>
<td></td>
</tr>
<tr>
<td>Inputs</td>
<td>2</td>
<td>2 (PLC levels only)</td>
</tr>
<tr>
<td>Type</td>
<td>opto-isolated with current limiters</td>
<td></td>
</tr>
<tr>
<td>Outputs</td>
<td>4</td>
<td>2 4 4</td>
</tr>
<tr>
<td>Type</td>
<td>high-side solid state relay</td>
<td>opto-isolated high-side solid state relay</td>
</tr>
<tr>
<td>Ethernet</td>
<td>100/1000 Mbit on RJ45</td>
<td>100/1000 Mbit on M12 x-coded</td>
</tr>
<tr>
<td>Optics</td>
<td>Lens Mount (Focal Distance)</td>
<td>C-mount (17.526 mm in air), CS-mount (12.5 mm in air), optional S-mount</td>
</tr>
<tr>
<td>Lens Protrusion</td>
<td>6 .. 8 mm</td>
<td></td>
</tr>
<tr>
<td>Environment</td>
<td>Ambient Temperature</td>
<td></td>
</tr>
<tr>
<td>Operation</td>
<td>0..45 deg C / 30 to 80% RH</td>
<td>0..45 deg C / 0 to 100% RH</td>
</tr>
<tr>
<td>Storage</td>
<td>-20..60 deg C / 20 to 90% RH</td>
<td>-20..60 deg C / 0 to 100% RH</td>
</tr>
<tr>
<td>Additional features</td>
<td>temperature sensor</td>
<td>temperature sensor video iris lens control (zoom, focus, iris)</td>
</tr>
<tr>
<td>Weight without lens</td>
<td>approx. 110 g</td>
<td>approx. 127 g</td>
</tr>
<tr>
<td>Power supply</td>
<td>DC 10...28 V</td>
<td>P_{max}</td>
</tr>
<tr>
<td>Max. current consumption</td>
<td>1.0 A @ 10 V</td>
<td></td>
</tr>
<tr>
<td>Reverse polarity protection</td>
<td>up to 28 V</td>
<td></td>
</tr>
</tbody>
</table>
## 10.6 Summary of advanced features

The following table shows an excerpt of the advanced features. For the complete list, please use the "Product Comparison" in the "Appendix C".

<table>
<thead>
<tr>
<th>Advanced features</th>
<th>mvBlueCOUGAR-X1xx</th>
<th>mvBlueCOUGAR-X2xx</th>
<th>mvBlueCOUGAR-XDxxx</th>
</tr>
</thead>
<tbody>
<tr>
<td>MultiFrame / SingleFrame / Continuous (p.170)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Trigger Overlap (p.176)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Temperature settings (p.387)</td>
<td>upper limit (&lt;= 255 deg C) lower limit (&gt;= 0 deg C)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>User Sets (p.195)</td>
<td>5 config sets storable</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>User Data (p.379)</td>
<td>512 byte of EEPR OM data</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>FPGA based look-up table (LUT) (p.182)</td>
<td>on the fly on the camera</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Timestamp (latch/reset) (p.194)</td>
<td>microsecond precise camera individual 64 bit counter</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Flat Field Correction (FFC) (p.294)</td>
<td>full AOI, 14 bit to 14 bit (12 bit coeff.), pixel to pixel</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td>Enhanced I/O functionality acc. to SFNC (counter, timer) (p.173)</td>
<td>timers can be used for pulse width modulation</td>
<td>4 counters, 2 timers</td>
<td>4 counters, 2 timers</td>
</tr>
<tr>
<td>Auto Exposure Control (AEC) (p.170)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Auto Gain Control (AGC) (p.170)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>White balance (p.144)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Auto white balance (p.144)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>RGB-to-YUV conversion</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Color correction matrix (p.177)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Timestamp for inter-camera synchronization</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Image recorder with pretrigger option (p.249)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Linescan mode (p. 249)</td>
<td>mvBlueCOUGA---R-Xx02e (1.3 Mpix [1280 x 1024]) (p. 641)</td>
<td>mvBlueCOUGA---GAR-Xx02eGE (1.3 Mpix [1280 x 1024]) (p. 647)</td>
<td>mvBlueCOUGA---R-Xx04e (1.9 Mpix [1600 x 1200]) (p. 655)</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------</td>
<td>---------------------------------------------------------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>Logic gates (p. 176)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Chunk Data (p. 179)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Introducing multicasting (p. 200)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Primary application switchover (p. 419)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Frame average (p. 313)</td>
<td>-</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td>Horizontal mirror (CCD)</td>
<td>-</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
11 Sensor overview

11.1 Image data flow

The following block diagrams show the data flow of the image data after being read from the sensor chip in the camera. The transfer latency (stream controller + frame buffer + resend buffer + packet generator) of the data is less than 100 us.

![Figure 1: Block diagram gray scale sensors](image1)

![Figure 2: Block diagram color sensors](image2)

11.2 CCD sensors

The CCD sensors are highly programmable imaging modules which will, for example, enable the following type of applications...
<table>
<thead>
<tr>
<th>Sensors</th>
<th>0.3 Mpix (x20a)</th>
<th>0.3 Mpix (x20b)</th>
<th>0.5 Mpix (x20d)</th>
<th>1.2 Mpix (x22)</th>
<th>1.4 Mpix (x23)</th>
<th>1.9 Mpix (x24)</th>
<th>5.1 Mpix (x25a)</th>
<th>5.1 Mpix (x225)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor supplier</td>
<td>Sony</td>
<td>Sony</td>
<td>Sony</td>
<td>Sony</td>
<td>Sony</td>
<td>Sony</td>
<td>Sony</td>
<td>Sony</td>
</tr>
<tr>
<td>Sensor name</td>
<td>ICX424 AL/AQ</td>
<td>ICX414 AL/AQ</td>
<td>ICX415 AL/AQ</td>
<td>ICX445 ALA/AQ</td>
<td>ICX267 AL/AQ</td>
<td>ICX274 AL/AQ</td>
<td>ICX655 ALA/AQ</td>
<td>ICX625 ALA/AQ</td>
</tr>
<tr>
<td>Resolution</td>
<td>640 x 480 gray scale or RGB</td>
<td>640 x 480 gray scale or RGB</td>
<td>776 x 580 gray scale or RGB</td>
<td>1280 x 960 gray scale or RGB</td>
<td>1360 x 1024 gray scale or RGB</td>
<td>1600 x 1200 gray scale or RGB</td>
<td>2448 x 2050 gray scale or RGB</td>
<td>2448 x 2050 gray scale or RGB</td>
</tr>
<tr>
<td>Pixel clock [MHz]</td>
<td>24.54 / 32 / 40</td>
<td>24.54 / 32 / 40</td>
<td>29.5 / 48 / 50</td>
<td>36 / 56</td>
<td>28 / 56</td>
<td>36 / 56 / 65</td>
<td>60 / 66</td>
<td>60 / 66</td>
</tr>
<tr>
<td>Max. F-PS</td>
<td>104</td>
<td>87</td>
<td>31</td>
<td>30</td>
<td>28</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>Frame rate ex-actness</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADC (on sensor board) resolution [bit]</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>Programmable analog gain and offset</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Frame integrating progressive scan sensor (no interlaced problems!)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>High resolution</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>High color reproduction (for color version)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>--------------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low dark current</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Continuous variable-speed shutter</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overlapped mode (p. 241)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low smear</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Excellent anti-blooming characteristics</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Programmable integration time</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trigger (HW / SW)</td>
<td>X / X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flash control output, synchronous to integration period</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windowing and partial scan</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Power consumption @ 24V

<table>
<thead>
<tr>
<th>Model</th>
<th>Power Consumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>mvBlueC</td>
<td>OUGA-R-Xx20a (0.3 Mpix [640 x 480])</td>
</tr>
<tr>
<td>mvBlueC</td>
<td>OUGA-R-Xx20b (0.3 Mpix [640 x 480])</td>
</tr>
<tr>
<td>mvBlueC</td>
<td>OUGA-R-Xx20d (0.5 Mpix [776 x 580])</td>
</tr>
<tr>
<td>mvBlueC</td>
<td>OUGA-R-Xx22 (1.2 Mpix [1280 x 960])</td>
</tr>
<tr>
<td>mvBlueC</td>
<td>OUGA-R-Xx23 (1.4 Mpix [1360 x 1024])</td>
</tr>
<tr>
<td>mvBlueC</td>
<td>OUGA-R-Xx24 (1.9 Mpix [1600 x 1200])</td>
</tr>
<tr>
<td>BlueC</td>
<td>OUGA-R-Xx25a (5.1 Mpix [2448 x 2050])</td>
</tr>
<tr>
<td>BlueC</td>
<td>OUGA-R-Xx25b (5.1 Mpix [2448 x 2050])</td>
</tr>
</tbody>
</table>

### More specific data

- mvBlueCOUGAR-XD specific CCD sensors

The CCD sensors are highly programmable imaging modules which will, for example, enable the following type of applications...
<table>
<thead>
<tr>
<th>Sensors</th>
<th>3 Mpix (-X→D124a)</th>
<th>6 Mpix (-X→D126)</th>
<th>6 Mpix (-X→D126a)</th>
<th>9.2 Mpix (-XD129)</th>
<th>9.2 Mpix (-XD129a)</th>
<th>12.1 Mpix (-XD1212a)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor supplier</td>
<td>Sony</td>
<td>Sony</td>
<td>Sony</td>
<td>Sony</td>
<td>Sony</td>
<td>Sony</td>
</tr>
<tr>
<td>Sensor name</td>
<td>ICX674 AL←G/AQG</td>
<td>ICX694 AL←G/AQG</td>
<td>ICX695 AL←G/AQG</td>
<td>ICX814 AL←G/AQG</td>
<td>ICX815 AL←G/AQG</td>
<td>ICX834 AL←G/AQG</td>
</tr>
<tr>
<td>Resolution</td>
<td>1936 x 1460</td>
<td>2752 x 2208</td>
<td>2752 x 2208</td>
<td>3384 x 2712</td>
<td>3384 x 2712</td>
<td>4248 x 2836</td>
</tr>
<tr>
<td>Pixel clock [MHz]</td>
<td>56 / 66</td>
<td>56 / 66</td>
<td>56</td>
<td>56 / 66</td>
<td>56</td>
<td>36 / 56</td>
</tr>
<tr>
<td>Max. FPS</td>
<td>64</td>
<td>33</td>
<td>14</td>
<td>22</td>
<td>9.5</td>
<td>7.5</td>
</tr>
<tr>
<td>Frame rate exactness (p.[170])</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADC (on sensor board) resolution / Out [bit]</td>
<td>14 / 8 .. 16</td>
<td>14 / 8 .. 16</td>
<td>14 / 8 .. 16</td>
<td>14 / 8 .. 16</td>
<td>14 / 8 .. 16</td>
<td>14 / 8 .. 16</td>
</tr>
<tr>
<td>Programmable analog gain and offset</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Frame integrating progressive scan sensor (no interlaced problems!)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>High resolution</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>High color reproducibility (for color version)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low dark current</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Continuous variable-speed shutter</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overlapped mode (p.[241])</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low smear</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Excellent antiblooming characteristics</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Programmable integration time from usec to sec.</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trigger (HW / SW)</td>
<td>X / X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature</td>
<td>Value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flash control output, synchronous to integration period</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windowing and partial scan</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power consumption @ 24V with 2x GigE lanes</td>
<td>approx. 6.← 9 W (2 taps, 56 MHz)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>approx. 7.← 7 W (4 taps, 56 MHz)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>approx. 8.← 6 W (2 taps, 66 MHz)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>approx. 8.← 8 W (2 taps, 66 MHz)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>approx. 8.14 W (4 taps, 56 MHz)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>More specific data</td>
<td>mvBlueC← OUGAR← XD124a (3 Mpix [1936 x 1460]) (p.484)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>mvBlue← COUGA← R-XD126 (6 Mpix [2752 x 2208]) (p.513)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>mvBlueC← OUGAR← XD126a (6 Mpix [2752 x 2208]) (p.523)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>mvBlueC← OUGAR← XD129 (9.2 Mpix [3384 x 2712]) (p.533)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>mvBlueC← OUGAR-X← D129a (9.2 Mpix [3384 x 2712]) (p.543)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>mvBlue← COUGA← R-XD1212a (12.1 Mpix [4248 x 2836]) (p.553)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 11.2.2 Negative gain

The mvBlueCOUGAR-X cameras use a minimum needed basic gain for typical sensor output signals to get near the maximum ADC (p.234) (analog-to-digital converter) saturation limit (Figure 3). This guarantees a higher sensitivity. However, to take advantage of the whole ADC (p.234) range of the sensor, you can use "negative gain". Together with enough illumination, you can get a better signal-to-noise ratio given that this setting will affect the analog signal before it is processed by the ADC (p.234).

![Figure 3: Negative gain diagram](image)

**Note**

Depending on the value of the adjusted negative gain, it can be possible that single pixels pass over their signal limits. In the live image, you will have the effect that some pixels do not change their behavior and the whole image seems to be "frozen".
11.3 mvBlueCOUGAR-X specific CMOS sensors

The CMOS sensor modules incorporate the following features:

### 11.3.1 Sony Pregius

<p>| Sensor supplier | Sensor name | Resolution | Frame rate exactness | ADC resolution / Out [bit] | SNRmax [dB]| |
|-----------------|-------------|------------|----------------------|---------------------------|-------------|
| Sony            | IMX287      | 1456 x 544 | -                    | 12 / 16, 12, 8            | 43.3        | |
| Sony            | IMX273      | 1608 x 1104| -                    | 12 / 16, 12, 8            | 40.2        | |
| Sony            | IMX432      | 1608 x 1104| -                    | 12 / 16, 12, 8            | 49.5        | 40.5 |
| Sony            | IMX425      | 1608 x 1248| -                    | 12 / 16, 12, 8            | 40.3        | 40.3 |
| Sony            | IMX430      | 1632 x 1544| -                    | 12 / 16, 12, 8            | 40.2        | 40.2 |
| Sony            | IMX249      | 1936 x 1544| -                    | 12 / 16, 12, 8            | 40.2        | 40.2 |
| Sony            | IMX265      | 2064 x 2056| -                    | 12 / 16, 12, 8            | 40.2        | 40.2 |
| Sony            | IMX264      | 2464 x 2208| -                    | 12 / 16, 12, 8            | 40.2        | 40.2 |
| Sony            | IMX267      | 3216 x 2176| -                    | 12 / 16, 12, 8            | 40.2        | 40.2 |
| Sony            | IMX304      | 4112 x 3008| -                    | 12 / 16, 12, 8            | 40.2        | 40.2 |</p>
<table>
<thead>
<tr>
<th>Feature</th>
<th>DR (normal)</th>
<th>HDR (p. 363) [dB]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Progressive scan sensor (no interlaced problems!)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Rolling shutter</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Global shutter</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Trigger (HW / SW)</td>
<td>X / X</td>
<td>X / X</td>
</tr>
<tr>
<td>Pipeline global shutter in trigger mode (p. 241)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Linescan mode</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>High color reproducibility (for color version)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Programmable read-out timing with free capture windows and partial scan</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
### 11.3 mvBlueCOUGAR-X specific CMOS sensors

<table>
<thead>
<tr>
<th>Many trigger modes (free-running, software-triggered, hardware-triggered)</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Flash control output, synchronous to integration period</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
</tr>
</thead>
</table>

| Power consumption @ 24V | approx. 3.6 W POE-I: 3.8 W | approx. 3.6 W POE-I: 3.8 W | approx. tbd W | approx. tbd W | approx. 3.4 W POE-I: 3.8 W | approx. 3.8 W POE-I: 4.1 W | approx. 3.7 W POE-I: 4.1 W | approx. tbd W | approx. 4.3 W POE-I: 4.75 W | approx. 4.3 W |

| More specific data | mvBlueCOUGAR-X \( \text{[D]} \) 100f (0.4 Mpix \( \text{[728 x 544]} \)) \( \text{[583]} \) | mvBlueCOUGAR-X \( \text{[D]} \) 102f (1.6 Mpix \( \text{[1456 x 1088]} \)) \( \text{[586]} \) | mvBlueCOUGAR-X \( \text{[D]} \) 103f (1.8 Mpix \( \text{[1608 x 1248]} \)) \( \text{[579]} \) | mvBlueCOUGAR-X \( \text{[D]} \) 104f (2.4 Mpix \( \text{[1936 x 1544]} \)) \( \text{[584]} \) | mvBlueCOUGAR-X \( \text{[D]} \) 105f (3.2 Mpix \( \text{[2064 x 1608]} \)) \( \text{[591]} \) | mvBlueCOUGAR-X \( \text{[D]} \) 106f (4.1 Mpix \( \text{[2316 x 1888]} \)) \( \text{[598]} \) | mvBlueCOUGAR-X \( \text{[D]} \) 107f (4.75 Mpix \( \text{[2608 x 2064]} \)) \( \text{[605]} \) | mvBlueCOUGAR-X \( \text{[D]} \) 108f (5.1 Mpix \( \text{[2916 x 2344]} \)) \( \text{[612]} \) | mvBlueCOUGAR-X \( \text{[D]} \) 109f (8.9 Mpix \( \text{[4112 x 3308]} \)) \( \text{[609]} \) | mvBlueCOUGAR-X \( \text{[D]} \) 110f (12.4 Mpix \( \text{[4112 x 3008]} \)) |

---

1 Measured accord. to EMVA1288 with gray scale version of the camera

#### 11.3.1.1 Sony Starvis

<table>
<thead>
<tr>
<th>Sensors</th>
<th>20.5 Mpix (-1020)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor supplier</td>
<td>Sony</td>
</tr>
<tr>
<td>Sensor name</td>
<td>IMX183</td>
</tr>
<tr>
<td>Res.</td>
<td>5544 x 3692</td>
</tr>
<tr>
<td>Sensor size</td>
<td>1&quot;</td>
</tr>
<tr>
<td>Max. FPS (in free-running full frame mode)</td>
<td>8.82 [burst mode ([270] required)] / 5.8 [streaming]</td>
</tr>
<tr>
<td><strong>Frame rate exactness</strong> (p. [170])</td>
<td>-</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>ADC resolution / Out</td>
<td>12 / 12, 10, 8</td>
</tr>
<tr>
<td>SNR$_{\text{max}}$ [dB]$^{\dagger}$</td>
<td>tbd</td>
</tr>
<tr>
<td>DR (normal / HDR (p. [362])) [dB]$^{\dagger}$</td>
<td>tbd</td>
</tr>
<tr>
<td>Rolling shutter</td>
<td>X</td>
</tr>
<tr>
<td>Global shutter</td>
<td>-</td>
</tr>
<tr>
<td>Global Reset</td>
<td>-</td>
</tr>
<tr>
<td>Trigger (HW / SW)</td>
<td>- / X</td>
</tr>
<tr>
<td>Pipelined global shutter in trigger mode (p. [241])</td>
<td>-</td>
</tr>
<tr>
<td>Linescan mode</td>
<td>-</td>
</tr>
<tr>
<td>High color reproductivity (for color version)</td>
<td>X</td>
</tr>
<tr>
<td>Power consumption (since FW 2.5.146) [W]</td>
<td>approx. tbd</td>
</tr>
<tr>
<td><strong>More specific data</strong></td>
<td>mvBlueCOUGAR-X1020 (20.5 Mpix [5544 x 3692]) (p. [621])</td>
</tr>
</tbody>
</table>

$^{\dagger}$ Measured accord. to EMVA1288 with gray scale version of the camera

### 11.3.1.2 Sony Polarsens

<table>
<thead>
<tr>
<th><strong>Sensors</strong></th>
<th>5.1 Mpix (-105p)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sensor supplier</strong></td>
<td>Sony</td>
</tr>
<tr>
<td><strong>Sensor name</strong></td>
<td>IMX250_POL</td>
</tr>
<tr>
<td><strong>Res.</strong></td>
<td>2464 x 2056</td>
</tr>
<tr>
<td><strong>gray scale / RGB</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Sensor size</strong></td>
<td>2/3”</td>
</tr>
<tr>
<td><strong>Max. FPS (in free-running full frame mode)</strong></td>
<td>tbd [burst mode (p. [270]) required] / tbd [streaming]</td>
</tr>
<tr>
<td><strong>Frame rate exactness</strong> (p. [170])</td>
<td>-</td>
</tr>
<tr>
<td>ADC resolution / Out</td>
<td>12 / 12, 10, 8</td>
</tr>
<tr>
<td>SNR$_{\text{max}}$ [dB]$^{\dagger}$</td>
<td>tbd</td>
</tr>
<tr>
<td>DR (normal / HDR (p. [362])) [dB]$^{\dagger}$</td>
<td>tbd</td>
</tr>
<tr>
<td>Rolling shutter</td>
<td>-</td>
</tr>
<tr>
<td>Global shutter</td>
<td>X</td>
</tr>
<tr>
<td>Global Reset</td>
<td>-</td>
</tr>
<tr>
<td>Trigger (HW / SW)</td>
<td>X / X</td>
</tr>
<tr>
<td>Pipelined global shutter in trigger mode (p. [241])</td>
<td>X</td>
</tr>
<tr>
<td>Linescan mode</td>
<td>-</td>
</tr>
<tr>
<td>High color reproductivity (for color version)</td>
<td>X</td>
</tr>
<tr>
<td>Power consumption (since FW 2.5.146) [W]</td>
<td>approx. 3.6</td>
</tr>
<tr>
<td><strong>More specific data</strong></td>
<td>mvBlueCOUGAR-X105p (5.1 Mpix [2464 x 2056]) (p. [625])</td>
</tr>
</tbody>
</table>

$^{\dagger}$ Measured accord. to EMVA1288 with gray scale version of the camera

### 11.3.2 Aptina, CMOSIS, e2v

<table>
<thead>
<tr>
<th><strong>Sensors</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.4 Mpix (-x00w)</td>
<td>1.2 Mpix (-x02b)$^{\dagger}$</td>
</tr>
<tr>
<td>1.2 Mpix (-x02d)$^{\dagger}$</td>
<td>1.3 Mpix (-x02e)$^{\dagger}$</td>
</tr>
<tr>
<td>1.3 Mpix (-x02e--GE)$^{\dagger}$</td>
<td>2.2 Mpix (-x04)$^{\dagger}$</td>
</tr>
<tr>
<td>4.2 Mpix (-x04b)</td>
<td>1.9 Mpix (-x04e)$^{\dagger}$</td>
</tr>
<tr>
<td>5 Mpix (-x05)</td>
<td>10 Mpix (-x010)</td>
</tr>
</tbody>
</table>

MATRIX VISION GmbH
<table>
<thead>
<tr>
<th>Sensor supplier</th>
<th>Aptina</th>
<th>Aptina</th>
<th>Aptina</th>
<th>e2v</th>
<th>e2v</th>
<th>CM OSIS</th>
<th>CM OSIS</th>
<th>e2v</th>
<th>Aptina</th>
<th>Aptina</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor name</td>
<td>MT9--- V034</td>
<td>MT9--- M021</td>
<td>MT9--- M034</td>
<td>E--- V76--- C560</td>
<td>E--- V76--- C661</td>
<td>CM--- V2000</td>
<td>CM--- V4000</td>
<td>E--- V76--- C570</td>
<td>MT9--- P031</td>
<td>MT9--- J003</td>
</tr>
<tr>
<td>Resolution</td>
<td>752 x 480 gray scale or RGB</td>
<td>1280 x 960 gray scale or RGB</td>
<td>1280 x 1024 gray scale or RGB</td>
<td>1280 x 1024 gray scale or RGB</td>
<td>2048 x 1088 gray scale or RGB</td>
<td>2048 x 1200 gray scale or RGB</td>
<td>1600 x 1444 gray scale or RGB</td>
<td>3856 x 2764 gray scale or RGB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensor size</td>
<td>1/3&quot;</td>
<td>1/3&quot;</td>
<td>1/3&quot;</td>
<td>1/1.8&quot;</td>
<td>1/1.8&quot;</td>
<td>2/3&quot;</td>
<td>1&quot;</td>
<td>1/1.8&quot;</td>
<td>1/2.5&quot;</td>
<td>1/2.3&quot;</td>
</tr>
<tr>
<td>Pixel clock [MHz]</td>
<td>27 / 40 / 50</td>
<td>40 / 66 / 74.25</td>
<td>40 / 66 / 74.25</td>
<td>85</td>
<td>85</td>
<td>40</td>
<td>40</td>
<td>85</td>
<td>40 / 98</td>
<td>81.25</td>
</tr>
<tr>
<td>Max. FPS (in freerunning full frame mode)</td>
<td>117</td>
<td>45.6</td>
<td>45.6</td>
<td>60</td>
<td>60</td>
<td>270 [burst mode (p.[270] required) / 106.5 [streaming] (-XD)</td>
<td>34.8 (-X)</td>
<td>149 [burst mode (p.[270] required) / 56.6 [streaming] (-XD)</td>
<td>18 (-X)</td>
<td>51</td>
</tr>
<tr>
<td>Frame rate exactness (p.[170])</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ADC resolution / Out [bit]</td>
<td>10 / 12, 10, 8</td>
<td>10 / 12, 10, 8</td>
<td>10 / 12, 10, 8</td>
<td>10 (10-8 companding) / 10, 8</td>
<td>10 (10-8 companding) / 10, 8</td>
<td>10 / 16, 12, 8 (-XD)</td>
<td>10 / 10, 8 (-X)</td>
<td>10 / 16, 12, 8 (-XD)</td>
<td>10 / 10, 8 (-X)</td>
<td>10 (10-8 companding) / 10, 8</td>
</tr>
<tr>
<td>S NRmax [dB]³</td>
<td>42.7</td>
<td>37.3</td>
<td>37.7</td>
<td>39.8</td>
<td>38.9</td>
<td>38.2</td>
<td>39.1</td>
<td>38.4</td>
<td>38.2</td>
<td>37.2</td>
</tr>
<tr>
<td>DR (normal / HDR (p.[363]) [dB]³</td>
<td>56.2 / &gt; 110</td>
<td>55.4 / &gt; 115</td>
<td>63.8 / &gt; 115</td>
<td>52.8 / &gt; 100</td>
<td>50.3 / &gt; 100</td>
<td>59.6 / &gt; 100</td>
<td>56.1 / &gt; 100</td>
<td>50.2 / &gt; 100</td>
<td>59.6 / &gt; 100</td>
<td>56.3 /</td>
</tr>
<tr>
<td>Feature</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>Progressive scan sensor (no interlaced problems!)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rolling shutter</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Global shutter</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Trigger (HW / SW)</td>
<td>X / X</td>
<td>X / X</td>
<td>X / X</td>
<td>X / X</td>
<td>X / X</td>
<td>X / X</td>
<td>X / X</td>
<td>X / X</td>
<td>X / X</td>
<td></td>
</tr>
<tr>
<td>Pipelined global shutter in trigger mode</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Linescan mode</td>
<td></td>
<td></td>
<td></td>
<td>available</td>
<td>available</td>
<td>available</td>
<td>available</td>
<td>available</td>
<td>available</td>
<td></td>
</tr>
<tr>
<td>High color reproducibility (for color version)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Programmable read-out timing with free capture windows and partial scan</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Many trigger modes (free-running, software triggered, hardware triggered)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
### 11.3 mvBlueCOUGAR-X specific CMOS sensors

<table>
<thead>
<tr>
<th>Power consumption @ 24V</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>approx. 2.5 W (-100w)</td>
<td>approx. 2.8 W (-102b)</td>
<td>approx. 3.2 W (-102d)</td>
<td>approx. 2.8 W (-102e)</td>
<td>approx. 3.5 W (-104)</td>
<td>approx. 3.5 W (-104b)</td>
<td>approx. 3 W (-104e)</td>
<td>approx. 3 W (-105)</td>
<td>approx. 3.3 W (-1010)</td>
<td></td>
</tr>
</tbody>
</table>

More specific data

<table>
<thead>
<tr>
<th>Sensors</th>
<th>Xx00w</th>
<th>Xx02d</th>
<th>Xx02e</th>
<th>Xx02eGE</th>
<th>Xx02e[D]x04</th>
<th>Xx04d</th>
<th>Xx04e</th>
<th>Xx05</th>
<th>Xx100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mpix</td>
<td>(752 x 480)</td>
<td>(1.2 Mpix)</td>
<td>(1.2 Mpix)</td>
<td>(1.3 Mpix)</td>
<td>(1.3 Mpix)</td>
<td>(1.3 Mpix)</td>
<td>(1.3 Mpix)</td>
<td>(5 Mpix)</td>
<td>(10 Mpix)</td>
</tr>
<tr>
<td>[p. 529]</td>
<td>[p. 533]</td>
<td>[p. 837]</td>
<td>[p. 641]</td>
<td>[p. 647]</td>
<td>[p. 661]</td>
<td>[p. 665]</td>
<td>[p. 669]</td>
<td>[p. 874]</td>
<td></td>
</tr>
</tbody>
</table>

1 No AEC with external trigger
2 Infrared enhanced
3 Measured accord. to EMVA1288 with gray scale version of the camera

**Note**

For further information about rolling shutter, please have a look at the practical report about rolling shutter on our website: [https://www.matrix-visions.com/tl_files/mvl1/Glossary/art__rolling_shutter_en.pdf](https://www.matrix-visions.com/tl_files/mvl1/Glossary/art__rolling_shutter_en.pdf)

For further information about image errors of image sensors, please have a look at *Correcting image errors of a sensor* (p. 286).

### 11.3.3 mvBlueCOUGAR-XD specific CMOS sensors

The CMOS sensor modules incorporate the following features:

<table>
<thead>
<tr>
<th>Sensors</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2 Mpix (-XD104)</td>
<td>4.2 Mpix (-XD04b)</td>
<td>2.4 Mpix (-XD104d)</td>
<td>3.2 Mpix (-XD105a)</td>
<td>5.1 Mpix (-XD107)</td>
<td>8.9 Mpix (-XD109b)</td>
<td>12.4 Mpix (-XD1012b)</td>
<td>31.5 Mpix (-XD1031)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/ (-X~D204)</td>
<td>/ (-X~D204)</td>
<td>/ (-X~D104a12)</td>
<td>/ (-X~D104b)</td>
<td>/ (-X~D104h)</td>
<td>/ (-X~D105a)</td>
<td>/ (-X~D105b)</td>
<td>/ (-X~D105c)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sensor supplier</th>
<th>CMO~SIS</th>
<th>CMO~SIS</th>
<th>Sony</th>
<th>Sony</th>
<th>Sony</th>
<th>Sony</th>
<th>Sony</th>
<th>Sony</th>
<th>Sony</th>
</tr>
</thead>
</table>

MATRIX VISION GmbH
<table>
<thead>
<tr>
<th>Sensor name</th>
<th>CMV2000</th>
<th>CMV4000</th>
<th>IMX174</th>
<th>IMX252</th>
<th>IMX250</th>
<th>IMX420</th>
<th>IMX267</th>
<th>IMX304</th>
<th>IMX342</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolution</td>
<td>2048 x 1088 gray scale or RGB</td>
<td>2048 x 1088 gray scale or RGB</td>
<td>1936 x 1216 gray scale or RGB</td>
<td>2064 x 1544 gray scale or RGB</td>
<td>2064 x 1544 gray scale or RGB</td>
<td>2056 x 1296 gray scale or RGB</td>
<td>2056 x 1296 gray scale or RGB</td>
<td>2208 x 1544 gray scale or RGB</td>
<td>2208 x 1544 gray scale or RGB</td>
</tr>
<tr>
<td>Sensor size</td>
<td>2/3”</td>
<td>1&quot;</td>
<td>1/1.2&quot;</td>
<td>1/1.8&quot;</td>
<td>2/3&quot;</td>
<td>1.1&quot;</td>
<td>1.1&quot;</td>
<td>APS-C</td>
<td></td>
</tr>
<tr>
<td>Pixel clock [MHz]</td>
<td>30</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max. FPS (in full frame mode)</td>
<td>270 [burst mode (p. 270 required)] / 106.5 [streaming]</td>
<td>270 [burst mode (p. 270 required)] / 106.5 [streaming]</td>
<td>164 [burst mode (p. 270 required)] / 74.5 [streaming]</td>
<td>123 [burst mode (p. 270 required)] / 46.8 [streaming]</td>
<td>80 [burst mode (p. 270 required)] / 33.4 [streaming]</td>
<td>60.6 [burst mode (p. 270 required)] / 26.5 [streaming]</td>
<td>32 [burst mode (p. 270 required)] / 26.5 [streaming]</td>
<td>23 [burst mode (p. 270 required)] / 26.5 [streaming]</td>
<td>14.4 [burst mode (p. 270 required)] / 7.5 [streaming]</td>
</tr>
<tr>
<td>Frame rate exactness (p. 170)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ADC resolution / Out [bit]</td>
<td>12 / 16, 12, 8</td>
<td>12 / 16, 12, 8</td>
<td>12 / 16, 12, 8</td>
<td>12 / 16, 12, 8</td>
<td>12 / 16, 12, 8</td>
<td>12 / 16, 12, 8</td>
<td>12 / 16, 12, 8</td>
<td>12 / 16, 12, 8</td>
<td>12 / 16, 12, 8</td>
</tr>
<tr>
<td>SNR$_{\text{max}}$ [dB]$^1$</td>
<td>39.2</td>
<td>38.4</td>
<td>45.1</td>
<td>40.3</td>
<td>40.3</td>
<td>43.9 [Dual ADC]</td>
<td>40.2</td>
<td>40.2</td>
<td>tbd</td>
</tr>
<tr>
<td>DR (normal / HDR (p. 363) [dB]$^1$</td>
<td>56.1 / 54.8 / 54.6 /</td>
<td>55.8 / 54.8 / 54.6 /</td>
<td>73.7 / 71.3 / 71.4 / 74.1 / 70.7 /</td>
<td>71.3 / 71.4 / 74.1 / 70.7 / 70.5 /</td>
<td>70.7 / 70.5 / tbd</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Progressive scan sensor (no interlaced problems!)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Rolling shutter</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Global shutter</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Trigger (HW / SW)</td>
<td>X / X</td>
<td>X / X</td>
<td>X / X</td>
<td>X / X</td>
<td>X / X</td>
<td>X / X</td>
<td>X / X</td>
<td>X / X</td>
<td>X / X</td>
</tr>
<tr>
<td>Pipelined global shutter in trigger mode (p. 241)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Linescan mode</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>High color reproducitivity (for color version)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Programmable readout timing with free capture windows and partial scan</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Many trigger modes (free-running, software-triggered, hardware-triggered)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Flash control output, synchronous to integration period</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Power consumption @ 24V</td>
<td>approx. 5.3 W</td>
<td>approx. 5.3 W</td>
<td>approx. 5.0 W</td>
<td>approx. 5.5 W</td>
<td>approx. 5.5 W</td>
<td>approx. 6.3 W</td>
<td>approx. 5.8 W</td>
<td>approx. 5.8 W</td>
<td>approx. 5.8 W</td>
</tr>
</tbody>
</table>
More specific data

<table>
<thead>
<tr>
<th>Model</th>
<th>Mpix</th>
<th>Resolution</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>BlueCOUGAR-X[D]x04</td>
<td>2.2 Mpix</td>
<td>[2048 x 1088]</td>
<td>661</td>
</tr>
<tr>
<td>BlueCOUGAR-X[D]x04b</td>
<td>4.2 Mpix</td>
<td>[2048 x 2048]</td>
<td>665</td>
</tr>
<tr>
<td>BlueCOUGAR-XD104d</td>
<td>2.4 Mpix</td>
<td>[1936 x 1216]</td>
<td>580</td>
</tr>
<tr>
<td>BlueCOUGAR-XD104h</td>
<td>3.2 Mpix</td>
<td>[2064 x 1544]</td>
<td>587</td>
</tr>
<tr>
<td>BlueCOUGAR-XD105a</td>
<td>5.1 Mpix</td>
<td>[2464 x 1894]</td>
<td>594</td>
</tr>
<tr>
<td>BlueCOUGAR-XD107</td>
<td>7.1 Mpix</td>
<td>[3216 x 2480]</td>
<td>602</td>
</tr>
<tr>
<td>BlueCOUGAR-XD109b</td>
<td>8.9 Mpix</td>
<td>[4112 x 3008]</td>
<td>609</td>
</tr>
<tr>
<td>BlueCOUGAR-XD1031</td>
<td>12.4 Mpix</td>
<td>[4112 x 4856]</td>
<td>613</td>
</tr>
<tr>
<td>BlueCOUGAR-XD1031</td>
<td>31.5 Mpix</td>
<td>[6480 x 4856]</td>
<td>617</td>
</tr>
</tbody>
</table>

1 Measured accord. to EMVA1288 with gray scale version of the camera

Note

For further information about rolling shutter, please have a look at the practical report about rolling shutter on our website: [https://www.matrix-vision.com/tl_files/mv11/Glossary/art_rolling_shutter_en.pdf](https://www.matrix-vision.com/tl_files/mv11/Glossary/art_rolling_shutter_en.pdf)

For further information about image errors of image sensors, please have a look at the practical report on our website: [https://www.matrix-vision.com/tl_files/mv11/Glossary/art_image_errors_sensors_en.pdf](https://www.matrix-vision.com/tl_files/mv11/Glossary/art_image_errors_sensors_en.pdf)

11.4 Output sequence of color sensors (RGB Bayer)

![Output sequence of RAW data](https://www.matrix-vision.com/tl_files/mv11/Glossary/art.png)

Figure 4: Output sequence of RAW data

11.5 Bilinear interpolation of color sensors (RGB Bayer)

For Bayer demosaicing in the camera, we use bilinear interpolation:

![Bilinear interpolation](https://www.matrix-vision.com/tl_files/mv11/Glossary/art.png)

Figure 5: Bilinear interpolation
1. Interpolation of green pixels: the average of the upper, lower, left and right pixel values is assigned as the G value of the interpolated pixel.

For example:

\[
G_8 = \frac{(G_3+G_7+G_9+G_{13})}{4}
\]

For \(G_7\):

\[
G_{7\_new} = 0.5 \times G_7 + 0.5 \times \frac{(G_1+G_3+G_{11}+G_{13})}{4}
\]

2. Interpolation of red/blue pixels:

   Interpolation of a red/blue pixel at a green position: the average of two adjacent pixel values in corresponding color is assigned to the interpolated pixel.

   For example:

   \[
   B_7 = \frac{(B_6+B_8)}{2}; \quad R_7 = \frac{(R_2+R_{12})}{2}
   \]

   Interpolation of a red/blue pixel at a blue/red position: the average of four adjacent diagonal pixel values is assigned to the interpolated pixel.

   For example:

   \[
   R_8 = \frac{(R_2+R_4+R_{12}+R_{14})}{4}; \quad B_{12} = \frac{(B_6+B_8+B_{16}+B_{18})}{4}
   \]

Any colored edge which might appear is due to Bayer false color artifacts.

**Note**

There are more advanced and adaptive methods (like edge sensitive ones) available if the host is doing this debayering.

11.6 Supported image formats

It depends on the sensor model which image formats are actually supported.

The parameter **PixelFormat** in the **ImageFormatControl** (p. 169) lists all the supported image formats of the specific camera model. In **wxPropView** (p. 115) you can find the **PixelFormat** parameter in "Setting -> Base -> Camera -> GenICam -> Image Format Control -> Pixel Format".

<table>
<thead>
<tr>
<th>Gray scale version</th>
<th>Color version</th>
<th>Color version (3 head model (p. 59))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mono8</td>
<td></td>
<td>Each color channel represents one sensor</td>
</tr>
<tr>
<td>Mono10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mono12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mono14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mono16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mono12Packed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BayerGR8</td>
<td>(GR, RG, GB, BG depends on camera type)</td>
<td></td>
</tr>
<tr>
<td>BayerGR10</td>
<td>(GR, RG, GB, BG depends on camera type)</td>
<td></td>
</tr>
<tr>
<td>Image Format</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BayerGR12 (GR, RG, GB, BG depends on camera type)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BayerGR16 (GR, RG, GB, BG depends on camera type)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BayerGR12Packed (GR, RG, GB, BG depends on camera type)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BayerGR12Packed (GR, RG, GB, BG depends on camera type)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RGB8Packed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BGR8Packed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BGRA8Packed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BGR10V2Packed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RGB8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BGR8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RGB10p32 (10 bit packed in 32 bit pixel)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>YUV422Packed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>YUV422_YUYVPacked</td>
<td></td>
<td></td>
</tr>
<tr>
<td>YUV444Packed</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

See also

For more details about the image formats, please have a look at the enums `TImageDestinationPixelFormat (C++)` and `TImageBufferPixelFormat (C++)` in the `mvIMPACT Acquire` API manuals.

An example application about the pixel formats called `ContinuousAllFormats` is also available.
12 Filters

MATRIX VISION offers several filters for the mvBlueCOUGAR camera. The hot mirror filter (p. 113) is part of the standard delivery condition.

12.1 Hot mirror filter

The hot mirror filter has great transmission in the visible spectrum and blocks out a significant portion of the IR energy.

<table>
<thead>
<tr>
<th>Technical data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diameter</td>
</tr>
<tr>
<td>Thickness</td>
</tr>
<tr>
<td>Material</td>
</tr>
<tr>
<td>Characteristics</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>AOI</td>
</tr>
<tr>
<td>Surface quality</td>
</tr>
<tr>
<td>Surface irregularity</td>
</tr>
</tbody>
</table>

![Figure 1: IR-CUT wavelengths and transmission diagram](image)

12.2 Cold mirror filter

The high-quality daylight cut filter has optically polished surfaces. The polished surface allows the use of the filter directly in the path of rays in image processing applications. The filter is protected against scratches during the transport by a protection film that has to be removed before installing the filter.

<table>
<thead>
<tr>
<th>Technical data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diameter</td>
</tr>
<tr>
<td>Thickness</td>
</tr>
<tr>
<td>Material</td>
</tr>
<tr>
<td>Characteristics</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
12.3 Glass filter

It is also possible to choose a glass filter with following characteristics:

<table>
<thead>
<tr>
<th>Technical data</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Glass thickness</td>
<td>1.0 mm</td>
</tr>
<tr>
<td>Material</td>
<td>Borofloat without coating ground with protection chamfer</td>
</tr>
<tr>
<td>Surface quality</td>
<td>polished on both sides P4</td>
</tr>
<tr>
<td>Surface irregularity</td>
<td>5/3x0.06 on both sides</td>
</tr>
</tbody>
</table>
13 Application Usage

13.1 wxPropView

wxPropView (p. 115) is an interactive GUI tool to acquire images and to configure the device and to display and modify the device properties of MATRIX VISION GmbH hardware. After the installation you can find wxPropView (p. 115)

- as an icon with the name "wxPropView" on the desktop (Windows) or
- in "~/mvimpact-acquire/apps/mvPropView/x86" (Linux).

**Note**

"With Linux": If you want to run the GUI tools as root for achieving a better performance, you have to export following paths (this is done automatically for simple user):

```bash
export GENICAM_ROOT=/home/x/tmp
export GENICAM_ROOT_V3_0=/home/x/tmp
export GENICAM_LOG_CONFIG=/home/x/tmp/share/genicam/log/config/DefaultLogging.properties
```

or easier run shell script "genicam.sh" before:

```bash
. /etc/profile.d/genicam.sh
```

Given that during the start wxPropView (p. 115) searches the network for connected mvBlueCOUGAR cameras, it could be possible that - depending on the Windows® firewall settings - a Windows® security alert appears. Please click on "Unblock" so that the program works properly.

![Windows Security Alert](image)

Figure 1: Windows security alert
Depending on the driver version, wxPropView starts with the Quick Setup Wizard (p.116) (as soon as a camera with the right firmware version was selected used or a single camera with the right firmware was found) or without it (p.119).

### Quick Setup Wizard

Since mvIMPACT Acquire 2.11.3

The Quick Setup Wizard is a tiny and powerful single window configuration tool to optimize the image quality automatically and to set the most important parameters, which affect the image quality, in an easy way manually and to get a preview of this changes. Settings will be accepted by clicking ok, otherwise the changes are cancelled.

![Quick Setup Wizard started](image_url)

**Figure 2: Quick Setup Wizard started**
Depending on the camera spectrum (gray or color sensor), it will automatically pre-set the camera so that image quality is usually as best as possible.

"For all cameras:"
Image format is chosen as 10 bit (if possible) as a good compromise on image quality and speed.
It will further set:

- "Exposure" to **Auto**,
- "Gain" to **Auto**,
- "Frame rate" to **Auto** based on current settings of the camera, and
- switches camera into continuous mode

"In case of gray:"
The above settings will be also applied whenever the "Gray Preset" button is pressed. For gray cameras it is herewith assumed that image processing prefers a linear camera response.

"In case of color:"
It will additionally set:

- "White balance" in the camera to **Auto**, and will apply
- a host based moderate "Gamma correction" (1.8), and lastly it will apply
- a host (PC) based sensor specific "Color Correction Matrix" and use the respective "sRGB display matrix".

These settings will also be applied whenever the "Color Preset" button is pressed. It is herewith assumed that color camera image is optimized for best human visual feedback.

13.1.1.1 Changing the Presets

There are 3 presets:

- Gray
- Color
- Factory

Factory can be used as a fall back to quickly skip or remove all presets and load the factory default settings.
13.1.1.2 Modifying Settings

All auto modes can be switched off and all settings, such as Gain, Exposure etc. can be subsequently modified by using:

- the sliders,
- the arrow keys, or
- entering real values with your keyboard.

Toggling **Gamma** button loads or unloads a host based 10 bit Gamma correction with a moderate value of 1.8 into the signal processing path. Switch Gamma on if you require a gray level camera image to appear natural for the human eye.

Toggling **Color+** button switches both CCM and sRGB display matrix on and off. This optimizes the sensor color response for the human eye and goes in conjunction with a display color response. Because sRGB displays are mostly used and this is the default color space in Windows OS, these are preselected. If you require other display matrices (e.g. Adobe or WideGamut) feel free to use the tree mode of wxPropView and select **Color Twist Output Correction** accordingly.

**Setting Black Level**
Black level can be used if you require dark portions in the image to appear even darker or brighter. Please note that this slider combines analog and digital settings meaningfully.

**Setting Gain**
Gain settings also combine analog and digital registers into one slider setting.

**Setting Saturation**
Saturation setting increases the color saturation to make the image appear more colored. It does not change uncolored parts in the image nor changes the color tone or hue.

13.1.1.3 How to disable Quick Setup Wizard

Uncheck the checkbox "Show This Display When A Device Is Opened" to disable the Quick Setup Wizard to be called automatically. Use the "Wizards" menu and select "Quick Setup" to open the Quick Setup Wizard once again.

13.1.1.4 How to Return to the Tree Mode

Use **OK** to use the values and settings of the Quick Setup Wizard and go back to the tree mode of wxPropView.

Use **Cancel** to discard the Quick Setup Wizard values and settings and go back to wxPropView and use the former (or default) settings.

13.1.1.5 Image Display Functions

Quick Setup Wizard allows zooming into the image by right clicking in the image area and unchecking "**Fit To Screen**" mode. Use the mouse wheel to zoom in or out. Check "**Fit To Screen**" mode, if you want the complete camera image to be sized in the window screen size.
13.1.1.6 Known Restrictions

In cases of Tungsten (artificial) light, camera brightness may tend to oscillations if Auto functions are used. This can be minimized or avoided by setting the frame frequency to an integer divisor of the mains frequency.

- Example:
  - Europe: 50 Hz; Set frame rate to 100, 50, 25 12.5 fps or appropriate.
  - In countries with 60 Hz use 120, 60, 30 or 15... accordingly.

13.1.1.2 First View of wxPropView

wxPropView (p. 115) consists of several areas:

- **"Menu Bar"**
  (to work with wxPropView (p. 115) using the menu)

- **"Upper Tool Bar"**
  (to select and initialize a device, acquire images, play a recorder sequence)

- **"Left Tool Bar"**
  (to hide and show parts of the GUI)
• "Status Tool Bar"

• "Main Window" with
  – "Grid"
    (tree control with the device settings accessible by the user)
  – "Display"
    (for the acquired images)

• "Analysis"
  (information about whole images or an AOI)

By clicking on F1 you will get the HELP dialog.

Now, you can initialize a device by

• selecting it in the drop down list in the "Upper Tool Bar" and
• clicking on "Use".

After having successfully initialized a device the tree control in the lower left part of the "Main Window" will display the properties (settings or parameters) (according to the "interface layout") accessible by the user.

You've also got the possibility to set your "User Experience". According to the chosen experience, the level of visibility is different:

• Beginner (basic camera settings/properties are visible)
• Expert (e.g. all advanced image processing are visible)
• Guru (all settings/properties are visible)

Properties displayed in light grey cannot be modified by the user. Only the properties, which actually have an impact on the resulting image, will be visible. Therefore, certain properties might appear or disappear when modifying another properties.

To permanently commit a modification made with the keyboard the ENTER must be pressed. If leaving the editor before pressing ENTER will restore the old value.

13.1.1.3 How to see the first image

As described earlier, for each recognized device in the system the devices serial number will appear in the drop down menu in the upper left corner of the "Upper Tool Bar". When this is the first time you start the application after the system has been booted this might take some seconds when working with devices that are not connected to the host system via PCI or PCIe.

Once you have selected the device of your choice from the drop down menu click on the "Use" button to open it.

When the device has been opened successfully, the remaining buttons of the dialog will be enabled:
Note

Following screenshots are representative and were made using a mvBlueCOUGAR-X camera as the capturing device.

For color sensors, it is recommended to perform a **white balance** (p. [144](#)) calibration before acquiring images. This will improve the quality of the resulting images significantly.

---

**Figure 4:** wxPropView - First start

Now, you can capture an image (“Acquisition Mode”: “SingleFrame”) or display live images (“Continuous”). Just

- select an “Acquisition Mode” e.g. “SingleFrame” and
- click the “Acquire” button.
The techniques behind the image acquisition can be found in the developers sections.

The frame rate depends on

- the camera,
- the pixel clock of the sensor and
- the "Acquisition Frame Rate".

If you want to have a fixed frame rate using the "Continuous" mode, GenICam (p. 234) offers the property "Setting -> Base -> Camera -> GenICam -> Acquisition Control -> Acquisition Frame Rate" (from 5 fps to maximum of the camera in 0.1 increments). Just adapt this property to your needs.

Alternatively, if you need frame rates below 5 fps, you can use Timers. In the use case Creating synchronized acquisitions using timers (p. 414), for example, a frame rate of 1 fps is generated.

13.1.1.3.1 Record Mode

It is also possible to record image sequences using wxPropView.

1. For this, you have to set the size of the recorder in "System Settings -> RequestCount" e.g. to 100. This will save the last 100 requests in the request queue of the driver, i.e. the image data including the request info like frame number, time stamp, etc.

2. Afterwards you can start the recording by clicking the Rec. button.

3. With the Next and Prev. buttons you can display the single images.

If you switched on the request info overlay (right-click on the display area and select the entry to activate this feature), these information will be displayed on the image, too. With the timestamp you can see the interval of the single frames in microseconds.

Figure 5: wxPropView - Using the record mode.
13.1.1.3.2 Hard Disk Recording

You can save acquired images to the hard disk the following way:

1. In the "Menu Bar" click on "Capture -> Recording -> Setup Hard Disk Recording".
2. Confirm with "Yes".
3. Afterwards select the target folder for the images.
4. Finally, choose the file format of the acquired images.

Figure 6: wxPropView - Hard Disk Recording.

13.1.1.4 Using the analysis plots

With the analysis plots you have the possibility to get image details and to export them (p.131).
13.1.1.4.1 Spatial noise histogram

The spatial noise histogram calculates and evaluates statistically the difference between two neighbouring pixels in vertical and horizontal direction. I.e. it shows the sensor’s spatial background pattern like the sensitivity shifts of each pixel. An ideal sensor or camera has a spatial noise of zero. However, you have to keep in mind the temporal noise as well.

![Spatial noise histogram](image)

Figure 7: wxPropView - Spatial noise histogram

Read: Channel::Direction (Mean difference, most frequent value count/value, Standard deviation)

**Example:** For a single channel (Mono) image the output of ‘C0Hor(3.43, 5086/0, 9.25), C0Ver(3.26, 4840/0, 7.30) will indicate that the mean difference between pixels in horizontal direction is 3.43, the most frequent difference is 0 and this difference is present 5086 times in the current AOI. The standard deviation in horizontal direction is 9.25. The C0Ver value list contains the same data but in vertical direction.

13.1.1.4.2 Temporal noise histogram

The temporal noise histogram shows the changes of a pixel from image to image. This method is more stable because it is relatively independent from the image content. By subtracting two images, the actual structure is eliminated, leaving the change of a pixel from image to image, that is, the noise. When capturing images, all parameters must be frozen, all automatic mechanisms have to be turned off and the image may not have underexposed or saturated areas. However, there are no picture signals without temporal noise. Light is a natural signal and the noise always increases with the signal strength. If the noise only follows the natural limits, then the camera is good. Only if additional noise is added the camera or the sensor has errors.
Read: Channel# (Mean difference, most frequent value count/ value, Standard deviation)

**Example:** For a single channel (Mono) image the output of ‘C0(3.43, 5086/ 0, 9.25) will indicate that the mean difference between pixels in 2 consecutive images is 3.43, the most frequent difference is 0 and this difference is present 5086 times in the current AOI. The standard deviation between pixels in these 2 images is 9.25. Please note the impact of the 'Update Interval' in this plot: It can be used to define a gap between 2 images to compare. E.g. if the update interval is set to 2, the differences between image 1 and 3, 3 and 5, 5 and 7 etc. will be calculated. In order to get the difference between 2 consecutive images the update interval must be set to 1!

13.1.1.5 Storing and restoring settings

When wxPropView (p.115) is started for the first time, the values of properties set to their default values will be displayed in green to indicate that these values have not been modified by the user so far. Modified properties (even if the value is the same as the default) will be displayed in black.
Settings can be stored in several ways (via the "Menu Bar": "Action -> Capture Settings -> Save Active Device Settings"): 

- "As Default Settings For All Devices Belonging To The Same Family (Per User Only)". As the start-up parameters for every device belonging to the same family, e.g. for mvBlueCOUGAR-X, mvBlueCOUGAR-XD.
- "As Default Settings For All Devices Belonging To The Same Family And Product Type". As the start-up parameters for every device belonging to the same product, e.g. for any mvBlueCOUGAR-X but not for mvBlueCOUGAR-XD.
- "As Default Settings For This Device(Serial Number)". As the start-up parameters for the currently selected device.
- "To A File": As an XML file that can be used e.g. to transport a setting from one machine to another or even to use the settings configured for one platform on another (Windows <-> Linux).

During the startup of a device, all these setting possibilities show different behaviors. The differences are described in chapter Settings behaviour during startup (p. 62).

Restoring of settings previously stored works in a similar way. After a device has been opened the settings will be loaded automatically as described in Settings behaviour during startup (p. 62).

However, at runtime the user has different load settings possibilities (via the "Menu Bar": "Action -> Capture Settings -> Load Active Device Settings")

- explicitly load the device family specific settings stored on this machine (from "The Default Settings Location For This Devices Family (Per User Only)")
- explicitly load the product specific settings stored on this machine (from "The Default Settings Location For This Devices Family And Product Type")
- explicitly load the device specific settings stored on this machine (from "The Default Settings Location For This Device(Serial Number)")
- explicitly load device family specific settings from a XML file previously created ("From A File")

**Warning**

Since mvIMPACT Acquire 2.9.0 GenICam devices will be able to save their properties in a XML File, only if the properties have the streamable attribute set (for more information refer to the GenICam standard specification). Properties with no streamable attribute set, will be silently ignored when saving, which means they will not be saved in the XML file. For MATRIX VISION GenICam cameras, starting with firmware version 1.6.414 the streamable attribute is set for all the necessary properties.

**Warning**

Since mvIMPACT Acquire 2.9.0 and again in version 2.11.0 storing and loading of camera settings in a XML file for the GenICam (p. 234) interface layout has been updated. As a result XML files created with newer versions of mvIMPACT Acquire might not be readable on systems with older version of mvIMPACT Acquire installed. XML files created on systems with earlier versions of mvIMPACT Acquire will always be readable this or newer versions. See the following table for details.
Attention

Since mvIMPACT Acquire 2.28.0 it is possible for devices operated in the GenICam interface layout to store settings including including sequencer set and user set (see SFNC for details) data by specifying appropriate flags during the storage operation. Settings stored like this cannot be loaded by previous mvIMPACT Acquire versions.

Note

For devices operated in the GenICam interface layout further restriction apply: Settings created with a certain product type can only be used with other devices belonging to the exact same type as defined by the property Product inside the device list (the one device specific property list that is accessible without initializing the device before). Even if a setting can be used with various firmware versions it is recommended to use one setting for multiple devices all updated to the very same firmware version to avoid compatibility problems.

With “Action -> Capture Settings -> Manage...” you can delete the settings which were saved on the system.

13.1.1.6 Properties

All properties and functions can be displayed in the list control on the lower left side of the dialog. To modify the value of a property select the edit control right of the properties name. Property values, which refer to the default value of the device, are displayed in green. A property value once modified by the user will be displayed in black (even if the value itself has not changed). To restore its default value of a single property

- right click on the name of the property and
• select "Restore Default".

To restore the default value for a complete list (which might include sub-lists)

• right click on the name of a list and
• select "Restore Default".

In this case a popup window will be opened and you have to confirm again.

![Figure 11:wxPropView - Restore the default value of a property](image)

Most properties store one value only, thus they will appear as a single entry in the property grid. However, properties are capable of storing more than one value, if this is desired. A property storing more than one value will appear as a parent list item with a WHITE background color (lists will be displayed with a grey background) and as many child elements as values stored by the property. The PARENT grid control will display the number of values stored by the property, every child element will display its corresponding value index.

If supported by the property, the user might increase or decrease the number of values stored by right clicking on the PARENT grid element. If the property allows the modification the pop up menu will contain additional entries now:

![Figure 12:wxPropView - A resizable property](image)
When a new value has been created it will be displayed as a new child item of the parent grid item:

<table>
<thead>
<tr>
<th>Camera Descriptions</th>
<th>Standard_Generic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>NonStandard</td>
</tr>
<tr>
<td>VideoOutput[0]</td>
<td>Composite</td>
</tr>
<tr>
<td>VideoOutput[1]</td>
<td>SVide</td>
</tr>
<tr>
<td>VideoOutput[2]</td>
<td>Undefined</td>
</tr>
<tr>
<td>ScanRate_KHz</td>
<td>14750</td>
</tr>
</tbody>
</table>

Currently, only the last value can be removed via the GUI and a value can’t be removed, when a property stores one value only.

Also the user might want to set all (or a certain range of) values for properties that store multiple values with a single operation. If supported by the property, this can also be achieved by right clicking on the PARENT grid element. If the property allows this modification the pop up menu will again contain additional entries:
It’s possible to either set all (or a range of) elements of the property to a certain value OR to define a value range, that then will be applied to the range of property elements selected by the user. The following example will explain how this works:

![Figure 15:wxPropView - Setting multiple property values within a certain value range](image1.png)

In this sample the entries 0 to 255 of the property will be assigned the value range of 0 to 255. This will result in the following values AFTER applying the values:

![Figure 16:wxPropView - After applying the value range to a property](image2.png)
13.1.1.7 Methods

Method appears as entries in the tree control as well. However, their name and behavior differs significantly from the behavior of properties. The names of method objects will appear in 'C' syntax like e.g. "int function( char, int )". This will specify a function returning an integer value and expecting a string and an integer as input parameters. To execute a method object:

- right click on the name of a method and
- select "Execute" from the popup menu:

![Figure 17:wxPropView - Calling a method object](image)

Parameters can be passed to methods by selecting the edit control left of a method object. Separate the parameters by blanks. So to call a function expecting a string and an integer value you e.g. might enter "testString 0" into the edit control left of the method.

The return value (in almost every case an error code as an integer) will be displayed in the lower right corner of the tree control. The values displayed here directly correspond the error codes defined in the interface reference and therefore will be of type TDMR_ERROR or TPROPHANDLING_ERROR.

13.1.1.8 Copy grid data to the clipboard

Since wxPropView (p. 115) version 1.11.0 it is possible to copy analysis data to the clipboard. The data will be copied in CSV style thus can be pasted directly into tools like Open Office™ or Microsoft® Office™.

Just

- right-click on the specific analysis grid when in numerical display mode and
- select "Copy grid to clipboard" from the pop up menu.
13.1.1.9 Import and Export images

**wxPropView** (p. 115) offers a wide range of image formats that can be used for exporting captured image to a file. Some formats e.g. like packed YUV 4:2:2 with 10 bit per component are rather special thus they can't be stored into a file like e.g. offered by the BMP file header. When a file is stored in a format, that does not support this data type **wxPropView** (p. 115) will convert this image into something that matches the original image format as close as possible. This, however, can result in the loss of data. In order to allow the storage of the complete information contained in a captured image **wxPropView** (p. 115) allows to store the data in a raw format as well. This file format will just contain a binary dump of the image with no leader or header information. However, the file name will automatically be extended by information about the image to allow the restoring of the data at a later time.

All image formats, that can be exported can also be imported again. Importing a file can be done in 3 different ways:

- via the menu (via the "Menu Bar": "Action -> Load image...")
- by dragging an image file into an image display within **wxPropView** (p. 115)
- by starting **wxPropView** (p. 115) from the command line passing the file to open as a command line parameter (p. 148) (under Windows® e.g. "wxPropView.exe MyImage.png" followed by [ENTER])

When importing a "*.raw" image file a small dialog will pop up allowing the user to define the dimensions and the pixel format of the image. When the file name has been generated using the image storage function offered by **wxPropView** (p. 115), the file name will be passed and the extracted information will automatically be set in the dialog thus the user simply needs to confirm this information is correct.
13.1.1.10 Setting up multiple display support and/or work with several capture settings in parallel

wxPropView (p. 115) is capable of

- dealing with multiple capture settings or acquisition sequences for a single device and in addition to that
- it can be configured to deal with multiple image displays.

The amount of parallel image displays can be configured via the command line parameter "dcx" and "dcy". In this step by step setup wxPropView (p. 115) has been started like this from the command line:

wxPropView dcx=1 dcy=2

This will result in 1 display in horizontal direction and 2 in vertical direction.

Since

mvIMPACT Acquire 2.18.1

Is is also possible to change the amount of display at runtime via "Settings -> Image Displays -> Configure Image Display Count".
Additional capture settings can be created via "Menu Bar": "Capture -> Capture Settings -> Create Capture Settings". The property grid will display these capture settings either in "Developers" or in "Multiple Settings View".

**Note**

In GenICam interface (p. 143) layout multiple capture settings are NOT supported. However, you can define different acquisition sets using the Sequencer Control (p. 190).

Now, in order to set up wxPropView (p. 115) to work with 2 instead of one capture setting,

1. Various additional capture setting can be created. In order to understand what a capture setting actually is please refer to
   - "Working with settings" chapter of the "mvIMPACT Acquire API" manuals.
   
   Creating a capture setting is done via "Capture -> Capture Settings -> Create Capture Setting".

2. Then, the user is asked for the name of the new setting.
3. And finally for the base this new setting shall be derived from.

Afterwards, in this example we end up having 2 capture settings:

- a "Base" setting, which is always available
- a "NewSetting1", which has been derived from "Base".
As "NewSetting1" has been derived from "Base" changing a property in "Base" will automatically change this property in "NewSetting1" if this property has not already been modified in "NewSetting1". Again to get an understanding for this behaviour please refer to

- "Working with settings" chapter of the "mvIMPACT Acquire API" manuals.

Now, to set up wxPropView (p.115) to display all images taken using capture setting "Base" in one display and all image taken using capture setting "NewSetting1" in another display the capture settings need to be assigned to image displays via "Capture -> Capture Settings -> Assign To Display(s)".
By default a new setting when created will be assigned to one of the available displays in a round-robin scheme, thus when there are 3 displays, the first (Base) setting will be assigned to "Display 0", the next to "Display 1", the next to "Display 2" and a fourth setting will be assigned to "Display 0" again. The setting to display relationships can be customized via "Capture -> Capture Settings -> Assign to Display(s)".

As each image display keeps a reference to the request, this image belongs to the driver can’t re-use the request buffer until a new request is blitted into this display. Thus, it might be necessary to increase the number of request objects the driver is working with if a larger number of displays are involved. The minimum number of requests needed is 2 times the amount of images displays. The number of requests used by the driver can be set up in the drivers property tree:
Finally, `wxPropView` (p. 115) must be configured in order to use all available capture settings in a round-robin scheme. This can be done by setting the capture setting usage mode to "Automatic" via "Capture -> Capture Settings -> Usage Mode".

That's it. Now, starting a live acquisition will display live images in both displays and each display is using a different set of capture parameters. If a device supports parallel acquisition from multiple input channels, this will increase

- the used bandwidth and also
- the CPU load
as *wxPropView* (p. 115) now needs to display more images per second. Each display can be configured independently thus e.g. one display can be used scaled while the other displays 1:1 data. The analysis plots can be assigned to a specific display by left-clicking on the corresponding image display, the info plot will plot a graph for each capture setting in parallel.

![Image of wxPropView interface](image)

**Figure 29:wxPropView - Running example**

When only one setting shall be used at a given time, this can be achieved by setting the capture setting usage mode back to "Manual" via "Capture -> Capture Settings -> Usage Mode". Then the setting that shall be used can be manually selected in the request control list:
This can even be changed during a running acquisition.

13.1.1.11 Bit-shifting an image

`wxPropView` (p. 115) shows snapped or live images in the display area of the GUI. The area, however, shows the most significant bits (msb) of the image in the 8 bit display.

The following image shows how a mid-grey 12 bit pixel of an image is displayed with 8 bit. Additionally, two shifts are shown.

```
Mono12 (12 bit):

0 1 1 1 0 0 1 1 1 1 0 0

Display (8 bit):

0 1 1 1 1 0 0 1 1 1 1 0 0

Display (8 bit); Shift: 1:

0 1 1 1 0 0 1 1 1 1 0 0

Display (8 bit); Shift: 2:

0 1 1 1 0 0 1 1 1 1 0 0
```

Figure 31: Mid-grey 12 bit pixel image and 8 bit display with 2 example shifts
13.1 \textit{wxPropView}

In this particular case, the pixel will be brighter (as the most significant bits are 1's). Perhaps you already recognized it. Each shift means that each pixel value is multiplied or divided by 2 according to the direction.

Anyway, there is one restriction in the 8 bit display:

If the pixel value is greater than 255, the pixel value will be clipped to 255. To describe this from a programmer's view; \( a \) represents the pixel value:

\[
a = (a > 255) ? 255 : a
\]

With \textit{wxPropView} \((p.115)\), you can shift the bits in the display using the left and right arrow keys. Furthermore, you can turn on the monitor display to compare the images synchronously.

13.1.1.12 Changing the view of the property grid to assist writing code that shall locate driver features

With \textit{wxPropView} \((p.115)\), it is possible to switch the views between "Standard View" (user-friendly) and "Developers View". While the first (default) view will display the device drivers feature tree in a way that might be suitable for most users of a GUI application it might present the features in a slightly different order as they actually are implemented in the device driver. The developers view switches the tree layout of the application to reflect the feature tree exactly like it is implemented and presented by the SDK. It can be helpful when writing code that shall locate a certain property in the feature tree of the driver using the C, C++, or .NET interface. The feature hierarchy displayed here can directly be used for searching for the features using the "ComponentLocator (C++/.NET)" objects or "DMR_FindList (C)" and "OBJ_GetHandleEx (C)" functions.
13.1.1.13 Accessing log files

Since mvIMPACT Acquire 2.11.9

Using Windows, it is possible to access the log files generated by MATRIX VISION via the Help menu. Sending us the log files will speed up support cases.

The options are to

- directly open the logs folder, to
- create a zip file with all the logs, and to
- open the systems default email client to send an email to support@matrix-vision.com.

See also

Accessing log files using Linux (p. 217)

13.1.2 How to configure a device

As described above, after the device has been initialized successfully in the “Grid” area of the GUI the available properties according to the chosen “interface layout” (e.g. GenICam) are displayed in a hierarchy tree.
The next chapter will show how to set the interface layout and which interface you should use according to your needs.

13.1.2.1 Changing interface to GenICam or device specific

The mvIMPACT Acquire interface internally uses the GenICam runtime libs, so that it can be considered as an user application written with the GenICam interface. This behavior has several advantages:

- The mvBlueCOUGAR can be used like any other mvIMPACT Acquire device.
- The current version of the mvGenTL-Acquire driver is meant to work with every GigE Vision and every USB3 Vision compliant device.
- Developers either can use the generic GenICam properties or the mvIMPACT Acquire properties.

You can change the property interfaceLayout with wxPropView to select the preferred interface.

- When GigE Vision and GenICam compliant devices from several vendors shall be used in the same application it's recommended to use the "GenICam" interface layout only in order to keep the application code simple.
- When several different MATRIX VISION devices (e.g. a frame grabber, a USB camera and a GigE Vision camera) shall be operated by the same application, it's recommended to use the device specific interface for the same reasons.
- When an application shall be able to work with every MATRIX VISION device and every GigE Vision and GenICam compliant device both approaches make sense however a mixture between the 2 worlds can't be avoided.

To specify the InterfaceLayout for all devices globally, you can do this via the "Action -> Default Device Interface Layout" in the "Menu Bar":

![Figure 34: Global selection of the interface layout for all devices](image-url)
If you want to specify the `InterfaceLayout` for the used device, you can do this via "Device Properties" in the section "Device -> InterfaceLayout":

![Image](image_url)

**Figure 35:** Selection of the interface layout for this specific device

See also

- GenICam (p. 234)
- Standard Feature Naming Convention of GenICam properties (p. 242)

13.1.2.2 White balance of a camera device (color version)

Start the `wxPropView` (p. 115) and initialize the device by clicking "Use" and start a "Continuous" acquisition.

![Image](image_url)

**Figure 36:** wxPropView - Starting window
While using a color version of the camera, the PC will calculate a color image from the original gray Bayer mosaic data. For getting correct colors when working with a Bayer mosaic filter you have to calibrate the white balance (this must be performed every time the lighting conditions change).

The "White Balance Control" can be found in "Setting -> Base -> Camera -> GenICam -> Analog Control -> Balance White Auto". Just select "Continuous" and you will get a white balanced image.

![User Experience: Guru](image)

**Figure 37: wxPropView - Selecting WhiteBalance profile**
"Optimizing white balance in the camera"

The gain is increased before the digital white balancing, which uses green as reference. Based on the 14 bit image raw data, the digital white balance adjusts the gain of red and blue. Afterwards, the 8 bit reducing and the transfer takes place.

To optimize the digital white balance within the camera you can select "Red" or "Blue" in "Balance Ratio Selector" (in "Analog Control") to adjust the gain. Afterwards, turn "Balance White Auto" to "Off" and "Once" to execute it.

<table>
<thead>
<tr>
<th>Analog Control</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Gain Selector</td>
<td>AnalogAll</td>
</tr>
<tr>
<td>Black Level Selector</td>
<td>All</td>
</tr>
<tr>
<td>Balance Ratio Selector</td>
<td>Red</td>
</tr>
<tr>
<td>Balance Ratio</td>
<td>1.000</td>
</tr>
<tr>
<td>Balance White Auto</td>
<td>Off</td>
</tr>
</tbody>
</table>

Figure 39: wxPropView - Optimizing white balance

13.1.2.3 Configuring different trigger modes

To configure a device for a triggered acquisition, in wxPropView (p.115) the property "Image Setting -> Camera -> TriggerMode" ("DeviceSpecific interface layout") or "Setting -> Base -> Camera -> GenICam -> Acquisition Control -> Trigger Selector" ("GenICam interface layout") is available.

Note

The supported trigger modes of each sensor are described in the More specific data (p.96) of each sensor.
13.1.2.4 Testing the digital inputs

**Note**

The following description will be significant if you are using the “DeviceSpecific interface layout”. In GenICam layout, the “Digital I/O” section can be found in “Setting -> Base -> Camera -> GenICam -> Digital I/O Control”.

For performance reasons, device drivers will not automatically update their digital input properties if nobody is interested in the current state. Therefore, in order to check the current state of a certain digital input, it is necessary to manually refresh the state of the properties. To do this please right-click on the property you are interested in and select “Force Refresh” from the pop-up menu.

**GenICam interface layout only:**

Some devices might also offer an event notification if a certain digital input changed its state. This event can then be enabled

- via the “EventSelector” in “Setting -> Base -> Camera -> GenICam -> Event Control”.
- Afterwards, a callback can be registered by right-clicking on the property you are interested in again.
- Now, select “Attach Callback” from the pop-up menu and switch to the “Output” tab in the lower right section of wxPropView (Analysis tabs).

Whenever an event is send by the device that updates one of the properties a callback has been attached to, the output window will print a message with some information about the detected change.

![Image](image_url)

**Figure 39: wxPropView - Call refresh**
13.1.2.5 Saving user settings in the non-volatile flash memory

The mvBlueCOUGAR-X camera offers the possibility, to save up to 4 user sets in the camera’s flash memory directly. This means that all camera specific settings you’ve adjusted via wxPropView (p. 115) can be saved in a non-volatile memory.

Example: You have connected a flash via exposure out of the camera and you want to avoid an overload of the flash by maloperation, you can save a suitable shutter time, with which the camera will start.

To save your specific settings, set you properties in the “Setting -> Camera -> GenICam” section of wxPropView (p. 115). Then, select in “User Set Control” your user set with the “User Set Selector”, for example “UserSet1”. Afterwards, save the user set with “int UserSetSave()”. Finally, if you want that the camera starts with a specific user set (after power up), you have to select it with the “User Set Default Selector”.

Figure 40: wxPropView - User set control

Attention

A firmware update will delete all saved register settings!

13.1.3 Command-line options

It is possible to start wxPropView via command line and controlling the starting behavior using parameters. The supported parameter are as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>width or w</td>
<td>Defines the startup width of wxPropView. Example: width=640</td>
</tr>
<tr>
<td>height or h</td>
<td>Defines the startup height of wxPropView. Example: height=460</td>
</tr>
<tr>
<td>xpos or x</td>
<td>Defines the startup x position of wxPropView.</td>
</tr>
<tr>
<td>ypos or y</td>
<td>Defines the startup y position of wxPropView.</td>
</tr>
<tr>
<td>splitterRatio</td>
<td>Defines the startup ratio of the position of the property grids splitter. Values between &gt; 0 and &lt; 1 are valid. Example: splitterRatio=0.5</td>
</tr>
<tr>
<td>propgridwidth or pgw</td>
<td>Defines the startup width of the property grid.</td>
</tr>
<tr>
<td>debuginfo or di</td>
<td>Will display debug information in the property grid.</td>
</tr>
<tr>
<td>dic</td>
<td>Will display invisible (currently shadowed) components in the property grid.</td>
</tr>
<tr>
<td>displayCountX or dcx</td>
<td>Defines the number of images displayed in horizontal direction.</td>
</tr>
<tr>
<td>displayCountY or dcy</td>
<td>Defines the number of images displayed in vertical direction.</td>
</tr>
<tr>
<td>fulltree or ft</td>
<td>Will display the complete property tree (including the data not meant to be accessed by the user) in the property grid. Example (Tree will be shown): fulltree=1</td>
</tr>
<tr>
<td>device or d</td>
<td>Will directly open a device with a particular serial number. * will take the first device. Example: d=GX000735</td>
</tr>
<tr>
<td>qsw</td>
<td>Will forcefully hide or show the Quick Setup Wizard, regardless of the default settings. Example (Quick Setup Wizard will be shown): qsw=1</td>
</tr>
<tr>
<td>live</td>
<td>Will directly start live acquisition from the device opened via device or d directly.</td>
</tr>
</tbody>
</table>
13.1.3.1 Sample (Windows)

wxPropView.exe d=* fulltree=1 qsw=0

This will start the first available device, will hide the Quick Setup Wizard, and will display the complete property tree.

13.2 mvGigEConfigure

mvGigEConfigure (p. 149) is an interactive GUI tool to install the GigE (p. 236) capture filter which adds a service to your Gigabit Ethernet interface.

**Note**

Windows 2000 does not support the "Resend" mechanism of GigE Vision (p. 236).

13.2.1 Install GigE Vision capture filter

To install the GigE (p. 236) capture filter, please follow these instructions:

1. Start the mvGigEConfigure (p. 149) by clicking the program in the start menu.
   The tool will start.

2. Click on the "Install driver" button.
   mvGigEConfigure (p. 149) will install the driver and adds a service to your Gigabit Ethernet interface automatically.

After installation of the capture filter, you will see your network devices and the connected devices:

![Figure 41: mvGigEConfigure displays connected devices](image-url)
When a GigE (p. 236) compliant device will be initialised, wxPropView (p. 115) it will display the used driver technology:

![Figure 42: wxPropView - Device Properties](image)

Once a device has been initialized successfully the property "mvStreamDriverTechnology" will display either "FilterDriver" when the high performance Kernel mode driver is used or "SocketAPI" when the driver is not installed or available. The property is located under "System Settings -> GenTL" in interface layout "DeviceSpecific" or under "Image Settings -> Camera -> GenTL" in interface layout "GenICam".

13.2.2 Remove GigE Vision capture filter

To remove the GigE (p. 236) capture filter, please follow these instructions:

1. Start the mvGigEConfigure (p. 149) by clicking the program in the start menu.
   The tool will start.
2. Click on the "Remove driver" button.
   mvGigEConfigure (p. 149) will remove the driver automatically.

13.2.3 Command-line options

mvGigEConfigure offers a couple of command-line options to allow driver installation without the need of user interaction. The available command-line options are

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>install</td>
<td>Will install the driver if it can be located under</td>
</tr>
<tr>
<td></td>
<td>- &quot;%MVIMPACT_ACQUIRE_DIR%\Kernel\Drivers\mvGigECaptureDriver(x86)(SHA1)&quot; for 32-bit versions of Windows 2000 and Windows XP</td>
</tr>
<tr>
<td></td>
<td>- &quot;%MVIMPACT_ACQUIRE_DIR%\Kernel\Drivers\mvGigECaptureDriver(x86)&quot; for 32-bit versions of Windows Vista, 7, 8 and 8.1</td>
</tr>
<tr>
<td></td>
<td>- &quot;%MVIMPACT_ACQUIRE_DIR%\Kernel\Drivers\mvGigECaptureDriver(x86)(SHA1)(SHA1-A256.EV)&quot; for 32-bit versions of Windows 10</td>
</tr>
<tr>
<td></td>
<td>- &quot;%MVIMPACT_ACQUIRE_DIR%\Kernel\Drivers\mvGigECaptureDriver(x64)(SHA1)&quot; for 64-bit versions of Windows XP</td>
</tr>
<tr>
<td></td>
<td>- &quot;%MVIMPACT_ACQUIRE_DIR%\Kernel\Drivers\mvGigECaptureDriver(x64)(SHA1-A256.EV)&quot; for 64-bit versions of Windows 10</td>
</tr>
<tr>
<td>install_retry_count</td>
<td>Defines the number of retry attempts for the driver installation. This can sometimes be useful when a network component is currently in use/shutting down. 45 seconds will be spent between 2 consecutive attempts. Currently this value will only have an effect, if specified BEFORE the 'install' parameter!</td>
</tr>
<tr>
<td>enable</td>
<td>Will enable the driver for interface index, all interfaces or all interfaces that have at least one GigE Vision device connected to it.</td>
</tr>
<tr>
<td>disable</td>
<td>Will disable the driver for interface index or every interface.</td>
</tr>
<tr>
<td>remove</td>
<td>Will remove the driver.</td>
</tr>
<tr>
<td>welcome</td>
<td>Will display a welcome message to the user explaining what is about to happen.</td>
</tr>
<tr>
<td>postInstallMessage</td>
<td>Will display a message to the user AFTER the filter driver has been installed on the system.</td>
</tr>
<tr>
<td>log_file=file_name or lf=file_name</td>
<td>Will write the content of the output window of the application into a file. Passing 'STDDLOGDIR' as the initial path (e.g. STDDLOGDIR/file_name.log) will write the file into MVIMPACT_ACQUIRE_DATA_DIR/logs thus the file containing the regular mvIMPACT Acquire log-files as well.</td>
</tr>
<tr>
<td>quit</td>
<td>Will automatically terminate the application once all the other command line parameters have been processed.</td>
</tr>
</tbody>
</table>

13.2.3.1 Sample (Windows)

So e.g. to automatically install the filter driver and terminate the application afterwards without any additional messages display to the user the following command line is needed:
mvGigEConfigure.exe install quit

This will install the filter driver, but will disable it on all the adapters and then will terminate automatically.

mvGigEConfigure install disable=all quit

This will enable the filter driver on adapters 1 and 3.

mvGigEConfigure enable=1 enable=3

This will be used during standard installation.

mvGigEConfigure welcome install postinstallmessage

13.3 mvIPConfigure

mvIPConfigure (p. 152) is an interactive GUI tool to configure the network behaviour of GigE (p. 236) devices. With mvIPConfigure (p. 152) it is possible

• to assign a user defined name to a GigE Vision™ compliant device,
• to change its IP address behaviour, and
• to find and fix incorrectly configured (e.g. wrong IP address) GigE Vision™ compliant devices.

The user can select, if the device should

• use a persistent IP address (if supported by the device) or
• use DHCP (p. 234) to obtain an IP address

LLA must always be enabled thus this control will never become modifiable in this GUI application.

Every GigE Vision™ compliant device must use the following IP protocol selection algorithm:

![Diagram of IP protocol selection sequence]

Figure 43: IP protocol selection sequence
13.3 mvIPConfigure

13.3.1 Configure a GigE Vision device

To configure the GigE device, please follow these instructions:

1. Start the mvIPConfigure by clicking the program in the start menu. The tool will start.

2. Select the device you want to configure from the list on left side of the tool.

3. Click on the "Configure" button. Now every feature that is supported by the device and that can be modified will become enabled.

Figure 44: mvIPConfigure - Start window
4. When the device shall use a persistent IP address not only check the check box "Use Persistent IP" but also enter all the required information into the text control in the group box "Persistent IP address".

**Note**

In order that the camera is certainly detected in the network with the new IP address, please switch the camera off and on after you applied the changes in the next step.

5. To write the changes to the device click on "Apply Changes" button. To discard the changes made in the GUI controls either close the application, select another device or click anywhere in the device list.

**Note**

If you use two GigE Vision cameras with two network adapters in one host device, please be sure that you use different subnets.

It is also possible to assign a temporary IP address to a certain device with a known MAC address via the "Action" menu item.

1. Select the "Action" menu item and click on "Manually Assign Temporary IPv4 Address".
A dialog opens, where you can enter the network details.

2. Click on "Execute" button.
   The device will now use this data until disconnected from its power supply or until it is assigned a new IP address.

13.3.2 Recover a incorrectly configured GigE Vision device

Because of the activated setting "Use Advanced Device Discovery", by default, incorrectly configured GigE Vision™ compliant devices will be listed within mvIPConfigure (p.152) even if the IP address does not match the local network.
Just select the device and configure it as described in the chapter above (p. 153).

13.4 mvDeviceConfigure

mvDeviceConfigure (p. 156) is an interactive GUI tool to configure MATRIX VISION devices. It shows all connected devices. With mvDeviceConfigure (p. 156) it is possible e.g.

- to check, if the camera is accessible on host PC,
- to set the device ID,
- to update firmware of the mvBlueCOUGAR or
- to upload a GenICam (p. 234) XML file.

**Note**

Given that during the start mvDeviceConfigure (p. 156) searches the network for connected mvBlueCOUGARs, it could be possible that - depending on the Windows firewall settings - a Windows security alert appears. Please click on "Unblock" so that the program works properly.
Various things can also be done without user interaction (e.g. updating the firmware of a device). To find out how to do this please start `mvDeviceConfigure` and have a look at the available command line options presented in the text window in the lower section (the text control) of the application.

### 13.4.1 How to set the device ID

**Note**

Currently you cannot set the device ID.

The device ID is used to identify the devices with a self defined ID. The default ID on the device's EEPROM is "0". If the user hasn't assigned unique device IDs to his devices, the serial number can be used to selected a certain device instead. However, certain third-party drivers and interface libraries might rely on these IDs to be set up in a certain way and in most of the cases this means, that each device needs to have a unique ID assigned and stored in the devices non-volatile memory. So after installing the device driver and connecting the devices setting up these IDs might be a good idea.

To set the ID please start the `mvDeviceConfigure` tool. You will see the following window:
Whenever there is a device that shares its ID with at least one other device belonging to the same device family, `mvDeviceConfigure` will display a warning and the devices in red color.

13.4.1.1 Step 1: Device selection

Select the device you want to set up. Select the menu item *Action* and click on *Set ID*. 
13.4 mvDeviceConfigure

It is also possible to select the action with a right click on the device.

![Figure 52: mvDeviceConfigure - Select action](image)

13.4.1.2 Step 2: Open dialog to set the ID

- Enter the new ID and click OK.
- Now the overview shows you the list with all devices as well as the new ID.

In case there has been an ID conflict before that has been resolved now mvDeviceConfigure (p. 156) will no longer highlight the conflict now.

13.4.2 How to update the firmware

With the mvDeviceConfigure (p. 156) tool it is also possible to update the firmware. In the device list, new firmware versions, if available, will be marked in blue.

**Note**

Since version 2.29.1 for GenICam compliant devices firmware updates will not be installed by the mvIMPACT Acquire driver installation package any more. Firmware archives can be downloaded from the MATRIX VISION ION website instead. The latest firmware (and previous versions) can always be found in the download area of the corresponding product. Downloaded archives should be copied into the \Firmware\<product name> folder as then mvDeviceConfigure will automatically check if your devices run with the latest firmware or not.

To update the firmware on a MATRIX VISION device, the following steps are necessary:
13.4.2.1 Step 1: Device selection

Select the device you want to set up. Select the menu item Action and click on Update firmware.

**Note**

It is also possible to select the action with a right click on the device.

![Figure 53: mvDeviceConfigure - Select action](image)

13.4.2.2 Step 2: Start firmware update

- You have to close applications using the device and click Ok.

![Figure 54: mvDeviceConfigure - Close all applications](image)

- You have to select the update file
  - `mvBlueCOUGAR-S:mvBlueCOUGAR-S[MODELNAME]_Update.fpg`
13.4 mvDeviceConfigure

- mvBlueCOUGAR-X: mvBlueCOUGAR-X [MODELNAME]_Update.mvu

- Afterwards, you have to select the GenICam (p. 234) file that came with the firmware e.g. MATRIXVISION_mv
  ION_mvBlueCOUGAR-[MODELNAME]_GigE-[VERSIONNUMBER].zip or MATRIXVISION_mv
  BlueFOX3_GigE-[VERSIONNUMBER].zip.

Figure 55: mvDeviceConfigure - Select firmware file

**Warning**

All current camera settings will be lost when updating the firmware. Network configuration settings (such as static IP settings etc.) on the other hand will not be affected. UserSet settings may or may not be lost, depending on whether the "Persistent UserSet Settings" parameter is set (this issue will be covered later in this chapter).

- Confirm the firmware update.
Afterwards, you will see a progress bar:

If the firmware update is successful, the dialog will disappear.

Note

The firmware update is only necessary in some special cases (e.g. to benefit from a new functionality added to the firmware, to fix a firmware related bug or to update the kernel driver). Before updating the firmware be sure what you are doing and have a look into the change log (versionInfo.txt and/or the manual to see if the update will fix your problem).

The firmware update takes several minutes and during this time the application will not respond!

13.4.3 Preserving UserSet settings when updating the Firmware

For devices that are capable of storing UserSet settings on the device itself (mvBlueCOUGAR-X/XD, mvBlueFOX3, etc.) these settings will by default be preserved during firmware updates since mvIMPACT Acquire 2.9.1. This may lead to slightly longer firmware update times. If UserSets are not used, and their persistence during firmware-updates is not desired, the “Persistent UserSet Settings” in the Settings Submenu can be unchecked: This will also accelerate the firmware update process.
13.4.4 How to disable CPU sleep states a.k.a. C states (< Windows 8)

Modern PC's, notebook's, etc. try to save energy by using a smart power management. For this several hardware manufacturers specified the ACPI standard. The standard defines several power states. For example, if processor load is not needed the processor changes to a power saving (sleep) state automatically and vice versa. Every state change will stop the processor for microseconds. This time is enough to cause image error counts!

See also


To disable the power management on the processor level (so-called "C states"), you can use mvDeviceConfigure:

**Note**

With Windows XP it is only possible to disable the C2 and C3 states. With Windows Vista / 7 all C states (1,2, and 3) will be disabled.

**Warning**

Please be sure you know what you do! To turn off the processor's sleep states will lead to a higher power consumption of your system.

**Note**

Modifying the sleep states using mvDeviceConfigure does only affects the current power scheme. For notebooks this will e.g. make a difference depending on whether the notebook is running on battery or not. E.g. if the sleep states have been disabled while running on battery and then the system is connected to an external power supply, the sleep states might be active again. Thus in order to permanently disable the sleep states, this needs to be done for all power schemes that will be used when operating devices.

1. Start mvDeviceConfigure.

2. Go to tab "Settings" and unselect "CPU Idle States Enabled".
The sleep states can also be enabled or disabled from a script by calling `mvDeviceConfigure` like this:

```bash
mvDeviceConfigure.exe set_processor_idle_states=1 quit
```

or

```bash
mvDeviceConfigure.exe set_processor_idle_states=0 quit
```

The additional `quit` will result in the application to terminate after the new value has been applied.

**Note**

With Windows Vista or newer `mvDeviceConfigure` must be started from a command shell with administrator privileges in order to modify the processors sleep states.

### 13.4.5 Command-line options

It is possible to start mvDeviceConfigure via command line and controlling the starting behavior using parameters. The supported parameter are as follows:
### Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>setid or id</td>
<td>Updates the firmware of one or many devices (syntax: setid=BF000666.5).</td>
</tr>
<tr>
<td>set_processor_idle_states</td>
<td>Changes the C1, C2 and C3 states for ALL processors in the current system.</td>
</tr>
<tr>
<td>set_userset_persistence</td>
<td>Sets the persistency of UserSet settings during firmware updates.</td>
</tr>
<tr>
<td>update_fw or ufw</td>
<td>Updates the firmware of one or many devices.</td>
</tr>
<tr>
<td>update_fw_file or uwf</td>
<td>Updates the firmware of one or many devices. Pass a full path to a text file</td>
</tr>
<tr>
<td>custom_genicam_file or cgf</td>
<td>Specifies a custom GenICam file to be used to open devices for firmware</td>
</tr>
<tr>
<td>update_kd or ukd</td>
<td>Updates the kernel driver of one or many devices.</td>
</tr>
<tr>
<td>ipv4_mask</td>
<td>Specifies an IPv4 address mask to use as a filter for the selected update</td>
</tr>
<tr>
<td>fw_file</td>
<td>Specifies a custom name for the firmware file to use.</td>
</tr>
<tr>
<td>fw_path</td>
<td>Specifies a custom path for the firmware files.</td>
</tr>
<tr>
<td>log_file or lf</td>
<td>Specifies a log file storing the content of this text control upon</td>
</tr>
<tr>
<td>quit or q</td>
<td>Ends the application automatically after all updates have been applied.</td>
</tr>
<tr>
<td>force or f</td>
<td>Forces a firmware update in unattended mode, even if it isn't a newer</td>
</tr>
<tr>
<td>*</td>
<td>Can be used as a wildcard, devices will be searched by serial number AND by</td>
</tr>
</tbody>
</table>

The number of commands that can be passed to the application is not limited.

#### 13.4.5.1 Sample (Windows)

mvDeviceConfigure ufw=BF000666

This will update the firmware of a mvBlueFOX with the serial number BF000666.

mvDeviceConfigure update_fw=BF*

This will update the firmware of ALL mvBlueFOX devices in the current system.

mvDeviceConfigure update_fw=mvBlueFOX-2* 1f=output.txt quit

This will update the firmware of ALL mvBlueFOX-2 devices in the current system, then will store a log file of the executed operations and afterwards will terminate the application.

mvDeviceConfigure setid=BF000666.5
This will assign the device ID ‘5’ to a mvBlueFOX with the serial number BF000666.

```
mvDeviceConfigure ufw=* 
```

This will update the firmware of every device in the system.

```
mvDeviceConfigure ufw=BF000666 ufw=BF000667 
```

This will update the firmware of 2 mvBlueFOX cameras.

```
mvDeviceConfigure ipv4_mask=169.254.*;192.168.100* update_fw=GX* 
```

This will update the firmware of all mvBlueCOUGAR-X devices with a valid IPv4 address that starts with ‘169.254.’ or ‘192.168.100.’.
14 GenICam and Advanced Features

14.1 Introduction

For new applications or to set the device via wxPropView (p.115) we recommend to use the GenICam (p.234) interface layout as it allows the most flexible access to the device features.

After you've set the interface layout to GenICam (p.234) (either programmed or using wxPropView (p.115)), all GenICam (p.234) controls of the device are available.

**Note**

It depends on the device, which controls are supported.

In wxPropView (p.115), you can see them in "Setting -> Base -> Camera -> GenICam":

![Figure 1: wxPropView - GenICam controls](image)

As you can see, there are some controls with and without the prefix "mv".

- "mv" prefix features are unique non-standard features developed by MATRIX VISION.
- Without "mv" are standard features as known from the Standard Feature Naming Convention of GenICam properties (p.242) (SFNC).

All those features are "camera based / device based" features which can also be accessed using the camera with other GenICam (p.234) / GigE Vision (p.238) compliant third-party software.
Note

In GigE Vision timestamps are denoted in "device ticks" but for MATRIX VISION devices this equals microseconds.

Do not mix up the camera based / device based features with the features available in "Setting -> Base -> Image Processing". Theses features are driver based features which are processed by the software and therefore need CPU load.

14.2 Device Control

The "Device Control" contains the features like

<table>
<thead>
<tr>
<th>Feature name (acc. to SFNC)</th>
<th>Property name (acc. to mvIMPA -&gt; CT Acquire (p. 240))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeviceType</td>
<td>deviceType</td>
<td>Returns the device type.</td>
</tr>
<tr>
<td>DeviceScanType</td>
<td>deviceScanType</td>
<td>Scan type of the sensor of the device.</td>
</tr>
<tr>
<td>DeviceVendorName</td>
<td>deviceVendorName</td>
<td>Name of the manufacturer of the device.</td>
</tr>
<tr>
<td>DeviceModelName</td>
<td>deviceModelName</td>
<td>Name of the device model.</td>
</tr>
<tr>
<td>DeviceManufacturerInfo</td>
<td>deviceManufacturerInfo</td>
<td>Manufacturer information about the device.</td>
</tr>
<tr>
<td>DeviceVersion</td>
<td>deviceVersion</td>
<td>Version of the device.</td>
</tr>
<tr>
<td>DeviceFirmwareVersion</td>
<td>deviceFirmwareVersion</td>
<td>Firmware version of the device.</td>
</tr>
<tr>
<td>DeviceSerialNumber</td>
<td>deviceSerialNumber</td>
<td>Serial number of the device.</td>
</tr>
<tr>
<td>DeviceUserID</td>
<td>deviceUserID</td>
<td>User-programmable device identifier.</td>
</tr>
<tr>
<td>DeviceTLVersionMajor</td>
<td>deviceTLVersionMajor</td>
<td>Major version of the transport layer of the device.</td>
</tr>
<tr>
<td>DeviceTLVersionMinor</td>
<td>deviceTLVersionMinor</td>
<td>Minor version of the transport layer of the device.</td>
</tr>
<tr>
<td>DeviceLinkSpeed</td>
<td>deviceLinkSpeed</td>
<td>Indicates the speed of transmission negotiated on the specified Link.</td>
</tr>
<tr>
<td>DeviceTemperature</td>
<td>deviceTemperature</td>
<td>Device temperature.</td>
</tr>
</tbody>
</table>

related to the device and its sensor.

Additionally, MATRIX VISION offers two temperature sensors and for this special features

- device temperature upper limit (mvDeviceTemperatureUpperLimit)
- device temperature lower limit (mvDeviceTemperatureLowerLimit)
- device temperature limit hysteresis (mvDeviceTemperatureLimitHysteresis)

The use case Working with the temperature sensors (p.387) shows how you can work with the temperature sensors.

MATRIX VISION offers also some information properties about the

- FPGA
  - mvDeviceFPGAVersion
- and the image sensor
  - SensorName (since SFNC 2.4, previous versions did use mvDeviceSensorName)
  - mvDeviceSensorColorMode
# 14.3 Image Format Control

The "Image Format Control" contains features like:

<table>
<thead>
<tr>
<th>Feature name (acc. to SFNC)</th>
<th>Property name (acc. to mvMPA→CT Acquire)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SensorWidth</td>
<td>sensorWidth</td>
<td>Effective width of the sensor in pixels.</td>
</tr>
<tr>
<td>SensorHeight</td>
<td>sensorHeight</td>
<td>Effective height of the sensor in pixels.</td>
</tr>
<tr>
<td>Width</td>
<td>width</td>
<td>Width of the image provided by the device (in pixels).</td>
</tr>
<tr>
<td>Height</td>
<td>height</td>
<td>Height of the image provided by the device (in pixels).</td>
</tr>
<tr>
<td>BinningHorizontal, BinningVertical</td>
<td>binningHorizontal, binningVertical</td>
<td>Number of horizontal/vertical photo-sensitive cells to combine together.</td>
</tr>
<tr>
<td>DecimationHorizontal, DecimationVertical</td>
<td>decimationHorizontal, decimationVertical</td>
<td>Sub-sampling of the image. This reduces the resolution (width) of the image by the specified decimation factor.</td>
</tr>
<tr>
<td>TestPattern</td>
<td>testPattern</td>
<td>Selects the type of test image that is sent by the device.</td>
</tr>
<tr>
<td>TestImageSelector (Deprecated)</td>
<td>testImageSelector</td>
<td>Selects the type of test image that is sent by the device. This feature is deprecated. Use 'TestPattern' instead.</td>
</tr>
</tbody>
</table>

related to the format of the transmitted image.

With the test image selector, for example, you can select the type of test image that is send by the device. Here, MATRIX VISION offers two special types:

- **mvBayerRaw** (the Bayer mosaic raw image)
- **mvFFCImage** (the flat-field correction image (p. 294))

Additionally, MATRIX VISION offers numerous additional features like:

- **mvMultiAreaMode**
  which can be used to define multiple AOIs (Areas of Interests) in one image.

  See also
  The use case Working with multiple AOIs (mv Multi Area Mode) (p. 266) shows how this feature works.
### 14.4 Acquisition Control

The "**Acquisition Control**" contains features like

<table>
<thead>
<tr>
<th>Feature name (acc. to SFNC)</th>
<th>Property name (acc. to mvIMPA-CT Acquire)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AcquisitionMode</td>
<td>acquisitionMode</td>
<td>Sets the acquisition mode of the device. The different modes configures a device to send</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• exactly one image (&quot;SingleFrame&quot;),</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• exactly the set number of frames (&quot;MultiFrame&quot;) or</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• images constantly until an application explicitly stops the acquisition again (&quot;Continuous&quot;).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>and can be used for asynchronously grabbing and sending image(s). It works with internal and external hardware trigger where the edge is selectable. The external trigger uses ImageRequestTimeout (ms) to time out.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The chapter <strong>How to see the first image</strong> (p.120) shows how to acquire images with wxPropView (p.115).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The use case <strong>Acquiring a number of images</strong> (p.248) shows how to acquire a number of images; also triggered externally.</td>
</tr>
<tr>
<td>AcquisitionStart</td>
<td>acquisitionStart</td>
<td>Starts the acquisition of the device.</td>
</tr>
<tr>
<td>AcquisitionStop</td>
<td>acquisitionStop</td>
<td>Stops the acquisition of the device at the end of the current Frame.</td>
</tr>
<tr>
<td>AcquisitionAbort</td>
<td>acquisitionAbort</td>
<td>Aborts the acquisition immediately.</td>
</tr>
<tr>
<td>AcquisitionFrameRate</td>
<td>acquisitionFrameRate</td>
<td>Controls the acquisition rate (in Hertz) at which the frames are captured. Some cameras support a special internal trigger mode that allows more exact frame rates. This feature keeps the frame rate constant to an accuracy of +/-0.005 fps at 200 fps. This is achieved using frames with a length difference of up to 1 us. Please check in the sensor summary (p.96) if this feature exists for the requested sensor.</td>
</tr>
</tbody>
</table>
### 14.4 Acquisition Control

<table>
<thead>
<tr>
<th>TriggerSelector</th>
<th>triggerSelector</th>
<th>Selects the type of trigger to configure. A possible option is <code>mvTimestampReset</code>. The use case about <code>mvTimestampReset</code> is available (p.351).</th>
</tr>
</thead>
<tbody>
<tr>
<td>TriggerOverlap[TriggerSelector]</td>
<td>triggerOverlap</td>
<td>Specifies the type trigger overlap permitted with the previous frame. <code>TriggerOverlap</code> is only intended for external trigger (which is usually non-overlapped: i.e. exposure and readout are sequentially). This leads to minimal latency / jitter between trigger and exposure. However, the camera accepts a new trigger (the exposure time earlier) before the end of the transmission of the current image. Maximum frame rate in triggered mode = frame rate of continuous mode. This however leads to higher latency / jitter between trigger and exposure. A trigger will be not latched if it occurs before this moment (trigger is accurate in time). See also Principles of overlapped and pipelined trigger (p.241).</td>
</tr>
<tr>
<td>ExposureMode</td>
<td>exposureMode</td>
<td>Sets the operation mode of the exposure (or shutter).</td>
</tr>
<tr>
<td>ExposureTime</td>
<td>exposureTime</td>
<td>Sets the exposure time (in microseconds) when <code>ExposureMode</code> is <code>Timed</code> and <code>ExposureAuto</code> is <code>Off</code>.</td>
</tr>
<tr>
<td>ExposureAuto</td>
<td>exposureAuto</td>
<td>Sets the automatic exposure mode when <code>ExposureMode</code> is <code>Timed</code>.</td>
</tr>
<tr>
<td>etc.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Related to the image acquisition, including the triggering mode.

Additionally, MATRIX VISION offers numerous additional features like:

- `mvShutterMode`  
  which selects the shutter mode of the CMOS sensors like rolling shutter or global shutter.

- `mvDefectivePixelEnable`  
  which activates the sensor's defective pixel correction.
See also


• **mvExposureAutoSpeed**
  which determines the increment or decrement size of exposure value from frame to frame.

• **mvExposureAutoDelayImages**
  the number of frames that the AEC must skip before updating the exposure register.

• **mvExposureAutoAverageGrey**
  common desired average grey value (in percent) used for Auto Gain Control (AGC) and Auto Exposure Control (AEC).

• **mvExposureAutoAOIMode**
  common AutoControl AOI used for Auto Gain Control (AGC), Auto Exposure Control (AEC) and Auto White Balance (AWB).

• **mvAcquisitionMemoryMode**
  MATRIX VISION offers three additional acquisition modes which use the internal memory of the mvBlueCOUGAR-X:

  – **mvRecord** *(p. 249)*
    which is used to store frames in memory.

  – **mvPlayback** *(p. 249)*
    which transfers stored frames.

  – **mvPretrigger** *(p. 249)*
    which stores frames in memory to be transferred after trigger.
    To define the number of frames to acquire before the occurrence of an AcquisitionStart or AcquisitionActive trigger, you can use **mvPretriggerFrameCount**.
    See also
    The use case **Recording sequences with pre-trigger** *(p. 249)* shows how this feature works.

• **mvAcquisitionMemoryMaxFrameCount**
  which shows the maximum of frames the internal memory can save.

  See also
  The use case **Working with burst mode buffer** *(p. 270)* lists some maximum frame counts of some mvBlueCOUGAR-X models.

• **mvSmearReduction**
  smear reduction in triggered and non-overlapped mode.

• **mvSmartFrameRecallEnable**
  which configures the internal memory to store each frame (that gets transmitted to the host) in full resolution.
14.5 Counter And Timer Control

The use case **SmartFrameRecall** (p. 274) shows how this feature works.

For "**Exposure Auto Mode**", in **wxPropView** (p. 115) just select in "Exposure Auto" "Continuous". Afterwards, you have the possibility to set lower and upper limit, average gray combined with AOI setting:

![Figure 2: Acquire Control -> Exposure Auto](image)

- **mvSmartFrameRecallFrameSkipRatio**
  When set to a value != 0, the smaller frames get thinned out. AOI requests can still be done for all frames.

- **mvSmartFrameRecallTimestampLookupAccuracy**
  is needed for the SkipRatio feature since you **don't know** the timestamps of the **internal** frames. This value defines the strictness of the timestamp-check for the recalled image (given in us).

14.5 Counter And Timer Control

The "**Counter And Timer Control**" is a powerful feature which MATRIX VISION customers already know under the name Hardware Real-Time Controller (HRTC). **mvBlueCOUGAR-X** provides:

- 4 counters for counting events or external signals (compare number of triggers vs. number of frames; over-trigger) and
- 2 timers.

Counter and Timers can be used, for example,
• for pulse width modulation (PWM) (p.339) and
• to generate output signals of variable length, depending on conditions in camera.

This achieves complete HRTC functionality which supports following applications:

• frame rate by timer
• exposure time by timer
• pulse width at input

The "Counter And Timer Control" contains features like

<table>
<thead>
<tr>
<th>Feature name (acc. to SFNC)</th>
<th>Property name (acc. to mvIMPA (\rightarrow) CT Acquire (p.240))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CounterSelector</td>
<td>counterSelector</td>
<td>Selects which counter to configure.</td>
</tr>
<tr>
<td>CounterEventSource[CounterSelector]</td>
<td>counterEventSource</td>
<td>Selects the events that will be the source to increment the counter.</td>
</tr>
<tr>
<td>CounterEventActivation[CounterSelector]</td>
<td>counterEventActivation</td>
<td>Selects the activation mode event source signal.</td>
</tr>
<tr>
<td>TimerSelector</td>
<td>timerSelector</td>
<td>Selects which timer to configure.</td>
</tr>
<tr>
<td>TimerDuration[TimerSelector]</td>
<td>timerDuration</td>
<td>Sets the duration (in microseconds) of the timer pulse.</td>
</tr>
<tr>
<td>TimerDelay[TimerSelector]</td>
<td>timerDelay</td>
<td>Sets the duration (in microseconds) of the delay.</td>
</tr>
<tr>
<td>etc.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

related to the usage of programmable counters and timers.

Because there are many ways to use this feature, the list of use cases is long and not finished yet:

• Processing triggers from an incremental encoder (p.336)
• Creating different exposure times for consecutive images (p.343)
• Creating synchronized acquisitions using timers (p.414)
• Generating a pulse width modulation (PWM) (p.339)
• Outputting a pulse at every other external trigger (p.341)
• Generating very long exposure times (p.264)

14.6 Analog Control

The "Analog Control" contains features like
### 14.6 Analog Control

<table>
<thead>
<tr>
<th>Feature name (acc. to SFNC (p. 242))</th>
<th>Property name (acc. to mvIMPA - CT Acquire (p. 240))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GainSelector</td>
<td>gainSelector</td>
<td>Selects which gain is controlled by the various gain features.</td>
</tr>
<tr>
<td>Gain[GainSelector]</td>
<td>gain</td>
<td>Controls the selected gain as an absolute physical value [in dB].</td>
</tr>
<tr>
<td>GainAuto[GainSelector]</td>
<td>gainAuto</td>
<td>Sets the automatic gain control (AGC) mode.</td>
</tr>
<tr>
<td>GainAutoBalance</td>
<td>gainAutoBalance</td>
<td>Sets the mode for automatic gain balancing between the sensor color channels or taps.</td>
</tr>
<tr>
<td>BlackLevelSelector</td>
<td>blackLevelSelector</td>
<td>Selects which black Level is controlled by the various black Level features.</td>
</tr>
<tr>
<td>BalanceWhiteAuto</td>
<td>balanceWhiteAuto</td>
<td>Controls the mode for automatic white balancing between the color channels.</td>
</tr>
<tr>
<td>Gamma</td>
<td>gamma</td>
<td>Controls the gamma correction of pixel intensity.</td>
</tr>
<tr>
<td>etc.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Related to the video signal conditioning in the analog domain.

Additionally, MATRIX VISION offers:

- `mvBalanceWhiteAuto` functions and
- `mvGainAuto` functions.

In `wxPropView` (p. 115) just select in "Gain Auto" (AGC) "Continuous". Afterwards, you have the possibility to set minimum and maximum limit combined with AOI setting:
See also

Optimizing the color fidelity of the camera (p. 298)

14.7 mv Logic Gate Control

The "mv Logic Gate Control" contains features like

<table>
<thead>
<tr>
<th>Feature name (acc. to SFNC)</th>
<th>Property name (acc. to mvMPA--CT Acquire)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mvLogicGateANDSelector</td>
<td>mvLogicGateANDSelector</td>
<td>Selects the AND gate to configure.</td>
</tr>
<tr>
<td>mvLogicGateANDSource1</td>
<td>mvLogicGateANDSource1</td>
<td>Selects the first input signal of the AND gate selected by mvLogic--GateANDSelector.</td>
</tr>
<tr>
<td>mvLogicGateANDSource2</td>
<td>mvLogicGateANDSource2</td>
<td>Selects the second input signal of the AND gate selected by mv--LogicGateANDSelector.</td>
</tr>
<tr>
<td>mvLogicGateORSelector</td>
<td>mvLogicGateORSelector</td>
<td>Selects the OR gate to configure.</td>
</tr>
<tr>
<td>mvLogicGateORSource1</td>
<td>mvLogicGateORSource1</td>
<td>Selects the first input signal of the OR gate selected by mvLogic--GateORSelector.</td>
</tr>
</tbody>
</table>
The use case **Creating different exposure times for consecutive images** (p. 343) shows how you can create different exposure times with timers, counters and the logic gate functionality.

See also

**Triggering of an indefinite sequence with precise starting time** (p. 350)

### 14.8 Color Transformation Control

The "**Color Transformation Control**" contains features like

<table>
<thead>
<tr>
<th>Feature name (acc. to SFNC)</th>
<th>Property name (acc. to mvIMPA←CT Acquire (p. 240))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ColorTransformationSelector</td>
<td>colorTransformationEnable</td>
<td>Activates the selected color transformation module.</td>
</tr>
<tr>
<td>ColorTransformationSelector</td>
<td>colorTransformationSelector</td>
<td>Selects which color transformation module is controlled by the various color transformation features.</td>
</tr>
<tr>
<td>ColorTransformationValue</td>
<td>colorTransformationValue</td>
<td>Represents the value of the selected Gain factor or Offset inside the transformation matrix.</td>
</tr>
<tr>
<td>ColorTransformationValueSelector</td>
<td>colorTransformationValueSelector</td>
<td>Selects the gain factor or Offset of the Transformation matrix to access in the selected color transformation module.</td>
</tr>
</tbody>
</table>

related to the control of the color transformation.

This control offers an enhanced color processing for optimum color fidelity using a **color correction matrix** (CCM) and enables

- 9 coefficients values \( \text{Gain}_{00} \ldots \text{Gain}_{22} \) and
- 3 offset values \( \text{Offset}_0 \ldots \text{Offset}_2 \)

to be entered for \( \text{RGB}_{\text{IN}} \rightarrow \text{RGB}_{\text{OUT}} \) transformation. This can be used to optimize specific colors or specific color temperatures.
Coefficients will be made available for sensor models and special requirements on demand.

See also

Optimizing the color fidelity of the camera (p. 298)

14.9 mv Flat Field Correction Control

The "mv Flat Field Correction Control" contains features like:

<table>
<thead>
<tr>
<th>Feature name (acc. to SFNC p. 242)</th>
<th>Property name (acc. to mvMPA&lt;sup&gt;CT Acquire&lt;/sup&gt; p. 240)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mvFFCEnable</td>
<td>mvFFCCalibrationImageCount</td>
<td>Enables the flat field correction.</td>
</tr>
<tr>
<td>mvFFCCalibrate</td>
<td></td>
<td>The number of images to use for the calculation of the correction image.</td>
</tr>
<tr>
<td>mvFFCCalibrate</td>
<td></td>
<td>Starts the calibration of the flat field correction.</td>
</tr>
</tbody>
</table>

related to control the devices Flat Field Correction parameters.

The use case Flat-Field Correction (p. 294) shows how this control can be used.

14.10 Event Control

The "Event Control" contains features like:

<table>
<thead>
<tr>
<th>Feature name (acc. to SFNC p. 242)</th>
<th>Property name (acc. to mvMPA&lt;sup&gt;CT Acquire&lt;/sup&gt; p. 240)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EventSelector</td>
<td>eventSelector</td>
<td>Selects which Event to signal to the host application.</td>
</tr>
<tr>
<td>EventNotification[EventSelector]</td>
<td>eventNotification</td>
<td>Activate or deactivate the notification to the host application of the occurrence of the selected Event.</td>
</tr>
<tr>
<td>EventFrameTriggerData</td>
<td>eventFrameTriggerData</td>
<td>Category that contains all the data features related to the FrameTrigger event.</td>
</tr>
<tr>
<td>EventFrameTrigger</td>
<td>eventFrameTrigger</td>
<td>Returns the unique identifier of the FrameTrigger type of event.</td>
</tr>
</tbody>
</table>
14.11 Chunk Data Control

The "Chunk Data Control" contains features like

<table>
<thead>
<tr>
<th>Feature name (acc. to SFNC (p. 242))</th>
<th>Property name (acc. to mvIMPA CT Acquire (p. 240))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChunkModeActive</td>
<td>chunkModeActive</td>
<td>Activates the inclusion of chunk data in the payload of the image.</td>
</tr>
<tr>
<td>ChunkSelector</td>
<td>chunkSelector</td>
<td>Selects which chunk to enable or control.</td>
</tr>
<tr>
<td>ChunkEnable[ChunkSelector]</td>
<td>chunkEnable</td>
<td>Enables the inclusion of the selected chunk data in the payload of the image.</td>
</tr>
<tr>
<td>ChunkImage</td>
<td>chunkImage</td>
<td>Returns the entire image data included in the payload.</td>
</tr>
<tr>
<td>ChunkOffsetX</td>
<td>ChunkOffsetX</td>
<td>Returns the offset x of the image included in the payload.</td>
</tr>
<tr>
<td>ChunkOffsetY</td>
<td>chunkOffsetY</td>
<td>Returns the offset y of the image included in the payload.</td>
</tr>
</tbody>
</table>

related to the generation of Event notifications by the device.

The use case Working with Event Control (p. 284) shows how this control can be used.
<table>
<thead>
<tr>
<th>ChunkWidth</th>
<th>chunkWidth</th>
<th>Returns the width of the image included in the payload.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChunkHeight</td>
<td>chunkHeight</td>
<td>Returns the height of the image included in the payload.</td>
</tr>
<tr>
<td>ChunkPixelFormat</td>
<td>chunkPixelFormat</td>
<td>Returns the pixel format of the image included in the payload.</td>
</tr>
<tr>
<td>ChunkPixelDynamicRangeMin</td>
<td>chunkPixelDynamicRangeMin</td>
<td>Returns the minimum value of dynamic range of the image included in the payload.</td>
</tr>
<tr>
<td>ChunkPixelDynamicRangeMax</td>
<td>chunkPixelDynamicRangeMax</td>
<td>Returns the maximum value of dynamic range of the image included in the payload.</td>
</tr>
<tr>
<td>ChunkTimestamp</td>
<td>chunkTimestamp</td>
<td>Returns the timestamp of the image included in the payload at the time of the FrameStart internal event.</td>
</tr>
</tbody>
</table>

etc.

related to the Chunk Data Control.

A description can be found in the image acquisition section of the "mvIMPACT Acquire API" manuals.

14.12 File Access Control

The "File Access Control" contains features like

<table>
<thead>
<tr>
<th>Feature name (acc. to SFNC (p. 242))</th>
<th>Property name (acc. to mvIMPA→CT Acquire (p. 240))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FileSelector</td>
<td>fileSelector</td>
<td>Selects the target file in the device.</td>
</tr>
<tr>
<td>FileOperationSelector[FileSelector]</td>
<td>fileOperationSelector</td>
<td>Selects the target operation for the selected file in the device.</td>
</tr>
<tr>
<td>FileOperationExecute[FileOperationSelector][FileOperationSelector]</td>
<td>fileOperationExecute</td>
<td>Executes the operation selected by FileOperationSelector on the selected file.</td>
</tr>
<tr>
<td>FileOpenMode[FileSelector]</td>
<td>fileOpenMode</td>
<td>Selects the access mode in which a file is opened in the device.</td>
</tr>
<tr>
<td>FileAccessBuffer</td>
<td>FileAccessBuffer</td>
<td>Defines the intermediate access buffer that allows the exchange.</td>
</tr>
</tbody>
</table>

etc.

related to the File Access Control that provides all the services necessary for generic file access of a device.

The use case Working with the UserFile section (Flash memory) (p. 384) shows how this control can be used.

14.13 mv Serial Interface Control

The "mv Serial Interface Control" contains features like
related to control the devices Serial Interface Control parameters. It enables the camera to be controlled via serial interface.

The use case Working with the serial interface (mv Serial Interface Control) (p. 397) shows how you can work with the serial interface control.

See also

Working with the serial interface (mv Serial Interface Control) (p. 397)

14.14 Digital I/O Control

The "Digital I/O Control" contains features like

<table>
<thead>
<tr>
<th>Feature name (acc. to SFNC) (p. 242)</th>
<th>Property name (acc. to mvIMPA← CT Acquire (p. 240))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mvSerialInterfaceMode</td>
<td>mvSerialInterfaceMode</td>
<td>States the interface mode of the serial interface.</td>
</tr>
<tr>
<td>mvSerialInterfaceEnable</td>
<td>mvSerialInterfaceEnable</td>
<td>Controls whether the serial interface is enabled or not.</td>
</tr>
<tr>
<td>mvSerialInterfaceBaudRate</td>
<td>mvSerialInterfaceBaudRate</td>
<td>Serial interface clock frequency.</td>
</tr>
<tr>
<td>mvSerialInterfaceASCIIBuffer</td>
<td>mvSerialInterfaceASCIIBuffer</td>
<td>Buffer for exchanging ASCII data over serial interface.</td>
</tr>
<tr>
<td>mvSerialInterfaceWrite</td>
<td>mvSerialInterfaceWrite</td>
<td>Command to write data from the serial interface.</td>
</tr>
<tr>
<td>mvSerialInterfaceRead</td>
<td>mvSerialInterfaceRead</td>
<td>Command to read data from the serial interface.</td>
</tr>
</tbody>
</table>

etc.

related to the control of the general input and output pins of the device.

Additionally, MATRIX VISION offers:

- mvLineDebounceTimeRisingEdge and
- mvLineDebounceTimeFallingEdge functionality.

A description of these functions can be found in the use case Creating a debouncing filter at the inputs (p. 355).

How you can test the digital inputs and outputs is described here: Testing the digital inputs (p. 147). The use case Creating synchronized acquisitions using timers (p. 414) is a further example which shows you how to work with digital inputs and outputs.
14.15 Encoder Control

The "Encoder Control" contains features like:

<table>
<thead>
<tr>
<th>Feature name (acc. to SFNC)</th>
<th>Property name (acc. to mvIMPA→CT Acquire)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EncoderSourceA</td>
<td>encoderSourceA</td>
<td>Selection of the A input line.</td>
</tr>
<tr>
<td>EncoderSourceB</td>
<td>encoderSourceB</td>
<td>Selection of the B input line.</td>
</tr>
<tr>
<td>EncoderMode [FourPhase]</td>
<td>encoderMode</td>
<td>The counter increments or decrements 1 for every full quadrature cycle.</td>
</tr>
<tr>
<td>EncoderDivider</td>
<td>encoderDivider</td>
<td>Sets how many Encoder increments/decrements that are needed generate an encoder output signal.</td>
</tr>
<tr>
<td>EncoderOutputMode</td>
<td>encoderOutputMode</td>
<td>Output signals are generated at all new positions in one direction. If the encoder reverses no output pulse are generated until it has again passed the position where the reversal started.</td>
</tr>
<tr>
<td>EncoderValue</td>
<td>encoderValue</td>
<td>Reads or writes the current value of the position counter of the selected Encoder. Writing to EncoderValue is typically used to set the start value of the position counter.</td>
</tr>
</tbody>
</table>

related to the usage if quadrature encoders.

The following figure explains the different EncoderOutputModes:

![Figure 5: EncoderOutputModes](image)

Additionally, the Encoder is also available as TriggerSource and as an EventSource.

A description of incremental encoder's principle can be found in the use case Processing triggers from an incremental encoder (p. 336).

14.16 LUT Control

The "LUT Control" contains features like...
14.16  LUT Control

<table>
<thead>
<tr>
<th>Feature name (acc. to SFNC p.242)</th>
<th>Property name (acc. to mvIMPA-CT Acquire p.240)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LUTSelector</td>
<td>LUTSelector</td>
<td>Selects which LUT to control.</td>
</tr>
<tr>
<td>LUTEnable[LUTSelector]</td>
<td>LUTEnable</td>
<td>Activates the selected LUT.</td>
</tr>
<tr>
<td>LUTIndex[LUTSelector]</td>
<td>LUTIndex</td>
<td>Controls the index (offset) of the coefficient to access in the selected LUT.</td>
</tr>
<tr>
<td>LUTValue[LUTSelector][LUTIndex]</td>
<td>LUTValue</td>
<td>Returns the value at entry LUT Index of the LUT selected by LUTSelector.</td>
</tr>
<tr>
<td>LUTValueAll[LUTSelector]</td>
<td>LUTValueAll</td>
<td>Allows access to all the LUT coefficients with a single read/write operation.</td>
</tr>
<tr>
<td>mvLUTType</td>
<td></td>
<td>Describes which type of LUT is used for the current LUTSelector.</td>
</tr>
<tr>
<td>mvLUTInputData</td>
<td></td>
<td>Describes the data the LUT is applied to (e.g. Bayer, RGB, or gray data).</td>
</tr>
<tr>
<td>mvLUTMapping</td>
<td></td>
<td>Describes the LUT mapping (e.g. 10 bit -&gt; 12 bit).</td>
</tr>
</tbody>
</table>

related to the look-up table (LUT) control.

The look-up table (LUT) is a part of the signal path in the camera and maps data of the ADC into signal values. The LUT can be used e.g. for:

- High precision gamma
- Non linear enhancement (e.g. S-Shaped)
- Inversion (default)
- Negative offset
- Threshold
- Level windows
- Binarization

This saves (approx. 5%) CPU load, works on the fly in the FPGA of camera, is less noisy and there are no missing codes after Gamma stretching. Three read-only registers describe the LUT that is selected using the LUTSelector register:

14.16.1  mvLUTType

There are two different types of LUTs available in MATRIX VISION cameras:

- Direct LUTs define a mapping for each possible input value, for example a 12 -> 10 bit direct LUT has $2^{12}$ entries and each entry has 10 bit.
- Interpolated LUTs do not define a mapping for every possible input value, instead the user defines an output value for equidistant nodes. In between the nodes linear interpolation is used to calculate the correct output value.

Considering a 10 -> 10 bit interpolated LUT with 256 nodes (as usually used in MATRIX VISION cameras), the user defines a 10 bit output value for 256 equidistant nodes beginning at input value 0, 4, 8, 12, 16 and so on. For input values in between the nodes linear interpolation is used.
14.16.2 mvLUTInputData

This register describes on which data the LUT is applied to:

- Bayer means that the LUT is applied to raw Bayer data, thus (depending on the de-Bayer algorithm) a manipulation of one pixel may also affect other pixels in its neighborhood.
- Gray means that the LUT is applied to gray data.
- RGB means that the LUT is applied to RGB data (i.e. after de-Bayering). Normally this is used to change the luminance on an RGB image and the LUT is applied to all three channels.

14.16.3 mvLUTMapping

This register describes the mapping of the currently selected LUT, e.g. "map_10To10" means that a 10 bit input value is mapped to a 10 bit output values whereas "map_12To10" means that a 12 bit input value is mapped to a 10 bit output value.

14.16.4 LUT support in MATRIX VISION cameras

<table>
<thead>
<tr>
<th>mvBlueCOUGAR</th>
<th>LUTSelector</th>
<th>LUT type</th>
<th>LUT mapping</th>
<th>LUT input data</th>
</tr>
</thead>
</table>

MATRIX VISION GmbH
<table>
<thead>
<tr>
<th>Luminance</th>
<th>Direct</th>
<th>map_10To10</th>
<th>Gray</th>
</tr>
</thead>
<tbody>
<tr>
<td>X100wG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X100wG-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X102bG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X102bG-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X102bG-POEI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X102dG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X102dG-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X102dG-POEI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X102eG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X102eGE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X104eG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X104eG-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X104eG-PLC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X102eG-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X102eGE-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X102eG-POEI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X102eGE-POEI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X105G</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X120aG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X120aG-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X120bG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X120bG-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X120bG-PLC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X120dG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X120dG-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X120dG-POEI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X122G</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X122G-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X122G-PLC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X122G-POEI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X123G</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X123G-PLC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X123G-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X123G-POEI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X124G</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X124G-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X1010G</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X1010G-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X225G-POEI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X225aG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X225G</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X225G-PLC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X225G-1211-ET</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Matrix Vision Model</td>
<td>Luminance</td>
<td>Interpolated</td>
<td>RGB</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------</td>
<td>--------------</td>
<td>------</td>
</tr>
<tr>
<td>X100wC</td>
<td>Red</td>
<td>Direct</td>
<td>Bayer</td>
</tr>
<tr>
<td>X100wC-POE</td>
<td>Green</td>
<td>Direct</td>
<td>Bayer</td>
</tr>
<tr>
<td>X102bC</td>
<td>Blue</td>
<td>Direct</td>
<td>Bayer</td>
</tr>
<tr>
<td>X102bC-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X102bC-POEI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X102dC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X102dC-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X102dC-POEI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X102eC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X104C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X104C-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X104C-POEI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X105C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X120aC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X120aC-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X120bC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X120bC-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X120bC-PLC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X120dC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X120dC-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X120dC-POEI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X122C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X122C-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X122C-PLC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X123C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X123C-PLC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X123C-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X123C-POEI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X124C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X124C-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X1010C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X1010C-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X225aC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X225C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X104G</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X104G-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X104aG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X104a12G</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X104bG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X104bG-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X125aG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X125aG-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X125aG-POEI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X104C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X104C-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X104aC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X104a12C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X104bC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X104bC-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X125aC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X125aC-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X125aC-POEI</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Notes:**
- **Luminance** refers to the color channel used for luminance.
- **Interpolated** indicates whether the output is interpolated or not.
- **RGB** indicates the color space.

**Additional Observations:**
- The table lists various models along with their corresponding luminance, interpolation, and RGB output types.
- The table is structured to show a clear comparison between different models and their output characteristics.
<table>
<thead>
<tr>
<th>Luminance</th>
<th>Direct</th>
<th>map_12To9</th>
<th>Gray</th>
</tr>
</thead>
<tbody>
<tr>
<td>X100fG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X100fG-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X100fG-POEI-X102fG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X102fG-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X102fG-POEI-X102kG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X102kG-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X102kG-POEI-X102mG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X102mG-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X102mG-POEI-X102nG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X102nG-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X102nG-POEI-X104fIG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X104fIG-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X104fIG-POEI-X104iG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X104iG-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X104iG-POEI-X105bG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X105bG-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X105bG-POEI-X105pG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X105pG-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X105pG-POEI-X107bG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X107bG-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X107bG-POEI-X109bG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X109bG-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X109bG-POEI-X1012bG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X1012bG-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X1012bG-POEI-X1020G</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X1020G-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X1020G-POEI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X100fC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X100fC-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X100fC-POEI-X102fC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X102fC-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X102fC-POEI-X102kC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X102kC-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X102kC-POEI-X102mC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X102mC-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X102mC-POEI-X102nC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X102nC-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X102nC-POEI-X104fIC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X104fIC-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X104fIC-POEI-X104iC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X104iC-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X104iC-POEI-X105bC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X105bC-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X105bC-POEI-X105pC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X105pC-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X105pC-POEI-X107bC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X107bC-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X107bC-POEI-X109bC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X109bC-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X109bC-POEI-X1012bC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X1012bC-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X1012bC-POEI-X1020C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X1020C-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X1020C-POEI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X2012bG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X2012bG-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X2012bG-POEI</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Red</th>
<th>Direct</th>
<th>map_10To9</th>
<th>Bayer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Green</td>
<td>Direct</td>
<td>map_10To9</td>
<td>Bayer</td>
</tr>
<tr>
<td>Blue</td>
<td>Direct</td>
<td>map_10To9</td>
<td>Bayer</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Luminance</th>
<th>Direct</th>
<th>map_12To9</th>
<th>Gray</th>
</tr>
</thead>
<tbody>
<tr>
<td>X2012bG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X2012bG-POE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X2012bG-POEI</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

MATRIX VISION GmbH
<table>
<thead>
<tr>
<th>Model</th>
<th>Red</th>
<th>Direct</th>
<th>Map</th>
<th>Bayer</th>
</tr>
</thead>
<tbody>
<tr>
<td>-X2012bC</td>
<td>Green</td>
<td>Direct</td>
<td>map_10To9</td>
<td>Bayer</td>
</tr>
<tr>
<td>-X2012bC-POE</td>
<td>Blue</td>
<td>Direct</td>
<td>map_10To9</td>
<td>Bayer</td>
</tr>
<tr>
<td>-X2012bC-POE1</td>
<td></td>
<td></td>
<td>map_10To9</td>
<td>Bayer</td>
</tr>
<tr>
<td>-XD124aG</td>
<td>Luminance</td>
<td>Direct</td>
<td>map_10To12</td>
<td>Gray</td>
</tr>
<tr>
<td>-XD126G</td>
<td>Red</td>
<td>Direct</td>
<td>map_10To12</td>
<td>Bayer</td>
</tr>
<tr>
<td>-XD129G</td>
<td>Green</td>
<td>Direct</td>
<td>map_10To12</td>
<td>Bayer</td>
</tr>
<tr>
<td>-XD129bG</td>
<td>Blue</td>
<td>Direct</td>
<td>map_10To12</td>
<td>Bayer</td>
</tr>
<tr>
<td>-XD126aG</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-XD129aG</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-XD204G</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-XD204aG</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-XD204bG</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-XD204baG</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-XD104G</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-XD104aG</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-XD104a12G</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-XD104bG</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-XD104baG</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-XD1212aG</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-XD124aC</td>
<td>Luminance</td>
<td>Interpolated</td>
<td>map_10To10</td>
<td>RGB</td>
</tr>
<tr>
<td>-XD126C</td>
<td>Red</td>
<td>Direct</td>
<td>map_8To12</td>
<td>Bayer</td>
</tr>
<tr>
<td>-XD129C</td>
<td>Green</td>
<td>Direct</td>
<td>map_8To12</td>
<td>Bayer</td>
</tr>
<tr>
<td>-XD129bC</td>
<td>Blue</td>
<td>Direct</td>
<td>map_8To12</td>
<td>Bayer</td>
</tr>
<tr>
<td>-XD126aC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-XD129aC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-XD204C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-XD204aC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-XD204bC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-XD204baC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-XD104C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-XD104aC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-XD104bC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-XD104baC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-XD1212aC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-XD104dG</td>
<td>Luminance</td>
<td>Direct</td>
<td>map_12To9</td>
<td>Gray</td>
</tr>
<tr>
<td>-XD105aG</td>
<td>Red</td>
<td>Direct</td>
<td>map_12To9</td>
<td>Gray</td>
</tr>
<tr>
<td>-XD104hG</td>
<td>Green</td>
<td>Direct</td>
<td>map_12To9</td>
<td>Gray</td>
</tr>
<tr>
<td>-XD107G</td>
<td>Blue</td>
<td>Direct</td>
<td>map_12To9</td>
<td>Gray</td>
</tr>
<tr>
<td>-XD107bG</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-XD109bG</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-XD1012bG</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-XD1031G</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-XD104aC</td>
<td>Luminance</td>
<td>Interpolated</td>
<td>map_10To10</td>
<td>RGB</td>
</tr>
<tr>
<td>-XD105aC</td>
<td>Red</td>
<td>Direct</td>
<td>map_10To9</td>
<td>Bayer</td>
</tr>
<tr>
<td>-XD104hC</td>
<td>Green</td>
<td>Direct</td>
<td>map_10To9</td>
<td>Bayer</td>
</tr>
<tr>
<td>-XD107C</td>
<td>Blue</td>
<td>Direct</td>
<td>map_10To9</td>
<td>Bayer</td>
</tr>
<tr>
<td>-XD107bC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-XD109bC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-XD1012bC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-XD1031C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-XD124aG-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD126G-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD129G-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD129bG-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD126aG-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD129aG-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD204G-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD204aG-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD204bG-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD204baG-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD104G-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD104aG-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD104a12G-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD104bG-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD104baG-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD1212aG-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD124aC-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD126C-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD129C-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD129bC-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD126aC-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD129aC-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD204C-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD204aC-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD204bC-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD204baC-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD104C-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD104aC-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD104bC-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD104baC-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD1212aC-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD104dG-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD105aG-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD104hG-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD107G-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD107bG-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD109bG-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD1012bG-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD1031G-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD104aC-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD105aC-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD104hC-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD107C-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD107bC-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD109bC-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD1012bC-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-XD1031C-POE</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

**Note:** The table includes various camera models with their respective color channels and mapping functions. The models are categorized by their unique identifiers. The color channels shown include Red, Green, and Blue, with each channel having different mapping functions such as Direct, Interpolated, and Map functions. The Bayer patterns are also listed, indicating the format in which the results are captured or displayed.
<table>
<thead>
<tr>
<th>mvBlueCOUGAR-3X123C</th>
<th>Red (1st head)</th>
<th>Interpolated</th>
<th>map_12To12</th>
<th>Bayer</th>
<th>Blue (3rd head)</th>
<th>map_12To9</th>
<th>Gray</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Green (2nd head)</td>
<td>Interpolated</td>
<td>map_12To12</td>
<td></td>
<td>Blue (3rd head)</td>
<td>map_12To9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Luminance</td>
<td>Direct</td>
<td>map_10To12</td>
<td></td>
<td></td>
<td></td>
<td>Gray</td>
</tr>
<tr>
<td>-1100 [G]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1140 [GW]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1031 [G]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1012d [G]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1012b [G]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1013 [G]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1013GE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1020 [G]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1020a [G]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1012bc</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1012dC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1013C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1020C - 1031C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1100C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1140C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2004G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2016G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2024G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2024aG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2032G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2032aG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2051G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2051aG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2051pG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2064G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2071aG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2089G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2089aG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2124G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2124aG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2124rG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2205G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-4315G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2004C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2016C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2024C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2024aC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2032C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2032aC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2051C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2051aC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2051pC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2064C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2071aC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2089C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2089aC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2124C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2124aC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2124rC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2205C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-4315C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
14.17 Sequencer Control

14.17.1 Sequencer overview

The purpose of a sequencer is to allow the user of a camera to define a series of feature sets for image acquisition which can consecutively be activated during the acquisition by the camera. Accordingly, the proposed sequence is configured by a list of parameter sets. Each of these sequencer sets contains the settings for a number of camera features. Similar to user sets, the actual settings of the camera are overwritten when one of these sequencer sets is loaded. The order in which the features are applied to the camera depends on the design of the vendor. It is recommended to apply all the image related settings to the camera, before the first frame of this sequence is captured. The sequencer sets can be loaded and saved by selecting them using SequencerSetSelector. The Execution of the sequencer is completely controlled by the device.

(quoted from the GenICam SFNC 2.3)

14.17.2 Configuration of a sequencer set

The index of the adjustable sequencer set is given by the SequencerSetSelector. The number of available sequencer sets is directly given by the range of this feature. The features which are actually part of a sequencer set are defined by the camera manufacturer. These features can be read by SequencerFeatureSelector and activated by SequencerFeatureEnable[SequencerFeatureSelector]. This configuration is the same for all Sequencer Sets. To configure a sequencer set the camera has to be switched into configuration mode by SequencerConfigurationMode. Then the user has to select the desired sequencer set he wants to modify with the SequencerSetSelector. After the user has changed all the needed camera settings it is possible to store all these settings within a selected sequencer set by executing SequencerSetSave[SequencerSetSelector]. The user can also read back these settings by executing SequencerSetLoad[SequencerSetSelector]. To permit a flexible usage, more than one possibility to go from one sequencer set to another can exist. Such a path is selected by SequencerPathSelector[SequencerSetSelector]. Each path and therefore the transition between different sequencer sets is based on a defined trigger and an aimed next sequencer set which is selectable by SequencerSetNext[SequencerSetSelector][SequencerPathSelector]. After the trigger occurs the settings of the next set are active. The trigger is defined by the features SequencerTriggerSource[SequencerSetSelector][SequencerPathSelector] and SequencerTriggerActivation[SequencerSetSelector][SequencerPathSelector]. The functions of these features are the same as TriggerSource and TriggerActivation. For a flexible sequencer implementation, the SequencerPathSelector[SequencerSetSelector] should be part of the sequencer sets.

(quoted from the GenICam SFNC 2.3)

The "Sequencer Control" contains features like

<table>
<thead>
<tr>
<th>Feature name (acc. to SFNC p. 242)</th>
<th>Property name (acc. to mvIMPA CT Acquire p. 240)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SequencerMode</td>
<td>sequencerMode</td>
<td>Controls if the sequencer mechanism is active. Possible values are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Off: Disables the sequencer.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- On: Enables the sequencer.</td>
</tr>
</tbody>
</table>
| **SequencerConfigurationMode** |sequencerConfigurationMode | Controls if the sequencer configuration mode is active. Possible values are:
- Off: Disables the sequencer configuration mode.
- On: Enables the sequencer configuration mode. |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SequencerFeatureSelector</strong></td>
<td>sequencerFeatureSelector</td>
<td>Selects which sequencer features to control. The feature lists all the features that can be part of a device sequencer set. All the device's sequencer sets have the same features. Note that the name used in the enumeration must match exactly the device's feature name.</td>
</tr>
<tr>
<td><strong>SequencerFeatureEnable</strong></td>
<td>sequencerFeatureEnable</td>
<td>Enables the selected feature and make it active in all the sequencer sets.</td>
</tr>
<tr>
<td><strong>SequencerSetSelector</strong></td>
<td>sequencerSetSelector</td>
<td>Selects the sequencer set to which further settings applies.</td>
</tr>
<tr>
<td><strong>SequencerSetSave</strong></td>
<td>sequencerSetSave</td>
<td>Saves the current device state to the selected sequencer set selected by <strong>SequencerSetSelector</strong>.</td>
</tr>
<tr>
<td><strong>SequencerSetLoad</strong></td>
<td>sequencerSetLoad</td>
<td>Loads the sequencer set selected by <strong>SequencerSetSelector</strong> in the device. Even if <strong>SequencerMode</strong> is <strong>Off</strong>, this will change the device state to the configuration of the selected set.</td>
</tr>
<tr>
<td><strong>SequencerSetActive</strong></td>
<td>sequencerSetActive</td>
<td>Contains the currently active sequencer set.</td>
</tr>
<tr>
<td><strong>SequencerSetStart</strong></td>
<td>sequencerSetStart</td>
<td>Sets the initial/start sequencer set, which is the first set used within a sequencer.</td>
</tr>
<tr>
<td><strong>SequencerPathSelector</strong></td>
<td>sequencerPathSelector</td>
<td>Selects to which branching path further path settings apply.</td>
</tr>
<tr>
<td><strong>SequencerSetNext</strong></td>
<td>sequencerSetNext</td>
<td>Selects the next sequencer set.</td>
</tr>
<tr>
<td>SequencerTriggerSource</td>
<td>sequencerTriggerSource</td>
<td></td>
</tr>
<tr>
<td>------------------------</td>
<td>------------------------</td>
<td></td>
</tr>
<tr>
<td>Specifies the internal signal or physical input line to use as the sequencer trigger source. Values supported by MATRIX VISION devices are:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Off: Disables the sequencer trigger.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• ExposureEnd: Starts with the reception of the ExposureEnd event.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Counter1End: Starts with the reception of the Counter1End event.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• UserOutput0: Specifies UserOutput0 bit signal to use as internal source for the trigger.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other possible values that might be supported by third party devices are:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• AcquisitionTrigger: Starts with the reception of the Acquisition Trigger.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• AcquisitionTriggerMissed: Starts with the reception of the missed Acquisition Trigger.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• AcquisitionStart: Starts with the reception of the Acquisition Start.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• AcquisitionEnd: Starts with the reception of the Acquisition End.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• FrameTrigger: Starts with the reception of the Frame Start Trigger.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• FrameTriggerMissed: Starts with the reception of the missed Frame Trigger.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• FrameStart: Starts with the reception of the Frame Start.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• FrameEnd: Starts with the reception of the Frame End.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• FrameBurstStart: Starts with the reception of the Frame Burst Start.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• FrameBurstEnd: Starts with the reception of the Frame Burst End.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• ExposureStart: Starts with the reception of the Exposure Start.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Line0 (If 0 based), Line1, Line2, ...: Starts when the specified TimerTrigger Activation condition is met on the chosen I/O Line.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• UserOutput1, UserOutput2, ...: Specifies which User Output bit signal to use as internal source for the trigger.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Counter0Start, Counter1Start, Counter2Start, ...: Starts with the reception of the Counter Start.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Counter0End, Counter2End, ...: Starts with the reception of the Counter End.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Timer0Start, Timer1Start, Timer2Start, ...: Starts with the reception of the Timer Start.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Timer0End, Timer1End, Timer2End, ...: Starts with the reception of the Timer End.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Encoder0, Encoder1, Encoder2, ...: Starts with the reception of the Encoder output signal.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• LogicBlock0, LogicBlock1, LogicBlock2, ...: Starts with the reception of the Logic Block output signal.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• SoftwareSignal0, SoftwareSignal1, SoftwareSignal2, ...: Starts on the reception of the Software Signal.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Action0, Action1, Action2, ...: Starts with the assertion of the chosen action signal.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• LinkTrigger0, LinkTrigger1, LinkTrigger2, ...: Starts with the reception of the chosen Link Trigger.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| • CC1, CC2, CC3, CC4: Index of the Camera Link physical line and associated I/O control block to use. This ensures a direct mapping between MATRIX VISION GmbH
The sequencer mode can be used to set a series of feature sets for image acquisition. The sets can consecutively be activated during the acquisition by the camera. The sequence is configured by a list of parameters sets.

**Note**

At the moment, the Sequencer Mode is only available for MATRIX VISION cameras with CCD sensors and Sony’s CMOS sensors. Please consult the "Device Feature and Property List’s" to get a summary of the actually supported features of each sensor.

The following features are currently available for using them inside the sequencer control:

- BinningHorizontal
- BinningVertical
- CounterDuration (can be used to configure a certain set of sequencer parameters to be applied for the next CounterDuration frames)
- DecimationHorizontal
- DecimationVertical
- ExposureTime
• Gain
• Height
• OffsetX
• OffsetY
• Width
• Multiple conditional sequencer paths.

**Note**

Configured sequencer programs are stored as part of the **User Sets** like any other feature.

Actual settings of the camera are overwritten when a sequencer set is loaded.

See also

- Define multiple exposure times using the Sequencer Control (p. 251)
- There are 3 C++ examples called **GenICamSequencerUsage**, **GenICamSequencerUsageWithPath**s and **GenICamSequencerParameterChangeAtRuntime** that show how to control the sequencer from an application. They can be found in the **Examples** section of the **C++ interface documentation** (p. 199)

<table>
<thead>
<tr>
<th>Feature name (acc. to SFNC (p. 242))</th>
<th>Property name (acc. to mvIMPA⇒CT Acquire (p. 240))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>mv Defective Pixel Count</strong></td>
<td>Contains the number of valid defective pixels.</td>
<td></td>
</tr>
<tr>
<td><strong>mv Defective Pixel Selector</strong></td>
<td>Controls the index of the defective pixel to access.</td>
<td></td>
</tr>
<tr>
<td><strong>mvDefectivePixelDataLoad</strong></td>
<td>Loads the defective pixels from the device non volatile memory.</td>
<td></td>
</tr>
<tr>
<td><strong>mvDefectivePixelDataSave</strong></td>
<td>Saves the defective pixels to the device non volatile memory.</td>
<td></td>
</tr>
</tbody>
</table>

related to control the devices Defective Pixel data.

See also

Correcting image errors of a sensor (p. 290)

14.19 Transport Layer Control

The **“Transport Layer Control”** contains features like
related to the Transport Layer Control.

In this section MATRIX VISION provides a bandwidth control feature (p. 315). You have to select `mvGevSCB<\rightarrow\>WControl` in the `GevStreamChannelSelector`. A new property with the same name will appear. Here you can set the maximum bandwidth in KBps.

Use cases related to the "Transport Layer Control" are:

- Using the primary application switchover functionality (p. 419)
- Disabling the heartbeat (p. 389)

14.20 User Set Control

The "User Set Control" contains features like

<table>
<thead>
<tr>
<th>Feature name (acc. to SFNC (p. 242))</th>
<th>Property name (acc. to mvIMPA&lt;\rightarrow&gt;CT Acquire (p. 240))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserSetSelector</td>
<td>userSetSelector</td>
<td>Selects the feature user set to load, save or configure.</td>
</tr>
<tr>
<td>UserSetLoad[UserSetSelector]</td>
<td>userSetLoad</td>
<td>Loads the user set specified by <code>UserSetSelector</code> to the device and makes it active.</td>
</tr>
<tr>
<td>UserSetSave[UserSetSelector]</td>
<td>userSetSave</td>
<td>Endianess of the device registers.</td>
</tr>
<tr>
<td>UserSetDefault</td>
<td>userSetDefault</td>
<td>Selects the feature user set to load and make active when the device is reset.</td>
</tr>
</tbody>
</table>

related to the User Set Control to save and load the user device settings.

The camera allows the storage of up to five configuration sets in the camera. This feature is similar to the storing settings in the registry but this way in the camera. It is possible to store

- exposure,
- gain,
- AOI,
• frame rate,
• LUT (p. 182),
• one Flat-Field Correction (p. 294),
• etc.

permanently. You can select, which user set comes up after hard reset.

Use case related to the "User Set Control" is:

• Creating user set entries (p. 381)

Another way to create user data is described here: Creating user data entries (p. 379)

14.21 mv Frame Average Control (only with specific models)

The "mv Frame Average Control" contains features like

<table>
<thead>
<tr>
<th>Feature name (acc. to SFNC (p. 242))</th>
<th>Property name (acc. to mvIMPA→CT Acquire (p. 240))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mvFrameAverageEnable</td>
<td>mvFrameAverageEnable</td>
<td>Enables the frame averaging engine.</td>
</tr>
<tr>
<td>mvFrameAverageSlope</td>
<td>mvFrameAverageSlope</td>
<td>The slope in full range of register.</td>
</tr>
</tbody>
</table>

related to the frame averaging engine.

The use case Reducing noise by frame averaging (p. 313) shows in detail how this feature works.

14.22 mv High Dynamic Range Control (only with specific sensor models)

The "mv High Dynamic Range Control" contains features like

<table>
<thead>
<tr>
<th>Feature name (acc. to SFNC (p. 242))</th>
<th>Property name (acc. to mvIMPA→CT Acquire (p. 240))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mvHDREnable</td>
<td>mvHDREnable</td>
<td>Enables the high dynamic range feature.</td>
</tr>
<tr>
<td>mvHDRPreset</td>
<td>mvHDRPreset</td>
<td>Selects the HDR parameter set.</td>
</tr>
<tr>
<td>mvHDRSelector</td>
<td>mvHDRSelector</td>
<td>Selects the HDR parameter set to configure.</td>
</tr>
<tr>
<td>mvHDRVoltage1</td>
<td>mvHDRVoltage1</td>
<td>First HDR voltage in mV.</td>
</tr>
<tr>
<td>mvHDRVoltage2</td>
<td>mvHDRVoltage2</td>
<td>Second HDR voltage in mV.</td>
</tr>
<tr>
<td>mvHDRExposure1</td>
<td>mvHDRExposure1</td>
<td>First HDR exposure in ppm.</td>
</tr>
<tr>
<td>mvHDRExposure2</td>
<td>mvHDRExposure2</td>
<td>Second HDR exposure in ppm.</td>
</tr>
</tbody>
</table>

related to the control of the device High Dynamic Range parameters.

The use cases Adjusting sensor -x00w (p. 363) and Adjusting sensor -x02d (-1012d) (p. 366) show the principle of the HDR.
14.23 Action Control

The "Action Control" contains features like

<table>
<thead>
<tr>
<th>Feature name (acc. to SFNC (p. 242))</th>
<th>Property name (acc. to mvMPA-CT Acquire (p. 240))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActionDeviceKey</td>
<td>actionDeviceKey</td>
<td>Provides the device key that allows the device to check the validity of action commands. The device internal assertion of an action signal is only authorized if the ActionDeviceKey and the action device key value in the protocol message are equal.</td>
</tr>
<tr>
<td>ActionSelector</td>
<td>actionSelector</td>
<td>Selects to which action signal further action settings apply.</td>
</tr>
<tr>
<td>ActionGroupKey</td>
<td>actionGroupKey</td>
<td>Provides the key that the device will use to validate the action on reception of the action protocol message.</td>
</tr>
<tr>
<td>ActionGroupMask</td>
<td>actionGroupMask</td>
<td>Provides the mask that the device will use to validate the action on reception of the action protocol message.</td>
</tr>
</tbody>
</table>

related to the Action control features.

The use case Using Action Commands (p. 407) shows in detail how this feature works.
15 C developers

The description for the mvIMPACT Acquire SDK for C developers is available as a separate file: mvIMPACT_Acquire_API_C_manual.chm which is

- either part of the installed package or

Here an online version of the documentation is available as well.
16 C++ developers

The description for the mvIMPACT Acquire SDK for C++ developers is available as a separate file: `mvIMPACT_Acquire_API_CPP_manual.chm` which is

- either part of the installed package or

Here an online version of the documentation is available as well.
17 .NET developers

The description for the mvIMPACT Acquire SDK for .NET developers is available as a separate file: `mvIMPACT\Acquire_API_NET_manual.chm` which is

- either part of the installed package or

Here an online version of the documentation is available as well.
18 Python developers

Note

There is no separate manual available for the Python API right now. For documentation please refer to the C++ manual instead. Almost everything stated there will be valid for Python as well!

18.1 Introduction

As supporting all the various distributions and versions of Python out there with a binary interface is almost impossible the mvIMPACT acquire Python API needs to be compiled for a specific version of Python it shall be used with. Because of that what is shipped at the moment is source code that has been generated using SWIG which before using it requires a compilation operation.

This requires a compiler matching the version of Python that shall be used. E.g. to use mvIMPACT Acquire with Python 2.7 on Windows systems requires Visual Studio 2008. Additional information about the compiler that works for a particular Python version on a particular platform can be found online. See e.g.

- https://docs.python.org/2/extending/building.html
- https://wiki.python.org/moin/WindowsCompilers

Note

Microsoft provides free-of-charge a ‘Microsoft Visual C++ Compiler for Python 2.7’ which can be used to compile ‘mvIMPACT Acquire for python’ for Python versions 2.7 to 3.2:  
The use of this compiler packet has been tested by MATRIX VISION and it is is highly recommended.

When installing the mvIMPACT Acquire Python API on a target system all files needed for building the actual extension module can be found in /mvIMPACT_Python.

18.2 Building

During the compilation process Pythons distutils package will be used

18.2.1 Windows

On Windows systems running /mvIMPACT_Python/compileWrapperCode.bat will build and install the Python API in the site-packages sub-folder of your Python installation provided a matching compiler could be found.

Note

The script will assume the Python interpreter can be found in the systems Path variable. If this is not the case you need to append the path to the directory containing Python.exe to this variable either permanently using the Systems environment variable dialog or temporary within the command shell you are calling the script from like this:

set Path=%Path%;C:\Python27  
compileWrapperCode.bat

The command shell your are calling the compilation script from depending on the version of Windows you are working with and the folder you have installed the mvIMPACT Acquire package to might require elevated rights thus you might need to start the command shell with the Run as administrator option.
18.2.2 Linux

On Linux systems running `/mvIMPACT_Python/setup.py` can be used to build and install the Python API in the `site-packages` sub-folder of your Python installation providing a matching compiler could be found. However one must be familiar with Python's `distutils` package.

**Note**

It is mandatory that the `python-dev` package is installed on the target Linux system, otherwise the binaries cannot be built!
The invoking user of the script must have the rights to install the generated binaries in the python directory of his system. If this is not the case, a recommended way to call the installation script is:

```
sudo -E python setup.py install
```

The building process may take literally a few minutes, so please be patient!

**Attention**

The SWIG generated wrapper code source file is very large! On some embedded systems or on 32-bit systems gcc might have trouble to digest this file as it is known to consume a lot of memory per source line in a given translation unit. When you encounter such a situation please get in touch with us!

18.3 Using

The actual API is almost the same as in C++ thus for now the C++ manual can be used as a reference and function description. There are just some minor differences between the C++ and the Python API which shall be explained here briefly:

- Stuff that has been declared deprecated at the time of publishing the Python API will not be available
- Simple getter functions may be wrapped as Python properties to have a more Python-like interface. So e.g. the function `Component::isValid()` will be a property in Python
- Code that in C++ resides in sub-namespaces like e.g. `mvIMPACT::acquire::GenICam` will all end up in `acquire` in Python (this is likely to change in future versions!)
- Some functions that use Python style built-in names like `mvIMPACT::acquire::Component::type()` will use a slightly different name in Python like `getType` in order to avoid confusion

Apart from that if someone is familiar with the C++ interface it shouldn't be too difficult to use the API. This is how an acquisition from a user selectable device can be done:

```python
from __future__ import print_function
import os
import platform
import string
import sys
# import all the stuff from mvIMPACT Acquire into the current scope
from mvIMPACT import acquire
# import all the mvIMPACT Acquire related helper function such as ‘conditionalSetProperty’ into the current scope
from mvIMPACT import acquire
# import all the mviMPACT Acquire related helper function such as ‘conditionalSetProperty’ into the current scope
# If you want to use this module in your code feel free to do so but make sure the ‘Common’ folder resides in a sub-folder of your project
from Common import *

# For systems with NO mvDisplay library support
# import ctypes
# import Image
```
#import numpy

devMgr = acquire.DeviceManager()
for i in range(devMgr.deviceCount()):
    pDev = devMgr.getDevice(i)
    print("[\{}\]: " + pDev.serial.read() + "(" + pDev.product.read() + ", " + pDev.family.read(), end='')
    if pDev.interfaceLayout.isValid:
        conditionalSetProperty(pDev.interfaceLayout, acquire.dilGenICam)
        print("interface layout: "+pDev.interfaceLayout.readS(), end='')
    if pDev.acquisitionStartStopBehaviour.isValid:
        conditionalSetProperty(pDev.acquisitionStartStopBehaviour, acquire.assbUser)
        print("acquisition start/stop behaviour: "+pDev.acquisitionStartStopBehaviour.readS(), end='')
    if pDev.isInUse():
        print("!!!ALREADY IN USE!!!", end='')
    print(')

print("Please enter the number in front of the listed device followed by [ENTER] to open it: ", end='')
devNr = int(raw_input())
if (devNr < 0) or (devNr >= devMgr.deviceCount()):
    print("Invalid selection!")
sys.exit(-1)

pDev = devMgr.getDevice(devNr)
pDev.open()

print("Please enter the number of buffers to capture followed by [ENTER]: ", end='')
framesToCapture = int(raw_input())
if framesToCapture < 1:
    print("Invalid input! Please capture at least one image")
sys.exit(-1)

# The mvDisplay library is only available on Windows systems for now
isDisplayModuleAvailable = platform.system() == "Windows"
if isDisplayModuleAvailable:
    display = acquire.ImageDisplayWindow("A window created from Python")
else:
    print("The mvIMPACT Acquire display library is not available on this('" + platform.system() + ") system.

# For systems with NO mvDisplay library support
channelType = numpy.uint16 if channelBitDepth > 8 else numpy.uint8

fi = acquire.FunctionInterface(pDev)
statistics = acquire.Statistics(pDev)

while fi.imageRequestSingle() == acquire.DMR_NO_ERROR:
    print("Buffer queued")
pPreviousRequest = None

manuallyStartAcquisitionIfNeeded(pDev, fi)
for i in range(framesToCapture):
    requestNr = fi.imageRequestWaitFor(-1)
    if fi.isRequestNrValid(requestNr):
        pRequest = fi.getRequest(requestNr)
        if pRequest.isOK():
            if i%100 == 0:
                print("Info from " + pDev.serial.read() + ": " + statistics.framesPerSecond.name() + ": " + statistics.framesPerSecond.readS() + ": " + statistics.errorCount.name() + ": " + statistics.errorCount.readS() + ": " + statistics.captureTime_s.name() + ": " + statistics.captureTime_s.readS())
            if isDisplayModuleAvailable:
                display.GetImageDisplay().SetImage(pRequest)
                display.GetImageDisplay().Update()
        # For systems with NO mvDisplay library support
        cbuf = (ctypes.c_char * imageSize).from_address(long(req.imageData.read()))
        arr = numpy.fromstring(cbuf, dtype = channelType)
        arr.shape = (height, width, channelCount)
        if channelCount == 1:
            img = Image.fromarray(arr)
        else:
            img = Image.fromarray(arr, 'RGBA' if alpha else 'RGB')
        if pPreviousRequest != None:
            pPreviousRequest.unlock()
pPreviousRequest = pRequest
fi.imageRequestSingle()
else:
    print("imageRequestWaitFor failed (" + str(requestNr) + ", " + ImpactAcquireException.getErrorCodeAsString(requestNr) + ")")
manuallyStopAcquisitionIfNeeded(pDev, fi)
raw_input("Press Enter to continue...")

Note

The above code uses the Python 3 style print. Because of the line
from __future__ import print_function
This will also work with Python versions starting with version 2.6. For smaller versions of Python the code needs to be changed!
19 DirectShow Interface

Note

DirectShow can only be used in combination with the Microsoft Windows operating system. Since Windows Vista, Movie Maker does not support capturing from a device registered for DirectShow anymore.

This is the documentation of the MATRIX VISION DirectShow_acquire interface. A MATRIX VISION specific property interface based on the IKsPropertySet has been added. All other features are related to standard DirectShow programming.

- Supported Interfaces (p. 205)
- Logging (p. 205)
- Registering and renaming devices for DirectShow usage (p. 206)

19.1 Supported Interfaces

19.1.1 IAMCameraControl

19.1.2 IAMDroppedFrames

19.1.3 IAMStreamConfig

19.1.4 IAMVideoProcAmp

19.1.5 IKsPropertySet

The DirectShow_acquire supports the IKsPropertySet Interface. For further information please refer to the Microsoft DirectX 9.0 Programmer's Reference.

Supported property set GUID's:

- AMPROPERTY_PIN_CATEGORY
- DIRECT_SHOW_ACQUIREPROPERTYSET

19.1.6 ISpecifyPropertyPages

19.2 Logging

The DirectShow_acquire logging procedure is equal to the logging of the MATRIX VISION products which uses mvIMPACT Acquire. The log output itself is based on XML.

If you want more information about the logging please have a look at the Logging chapter of the respective "mvIMPACT Acquire API" manual.
19.3 Registering and renaming devices for DirectShow usage

Note

Please be sure to register the MV device for DirectShow with the right version of `mvDeviceConfigure` (p. 156) . I.e. if you have installed the 32 bit version of the VLC Media Player, Virtual Dub, etc., you have to register the MV device with the 32 bit version of `mvDeviceConfigure` (p. 156) ("C:\Program Files\MATRIX VISION\mvI-\MPACT Acquire\bin") !

19.3.1 Registering devices

To register a device/devices for access under DirectShow please perform the following registration procedure:

1. Start `mvDeviceConfigure`.
   
   If no device has been registered the application will more or less (depending on the installed devices) look like this.

   ![mvDeviceConfigure start window](image)

   **Figure 1: mvDeviceConfigure - start window**

2. To register every installed device for DirectShow access click on the menu item "DirectShow" → "Register all devices".
19.3 Registering and renaming devices for DirectShow usage

3. After a successful registration the column "registered for DirectShow" will display 'yes' for every device and the devices will be registered with a default DirectShow friendly name.
19.3.2 Renaming devices

If you want to modify the friendly name of a device under DirectShow, please perform the following procedure:

1. If mvDeviceConfigure is already not running, please start it.
2. Now, select the device you want to rename, click the right mouse button and select "Set DirectShow friendly name":

3. Then, a dialog will appear. Please enter the new name and confirm it with "OK".
19.3 Registering and renaming devices for DirectShow usage

Figure 5: mvDeviceConfigure - enter new name

4. Afterwards the column "DirectShow friendly name" will display the newly assigned friendly name.

Figure 6: mvDeviceConfigure - renamed device

Note

Please do not select the same friendly name for two different devices. In theory this is possible, however the mvDeviceConfigure GUI will not allow this to avoid confusion.

19.3.3 Make silent registration

To make a silent registration without dialogs, the Windows tool "regsvr32" via command line can be used.

The following command line options are available and can be passed during the silent registration:

EXAMPLES:

Register ALL devices that are recognized by mvIMPACT Acquire (this will only register devices which have drivers installed).

regsvr32 <path>\DirectShow_acquire.ax /s
20 Troubleshooting

20.1 I can see the camera in mvIPConfigure, but I cannot access the camera

In some cases, it can be possible that you can see the camera in mvIPConfigure (p.152), however,

- there are error messages in mvIPConfigure (p.152) like "Cannot Access Device",
- you cannot Use the camera in wxPropView (p.115), or
- the camera is even not listed in wxPropView (p.115).

There could be two reasons for this problem:

1. The communication is blocked by a firewall.
2. There are network configuration problems, which means that either the network is incorrectly configured or
   with regards to the network settings the camera itself.

To solve this issue, please check the firewall settings of your system. In the latter case, please have a look at the
following chapters of the manual:

- mvIPConfigure (p.152)
- Optimizing the network configuration (p.47) (Windows)
- Optimizing the network configuration (p.57) (Linux)
- Additional network configuration for mvBlueCOUGAR-XD (p.51) (Windows)
- Additional network configuration for mvBlueCOUGAR-XD (p.58) (Linux)

20.2 Image Data is lost (incomplete frames)

Incomplete frames indicate network problems or suboptimal network settings. Please have a look at chapters
Optimizing the network configuration (p.47) (Windows) or Optimizing the network configuration (p.57) (Linux)
for more information.

20.3 There are image error counts

Modern PC’s, notebook’s, etc. try to save energy by using a smart power management. For example, if processor
load is not needed the processor changes to a power saving (sleep) state automatically and vice versa. Every state
change will stop the processor for microseconds. This time is enough to cause image error counts!

With mvDeviceConfigure you can disable the CPU sleep states on Windows (p.163) systems.

A description for Linux users is in the pipeline.
20.4 I get an exception when setting offsetX or offsetY

offsetX and offsetY are horizontal or vertical offsets from the origin to the region of interest (in pixels). The max. value of both offsetX and offsetY changes in reference to width or height. For this reason, if you want to change both values, offsetX / offsetY and width / height, you first have to change width / height and then offsetX / offsetY.

20.5 Cognex VisionPro

20.5.1 I cannot use mvBlueCOUGAR-XD

By default, VisionPro 7.2 by Cognex does not support Dual-GigE cameras like mvBlueCOUGAR-XD. However, there is a workaround to include Dual-GigE cameras.

**Note**

Since VisionPro 8.0, you have to start the program as administrator. Afterwards, the products using the MATRIX VISION driver ("Device: " are displayed.

Instead of choosing "GigE Vision: MATRIX VISION..." from the image acquisition devices dropdown list, select the item "Device: mvBlueCOUGAR-XD ...". Afterwards, the MATRIX VISION driver will be used:

![VisionPro - Choosing the Image Acquisition Device](image.png)

Figure 1: VisionPro - Choosing the Image Acquisition Device
20.5.2 I cannot initialize 4 Tap sensor mvBlueCOUGAR-XDs

VisionPro 7.2 by Cognex does not support 4 Tap sensors, like the CCD sensors by Sony available for the mvBlueCOUGAR-XD.

**Note**

Since VisionPro 8.0, you have to start the program as administrator. Afterwards, the products using the MATRIX VISION driver ("Device: " ) are displayed.

If you use the workaround to use Dual-GigE cameras (p.211) (1) and click on "Initialize Image Acquisition" (2), ...

![Figure 2: VisionPro - Initialize Image Acquisition](image)

... you can get following error message:

![Figure 3: VisionPro - Error message](image)
In this case, the 4 Tap mode of the mvBlueCOUGAR-XD is activated, which cannot be changed by VisionPro directly:

1. Close VisionPro,

2. open wxPropView (p.115) and

3. set SensorTaps to "Two" in "Setting -> Base -> Camera -> GenICam -> ImageFormatControl". The frame rate will be halved.

4. In order that these settings are using during the start, you can save them in "Setting -> Base -> Camera -> GenICam -> UserSetControl". Choose a UserSet, e.g. "UserSet1" (1).

5. save the UserSet by clicking on "int UserSetSave()" (2), and

6. select the UserSet in "UserSetDefault" (3).

Figure 4: wxPropView - Setting SensorTaps
Afterwards it is possible to initialize the mvBlueCOUGAR-XD by clicking on "Initialize Image Acquisition":

---

**Figure 5: wxPropView - Setting UserSet**

7. Afterwards it is possible to initialize the mvBlueCOUGAR-XD by clicking on "Initialize Image Acquisition":

---
8. By clicking on the camera icon (top left), you should see a live image. Perhaps you have to adapt the acquisition parameters:
20.6 I get an oscillating frame rate

If your camera supports **FrameRateExactness** it is possible that you may receive an oscillating frame rate. This is due to the fact, that frames of different lengths are used to achieve an overall stable and exact frame rate. We use 10 frames to achieve an more exact frame rate, i.e. if you capture 10 frames, your frame rate will be exact and stable.

Example:

1. Assume we want to achieve a frame rate of 150 fps.
2. This means we need to wait 6666,6667 us until the next frame start.
3. The possible step size is 1 us.
4. If we set 6666 us, the frame rate will be stable, but not exact enough.
5. $\Rightarrow$ 6 frames of length 6666 and 4 frames of length 6667 make the frame rate more exact.

20.7 I cannot focus on distant objects

In most vision applications it is necessary to focus on close objects. For this reason we adjust the lensholder to meet the lens' MOD (= minimum object distance).

If you need to focus on distant objects, you can adjust the C-mount lensholder as described in **Adjusting the C-mount** (p. 26).

20.8 Accessing log files

If you need support using our products, you can shorten response times by sending us your log files. Accessing the log files is different in Windows and Linux:

20.8.1 Windows

Since **mvIMPACT Acquire 2.11.9**

Since mvIMPACT Acquire driver version 2.11.9 you can access the log files in Windows using **wxPropView** (p. 115). The way to do this is described in **Accessing log files** (p. 142).
20.8 Accessing log files

20.8.2 Linux

Since mvIMPACT Acquire 2.24.0

You can access the log files in Linux via /opt/mvIMPACT_Acquire/data/logs.

You can also extract the directory using the following command

```bash
env | grep MVIMPACT_ACQUIRE_DATA_DIR
```

or change the directory directly via

```bash
cd $MVIMPACT_ACQUIRE_DATA_DIR/logs
```

**For older versions:**

Like in Windows, log files will be generated, if the activation flag for loggings called `mvDebugFlags.mvd` is available in the same folder as the application (however, using Windows log files will be generated automatically, because the applications are started from the same folder). By default, in Linux the `mvDebugFlags.mvd` will be installed in the installation's destination folder in the sub-folder "apps". For example, if the destination folder was "/home/workspace", you can locate the `mvDebugFlags.mvd` like the following way:

```bash
user@linux-desktop:~$ // <- Starting the console window, you will be in the home directory: /home/
user@linux-desktop:~$ cd workspace/apps/ // <- Change the directory
user@linux-desktop:/home/workspace/apps$ ls -l // <- list the directory
insgesamt 144
drwxr-xr-x 9 user user 4096 Mai 21 15:08 Callback
drwxr-xr-x 8 user user 4096 Mai 21 15:08 Callback_C
drwxr-xr-x 9 user user 4096 Mai 21 15:08 CaptureToUserMemory_C
drwxr-xr-x 3 user user 4096 Mai 21 15:03 Common
drwxr-xr-x 9 user user 4096 Mai 21 15:09 ContinuousCaptureAllDevices
drwxr-xr-x 9 user user 4096 Mai 21 15:09 ContinuousCapture_C
drwxr-xr-x 11 user user 4096 Mai 21 15:09 DigitalIOs
drwxr-xr-x 9 user user 4096 Mai 21 15:09 FirmwareUpgrade
drwxr-xr-x 11 user user 4096 Mai 21 15:09 GenericInterfaceLayout
drwxr-xr-x 11 user user 4096 Mai 21 15:09 GenICamInterfaceLayout
drwxr-xr-x 11 user user 4096 Mai 21 15:09 LiveSnap
-rw-r--r-- 6 user user 4096 Mai 21 15:09 LiveSnapFLTK
-rw-r--r-- 1 user user 854 Mai 21 15:03 Makefile
-rw-r--r-- 1 user user 7365 Mai 21 15:03 Makefile.samp.inc
-rwxr-xr-x 1 user user 7166 Mai 21 15:03 mknewwappl.sh
-rw-r--r-- 1 user user 20713 Mai 21 15:03 mvDebugFlags.mvd // <- Log activation flag
drwxr-xr-x 7 user user 4096 Mai 21 15:09 mvDeviceConfigure
drwxr-xr-x 6 user user 4096 Mai 21 15:10 mvIPConfigure
drwxr-xr-x 6 user user 4096 Mai 21 15:11 mvPropView
drwxr-xr-x 9 user user 4096 Mai 21 15:11 SingleCapture
drwxr-xr-x 9 user user 4096 Mai 21 15:11 SingleCaptureStorage
```

In order that log files are generated, you have to execute your app from the folder, where `mvDebugFlags.mvd` is located. E.g. if you want to start `wxPropView`:

```bash
user@linux-desktop:/home/workspace/apps$ ./wxPropView // <- Start the executable from the folder, where `mvDebugFlags.mvd` is located.
```

Another possibility would be, to copy the `mvDebugFlags.mvd` file to the folder of the executable:

```bash
user@linux-desktop:/home/workspace/apps$ cp mvDebugFlags.mvd ./wxPropView // <- Copy the log activation flag
user@linux-desktop:/home/workspace/apps$ cd ./wxPropView/x86/
user@linux-desktop:/home/workspace/apps$ ./wxPropView // <- Start the executable
```

Afterwards, several log files are generated which are listed in `files.mvloglist`. The log files have the file extension `.mvlog`. Please send these files to our support team.
20.9 mvGigEConfigure freezes when trying to activate/deactivate the filter driver a Link Aggregation Group (LAG)

20.9.1 Symptoms

Activating or deactivating the filter driver on a Link Aggregation Group (LAG for short) using mvGigEConfigure (p.149), the NDIS filter driver will freeze and the process cannot be killed even with task manager. After a restart, mvGigEConfigure (p.149) will show the filter driver status correctly however. Also when connecting a debugger the situation as described here: https://social.msdn.microsoft.com/ Forums/windowsdesktop/en-US/0735c73e-6d2e-49ab-a947-e10fe8137ac3/ndis-driver-freezes-during-install-via-inetcfgclasssetupinstall?forum=wdk can be observed.

20.9.2 Cause

The exact cause for this issue is unknown but it is believed to be a bug in the network cards kernel module.

20.9.3 Resolution

So far, to avoid this behavior disconnect both network cables just before activating or deactivating the filter driver. Afterwards reconnect the network cables again.

20.10 mvBlueCOUGAR-X105G with frame rates < 5.75 fps

20.10.1 Cause

The frame rate can not be reduced below 5.75fps because of some technical limitation related to the free running mode of the sensor.

20.10.2 Resolution

As soon as the sensor is used in triggered mode, the frame rate can be below 5.75 fps.

To get the camera working at lower frame rates, it is necessary to configure two timers which are used to trigger the sensor.

E.g., the camera is triggered at 4 fps.

- **Timer1** represents the duration of the "low" signal. It is configured to 240,000 us. TimerTriggerSource is configured to "Timer2End".

- **Timer2** represents the "high" signal. It is configured to 10,000us. TimerTriggerSource of Timer2 is "Timer1End". The sum of both durations is 250,000 us which means the sequence runs with 4 cycles per second. (1s/250,000us=4)

Afterwards the "TriggerSource" has to be configured to "Timer2Start" and the "TriggerMode" has to be switched to "On".

Once you start the acquisition by clicking the "Acquire" button, the camera should run at 4 fps.

To get the camera working at different frame rates, **Timer1** needs to be configured accordingly. E.g. 500,000 us for 2 fps.
20.11 Why does updating the device list take so long

Since mvIMPACT Acquire 2.24.0

The GenTL driver scans both the GigE Vision and the USB3 Vision interfaces for compliant devices. Given that the GigE Vision standard allows detection timeouts of 1000 ms, it could take long, especially for USB3 Vision users which even do not have GigE Vision in their application, until the device update is finished.

For this reason, we upgraded the MATRIX VISION application tools (p. 115) to excluded interfaces or single devices to be scanned or detected. The easiest way to do so is as follows:

1. Open wxPropView (p. 115).
2. Open the device list drop down and
3. select "Missing A Device?".

![Figure 8: "Missing A Device? Click here..."](image)

Now, you will see the interfaces and the detected device.

4. Finally, you can deselect the interfaces or devices which you do not want to scan for updating the device list. The following figure shows that the GigE Vision interface should be skipped:
20.12 Why is there no GenICam device or interface found on my Linux system

The environment variables which are necessary to us the mvGenTL Acquire package are defined by two shell scripts which are set by profile.d. This causes the variables to be available within every login-shell run on the system. Once an application is started from a non-login shell which is using mvIMPACT Acquire libraries the variables from profile.d are not used. This leads to the situation that the necessary runtimes could not be found which leads to missing interfaces and devices.

To avoid this situation there are two possible solutions:

- Use a login-shell to run the application using the MATRIX VISION driver package.
- Insert the variables from: /etc/profile.d/genicam.sh and /etc/profile.d/acquire.sh to a shell configuration file which is read to configure environment variables.
<table>
<thead>
<tr>
<th>Numerical Value</th>
<th>String Representation</th>
<th>Brief Description</th>
<th>Detailed Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2000</td>
<td>PROPHANDLING_NOT←A_LIST</td>
<td>This component is not a list.</td>
<td>A list operation for this component has been called but this component does not reference a list. [-2000]</td>
</tr>
<tr>
<td>-2001</td>
<td>PROPHANDLING_NOT←APROPERTY</td>
<td>This component is not a property.</td>
<td>A property operation for this component has been called but this component does not reference a property. [-2001]</td>
</tr>
<tr>
<td>-2002</td>
<td>PROPHANDLING_NOT←AMETHOD</td>
<td>This component is not a method.</td>
<td>A method operation for this component has been called but this component does not reference a method. [-2002]</td>
</tr>
<tr>
<td>-2003</td>
<td>PROPHANDLING_NO_READ←READ_RIGHTS</td>
<td>The caller has no read rights for this component.</td>
<td>It has been tried to read data from this component, but the caller has no read rights for this component. [-2003]</td>
</tr>
<tr>
<td>-2004</td>
<td>PROPHANDLING_NO_WRITE←WRITE_RIGHTS</td>
<td>The caller has no write rights for this component.</td>
<td>It has been tried to modify data of this component, but the caller has no write rights for this component. [-2004]</td>
</tr>
<tr>
<td>-2005</td>
<td>PROPHANDLING_NO_MODIFY_SIZE←MODIFY_SIZE_RIGHTS</td>
<td>The caller can't modify the size of this component.</td>
<td>It has been tried to modify the size of this list or the number of values stored by a property, but the caller doesn't have the required right to do this. This error will also be reported if the user tried to increase the number of values handled by a property above the maximum number of values it can handle. Therefore before resizing a property check if the new size might exceed this maximum value by calling the appropriate function. [-2005]</td>
</tr>
<tr>
<td>-2006</td>
<td>PROPHANDLING_INCOMPATIBLE_COMPONENTS ←</td>
<td>The two involved components are not compatible.</td>
<td>An operation requiring two compatible components has been called with two components, which are not compatible. [-2006]</td>
</tr>
<tr>
<td>-2008</td>
<td>PROPHANDLING_UNSUPPORTED_PARAMETER ←</td>
<td>One or more of the specified parameters are not supported by the function.</td>
<td>This error might also be generated if a certain feature is not available on the current platform. [-2008]</td>
</tr>
<tr>
<td>Year</td>
<td>Error Code</td>
<td>Description</td>
<td>Details</td>
</tr>
<tr>
<td>------</td>
<td>------------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td>2009</td>
<td>PROPHANDLING_SIZE_MISMATCH</td>
<td>Different sized value buffers have been passed.</td>
<td>While trying to read value pairs the caller passed two different sized value buffers to a function while one is too small to hold all the information. [-2009]</td>
</tr>
<tr>
<td>2010</td>
<td>PROPHANDLING_IMPLEMENTATION_MISSING</td>
<td>A feature that is not implemented so far has been requested.</td>
<td>The caller requested a feature, that hasn't been implemented so far. This error code is only provided for compatibility and will be set in very rare cases only. [-2010]</td>
</tr>
<tr>
<td>2011</td>
<td>PROPHANDLING_ACS_TOKEN_CREATION_FAILED</td>
<td>An access token object couldn't be created.</td>
<td>This can either happen, because the caller has not the rights required to create an access token or because the system runs very low on memory. DeprecatedThis error code currently is not used anywhere within this framework. It might be removed in a future version. [-2011]</td>
</tr>
<tr>
<td>2012</td>
<td>PROPHANDLING_INVALID_ID_PROP_VALUE</td>
<td>It has been tried to assign an invalid value to a property.</td>
<td>This can either happen if the value lies above or below the min. or max. value for a property or when it has been tried to write a value to a property, which is not in the properties translation dictionary (if it defines one). To find out, which values are allowed for the property in question the user should check if the property defines a translation dictionary check the allowed values within a translation dictionary if one is defined check the min and max value for properties, that define limits [-2012]</td>
</tr>
<tr>
<td>2013</td>
<td>PROPHANDLING_PROP_TRANSLATION_TABLE_CORRUPTED</td>
<td>The properties translation table has been corrupted.</td>
<td>The properties translation table has been corrupted for an unknown reason and can't be used anymore. [-2013]</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td>Error Message</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>-2014</td>
<td>Invalid value index.</td>
<td>The caller tried to read a value from an invalid index from a property. Most properties store one value only, thus the only valid positive value index will be 0 (some negative index values are reserved for special values like e.g. the min/max value of a property). However some properties might store more than one value, thus the max. allowed index might be higher. The highest index allowed will always be the value count of a property minus one for properties with the cfFixedSize flag set. Other properties will automatically adjust the size once the user writes to an index out of bounds.</td>
<td></td>
</tr>
<tr>
<td>-2015</td>
<td>This property doesn't define a translation table.</td>
<td>The caller tried to modify a translation table, that hasn't been defined for this property.</td>
<td></td>
</tr>
<tr>
<td>-2016</td>
<td>An invalid value has been passed to the property.</td>
<td>Although properties are quite tolerant regarding the allowed assignment for them some value types can't be used to write all properties. As an example assigning a float value to an integer property would result in this error. Another reason for this error might be when a user tried to access e.g. a float property with functions meant to be used for int properties.</td>
<td></td>
</tr>
<tr>
<td>-2017</td>
<td>A too large value has been passed.</td>
<td>One or more of the values the caller tried to write to the property are larger than the max. allowed value for this property.</td>
<td></td>
</tr>
<tr>
<td>-2018</td>
<td>A too small value has been passed.</td>
<td>One or more of the values the caller tried to write to the property are smaller than the min. allowed value for this property.</td>
<td></td>
</tr>
<tr>
<td>-2019</td>
<td>The specified component could not be found.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2020</td>
<td>An invalid list has been referenced.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code</td>
<td>Definition</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>---------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>-2021</td>
<td>PROPHANDLING_COMPONENT_ID_INVALID</td>
<td>An invalid component within a list has been referenced.</td>
<td></td>
</tr>
<tr>
<td>-2022</td>
<td>PROPHANDLING_LIST_ENTRY_OCCUPIED</td>
<td>The specified list index is occupied.</td>
<td></td>
</tr>
<tr>
<td>-2023</td>
<td>PROPHANDLING_COMPONENT_HAS_OWNER_ALREADY</td>
<td>The specified component already has an owner.</td>
<td></td>
</tr>
<tr>
<td>-2024</td>
<td>PROPHANDLING_COMPONENT_ALREADY_REGISTERED</td>
<td>It has been tried to register the same component at twice in the same list.</td>
<td></td>
</tr>
<tr>
<td>-2025</td>
<td>PROPHANDLING_LIST_CANT_ACCESS_DATA</td>
<td>The desired data can't be accessed or found.</td>
<td></td>
</tr>
<tr>
<td>-2026</td>
<td>PROPHANDLING_METHOD_PTR_INVALID</td>
<td>The function pointer of the referenced method object is invalid.</td>
<td></td>
</tr>
<tr>
<td>-2027</td>
<td>PROPHANDLING_METHOD_INVALID_PARAM_LIST</td>
<td>A method object has an invalid parameter list.</td>
<td></td>
</tr>
<tr>
<td>-2028</td>
<td>PROPHANDLING_SWIG_ERROR</td>
<td>This indicates an internal error occurred within the SWIG generated wrapper code, when working under Python.</td>
<td></td>
</tr>
<tr>
<td>-2029</td>
<td>PROPHANDLING_INVALID_ID_INPUT_PARAMETER</td>
<td>A invalid input parameter has been passed to a function of this module.</td>
<td></td>
</tr>
<tr>
<td>-2030</td>
<td>PROPHANDLING_COMPONENT_NO_CALLBACK_REGISTERED</td>
<td>The user tried to modify a registered callback, but no callback has been registered for this component.</td>
<td></td>
</tr>
<tr>
<td>-2031</td>
<td>PROPHANDLING_INPUT_BUFFER_TOO_SMALL</td>
<td>The user tried to read data into a user supplied storage location, but the buffer was too small to accommodate the result.</td>
<td></td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td>Error Message</td>
<td>Example</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>-2032</td>
<td>PROPHANDLING_WRO--NG_PARAM_COUNT</td>
<td>The number of parameters is incorrect.</td>
<td>This error might occur if the user called a function with a variable number of input or output parameters and the number of parameters passed to the function does not match the number of required parameters. [-2032]</td>
</tr>
<tr>
<td>-2033</td>
<td>PROPHANDLING_UNSU--PPORTED_OPERATION</td>
<td>The user tried to execute an operation, which is not supported by the component he is referring to.</td>
<td>[-2033]</td>
</tr>
<tr>
<td>-2034</td>
<td>PROPHANDLING_CANT--_SERIALIZE_DATA</td>
<td>The user tried to save(serialize) a property list without having the right to do this.</td>
<td>[-2034]</td>
</tr>
<tr>
<td>-2035</td>
<td>PROPHANDLING_INVAL--ID_FILE_CONTENT</td>
<td>The user tried to use a file to update or create a component list, that does not contain valid data for this operation.</td>
<td>This e.g. might happen, if the file does not contain valid XML data or XML data that is not well formed. [-2035]</td>
</tr>
<tr>
<td>-2036</td>
<td>PROPHANDLING_CANT--_ALLOCATE_LIST</td>
<td>This error will occur when the modules internal representation of the tree structure does not allow the allocation of a new list.</td>
<td>In this case either new list can't be allocated. The only way to solve this problem is to delete another list. [-2036]</td>
</tr>
<tr>
<td>-2037</td>
<td>PROPHANDLING_CANT--_REGISTER_COMPONE--NT</td>
<td>The referenced list has no space left to register this component at the desired position.</td>
<td>There might however be an empty space within the list where this element could be registered, but no more components can be registered at the end of this list. [-2037]</td>
</tr>
<tr>
<td>-2038</td>
<td>PROPHANDLING_PROP--_VALIDATION_FAILED</td>
<td>The user tried to assign a value to a property, that is invalid.</td>
<td>This will result in a detailed error message in the log-file. This error might arise e.g. when a string property doesn't allow the string to contain numbers. In this case trying to set the properties value to 'blabla7bla' would cause this error. [-2038]</td>
</tr>
<tr>
<td>-2099</td>
<td>PROPHANDLING_LAST--_VALID_ERROR_CODE</td>
<td>Defines the last valid error code value for the property module.</td>
<td>[-2099]</td>
</tr>
<tr>
<td>-2100</td>
<td>DMR_DEV_NOT_FOUND</td>
<td>The specified device can't be found.</td>
<td>This error occurs either if an invalid device ID has been passed to the device manager or if the caller tried to close a device which currently isn't initialized. [-2100]</td>
</tr>
<tr>
<td>-2101</td>
<td>DMR_INIT_FAILED</td>
<td>The device manager couldn't be initialized.</td>
<td>This is an internal error. [-2101]</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td>Error Message</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>---------------------------------------</td>
<td>--------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>-2102</td>
<td>DMR_DRV_ALREADY_IN_USE</td>
<td>The device is already in use. This error e.g. will occur if this or another process has initialized this device already and an application tries to open the device once more or if a certain resource is available only once but shall be used twice. [-2102]</td>
<td></td>
</tr>
<tr>
<td>-2103</td>
<td>DMR_DEV_CANNOT_OPEN</td>
<td>The specified device couldn't be initialized. [-2103]</td>
<td></td>
</tr>
<tr>
<td>-2104</td>
<td>DMR_NOT_INITIALIZED</td>
<td>The device manager or another module hasn't been initialized properly. This error occurs if the user tries e.g. to close the de-vice manager without hav-ing initialized it before or if a library used internally or a module or device associated with that library has not been initialized properly or if [-2104]</td>
<td></td>
</tr>
<tr>
<td>-2105</td>
<td>DMR_DRV_CANNOT_OPEN</td>
<td>A device could not be initialized. In this case the log-file will contain detailed information about the source of the prob-lem. [-2105]</td>
<td></td>
</tr>
<tr>
<td>-2106</td>
<td>DMR_DEV_REQUEST_QUEUE_EMPTY</td>
<td>The devices request queue is empty. This error e.g. occurs if the user waits for an image re-quest to become available at a result queue without having send an image re-quest to the device before. It might also arise when try-ing to trigger an image with a software trigger mecha-nism before the acquisition engine has been completely started. In this case a small delay and then again calling the software trigger function will succeed. [-2106]</td>
<td></td>
</tr>
<tr>
<td>-2107</td>
<td>DMR_DEV_REQUEST_CREATION_FAILED</td>
<td>A request object couldn't be created. The creation of a request ob-ject failed. This might e.g. happen, if the system runs extremely low on memory. [-2107]</td>
<td></td>
</tr>
<tr>
<td>-2108</td>
<td>DMR_INVALID_PARAMETER</td>
<td>An invalid parameter has been passed to a function. This might e.g. happen if a function requiring a pointer to a structure has been passed an unassigned pointer or if a value has been passed, that is either too large or too small in that con-text. [-2108]</td>
<td></td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td>Reason</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>----------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>-2109</td>
<td>DMR_EXPORTED_SYM→BOL_NOT_FOUND</td>
<td>One or more symbols needed in a detected driver library couldn’t be resolved. In most cases this is an error handled internally. So the user will not receive this error code as a result of a call to an API function. However when the user tries to get access to an IMPACT buffer type while the needed IMPACT Base libraries are not installed on the target system this error code also might be returned to the user.</td>
<td>[-2109]</td>
</tr>
<tr>
<td>-2110</td>
<td>DEV_UNKNOWN_ERROR</td>
<td>An unknown error occurred while processing a user called driver function.</td>
<td></td>
</tr>
<tr>
<td>-2111</td>
<td>DEV_HANDLE_INVALID</td>
<td>A driver function has been called with an invalid device handle.</td>
<td></td>
</tr>
<tr>
<td>-2112</td>
<td>DEV_INPUT_PARAM_INVALID←VALID</td>
<td>A driver function has been called but one or more of the input parameters are invalid. There are several possible reasons for this error: an unassigned pointer has been passed to a function, that requires a valid pointer one or more of the passed parameters are of an incorrect type one or more parameters contain an invalid value (e.g. a filename that points to a file that can’t be found, a value, that is larger or smaller than the allowed values.</td>
<td>[-2112]</td>
</tr>
<tr>
<td>-2113</td>
<td>DEV_WRONG_INPUT_PARAM←ARAM_COUNT</td>
<td>A function has been called with an invalid number of input parameters.</td>
<td></td>
</tr>
<tr>
<td>-2114</td>
<td>DEV_CREATE_SETTING←_FAILED</td>
<td>The creation of a setting failed. This can either happen, when a setting with the same name as the one the user tried to create already exists or if the system can’t allocate memory for the new setting.</td>
<td>[-2114]</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td>Details</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td>-2115</td>
<td>DEV_REQUEST_CANT_BE_UNLOCKED</td>
<td>The unlock for a Request object failed. This might happen, if the Request is not locked at the time of calling the unlock function. It either has been unlocked by the user already or this request has never been locked as the request so far has not been used to capture image data into its buffer. Another reason for this error might be that the user tries to unlock a request that is currently processed by the device driver.</td>
<td></td>
</tr>
<tr>
<td>-2116</td>
<td>DEV_INVALID_REQUEST_NUMBER</td>
<td>The number for the Request object is invalid. The max. number for a Request object is the value of the property RequestCount in the SystemSettings list - 1.</td>
<td></td>
</tr>
<tr>
<td>-2117</td>
<td>DEV_LOCKED_REQUEST_IN_QUEUE</td>
<td>A Request that hasn't been unlocked has been passed back to the driver. This error might occur if the user requested an image from the driver but hasn't unlocked the Request that will be used for this new image.</td>
<td></td>
</tr>
<tr>
<td>-2118</td>
<td>DEV_NO_FREE_REQUEST_AVAILABLE</td>
<td>The user requested a new image, but no free Request object is available to process this request.</td>
<td></td>
</tr>
<tr>
<td>-2119</td>
<td>DEV_WAIT_FOR_REQUEST_FAILED</td>
<td>The wait for a request failed. This might have several reasons: the user waited for an image, but no image has been requested before. the user waited for a requested image, but the image is still not ready(e.g. because of a short timeout and a long exposure time). a triggered image has been requested but no trigger signal has been detected within the wait period. a plug and play device(e.g. an USB device) has been unplugged and therefore can't deliver images anymore. In this case the 'state' property should be checked to find out if the device is still present or not.</td>
<td></td>
</tr>
<tr>
<td>-2120</td>
<td>DEV_UNSUPPORTED_PARAMETER</td>
<td>The user tried to get/set a parameter, which is not supported by this device.</td>
<td></td>
</tr>
<tr>
<td>Code</td>
<td>Error Description</td>
<td>Details</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------------------------------------------------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td>-2121</td>
<td>DEV_INVALID_RTC_NMBR</td>
<td>The requested real time controller is not available for this device.</td>
<td></td>
</tr>
<tr>
<td>-2122</td>
<td>DMR_INTERNAL_ERROR</td>
<td>Some kind of internal error occurred. More information can be found in the +.log-file or the debug output.</td>
<td></td>
</tr>
<tr>
<td>-2123</td>
<td>DMR_INPUT_BUFFER_TOO_SMALL</td>
<td>The user allocated input buffer is too small to accommodate the result.</td>
<td></td>
</tr>
<tr>
<td>-2124</td>
<td>DEV_INTERNAL_ERROR</td>
<td>Some kind of internal error occurred in the device driver. More information can be found in the +.log-file or the debug output.</td>
<td></td>
</tr>
<tr>
<td>-2125</td>
<td>DMR_LIBRARY_NOT_FOUND</td>
<td>One or more needed libraries are not installed on the system.</td>
<td></td>
</tr>
<tr>
<td>-2126</td>
<td>DMR_FUNCTION_NOT_IMPLEMENTED</td>
<td>A called function or accessed feature is not available for this device.</td>
<td></td>
</tr>
<tr>
<td>-2127</td>
<td>DMR_FEATURE_NOT_AVAIBLE</td>
<td>The feature in question is (currently) not available for this device or driver. This might be because another feature currently blocks the one in question from being accessible. More information can be found in the +.log-file or the debug output.</td>
<td></td>
</tr>
<tr>
<td>-2128</td>
<td>DMR_EXECUTION_PROHIBITED</td>
<td>The user is not permitted to perform the requested operation. This e.g. might happen if the user tried to delete user data without specifying the required password.</td>
<td></td>
</tr>
<tr>
<td>-2129</td>
<td>DMR_FILE_NOT_FOUND</td>
<td>The specified file can't be found. This might e.g. happen if the current working directory doesn't contain the file specified.</td>
<td></td>
</tr>
<tr>
<td>-2130</td>
<td>DMR_INVALID_LICENCE</td>
<td>The licence doesn't match the device it has been assigned to. When e.g. upgrading a device feature each licence file is bound to a certain device. If the device this file has been assigned to has a different serial number than the one used to create the licence this error will occur.</td>
<td></td>
</tr>
<tr>
<td>-2131</td>
<td>DEV_SENSOR_TYPE_ERROR</td>
<td>There is no sensor found or the found sensor type is wrong or not supported.</td>
<td></td>
</tr>
<tr>
<td>-2132</td>
<td>DMR_CAMERA_DESCRIPTION_INVALID</td>
<td>A function call was associated with a camera description, that is invalid. One possible reason might be, that the camera description has been deleted (driver closed?). 1.5.0</td>
<td></td>
</tr>
<tr>
<td>Error Code</td>
<td>Error Name</td>
<td>Description</td>
<td>Additional Information</td>
</tr>
<tr>
<td>-----------</td>
<td>------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>-2133</td>
<td>DMR_NEWER_LIBRARY-REQUIRED</td>
<td>A suitable driver library to work with the device manager has been detected, but it is too old to work with this version of the mvDevice Manager library.</td>
<td>This might happen if two different drivers have been installed on the target system and one introduces a newer version of the device manager that is not compatible with the older driver installed on the system. In this case this error message will be written into the log-file together with the name of the library that is considered to be too old. The latest drivers will always be available online under <a href="http://www.matrix-vision.de">www.matrix-vision.de</a>. There will always be an updated version of the library considered to be too old for download from here. 1.6.6</td>
</tr>
<tr>
<td>-2134</td>
<td>DMR_TIMEOUT</td>
<td>A general timeout occurred.</td>
<td>This is the typical result of functions that wait for some condition to be met with a timeout among their parameters. More information can be found in the *.log-file or the debug output. 1.7.2</td>
</tr>
<tr>
<td>-2135</td>
<td>DMR_WAIT_ABANDONED</td>
<td>A wait operation has been aborted.</td>
<td>This e.g. might occur if the user waited for some message to be returned by the driver and the device driver has been closed within another thread. In order to inform the user that this waiting operation terminated in an unusual wait, DMR_WAIT_ABANDONED will be returned then. 1.7.2</td>
</tr>
<tr>
<td>-2136</td>
<td>DMR_EXECUTION_FAIL</td>
<td>The execution of a method object or reading/writing to a feature failed.</td>
<td>More information can be found in the log-file. 1.9.0</td>
</tr>
<tr>
<td>-2137</td>
<td>DEV_REQUEST_ALREADY_IN_USE</td>
<td>This request is currently used by the driver.</td>
<td>This error may occur if the user tries to send a certain request object to the driver by a call to the corresponding image request function. 1.10.31</td>
</tr>
<tr>
<td>Code</td>
<td>Error Code</td>
<td>Description</td>
<td>Exception</td>
</tr>
<tr>
<td>--------</td>
<td>---------------------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>-2138</td>
<td>DEV_REQUEST_BUFFE←R_INVALID</td>
<td>A request has been configured to use a user supplied buffer, but the buffer pointer associated with the request is invalid.</td>
<td>1.10.31 [-2138]</td>
</tr>
<tr>
<td>-2139</td>
<td>DEV_REQUEST_BUFFE←R_MISALIGNED</td>
<td>A request has been configured to use a user supplied buffer, but the buffer pointer associated with the request has an incorrect alignment.</td>
<td>Certain devices need aligned memory to perform efficiently thus when a user supplied buffer shall be used to capture data into this buffer must follow these alignment constraints 1.10.31 [-2139]</td>
</tr>
<tr>
<td>-2140</td>
<td>DEV_ACCESS_DENIED</td>
<td>The requested access to a device could not be granted.</td>
<td>This might e.g. happen if an application tries to access a device exclusively that is already open in another process. This could also happen if a network device has already been opened with control access from another system and the current system also tries to establish control access to the device. 1.10.39 [-2140]</td>
</tr>
<tr>
<td>-2141</td>
<td>DMR_PRELOAD_CHEC←K_FAILED</td>
<td>A pre-load condition for loading a device driver failed.</td>
<td>Certain device drivers may depend on certain changes applied to the system in order to operate correctly. E.g. a device driver might need a certain environment variable to exist. When the device manager tries to load a device driver it performs some basic checks to detect problems like this. When one of these checks fails the device manager will not try to load the device driver and an error message will be written to the selected log outputs. 1.10.52 [-2141]</td>
</tr>
<tr>
<td>Error Code</td>
<td>Description</td>
<td>Reason</td>
<td>Possible Causes</td>
</tr>
<tr>
<td>-------------</td>
<td>------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>-1242</td>
<td>DMR_CAMERA_DESCRIPTION_INVALID_PARAMETER</td>
<td>One or more of the camera description parameters are invalid for the grabber it is used with.</td>
<td>There are multiple reasons for this error code. Detailed information can be found in the .log-file. Possible causes: The TapsXGeometry or TapsYGeometry parameter of the selected camera description cannot be used with a user defined AOI. A scan standard has been selected, that is not supported by this device. An invalid scan rate has been selected. ... This error code will be returned by frame grabbers only. 1.10.57</td>
</tr>
<tr>
<td>-1243</td>
<td>DMR_FILE_ACCESS_ERROR</td>
<td>A general error returned whenever there has been a problem with accessing a file.</td>
<td>There can be multiple reasons for this error and a detailed error message will be sent to the log-output whenever this error code is returned. Possible causes: The driver tried to modify a file, for which it has no write access. The driver tried to read from a file, for which it has no read access ...</td>
</tr>
<tr>
<td>-1244</td>
<td>DMR_INVALID_QUEUE_SELECTION</td>
<td>An error returned when the user application attempts to operate on an invalid queue.</td>
<td>1.11.0</td>
</tr>
<tr>
<td>-1245</td>
<td>DMR_ACQUISITION_ENGINE_BUSY</td>
<td>An error returned when the user application attempts to start the acquisition engine at a time, where it is already running.</td>
<td>2.5.3</td>
</tr>
<tr>
<td>-1246</td>
<td>DMR_BUSY</td>
<td>An error returned when the user application attempts to perform any operation that currently for any reason cannot be started because something else already running.</td>
<td>The log-output will provide additional information. 2.32.0</td>
</tr>
<tr>
<td>-1247</td>
<td>DMR_OUT_OF_MEMORY</td>
<td>An error returned when for any reason internal resources (memory, handles, ...) cannot be allocated.</td>
<td>The log-output will provide additional information. 2.32.0</td>
</tr>
<tr>
<td>-2199</td>
<td>DMR_LAST_VALID_ERROR_CODE</td>
<td>Defines the last valid error code value for device and device manager related errors.</td>
<td>[-2199]</td>
</tr>
</tbody>
</table>
## 21 Glossary

<table>
<thead>
<tr>
<th><strong>Term</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application programming interface (API). The standard API for MATRIX VISION products is called mvIMPACT_Acquire (p. 240).</td>
</tr>
<tr>
<td>BusyBox</td>
<td>Configurable monolithic application including shell and other useful command line tools - often called the &quot;swiss army knife&quot; for embedded systems. Even desktop distributions are sometimes relying on BusyBox due to its robustness. Please see <a href="http://www.busybox.net">http://www.busybox.net</a> for details.</td>
</tr>
<tr>
<td>CIFS</td>
<td>Common Internet file system (CIFS) replaced Samba in 2006. It gets rid of NetBIOS packets an introduced Unix features like soft/hard links and allows larger files.</td>
</tr>
<tr>
<td>CPU</td>
<td>Central processing unit aka processor.</td>
</tr>
<tr>
<td>DAC</td>
<td>Digital to analog converter (D/A converter).</td>
</tr>
<tr>
<td>Defaults</td>
<td>Standard system settings.</td>
</tr>
<tr>
<td>DHCP</td>
<td>Dynamic Host Configuration Protocol (DHCP). DHCP is a protocol used by networked devices (clients) to obtain various parameters necessary for the clients to operate in an Internet Protocol (IP) network.</td>
</tr>
<tr>
<td>Digital I/O</td>
<td>Digital inputs and outputs.</td>
</tr>
<tr>
<td>GDB</td>
<td>GDB, the GNU Project debugger.</td>
</tr>
<tr>
<td>GenICam</td>
<td>GenICam stands for GENeric programming Interface for CAMeras. It's a generic way to access and modify device parameters with a unified interface. A GenICam compliant device either directly provides a GenICam compliant description file (in internal memory) or the description file can be obtained from a local (hard disk etc.) or web location. A GenICam description file is something like a machine readable device manual. It provides a user readable name and value range for parameters (p. 242) that are offered by the device for reading and/or writing and instructions on what command must be send to a device when the user modifies or reads a certain parameter from the device. These description files are written in XML. An excerpt from such a file can be seen in the figure below:</td>
</tr>
</tbody>
</table>

Excerpt of a GenICam description file (in XML)

For further information on this topic please have a look at [https://en.wikipedia.org/wiki/GenICam](https://en.wikipedia.org/wiki/GenICam). |
| GenTL     | GenTL is the transport layer interface for cameras, acquiring images from the camera, and moving them to the user application. |
| **Gigabit Ethernet (GigE)** | The term Gigabit Ethernet (defined by the IEEE 802.3-2008 standard) represents various technologies for transmitting Ethernet frames at a rate of a gigabit per second (1,000,000,000 bits per second). |
GigE Vision is a network protocol designed for the communication between an imaging device and an application. This protocol completely describes:

- device discovery
- data transmission
  - image data
  - additional data
- read/write of parameters.

GigE Vision uses UDP for data transmission to reduce overhead introduced by TCP.

**Note**

UDP does not guarantee the order in which packets reach the client nor does it guarantee that packets arrive at the client at all. However, GigE Vision defines mechanisms that can detect lost packets. This allows capture driver manufacturers to implement algorithms that can reconstruct images and other data by requesting the device to resend lost data packets until the complete buffer has been assembled. For further information please have a look at https://en.wikipedia.org/wiki/GigE_Vision

The MATRIX VISION GigE Vision capture filter driver as well as the socket based acquisition driver and all MATRIX VISION GigE Vision compliant devices support resending thus lost data can be detected and in most cases reconstructed. This of course can not enhance the max. bandwidth of the transmission line thus if e.g. parts of the transmission line are overloaded for a longer period of time data will be lost anyway.

Both capture drivers will allow to fine tune the resend algorithm used internally and both drivers will also provide information about the amount of data lost and the amount of data that was re-requested. This information/configuration will be part of the drivers SDK. More information about it can be found in the corresponding interface description.

**Note**

Under Windows 2000 the filter driver does not support the "Resend" mechanism.
<table>
<thead>
<tr>
<th><strong>High Dynamic Range (HDR)</strong></th>
<th>The HDR (High Dynamic Range) mode increases the usable contrast range. This is achieved by dividing the integration time in two or three phases. The exposure time proportion of the three phases can be set independently. Furthermore, it can be set, how many signal of each phase is charged.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HRTC</strong></td>
<td>With a Hardware Real-Time Controller (HRTC) built inside the FPGA users can define a PLC like sequence of operating steps to control basic time critical functions like exposure time, image trigger and I/O ports. Timing is hard real-time with sub microsecond high resolution.</td>
</tr>
<tr>
<td><strong>IDE</strong></td>
<td>a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of a source code editor, a compiler and/or interpreter, build automation tools, and (usually) a debugger.</td>
</tr>
<tr>
<td><strong>IPKG</strong></td>
<td>Ipsy package management system originally designed for embedded systems. Please have a look at the project homepage <a href="http://handhelds.org/moin/moin.cgi/Ipkg">http://handhelds.org/moin/moin.cgi/Ipkg</a> and the more sophisticated documentation at <a href="http://buffalo.nas-central.org/index.php">http://buffalo.nas-central.org/index.php</a> &quot;Overview--of_the_ipkg_package_management_system&quot; MATRIX VISION distributes all non-firmware, i.e. optional software as ipk packages.</td>
</tr>
</tbody>
</table>
| **JFFS2**                     | JFFS2 is a file system which supports wear levelling. See also Sources about the JFFS file system:  
  - http://sources.redhat.com/jffs2/  
| **Link Aggregation (LAG)**    | Dual-GigE cameras need a network interface card with two network interfaces. However, both network interfaces have to work as a unit. Link aggregation (LAG) or bonding is the name of the game and has to be supported by the network interface card's driver. With it you can bond the two network interfaces so that the work as one interface. |
Logical link address (LLA) is a type of mechanism to obtain a valid IP address without a DHCP server being present. Whether an IP address is available or not is resolved using address resolution protocol (ARP) packets. If no ARP response is received on a given address it is considered unused and will be assigned to the interface. LLA space is 169.254.x.y, i.e. 16bit netmask yielding 64K possible device addresses.

With Linux you have to add LLA as an additional interface. By default, you can find one interface in Connections:

(This description uses "Gnome Network Manager", however using KDE should be similar)

In Wired, you can add interfaces via Add:
In the tab "IPv4 Setting" you have to set "Link-Local Only":

After saving, you will find both connections in the summary:

Now, you can select the wished connection using the left mouse button in the "Network Manager" menu. In the LLA case it is just the new created connection:

<p>| MAC address | Media Access Control address (MAC address) is a quasi-unique identifier attached to most network adapters (NICs) in computer networking. |</p>
<table>
<thead>
<tr>
<th>MTU</th>
<th>Maximum transmission unit (MTU) refers to the size (in bytes) of the largest packet that a given layer of a communications protocol can pass onwards. The default MTU for Ethernet is 1500. The optimum for Gigabit Ethernet is 8000 - 12000. Different MTU settings in the same subnet can cause package losses, i.e. never ever change the MTU unless you know what you're doing. Changing the MTU to other values than 1500 when using file or web services from other computers are almost always a bug. It's save to increase MTU when working in peer-to-peer mode with both devices sharing the same MTU. Please not that few network cards support 16K, most Gigabit Ethernet cards are limited to 9k, some don't support Jumbo Frames (MTU &gt; 1500) at all.</th>
</tr>
</thead>
</table>
| mvIMPACT Acquire | This driver supplied with MATRIX VISION products represents the port between the programmer and the hardware. The driver concept of MATRIX VISION provides a standardized programming interface to all image processing products made by MATRIX VISION GmbH. The advantage of this concept for the programmer is that a developed application runs without the need for any major modifications to the various image processing products made by MATRIX VISION GmbH. You can also incorporate new driver versions, which are available for download free of charge on our website: https://www.matrix-vision.com. The developer interface description of the mvIMPACT Acquire is called
| • mvIMPACT_Acquire_API_CPP_manual.chm |
| • mvIMPACT_Acquire_API_C_manual.chm |
| • mvIMPACT_Acquire_API_NET_manual.chm |
| and can be downloaded from our website. |
| Netboot | With netboot you can boot mvBlueCOUGAR over network. This is especially useful when several devices share the same pieces of software, i.e. same root file system, which might be subject to change frequently. |
| NFS | Network File System (NFS) is a network file system protocol, allowing clients to access files over LAN. Given that you need a NFS server are uncommon on Windows, this protocol best fits for Linux-Linux connections. |
By default, the steps exposure and readout out of an image sensor are done one after the other.

- By design, CCD sensors support overlap capabilities also combined with trigger (see figure).
- In contrast, so-called pipelined CMOS sensors only support the overlapped mode. Even less CMOS sensors support the overlapped mode combined with trigger.

Please check the sensor summary (p. 96).

In overlapping mode, the exposure starts the exposure time earlier during readout.

**Note**

In overlapped trigger mode, you have to keep in mind the following formula:

\[
\text{interval between two trigger events} \geq (\text{readout time} - \text{exposure time})
\]
| PoE | Power over Ethernet. You can power the mvBlueCOUGAR over network cable (IEEE 802.3af (IEEE 802.3at Type 1)).

See also
https://en.wikipedia.org/wiki/Power_over_Ethernet). The IEEE specification allows the power to be injected onto the CAT5e cable in a number of ways. We support power supplied over the data pins as shown in the "RJ45 table" in the chapter "Technical data". This alternative is most common. For more information please have a look at http://www.poweroverethernet.com/most-Recent.php?article_id=52 |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PSE</td>
<td>Power sourcing equipment. The network PoE element that inserts power onto an Ethernet cable.</td>
</tr>
<tr>
<td>SDK</td>
<td>Software development kit (SDK). The standard image processing software library from MARTIX VISION is mvIMPACT.</td>
</tr>
</tbody>
</table>
| SFNC | Standard Feature Naming Convention of GenICam (p.231). See also
The latest GenICam properties list can be found here: http://www.emva.org/standards-technology/genicam/genicam-downloads/ The file is called "GenICam Standard Features Naming Convention (PDF)"

Shell | In computing, a shell is a piece of software that provides an interface for users. Command-line shells provide a command-line interface (CLI) to the operating system. The primary purpose of the shell is to invoke or "launch" another program; however, shells frequently have additional capabilities such as viewing the contents of directories. |
| USB3 Vision | A closed source framework, defined and administered by the Automated Imaging Association (AIA), for transmitting video and related control data over USB 3. Sometimes U3V is used as an acronym. |
| UDP | The User Datagram Protocol (UDP) is an Internet protocol. It is used by applications to send messages to other hosts on an Internet Protocol (IP) network. |
| Virtual Network Computing (VNC) | Virtual Network Computing (VNC) is a graphical desktop sharing system that uses the RFB protocol to remotely control another computer. Over a network, it transmits the mouse and keyboard events from one computer to another, relaying the graphical screen updates back in the other direction.

To access the camera's desktop from a PC via VNC,

- you have to know the IP address of the remote system.
- Start a VNC viewer and
- point it to the remote system.

You won't need a password. Of course, you won't get a very fast live image display via the network with VNC but you should be able to start *wxPropView* (p.115) and capture images. |
|---|---|
| wxWidgets | wxWidgets is a cross-platform GUI library. It can be used from languages such as C++, Python, Perl, and C#/.NET.  

See also  
http://www.wxwidgets.org |
22 Use cases

22.1 GenICam to mvIMPACT Acquire code generator

22.1.1 Using the code generator

As any GenICam™ compliant device for which there is a GenICam™ GenTL compliant capture driver in mvIMPACT Acquire can be used using the mvIMPACT Acquire interface and it can't be known which features are supported by a device until a device has been initialised and its GenICam™ XML file has been processed it is not possible to provide a complete C++ wrapper for every device statically.

Therefore an interface code generator has been embedded into each driver capable of handling arbitrary devices. This code generator can be used to create a convenient C++ interface file that allows access to every feature offered by a device.

**Warning**

A code generated interface can result in incompatibility issues, because the interface name will be constructed from the version and product information that comes with the GenICam XML file (see comment in code sample below). To avoid incompatibility, please use the common interface from the namespace mvIMPACT::acquire::GenICam whenever possible.

To access features needed to generate a C++ wrapper interface a device needs to be initialized. Code can only be generated for the interface layout selected when the device was opened. If interfaces shall be created for more than a single interface layout the steps that will explain the creation of the wrapper files must be repeated for each interface layout.

Once the device has been opened the code generator can be accessed by navigating to "System Settings -> CodeGeneration".

![wxPropView - Code Generation section](image-url)
To generate code first of all an appropriate file name should be selected. In order to prevent file name clashes the following hints should be kept in mind when thinking about a file name:

- If several devices from different families or different vendors shall later be included into an application each device or device family will need its own header file thus either the files should be organized in different subfolders or must have unique names.
- If a device shall be used using different interface layouts again different header files must be generated.

If only a single device family is involved but 2 interface layouts will be used later a suitable file name might be "mvIMPACT_acquire_GenICam_Wrapper_DeviceSpecific.h".

For a more complex application involving different device families using the GenICam interface layout (p. 143) only something like this might make sense:

- mvIMPACT_acquire_GenICam_Wrapper_MyDeviceA.h
- mvIMPACT_acquire_GenICam_Wrapper_MyDeviceB.h
- mvIMPACT_acquire_GenICam_Wrapper_MyDeviceC.h
- ...

Once a file name has been selected the code generator can be invoked by executing the "int GenerateCode()" method:

The result of the code generator run will be written into the "LastResult" property afterwards:
22.1.2 Using the result of the code generator in an application

Each header file generated by the code generator will include "mvIMPACT_CPP/mvIMPACT_acquire.h" thus when an application is compiled with files that have been automatically generated these header files must have access to this file. This can easily achieved by appropriately settings up the build environment / Makefile.

To avoid problems of multiple includes the file will use an include guard build from the file name. Within each header file the generated data types will reside in a sub namespace of "mvIMPACT::acquire" in order to avoid name clashes when working with several different created files in the same application. The namespace will automatically be generated from the ModelName tag and the file version tags in the devices GenICam (p. 234) XML file and the interface layout. For a device with a ModelName tag mvBlueIntelligentDevice and a file version of 1.1.0 something like this will be created:

```
namespace mvIMPACT {
    namespace acquire {
        namespace DeviceSpecific { // the name of the interface layout used during the process of code creation
            namespace MATRIX_VISION_mvBlueIntelligentDevice_1 { // this name will be constructed from the version
                // information that comes with the GenICam XML file
                // by the GenICam standard, different major versions
                // XML file may not be compatible thus different

                // all code will reside in this inner namespace

                } // namespace MATRIX_VISION_mvBlueIntelligentDevice_1
            } // namespace DeviceSpecific
        } // namespace acquire
    } // namespace mvIMPACT
}
```

In the application the generated header files can be used like normal include files:

```c++
#include "mvIMPACT_CPP/mvIMPACT_acquire.h"
```
#include <string>
#include <mvIMPACT_CPP/mvIMPACT_acquire.h>
#include <mvIMPACT_CPP/mvIMPACT_acquire_GenICam_Wrapper_DeviceSpecific.h>

Now to access data types from the header files of course the namespaces must somehow be taken into account. When there is just a single interface that has been created automatically the easiest thing to do would probably an appropriate using statement:

```
using namespace mvIMPACT::acquire::DeviceSpecific::MATRIX_VISION_mvBlueIntelligentDevice_1;
```

If several files created from different devices shall be used and these devices define similar features in a slightly different way this however might result in name clashes and/or unexpected behaviour. In that case the namespaces should be specified explicitly when creating data type instances from the header file in the application:

```cpp
#include <string>
#include <mvIMPACT_CPP/mvIMPACT_acquire.h>
#include <mvIMPACT_CPP/mvIMPACT_acquire_GenICam_Wrapper_DeviceSpecific.h>

void fn( Device* pDev )
{
    DeviceControl dc( pDev );
    if( dc.timestampLatch.isValid() )
    {
        dc.timestampLatch.call();
    }
}
```

When working with a using statement the same code can be written like this as well:

```cpp
#include <string>
#include <mvIMPACT_CPP/mvIMPACT_acquire.h>
#include <mvIMPACT_CPP/mvIMPACT_acquire_GenICam_Wrapper_DeviceSpecific.h>

#ifndef _USE_EXTERNAL_MVBLUEINTelligentDevice
    using namespace mvIMPACT::acquire::DeviceSpecific::MATRIX_VISION_mvBlueIntelligentDevice_1;
#endif

void fn( Device* pDev )
{
    DeviceControl dc( pDev );
    if( dc.timestampLatch.isValid() )
    {
        dc.timestampLatch.call();
    }
}
```

## 22.2 Introducing acquisition / recording possibilities

There are several use cases concerning the acquisition / recording possibilities of the camera:
22.2.1 Acquiring a number of images

As described in chapter Acquisition Control (p.170), if you want to acquire a number of images, you can use as "Setting -> Base -> Camera -> Acquisition Control -> Acquisition Mode" "MultiFrame" and you have to set the "Acquisition Frame Count".

Afterwards, if you start the acquisition via the button "Acquire", the camera will deliver the number of images.

The "MultiFrame" functionality can be combined with an external signal to start the acquisition.

There are several ways and combinations possible, e.g.:

- A trigger starts the acquisition (Figure 1).
- A trigger starts the acquisition start event and a second trigger starts the grabbing itself (Figure 2).

A rising edge at line 4 will start the acquisition of 20 images.
22.2 Introducing acquisition / recording possibilities

22.2.2 Recording sequences in the camera

22.2.2.1 Introduction

Beside the mvPretrigger (p. 249), there are two further mv Acquisition Memory Mode (p. 170) available:

1. mvRecord (p. 170)
   which uses frame rate and trigger settings.

2. mvPlayback (p. 170)
   which disregards the frame rate and trigger settings and outputs the images as fast as possible (throttling by bandwidth setting (p. 315)).

To use mvRecord (p. 170), just

1. set mv Acquisition Memory Mode (p. 170) to mvRecord. (p. 170)
2. Start the camera with "Use" and "Acquire".

The camera will record mv Acquisition Memory Max Frame Count (p. 170) images into the memory.

To use mvPlayback (p. 170), just

1. set mv Acquisition Memory Mode (p. 170) to mvPlayback. (p. 170)
2. Start the camera with "Use" and "Acquire".

The camera will playback (transfer) mv Acquisition Memory Max Frame Count (p. 170) images into the PC memory.

Note

mv Acquisition Memory Max Frame Count (p. 170) can be increased by reducing the image height.

22.2.3 Recording sequences with pre-trigger

22.2.3.1 What is pre-trigger?

With pre-trigger it is possible to record frames before and after a trigger event.
22.2.3.2 How it works

To use this functionality you have to set the **mv Acquisition Memory Mode** (p.170) to "mvPretrigger". This can be used to define the number of frames which will be recorded before the trigger event occurs:

![Image of wxPropView - setting pre-trigger](image1)

Figure 1: wxPropView - setting pre-trigger

Afterwards, you have to define the "AcquisitionStart" or "AcquisitionActive" event. In figure 1 this is **Line4** as trigger event, which starts the regular camera streaming.

Now, start the camera by pressing "**Live**" and generate the acquisition event.

The camera will output the number of pre-trigger frames as fast as possible followed by the frames in live mode as fast as possible until the frame rate is in sync:

**Pretrigger principle:**

![Image of pretrigger principle](image2)

Figure 2: wxPropView - recorded / output images
22.2.4 Creating acquisition sequences (Sequencer Control)

22.2.4.1 Introduction

As mentioned in GenICam and Advanced Features section of this manual, the Sequencer Mode (p. 190) is a feature to define feature sets with specific settings. The sets are activated by a user-defined trigger source and event.

Note

At the moment, the Sequencer Mode is only available for MATRIX VISION cameras with CCD sensors and Sony’s CMOS sensors. Please consult the "Device Feature and Property List"s to get a summary of the actually supported features of each sensor.

The following features are currently available for using them inside the sequencer control:

- BinningHorizontal
- BinningVertical
- CounterDuration (can be used to configure a certain set of sequencer parameters to be applied for the next CounterDuration frames)
- DecimationHorizontal
- DecimationVertical
- ExposureTime
- Gain
- Height
- OffsetX
- OffsetY
- Width
- UserOutputValueAll
- UserOutputValueAllMask
- Multiple conditional sequencer paths.

The Sequencer Control (p. 190) uses Counter And Timer Control (p. 173) Counter1 to work. If you already preset Counter1 and you save a new acquisition sequence, the settings of Counter1 will be overwritten.

Note

Configured sequencer programs are stored as part of the User Sets (p. 381) like any other feature.
22.2.4.2 Creating a sequence using the Sequencer Control in wxPropView

In this sample, we define an acquisition sequence with five different exposure times on the device whereby the last step should be repeated five times. We also activate the digital outputs 0..3 to the sets accordingly - this, for example, can be used as flash signals. All the configuration is done on the device itself so after finishing the configuration and starting the acquisition the device itself will apply the parameter changes when necessary. The host application only needs to acquire the images then. This results in a much faster overall frame rate compared to when applying these changes on a frame to frame basis from the host application.

- 1000 us
- 5000 us
- 10000 us
- 20000 us
- 50000 us (5x)

This will result in the following flow diagram:

![Flow Diagram](image)

Figure 1: Working diagram of the sample

As a consequence the following exposure times will be used to expose images inside an endless loop once the sequencer has been started:

- Frame 1: 1000 us
- Frame 2: 5000 us
- Frame 3: 10000 us
- Frame 4: 20000 us
- Frame 5: 50000 us
- Frame 6: 50000 us
22.2 Introducing acquisition / recording possibilities

- Frame 7: 50000 us
- Frame 8: 50000 us
- Frame 9: 50000 us
- Frame 10: 1000 us
- Frame 11: 5000 us
- Frame 12: 10000 us
- ...

So the actual sequence that will be executed on the device later on will be like this

```cpp
while( sequencerMode == On )
{
    take 1 image using set 0
    take 1 image using set 1
    take 1 image using set 2
    take 1 image using set 3
    take 5 images using set 4
}
```

- There are 2 C++ examples called GenICamSequencerUsage and GenICamSequencerParameterChangeAtRuntime that show how to control the sequencer from an application. They can be found in the Examples section of the C++ interface documentation (p. 199)

The following steps are needed to configure the device as desired:

1. First, switch into the "Configuration Mode": "Sequencer Configuration Mode" = "On". Only then the sequencer on a device can be configured.

**Note**

This is the SFNC (p. 242) way how to create an acquisition sequence and consequently how you have to program it. However, wxPropView offers a wizard (p. 261) to define an acquisition sequence in a much easier way.
2. Set the "Sequencer Feature Selector", if it is not already active (pink box, figure 3): "Sequencer Feature Selector" = "ExposureTime" and "Sequencer Feature Enable" = "1".

3. Set the "Sequencer Feature Selector" for the duration counter (pink box, figure 4): "Sequencer Feature Selector" = "CounterDuration" and "Sequencer Feature Enable" = "1".

4. Then, each sequencer set must be selected by the "Sequencer Set Selector" (orange box, figure 3): "Sequencer Set Selector" = "0".
5. Set the following sequencer set using "Sequencer Set Next" (brown box, figure 3): "Sequencer Set Next" = "1".

6. Set the "Exposure Time" (red box, figure 3): "Exposure Time" = "1000".

7. Finally, save the "Sequencer Set" (green box, figure 3): "int SequencerSetSave()". This ends the configuration of this sequencer set and all the relevant parameters have been stored inside the device's RAM.

8. Set the "UserOutputValueAllMask" (purple box, figure 4) to a suitable value. In this case we want to use all UserOutputs, so we set it to "0xf".

9. Set the "UserOutputValueAll" (red box, figure 4): "UserOutputValueAll" = "0x1".
10. Repeat these steps for the following 3 sequencer sets (Exposure Times 5000, 10000, 20000; UserOutputValueAll 0x2, 0x4, 0x8).

11. For the last sequencer set, set the desired sequencer set with "Sequencer Set Selector" (orange box, figure 5): "Sequencer Set Selector" = "4".

12. Set the following sequencer set with "Sequencer Set Next" and trigger source with "Sequencer Trigger Source" (brown box, figure 5): "Sequencer Set Next" = "0". This will close the loop of sequencer sets by jumping from here back to the first one. "Sequencer Trigger Source" = "Counter1End".

13. Set the "Exposure Time" (red box, figure 5): "Exposure Time" = "50000".
22.2 Introducing acquisition / recording possibilities

14. Set the "Counter Duration" in "Counter And Timer Control" (red box, figure 6): "Counter Duration" = "5".

Figure 5: wxPropView - Sequencer set 4

Figure 6: wxPropView - "Sequencer Mode" = "On"
15. As there are only four UserOutputs we decided not to show sequencer set "4" on the output lines.

16. Finally, save the "Sequencer Set" (green box, figure 3): "int SequencerSetSave()".

17. Leave the "Configuration Mode" (red box, figure 4: "Sequencer Configuration Mode" = "Off").

18. Activate the "Sequencer Mode" (red box, figure 4): "Sequencer Mode" = "On".

Figure 7: wxPropView - "Sequencer Mode" = "On"
The "Sequencer Mode" will overwrite the current device settings.

You will now see that the sequencer sets are processed endlessly. Via the chunk data (activate chunk data (p.179) via "Setting -&gt; Base -&gt; Camera -&gt; GenICam -&gt; Chunk Data Control" - activate "Chunk Mode Active"), the "Info Plot" of the analysis grid can be used to visualize the exposure times:

![Image of wxPropView - Info Plot shows the exposure times]

22.2.4.2.1 Adapting the active time on the output lines via logic gates

If you do not want to see the whole active time of a given sequencer set but only the exposure time of a given sequencer set, you can combine your signal with the logic gates in mvLogicGateControl (p.176). Figure 8 shows sample settings of Line3:

![Image of wxPropView - mvLogicGateControl]
This produces the following output on the output lines:

```
#include <mvIMPACT_CPP/mvIMPACT_acquire_GenICam.h>
GenICam::DigitalIOControl dio = new GenICam::DigitalIOControl(pDev);
dio.userOutputValueAllMask.write( 0xF );
dio.userOutputValueAll.write( 0x1 ); // 0x2, 0x4, 0x8, 0x0
GenICam::mvLogicGateControl mvlgc = new GenICam::mvLogicGateControl(pDev);
mvlgc.mvLogicGateANDSelector.writeS("mvLogicGateAND1");
mvlgc.mvLogicGateANDSource1.writeS("UserOutput0");
mvlgc.mvLogicGateANDSource2.writeS("ExposureActive");
mvlgc.mvLogicGateORSelector.writeS("mvLogicGateOR1");
mvlgc.mvLogicGateORSource1.writeS("mvLogicGateAND1Output");
mvlgc.mvLogicGateORSource2.writeS("Off");
mvlgc.mvLogicGateORSource3.writeS("Off");
mvlgc.mvLogicGateORSource4.writeS("Off");
dio.lineSelector.writeS("Line0");
dio.lineSource.writeS("mvLogicGateOR1Output");
// To output the UserOutputs directly on the output lines would be like this:
// dio.lineSource.writeS("UserOutput0");
```
22.2.4.3 Using the Sequencer Control wizard

Since mvIMPACT Acquire 2.18.0

**wxPropView** (p. 115) offers a wizard for the **Sequencer Control** (p. 190) usage:

![Figure 10: wxPropView - Wizard button](image)

The wizard can be used to get a comfortable overview about the settings of the sequence sets and to create and set a sequence sets in a much easier way:

![Figure 11: wxPropView - Sequencer Control wizard](image)
Just

- select the desired properties,
- select the desired Set tabs,
- set the properties in the table directly, and
- "Auto-Assign Displays To Sets", if you like to show each sequence set in a different display.

Do not forget to save the settings at the end.

22.2.4.4 Working with sequence paths

Since 
mvIMPACT Acquire 2.28.0

It is possible to define sets with a maximum of two active paths. The following diagram shows that two paths are defined in "Set 0": "Path 0" ("SequencePathSelector = 0") is the "standard" path that in this configuration loops and "Path 1" ("SequencePathSelector = 1") will jump to "Set 1" after it is activated via a "RisingEdge" ("Sequencer←TriggerActivation = RisingEdge") signal at "UserOutput0" ("SequencerTriggerSource = 0"). "UserOutput0" can be connected, for example, to one of the digital input lines of the camera:

![Diagram of sequence paths](image)

Figure 12: Working diagram of a sample with sequence paths
There are some specifications concerning the sequencer path feature:

- A path is inactive as soon as the "SequencerTriggerSource" is Off.
- If none of the paths are triggered or both parts are inactive, the sequencer will remain in the current set.
- If both paths were triggered, the path with the trigger that happened first, will be followed.
- As the next sequencer set parameters (like ExposureTime) need to be prepared beforehand, the set sequence might not seem straightforward at first glance. The sequencer will always need one frame to switch to the new set; this frame will be the already prepared set.

22.2.4.5 Programming a sequence with paths using the Sequencer Control

First the sequencer has to be configured.

```cpp
#include <mvIMPACT_CPP/mvIMPACT_acquire_GenICam.h>
GenICam::SequencerControl sc = new GenICam::SequencerControl( pDev );
GenICam::AcquisitionControl ac = new GenICam::AcquisitionControl( pDev );
TDMR_ERROR result = DMR_NO_ERROR;

// general sequencer settings
sc.sequencerMode.writeS( "Off" );
sc.sequencerConfigurationMode.writeS( "On" );
sc.sequencerFeatureSelector.writeS( "ExposureTime" );
sc.sequencerFeatureEnable.write( bTrue );
sc.sequencerFeatureSelector.writeS( "CounterDuration" );
sc.sequencerFeatureEnable.write( bFalse );
sc.sequencerFeatureSelector.writeS( "Gain" );
sc.sequencerFeatureEnable.write( bFalse );
sc.sequencerSetStart.write( 0 );

// set0
sc.sequencerSetSelector.write( 0LL );
ac.exposureTime.write( 1000 );

// set0 path0
sc.sequencerPathSelector.write( 0LL );
sc.sequencerTriggerSource.writeS( "ExposureEnd" );
sc.sequencerSetNext.write( 0LL );

// set0 path1
sc.sequencerPathSelector.write( 1LL );
sc.sequencerTriggerSource.writeS( "UserOutput0" );
sc.sequencerTriggerActivation.writeS( "RisingEdge" );
sc.sequencerSetNext.write( 1LL );

// save set
if( ( result = static_cast<TDMR_ERROR>( sc.sequencerSetSave.call() ) ) != DMR_NO_ERROR )
{
    std::cout << "An error was returned while calling function " << sc.sequencerSetSave.displayName() << " on device " << pDev->serial.read() << " (" << pDev->product.read() << ")": " << ImpactAcquireException::getErrorCodeAsString( result ) << std::endl;
}

// set1
sc.sequencerSetSelector.write( 1LL );
ac.exposureTime.write( 5000 );

// set1 path0
sc.sequencerPathSelector.write( 0LL );
sc.sequencerTriggerSource.writeS( "ExposureEnd" );
sc.sequencerSetNext.write( 0LL );

// set1 path1
sc.sequencerPathSelector.write( 1LL );
sc.sequencerTriggerSource.writeS( "Off" );

// save set
if( ( result = static_cast<TDMR_ERROR>( sc.sequencerSetSave.call() ) ) != DMR_NO_ERROR )
{
    std::cout << "An error was returned while calling function " << sc.sequencerSetSave.displayName() << " on device " << pDev->serial.read() << " (" << pDev->product.read() << ")": " << ImpactAcquireException::getErrorCodeAsString( result ) << std::endl;
}

// final general sequencer settings
sc.sequencerConfigurationMode.writeS( "Off" );
sc.sequencerMode.writeS( "On" );
```

MATRIX VISION GmbH
Then it can later be triggered during runtime.

```cpp
GenICam::DigitalIOControl dic = new GenICam::DigitalIOControl( pDev );
dic.userOutputSelector.write( 0 );
dic.userOutputValue.write( bTrue );
dic.userOutputValue.write( bFalse );
```

This will set an internal event that will cause the sequencer to use set0-path1 at the next possible time, i.e. the next time we are in set0.

There is a C++ example called **GenICamSequencerUsageWithPaths** that shows how to control the sequencer with paths from an application. It can be found in the Examples section of the **C++ Interface documentation** (p.199)

### 22.2.5 Generating very long exposure times

#### 22.2.5.1 Basics

At the moment the exposure time is limited to a maximum of 1 up to 20 seconds depending on certain internal sensor register restrictions. So each device might report a different maximum exposure time.

Since $mvIMPACT$ Acquire 2.28.0

Firmware version 2.28 did contain a major overhaul here so updating to at least this version can result in a much higher maximum exposure time. However, current sensor controllers can be configured to use even longer exposure times if needed using one of the devices timers to create an external exposure signal that can be fed back into the sensor. This use case will explain how this can be done.

This approach of setting up long exposure times requires the sensor of the camera to allow the configuration of an external signal to define the length of the exposure time, so only devices offering the **ExposureMode Trigger Width** can be used for this setup.

**Note**

The maximum exposure time in microseconds that can be achieved in this configuration is the maximum value offered by the timer used.

With **GenICam** (p.234) compliant devices that support all the needed features the setup is roughly like this:

1. Select “Setting -> Base -> Camera -> GenICam -> Counter And Timer Control -> Timer Selector -> Timer 1” and .
2. Set “Timer Trigger Source” = "UserOutput0".
3. Set “Timer Trigger Activation” = "RisingEdge".  
   I.e. a rising edge on UserOutput0 will start Timer1.
4. Then set the "Timer Duration" property to the desired exposure time in us.

5. In "Setting -> Base -> Camera -> GenICam -> Acquisition Control" set the Trigger Selector = "FrameStart". I.e. the acquisition of one frame will start when

6. **Timer1** is active: "Trigger Source" = "Timer1Active".

7. **Exposure** time will be the trigger width: "Exposure Mode" = "TriggerWidth".

The following diagram illustrates all the signals involved in this configuration:

![Figure 1: Long exposure times using GenICam](image)

To start the acquisition of one frame a rising edge must be detected on **UserOutput0** in this example but other configurations are possible as well.

### 22.2.5.2 Setting up the device

The easiest way to define a long exposure time would be by using a single timer. The length of the timer active signal is then used as trigger signal and the sensor is configured to expose while the trigger signal is active. This allows to define exposure time with micro-second precision up the the maximum value of the timer register. With a 32 bit timer register this results in a maximum exposure time of roughly 4295 seconds (so roughly 71.5 minutes).

When writing code e.g. in C# this could look like this:

```csharp
private static void configureDevice(Device pDev, int exposureSec, GenICam.DigitalIOControl ioc)
{
    try
    {
        // establish access to the CounterAndTimerControl interface
        GenICam.CounterAndTimerControl ctc = new mv.impact.acquire.GenICam.CounterAndTimerControl(pDev);
        // set TimerSelector to Timer1 and TimerTriggerSource to UserOutput0
        ctc.timerSelector.writeS("Timer1");
        ctc.timerTriggerSource.writeS("UserOutput0");
        ctc.timerTriggerActivation.writeS("RisingEdge");
        // Set timer duration for Timer1 to value from user input
        ctc.timerDuration.write(exposureSec * 1000000);
    }
    catch
    {
        // handle exception
    }
}
```
// set userOutputSelector to UserOutput0 and set UserOutput0 to inactive. We will later generate a pulse here
ioc.userOutputSelector.writeS("UserOutput0");
ioc.userOutputValue.write(TBoolean.bFalse);

// establish access to the AcquisitionControl interface
GenICam.AcquisitionControl ac = new mv.impact.acquire.GenICam.AcquisitionControl(pDev);
ac.triggerSelector.writeS("FrameStart");
// set TriggerSelector to FrameStart and try to set ExposureMode to TriggerWidth
ac.triggerSource.writeS("Timer1Active");
ac.triggerMode.writeS("On");

// expose as long as we have a high level from Timer1
ac.exposureMode.writeS("TriggerWidth");
}

Note

Make sure that you adjust the ImageRequestTimeout_ms either to 0 (infinite)(this is the default value) or
to a reasonable value that is larger than the actual exposure time in order not to end up with timeouts resulting
from the buffer timeout being smaller than the actual time needed for exposing, transferring and capturing the
image:

ImageRequestTimeout_ms = 0 # or reasonable value

See also

Counter And Timer Control (p. 173)
Digital I/O Control (p. 181)
Acquisition Control (p. 170)

22.2.6 Working with multiple AOIs (mv Multi Area Mode)

Since

mvIMPACT Acquire 2.18.3

22.2.6.1 Introduction

A special feature of Pregius sensors (a.k.a. IMX) from Sony is the possibility to define multiple AOIs (Areas of
Interests - a.k.a. ROI - Regions of Interests) and to transfer them at the same time. Because many applications just
need one or several specific parts in an image to be checked, this functionality can increase the frame rate.

Once activated, the "mv Multi Area Mode" allows you, depending on the sensor, to define up to eight AOIs (mvArea0
to mvArea7) in one image. There are several parameters in combination with the AOIs which are illustrated in the
following figure:
22.2 Introducing acquisition / recording possibilities

Figure 1: Multiple AOIs principle

The “Resulting Offset X” and “Resulting Offset Y” indicates the starting point of the specific AOI in the output image. To complete the rectangular output image, the "missing" areas are filled up with the image data horizontally and vertically. We recommend to use the wizard as a starting point - the wizard provides a live preview of the final merged output image.

22.2.6.2 Using wxPropView

To create multiple AOIs with wxPropView (p.115), you have to do the following step:

1. Start **wxPropView** (p.115) and
2. Connect to the camera.
3. Then change in "Setting -> Base -> Camera -> GenICam -> Image Format Control" the "mv Multi Area Mode" to "mvMultiAreasCombined". Afterwards, "mv Area Selector" is available.
4. Now, select the area a.k.a. AOI you want to create via "mv Area Selector", e.g. "mvArea3" and
5. Set the parameters "mv Area Width", "mv Area Height", "mv Area Offset X", and "mv Area Offset Y" to your needs.
6. Activate the area a.k.a. AOI by checking the box of "mv Area Enable".
7. Finally, start the acquisition by clicking the button "Acquire".

22.2.6.2.1 Using the Multi AOI wizard

Since mvIMPACT Acquire 2.19.0

wxPropView (p.115) offers a wizard for the Multi AOI usage:

The wizard can be used to get a comfortable overview about the settings of the AOIs and to create and set the AOIs in a much easier way:
Figure 4: wxPropView - Multi AOI wizard

Just

- select the desired `mvArea` tabs,
- set the properties like offset, width, and height in the table directly, and
- confirm the changes at the end using the **Ok** button.

The live image shows the created AOIs and the merged or "missing" areas which are used to get the final rectangular output image.

Figure 4: wxPropView - Multi AOI wizard - Live image

22.2.6.3 Programming multiple AOIs

```cpp
#include <mvIMPACT_CPP/mvIMPACT_acquire.h>
#include <mvIMPACT_CPP/mvIMPACT_acquire_GenICam.h>
...
GenICam::ImageFormatControl ifc( pDev );
ifc.mvMultiAreaMode.writeS( "mvMultiAreasCombined" );
ifc.mvAreaSelector.writeS( "mvArea0" );
ifc.mvAreaEnable.write( bTrue );
ifc.mvAreaOffsetX.write( 0 );
ifc.mvAreaOffsetY.write( 0 );
ifc.mvAreaWidth.write( 256 );
ifc.mvAreaHeight.write( 152 );
ifc.mvAreaSelector.writeS( "mvArea1" );
ifc.mvAreaEnable.write( bFalse );
ifc.mvAreaSelector.writeS( "mvArea2" );
ifc.mvAreaEnable.write( bFalse );
```
22.2.7 Working with burst mode buffer

If you want to acquire a number of images at sensor's maximum frame rate while at the same time the image transfer should be at a lower frame rate, you can use the internal memory of the mvBlueCOUGAR-X.

![Burst mode buffer principle](image)

**Figure 1: Principle of burst mode buffering of images**

**Note**

The maximum buffer size can be found in "Setting -> Base -> Camera -> GenICam -> Acquisition Control -> mv Acquisition Memory Max Frame Count".

To create a burst mode buffering of images, please follow these steps:

1. Set image acquisition parameter ("Setting -> Base -> Camera -> GenICam -> Acquisition Control -> mv Acquisition Frame Rate Limit Mode") to "mvDeviceMaxSensorThroughput".

2. Finally, set the acquisition parameter "mv Acquisition Frame Rate Enable" to "Off".
22.2 Introducing acquisition / recording possibilities

Figure 2: wxPropView - Setting the bandwidth using "mv Acquisition Frame Rate Limit Mode"

Alternatively, you can set the burst mode via the desired input frames and the desired output bandwidth:

1. Set image acquisition parameters to the desired input frames per second value ("Setting -> Base -> Camera -> GenICam -> Acquisition Control").

Figure 3: wxPropView - Setting the "Acquisition Frame Rate"
2. Set bandwidth control to the desired MByte/s out value in "Setting -> Base -> Camera -> GenICam -> Device Control -> Device Link Selector -> Device Link Throughput Limit Mode" to "On" and

3. set the desired "Device Link Throughput Limit" in Bits per second (Bps).

![Figure 4: wxPropView - Setting the bandwidth using "Device Link Throughput Limit Mode"](image)

Now, the camera will buffer burst number of images in internal memory and readout at frames per second out.

See also

**Limiting the bandwidth of the imaging device** (p. 315)

22.2.7.1 Triggered frame burst mode

With the `triggerSelector "FrameBurstStart"`, you can also start a frame burst acquisition by a trigger. A defined number of images ("AcquisitionBurstFrameCount") will be acquired directly one after the other. With the "mv Acquistion Frame Rate Limit Mode" set to `mvDeviceMaxSensorThroughput`, the there won't be hardly any gap between these images.

As shown in figure 5, "FrameBurstStart" can be trigger by a software trigger, too.
22.2 Introducing acquisition / recording possibilities

Figure 5: wxPropView - Setting the frame burst mode triggered by software

FrameBurstStart

Figure 6: Principle of FrameBurstStart
22.2.8 Using the SmartFrameRecall feature

Since mvIMPACT Acquire 2.18.0

22.2.8.1 Introduction

The SmartFrameRecall is a new, FPGA based smart feature which takes the data handling of industrial cameras to a new level.

So far, the entire data amount has to transferred to the host PC whereby the packetizer in the camera split the data packages and distributed them to the two Gigabit Ethernet lines. On the host PC, the data was merged again, shrank, and this AOI was possibly processed (Figure 1).

![Figure 1: Data handling so far](image)

This procedure has several disadvantages:

- Both data lines (in case of Dual GigE cameras for example) for each camera are required which
  - leads to high cabling efforts.
- A high end PC is needed to process the data which
  - leads to high power consumptions.
- In USB 3 multi-camera solutions (depending on the resolution) each camera requires a separate connection line to the host PC which
  - limits the cabling possibilities, the possible distances, and makes an installation more complex (without the possibilities to use hubs, for example).
- Last but not least, the frame rates a limited to the bandwidth.

The SmartFrameRecall is a new data handling approach which buffers the hi-res images in the camera and only transfers thumbnails. You or your software decides on the host PC which AOI should be sent to the host PC (Figure 2).
22.2 Introducing acquisition / recording possibilities

This approach allows

- higher bandwidths,
- less CPU load and power consumption,
- higher frame rates, and
- less complex cabling.

22.2.8.2 Implementing a SmartFrameRecall application

First of all, clarify if the SmartFrameRecall makes sense to be used in your application:

- Taking a short look on the thumbnails, could you specify which frames don't interest you at all or which portions of the frame suffice to you for further processing?

If you can, your application can do so too, and then SmartFrameRecall is the right framework for you. To use the SmartFrameRecall follow these steps:

- Activate the Chunk Data Control:
#include <mvIMPACT_CPP/mvIMPACT_acquire.h>
#include <mvIMPACT_CPP/mvIMPACT_acquire_GenICam.h>

... GenICam::ChunkDataControl cdc( pDev );
cdc.chunkModeActive.write( bTrue );
cdc.chunkSelector.writeS( "Image" );
cdc.chunkEnable.write( bTrue );
cdc.chunkSelector.writeS( "mvCustomIdentifier" );
cdc.chunkEnable.write( bTrue );
...

It is necessary to activate the chunk data (p. [179]) that your application can easily distinguish frames belonging to the normal stream from the ones requested by the host application.

- Reduce the size of the streamed images. This can reduce the size using Decimation, both horizontally and vertically. E.g. setting decimation to 16, a normal image will only consume 16*16 thus 1/256th of the bandwidth:

  ... 
  GenICam::ImageFormatControl ifc( pDev );
  ifc.decimationHorizontal.write( 16 );
  ifc.decimationVertical.write( 16 );
  ...

- Make sure that the resulting image width is a multiple of 8! If this is not the case the SmartFrameRecall feature cannot be activated.

- Activate the SmartFrameRecall feature:

  ... 
  GenICam::AcquisitionControl ac( pDev );
  ac.mvSmartFrameRecallEnable.write( bTrue );
  ...

This will configure the devices internal memory to store each frame (that gets transmitted to the host) in full resolution. These images can be requested by an application when needed. As soon as the memory is full, the oldest one will be removed from the memory whenever a new one becomes ready (FIFO).

- Analyze the images of the reduced data stream.

- If necessary, request the desired image in full resolution:

  ... 
  struct ThreadParameter
  {   
    Device* pDev;
    GenICam::CustomCommandGenerator ccg;
  } ...
  
  unsigned int DMR_CALL liveThread( void* pData )
  {
    ThreadParameter* pThreadParameter = reinterpret_cast<ThreadParameter*>(pData);
    ...
    pThreadParameter->ccg.setRequestTransmission( pRequest, x, y, w, h, rtmFullResolution, cnt );
    ...
  } ...

 MATRIX VISION GmbH
The last parameter of requestTransmission will be written into the chunkmvCustomIdentifier and your application can recognize the request.

- Finally do your analysis/processing with the requested image in full resolution.

We provide

- a basic SmartFrameRecall C++ sample called GenICamSmartFrameRecallUsage.cpp which will be installed together with "mvGenTL Acquire" driver.
  You find the sample’s description in the online manual of the mvIMPACT Acquire API C++: https://www.matrix-vision.com/manuals/SDK_CPP/GenICamSmartFrameRecallUsage.cpp-example.html

- a advanced SmartFrameRecall C# sample called SmartFrameRecall.cs which will be installed together with "mvGenTL Acquire" driver.
  You find the sample’s description in the online manual of the mvIMPACT Acquire API .NET: https://www.matrix-vision.com/manuals/SDK_NET/SmartFrameRecall cs-example.html

22.2.9 Using VLC Media Player

With the DirectShow Interface (p. 205) MATRIX VISION devices become a (acquisition) video device for the VLC Media Player.

Figure 1: VLC Media Player with a connected device via DirectShow
22.2.9.1 System requirements

It is necessary that following drivers and programs are installed on the host device (laptop or PC):

- Windows 7, 32 bit or 64 bit
- up-do-date VLC Media Player, 32 bit or 64 bit (here: version 2.0.6)
- up-do-date MATRIX VISION driver, 32 bit or 64 bit (here: version 2.5.6)

Attention

Using Windows 10: VLC Media Player with versions 2.2.0 or smaller have been tested successfully. Newer versions do NOT work with mvIMPACT Acquire! There are some bug tickets in the VLC repository that might be related (At the time of writing none of these seem to have a fix):

- [https://trac.videolan.org/vlc/ticket/14278](https://trac.videolan.org/vlc/ticket/14278) (According to [http://www.digital-digest.com/software/videolan_history.html](http://www.digital-digest.com/software/videolan_history.html#83) may have been fixed but the trac system suggests otherwise)
- [https://trac.videolan.org/vlc/ticket/14489](https://trac.videolan.org/vlc/ticket/14489) (According to [http://www.digital-digest.com/software/videolan_history.html](http://www.digital-digest.com/software/videolan_history.html#83) may have been fixed but the trac system suggests otherwise)
- [https://trac.videolan.org/vlc/ticket/15511](https://trac.videolan.org/vlc/ticket/15511)

22.2.9.2 Installing VLC Media Player

1. Download a suitable version of the VLC Media Player from the VLC Media Player website mentioned below.
2. Run the setup.
3. Follow the installation process and use the default settings.

A restart of the system is not required.

See also


22.2.9.3 Setting up MV device for DirectShow

Note

Please be sure to register the MV device for DirectShow with the right version of `mvDeviceConfigure` (p. 156). I.e. if you have installed the 32 bit version of the VLC Media Player, you have to register the MV device with the 32 bit version of `mvDeviceConfigure` (p. 156) (“C:/Program Files/MATRIX VISION/mvIMPACT Acquire/bin”) !

1. Connect the MV device to the host device directly or via GigE switch using an Ethernet cable.
2. Power the camera using a power supply at the power connector.
3. Wait until the status LED turns blue.
4. Open the tool `mvDeviceConfigure` (p. 156).
5. set a friendly name (p. 208),
6. and register the MV device for DirectShow (p. 206).

Note

In some cases it could be necessary to repeat step 5.
22.2.9.4 Working with VLC Media Player

1. Start VLC Media Player.
2. Click on "Media -> Open Capture Device...".

![Open Capture Device](image1.png)

Figure 2: Open Capture Device...

3. Select the tab "Device Selection".
4. In the section "Video device name", select the friendly name of the MV device:

![Video device name](image2.png)

Figure 3: Video device name

5. Finally, click on "Play".
   After a short delay you will see the live image of the camera.
22.2.10 Using the linescan mode

Both CMOS sensors from e2v offer a line scan mode. One (gray scale sensor) or two lines (in terms of color sensor) can be selected to be read out of the full line height of 1024 or 1200 lines. This or these lines are grouped to a pseudo frame of selectable height in the internal buffer of the camera.

Complete instructions for using the line scan mode are provided here:

- sensor description of line scan mode (-x02e) (p. 643)
- sensor description of line scan mode (-x02eGE) (p. 651)
- sensor description of line scan mode (-x04e) (p. 657)

22.2.10.1 System requirements

- mvBlueCOUGAR-X
  - "firmware version" at least "1.6.32.0"
- mvBlueFOX3
  - "firmware version" at least "1.6.139.0"

22.2.10.2 Initial situation and settings

Generally, line scan cameras are suitable for inspections of moving, continuous materials. In order that the line scan camera acquires the line at the right time, an incremental encoder (p. 336), for example, at a conveyor belt triggers the line scan camera. Normally, an incremental encoder (p. 336) does this using a specific frequency like 1:1 which means that there is a signal for every line. However, during the adjustment of a line trigger application or while choosing a specific incremental encoder you have to keep the specific frequency in mind.

**Note**

Using timers and counters (p. 173) it is possible to skip trigger signals.

In line scan mode, the camera adds the single lines to one image of the height of max. 1024 or 1200 lines (according to the used sensor). The images are provided with no gap.

**Note**

While using the line scan mode with a gray scale sensor, one trigger signal will lead to one acquired image line. Using a color sensor, one trigger signal will lead to two acquired image lines.

Due to aberrations at the edges of the lens, you should set an offset in the direction of Y ("Offset Y", see the following figure), generally around the half of the sensor's height (a.k.a. sensor's Y center). With Offset Y you can adjust the scan line in the direction of motion.
22.2 Introducing acquisition / recording possibilities

22.2.10.2.1 Scenarios

With regards to the external trigger signals provided by an incremental encoder (p. 336), there are two possible scenarios:

1. A conveyor belt runs continuously and so does the incremental encoder (p. 336), or - like in a reverse vending machine,
2. a single item is analyzed and the conveyor belt and so the incremental encoder (p. 336) stops after the inspection and restarts for the next item.

In the first scenario you can use the standard settings of the MATRIX VISION devices. Please have a look at the sample Sample 2: Triggered line scan acquisition with a specified number of image blocks and pausing trigger signals (p. 283) scan. Sample "Triggered line scan acquisition with exposure time of 250 us" which shows how you can set the line scan mode with continuous materials and signals from the encoder. However, it is absolutely required that the external trigger is always present. During a trigger interruption, controlling or communication to the camera is not possible.

In the second scenario, the external trigger stops. If there is a heartbeat functionality in the system (e.g. with GigE Vision), there can be a halt of the system. Please have a look at the sample Triggered line scan acquisition with a specified number of image blocks and pausing trigger signals (p. 281) which shows how you can handle pausing trigger signals.

22.2.10.3 Sample 1: Triggered linescan acquisition with exposure time of 250 us

This sample will show you how to use the line scan mode of the sensors -x02e (p. 643) and -x02eGE (p. 651) using an external trigger provided by an incremental encoder (p. 336) which sends a "trigger for every line" (1:1)

Note

You can also use the sensor -x04e (p. 655). However, the sensor is slower due to the higher number of pixels.

In this sample, we chose an exposure time of "250 us" and to ease the calculations we used "1000 px image height".

MATRIX VISION GmbH
Note

To get suitable image results, it might be necessary to increase the gain or the illumination.

These settings result in a max. "frame rate" of "2.5 frames per second".

To adjust the opto-mechanics (focus, distance, illumination, etc.), you can use the area mode of the sensor. That's a main advantage of an area sensor with line scan mode compared to a line scan camera!

You will need the following pins of the mvBlueCOUGAR-X:

<table>
<thead>
<tr>
<th>Pin.</th>
<th>Signal (Standard version)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GND</td>
<td>Common ground</td>
</tr>
<tr>
<td>2</td>
<td>12V .. 24V</td>
<td>Power supply</td>
</tr>
<tr>
<td>4</td>
<td>Opto DigIn0 (line4)</td>
<td>The output signal A of the incremental encoder</td>
</tr>
</tbody>
</table>

22.2.10.3.1 Setting the application in wxPropView

Summary of our sample:

<table>
<thead>
<tr>
<th>Property name wxPropView</th>
<th>Setting</th>
<th>GenICam Control (p.167)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device Scan Type</td>
<td>line</td>
<td>Scan</td>
<td></td>
</tr>
<tr>
<td>Height (in pixels)</td>
<td>1000</td>
<td>1000</td>
<td>Image Format Control</td>
</tr>
<tr>
<td>Offset Y (in pixels)</td>
<td>500</td>
<td>500</td>
<td>Image Format Control</td>
</tr>
<tr>
<td>Exposure Time (in micro-sec)</td>
<td>250.000</td>
<td>250.000</td>
<td>Acquisition Control</td>
</tr>
<tr>
<td>Trigger Mode</td>
<td>On</td>
<td></td>
<td>Acquisition Control</td>
</tr>
<tr>
<td>Trigger Source</td>
<td>line4</td>
<td></td>
<td>Acquisition Control</td>
</tr>
<tr>
<td>ImageRequestTimeout_ms (in milliseconds)</td>
<td>0 ms</td>
<td>-</td>
<td>This is necessary otherwise there will be error counts and no frames are created.</td>
</tr>
</tbody>
</table>
22.2 Introducing acquisition / recording possibilities

22.2.10.4 Sample 2: Triggered line scan acquisition with a specified number of image blocks and pausing trigger signals

This section will provide you with some information you have to keep in mind while working with pausing triggers and specified number of image blocks.

First of all, using mvBlueCOUGAR-X or mvBlueCOUGAR-XD it is necessary to disable the heartbeat of the GigE Vision control protocol (GVCP) (“Device Link Heartbeat Mode = On”) otherwise a paused trigger signal can be misinterpreted as a lost connection:
Secondly, since the conveyor belt stops sometime, the trigger will do so, too. Be sure, that the trigger signal is available until the last image block was received.

Thirdly, if you know the number of image blocks, you can use the How to see the first image (p.[120] MultiFrame functionality (in “Setting -> Base -> Camera -> GenICam -> Acquisition Control” set “Acquisition Mode = MultiFrame” and “Acquisition Frame Count”). This will acquire the number of image blocks and will stop the acquisition afterwards.

22.2.11 Working with Event Control

The mvBlueCOUGAR-X camera generates Event notifications. An Event is a message that is sent to the host application to notify it of the occurrence of an internal event. With “Setting -> Base -> Camera -> GenICam -> Event Control” you can handle these notifications.

At the moment, it is possible to handle
22.2 Introducing acquisition / recording possibilities

- Exposure End (= sensor’s exposure end)
- Line 4 (= DigIn0) Rising Edge
- Line 5 (= DigIn1) Rising Edge
- Frame End (= the camera is ready for a new trigger)

22.2.11.1 Setting Event notifications using wxPropView

To activate the notifications, just

1. Select the Event via “Setting -> Base -> Camera -> GenICam -> Event Control -> Event Selector”, e.g. ExposureEnd.
2. Set the “Event Notification” to “On”.

Afterwards, it is possible to attach a custom callback that gets called whenever the property is modified. E.g. if you want to attach a callback to the Frame ID after the exposure was finished, you have to

1. select “Setting -> Base -> Camera -> GenICam -> Event Control -> Event Exposure End Data -> Event Exposure End Frame ID”;
2. right-click on the property, and
3. click on “Attach Callback”.

![Figure 1: wxPropView - "Attach Callback" to Event Exposure End Frame ID](image)
Now, you can track the property modifications in the output window:

Figure 2: wxPropView - Output window with the Event notifications

You can find a detailed Callback code example in the C++ API manual.

22.3 Improving the acquisition / image quality

There are several use cases concerning the acquisition / image quality of the camera:

22.3.1 Correcting image errors of a sensor

Due to random process deviations, technical limitations of the sensors, etc. there are different reasons that image sensors have image errors. MATRIX VISION provides several procedures to correct these errors, by default these are host-based calculations.

However, the mvBlueCOUGAR-X, for example, also supports a camera-based Flat-Field Correction, which saves dozens of % CPU load and lowers latency.

Provided image corrections procedures are
1. Defective Pixels Correction (p. 288),
2. Dark Current Correction (p. 292), and
3. Flat-Field Correction (p. 294).

**Note**

If you execute all correction procedures, you have to keep this order. All gray value settings of the corrections below assume an 8-bit image.

![Host-based image corrections](image.png)

Figure 1: Host-based image corrections

The path "Setting -> Base -> ImageProcessing -> ..." indicates that these corrections are host-based corrections.

Before starting consider the following hints:

- To correct the complete image, you have to make sure no user defined AOI has been selected: Right-click "Restore Default" on the devices AOI parameters Width and Height or "Setting -> Base -> Camera -> GenICam -> Image Format Control" using the GenICam interface layout (p. 143).

- You have several options to save the correction data. The chapter Storing and restoring settings (p. 125) describes the different ways.

**See also**

There is a white paper about image error corrections with extended information available on our website: http://www.matrix-vision.com/ti_files/mv11/Glossary/art_image_errors←sensors_en.pdf
22.3.1.1 Defective Pixels Correction

Due to random process deviations, not all pixels in an image sensor array will react in the same way to a given light condition. These variations are known as blemishes or defective pixels.

There are three types of defective pixels:

1. **leaky pixel** (in the dark)
   which indicates pixels that produce a higher read out code than the average

2. **hot pixel** (in standard light conditions)
   which indicates pixels that produce a higher non-proportional read out code when temperatures are rising

3. **cold pixel** (in standard light conditions)
   which indicates pixels that produce a lower read out code than average when the sensor is exposed (e.g. caused by dust particles on the sensor)

**Note**

Please use either an 8 bit Mono or Bayer image format when correcting the image. After the correction, all image formats will be corrected.

To correct the defective pixels there are three neighbor replace methods:

1. **"Replace 3x1 average"**
   which replaces the detected defective pixels with the average value from the left and right neighbor pixel (3x1)

2. **"Replace 3x3 median"**
   which replaces the detected defective pixels with the median value calculated from the nearest neighbors (3x3)

3. **"Replace 3x3 Filtered Data Averaged"**
   which replaces and treats the detected defective pixels as being fed into a 3x3 debayer algorithm before reaching the filter
   Only for devices which do not offer a defective pixel compensation; packed RGB or packed YUV444 data is needed.

22.3.1.1.1 Correcting leaky pixels

To correct **leaky pixels** the following steps are necessary:

1. Set gain ("Setting -> Base -> Camera -> GenICam -> Analog Control -> Gain = 0 dB") and exposure time
   "Setting -> Base -> Camera -> GenICam -> Acquisition Control -> ExposureTime = 360 msec" to the given operating conditions
   The total number of defective pixels found in the array depend on the gain and the exposure time.

2. Black out the lens completely

3. Set the (Filter-) "Mode = Calibrate leaky pixel"

4. Snap an image ("Acquire" with "Acquisition Mode = SingleFrame")

5. To activate the correction, choose one of the neighbor replace methods: "Replace 3x1 average", 
   "Replace 3x3 median" or "Replace 3x3 Filtered Data Averaged"

6. Save the settings including the correction data via "Action -> Capture Settings -> Save Active Device Settings"
   (Settings can be saved in the Windows registry or in a file)
22.3 Improving the acquisition / image quality

Note

After having re-started the camera you have to reload the capture settings vice versa.

The filter checks:

Pixel > LeakyPixelDeviation_ADCLimit // (default value: 50)

All pixels above this value are considered as leaky pixel.

22.3.1.1.2 Correcting hot pixels

To correct hot pixels the following steps are necessary:

1. Set gain ("Setting -> Base -> Camera -> GenICam -> Analog Control -> Gain = 0 dB") and exposure time "Setting -> Base -> Camera -> GenICam -> Acquisition Control -> ExposureTime = 360 msec" to the given operating conditions. The total number of defective pixels found in the array depend on the gain and the exposure time.

2. Black out the lens completely.

3. Set the (Filter) "Mode = Calibrate Hot Pixel"

4. Snap an image ("Acquire" with "Acquisition Mode = SingleFrame")

5. To activate the correction, choose one of the neighbor replace methods: "Replace 3x1 average", "Replace 3x3 median" or "Replace 3x3 Filtered Data Averaged"

6. Save the settings including the correction data via "Action -> Capture Settings -> Save Active Device Settings"
   (Settings can be saved in the Windows registry or in a file)

Note

After having re-started the camera you have to reload the capture settings vice versa.

22.3.1.1.3 Correcting cold pixels

To correct cold pixels the following steps are necessary:

1. You will need a uniform sensor illumination approx. 50 - 70 % saturation (which means an average gray value between 128 and 180)

2. Set the (Filter) "Mode = Calibrate cold pixel" (Figure 2)

3. Snap an image ("Acquire" with "Acquisition Mode = SingleFrame")

4. To activate the correction, choose one of the neighbor replace methods: "Replace 3x1 average", "Replace 3x3 median" or "Replace 3x3 Filtered Data Averaged"

5. Save the settings including the correction data via "Action -> Capture Settings -> Save Active Device Settings"
   (Settings can be saved in the Windows registry or in a file)
Note

After having re-started the camera you have to reload the capture settings vice versa.

The filter checks:

Pixel < T[cold] // (default value: 15 %)
// T[cold] = deviation of the average gray value (ColdPixelDeviation_pc)

All pixels below this value have a dynamic below normal behavior.

Figure 2: Image corrections: DefectivePixelsFilter

Note

Repeating the defective pixel corrections will accumulate the correction data which leads to a higher value in "DefectivePixelsFound". If you want to reset the correction data or repeat the correction process you have to set the (Filter) "Mode = Reset Calibration Data".
22.3.1.1.4 Storing pixel data on the device

To save and load the defective pixel data, appropriate functions are available:

- `int mvDefectivePixelDataLoad( void )`
- `int mvDefectivePixelDataSave( void )`

The section "Setting -> Base -> ImageProcessing -> DefectivePixelsFilter" was also extended (see Figure 2a). First, the DefectivePixelsFound indicates the number of found defective pixels. The coordinates are available through the properties DefectivePixelOffsetX and DefectivePixelOffsetY now. In addition to that it is possible to edit, add and delete these values manually (via right-click on the "DefectivePixelOffset" and select "Append Value" or "Delete Last Value"). Second, with the function

- `int mvDefectivePixelReadFromDevice( void )`
- `int mvDefectivePixelWriteToDevice( void )`

you can exchange the data from the filter with the camera and vice versa.

![Figure 2a: Image corrections: DefectivePixelsFilter (since driver version 2.17.1 and firmware version 2.12.406)]
Just right-click on `mvDefectivePixelWriteToDevice` and click on "Execute" to write the data to the camera (and hand over the data to the mv Defective Pixel Correction Control (p. 194)). To permanently store the data inside the cameras non-volatile memory afterwards `mvDefectivePixelDataSave` must be called as well!

![Image](image.png)

Figure 2b: Defective pixel data are written to the camera (since driver version 2.17.1 and firmware version 2.12.406)

While opening the camera, the camera will load the defective pixel data from the camera. If there are pixels in the filter available (via calibration), nevertheless you can load the values from the camera. In this case the values will be merged with the existing ones. I.e., new ones are added and duplicates are removed.

### 22.3.1.2 Dark Current Correction

Dark current is a characteristic of image sensors, which means, that image sensors also deliver signals in total darkness by warmness, for example, which creates charge carriers spontaneously. This signal overlays the image information. Dark current depends on two circumstances:

1. **Exposure time**
   The longer the exposure, the greater the dark current part. I.e. using long exposure times, the dark current itself could lead to an overexposed sensor chip

2. **Temperature**
   By cooling the sensor chips the dark current production can be highly dropped (approx. every 6 °C the dark current is cut in half)
22.3.1.2.1 Correcting Dark Current

The dark current correction is a pixel wise correction where the dark current correction image removes the dark current from the original image. To get a better result it is necessary to snap the original and the dark current images with the same exposure time and temperature.

**Note**

Dark current snaps generally show noise.

To correct the **dark current pixels** following steps are necessary:

1. Black out the lens completely
2. Set exposure time according to the application
3. Set the number of image for calibration in "Setting -> Base -> ImageProcessing -> DarkCurrentFilter -> CalibrationImageCount" (Figure 3).
4. Set "Setting -> Base -> ImageProcessing -> DarkCurrentFilter -> Mode" to "Calibrate" (Figure 3)
5. Snap an image ("Acquire" with "Acquisition Mode = SingleFrame")
6. Finally, you have to activate the correction: Set "Setting -> Base -> ImageProcessing -> DarkCurrentFilter -> Mode" to "On"
7. Save the settings including the correction data via "Action -> Capture Settings -> Save Active Device Settings" (Settings can be saved in the Windows registry or in a file)

The filter snaps a number of images and averages the dark current images to one correction image.

**Note**

After having re-started the camera you have to reload the capture settings vice versa.

![Figure 3: Image corrections: CalibrationImageCount](image-url)
22.3.1.3 Flat-Field Correction

Each pixel of a sensor chip is a single detector with its own properties. Particularly, this pertains to the sensitivity as the case may be the spectral sensitivity. To solve this problem (including lens and illumination variations), a plain and equally "colored" calibration plate (e.g. white or gray) as a flat-field is snapped, which will be used to correct the original image. Between flat-field correction and the future application you must not change the optic. To reduce errors while doing the flat-field correction, a saturation between 50 % and 75 % of the flat-field in the histogram is convenient.

**Note**

Flat-field correction can also be used as a destructive watermark and works for all f-stops.

To make a **flat field correction** following steps are necessary:

1. You need a plain and equally "colored" calibration plate (e.g. white or gray)
2. No single pixel may be saturated - that's why we recommend to set the maximum gray level in the brightest area to max. 75% of the gray scale (i.e., to gray values below 190 when using 8-bit values)
3. Choose a BayerXY in "Setting -> Base -> Camera -> GenICam -> Image Format Control -> PixelFormat".
4. Set the (Filter-) "Mode = Calibrate" (Figure 6)
5. Start a Live snap ("Acquire" with "Acquisition Mode = Continuous")
6. Finally, you have to activate the correction: Set the (Filter-) "Mode = On"
7. Save the settings including the correction data via "Action -> Capture Settings -> Save Active Device Settings"
   (Settings can be saved in the Windows registry or in a file)
22.3 Improving the acquisition / image quality

Note

After having re-started the camera you have to reload the capture settings vice versa.

The filter snaps a number of images (according to the value of the `CalibrationImageCount`, e.g. 5) and averages the flat-field images to one correction image.

![Figure 6: Image corrections: Host-based flat field correction](image)

### 22.3.1.3.1 Camera-based Flat-Field Correction

The camera-based Flat-Field Correction feature supports full AOI and 14 bit to 14 bit correction (12 bit coefficients). This enables a pixel to pixel correction and saves dozens of % CPU load and lowers latency. To reduce noise averaging of a number of "mv Flat-Field Correction Calibration Images" is possible. A correction image in the camera is then calculated. This may take some time / number of images, however, the camera blinks green. One correction image can be stored for all user settings.

The camera-based Flat-Field Correction is independent of the offset and uses a run-mode trigger (e.g. external trigger).

You can set the "camera-based flat field correction" in the following way:
1. Set "**mv Flat-Field Correction Calibration Image Count**" to, for example, 5.

2. This will average 5 images before calculating the FFC factors to reduce impact of noise.

3. Stop "**Continuous**" acquisition mode, then right click on "int mvFFCCalibrate()" -> "**Execute**".

4. Finally, you have to activate the correction: Set the "**mv Flat-Field Correction Enable = 1**".

Depending on the sensor, this need some time, because the data is stored in the internal flash memory (yellow LED lights up).

![Figure 7: wxPropView - settings](image)

**Example**

![Figure 8: wxPropView - inhomogeneous light / pixel histogram](image)
Figure 9: wxPropView - inhomogeneous light / pixel histogram (horizontal)

Figure 10: wxPropView - compensated light / pixel histogram
22.3.2 Optimizing the color fidelity of the camera

Purpose of this chapter is to optimize the color image of a camera, so that it looks as natural as possible on different displays and for human vision.

This implies some linear and nonlinear operations (e.g. display color space or Gamma viewing LUT) which are normally not necessary or recommended for machine vision algorithms. A standard monitor offers, for example, several display modes like sRGB, "Adobe RGB", etc., which reproduce the very same color of a camera color differently.

It should also be noted that users can choose for either

- camera based settings and adjustments or
- host based settings and adjustments or
- a combination of both.

Camera based settings are advantageous to achieve highest calculating precision, independent of the transmission bit depth, lowest latency, because all calculations are performed in FPGA on the fly and low CPU load, because the host is not invoked with these tasks. These camera based settings are

- **gamma correction** (p. 301)
• negative gain / gain (p. 301)
• look-up table (LUT) (p. 301)
• white balance (p. 305)
• offset (p. 307)
• saturation and color correction (p. 309)

Host based settings save transmission bandwidth at the expense of accuracy or latency and CPU load. Especially performing gain, offset, and white balance in the camera while outputting RAW data to the host can be recommended.

Of course host based settings can be used with all families of cameras (e.g. also mvBlueFOX).

Host based settings are:

• look-up table (LUTOperations)
• color correction (ColorTwist)

The block diagram of the camera (p. 96) shows where the different settings can be found in the data flow of the image data.

To show the different color behaviors, we take a color chart as a starting point:

![Color chart](image)

Figure 1: Color chart as a starting point

If we take a SingleFrame image without any color optimizations, an image can be like this:
Figure 2: SingleFrame snap without color optimization

Figure 3: Corresponding histogram of the horizontal white to black profile

As you can see,

- saturation is missing,
- white is more light gray,
- black is more dark gray,
- etc.

**Note**

You have to keep in mind that there are two types of images: the one generated in the camera and the other one displayed on the computer monitor. Up-to-date monitors offer different display modes with different color spaces (e.g. sRGB). According to the chosen color space, the display of the colors is different.

The following figure shows the way to a perfect colored image
22.3 Improving the acquisition / image quality

including these process steps:

1. Do a **Gamma correction (Luminance)** (p. 301),
2. make a **White balance** (p. 305) and
3. Improve the **Contrast** (p. 307).
4. Improve **Saturation** (p. 309), and use a "color correction matrix" for both
   (a) the sensor and / or
   (b) the monitor.

The following sections will describe the single steps in detail.

22.3.2.1 Step 1: Gamma correction (Luminance)

First of all, a **Gamma correction** (Luminance) can be performed to change the image in a way how humans perceive light and color.

For this, you can change either

- the exposure time,
- the aperture or
- the gain.

You can change the gain via **wxPropView** (p. 115) like the following way:

1. Click on "Setting -> Base -> Camera -> GenICam -> LUT Control -> LUT Selector".
2. Afterwards, click on "Wizard" to start the LUT Control wizard tool. The wizard will load the data from the camera.
Figure 5: Selected LUT Selector and click on wizard will start wizard tool
3. Now, click on the "Gamma..." button

4. and enter e.g. "2.2" as the Gamma value:

5. Then, click on "Copy to..." and select "All" and

6. and click on "Enable All".

7. Finally, click on Synchronize and play the settings back to the device (via "Cache -> Device").
After gamma correction, the image will look like this:

![Image](image_url)

Figure 9: After gamma correction

After gamma correction, the image will look like this:

![Image](image_url)

Figure 10: Corresponding histogram after gamma correction
As mentioned above, you can do a gamma correction via ("Setting -> Base -> ImageProcessing -> LUTControl"). Here, the changes will affect the 8 bit image data and the processing needs the CPU of the host system:

![Figure 11: LUTControl dialog](image)

Just set "LUTEnable" to "On" and adapt the single LUTs like (LUT-0, LUT-1, etc.).

### 22.3.2.2 Step 2: White Balance

As you can see in the histogram, the colors red and blue are above green. Using green as a reference, we can optimize the white balance via "Setting -> Base -> Camera -> GenICam -> Analog Control -> Balance Ratio Selector" ("Balance White Auto" has to be "Off"):

1. Just select "Blue" and
2. adjust the "Balance Ratio" value until the blue line reaches the green one.
3. Repeat this for “Red”.

After optimizing white balance, the image will look like this:
22.3 Improving the acquisition / image quality

22.3.2.3 Step 3: Contrast

Still, black is more a darker gray. To optimize the contrast you can use "Setting -> Base -> Camera -> GenICam -> Analog Control -> Black Level Selector":

1. Select "DigitalAll" and

2. adjust the "Black Level" value until black seems to be black.
Figure 15: Back level adjustment
22.3 Improving the acquisition / image quality

22.3.2.4 Step 4: Saturation and Color Correction Matrix (CCM)

Still saturation is missing. To change this, the "Color Transformation Control" can be used ("Setting -> Base -> Camera -> GenICam -> Color Transformation Control"):

1. Click on "Color Transformation Enable" and

2. click on "Wizard" to start the saturation via "Color Transformation Control" wizard tool (since firmware version 1.4.57).
3. Now, you can adjust the saturation e.g. "1.1".

4. Afterwards, click on "Enable".
5. Since driver version 2.2.2, it is possible to set the special color correction matrices at
   (a) the input (sensor),
   (b) the output side (monitor) and
   (c) the saturation itself using this wizard.

6. Select the specific input and output matrix and

7. click on "Enable".

8. As you can see, the correction is done by the host by default ("Host Color Correction Controls"). However, you can decide, if the color correction is done by the device by clicking on "Write To Device And Switch Off Host Processing". The wizard will take the settings of the "Host Color Correction Controls" and will save it in the device.

9. Finally, click on "Apply".

After the saturation, the image will look like this:

![Figure 20: After adapting saturation](image)

![Figure 21: Corresponding histogram after adapting saturation](image)
As mentioned above, you can change the saturation and the color correction matrices via ("Setting -> Base -> ImageProcessing -> ColorTwist"). Here, the changes will affect the 8 bit image data and the processing needs the CPU of the host system:

![Figure 22: ColorTwist dialog](image1)

![Figure 23: Input and output color correction matrix](image2)
22.3 Improving the acquisition / image quality

22.3.3 Reducing noise by frame averaging

22.3.3.1 What is frame average?

As the name suggests, the functionality averages the gray values of each pixel using the information of subsequent frames. This can be used to

- reduce the noise in an image and
- compensate motion in an image.

MATRIX VISION implemented a dynamic version of the frame averaging in the FPGA, which not need any CPU of the host system. However, this mode is only available for the

- mvBlueCOUGAR-X2xx cameras with larger FPGAs.

22.3.3.2 Dynamic mode of mvBlueCOUGAR-X2xx

This mode uses an adaptive recursive filter with an average slope. The slope sets the amount of new image versus averaged image in relation to the gray scale variation of the pixel. With it, static noise can be removed at full bit depth and full frame rate:

Figure 2: Frame average: Functional principle

This method is well known and is used in the same or a similar way in all flat screen televisions. The amount of de-noising can be set with the slope factor: the smaller the value, the greater the feedback and therefore the de-noising but also the motion blur in the image:

\[
\text{Slope: } 256 = 100\% \text{ pixel\_difference} = 100\% \text{ signal\_in}
\]

Figure 3: Frame average: Slope

There are no delays with this option because de-noising is recursive and the Signal_{OUT} is extracted before the frame memory.
22.3.3.2.1 Using wxPropView

Using the dynamic frame average mode, you have to do the following step:

1. Start **wxPropView** (p. [115](#)) and
2. connect to the camera.
3. Then specify in "Setting -> Base -> Camera -> GenICam -> Device Control" which processing unit of the camera should do the frame averaging, e.g. unit 0 should do the frame averaging
   "mv Device Processing Unit Selector = 0"
   "mv Device Processing Unit = mvFrameAverage".
   Afterwards, "mv Frame Average Control" is available.
4. Now open "Setting -> Base -> Camera -> GenICam -> mv Frame Average Control" and
5. set the slope, e.g. 5000: "mv Frame Average Slope = 5000".
6. Activate frame averaging by setting "mv Frame Average Enable = 1".

![Figure 4: wxPropView: Setting the dynamic frame average](image)

For "**static images**", setting average slope to small numbers (10 .. 1000) gives the best noise enhancement at the expense of motion blur.

![Figure 5: wxPropView - noisy image](image)

For "**dynamic images**", setting average slope to higher values (1000 .. 5000) reduces motion blur at the expense of noise enhancement. This is especially needed in video application to enhance noise.

![Figure 6: wxPropView - reduced noise in image](image)
22.3 Improving the acquisition / image quality

22.3.4 Optimizing the bandwidth

22.3.4.1 Calculating the needed bandwidth

In some applications, for example, when multiple cameras share a single network path, you have to keep the bandwidth in mind.

Note

Within a GigE network you have a bandwidth of 125 MByte/s; a 100 MBit network has 12.5 MByte/s. The result of this formula is a rough guideline only. Some additional bandwidth is needed by the communication protocol and some other non GigE Vision related network traffic. Apart from that not every network controller can cope with a full 1 GBit/s stream of data, thus “real” results may vary.

22.3.4.2 Limiting the bandwidth of the imaging device

It is possible to limit the used bandwidth like the following way:

Since mvIMPACT Acquire 2.25.0

1. In "Setting -> Base -> Camera -> GenICam -> Device Control -> Device Link Selector" set property "Device Link Throughput Limit Mode" to "On".

2. Now, you can set the bandwidth with "Device Link Throughput Limit" to your desired bandwidth in bits per second

![Figure 1: wxPropView - Setting Device Link Throughput Limit](image-url)
Since

mvIMPACT Acquire < 2.25.0

1. In "Setting -> Base -> Camera -> GenICam -> Transport Layer Control -> Gev Stream Channel Selector" set property "mv Gev SCBW Control" to "mvGevSCBW".

![Figure 2: wxPropView - Setting mvGevSCBW to mvGevSCBWControl](image)

2. Now, you can set the bandwidth with "mvGevSCBW". E.g. 10000 for 10 MB. According to the image size and acquisition settings, the frame rate will be adjusted.

![Figure 3: wxPropView - Setting bandwidth size](image)

In contrast to this smart bandwidth control mechanism of mvBlueCOUGAR-X cameras, with other cameras you have to know and optimize the Inter-Packet Delay of the camera to avoid congestion in the switch (the loss of packages is an indicator of congestion). You can get the Inter-Packet Delay with following calculator:
22.3 Improving the acquisition / image quality

22.3.5 Setting a flicker-free auto expose and auto gain

22.3.5.1 Introduction

In order to prevent oscillations it is important to adapt the camera frequency to the frequency of AC light.

This is, for example, in

- Europe 50 cycles (100 fluctuations/s) whereas in

- USA, Japan and other countries it is 60 Hz.

This means the camera must strictly be coupled to this frequency. In conjunction with auto exposure this can only be maintained by using a timer based generation of external trigger pulses. This is a behavior of both sensor types: CCD and CMOS.

Note

It is not enough to use "Setting -> Base -> Camera -> GenICam -> Acquisition Control -> Acquisition Frame Rate" for this, as there are small fluctuations in this frame rate if the exposure time changes. These fluctuations lead to oscillations (see settings marked with red boxes in Figure 1). The "Acquisition Frame Rate" will only provide the exact frame rate if auto exposure is turned off.

As shown in Figure 1, it is possible to set ("mv Exposure Auto Mode") which part of the camera handles the auto expose (device or the sensor itself; pink boxes). Using "mvSensor" as "mv Exposure Auto Mode", it is possible to avoid oscillations in some cases. The reason for this behavior is that you can set more parameters like "mv Exposure Auto Delay Images" in contrast to "mvDevice". However, as mentioned above it is recommended to use a timer based trigger when using auto expose together with continuous acquisition.
Figure 1: wxPropView - Auto expose is turned on and the frame rate is set to 25 fps

22.3.5.2 Example of using a timer for external trigger

Figure 2 shows how to generate a 25 Hz signal, which triggers the camera:

- "Setting -> Base -> Camera -> GenICam -> Counter & Timer Control -> Timer Selector -> Timer 1"
– “Timer Trigger Source” = “Timer1End”
– “Timer Duration” = “40000”

\[
\text{FPS}_{\text{max}} = \frac{1}{40000 \text{ us}} = 25
\]

– “Setting -> Base -> Camera -> GenICam -> Acquisition Control -> Trigger Selector -> FrameStart”:
  – “Trigger Mode” = “On”
  – “Trigger Source” = “Timer1End”

Figure 2: wxPropView - 25 Hz timer for external trigger
No oscillation occurs, regardless of DC ambient vs. AC indoor light. This operation mode is known as flicker-free or flicker-less operation.

What it mainly does is to adjust the frame frequency to precisely the frequency of the power line. Usually the line frequency is very stable and therefore is the harmonic frequency difference of the two signals are very slow; probably in the range of $< 0.1 \text{ Hz}$.

The fact that we do not know the phase relation between the two frequencies means that we scan the alternating ambient light source with our camera. The shorter the exposure time, the more we see a slow change in brightness. Using AutoExposure/AutoGain can completely eliminate this change because the frequency of change is very low. That means it will be legal if we calculate a brightness difference in one picture and apply it for the next one, because the change is also valid in the next one; as we fulfill the Nyquist theorem.

If we use an arbitrary scanning frequency like 20 fps or whatever your algorithm and data flow is accepting, is wrong in this aspect and leads to oscillations and undesired flicker.

Pointing to a 60 Hz display with flashing backlight an oscillation of 10 Hz can be seen of course.

Figure 3: wxPropView - Intensity plot while pointing the camera to a 60 Hz display

22.3.5.3 Conclusion

To avoid oscillations, it is necessary to adapt the camera frequency to the frequency of AC light. When using auto expose a flicker-free mode (timer based external trigger) is needed. If the camera is used throughout the world it is necessary that the frequency of AC light can be set in the software and the software adapts the camera to this specific environment.

22.3.6 Working with binning

With binning it is possible to combine adjacent pixels vertically and/or horizontally. Depending on the sensor, up to 16 adjacent pixels can be combined.

See also

https://www.matrix-vision.com/manuals/SDK_CPF/Binning_modes.png

Binning will lighten the image at the expense of the resolution. This is a neat solution for applications with low light and low noise.

The following results were achieved with the mvBlueFOX3-2124G, however, binning is also available with the mvBlueCOUGAR-X camera family.
### 22.3 Improving the acquisition / image quality

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2500</td>
<td>-</td>
<td>0</td>
<td>-</td>
<td><img src="image1.jpg" alt="Image" /></td>
</tr>
<tr>
<td>2500</td>
<td>-</td>
<td>30</td>
<td>-</td>
<td><img src="image2.jpg" alt="Image" /></td>
</tr>
<tr>
<td>2500</td>
<td>2H 2V</td>
<td>30</td>
<td>-</td>
<td><img src="image3.jpg" alt="Image" /></td>
</tr>
</tbody>
</table>

**Exposure [in us]**
- 2500 us

**Binning**
- None

**Gain [in dB]**
- 0 dB
- 30 dB
- 2H 2V

**Averager**
- None
The last image shows, that you can reduce the noise, caused by the increased gain, using frame averaging (p. 313).

22.3.7 Working with Sony’s 4 Tap CCD sensors

To achieve high frame rates, Sony’s high-resolution sensors offer a built-in four channel output a.k.a Taps. I.e., the sensor area is divided into 4 parts in 4-Tap mode and 2 parts in 2-Tap mode:

Each Tap has its own gain control and for this reason each image area could behave differently. The following image shows an uncalibrated image where you can see the four different Tap areas of the sensor:
Therefore it is necessary to calibrate the Taps to get a good image. We can use wxPropView (p. 115) to calibrate the camera.

**Note**

Before calibrating, you should fix the acquisition parameters like number of Taps, device clock frequency, pixel format and exposure time.

1. Open **wxPropView** (p. 115) and
2. open the **Analog Control** (p. 174) ("Setting -> Base -> Camera -> GenICam -> Analog Control").
3. Now, you can adapt the **Gain** of each Tap ("AnalogTap1", "AnalogTap2", "AnalogTap3", "AnalogTap4"): 
Figure 3: wxPropView - Setting the gain of each Tap area (you will find the exposure time under acquisition control)

In our example the calibrated image would look like this:
22.3 Improving the acquisition / image quality

Figure 4: wxPropView - Calibrated Taps

Note

Using 2 Taps, the adapting "AnalogTap3" and "AnalogTap4" will have no effect.

The mvBlueCOUGAR-XD sensor range contains three 4-Taps sensors by Sony:

- mvBlueCOUGAR-XD124a (3 Mpix [1936 x 1460]) (p. 484)
- mvBlueCOUGAR-XD126 (6 Mpix [2752 x 2208]) (p. 513)
- mvBlueCOUGAR-XD129 (9.2 Mpix [3384 x 2712]) (p. 533)

22.3.8 Minimizing sensor pattern of mvBlueCOUGAR-X1010G

Sometimes the gray scale version of Aptinas sensor MT9J003 shows structures comparable with Bayer patterns of color sensors. This pattern is particularly apparent in scaled images:
To minimized this pattern, you can balance the sensor patterns since Firmware version 2.3.70.0. This procedure works like the white balancing used with color sensors. For this reason the same terms are used (red, blue, green).

The balance reference is the "green" pixel value from the "blue-green" line of the sensor.

See: Output sequence of color sensors (RGB Bayer) (p.110)

I.e. all gray scale values of the these "green" pixels are averaged.

With "Setting -> Base -> Camera -> GenICam -> Analog Control -> Balance Ration Selector" you can select each color to set the "Balance Ratio":

- "Red" averages the "red" pixel values from the "red-green" line of the sensor.
- "Green" averages the "green" pixel values from the "red-green" line of the sensor, too.
- "Blue" averages the "blue" pixel values from the "blue-green" line of the sensor.

I.e. there are 4 average values (reference value, red value, green value, blue value). The lowest value will be unchanged, the other values are increased using each "Balance Ratio".

However by using the property "Balance White Auto" you can balance the sensor automatically:
22.3 Improving the acquisition / image quality

Figure 2: Balance White Auto

After balancing, we recommend to save these settings to a UserSet (p. 148).
22.3.9 Working with the dual exposure feature ("mvMultiZone") of mvBlueCOUGAR-XD107

22.3.9.1 Introduction

The IMX420 used in the mvBlueCOUGAR-XD107 is a Pregius sensor of the third generation. This sensor features a dual exposure mode, i.e. after a trigger event, for example, different image areas can be exposed to light at the same time differently.

To activate the dual exposure mode we introduced a new exposure mode called mvMultiZone.

If this mode is selected you can set the "mv Exposure Horizontal Zone Divider". This property indicates where the zones are divided horizontally. The step size is 12.5%.

The exposure time of the different zones can be specified using the "Exposure Time Selector". Possible zones are

- mvHorizontalZone0
- mvHorizontalZone1

Figure 3: Calibrated sensor
22.3 Improving the acquisition / image quality

22.3.9.2 Setting the dual exposure using wxPropView

To activate the dual exposure, just

1. Select **mvMultiZone** in "Setting -> Base -> Camera -> GenICam -> Acquisition Control -> Exposure Mode".

2. Set the "Exposure Time" of each zone by selecting the specific "Exposure Time Selector".

3. Finally, specify the "mv Exposure Horizontal Zone Divider" (e.g. 50, i.e. you have two equal image halves).

![Figure 1: wxPropView - new Exposure Mode](image-url)
22.3.9.2.1 Programming the dual exposure

```cpp
#include <mvIMPACT_CPP/mvIMPACT_acquire_GenICam.h>

// more code
GenICam::AcquisitionControl ac( pDev );
ac.exposureMode.writeS( "mvMultiZone" );
ac.exposureTimeSelector.writeS( "mvHorizontalZone1" );
ac.exposureTime.write( 5000 );
ac.exposureHorizontalDivider.write( 50 );
// more code
```

22.3.10 Working with the dual gain feature of mvBlueCOUGAR-XD107/XD1031 and mvBlueCOUGAR-X102m/X107

22.3.10.1 Introduction

The IMX420/425/428 used in the mvBlueCOUGAR-XD107/XD1031 and mvBlueCOUGAR-X102m/X107 are Pregius sensors of the third generation.

Those sensors feature a dual gain mode, i.e. after a trigger event, for example, different image areas can be amplified at the same time differently.

To activate the dual gain mode it is necessary to enable the multi area mode by using the `mvMultiAreaMode` property. At least two different AOIs have to be defined.
22.3 Improving the acquisition / image quality

Note
Both AOIs must not overlap or touch each other!

Once the AOIs are configured the GainSelector has to be set to one of the new implemented options called \texttt{mvHorizontalZone0} and \texttt{mvHorizontalZone1}.

The gain value of the different zones can be specified using the \texttt{"Gain Selector"}. And the corresponding gain property. Possible zones are

\begin{itemize}
  \item \texttt{mvHorizontalZone0}
  \item \texttt{mvHorizontalZone1}
\end{itemize}

If this mode is selected you can set the \texttt{"mv Gain Horizontal Zone Divider"}. This property indicates where the zones are divided horizontally once more than two AOIs are configured. In this case the e.g. 25\% means that the upper 25\% of the image are defined by the gain value of \texttt{mvHorizontalZone0} and the lower 75\% are defined by the gain value of \texttt{mvHorizontalZone1}.

Note
Some sensors may only allow to change the gain at certain positions e.g. the last line of a defined ROI. In this case the first possible switching point above the actual line will be used.

22.3.10.2 Setting the dual gain using \texttt{wxPropView}

To activate the dual gain, just

1. Use the Multi AOI Wizard to adjust the different AOIs. (They must not overlap or touch each other!)
2. Select \texttt{mvMultiZone} in \texttt{"Setting -> Base -> Camera -> GenICam -> Analog Control -> mvGainMode"}.
3. Select \texttt{mvHorizontalZone0} in \texttt{"Setting -> Base -> Camera -> GenICam -> Analog Control -> GainSelector"}.
4. Adjust the gain value for the first AOI in \texttt{"Setting -> Base -> Camera -> GenICam -> Analog Control -> GainSelector -> Gain"}.
5. Adjust the gain divider position \texttt{"Setting -> Base -> Camera -> GenICam -> Analog Control -> GainSelector -> mvGainHorizontalZoneDivider"}.
6. Select \texttt{mvHorizontalZone1} in \texttt{"Setting -> Base -> Camera -> GenICam -> Analog Control -> GainSelector"}.
7. Adjust the gain value for the second AOI in \texttt{"Setting -> Base -> Camera -> GenICam -> Analog Control -> GainSelector -> Gain"}. 

MATRIX VISION GmbH
Figure 1: wxPropView - configuring multiple AOIs

Figure 2: wxPropView - configuring the gain of the first zone
22.3 Improving the acquisition / image quality

22.3.10.3 Programming the dual gain mode

As an example the IMX425 sensor is used for the sample. The goal is to configure three AOIs which have a similar height. Since the AOIs must not overlap or touch each other, it is important to increase the offset of the next AOI by the smallest increment size. Which is 8 in this case.

```c++
#include <mvIMPACT_CPP/mvIMPACT_acquire_GenICam.h>

GenICam::ImageFormatControl ifc( pDev );

ifc.mvMultiAreaMode.writeS( "mvMultiAreasCombined" );
ifc.mvAreaSelector.writeS( "mvArea0" );
ifc.mvAreaOffsetX.write( 0 );
ifc.mvAreaOffsetY.write( 0 );
ifc.mvAreaWidth.write( ifc.mvAreaWidth.getMaxValue() );
ifc.mvAreaHeight.write( 360 );
ifc.mvAreaEnable.writeS( "1" );

ifc.mvAreaSelector.writeS( "mvArea1" );
ifc.mvAreaOffsetX.write( 0 );
ifc.mvAreaOffsetY.write( 368 );
ifc.mvAreaWidth.write( ifc.mvAreaWidth.getMaxValue() );
ifc.mvAreaHeight.write( 360 );
ifc.mvAreaEnable.writeS( "1" );

ifc.mvAreaSelector.writeS( "mvArea2" );
ifc.mvAreaOffsetX.write( 0 );
```

Figure 3: Example image with two different amplified zones

MATRIX VISION GmbH
22.3.11 Working with the dual ADC feature of mvBlueCOUGAR-XD107

22.3.11.1 Introduction

The IMX420 used in the mvBlueCOUGAR-XD107C/G is a Pregius sensor of the third generation. This sensor features a dual ADC mode which allows to increase the dynamic range of the sensor.

To activate the dual ADC mode it is sufficient to use the "mv Dual ADC Mode" property.

22.3.11.2 Enabling the dual ADC mode using wxPropView

To activate the dual ADC mode for a color model, just

1. Change the pixel format to "BayerRG16" "Setting -> Base -> Camera -> GenICam -> Image Format Control -> PixelFormat".

2. Enable "mvDualAdcMode" "Setting -> Base -> Camera -> GenICam -> Image Format Control -> mvAdcMode".

To activate the dual ADC mode for a monochrome model, just

1. Change the pixel format to "Mono16" "Setting -> Base -> Camera -> GenICam -> Image Format Control -> PixelFormat".

2. Enable "mvDualAdcMode" "Setting -> Base -> Camera -> GenICam -> Image Format Control -> mvAdcMode".
Note

Changing the pixel format to a pixel format which utilizes more than one byte per pixel (e.g. BayerRG16 or Mono16) is necessary to present the higher dynamic range in the image. The maximum frame rate will be halved because of the multi byte pixel format. Please be aware that in case of histogram analysis this mode might not be the best option since the transition between both ADCs might show a slight gain offset error.

![ImageFormatControl](image)

Figure 1: wxPropView - enabling the dual ADC mode

### 22.3.11.2.1 Programming the dual ADC mode

```cpp
#include <mvIMPACT_CPP/mvIMPACT_acquire_GenICam.h>

// more code
GenICam::ImageFormatControl ifc( pDev );
ifc.pixelFormat.writeString("BayerRG16");
ifc.mvDualAdcMode.writeS( "1" );
// more code
```

### 22.4 Working with triggers

There are several use cases concerning trigger:
22.4.1 Processing triggers from an incremental encoder

22.4.1.1 Basics

The following figure shows the principle of an incremental encoder:

![Incremental encoder principle](image)

This incremental encoder will send an A, B, and Z pulse. With these pulses there are several ways to synchronize an image with an incremental encoder.

22.4.1.2 Using Encoder Control

To create an external trigger event by an incremental encoder, please follow these steps:

1. Connect the incremental encoder output signal A, for example, to the digital input 0 ("Line4") of the mvBlueCOUGAR-X.
   This line counts the forward pulses of the incremental encoder.

2. Depending on the signal quality, it could be necessary to set a debouncing filter at the input (p.355) (red box in Figure 3):
   Adapt in "Setting -> Base -> Camera -> GenICam -> Digital I/O Control" the "Line Selector" to "Line4" and set "mv Line Debounce Time Falling Edge" and "mv Line Debounce Time Rising Edge" according to your needs.

3. Set the trigger "Setting -> Base -> Camera -> GenICam -> Acquisition Control -> Trigger Selector" "FrameStart" to the "Encoder0" ("Trigger Source") signal.

4. Then set "Setting -> Base -> Camera -> GenICam -> Encoder Control -> Encoder Selector" to "Encoder0" and

5. adapt the parameters to your needs.
22.4 Working with triggers

See also Encoder Control (p. 182)

Figure 2: wxPropView settings

Note

The max. possible frequency is 5 KHz.

22.4.1.2.1 Programming the Encoder Control

```
#include <mvIMPACT_CPP/mvIMPACT_acquire_GenICam.h>

// more code
GenICam::EncoderControl ec( pDev );
ec.encoderSelector.writeS( "Encoder0" );
ec.encoderSourceA.writeS( "Line4" );
ec.encoderMode.writeS( "FourPhase" );
ec.encoderOutputMode.writeS( "PositionUp" );
// more code
```
22.4.1.3 Using Counter

It is also possible to use **Counter** and **CounterEnd** as the trigger event for synchronizing images with an incremental encoder.

To create an external trigger event by an incremental encoder, please follow these steps:

1. Connect the incremental encoder output signal A, for example, to the digital input 0 (**Line4**) of the BlueCOUGAR-X. This line counts the forward pulses of the incremental encoder.
2. Set "Setting -> Base -> Camera -> GenICam -> Counter and Timer Control -> Counter Selector" to "Counter1" and
3. "Counter Event Source" to "Line4" to count the number of pulses e.g. as per revolution (e.g. "Counter Duration" to 3600).
4. Then set the trigger "Setting -> Base -> Camera -> GenICam -> Acquisition Control -> Trigger Selector" "FrameStart" to the "Counter1End" ("Trigger Source") signal.

![Figure 3: wxPropView setting](image-url)

To reset "**Counter1**" at Zero degrees, you can connect the digital input 1 (**Line5**) to the encoder Z signal.
22.4 Working with triggers

22.4.2 Generating a pulse width modulation (PWM)

22.4.2.1 Basics

To dim a laser line generator, for example, you have to generate a pulse width modulation (PWM).

For this, you will need

- 2 timers and
- the active signal of the second timer at an output line

22.4.2.2 Programming the pulse width modulation

You will need two timers and you have to set a trigger.

- **Timer1** defines the interval between two triggers.
- **Timer2** generates the trigger pulse at the end of **Timer1**.

The following sample shows a trigger

- which is generated every second and
- the pulse width is 10 ms:

```c
#include <mvIMPACT_CPP/mvIMPACT_acquire.h>
#include <mvIMPACT_CPP/mvIMPACT_acquire_GenICam.h>
...

// Master: Set timers to trig image: Start after queue is filled
GenICam::CounterAndTimerControl catcMaster(pDev);
catcMaster.timerSelector.writeS( "Timer1" );
catcMaster.timerDelay.write( 0. );
catcMaster.timerDuration.write( 1000000. );
catcMaster.timerTriggerSource.writeS( "Timer1End" );

catcMaster.timerSelector.writeS( "Timer2" );
catcMaster.timerDelay.write( 0. );
catcMaster.timerDuration.write( 10000. );
catcMaster.timerTriggerSource.writeS( "Timer1End" );
```

See also

*Counter And Timer Control* (p. 173)
Note

Make sure the *Timer1* interval must be larger than the processing time. Otherwise, the images are lost.

Now, the two timers will work like the following figure illustrates, which means

- *Timer1* is the trigger event and
- *Timer2* the trigger pulse width:

```
2 Timers sample: Timer 2 and Timer 1 start at the end of Timer1

<table>
<thead>
<tr>
<th>Timer 1</th>
<th>Duration 1s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timer 2</td>
<td>Duration 10ns</td>
</tr>
</tbody>
</table>
```

*Figure 1: Timers*

The timers are defined, now you have to set the digital output, e.g. "Line 0":

```c
// Set Digital I/O
GenICam::DigitalIOControl io(pDev);
io.lineSelector.writeS( "Line0" );
io.lineSource.writeS( "Timer2Active" );
```

See also

**Digital I/O Control** *(p. 181)*

This signal has to be connected with the digital inputs of the application.

### 22.4.2.3 Programming the pulse width modulation with wxPropView

The following figures show, how you can set the timers using the GUI tool *wxPropView* *(p. 115)*

1. Setting of *Timer1* (blue box) on the master camera:
22.4 Working with triggers

22.4.3 Outputting a pulse at every other external trigger

To do this, please follow these steps:

1. Setting of Timer1 (purple on the master camera):

![Figure 2: wxPropView - Setting of Timer1](image)

2. Setting of Timer2 (purple on the master camera):

![Figure 3: wxPropView - Setting of Timer2](image)

3. Assigning timer to DigOut (orange box in Figure 2).

---

MATRIX VISION GmbH
1. Switch "Trigger Mode" to "On" and
2. Select the "Trigger Source", e.g. "Line5".
3. Use "Counter1" and count the number of input trigger by setting the "Counter Duration" to "2".
4. Afterwards, start "Timer1" at the end of "Counter1":

![Image of wxPropView settings]

Figure 1: wxPropView - Setting the sample

The "Timer1" appears every second image.
Now, you can assign "Timer1Active" to a digital output e.g. "Line3":

![Figure 2: Assigning the digital output](image)

**Note**

You can delay the pulse if needed.

### 22.4.4 Creating different exposure times for consecutive images

If you want to create a sequence of exposure times, you have to trigger the camera "externally" via pulse width:

1. Use **Timer** and **Counter** to build a sequence of different pulse widths.
2. Use the **Counter** for the time between the exposures (with respect to the readout times).
3. Afterwards, use an **AND** gate followed by **OR** gate to combine different exposure times.

**Note**

Please be sure that the sensor can output the complete image during **Counter1** or **Counter2**. Otherwise, only one integration time will be used.

**Sequence of different exposure times:**

![Figure 1: wxPropView - Logic gate principle](image)
You can set this sample in `wxPropView` (p. 115). E.g. the sensor makes 22.7 frames per second in **Continuous Mode** (p. 170). This means that the sensor needs 44 ms to output the complete image.

\[
\frac{1}{22.7} \approx \text{approx. } 44 \text{ ms}
\]

We take 55 ms to be sure. Now, as different exposure times we take 1 ms (`Timer1`) and 5 ms (`Timer2`). To get the 55 ms, we have to add 54000 us (`Counter1`) and 50000 us (`Counter2`).

Finally, you have to set the logic gate as shown in the figure:

![Figure 2: wxPropView - Logic gate setting](image)

**Note**

Because there are 4 counters and 2 timers you can only add one further exposure time using one counter as a timer.

So if you want other sequences, you have to use the counters and timers in a flexible way as show in the next sample:
22.4.4.1 Sequence with 4 times exposure A followed by 1 time exposure B

If you have an external trigger, you can use the counter and timer to create longer exposure sequences.

For example, if you want a sequence with 4 times exposure A followed by 1 time exposure B you can count the trigger events. That means practically:

1. Use **Counter1** to count 5 trigger signals then
2. issuing **Timer2** for the long exposure time (re-triggered by **Counter1End**).
3. Every trigger issues **Timer1** for the short exposure.
4. Afterwards, an **AND** gate followed by **OR** gate combines the different exposure times.

In **wxPropView** (p. 115) it will look like this:

![Figure 3: wxPropView - Logic gate setting 2](image-url)
22.4.5 Detecting overtriggering

22.4.5.1 Scenario

The image acquisition of a camera consists of two steps:

- exposure of the sensor and
- readout of the sensor data

During these steps, a trigger signal will be skipped:

![Trigger counter increases but the start exposure counter not]

To notice overtriggering, you can use counters (p. 173):

- One counter counts the incoming trigger signals, the
- second counter counts the ExposureStart signals.

Using the chunk data (p. 179) you can overlay the counters in the live image.

22.4.5.2 Setting the overtrigger detector using wxPropView

First of all, we have to set the trigger in "Setting -> Base -> Camera -> GenICam -> Acquisition Control" with following settings:

<table>
<thead>
<tr>
<th>Property name wxPropView</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trigger Selector</td>
<td>FrameStart</td>
</tr>
<tr>
<td>Trigger Mode</td>
<td>On</td>
</tr>
<tr>
<td>Trigger Source</td>
<td>Line4</td>
</tr>
<tr>
<td>Trigger Activation</td>
<td>RisingEdge</td>
</tr>
<tr>
<td>Exposure Mode</td>
<td>Timed</td>
</tr>
</tbody>
</table>
This trigger will start an acquisition after a rising edge signal on line 4 (= DigIn0 (p. 69)).

Now, set the two counters. Both counters (Counter1 and Counter2) will be reset and start after the acquisition (AcquisitionStart) has started.

While Counter1 increases with every ExposureStart event (see figure above for the event and acquisition details) ...

... Counter2 increases with every RisingEdge of the trigger signal:
Now, you can check if the trigger signal is skipped (when a rising edge signal is active during readout) or not by comparing the two counters.

Enable the inclusion of the selected chunk data ("Chunk Mode Active = 1") in the payload of the image in "Setting -> Base -> Camera -> GenICam -> Chunk Data Control".

Activate the info overlay in the display area. Right-click on the live display and select: "Request Info Overlay"
22.4 Working with triggers

Figure 5: Show chunk data

The following figure shows that no trigger signal is skipped:

Figure 6: Trigger Signal counter equals ExposureStart counter

The following figure shows that the acquisition is overtriggered:
22.4.6 Triggering of an indefinite sequence with precise starting time

22.4.6.1 Scenario

Especially in the medical area, there are applications where a triggered acquisition is started, for example, with a foot switch. Following challenges have to be solved in combination with these applications:

- The user wants the acquired image immediately (precise starting time).
- It is not known, when the user stops the acquisition (indefinite sequence).

Using AcquisitionStart as the trigger source, it could take between 10 and 40 ms until the camera acquires the first frame. That's not really an immediately acquisition start. It is recommended to use FrameStart as the trigger source instead. However, according to the time the trigger event occurs, there will be a timeout during the first frame in nearly all cases.

You can avoid this using a timer which generates a "high" every 100 us and which is connected to the trigger input Line4 using a logical AND gate. I.e. if the timer is "high" and there is a trigger signal at Line4 then the logical conjunction is true. The AND gate result is then connected as TriggerSource of the FrameStart trigger using a logical OR gate. I.e. as soon as the logical AND conjunction is true, the trigger source is true and the image acquisition will start.

The following figure illustrates the settings:
22.4 Working with triggers

With this setting, there is still an acceptable time delay of approx. 100 to 130 us possible.

22.4.6.2 Creating the use case using wxPropView

First of all, we have to set the timer in "Setting -> Base -> Camera -> GenICam -> Counter And Timer Control" with following settings:

<table>
<thead>
<tr>
<th>Property name wxPropView</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timer Selector</td>
<td>Timer1</td>
</tr>
<tr>
<td>Timer Trigger Source</td>
<td>Timer1End</td>
</tr>
<tr>
<td>Timer Duration</td>
<td>100.000</td>
</tr>
</tbody>
</table>

Afterwards, we have to set the logical gates in "Setting -> Base -> Camera -> GenICam -> mv Logic Gate Control" with following settings:

<table>
<thead>
<tr>
<th>Property name wxPropView</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>mv Logic Gate AND Selector</td>
<td>mvLogicGateAND1</td>
</tr>
<tr>
<td>mv Logic Gate AND Source 1</td>
<td>Line4</td>
</tr>
<tr>
<td>mv Logic Gate AND Source 2</td>
<td>Time1Active</td>
</tr>
<tr>
<td>mv Logic Gate OR Selector</td>
<td>mvLogicGateOR1</td>
</tr>
<tr>
<td>mv Logic Gate OR Source 1</td>
<td>mvLogicGateAND1Output</td>
</tr>
</tbody>
</table>

Finally, we have to set the trigger in "Setting -> Base -> Camera -> GenICam -> Acquisition Control" with following settings:

<table>
<thead>
<tr>
<th>Property name wxPropView</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trigger Selector</td>
<td>FrameStart</td>
</tr>
<tr>
<td>Trigger Mode</td>
<td>On</td>
</tr>
<tr>
<td>Trigger Source</td>
<td>mvLogicGateAND1Output</td>
</tr>
<tr>
<td>Trigger Activation</td>
<td>RisingEdge</td>
</tr>
</tbody>
</table>
### 22.5 Working with I/Os

There are several use cases concerning I/Os:
22.5 Working with I/Os

22.5.1 Controlling strobe or flash at the outputs

Of course, the mvBlueCOUGAR-X supports strobe or flash lights. However, there are several things you have to keep in mind when using strobes or flash:

1. Be sure that the illumination fits with the movement of the device under test.
2. Bright illumination and careful control of exposure time are usually required.
3. To compensate blur in the image, short exposure times are needed.

Alternatively, you can use flash with short burn times. For this, you can control the flash using the camera. The following figures show, how you can do this using **wxPropView** (p. 115)

1. Select in "Setting -> Base -> Camera -> Digital I/O Control" the output line with the "Line Selector" to which the strobe or flash is connected.
2. Now, set the "Line Source" to "mvExposureAndAcquisitionActive". This means that the signal will be high for the exposure time and only while acquisition of the camera is active.

![Figure 1: Setting the "Line Source" to "mvExposureAndAcquisitionActive"](image)

**Note**

This can be combined using an external trigger.
22.5.2 Compensating delay of strobe or flash

Normally, the input circuitry of flash has a delay (e.g. low pass filtering). Using "ExposureActive" to fire strobe would actually illuminate delayed with respect to exposure of the sensor. Figure 2 shows the problem:

![Figure 2: Flash delay with "ExposureActive"](image)

To solve this issue, you can use following procedure:

1. Do not use "ExposureActive" for triggering strobe.
2. Build flash signal with Timer,
3. trigger Timer with external trigger (e.g. "Line5").
4. Use "Trigger Delay" to delay exposure of the sensor accordingly.

In wxPropView (p. 115) it will look like this:

![Figure 3: Working with Timer and "Trigger Delay"](image)
22.5.3 Creating a debouncing filter at the inputs

In some cases, it is necessary to eliminate noise on trigger lines. This can become necessary when either

- the edges of a trigger signal are not perfect in terms of slope or
- if, because of the nature of the trigger signal source, multiple trigger edges are generated within a very short period of time even if there has just been a single trigger event.

The latter one is also called bouncing.

Bouncing is the tendency of any two metal contacts in an electronic device to generate multiple signals as the contacts close or open; now debouncing is any kind of hardware device or software that ensures that only a single signal will be acted upon for a single opening or closing of a contact.

To address problems that can arise from these kinds of trigger signals MATRIX VISION offers debouncing filters at the digital inputs of a device.

The debouncing filters can be found under "Setting -> Base -> Camera -> GenICam -> Digital I/O Control -> LineSelector" (red box in Figure 1) for each digital input:
Each digital input (LineMode equals Input) that can be selected via the LineSelector property will offer its own property to configure the debouncing time for falling edge trigger signals ("mv Line Debounce Time Falling Edge") and rising edge ("mv Line Debounce Time Rising Edge") trigger signals.

The line debounce time can be configured in micro-seconds with a maximum value of up to 5000 micro-seconds. Internally each time an edge is detected at the corresponding digital input a timer will start (orange ∗ in figure 2 and 3), that is reset whenever the signal applied to the input falls below the threshold again. Only if the signal stays at a constant level for a full period of the defined \( mvLineDebounceTime \) the input signal will be considered as a valid trigger signal.
22.5 Working with I/Os

Note

Of course this mechanism delays the image acquisition by the debounce time.

Figure 2: mvLineDebounceTimeRisingEdge Behaviour

Figure 3: mvLineDebounceTimeFallingEdge Behaviour

22.5.4 Using motorized lenses with mvBlueCOUGAR-XD

22.5.4.1 Introduction

In contrast to machine vision applications with constant lightings and fixed focal lengths, outdoor applications such as traffic monitoring, security, or sports demand to control image brightness, field of view, zoom, focus, or iris. Lenses with motors offer the possibility to remotely manipulate these settings and the mvBlueCOUGAR-XD offers the possibility to control motorized lenses.
22.5.4.2 Types of motorized lenses and their controls

Motorized lenses differ by the fact what element is motorized and how:

- Zoom
- Focus
- Iris
  - Motor
  - Video
  - DC

Lenses with motors differ by the voltage they accept and by certain wiring specialties. Driving voltages may be between 3 and 12V DC, wiring may be 2 wires per motor (a.k.a. bipolar) or with one wire per motor and common ground. Some lenses offer potentiometers so that the actual position can be measured by a resistance. These potentiometers are not supported by the mvBlueCOUGAR-XD camera.

22.5.4.2.1 List of usable lenses

Motorized lenses differ by the max. sensor diameter they support and by resolution limits. mvBlueCOUGAR-XD cameras use sensors with 2/3” to 1” diameter. Lens/camera combinations must be selected having these properties in mind. The following is a list of usable lenses. It is provided for reference only. Exclusion from this list does not mean that the product is not usable with the camera per se.

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Details</th>
<th>Motorized Iris</th>
<th>Motorized Focus</th>
<th>Motorized Zoom</th>
<th>Video Iris</th>
</tr>
</thead>
<tbody>
<tr>
<td>KOWA</td>
<td>Motorized LMZ-series up to 1” and 5 MPix resolution</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Goyo Optical</td>
<td>GAZ series 2/3” – 1”</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Linos</td>
<td>Mevis motorized</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Schneider Optics</td>
<td>Cinegon/Xenoplan: motorized iris</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Computar</td>
<td>2/3” M6Z series</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fujinon</td>
<td>2/3” and 1” lenses</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

22.5.4.3 Connecting the motorized lens to a camera

22.5.4.3.1 Connecting the direct drive lens motors

mvBlueCOUGAR-XD offers two connectors at the back. Use the female one on the right side for lens connection. Pinning is shown in the table on the left side below:
Figure 1: mvBlueCOUGAR-XD connectors

<table>
<thead>
<tr>
<th>Female connector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pin.</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>11</td>
</tr>
<tr>
<td>12</td>
</tr>
</tbody>
</table>

Figure 2: Wiring of full motorized zoom lens

Image above shows a typical lens wiring. The three independent motors can be seen.

- Connect Pin4 (Focus+) to CN Pin4.
- Connect Pin5 (Focus-) to CN Pin3.
- etc.

mvBlueCOUGAR-XD can deliver up to 100mA current with a selectable voltage to the outputs Focus, Zoom and Iris. Please note that the voltage applied is independent of the supply voltage of the camera. Channel 4 can be left open.
22.5.4.3.2 Connecting the video iris

mvBlueCOUGAR-XD camera generates a video like signal containing average brightness information and sync signals to drive a video iris type lens. The diaphragm of the lens closes at increasing brightness keeping the resulting overall brightness reaching the sensor constant.

Advantage of video iris against AutoExposure: Bigger range of brightness variation avoids smear in CCD as it blocks extreme bright light to hit the sensor; but is slower than AutoExposure and AutoGain which is also supported by the camera. Pining of the standardized video iris connector (4 pin EIAJ) is shown below.

Figure 3: Pining video iris

Use the square 4 pin iris connector of the camera to directly connect the video iris.

22.5.4.4 Controlling the lens via viewer or the API (aka mvIMPACT Acquire)

Usage of the lens control wizard of wxPropView (p. 115) is recommended for setup:

Figure 4: Lens control wizard

Select the "Drive Level" voltage according to the lens type. Focus, Zoom and Iris buttons drive the motors at a selectable speed.

"Video Iris" can be selected to open or completely close the Iris (for setup) and for auto mode.

Note

Additional settings such as level (sensitivity) and/or ALC (peak or average) may be possible directly at the lens (via poti).
ALC settings do not have effect due to digital video signal!
Consult the manual of the lens for more details.

Using AGC/AEC & mvIrisAuto may lead to oscillating brightness.
22.5.4.5 Setting the video iris (example)

Purpose is to bring the video iris into a usable range so that during operation it can open if brightness goes down and further close if brightness goes up.

1. Open iris by using `mvIrisOpen` command: This opens iris to the min. F number (f/N) supported by the lens, e.g. f/1.2 (see lens manual).
2. Set exposure of camera so that the image is not saturated.
3. Set exp = 4 x min.
4. Iris auto will move the lens to f/2.4.
5. Set working exposure = 16 X min will move the Iris auto to f/4.8.

22.5.4.6 Controlling the lens via 3rd party libraries or APIs

The properties for "mv Lens Control" are MATRIX VISION specific but appear in the camera’s XML-file according to GigE Vision (p. 236) standards and SFNC (p. 242) thanks to the standard. This makes it possible to use the features from third party applications or programs without problems.

![Figure 5: wxPropView - mv Lens Control](image)

The screenshot below shows how the properties appear under MVTec HALCON's image acquisition assistant:

![Figure 6: HALCON's image acquisition assistant](image)
The next screenshot shows the respective HDevelop example under Halcon for the same settings:

![HDevelop example](image)

**Figure 7: HDevelop example**

22.6 Working with HDR (High Dynamic Range Control)

There are several use cases concerning High Dynamic Range Control:
22.6.1 Adjusting sensor -x00w

22.6.1.1 Introduction

The HDR (High Dynamic Range) mode of the sensor -x00w increases the usable contrast range. This is achieved by dividing the integration time in two or three phases. The exposure time proportion of the three phases can be set independently. Furthermore, it can be set, how much signal of each phase is charged.

22.6.1.2 Functionality

![Diagram of the -x00w sensor's HDR mode](image)

Figure 1: Diagram of the -x00w sensor's HDR mode

22.6.1.2.1 Description

- **"Phase 0"**
  - During T1 all pixels are integrated until they reach the defined signal level of Knee Point 1.
  - If one pixel reaches the level, the integration will be stopped.
  - During T1 no pixel can reached a level higher than P1.

- **"Phase 1"**
  - During T2 all pixels are integrated until they reach the defined signal level of Knee Point 2.
  - T2 is always smaller than T1 so that the percentage compared to the total exposure time is lower.
  - Thus, the signal increase during T2 is lower as during T1.
  - The max. signal level of Knee Point 2 is higher than of Knee Point 1.

- **"Phase 2"**
  - During T2 all pixels are integrated until the possible saturation.
  - T3 is always smaller than T2, so that the percentage compared to the total exposure time is again lower here.
Thus, the signal increase during T3 is lower as during T2.

For this reason, darker pixels can be integrated during the complete integration time and the sensor reaches its full sensitivity. Pixels, which are limited at each Knee Points, lose a part of their integration time - even more, if they are brighter.

![Figure 2: Integration time of different bright pixels](image)

In the diagram you can see the signal line of three different bright pixels. The slope depends of the light intensity, thus it is per pixel the same here (granted that the light intensity is temporally constant). Given that the very light pixel is limited soon at the signal levels S1 and S2, the whole integration time is lower compared to the dark pixel. In practice, the parts of the integration time are very different. T1, for example, is 95% of $T_{total}$, T2 only 4% and T3 only 1%. Thus, a high decrease of the very light pixels can be achieved. However, if you want to divide the integration thresholds into three parts that is $S2 = 2 \times S1$ and $S3 = 3 \times S1$, a hundredfold brightness of one pixel's step from S2 to S3, compared to the step from 0 and S1 is needed.

22.6.1.3 Using HDR with mvBlueCOUGAR-X-x00w

Figure 3 is showing the usage of the HDR mode. You can reach the HDR settings via "Setting -> Base -> Camera -> GenICam -> mv High Dynamic Range Control".
22.6 Working with HDR (High Dynamic Range Control)

22.6.1.3.1 Notes about the usage of the HDR mode with mvBlueCOUGAR-X-x00w

- In the HDR mode, the basic amplification is reduced by approx. 0.7, to utilize a huge, dynamic area of the sensor.
- If the manual gain is raised, this effect will be reversed.
- Exposure times, which are too low, make no sense. During the third phase, if the exposure time reaches a possible minimum (one line period), a sensible lower limit is reached.

22.6.1.3.2 Possible settings using mvBlueCOUGAR-X-x00w

Possible settings of the mvBlueCOUGAR-X-x00w in HDR mode are:

"mv HDR Enable":

- "Off": Standard mode
- "On": HDR mode on, reduced amplification:
• "mv HDR Preset":
  
  - "mvDualKneePoint0": Fixed setting with 2 Knee Points. modulation Phase 0 .. 33% / 1 .. 66% / 2 .. 100%
  
  - "mvDualKneePoint1": Phase 1 exposure 12.5% , Phase 2 31.25% of total exposure
  
  - "mvDualKneePoint2": Phase 1 exposure 6.25% , Phase 2 1.56% of total exposure
  
  - "mvDualKneePoint3": Phase 1 exposure 3.12% , Phase 2 0.78% of total exposure
  
  - "mvDualKneePoint4": Phase 1 exposure 1.56% , Phase 2 0.39% of total exposure
  
  - "mvDualKneePoint5": Phase 1 exposure 0.78% , Phase 2 0.195% of total exposure
  
  - "mvDualKneePoint6": Phase 1 exposure 0.39% , Phase 2 0.049% of total exposure
  
  - "mvDualKneePointUser": Variable setting of the Knee Point (1..2), threshold and exposure time proportion

• "mv HDR Selector":
  
  - "mv HDR Voltage first knee point": Control voltage for exposure threshold of first Knee Point (3030mV is equivalent to approx. 33%)
  
  - "mv HDR Voltage second knee point": Control voltage for exposure threshold of first Knee Point (2630mV is equivalent to approx. 66%)
  
  - "mv HDR Exposure first knee point": Proportion of Phase 0 compared to total exposure in parts per million (ppm)
  
  - "mv HDR Exposure second knee point": Proportion of Phase 1 compared to total exposure in parts per million (ppm)

22.6.2 Adjusting sensor -x02d (-1012d)

22.6.2.1 Introduction

The HDR (High Dynamic Range) mode of the Aptina sensor increases the usable contrast range. This is achieved by dividing the integration time in three phases. The exposure time proportion of the three phases can be set independently.

22.6.2.2 Functionality

To exceed the typical dynamic range, images are captured at 3 exposure times with given ratios for different exposure times. The figure shows a multiple exposure capture using 3 different exposure times.
Figure 1: Multiple exposure capture using 3 different exposure times

Note

The longest exposure time (T1) represents the *Exposure_us* parameter you can set in wxPropView.

Afterwards, the signal is fully linearized before going through a compander to be output as a piece-wise linear signal. The next figure shows this.
22.6.2.2.1 Description

Exposure ratios can be controlled by the program. Two ratios are used: $R_1 = \frac{T_1}{T_2}$ and $R_2 = \frac{T_2}{T_3}$.

Increasing $R_1$ and $R_2$ will increase the dynamic range of the sensor at the cost of lower signal-to-noise ratio (and vice versa).

22.6.2.2 Possible settings

Possible settings of the mvBlueCOUGAR-X-x02d in HDR mode are:

- "mv HDR Enable":
  - "Off": Standard mode
  - "On": HDR mode on, reduced amplification

- "mv HDR Preset":
  - "mvDualKneePoint0": Fixed setting with exposure-time-ratios: $T_1 \rightarrow T_2$ ratio / $T_2 \rightarrow T_3$ ratio
  - "mvDualKneePoint1": 8 / 4
  - "mvDualKneePoint2": 4 / 8
  - "mvDualKneePoint3": 8 / 8
  - "mvDualKneePoint4": 8 / 16
  - "mvDualKneePoint5": 16 / 16
  - "mvDualKneePoint6": 16 / 32

Figure 3: wxPropView - Working with the HDR mode
22.6.3 Adjusting sensor -x02e (-1013) / -x04e (-1020)

22.6.3.1 Introduction

The HDR (High Dynamic Range) mode of the e2v sensors increases the usable contrast range. This is achieved by adjusting the logarithmic response of the pixels.

22.6.3.2 Functionality

MATRIX VISION offers the "mv Linear Logarithmic Mode" to use the HDR mode of the e2v sensors. With this mode you can set the low voltage of the reset signal at pixel level.

![Knee-Point of the e2v HDR mode shifts the linear / logarithmic level](image)

Figure 1: Knee-Point of the e2v HDR mode shifts the linear / logarithmic level

You can find the "mv Linear Logarithmic Mode" in "Setting -> Base -> Camera -> GenICam -> Analog Control":

![User Experience](image)
The following figure shows the measured curves at 2 ms and 20 ms exposure time with four different "mv Linear Logarithmic Mode" settings:

![Image of measured curves](image)

The curves are at Gain 1, lambda = 670 nm (40 nm width), room temperature, nominal power supply values, on a 100 x 100 pixel centered area.

<table>
<thead>
<tr>
<th>&quot;mv Linear Logarithmic Mode&quot; value</th>
<th>Dynamic max (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>T = 2 ms</td>
</tr>
<tr>
<td>4</td>
<td>47</td>
</tr>
<tr>
<td>5</td>
<td>74</td>
</tr>
<tr>
<td>6</td>
<td>85</td>
</tr>
<tr>
<td>7</td>
<td>92</td>
</tr>
</tbody>
</table>

22.7 Working with LUTs

There are several use cases concerning LUTs (Look-Up-Tables):
22.7 Working with LUTs

22.7.1 Introducing LUTs

22.7.1.1 Introduction

Look-Up-Tables (LUT) are used to transform input data into a desirable output format. For example, if you want to invert an 8 bit image, a Look-Up-Table will look like the following:

![Look-Up-Table which inverts a pixel of an 8 bit mono image](image)

I.e., a pixel which is white in the input image (value 255) will become black (value 0) in the output image.

All MATRIX VISION devices use a hardware based LUT which means that

- no host CPU load is needed and
- the LUT operations are independent of the transmission bit depth.

22.7.1.2 Setting the hardware based LUTs via LUT Control

On the mvBlueCOUGAR-X using wxPropView (p.115), you will find the LUT Control (p.182) via "Setting -> Base -> Camera -> GenICam -> LUT Control".

wxPropView (p.115) offers a wizard for the LUT Control (p.182) usage:

1. Click on "Setting -> Base -> Camera -> GenICam -> LUT Control". Now, the "Wizard" button becomes active.
2. Click on the "Wizard" button to start the LUT Control wizard tool. The wizard will load the LUT data from the camera.
It is easy to change settings like the Gamma value of the Luminance or of each color channel (in combination with a color sensor of course) with the help of the wizard. You can also invert the values of each pixel with the wizard. It is not possible to set a LUT mode and the "mv LUT Mapping" is fixed.

Make your changes and do not forget to

1. click on "Copy to..." and select "All" or the color channel you need, to
2. click on "Enable All", and finally, to
3. click on Synchronize and play the settings back to the device (via "Cache -> Device").

**Note**

If you select "Enable All" without entering any value the image will be inverted.
22.7.1.3 Setting the Host based LUTs via LUTOperations

Host based LUTs are also available via "Setting -> Base -> ImageProcessing -> LUTOperations"). Here, the changes will affect the 8 bit image data and the processing needs the CPU of the host system.

Three "LUTMode"s are available:

- "Gamma"
  You can use "Gamma" to lift darker image areas and to flatten the brighter ones. This compensates the contrast of the object. The calculation is described here. It makes sense to set the "← GammaStartThreshold" higher than 0 to avoid a too extreme lift or noise in the darker areas.

- "Interpolated"
  With "Interpolated" you can set the key points of a characteristic line. You can defined the number of key points. The following figure shows the behavior of all 3 LUTInterpolationModes with 3 key points:

![Interpolated LUTModes](image)

Figure 4: LUTMode "Interpolated" -> LUTInterpolationMode

- "Direct"
  With "Direct" you can set the LUT values directly.

22.7.1.3.1 Example 1: Inverting an Image

To get an inverted 8 bit mono image like shown in Figure 1, you can set the LUT using wxPropView (p.115). After starting wxPropView (p.115) and using the device,

1. Set "LUTEnable" to "On" in "Setting -> Base -> ImageProcessing -> LUTOperations".
2. Afterwards, set "LUTMode" to "Direct".
3. Right-click on "LUTs -> LUT0 -> DirectValues[256]" and select "Set Multiple Elements... -> Via A User Defined Value Range". This is one way to get an inverted result. It is also possible to use the "LUTMode" - "Interpolated".

4. Now you can set the range from 0 to 255 and the values from 255 to 0 as shown in Figure 2.

![Figure 5: Inverting an image using wxPropView with LUTMode "Direct"

22.7.2 Working with LUTValueAll

Working with the LUTValueAll feature requires a detailed understanding on both Endianess and the camera's internal format for storing LUT data. LUTValueAll typically references the same section in the camera's memory as when accessing the LUT via the features LUTIndex and LUTValue.

LUT data can either be written to a device like this (C++ syntax):

```cpp
const size_t LUT_VALUE_COUNT = 256;
int64_type LUTData[LUT_VALUE_COUNT] = getLUTDataToWriteToTheDevice();
mvIMPACT::acquire::GenICam::LUTControl lut(getDevicePointerFromSomewhere());
for(int64_type i=0; i< static_cast<int64_type>(LUT_VALUE_COUNT); i++)
{
    lut.LUTIndex.write( i );
    lut.LUTValue.write( LUTData[i] );
}
```
When using this approach all the Endianess related issues will be handled completely by the GenICam runtime library. So this code is straight forward and easy to understand but might be slower than desired as it requires a lot of direct register accesses to the device.

In order to allow a fast efficient way to read/write LUT data from/to a device the LUTValueAll feature has been introduced. When using this feature the complete LUT can be written to a device like this:

```c
const size_t LUT_VALUE_COUNT = 256;
int LUTData[LUT_VALUE_COUNT] = getLUTDataToWriteToTheDevice();
mvIMPACT::acquire::GenICam::LUTControl lut(getDevicePointerFromSomewhere());
std::string buf(reinterpret_cast<std::string::value_type*>(&LUTData), sizeof(LUTData));
lut.LUTValueAll.writeBinary(buf);
```

**BUT** as this simply writes a raw block of memory to the device it suddenly becomes important to know exactly how the LUT data is stored inside the camera. This includes:

- The size of one individual LUT entry (this could be anything from 1 up to 8 bytes)
- The Endianess of the device
- The Endianess of the host system used for sending/receiving the LUT data

The first item has impact on how the memory must be allocated for receiving/sending LUT data. For example when the LUT data on the device uses a ‘one 32-bit integer per LUT entry with 256 entries’ layout then of course this is needed on the host system:

```c
const size_t LUT_VALUE_COUNT = 256;
int LUTData[LUT_VALUE_COUNT];
```

When the Endianess of the host system differs from the Endianess used by the device the application communicates with, then before sending data assembled on the host might require Endianess swapping. For the example from above this would e.g. require something like this:

```c
#define SWAP_32(l) \
 (((((l) & 0xff000000) >> 24) | \ 
 (((l) & 0x00ff0000) >> 8) | \ 
 (((l) & 0x0000ff00) << 8) | \ 
 (((l) & 0x000000ff) << 24))

void fn() 
{
    const size_t LUT_VALUE_COUNT = 256;
    int LUTData[LUT_VALUE_COUNT] = getLUTDataToWriteToTheDevice();
    mvIMPACT::acquire::GenICam::LUTControl lut(getDevicePointerFromSomewhere());
    for( size_t i=0; i<LUT_VALUE_COUNT; i++ )
    {
        LUTData[i] = SWAP_32(LUTData[i]);
    }
    std::string buf(reinterpret_cast<std::string::value_type*>(&LUTData), sizeof(LUTData));
    lut.LUTValueAll.writeBinary(buf);
}
```

For details on how the LUT memory is organized for certain sensors please refers to the Sensor overview. Please note that all mvBlueCOUGAR-S, mvBlueCOUGAR-X and mvBlueCOUGAR-XD devices are using Big Endian while almost any Windows or Linux distribution on the market uses Little Endian, thus the swapping of the data will most certainly be necessary when using the LUTValueAll feature.
22.7 Working with LUTs

22.7.3 Implementing a hardware-based binarization

If you like to have binarized images from the camera, you can use the hardware-based Look-Up-Tables (LUT) (p. 182) which you can access via "Setting -> Base -> Camera -> GenICam -> LUT Control".

To get binarized images from the camera, please follow these steps:

1. Set up the camera and the scenery, e.g.

![Scenery](image)

Figure 1: Scenery

2. Open the LUT wizard via the menu "Wizards -> LUT Control...".

3. Export the current LUT as a "*.csv" file.

The "*.csv" file contains just one column for the output gray scale values. Each row of the "*.csv" represents the input gray scale value. In our example, the binarization threshold is 1024 in a 12-to-9 bit LUT. I.e., we have 4096 (= 12 bit) input values (= rows) and 512 (= 9 bit) output values (column values). To binarize the image according to the threshold, you have to

1. set all values below the binarization threshold to 0.

2. Set all values above the binarization threshold to 511:
3. Now, save the "*.csv" file and
4. import it via the LUT Control wizard.
5. Click on synchronize and
6. finally check "Enable".

Afterwards the camera will output binarized images like the following:
22.8 Saving data on the device

**Note**

As described in *Storing and restoring settings* (p. 125), it is also possible to save the settings as an XML file on the host system. You can find further information about for example the XML compatibilities of the different driver versions in the mvIMPACT Acquire SDK manuals and the according setting classes: https://www.matrix-vision.com/manuals/SDK_CPP/classmvIMPACT_1___acquire_1_1FunctionInterface.html (C++)

There are several use cases concerning device memory:

### 22.8.1 Creating user data entries

#### 22.8.1.1 Basics about user data

It is possible to save arbitrary user specific data on the hardware's non-volatile memory. The amount of possible entries depends on the length of the individual entries as well as the size of the devices non-volatile memory reserved for storing:

- mvBlueFOX,
- mvBlueFOX-M,
- mvBlueFOX-MLC,
- mvBlueFOX3, and
- mvBlueCOUGAR-X

Currently offer 512 bytes of user accessible non-volatile memory of which 12 bytes are needed to store header information leaving 500 bytes for user specific data.

One entry will currently consume:

\[
1 + \text{<length_of_name (up to 255 chars)> } + 2 + \text{<length_of_data (up to 65535 bytes)> } + 1 \text{ (access mode) bytes }
\]

as well as an optional:

\[
1 + \text{<length_of_password> bytes per entry if a password has been defined for this particular entry }
\]

It is possible to save either String or Binary data in the data property of each entry. When storing binary data please note, that this data internally will be stored in Base64 format thus the amount of memory required is 4/3 time the binary data size.

The UserData can be accessed and created using wxPropView (p. 115) (the device has to be closed). In the section "UserData" you will find the entries and following methods:

- "CreateUserDataEntry"
- "DeleteUserDataEntry"
- "WriteDataToHardware"
To create a user data entry, you have to

- **Right click on** "CreateUserDataEntry"

- **Select** "Execute" from the popup menu.

  An entry will be created.

- **In** "Entries" **click on the entry you want to adjust and modify the data fields.**

  To permanently commit a modification made with the keyboard the **ENTER key must be pressed.**

- **To save the data on the device, you have to execute** "WriteDataToHardware". Please have a look at the "Output" tab in the lower right section of the screen as shown in Figure 2, to see if the write process returned with no errors. If an error occurs a message box will pop up.

---

**Figure 1: wxPropView - section "UserData -> Entries"**

---

**Figure 2: wxPropView - analysis tool "Output"**
22.8 Saving data on the device

22.8.1.2 Coding sample

If you e.g. want to use the UserData as dongle mechanism (with binary data), it is not suggested to use wxPropView (p. 115). In this case you have to program the handling of the user data.

See also

mvIMPACT::acquire::UserDataEntry in mvIMPACT_Acquire_API_CPP_manual.chm.

22.8.2 Creating user set entries

With mvBlueCOUGAR-X it is possible to store up to five configuration sets (4 user plus one factory default) in the camera.

This feature is similar to the storing settings functionality (p. 379), which saves the settings in the registry. However, as mentioned before the user sets are stored in the camera.

The user set stores

- exposure,
- gain,
- AOI,
- frame rate,
- LUT (p. 182),
- one Flat-Field Correction (p. 294),
- defective pixels (p. 288),
- etc.

permanently and is independent of the computer which is used.

Additionally, you can select, which user set comes up after hard reset.

**Attention**

The storage of user data in the registry can still override user set data!
User sets are cleared after firmware change.
22.8.2.1 List of ignored properties

Following properties are not stored in the user set:

- ActionUnconditionalMode
- DeviceLinkHeartbeatMode
- DeviceLinkHeartbeatTimeout
- DeviceStreamChannelEndianness
- DeviceTLType
- EventExposureEndData
- EventFrameEndData
- EventLine4RisingEdgeData
- EventLine5RisingEdgeData
- FileAccessBuffer
- FileAccessLength
- FileAccessOffset
- FileOpenMode
- GevCCP
- GevCurrentIPConfigurationDHCP
- GevCurrentIPConfigurationPersistentIP
- GevDiscoveryAckDelay
- GevGVCPExtendedStatusCodes
- GevGVCPHeartbeatDisable
- GevGVCPPendingAck
- GevGVSPExtendedIDMode
- GevHeartbeatTimeout
- GevMCDA
- GevWCMHostPort
- GevMCRC
- GevMCTT
- GevPersistentDefaultGateway
- GevPersistentIPAddress
- GevPersistentSubnetMask
- GevPhysicalLinkConfiguration
- GevPrimaryApplicationSwitchoverKey
- GevSCCFGAllInTransmission
- GevSCCFGExtendedChunkData
- GevSCCPLargeLeaderTrailer
- GevSCCFOMultiPart
- GevSCCFGPacketResendDestination
- GevSCCFGUnconditionalStreaming
- GevSCDA
- GevSCP0
- GevSCPMHostPort
- GevSCPInterfaceIndex
- GevSCPSBigEndian
- GevSCPSSoNotFragment
- GevSCPGFireTestPacket
- GevSCPSPacketSize
- LUTIndex
- LUTValueAll
- PtpControl
- PtpEnable
- UserSetDefault
- mvADCGain
- mvDefectivePixelCount
- mvDefectivePixelOffsetX
- mvDefectivePixelOffsetY
- mvDigitalGainOffset
- mvFFCAutoLoadMode
- mvI2cInterfaceASCIIBuffer
- mvI2cInterfaceBinaryBuffer
- mvI2cInterfaceBytesToRead
- mvI2cInterfaceBytesToWrite
- mvPreGain
- mvSerialInterfaceASCIIBuffer
- mvSerialInterfaceBinaryBuffer
- mvSerialInterfaceBytesToRead
- mvSerialInterfaceBytesToWrite
- mvUserData
- mvVRamp
22.8.2.2 Working with the user sets

You can find the user set control in "Setting -> Base -> Camera -> GenICam -> User Set Control".

With "User Set Selector" you can select the user set ("Default", "UserSet1 - UserSet4"). To save or load the specific user set, you have two functions:

- "int UserSetLoad()"
- "int UserSetSave()"

"User Set Default" is the property, where you can select the user set, which comes up after hard reset.

Finally, with "mv User Data" you have the possibility to store arbitrary user data.
22.8.3 Working with the UserFile section (Flash memory)

The mvBlueCOUGAR-X offers a 64 KByte section in the Flash memory that can be used to upload a custom file to (UserFile).

To read or write this file you can use the following GenICam File Access Control (p. 180) and its interfaces:

- IDevFileStream (read)
- ODevFileStream (write)

**Attention**

The UserFile is lost each time a firmware update is applied to the device.

22.8.3.1 Using wxPropView

wxPropView (p. 115) offers a wizard for the File Access Control (p. 180) usage:

1. Click on "Setting -> Base -> Camera -> GenICam -> File Access Control -> File Selector -> File Operator Selector". Now, the "Wizard" button becomes active.
2. Click on the "Wizard" button. Now, a dialog appears where you can choose either to upload or download a file.
3. Make your choice and click on "OK". Now, a dialog appears where you can select the File.

4. Select "UserFile" follow the instructions.

22.8.3.2 Manually control the file access from an application (C++)

The header providing the file access related classes must be included into the application:

```
#include <mvIMPACT_CPP/mvIMPACT_acquire_GenICam_FileStream.h>
```
A write access then will look like:

```cpp
class mvIMPACT::acquire::GenICam::ODevFileStream

const string fileNameDevice("UserFile");

// uploading a file
mvIMPACT::acquire::GenICam::ODevFileStream file;
file.open( pDev, fileNameDevice.c_str() );
if( !file.fail() )
{
   // Handle the successful upload.
}
else
{
   // Handle the error.
}
```

A read access will look like:

```cpp
class mvIMPACT::acquire::GenICam::IDevFileStream

const string fileNameDevice("UserFile");

// downloading a file works in a similar way
mvIMPACT::acquire::GenICam::IDevFileStream file;
file.open( pDev, fileNameDevice.c_str() );
if( !file.fail() )
{
   // Handle the successful upload.
}
else
{
   // Handle the error.
}
```

You can find a detailed code example in the C++ API manual in the documentation of the classes `mvIMPACT::acquire::GenICam::IDevFileStream` and `mvIMPACT::acquire::GenICam::ODevFileStream`.

22.9 Working with device features

There are several use cases concerning device features:

22.9.1 Working with the temperature sensors

The `mvBlueCOUGAR-X` offers two different temperature sensors:

- on the sensor board (typically lower)
- on FGPA board (typically higher)
Note

Avoid temperatures higher than 80 C by lowering thermal resistance of mvBlueCOUGAR-X housing to connecting structure active cooling means.

It is possible to regulate the temperature of the camera. The limits of this feature are

- upper limit (≤ 255 C) and
- lower limit (≥ 0 C).

Furthermore, the hysteresis (0, 1, 5, 3, 6 C) difference between on/off prevents oscillation. The temperature out of range will set selected output high (see Figure 2). The output can directly drive a fan or a (modest) heating system (but not both due to limit of IC).
22.9.2 Disabling the heartbeat

The GigE Vision Control Protocol (GVCP) is an application protocol relying on the UDP (p. 242) transport layer protocol. Because UDP (p. 242) is a connectionless protocol, a recovery mechanism has been defined to detect that one of the two participants has been detached without closing the control channel before. A heartbeat message is used to achieve this.

In the default operation mode, these heartbeat messages are exchanged with a user selectable frequency ("Setting -> Base -> Camera -> GenICam -> Device Control -> Device Link Heartbeat Timeout"; orange box in Figure 1). As a result, during this interval a certain register (GevCCP = Control Channel Privilege) is read several times from the device to reset the heartbeat timer interval in order to ensure the control channel stays open.
If the heartbeat bit is set to disable ("Device Link Heartbeat Mode = On"; red box in Figure 1), the transport layer lib will stop reading the GevCCP register periodically. To disable the heartbeat makes sense for at least two scenarios:

- while debugging a program and
- in Triggered line scan acquisition with pausing trigger signals (p. 283).

In both scenarios, one participant pauses the connection on purpose.

If the heartbeat disable bit is reset again, the transport layer lib will start reading the CCP register periodically again.
22.9.3 Reset timestamp by hardware

This feature can be used:

- for precise control of timestamp
  - for one camera or
  - to synchronize timestamp of multitude of cameras.

The latter sample, can be achieved by following steps:

1. Define the input line ("TriggerSource") to reset the timestamp, e.g. "Line5" and
2. set the "Trigger Selector" to "mvTimestampReset".
3. Connect all input lines of all cameras together.
4. Finally, use one output of one camera to generate reset edge:

![wxPropView - Setting the sample](image-url)
Note

Be aware of the drift of the individual timestamps.

The timestamp is generated via FPGA in the camera which itself is clocked by a crystal oscillator. This is done independently in each camera and by default not synchronized among cameras or the host system.

Typical stability of crystal oscillators is in the range 100ppm (parts per million).

I.e. for longer operation times (say in excess of hours) there is a tendency that timestamps of individual cameras drift against each other and against the time in the operating system of the host.

Customers wishing to use the individual camera timestamps for synchronization and identification of images via timestamps for multi-camera systems will have in the meantime

- to reset all timestamps either by hardware signal or by command and regularly resynchronize or check the drift algorithmically
- in order to make sure that the drift is less half an image frame time.

22.9.4 Synchronizing camera timestamps without IEEE 1588

22.9.4.1 Introduction

Camera timestamps are a recommended Genicam (p.234) / SFNC (p.242) feature to add the information when an image was taken (exactly: when the exposure of the image started).

Without additional synchronization it is merely a camera individual timer with a vendor specific increment and implementation dependent accuracy. Each camera starts its own timestamp beginning with zero and there are no means to adjust or synchronize them among cameras or host PCs. There is effort ongoing to widely establish the precision timestamp according to "IEEE 1588" into GigE cameras. This involves cameras which are able to perform the required synchronization as well as specific network hardware and driver software and procedures to do and maintain the synchronization.

There are many applications which do not or cannot profit from "IEEE 1588" but have certain synchronization needs. Solutions for these scenarios are described as follows.

22.9.4.2 Resetting timestamp using mvTimestampReset

First of all the standard does not provide hardware means to reset the timestamp in a camera other than plug off and on again. Therefore MATRIX VISION has created its own mechanism mvTimestampReset to reset the timestamp by a hardware input.
This can be elegantly used for synchronization purposes by means of wiring an input of all cameras together and reset all camera timestamps at the beginning by a defined signal edge from the process. From this reset on all cameras start at zero local time and will increment independently their timestamp so that we achieve a basic accuracy only limited by drift of the clock main frequency (e.g. a 1 MHz oscillator in the FPGA) over time.

In order to compensate for this drift we can in addition reset the timestamp every second or minute or so and count the reset pulse itself by a counter in each camera. Assuming this reset pulse is generated by the master camera itself by means of a timer and output as the hardware reset signal for all cameras, we now can count the reset pulse with all cameras and put both the count and the reset timestamp as so called chunk data in the images.

We thus have achieved a synchronized timestamp with the precision of the master camera among all connected cameras.

Settings required are shown using MATRIX VISION’s `wxPropView` tool:
An example of the chunk data attached to the image can be seen below. The timestamp is in µs and Counter1 counts the reset pulses, in this case itself generated by the camera via Timer1.

The task of resetting the counter at the beginning of the acquisition can be done by setting the reset property accordingly. Of course is this all independent whether the camera is acquiring images in triggered or continuous mode.

22.9.4.3 Synchronizing timestamp using a pulse-per-second signal

In order to eliminate the unknown drifts of different devices, a PPS (pulse per second) signal can be fed into each camera using a PC with NTP (network time protocol software), GPS devices or even a looped-back camera timer.

From these pulses the device will find out how long one second is. When a device detects that it is no longer running precisely it will adapt it's internal clock leading to a "stabilized oscillator".

The device would then maintain their timestamp-differences over long times and stay synchronized. The initial difference between the timers - before the PPS was used - remains however. If you aim to eliminate that as well, you can use the mvTimestampReset up front with the same PPS input signal. In an application this can be configured like this (C# syntax):

```csharp
// --------------------------------------------------------------------------
bool waitForNextPulse(Device pDev, String triggerLine)
// --------------------------------------------------------------------------
{
    GenICam.CounterAndTimerControl ctc = new GenICam.CounterAndTimerControl(pDev);
    ctc.counterEventSource.writeS(triggerLine);
    long momentaryValue = ctc.counterValue.read();
    for (int i=0; i<12; i++)
    {
        System.Threading.Thread.Sleep(100);
        if (momentaryValue != ctc.counterValue.read())
        {
            return true;
        }
    }
    return false;
}
// --------------------------------------------------------------------------
void SetupPPS(Device[] pDevs)
// --------------------------------------------------------------------------
{
    string TriggerLine = "Line4";

    if (!waitForNextPulse(pDevs[0], TriggerLine))
```
```csharp
if (!waitForNextPulse(pDevs[0], TriggerLine))
{
    Console.WriteLine("the pulses aren’t coming fast enough ...");
    return;
}

// Now switch off the reset of the timestamp again. All devices did restart their
// timestamp counters and will stay in sync using the PPS signal now
foreach (Device aDevice in pDevs)
{
    GenICam.AcquisitionControl ac = new GenICam.AcquisitionControl(aDevice);
    ac.triggerMode.writeS("Off");
    GenICam.DeviceControl dc = new GenICam.DeviceControl(aDevice);
    dc.mvTimestampPPSSync.writeS("Line4");
}
```
Setting the Timer1 to 1 s seems like an easy task but due to some internal dependencies you should be carefully here. At the moment two different timer implementations are present in our products.

- **Type 1**: For mvBlueCOUGAR-X cameras with sensors other than the IMX family from Sony please set the duration to the theoretical value of 1000000.
- **Type 2**: For all the other cameras please use 999997 us duration since the self-triggering will consume the other 3 us.
- Please refrain from switching on PPSSync inside the Master camera since (at least in Type 1 cameras) since this will lead to an instable feedback loop.

22.9.5 Working with the 3 head model

The 3 head model behaves like a mvBlueCOUGAR-X standard camera and for this reason it can be used with every GigE Vision compliant software. The information of the three sensors are returned as a packed pseudo RGB image, with each color channel representing one sensor.

While working with trigger signals (p. 335), a trigger signal will trigger all sensors simultaneously.

Because of the three sensor heads, you can set the gain and the offset (black level) of each sensor separately in "Setting -> Base -> Camera -> GenICam -> Analog Control -> "

- "Gain Selector"
22.9 Working with device features

• "Black Level Selector"

![Figure 1: wxPropView - Gain and Black Level Selector](image)

22.9.6 Working with the serial interface (mv Serial Interface Control)

22.9.6.1 Introduction

As mentioned in GenICam and Advanced Features section of this manual, the mv Serial Interface Control (p. 180) is a feature which allows an easy integration of motor lenses or other peripherals based on RS232.

• Available message buffer size: 1 KByte

**Note**

Use the Power GND for the RS232 signal.
22.9.6.2 Setting up the device

Follow these steps to prepare the camera for the communication via RS232:

1. Start *wxPropView* (p. 115)

2. Connect to the camera

3. Under "Setting -> Base -> Camera -> GenICam -> mv Serial Interface Control" activate the serial interface by enabling "mv Serial Interface Enable" (1). Afterwards "mv Serial Interface Control" is available.

4. Set up the connection settings to your needs (2).

5. To test the settings you can send a test message (3).

6. Send messages by executing the function "int mvSerialInterfaceWrite( void )" by either clicking on the 3 dots next to the function name or by right-clicking on the command and then selecting Execute from the context menu.(4)

If you listening to the RS232 serial line using a tool like *PuTTY* with matching settings...
you will see the test message:

```
Test Test Test
```
22.9.6.3 Programming the serial interface

```cpp
#include <mvIMPACT_CPP/mvIMPACT_acquire_GenICam.h>

// more code
GenICam::mvSerialInterfaceControl sic( pDev );
sic.mvSerialInterfaceBaudRate.writeS( "Hz_115200" );
sic.mvSerialInterfaceASCIIBuffer.writeS( "Test Test Test" );
sic.mvSerialInterfaceWrite();
// more code
```

22.10 Working with several cameras simultaneously

There are several use cases concerning multiple cameras:

22.10.1 Introducing multicasting

**GigE Vision** supports to stream data over a network via Multicast to multiple destinations simultaneously. Multicast means, that one source streams data to a group of recipients in the same subnet. As long as the recipients are members of this Multicast group they will get the packages. Other members of the same local subnet will skip the packages (but the packages are sent and so they need bandwidth).

See also

http://en.wikipedia.org/wiki/Multicast

To set up a Multicast environment, **GigE Vision** introduced camera access types. The most important ones are

- Control
- Read.

With **Control**, a primary application can used to set up the camera which will stream the data via Multicast. With **Read** you can set up secondary applications which will playback the data stream.
Because the mvBlueCOUGAR cameras are GigE Vision compliant devices, you can set up a Multicast environment using wxPropView. This use case will show how you can do this.

22.10.1.1 Sample

On (the primary) application you have to establish "Control" access.

For this,

1. please start wxPropView and select the camera.

2. Click on the "Device" section.

3. Click on "DesiredAccess" and choose "Control".
See also
desiredAccess and grantedAccess in

- `mvlIMPACT::acquire::Device` (in `mvIMPACT_Acquire_API_CPP_manual.chm`)
- `TDeviceAccessMode` (in `mvIMPACT_Acquire_API_CPP_manual.chm`)

4. Click on "Use".

5. Now, select the "Setting -> Base -> Camera -> GenICam" section and open the "Transport Layer Control" subsection (using the device specific interface (p. 143): "System Settings" and "TransportLayer").

6. In "GevSCDA" enter a Multicast address like "239.255.255.255".
Figure 3: wxPropView - Primary application setting GevSCDA in “GevSCDA”
One or more applications running on different machines can then establish "read-only" access to the very same device.

**Note**

The machines of the secondary applications have to be connected to the same network as the primary application.

1. Please start **wxPropView** (p. 115) on the other machine and click on the "Device" section.

![Figure 4: wxPropView - Secondary application setting DesiredAccess to "Read"](image-url)
2. Features will not be writeable as you can see at the "Transport Layer Control" parameters in Figure 4.

![Figure 5: wxPropView - Secondary application read-only "Transport Layer Control" parameters](image)

3. Once the primary application starts to request images, the secondary applications will be able to receive these images as well. Please click on "Use" and then "Acquire" ("Acquisition Mode" = "Continuous").
Figure 6: wxPropView - Secondary application receives images from the primary application

**Note**

The machine that has "Control" access automatically joins the streaming multicast group of the camera it is controlling. If this is not desired, the "mv Auto Join Multicast Groups" property has to be set to **false**.
22.10 Working with several cameras simultaneously

Figure 7: wxPropView - Disable mv Auto Join Multicast Groups to prevent the Control machine to receive multicast streaming data

**Attention**

**GigE Vision** (p. 236) (GEV) does not allow packet fragmentation for performance reasons. Because of that and the nature of multicasting and network data transfer in general it is crucial to select a packet size every client with respect to its **NIC** (p. 240)’s **MTU** (p. 240) can handle as otherwise not all clients will be able to receive the image data. See the chapter

- Optimizing the network configuration (p. 47) (Windows)
- Optimizing the network configuration (p. 57) (Linux)

for details.

22.10.2 Using Action Commands

**GigE Vision** (p. 236) specifies so called **Action Commands** to trigger an action in multiple devices at roughly the same time.

**Note**

Due to the nature of Ethernet, this is not as synchronous as a hardware trigger since different network segments can have different latencies. Nevertheless in a switched network, the resulting jitter is acceptable for a broad range of applications and this scheme provides a convenient way to synchronize devices by means of a software command.

Action commands can be unicasted or broadcasted by applications with either **exclusive**, **write** or **read** (only when the device is configured accordingly) access to the device. They can be used e.g. to

- increment or reset counters
- reset timers
- act as trigger sources
- ...

The most typical scenario is when an application desires to trigger a simultaneous action on multiple devices. This case is showed by the figure below. The application fires a broadcast action command that will reach all the devices on the subnet.
Figure 1: Action command sent as a broadcast to all devices in the subnet

**Attention**

Due to the nature of Ethernet an action command at a given time can only leave a single network interface thus sending action commands via several different network interfaces to different devices that are e.g. directly connected to a given network interface card in a PC only works sequential and therefore these commands will \textbf{NOT} reach the devices at the same time. This can only be achieved using \textit{ScheduledActionCommands}. Depending on whether an action command is unicasted or broadcasted the command will either reach a single device on a given subnet or multiple devices.

**Note**

The following diagrams assume the connecting line between the PC and the devices to be the GVCP (GigE Vision control protocol)’s control port socket. Therefore these diagram do no make any assumption about the physical connection between the devices and the PC. However for the examples to work they must all be connected to the same network interface of the PC using a network switch as otherwise as stated above it would not be possible to sent the very same action command using a single network packet to all the devices.

But action commands can also be used by secondary applications. This can even be another device on the same subnet. This is depicted in the following figure.

Figure 2: Action command sent as a broadcast by one device to all other devices in the subnet
Upon reception of an action command, the device will decode the information to identify which internal action signal is requested. An action signal is a device internal signal that can be used as a trigger for functional units inside the device (ex: a frame trigger). It can be routed to all signal sinks of the device.

Each action command message contains information for the device to validate the requested operation:

1. `device_key` to authorize the action on this device.
2. `group_key` to define a group of devices on which actions have to be executed.
3. `group_mask` to be used to filter out some of these devices from the group.

Action commands can only be asserted if the device has an open primary control channel (so if an application has established write or exclusive access to a device) or when unconditional action mode is enabled.

A device can define several action commands which can be selected via the `ActionSelector` in the device features property tree.

The conditions for an action command to be asserted by the device are:

1. the device has an open primary control channel or unconditional action mode is enabled.
2. the `device_key` in the action command sent by the application and the `ActionDeviceKey` property of the device must be equal.
3. the `group_key` in the action command sent by the application and the `ActionGroupKey` property for the corresponding action of the device must be equal.
4. the logical AND-wise operation of `group_mask` in the action command sent by the application against the `ActionGroupMask` for the corresponding action of the device must be non-zero. Therefore, they must have at least one common bit set at the same position in the register.

When these 4 conditions are met for at least one action signal configured on the device, then the device internally asserts the requested action. As these conditions could be met on more than one action, the device could assert more than one action signal in parallel. When one of the 4 conditions is not met for any supported action signal, then the device ignores the action command.

This first condition asks for the write or exclusive access being established between the application and the device unless the `ActionUnconditionalMode` has been enabled. When this bit is set, the device can assert the requested action even if no application has established write or exclusive access as long as the 3 other conditions are met.

22.10.2.1 Scheduled Action Command

Scheduled Action Commands provide a way to trigger actions in a device at a specific time in the future. The typical use case is depicted in the following diagram:

![Figure 3: Principle of scheduled action commands](image-url)
The transmitter of an action command records the exact time, when the source signal is asserted (External signal). This time \( t_0 \) is incremented by a delta time \( \text{DELTA}_L \) and transmitted in an action command to the receivers. The delta time \( \text{DELTA}_L \) has to be larger than the longest possible transmission and processing latency of the action command in the network.

If the packet passes the action command filters in the receiver, then the action signal is put into a time queue (the depth of this queue is indicated by the \text{ActionQueueSize} property).

When the time of the local clock is greater or equal to the time of an action signal in the queue, the signal is removed from the queue and asserted. Combined with the timestamp precision of IEEE 1588 which can be sub-microseconds, a Scheduled Action Command provides a way to allow low-jitter software trigger. If the sender of an action command is not capable to set a future time into the packet, the action command has a flag to fall back to legacy mode (bit 0 of the flag field). In this mode the signal is asserted in the moment the packet passes the action command filters.

**Attention**

Scheduled action commands are not supported by every device. A device supporting scheduled action commands should also support time stamp synchronization based on IEEE1588. MATRIX VISION devices currently do **NOT** support these features.

### 22.10.2.2 Examples

The following examples illustrate the behavior of action commands in various scenarios. The figure below shows 4 different action commands sent by an application. The content of the action command packet is illustrated on the left side of the figure.
The content of the action command must be examined against the conditions listed above for each supported action signal.

For the first request (ACTION_CMD #1)

<table>
<thead>
<tr>
<th>Action Command 1</th>
<th>Device 0</th>
<th>Device 1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ACTION 0</td>
<td>ACTION 1</td>
</tr>
<tr>
<td>device_key</td>
<td>0x34638452</td>
<td>0x34638452</td>
</tr>
<tr>
<td>group_key</td>
<td>0x00000024</td>
<td>0x00000024</td>
</tr>
<tr>
<td>group_mask</td>
<td>0x00000003</td>
<td>0x00000001</td>
</tr>
</tbody>
</table>

Device 0 receives the request and looks for the 4 conditions.

1. exclusive or write access has been established between the application and the device (or unconditional action is enabled)
2. device_key matches
3. group_key matches
4. Logical AND-wise comparison of requested \texttt{group\_mask} is non-zero

All 4 conditions are met only for ACTION\_0, thus the device immediately asserts the internal signal represented by ACTION\_0. The same steps are followed by Device 1. Only the \texttt{group\_mask} is different, but nevertheless the logical bitwise AND operation produces a non-zero value, leading to the assertion of ACTION\_0 by Device 1.

For the second request (ACTION\_CMD \#2)

<table>
<thead>
<tr>
<th>Action Command 2</th>
<th>Device 0</th>
<th>Device 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{device_key}</td>
<td>0x34638452</td>
<td>0x34638452</td>
</tr>
<tr>
<td>\texttt{group_key}</td>
<td>0x00000042</td>
<td>0x000000024</td>
</tr>
<tr>
<td>\texttt{group_mask}</td>
<td>0x000000F2</td>
<td>0x000000001</td>
</tr>
</tbody>
</table>

Looking for the 4 conditions, Device 0 will assert ACTION\_1 while Device 1 will not assert any signal because the \texttt{group\_key} condition is not met. Therefore, Device 1 ignores the request.

For the third request (ACTION\_CMD \#3)

<table>
<thead>
<tr>
<th>Action Command 3</th>
<th>Device 0</th>
<th>Device 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{device_key}</td>
<td>0x34638452</td>
<td>0x34638452</td>
</tr>
<tr>
<td>\texttt{group_key}</td>
<td>0x00000024</td>
<td>0x000000024</td>
</tr>
<tr>
<td>\texttt{group_mask}</td>
<td>0x00000002</td>
<td>0x000000001</td>
</tr>
</tbody>
</table>

In the third example, the \texttt{group\_mask} and \texttt{group\_key} of Device 0 do not match with ACTION\_CMD \#3 for any of the ACTION\_0 to ACTION\_2. Therefore, Device 0 ignores the request. Device 1 asserts ACTION\_0 since the 4 conditions are met.

The ACTION\_CMD is flexible enough to accommodate “simultaneous” triggering of the same functional action in functionally different devices.

For instance, let’s assume the software trigger of Device 0 can only be associated to its ACTION\_3 and that the software trigger of Device 1 can only be associated to its ACTION\_1. And the action command supports to trigger the same functional action provided that their respective action group key and masks are set in order to meet the conditions from the previous list.

For the fourth request (ACTION\_CMD \#4)

<table>
<thead>
<tr>
<th>Action Command 4</th>
<th>Device 0</th>
<th>Device 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{device_key}</td>
<td>0x34638452</td>
<td>0x34638452</td>
</tr>
<tr>
<td>\texttt{group_key}</td>
<td>0x00000001</td>
<td>0x00000001</td>
</tr>
<tr>
<td>\texttt{group_mask}</td>
<td>0x00000001</td>
<td>0xFFFFFFFF</td>
</tr>
</tbody>
</table>

In this case, Device 0 asserts ACTION\_3 and Device 1 asserts ACTION\_1 since the conditions are met. As a result of this, the software trigger of both devices can be “simultaneously” triggered even though they are associated to different action numbers.

22.10.2.3 Writing Code Using Action Commands

The following section uses C# code snippets but the same thing can be done using a variety of other programming languages as well.
To set up an action command on the device something like this is needed:

```csharp
private static void setupActionCommandOnDevice(GenICam.ActionControl ac, Int64 deviceKey, Int64 actionNumber, Int64 groupKey, Int64 groupMask)
{
    if ((deviceKey == 0) && (groupKey == 0) && (groupMask == 0))
    {
        Console.WriteLine("Switching off action {0}.", actionNumber);
    }
    else
    {
        Console.WriteLine("Setting up action {0}. Device key: 0x{1:X8}, group key: 0x{2:X8}, group mask: 0x{3:X8}.", actionNumber, deviceKey, groupKey, groupMask);
        ac.actionDeviceKey.write(deviceKey);
        ac.actionSelector.write(actionNumber);
        ac.actionGroupKey.write(groupKey);
        ac.actionGroupMask.write(groupMask);
    }
}
```

**Note**

In a typical scenario the `deviceKey` parameter will only be set up once as there is only one register for it on each device while there can be various action commands.

Now to send action commands to devices connected to a certain interface it might be necessary to locate the correct instance of the `InterfaceModule` class first. One way to do this would be like this:

```csharp
private static List<GenICam.InterfaceModule> getGenTLInterfaceListForDevice(Device pDev)
{
    // first get a list of ALL interfaces in the current system
    GenICam.SystemModule systemModule = new GenICam.SystemModule(pDev);
    Dictionary<String, Int64> interfaceIDToIndexMap = new Dictionary<string, long>();
    Int64 interfaceCount = systemModule.interfaceSelector.maxValue + 1;
    for (Int64 i = 0; i < interfaceCount; i++)
    {
        systemModule.interfaceSelector.write(i);
        interfaceIDToIndexMap.Add(systemModule.interfaceID.read(), i);
    }
    // now try to get access to the interfaces the device in question is connected to
    PropertyI64 interfaceID = new PropertyI64();
    DeviceComponentLocator locator = new DeviceComponentLocator(pDev.hDev);
    locator.bindComponent(interfaceID, "InterfaceID");
    if (interfaceID.isValid == false)
    {
        return null;
    }
    ReadOnlyCollection<String> interfacesTheDeviceIsConnectedTo = interfaceID.listOfValidStrings;
    // create an instance of the GenICam.InterfaceModule class for each interface the device is connected to
    List<GenICam.InterfaceModule> interfaces = new List<GenICam.InterfaceModule>();
    foreach (String interfaceIDString in interfacesTheDeviceIsConnectedTo)
    {
        interfaces.Add(new GenICam.InterfaceModule(pDev, interfaceIDToIndexMap[interfaceIDString]));
    }
    return interfaces;
}
```

Once the desired interface has been located it could be configured to send an action command like this:

```csharp
private static void setupActionCommandOnInterface(GenICam.InterfaceModule im, Int32 destinationIP, Int64 deviceKey, Int64 groupKey, Int64 groupMask, bool boScheduledAction)
{
    im.mvActionDestinationIPAddress.write(destinationIP);
    im.mvActionDeviceKey.write(deviceKey);
    im.mvActionGroupKey.write(groupKey);
    im.mvActionGroupMask.write(groupMask);
    im.mvActionScheduledTimeEnable.write(boScheduledAction ? TBoolean.bTrue : TBoolean.bFalse);
    // here the desired execution time must also be configured for scheduled action commands if desired by writing
}```
Now the interface is set up completely and sending an action command works like this:

```java
private static void sendActionCommand(GenICam.InterfaceModule im)
{
    im.mvActionSend.call();
}
```

Depending on the value of `destinationIP` when actually firing the action command either a broadcast or a unicast message will be generated and send to either one or all devices in the subnet and depending on whether one or more action commands on the device are set up to assert the command they will react appropriately or will silently ignore the command.

### 22.10.3 Creating synchronized acquisitions using timers

#### 22.10.3.1 Basics

Getting images from several cameras exactly at the same time is a major task in

- 3D image acquisitions
  (the images must be acquired at the same time using two cameras) or
- acquisitions of larger objects
  (if more than one camera is required to span over the complete image, like in the textile and printing industry).

To solve this task, the mvBlueCOUGAR-X offers timers that can be used to generate pulse at regular intervals. This pulse can be connected to a digital output. The digital output can be connected digital to the digital input of one or more cameras to use it as a trigger.

#### 22.10.3.2 Connecting the hardware

**Note**

We recommend to use the cable "KS-BCX-HR12" to connect an external power supply (at pin 2 and 10) and the I/Os.

One camera is used as master (M), which generates the trigger signal. The other ones receive the trigger signal and act as slaves (S).

#### 22.10.3.2.1 On the master camera

1. Connect power supply GND to "pin 1" and "pin 7".
2. Connect power supply +12 V to "pin 2" and "pin 10 (power supply for the outputs)".
3. Connect DigOut0 of master ("pin 6") to DigIn0 of master ("pin 4").
22.10 Working with several cameras simultaneously

22.10.3.2.2 On each slave camera

1. Connect power supply GND to "pin 1" and "pin 7".
2. Connect power supply +12 V to "pin 2" and "pin 10 (power supply for the outputs)".

22.10.3.2.3 Between the cameras

1. Connect DigOut0 of master ("pin 6") to DigIn0 of slave ("pin 4").

Note
If more than one slave is used, connect the same "pin 6" of master to all "pin 4" of the slaves. If each camera has its own power supply, then connect all grounds (GND) together.

For the master camera, there are 2 possibilities how it is triggered:

1. The master camera triggers itself logically (so called "Master - Slave", see Figure 1), or
2. the master camera uses the external trigger signal, which was created by itself, via digital input (so called "Slave - Slave", see Figure 2).

![Figure 1: Master - Slave connecting](image1)

![Figure 2: Slave - Slave connecting](image2)

Note
With "Master - Slave" and according to the delay of the opto-isolated inputs of the slave cameras, you have to adapted the property "Trigger delay" of the master camera to synchronize the cameras exactly.

MATRIX VISION GmbH
22.10.3.3 Programming the acquisition

You will need two timers and you have to set a trigger.

22.10.3.3.1 Start timer

Two timers are used for the "start timer". Timer1 defines the interval between two triggers. Timer2 generates the trigger pulse at the end of Timer1.

The following sample shows a trigger

- which is generated every second and
- the pulse width is 10 ms:

```c
#include <mvIMPACT_CPP/mvIMPACT_acquire.h>
#include <mvIMPACT_CPP/mvIMPACT_acquire_GenICam.h>
...

// Master: Set timers to trig image: Start after queue is filled
GenICam::CounterAndTimerControl catcMaster(pDev);
catcMaster.timerSelector.writeS( "Timer1" );
catcMaster.timerDelay.write( 0. );
catcMaster.timerDuration.write( 1000000. );
catcMaster.timerTriggerSource.writeS( "Timer1End" );

catcMaster.timerSelector.writeS( "Timer2" );
catcMaster.timerDelay.write( 0. );
catcMaster.timerDuration.write( 10000. );
catcMaster.timerTriggerSource.writeS( "Timer1End" );
```

See also

**Counter And Timer Control** (p. 173)

**Note**

Make sure the Timer1 interval must be larger than the processing time. Otherwise, the images are lost.

The timers are defined, now you have to do following steps:

1. Set the digital output, e.g. "Line 0",
2. connect the digital output with the inputs of the slave cameras (and master camera if using "Slave - Slave"),
   and finally
3. set the trigger source to the digital input, e.g. "Line4".

22.10.3.3.2 Set digital I/O

In this step, the signal has to be connected to the digital output, e.g. "Line0":

```c
// Set Digital I/O
GenICam::DigitalIOControl io(pDev);
io.lineSelector.writeS( "Line0" );
io.lineSource.writeS( "Timer2Active" );
```

See also

**Digital I/O Control** (p. 181)

This signal has to be connected with the digital inputs of the slave cameras as shown in Figure 1 and 2.
22.10 Working with several cameras simultaneously

22.10.3.3 Set trigger

"If you want to use Master - Slave":

```cpp
// Set Trigger of Master camera
GenICam::AcquisitionControl ac(pDev);
ac.triggerSelector.writeS( "FrameStart" );
ac.triggerMode.writeS( "On" );
ac.triggerSource.writeS( "Timer1Start" );
// or ac.triggerSource.writeS( "Timer1End" );
```

```cpp
// Set Trigger of Slave camera
GenICam::AcquisitionControl ac(pDev);
ac.triggerSelector.writeS( "FrameStart" );
ac.triggerMode.writeS( "On" );
ac.triggerSource.writeS( "Line4" );
ac.triggerActivation.writeS( "RisingEdge" );
```

"If you want to use Slave - Slave":

```cpp
// Set Trigger of Master and Slave camera
GenICam::AcquisitionControl ac(pDev);
ac.triggerSelector.writeS( "FrameStart" );
ac.triggerMode.writeS( "On" );
ac.triggerSource.writeS( "Line4" );
ac.triggerActivation.writeS( "RisingEdge" );
```

See also

Acquisition Control (p. 170)

Now, the two timers will work like the following figure illustrates, which means

- **Timer1** is the trigger event and
- **Timer2** the trigger pulse width:

![Timers](image)

**Figure 3: Timers**

By the way, this is a simple "pulse width modulation (PWM)" example.

---

MATRIX VISION GmbH
22.10.3.4 Setting the synchronized acquisition using wxPropView

The following figures show, how you can set the timers and trigger using the GUI tool **wxPropView** (p. 115)

1. Setting of *Timer1* (blue box) on the master camera:

   ![Figure 4: wxPropView - Setting of Timer1 on the master camera](image)

2. Setting of *Timer2* (purple box) on the master camera:
3. Setting the trigger slave camera(s)
   - The red box in Figure 4 is showing "Slave - Slave", which means that both master and slave camera are connected via digital input.
   - The red box in Figure 6 is showing "Master - Slave"), which means that the master is triggered internally and the slave camera is set as shown in Figure 4.

4. Assigning timer to DigOut (orange box in Figure 4).

22.10.4 Using the primary application switchover functionality

There are scenarios where a second application should take control over a device that is already under control of another (primary) application (e.g. systems which requires redundancy, fault recovery or systems with a higher level management entity).

The switchover procedure will look like this: The primary application

1. requests (and gets granted) exclusive access,
2. verifies that the device supports switchover consulting GVCP capability register,
3. sets the control switchover key register,
4. requests (and gets granted) control access with enabled switchover (this is done without closing the control channel),

Another application that knows the key, can request (and gets granted) device control.

You can enable the switchover via "Device -> PrimaryApplicationSwitchoverEnable". Set this register to "On" to allow other applications to take control over the device.
If the control access has been granted, "DesiredAccess", "PrimaryApplicationSwitchoverEnable" and "PrimaryApplicationSwitchoverKey" will become read-only.

Now, in "Setting -> Base -> Camera -> GenICam -> Transport Layer Control" the property "Gev Primary Application Switchover Key" can be used by the control application to define a value that must be specified by an application that wants to take over control over the device. E.g. "666":

```
Figure 1: wxPropView - Device PrimaryApplicationSwitchoverEnable
```
The other application now tries to take over control with the correct switchover key and this access is granted. As a result the first application can no longer write to the device (executing "int timestampLatch()" fails with "DEV_ACCESS_DENIED").
Figure 3: wxPropView - Application Switchover

Note

If the other application tries to take over control without specifying the correct switchover key, an "DEV_ACCESS_DENIED" error will appear.

22.10.4.1 Code samples

The following code samples show, how to create functions which

- bind and set a property,
- how to configure the switchover access and
- how to take over the control.

```cpp
//template<typename _Ty, typename _Tx>
void bindAndSetProperty( _Ty& prop, ComponentLocatorBase& locator, const string& propName, const _Tx& value )
{
    locator.bindComponent( prop, propName );
    if( prop.isValid() )
```
try {
    prop.write( value );
    cout << "Property '"" << propName << '"' set to '"' << prop.readS() << '"'." << endl; // read back the value
} catch( const ImpactAcquireException& e ) {
    cout << "Failed to write '"" << value << '"' to property '"' << propName << '"'. Error message: " << e.getErrorString() << endl;
    displayPropertyData( prop );
} else {
    cout << "Property '"" << propName << '"' is not supported by this device or could not be located." << endl;
}

//------------------------------------------------------------------------------
void configureControlAccessWithSwitchoverEnabled( Device* pDev, int64_type key )
//------------------------------------------------------------------------------
{
    try {
        ComponentLocator locator(pDev->hDev());
        PropertyIBoolean primaryApplicationSwitchoverEnable;
        bindAndSetProperty( primaryApplicationSwitchoverEnable, locator, "PrimaryApplicationSwitchoverEnable", bTrue );
        pDev->interfaceLayout.write( dilGenICam );
        pDev->desiredAccess.write( damControl );
        pDev->open();
        DeviceComponentLocator devLocator(pDev, dltSetting);
        PropertyI64 gevPrimaryApplicationSwitchoverKey;
        bindAndSetProperty( gevPrimaryApplicationSwitchoverKey, locator, "GevPrimaryApplicationSwitchoverKey", key );
    } catch( const ImpactAcquireException& e ) {
        cout << string(__FUNCTION__) << ": " << e.getErrorString() << "(" << e.getErrorCodeAsString() << ") occurred at line " << e.getErrorOrigin() << endl;
    }
}

//------------------------------------------------------------------------------
void takeOverControlAccess( Device* pDev, int64_type key, bool boKeepSwitchoverPossible )
//------------------------------------------------------------------------------
{
    try {
        ComponentLocator locator(pDev->hDev());
        PropertyIBoolean primaryApplicationSwitchoverEnable;
        bindAndSetProperty( primaryApplicationSwitchoverEnable, locator, "PrimaryApplicationSwitchoverEnable", boKeepSwitchoverPossible ? bTrue : bFalse );
        PropertyI64 primaryApplicationSwitchoverKey;
        bindAndSetProperty( primaryApplicationSwitchoverKey, locator, "PrimaryApplicationSwitchoverKey", key );
        pDev->interfaceLayout.write( dilGenICam );
        pDev->desiredAccess.write( damControl );
        pDev->open();
    } catch( const ImpactAcquireException& e ) {
        cout << string(__FUNCTION__) << ": " << e.getErrorString() << "(" << e.getErrorCodeAsString() << ") occurred at line " << e.getErrorOrigin() << endl;
    }
}
23 Appendix A.1 CCD specific camera / sensor data

23.1 mvBlueCOUGAR-Xx20a (0.3 Mpix [640 x 480])

23.1.1 Introduction

The CCD sensor is a highly programmable imaging module which will, for example, enable the following type of applications

**Industrial applications:**

- triggered image acquisition with precise control of image integration start by hardware trigger input.
- image acquisition of fast moving objects due to:
  - frame integration, integrating all pixels at a time in contrast to CMOS imager which typically integrate line-by-line.
  - short shutter time, to get sharp images.
  - flash control output to have enough light for short time.

**Scientific applications:**

- long time integration for low light conditions.
- optimizing image quality using the variable shutter control.

23.1.2 Details of operation

The process of getting an image from the CCD sensor can be separated into three different phases.

23.1.2.1 Trigger

When coming out of reset or ready with the last readout the CCD controller is waiting for a Trigger signal.

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Setting (GenICam (p. 143))</th>
<th>Mode / Setting (Device Specific (p. 143))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = Off&quot;</td>
<td>Continuous</td>
<td>Free running, no external trigger signal needed.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = Software&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnDemand</td>
<td>Image acquisition triggered by command (software trigger).</td>
</tr>
</tbody>
</table>

To trigger one frame execute the TriggerSoftware@i command then.
<table>
<thead>
<tr>
<th>TriggerSelector</th>
<th>OnLowLevel</th>
<th>Start an exposure of a frame as long as the trigger input is below the trigger threshold. (No Frame---Trigger!)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AcquisitionActive</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerMode = On</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerSource = &lt;desired Line&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerActivation = LevelLow&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ExposureMode = Timed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerSelector =</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AcquisitionActive</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerMode = On</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerSource = &lt;desired Line&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerActivation = LevelHigh&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ExposureMode = Timed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerSelector =</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FrameStart</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerMode = On</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerSource = &lt;desired Line&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerActivation = FallingEdge&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ExposureMode = Timed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerSelector =</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FrameStart</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerMode = On</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerSource = &lt;desired Line&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerActivation = RisingEdge&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ExposureMode = Timed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerSelector = FrameStart&quot;</td>
<td></td>
<td>Each falling edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>TriggerMode = On</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerSource = &lt;desired Line&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerActivation = LevelHigh&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ExposureMode = Timed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerSelector = FrameStart&quot;</td>
<td></td>
<td>Each rising edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>TriggerMode = On</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerSource = &lt;desired Line&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerActivation = LevelLow&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ExposureMode = TriggerWidth&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerSelector = FrameStart&quot;</td>
<td></td>
<td>Each rising edge of trigger signal acquires one image, exposure time corresponds to pulse width.</td>
</tr>
<tr>
<td>TriggerMode = On</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerSource = &lt;desired Line&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerActivation = LevelHigh&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ExposureMode = TriggerWidth&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerSelector = FrameStart&quot;</td>
<td></td>
<td>Each falling edge of trigger signal acquires one image, exposure time corresponds to pulse width.</td>
</tr>
<tr>
<td>TriggerMode = On</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerSource = &lt;desired Line&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerActivation = LevelLow&quot;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
"TriggerSelector = FrameStart"
"TriggerMode = On"
"TriggerSource = <desired Line>"
"TriggerActivation = AnyEdge"
"ExposureMode = Timed"

OnAnyEdge
Start the exposure of a frame when the trigger input level changes from high to low or from low to high.

Line Mapping (TriggerSource mvIMPACT Acquire -> TriggerSource GenICam (valid values for <desired Line>)):

<table>
<thead>
<tr>
<th>TriggerSource mvIMPACT Acquire</th>
<th>TriggerSource GenICam(BCX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP-IN0</td>
<td>Line4</td>
</tr>
<tr>
<td>GP-IN1</td>
<td>Line5</td>
</tr>
</tbody>
</table>

See also
For detailed description about the trigger modes (https://www.matrix-vision/manuals/ [mvIMPACT Acquire API])

- C: TCameraTriggerMode
- C++: mvIMPACT::acquire::TCameraTriggerMode

**Note**
Trigger modes which use an external input (ctmOnLowLevel, ctmOnHighLevel, ctmOnRisingEdge, ctmOnFallingEdge) will use digital input 0 as input for the trigger signal. Input 0 is not restricted to the trigger function. It can always also be used as general purpose digital input. The input switching threshold of all inputs can be programmed with write_dac(level_in_mV). The best is to set this to the half of the input voltage. So for example if you apply a 24V switching signal to the digital inputs set the threshold to 12000 mV.

23.1.2.2 Exposure aka Integration

After an active trigger, the integration phase starts with a maximum jitter of \( t_{\text{trig}} \). If flash illumination is enabled in software the flash output will be activated exactly while the sensor chip is integrating light. Exposure time is adjustable by software in increments of \( t_{\text{readline}} \).

23.1.2.3 Readout

When integration is finished, the image is transferred to hidden storage cells on the CCD. Image data is then shifted out line-by-line and transferred to memory. Shifting out non active lines takes \( t_{\text{shift}} \), while shifting out active lines will consume \( t_{\text{readline}} \). The number of active pixels per line will not have any impact on readout speed.
23.1.3 CCD Timing

### Trigger Overlap: No

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Pixel clock</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_{trig}$</td>
<td>Time from trigger (internal or external) to exposure start</td>
<td>24.54 MHz</td>
</tr>
<tr>
<td>$t_{wait}$</td>
<td>Time after a new trigger is accepted</td>
<td>32 MHz</td>
</tr>
<tr>
<td>$t_{trans}$</td>
<td>Image transfer time (move image to readout cells in CCD)</td>
<td>40 MHz</td>
</tr>
<tr>
<td>$t_{readline}$</td>
<td>Time needed to readout a line</td>
<td>7.99us</td>
</tr>
<tr>
<td>$t_{vshift}$</td>
<td>Time needed to shift unused lines away</td>
<td>6.13us</td>
</tr>
<tr>
<td>$t_{exposure}$</td>
<td>Exposure time</td>
<td>4.90us</td>
</tr>
<tr>
<td>$t_{readout}$</td>
<td>Image readout time (move image from readout cells to memory)</td>
<td>31.95us</td>
</tr>
</tbody>
</table>

**Note**

In the "Trigger Overlap ReadOut" mode, the camera accepts a trigger signal the $t_{exposure}$ period earlier than $t_{readout}$ is finished.

23.1.3.1 Timings

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Pixel clock</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_{trig}$</td>
<td>Time from trigger (internal or external) to exposure start</td>
<td>24.54 MHz</td>
</tr>
<tr>
<td>$t_{wait}$</td>
<td>Time after a new trigger is accepted</td>
<td>32 MHz</td>
</tr>
<tr>
<td>$t_{trans}$</td>
<td>Image transfer time (move image to readout cells in CCD)</td>
<td>40 MHz</td>
</tr>
<tr>
<td>$t_{readline}$</td>
<td>Time needed to readout a line</td>
<td>7.99us</td>
</tr>
<tr>
<td>$t_{vshift}$</td>
<td>Time needed to shift unused lines away</td>
<td>6.13us</td>
</tr>
<tr>
<td>$t_{exposure}$</td>
<td>Exposure time</td>
<td>4.90us</td>
</tr>
<tr>
<td>$t_{readout}$</td>
<td>Image readout time (move image from readout cells to memory)</td>
<td>31.95us</td>
</tr>
</tbody>
</table>

**Note**

In partial scan mode (readout window ysize < 480 lines).

To calculate the maximum frames per second (FPS$_{max}$) you will need following formula (Expose mode: No overlap):

$$FPS_{max} = \frac{1}{t_{readout} + t_{exposure}}$$
(Expose mode: Overlapped):

\[
\begin{align*}
\text{t\textsubscript{readout}} & \text{ < t\textsubscript{exposure}}: \quad \text{FPS\textsubscript{max}} = \frac{1}{\text{t\textsubscript{exposure}}} \\
\text{t\textsubscript{readout}} & \text{ > t\textsubscript{exposure}}: \quad \text{FPS\textsubscript{max}} = \frac{1}{\text{t\textsubscript{readout}}} 
\end{align*}
\]

23.1.3.1 Example: Frame rate as function of lines & exposure time

Now, when we insert the values using exposure time of, for example, 8000 us, 480 lines and 40MHz pixel clock (Expose mode: No overlap):

\[
\text{FPS\textsubscript{max}} = \frac{1}{((480 + 1) \times 19.60 \text{ us}) + ((504 - 480) \times 1.80 \text{ us}) + 4.90 \text{ us}) + 8000 \text{ us}}
= 0.00005722231441372878 \text{ 1 / us}
= 57.2
\]

23.1.3.2 Frame rate calculator

Note

The calculator returns the max. frame rate supported by the sensor. Please keep in mind that it will depend on the interface and the used image format if this frame rate can be transferred.

23.1.4 Reprogramming CCD Timing

Reprogramming the CCD Controller will happen when the following changes occur

- Changing the exposure time
- Changing the capture window
- Changing Trigger Modes

Reprogram time consists of two phases

1. Time needed to send data to the CCD controller depending on what is changed
   - exposure: abt 2.3ms
   - window: abt 4.6ms
   - trigger mode: from 5.90ms, varies with oldmode/newmode combination
2. Time to initialize (erase) the CCD chip after reprogramming this is fixed, abt 4.5 ms

So for example when reprogramming the capture window you will need (average values)

\[
\text{t\textsubscript{regprog}} = \text{change\_window} + \text{init\_ccd}
\]

\[
\begin{align*}
\text{t\textsubscript{regprog}} &= 5\text{ms} + 4.5\text{ms} \\
\text{t\textsubscript{regprog}} &= 9.5\text{ms}
\end{align*}
\]

MATRIX VISION GmbH
23.1.5  CCD Sensor Data

Device Structure

- Interline CCD image sensor
- Image size: Diagonal 6mm (Type 1/3)
- Number of effective pixels: 659 (H) x 494 (V) approx. 330K pixels
- Total number of pixels: 692 (H) x 504 (V) approx. 350K pixels
- Chip size: 5.79mm (H) x 4.89mm (V)
- Unit cell size: 7.4um (H) x 7.4um (V)
- Optical black:
  - Horizontal (H) direction: Front 2 pixels, rear 31 pixels
  - Vertical (V) direction: Front 8 pixels, rear 2 pixels
- Number of dummy bits: Horizontal 16 Vertical 5
- Substrate material: Silicon

23.1.5.1  Characteristics

These zone definitions apply to both the color and gray scale version of the sensor.
23.1.5.2 Color version

Image Sensor Characteristics  
(Ta = 25°C)

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Unit</th>
<th>Measurement method</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>G Sensitivity</td>
<td>Sg</td>
<td>600</td>
<td>750</td>
<td></td>
<td>mV</td>
<td>1</td>
<td>1/30s accumulation</td>
</tr>
<tr>
<td>Sensitivity comparison</td>
<td>Rr</td>
<td>0.4</td>
<td>0.55</td>
<td>0.7</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Rb</td>
<td>0.3</td>
<td>0.45</td>
<td>0.6</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Saturation signal</td>
<td>Vsat</td>
<td>500</td>
<td></td>
<td></td>
<td>mV</td>
<td>2</td>
<td>Ta = 60°C</td>
</tr>
<tr>
<td>Smear</td>
<td>Sm</td>
<td>−100</td>
<td>−92</td>
<td></td>
<td>dB</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Video signal shading</td>
<td>SHg</td>
<td></td>
<td></td>
<td></td>
<td>%</td>
<td>4</td>
<td>Zone 0 and I</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>%</td>
<td>4</td>
<td>Zone 0 to 11'</td>
</tr>
<tr>
<td>Uniformity between</td>
<td>ΔSrG</td>
<td></td>
<td></td>
<td></td>
<td>%</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>video signal channels</td>
<td>ΔSbg</td>
<td></td>
<td></td>
<td></td>
<td>%</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Dark signal</td>
<td>Vdt</td>
<td></td>
<td></td>
<td></td>
<td>mV</td>
<td>6</td>
<td>Ta = 60°C</td>
</tr>
<tr>
<td>Dark signal shading</td>
<td>ΔVdt</td>
<td></td>
<td></td>
<td>0.5</td>
<td>mV</td>
<td>7</td>
<td>Ta = 60°C</td>
</tr>
<tr>
<td>Line crawl G</td>
<td>Lcg</td>
<td>3.8</td>
<td></td>
<td></td>
<td>%</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Line crawl R</td>
<td>Lcr</td>
<td>3.8</td>
<td></td>
<td></td>
<td>%</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Line crawl B</td>
<td>Lcb</td>
<td>3.8</td>
<td></td>
<td></td>
<td>%</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Lag</td>
<td>Lag</td>
<td>0.5</td>
<td></td>
<td></td>
<td>%</td>
<td>9</td>
<td></td>
</tr>
</tbody>
</table>

Spectral Sensitivity Characteristics  
(Includes lens characteristics, excludes light source characteristics)
23.1.5.3 Gray scale version

### Image Sensor Characteristics

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Unit</th>
<th>Measurement method</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>S</td>
<td>700</td>
<td>880</td>
<td>mV</td>
<td>1</td>
<td>1/30s accumulation conversion value</td>
<td></td>
</tr>
<tr>
<td>Saturation signal</td>
<td>V_{sat}</td>
<td>500</td>
<td></td>
<td>mV</td>
<td>2</td>
<td>Ta = 60°C</td>
<td></td>
</tr>
<tr>
<td>Smear</td>
<td>Sm</td>
<td>-100</td>
<td>-92</td>
<td>dB</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Video signal shading</td>
<td>SH</td>
<td>20</td>
<td>%</td>
<td>4</td>
<td>Zone 0, 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dark signal</td>
<td>V_{dt}</td>
<td>2</td>
<td>mV</td>
<td>5</td>
<td>Ta = 60°C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dark signal shading</td>
<td>\Delta V_{dt}</td>
<td>0.5</td>
<td>mV</td>
<td>6</td>
<td>Ta = 60°C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lag</td>
<td>Lag</td>
<td>0.5</td>
<td>%</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Spectral Sensitivity Characteristics

(Excludes lens characteristics and light source characteristics)
Measurement conditions

1) In the following measurements, the device drive conditions are at the typical values of the bias and clock voltage conditions.

2) In the following measurements, spot blemishes are excluded and, unless otherwise specified, the optical black level (OB) is used as the reference for the signal output, which is taken as the value measured at point [OB] of the measurement system.

Definition of standard imaging conditions

1) Standard imaging condition I:
   Use a pattern box (luminance: 706cd/m², color temperature of 3200K halogen source) as a subject. (pattern for evaluation is not applicable.) Use a testing standard lens with CM505S (t = 1.0mm) as an IR cut filter and image at F8. The luminous intensity to the sensor receiving surface at this point is defined as the standard sensitivity testing luminous intensity.

2) Standard imaging condition II:
   Image a light source (color temperature of 3200K) with a uniformity of brightness within 2% at all angles. Use a testing standard lens with CM505S (t = 1.0mm) as an IR cut filter. The luminous intensity is adjusted to the value indicated in each testing item by the lens diaphragm.

1. Sensitivity
   Set to standard imaging condition I. After selecting the electronic shutter mode with a shutter speed of 1/100s, measure the signal output (Vs) at the center of the screen, and substitute the value into the following formula.

   \[ S = Vs \times \frac{250}{30} \ [mV] \]

2. Saturation signal
   Set to standard imaging condition II. After adjusting the luminous intensity to 10 times the intensity with the average value of the signal output, 150mV, measure the minimum value of the signal output.

3. Smear
   Set to standard imaging condition II. With the lens diaphragm at F5.6 to F8, first adjust the luminous intensity to 500 times the intensity with the average value of signal output, 150mV. Then after the readout clock is stopped and the charge drain is executed by the electronic shutter at the respective H blankings, measure the maximum value (VSm [mV]) of the signal output and substitute the value into the following formula.

   \[ Sm = 20 \times \log \left( \frac{VSm}{150} \times \frac{1}{500} \times \frac{1}{10} \right) \ [dB] \ (1/10V \ method \ conversion \ value) \]

4. Video signal shading
   Set to standard imaging condition II. With the lens diaphragm at F5.6 to F8, adjust the luminous intensity so that the average value of the signal output is 150mV. Then measure the maximum (Vmax [mV]) and minimum (Vmin [mV]) values of the signal output and substitute the values into the following formula.

   \[ SH = (Vmax - Vmin)/150 \times 100 \ [%] \]

5. Dark signal
   Measure the average value of the signal output (Vdt [mV]) with the device ambient temperature 60°C and the device in the light-obstructed state, using the horizontal idle transfer level as a reference.
6. Dark signal shading
After measuring 5, measure the maximum (Vdmax [mV]) and minimum (Vdmin [mV]) values of the dark signal output and substitute the values into the following formula.

\[ \Delta Vdt = Vdmax - Vdmin \text{[mV]} \]

7. Lag
Adjust the signal output value generated by strobe light to 150mV. After setting the strobe light so that it strobes with the following timing, measure the residual signal (Vlag). Substitute the value into the following formula.

\[ \text{Lag} = \left( \frac{Vlag}{150} \right) \times 100 \% \]

---

**Measurement System**

**Note** Adjust the amplifier gain so that the gain between \[^A\] and \[^B\] equals 1.

23.1.6 Device Feature And Property List

23.1.6.1 Gray scale version

23.1.6.2 Color version
23.2  mvBlueCOUGAR-Xx20b (0.3 Mpix [640 x 480])

23.2.1  Introduction

The CCD sensor is a highly programmable imaging module which will, for example, enable the following type of applications

**Industrial applications:**

- triggered image acquisition with precise control of image exposure start by hardware trigger input.
- image acquisition of fast moving objects due to:
  - frame exposure, integrating all pixels at a time in contrast to CMOS imager which typically integrate line-by-line.
  - short shutter time, to get sharp images.
  - flash control output to have enough light for short time.

**Scientific applications:**

- long time exposure for low light conditions.
- optimizing image quality using the variable shutter control.

23.2.2  Details of operation

The process of getting an image from the CCD sensor can be separated into three different phases.

23.2.2.1  Trigger

When coming out of reset or ready with the last readout the CCD controller is waiting for a Trigger signal.

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Setting (GenICam) (p.143)</th>
<th>Mode / Setting (Device Specific) (p.143)</th>
<th>Description</th>
</tr>
</thead>
</table>
| "TriggerSelector = FrameStart"  
  "TriggerMode = Off" | Continuous | Free running, no external trigger signal needed. |
| "TriggerSelector = FrameStart"  
  "TriggerMode = On"  
  "TriggerSource = Software"  
  "ExposureMode = Timed" | OnDemand | Image acquisition triggered by command (software trigger). |

To trigger one frame execute the `TriggerSoftware@i` command then.
<table>
<thead>
<tr>
<th>TriggerSelector = AcquisitionActive</th>
<th>OnLowLevel</th>
<th>Start an exposure of a frame as long as the trigger input is below the trigger threshold. (No Frame←Trigger!)</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerMode = On&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSource = &lt;desired Line&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerActivation = LevelLow&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;ExposureMode = Timed&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot;</td>
<td>OnHighLevel</td>
<td>Start an exposure of a frame as long as the trigger input is above the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerMode = On&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSource = &lt;desired Line&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerActivation = LevelHigh&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;ExposureMode = Timed&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot;</td>
<td>OnFallingEdge</td>
<td>Each falling edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>&quot;TriggerMode = On&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSource = &lt;desired Line&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerActivation = FallingEdge&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;ExposureMode = Timed&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot;</td>
<td>OnRisingEdge</td>
<td>Each rising edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>&quot;TriggerMode = On&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSource = &lt;desired Line&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerActivation = RisingEdge&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;ExposureMode = Timed&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot;</td>
<td>OnHighExpose</td>
<td>Each rising edge of trigger signal acquires one image, exposure time corresponds to pulse width.</td>
</tr>
<tr>
<td>&quot;TriggerMode = On&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSource = &lt;desired Line&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerActivation = LevelHigh&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;ExposureMode = TriggerWidth&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot;</td>
<td>OnLowExpose</td>
<td>Each falling edge of trigger signal acquires one image, exposure time corresponds to pulse width.</td>
</tr>
<tr>
<td>&quot;TriggerMode = On&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSource = &lt;desired Line&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerActivation = LevelLow&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;ExposureMode = TriggerWidth&quot;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
"TriggerSelector = FrameStart"
"TriggerMode = On"
"TriggerSource = <desired Line>"
"TriggerActivation = AnyEdge"
"ExposureMode = Timed"

OnAnyEdge
Start the exposure of a frame when the trigger input level changes from high to low or from low to high.

Line Mapping (TriggerSource mvIMPACT Acquire -> TriggerSource GenICam (valid values for <desired Line>)):

<table>
<thead>
<tr>
<th>TriggerSource mvIMPACT Acquire</th>
<th>TriggerSource GenICam(BCX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP-IN0</td>
<td>Line4</td>
</tr>
<tr>
<td>GP-IN1</td>
<td>Line5</td>
</tr>
</tbody>
</table>

23.2.2.2 Exposure aka Integration

After an active trigger, the exposure phase starts with a maximum jitter of t_{trig}. If flash illumination is enabled in software the flash output will be activated exactly while the sensor chip is integrating light. Exposure time is adjustable by software in increments of t_{readline}.

23.2.2.3 Readout

When exposure is finished, the image is transferred to hidden storage cells on the CCD. Image data is then shifted out line-by-line and transferred to memory. Shifting out non active lines takes t_{vshift}, while shifting out active lines will consume t_{readline}. The number of active pixels per line will not have any impact on readout speed.

23.2.3 CCD Timing

Note

In the "Trigger Overlap ReadOut" mode, the camera accepts a trigger signal the t_{exposure} period earlier than t_{readout} is finished.
23.2.3.1 Timings
### Table of Values

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Pixel clock</th>
</tr>
</thead>
<tbody>
<tr>
<td>t trig</td>
<td>Time from trigger (internal or external) to exposure start</td>
<td>24.54 MHz 32 MHz 40 MHz</td>
</tr>
<tr>
<td>t trig</td>
<td>t trig = t shift + t turn_on_delay_opto (t turn_on_delay_opto = approx. 2us)</td>
<td></td>
</tr>
<tr>
<td>t wait</td>
<td>Time after a new trigger is accepted</td>
<td>&lt; 1us</td>
</tr>
<tr>
<td>t trans</td>
<td>Image transfer time (move image to readout cells in CCD)</td>
<td>7.99us 6.13us 4.90us</td>
</tr>
<tr>
<td>t readline</td>
<td>Time needed to readout a line</td>
<td>31.95us 24.50us 19.60us</td>
</tr>
<tr>
<td>t vshift</td>
<td>Time needed to shift unused lines away</td>
<td>2.93us 2.25us 1.80us</td>
</tr>
<tr>
<td>t exposure</td>
<td>Exposure time</td>
<td>10us..20s 10us..20s 10us..20s</td>
</tr>
<tr>
<td>t readout</td>
<td>Image readout time (move image from readout cells to memory)</td>
<td>t readout = ((ActiveLines + 1) * t readline) + ((504 - ActiveLines) * t vshift) + t trans</td>
</tr>
</tbody>
</table>

### Note

In partial scan mode (readout window ysize < 480 lines).

To calculate the maximum frames per second (FPS\(_{\text{max}}\)) you will need following formula (Expose mode: No overlap):

\[
FPS_{\text{max}} = \frac{1}{t_{\text{readout}} + t_{\text{exposure}}}
\]

(Expose mode: Overlapped):

\[
\begin{align*}
\text{t readout} < \text{t exposure: } & \quad FPS_{\text{max}} = \frac{1}{t_{\text{exposure}}} \\
\text{t readout} > \text{t exposure: } & \quad FPS_{\text{max}} = \frac{1}{t_{\text{readout}}}
\end{align*}
\]

**23.2.3.1.1 Example: Frame rate as function of lines & exposure time (Expose mode: No overlap)**

Now, when we insert the values using exposure time of, for example, 8000 us, 480 lines and 40MHz pixel clock:

\[
FPS_{\text{max}} = \frac{1}{((480 + 1) * 19.60 \text{ us}) + ((504 - 480) * 1.80 \text{ us}) + 4.90 \text{ us}) + 8000 \text{ us}}
\]

\[
= \frac{1}{(9606.6 + 24.0 + 4.90 + 8000)}
\]

\[
= \frac{1}{17793.50}
\]

\[
= 0.0000572223141372878 \text{ 1 / us}}
\]

\[
= 57.2
\]

**23.2.4 Reprogramming CCD Timing**

Reprogramming the CCD Controller will happen when the following changes occur:

- Changing the exposure time
- Changing the capture window
- Changing trigger modes
Reprogram time consists of two phases

1. Time needed to send data to the CCD controller depending on what is changed
   - exposure: about 2.3 ms
   - window: about 4.6 ms
   - trigger mode: from 5.9 ms, varies with old mode/new mode combination

2. Time to initialize (erase) the CCD chip after reprogramming this is fixed, about 4.5 ms

So for example when reprogramming the capture window you will need (average values)

\[ t_{\text{regprog}} = \text{change\_window} + \text{init\_ccd} \]
\[ t_{\text{regprog}} = 5 \text{ms} + 4.5 \text{ms} \]
\[ t_{\text{regprog}} = 9.5 \text{ms} \]

23.2.5 CCD Sensor Data

Device Structure

- Interline CCD image sensor
- Image size: Diagonal 6 mm (Type 1/2)
- Number of effective pixels: 659 (H) x 494 (V) approx. 330 K pixels
- Total number of pixels: 692 (H) x 504 (V) approx. 350 K pixels
- Chip size: 7.48 mm (H) x 6.15 mm (V)
- Unit cell size: 9.9 um (H) x 9.9 um (V)
- Optical black:
  - Horizontal (H) direction: Front 2 pixels, rear 31 pixels
  - Vertical (V) direction: Front 8 pixels, rear 2 pixels
- Number of dummy bits: Horizontal 16 Vertical 5
- Substrate material: Silicon

23.2.5.1 Characteristics

These zone definitions apply to both the color and gray scale version of the sensor.
23.2.5.2 Color version

Image Sensor Characteristics  

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Unit</th>
<th>Measurement method</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>G Sensitivity</td>
<td>Sg</td>
<td>600</td>
<td>750</td>
<td></td>
<td>mV</td>
<td>1</td>
<td>1/30s accumulation</td>
</tr>
<tr>
<td>Sensitivity comparison</td>
<td>Rr</td>
<td>0.4</td>
<td>0.55</td>
<td>0.7</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Rb</td>
<td>0.3</td>
<td>0.45</td>
<td>0.6</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Saturation signal</td>
<td>Vsat</td>
<td>500</td>
<td></td>
<td></td>
<td>mV</td>
<td>2</td>
<td>Ta = 60°C</td>
</tr>
<tr>
<td>Smear</td>
<td>Sm</td>
<td></td>
<td>-100</td>
<td>-92</td>
<td>dB</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Video signal shading</td>
<td>SHg</td>
<td></td>
<td></td>
<td></td>
<td>%</td>
<td>4</td>
<td>Zone 0 and I</td>
</tr>
<tr>
<td>Uniformity between video signal channels</td>
<td>ΔSr</td>
<td>8</td>
<td></td>
<td></td>
<td>%</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ΔSbg</td>
<td>8</td>
<td></td>
<td></td>
<td>%</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Dark signal</td>
<td>Vdt</td>
<td>2</td>
<td></td>
<td></td>
<td>mV</td>
<td>6</td>
<td>Ta = 60°C</td>
</tr>
<tr>
<td>Dark signal shading</td>
<td>ΔVdt</td>
<td>0.5</td>
<td></td>
<td></td>
<td>mV</td>
<td>7</td>
<td>Ta = 60°C</td>
</tr>
<tr>
<td>Line crawl G</td>
<td>Lcg</td>
<td>3.8</td>
<td></td>
<td></td>
<td>%</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Line crawl R</td>
<td>Lcr</td>
<td>3.8</td>
<td></td>
<td></td>
<td>%</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Line crawl B</td>
<td>Lcb</td>
<td>3.8</td>
<td></td>
<td></td>
<td>%</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Lag</td>
<td>Lcg</td>
<td>0.5</td>
<td></td>
<td></td>
<td>%</td>
<td>9</td>
<td></td>
</tr>
</tbody>
</table>

Note: All image sensor characteristic data noted above is for operation in 1/60s progressive scan mode.

Spectral Sensitivity Characteristics (Includes lens characteristics, excludes light source characteristics)
23.2.5.3 Gray scale version

### Image Sensor Characteristics

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Unit</th>
<th>Measurement method</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>$S$</td>
<td>700</td>
<td>880</td>
<td>1150</td>
<td>mV</td>
<td>1</td>
<td>1/30s accumulation conversion value</td>
</tr>
<tr>
<td>Saturation signal</td>
<td>$V_{sat}$</td>
<td>500</td>
<td></td>
<td></td>
<td>mV</td>
<td>2</td>
<td>$Ta = 60^\circ C$</td>
</tr>
<tr>
<td>Smear</td>
<td>$S_m$</td>
<td>-100</td>
<td></td>
<td>-92</td>
<td>dB</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Video signal shading</td>
<td>$S_H$</td>
<td>25</td>
<td></td>
<td></td>
<td>%</td>
<td>4</td>
<td>Zone 0</td>
</tr>
<tr>
<td>Dark signal</td>
<td>$V_{dt}$</td>
<td>2</td>
<td></td>
<td></td>
<td>mV</td>
<td>5</td>
<td>$Ta = 60^\circ C$</td>
</tr>
<tr>
<td>Dark signal shading</td>
<td>$\Delta V_{dt}$</td>
<td>1</td>
<td></td>
<td></td>
<td>mV</td>
<td>6</td>
<td>$Ta = 60^\circ C$</td>
</tr>
<tr>
<td>Lag</td>
<td>$Lag$</td>
<td>0.5</td>
<td></td>
<td></td>
<td>%</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

**Note**: All image sensor characteristic data noted above is for operation in 1/60s progressive scan mode.

### Spectral Sensitivity Characteristics (Excludes lens characteristics and light source characteristics)

![Spectral Sensitivity Characteristics Graph](image)
**Measurement conditions**

1. In the following measurements, the substrate voltage is set to the value indicated on the device, and the device drive conditions are at the typical values of the bias and clock voltage conditions.

2. In the following measurements, spot blemishes are excluded and, unless otherwise specified, the optical black level (OB) is used as the reference for the signal output, which is taken as the value measured at point ["B"] of the measurement system.

3. In the following measurements, this image sensor is operated in 1/50s progressive scan mode.

**Definition of standard imaging conditions**

1. Standard imaging condition I:
   Use a pattern box (luminance: 706cd/m², color temperature of 3200K halogen source) as a subject (Pattern for evaluation is not applicable.) Use a testing standard lens with CM500S (t = 1.0mm) as an IR cut filter and image at F8. The luminous intensity to the sensor receiving surface at this point is defined as the standard sensitivity testing luminous intensity.

2. Standard imaging condition II:
   Image a light source (color temperature of 3200K) with a uniformity of brightness within 2% at all angles. Use a testing standard lens with CM500S (t = 1.0mm) as an IR cut filter. The luminous intensity is adjusted to the value indicated in each testing item by the lens diaphragm.

1. Sensitivity
   Set to standard imaging condition I. After setting the electronic shutter mode with a shutter speed of 1/250s, measure the signal voltage (Vs) at the center of the screen, and substitute the value into the following formula.
   \[
   S = Vs \times \frac{250}{50} [mV]
   \]

2. Saturation signal
   Set to standard imaging condition II. After adjusting the luminous intensity to 10 times the intensity with the average value of the signal output, 150mV, measure the minimum value of the signal output.

3. Smear
   Set to standard imaging condition II. With the lens diaphragm at F5.6 to F8, first adjust the luminous intensity to 500 times the intensity with the average value of signal output, 150mV. Then after the readout clock is stopped and the charge drain is executed by the electronic shutter at the respective H blankings, measure the maximum value (VSm [mV]) of the signal output and substitute the value into the following formula.
   \[
   Sm = 20 \times \log \left( \frac{VSm}{150} \times \frac{1}{500} \times \frac{1}{10} \right) [dB] \text{ (1/10V method conversion value)}
   \]

4. Video signal shading
   Set to standard imaging condition II. With the lens diaphragm at F5.6 to F8, adjust the luminous intensity so that the average value of the signal output is 150mV. Then measure the maximum (Vmax [mV]) and minimum (Vmin [mV]) values of the signal output and substitute the values into the following formula.
   \[
   SH = (Vmax - Vmin) / 150 \times 100 \%
   \]
5. Dark signal

Measure the average value of the signal output (Vd [mV]) with the device ambient temperature 60°C and the device in the light-obstructed state, using the horizontal idle transfer level as a reference.

6. Dark signal shading

After measuring 5, measure the maximum (Vdmax [mV]) and minimum (Vdmin [mV]) values of the dark signal output and substitute the values into the following formula.

\[ \Delta V_d = V_{d\text{max}} - V_{d\text{min}} \text{ [mV]} \]

7. Lag

Adjust the signal output generated by strobe light to 150mV. After setting the strobe light so that it strobes with the following timing, measure the residual signal (\(V_{\text{lag}}\)). Substitute the value into the following formula.

\[ \text{Lag} = \left( \frac{V_{\text{lag}}}{150} \right) \times 100 \% \]

---

**Measurement System**

- **CCD signal output [A]**
  - **C.D.S**
  - **AMP**
  - **GnGo**
  - **GrGo channel signal output [B]**
  - **R/B**
  - **GnGo + R/B**
  - **SrH**
  - **GrGo channel signal output [C]**

**Note:** Adjust the amplifier gain so that the gain between [A] and [B], and between [A] and [C] equals 1.

---

23.2.6 Device Feature And Property List

23.2.6.1 Gray scale version

23.2.6.2 Color version
23.3 mvBlueCOUGAR-Xx20d (0.5 Mpix [776 x 580])

23.3.1 Introduction

The CCD sensor is a highly programmable imaging module which will, for example, enable the following type of applications:

**Industrial applications:**

- triggered image acquisition with precise control of image exposure start by hardware trigger input.
- image acquisition of fast moving objects due to:
  - frame exposure, integrating all pixels at a time in contrast to CMOS imager which typically integrate line-by-line.
  - short shutter time, to get sharp images.
  - flash control output to have enough light for short time.

**Scientific applications:**

- long time exposure for low light conditions.
- optimizing image quality using the variable shutter control.

23.3.2 Details of operation

The process of getting an image from the CCD sensor can be separated into three different phases.

23.3.2.1 Trigger

When coming out of reset or ready with the last readout the CCD controller is waiting for a Trigger signal.

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Setting (GenICam p.143)</th>
<th>Mode / Setting (Device Specific p.143)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = Off&quot;</td>
<td>Continuous</td>
<td>Free running, no external trigger signal needed.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = Software&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnDemand</td>
<td>Image acquisition triggered by command (software trigger).</td>
</tr>
</tbody>
</table>

To trigger one frame execute the `TriggerSoftware@i` command then.
<table>
<thead>
<tr>
<th>Trigger Configuration</th>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelLow&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnLowLevel</td>
<td>Start an exposure of a frame as long as the trigger input is below the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelHigh&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnHighLevel</td>
<td>Start an exposure of a frame as long as the trigger input is above the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = FallingEdge&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnFallingEdge</td>
<td>Each falling edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = RisingEdge&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnRisingEdge</td>
<td>Each rising edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelHigh&quot; &quot;ExposureMode = TriggerWidth&quot;</td>
<td>OnHighExpose</td>
<td>Each rising edge of trigger signal acquires one image, exposure time corresponds to pulse width.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelLow&quot; &quot;ExposureMode = TriggerWidth&quot;</td>
<td>OnLowExpose</td>
<td>Each falling edge of trigger signal acquires one image, exposure time corresponds to pulse width.</td>
</tr>
</tbody>
</table>
"TriggerSelector = FrameStart"
"TriggerMode = On"
"TriggerSource = <desired Line>"
"TriggerActivation = AnyEdge"
"ExposureMode = Timed"

| OnAnyEdge | Start the exposure of a frame when the trigger input level changes from high to low or from low to high. |

Line Mapping (TriggerSource mvIMPACT Acquire -> TriggerSource GenICam (valid values for <desired Line>):

<table>
<thead>
<tr>
<th>TriggerSource mvIMPACT Acquire</th>
<th>TriggerSource GenICam(BCX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP-IN0</td>
<td>Line4</td>
</tr>
<tr>
<td>GP-IN1</td>
<td>Line5</td>
</tr>
</tbody>
</table>

### 23.3.2.2 Exposure aka Integration

After an active trigger, the exposure phase starts with a maximum jitter of $t_{\text{trig}}$. If flash illumination is enabled in software the flash output will be activated exactly while the sensor chip is integrating light. Exposure time is adjustable by software in increments of $t_{\text{readline}}$.

### 23.3.2.3 Readout

When exposure is finished, the image is transferred to hidden storage cells on the CCD. Image data is then shifted out line-by-line and transferred to memory. Shifting out non active lines takes $t_{\text{vshift}}$, while shifting out active lines will consume $t_{\text{readline}}$. The number of active pixels per line will not have any impact on readout speed.

### 23.3.3 CCD Timing

Note

In the "Trigger Overlap ReadOut" mode, the camera accepts a trigger signal the $t_{\text{exposure}}$ period earlier than $t_{\text{readout}}$ is finished.
23.3.3.1 Timings
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Pixel clock</th>
</tr>
</thead>
<tbody>
<tr>
<td>t_trig</td>
<td>Time from trigger (internal or external) to exposure start</td>
<td>t_trig = t_vshift + t_turn_on_delay_opto ((t_{\text{turn on delay opto}} = \text{approx. 2us}))</td>
</tr>
<tr>
<td>t_wait</td>
<td>Time after a new trigger is accepted</td>
<td>&lt; 1us</td>
</tr>
<tr>
<td>t_trans</td>
<td>Image transfer time (move image to readout cells in CCD)</td>
<td>30.51us</td>
</tr>
<tr>
<td>t_readline</td>
<td>Time needed to readout a line</td>
<td>32us</td>
</tr>
<tr>
<td>t_vshift</td>
<td>Time needed to shift unused lines away</td>
<td>3.25us</td>
</tr>
<tr>
<td>t_exposure</td>
<td>Exposure time</td>
<td>10us..20s</td>
</tr>
<tr>
<td>t_readout</td>
<td>Image readout time (move image from readout cells to memory)</td>
<td>t_{\text{readout}} = \left( (\text{ActiveLines} + 1) \times t_{\text{readline}} \right) + \left( (592 - \text{ActiveLines}) \times t_{\text{vshift}} \right) + t_{\text{trans}}</td>
</tr>
</tbody>
</table>

**Note**

In partial scan mode (readout window ysize < 580 lines).

To calculate the maximum frames per second \((\text{FPS}_{\text{max}})\) you will need following formula (Exposure mode: No overlap):

\[
\text{FPS}_{\text{max}} = \frac{1}{t_{\text{readout}} + t_{\text{exposure}}}
\]

(Exposure mode: Overlapped):

- \(t_{\text{readout}} < t_{\text{exposure}}\): \[
\text{FPS}_{\text{max}} = \frac{1}{t_{\text{exposure}}}
\]
- \(t_{\text{readout}} > t_{\text{exposure}}\): \[
\text{FPS}_{\text{max}} = \frac{1}{t_{\text{readout}}}
\]

23.3.3.1 Example: Frame rate as function of lines & exposure time (Exposure mode: No overlap)

Now, when we insert the values using exposure time of, for example, 8000 us, 580 lines and 48MHz pixel clock:

\[
\text{FPS}_{\text{max}} = \frac{1}{\left( (\text{580} + 1) \times 19.67 \text{ us} \right) + \left( (\text{592} - \text{580}) \times 2 \text{ us} \right) + 18.75 \text{ us}} = 0.00008717620577769021 \text{ 1 / us} = 87.2
\]

23.3.4 Reprogramming CCD Timing

Reprogramming the CCD Controller will happen when the following changes occur:

- Changing the exposure time
- Changing the capture window
- Changing trigger modes
Reprogram time consists of two phases

1. Time needed to send data to the CCD controller depending on what is changed
   exposure: abt 2..3ms
   window: abt 4..6ms
   trigger mode: from 5..90ms,
   varies with oldmode/newmode combination

2. Time to initialize (erase) the CCD chip after reprogramming this is fixed, abt 4.5 ms

So for example when reprogramming the capture window you will need (average values)

\[ t_{\text{regprog}} = \text{change\_window} + \text{init\_ccd} \]

\[ t_{\text{regprog}} = 5ms + 4.5ms \]

\[ t_{\text{regprog}} = 9.5ms \]

23.3.5 CCD Sensor Data

Device Structure

- Interline CCD image sensor
- Image size: Diagonal 6mm (Type 1/2)
- Number of effective pixels: 782 (H) x 582 (V) approx. 460 K pixels
- Total number of pixels: 823 (H) x 592 (V) approx. 490 K pixels
- Chip size: 7.48mm (H) x 6.15mm (V)
- Unit cell size: 8.3um (H) x 8.3um (V)
- Optical black:
  - Horizontal (H) direction: Front 3 pixels, rear 38 pixels
  - Vertical (V) direction: Front 8 pixels, rear 2 pixels
- Number of dummy bits: Horizontal 19 Vertical 5
- Substrate material: Silicon
23.3.5.1 Characteristics

These zone definitions apply to both the color and gray scale version of the sensor.

![Zone Definition of Video Signal Shading](image)

23.3.5.2 Color version

### Image Sensor Characteristics

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Unit</th>
<th>Measurement method</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>G Sensitivity</td>
<td>Sg</td>
<td>570</td>
<td>720</td>
<td>940</td>
<td>mV</td>
<td>1</td>
<td>1/25s accumulation conversion value</td>
</tr>
<tr>
<td>Sensitivity comparison</td>
<td>Rr</td>
<td>0.4</td>
<td>0.55</td>
<td>0.7</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Rb</td>
<td>0.3</td>
<td>0.45</td>
<td>0.6</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Saturation signal</td>
<td>Vsat</td>
<td>375</td>
<td></td>
<td></td>
<td>mV</td>
<td>2</td>
<td>Ta = 60°C</td>
</tr>
<tr>
<td>Smear</td>
<td>Sm</td>
<td>-100</td>
<td>-92</td>
<td></td>
<td>dB</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Video signal shading</td>
<td>SHg</td>
<td>25</td>
<td></td>
<td></td>
<td>%</td>
<td>4</td>
<td>Zone 0</td>
</tr>
<tr>
<td>Uniformity between video signal channels</td>
<td>ΔSrg</td>
<td>8</td>
<td></td>
<td>%</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ΔShg</td>
<td>8</td>
<td></td>
<td>%</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dark signal</td>
<td>Vdt</td>
<td>2</td>
<td>mV</td>
<td></td>
<td></td>
<td>6</td>
<td>Ta = 60°C</td>
</tr>
<tr>
<td>Dark signal shading</td>
<td>ΔVdt</td>
<td>1</td>
<td>mV</td>
<td></td>
<td></td>
<td>7</td>
<td>Ta = 60°C</td>
</tr>
<tr>
<td>Lag</td>
<td>Lag</td>
<td>0.5</td>
<td>%</td>
<td></td>
<td></td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** All image sensor characteristic data noted above is for operation in 1/50s progressive scan mode.
23.3.5.3 Gray scale version

Image Sensor Characteristics

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Unit</th>
<th>Measurement method</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>S</td>
<td>650</td>
<td>620</td>
<td>1070</td>
<td>mV</td>
<td>1</td>
<td>1/255s accumulation conversion value</td>
</tr>
<tr>
<td>Saturation signal</td>
<td>Vsat</td>
<td>375</td>
<td></td>
<td></td>
<td>mV</td>
<td>2</td>
<td>Ta = 60°C</td>
</tr>
<tr>
<td>Smear</td>
<td>Sm</td>
<td>-100</td>
<td>-92</td>
<td>-92</td>
<td>dB</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Video signal shading</td>
<td>SH</td>
<td>25</td>
<td></td>
<td></td>
<td>%</td>
<td>4</td>
<td>Zone 0</td>
</tr>
<tr>
<td>Dark signal</td>
<td>Vdt</td>
<td>2</td>
<td></td>
<td></td>
<td>mV</td>
<td>5</td>
<td>Ta = 60°C</td>
</tr>
<tr>
<td>Dark signal shading</td>
<td>ΔVdt</td>
<td>1</td>
<td></td>
<td></td>
<td>mV</td>
<td>6</td>
<td>Ta = 60°C</td>
</tr>
<tr>
<td>Lag</td>
<td>Lag</td>
<td>0.5</td>
<td></td>
<td></td>
<td>%</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

Note: All image sensor characteristic data noted above is for operation in 1/50s progressive scan mode.
Spectral Sensitivity Characteristics (Excludes lens characteristics and light source characteristics)
Measurement conditions

1. In the following measurements, the substrate voltage is set to the value indicated on the device, and the device drive conditions are at the typical values of the bias and clock voltage conditions.

2. In the following measurements, spot blemishes are excluded and, unless otherwise specified, the optical black level (OB) is used as the reference for the signal output, which is taken as the value measured at point [SB] of the measurement system.

3. In the following measurements, this image sensor is operated in 1/50s progressive scan mode.

Definition of standard imaging conditions

1. Standard imaging condition I:
   Use a pattern box (luminance: 700cd/m², color temperature of 3200K halogen source) as a subject. (Pattern for evaluation is not applicable.) Use a testing standard lens with CM500S (f = 1.0mm) as an IR cutoff filter and image at F8. The luminous intensity to the sensor receiving surface at this point is defined as the standard sensitivity testing luminous intensity.

2. Standard imaging condition II:
   Image a light source (color temperature of 3200K) with a uniformity of brightness within 2% at all angles. Use a testing standard lens with CM500S (f = 1.0mm) as an IR cutoff filter. The luminous intensity is adjusted to the value indicated in each testing item by the lens diaphragm.

1. Sensitivity
   Set to standard imaging condition I. After setting the electronic shutter mode with a shutter speed of 1/250s, measure the signal voltage (Vs) at the center of the screen, and substitute the value into the following formula.
   \[ S = Vs \times \frac{250}{23} \text{ [mV]} \]

2. Saturation signal
   Set to standard imaging condition II. After adjusting the luminous intensity to 10 times the intensity with the average value of the signal output, 120mV, measure the minimum value of the signal output.

3. Smear
   Set to standard imaging condition II. With the lens diaphragm at F5.6 to F8, first adjust the luminous intensity to 500 times the intensity with the average value of signal output, 120mV. Then after the readout clock is stopped and the charge drain is executed by the electronic shutter at the respective H blankings, measure the maximum value (YSm [mV]) of the signal output and substitute the value into the following formula.
   \[ Sm = 20 \times \log \left( \frac{V_{Sm}}{120} \times \frac{1}{500} \times \frac{1}{10} \right) \text{ [dB] (1/10V method conversion value)} \]

4. Video signal shading
   Set to standard imaging condition II. With the lens diaphragm at F5.6 to F8, adjust the luminous intensity so that the average value of the signal output is 120mV. Then measure the maximum (Vmax [mV]) and minimum (Vmin [mV]) values of the signal output and substitute the values into the following formula.
   \[ SH = \frac{V_{max} - V_{min}}{120} \times 100 \% \]
5. Dark signal
   Measure the average value of the signal output (V_{dt} [mV]) with the device ambient temperature 50°C and the device in the light-obstructed state, using the horizontal idle transfer level as a reference.

6. Dark signal shading
   After measuring 5, measure the maximum (V_{dmax} [mV]) and minimum (V_{dmin} [mV]) values of the dark signal output and substitute the values into the following formula.

   \[ \Delta V_{dt} = V_{dmax} - V_{dmin} [mV] \]

7. Leg
   Adjust the signal output generated by strobe light to 120mV. After setting the strobe light so that it strobos with the following timing, measure the residual signal (V_{lag}). Substitute the value into the following formula.

   \[ L_{eg} = \left( \frac{V_{lag}}{120} \right) \times 100 \% \]

---

**Measurement System**

- CCD signal output (*A*)
- Gr/Gr channel signal output (*B*)
- R/B channel signal output (*C*)

**Note** Adjust the amplifier gain so that the gain between (*A*) and (*B*), and between (*A*) and (*C*) equals 1.

---

23.3.6 Device Feature And Property List

23.3.6.1 Gray scale version

23.3.6.2 Color version

<bsensor data.r>
23.4 mvBlueCOUGAR-Xx22 (1.2 Mpix [1280 x 960])

23.4.1 Introduction

The CCD sensor is a highly programmable imaging module which will, for example, enable the following type of applications

**Industrial applications:**

- triggered image acquisition with precise control of image exposure start by hardware trigger input.
- image acquisition of fast moving objects due to:
  - frame exposure, integrating all pixels at a time in contrast to CMOS imager which typically integrate line-by-line.
  - short shutter time, to get sharp images.
  - flash control output to have enough light for short time.

**Scientific applications:**

- long time exposure for low light conditions.
- optimizing image quality using the variable shutter control.

23.4.2 Details of operation

The process of getting an image from the CCD sensor can be separated into three different phases.

23.4.2.1 Trigger

When coming out of reset or ready with the last readout the CCD controller is waiting for a Trigger signal.

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Setting (GenICam (p.143))</th>
<th>Mode / Setting (Device Specific (p.143))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = Off&quot;</td>
<td>Continuous</td>
<td>Free running, no external trigger signal needed.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = Software&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnDemand</td>
<td>Image acquisition triggered by command (software trigger).</td>
</tr>
</tbody>
</table>

To trigger one frame execute the **TriggerSoftware@i** command then.
<table>
<thead>
<tr>
<th>TriggerSelector</th>
<th>TriggerSource</th>
<th>TriggerActivation</th>
<th>ExposureMode</th>
<th>Description</th>
</tr>
</thead>
</table>
| AcquisitionActive | <desired Line> | LevelLow | Timed | OnLowLevel
Start an exposure of a frame as long as the trigger input is below the trigger threshold. (No Frame ← Trigger!) |
| AcquisitionActive | <desired Line> | LevelHigh | Timed | OnHighLevel
Start an exposure of a frame as long as the trigger input is above the trigger threshold. (No Frame ← Trigger!) |
| FrameStart | <desired Line> | FallingEdge | Timed | OnFallingEdge
Each falling edge of trigger signal acquires one image. |
| FrameStart | <desired Line> | RisingEdge | Timed | OnRisingEdge
Each rising edge of trigger signal acquires one image. |
| FrameStart | <desired Line> | LevelHigh | TriggerWidth | OnHighExpose
Each rising edge of trigger signal acquires one image, exposure time corresponds to pulse width. |
| FrameStart | <desired Line> | LevelLow | TriggerWidth | OnLowExpose
Each falling edge of trigger signal acquires one image, exposure time corresponds to pulse width. |
"TriggerSelector = FrameStart"
"TriggerMode = On"
"TriggerSource = <desired Line>"
"TriggerActivation = AnyEdge"
"ExposureMode = Timed"

<table>
<thead>
<tr>
<th>TriggerSource mvIMPACT Acquire</th>
<th>TriggerSource GenICam (valid values for &lt;desired Line&gt;)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP-IN0</td>
<td>Line4</td>
</tr>
<tr>
<td>GP-IN1</td>
<td>Line5</td>
</tr>
</tbody>
</table>

23.4.2.2 Exposure aka Exposure

After an active trigger, the exposure phase starts with a maximum jitter of $t_{\text{trig}}$. If flash illumination is enabled in software the flash output will be activated exactly while the sensor chip is integrating light. Exposure time is adjustable by software in increments of $t_{\text{readline}}$.

23.4.2.3 Readout

When exposure is finished, the image is transferred to hidden storage cells on the CCD. Image data is then shifted out line-by-line and transferred to memory. Shifting out non active lines takes $t_{\text{shift}}$, while shifting out active lines will consume $t_{\text{readline}}$. The number of active pixels per line will not have any impact on readout speed.

23.4.3 CCD Timing

**Note**

In the "Trigger Overlap ReadOut" mode, the camera accepts a trigger signal the $t_{\text{exposure}}$ period earlier than $t_{\text{readout}}$ is finished.
23.4.3.1 Timings
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Pixel clock (MHz)</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>t\text{trig}</td>
<td>Time from trigger (internal or external) to exposure start</td>
<td>36 MHz</td>
<td>50 MHz</td>
<td></td>
</tr>
<tr>
<td>t\text{wait}</td>
<td>Time after a new trigger is accepted</td>
<td>&lt; 1us</td>
<td></td>
<td></td>
</tr>
<tr>
<td>t\text{trans}</td>
<td>Image transfer time (move image to readout cells in CCD)</td>
<td>16.25us</td>
<td>11.70us</td>
<td></td>
</tr>
<tr>
<td>t\text{ readline}</td>
<td>time needed to readout a line</td>
<td>45.83us</td>
<td>33.00us</td>
<td></td>
</tr>
<tr>
<td>t\text{vshift}</td>
<td>time needed to shift unused lines away</td>
<td>2.06us</td>
<td>1.48us</td>
<td></td>
</tr>
<tr>
<td>t\text{exposure}</td>
<td>Exposure time</td>
<td>10us..10s</td>
<td>10us..10s</td>
<td></td>
</tr>
<tr>
<td>t\text{readout}</td>
<td>Image readout time (move image from readout cells to memory)</td>
<td>t\text{readout} = ((ActiveLines + 1) * t\text{ readline}) + ((976 - ActiveLines) * t\text{vshift}) + t\text{trans}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note**

In partial scan mode (readout window ysize < 964 lines).

To calculate the maximum frames per second (FPS\text{max}) you will need following formula (Expose mode: No overlap):

\[
\text{FPS\text{max}} = \frac{1}{t_{\text{readout}} + t_{\text{exposure}}}
\]

(Expose mode: Overlapped):

\[
\begin{align*}
\text{t\_readout} < \text{t\_exposure} : & \quad \text{FPS\text{max}} = \frac{1}{t_{\text{exposure}}} \\
\text{t\_readout} > \text{t\_exposure} : & \quad \text{FPS\text{max}} = \frac{1}{t_{\text{readout}}}
\end{align*}
\]

23.4.3.1.1 Example: Frame rate as function of lines & exposure time

Now, when we insert the values using exposure time of, for example, 8000us and 960 lines and 50MHz pixel clock (Expose mode: No overlap):

\[
\text{FPS\text{max}} = \frac{1}{(((960 + 1) * 33.00 \text{us}) + ((976 - 960) * 1.48 \text{us}) + 11.70 \text{us}) + 8000 \text{us}}
\]

\[
= 0.00002515825802208794 \text{ / us}
\]

\[
= 25.2
\]

23.4.4 Reprogramming CCD Timing

Reprogramming the CCD Controller will happen when the following changes occur:

- Changing the exposure time
- Changing the capture window

MATRIX VISION GmbH
• Changing Trigger Modes

Reprogram time consists of two phases

1. Time needed to send data to the CCD controller depending on what is changed
   exposure: about 2.3 ms
   window: about 4.6 ms
   trigger mode: from 5.90 ms, varies with oldmode/newmode combination

2. Time to initialize (erase) the CCD chip after reprogramming this is fixed, about 4.5 ms

So for example when reprogramming the capture window you will need (average values)

\[ t_{regprog} = \text{change\_window} + \text{init\_ccd} \]

\[ t_{regprog} = 5\text{ms} + 4.5\text{ms} \]

\[ t_{regprog} = 9.5\text{ms} \]

23.4.5 CCD Sensor Data

Device Structure

• Interline CCD image sensor
• Image size: Diagonal 6mm (Type 1/3)
• Number of effective pixels: 1296 (H) x 966 (V) approx. 1.25 M pixels
• Total number of pixels: 1348 (H) x 976 (V) approx. 1.32 M pixels
• Chip size: 6.26mm (H) x 5.01mm (V)
• Unit cell size: 3.75um (H) x 3.75um (V)
• Optical black:
  – Horizontal (H) direction: Front 12 pixels, rear 40 pixels
  – Vertical (V) direction: Front 8 pixels, rear 2 pixels
• Number of dummy bits: Horizontal 4 Vertical 2
• Substrate material: Silicon
23.4.5.1 Characteristics

These zone definitions apply to both the color and gray scale version of the sensor.

Image Sensor Characteristics (Center cut-out drive, 30 frame/s)

(Ta = 25°C)

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Unit</th>
<th>Measurement method</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>G sensitivity</td>
<td>Sg</td>
<td>300</td>
<td>360</td>
<td></td>
<td>mV</td>
<td>1</td>
<td>1/30s accumulation</td>
</tr>
<tr>
<td>Sensitivity ratio</td>
<td>Rr</td>
<td>0.55</td>
<td>0.81</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Rb</td>
<td>0.23</td>
<td>0.49</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Saturation signal</td>
<td>Ysat</td>
<td></td>
<td></td>
<td></td>
<td>mV</td>
<td>2</td>
<td>Ta = 60°C</td>
</tr>
<tr>
<td>Smear</td>
<td>Sm</td>
<td>-104</td>
<td>-96</td>
<td></td>
<td>dB</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Video signal shading</td>
<td>SHg</td>
<td></td>
<td></td>
<td></td>
<td>%</td>
<td>4</td>
<td>Zone 0 and I</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td>Zone 0 to II'</td>
</tr>
<tr>
<td>Uniformity between video signal channels</td>
<td>ΔSrg</td>
<td></td>
<td></td>
<td>8</td>
<td>%</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ΔSbg</td>
<td></td>
<td></td>
<td>8</td>
<td>%</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Dark signal</td>
<td>Vdt</td>
<td></td>
<td></td>
<td>2</td>
<td>mV</td>
<td>6</td>
<td>Ta = 60°C, 1/30s accumulation</td>
</tr>
<tr>
<td>Dark signal shading</td>
<td>ΔVdt</td>
<td></td>
<td></td>
<td>1</td>
<td>mV</td>
<td>7</td>
<td>Ta = 60°C, 1/30s accumulation*1</td>
</tr>
<tr>
<td>Line crawl R</td>
<td>Lcr</td>
<td></td>
<td></td>
<td>3.8</td>
<td>%</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Line crawl B</td>
<td>Lcb</td>
<td></td>
<td></td>
<td>3.8</td>
<td>%</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Lag</td>
<td>Lag</td>
<td></td>
<td></td>
<td>0.5</td>
<td>%</td>
<td>9</td>
<td></td>
</tr>
</tbody>
</table>

*1 Excludes vertical dark signal shading caused by vertical register high-speed transfer.
Spectral Sensitivity Characteristics (excludes lens characteristics and light source characteristics)

23.4.5.3 Gray scale version

Image Sensor Characteristics (Center cut-out drive, 38 frames)

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Unit</th>
<th>Measurement method</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity 1</td>
<td>S1</td>
<td>300</td>
<td>480</td>
<td></td>
<td>mV/°</td>
<td>1</td>
<td>V/0s accumulation</td>
</tr>
<tr>
<td>Sensitivity 2</td>
<td>S2</td>
<td>1000</td>
<td>1500</td>
<td></td>
<td>mV/°</td>
<td>2</td>
<td>V/0s accumulation</td>
</tr>
<tr>
<td>Saturation signal</td>
<td>Vsat</td>
<td>350</td>
<td></td>
<td></td>
<td>mV/°</td>
<td>3</td>
<td>Ta = 0°C</td>
</tr>
<tr>
<td>Spread</td>
<td>Sn</td>
<td>-604</td>
<td>-66</td>
<td>6</td>
<td>dB</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Video signal shading</td>
<td>SH</td>
<td>20%</td>
<td>5</td>
<td>Zone 0 and I</td>
<td>%</td>
<td>5</td>
<td>Zone 0 to II</td>
</tr>
<tr>
<td>Dark signal</td>
<td>VDS</td>
<td>2</td>
<td>6</td>
<td>Ta = 0°C, 1/30s accumulation</td>
<td>mV/°</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dark signal shading</td>
<td>ΔVDS</td>
<td>1</td>
<td>7</td>
<td>Ta = 0°C, 1/30s accumulation</td>
<td>mV/°</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lag</td>
<td>Lg</td>
<td>0.5</td>
<td>8</td>
<td>0</td>
<td>%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*1: Excludes vertical dark signal shading caused by the vertical register high-speed transfer.

Spectral Sensitivity Characteristics (excludes lens characteristics and light source characteristics)
Measurement conditions

1. In the following measurements, the device drive conditions are at the typical values of the bias and clock voltage conditions.

2. In the following measurements, spot images are excluded and, unless otherwise specified, the optical black (OB) level is used as the reference for the signal output, and the value measured at point MB of the measurement system is used.

Definition of Standard Imaging Conditions

1. Standard imaging condition I:
   Use a pattern box (luminance: 706cd/m², color temperature of 3200K halogen source) as a subject. (Pattern for evaluation is not applicable.) Use a testing standard lens with CM500S (t = 1.0mm) as an IR cut filter and image at F8. The luminous intensity to the sensor receiving surface at this point is defined as the standard sensitivity testing luminous intensity.

2. Standard imaging condition II:
   This indicates the standard imaging condition I with the IR cut filter removed.

3. Standard imaging condition III:
   Image a light source (color temperature of 3200K) with a uniformity of brightness within 2% at all angles. Use a testing standard lens with CM500S (t = 1.0mm) as an IR cut filter. The luminous intensity is adjusted to the value indicated in each testing item by the lens diaphragm.

   1. Sensitivity 1
      Set the measurement condition to the standard imaging condition I. After setting the electronic shutter mode with a shutter speed of 1/100s, measure the signal output (V51) at the center of the screen, and substitute the value into the following formula.

      \[ S1 = \frac{V51 \times (100/30)}{mV} \]

   2. Sensitivity 2
      Set the measurement condition to the standard imaging condition II. After setting the electronic shutter mode with a shutter speed of 1/500s, measure the signal output (V52) at the center of the screen, and substitute the value into the following formula.

      \[ S2 = \frac{V52 \times (500/30)}{mV} \]

   3. Saturation signal
      Set the measurement condition to the standard imaging condition III. After adjusting the luminous intensity to 10 times the intensity with the average value of the signal output, 150mV, measure the minimum value of the signal output.

   4. Smear
      Set the measurement condition to the standard imaging condition III. With the lens diaphragm at F5.6 to F8, adjust the luminous intensity to 500 times the intensity with the average value of the signal output, 150mV. When the readout clock is stopped and the charge drain is executed by the electronic shutter at the respective H blankings, measure the maximum value (V5m) of the signal output, and substitute the value into the following formula.

      \[ Sm = 20 \times \log \left( \frac{V5m}{150} \times \frac{1}{500} \times \frac{1}{10} \right) [dB] (1/10V method conversion value) \]
5. Video signal shading
   Set the measurement condition to the standard imaging condition. With the lens diaphragm at F5.6 to
   F8, adjust the luminous intensity so that the average value of the signal output is 150mV. Then measure
   the maximum value (Vmax) and the minimum value (Vmin) of the signal output, and substitute the values
   into the following formula.

   \[
   \text{SH} = \frac{(V_{\text{max}} - V_{\text{min}})}{150} \times 100 \\% 
   \]

6. Dark signal
   Measure the average value (Vdt) of the signal output with the device ambient temperature of \(60^\circ\text{C}\) and the
   device in the light-obstructed state, using the horizontal idle transfer level as the reference.

7. Dark signal shading
   After the measurement item 6, measure the maximum value (Vdmax) and the minimum value (Vdmin) of
   the dark signal output, and substitute the values into the following formula.

   \[
   \Delta V_{\text{dt}} = V_{\text{dmax}} - V_{\text{dmin}} \text{ [mV]} 
   \]

8. Lag
   Adjust the signal output value generated by the strobe light to 150mV. After setting the strobe light so that
   it strobos with the following timing, measure the residual signal level (Vlag), and substitute the value into
   the following formula.

   \[
   \text{Lag} = \frac{(V_{\text{lag}}/150)}{100} \times 100 \\% 
   \]

---

**Measurement System**

\[
\text{CCD signal output \((\text{A})\)} \\
\text{CCD} \quad \xrightarrow{\text{GDD}} \quad \text{AMP} \quad \xrightarrow{\text{SH}} \quad \text{CCD signal output \((\text{B})\)}
\]

**Note** Adjust the amplifier gain so that the gain between \((\text{A})\) and \((\text{B})\) equals 1.

---

23.5 mvBlueCOUGAR-Xx23 (1.4 Mpix [1360 x 1024])
23.5.1 Introduction

The CCD sensor is a highly programmable imaging module which will, for example, enable the following type of applications:

**Industrial applications:**

- triggered image acquisition with precise control of image exposure start by hardware trigger input.
- image acquisition of fast moving objects due to:
  - frame exposure, integrating all pixels at a time in contrast to CMOS imager which typically integrate line-by-line.
  - short shutter time, to get sharp images.
  - flash control output to have enough light for short time.

**Scientific applications:**

- long time exposure for low light conditions.
- optimizing image quality using the variable shutter control.

23.5.2 Details of operation

The process of getting an image from the CCD sensor can be separated into three different phases.

23.5.2.1 Trigger

When coming out of reset or ready with the last readout the CCD controller is waiting for a Trigger signal.

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Setting (GenICam (p.143))</th>
<th>Mode / Setting (Device Specific (p.143))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = Off&quot;</td>
<td>Continuous</td>
<td>Free running, no external trigger signal needed.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = Software&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnDemand</td>
<td>Image acquisition triggered by command (software trigger).</td>
</tr>
<tr>
<td>To trigger one frame execute the TriggerSoftware@i command then.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelLow&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnLowLevel</td>
<td>Start an exposure of a frame as long as the trigger input is below the trigger threshold. (No Frame ← Trigger!)</td>
</tr>
</tbody>
</table>
"TriggerSelector = AcquisitionActive"  
"TriggerMode = On"  
"TriggerSource = <desired Line>"  
"TriggerActivation = LevelHigh"  
"ExposureMode = Timed"

OnHighLevel  
Start an exposure of a frame as long as the trigger input is above the trigger threshold. (No Frame→Trigger!)

"TriggerSelector = FrameStart"  
"TriggerMode = On"  
"TriggerSource = <desired Line>"  
"TriggerActivation = FallingEdge"  
"ExposureMode = Timed"

OnFallingEdge  
Each falling edge of trigger signal acquires one image.

"TriggerSelector = FrameStart"  
"TriggerMode = On"  
"TriggerSource = <desired Line>"  
"TriggerActivation = RisingEdge"  
"ExposureMode = Timed"

OnRisingEdge  
Each rising edge of trigger signal acquires one image.

"TriggerSelector = FrameStart"  
"TriggerMode = On"  
"TriggerSource = <desired Line>"  
"TriggerActivation = LevelHigh"  
"ExposureMode = TriggerWidth"

OnHighExpose  
Each rising edge of trigger signal acquires one image, exposure time corresponds to pulse width.

"TriggerSelector = FrameStart"  
"TriggerMode = On"  
"TriggerSource = <desired Line>"  
"TriggerActivation = LevelLow"  
"ExposureMode = TriggerWidth"

OnLowExpose  
Each falling edge of trigger signal acquires one image, exposure time corresponds to pulse width.

"TriggerSelector = FrameStart"  
"TriggerMode = On"  
"TriggerSource = <desired Line>"  
"TriggerActivation = AnyEdge"  
"ExposureMode = Timed"

OnAnyEdge  
Start the exposure of a frame when the trigger input level changes from high to low or from low to high.

Line Mapping (TriggerSource mvIMPACT Acquire -> TriggerSource GenICam (valid values for <desired Line>)):

<table>
<thead>
<tr>
<th>TriggerSource mvIMPACT Acquire</th>
<th>TriggerSource GenICam(BCX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP-IN0</td>
<td>Line4</td>
</tr>
<tr>
<td>GP-IN1</td>
<td>Line5</td>
</tr>
</tbody>
</table>
See also

For detailed description about the trigger modes (https://www.matrix-vision/manuals/ [mvIMPACT Acquire API])

- C: TCameraTriggerMode
- C++: mvIMPACT::acquire::TCameraTriggerMode

23.5.2.2 Exposure aka Integration

After an active trigger, the exposure phase starts with a maximum jitter of $t_{trig}$. If flash illumination is enabled in software the flash output will be activated exactly while the sensor chip is integrating light. Exposure time is adjustable by software in increments of $t_{readline}$.

23.5.2.3 Readout

When exposure is finished, the image is transferred to hidden storage cells on the CCD. Image data is then shifted out line-by-line and transferred to memory. Shifting out non active lines takes $t_{shift}$, while shifting out active lines will consume $t_{readline}$. The number of active pixels per line will not have any impact on readout speed.

23.5.3 CCD Timing

**Trigger Overlap: No**

![Diagram of Trigger Overlap: No]

**Trigger Overlap: ReadOut**

![Diagram of Trigger Overlap: ReadOut]

---

**Note**

In the "Trigger Overlap ReadOut" mode, the camera accepts a trigger signal the $t_{exposure}$ period earlier than $t_{readout}$ is finished.

23.5.3.1 Timings
### Name | Description | Pixel clock
--- | --- | ---
| t\text{trig} | Time from trigger (internal or external) to exposure start | 28 MHz | 56 MHz
| t\text{wait} | Time after a new trigger is accepted | < 1us
| t\text{trans} | Image transfer time (move image to readout cells in CCD) | 42.86us | 21.43us
| t\text{readline} | Time needed to readout a line | 63.93us | 31.96us
| t\text{vshift} | Time needed to shift unused lines away | 8.57us | 4.29us
| t\text{exposure} | Exposure time | 20us..10s | 20us..10s
| t\text{readout} | Image readout time (move image from readout cells to memory) | \(t_{\text{readout}} = ((\text{ActiveLines} + 1) \times \text{t_{\text{readline}}}) + ((1050 - \text{ActiveLines}) \times \text{t_{\text{vshift}}}) + \text{t_{\text{trans}}}\)

#### Note

In partial scan mode (readout window ysize < 1024 lines).

To calculate the maximum frames per second (FPS\text{max}) you will need following formula (Expose mode: No overlap):

\[
\text{FPS}_{\text{max}} = \frac{1}{t_{\text{readout}} + t_{\text{exposure}}}
\]

(Expose mode: Overlapped):

- If \(t_{\text{readout}} < t_{\text{exposure}}\): 
  \[
  \text{FPS}_{\text{max}} = \frac{1}{t_{\text{exposure}}}
  \]
- If \(t_{\text{readout}} > t_{\text{exposure}}\): 
  \[
  \text{FPS}_{\text{max}} = \frac{1}{t_{\text{readout}}}
  \]

#### 23.5.3.1.1 Example: Frame rate as function of lines & exposure time

Now, when we insert the values using exposure time of, for example, 8000 us, 1024 lines and 56MHz pixel clock (Expose mode: No overlap):

\[
\begin{align*}
\text{FPS}_{\text{max}} &= \frac{1}{t_{\text{readout}} + t_{\text{exposure}}} \\
&= \frac{1}{(((1024 + 1) \times 31.96 \text{ us}) + ((1050 - 1024) \times 4.29 \text{ us}) + 8.57 \text{ us}) + 8000 \text{ us}} \\
&= \frac{1}{0.00002445678999324316 \text{ us}} \\
&= 24.5
\end{align*}
\]

#### 23.5.3.2 Frame rate calculator

The calculator returns the max. frame rate supported by the sensor. Please keep in mind that it will depend on the interface and the used image format if this frame rate can be transferred.
23.5.4 Reprogramming CCD Timing

Reprogramming the CCD Controller will happen when the following changes occur

- Changing the exposure time
- Changing the capture window
- Changing Trigger Modes

Reprogram time consists of two phases

1. Time needed to send data to the CCD controller depending on what is changed
   - exposure: abt 2.3ms
   - window: abt 4.6ms
   - trigger mode: from 5.90ms, varies with oldmode/newmode combination

2. Time to initialize (erase) the CCD chip after reprogramming this is fixed, abt 4.5 ms

So for example when reprogramming the capture window you will need (average values)

\[ t_{\text{regprog}} = \text{change\_window} + \text{init\_ccd} \]

\[ t_{\text{regprog}} = 5\text{ms} + 4.5\text{ms} \]

\[ t_{\text{regprog}} = 9.5\text{ms} \]

23.5.5 CCD Sensor Data

Device Structure

- Interline CCD image sensor
- Image size: Diagonal 8mm (Type 1/2)
- Number of effective pixels: 1392 (H) x 1040 (V) approx. 1.45M pixels
- Total number of pixels: 1434 (H) x 1050 (V) approx. 1.5M pixels
- Chip size: 7.60mm (H) x 6.2mm (V)
- Unit cell size: 4.65um (H) x 4.65um (V)
- Optical black:
  - Horizontal (H) direction: Front 2 pixels, rear 40 pixels
  - Vertical (V) direction: Front 8 pixels, rear 2 pixels
- Number of dummy bits: Horizontal 20 Vertical 3
- Substrate material: Silicon
23.5.5.1 Characteristics

These zone definitions apply to both the color and gray scale version of the sensor.
### Image Sensor Characteristics

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Unit</th>
<th>Measurement method</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>G sensitivity</td>
<td>$S_g$</td>
<td>320</td>
<td>400</td>
<td></td>
<td>mV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensitivity comparison</td>
<td>R</td>
<td>0.4</td>
<td>0.55</td>
<td>0.7</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>0.3</td>
<td>0.46</td>
<td>0.6</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Saturation signal</td>
<td>$V_{sat}$</td>
<td>450</td>
<td></td>
<td></td>
<td>mV</td>
<td>2</td>
<td>Progressive scan readout mode</td>
</tr>
<tr>
<td></td>
<td>$V_{sat2}$</td>
<td>380</td>
<td></td>
<td></td>
<td>mV</td>
<td>2</td>
<td>High frame rate readout mode</td>
</tr>
<tr>
<td></td>
<td>$V_{sat4}$</td>
<td>380</td>
<td></td>
<td></td>
<td>mV</td>
<td>2</td>
<td>High frame rate readout two pixels addition</td>
</tr>
<tr>
<td>Smear</td>
<td>$S_m$</td>
<td>0.001</td>
<td>0.0025</td>
<td>%</td>
<td>3</td>
<td></td>
<td>Progressive scan readout, high frame rate readout two pixels addition</td>
</tr>
<tr>
<td>Video signal shading</td>
<td>$S_{Hg}$</td>
<td>20</td>
<td></td>
<td></td>
<td>%</td>
<td>4</td>
<td>Zone 0 and 1</td>
</tr>
<tr>
<td>Uniformity between video signal channels</td>
<td>$\Delta S_{rg}$</td>
<td>8</td>
<td>%</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dark signal</td>
<td>$Y_{dt}$</td>
<td>8</td>
<td>mV</td>
<td>6</td>
<td></td>
<td>Ta = 60°C, 15 frames/s</td>
<td></td>
</tr>
<tr>
<td>Dark signal shading</td>
<td>$\Delta Y_{dt}$</td>
<td>2</td>
<td>mV</td>
<td>7</td>
<td></td>
<td>Ta = 60°C, 15 frames/s</td>
<td></td>
</tr>
<tr>
<td>Line crawl G</td>
<td>Lg</td>
<td>3.8</td>
<td></td>
<td></td>
<td>%</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Line crawl R</td>
<td>Lcr</td>
<td>3.8</td>
<td></td>
<td></td>
<td>%</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Line crawl B</td>
<td>Lcb</td>
<td>3.8</td>
<td></td>
<td></td>
<td>%</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Lag</td>
<td>Lag</td>
<td>0.5</td>
<td></td>
<td></td>
<td>%</td>
<td>9</td>
<td></td>
</tr>
</tbody>
</table>

*1 $V_{sat4}$ is the saturation signal amount at two pixels addition, and it is 190mV per one pixel. $V_{sat}$ internal generation value ensures 190mV per one pixel of the saturation signal amount in high frame rate two pixels addition mode.

*2 Eliminates the dark signal shading in the vertical direction by the high-speed transfer of the vertical register.

### Spectral Sensitivity Characteristics

(Spectral Sensitivity Characteristics (excludes lens characteristics and light source characteristics))
23.5.5.3 Gray scale version

**Image Sensor Characteristics**

(Ta = 25°C)

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Unit</th>
<th>Measurement method</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>S</td>
<td>350</td>
<td>450</td>
<td>mV</td>
<td>1/30s accumulation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Saturation signal</td>
<td>Vsat</td>
<td>450</td>
<td></td>
<td>mV</td>
<td>2</td>
<td>Progressive scan readout mode</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Vsat2</td>
<td>380</td>
<td></td>
<td>mV</td>
<td>2</td>
<td>High frame rate readout mode</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Vsat4</td>
<td>380</td>
<td></td>
<td>mV</td>
<td>2</td>
<td>High frame rate readout two pixels addition³¹</td>
<td></td>
</tr>
<tr>
<td>Smear</td>
<td>Sm</td>
<td>0.001</td>
<td>0.0025</td>
<td>%</td>
<td>3</td>
<td>Progressive scan readout, high frame rate readout two pixels addition</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.002</td>
<td>0.005</td>
<td>%</td>
<td>3</td>
<td>High frame rate readout mode</td>
<td></td>
</tr>
<tr>
<td>Video signal shading</td>
<td>SHg</td>
<td>25</td>
<td></td>
<td>%</td>
<td>4</td>
<td>Zone 0 and 1²</td>
<td></td>
</tr>
<tr>
<td>Dark signal</td>
<td>Vdst</td>
<td>8</td>
<td></td>
<td>mV</td>
<td>5</td>
<td>Zone 0 to 1'³²</td>
<td></td>
</tr>
<tr>
<td>Dark signal shading</td>
<td>ΔVdst</td>
<td>2</td>
<td></td>
<td>mV</td>
<td>6</td>
<td>Ta = 60°C, 15 frames/s</td>
<td></td>
</tr>
<tr>
<td>Lag</td>
<td>Lag</td>
<td>0.5</td>
<td></td>
<td>%</td>
<td>7</td>
<td>Ta = 60°C, 15 frames/s³²</td>
<td></td>
</tr>
</tbody>
</table>

³¹ Vsat4 is the saturation signal amount at two pixels addition, and it is 190 mV per one pixel. Vsus internal generation value ensures 190 mV per one pixel of the saturation signal amount in high frame rate two pixels addition mode.

³² Eliminates the dark signal shading in the vertical direction by the high-speed transfer of the vertical register.

**Spectral Sensitivity Characteristics** (excludes lens characteristics and light source characteristics)

![Graph showing spectral sensitivity characteristics](image)

MATRIX VISION GmbH
1. **G sensitivity, sensitivity comparison**
   Set to standard imaging condition I. After selecting the electronic shutter mode with a shutter speed of 1/100s, measure the signal outputs ($V_{Gr}$, $V_{Gb}$, $V_{R}$, and $V_{B}$) at the center of each $Gr$, $Gb$, $R$ and $B$ channel screen, and substitute the values into the following formulas:

   \[
   \begin{align*}
   V_0 &= (V_{Gr} + V_{Gb})/2 \\
   S_g &= V_G \times \frac{100}{90} \text{ [mV]} \\
   R_t &= V_R/V_0 \\
   R_b &= V_B/V_0
   \end{align*}
   \]

2. **Saturation signal**
   Set to standard imaging condition I. After adjusting the luminous intensity to 20 times the intensity with the average value of the $Gr$ signal output, 150mV, measure the minimum values of the $Gr$, $Gb$, $R$ and $B$ signal outputs.

3. **Smear**
   Set to standard imaging condition I. With the lens diaphragm at F5.6 to F8, first adjust the average value of the $Gr$ signal output to 150mV. Measure the average values of the $Gr$ signal output, $Gb$ signal output, $R$ signal output and $B$ signal output ($Gra$, $Gba$, $Ra$, $Ba$), and then adjust the luminous intensity to 500 times the intensity with the average value of the $Gr$ signal output, 150mV. After the readout clock is stopped and the charge drain is executed by the electronic shutter at the respective H blankings, measure the maximum value ($V_{sm}$ [mV]) independent of the $Gr$, $Gb$, $R$ and $B$ signal outputs, and substitute the values into the following formula.

   \[
   Sm = V_{sm} - \frac{Gra + Gba + Ra + Ba}{4} \times \frac{1}{500} \times \frac{1}{10} \times 100 \text{ [%]} \text{ (1/10V method conversion value)}
   \]
4. Video signal shading
Set to standard imaging condition 1. With the lens diaphragm at F5.6 to F8, adjust the luminous intensity so that the average value of the Gr signal output is 150mV, then measure the maximum (Grmax [mV]) and minimum (Grmin [mV]) values of the Gr signal output and substitute the values into the following formula.

\[ SHg = \frac{(Grmax - Grmin)}{150} \times 100 \% \]

5. Uniformity between video signal channels
After measuring 4, measure the maximum (Rmax [mV]) and minimum (Rmin [mV]) values of the R signal and the maximum (Bmax [mV]) and minimum (Bmin [mV]) values of the B signal, and substitute the values into the following formulas.

\[ \Delta Srg = \frac{(Rmax - Rmin)}{150} \times 100 \% \]
\[ \Delta Sbg = \frac{(Bmax - Bmin)}{150} \times 100 \% \]

6. Dark signal
Measure the average value of the signal output (Vdt [mV]) with the device ambient temperature 60°C and the device in the light-obstructed state, using the horizontal idle transfer level as a reference.

7. Dark signal shading
After measuring 6, measure the maximum (Vdmax [mV]) and minimum (Vdmin [mV]) values of the dark signal output and substitute the values into the following formula.

\[ \Delta Vdt = Vdmax - Vdmin \ [mV] \]

8. Line crawl
Set to standard imaging condition 1. Adjusting the luminous intensity so that the average value of the Gr signal output is 150mV, and then insert R, G and B filters and measure the difference between G signal lines (ΔGr, ΔGg, ΔGlb [mV]) as well as the average value of the G signal output (Gar, Gag, Gab). Substitute the values into the following formula.

\[ Lci = \frac{AGi}{Gar} \times 100 \% \ (i = r, g, h) \]

9. Lag
Adjust the Gr signal output value generated by strobe light to 150mV. After setting the strobe light so that it strobes with the following timing, measure the residual signal (Vlag). Substitute the value into the following formula.

\[ Lag = \frac{(Vlag/150)}{150} \times 100 \% \]

---

23.5.6 Device Feature And Property List

23.5.6.1 Gray scale version
23.6 mvBlueCOUGAR-Xx24 (1.9 Mpix [1600 x 1200])

23.6.1 Introduction

The CCD sensor is a highly programmable imaging module which will, for example, enable the following type of applications

**Industrial applications:**

- triggered image acquisition with precise control of image exposure start by hardware trigger input.
- image acquisition of fast moving objects due to:
  - frame exposure, integrating all pixels at a time in contrast to CMOS imager which typically integrate line-by-line.
  - short shutter time, to get sharp images.
  - flash control output to have enough light for short time.

**Scientific applications:**

- long time exposure for low light conditions.
- optimizing image quality using the variable shutter control.

23.6.2 Details of operation

The process of getting an image from the CCD sensor can be separated into three different phases.

23.6.2.1 Trigger

When coming out of reset or ready with the last readout the CCD controller is waiting for a Trigger signal.

The following trigger modes are available:
<table>
<thead>
<tr>
<th>Setting (GenICam (p.143))</th>
<th>Mode / Setting (Device Specific (p.143))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = Off&quot;</td>
<td>Continuous</td>
<td>Free running, no external trigger signal needed.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = Software&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnDemand</td>
<td>Image acquisition triggered by command (software trigger).</td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelLow&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnLowLevel</td>
<td>Start an exposure of a frame as long as the trigger input is below the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelHigh&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnHighLevel</td>
<td>Start an exposure of a frame as long as the trigger input is above the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = FallingEdge&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnFallingEdge</td>
<td>Each falling edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = RisingEdge&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnRisingEdge</td>
<td>Each rising edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelHigh&quot; &quot;ExposureMode = TriggerWidth&quot;</td>
<td>OnHighExpose</td>
<td>Each rising edge of trigger signal acquires one image, exposure time corresponds to pulse width.</td>
</tr>
</tbody>
</table>
"TriggerSelector = FrameStart"
"TriggerMode = On"
"TriggerSource = <desired Line>"
"TriggerActivation = LevelLow"
"ExposureMode = TriggerWidth"

<table>
<thead>
<tr>
<th>OnLowExposure</th>
<th>Each falling edge of trigger signal acquires one image, exposure time corresponds to pulse width.</th>
</tr>
</thead>
</table>

"TriggerSelector = FrameStart"
"TriggerMode = On"
"TriggerSource = <desired Line>"
"TriggerActivation = AnyEdge"
"ExposureMode = Timed"

<table>
<thead>
<tr>
<th>OnAnyEdge</th>
<th>Start the exposure of a frame when the trigger input level changes from high to low or from low to high.</th>
</tr>
</thead>
</table>

Line Mapping (TriggerSource mvIMPACT Acquire -> TriggerSource GenICam (valid values for <desired Line>):

<table>
<thead>
<tr>
<th>TriggerSource mvIMPACT Acquire</th>
<th>TriggerSource GenICam(BCX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP-IN0</td>
<td>Line4</td>
</tr>
<tr>
<td>GP-IN1</td>
<td>Line5</td>
</tr>
</tbody>
</table>

See also

For detailed description about the trigger modes (https://www.matrix-vision/manuals/[mvIMPACT Acquire API])

- C: TCameraTriggerMode
- C++: mvIMPACT::acquire::TCameraTriggerMode

23.6.2.2 Exposure aka Integration

After an active trigger, the exposure phase starts with a maximum jitter of \( t_{\text{trig}} \). If flash illumination is enabled in software the flash output will be activated exactly while the sensor chip is integrating light. Exposure time is adjustable by software in increments of \( t_{\text{readline}} \).

23.6.2.3 Readout

When exposure is finished, the image is transferred to hidden storage cells on the CCD. Image data is then shifted out line-by-line and transferred to memory. Shifting out non active lines takes \( t_{\text{shift}} \), while shifting out active lines will consume \( t_{\text{readline}} \). The number of active pixels per line will not have any impact on readout speed.
23.6.3 CCD Timing

**Trigger Overlap: No**

**Note**

In the "Trigger Overlap ReadOut" mode, the camera accepts a trigger signal the exposure period earlier than \( t_{\text{readout}} \) is finished.

### Timings

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Pixel clock</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_{\text{trig}} )</td>
<td>Time from trigger (internal or external) to exposure start</td>
<td>36 MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>56 MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>65 MHz</td>
</tr>
<tr>
<td>( t_{\text{wait}} )</td>
<td>Time after a new trigger is accepted</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt; 1us</td>
</tr>
<tr>
<td>( t_{\text{trans}} )</td>
<td>Image transfer time (move image to readout cells in CCD)</td>
<td>35.56us</td>
</tr>
<tr>
<td></td>
<td></td>
<td>22.86us</td>
</tr>
<tr>
<td></td>
<td></td>
<td>19.69us</td>
</tr>
<tr>
<td>( t_{\text{readline}} )</td>
<td>Time needed to readout a line</td>
<td>53.33us</td>
</tr>
<tr>
<td></td>
<td></td>
<td>34.29us</td>
</tr>
<tr>
<td></td>
<td></td>
<td>29.54us</td>
</tr>
<tr>
<td>( t_{\text{vshift}} )</td>
<td>Time needed to shift unused lines away</td>
<td>5.56us</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.57us</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.08us</td>
</tr>
<tr>
<td>( t_{\text{exposure}} )</td>
<td>Exposure time</td>
<td>20us..10s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20us..10s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20us..10s</td>
</tr>
<tr>
<td>( t_{\text{readout}} )</td>
<td>Image readout time (move image from readout cells to memory)</td>
<td>( t_{\text{readout}} = ((\text{ActiveLines} + 1) \times t_{\text{readline}}) + ((1248 - \text{ActiveLines}) \times t_{\text{vshift}}) )</td>
</tr>
</tbody>
</table>

**Note**

In partial scan mode (readout window ysize < 1200 lines).

To calculate the maximum frames per second (FPS\(_{\text{max}}\)) you will need following formula (Expose mode: No overlap):

\[
\text{FPS}_{\text{max}} = \frac{1}{t_{\text{readout}} + t_{\text{exposure}}}
\]
### Example: Frame rate as function of lines & exposure time

Now, when we insert the values using exposure time of, for example, 8000 us, 1200 lines and 65MHz pixel clock (Expose mode: No overlap):

\[
FPS_{\text{max}} = \frac{1}{((1200 + 1) \times 29.54 \text{ us}) + ((1248 - 1200) \times 3.08 \text{ us}) + 19.69 \text{ us}) + 8000 \text{ us}}
\]

\[
= 0.0000229120952263 \text{ 1 / us}
\]

\[
= 22.9
\]

### Frame rate calculator

**Note**

The calculator returns the max. frame rate supported by the sensor. Please keep in mind that it will depend on the interface and the used image format if this frame rate can be transferred.

### Reprogramming CCD Timing

Reprogramming the CCD Controller will happen when the following changes occur:

- Changing the exposure time
- Changing the capture window
- Changing Trigger Modes

Reprogram time consists of two phases:

1. Time needed to send data to the CCD controller depending on what is changed **exposure**: abt 2..3ms  
   **window**: abt 4..6ms  
   **trigger mode**: from 5..90ms, varies with oldmode/newmode combination

2. Time to initialize (erase) the CCD chip after reprogramming this is fixed, abt 4.5 ms

So for example when reprogramming the capture window you will need (average values)

\[
t_{\text{regprog}} = \text{change\_window} + \text{init\_ccd}
\]

\[
t_{\text{regprog}} = 5\text{ms} + 4.5\text{ms}
\]

\[
t_{\text{regprog}} = 9.5\text{ms}
\]
23.6.5  CCD Sensor Data

Device Structure

- Interline CCD image sensor
- Image size: Diagonal 8.923mm (Type 1/1.8)
- Number of effective pixels: 1600 (H) x 1200 (V) approx. 1.92M pixels
- Total number of pixels: 1688 (H) x 1248 (V) approx. 2.11M pixels
- Chip size: 8.50mm (H) x 6.8mm (V)
- Unit cell size: 4.4um (H) x 4.4um (V)
- Optical black:
  - Horizontal (H) direction: Front 12 pixels, rear 48 pixels
  - Vertical (V) direction: Front 10 pixels, rear 2 pixels
- Number of dummy bits: Horizontal 28 Vertical 1
- Substrate material: Silicon

23.6.5.1 Characteristics

These zone definitions apply to both the color and gray scale version of the sensor.
### Image Sensor Characteristics

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Unit</th>
<th>Measurement method</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>G Sensitivity</td>
<td>Sg</td>
<td>335</td>
<td>420</td>
<td>545</td>
<td>mV</td>
<td>1</td>
<td>1/30s accumulation</td>
</tr>
<tr>
<td>Sensitivity comparison</td>
<td>R</td>
<td>0.35</td>
<td>0.5</td>
<td>0.65</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>0.45</td>
<td>0.6</td>
<td>0.75</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Saturation signal</td>
<td>Vsat (^{1})</td>
<td>400</td>
<td>mV</td>
<td>2</td>
<td>Ta = 60°C</td>
<td>No line addition (^{2})</td>
<td></td>
</tr>
<tr>
<td>Vsat2 (^{3})</td>
<td>400</td>
<td>2</td>
<td>2-line addition (^{4})</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Smear</td>
<td>Sm</td>
<td>–100</td>
<td>–92</td>
<td>3</td>
<td>Progressive scan mode (^{5})</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>–94</td>
<td>–86</td>
<td>2/4-line readout mode (^{6})</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>–88</td>
<td>–80</td>
<td>2/8-line readout mode (^{6})</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Video signal shading</td>
<td>SH</td>
<td>20</td>
<td>%</td>
<td>4</td>
<td>Zone 0 and 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>25</td>
<td>Zone 0 to II</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Uniformity between video signal channels</td>
<td>ΔSrg</td>
<td>8</td>
<td>%</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ΔSbg</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dark signal</td>
<td>Vdt</td>
<td>8</td>
<td>mV</td>
<td>6</td>
<td>Ta = 60°C, 14,955 frames/s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dark signal shading</td>
<td>ΔVdt</td>
<td>2</td>
<td>mV</td>
<td>7</td>
<td>Ta = 60°C, 14,955 frames/s, (^{10})</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Line crawl G</td>
<td>Lcg</td>
<td>3.8</td>
<td>%</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Line crawl R</td>
<td>Lcr</td>
<td>3.8</td>
<td>%</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Line crawl B</td>
<td>Lcb</td>
<td>3.8</td>
<td>%</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Leg</td>
<td>Lag</td>
<td>0.5</td>
<td>%</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\(^{1}\) Vsat2 is the saturation signal level in 2-line addition mode, and is 200 mV per pixel.

\(^{2}\) Progressive scan mode, 2/8-line readout mode, 2/4-line readout mode, and center scan modes (1) and (3).

\(^{3}\) 2-line addition mode and center scan mode (2).

\(^{4}\) Same for 2-line addition mode and center scan modes (2) and (3).

\(^{5}\) Same for center scan mode (1).

\(^{6}\) Same for AF modes (1) and (2).

\(^{7}\) Excludes vertical dark signal shading caused by vertical register high-speed transfer.

### Spectral Sensitivity Characteristics (excludes lens characteristics and light source characteristics)

![Spectral Sensitivity Characteristics](image-url)
23.6.5.3 Gray scale version

### Image Sensor Characteristics

(\(T_a = 25^\circ C\))

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Unit</th>
<th>Measurement method</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>S</td>
<td>335</td>
<td>420</td>
<td>545</td>
<td>mV</td>
<td>1</td>
<td>1/30s accumulation</td>
</tr>
<tr>
<td>Saturation signal</td>
<td>(V_{sat})</td>
<td>400</td>
<td></td>
<td></td>
<td>mV</td>
<td>2</td>
<td>(T_a = 60^\circ C) No line addition(^{22}) 2-line addition(^{23})</td>
</tr>
<tr>
<td></td>
<td>(V_{sat2}^{21})</td>
<td>400</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Smear</td>
<td>(S_m)</td>
<td></td>
<td>-100</td>
<td>-92</td>
<td>dB</td>
<td>3</td>
<td>Progressive scan mode(^{24})</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>-94</td>
<td>-86</td>
<td></td>
<td></td>
<td>2/4-line readout mode(^{25})</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>-88</td>
<td>-80</td>
<td></td>
<td></td>
<td>2/6-line readout mode(^{26})</td>
</tr>
<tr>
<td>Video signal shading</td>
<td>(S_H)</td>
<td>20</td>
<td></td>
<td></td>
<td>%</td>
<td>4</td>
<td>Zone 0 and I</td>
</tr>
<tr>
<td></td>
<td></td>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Zone 0 to II</td>
</tr>
<tr>
<td>Dark signal</td>
<td>(V_{dlt})</td>
<td>8</td>
<td></td>
<td></td>
<td>mV</td>
<td>5</td>
<td>(T_a = 60^\circ C, 14.985) frame/s</td>
</tr>
<tr>
<td>Dark signal shading</td>
<td>(\Delta V_{dlt})</td>
<td>2</td>
<td></td>
<td></td>
<td>mV</td>
<td>6</td>
<td>(T_a = 60^\circ C, 14.985) frame/s, (^{27})</td>
</tr>
<tr>
<td>Lag</td>
<td>Lag</td>
<td>0.5</td>
<td></td>
<td></td>
<td>%</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

\(^{21}\) \(V_{sat2}\) is the saturation signal level in 2-line addition mode, and is 200 mV per pixel.

\(^{22}\) Progressive scan mode, 2/6-line readout mode, 2/4-line readout mode, and center scan modes (1) and (3).

\(^{23}\) 2-line addition mode and center scan mode (2).

\(^{24}\) Same for 2-line addition mode and center scan modes (2) and (3).

\(^{25}\) Same for center scan mode (1).

\(^{26}\) Same for AF modes (1) and (2).

\(^{27}\) Excludes vertical dark signal shading caused by vertical register high-speed transfer.

---

### Spectral Sensitivity Characteristics

(excludes lens characteristics and light source characteristics)

![Spectral Sensitivity Characteristics Graph](image-url)
1. Sensitivity
   Set to the standard imaging condition I. After selecting the electronic shutter mode with a shutter speed of 1/100s, measure the signal output \( (V_s) \) at the center of the screen, and substitute the values into the following formulas.

\[
S = V_s \times \frac{100}{30} \text{ [mV]}
\]

2. Saturation signal
   Set to the standard imaging condition II. After adjusting the luminous intensity to 10 times the intensity with the average value of the G channel signal output, 150mV, measure the minimum values of the signal outputs.

3. Smear
   Set to standard imaging condition II. With the lens diaphragm at F5.6 to F8, adjust the luminous intensity to 500 times the intensity with the average value of the signal output, 150mV. After the readout clock is stopped and the charge drain is executed by the electronic shutter at the respective H blankings, measure the maximum value \( (V_{sm} \text{ [mV]}) \) of the signal outputs, and substitute the values into the following formula. Smear in modes other than progressive scan mode is calculated from the storage time and signal addition method. As a result, 2-line addition mode and center scan modes (2) and (3) are the same as progressive scan mode, 2/4-line readout mode and center scan mode (1) are two times progressive scan mode, and 2/6-line readout mode and AF modes (1) and (2) are four times progressive scan mode.

\[
Sm = 20 \times \log \left( \frac{V_{sm}}{200} \times \frac{1}{500} \times \frac{1}{10} \right) \text{ [dB]} \text{ (1/10 V method conversion value)}
\]

4. Video signal shading
   Set to the standard imaging condition II. With the lens diaphragm at F5.6 to F8, adjusting the luminous intensity so that the average value of the signal output is 150mV. Then measure the maximum value \( (V_{max} \text{ [mV]}) \) and minimum value \( (V_{min} \text{ [mV]}) \) of the G signal output and substitute the values into the following formula.

\[
SH = \frac{(V_{max} - V_{min})}{150} \times 100 \% \]

5. Dark signal
   Measure the average value of the signal output \( (V_{dt} \text{ [mV]}) \) with the device ambient temperature of 60°C and the device in the light-obstructed state, using the horizontal idle transfer level as a reference.

6. Dark signal shading
   After measuring 5, measure the maximum \( (V_{dmax} \text{ [mV]}) \) and minimum \( (V_{dmin} \text{ [mV]}) \) values of the dark signal output and substitute the values into the following formula.

\[
\Delta V_{dt} = V_{dmax} - V_{dmin} \text{ [mV]}
\]

7. Lag
   Adjust the signal output generated by the strobe light to 150mV. After setting the strobe light so that it strobes with the following timing, measure the residual signal amount \( (V_{lag}) \). Substitute the value into the following formula.
23.7.1 Introduction

The CCD sensor is a highly programmable imaging module which will, for example, enable the following type of applications:

**Industrial applications:**
- triggered image acquisition with precise control of image exposure start by hardware trigger input.
- image acquisition of fast moving objects due to:
  - frame exposure, integrating all pixels at a time in contrast to CMOS imager which typically integrate line-by-line.
  - short shutter time, to get sharp images.
  - flash control output to have enough light for short time.

**Scientific applications:**
- long time exposure for low light conditions.
- optimizing image quality using the variable shutter control.

23.7.2 Details of operation

The process of getting an image from the CCD sensor can be separated into three different phases.

23.7.2.1 Trigger

When coming out of reset or ready with the last readout the CCD controller is waiting for a Trigger signal. The following trigger modes are available:
<table>
<thead>
<tr>
<th>Setting (GenICam)</th>
<th>Mode / Setting (Device Specific)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot;</td>
<td>Continuous</td>
<td>Free running, no external trigger signal needed.</td>
</tr>
<tr>
<td>&quot;TriggerMode = Off&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot;</td>
<td>OnDemand</td>
<td>Image acquisition triggered by command (software trigger).</td>
</tr>
<tr>
<td>&quot;TriggerMode = On&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSource = Software&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;ExposureMode = Timed&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>To trigger one frame execute the TriggerSoftw@i command then.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot;</td>
<td>OnLowLevel</td>
<td>Start an exposure of a frame as long as the trigger input is below the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerMode = On&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSource = &lt;desired Line&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerActivation = LevelLow&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;ExposureMode = Timed&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot;</td>
<td>OnHighLevel</td>
<td>Start an exposure of a frame as long as the trigger input is above the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerMode = On&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSource = &lt;desired Line&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerActivation = LevelHigh&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;ExposureMode = Timed&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot;</td>
<td>OnFallingEdge</td>
<td>Each falling edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>&quot;TriggerMode = On&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSource = &lt;desired Line&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerActivation = FallingEdge&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;ExposureMode = Timed&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot;</td>
<td>OnRisingEdge</td>
<td>Each rising edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>&quot;TriggerMode = On&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSource = &lt;desired Line&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerActivation = RisingEdge&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;ExposureMode = Timed&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot;</td>
<td>OnHighExpose</td>
<td>Each rising edge of trigger signal acquires one image, exposure time corresponds to pulse width.</td>
</tr>
<tr>
<td>&quot;TriggerMode = On&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSource = &lt;desired Line&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerActivation = LevelHigh&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;ExposureMode = TriggerWidth&quot;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
"TriggerSelector = FrameStart"
"TriggerMode = On"
"TriggerSource = <desired Line>"
"TriggerActivation = LevelLow"
"ExposureMode = TriggerWidth"

| OnLowExpose | Each falling edge of trigger signal acquires one image, exposure time corresponds to pulse width. |

| OnAnyEdge | Start the exposure of a frame when the trigger input level changes from high to low or from low to high. |

| "TriggerSelector = FrameStart"
"TriggerMode = On"
"TriggerSource = <desired Line>"
"TriggerActivation = AnyEdge"
"ExposureMode = Timed" | |

Line Mapping (TriggerSource mvIMPACT Acquire -> TriggerSource GenICam (valid values for <desired Line>)): |

| TriggerSource mvIMPACT Acquire | TriggerSource GenICam(BCX) |
| GP-IN0 | Line4 |
| GP-IN1 | Line5 |

23.7.2.2 Exposure aka Integration

After an active trigger, the exposure phase starts with a maximum jitter of $t_{\text{trig}}$. If flash illumination is enabled in software the flash output will be activated exactly while the sensor chip is integrating light. Exposure time is adjustable by software in increments of $t_{\text{readline}}$.

23.7.2.3 Readout

When exposure is finished, the image is transferred to hidden storage cells on the CCD. Image data is then shifted out line-by-line and transferred to memory. Shifting out non active lines takes $t_{\text{vshift}}$, while shifting out active lines will consume $t_{\text{readline}}$. The number of active pixels per line will not have any impact on readout speed.
23.7.3 CCD Timing

**Trigger Overlap: No**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Pixel clock</th>
</tr>
</thead>
<tbody>
<tr>
<td>trig</td>
<td>Time from trigger (internal or external) to exposure start</td>
<td>56 MHz</td>
</tr>
<tr>
<td>wait</td>
<td>Time after a new trigger is accepted</td>
<td>&lt; 1us</td>
</tr>
<tr>
<td>trans</td>
<td>Image transfer time (move image to readout cells in CCD)</td>
<td>42.30us</td>
</tr>
<tr>
<td>readline</td>
<td>time needed to readout a line</td>
<td>24.91us</td>
</tr>
<tr>
<td>vshift</td>
<td>time needed to shift unused lines away</td>
<td>4.57us</td>
</tr>
<tr>
<td>exposure</td>
<td>Exposure time</td>
<td>30us..20s</td>
</tr>
<tr>
<td>taps</td>
<td>Number of Taps</td>
<td>2 or 4</td>
</tr>
<tr>
<td>readout</td>
<td>Image readout time (move image from readout cells to memory)</td>
<td>t_readout = ((ActiveLines + 1) * t_readline / (taps / 2)) + ((1476 - ActiveLines) / (taps / 2))</td>
</tr>
</tbody>
</table>

**Note**

In partial scan mode (readout window ysize < 1460 lines).

To calculate the maximum frames per second (FPS\textsubscript{max}) you will need following formula (Exposure: No overlap):

\[
\text{FPS}_{\text{max}} = \frac{1}{t_{\text{readout}} + t_{\text{exposure}}}
\]

MATRIX VISION GmbH
(Expose mode: Overlapped):

\[
t_{\text{readout}} < t_{\text{exposure}}: \quad \text{FPS}\_\text{max} = \frac{1}{t_{\text{exposure}}}
\]

\[
t_{\text{readout}} > t_{\text{exposure}}: \quad \text{FPS}\_\text{max} = \frac{1}{t_{\text{readout}}}
\]

23.7.3.1.1 Example: Frame rate as function of lines & exposure time (Expose mode: No overlap)

Now, when we insert the values using exposure time of, for example, 8000 us, 1460 lines, 4 taps and 66MHz pixel clock:

\[
\text{FPS}\_\text{max} = \frac{1}{\frac{(1460 + 1) \times 21.14 \text{ us}}{4} + \frac{(1476 - 1460) \times 3.88 \text{ us}}{4} + 35.89 \text{ us}}
\]

\[
= \frac{1}{0.0006453836353943874} \approx 64.5 \text{ 1 / us}
\]

23.7.4 Reprogramming CCD Timing

Reprogramming the CCD Controller will happen when the following changes occur:

- Changing the exposure time
- Changing the capture window
- Changing trigger modes

Reprogram time consists of two phases

1. Time needed to send data to the CCD controller depending on what is changed
   exposure: abt 2..3ms
   window: abt 4..6ms
   trigger mode: from 5..90ms,
   varies with oldmode/newmode combination

2. Time to initialize (erase) the CCD chip after reprogramming this is fixed, abt 4.5 ms

So for example when reprogramming the capture window you will need (average values)

\[
t_{\text{regprog}} = \text{change}_\text{window} + \text{init}_\text{ccd}
\]

\[
t_{\text{regprog}} = 5\text{ms} + 4.5\text{ms}
\]

\[
t_{\text{regprog}} = 9.5\text{ms}
\]
23.7.5 CCD Sensor Data

Device Structure

- Interline CCD image sensor
- Image size: Diagonal 10.972mm (Type 2/3)
- Number of effective pixels: 1940 (H) x 1460 (V) approx. 2.83 Mpixels
- Total number of pixels: 2020 (H) x 1476 (V) approx. 2.98 Mpixels
- Chip size: 10.7mm (H) x 9.2mm (V)
- Unit cell size: 4.54um (H) x 4.54um (V)
- Optical black:
  - Horizontal (H) direction: Front 40 pixels each channel
  - Vertical (V) direction: Front 8 pixels each channel
- Number of dummy bits: Front 1 each channel
- Substrate material: Silicon

23.7.5.1 Characteristics

These zone definitions apply to both the color and gray scale version of the sensor.

Zone Definition of Video Signal Shading
23.7.5.2 Color version

**Image Sensor Characteristics**

\[(Ta = 25^\circ C)\]

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Unit</th>
<th>Measurement method</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>G sensitivity</td>
<td>Sg</td>
<td>640</td>
<td>800</td>
<td></td>
<td>mV</td>
<td>1</td>
<td>1/30s storage</td>
</tr>
<tr>
<td>Sensitivity ratio</td>
<td>R</td>
<td>0.40</td>
<td></td>
<td>0.70</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>0.22</td>
<td></td>
<td>0.52</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Saturation signal</td>
<td>Vsat</td>
<td>800</td>
<td></td>
<td></td>
<td>mV</td>
<td>2</td>
<td>Ta = 60\degree C</td>
</tr>
<tr>
<td>Smear</td>
<td>Sm</td>
<td></td>
<td>-105</td>
<td>-95</td>
<td>dB</td>
<td>3</td>
<td>Progressive scan mode</td>
</tr>
<tr>
<td>Video signal shading</td>
<td>SHg</td>
<td></td>
<td>20</td>
<td></td>
<td>%</td>
<td>4</td>
<td>Zone 0 and I</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td>Zone 0 to II'</td>
</tr>
<tr>
<td>Dark signal</td>
<td>Vdt</td>
<td>4</td>
<td></td>
<td></td>
<td>mV</td>
<td>5</td>
<td>Ta = 60\degree C, 15 frame/s</td>
</tr>
<tr>
<td>Dark signal shading</td>
<td>(\Delta V_{dt})</td>
<td>2</td>
<td></td>
<td></td>
<td>mV</td>
<td>6</td>
<td>Ta = 60\degree C, 15 frame/s *1</td>
</tr>
<tr>
<td>Line crawl R</td>
<td>Lcr</td>
<td>3.8</td>
<td></td>
<td></td>
<td>%</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Line crawl G</td>
<td>Lcg</td>
<td>3.8</td>
<td></td>
<td></td>
<td>%</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Line crawl B</td>
<td>Lcb</td>
<td>3.8</td>
<td></td>
<td></td>
<td>%</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Lag</td>
<td>Lag</td>
<td>0.5</td>
<td></td>
<td></td>
<td>%</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

The difference between the output signal for other channels (Vout2, Vout3, and Vout4) and the Vout1 output signal is maximum 10%.

*1 Excludes vertical dark signal shading caused by vertical register high-speed transfer.

---

**Spectral Sensitivity Characteristics** (Excludes lens characteristics and light source characteristics)
23.7.5.3 Gray scale version

**Image Sensor Characteristics**

\((T_a = 25^\circ C)\)

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Unit</th>
<th>Measurement method</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity 1</td>
<td>S1</td>
<td>760</td>
<td>950</td>
<td></td>
<td>mV</td>
<td>1</td>
<td>1/30s storage</td>
</tr>
<tr>
<td>Sensitivity 2</td>
<td>S2</td>
<td>3100</td>
<td></td>
<td></td>
<td>mV</td>
<td>2</td>
<td>1/30s storage</td>
</tr>
<tr>
<td>Saturation signal</td>
<td>Ysat</td>
<td>600</td>
<td></td>
<td></td>
<td>mV</td>
<td>3</td>
<td>(T_a = 60^\circ C)</td>
</tr>
<tr>
<td>Smear</td>
<td>Sm</td>
<td>−105</td>
<td>−95</td>
<td></td>
<td>dB</td>
<td>4</td>
<td>Progressive scan mode</td>
</tr>
<tr>
<td>Video signal shading</td>
<td>SH</td>
<td></td>
<td></td>
<td>20</td>
<td>%</td>
<td>5</td>
<td>Zone 0 and zone 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>25</td>
<td>%</td>
<td></td>
<td>Zone 0, zone I, zone II and zone II*</td>
</tr>
<tr>
<td>Dark signal</td>
<td>Vdt</td>
<td>4</td>
<td></td>
<td>6</td>
<td>mV</td>
<td>6</td>
<td>(T_a = 60^\circ C, 15) frame/s</td>
</tr>
<tr>
<td>Dark signal shading</td>
<td>(\Delta Vdt)</td>
<td>2</td>
<td></td>
<td>7</td>
<td>mV</td>
<td></td>
<td>(T_a = 60^\circ C, 15) frame/s, (*1)</td>
</tr>
<tr>
<td>Lag</td>
<td>Lag</td>
<td>0.5</td>
<td></td>
<td></td>
<td>%</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

The difference between the output signal for other channels (Vout2, Vout3, and Vout4) and the Vout1 output signal is maximum 10%.

\(*1\) Excludes vertical dark signal shading caused by vertical register high-speed transfer.

**Spectral Sensitivity Characteristics** (Excludes lens characteristics and light source characteristics)
Measurement conditions

1. In the following measurements, the device drive conditions are at the typical values of the bias and clock voltage conditions.

2. In the following measurements, spot pixels are excluded and, unless otherwise specified, the optical black (OB) level is used as the reference for the signal output, which is measured at point [FB] of the measurement system.

Definition of Standard Imaging Conditions

- **Standard imaging condition I:**
  Use a pattern box (luminance: 700cd/m², color temperature of 3200K halogen source) as a subject. (Pattern for evaluation is not applicable.) Use a testing standard lens with CM500S (f = 1.0mm) as an IR cut filter and image at F8. The luminous intensity to the sensor receiving surface at this point is defined as the standard sensitivity testing luminous intensity.

- **Standard imaging condition II:**
  Indicates that the IR cut filter is removed from the standard imaging condition I.

- **Standard imaging condition III:**
  Image a light source (color temperature of 3200K) with a uniformity of brightness within 2% at all angles. Use a testing standard lens with CM500S (f = 1.0mm) as an IR cut filter. The luminous intensity is adjusted to the value indicated in each testing item by the lens diaphragm.

1. **Sensitivity 1**
   Set the measurement condition to standard imaging condition I. After setting the electronic shutter mode with a shutter speed of 1/100s, measure the signal output (Vs1) at the center of the screen, and substitute the value into the following formula.

   \[ S1 = Vs1 \times (100/30) \text{[mV]} \]

2. **Sensitivity 2**
   Set the measurement condition to standard imaging condition II. After setting the electronic shutter mode with a shutter speed of 1/500s, measure the signal output (Vs2) at the center of the screen, and substitute the value into the following formula.

   \[ S2 = Vs2 \times (500/30) \text{[mV]} \]

3. **Saturation signal**
   Set the measurement condition to standard imaging condition III. After adjusting the luminous intensity to 10 times the intensity with the average value of the signal output, 200mV, measure the minimum value of the signal output.

4. **Smear**
   Set the measurement condition to standard imaging condition III. With the lens diaphragm at F5.6 to F8, adjust the luminous intensity to 500 times the intensity with the average value of the signal output, 200mV. When the readout clock is stopped and the charge drain is executed by the electronic shutter at the respective H blankings, measure the maximum value (VSm) of the signal output, and substitute the value into the following formula.

   \[ Sm = 20 \times \log \left( \frac{VSm}{200} \times \left( \frac{1}{500} \times \frac{1}{10} \right) \right) \text{[dB]} \text{ (1/10V method conversion value)} \]
5. Video signal shading
Set the measurement condition to standard imaging condition III. With the lens diaphragm at F5.6 to F8, adjust the luminous intensity so that the average value of the signal output is 200mV. Then measure the maximum value \( (V_{\text{max}}) \) and minimum value \( (V_{\text{min}}) \) of the output signal and substitute the values into the following formula.

\[
SH = \frac{(V_{\text{max}} - V_{\text{min}})}{200\text{mV}} \times 100\% 
\]

6. Dark signal
Measure the average value of the signal output \( (V_{\text{d}}) \) based on the horizontal idle transfer level at the device ambient temperature of 50°C placing the device in the light-obstructed state.

7. Dark signal shading
After the measurement item 6, measure the maximum value \( (V_{\text{dmax}}) \) and minimum value \( (V_{\text{dmin}}) \) of the dark signal output, and substitute the values into the following formula.

\[
\Delta V_{\text{d}} = V_{\text{dmax}} - V_{\text{dmin}} 
\]

8. Lag
Adjust the signal output value generated by strobe light to 200mV. After setting the strobe light so that it strobes with the following timing, measure the residual signal level \( (V_{\text{lag}}) \), and substitute the value into the following formula.

\[
\text{Lag} = \left(\frac{V_{\text{lag}}}{200}\right) \times 100\% 
\]

---

**Measurement System**

---

---

23.7.6 Device Feature And Property List

23.7.6.1 Gray scale version

23.7.6.2 Color version
23.8  mvBlueCOUGAR-X225 (5.1 Mpix [2448 x 2050])

23.8.1 Introduction

The CCD sensor is a highly programmable imaging module which will, for example, enable the following type of applications

**Industrial applications:**

- triggered image acquisition with precise control of image exposure start by hardware trigger input.
- image acquisition of fast moving objects due to:
  - frame exposure, integrating all pixels at a time in contrast to CMOS imager which typically integrate line-by-line.
  - short shutter time, to get sharp images.
  - flash control output to have enough light for short time.

**Scientific applications:**

- long time exposure for low light conditions.
- optimizing image quality using the variable shutter control.

23.8.2 Details of operation

The process of getting an image from the CCD sensor can be separated into three different phases.

23.8.2.1 Trigger

When coming out of reset or ready with the last readout the CCD controller is waiting for a Trigger signal.

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Setting (GenICam (p.143))</th>
<th>Mode / Setting (Device Specific (p.143))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = Off&quot;</td>
<td>Continuous</td>
<td>Free running, no external trigger signal needed.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = Software&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnDemand</td>
<td>Image acquisition triggered by command (software trigger).</td>
</tr>
</tbody>
</table>

To trigger one frame execute the `TriggerSoftware@i` command then.
<table>
<thead>
<tr>
<th>Trigger Configuration</th>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
</table>
| TriggerSelector = AcquisitionActive  
TriggerMode = On  
TriggerSource = <desired Line>  
TriggerActivation = LevelLow  
ExposureMode = Timed | OnLowLevel | Start an exposure of a frame as long as the trigger input is below the trigger threshold. (No Frame←Trigger!) |
| TriggerSelector = AcquisitionActive  
TriggerMode = On  
TriggerSource = <desired Line>  
TriggerActivation = LevelHigh  
ExposureMode = Timed | OnHighLevel | Start an exposure of a frame as long as the trigger input is above the trigger threshold. (No Frame←Trigger!) |
| TriggerSelector = FrameStart  
TriggerMode = On  
TriggerSource = <desired Line>  
TriggerActivation = FallingEdge  
ExposureMode = Timed | OnFallingEdge | Each falling edge of trigger signal acquires one image. |
| TriggerSelector = FrameStart  
TriggerMode = On  
TriggerSource = <desired Line>  
TriggerActivation = RisingEdge  
ExposureMode = Timed | OnRisingEdge | Each rising edge of trigger signal acquires one image. |
| TriggerSelector = FrameStart  
TriggerMode = On  
TriggerSource = <desired Line>  
TriggerActivation = LevelHigh  
ExposureMode = TriggerWidth | OnHighExpose | Each rising edge of trigger signal acquires one image, exposure time corresponds to pulse width. |
| TriggerSelector = FrameStart  
TriggerMode = On  
TriggerSource = <desired Line>  
TriggerActivation = LevelLow  
ExposureMode = TriggerWidth | OnLowExpose | Each falling edge of trigger signal acquires one image, exposure time corresponds to pulse width. |
| TriggerSelector = FrameStart | OnAnyEdge | Start the exposure of a frame when the trigger input level changes from high to low or from low to high. |
| "TriggerMode = On" | "TriggerSource = <desired Line>" | "TriggerActivation = AnyEdge" |
| "ExposureMode = Timed" |

23.8.2.2 Exposure aka Integration

After an active trigger, the exposure phase starts with a maximum jitter of \( t_{\text{trig}} \). If flash illumination is enabled in software the flash output will be activated exactly while the sensor chip is integrating light. Exposure time is adjustable by software in increments of \( t_{\text{readline}} \).

23.8.2.3 Readout

When exposure is finished, the image is transferred to hidden storage cells on the CCD. Image data is then shifted out line-by-line and transferred to memory. Shifting out non active lines takes \( t_{\text{vshift}} \), while shifting out active lines will consume \( t_{\text{readline}} \). The number of active pixels per line will not have any impact on readout speed.

### 23.8.3 CCD Timing

**Note**

In the "Trigger Overlap ReadOut" mode, the camera accepts a trigger signal the \( t_{\text{exposure}} \) period earlier than \( t_{\text{readout}} \) is finished.
23.8.3.1 Timings
### Table

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Pixel clock</th>
</tr>
</thead>
<tbody>
<tr>
<td>t_trig</td>
<td>Time from trigger (internal or external) to exposure start</td>
<td>60 MHz 66 MHz</td>
</tr>
<tr>
<td>t_wait</td>
<td>Time after a new trigger is accepted</td>
<td>&lt; 1us</td>
</tr>
<tr>
<td>t_trans</td>
<td>Image transfer time (move image to readout cells in CCD)</td>
<td>30.87us 28.06us</td>
</tr>
<tr>
<td>t_readline</td>
<td>time needed to readout a line</td>
<td>32.07us 29.15us</td>
</tr>
<tr>
<td>t_vshift</td>
<td>time needed to shift unused lines away</td>
<td>10.00us 9.09us</td>
</tr>
<tr>
<td>t_exposure</td>
<td>Exposure time</td>
<td>20us..20s 20us..20s</td>
</tr>
<tr>
<td>t_readout</td>
<td>Image readout time (move image from readout cells to memory)</td>
<td>t_readout = ((ActiveLines + 1) * t_readline) + ((2068 - ActiveLines) * t_vshift) + t_trans</td>
</tr>
</tbody>
</table>

### Note

In partial scan mode (readout window ysize < 2050 lines).

To calculate the maximum frames per second (FPS<sub>max</sub>) you will need following formula (Expose mode: No overlap):

\[
FPS_{\text{max}} = \frac{1}{t_{\text{readout}} + t_{\text{exposure}}}
\]

(Expose mode: Overlapped):

- \(t_{\text{readout}} < t_{\text{exposure}}\): \(FPS_{\text{max}} = \frac{1}{t_{\text{exposure}}}\)
- \(t_{\text{readout}} > t_{\text{exposure}}\): \(FPS_{\text{max}} = \frac{1}{t_{\text{readout}}}\)

### 23.8.3.1.1 Example: Frame rate as function of lines & exposure time (Expose mode: No overlap)

Now, when we insert the values using exposure time of, for example, 8000 us, 2050 lines and 60MHz pixel clock:

\[
FPS_{\text{max}} = \frac{1}{((2050 + 1) * 32.07 \text{ us}) + ((2068 - 2050) * 10.00 \text{ us}) + 30.87 \text{ us}) + 8000 \text{ us}} = 0.0000135159902274 \text{ 1 / us} = 13.5
\]

### 23.8.4 Reprogramming CCD Timing

Reprogramming the CCD Controller will happen when the following changes occur:

- Changing the exposure time
- Changing the capture window
- Changing Trigger Modes
Reprogram time consists of two phases

1. Time needed to send data to the CCD controller depending on what is changed
   - exposure: about 2.3 ms
   - window: about 4.6 ms
   - trigger mode: from 5.90 ms, varies with oldmode/newmode combination

2. Time to initialize (erase) the CCD chip after reprogramming this is fixed, about 4.5 ms

So for example when reprogramming the capture window you will need (average values)

\[ t_{\text{regprog}} = \text{change\_window} + \text{init\_ccd} \]

\[ t_{\text{regprog}} = 5\text{ms} + 4.5\text{ms} \]

\[ t_{\text{regprog}} = 9.5\text{ms} \]

23.8.5 CCD Sensor Data

Device Structure

- Interline CCD image sensor
- Image size: Diagonal 11.016 mm (Type 2/3)
- Number of effective pixels: 2456 (H) x 2058 (V) approximate 5.05 M pixels
- Total number of pixels: 2536 (H) x 2068 (V) approximate 5.24 M pixels
- Chip size: 9.93 mm (H) x 8.70 mm (V)
- Unit cell size: 3.45 um (H) x 3.45 um (V)
- Optical black:
  - Horizontal (H) direction: Front 40 pixels per channel
  - Vertical (V) direction: Front 8 pixels, rear 2 pixels
- Number of dummy bits: Horizontal 1 per channel Vertical 1
- Substrate material: Silicon

23.8.5.1 Characteristics

These zone definitions apply to both the color and gray scale version of the sensor.
23.8.5.2 Color version

**Image Sensor Characteristics (Ta = 25°C)**

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Unit</th>
<th>Measurement method</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>G sensitivity</td>
<td>Sg</td>
<td>330</td>
<td>420</td>
<td></td>
<td>mV</td>
<td>1</td>
<td>1/30s storage</td>
</tr>
<tr>
<td>Sensitivity ratio</td>
<td>R Rr</td>
<td>0.45</td>
<td>0.75</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B Rb</td>
<td>0.35</td>
<td>0.65</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Saturation signal</td>
<td>Vsat</td>
<td>360</td>
<td></td>
<td></td>
<td>mV</td>
<td>2</td>
<td>Ta = 60°C</td>
</tr>
<tr>
<td>Smear</td>
<td>Sm</td>
<td>-95</td>
<td>-90</td>
<td></td>
<td>dB</td>
<td>3</td>
<td>All-pixel scan mode*1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-83</td>
<td>-78</td>
<td></td>
<td></td>
<td>4/16-line readout mode</td>
<td></td>
</tr>
<tr>
<td>Video signal shading</td>
<td>SHg</td>
<td>20</td>
<td></td>
<td></td>
<td>%</td>
<td>4</td>
<td>Zone 0 and I</td>
</tr>
<tr>
<td></td>
<td></td>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Zone 0 to II*</td>
</tr>
<tr>
<td>Dark signal</td>
<td>Vdt</td>
<td>4</td>
<td></td>
<td></td>
<td>mV</td>
<td>5</td>
<td>Ta = 60°C, 15 frame/s</td>
</tr>
<tr>
<td>Dark signal shading</td>
<td>ΔVdt</td>
<td>2</td>
<td></td>
<td></td>
<td>mV</td>
<td>6</td>
<td>Ta = 60°C, 15 frame/s, *2</td>
</tr>
<tr>
<td>Line crawl R</td>
<td>Lcr</td>
<td>3.8</td>
<td></td>
<td></td>
<td>%</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Line crawl G</td>
<td>Lcg</td>
<td>3.8</td>
<td></td>
<td></td>
<td>%</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Line crawl B</td>
<td>Lcb</td>
<td>3.8</td>
<td></td>
<td></td>
<td>%</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Lag</td>
<td>Lag</td>
<td>0.5</td>
<td></td>
<td></td>
<td>%</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

The maximum difference of the output signals between right and left output (Vout1, Vout2) is 10%.

*1 Same for center scan mode.

*2 Excludes vertical dark signal shading caused by vertical register high-speed transfer.

**Spectral Sensitivity Characteristics (Includes lens characteristics, excludes light source characteristics)**

![Spectral Sensitivity Characteristics Graph]
23.8.5.3 Gray scale version

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Unit</th>
<th>Measurement method</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>S</td>
<td>330</td>
<td>420</td>
<td></td>
<td>mV</td>
<td>1</td>
<td>1/30s accumulation</td>
</tr>
<tr>
<td>Saturation signal</td>
<td>Vsat</td>
<td>360</td>
<td></td>
<td></td>
<td>mV</td>
<td>2</td>
<td>Ta = 60°C</td>
</tr>
<tr>
<td>Smear</td>
<td>Sm</td>
<td></td>
<td></td>
<td></td>
<td>dB</td>
<td>3</td>
<td>All-pixel scan mode*1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4/16-line readout mode</td>
</tr>
<tr>
<td>Video signal shading</td>
<td>SH</td>
<td></td>
<td></td>
<td></td>
<td>%</td>
<td>4</td>
<td>Zone 0 and I</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Zone 0 to II'</td>
</tr>
<tr>
<td>Dark signal</td>
<td>Vdt</td>
<td></td>
<td>4</td>
<td></td>
<td>mV</td>
<td>5</td>
<td>Ta = 60°C, 15 frame/s</td>
</tr>
<tr>
<td>Dark signal shading</td>
<td>ΔVdt</td>
<td></td>
<td>2</td>
<td></td>
<td>mV</td>
<td>6</td>
<td>Ta = 60°C, 15 frame/s, *2</td>
</tr>
<tr>
<td>Lag</td>
<td>Lag</td>
<td></td>
<td>0.5</td>
<td></td>
<td>%</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

The maximum difference of the output signals between right and left output (Vout1, Vout2) is 10%.

*1 Same for center scan mode.

*2 Excludes vertical dark signal shading caused by vertical register high-speed transfer.

Spectral Sensitivity Characteristics (Excludes lens characteristics and light source characteristics)
Measurement conditions

1. In the following measurements, the device drive conditions are at the typical values of the bias and clock voltage conditions.

2. In the following measurements, spot pixels are excluded and, unless otherwise specified, the optical black (OB) level is used as the reference for the signal output, which is taken as the value of the Gr/Gb signal output or the R/B signal output of the measurement system.

Definition of standard imaging conditions

- **Standard imaging condition I:**
  Use a pattern box (luminance: 706cd/m², color temperature of 3200K halogen source) as a subject. (Pattern for evaluation is not applicable.) Use a testing standard lens with CM500S (f = 1.0mm) as an IR cut filter and image at F5.6. The luminous intensity to the sensor receiving surface at this point is defined as the standard sensitivity testing luminous intensity.

- **Standard imaging condition II:**
  Image a light source (color temperature of 3200K) with a uniformity of brightness within 2% at all angles.
  Use a testing standard lens with CM500S (f = 1.0mm) as an IR cut filter. The luminous intensity is adjusted to the value indicated in each testing item by the lens diaphragm.

1. G sensitivity, sensitivity ratio
   Set the measurement condition to the standard imaging condition I. After setting the electronic shutter mode with a shutter speed of 1/100s, measure the signal outputs (VGr, VGb, VR, VB) at the center of each Gr, Gb, R and B channel screen, and substitute the values into the following formula.
   
   \[
   \begin{align*}
   V_G &= (V_{Gr} + V_{Gb})/2 \\
   S_g &= V_G \times 100/30 \ [mV] \\
   R_r &= V_R/V_G \\
   R_b &= V_B/V_G
   \end{align*}
   \]

2. Saturation signal
   Set the measurement condition to the standard imaging condition II. After adjusting the luminous intensity to 10 times the intensity with the average value of the Gr signal output, 150mV, measure the minimum values of the Gr, Gb, R and B signals.

3. Snear
   Set the measurement condition to the standard imaging condition II. With the lens diaphragm at F5.6 to F8, first adjust the average value of the Gr signal output to 150mV. Measure the average values of the Gr, Gb, R and B signal outputs (Gra, Gba, Ra, Ba), and then adjust the luminous intensity to 500 times the intensity with the average value of the Gr signal output, 150mV.
   After the readout clock is stopped and the charge drain is executed by the electronic shutter at the respective H blanking, measure the maximum value (VSm [mV]) independent of the Gr, Gb, R and B signal outputs, and substitute the values into the following formula.
   
   \[
   S_m = 20 \times \log \left[ \left( V_{Sm} \times (Gra + Gba + Ra + Ba/4) \right) \times (1/500) \times (1/10) \right] \ [dB] \ (1/10V \ method \ conversion \ value)
   \]

4. Video signal shading
   Set the measurement condition to the standard imaging condition II. With the lens diaphragm at F5.6 to F8, adjust the luminous intensity so that the average value of the signal output is 150mV. Then measure the maximum value (Gmax [mV]) and the minimum value (Gmin [mV]) of the Gr signal output, and substitute the values into the following formula.
   
   \[
   \text{SHg} = (G_{max} - G_{min})/150mV \times 100 \ [%]
   \]
5. Dark signal
   Measure the average value of the signal output (\(V_{\text{dlt}}\) [mV]) at the device ambient temperature of 60°C and the device in the light-obstructed state using the horizontal idle transfer level as a reference.

6. Dark signal shading
   After the measurement item 5, measure the maximum value (\(V_{\text{dmax}}\)) and the minimum value (\(V_{\text{dmin}}\)) of the dark signal output, and substitute the values into the following formula.
\[
\Delta V_{\text{dlt}} = V_{\text{dmax}} - V_{\text{dmin}}
\]

7. Line crawl
   Set the measurement condition to the standard imaging condition II. After adjusting the average value of the Gr signal output to 150mV, insert R, G and B filters and measure the difference between G signal lines (\(\Delta G_{\text{r}}, \Delta G_{\text{g}}, \Delta G_{\text{b}}\) [mV]) as well as the average value of the G signal output (\(G_{\text{r}}, G_{\text{g}}, G_{\text{b}}\)). Substitute the values into the following formula.
\[
L_{\text{ci}} = \frac{\Delta G_i}{G_{\text{ai}}} \times 100\% (i = r, g, b)
\]

8. Lag
   Adjust the Gr signal output value generated by strobe light to 150mV. After setting the strobe light so that it strobes with the following timing, measure the residual signal level (\(V_{\text{lag}}\)), and substitute the value into the following formula.
\[
L_{\text{ag}} = \frac{V_{\text{lag}}}{150} \times 100\%
\]
23.9 mvBlueCOUGAR-Xx25a (5.1 Mpix [2448 x 2050])

23.9.1 Introduction

The CCD sensor is a highly programmable imaging module which will, for example, enable the following type of applications

**Industrial applications:**

- triggered image acquisition with precise control of image exposure start by hardware trigger input.
- image acquisition of fast moving objects due to:
  - frame exposure, integrating all pixels at a time in contrast to CMOS imager which typically integrate line-by-line.
  - short shutter time, to get sharp images.
  - flash control output to have enough light for short time.

**Scientific applications:**

- long time exposure for low light conditions.
- optimizing image quality using the variable shutter control.

23.9.2 Details of operation

The process of getting an image from the CCD sensor can be separated into three different phases.

23.9.2.1 Trigger

When coming out of reset or ready with the last readout the CCD controller is waiting for a Trigger signal.

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Setting (GenICam (p.143))</th>
<th>Mode / Setting (Device Specific (p.143))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot;  &quot;TriggerMode = Off&quot;</td>
<td>Continuous</td>
<td>Free running, no external trigger signal needed.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot;  &quot;TriggerMode = On&quot;  &quot;TriggerSource = Software&quot;  &quot;ExposureMode = Timed&quot;</td>
<td>OnDemand</td>
<td>Image acquisition triggered by command (software trigger).</td>
</tr>
</tbody>
</table>

To trigger one frame execute the TriggerSoftware@i command then.
<table>
<thead>
<tr>
<th>Trigger Configuration</th>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OnLowLevel</td>
<td></td>
<td>Start an exposure of a frame as long as the trigger input is below the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>OnHighLevel</td>
<td></td>
<td>Start an exposure of a frame as long as the trigger input is above the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>OnFallingEdge</td>
<td></td>
<td>Each falling edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>OnRisingEdge</td>
<td></td>
<td>Each rising edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>OnHighExposure</td>
<td></td>
<td>Each rising edge of trigger signal acquires one image, exposure time corresponds to pulse width.</td>
</tr>
<tr>
<td>OnLowExposure</td>
<td></td>
<td>Each falling edge of trigger signal acquires one image, exposure time corresponds to pulse width.</td>
</tr>
</tbody>
</table>
"TriggerSelector = FrameStart"
"TriggerMode = On"
"TriggerSource = <desired Line>"
"TriggerActivation = AnyEdge"
"ExposureMode = Timed"

<table>
<thead>
<tr>
<th>TriggerSource mvIMPACT Acquire</th>
<th>TriggerSource GenICam(BCX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP-IN0</td>
<td>Line4</td>
</tr>
<tr>
<td>GP-IN1</td>
<td>Line5</td>
</tr>
</tbody>
</table>

23.9.2.2  Exposure aka Integration

After an active trigger, the exposure phase starts with a maximum jitter of \(t_{\text{trig}}\). If flash illumination is enabled in software the flash output will be activated exactly while the sensor chip is integrating light. Exposure time is adjustable by software in increments of \(t_{\text{readline}}\).

23.9.2.3  Readout

When exposure is finished, the image is transferred to hidden storage cells on the CCD. Image data is then shifted out line-by-line and transferred to memory. Shifting out non active lines takes \(t_{\text{vshift}}\), while shifting out active lines will consume \(t_{\text{readline}}\). The number of active pixels per line will not have any impact on readout speed.

23.9.3  CCD Timing

**Note**

In the "Trigger Overlap ReadOut" mode, the camera accepts a trigger signal the \(t_{\text{exposure}}\) period earlier than \(t_{\text{readout}}\) is finished.
23.9.3.1 Timings
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Pixel clock</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_{\text{trig}}$</td>
<td>Time from trigger (internal or external) to exposure start</td>
<td>60 MHz</td>
</tr>
<tr>
<td>$t_{\text{wait}}$</td>
<td>Time after a new trigger is accepted</td>
<td>$&lt; 1\text{us}$</td>
</tr>
<tr>
<td>$t_{\text{trans}}$</td>
<td>Image transfer time (move image to readout cells in CCD)</td>
<td>30.87us</td>
</tr>
<tr>
<td>$t_{\text{readline}}$</td>
<td>Time needed to readout a line</td>
<td>53.20us</td>
</tr>
<tr>
<td>$t_{\text{vshift}}$</td>
<td>Time needed to shift unused lines away</td>
<td>10.00us</td>
</tr>
<tr>
<td>$t_{\text{exposure}}$</td>
<td>Exposure time</td>
<td>20us..20s</td>
</tr>
<tr>
<td>$t_{\text{readout}}$</td>
<td>Image readout time (move image from readout cells to memory)</td>
<td>$t_{\text{readout}} = ((\text{ActiveLines} + 1) \times t_{\text{readline}}) + ((2068 - \text{ActiveLines}) \times t_{\text{vshift}}) + t_{\text{trans}}$</td>
</tr>
</tbody>
</table>

**Note**

In partial scan mode (readout window ysize $< 2050$ lines).

To calculate the maximum frames per second ($\text{FPS}_{\text{max}}$) you will need following formula (Expose mode: No overlap):

$$\text{FPS}_{\text{max}} = \frac{1}{t_{\text{readout}} + t_{\text{exposure}}}$$

(Expose mode: Overlapped):

$$t_{\text{readout}} < t_{\text{exposure}}: \quad \text{FPS}_{\text{max}} = \frac{1}{t_{\text{exposure}}}$$

$$t_{\text{readout}} > t_{\text{exposure}}: \quad \text{FPS}_{\text{max}} = \frac{1}{t_{\text{readout}}}$$

**23.9.3.1.1 Example: Frame rate as function of lines & exposure time (Expose mode: No overlap)**

Now, when we insert the values using exposure time of, for example, 8000 us, 2050 lines and 60MHz pixel clock:

$$\text{FPS}_{\text{max}} = \frac{1}{((2050 + 1) \times 53.20 \text{us}) + ((2068 - 2050) \times 10.00 \text{us}) + 30.87 \text{us}) + 8000 \text{us}}$$

$$= 0.00000852340018549 \frac{1}{\text{us}} = 8.5$$

**23.9.4 Reprogramming CCD Timing**

Reprogramming the CCD Controller will happen when the following changes occur:

- Changing the exposure time
- Changing the capture window
- Changing Trigger Modes
Reprogram time consists of two phases

1. Time needed to send data to the CCD controller depending on what is changed
   exposure: abt 2.3ms
   window: abt 4.6ms
   trigger mode: from 5.90ms,
   varies with oldmode/newmode combination

2. Time to initialize (erase) the CCD chip after reprogramming this is fixed, abt 4.5 ms

So for example when reprogramming the capture window you will need (average values)

\[ t_{\text{regprog}} = \text{change\_window} + \text{init\_ccd} \]

\[ t_{\text{regprog}} = 5\text{ms} + 4.5\text{ms} \]

\[ t_{\text{regprog}} = 9.5\text{ms} \]

23.9.5 CCD Sensor Data

Device Structure

- Interline CCD image sensor
- Image size: Diagonal 11.016mm (Type 2/3)
- Number of effective pixels: 2456 (H) x 2058 (V) approx. 5.05 M pixels
- Total number of pixels: 2536 (H) x 2068 (V) approx. 5.24 M pixels
- Chip size: 9.93mm (H) x 8.70mm (V)
- Unit cell size: 3.45um (H) x 3.45um (V)
- Optical black:
  - Horizontal (H) direction: Front 40 pixels, rear 40 pixels
  - Vertical (V) direction: Front 8 pixels, rear 2 pixels
- Number of dummy bits: Horizontal 1 per channel Vertical 1
- Substrate material: Silicon

23.9.5.1 Characteristics

These zone definitions apply to both the color and gray scale version of the sensor.

---

**Zone Definition of Video Signal Shading**
### Image Sensor Characteristics

$\text{(Ta} = 25^\circ\text{C)}$

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Unit</th>
<th>Measurement method</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>Sg</td>
<td>330</td>
<td>420</td>
<td></td>
<td>mV</td>
<td>1</td>
<td>1/30s storage</td>
</tr>
<tr>
<td>Sensitivity ratio</td>
<td>R Rr</td>
<td>0.45</td>
<td>0.75</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B Rb</td>
<td>0.35</td>
<td>0.65</td>
<td></td>
<td></td>
<td></td>
<td>Ta = 60°C</td>
</tr>
<tr>
<td>Saturation signal</td>
<td>Vsat</td>
<td>360</td>
<td></td>
<td></td>
<td>mV</td>
<td>2</td>
<td>Ta = 60°C</td>
</tr>
<tr>
<td>Smear</td>
<td>Sm</td>
<td>−95</td>
<td>−90</td>
<td>−83</td>
<td>−78</td>
<td>dB</td>
<td>3</td>
</tr>
<tr>
<td>Video signal shading</td>
<td>SHg</td>
<td>20</td>
<td></td>
<td>25</td>
<td>%</td>
<td>4</td>
<td>Zones 0 and I</td>
</tr>
<tr>
<td>Dark signal</td>
<td>Vdt</td>
<td>4</td>
<td></td>
<td></td>
<td>mV</td>
<td>5</td>
<td>Ta = 60°C, 9 frame/s</td>
</tr>
<tr>
<td>Dark signal shading</td>
<td>ΔVdt</td>
<td>2</td>
<td></td>
<td></td>
<td>mV</td>
<td>6</td>
<td>Ta = 60°C, 9 frame/s$^*$2</td>
</tr>
<tr>
<td>Line crawl R</td>
<td>Lcr</td>
<td>3.8</td>
<td></td>
<td></td>
<td>%</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Line crawl G</td>
<td>Lcg</td>
<td>3.8</td>
<td></td>
<td></td>
<td>%</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Line crawl B</td>
<td>Lcb</td>
<td>3.8</td>
<td></td>
<td></td>
<td>%</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Lag</td>
<td>Lag</td>
<td>0.5</td>
<td></td>
<td></td>
<td>%</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

$^*$1 Same for center scan mode.

$^*$2 Excludes vertical dark signal shading caused by vertical register high-speed transfer.

---

**Spectral Sensitivity Characteristics** *(Includes lens characteristics, excludes light source characteristics)*

![Spectral Sensitivity Characteristics Graph](image)
23.9.5.3  Gray scale version

### Image Sensor Characteristics

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Unit</th>
<th>Measurement method</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>S</td>
<td>330</td>
<td>420</td>
<td></td>
<td>mV</td>
<td>1</td>
<td>1/30s storage</td>
</tr>
<tr>
<td>Saturation signal</td>
<td>Vsat</td>
<td>360</td>
<td></td>
<td></td>
<td>mV</td>
<td>2</td>
<td>Ta = 60°C</td>
</tr>
<tr>
<td>Smear</td>
<td>Sm</td>
<td></td>
<td></td>
<td>-95</td>
<td>-90</td>
<td>dB</td>
<td>4/16-line readout mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-83</td>
<td>-78</td>
<td></td>
<td>All-pixel scan mode *¹</td>
</tr>
<tr>
<td>Video signal shading</td>
<td>SH</td>
<td></td>
<td></td>
<td>20</td>
<td>%</td>
<td>4</td>
<td>Zones 0 and I</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>25</td>
<td></td>
<td></td>
<td>Zones 0 to II</td>
</tr>
<tr>
<td>Dark signal</td>
<td>Vdt</td>
<td></td>
<td></td>
<td>4</td>
<td>mV</td>
<td>5</td>
<td>Ta = 60°C, 9 frame/s</td>
</tr>
<tr>
<td>Dark signal shading</td>
<td>ΔVdt</td>
<td></td>
<td></td>
<td>2</td>
<td>mV</td>
<td>6</td>
<td>Ta = 60°C, 9 frame/s *²</td>
</tr>
<tr>
<td>Lag</td>
<td>Lag</td>
<td></td>
<td>0.5</td>
<td></td>
<td>%</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

*¹  Same for center scan mode.
*²  Excludes vertical dark signal shading caused by vertical register high-speed transfer.

### Spectral Sensitivity Characteristics

(Spectra are given for the following wavelengths: 400–1000 nm. 
Excludes lens characteristics and light source characteristics. 

![Spectral Sensitivity Characteristics Graph](image_url)
Measurement conditions

1. In the following measurements, the device drive conditions are at the typical values of the bias and clock voltage conditions.

2. In the following measurements, spot pixels are excluded and, unless otherwise specified, the optical black (OB) level is used as the reference for the signal output, which is taken as the value of the Gr/Gb signal output or the R/B signal output of the measurement system.

Definition of standard imaging conditions

* Standard imaging condition I:
  Use a pattern box (luminance: 706cd/m², color temperature of 3200K halogen source) as a subject. (Pattern for evaluation is not applicable.) Use a testing standard lens with CM500S (t = 1.0mm) as an IR cut filter and image at F5.6. The luminous intensity to the sensor receiving surface at this point is defined as the standard sensitivity testing luminous intensity.

* Standard imaging condition II:
  Image a light source (color temperature of 3200K) with a uniformity of brightness within 2% at all angles. Use a testing standard lens with CM500S (t = 1.0mm) as an IR cut filter. The luminous intensity is adjusted to the value indicated in each testing item by the lens diaphragm.

1. G sensitivity, sensitivity ratio
   Set the measurement condition to the standard imaging condition I. After setting the electronic shutter mode with a shutter speed of 1/100s, measure the signal outputs (VGr, VGb, VR, VB) at the center of each Gr, Gb, R and B channel screen, and substitute the values into the following formula.
   \[ V_G = (V_{Gr} + V_{Gb})/2 \]
   \[ S_G = V_G \times 100/30 \text{ [mV]} \]
   \[ R_R = V_R/V_G \]
   \[ R_B = V_B/V_G \]

2. Saturation signal
   Set the measurement condition to the standard imaging condition II. After adjusting the luminous intensity to 10 times the intensity with the average value of the Gr signal output, 150mV, measure the minimum values of the Gr, Gb, R and B signals.

3. Snear
   Set the measurement condition to the standard imaging condition II. With the lens diaphragm at F5.6 to F8, first adjust the average value of the Gr signal output to 150mV. Measure the average values of the Gr, Gb, R and B signal outputs (Gra, Gba, Ra, Ba), and then adjust the luminous intensity to 500 times the intensity with the average value of the Gr signal output, 150mV. After the readout clock is stopped and the charge drain is executed by the electronic shutter at the respective H blankings, measure the maximum value (VSm [mV]) independent of the Gr, Gb, R and B signal outputs, and substitute the values into the following formula.
   \[ Sm = 20 \times \log [(V_{Sm} \times (Gra + Gba + Ra + Ba/4)) \times (1/500) \times (1/10)] \text{ [dB]} \text{ (1/10V method conversion value)} \]

4. Video signal shading
   Set the measurement condition to the standard imaging condition II. With the lens diaphragm at F5.6 to F8, adjust the luminous intensity so that the average value of the signal output is 150mV. Then measure the maximum value (Gmax [mV]) and the minimum value (Gmin [mV]) of the Gr signal output, and substitute the values into the following formula.
   \[ SH_G = (G_{max} - G_{min})/150mV \times 100 \text{ [%]} \]
5. Dark signal
   Measure the average value of the signal output (Vdt) based on the horizontal idle transfer level at the
device ambient temperature of 60°C placing the device in the light-obstructed state.

6. Dark signal shading
   After the measurement item 5, measure the maximum (Vdmax) and minimum (Vdmin) values of the dark
signal output and substitute the values into the following formula.
   \[ \Delta Vdt = Vdmax - Vdmin \]

7. Line crawl
   Set the measurement condition to standard imaging condition II. Adjust the luminous intensity so that the
average value of the Gr signal output is 150 mV, and then insert R, G, and B filters and measure the
difference between G signal lines (\( \Delta G_{ir}, \Delta G_{ig}, \Delta G_{ib} \) [mV]) as well as the average value of the G signal
output (Gar, Gag, Gab). Substitute the values into the following formula.
   \[ Lci = \Delta G_{ii} / Gai \times 100 \% \] (i = r, g, b)

8. Lag
   Adjust the Gr signal output value generated by strobe light to 150 mV. After setting the strobe light so that it
strokes with the following timing, measure the residual signal (Vlag), and substitute the value into the following formula.
   \[ Lag = (Vlag / 150) \times 100 \% \]

---

Measurement System

The measurement system diagram includes the following components:

- **CCD**
- **C.D.S**
- **AMP**
- **S/H**
- **Gr/Gb channel signal output**
- **R/B channel signal output**
23.10.1 Introduction

The CCD sensor is a highly programmable imaging module which will, for example, enable the following type of applications:

**Industrial applications:**

- triggered image acquisition with precise control of image exposure start by hardware trigger input.
- image acquisition of fast moving objects due to:
  - frame exposure, integrating all pixels at a time in contrast to CMOS imager which typically integrate line-by-line.
  - short shutter time, to get sharp images.
  - flash control output to have enough light for short time.

**Scientific applications:**

- long time exposure for low light conditions.
- optimizing image quality using the variable shutter control.

23.10.2 Details of operation

The process of getting an image from the CCD sensor can be separated into three different phases.

23.10.2.1 Trigger

When coming out of reset or ready with the last readout the CCD controller is waiting for a Trigger signal.

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Setting (GenICam [p.143])</th>
<th>Mode / Setting (Device Specific [p.143])</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot;</td>
<td>Continuous</td>
<td>Free running, no external trigger signal needed.</td>
</tr>
<tr>
<td>&quot;TriggerMode = Off&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot;</td>
<td>OnDemand</td>
<td>Image acquisition triggered by command (software trigger).</td>
</tr>
<tr>
<td>&quot;TriggerMode = On&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSource = Software&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;ExposureMode = Timed&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>To trigger one frame execute the TriggerSoftware@i command then.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot;</td>
<td>OnLowLevel</td>
<td>Start an exposure of a frame as long as the trigger input is below the trigger threshold. (No Frame ← Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerMode = On&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSource = &lt;desired Line&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerActivation = LevelLow&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;ExposureMode = Timed&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerSelector</td>
<td>TriggerMode</td>
<td>TriggerSource</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------</td>
<td>---------------</td>
</tr>
<tr>
<td>TriggerSelector =</td>
<td>TriggerMode =</td>
<td>TriggerSource =</td>
</tr>
<tr>
<td>AcquisitionActive</td>
<td>On</td>
<td>&lt;desired Line&gt;</td>
</tr>
<tr>
<td>TriggerSelector =</td>
<td>TriggerMode =</td>
<td>TriggerSource =</td>
</tr>
<tr>
<td>FrameStart</td>
<td>On</td>
<td>&lt;desired Line&gt;</td>
</tr>
<tr>
<td>TriggerSelector =</td>
<td>TriggerMode =</td>
<td>TriggerSource =</td>
</tr>
<tr>
<td>FrameStart</td>
<td>On</td>
<td>&lt;desired Line&gt;</td>
</tr>
<tr>
<td>TriggerSelector =</td>
<td>TriggerMode =</td>
<td>TriggerSource =</td>
</tr>
<tr>
<td>FrameStart</td>
<td>On</td>
<td>&lt;desired Line&gt;</td>
</tr>
<tr>
<td>TriggerSelector =</td>
<td>TriggerMode =</td>
<td>TriggerSource =</td>
</tr>
<tr>
<td>FrameStart</td>
<td>On</td>
<td>&lt;desired Line&gt;</td>
</tr>
</tbody>
</table>

Line Mapping (TriggerSource mvIMPACT Acquire → TriggerSource GenICam (valid values for <desired Line>)):

<table>
<thead>
<tr>
<th>TriggerSource mvIMPACT Acquire</th>
<th>TriggerSource GenICam(BCX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP-IN0</td>
<td>Line4</td>
</tr>
<tr>
<td>GP-IN1</td>
<td>Line5</td>
</tr>
</tbody>
</table>
23.10.2.2 Exposure aka Integration

After an active trigger, the exposure phase starts with a maximum jitter of \( t_{\text{trig}} \). If flash illumination is enabled in software the flash output will be activated exactly while the sensor chip is integrating light. Exposure time is adjustable by software in increments of \( t_{\text{readline}} \).

23.10.2.3 Readout

When exposure is finished, the image is transferred to hidden storage cells on the CCD. Image data is then shifted out line-by-line and transferred to memory. Shifting out non active lines takes \( t_{\text{vshift}} \), while shifting out active lines will consume \( t_{\text{readline}} \). The number of active pixels per line will not have any impact on readout speed.

23.10.3 CCD Timing

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Pixel clock</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_{\text{trig}} )</td>
<td>Time from trigger (internal or external) to exposure start</td>
<td>56 MHz</td>
</tr>
<tr>
<td>( t_{\text{wait}} )</td>
<td>Time after a new trigger is accepted</td>
<td>( t_{\text{trig}} = t_{\text{vshift}} + t_{\text{turn_on_delay_opto}} ) ( t_{\text{turn_on_delay_opto}} = \approx 2,\mu\text{s} )</td>
</tr>
<tr>
<td>( t_{\text{trans}} )</td>
<td>Image transfer time (move image to readout cells in CCD)</td>
<td>(&lt; 1,\mu\text{s})</td>
</tr>
<tr>
<td>( t_{\text{readline}} )</td>
<td>time needed to readout a line</td>
<td>42.3,\mu\text{s}</td>
</tr>
<tr>
<td>( t_{\text{vshift}} )</td>
<td>time needed to shift unused lines away</td>
<td>32.21,\mu\text{s}</td>
</tr>
<tr>
<td>( t_{\text{exposure}} )</td>
<td>Exposure time</td>
<td>30,\mu\text{s}\ldots20,\second</td>
</tr>
<tr>
<td>( t_{\text{readout}} )</td>
<td>Image readout time (move image from readout cells to memory)</td>
<td>( t_{\text{readout}} ) = ( (\text{ActiveLines} + 1) \times t_{\text{readline}} / (\text{taps}/2) ) ( (2224 - \text{ActiveLines}) / (\text{taps}/2) )</td>
</tr>
</tbody>
</table>
Note

In partial scan mode (readout window ysize < 2208 lines).

To calculate the maximum frames per second (FPS\textsubscript{max}) you will need following formula (Expose mode: No overlap):

\[
\text{FPS\textsubscript{max}} = \frac{1}{t_{\text{readout}} + t_{\text{exposure}}}
\]

(Expose mode: Overlapped):

\[
\begin{align*}
t_{\text{readout}} < t_{\text{exposure}}: & \quad \text{FPS\textsubscript{max}} = \frac{1}{t_{\text{exposure}}} \\
t_{\text{readout}} > t_{\text{exposure}}: & \quad \text{FPS\textsubscript{max}} = \frac{1}{t_{\text{readout}}}
\end{align*}
\]

23.10.3.1.1 Example: Frame rate as function of lines & exposure time (Expose mode: No overlap)

Now, when we insert the values using exposure time of, for example, 8000 us, 2208 lines, 4 taps and 66MHz pixel clock:

\[
\begin{align*}
\text{FPS\textsubscript{max}} = \frac{1}{((2208 + 1) \times 27.33 \text{ us} / (4 / 2)) + (((2224 - 2208) / (4 / 2)) \times 3.88 \text{ us}) + 35.89 \text{ us}}
\end{align*}
\]

\[
= \frac{1}{0.000033054666597185759 \text{ 1 / us}}
\]

\[
= 33.1
\]

23.10.4 Reprogramming CCD Timing

Reprogramming the CCD Controller will happen when the following changes occur

- Changing the exposure time
- Changing the capture window
- Changing trigger modes

Reprogram time consists of two phases

1. Time needed to send data to the CCD controller depending on what is changed
   - exposure: abt 2.3ms
   - window: abt 4.6ms
   - trigger mode: from 5..90ms,
   - varies with oldmode/newmode combination

2. Time to initialize (erase) the CCD chip after reprogramming this is fixed, abt 4.5 ms

So for example when reprogramming the capture window you will need (average values)

\[
\begin{align*}
t_{\text{regprog}} &= \text{change_window} + \text{init_ccd} \\
t_{\text{regprog}} &= 5\text{ms} + 4.5\text{ms} \\
t_{\text{regprog}} &= 9.5\text{ms}
\end{align*}
\]
Device Structure

- Interline CCD image sensor

- Image size: Diagonal 15.989mm (Type 1)

- Number of effective pixels: 2758 (H) x 2208 (V) approx. 6.09 Mpixels

- Total number of pixels: 2838 (H) x 2224 (V) approx. 6.31 Mpixels

- Chip size: 14.6mm (H) x 12.8mm (V)

- Unit cell size: 4.54um (H) x 4.54um (V)

- Optical black:
  - Horizontal (H) direction: Front 40 pixels each channel
  - Vertical (V) direction: Front 8 pixels each channel

- Number of dummy bits: Front 1 each channel

- Substrate material: Silicon

23.10.5.1 Characteristics

These zone definitions apply to both the color and gray scale version of the sensor.
Image Sensor Characteristics

\((Ta = 25°C)\)

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Unit</th>
<th>Measurement method</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>G Sensitivity</td>
<td>Sg</td>
<td>800</td>
<td>1000</td>
<td>519</td>
<td>mV</td>
<td>1/30s storage</td>
<td></td>
</tr>
<tr>
<td>Sensitivity ratio</td>
<td>R</td>
<td>0.40</td>
<td>0.70</td>
<td>0.52</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>Rb</td>
<td>0.22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Saturation signal</td>
<td>Vsat</td>
<td>800</td>
<td></td>
<td></td>
<td>dB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Smear</td>
<td>Sm</td>
<td>-110</td>
<td>-100</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Video signal shading</td>
<td>SH</td>
<td>20</td>
<td>25</td>
<td></td>
<td>%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dark signal</td>
<td>Vdt</td>
<td>8</td>
<td></td>
<td></td>
<td>mV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dark signal shading</td>
<td>ΔVdt</td>
<td>4</td>
<td></td>
<td></td>
<td>mV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Line crawl R</td>
<td>Lcr</td>
<td>3.8</td>
<td></td>
<td></td>
<td>%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Line crawl G</td>
<td>Lcg</td>
<td>3.8</td>
<td></td>
<td></td>
<td>%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Line crawl B</td>
<td>Lcb</td>
<td>3.8</td>
<td></td>
<td></td>
<td>%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lag</td>
<td>Lg</td>
<td>0.5</td>
<td></td>
<td></td>
<td>%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The maximum difference of the output signal right and left output \((V_{OUTL}, V_{OUTR})\) is 10 %

*1 Note that the video signal shading may increase when using this sensor combined with a lens of which exit pupil distance short.

*2 Excludes vertical dark signal shading caused by vertical register high-speed transfer.

Spectral Sensitivity Characteristics (Excludes lens characteristics and light source characteristics)
Image Sensor Characteristics

\( (T_a = 25^\circ C) \)

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Unit</th>
<th>Measurement method</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity 1</td>
<td>S1</td>
<td>800</td>
<td>1000</td>
<td>mV</td>
<td>1</td>
<td>1/30 s storage</td>
<td></td>
</tr>
<tr>
<td>Sensitivity 2</td>
<td>S2</td>
<td>3250</td>
<td></td>
<td>mV</td>
<td>2</td>
<td>1/30 s storage</td>
<td></td>
</tr>
<tr>
<td>Saturation signal</td>
<td>Vsat</td>
<td>800</td>
<td></td>
<td>mV</td>
<td>3</td>
<td>Ta = 60 °C</td>
<td></td>
</tr>
<tr>
<td>Smear</td>
<td>Sm</td>
<td>-110</td>
<td>-100</td>
<td>dB</td>
<td>4</td>
<td>Progressive scan mode</td>
<td></td>
</tr>
<tr>
<td>Video signal shading</td>
<td>SH</td>
<td>20</td>
<td></td>
<td>%</td>
<td>5</td>
<td>Zone 0 and zone 1</td>
<td>Zone 0, zone I, zone II and zone II</td>
</tr>
<tr>
<td>Dark signal</td>
<td>Vdt</td>
<td>4</td>
<td>mV</td>
<td>6</td>
<td>Ta = 60 °C, 15 frame/s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dark signal shading</td>
<td>( \Delta V_{dt} )</td>
<td>2</td>
<td>mV</td>
<td>7</td>
<td>Ta = 60 °C, 15 frame/s, (^1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lag</td>
<td>Lag</td>
<td>0.5</td>
<td>%</td>
<td>8</td>
<td>(^1) Excludes vertical dark signal shading caused by vertical register high-speed transfer.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The difference between the output signal for other channels (Vout2, Vout3, and Vout4) and the Vout1 output signal is maximum 10 %.

\(^1\) Excludes vertical dark signal shading caused by vertical register high-speed transfer.

Spectral Sensitivity Characteristics (Excludes lens characteristics and light source characteristics)
Measurement conditions

1. In the following measurements, the device drive conditions are at the typical values of the bias and clock voltage conditions.
2. In the following measurements, spot pixels are excluded and, unless otherwise specified, the optical black (OB) level is used as the reference for the signal output, which is measured at point [B] of the measurement system.

Definition of Standard Imaging Conditions

- **Standard imaging condition I:**
  Use a pattern box (luminance: 706 cd/m², color temperature of 3200 K halogen source) as a subject. (Pattern for evaluation is not applicable.) Use a testing standard lens with CM500S (f = 1.0 mm) as an IR cut filter and image at F8. The luminous intensity to the sensor receiving surface at this point is defined as the standard sensitivity testing luminous intensity.

- **Standard imaging condition II:**
  Indicates a state that the IR cut filter is removed from the standard imaging condition I.

- **Standard imaging condition III:**
  Image a light source (color temperature of 3200 K) with a uniformity of brightness within 2 % at all angles. Use a testing standard lens with CM500S (f = 1.0 mm) as an IR cut filter. The luminous intensity is adjusted to the value indicated in each testing item by the lens diaphragm.

1. **Sensitivity 1**
   Set the measurement condition to standard imaging condition I. After setting the electronic shutter mode with a shutter speed of $1/100$ s, measure the signal output ($V_{s1}$) at the center of the screen, and substitute the value into the following formula.
   \[ S1 = V_{s1} \times (100/30) \text{ [mV]} \]

2. **Sensitivity 2**
   Set the measurement condition to standard imaging condition II. After setting the electronic shutter mode with a shutter speed of $1/500$ s, measure the signal output ($V_{s2}$) at the center of the screen, and substitute the value into the following formula.
   \[ S2 = V_{s2} \times (500/30) \text{ [mV]} \]

3. **Saturation signal**
   Set the measurement condition to standard imaging condition III. After adjusting the luminous intensity to 10 times the intensity with the average value of the signal output, 200 mV, measure the minimum value of the signal output.

4. **Smear**
   Set the measurement condition to standard imaging condition III. With the lens diaphragm at F5.6 to F8, adjust the luminous intensity to 500 times the intensity with the average value of the signal output, 200 mV. When the readout clock is stopped and the charge drain is executed by the electronic shutter at the respective H blankings, measure the maximum value ($V_{Sm}$) of the signal output, and substitute the value into the following formula.
   \[ Sm = 20 \times \log ((V_{Sm}/200) \times (1/500) \times (1/10)) \text{ [dB]} (1/10 \text{ V method conversion value}) \]
5. Video signal shading
Set the measurement condition to standard imaging condition III. With the lens diaphragm at F5.6 to F8, adjust the luminous intensity so that the average value of the signal output is 200 mV. Then measure the maximum value ($V_{\text{max}}$) and minimum value ($V_{\text{min}}$) of the output signal and substitute the values into the following formula.

$$SH = \frac{(V_{\text{max}} - V_{\text{min}})}{200 \text{ mV}} \times 100 \text{ [%]}$$

6. Dark signal
Measure the average value of the signal output ($V_{\text{dlt}}$) based on the horizontal idle transfer level at the device ambient temperature of 60 °C placing the device in the light-obstructed state.

7. Dark signal shading
After the measurement item 6, measure the maximum value ($V_{\text{dmax}}$) and minimum value ($V_{\text{dmin}}$) of the dark signal output, and substitute the values into the following formula.

$$\Delta V_{\text{dlt}} = V_{\text{dmax}} - V_{\text{dmin}}$$

8. Lag
Adjust the signal output value generated by strobe light to 200 mV. After setting the strobe light so that it strobos with the following timing, measure the residual signal level ($V_{\text{lag}}$), and substitute the value into the following formula.

$$\text{Lag} = \left( \frac{V_{\text{lag}}}{200} \right) \times 100 \text{ [%]}$$

---

**Measurement System**

![Measurement System Diagram](attachment:image.png)

23.10.6 Device Feature And Property List

23.10.6.1 Gray scale version

23.10.6.2 Color version
23.11 Introduction

The CCD sensor is a highly programmable imaging module which will, for example, enable the following type of applications:

**Industrial applications:**

- triggered image acquisition with precise control of image exposure start by hardware trigger input.
- image acquisition of fast moving objects due to:
  - frame exposure, integrating all pixels at a time in contrast to CMOS imager which typically integrate line-by-line.
  - short shutter time, to get sharp images.
  - flash control output to have enough light for short time.

**Scientific applications:**

- long time exposure for low light conditions.
- optimizing image quality using the variable shutter control.

23.11.2 Details of operation

The process of getting an image from the CCD sensor can be separated into three different phases.

23.11.2.1 Trigger

When coming out of reset or ready with the last readout the CCD controller is waiting for a Trigger signal.

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Setting (GenICam (p.143))</th>
<th>Mode / Setting (Device Specific (p.143))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = Off&quot;</td>
<td>Continuous</td>
<td>Free running, no external trigger signal needed.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = Software&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnDemand</td>
<td>Image acquisition triggered by command (software trigger).</td>
</tr>
</tbody>
</table>

To trigger one frame execute the TriggerSoftware@i command then.
<table>
<thead>
<tr>
<th>TriggerSelector</th>
<th>Mode</th>
<th>Source</th>
<th>Activation</th>
<th>ExposureMode</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>AcquisitionActive</td>
<td>On</td>
<td>&lt;desired Line&gt;</td>
<td>LevelLow</td>
<td>Timed</td>
<td>OnLowLevel - Start an exposure of a frame as long as the trigger input is below the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>OnLowLevel</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AcquisitionActive</td>
<td>On</td>
<td>&lt;desired Line&gt;</td>
<td>LevelHigh</td>
<td>Timed</td>
<td>OnHighLevel - Start an exposure of a frame as long as the trigger input is above the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>OnHighLevel</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FrameStart</td>
<td>On</td>
<td>&lt;desired Line&gt;</td>
<td>FallingEdge</td>
<td>Timed</td>
<td>OnFallingEdge - Each falling edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>OnFallingEdge</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FrameStart</td>
<td>On</td>
<td>&lt;desired Line&gt;</td>
<td>RisingEdge</td>
<td>Timed</td>
<td>OnRisingEdge - Each rising edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>OnRisingEdge</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FrameStart</td>
<td>On</td>
<td>&lt;desired Line&gt;</td>
<td>LevelHigh</td>
<td>TriggerWidth</td>
<td>OnHighExpose - Each rising edge of trigger signal acquires one image, exposure time corresponds to pulse width.</td>
</tr>
<tr>
<td>OnHighExpose</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FrameStart</td>
<td>On</td>
<td>&lt;desired Line&gt;</td>
<td>LevelLow</td>
<td>TriggerWidth</td>
<td>OnLowExpose - Each falling edge of trigger signal acquires one image, exposure time corresponds to pulse width.</td>
</tr>
<tr>
<td>OnLowExpose</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
"TriggerSelector = FrameStart"
"TriggerMode = On"
"TriggerSource = <desired Line>"
"TriggerActivation = AnyEdge"
"ExposureMode = Timed"

| OnAnyEdge | Start the exposure of a frame when the trigger input level changes from high to low or from low to high. |

Line Mapping (TriggerSource mvIMPACT Acquire -> TriggerSource GenICam (valid values for <desired Line>)):

<table>
<thead>
<tr>
<th>TriggerSource mvIMPACT Acquire</th>
<th>TriggerSource GenICam(BCX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP-IN0</td>
<td>Line4</td>
</tr>
<tr>
<td>GP-IN1</td>
<td>Line5</td>
</tr>
</tbody>
</table>

23.11.2.2 Exposure aka Integration

After an active trigger, the exposure phase starts with a maximum jitter of $t_{\text{trig}}$. If flash illumination is enabled in software the flash output will be activated exactly while the sensor chip is integrating light. Exposure time is adjustable by software in increments of $t_{\text{readline}}$.

23.11.2.3 Readout

When exposure is finished, the image is transferred to hidden storage cells on the CCD. Image data is then shifted out line-by-line and transferred to memory. Shifting out non active lines takes $t_{\text{vshift}}$, while shifting out active lines will consume $t_{\text{readline}}$. The number of active pixels per line will not have any impact on readout speed.

23.11.3 CCD Timing

In the "Trigger Overlap ReadOut" mode, the camera accepts a trigger signal the $t_{\text{exposure}}$ period earlier than $t_{\text{readout}}$ is finished.
23.11.3.1 Timings
### Table 1: Timing Parameters

| Name            | Description                                                                 | Pixel Clock  
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_{\text{trig}}$</td>
<td>Time from trigger (internal or external) to exposure start</td>
<td>56 MHz</td>
</tr>
<tr>
<td>$t_{\text{wait}}$</td>
<td>Time after a new trigger is accepted</td>
<td>$&lt; 1\text{us}$</td>
</tr>
<tr>
<td>$t_{\text{trans}}$</td>
<td>Image transfer time (move image to readout cells in CCD)</td>
<td>42.3 us</td>
</tr>
<tr>
<td>$t_{\text{readline}}$</td>
<td>Time needed to readout a line</td>
<td>32.21 us</td>
</tr>
<tr>
<td>$t_{\text{vshift}}$</td>
<td>Time needed to shift unused lines away</td>
<td>4.57 us</td>
</tr>
<tr>
<td>$t_{\text{exposure}}$</td>
<td>Exposure time</td>
<td>30us..20s</td>
</tr>
<tr>
<td>$t_{\text{readout}}$</td>
<td>Image readout time (move image from readout cells to memory)</td>
<td>$t_{\text{readout}} = ((\text{ActiveLines} + 1) * t_{\text{readline}} / (\text{taps} / 2)) + (((2224 \text{ - ActiveLines}) / (\text{taps} / 2)) * t_{\text{vshift}}) + t_{\text{trans}}$</td>
</tr>
</tbody>
</table>

**Note**

In partial scan mode (readout window ysize $< 2208$ lines).

To calculate the maximum frames per second ($FPS_{\text{max}}$) you will need following formula (Expose mode: No overlap):

\[
FPS_{\text{max}} = \frac{1}{t_{\text{readout}} + t_{\text{exposure}}}
\]

(Expose mode: Overlapped):

- $t_{\text{readout}} < t_{\text{exposure}}$:  
  \[
  FPS_{\text{max}} = \frac{1}{t_{\text{exposure}}}
  \]

- $t_{\text{readout}} > t_{\text{exposure}}$:  
  \[
  FPS_{\text{max}} = \frac{1}{t_{\text{readout}}}
  \]

**Example: Frame rate as function of lines & exposure time**

Now, when we insert the values using exposure time of, for example, 8000 us, 2208 lines, 2 taps and 56MHz pixel clock:

\[
FPS_{\text{max}} = \frac{1}{(((2208 + 1) \times 32.21\text{us} / (2 / 2)) + (((2224 - 2208) / (2 / 2)) \times 4.57\text{us}) + 42.3\text{us})}\]
\[
= 0.000001403016787599 \text{ 1 / us}
\]
\[
= 14
\]
23.11.4 Reprogramming CCD Timing

Reprogramming the CCD Controller will happen when the following changes occur

- Changing the exposure time
- Changing the capture window
- Changing trigger modes

Reprogram time consists of two phases

1. Time needed to send data to the CCD controller depending on what is changed
   - exposure: abt 2...3ms
   - window: abt 4..6ms
   - trigger mode: from 5..90ms, varies with oldmode/newmode combination

2. Time to initialize (erase) the CCD chip after reprogramming this is fixed, abt 4.5 ms

So for example when reprogramming the capture window you will need (average values)

\[ t_{\text{regprog}} = \text{change\_window} + \text{init\_ccd} \]

\[ t_{\text{regprog}} = 5\text{ms} + 4.5\text{ms} \]

\[ t_{\text{regprog}} = 9.5\text{ms} \]

23.11.5 CCD Sensor Data

Device Structure

- Interline CCD image sensor
- Image size: Diagonal 15.989mm (Type 1)
- Number of effective pixels: 2758 (H) x 2208 (V) approx. 6.09 Mpxels
- Total number of pixels: 2838 (H) x 2224 (V) approx. 6.31 Mpxels
- Chip size: 14.6mm (H) x 12.8mm (V)
- Unit cell size: 4.54um (H) x 4.54um (V)
- Optical black:
  - Horizontal (H) direction: Front 40 pixels each channel
  - Vertical (V) direction: Front 8 pixels each channel
- Number of dummy bits: Front 1 each channel
- Substrate material: Silicon
23.11.5.1 Characteristics

These zone definitions apply to both the color and gray scale version of the sensor.

![Zone Definition of Video Signal Shading](image)

23.11.5.2 Color version

**Image Sensor Characteristics**

\[ Ta = 25^\circ C \]

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Unit</th>
<th>Measurement method</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>G Sensitivity</td>
<td>Gs</td>
<td>800</td>
<td>1000</td>
<td></td>
<td>mV</td>
<td>1</td>
<td>1/30 s storage</td>
</tr>
<tr>
<td>Sensitivity ratio</td>
<td>Rr</td>
<td>0.40</td>
<td>0.70</td>
<td></td>
<td>mV</td>
<td>2</td>
<td>Tj = 60 °C</td>
</tr>
<tr>
<td>B</td>
<td>Rb</td>
<td>0.22</td>
<td>0.52</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Saturation signal</td>
<td>Vsat</td>
<td>800</td>
<td></td>
<td></td>
<td>mV</td>
<td>3</td>
<td>Progressive scan mode</td>
</tr>
<tr>
<td>Smear</td>
<td>Sm</td>
<td>-110</td>
<td>-100</td>
<td></td>
<td>dB</td>
<td>4</td>
<td>Zone 0 and zone I, zone II and zone III [1]</td>
</tr>
<tr>
<td>Video signal shading</td>
<td>SH</td>
<td>20</td>
<td>25</td>
<td></td>
<td>%</td>
<td>4</td>
<td>Zone 0, zone I, zone II and zone III [1]</td>
</tr>
<tr>
<td>Dark signal</td>
<td>Vdt</td>
<td>8</td>
<td></td>
<td></td>
<td>mV</td>
<td>5</td>
<td>Tj = 60 °C, 7.5 frames/s [2]</td>
</tr>
<tr>
<td>Dark signal shading</td>
<td>△Vdt</td>
<td>4</td>
<td></td>
<td></td>
<td>mV</td>
<td>6</td>
<td>Tj = 60 °C, 7.5 frames/s [2]</td>
</tr>
<tr>
<td>Line crawl R</td>
<td>Lcr</td>
<td>3.8</td>
<td></td>
<td></td>
<td>%</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Line crawl G</td>
<td>Log</td>
<td>3.8</td>
<td></td>
<td></td>
<td>%</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Line crawl B</td>
<td>Lcb</td>
<td>3.8</td>
<td></td>
<td></td>
<td>%</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Lag</td>
<td>Lag</td>
<td>0.5</td>
<td></td>
<td></td>
<td>%</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

*1 Note that the video signal shading may increase when using this sensor combined with a lens of which exit pupil distance short
*2 Excludes vertical dark signal shading caused by vertical register high-speed transfer

![Spectral Sensitivity Characteristics](image)

*FILTER IR-CUT
### Image Sensor Characteristics

\( (T_a = 25°C) \)

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Unit</th>
<th>Measurement method</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity 1</td>
<td>S1</td>
<td>800</td>
<td>1000</td>
<td></td>
<td>mV</td>
<td>1/30 s storage</td>
<td></td>
</tr>
<tr>
<td>Sensitivity 2</td>
<td>S2</td>
<td></td>
<td></td>
<td>3250</td>
<td>mV</td>
<td>1/30 s storage</td>
<td></td>
</tr>
<tr>
<td>Saturation signal</td>
<td>Vsat</td>
<td>800</td>
<td></td>
<td></td>
<td>mV</td>
<td>3</td>
<td>( T_j = 60°C )</td>
</tr>
<tr>
<td>Smear</td>
<td>Sm</td>
<td></td>
<td></td>
<td></td>
<td>dB</td>
<td>4</td>
<td>Progressive scan mode</td>
</tr>
<tr>
<td>Video signal shading</td>
<td>SH</td>
<td></td>
<td></td>
<td></td>
<td>%</td>
<td>5</td>
<td>Zone 0, zone I, zone II and zone III ( ^1 )</td>
</tr>
<tr>
<td>Dark signal</td>
<td>Vdt</td>
<td>8</td>
<td></td>
<td></td>
<td>mV</td>
<td>6</td>
<td>( T_j = 60°C, 7.5 \text{ frame/s} ) ( ^2 )</td>
</tr>
<tr>
<td>Dark signal shading</td>
<td>( \Delta \text{Vdt} )</td>
<td></td>
<td></td>
<td></td>
<td>mV</td>
<td>7</td>
<td>( T_j = 60°C, 7.5 \text{ frame/s} ) ( ^2 )</td>
</tr>
<tr>
<td>Lag</td>
<td>Lag</td>
<td></td>
<td></td>
<td></td>
<td>%</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

1. The maximum difference of the output signal right and left output \( | V_{G1} - V_{G2} | \leq 10 \% \)

2. Note that the video signal shading may increase when using this sensor combined with a lens of which exit pupil distance is short.

2. Excludes vertical dark signal shading caused by vertical register high-speed transfer.

### Spectral Sensitivity Characteristics

(Excludes lens characteristics and light source characteristics)

![Spectral Sensitivity Graph](image)
Measurement conditions
1. in the following measurements, the device drive conditions are at the typical values of the bias and clock voltage conditions.
2. in the following measurements, spot pixels are excluded and, unless otherwise specified, the optical black (OB) level is used as the reference for the signal output, which is taken as the value of the Gr/GB signal output or the R/B signal output of the measurement system.

Definition of Standard Imaging Conditions
- **Standard imaging condition I:**
  Use a pattern box (luminance: 706 cd/m², color temperature of 3200 K halogen source) as a subject. (Pattern for evaluation is not applicable.) Use a testing standard lens with CM500S (t = 1.0 mm) as an IR cut filter and image at F5.6. The luminous intensity to the sensor receiving surface at this point is defined as the standard sensitivity testing luminous intensity.

- **Standard imaging condition II:**
  Image a light source (color temperature of 3200 K) with a uniformity of brightness within 2 % at all angles. Use a testing standard lens with CM500S (t = 1.0 mm) as an IR cut filter. The luminous intensity is adjusted to the value indicated in each testing item by the lens diaphragm.

1. G sensitivity, sensitivity ratio
   Set the measurement condition to the standard imaging condition I. After setting the electronic shutter mode with a shutter speed of 1/100 s, measure the signal output (Vgr, Vga, Vra, Vba) at the center of each Gr, Gb, R and B channel screen, and substitute the values into the following formula.
   
   \[
   \begin{align*}
   V_g & = (V_g + V_a) / 2 \\
   S_g & = V_g 	imes (100/30) \text{ [mV]} \\
   R_g & = V_g/V_a \\
   R_b & = V_b/V_a
   \end{align*}
   \]

2. Saturation signal
   Set the measurement condition to standard imaging condition II. After adjusting the luminous intensity to 20 times the intensity with the average value of the Gr signal output, 200 mV, measure the minimum value of the Gr, Gb, R and B signals.

3. Smaar
   Set the measurement condition to standard imaging condition II. With the lens diaphragm at F5.6 to F8, first adjust the average value of the Gr signal output to 200 mV. Measure the average values of the Gr, Gb, R and B signal outputs (Gra, Gba, Ra, Ba), and then adjust the luminous intensity to 500 times the intensity with the average value of the Gr signal output, 200 mV. After the readout clock is stopped and the charge drain is executed by the electronic shutter at the respective H blankings, measure the maximum value (VSm [mV]) independent of the Gr, Gb, R and B signal outputs, and substitute the value into the following formula.
   
   \[
   S_m = 20 \times \log \left( \frac{V_Sm}{(Gra + Gba + Ra + Ba/4)} \times \frac{1}{500} \right) \times (1/10) \text{ [dB]} \quad (1/10 \text{ V method conversion value})
   \]

4. Video signal shading
   Set the measurement condition to standard imaging condition II. With the lens diaphragm at F5.6 to F8, adjust the luminous intensity so that the average value of the signal output is 200 mV. Then measure the maximum value (Gmax) and the minimum value (Gmin) of the Gr signal output, and substitute the values into the following formula.
   
   \[
   \text{Shg} = \frac{(G_{\text{max}} - G_{\text{min}})}{200 \text{ mV}} \times 100 \% \]
5. Dark signal
Measure the average value of the signal output (\(V_{dt}\)) based on the horizontal idle transfer level at the junction temperature of 60 °C placing the device in the light-obstructed state.

8. Dark signal shading
After the measurement item 5, measure the maximum value (\(V_{dmax}\)) and minimum value (\(V_{dmin}\)) of the dark signal output, and substitute the values into the following formula.

\[
\Delta V_{dt} = V_{dmax} - V_{dmin}
\]

7. Line crawl
Set the measurement condition to the standard imaging condition II. After adjusting the average value of the Gr signal output to 200 mV, intert R, G and B filters and measure the difference between G signal lines (\(\Delta G_{ri}, \Delta G_{gi}, \Delta G_{bi}\) [mV]) as well as the average value of the G signal output (Gar, Gag, Gab). Substitute the values into the following formula.

\[
L_{ci} = \frac{\Delta G_{ri}}{G_{ci}} \times 100 \% (i = r, g, b)
\]

8. Lag
Adjust the signal output value generated by strobe light to 200 mV. After setting the strobe light so that it strobes with the following timing, measure the residual signal level (\(V_{lag}\)), and substitute the value into the following formula.

\[
L_{ag} = \frac{V_{lag}}{200} \times 100 \%
\]

---

**Measurement System**

![Diagram of measurement system](image)

23.11.6 Device Feature And Property List

23.11.6.1 Gray scale version

23.11.6.2 Color version
23.12  mvBlueCOUGAR-XD129 (9.2 Mpix [3384 x 2712])

23.12.1 Introduction

The CCD sensor is a highly programmable imaging module which will, for example, enable the following type of applications

**Industrial applications:**

- triggered image acquisition with precise control of image exposure start by hardware trigger input.
- image acquisition of fast moving objects due to:
  - frame exposure, integrating all pixels at a time in contrast to CMOS imager which typically integrate line-by-line.
  - short shutter time, to get sharp images.
  - flash control output to have enough light for short time.

**Scientific applications:**

- long time exposure for low light conditions.
- optimizing image quality using the variable shutter control.

23.12.2 Details of operation

The process of getting an image from the CCD sensor can be separated into three different phases.

23.12.2.1 Trigger

When coming out of reset or ready with the last readout the CCD controller is waiting for a Trigger signal.

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Setting (GenICam (p.143))</th>
<th>Mode / Setting (Device Specific (p.143))</th>
<th>Description</th>
</tr>
</thead>
</table>
| "TriggerSelector = FrameStart"
  "TriggerMode = Off"         | Continuous                                 | Free running, no external trigger signal needed. |
| "TriggerSelector = FrameStart"
  "TriggerMode = On"
  "TriggerSource = Software"
  "ExposureMode = Timed"      | OnDemand                                   | Image acquisition triggered by command (software trigger). |

To trigger one frame execute the `TriggerSoftware@i` command then.
<table>
<thead>
<tr>
<th>TriggerSelector</th>
<th>TriggerMode</th>
<th>TriggerSource</th>
<th>TriggerActivation</th>
<th>ExposureMode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AcquisitionActive</td>
<td>On</td>
<td>&lt;desired Line&gt;</td>
<td>LevelLow</td>
<td>Timed</td>
<td>OnLowLevel Start an exposure of a frame as long as the trigger input is below the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>AcquisitionActive</td>
<td>On</td>
<td>&lt;desired Line&gt;</td>
<td>LevelHigh</td>
<td>Timed</td>
<td>OnHighLevel Start an exposure of a frame as long as the trigger input is above the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>FrameStart</td>
<td>On</td>
<td>&lt;desired Line&gt;</td>
<td>FallingEdge</td>
<td>Timed</td>
<td>OnFallingEdge Each falling edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>FrameStart</td>
<td>On</td>
<td>&lt;desired Line&gt;</td>
<td>RisingEdge</td>
<td>Timed</td>
<td>OnRisingEdge Each rising edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>FrameStart</td>
<td>On</td>
<td>&lt;desired Line&gt;</td>
<td>LevelHigh</td>
<td>TriggerWidth</td>
<td>OnHighExpose Each rising edge of trigger signal acquires one image, exposure time corresponds to pulse width.</td>
</tr>
<tr>
<td>FrameStart</td>
<td>On</td>
<td>&lt;desired Line&gt;</td>
<td>LevelLow</td>
<td>TriggerWidth</td>
<td>OnLowExpose Each falling edge of trigger signal acquires one image, exposure time corresponds to pulse width.</td>
</tr>
</tbody>
</table>
"TriggerSelector = FrameStart"
"TriggerMode = On"
"TriggerSource = <desired Line>">
"TriggerActivation = AnyEdge"
"ExposureMode = Timed"

| OnAnyEdge | Start the exposure of a frame when the trigger input level changes from high to low or from low to high. |

OnAnyEdge

Line Mapping (TriggerSource mvIMPACT Acquire -> TriggerSource GenICam (valid values for <desired Line>)):

<table>
<thead>
<tr>
<th>TriggerSource mvIMPACT Acquire</th>
<th>TriggerSource GenICam(BCX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP-IN0</td>
<td>Line4</td>
</tr>
<tr>
<td>GP-IN1</td>
<td>Line5</td>
</tr>
</tbody>
</table>

23.12.2.2 Exposure aka Integration

After an active trigger, the exposure phase starts with a maximum jitter of $t_{\text{trig}}$. If flash illumination is enabled in software the flash output will be activated exactly while the sensor chip is integrating light. Exposure time is adjustable by software in increments of $t_{\text{readline}}$.

23.12.2.3 Readout

When exposure is finished, the image is transferred to hidden storage cells on the CCD. Image data is then shifted out line-by-line and transferred to memory. Shifting out non active lines takes $t_{\text{vshift}}$, while shifting out active lines will consume $t_{\text{readline}}$. The number of active pixels per line will not have any impact on readout speed.

23.12.3 CCD Timing

Note

In the "Trigger Overlap ReadOut" mode, the camera accepts a trigger signal the $t_{\text{exposure}}$ period earlier than $t_{\text{readout}}$ is finished.

MATRIX VISION GmbH
23.12.3.1 Timings
### Table

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Pixel clock</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_{\text{trig}} )</td>
<td>Time from trigger (internal or external) to exposure start ( t_{\text{trig}} = t_{\text{vshift}} + t_{\text{turn_on_delay_opto}} ) ( (t_{\text{turn_on_delay_opto}} = \text{approx.} 2 \text{us}) )</td>
<td>56 MHz</td>
</tr>
<tr>
<td>( t_{\text{wait}} )</td>
<td>Time after a new trigger is accepted ( &lt; 1 \text{us} )</td>
<td>( &lt; 1 \text{us} )</td>
</tr>
<tr>
<td>( t_{\text{trans}} )</td>
<td>Image transfer time (move image to readout cells in CCD) ( 59.64 \text{us} )</td>
<td>( 50.61 \text{us} )</td>
</tr>
<tr>
<td>( t_{\text{readline}} )</td>
<td>Time needed to readout a line ( 38.88 \text{us} )</td>
<td>( 32.98 \text{us} )</td>
</tr>
<tr>
<td>( t_{\text{vshift}} )</td>
<td>Time needed to shift unused lines away ( 5.86 \text{us} )</td>
<td>( 4.97 \text{us} )</td>
</tr>
<tr>
<td>( t_{\text{exposure}} )</td>
<td>Exposure time ( 30 \text{us}..20 \text{s} )</td>
<td>( 30 \text{us}..20 \text{s} )</td>
</tr>
<tr>
<td>( t_{\text{readout}} )</td>
<td>Image readout time (move image from readout cells to memory) ( t_{\text{readout}} = ((\text{ActiveLines} + 1) + t_{\text{readline}} / (\text{taps} / 2)) + (((2728 - \text{ActiveLines}) / (\text{taps} / 2)) \ast t_{\text{vshift}}) + t_{\text{trans}} )</td>
<td>( t_{\text{readout}} )</td>
</tr>
</tbody>
</table>

### Note

In partial scan mode \( (\text{readout window ysize} < 2712 \text{ lines}) \).

To calculate the maximum frames per second \( (\text{FPS}_{\text{max}}) \) you will need following formula \( (\text{Expose mode: No overlap}) \):

\[
\text{FPS}_{\text{max}} = \frac{1}{t_{\text{readout}} + t_{\text{exposure}}}
\]

(Exposure mode: Overlapped):

\[
\begin{align*}
\text{t}_{\text{readout}} < \text{t}_{\text{exposure}}: & \quad \text{FPS}_{\text{max}} = \frac{1}{\text{t}_{\text{exposure}}} \\
\text{t}_{\text{readout}} > \text{t}_{\text{exposure}}: & \quad \text{FPS}_{\text{max}} = \frac{1}{\text{t}_{\text{readout}}}
\end{align*}
\]

#### Example: Frame rate as function of lines & exposure time (Expose mode: No overlap)

Now, when we insert the values using exposure time of, for example, 8000 us, 2712 lines, 4 taps and 66MHz pixel clock:

\[
\text{FPS}_{\text{max}} = \frac{1}{((2712 + 1) \ast 32.98 \text{ us} / (4 / 2)) + (((2728 - 2712) / (4 / 2)) \ast 4.97 \text{ us}) + 50.61 \text{ us})}
\]

\[
= 0.000022307615775410495 \text{ 1 / us}
\]

\[
= 22.3
\]
23.12.4 Reprogramming CCD Timing

Reprogramming the CCD Controller will happen when the following changes occur:

- Changing the exposure time
- Changing the capture window
- Changing trigger modes

Reprogram time consists of two phases:

1. Time needed to send data to the CCD controller depending on what is changed:
   - Exposure: abt 2..3ms
   - Window: abt 4..6ms
   - Trigger mode: from 5..90ms, varies with oldmode/newmode combination

2. Time to initialize (erase) the CCD chip after reprogramming this is fixed, abt 4.5 ms

So for example when reprogramming the capture window you will need (average values):

\[ t_{\text{regprog}} = \text{change\_window} + \text{init\_ccd} \]

\[ t_{\text{regprog}} = 5\text{ms} + 4.5\text{ms} \]

\[ t_{\text{regprog}} = 9.5\text{ms} \]

23.12.5 CCD Sensor Data

Device Structure

- Interline CCD image sensor
- Image size: Diagonal 15.972mm (Type 1)
- Number of effective pixels: 3388 (H) x 2712 (V) approx. 9.19 Mpxels
- Total number of pixels: 3468 (H) x 2728 (V) approx. 9.46 Mpxels
- Chip size: 14.6mm (H) x 12.8mm (V)
- Unit cell size: 3.69um (H) x 3.69um (V)
- Optical black:
  - Horizontal (H) direction: Front 40 pixels each channel
  - Vertical (V) direction: Front 8 pixels each channel
- Number of dummy bits: Front 1 each channel
- Substrate material: Silicon
23.12.5.1 Characteristics

These zone definitions apply to both the color and gray scale version of the sensor.

Zone Definition of Video Signal Shading

![Zone Definition of Video Signal Shading](image)

23.12.5.2 Color version

**Image Sensor Characteristics**

\[
(\text{Ta} = 25^\circ \text{C})
\]

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Unit</th>
<th>Measurement method</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>G Sensitivity</td>
<td>Sg</td>
<td>465</td>
<td>590</td>
<td>1</td>
<td>mV</td>
<td>1/10 s storage</td>
<td></td>
</tr>
<tr>
<td>Sensitivity ratio</td>
<td>R</td>
<td>0.40</td>
<td>0.70</td>
<td>0.52</td>
<td>dB</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>0.22</td>
<td>0.30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Saturation signal</td>
<td>Vsat</td>
<td>520</td>
<td></td>
<td></td>
<td>dB</td>
<td>2</td>
<td>Tj = 60 °C</td>
</tr>
<tr>
<td>Smear</td>
<td>Sm</td>
<td>-105</td>
<td>-95</td>
<td></td>
<td></td>
<td>3</td>
<td>Progressive scan mode</td>
</tr>
<tr>
<td>Video signal shading</td>
<td>SH</td>
<td>20</td>
<td>25</td>
<td></td>
<td>%</td>
<td>4</td>
<td>Zone 0 and zone I</td>
</tr>
<tr>
<td>Dark signal</td>
<td>Vdt</td>
<td>12</td>
<td></td>
<td></td>
<td>mV</td>
<td>5</td>
<td>Tj = 60 °C, 5 frame/s</td>
</tr>
<tr>
<td>Dark signal shading</td>
<td>ΔVdt</td>
<td>6</td>
<td></td>
<td></td>
<td>mV</td>
<td>6</td>
<td>Tj = 60 °C, 5 frame/s, (^1)</td>
</tr>
<tr>
<td>Line crawl R</td>
<td>Lcr</td>
<td>3.8</td>
<td>%</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Line crawl G</td>
<td>Lcg</td>
<td>3.8</td>
<td>%</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Line crawl B</td>
<td>Lcb</td>
<td>3.8</td>
<td>%</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lag</td>
<td>Lag</td>
<td>0.5</td>
<td>%</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The difference between the output signal for other channels (Vout2, Vout3, and Vout4) and the Vout1 output signal is maximum 10%.

\(^1\) Excludes vertical dark signal shading caused by vertical register high-speed transfer.

**Spectral Sensitivity Characteristics** (Excludes lens characteristics and light source characteristics)

![Spectral Sensitivity Characteristics](image)
23.12.5.3 Gray scale version

### Image Sensor Characteristics

\[(Ta = 25°C)\]

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Unit</th>
<th>Measurement method</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity 1</td>
<td>S1</td>
<td>530</td>
<td>660</td>
<td></td>
<td>mV</td>
<td>1</td>
<td>1/30 s storage</td>
</tr>
<tr>
<td>Sensitivity 2</td>
<td>S2</td>
<td>2150</td>
<td></td>
<td></td>
<td>mV</td>
<td>2</td>
<td>1/30 s storage</td>
</tr>
<tr>
<td>Saturation signal</td>
<td>Vsat</td>
<td>520</td>
<td></td>
<td></td>
<td>mV</td>
<td>3</td>
<td>[T_j = 60 °C]</td>
</tr>
<tr>
<td>Smear</td>
<td>Sm</td>
<td>-105</td>
<td>-95</td>
<td></td>
<td>dB</td>
<td>4</td>
<td>Progressive scan mode</td>
</tr>
<tr>
<td>Video signal shading</td>
<td>SH</td>
<td></td>
<td></td>
<td>20</td>
<td>%</td>
<td>5</td>
<td>Zone 0 and zone I</td>
</tr>
<tr>
<td>Dark signal</td>
<td>Vdt</td>
<td></td>
<td></td>
<td>12</td>
<td>mV</td>
<td>6</td>
<td>[T_j = 60 °C, 5 frame/s]</td>
</tr>
<tr>
<td>Dark signal shading</td>
<td>ΔVdt</td>
<td></td>
<td></td>
<td>6</td>
<td>mV</td>
<td>7</td>
<td>[T_j = 60 °C, 5 frame/s, *]</td>
</tr>
<tr>
<td>Lag</td>
<td>Lag</td>
<td></td>
<td></td>
<td>0.5</td>
<td>%</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

The difference between the output signal for other channels (\(V_{out2}, V_{out3}, \) and \(V_{out4}\)) and the \(V_{out1}\) output signal is maximum 10 %.

\*1 Excludes vertical dark signal shading caused by vertical register high-speed transfer.

### Spectral Sensitivity Characteristics (Excludes lens characteristics and light source characteristics)
Measurement conditions

1. In the following measurements, the device drive conditions are at the typical values of the bias and clock voltage conditions.
2. In the following measurements, spot pixels are excluded and, unless otherwise specified, the optical black (OB) level is used as the reference for the signal output, which is measured at point [F8] of the measurement system.

Definition of Standard Imaging Conditions

- **Standard imaging condition I:**
  Use a pattern box (luminance: 706 candelas/m², color temperature of 3200 K halogen source) as a subject. (Pattern for evaluation is not applicable.) Use a testing standard lens with CM500S (t = 1.0 mm) as an IR cut filter and image at F8. The luminous intensity to the sensor receiving surface at this point is defined as the standard sensitivity testing luminous intensity.

- **Standard imaging condition II:**
  Indicates a state that the IR cut filter is removed from the standard imaging condition I.

- **Standard imaging condition III:**
  Image a light source (color temperature of 3200 K) with a uniformity of brightness within 2 % at all angles. Use a testing standard lens with CM500S (t = 1.0 mm) as an IR cut filter. The luminous intensity is adjusted to the value indicated in each testing item by the lens diaphragm.

1. **Sensitivity 1**
   Set the measurement condition to standard imaging condition I. After setting the electronic shutter mode with a shutter speed of 1/100 s, measure the signal output (Vso1) at the center of the screen, and substitute the value into the following formula.
   \[ S1 = V_{so1} \times (100/30) \text{ [mV]} \]

2. **Sensitivity 2**
   Set the measurement condition to standard imaging condition II. After setting the electronic shutter mode with a shutter speed of 1/500 s, measure the signal output (Vso2) at the center of the screen, and substitute the value into the following formula.
   \[ S2 = V_{so2} \times (500/30) \text{ [mV]} \]

3. **Saturation signal**
   Set the measurement condition to standard imaging condition III. After adjusting the luminous intensity to 10 times the intensity with the average value of the signal output, 200 mV, measure the minimum value of the signal output.

4. **Smear**
   Set the measurement condition to standard imaging condition III. With the lens diaphragm at F5.6 to F8, adjust the luminous intensity to 500 times the intensity with the average value of the signal output, 200 mV. When the readout clock is stopped and the charge drain is executed by the electronic shutter at the respective H blankings, measure the maximum value (VSm) of the signal output, and substitute the value into the following formula.
   \[ Sm = 20 \times \log ((V_{Sm}/200) \times (1/500) \times (1/10)) \text{ [dB]} \times (1/10 \text{ V method conversion value}) \]
5. Video signal shading
Set the measurement condition to standard imaging condition III. With the lens diaphragm at F5.6 to F8, adjust the luminous intensity so that the average value of the signal output is 200 mV. Then measure the maximum value (Vmax) and minimum value (Vmin) of the output signal and substitute the values into the following formula.

\[ SH = \frac{(V_{\text{max}} - V_{\text{min}})}{200 \text{ mV}} \times 100\% \]

6. Dark signal
Measure the average value of the signal output (Vid) based on the horizontal idle transfer level at the device ambient temperature of 60 °C placing the device in the light-obstructed state.

7. Dark signal shading
After the measurement item 6, measure the maximum value (Vdmax) and minimum value (Vdmin) of the dark signal output, and substitute the values into the following formula.

\[ \Delta V_{\text{d}} = V_{\text{dmax}} - V_{\text{dmin}} \]

8. Lag
Adjust the signal output value generated by strobe light to 200 mV. After setting the strobe light so that it strobes with the following timing, measure the residual signal level (Vlag), and substitute the value into the following formula.

\[ \text{Lag} = \left( \frac{V_{\text{lag}}}{200} \right) \times 100\% \]

![Diagram of measurement system](image_url)

Measurement System

23.12.6 Device Feature And Property List

23.12.6.1 Gray scale version

23.12.6.2 Color version
23.13.1 Introduction

The CCD sensor is a highly programmable imaging module which will, for example, enable the following type of applications

**Industrial applications:**

- triggered image acquisition with precise control of image exposure start by hardware trigger input.
- image acquisition of fast moving objects due to:
  - frame exposure, integrating all pixels at a time in contrast to CMOS imager which typically integrate line-by-line.
  - short shutter time, to get sharp images.
  - flash control output to have enough light for short time.

**Scientific applications:**

- long time exposure for low light conditions.
- optimizing image quality using the variable shutter control.

23.13.2 Details of operation

The process of getting an image from the CCD sensor can be separated into three different phases.

23.13.2.1 Trigger

When coming out of reset or ready with the last readout the CCD controller is waiting for a Trigger signal.

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Setting (GenICam (p.143))</th>
<th>Mode / Setting (Device Specific (p.143))</th>
<th>Description</th>
</tr>
</thead>
</table>
| "TriggerSelector = FrameStart"
"TriggerMode = Off"       | Continuous                              | Free running, no external trigger signal needed. |
| "TriggerSelector = FrameStart"
"TriggerMode = On"
"TriggerSource = Software"
"ExposureMode = Timed"    | OnDemand                                 | Image acquisition triggered by command (software trigger). |

To trigger one frame execute the `TriggerSoftware@i` command then.
<table>
<thead>
<tr>
<th>TriggerSelector = AcquisitionActive</th>
<th>OnLowLevel</th>
<th>Start an exposure of a frame as long as the trigger input is below the trigger threshold. (No Frame←Trigger!)</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerMode = On&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSource = &lt;desired Line&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerActivation = LevelLow&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;ExposureMode = Timed&quot;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TriggerSelector = AcquisitionActive</th>
<th>OnHighLevel</th>
<th>Start an exposure of a frame as long as the trigger input is above the trigger threshold. (No Frame←Trigger!)</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerMode = On&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSource = &lt;desired Line&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerActivation = LevelHigh&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;ExposureMode = Timed&quot;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TriggerSelector = FrameStart</th>
<th>OnFallingEdge</th>
<th>Each falling edge of trigger signal acquires one image.</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerMode = On&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSource = &lt;desired Line&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerActivation = FallingEdge&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;ExposureMode = Timed&quot;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TriggerSelector = FrameStart</th>
<th>OnRisingEdge</th>
<th>Each rising edge of trigger signal acquires one image.</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerMode = On&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSource = &lt;desired Line&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerActivation = RisingEdge&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;ExposureMode = Timed&quot;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TriggerSelector = FrameStart</th>
<th>OnHighExpose</th>
<th>Each rising edge of trigger signal acquires one image, exposure time corresponds to pulse width.</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerMode = On&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSource = &lt;desired Line&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerActivation = LevelHigh&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;ExposureMode = TriggerWidth&quot;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TriggerSelector = FrameStart</th>
<th>OnLowExpose</th>
<th>Each falling edge of trigger signal acquires one image, exposure time corresponds to pulse width.</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerMode = On&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSource = &lt;desired Line&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerActivation = LevelLow&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;ExposureMode = TriggerWidth&quot;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
23.13.2.2 Exposure aka Integration

After an active trigger, the exposure phase starts with a maximum jitter of $t_{\text{trig}}$. If flash illumination is enabled in software the flash output will be activated exactly while the sensor chip is integrating light. Exposure time is adjustable by software in increments of $t_{\text{readline}}$.

23.13.2.3 Readout

When exposure is finished, the image is transferred to hidden storage cells on the CCD. Image data is then shifted out line-by-line and transferred to memory. Shifting out non active lines takes $t_{\text{vshift}}$, while shifting out active lines will consume $t_{\text{readline}}$. The number of active pixels per line will not have any impact on readout speed.

23.13.3 CCD Timing

In the "Trigger Overlap ReadOut" mode, the camera accepts a trigger signal the $t_{\text{exposure}}$ period earlier than $t_{\text{readout}}$ is finished.
23.13.3.1 Timings
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Pixel clock</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_{\text{trig}}$</td>
<td>Time from trigger (internal or external) to exposure start</td>
<td>56 MHz</td>
</tr>
<tr>
<td>$t_{\text{trig}}$</td>
<td>$t_{\text{trig}} = t_{\text{vshift}} + t_{\text{turn_on_delay_opto}}$ (approx. 2us)</td>
<td></td>
</tr>
<tr>
<td>$t_{\text{wait}}$</td>
<td>Time after a new trigger is accepted</td>
<td>&lt; 1us</td>
</tr>
<tr>
<td>$t_{\text{trans}}$</td>
<td>Image transfer time (move image to readout cells in CCD)</td>
<td>59.64us</td>
</tr>
<tr>
<td>$t_{\text{readline}}$</td>
<td>Time needed to readout a line</td>
<td>38.88us</td>
</tr>
<tr>
<td>$t_{\text{vshift}}$</td>
<td>Time needed to shift unused lines away</td>
<td>5.86us</td>
</tr>
<tr>
<td>$t_{\text{exposure}}$</td>
<td>Exposure time</td>
<td>30us..20s</td>
</tr>
<tr>
<td>$t_{\text{readout}}$</td>
<td>Image readout time (move image from readout cells to memory)</td>
<td>$t_{\text{readout}} = ((\text{ActiveLines} + 1) * t_{\text{readline}} / (\text{taps} / 2)) + (((2728 - \text{ActiveLines}) / (\text{taps} / 2)) * t_{\text{vshift}}) + t_{\text{trans}}$</td>
</tr>
</tbody>
</table>

**Note**

In partial scan mode (readout window ysize < 2712 lines).

To calculate the maximum frames per second ($\text{FPS}_{\text{max}}$) you will need following formula (Expose mode: No overlap):

$$\text{FPS}_{\text{max}} = \frac{1}{t_{\text{readout}} + t_{\text{exposure}}}$$

(Expose mode: Overlapped):

- $t_{\text{readout}} < t_{\text{exposure}}$: 
  $$\text{FPS}_{\text{max}} = \frac{1}{t_{\text{exposure}}}$$
- $t_{\text{readout}} > t_{\text{exposure}}$: 
  $$\text{FPS}_{\text{max}} = \frac{1}{t_{\text{readout}}}$$

### 23.13.3.1.1 Example: Frame rate as function of lines & exposure time (Expose mode: No overlap)

Now, when we insert the values using exposure time of, for example, 8000 us, 2712 lines, 2 taps and 56MHz pixel clock:

$$\text{FPS}_{\text{max}} = \frac{1}{((2712 + 1) * 38.88 \text{ us} / (2 / 2)) + (((2728 - 2712) / (2 / 2)) * 5.86 \text{ us}) + 59.64 \text{ us}}$$

$$= 0.000009466573717 \text{ 1 / us}$$

$$= 9.5$$
23.13.4 Reprogramming CCD Timing

Reprogramming the CCD Controller will happen when the following changes occur

- Changing the exposure time
- Changing the capture window
- Changing trigger modes

Reprogram time consists of two phases

1. Time needed to send data to the CCD controller depending on what is changed
   - exposure: about 2.3ms
   - window: about 4.6ms
   - trigger mode: from 5.90ms, varies with oldmode/newmode combination
2. Time to initialize (erase) the CCD chip after reprogramming this is fixed, about 4.5ms

So for example when reprogramming the capture window you will need (average values)

\[ t_{\text{regprog}} = \text{change\_window} + \text{init\_ccd} \]

\[ t_{\text{regprog}} = 5\text{ms} + 4.5\text{ms} \]

\[ t_{\text{regprog}} = 9.5\text{ms} \]

23.13.5 CCD Sensor Data

Device Structure

- Interline CCD image sensor
- Image size: Diagonal 15.972mm (Type 1)
- Number of effective pixels: 3388 (H) x 2712 (V) approx. 9.19 Mpixels
- Total number of pixels: 3468 (H) x 2728 (V) approx. 9.46 Mpixels
- Chip size: 14.6mm (H) x 12.8mm (V)
- Unit cell size: 3.69um (H) x 3.69um (V)
- Optical black:
  - Horizontal (H) direction: Front 40 pixels each channel
  - Vertical (V) direction: Front 8 pixels each channel
- Number of dummy bits: Front 1 each channel
- Substrate material: Silicon
23.13.5.1 Characteristics

These zone definitions apply to both the color and gray scale version of the sensor.

![Zone Definition of Video Signal Shading](image)

23.13.5.2 Color version

**Image Sensor Characteristics**

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Unit</th>
<th>Measurement method</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>G Sensitivity</td>
<td>Sg</td>
<td>465</td>
<td>590</td>
<td></td>
<td>mV</td>
<td>1</td>
<td>1/10 s storage</td>
</tr>
<tr>
<td>Sensitivity ratio</td>
<td>R</td>
<td>0.40</td>
<td></td>
<td>0.70</td>
<td>mV</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>0.22</td>
<td></td>
<td>0.52</td>
<td>mV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Saturation signal</td>
<td>Vsat</td>
<td>520</td>
<td></td>
<td></td>
<td>mV</td>
<td>2</td>
<td>Tj = 60 °C</td>
</tr>
<tr>
<td>Smear</td>
<td>Sm</td>
<td>-105</td>
<td>-95</td>
<td></td>
<td>dB</td>
<td>3</td>
<td>Progressive scan mode</td>
</tr>
<tr>
<td>Video signal shading</td>
<td>SH</td>
<td></td>
<td></td>
<td></td>
<td>%</td>
<td>4</td>
<td>Zone 0 and zone I</td>
</tr>
<tr>
<td>Dark signal</td>
<td>Vdt</td>
<td>12</td>
<td></td>
<td></td>
<td>mV</td>
<td>5</td>
<td>Tj = 60 °C, 5 frames/s</td>
</tr>
<tr>
<td>Dark signal shading</td>
<td>ΔVdt</td>
<td>6</td>
<td></td>
<td></td>
<td>mV</td>
<td>6</td>
<td>Tj = 60 °C, 5 frames/s</td>
</tr>
<tr>
<td>Line crawl R</td>
<td>Lcr</td>
<td>3.8%</td>
<td></td>
<td></td>
<td>%</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Line crawl G</td>
<td>Lcg</td>
<td>3.8%</td>
<td></td>
<td></td>
<td>%</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Line crawl B</td>
<td>Lcb</td>
<td>3.8%</td>
<td></td>
<td></td>
<td>%</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Lag</td>
<td>Lag</td>
<td>0.5%</td>
<td></td>
<td></td>
<td>%</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

The maximum difference of the output signal right and left output (\(V_{O1} - V_{O2}\)) is 10 %

1. Note that the video signal shading may increase when using this sensor combined with a lens of which pupil distance sheet.
2. Excludes vertical dark signal shading caused by vertical resolution high-speed character.

**Spectral Sensitivity Characteristics** (Excludes lens characteristics and light source characteristics)
### Image Sensor Characteristics

$$(Ta = 25°C)$$

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Unit</th>
<th>Measurement method</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity 1</td>
<td>S1</td>
<td>530</td>
<td>660</td>
<td></td>
<td>mV</td>
<td>1/30 s storage</td>
<td></td>
</tr>
<tr>
<td>Sensitivity 2</td>
<td>S2</td>
<td>2150</td>
<td></td>
<td></td>
<td>mV</td>
<td>1/30 s storage</td>
<td></td>
</tr>
<tr>
<td>Saturation signal</td>
<td>Vsat</td>
<td>520</td>
<td></td>
<td></td>
<td>mV</td>
<td>3</td>
<td>$T_j = 60 , ^{°}C$</td>
</tr>
<tr>
<td>Smear</td>
<td>Sm</td>
<td></td>
<td>$-105$</td>
<td>$-95$</td>
<td>dB</td>
<td>4</td>
<td>Progressive scan mode</td>
</tr>
<tr>
<td>Video signal shading</td>
<td>SH</td>
<td></td>
<td>20</td>
<td></td>
<td>%</td>
<td>5</td>
<td>Zone 0 and zone I **1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>25</td>
<td></td>
<td>%</td>
<td></td>
<td>Zone 0, zone I, zone II **1 and zone II **1</td>
</tr>
<tr>
<td>Dark signal</td>
<td>Vdt</td>
<td>12</td>
<td></td>
<td></td>
<td>mV</td>
<td>6</td>
<td>$T_j = 60 , ^{°}C$, 5 frame/s</td>
</tr>
<tr>
<td>Dark signal shading</td>
<td>ΔVdt</td>
<td>6</td>
<td></td>
<td></td>
<td>mV</td>
<td>7</td>
<td>$T_j = 60 , ^{°}C$, 5 frame/s, **2</td>
</tr>
<tr>
<td>Lag</td>
<td>Lag</td>
<td>0.5</td>
<td></td>
<td></td>
<td>%</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

1. The maximum difference of the output signal right and left output ($V_{0,101}$ $V_{0,102}$) is 10 %

2. Note that the video signal shading may increase when using this sensor combined with a lens of which exit pupil distance short

**2** Excludes vertical dark signal shading caused by vertical register high-speed transfer.
Measurement conditions

1. In the following measurements, the device drive conditions are at the typical values of the bias and clock voltage conditions.

2. In the following measurements, spot pixels are excluded and, unless otherwise specified, the optical black (OB) level is used as the reference for the signal output, which is taken as the value of the Gr/Gb signal output or the R/B signal output of the measurement system.

Definition of Standard Imaging Conditions

1. Standard imaging condition I:
   Use a pattern box (luminance: 706 cd/m², color temperature of 3200 K halogen source) as a subject. (Pattern for evaluation is not applicable.) Use a testing standard lens with CM5G0S (f = 1.0 mm) as an IR cut filter and image at F5.6. The luminous intensity to the sensor receiving surface at this point is defined as the standard sensitivity testing luminous intensity.

2. Standard imaging condition II:
   Image a light source (color temperature of 3200 K) with uniformity of brightness within 2% at all angles. Use a testing standard lens with CM5G0S (f = 1.6 mm) as an IR cut filter. The luminous intensity is adjusted to the value indicated in each testing item by the lens diaphragm.

1. Gain sensitivity, sensitivity ratio
   Set the measurement condition to the standard imaging condition I. After setting the electronic shutter mode with a shutter speed of 1/100 s, measure the signal output (Vc, Vd, Vc, Va, Vb) at the center of each Gr, Gb, R and B channel screen, and substitute the values into the following formula.

   \[
   V_{cl} = \frac{(V_c + V_{dc})}{2} \\
   S_g = V_a \times \left(\frac{100}{30}\right) \text{[mV]} \\
   R_v = V_a V_a \\
   R_b = V_a V_a
   \]

2. Saturation signal
   Set the measurement condition to standard imaging condition II. After adjusting the luminous intensity to 20 times the intensity with the average value of the Gr signal output, 200 mV, measure the minimum value of the Gr, Gb, R and B signals.

3. Smax
   Set the measurement condition to standard imaging condition II. With the lens diaphragm at F5.6 to F8, first adjust the average value of the Gr signal output to 200 mV. Measure the average values of the Gr, Gb, R and B signal outputs (Gr_a, Gb_a, Ra, Ba), and then adjust the luminous intensity to 500 times the intensity with the average value of the Gr signal output, 200 mV. After the readout clock is stopped and the charge drain is executed by the electronic shutter at the respective H blankings, measure the maximum value (VSm [mV]) independent of the Gr, Gb, R and B signal outputs, and substitute the value into the following formula.

   \[
   S_{sm} = 20 \times \log \frac{V_{Sm}}{(V_{Sna} (Gra + Gba + Ra + Ba)/(4))} \times \left(\frac{15000}{1/10}\right) \text{[dB]} \times \left(1/10 \text{V method conversion value}\right)
   \]

4. Video signal shading
   Set the measurement condition to standard imaging condition II. With the lens diaphragm at F5.6 to F8, adjust the luminous intensity so that the average value of the signal output is 200 mV. Then measure the maximum value (Gmax) and the minimum value (Gmin) of the Gr signal output, and substitute the values into the following formula.

   \[
   SH_g = \frac{(G_{max} - G_{min})}{200 \text{ mV} \times 100} \%
   \]
5. Dark signal
Measure the average value of the signal output \( (V_{di}) \) based on the horizontal idle transfer level at the junction temperature of 60 °C placing the device in the light-obstructed state.

6. Dark signal shading
After the measurement item 5, measure the maximum value \( (V_{dmax}) \) and minimum value \( (V_{dmin}) \) of the dark signal output, and substitute the values into the following formula.

\[
\Delta V_{di} = V_{dmax} - V_{dmin}
\]

7. Line crawl
Set the measurement condition to the standard imaging condition II. After adjusting the average value of the Gr signal output to 200 mV, inter R, G and B filters and measure the difference between Gr signal lines \( (\Delta G_r, \Delta G_g, \Delta G_b \text{mV}) \) as well as the average value of the G signal output \( (G_{ar}, G_{ag}, G_{ab}) \). Substitute the values into the following formula:

\[
L_i = \frac{\Delta G_i}{G_{ai}} \times 100 \text{[%]} \quad (i = r, g, b)
\]

8. Lag
Adjust the signal output value generated by strobe light to 200 mV. After setting the strobe light so that it strobes with the following timing, measure the residual signal level \( (V_{lag}) \), and substitute the value into the following formula.

\[
L_{ag} = \left( \frac{V_{lag}}{200} \right) \times 100 \text{[%]}
\]

---

23.13.6 Device Feature And Property List

23.13.6.1 Gray scale version

23.13.6.2 Color version
23.14 mvBlueCOUGAR-XD1212a (12.1 Mpix [4248 x 2836])

23.14.1 Introduction

The CCD sensor is a highly programmable imaging module which will, for example, enable the following type of applications

**Industrial applications:**

- triggered image acquisition with precise control of image exposure start by hardware trigger input.
- image acquisition of fast moving objects due to:
  - frame exposure, integrating all pixels at a time in contrast to CMOS imager which typically integrate line-by-line.
  - short shutter time, to get sharp images.
  - flash control output to have enough light for short time.

**Scientific applications:**

- long time exposure for low light conditions.
- optimizing image quality using the variable shutter control.

23.14.2 Details of operation

The process of getting an image from the CCD sensor can be separated into three different phases.

23.14.2.1 Trigger

When coming out of reset or ready with the last readout the CCD controller is waiting for a Trigger signal.

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Setting (GenICam) (p. 143)</th>
<th>Mode / Setting (Device Specific) (p. 143)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = Off&quot;</td>
<td>Continuous</td>
<td>Free running, no external trigger signal needed.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = Software&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnDemand</td>
<td>Image acquisition triggered by command (software trigger).</td>
</tr>
</tbody>
</table>

To trigger one frame execute the `TriggerSoftware@i` command then.
<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelLow&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnLowLevel</td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelHigh&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnHighLevel</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = FallingEdge&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnFallingEdge</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = RisingEdge&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnRisingEdge</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelHigh&quot; &quot;ExposureMode = TriggerWidth&quot;</td>
<td>OnHighExpose</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelLow&quot; &quot;ExposureMode = TriggerWidth&quot;</td>
<td>OnLowExpose</td>
</tr>
</tbody>
</table>
"TriggerSelector = FrameStart"
"TriggerMode = On"
"TriggerSource = <desired Line>"
"TriggerActivation = AnyEdge"
"ExposureMode = Timed"

<table>
<thead>
<tr>
<th>OnAnyEdge</th>
<th>Start the exposure of a frame when the trigger input level changes from high to low or from low to high.</th>
</tr>
</thead>
</table>

Line Mapping (TriggerSource mvIMPACT Acquire -> TriggerSource GenICam (valid values for <desired Line>)):

<table>
<thead>
<tr>
<th>TriggerSource mvIMPACT Acquire</th>
<th>TriggerSource GenICam (BCX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP-IN0</td>
<td>Line4</td>
</tr>
<tr>
<td>GP-IN1</td>
<td>Line5</td>
</tr>
</tbody>
</table>

23.14.2.2 Exposure aka Integration

After an active trigger, the exposure phase starts with a maximum jitter of \( t_{\text{trig}} \). If flash illumination is enabled in software the flash output will be activated exactly while the sensor chip is integrating light. Exposure time is adjustable by software in increments of \( t_{\text{readline}} \).

23.14.2.3 Readout

When exposure is finished, the image is transferred to hidden storage cells on the CCD. Image data is then shifted out line-by-line and transferred to memory. Shifting out non active lines takes \( t_{\text{vshift}} \), while shifting out active lines will consume \( t_{\text{readline}} \). The number of active pixels per line will not have any impact on readout speed.

23.14.3 CCD Timing

**Note**

In the "Trigger Overlap ReadOut" mode, the camera accepts a trigger signal the exposure period earlier than \( t_{\text{readout}} \) is finished.
23.14.3.1 Timings
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Pixel clock</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>t&lt;sub&gt;trig&lt;/sub&gt;</td>
<td>Time from trigger (internal or external) to exposure start</td>
<td>36 MHz</td>
<td>56 MHz</td>
</tr>
<tr>
<td>t&lt;sub&gt;wait&lt;/sub&gt;</td>
<td>Time after a new trigger is accepted</td>
<td>&lt; 1us</td>
<td></td>
</tr>
<tr>
<td>t&lt;sub&gt;trans&lt;/sub&gt;</td>
<td>Image transfer time (move image to readout cells in CCD)</td>
<td>92.78us</td>
<td>59.64us</td>
</tr>
<tr>
<td>t&lt;sub&gt;readline&lt;/sub&gt;</td>
<td>time needed to readout a line</td>
<td>71.56us</td>
<td>46.00us</td>
</tr>
<tr>
<td>t&lt;sub&gt;vshift&lt;/sub&gt;</td>
<td>time needed to shift unused lines away</td>
<td>8.22us</td>
<td>5.29us</td>
</tr>
<tr>
<td>t&lt;sub&gt;exposure&lt;/sub&gt;</td>
<td>Exposure time</td>
<td>30us..20s</td>
<td>30us..20s</td>
</tr>
<tr>
<td>t&lt;sub&gt;readout&lt;/sub&gt;</td>
<td>Image readout time (move image from readout cells to memory)</td>
<td>t&lt;sub&gt;readout&lt;/sub&gt; = (ActiveLines + 1) * t&lt;sub&gt;readline&lt;/sub&gt; / (taps / 2)) + ((2854 - ActiveLines) / (taps / 2)) * t&lt;sub&gt;vshift&lt;/sub&gt; + t&lt;sub&gt;trans&lt;/sub&gt;</td>
<td></td>
</tr>
</tbody>
</table>

**Note**

In partial scan mode (readout window ysize < 2838 lines).

To calculate the maximum frames per second (FPS<sub>max</sub>) you will need following formula (Expose mode: No overlap):

\[
FPS_{\text{max}} = \frac{1}{t_{\text{readout}} + t_{\text{exposure}}}
\]

(Expose mode: Overlapped):

\[
\text{t_readout} < \text{t_exposure}: \quad FPS_{\text{max}} = \frac{1}{t_{\text{exposure}}}
\]

\[
\text{t_readout} > \text{t_exposure}: \quad FPS_{\text{max}} = \frac{1}{t_{\text{readout}}}
\]

23.14.3.1.1 Example: Frame rate as function of lines & exposure time (Expose mode: No overlap)

Now, when we insert the values using exposure time of, for example, 8000 us, 2838 lines, 2 taps and 56MHz pixel clock:

\[
FPS_{\text{max}} = \frac{1}{(12838 + 1) \times 46.00 \text{ us} / (2 / 2) + (2854 - 2838) / (2 / 2) + 5.29 \text{ us} + 59.64 \text{ us}}
\]

\[
= 0.0000765156136231 \text{ 1 / us}
\]

\[
= 7.7
\]
23.14.4 Reprogramming CCD Timing

Reprogramming the CCD Controller will happen when the following changes occur:

- Changing the exposure time
- Changing the capture window
- Changing trigger modes

Reprogram time consists of two phases:

1. Time needed to send data to the CCD controller depending on what is changed:
   - Exposure: about 2..3 ms
   - Window: about 4..6 ms
   - Trigger mode: from 5..90 ms, varies with oldmode/newmode combination

2. Time to initialize (erase) the CCD chip after reprogramming: this is fixed, about 4.5 ms

So for example, when reprogramming the capture window, you will need (average values):

\[ t_{\text{regprog}} = \text{change\_window} + \text{init\_ccd} \]

\[ t_{\text{regprog}} = 5 \text{ ms} + 4.5 \text{ ms} \]

\[ t_{\text{regprog}} = 9.5 \text{ ms} \]

23.14.5 CCD Sensor Data

Device Structure:

- Interline CCD image sensor
- Image size: Diagonal 15.808 mm (Type 1)
- Number of effective pixels: 4248 (H) \times 2836 (V) approx. 12.06 Mpixels
- Total number of pixels: 4330 (H) \times 2854 (V) approx. 12.36 Mpixels
- Chip size: 15.3 mm (H) \times 12.0 mm (V)
- Unit cell size: 3.1 \text{ um} (H) \times 3.1 \text{ um} (V)
- Optical black:
  - Horizontal (H) direction: Front 40 pixels each channel
  - Vertical (V) direction: Front 8 pixels each channel
- Number of dummy bits: Front 1 each channel
- Substrate material: Silicon
23.14.5.1 Characteristics

These zone definitions apply to both the color and gray scale version of the sensor.

**Zone Definition of Video Signal Shading**

23.14.5.2 Color version

**Image Sensor Characteristics**

(Ta = 25°C)

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Unit</th>
<th>Measurement method</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>G Sensitivity</td>
<td>Sg</td>
<td>420</td>
<td></td>
<td></td>
<td>mV</td>
<td>1</td>
<td>1/30 s storage</td>
</tr>
<tr>
<td>Sensitivity ratio</td>
<td>R Rr</td>
<td>0.40</td>
<td>0.70</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B Rb</td>
<td>0.22</td>
<td>0.52</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Saturation signal</td>
<td>Vsat</td>
<td>360</td>
<td></td>
<td></td>
<td>mV</td>
<td>2</td>
<td>Tj = 60 °C</td>
</tr>
<tr>
<td>Smear</td>
<td>Sm</td>
<td>−105</td>
<td>−95</td>
<td></td>
<td>dB</td>
<td>3</td>
<td>Progressive scan mode</td>
</tr>
<tr>
<td>Video signal shading</td>
<td>SH</td>
<td>20</td>
<td></td>
<td></td>
<td>%</td>
<td>4</td>
<td>Zone 0 and zone I</td>
</tr>
<tr>
<td></td>
<td></td>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Zone 0, zone I, zone II, and zone III</td>
</tr>
<tr>
<td>Dark signal</td>
<td>Vdt</td>
<td>15.2</td>
<td></td>
<td></td>
<td>mV</td>
<td>5</td>
<td>Tj = 60 °C, 3.9 frames'</td>
</tr>
<tr>
<td>Dark signal shading</td>
<td>ΔVdt</td>
<td>7.6</td>
<td></td>
<td></td>
<td>mV</td>
<td>6</td>
<td>Tj = 60 °C, 3.9 frames'</td>
</tr>
<tr>
<td>Line crawl R</td>
<td>Lcr</td>
<td>3.8</td>
<td></td>
<td></td>
<td>%</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Line crawl G</td>
<td>Lcg</td>
<td>3.8</td>
<td></td>
<td></td>
<td>%</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Line crawl B</td>
<td>Lcb</td>
<td>3.8</td>
<td></td>
<td></td>
<td>%</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Lag</td>
<td>Lag</td>
<td>0.5</td>
<td></td>
<td></td>
<td>%</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

The difference between the outputs signals for other channels (Vout2, Vout3, and Vout4) and the Vout1 output signal is maximum 10%.

*1 Note that the video signal shading may increase when using this sensor combined with a lens with which exit pupil distances short.
*2 Excludes vertical dark signal shading caused by vertical register high-speed transfer.
Spectral Sensitivity Characteristics (Includes lens characteristics, excludes light source characteristics)

23.14.5.3 Gray scale version

**Image Sensor Characteristics**

\( Ta = 25^\circ C \)

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Unit</th>
<th>Measurement method</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity 1</td>
<td>S1</td>
<td>-420</td>
<td></td>
<td></td>
<td>mV</td>
<td>1</td>
<td>1/30 s storage</td>
</tr>
<tr>
<td>Sensitivity 2</td>
<td>S2</td>
<td>1370</td>
<td></td>
<td></td>
<td>mV</td>
<td>2</td>
<td>1/30 s storage</td>
</tr>
<tr>
<td>Saturation signal</td>
<td>Vsat</td>
<td>350</td>
<td></td>
<td></td>
<td>mV</td>
<td>3</td>
<td>( T_j = 60^\circ C )</td>
</tr>
<tr>
<td>Smear</td>
<td>Sm</td>
<td>-105</td>
<td>-95</td>
<td>dB</td>
<td></td>
<td>4</td>
<td>Progressive scan mode</td>
</tr>
<tr>
<td>Video signal shading</td>
<td>SH</td>
<td>20</td>
<td></td>
<td></td>
<td>%</td>
<td>5</td>
<td>Zone 0 and zone I</td>
</tr>
<tr>
<td>Dark signal</td>
<td>Vdt</td>
<td>15.2</td>
<td></td>
<td>mV</td>
<td>6</td>
<td>( T_j = 60^\circ C, 3.9 ) frame/s</td>
<td></td>
</tr>
<tr>
<td>Dark signal shading</td>
<td>( \Delta Vdt )</td>
<td>7.6</td>
<td></td>
<td>mV</td>
<td>7</td>
<td>( T_j = 60^\circ C, 3.9 ) frame/s</td>
<td></td>
</tr>
<tr>
<td>Lag</td>
<td>Lag</td>
<td>0.5</td>
<td></td>
<td>%</td>
<td>8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The difference between the output signal for other channels (\( Y_{Vdt}, Y_{Vsat} \) and \( Y_{Vdt} \)) and the \( Y_{Vdt} \) output signal is maximum 10%.

*1 Note that the video signal shading may increase when using this sensor combined with a lens of which exit pupil distances short.

*2 Excludes vertical dark signal shading caused by vertical register high-speed transfer.

Spectral Sensitivity Characteristics (Excludes lens characteristics and light source characteristics)
Measurement conditions

1. In the following measurements, the device drive conditions are at the typical values of the bias and clock voltage conditions.

2. In the following measurements, spot pixels are excluded and, unless otherwise specified, the optical black (OB) level is used as the reference for the signal output, which is measured at point [B] of the measurement system.

Definition of Standard Imaging Conditions

+ Standard imaging condition I:
  Use a pattern box (luminance: 706 cd/m², color temperature of 3200 K halogen source) as a subject.
  (Pattern for evaluation is not applicable.) Use a testing standard lens with CM500S (t = 1.0 mm) as an IR cut filter and image at F8. The luminous intensity to the sensor receiving surface at this point is defined as the standard sensitivity testing luminous intensity.

+ Standard imaging condition II:
  Indicates a state that the IR cut filter is removed from the standard imaging condition I.

+ Standard imaging condition III:
  Image a light source (color temperature of 3200 K) with a uniformity of brightness within 2 % at all angles.
  Use a testing standard lens with CM500S (t = 1.0 mm) as an IR cut filter. The luminous intensity is adjusted to the value indicated in each testing item by the lens diaphragm.

1. Sensitivity 1
   Set the measurement condition to standard imaging condition I. After setting the electronic shutter mode with a shutter speed of 1/100 s, measure the signal output (Vs1) at the center of the screen, and substitute the value into the following formula.

   \[ S1 = Vs1 \times \left( \frac{100}{30} \right) [mV] \]

2. Sensitivity 2
   Set the measurement condition to standard imaging condition II. After setting the electronic shutter mode with a shutter speed of 1/500 s, measure the signal output (Vs2) at the center of the screen, and substitute the value into the following formula.

   \[ S2 = Vs2 \times \left( \frac{500}{30} \right) [mV] \]

3. Saturation signal
   Set the measurement condition to standard imaging condition III. After adjusting the luminous intensity to 10 times the intensity with the average value of the signal output, 200 mV, measure the minimum value of the signal output.

4. Smear
   Set the measurement condition to standard imaging condition III. With the lens diaphragm at F5.6 to F8, adjust the luminous intensity to 500 times the intensity with the average value of the signal output, 200 mV. When the readout clock is stopped and the charge drain is executed by the electronic shutter at the respective H blankings, measure the maximum value (VsM) of the signal output, and substitute the value into the following formula.

   \[ Sm = 20 \times \log \left( \frac{VsM}{200} \times \left( \frac{1}{500} \times \frac{110}{10} \right) \right) [dB] \left( \frac{1}{10} \text{ V method conversion value} \right) \]
5. **Video signal shading**

   Set the measurement condition to standard imaging condition III. With the lens diaphragm at F5.6 to F8, adjust the luminous intensity so that the average value of the signal output is 200 mV. Then measure the maximum value (Vmax) and minimum value (Vmin) of the output signal and substitute the values into the following formula.

   \[
   SH = \frac{(V_{\text{max}} - V_{\text{min}})}{200 \text{ mV}} \times 100 \% 
   \]

6. **Dark signal**

   Measure the average value of the signal output (Vdt) based on the horizontal idle transfer level at the device ambient temperature of 60 °C placing the device in the light-obstructed state.

7. **Dark signal shading**

   After the measurement item 6, measure the maximum value (Vdmax) and minimum value (Vdmin) of the dark signal output, and substitute the values into the following formula.

   \[
   \Delta V_{\text{dt}} = V_{\text{dmax}} - V_{\text{dmin}}
   \]

8. **Lag**

   Adjust the signal output value generated by strobe light to 200 mV. After setting the strobe light so that it strobes with the following timing, measure the residual signal level (Vlag), and substitute the value into the following formula.

   \[
   \text{Lag} = \frac{(V_{\text{lag}}/200)}{\times 100 \%}
   \]

---

**Measurement System**

![Diagram of measurement system](image)

---

23.14.6 **Device Feature And Property List**

23.14.6.1 **Gray scale version**

23.14.6.2 **Color version**
Appendix A.2 Pregius CMOS specific camera / sensor data

24.1 mvBlueCOUGAR-X100f (0.4 Mpix [728 x 544])

24.1.1 Introduction

The CMOS sensor module (IMX287) incorporates the following features:

- resolution to 728 x 544 gray scale or RGB Bayer mosaic
- supports window AOI mode with faster readout
- global shutter
- programmable readout timing with free capture windows and partial scan
- many trigger modes (free-running, hardware-triggered)

24.1.2 Details of operation

The sensor uses a global shutter, i.e. light exposure takes place on all pixels in parallel, although subsequent readout is sequential.

24.1.2.1 Free running mode

In free running mode, the sensor reaches its maximum frame rate. This is done by overlapping erase, exposure and readout phase. The sensor timing in free running mode is fixed, so there is no control when to start an acquisition. This mode is used with trigger mode Continuous.

To calculate the maximum frames per second (FPS_{max}) in free running mode you will need following formula:

\[
\text{FrameTime} = \frac{\text{InternalLineLength} \times 8 \text{ ImageHeight} + \text{VerticalBlankLines}}{\text{SensorInClock} \times \text{NumberOfLVDS} \times 1000}
\]

If exposure time is lower than frame time:
\[
\text{FPS}_{\text{max}} = \frac{1}{\text{FrameTime}}
\]

If exposure time is greater than frame time:

\[
\text{FPS}_{\text{max}} = \frac{1}{\text{ExposureTime}}
\]

**Note**

The exposure time step width is limited to the sensor's row time of 3.9 us and therefore
- auto exposure with very low exposure times will perform with relatively large increments and
- exposure mode = TriggerWidth (if available) will perform with a jitter corresponding to the row time

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Setting (GenICam (p.143))</th>
<th>Mode / Setting (Device Specific (p.143))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot;</td>
<td>Continuous</td>
<td>Free running, no external trigger signal needed.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot;</td>
<td>OnDemand</td>
<td>Image acquisition triggered by command (software trigger).</td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot;</td>
<td>OnLowLevel</td>
<td>Start an exposure of a frame as long as the trigger input is below the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot;</td>
<td>OnHighLevel</td>
<td>Start an exposure of a frame as long as the trigger input is above the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot;</td>
<td>OnFallingEdge</td>
<td>Each falling edge of trigger signal acquires one image.</td>
</tr>
</tbody>
</table>
### Trigger Modes

<table>
<thead>
<tr>
<th>Trigger Configuration</th>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TriggerSelector = FrameStart</td>
<td>OnRisingEdge</td>
<td>Each rising edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>TriggerMode = On</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerSource = &lt;desired Line&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerActivation = RisingEdge</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ExposureMode = Timed</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TriggerConfiguration</th>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TriggerSelector = FrameStart</td>
<td>OnAnyEdge</td>
<td>Start the exposure of a frame when the trigger input level changes from high to low or from low to high.</td>
</tr>
<tr>
<td>TriggerMode = On</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerSource = &lt;desired Line&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerActivation = AnyEdge</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ExposureMode = Timed</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Line Mapping

<table>
<thead>
<tr>
<th>TriggerSource mvIMPACT Acquire</th>
<th>TriggerSource GenICam (BCX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP-IN0</td>
<td>Line4</td>
</tr>
<tr>
<td>GP-IN1</td>
<td>Line5</td>
</tr>
</tbody>
</table>

### 24.1.3 Sensor Data

#### Device Structure

- CMOS image sensor (Type 1/2.9")
- Number of effective pixels: 728 (H) x 544 (V)
- Unit cell size: 6.9um (H) x 6.9um (V)

### 24.1.3.1 Characteristics

### 24.1.3.2 Color version
24.2.1 Introduction

The CMOS sensor module (IMX273) incorporates the following features:

- resolution to 1456 x 1088 gray scale or RGB Bayer mosaic
- supports window AOI mode with faster readout
- global shutter
- programmable readout timing with free capture windows and partial scan
- many trigger modes (free-running, hardware-triggered)

24.2.2 Details of operation

The sensor uses a global shutter, i.e. light exposure takes place on all pixels in parallel, although subsequent readout is sequential.

24.2.2.1 Free running mode

In free running mode, the sensor reaches its maximum frame rate. This is done by overlapping erase, exposure and readout phase. The sensor timing in free running mode is fixed, so there is no control when to start an acquisition. This mode is used with trigger mode Continuous.

To calculate the maximum frames per second ($\text{FPS}_{\text{max}}$) in free running mode you will need following formula:
<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>InternalLineLength</td>
<td>520 (mvBlueCOUGAR-X) / 290 (mvBlueCOUGAR-XD)</td>
</tr>
<tr>
<td>VerticalBlankLines</td>
<td>42</td>
</tr>
<tr>
<td>SensorInClock</td>
<td>74.25 (@50 MHz Pixel clock)</td>
</tr>
<tr>
<td>NumberOfLVDS</td>
<td>4 (mvBlueCOUGAR-X) / 8 (mvBlueCOUGAR-XD)</td>
</tr>
</tbody>
</table>

\[
\text{FrameTime} = \frac{\text{InternalLineLength} \times 8 + \text{ImageHeight} + \text{VerticalBlankLines}}{\text{SensorInClock} \times \text{NumberOfLVDS}} \times \frac{1000}{1000}
\]

If exposure time is lower than frame time:

\[
\text{FPS}_{\text{max}} = \frac{1}{\text{FrameTime}}
\]

If exposure time is greater than frame time:

\[
\text{FPS}_{\text{max}} = \frac{1}{\text{ExposureTime}}
\]

**Note**

The exposure time step width is limited to the sensor’s row time of 7.01 us (mvBlueCOUGAR-X) or 3.91 us (mvBlueCOUGAR-XD) and therefore
- auto exposure with very low exposure times will perform with relatively large increments and
- exposure mode = TriggerWidth (if available) will perform with a jitter corresponding to the row time.

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Setting (GenICam p.143)</th>
<th>Mode / Setting (Device Specific p.143)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = Off&quot;</td>
<td>Continuous</td>
<td>Free running, no external trigger signal needed.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = Software&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnDemand</td>
<td>Image acquisition triggered by command (software trigger).</td>
</tr>
</tbody>
</table>

To trigger one frame execute the `TriggerSoftware@i` command then.
### Contents

<table>
<thead>
<tr>
<th>TriggerSelector</th>
<th>TriggerMode</th>
<th>TriggerSource</th>
<th>TriggerActivation</th>
<th>ExposureMode</th>
<th>OnLowLevel</th>
<th>Start an exposure of a frame as long as the trigger input is below the trigger threshold. (No Frame←Trigger!)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AcquisitionActive</td>
<td>On</td>
<td>&lt;desired Line&gt;</td>
<td>LevelLow</td>
<td>Timed</td>
<td>OnLowLevel</td>
<td></td>
</tr>
<tr>
<td>TriggerSelector</td>
<td>TriggerMode</td>
<td>TriggerSource</td>
<td>TriggerActivation</td>
<td>ExposureMode</td>
<td>OnHighLevel</td>
<td>Start an exposure of a frame as long as the trigger input is above the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>FrameStart</td>
<td>On</td>
<td>&lt;desired Line&gt;</td>
<td>LevelHigh</td>
<td>Timed</td>
<td>OnHighLevel</td>
<td></td>
</tr>
<tr>
<td>TriggerSelector</td>
<td>TriggerMode</td>
<td>TriggerSource</td>
<td>TriggerActivation</td>
<td>ExposureMode</td>
<td>OnFallingEdge</td>
<td>Each falling edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>FrameStart</td>
<td>On</td>
<td>&lt;desired Line&gt;</td>
<td>FallingEdge</td>
<td>Timed</td>
<td>OnFallingEdge</td>
<td></td>
</tr>
<tr>
<td>TriggerSelector</td>
<td>TriggerMode</td>
<td>TriggerSource</td>
<td>TriggerActivation</td>
<td>ExposureMode</td>
<td>OnRisingEdge</td>
<td>Each rising edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>FrameStart</td>
<td>On</td>
<td>&lt;desired Line&gt;</td>
<td>RisingEdge</td>
<td>Timed</td>
<td>OnRisingEdge</td>
<td></td>
</tr>
<tr>
<td>TriggerSelector</td>
<td>TriggerMode</td>
<td>TriggerSource</td>
<td>TriggerActivation</td>
<td>ExposureMode</td>
<td>OnAnyEdge</td>
<td>Start the exposure of a frame when the trigger input level changes from high to low or from low to high.</td>
</tr>
<tr>
<td>FrameStart</td>
<td>On</td>
<td>&lt;desired Line&gt;</td>
<td>AnyEdge</td>
<td>Timed</td>
<td>OnAnyEdge</td>
<td></td>
</tr>
</tbody>
</table>

### Line Mapping

(Line Mapping (TriggerSource mvIMPACT Acquire -> TriggerSource GenICam (valid values for <desired Line>))::

<table>
<thead>
<tr>
<th>TriggerSource mvIMPACT Acquire</th>
<th>TriggerSource GenICam(BCX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP-IN0</td>
<td>Line4</td>
</tr>
<tr>
<td>GP-IN1</td>
<td>Line5</td>
</tr>
</tbody>
</table>

### 24.2.3 Sensor Data

#### Device Structure

- CMOS image sensor (Type 1/2.9"
- Number of effective pixels: 1456 (H) x 1088 (V)
- Unit cell size: 3.45um (H) x 3.45um (V)
24.2.3.1 Characteristics

24.2.3.2 Color version

![Color response graph](image1)

24.2.3.3 Gray scale version

![Gray scale response graph](image2)

24.2.4 Device Feature And Property List

24.2.4.1 Gray scale version

24.2.4.2 Color version
24.3 mvBlueCOUGAR-X102k (1.8 Mpix [1608 x 1104])

24.3.1 Introduction

The CMOS sensor module (IMX432) incorporates the following features:

- resolution to 1608 x 1104 gray scale or RGB Bayer mosaic
- supports window AOI mode with faster readout
- global shutter
- programmable readout timing with free capture windows and partial scan
- many trigger modes (free-running, hardware-triggered)

24.3.2 Details of operation

The sensor uses a global shutter, i.e. light exposure takes place on all pixels in parallel, although subsequent readout is sequential.

24.3.2.1 Free running mode

In free running mode, the sensor reaches its maximum frame rate. This is done by overlapping erase, exposure and readout phase. The sensor timing in free running mode is fixed, so there is no control when to start an acquisition. This mode is used with trigger mode Continuous.

To calculate the maximum frames per second ($\text{FPS}_{\text{max}}$) in free running mode you will need following formula:

$$\text{FrameTime} = \frac{\text{InternalLineLength}}{\text{SensorInClock}} \times \frac{\text{ImageHeight} + \text{VerticalBlankLines}}{1000}$$

If exposure time is lower than frame time:

$$\text{FPS}_{\text{max}} = \frac{1}{\text{FrameTime}}$$

If exposure time is greater than frame time:

$$\text{FPS}_{\text{max}} = \frac{1}{\text{ExposureTime}}$$

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>InternalLineLength</td>
<td>466</td>
</tr>
<tr>
<td>VerticalBlankLines</td>
<td>74</td>
</tr>
<tr>
<td>SensorInClock</td>
<td>53.977 (@54 MHz Pixel clock)</td>
</tr>
<tr>
<td>NumberOfLVDS</td>
<td>4</td>
</tr>
</tbody>
</table>
The exposure time step width is limited to the sensor’s row time of 8.63 us and therefore:
- auto exposure with very low exposure times will perform with relatively large increments and
- exposure mode = TriggerWidth (if available) will perform with a jitter corresponding to the row time

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Setting (GenICam (p.143))</th>
<th>Mode / Setting (Device Specific (p.143))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot;</td>
<td>Continuous</td>
<td>Free running, no external trigger signal needed.</td>
</tr>
<tr>
<td>&quot;TriggerMode = Off&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot;</td>
<td>OnDemand</td>
<td>Image acquisition triggered by command (software trigger).</td>
</tr>
<tr>
<td>&quot;TriggerMode = On&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSource = Software&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;ExposureMode = Timed&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>To trigger one frame execute the TriggerSoftware@i command then.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot;</td>
<td>OnLowLevel</td>
<td>Start an exposure of a frame as long as the trigger input is below the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerMode = On&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSource = &lt;desired Line&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerActivation = LevelLow&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;ExposureMode = Timed&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot;</td>
<td>OnHighLevel</td>
<td>Start an exposure of a frame as long as the trigger input is above the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerMode = On&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSource = &lt;desired Line&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerActivation = LevelHigh&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;ExposureMode = Timed&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot;</td>
<td>OnFallingEdge</td>
<td>Each falling edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>&quot;TriggerMode = On&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSource = &lt;desired Line&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerActivation = FallingEdge&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;ExposureMode = Timed&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot;</td>
<td>OnRisingEdge</td>
<td>Each rising edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>&quot;TriggerMode = On&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSource = &lt;desired Line&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerActivation = RisingEdge&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;ExposureMode = Timed&quot;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
"TriggerSelector = FrameStart"
"TriggerMode = On"
"TriggerSource = <desired Line>"
"TriggerActivation = AnyEdge"
"ExposureMode = Timed"

| OnAnyEdge | Start the exposure of a frame when the trigger input level changes from high to low or from low to high. |

Line Mapping (TriggerSource mvIMPACT Acquire -> TriggerSource GenICam (valid values for <desired Line>)):

<table>
<thead>
<tr>
<th>TriggerSource mvIMPACT Acquire</th>
<th>TriggerSource GenICam(BCX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP-IN0</td>
<td>Line4</td>
</tr>
<tr>
<td>GP-IN1</td>
<td>Line5</td>
</tr>
</tbody>
</table>

24.3.3 Sensor Data

Device Structure

- CMOS image sensor (Type 1.1")
- Number of effective pixels: 1608 (H) x 1104 (V)
- Unit cell size: 9.0um (H) x 9.0um (V)

24.3.3.1 Characteristics

24.3.3.2 Color version
24.4.1 Introduction

The CMOS sensor module (IMX425) incorporates the following features:

- resolution to 1608 x 1104 gray scale or RGB Bayer mosaic
- supports window AOI mode with faster readout
- global shutter
- programmable readout timing with free capture windows and partial scan
- many trigger modes (free-running, hardware-triggered)

24.4.2 Details of operation

The sensor uses a global shutter, i.e. light exposure takes place on all pixels in parallel, although subsequent readout is sequential.

24.4.2.1 Free running mode

In free running mode, the sensor reaches its maximum frame rate. This is done by overlapping erase, exposure and readout phase. The sensor timing in free running mode is fixed, so there is no control when to start an acquisition. This mode is used with trigger mode Continuous.

To calculate the maximum frames per second ($\text{FPS}_{\text{max}}$) in free running mode you will need following formula:
<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>InternalLineLength</td>
<td>390 (&lt; 12 bit sensor digitization bit depth)</td>
</tr>
<tr>
<td>VerticalBlankLines</td>
<td>98</td>
</tr>
<tr>
<td>SensorInClock</td>
<td>53.977 (@54 MHz Pixel clock)</td>
</tr>
<tr>
<td>NumberOfLVDS</td>
<td>4</td>
</tr>
</tbody>
</table>

\[
\text{FrameTime} = \frac{\text{InternalLineLength}}{\text{SensorInClock}} \cdot \frac{\text{ImageHeight} + \text{VerticalBlankLines}}{1000}
\]

If exposure time is lower than frame time:

\[
\text{FPS}_{\text{max}} = \frac{1}{\text{FrameTime}}
\]

If exposure time is greater than frame time:

\[
\text{FPS}_{\text{max}} = \frac{1}{\text{ExposureTime}}
\]

**Note**

The exposure time step width is limited to the sensor’s row time of 7.23 us and therefore
- auto exposure with very low exposure times will perform with relatively large increments and
- exposure mode = TriggerWidth (if available) will perform with a jitter corresponding to the row time

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Setting (GenICam (p.143))</th>
<th>Mode / Setting (Device Specific (p.143))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot;</td>
<td>Continuous</td>
<td>Free running, no external trigger signal needed.</td>
</tr>
<tr>
<td>&quot;TriggerMode = Off&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot;</td>
<td>OnDemand</td>
<td>Image acquisition triggered by command (software trigger).</td>
</tr>
<tr>
<td>&quot;TriggerMode = On&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSource = Software&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;ExposureMode = Timed&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>To trigger one frame execute the TriggerSoftware@i command then.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Line Mapping (TriggerSource mvIMPACT Acquire -> TriggerSource GenICam (valid values for <desired Line>)):

<table>
<thead>
<tr>
<th>TriggerSource mvIMPACT Acquire</th>
<th>TriggerSource GenICam(BCX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP-IN0</td>
<td>Line4</td>
</tr>
<tr>
<td>GP-IN1</td>
<td>Line5</td>
</tr>
</tbody>
</table>

24.4.3 Sensor Data

**Device Structure**

- CMOS image sensor (Type 1.1")
- Number of effective pixels: 1608 (H) x 1104 (V)
- Unit cell size: 9.0um (H) x 9.0um (V)
24.4.3.1 Characteristics

24.4.3.2 Color version

24.4.3.3 Gray scale version

24.4.4 Device Feature And Property List

24.4.4.1 Gray scale version

24.4.4.2 Color version
24.5 mvBlueCOUGAR-X102n (2 Mpix [1632 x 1248])

24.5.1 Introduction

The CMOS sensor module (IMX430) incorporates the following features:

- resolution to 1632 x 1248 gray scale or RGB Bayer mosaic
- supports window AOI mode with faster readout
- global shutter
- programmable readout timing with free capture windows and partial scan
- many trigger modes (free-running, hardware-triggered)

24.5.2 Details of operation

The sensor uses a global shutter, i.e. light exposure takes place on all pixels in parallel, although subsequent readout is sequential.

24.5.2.1 Free running mode

In free running mode, the sensor reaches its maximum frame rate. This is done by overlapping erase, exposure and readout phase. The sensor timing in free running mode is fixed, so there is no control when to start an acquisition. This mode is used with trigger mode **Continuous**.

To calculate the maximum frames per second (FPS\(_{\text{max}}\)) in free running mode you will need following formula:

\[
\text{FrameTime} = \frac{\text{InternalLineLength}}{\text{SensorInClock}} \times \frac{\text{ImageHeight} + \text{VerticalBlankLines}}{1000}
\]

If exposure time is lower than frame time:

\[
\text{FPS}_{\text{max}} = \frac{1}{\text{FrameTime}}
\]

If exposure time is greater than frame time:

\[
\text{FPS}_{\text{max}} = \frac{1}{\text{ExposureTime}}
\]
The exposure time step width is limited to the sensor's row time of 8.74 us and therefore
- auto exposure with very low exposure times will perform with relatively large increments and
- exposure mode = TriggerWidth (if available) will perform with a jitter corresponding to the row time

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Setting (GenICam (p.143))</th>
<th>Mode / Setting (Device Specific (p.143))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = Off&quot;</td>
<td>Continuous</td>
<td>Free running, no external trigger signal needed.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = Software&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnDemand</td>
<td>Image acquisition triggered by command (software trigger).</td>
</tr>
<tr>
<td>To trigger one frame execute the TriggerSoftware@i command then.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelLow&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnLowLevel</td>
<td>Start an exposure of a frame as long as the trigger input is below the trigger threshold. (No Frame ← Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelHigh&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnHighLevel</td>
<td>Start an exposure of a frame as long as the trigger input is above the trigger threshold. (No Frame ← Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = FallingEdge&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnFallingEdge</td>
<td>Each falling edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = RisingEdge&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnRisingEdge</td>
<td>Each rising edge of trigger signal acquires one image.</td>
</tr>
</tbody>
</table>
"TriggerSelector = FrameStart"
"TriggerMode = On"
"TriggerSource = <desired Line>"
"TriggerActivation = AnyEdge"
"ExposureMode = Timed"

OnAnyEdge

Start the exposure of a frame when the trigger input level changes from high to low or from low to high.

<table>
<thead>
<tr>
<th>TriggerSource mvIMPACT Acquire</th>
<th>TriggerSource GenICam (BCX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP-IN0</td>
<td>Line4</td>
</tr>
<tr>
<td>GP-IN1</td>
<td>Line5</td>
</tr>
</tbody>
</table>

24.5.3 Sensor Data

Device Structure

- CMOS image sensor (Type 1/1.7")
- Number of effective pixels: 1632 (H) x 1248 (V)
- Unit cell size: 4.5um (H) x 4.5um (V)

24.5.3.1 Characteristics

24.5.3.2 Color version
24.6.1 Introduction

The CMOS sensor module (IMX174) incorporates the following features:

- resolution to 1936 x 1216 gray scale or RGB Bayer mosaic
- supports window AOI mode with faster readout
- high dynamic range (p.363)
- global shutter
- programmable readout timing with free capture windows and partial scan
- many trigger modes (free-running, hardware-triggered)

24.6.2 Details of operation

The sensor uses a global shutter, i.e. light exposure takes place on all pixels in parallel, although subsequent readout is sequential.

24.6.2.1 Free running mode

In free running mode, the sensor reaches its maximum frame rate. This is done by overlapping erase, exposure and readout phase. The sensor timing in free running mode is fixed, so there is no control when to start an acquisition. This mode is used with trigger mode Continuous.

To calculate the maximum frames per second (FPS<sub>max</sub>) in free running mode you will need following formula:
FrameTime = \frac{\text{InternalLineLength} \times 8 + \text{VerticalBlankLines}}{\frac{\text{SensorInClock} \times \text{NumberOfLVDS} \times 1000}{\text{ImageHeight}}}

If exposure time is lower than frame time:

\[
\text{FPS}_{\text{max}} = \frac{1}{\text{FrameTime}}
\]

If exposure time is greater than frame time:

\[
\text{FPS}_{\text{max}} = \frac{1}{\text{ExposureTime}}
\]

**Note**

The exposure time step width is limited to the sensor’s row time of 4.86 us and therefore
- auto exposure with very low exposure times will perform with relatively large increments and
- exposure mode = TriggerWidth (if available) will perform with a jitter corresponding to the row time

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Setting (GenICam (p.143))</th>
<th>Mode / Setting (Device Specific (p.143))</th>
<th>Description</th>
</tr>
</thead>
</table>
| "TriggerSelector = FrameStart"  
  "TriggerMode = Off" | Continuous | Free running, no external trigger signal needed. |
| "TriggerSelector = FrameStart"   
  "TriggerMode = On"   
  "TriggerSource = Software"  
  "ExposureMode = Timed" | OnDemand | Image acquisition triggered by command (software trigger). |

To trigger one frame execute the `TriggerSoftware@i` command then.
"TriggerSelector = AcquisitionActive"  
"TriggerMode = On"  
"TriggerSource = <desired Line>"  
"TriggerActivation = LevelLow"  
"ExposureMode = Timed"

<table>
<thead>
<tr>
<th>OnLowLevel</th>
<th>Start an exposure of a frame as long as the trigger input is below the trigger threshold. (No Frame←Trigger!)</th>
</tr>
</thead>
</table>

"TriggerSelector = AcquisitionActive"  
"TriggerMode = On"  
"TriggerSource = <desired Line>"  
"TriggerActivation = LevelHigh"  
"ExposureMode = Timed"

<table>
<thead>
<tr>
<th>OnHighLevel</th>
<th>Start an exposure of a frame as long as the trigger input is above the trigger threshold. (No Frame←Trigger!)</th>
</tr>
</thead>
</table>

"TriggerSelector = FrameStart"  
"TriggerMode = On"  
"TriggerSource = <desired Line>"  
"TriggerActivation = FallingEdge"  
"ExposureMode = Timed"

<table>
<thead>
<tr>
<th>OnFallingEdge</th>
<th>Each falling edge of trigger signal acquires one image.</th>
</tr>
</thead>
</table>

"TriggerSelector = FrameStart"  
"TriggerMode = On"  
"TriggerSource = <desired Line>"  
"TriggerActivation = RisingEdge"  
"ExposureMode = Timed"

<table>
<thead>
<tr>
<th>OnRisingEdge</th>
<th>Each rising edge of trigger signal acquires one image.</th>
</tr>
</thead>
</table>

"TriggerSelector = FrameStart"  
"TriggerMode = On"  
"TriggerSource = <desired Line>"  
"TriggerActivation = AnyEdge"  
"ExposureMode = Timed"

<table>
<thead>
<tr>
<th>OnAnyEdge</th>
<th>Start the exposure of a frame when the trigger input level changes from high to low or from low to high.</th>
</tr>
</thead>
</table>

Line Mapping (TriggerSource mvIMPACT Acquire -> TriggerSource GenICam (valid values for <desired Line>)):

<table>
<thead>
<tr>
<th>TriggerSource mvIMPACT Acquire</th>
<th>TriggerSource GenICam(BCX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP-IN0</td>
<td>Line4</td>
</tr>
<tr>
<td>GP-IN1</td>
<td>Line5</td>
</tr>
</tbody>
</table>

24.6.3 Sensor Data

Device Structure

- CMOS image sensor (Type 1/1.2")
- Number of effective pixels: 1936 (H) x 1216 (V)
- Unit cell size: 5.86um (H) x 5.86um (V)
24.6.3.1 Characteristics

24.6.3.2 Color version

![Color version graph]

24.6.3.3 Gray scale version

![Gray scale version graph]

24.6.4 Device Feature And Property List

24.6.4.1 Gray scale version

24.6.4.2 Color version
24.7 mvBlueCOUGAR-Xx04f (2.4 Mpix [1936 x 1216])

24.7.1 Introduction

The CMOS sensor module (IMX249) incorporates the following features:

- resolution to 1936 x 1216 gray scale or RGB Bayer mosaic
- supports window AOI mode with faster readout
- global shutter
- programmable readout timing with free capture windows and partial scan
- many trigger modes (free-running, hardware-triggered)

24.7.2 Details of operation

The sensor uses a global shutter, i.e. light exposure takes place on all pixels in parallel, although subsequent readout is sequential.

24.7.2.1 Free running mode

In free running mode, the sensor reaches its maximum frame rate. This is done by overlapping erase, exposure and readout phase. The sensor timing in free running mode is fixed, so there is no control when to start an acquisition. This mode is used with trigger mode Continuous.

To calculate the maximum frames per second (FPS\text{max}) in free running mode you will need following formula:

\[
\text{FrameTime} = \frac{\text{InternalLineLength} \times 8}{\text{SensorInClock} \times \text{NumberOfLVDS}} \times \frac{\text{ImageHeight} + \text{VerticalBlankLines}}{1000}
\]

If exposure time is lower than frame time:

\[
\text{FPS}\text{max} = \frac{1}{\text{FrameTime}}
\]

If exposure time is greater than frame time:

\[
\text{FPS}\text{max} = \frac{1}{\text{ExposureTime}}
\]
The exposure time step width is limited to the sensor’s row time of 19.4 us and therefore
- auto exposure with very low exposure times will perform with relatively large increments and
- exposure mode = TriggerWidth (if available) will perform with a jitter corresponding to the row time

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Setting (GenICam (\text{p.}^{143}))</th>
<th>Mode / Setting (Device Specific (\text{p.}^{143}))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot;</td>
<td>Continuous</td>
<td>Free running, no external trigger signal needed.</td>
</tr>
<tr>
<td>&quot;TriggerMode = Off&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot;</td>
<td>OnDemand</td>
<td>Image acquisition triggered by command (software trigger).</td>
</tr>
<tr>
<td>&quot;TriggerMode = On&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSource = Software&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;ExposureMode = Timed&quot;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To trigger one frame execute the `TriggerSoftware@i` command then.

| "TriggerSelector = FrameStart" | OnFallingEdge | Each falling edge of trigger signal acquires one image. |
| "TriggerMode = On" | | |
| "TriggerSource = <desired Line>" | | |
| "TriggerActivation = FallingEdge" | | |
| "ExposureMode = Timed" | | |

| "TriggerSelector = FrameStart" | OnRisingEdge | Each rising edge of trigger signal acquires one image. |
| "TriggerMode = On" | | |
| "TriggerSource = <desired Line>" | | |
| "TriggerActivation = RisingEdge" | | |
| "ExposureMode = Timed" | | |

| "TriggerSelector = FrameStart" | OnAnyEdge | Start the exposure of a frame when the trigger input level changes from high to low or from low to high. |
| "TriggerMode = On" | | |
| "TriggerSource = <desired Line>" | | |
| "TriggerActivation = AnyEdge" | | |
| "ExposureMode = Timed" | | |

Line Mapping (TriggerSource mvIMPACT Acquire \(\rightarrow\) TriggerSource GenICam (valid values for \(<\text{desired Line}>\))):

<table>
<thead>
<tr>
<th>TriggerSource mvIMPACT Acquire</th>
<th>TriggerSource GenICam(BCX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP-IN0</td>
<td>Line4</td>
</tr>
<tr>
<td>GP-IN1</td>
<td>Line5</td>
</tr>
</tbody>
</table>

MATRIX VISION GmbH
24.7.3 Sensor Data

Device Structure

- CMOS image sensor (Type 1/1.2"
- Number of effective pixels: 1936 (H) x 1216 (V)
- Unit cell size: 5.86um (H) x 5.86um (V)

24.7.3.1 Characteristics

24.7.3.2 Color version

24.7.3.3 Gray scale version
24.8.4.1 Gray scale version

24.8.4.2 Color version

24.8.1 Introduction

The CMOS sensor module (IMX252) incorporates the following features:

- resolution to 2064 x 1544 gray scale or RGB Bayer mosaic
- supports window AOI mode with faster readout
- global shutter
- programmable readout timing with free capture windows and partial scan
- many trigger modes (free-running, hardware-triggered)

24.8.2 Details of operation

The sensor uses a global shutter, i.e. light exposure takes place on all pixels in parallel, although subsequent readout is sequential.

24.8.2.1 Free running mode

In free running mode, the sensor reaches its maximum frame rate. This is done by overlapping erase, exposure and readout phase. The sensor timing in free running mode is fixed, so there is no control when to start an acquisition. This mode is used with trigger mode Continuous.

To calculate the maximum frames per second (FPS$_{max}$) in free running mode you will need following formula:

\[
\text{FrameTime} = \frac{\text{InternalLineLength}}{\text{SensorInClock}} \times \frac{\text{ImageHeight}}{\text{NumberOfLVDS}} \times \frac{8}{1000}
\]

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>InternalLineLength</td>
<td>380</td>
</tr>
<tr>
<td>VerticalBlankLines</td>
<td>37</td>
</tr>
<tr>
<td>SensorInClock</td>
<td>74.25 (@50 MHz Pixel clock)</td>
</tr>
<tr>
<td>NumberOfLVDS</td>
<td>8</td>
</tr>
</tbody>
</table>
If exposure time is lower than frame time:

\[
1 \quad \text{FPS}_{\text{max}} = \frac{1}{\text{FrameTime}}
\]

If exposure time is greater than frame time:

\[
1 \quad \text{FPS}_{\text{max}} = \frac{1}{\text{ExposureTime}}
\]

**Note**

The exposure time step width is limited to the sensor's row time of 5.13 us and therefore
- auto exposure with very low exposure times will perform with relatively large increments and
- exposure mode = TriggerWidth (if available) will perform with a jitter corresponding to the row time

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Setting (GenICam (p.143))</th>
<th>Mode / Setting (Device Specific (p.143))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = Off&quot;</td>
<td>Continuous</td>
<td>Free running, no external trigger signal needed.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = Software&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnDemand</td>
<td>Image acquisition triggered by command (software trigger).</td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = \langle\text{desired Line}\rangle&quot; &quot;TriggerActivation = LevelLow&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnLowLevel</td>
<td>Start an exposure of a frame as long as the trigger input is below the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = \langle\text{desired Line}\rangle&quot; &quot;TriggerActivation = LevelHigh&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnHighLevel</td>
<td>Start an exposure of a frame as long as the trigger input is above the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>TriggerSelector = FrameStart</td>
<td>OnFallingEdge</td>
<td>Each falling edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>----------------------------</td>
<td>--------------</td>
<td>------------------------------------------------------</td>
</tr>
<tr>
<td>&quot;TriggerMode = On&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSource = &lt;desired Line&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerActivation = FallingEdge&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;ExposureMode = Timed&quot;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>&quot;TriggerSelector = FrameStart&quot;</th>
<th>OnRisingEdge</th>
<th>Each rising edge of trigger signal acquires one image.</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerMode = On&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSource = &lt;desired Line&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerActivation = RisingEdge&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;ExposureMode = Timed&quot;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>&quot;TriggerSelector = FrameStart&quot;</th>
<th>OnAnyEdge</th>
<th>Start the exposure of a frame when the trigger input level changes from high to low or from low to high.</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerMode = On&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSource = &lt;desired Line&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerActivation = AnyEdge&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;ExposureMode = Timed&quot;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Line Mapping (TriggerSource mvIMPACT Acquire -> TriggerSource GenICam (valid values for <desired Line>)):

<table>
<thead>
<tr>
<th>TriggerSource mvIMPACT Acquire</th>
<th>TriggerSource GenICam(BCX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP-IN0</td>
<td>Line4</td>
</tr>
<tr>
<td>GP-IN1</td>
<td>Line5</td>
</tr>
</tbody>
</table>

24.8.3 Sensor Data

Device Structure

- CMOS image sensor (Type 1/1.8")

- Number of effective pixels: 2064 (H) x 1544 (V)

- Unit cell size: 3.45um (H) x 3.45um (V)
24.8.3.2 Color version

![Color version graph]

24.8.3.3 Gray scale version

![Gray scale version graph]

24.8.4 Device Feature And Property List

24.8.4.1 Gray scale version

24.8.4.2 Color version
24.9.1 Introduction

The CMOS sensor module (IMX265) incorporates the following features:

- resolution to 2064 x 1544 gray scale or RGB Bayer mosaic
- supports window AOI mode with faster readout
- global shutter
- programmable readout timing with free capture windows and partial scan
- many trigger modes (free-running, hardware-triggered)

24.9.2 Details of operation

The sensor uses a global shutter, i.e. light exposure takes place on all pixels in parallel, although subsequent readout is sequential.

24.9.2.1 Free running mode

In free running mode, the sensor reaches its maximum frame rate. This is done by overlapping erase, exposure and readout phase. The sensor timing in free running mode is fixed, so there is no control when to start an acquisition. This mode is used with trigger mode Continuous.

To calculate the maximum frames per second ($FPS_{\text{max}}$) in free running mode you will need following formula:

\[
\text{FrameTime} = \frac{\text{InternalLineLength} \times 8 \times \text{ImageHeight} + \text{VerticalBlankLines}}{\text{SensorInClock} \times \text{NumberOfLVDS} \times 1000}
\]

If exposure time is lower than frame time:

\[
FPS_{\text{max}} = \frac{1}{\text{FrameTime}}
\]

If exposure time is greater than frame time:

\[
FPS_{\text{max}} = \frac{1}{\text{ExposureTime}}
\]
The exposure time step width is limited to the sensor's row time of 11.4 us and therefore
- auto exposure with very low exposure times will perform with relatively large increments and
- exposure mode = TriggerWidth (if available) will perform with a jitter corresponding to the row time.

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Setting (GenICam (p. 143))</th>
<th>Mode / Setting (Device Specific (p. 143))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = Off&quot;</td>
<td>Continuous</td>
<td>Free running, no external trigger signal needed.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = Software&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnDemand</td>
<td>Image acquisition triggered by command (software trigger).</td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelLow&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnLowLevel</td>
<td>Start an exposure of a frame as long as the trigger input is below the trigger threshold. (No Frame ← Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelHigh&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnHighLevel</td>
<td>Start an exposure of a frame as long as the trigger input is above the trigger threshold. (No Frame ← Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = FallingEdge&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnFallingEdge</td>
<td>Each falling edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = RisingEdge&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnRisingEdge</td>
<td>Each rising edge of trigger signal acquires one image.</td>
</tr>
</tbody>
</table>
"TriggerSelector = FrameStart"
"TriggerMode = On"
"TriggerSource = <desired Line>"
"TriggerActivation = AnyEdge"
"ExposureMode = Timed"

<table>
<thead>
<tr>
<th>OnAnyEdge</th>
<th>Start the exposure of a frame when the trigger input level changes from high to low or from low to high.</th>
</tr>
</thead>
</table>

Line Mapping (TriggerSource mvIMPACT Acquire -> TriggerSource GenICam (valid values for <desired Line>)):

<table>
<thead>
<tr>
<th>TriggerSource mvIMPACT Acquire</th>
<th>TriggerSource GenICam (BCX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP-IN0</td>
<td>Line4</td>
</tr>
<tr>
<td>GP-IN1</td>
<td>Line5</td>
</tr>
</tbody>
</table>

24.9.3 Sensor Data

Device Structure

- CMOS image sensor (Type 1/1.8")
- Number of effective pixels: 2064 (H) x 1544 (V)
- Unit cell size: 3.45um (H) x 3.45um (V)

24.9.3.1 Characteristics

24.9.3.2 Color version
24.10.1 Introduction

The CMOS sensor module (IMX250) incorporates the following features:

- resolution to 2464 x 2056 gray scale or RGB Bayer mosaic
- supports window AOI mode with faster readout
- global shutter
- programmable readout timing with free capture windows and partial scan
- many trigger modes (free-running, hardware-triggered)

24.10.2 Details of operation

The sensor uses a global shutter, i.e. light exposure takes place on all pixels in parallel, although subsequent readout is sequential.

24.10.2.1 Free running mode

In free running mode, the sensor reaches its maximum frame rate. This is done by overlapping erase, exposure and readout phase. The sensor timing in free running mode is fixed, so there is no control when to start an acquisition. This mode is used with trigger mode **Continuous**.

To calculate the maximum frames per second ($\text{FPS}_{\text{max}}$) in free running mode you will need following formula:
InternalLineLength * 8 ImageHeight + VerticalBlankLines
FrameTime = ------------------------------ * ------------------------------
SensorInClock * NumberOfLVDS 1000

If exposure time is lower than frame time:

\[
FPS_{\text{max}} = \frac{1}{\text{FrameTime}}
\]

If exposure time is greater than frame time:

\[
FPS_{\text{max}} = \frac{1}{\text{ExposureTime}}
\]

**Note**

The exposure time step width is limited to the sensor's row time of 5.94 us and therefore
- auto exposure with very low exposure times will perform with relatively large increments and
- exposure mode = TriggerWidth (if available) will perform with a jitter corresponding to the row time.

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Setting (GenICam (p.143))</th>
<th>Mode / Setting (Device Specific (p.143))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = Off&quot;</td>
<td>Continuous</td>
<td>Free running, no external trigger signal needed.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = Software&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnDemand</td>
<td>Image acquisition triggered by command (software trigger).</td>
</tr>
</tbody>
</table>

To trigger one frame execute the `TriggerSoftware@i` command then.
"TriggerSelector = AcquisitionActive"
"TriggerMode = On"
"TriggerSource = <desired Line>"
"TriggerActivation = LevelLow"
"ExposureMode = Timed"

OnLowLevel
Start an exposure of a frame as long as the trigger input is below the trigger threshold. (No Frame←Trigger!)

"TriggerSelector = AcquisitionActive"
"TriggerMode = On"
"TriggerSource = <desired Line>"
"TriggerActivation = LevelHigh"
"ExposureMode = Timed"

OnHighLevel
Start an exposure of a frame as long as the trigger input is above the trigger threshold. (No Frame←Trigger!)

"TriggerSelector = FrameStart"
"TriggerMode = On"
"TriggerSource = <desired Line>"
"TriggerActivation = FallingEdge"
"ExposureMode = Timed"

OnFallingEdge
Each falling edge of trigger signal acquires one image.

"TriggerSelector = FrameStart"
"TriggerMode = On"
"TriggerSource = <desired Line>"
"TriggerActivation = RisingEdge"
"ExposureMode = Timed"

OnRisingEdge
Each rising edge of trigger signal acquires one image.

"TriggerSelector = FrameStart"
"TriggerMode = On"
"TriggerSource = <desired Line>"
"TriggerActivation = AnyEdge"
"ExposureMode = Timed"

OnAnyEdge
Start the exposure of a frame when the trigger input level changes from high to low or from low to high.

Line Mapping (TriggerSource mvIMPACT Acquire -> TriggerSource GenICam (valid values for <desired Line>)):

<table>
<thead>
<tr>
<th>TriggerSource mvIMPACT Acquire</th>
<th>TriggerSource GenICam(BCX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP-IN0</td>
<td>Line4</td>
</tr>
<tr>
<td>GP-IN1</td>
<td>Line5</td>
</tr>
</tbody>
</table>

24.10.3 Sensor Data

Device Structure

- CMOS image sensor (Type 2/3")
- Number of effective pixels: 2464 (H) x 2056 (V)
- Unit cell size: 3.45um (H) x 3.45um (V)
24.10.3.1 Characteristics

24.10.3.2 Color version

![Color version graph](image1)

24.10.3.3 Gray scale version

![Gray scale version graph](image2)
24.10.4 Device Feature And Property List

24.10.4.1 Gray scale version

24.10.4.2 Color version

24.11 mvBlueCOUGAR-Xx05b (5.1 Mpix [2464 x 2056])

24.11.1 Introduction

The CMOS sensor module (IMX264) incorporates the following features:

- resolution to 2464 x 2056 gray scale or RGB Bayer mosaic
- supports window AOI mode with faster readout
- global shutter
- programmable readout timing with free capture windows and partial scan
- many trigger modes (free-running, hardware-triggered)

24.11.2 Details of operation

The sensor uses a global shutter, i.e. light exposure takes place on all pixels in parallel, although subsequent readout is sequential.

24.11.2.1 Free running mode

In free running mode, the sensor reaches its maximum frame rate. This is done by overlapping erase, exposure and readout phase. The sensor timing in free running mode is fixed, so there is no control when to start an acquisition. This mode is used with trigger mode Continuous.

To calculate the maximum frames per second ($\text{FPS}_{\text{max}}$) in free running mode you will need following formula:

\[
\text{FrameTime} = \frac{\text{InternalLineLength} \times 8 + \text{ImageHeight} \times \text{VerticalBlankLines}}{\text{SensorInClock} \times \text{NumberOfLVDS}} \times \frac{1000}{1000}
\]

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>InternalLineLength</td>
<td>498</td>
</tr>
<tr>
<td>VerticalBlankLines</td>
<td>37</td>
</tr>
<tr>
<td>SensorInClock</td>
<td>74.25 (@50 MHz Pixel clock)</td>
</tr>
<tr>
<td>NumberOfLVDS</td>
<td>4</td>
</tr>
</tbody>
</table>
If exposure time is lower than frame time:

\[
 FPS_{\text{max}} = \frac{1}{\text{FrameTime}}
\]

If exposure time is greater than frame time:

\[
 FPS_{\text{max}} = \frac{1}{\text{ExposureTime}}
\]

**Note**

The exposure time step width is limited to the sensor's row time of 13.4 us and therefore
- auto exposure with very low exposure times will perform with relatively large increments and
- exposure mode = TriggerWidth (if available) will perform with a jitter corresponding to the row time.

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Setting (GenICam (p.143))</th>
<th>Mode / Setting (Device Specific (p.143))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = Off&quot;</td>
<td>Continuous</td>
<td>Free running, no external trigger signal needed.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = Software&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnDemand</td>
<td>Image acquisition triggered by command (software trigger).</td>
</tr>
<tr>
<td>To trigger one frame execute the TriggerSoftware@i command then.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelLow&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnLowLevel</td>
<td>Start an exposure of a frame as long as the trigger input is below the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelHigh&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnHighLevel</td>
<td>Start an exposure of a frame as long as the trigger input is above the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>TriggerSelector = FrameStart</td>
<td>OnFallingEdge</td>
<td>Each falling edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------</td>
<td>------------------------------------------------------</td>
</tr>
<tr>
<td>TriggerMode = On</td>
<td>OnRisingEdge</td>
<td>Each rising edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>TriggerSource = &lt;desired Line&gt;</td>
<td>OnAnyEdge</td>
<td>Start the exposure of a frame when the trigger input level changes from high to low or from low to high.</td>
</tr>
<tr>
<td>TriggerActivation = FallingEdge</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ExposureMode = Timed</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Line Mapping (TriggerSource mvIMPACT Acquire -> TriggerSource GenICam (valid values for <desired Line>)):

<table>
<thead>
<tr>
<th>TriggerSource mvIMPACT Acquire</th>
<th>TriggerSource GenICam(BCX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP-IN0</td>
<td>Line4</td>
</tr>
<tr>
<td>GP-IN1</td>
<td>Line5</td>
</tr>
</tbody>
</table>

24.11.3 Sensor Data

Device Structure

- CMOS image sensor (Type 2/3")
- Number of effective pixels: 2464 (H) x 2056 (V)
- Unit cell size: 3.45um (H) x 3.45um (V)

24.11.3.1 Characteristics
24.11.3.2 Color version

![Color version diagram]

24.11.3.3 Gray scale version

![Gray scale version diagram]

24.11.4 Device Feature And Property List

24.11.4.1 Gray scale version

24.11.4.2 Color version
24.12  mvBlueCOUGAR-XD107 (7.1 Mpix [3216 x 2208])

24.12.1 Introduction

The CMOS sensor module (IMX420) incorporates the following features:

- resolution to 3216 x 2208 gray scale or RGB Bayer mosaic
- supports window AOI mode with faster readout
- global shutter
- programmable readout timing with free capture windows and partial scan
- many trigger modes (free-running, hardware-triggered)

24.12.2 Details of operation

The sensor uses a global shutter, i.e. light exposure takes place on all pixels in parallel, although subsequent readout is sequential.

24.12.2.1 Free running mode

In free running mode, the sensor reaches its maximum frame rate. This is done by overlapping erase, exposure and readout phase. The sensor timing in free running mode is fixed, so there is no control when to start an acquisition. This mode is used with trigger mode Continuous.

To calculate the maximum frames per second ($FPS_{\text{max}}$) in free running mode you will need following formula:

\[
\text{FrameTime} = \frac{\text{InternalLineLength} \times 8 \times \text{ImageHeight} + \text{VerticalBlankLines}}{\text{SensorInClock} \times \text{NumberOfLVDS} \times 1000}
\]

If exposure time is lower than frame time:

\[
FPS_{\text{max}} = \frac{1}{\text{FrameTime}}
\]

If exposure time is greater than frame time:

\[
FPS_{\text{max}} = \frac{1}{\text{ExposureTime}}
\]
The exposure time step width is limited to the sensor’s row time of 7.23 us and therefore
- auto exposure with very low exposure times will perform with relatively large increments and
- exposure mode = TriggerWidth (if available) will perform with a jitter corresponding to the row time

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Setting (GenICam (p.143))</th>
<th>Mode / Setting (Device Specific (p.143))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = Off&quot;</td>
<td>Continuous</td>
<td>Free running, no external trigger signal needed.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = Software&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnDemand</td>
<td>Image acquisition triggered by command (software trigger).</td>
</tr>
</tbody>
</table>

To trigger one frame execute the TriggerSoftware@i command then.

| "TriggerSelector = AcquisitionActive" "TriggerMode = On" "TriggerSource = <desired Line>" "TriggerActivation = LevelLow" "ExposureMode = Timed" | OnLowLevel | Start an exposure of a frame as long as the trigger input is below the trigger threshold. (No Frame←Trigger!) |
| "TriggerSelector = AcquisitionActive" "TriggerMode = On" "TriggerSource = <desired Line>" "TriggerActivation = LevelHigh" "ExposureMode = Timed" | OnHighLevel | Start an exposure of a frame as long as the trigger input is above the trigger threshold. (No Frame←Trigger!) |
| "TriggerSelector = FrameStart" "TriggerMode = On" "TriggerSource = <desired Line>" "TriggerActivation = FallingEdge" "ExposureMode = Timed" | OnFallingEdge | Each falling edge of trigger signal acquires one image. |
| "TriggerSelector = FrameStart" "TriggerMode = On" "TriggerSource = <desired Line>" "TriggerActivation = RisingEdge" "ExposureMode = Timed" | OnRisingEdge | Each rising edge of trigger signal acquires one image. |
"TriggerSelector = FrameStart"
"TriggerMode = On"
"TriggerSource = <desired Line>
"TriggerActivation = AnyEdge"
"ExposureMode = Timed"

<table>
<thead>
<tr>
<th>OnAnyEdge</th>
<th>Start the exposure of a frame when the trigger input level changes from high to low or from low to high.</th>
</tr>
</thead>
</table>

Line Mapping (TriggerSource mvIMPACT Acquire -> TriggerSource GenICam (valid values for <desired Line>):

<table>
<thead>
<tr>
<th>TriggerSource mvIMPACT Acquire</th>
<th>TriggerSource GenICam(BCX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP-IN0</td>
<td>Line4</td>
</tr>
<tr>
<td>GP-IN1</td>
<td>Line5</td>
</tr>
</tbody>
</table>

24.12.3 Sensor Data

Device Structure

- CMOS image sensor (Type 1.1")
- Number of effective pixels: 3216 (H) x 2208 (V)
- Unit cell size: 4.5µm (H) x 4.5µm (V)

24.12.3.1 Characteristics

24.12.3.2 Color version

![Graph showing spectral response](attachment:filter_ir_cut.png)
24.13.1 Introduction

The CMOS sensor module (IMX428) incorporates the following features:

- resolution to 3216 x 2208 gray scale or RGB Bayer mosaic
- supports window AOI mode with faster readout
- global shutter
- programmable readout timing with free capture windows and partial scan
- many trigger modes (free-running, hardware-triggered)
24.13.2 Details of operation

The sensor uses a global shutter, i.e. light exposure takes place on all pixels in parallel, although subsequent readout is sequential.

24.13.2.1 Free running mode

In free running mode, the sensor reaches its maximum frame rate. This is done by overlapping erase, exposure and readout phase. The sensor timing in free running mode is fixed, so there is no control when to start an acquisition. This mode is used with trigger mode Continuous.

To calculate the maximum frames per second (FPS_max) in free running mode you will need following formula:

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>InternalLineLength</td>
<td>466</td>
</tr>
<tr>
<td>VerticalBlankLines</td>
<td>75</td>
</tr>
<tr>
<td>SensorInClock</td>
<td>53.977 (@54 MHz Pixel clock)</td>
</tr>
<tr>
<td>NumberOfLVDS</td>
<td>8</td>
</tr>
</tbody>
</table>

\[
\text{FrameTime} = \frac{\text{InternalLineLength} \times 8}{\text{SensorInClock} \times \text{NumberOfLVDS}} - \frac{\text{VerticalBlankLines} + \text{ImageHeight}}{1000}
\]

If exposure time is lower than frame time:

\[
\text{FPS}_{\text{max}} = \frac{1}{\text{FrameTime}}
\]

If exposure time is greater than frame time:

\[
\text{FPS}_{\text{max}} = \frac{1}{\text{ExposureTime}}
\]

**Note**

The exposure time step width is limited to the sensor's row time of 8.63 us and therefore

- auto exposure with very low exposure times will perform with relatively large increments and
- exposure mode = TriggerWidth (if available) will perform with a jitter corresponding to the row time

The following trigger modes are available:
<table>
<thead>
<tr>
<th>Setting (GenICam (p.143))</th>
<th>Mode / Setting (Device Specific (p.143))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = Off&quot;</td>
<td>Continuous</td>
<td>Free running, no external trigger signal needed.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = Software&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnDemand</td>
<td>Image acquisition triggered by command (software trigger).</td>
</tr>
<tr>
<td>To trigger one frame execute the TriggerSoftware@i command then.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelLow&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnLowLevel</td>
<td>Start an exposure of a frame as long as the trigger input is below the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelHigh&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnHighLevel</td>
<td>Start an exposure of a frame as long as the trigger input is above the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = FallingEdge&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnFallingEdge</td>
<td>Each falling edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = RisingEdge&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnRisingEdge</td>
<td>Each rising edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = AnyEdge&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnAnyEdge</td>
<td>Start the exposure of a frame when the trigger input level changes from high to low or from low to high.</td>
</tr>
</tbody>
</table>
Line Mapping (TriggerSource mvIMPACT Acquire -> TriggerSource GenICam (valid values for <desired Line>)):

<table>
<thead>
<tr>
<th>TriggerSource mvIMPACT Acquire</th>
<th>TriggerSource GenICam(BCX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP-IN0</td>
<td>Line4</td>
</tr>
<tr>
<td>GP-IN1</td>
<td>Line5</td>
</tr>
</tbody>
</table>

24.13.3 Sensor Data

Device Structure

- CMOS image sensor (Type 1.1“)
- Number of effective pixels: 3216 (H) x 2208 (V)
- Unit cell size: 4.5um (H) x 4.5um (V)

24.13.3.1 Characteristics

24.13.3.2 Color version

![Diagram of relative response versus wavelength with FILTER IR-CUT highlighted](image-url)
24.14 mvBlueCOUGAR-X[D]109b (8.9 Mpix [4112 x 2176])

24.14.1 Introduction

The CMOS sensor module (IMX267) incorporates the following features:

- resolution to 4112 x 2176 gray scale or RGB Bayer mosaic
- supports window AOI mode with faster readout
- global shutter
- programmable readout timing with free capture windows and partial scan
- many trigger modes (free-running, hardware-triggered)
24.14.2 Details of operation

The sensor uses a global shutter, i.e. light exposure takes place on all pixels in parallel, although subsequent readout is sequential.

24.14.2.1 Free running mode

In free running mode, the sensor reaches its maximum frame rate. This is done by overlapping erase, exposure and readout phase. The sensor timing in free running mode is fixed, so there is no control when to start an acquisition. This mode is used with trigger mode Continuous.

To calculate the maximum frames per second (FPS_{max}) in free running mode you will need following formula:

\[
\text{FPS}_{\text{max}} = \frac{1}{\text{FrameTime}}
\]

If exposure time is lower than frame time:

\[
\text{FPS}_{\text{max}} = \frac{1}{\text{FrameTime}}
\]

If exposure time is greater than frame time:

\[
\text{FPS}_{\text{max}} = \frac{1}{\text{ExposureTime}}
\]

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>InternalLineLength</td>
<td>807 (mvBlueCOUGAR-X) / 1041 (mvBlueCOUGAR-XD)</td>
</tr>
<tr>
<td>VerticalBlankLines</td>
<td>54 (mvBlueCOUGAR-X) / 54 (mvBlueCOUGAR-XD)</td>
</tr>
<tr>
<td>SensorInClock</td>
<td>74.25 (@50 MHz Pixel clock)</td>
</tr>
<tr>
<td>NumberOfLVDS</td>
<td>4 (mvBlueCOUGAR-X) / 8 (mvBlueCOUGAR-XD)</td>
</tr>
</tbody>
</table>

FrameTime = \frac{\text{InternalLineLength} \times 8 + \text{ImageHeight} + \text{VerticalBlankLines}}{\text{SensorInClock} \times \text{NumberOfLVDS} \times 1000}

Note

The exposure time step width is limited to the sensor's row time of 21.78 us (mvBlueCOUGAR-X) or 14.04 us (mvBlueCOUGAR-XD) and therefore
- auto exposure with very low exposure times will perform with relatively large increments and
- exposure mode = TriggerWidth (if available) will perform with a jitter corresponding to the row time

The following trigger modes are available:
<table>
<thead>
<tr>
<th>Setting (GenICam (p.143))</th>
<th>Mode / Setting (Device Specific (p.143))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = Off&quot;</td>
<td>Continuous</td>
<td>Free running, no external trigger signal needed.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = Software&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnDemand</td>
<td>Image acquisition triggered by command (software trigger).</td>
</tr>
<tr>
<td>To trigger one frame execute the TriggerSoftware@i command then.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelLow&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnLowLevel</td>
<td>Start an exposure of a frame as long as the trigger input is below the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelHigh&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnHighLevel</td>
<td>Start an exposure of a frame as long as the trigger input is above the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = FallingEdge&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnFallingEdge</td>
<td>Each falling edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = RisingEdge&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnRisingEdge</td>
<td>Each rising edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = AnyEdge&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnAnyEdge</td>
<td>Start the exposure of a frame when the trigger input level changes from high to low or from low to high.</td>
</tr>
</tbody>
</table>
Line Mapping: \( \text{TriggerSource mvIMPACT Acquire} \rightarrow \text{TriggerSource GenICam} \) (valid values for \(< \text{desired Line} >\)):

<table>
<thead>
<tr>
<th>TriggerSource mvIMPACT Acquire</th>
<th>TriggerSource GenICam(BCX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP-IN0</td>
<td>Line4</td>
</tr>
<tr>
<td>GP-IN1</td>
<td>Line5</td>
</tr>
</tbody>
</table>

24.14.3 Sensor Data

Device Structure

- CMOS image sensor (Type 1")
- Number of effective pixels: 4112 (H) x 2176 (V)
- Unit cell size: 3.45um (H) x 3.45um (V)

24.14.3.1 Characteristics

24.14.3.2 Color version

---

FILTER IR-CUT

---
24.15 mvBlueCOUGAR-X[D]1012b (12.4 Mpix [4112 x 3008])

24.15.1 Introduction

The CMOS sensor module (IMX304) incorporates the following features:

- resolution to 4112 x 3008 gray scale or RGB Bayer mosaic
- supports window AOI mode with faster readout
- global shutter
- programmable readout timing with free capture windows and partial scan
- many trigger modes (free-running, hardware-triggered)
24.15.2 Details of operation

The sensor uses a global shutter, i.e. light exposure takes place on all pixels in parallel, although subsequent readout is sequential.

24.15.2.1 Free running mode

In free running mode, the sensor reaches its maximum frame rate. This is done by overlapping erase, exposure and readout phase. The sensor timing in free running mode is fixed, so there is no control when to start an acquisition. This mode is used with trigger mode Continuous.

To calculate the maximum frames per second ($\text{FPS}_{\text{max}}$) in free running mode you will need following formula:

\[
\text{FrameTime} = \frac{\text{InternalLineLength} \times 8 + \text{VerticalBlankLines} \times \text{ImageHeight}}{\text{SensorInClock} \times \text{NumberOfLVDS} \times 1000}
\]

If exposure time is lower than frame time:

\[
\text{FPS}_{\text{max}} = \frac{1}{\text{FrameTime}}
\]

If exposure time is greater than frame time:

\[
\text{FPS}_{\text{max}} = \frac{1}{\text{ExposureTime}}
\]

Note

The exposure time step width is limited to the sensor’s row time of 21.78 us (mvBlueCOUGAR-X) or 14.04 us (mvBlueCOUGAR-XD) and therefore
- auto exposure with very low exposure times will perform with relatively large increments and
- exposure mode = TriggerWidth (if available) will perform with a jitter corresponding to the row time

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>InternalLineLength</td>
<td>807 (mvBlueCOUGAR-X) / 1041 (mvBlueCOUGAR-XD)</td>
</tr>
<tr>
<td>VerticalBlankLines</td>
<td>54 (mvBlueCOUGAR-X) / 54 (mvBlueCOUGAR-XD)</td>
</tr>
<tr>
<td>SensorInClock</td>
<td>74.25 (@50 MHz Pixel clock)</td>
</tr>
<tr>
<td>NumberOfLVDS</td>
<td>4 (mvBlueCOUGAR-X) / 8 (mvBlueCOUGAR-XD)</td>
</tr>
<tr>
<td>Setting (GenICam (p.143))</td>
<td>Mode / Setting (Device Specific (p.143))</td>
</tr>
<tr>
<td>---------------------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = Off&quot;</td>
<td>Continuous</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = Software&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnDemand</td>
</tr>
<tr>
<td>To trigger one frame execute the TriggerSoftware@i command then.</td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelLow&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnLowLevel</td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelHigh&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnHighLevel</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = FallingEdge&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnFallingEdge</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = RisingEdge&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnRisingEdge</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = AnyEdge&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnAnyEdge</td>
</tr>
</tbody>
</table>
Line Mapping (TriggerSource mvIMPACT Acquire -> TriggerSource GenICam (valid values for <desired Line>>):

<table>
<thead>
<tr>
<th>TriggerSource mvIMPACT Acquire</th>
<th>TriggerSource GenICam(BCX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP-IN0</td>
<td>Line4</td>
</tr>
<tr>
<td>GP-IN1</td>
<td>Line5</td>
</tr>
</tbody>
</table>

24.15.3 Sensor Data

Device Structure

- CMOS image sensor (Type 1.1“)
- Number of effective pixels: 4112 (H) x 3008 (V)
- Unit cell size: 3.45um (H) x 3.45um (V)

24.15.3.1 Characteristics

24.15.3.2 Color version

![Color version graph](image)

24.15.3.3 Gray scale version

![Gray scale version graph](image)
24.15.4 Device Feature And Property List

24.15.4.1 Gray scale version

24.15.4.2 Color version

24.16 mvBlueCOUGAR-XD1031 (31.5 Mpix [6480 x 4856])

24.16.1 Introduction

The CMOS sensor module (IMX342) incorporates the following features:

- resolution to 6480 x 4856 gray scale or RGB Bayer mosaic
- supports window AOI mode with faster readout
- global shutter
- programmable readout timing with free capture windows and partial scan
- many trigger modes (free-running, hardware-triggered)

24.16.2 Details of operation

The sensor uses a global shutter, i.e. light exposure takes place on all pixels in parallel, although subsequent readout is sequential.

24.16.2.1 Free running mode

In free running mode, the sensor reaches its maximum frame rate. This is done by overlapping erase, exposure and readout phase. The sensor timing in free running mode is fixed, so there is no control when to start an acquisition. This mode is used with trigger mode **Continuous**.

To calculate the maximum frames per second (\(\text{FPS}_{\text{max}}\)) in free running mode you will need following formula:

\[
\text{FrameTime} = \frac{\text{InternalLineLength} \times 8 \times \text{NumberOfLVDS}}{\text{SensorInClock} \times \text{NumberOfLVDS}} \times \frac{\text{ImageHeight} + \text{VerticalBlankLines}}{1000}
\]

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>InternalLineLength</td>
<td>910</td>
</tr>
<tr>
<td>VerticalBlankLines</td>
<td>50</td>
</tr>
<tr>
<td>SensorInClock</td>
<td>53.977 (@54 MHz Pixel clock)</td>
</tr>
<tr>
<td>NumberOfLVDS</td>
<td>8</td>
</tr>
</tbody>
</table>
If exposure time is lower than frame time:

\[
FPS_{\text{max}} = \frac{1}{\text{FrameTime}}
\]

If exposure time is greater than frame time:

\[
FPS_{\text{max}} = \frac{1}{\text{ExposureTime}}
\]

**Note**

The exposure time step width is limited to the sensor's row time of 16.86 us and therefore
- auto exposure with very low exposure times will perform with relatively large increments and
- exposure mode = TriggerWidth (if available) will perform with a jitter corresponding to the row time

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Setting (GenICam (p.143))</th>
<th>Mode / Setting (Device Specific (p.143))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = Off&quot;</td>
<td>Continuous</td>
<td>Free running, no external trigger signal needed.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = Software&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnDemand</td>
<td>Image acquisition triggered by command (software trigger).</td>
</tr>
<tr>
<td>To trigger one frame execute the TriggerSoftware@i command then.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelLow&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnLowLevel</td>
<td>Start an exposure of a frame as long as the trigger input is below the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelHigh&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnHighLevel</td>
<td>Start an exposure of a frame as long as the trigger input is above the trigger threshold. (No Frame←Trigger!)</td>
</tr>
</tbody>
</table>
"TriggerSelector = FrameStart"
"TriggerMode = On"
"TriggerSource = <desired Line>">
"TriggerActivation = FallingEdge"
"ExposureMode = Timed"

<table>
<thead>
<tr>
<th>TriggerSelector = FrameStart</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerMode = On&quot;</td>
</tr>
<tr>
<td>&quot;TriggerSource = &lt;desired Line&gt;&quot;&gt;</td>
</tr>
<tr>
<td>&quot;TriggerActivation = FallingEdge&quot;</td>
</tr>
<tr>
<td>&quot;ExposureMode = Timed&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OnFallingEdge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Each falling edge of trigger signal acquires one image.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TriggerSelector = FrameStart</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerMode = On&quot;</td>
</tr>
<tr>
<td>&quot;TriggerSource = &lt;desired Line&gt;&quot;&gt;</td>
</tr>
<tr>
<td>&quot;TriggerActivation = RisingEdge&quot;</td>
</tr>
<tr>
<td>&quot;ExposureMode = Timed&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OnRisingEdge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Each rising edge of trigger signal acquires one image.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TriggerSelector = FrameStart</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerMode = On&quot;</td>
</tr>
<tr>
<td>&quot;TriggerSource = &lt;desired Line&gt;&quot;&gt;</td>
</tr>
<tr>
<td>&quot;TriggerActivation = AnyEdge&quot;</td>
</tr>
<tr>
<td>&quot;ExposureMode = Timed&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OnAnyEdge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start the exposure of a frame when the trigger input level changes from high to low or from low to high.</td>
</tr>
</tbody>
</table>

Line Mapping (TriggerSource mvIMPACT Acquire -> TriggerSource GenICam (valid values for <desired Line>>):

<table>
<thead>
<tr>
<th>TriggerSource mvIMPACT Acquire</th>
<th>TriggerSource GenICam(BCX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP-IN0</td>
<td>Line4</td>
</tr>
<tr>
<td>GP-IN1</td>
<td>Line5</td>
</tr>
</tbody>
</table>

24.16.3 Sensor Data

Device Structure

- CMOS image sensor (Type APS-C)
- Number of effective pixels: 6480 (H) x 4860 (V)
- Unit cell size: 3.45um (H) x 3.45um (V)

24.16.3.1 Characteristics
24.16.3.2 Color version

24.16.3.3 Gray scale version

24.16.4 Device Feature And Property List

24.16.4.1 Gray scale version

24.16.4.2 Color version
Appendix A.3 Starvis CMOS specific camera / sensor data

25.1 mvBlueCOUGAR-X1020 (20.5 Mpix [5544 x 3692])

25.1.1 Introduction

The CMOS sensor module (IMX183) incorporates the following features:

- resolution to 5544 x 3692 gray scale or RGB Bayer mosaic
- supports window AOI mode with faster readout
- rolling shutter
- programmable readout timing with free capture windows and partial scan
- many trigger modes (free-running, hardware-triggered)

25.1.2 Details of operation

The sensor uses a global shutter, i.e. light exposure takes place on all pixels in parallel, although subsequent readout is sequential.

25.1.2.1 Free running mode

The sensor uses a rolling shutter, i.e. all pixels of a given line are exposed to light the same time or synchronous shutter, i.e. light exposure takes place on all pixels in parallel, although subsequent readout is sequential. Therefore the sensor offers two different modes of operation:

- free running mode (Overlapping exposure and readout)
- snapshot mode (Sequential exposure and readout)

25.1.2.2 Free running mode

In free running mode, the sensor reaches its maximum frame rate. This is done by overlapping erase, exposure and readout phase. The sensor timing in free running mode is fixed, so there is no control when to start an acquisition. This mode is used with trigger mode Continuous.

To calculate the maximum frames per second ($\text{FPS}_{\text{max}}$) in free running mode you will need following formula:

\[
\text{FPS}_{\text{max}} = \frac{\text{InternalLineLength} \times \text{VerticalBlankLines} \times \text{SensorInClock}}{\text{NumberOfLVDS}}
\]

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>InternalLineLength</td>
<td>2190</td>
</tr>
<tr>
<td>VerticalBlankLines</td>
<td>37</td>
</tr>
<tr>
<td>SensorInClock</td>
<td>72 MHz</td>
</tr>
<tr>
<td>NumberOfLVDS</td>
<td>4</td>
</tr>
</tbody>
</table>
If exposure time is lower than frame time:

\[
FPS_{\text{max}} = \frac{1}{\text{FrameTime}}
\]

If exposure time is greater than frame time:

\[
FPS_{\text{max}} = \frac{1}{\text{ExposureTime}}
\]

**Note**

The exposure time step width is limited to the sensor's row time of 30.42 \(\mu\)s and therefore
- auto exposure with very low exposure times will perform with relatively large increments and
- exposure mode = TriggerWidth (if available) will perform with a jitter corresponding to the row time

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Setting (GenICam (p.143))</th>
<th>Mode / Setting (Device Specific (p.143))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = Off&quot;</td>
<td>Continuous</td>
<td>Free running, no external trigger signal needed.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = Software&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnDemand</td>
<td>Image acquisition triggered by command (software trigger).</td>
</tr>
<tr>
<td>To trigger one frame execute the TriggerSoftware@i command then.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = \langle\text{desired Line}\rangle&quot; &quot;TriggerActivation = LevelLow&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnLowLevel</td>
<td>Start an exposure of a frame as long as the trigger input is below the trigger threshold. (No Frame←Trigger!)</td>
</tr>
</tbody>
</table>
"TriggerSelector = AcquisitionActive"
"TriggerMode = On"
"TriggerSource = <desired Line>">
"TriggerActivation = LevelHigh"
"ExposureMode = Timed"

OnHighLevel
Start an exposure of a frame as long as the trigger input is above the trigger threshold. (No Frame—Trigger!)

"TriggerSelector = FrameStart"
"TriggerMode = On"
"TriggerSource = <desired Line>">
"TriggerActivation = FallingEdge"
"ExposureMode = Timed"

OnFallingEdge
Each falling edge of trigger signal acquires one image.

"TriggerSelector = FrameStart"
"TriggerMode = On"
"TriggerSource = <desired Line>">
"TriggerActivation = RisingEdge"
"ExposureMode = Timed"

OnRisingEdge
Each rising edge of trigger signal acquires one image.

"TriggerSelector = FrameStart"
"TriggerMode = On"
"TriggerSource = <desired Line>">
"TriggerActivation = AnyEdge"
"ExposureMode = Timed"

OnAnyEdge
Start the exposure of a frame when the trigger input level changes from high to low or from low to high.

Line Mapping (TriggerSource mvIMPACT Acquire -> TriggerSource GenICam (valid values for <desired Line>):

<table>
<thead>
<tr>
<th>TriggerSource mvIMPACT Acquire</th>
<th>TriggerSource GenICam(BCX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP-IN0</td>
<td>Line4</td>
</tr>
<tr>
<td>GP-IN1</td>
<td>Line5</td>
</tr>
</tbody>
</table>

25.1.3 Sensor Data

Device Structure

- CMOS image sensor (Type 1)
- Number of effective pixels: 5544 (H) x 3692 (V)
- Unit cell size: 2.4um (H) x 2.4um (V)
25.1.3.2 Color version

25.1.3.3 Gray scale version

25.1.4 Device Feature And Property List

25.1.4.1 Gray scale version

25.1.4.2 Color version
26.1 mvBlueCOUGAR-X105p (5.1 Mpix [2464 x 2056])

26.1.1 Introduction

The sensor can acquire a four directional polarization image in one shot. Because the polarizer is under the on-chip lens layer

- reflectances are reduced and
- flare and ghost characteristics are avoided.

The pixel array of the sensor is as follows:

![Pixel array image]

Figure 1: Pixel array gray scale / color

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manufacturer</td>
<td>Sensor</td>
</tr>
<tr>
<td>Sensor name</td>
<td>IMX250_POL</td>
</tr>
<tr>
<td>Max. frames per second</td>
<td>80</td>
</tr>
<tr>
<td>Device Structure</td>
<td>CMOS image sensor</td>
</tr>
<tr>
<td>$\text{SNR}_{\text{max}}$ [dB]</td>
<td>tbd</td>
</tr>
<tr>
<td>DNR (normal / HDR) [dB]</td>
<td>tbd / -</td>
</tr>
<tr>
<td>Image size</td>
<td>2/3</td>
</tr>
<tr>
<td>Number of effective pixels</td>
<td>2464 (H) x 2056 (V)</td>
</tr>
<tr>
<td>Unit cell size</td>
<td>3.45μm (H) x 3.45μm (V)</td>
</tr>
<tr>
<td>ADC resolution / output</td>
<td>12 bit $\rightarrow$ 8/10/(12)</td>
</tr>
</tbody>
</table>

MATRIX VISION GmbH
1 Measured accord. to EMVA1288 with gray scale version of the camera

26.1.2 Spectral Sensitivity

![Spectral sensitivity graph](image)

**Figure 2:** Spectral sensitivity mvBlueCOUGAR-X105pG

26.1.3 Timings

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>InternalLineLength</td>
<td>835</td>
</tr>
<tr>
<td>VerticalBlankLines</td>
<td>37</td>
</tr>
<tr>
<td>SensorInClock</td>
<td>74.25 (@50 MHz Pixel clock)</td>
</tr>
<tr>
<td>NumberOfLVDS</td>
<td>4</td>
</tr>
</tbody>
</table>

26.1.3.1 Free running mode

In free running mode, the sensor reaches its maximum frame rate. This is done by overlapping erase, exposure and readout phase. The sensor timing in free running mode is fixed, so there is no control when to start an acquisition. This mode is used with trigger mode Continuous.

To calculate the maximum frames per second \( (\text{FPS}_{\text{max}}) \) in free running mode you will need following formula:

\[
\text{FrameTime} = \frac{\text{InternalLineLength}}{\text{SensorInClock}} \cdot \frac{\text{ImageHeight} + \text{VerticalBlankLines}}{1000}
\]

MATRIX VISION GmbH
If exposure time is lower than frame time:

\[
\text{FPS}_{\text{max}} = \frac{1}{\text{FrameTime}}
\]

If exposure time is greater than frame time:

\[
\text{FPS}_{\text{max}} = \frac{1}{\text{ExposureTime}}
\]

**Note**

The exposure time step width is limited to the sensor’s row time of 11.27 us and therefore
- auto exposure with very low exposure times will perform with relatively large increments and
- exposure mode = TriggerWidth (if available) will perform with a jitter corresponding to the row time.

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Setting (GenICam)</th>
<th>Mode / Setting (obsolete “Device Specific”)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = Off&quot;</td>
<td>Continuous</td>
<td>Free running, no external trigger signal needed.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = Software&quot; &quot;ExposureMode = Timed&quot;</td>
<td>On Demand</td>
<td>Image acquisition triggered by command (software trigger).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>To trigger one frame execute the TriggerSoftware@i command then.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelLow&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnLowLevel</td>
<td>Start an exposure of a frame as long as the trigger input is below the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelHigh&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnHighLevel</td>
<td>Start an exposure of a frame as long as the trigger input is above the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>TriggerSelector = FrameStart</td>
<td>OnFallingEdge</td>
<td>Each falling edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------</td>
<td>------------------------------------------------------</td>
</tr>
<tr>
<td>TriggerMode = On</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerSource = &lt;desired Line&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerActivation = FallingEdge</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ExposureMode = Timed</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TriggerSelector = FrameStart</th>
<th>OnRisingEdge</th>
<th>Each rising edge of trigger signal acquires one image.</th>
</tr>
</thead>
<tbody>
<tr>
<td>TriggerMode = On</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerSource = &lt;desired Line&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerActivation = RisingEdge</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ExposureMode = Timed</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TriggerSelector = FrameStart</th>
<th>OnAnyEdge</th>
<th>Start the exposure of a frame when the trigger input level changes from high to low or from low to high.</th>
</tr>
</thead>
<tbody>
<tr>
<td>TriggerMode = On</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerSource = &lt;desired Line&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TriggerActivation = AnyEdge</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ExposureMode = Timed</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

26.1.4 Device Feature And Property List

26.1.5 Gray scale version

26.1.6 Color version
27 Appendix A.5 CMOS specific camera / sensor data

27.1 mvBlueCOUGAR-Xx00w (0.4 Mpix [752 x 480])

27.1.1 Introduction

The CMOS sensor module (MT9V034) incorporates the following features:

- resolution to 752 x 480 gray scale or RGB Bayer mosaic
- supports window AOI mode with faster readout
- **high dynamic range** (p. 363) 110 dB
- programmable analog gain (0..12 dB)
- progressive scan sensor (no interlaced problems!)
- full frame shutter
- programmable readout timing with free capture windows and partial scan
- many trigger modes (free-running, hardware-triggered)

27.1.2 Details of operation

The sensor uses a full **frame shutter** (ShutterMode = "FrameShutter"), i.e. all pixels are reset at the same time and the exposure commences. It ends with the charge transfer of the voltage sampling. Furthermore, the sensor offers two different modes of operation:

- free running mode (Overlapping exposure and readout)
- snapshot mode (Sequential exposure and readout)

27.1.2.1 Free running mode

In free running mode, the sensor reaches its maximum frame rate. This is done by overlapping erase, exposure and readout phase. The sensor timing in free running mode is fixed, so there is no control when to start an acquisition. This mode is used with trigger mode **Continuous**.

To calculate the maximum frames per second (FPS\textsubscript{max}) in free running mode you will need following formula:

\[
\text{FrameTime} = (\text{ImageWidth} + 61) \times ((\text{ImageHeight} + 45) / \text{PixelClock})
\]

If exposure time is lower than frame time:

\[
\text{FPS}_{\text{max}} = \frac{1}{\text{FrameTime}}
\]

If exposure time is greater than frame time:

\[
\text{FPS}_{\text{max}} = \frac{1}{\text{ExposureTime}}
\]
27.1.2.2 Snapshot mode

In snapshot mode, the image acquisition process consists of several sequential phases:

27.1.2.2.1 Trigger

Snapshot mode starts with a trigger. This can be either a hardware or a software signal.

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Setting (GenICam (p.143))</th>
<th>Mode / Setting (Device Specific (p.143))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot;</td>
<td>Continuous</td>
<td>Free running, no external trigger signal needed.</td>
</tr>
<tr>
<td>&quot;TriggerMode = Off&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>OnDemand</td>
<td>Image acquisition triggered by command (software trigger).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot;</td>
<td>OnLowLevel</td>
<td>Start an exposure of a frame as long as the trigger input is below the trigger threshold. (No Frame→Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerMode = On&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSource = &lt;desired Line&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerActivation = LevelLow&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;ExposureMode = Timed&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>OnHighLevel</td>
<td>Start an exposure of a frame as long as the trigger input is above the trigger threshold. (No Frame→Trigger!)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot;</td>
<td>OnFallingEdge</td>
<td>Each falling edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>&quot;TriggerMode = On&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSource = &lt;desired Line&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerActivation = FallingEdge&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;ExposureMode = Timed&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>OnRisingEdge</td>
<td>Each rising edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerMode = On&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSource = &lt;desired Line&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerActivation = RisingEdge&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;ExposureMode = Timed&quot;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
"TriggerSelector = AcquisitionActive"
"TriggerMode = On"
"TriggerSource = <desired Line>"
"TriggerActivation = LevelHigh"
"ExposureMode = TriggerWidth"

OnHighExpose
Each rising edge of trigger signal acquires one image, exposure time corresponds to pulse width.

"TriggerSelector = AcquisitionActive"
"TriggerMode = On"
"TriggerSource = <desired Line>"
"TriggerActivation = LevelLow"
"ExposureMode = TriggerWidth"

OnLowExpose
Each falling edge of trigger signal acquires one image, exposure time corresponds to pulse width.

"TriggerSelector = FrameStart"
"TriggerMode = On"
"TriggerSource = <desired Line>"
"TriggerActivation = AnyEdge"
"ExposureMode = Timed"

OnAnyEdge
Start the exposure of a frame when the trigger input level changes from high to low or from low to high.

Line Mapping (TriggerSource mvIMPACT Acquire -> TriggerSource GenICam (valid values for <desired Line>)):  

<table>
<thead>
<tr>
<th>TriggerSource mvIMPACT Acquire</th>
<th>TriggerSource GenICam(BCX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP-IN0</td>
<td>Line4</td>
</tr>
<tr>
<td>GP-IN1</td>
<td>Line5</td>
</tr>
</tbody>
</table>

27.1.2.2.2 Erase, exposure and readout

All pixels are light sensitive at the same period of time. The whole pixel core is reset simultaneously and after the exposure time all pixel values are sampled together on the storage node inside each pixel. The pixel core is read out line-by-line after exposure.

Note
Exposure and read out cycle is carry-out in serial; that causes that no exposure is possible during read out.

The step width for the exposure time is 1 us.

Image data is then shifted out line-by-line and transferred to memory.

To calculate the maximum frames per second \( (FPS_{max}) \) in snapshot mode you will need following formula:

\[
FrameTime = (ImageWidth + 61) \times ((ImageHeight + 45) / PixelClock)
\]

\[
FPS_{max} = \frac{1}{FrameTime + ExposureTime}
\]
27.1.3 Sensor Data

Device Structure

- Progressive scan CMOS image sensor
- Image size: 4.51(H)×2.88(V)mm (Type 1/3")
- Number of effective pixels: 752 (H) × 480 (V)
- Unit cell size: 6um (H) × 6um (V)

27.1.3.1 Characteristics

27.1.3.2 Color version

![Color version graph]

27.1.3.3 Gray scale version

![Gray scale version graph]
27.2 mvBlueCOUGAR-Xx02b (1.2 Mpix [1280 x 960])

27.2.1 Introduction

The CMOS sensor module (MT9M021) incorporates the following features:

- resolution to 1280 x 960 gray scale or RGB Bayer mosaic
- supports window AOI mode with faster readout
- programmable analog gain (0..12 dB)
- progressive scan sensor (no interlaced problems!)
- pipelined global shutter
- programmable readout timing with free capture windows and partial scan
- many trigger modes (free-running, hardware-triggered)

27.2.2 Details of operation

The sensor uses a pipelined global snapshot shutter, i.e. light exposure takes place on all pixels in parallel, although subsequent readout is sequential. Therefore the sensor offers two different modes of operation:

- free running mode (Overlapping exposure and readout)
- snapshot mode (Sequential exposure and readout)

27.2.2.1 Free running mode

In free running mode, the sensor reaches its maximum frame rate. This is done by overlapping erase, exposure and readout phase. The sensor timing in free running mode is fixed, so there is no control when to start an acquisition. This mode is used with trigger mode **Continuous**.

To calculate the maximum frames per second ($FPS_{max}$) in free running mode you will need following formula:

$$FrameTime = (ImageHeight \times \frac{1650}{PixelClock}) + (25 \times \frac{1650}{PixelClock})$$

If exposure time is lower than frame time:

$$FPS_{max} = \frac{1}{FrameTime}$$

If exposure time is greater than frame time:

$$FPS_{max} = \frac{1}{ExposureTime}$$
### 27.2.2.2 Snapshot mode

In snapshot mode, the image acquisition process consists of several sequential phases:

#### 27.2.2.2.1 Trigger

Snapshot mode starts with a trigger. This can be either a hardware or a software signal.

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Setting (GenICam (p.143))</th>
<th>Mode / Setting (Device Specific (p.143))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = Off&quot;</td>
<td>Continuous</td>
<td>Free running, no external trigger signal needed.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = Software&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnDemand</td>
<td>Image acquisition triggered by command (software trigger).</td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelLow&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnLowLevel</td>
<td>Start an exposure of a frame as long as the trigger input is below the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelHigh&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnHighLevel</td>
<td>Start an exposure of a frame as long as the trigger input is above the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = FallingEdge&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnFallingEdge</td>
<td>Each falling edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = RisingEdge&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnRisingEdge</td>
<td>Each rising edge of trigger signal acquires one image.</td>
</tr>
</tbody>
</table>
"TriggerSelector = FrameStart"
"TriggerMode = On"
"TriggerSource = <desired Line>"
"TriggerActivation = AnyEdge"
"ExposureMode = Timed"

<table>
<thead>
<tr>
<th>OnAnyEdge</th>
<th>Start the exposure of a frame when the trigger input level changes from high to low or from low to high.</th>
</tr>
</thead>
</table>

Line Mapping (\texttt{TriggerSource mvIMPACT Acquire -> TriggerSource GenICam} (valid values for \texttt{<desired Line>})):

<table>
<thead>
<tr>
<th>TriggerSource mvIMPACT Acquire</th>
<th>TriggerSource GenICam(BCX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP-IN0</td>
<td>Line4</td>
</tr>
<tr>
<td>GP-IN1</td>
<td>Line5</td>
</tr>
</tbody>
</table>

27.2.2.2 Erase, exposure and readout

All pixels are light sensitive at the same period of time. The whole pixel core is reset simultaneously and after the exposure time all pixel values are sampled together on the storage node inside each pixel. The pixel core is read out line-by-line after exposure.

\textbf{Note}

Exposure and read out cycle is carry-out in serial; that causes that no exposure is possible during read out.

The step width for the exposure time is 1 us.

Image data is then shifted out line-by-line and transferred to memory.

To calculate the maximum frames per second (\(FPS_{\text{max}}\)) in snapshot mode you will need following formula:

\[
FrameTime = (ImageHeight \times (1650 / \text{PixelClock})) + (25 \times (1650 / \text{PixelClock}))
\]

\[
FPS_{\text{max}} = \frac{1}{FrameTime + ExposureTime}
\]

27.2.3 Sensor Data

\textbf{Device Structure}

- CMOS image sensor (Type 1/3“)
- Number of effective pixels: 1280 (H) x 960 (V)
- Unit cell size: 3.75um (H) x 3.75um (V)
27.2.3.1 Characteristics

27.2.3.2 Color version

![Graph showing quantum efficiency vs. wavelength for different colors and filters.]

27.2.3.3 Gray scale version

![Graph showing quantum efficiency vs. wavelength for different filters.]

27.2.4 Device Feature And Property List

27.2.4.1 Gray scale version

27.2.4.2 Color version
27.3 mvBlueCOUGAR-Xx02d (1.2 Mpix [1280 x 960])

27.3.1 Introduction

The CMOS sensor module (MT9M024) incorporates the following features:

- resolution to 1280 x 960 gray scale or RGB Bayer mosaic
- supports window AOI mode with faster readout
- programmable analog gain (0..12 dB)
- progressive scan sensor (no interlaced problems!)
- high dynamic range (p. [366]) 115 dB (with gray scale version)
- rolling shutter
- programmable readout timing with free capture windows and partial scan
- many trigger modes (free-running, hardware-triggered)

27.3.2 Details of operation

The sensor uses following acquisition mode:

- rolling shutter With the rolling shutter the lines are exposed for the same duration, but at a slightly different point in time.

**Note**

Moving objects together with a rolling shutter can cause a shear in moving objects.

Furthermore, the sensor offers following operating modes:

- free running mode (Overlapping exposure and readout)
- snapshot mode (Sequential exposure and readout)

27.3.2.1 Free running mode

In free running mode, the sensor reaches its maximum frame rate. This is done by overlapping erase, exposure and readout phase. The sensor timing in free running mode is fixed, so there is no control when to start an acquisition. This mode is used with trigger mode Continuous.

To calculate the maximum frames per second \( (\text{FPS}_{\text{max}}) \) in free running mode you will need following formula:

\[
\text{FrameTime} = (\text{ImageHeight} \times (1650 / \text{PixelClock})) + (25 \times (1650 / \text{PixelClock}))
\]

If exposure time is lower than frame time:

\[
\text{FPS}_{\text{max}} = \frac{1}{\text{FrameTime}}
\]

If exposure time is greater than frame time:

\[
\text{FPS}_{\text{max}} = \frac{1}{\text{ExposureTime}}
\]
27.3.2.2 Snapshot mode

In snapshot mode, the image acquisition process consists of several sequential phases:

27.3.2.2.1 Trigger

Snapshot mode starts with a trigger. This can be either a hardware or a software signal.

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Setting (GenICam (p.143))</th>
<th>Mode / Setting (Device Specific (p.143))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = Off&quot;</td>
<td>Continuous</td>
<td>Free running, no external trigger signal needed.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = Software&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnDemand</td>
<td>Image acquisition triggered by command (software trigger).</td>
</tr>
<tr>
<td>To trigger one frame execute the TriggerSoftware@i command then.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelLow&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnLowLevel</td>
<td>Start an exposure of a frame as long as the trigger input is below the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelHigh&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnHighLevel</td>
<td>Start an exposure of a frame as long as the trigger input is above the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = FallingEdge&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnFallingEdge</td>
<td>Each falling edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = RisingEdge&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnRisingEdge</td>
<td>Each rising edge of trigger signal acquires one image.</td>
</tr>
</tbody>
</table>
"TriggerSelector = FrameStart"
"TriggerMode = On"
"TriggerSource = <desired Line>"
"TriggerActivation = AnyEdge"
"ExposureMode = Timed"

OnAnyEdge

Start the exposure of a frame when the trigger input level changes from high to low or from low to high.

Line Mapping (TriggerSource mvIMPACT Acquire -> TriggerSource GenICam (valid values for <desired Line>)):

<table>
<thead>
<tr>
<th>TriggerSource mvIMPACT Acquire</th>
<th>TriggerSource GenICam(BCX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP-IN0</td>
<td>Line4</td>
</tr>
<tr>
<td>GP-IN1</td>
<td>Line5</td>
</tr>
</tbody>
</table>

27.3.2.2.2 Erase, exposure and readout

All pixels are light sensitive at the same period of time. The whole pixel core is reset simultaneously and after the exposure time all pixel values are sampled together on the storage node inside each pixel. The pixel core is read out line-by-line after exposure.

**Note**

Exposure and read out cycle is carry-out in serial; that causes that no exposure is possible during read out.

The step width for the exposure time is 1 us.

Image data is then shifted out line-by-line and transferred to memory.

To calculate the maximum frames per second (FPS\_max) in snapshot mode you will need following formula:

\[
\text{FrameTime} = (\text{ImageHeight} \times (1650 / \text{PixelClock})) + (25 \times (1650 / \text{PixelClock}))
\]

\[
\text{FPS}_{\text{max}} = \frac{1}{\text{FrameTime} + \text{ExposureTime}}
\]

27.3 Sensor Data

**Device Structure**

- CMOS image sensor (Type 1/3")
- Number of effective pixels: 1280 (H) x 960 (V)
- Unit cell size: 3.75um (H) x 3.75um (V)
27.3.3.1 Characteristics

27.3.3.2 Color version

27.3.3.3 Gray scale version

27.3.4 Device Feature And Property List

27.3.4.1 Gray scale version

27.3.4.2 Color version
27.4.1 Introduction

The CMOS sensor module (EV76C560) incorporates the following features:

- resolution to 1280 x 1024 gray scale or RGB Bayer mosaic
- supports window AOI mode with faster readout
- high dynamic range (p. 369)
- pipelined global snapshot shutter
- Line scan mode (p. 643)
  - programmable readout timing with free capture windows and partial scan
  - many trigger modes (free-running, hardware-triggered)

27.4.2 Details of operation

The sensor uses a rolling shutter, i.e. all pixels of a given line are exposed to light the same time or synchronous shutter, i.e. light exposure takes place on all pixels in parallel, although subsequent readout is sequential.

Therefore the sensor offers two different modes of operation:

- free running mode (Overlapping exposure and readout)
- snapshot mode (Sequential exposure and readout)

27.4.2.1 Free running mode

In free running mode, the sensor reaches its maximum frame rate. This is done by overlapping erase, exposure and readout phase. The sensor timing in free running mode is fixed, so there is no control when to start an acquisition. This mode is used with trigger mode Continuous.

To calculate the maximum frames per second (FPS_{max}) in free running mode you will need following formula:

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>InternalLineLength</td>
<td>944</td>
</tr>
<tr>
<td>InternalADCClock us</td>
<td>120 MHz</td>
</tr>
</tbody>
</table>

\[
\text{FrameTime (ms)} = \frac{2 \times (\text{ImageHeight} + 10)}{\text{InternalLineLength} \times \frac{\text{InternalADCClock}}{1000}}
\]

If exposure time is lower than frame time:

\[
\text{FPS}_{\max} = \frac{1000}{\text{FrameTime}}
\]

If exposure time is greater than frame time:

\[
\text{FPS}_{\max} = \frac{1}{\text{ExposureTime}}
\]
27.4.2.2 Snapshot mode

In snapshot mode, the image acquisition process consists of several sequential phases:

27.4.2.2.1 Trigger

Snapshot mode starts with a trigger. This can be either a hardware or a software signal.

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Setting (GenICam) (p.143)</th>
<th>Mode / Setting (Device Specific)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = Off&quot;</td>
<td>Continuous</td>
<td>Free running, no external trigger signal needed.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = Software&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnDemand</td>
<td>Image acquisition triggered by command (software trigger).</td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelLow&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnLowLevel</td>
<td>Start an exposure of a frame as long as the trigger input is below the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelHigh&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnHighLevel</td>
<td>Start an exposure of a frame as long as the trigger input is above the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = FallingEdge&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnFallingEdge</td>
<td>Each falling edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = RisingEdge&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnRisingEdge</td>
<td>Each rising edge of trigger signal acquires one image.</td>
</tr>
</tbody>
</table>
"TriggerSelector = FrameStart"
"TriggerMode = On"
"TriggerSource = <desired Line>"
"TriggerActivation = AnyEdge"
"ExposureMode = Timed"

<table>
<thead>
<tr>
<th>TriggerSource mvIMpACT Acquire</th>
<th>TriggerSource GenICam (valid values for &lt;desired Line&gt;)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP-IN0</td>
<td>Line4</td>
</tr>
<tr>
<td>GP-IN1</td>
<td>Line5</td>
</tr>
</tbody>
</table>

27.4.2.2 Erase, exposure and readout

All pixels are light sensitive at the same period of time. The whole pixel core is reset simultaneously and after the exposure time all pixel values are sampled together on the storage node inside each pixel. The pixel core is read out line-by-line after exposure.

**Note**

Exposure and read out cycle is carry-out in serial; that causes that no exposure is possible during read out.

The step width for the exposure time is 1 us.

Image data is then shifted out line-by-line and transferred to memory.

To calculate the maximum frames per second (FPS\textsubscript{max}) in snapshot mode you will need following formula:

\[
\text{FrameTime} = \frac{2 \cdot \text{ImageHeight} + 10}{\text{InternalLineLength} \cdot \frac{\text{InternalADCClock}}{1000}}
\]

\[
\text{FPS}_{\text{max}} = \frac{1}{\text{FrameTime} + \text{ExposureTime}}
\]

27.4.2.3 Line scan mode

The sensor offers also a usage as a line scan sensor. One (gray scale sensor) or two lines (in terms of color sensor) can be selected to be read out of the full line height of 1024 lines. This or these lines are grouped to a pseudo frame of selectable height in the internal buffer of the camera.

The camera then outputs these frames which contain multiples of the same scan line(s) without gaps or interruptions.

To operate in line scan mode, use the following properties:

1. In "Setting -> Base -> Camera -> GenICam -> Device Control", please change the "Device Scan Type" to "line scan".

2. In "Setting -> Base -> Camera -> GenICam -> Image Format Control", please set "Width" and "Height". "Height" specifies how often the same line(s) will be used to reach the height of the image. Use "Offset X" to shift horizontally the starting point of the exposed line. Use "Offset Y" to shift the scan line vertically (typically to 512 which represents the optical center).
Note

The sensor will not get faster by windowing in x direction.

3. Finally, in "Setting -> Base -> Camera -> GenICam -> Acquisition Control", please adapt the "Exposure Time". For high line scan rates the exposure time has to be proportionally low. To achieve the maximum line scan rate of 12.6 kHz (gray scale sensor with rolling shutter) or 10.6 kHz (gray scale sensor with global shutter) the exposure time has to be set to lower than 62us.

You may use longer exposure times at the expense of scanning frequency.

Note

Using more than one line e.g. 20, is like having an area scan with ImageHeight = 20. In line scan mode, the min. ImageHeight is at least 16.
wxPropView settings for line scan usage
Note

You can use either "Continuous" or a trigger mode as "Acquisition Mode" in "Setting -> Base -> Camera -> GenICam -> Acquisition Control". However, if an external (line) trigger will have to be used, it is absolutely required that the external trigger is always present. During a trigger interruption, controlling or communication to the camera is not possible.

27.4.3 Sensor Data

Device Structure

- CMOS image sensor (Type 1/1.8"
- Number of effective pixels: 1280 (H) x 1024 (V)
- Unit cell size: 5.3um (H) x 5.3um (V)

27.4.3.1 Characteristics

27.4.4 Device Feature And Property List

27.4.4.1 Gray scale version

27.4.4.2 Color version
27.5 mvBlueCOUGAR-Xx02eGE (1.3 Mpix [1280 x 1024])

27.5.1 Introduction

The CMOS sensor module (EV76C661) incorporates the following features:

- resolution to 1280 x 1024 gray scale or RGB Bayer mosaic
- supports window AOI mode with faster readout
- **high dynamic range** (p. 369)
- enhanced infrared sensitivity
- global snapshot shutter
- **Line scan mode** (p. 651)
  - programmable readout timing with free capture windows and partial scan
  - many trigger modes (free-running, hardware-triggered)

27.5.2 Details of operation

The sensor uses two acquisition modes:

- **rolling shutter** (mvShutterMode = "RollingShutter") and
- **global reset release shutter** (mvShutterMode = "mvGlobalShutter").

With the **rolling shutter** the lines are exposed for the same duration, but at a slightly different point in time:
Note

Moving objects together with a rolling shutter can cause a shear in moving objects.

The **global reset release shutter**, which is only available in triggered operation, starts the exposure of all rows simultaneously and the reset to each row is released simultaneously, too. However, the readout of the lines is equal to the readout of the rolling shutter: line by line:

![Diagram](https://via.placeholder.com/150)

Note

This means, the bottom lines of the sensor will be exposed to light longer! For this reason, this mode will only make sense, if there is no extraneous light and the flash duration is shorter or equal to the exposure time.

Furthermore, the sensor offers two operating modes:

- free running mode (Overlapping exposure and readout)
- snapshot mode (Sequential exposure and readout) in triggered operation

### 27.5.2.1 Free running mode

In free running mode, the sensor reaches its maximum frame rate. This is done by overlapping erase, exposure and readout phase. The sensor timing in free running mode is fixed, so there is no control when to start an acquisition. This mode is used with trigger mode **Continuous**.

To calculate the maximum frames per second (FPS\(_{\text{max}}\)) in free running mode you will need following formula:

\[
\text{FrameTime} = \frac{\text{InternalLineLength} \times \text{ImageHeight} + 10}{\text{InternalADCClock} \times 1000}
\]

If exposure time is lower than frame time:

\[
\text{FPS}_{\text{max}} = \frac{1}{\text{FrameTime}}
\]
If exposure time is greater than frame time:

\[
\text{FPS\_max} = \frac{1}{\text{ExposureTime}}
\]

### 27.5.2.2 Snapshot mode

In snapshot mode, the image acquisition process consists of several sequential phases:

#### 27.5.2.2.1 Trigger

Snapshot mode starts with a trigger. This can be either a hardware or a software signal.

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Setting (GenICam (p. 143))</th>
<th>Mode / Setting (Device Specific (p. 143))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = Off&quot;</td>
<td>Continuous</td>
<td>Free running, no external trigger signal needed.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = Software&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnDemand</td>
<td>Image acquisition triggered by command (software trigger).</td>
</tr>
<tr>
<td>To trigger one frame execute the TriggerSoftware@i command then.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelLow&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnLowLevel</td>
<td>Start an exposure of a frame as long as the trigger input is below the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelHigh&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnHighLevel</td>
<td>Start an exposure of a frame as long as the trigger input is above the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = FallingEdge&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnFallingEdge</td>
<td>Each falling edge of trigger signal acquires one image.</td>
</tr>
</tbody>
</table>
"TriggerSelector = FrameStart"
"TriggerMode = On"
"TriggerSource = <desired Line>"
"TriggerActivation = RisingEdge"
"ExposureMode = Timed"

OnRisingEdge
Each rising edge of trigger signal acquires one image.

"TriggerSelector = FrameStart"
"TriggerMode = On"
"TriggerSource = <desired Line>"
"TriggerActivation = AnyEdge"
"ExposureMode = Timed"

OnAnyEdge
Start the exposure of a frame when the trigger input level changes from high to low or from low to high.

Line Mapping (TriggerSource mvIMPACT Acquire -> TriggerSource GenICam (valid values for <desired Line>>):

<table>
<thead>
<tr>
<th>TriggerSource mvIMPACT Acquire</th>
<th>TriggerSource GenICam(BCX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP-IN0</td>
<td>Line4</td>
</tr>
<tr>
<td>GP-IN1</td>
<td>Line5</td>
</tr>
</tbody>
</table>

27.5.2.2.2 Erase, exposure and readout

All pixels are light sensitive at the same period of time. The whole pixel core is reset simultaneously and after the exposure time all pixel values are sampled together on the storage node inside each pixel. The pixel core is read out line-by-line after exposure.

Note

Exposure and read out cycle is carry-out in serial; that causes that no exposure is possible during read out.

The step width for the exposure time is 1 us.

Image data is then shifted out line-by-line and transferred to memory.

To calculate the maximum frames per second (FPS<sub>max</sub>) in snapshot mode you will need following formula:

\[
\frac{2 \times \text{ImageHeight} + 10}{\frac{\text{InternalLineLength}}{\text{InternalADCClock}} \times \frac{1}{1000}}
\]

\[
\text{FPS}_{\text{max}} = \frac{1}{\text{FrameTime} + \text{ExposureTime}}
\]
27.5.2.3 Line scan mode

The sensor offers also a usage as a line scan sensor. One (gray scale sensor) or two lines (in terms of color sensor) can be selected to be read out of the full line height of 1024 lines. This or these lines are grouped to a pseudo frame of selectable height in the internal buffer of the camera.

The camera then outputs these frames which contain multiples of the same scan line(s) without gaps or interruptions.

To operate in line scan mode, use the following properties:

1. In "Setting -> Base -> Camera -> GenICam -> Device Control", please change the "Device Scan Type" to "line scan".

2. In "Setting -> Base -> Camera -> GenICam -> Image Format Control", please set "Width" and "Height". "Height" specifies how often the same line(s) will be used to reach the height of the image. Use "Offset X" to shift horizontally the starting point of the exposed line. Use and "Offset Y" to shift the scan line vertically.

   **Note**

   The sensor will not get faster by windowing in x direction.

3. Finally, in "Setting -> Base -> Camera -> GenICam -> Acquisition Control", please adapt the "Exposure Time". For high line scan rates the exposure time has to be proportionally low. To achieve the maximum line scan rate of 12.6 kHz (gray scale sensor with rolling shutter) or 10.6 kHz (gray scale sensor with global shutter) the exposure time has to be set to lower than 62us.

   You may use longer exposure times at the expense of scanning frequency.
Note

Using more than one line e.g. 20, is like having an area scan with ImageHeight = 20. In line scan mode, the min. ImageHeight is at least 16.
wxPropView settings for line scan usage
You can use either "Continuous" or a trigger mode as "Acquisition Mode" in "Setting -> Base -> Camera -> GenICam -> Acquisition Control". However, if an external (line) trigger will have to be used, it is absolutely required that the external trigger is always present. During a trigger interruption, controlling or communication to the camera is not possible.

27.5.3 Sensor Data

Device Structure

- CMOS image sensor (Type 1/1.8")
- Number of effective pixels: 1280 (H) x 1024 (V)
- Unit cell size: 5.3um (H) x 5.3um (V)

27.5.3.1 Characteristics

27.5.4 Device Feature And Property List

27.5.4.1 Gray scale version
27.6.1 Introduction

The CMOS sensor module (EV76C570) incorporates the following features:

- resolution to 1600 x 1200 gray scale or RGB Bayer mosaic
- supports window AOI mode with faster readout
- **high dynamic range** (*p. 363*)
- pipelined global snapshot shutter
- **Line scan mode** (*p. 657*)
  - programmable readout timing with free capture windows and partial scan
  - many trigger modes (free-running, hardware-triggered)

27.6.2 Details of operation

The sensor uses a pipelined global snapshot shutter, i.e. light exposure takes place on all pixels in parallel, although subsequent readout is sequential. Therefore the sensor offers two different modes of operation:

- free running mode (Overlapping exposure and readout)
- snapshot mode (Sequential exposure and readout)

27.6.2.1 Free running mode

In free running mode, the sensor reaches its maximum frame rate. This is done by overlapping erase, exposure and readout phase. The sensor timing in free running mode is fixed, so there is no control when to start an acquisition. This mode is used with trigger mode **Continuous**.

To calculate the maximum frames per second (FPS\textsubscript{max}) in free running mode you will need following formula:

\[
\text{FrameTime} = \frac{\text{InternalLineLength}}{\text{InternalADCClock}} \cdot \frac{\text{ImageHeight} + 10}{1000}
\]

\[
\text{FPS}_{\text{max}} = \frac{1}{\text{FrameTime}}
\]

If exposure time is lower than frame time:

\[
\text{FPS}_{\text{max}} = \frac{1}{\text{ExposureTime}}
\]
27.6.2.2 Snapshot mode

In snapshot mode, the image acquisition process consists of several sequential phases:

"Trigger"

Snapshot mode starts with a trigger. This can be either a hardware or a software signal.

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Setting (GenICam (p.143))</th>
<th>Mode / Setting (Device Specific (p.143))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = Off&quot;</td>
<td>Continuous</td>
<td>Free running, no external trigger signal needed.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = Software&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnDemand</td>
<td>Image acquisition triggered by command (software trigger).</td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelLow&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnLowLevel</td>
<td>Start an exposure of a frame as long as the trigger input is below the trigger threshold. (No Frame---Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelHigh&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnHighLevel</td>
<td>Start an exposure of a frame as long as the trigger input is above the trigger threshold. (No Frame---Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = FallingEdge&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnFallingEdge</td>
<td>Each falling edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = RisingEdge&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnRisingEdge</td>
<td>Each rising edge of trigger signal acquires one image.</td>
</tr>
</tbody>
</table>
"TriggerSelector = FrameStart"
"TriggerMode = On"
"TriggerSource = <desired Line>">
"TriggerActivation = AnyEdge"
"ExposureMode = Timed"

OnAnyEdge

Start the exposure of a frame when the trigger input level changes from high to low or from low to high.

Line Mapping (TriggerSource mvImpact Acquire -> TriggerSource GenICam (valid values for <desired Line>)):

<table>
<thead>
<tr>
<th>TriggerSource mvImpact Acquire</th>
<th>TriggerSource GenICam(BCX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP-IN0</td>
<td>Line4</td>
</tr>
<tr>
<td>GP-IN1</td>
<td>Line5</td>
</tr>
</tbody>
</table>

"Erase, exposure and readout"

All pixels are light sensitive at the same period of time. The whole pixel core is reset simultaneously and after the exposure time all pixel values are sampled together on the storage node inside each pixel. The pixel core is read out line-by-line after exposure.

**Note**

Exposure and readout cycle is carry-out in serial; that causes that no exposure is possible during read out.

The step width for the exposure time is 1 us.

Image data is then shifted out line-by-line and transferred to memory.

To calculate the maximum frames per second (FPS max) in snapshot mode you will need following formula:

\[
\frac{2 \times \text{ImageHeight} + 10}{\frac{\text{InternalLineLength}}{\text{InternalADCClock}}} \times \frac{1}{\text{FrameTime} + \text{ExposureTime}}
\]

27.6.2.3 Line scan mode

The sensor offers also a usage as a line scan sensor. One (gray scale sensor) or two lines (in terms of color sensor) can be selected to be read out of the full line height of 1024 lines. This or these lines are grouped to a pseudo frame of selectable height in the internal buffer of the camera.

The camera then outputs these frames which contain multiples of the same scan line(s) without gaps or interruptions.

To operate in line scan mode, use the following properties:

1. In "Setting -> Base -> Camera -> GenICam -> Device Control", please change the "Device Scan Type" to "line scan".
2. In "Setting -> Base -> Camera -> GenICam -> Image Format Control", please set "Width" and "Height". "Height" specifies how often the same line(s) will be used to reach the height of the image. Use "Offset X" to shift horizontally the starting point of the exposed line. Use "Offset Y" to shift the scan line vertically (typically to 512 which represents the optical center).
3. Finally, in “Setting -> Base -> Camera -> GenICam -> Acquisition Control”, please adapt the “Exposure Time”. For high line scan rates the exposure time has to be proportionally low. To achieve the maximum line scan rate of 12.6 kHz (gray scale sensor) the exposure time has to be set to approx. 80us.

You may use longer exposure times at the expense of scanning frequency.
wxPropView settings for line scan usage (screenshot of -x02e but the properties are similar to -x04e)
Note

You can use either "Continuous" or a trigger mode as "Acquisition Mode" in "Setting -> Base -> Camera -> GenICam -> Acquisition Control". However, if an external (line) trigger will have to be used, it is absolutely required that the external trigger is always present. During a trigger interruption, controlling or communication to the camera is not possible.

27.6.3 Sensor Data

Device Structure

- CMOS image sensor (Type 1/1.8")
- Number of effective pixels: 1600 (H) x 1200 (V)
- Unit cell size: 4.5um (H) x 4.5um (V)

27.6.3.1 Characteristics

27.6.4 Device Feature And Property List

27.6.4.1 Gray scale version

27.6.4.2 Color version
27.7 mvBlueCOUGAR-X[D]x04 (2.2 Mpix [2048 x 1088])

27.7.1 Introduction

The CMOS sensor module (CMV2000) incorporates the following features:

- resolution to 2048 x 1088 gray scale or RGB Bayer mosaic
- supports window AOI mode with faster readout
- programmable analog gain (0..18 dB)
- progressive scan sensor (no interlaced problems!)
- pipelined global shutter
- programmable readout timing with free capture windows and partial scan
- many trigger modes (free-running, hardware-triggered)

27.7.2 Details of operation

The sensor uses a pipelined global snapshot shutter, i.e. light exposure takes place on all pixels in parallel, although subsequent readout is sequential. Therefore the sensor offers two different modes of operation:

- free running mode (Overlapping exposure and readout)
- snapshot mode (Sequential exposure and readout)

27.7.2.1 Free running mode

In free running mode, the sensor reaches its maximum frame rate. This is done by overlapping erase, exposure and readout phase. The sensor timing in free running mode is fixed, so there is no control when to start an acquisition. This mode is used with trigger mode Continuous.

To calculate the maximum frames per second (FPS_{max}) in free running mode you will need following formula (in us):

\[
\text{FrameOverheadTime} = \frac{129}{\text{PixelClock}} \times \left( 10 + \frac{32}{\text{LVDSOutputs}} \right)
\]

\[
\text{ReadoutTime} = \frac{129}{\text{PixelClock}} \times \frac{16 \times \text{ImageHeight}}{\text{LVDSOutputs}}
\]

\[
\text{FrameTime} = \text{FrameOverheadTime} + \text{ReadoutTime}
\]

If exposure time is lower than frame time:
FPS\_max = \frac{1000000}{\text{FrameTime}}

If exposure time is greater than frame time:

FPS\_max = \frac{1000000}{\text{ExposureTime} + \text{FrameOverheadTime}}

### 27.7.2.2 Snapshot mode

In snapshot mode, the image acquisition process consists of several sequential phases:

#### 27.7.2.2.1 Trigger

Snapshot mode starts with a trigger. This can be either a hardware or a software signal.

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Setting (GenICam (p.143))</th>
<th>Mode / Setting (Device Specific (p.143))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = Off&quot;</td>
<td>Continuous</td>
<td>Free running, no external trigger signal needed.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = Software&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnDemand</td>
<td>Image acquisition triggered by command (software trigger).</td>
</tr>
</tbody>
</table>

To trigger one frame execute the `TriggerSoftware@i` command then.

| "TriggerSelector = FrameStart" "TriggerMode = On" "TriggerSource = \langle\text{desired Line}\rangle" "TriggerActivation = FallingEdge" "ExposureMode = Timed" | OnFallingEdge | Each falling edge of trigger signal acquires one image. |
| "TriggerSelector = FrameStart" "TriggerMode = On" "TriggerSource = \langle\text{desired Line}\rangle" "TriggerActivation = RisingEdge" "ExposureMode = Timed" | OnRisingEdge | Each rising edge of trigger signal acquires one image. |

Line Mapping (`TriggerSource` mvIMPACT Acquire -> `TriggerSource` GenICam (valid values for `\langle\text{desired Line}\rangle`):
27.7.2.2 Erase, exposure and readout

All pixels are light sensitive at the same period of time. The whole pixel core is reset simultaneously and after the exposure time all pixel values are sampled together on the storage node inside each pixel. The pixel core is read out line-by-line after exposure.

*Note*

Exposure and read out cycle is carry-out in serial; that causes that no exposure is possible during read out.

The step width for the exposure time is 1 us.

Image data is then shifted out line-by-line and transferred to memory.

To calculate the maximum frames per second (FPS\_max) in snapshot mode you will need following formula (in us):

\[
\text{FrameTime} = \frac{129}{\text{PixelClock}} + \frac{32 \times (16 \times \text{ImageHeight})}{1000000}
\]

\[
\text{FPS\_max} = \frac{1000000}{\text{FrameTime} + \text{ExposureTime}}
\]

27.7.3 Sensor Data

**Device Structure**

- CMOS image sensor (Type 2/3")
- Unit cell size: 5.5um (H) x 5.5um (V)

27.7.3.1 Characteristics

27.7.3.2 Color version
27.7.3.3 Gray scale version

27.7.3.4 Gray scale version (-XD104a12)

27.7.4 Device Feature And Property List

27.7.4.1 Gray scale version -XD

27.7.4.2 Color version -XD

27.7.4.3 Gray scale version -X

27.7.4.4 Color version -X
27.8.1 Introduction

The CMOS sensor module (CMV4000) incorporates the following features:

- resolution to 2048 x 2048 gray scale or RGB Bayer mosaic
- supports window AOI mode with faster readout
- programmable analog gain (0..18 dB)
- progressive scan sensor (no interlaced problems!)
- pipelined global shutter
- programmable readout timing with free capture windows and partial scan
- many trigger modes (free-running, hardware-triggered)

27.8.2 Details of operation

The sensor uses a pipelined global snapshot shutter, i.e. light exposure takes place on all pixels in parallel, although subsequent readout is sequential.

Therefore the sensor offers two different modes of operation:

- free running mode (Overlapping exposure and readout)
- snapshot mode (Sequential exposure and readout)

27.8.2.1 Free running mode

In free running mode, the sensor reaches its maximum frame rate. This is done by overlapping erase, exposure and readout phase. The sensor timing in free running mode is fixed, so there is no control when to start an acquisition.

This mode is used with trigger mode Continuous.

To calculate the maximum frames per second (FPS\text{max}) in free running mode you will need following formula (in us):

\[
\text{FrameOverheadTime} = \frac{129}{\text{PixelClock}} \times \left( \frac{32}{\text{LVDSOutputs}} + \frac{20}{\text{LVDSOutputs}} \right)
\]

\[
\text{ReadoutTime} = \frac{129}{\text{PixelClock}} \times \frac{16 \times \text{ImageHeight}}{\text{LVDSOutputs}}
\]

\[
\text{FrameTime} = \text{FrameOverheadTime} + \text{ReadoutTime}
\]

If exposure time is lower than frame time:
FPS_max = \frac{1000000}{\text{FrameTime}}

If exposure time is greater than frame time:

\[
FPS_{\text{max}} = \frac{1000000}{\text{ExposureTime} + \text{FrameOverheadTime}}
\]

### 27.8.2.2 Snapshot mode

In snapshot mode, the image acquisition process consists of several sequential phases:

#### 27.8.2.2.1 Trigger

Snapshot mode starts with a trigger. This can be either a hardware or a software signal.

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Setting (GenICam (p.143))</th>
<th>Mode / Setting (Device Specific (p.143))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = Off&quot;</td>
<td>Continuous Free running, no external trigger signal needed.</td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = Software&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnDemand Image acquisition triggered by command (software trigger).</td>
<td></td>
</tr>
<tr>
<td>To trigger one frame execute the \texttt{TriggerSoftware@i} command then.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = FallingEdge&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnFallingEdge Each falling edge of trigger signal acquires one image.</td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = RisingEdge&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnRisingEdge Each rising edge of trigger signal acquires one image.</td>
<td></td>
</tr>
</tbody>
</table>

Line Mapping (\texttt{TriggerSource mvIMPACT Acquire -> TriggerSource GenICam (valid values for <desired Line>)):
27.8.2.2.2 Erase, exposure and readout

All pixels are light sensitive at the same period of time. The whole pixel core is reset simultaneously and after the exposure time all pixel values are sampled together on the storage node inside each pixel. The pixel core is read out line-by-line after exposure.

**Note**

Exposure and read out cycle is carry-out in serial; that causes that no exposure is possible during read out.

The step width for the exposure time is 1 us.

Image data is then shifted out line-by-line and transferred to memory.

To calculate the maximum frames per second (FPS\textsubscript{max}) in snapshot mode you will need following formula (in us):

\[
\frac{129}{\text{PixelClock}} \times \frac{32}{16 \times \text{ImageHeight}} + \left( \frac{20}{\text{LVDSOutputs}} + \frac{1000000}{\text{FrameTime} + \text{ExposureTime}} \right)
\]

27.8.3 Sensor Data

**Device Structure**

- CMOS image sensor (Type 1")
- Unit cell size: 5.5um (H) x 5.5um (V)

27.8.3.1 Characteristics

27.8.3.2 Color version
27.8.3.3 Gray scale version

27.8.3.4 Gray scale version (-X104bUV)

27.8.4 Device Feature And Property List

27.8.4.1 Gray scale version

27.8.4.2 Color version
27.9 mvBlueCOUGAR-Xx05 (5 Mpix [2592 x 1944])

27.9.1 Introduction

The CMOS sensor module (MT9P031) incorporates the following features:

- resolution to 2592 x 1944 gray scale or RGB Bayer mosaic
- supports window AOI mode with faster readout
- programmable analog gain (0..12 dB)
- progressive scan sensor (no interlaced problems!)
- rolling shutter / global reset
- programmable readout timing with free capture windows and partial scan
- many trigger modes (free-running, hardware-triggered)

27.9.2 Details of operation

The sensor uses two acquisition modes:

- rolling shutter (mvShutterMode = "RollingShutter") and
- global reset release shutter (mvShutterMode = "GlobalResetRelease").

With the rolling shutter the lines are exposed for the same duration, but at a slightly different point in time:

Note

Moving objects together with a rolling shutter can cause a smear in moving objects.

The global reset release shutter, which is only available in triggered operation, starts the exposure of all rows simultaneously and the reset to each row is released simultaneously, too. However, the readout of the lines is equal to the readout of the rolling shutter: line by line:
Note

This means, the bottom lines of the sensor will be exposed to light longer! For this reason, this mode will only make sense, if there is no extraneous light and the flash duration is shorter or equal to the exposure time.

Furthermore, the sensor offers two operating modes:

- free running mode (Overlapping exposure and readout)
- snapshot mode (Sequential exposure and readout) in triggered operation

27.9.2.1 Free running mode

In free running mode, the sensor reaches its maximum frame rate. This is done by overlapping erase, exposure and readout phase. The sensor timing in free running mode is fixed, so there is no control when to start an acquisition. This mode is used with trigger mode Continuous.

To calculate the maximum frames per second (FPS_max) in free running mode you will need following formula:

\[
\text{FrameTime} = (\text{ImageHeight} \times \text{RowTime}) + (25 \times \text{RowTime})
\]

If exposure time is lower than frame time:

\[
\text{FPS}_{\text{max}} = \frac{1}{\text{FrameTime}}
\]

If exposure time is greater than frame time:

\[
\text{FPS}_{\text{max}} = \frac{1}{\text{ExposureTime}}
\]

27.9.2.2 Snapshot mode

In snapshot mode, the image acquisition process consists off several sequential phases:

27.9.2.2.1 Trigger

Snapshot mode starts with a trigger. This can be either a hardware or a software signal.

The following trigger modes are available:
<table>
<thead>
<tr>
<th>Setting (GenICam (p.143))</th>
<th>Mode / Setting (Device Specific (p.143))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = Off&quot;</td>
<td>Continuous</td>
<td>Free running, no external trigger signal needed.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = Software&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnDemand</td>
<td>Image acquisition triggered by command (software trigger).</td>
</tr>
<tr>
<td>To trigger one frame execute the <strong>TriggerSoftware@i</strong> command then.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelLow&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnLowLevel</td>
<td>Start an exposure of a frame as long as the trigger input is below the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelHigh&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnHighLevel</td>
<td>Start an exposure of a frame as long as the trigger input is above the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = FallingEdge&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnFallingEdge</td>
<td>Each falling edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = RisingEdge&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnRisingEdge</td>
<td>Each rising edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = AnyEdge&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnAnyEdge</td>
<td>Start the exposure of a frame when the trigger input level changes from high to low or from low to high.</td>
</tr>
</tbody>
</table>
Line Mapping (TriggerSource mvIMPACT Acquire -> TriggerSource GenICam (valid values for <desired Line>):

<table>
<thead>
<tr>
<th>TriggerSource mvIMPACT Acquire</th>
<th>TriggerSource GenICam(BCX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP-IN0</td>
<td>Line4</td>
</tr>
<tr>
<td>GP-IN1</td>
<td>Line5</td>
</tr>
</tbody>
</table>

27.9.2.2.2 Erase, exposure and readout

All pixels are light sensitive at the same period of time. The whole pixel core is reset simultaneously and after the exposure time all pixel values are sampled together on the storage node inside each pixel. The pixel core is read out line-by-line after exposure.

**Note**

Exposure and read out cycle is carry-out in serial; that causes that no exposure is possible during read out.

The step width for the exposure time is 1 us.

Image data is then shifted out line-by-line and transferred to memory.

To calculate the maximum frames per second (FPS\text{max}) in snapshot mode you will need following formula:

\[
\text{FrameTime} = (\text{ImageHeight} \times \text{RowTime}) + (25 \times \text{RowTime})
\]

\[
\text{FPS}_{\text{max}} = \frac{1}{\text{FrameTime} + \text{ExposureTime}}
\]

27.9.2.3 Use cases

As mentioned before, “Global reset release” will only make sense, if a flash is used which is brighter than the ambient light. The settings in \text{wxPropView} (p. 115) will look like this:

In this case, \text{DigOut0} gets a high signal as long as the exposure time (which is synchronized with the GlobalResetRelease). This signal can start a flash light.

27.9.3 Sensor Data

**Device Structure**

- CMOS image sensor
- Image size: 5.70(H)x4.28(V)mm (Type 1/2.5")
- Number of effective pixels: 2592 (H) x 1944 (V)
- Unit cell size: 2.2um (H) x 2.2um (V)
27.9.3.1 Characteristics

27.9.3.2 Color version

27.9.3.3 Gray scale version

27.9.4 Device Feature And Property List

27.9.4.1 Gray scale version

27.9.4.2 Color version
27.10  mvBlueCOUGAR-Xx010 (10 Mpix [3856 x 2764])

27.10.1  Introduction

The CMOS sensor module (MT9J003) incorporates the following features:

- resolution to 3856 x 2764 gray scale or RGB Bayer mosaic
- supports window AOI mode with faster readout
- programmable analog gain (0..12 dB)
- progressive scan sensor (no interlaced problems!)
- rolling shutter / global reset
- programmable readout timing with free capture windows and partial scan
- many trigger modes (free-running, hardware-triggered)

27.10.2  Details of operation

The sensor uses two acquisition modes:

- **rolling shutter** (mvShutterMode = "ElectronicRollingShutter") and
- **global reset release shutter** (mvShutterMode = "GlobalResetRelease").

With the **rolling shutter** the lines are exposed for the same duration, but at a slightly different point in time:

![Diagram showing rolling shutter](image)

**Note**

Moving objects together with a rolling shutter can cause a shear in moving objects.

The **global reset release shutter**, which is only available in triggered operation, starts the exposure of all rows simultaneously and the reset to each row is released simultaneously, too. However, the readout of the lines is equal to the readout of the rolling shutter: line by line:

![Diagram showing global reset release shutter](image)
27.10 mvBlueCOUGAR-Xx010 (10 Mpix [3856 x 2764])

Note

This means, the bottom lines of the sensor will be exposed to light longer! For this reason, this mode will only make sense, if there is no extraneous light and the flash duration is shorter or equal to the exposure time.

Furthermore, the sensor offers two operating modes:

- free running mode (Overlapping exposure and readout)
- snapshot mode (Sequential exposure and readout) in triggered operation

27.10.2.1 Free running mode

In free running mode, the sensor reaches its maximum frame rate. This is done by overlapping erase, exposure and readout phase. The sensor timing in free running mode is fixed, so there is no control when to start an acquisition. This mode is used with trigger mode Continuous.

To calculate the maximum frames per second (FPS\textsubscript{max}) in free running mode you will need following formula:

\[
\text{FrameTime} = \frac{(\text{ImageHeight} \times \left(1648\right)) + (\text{VerticalBlank} \times 1648)}{\text{PixelClock} \times \text{PixelClock}}
\]

If exposure time is lower than frame time:

\[
\text{FPS}\textsubscript{max} = \frac{1000000}{\text{FrameTime}}
\]

If exposure time is greater than frame time:

\[
\text{FPS}\textsubscript{max} = \frac{1000000}{\text{ExposureTime}}
\]

MATRIX VISION GmbH
27.10.2.2 Snapshot mode

In snapshot mode, the image acquisition process consists of several sequential phases:

27.10.2.2.1 Trigger

Snapshot mode starts with a trigger. This can be either a hardware or a software signal.

The following trigger modes are available:

<table>
<thead>
<tr>
<th>Setting (GenICam (p.143))</th>
<th>Mode / Setting (Device Specific (p.143))</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = Off&quot;</td>
<td>Continuous</td>
<td>Free running, no external trigger signal needed.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = Software&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnDemand</td>
<td>Image acquisition triggered by command (software trigger).</td>
</tr>
<tr>
<td>To trigger one frame execute the TriggerSoftware@i command then.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelLow&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnLowLevel</td>
<td>Start an exposure of a frame as long as the trigger input is below the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerSelector = AcquisitionActive&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = LevelHigh&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnHighLevel</td>
<td>Start an exposure of a frame as long as the trigger input is above the trigger threshold. (No Frame←Trigger!)</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = FallingEdge&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnFallingEdge</td>
<td>Each falling edge of trigger signal acquires one image.</td>
</tr>
<tr>
<td>&quot;TriggerSelector = FrameStart&quot; &quot;TriggerMode = On&quot; &quot;TriggerSource = &lt;desired Line&gt;&quot; &quot;TriggerActivation = RisingEdge&quot; &quot;ExposureMode = Timed&quot;</td>
<td>OnRisingEdge</td>
<td>Each rising edge of trigger signal acquires one image.</td>
</tr>
</tbody>
</table>
"TriggerSelector = FrameStart"
"TriggerMode = On"
"TriggerSource = <desired Line>"
"TriggerActivation = AnyEdge"
"ExposureMode = Timed"

<table>
<thead>
<tr>
<th>OnAnyEdge</th>
<th>Start the exposure of a frame when the trigger input level changes from high to low or from low to high.</th>
</tr>
</thead>
</table>

Line Mapping (TriggerSource mvIMPACT Acquire -> TriggerSource GenICam (valid values for <desired Line>):

<table>
<thead>
<tr>
<th>TriggerSource mvIMPACT Acquire</th>
<th>TriggerSource GenICam(BCX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP-IN0</td>
<td>Line4</td>
</tr>
<tr>
<td>GP-IN1</td>
<td>Line5</td>
</tr>
</tbody>
</table>

27.10.2.2 Erase, exposure and readout

All pixels are light sensitive at the same period of time. The whole pixel core is reset simultaneously and after the exposure time all pixel values are sampled together on the storage node inside each pixel. The pixel core is read out line-by-line after exposure.

**Note**

Exposure and read out cycle is carry-out in serial; that causes that no exposure is possible during read out.

The step width for the exposure time is 1 us.

Image data is then shifted out line-by-line and transferred to memory.

To calculate the maximum frames per second (FPS$_{\text{max}}$) in snapshot mode you will need following formula:

$$\text{FrameTime} = \frac{(\text{ImageHeight} \times 3980)}{\text{PixelClock}} + \frac{(\text{VerticalBlank} \times 3980)}{\text{PixelClock}}$$

$$\text{FPS}_{\text{max}} = \frac{1}{\text{FrameTime}}$$

**Note**

The maximum duration of the signal ExposureActive used in Counter, Timer, or as digital output signal is limited to 172ms (96.8 MHz pixel clock) or 207ms (80 MHz pixel clock). These limits also apply to the chunk data parameter ChunkExposureTime. It does not matter which exposure time is set, the value of this parameter will always be below or equal to 172ms (96.8 MHz pixel clock) or 207ms (80 MHz pixel clock). ExposureEndEvents are generated with the help of this signal, too and for this reason they have too small timestamps above this limit.
27.10.2.3 Use cases

As mentioned before, “Global reset release” will only make sense, if a flash is used which is brighter than the ambient light. The settings in wxPropView (p. 115) will look like this:

<table>
<thead>
<tr>
<th>FlashMode</th>
<th>DigOut0</th>
</tr>
</thead>
<tbody>
<tr>
<td>FlashType</td>
<td>RollingShutterFlash</td>
</tr>
<tr>
<td>TestMode</td>
<td>Off</td>
</tr>
<tr>
<td>ShutterMode</td>
<td>GlobalResetRelease</td>
</tr>
</tbody>
</table>

In this case, DigOut0 gets a high signal as long as the exposure time (which is synchronized with the GlobalResetRelease). This signal can start a flash light.

27.10.3 Sensor Data

Device Structure

- CMOS image sensor
- Image size: 6.119(H)x4.589(V)mm (Type 1/2.3")
- Number of effective pixels: 3856 (H) x 2764 (V)
- Unit cell size: 1.67um (H) x 1.67um (V)

27.10.3.1 Characteristics

27.10.3.2 Color version
27.10.3.3  Gray scale version

![Graph showing quantum efficiency vs. wavelength for different filters.]

27.10.4  Device Feature And Property List

27.10.4.1  Gray scale version

27.10.4.2  Color version