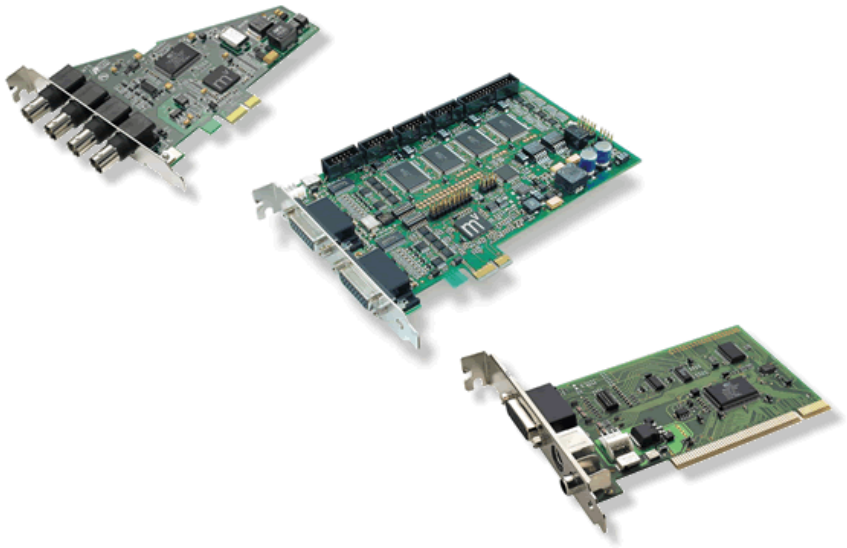
BALLUFF

A brand of Balluff MATRIX VISION



mvDELTA / mvSIGMA

Technical Manual

English - Version 3.00

1.1 About This Manual

1.1.1 Composition of the manual

The manual starts with technical data of the frame grabber as well as a quick start chapter.

Afterwards, you will find the different books:

- **GUI tools** (p. 34)
  - The frame grabbers can also be managed via user interface. The program is called **wxPropView** (p. 34).
- **Developing applications using the mvIMPACT Acquire SDK** (p. 35)
- **DirectShow developers** (p. 36)
  - This is the documentation of the MATRIX VISION DirectShow_acquire interface.

1.1.2 How to get started?

1.1.2.1 Introduction

This chapter gives you a short overview, how to get started with a MATRIX VISION frame grabber and where to find the necessary information in the manual. It will also explain or link to the concepts behind the driver and the image acquisition. Furthermore it shows you how to get start programming own applications.

1.1.2.2 Basics

1.1.2.2.1 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.

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

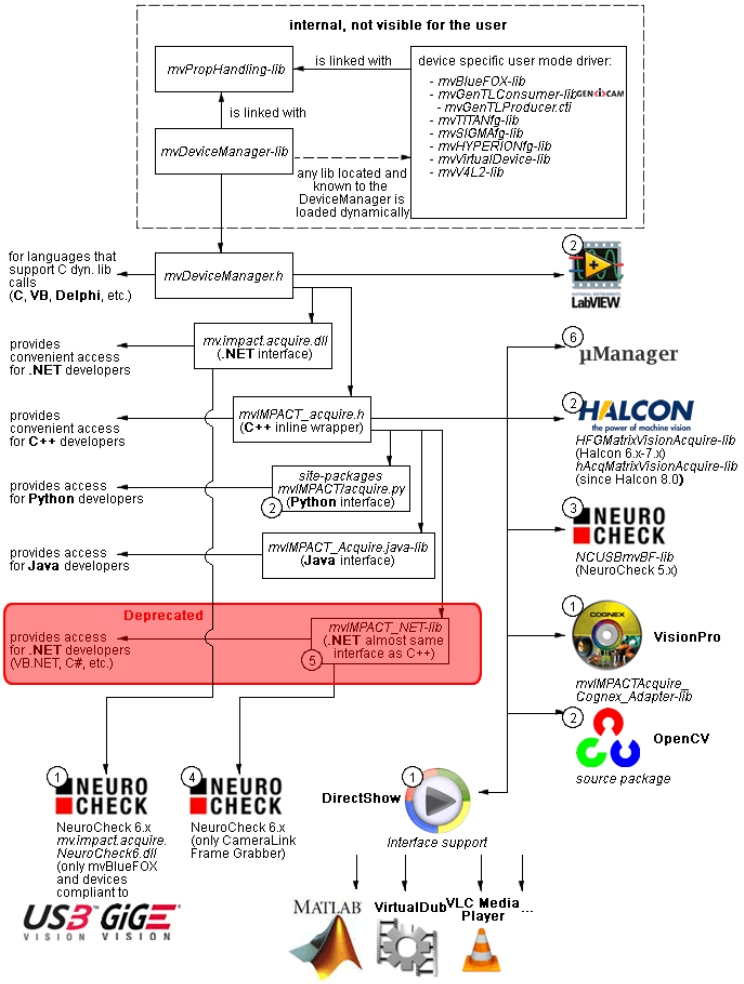


Figure 1: Driver concept

- **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.

1.1.2.2.2 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:

Device	Additional software needed
mvTITAN-G1	mvSDK driver for mvTITAN/mvGAMMA devices
mvTITAN-CL	mvSDK driver for mvTITAN/mvGAMMA devices
mvGAMMA-CL	mvSDK driver for mvTITAN/mvGAMMA devices
mvBlueFOX	mvIMPACT Acquire driver for mvBlueFOX devices, `"NCUSBmvBF.dll"`

For NeuroCheck 6.0 the following devices are supported:

Device	Additional software needed
mvTITAN-G1	mvIMPACT Acquire driver for mvTITAN/mvGAMMA devices
mvTITAN-CL	mvIMPACT Acquire driver for mvTITAN/mvGAMMA devices
mvGAMMA-CL	mvIMPACT Acquire driver for mvTITAN/mvGAMMA devices
mvHYPERION-CLb	mvIMPACT Acquire driver for mvHYPERION devices
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:

Device	Additional software needed
mvTITAN-G1	mvIMPACT Acquire driver for mvTITAN/mvGAMMA devices
mvTITAN-CL	mvIMPACT Acquire driver for mvTITAN/mvGAMMA devices
mvGAMMA-CL	mvIMPACT Acquire driver for mvTITAN/mvGAMMA devices
mvHYPERION-CLb	mvIMPACT Acquire driver for mvHYPERION devices
Every other mvIMPACT Acquire compliant device	mvIMPACT Acquire driver for the corresponding device family, "`mv.impact.acquire.NeuroCheck6_1.dll`" (comes with the driver package, but the driver package must be installed **AFTER** installing NeuroCheck 6.1

1.1.2.2.3 VisionPro support Every *mvIMPACT Acquire* driver package on 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.

1.1.2.2.4 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 or USB3 Vision standard don't need any software at all, but can also use HALCON's built-in GigE Vision or USB3 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.

1.1.2.2.5 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.

1.1.2.2.6 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. 36).

1.1.2.2.7 Micro-Manager support Every *mvIMPACT Acquire* compliant device can be operated under `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`.

1.1.2.3 Image acquisition concept

The image acquisition is based on queues to avoid the loss of single images. With this concept you can acquire images via single acquisition or triggered acquisition. For detailed description of the acquisition concept, please have a look at **"How the capture process works"** in the **mvIMPACT_Acquire_API** manual matching the programming language you are working with.

1.1.2.4 Installation

To install the frame grabber properly you have to follow these steps:
(Please follow the links for detailed descriptions.)

- **Windows:**
  - Please check the **system requirements** (p. 8).
  - Please **install the software and driver** (p. 9).
  - Please **install the hardware** (p. 7).
- **Linux:**
  - Please check the **system requirements** (p. 14).
  - Please **install the software and driver** (p. 15).
  - Please **install the hardware** (p. 7).

1.1.2.5 Programming

To control the camera and handle the images, you will have a good introduction by reading the main pages of the **"mvIMPACT Acquire"** interface references. Additionally, please have a look at the example programs. Several basic examples are available. The separate mvIMPACT Acquire manuals

- **"mvIMPACT_Acquire_API_CPP_manual.chm"**,
- **"mvIMPACT_Acquire_API_C_manual.chm"**, and
- **"mvIMPACT_Acquire_API_NET_manual.chm"**
  are available as downloads from our website `http://www.matrix-vision.com`.

Only mvTITAN/mvGAMMA and mvSIGMA/mvDELTA series using Linux:
Please have a look at the chapter **Writing your own applications** (p. 20) for details.

1.2 Imprint

MATRIX VISION GmbH
Talstrasse 16
DE - 71570 Oppenweiler
Telephone: +49-7191-9432-0
Fax: +49-7191-9432-288
Website: http://www.matrix-vision.de
E-Mail:

- info@matrix-vision.de
- jobs@matrix-vision.de

Author

U. Lansche

Date

2020

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.

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.

1.3 Revisions

Date	Rev.	Author	Description	Driver / Firmware version
03. May 2021	V3.00	LAN	Corrected **Symbols and Conventions** (p. 6).	
13. January 2021	V2.00	LAN	Seperated **GUI tools** (p. 34) .	

1.4 Symbols and Conventions

Note

This symbol indicates general notes.

1.4.1 Explanation of the warnings

Always observe the warnings in these instructions and the measures described to avoid hazards. The warnings used here contain various signal words and are structured as follows:

Attention

SIGNAL WORD
"Type and source of the hazard"

Consequences if not complied with

→ Measures to avoid hazards.

The individual signal words mean:

Attention

Indicates a **danger** that can lead to **damage** or **destruction** of the product.

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.

1.5 Important Information

	We cannot and do not take any responsibility for the damage caused to you or to any other equipment connected to the mvDELTA / mvSIGMA frame grabber. Similarly, warranty will be void, if a damage is caused by not following the manual.
	Handle the mvDELTA / mvSIGMA frame grabber with care. Do not misuse the mvDELTA / mvSIGMA frame grabber. Avoid shaking, striking, etc. The mvDELTA / mvSIGMA frame grabber could be damaged by faulty handling or shortage.
	• 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 antistatic versions) around printed circuit boards. – Do not touch components on the printed circuit board with your hands or with conductive devices.

1.5.1 European Union Declaration of Conformity statement

	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.

1.6 Introduction

The frame grabbers of the mvDELTA and mvSIGMA series are powerful image capture boards suitable for different capturing and digitizing signals. For more information about the single frame grabbers and the supported signals, please have a look at the **Technical Data** (p. 24).

1.6.1 What's inside and accessories

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

- **Frame grabber**

For the first use of the frame grabber a cable for the camera connection is needed.

1.7 Quickstart

1.7.1 Hardware installation

Attention

"Electro Static Discharge (ESD)"

Please take all proper Electro Static Discharge (ESD) precautions during the installation of your new hardware!

Before starting the installation, turn off your computer and all peripheral devices. Disconnect the computer from the power supply and all necessary components.

Note

To avoid doing damage to the hardware, discharge yourself of static charge by touching e.g. the casing. Beware of touching contacts of the frame grabber or of the computer.

- Select a free busmaster slot (PCI). Remove the slot's cover at the back of the computer and keep the screw.
- Carefully insert the board into the slot by holding the board at the top and gently pushing both ends into the slot at the same time. Press onto the upper edge of the board to make sure it is seated in the slot firmly.
- Do not force the board into the slot! You run the risk of bending the contacts. If the board does not fit easily, pull it back out, and try again.
- Fasten the board's bracket at the back of the computer using the screws you saved from the shield.
- Put the cover back on the computer and reconnect the peripheral devices.
- Start the computer.

1.7.2 Windows

1.7.2.1 System Requirements

Currently supported Windows versions are:

- Microsoft Windows 7 (32-bit, 64-bit) (requires min. 2 GB main memory)
- Microsoft Windows 8.1 (32-bit, 64-bit) (requires min. 2 GB main memory)

Note

Working on Windows XP you must enable at least the following advanced user right when not working with an Administrator account: "`Increase scheduling priority`" (for the German version of Windows: "`Anheben der Zeitplanungsprioritaet`"). These settings can be made on Windows XP via 'Settings -> Control Panel -> Administrative Tools -> Local Security Policy -> Local Policies -> User Rights Assignment').

Consecutively the installation for Windows will be described. The description for the Linux installation can be found here: **Linux** (p. 14) .

1.7.2.2 Software installation

All necessary drivers for Windows and Linux are contained in the mvIMPACT CD-ROM or DVD-ROM. For newer driver versions we recommend to visit the MATRIX VISION website at www.matrix-vision.de, section Support/↩ Download/Hardware.

After the **Hardware installation** (p. 7) the boot sequence shows **"Found New Hardware"** and starts the Windows Hardware Wizard. Closed this windows and insert the mvIMPACT CD-ROM or DVD-ROM into your drive and select **"Driver installation ..."** and the needed mvIMPACT Acquire driver (e.g. "mvTITAN / mvGAMMA").

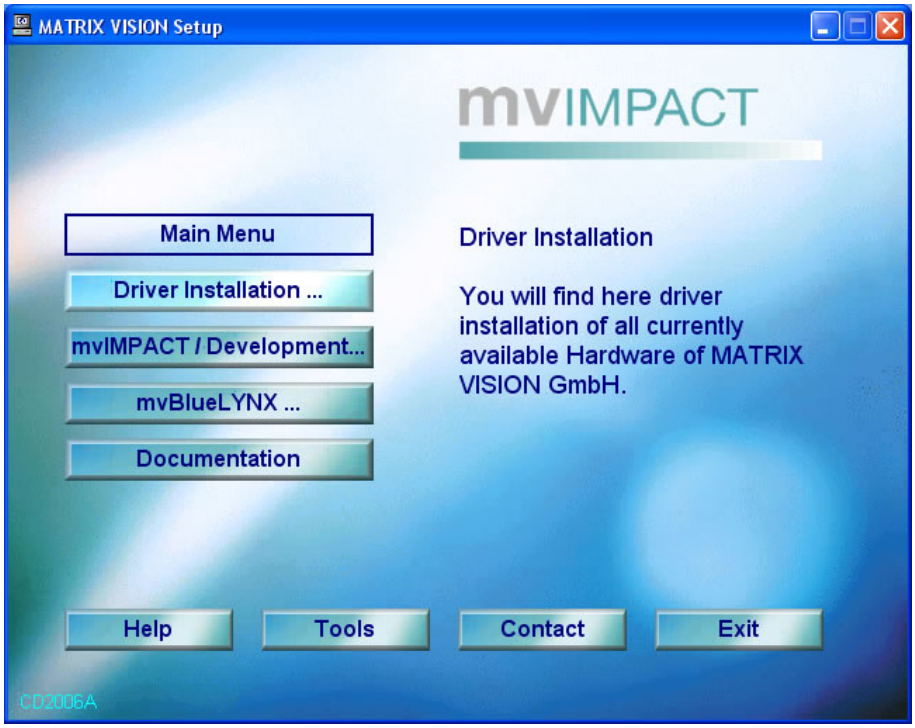


Figure 1: Start window

After the click on the needed driver the installation process starts.

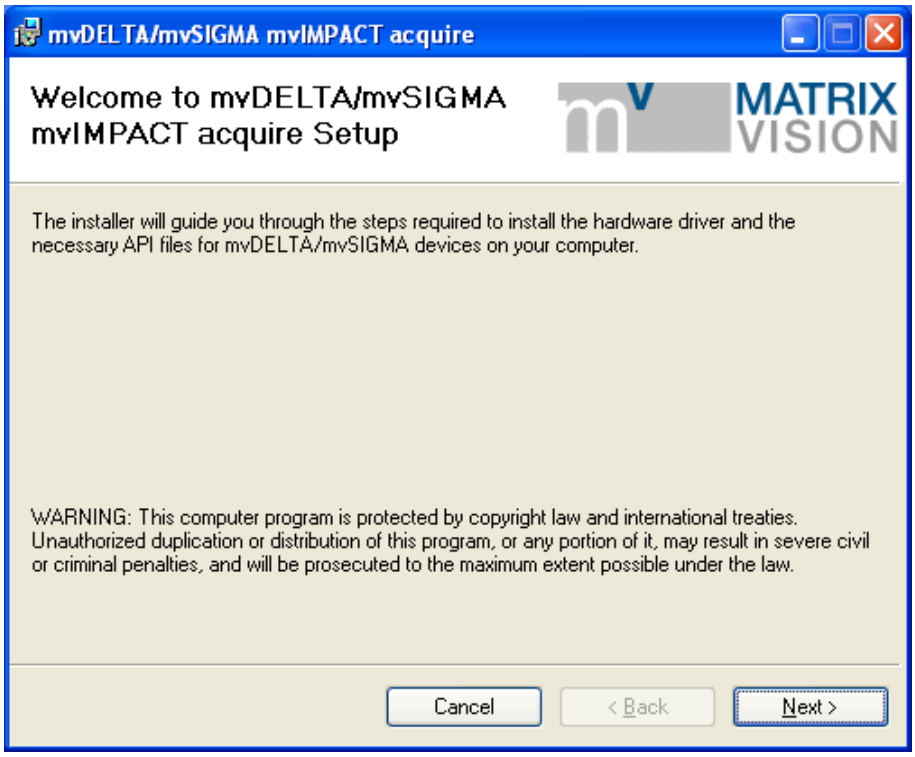


Figure 2: mvDELTA/mvSIGMA installer - Start window

Select the folder, where you want to install the software.

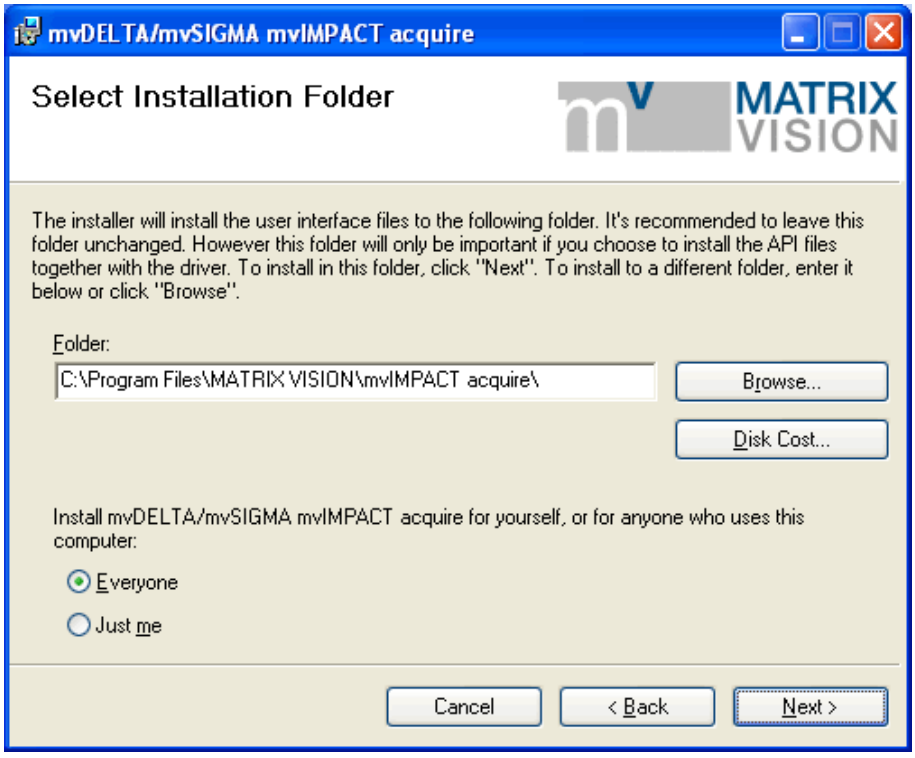


Figure 3: mvDELTA/mvSIGMA installer - Select folder

Select the features, which you want to install. Following features exist:

- **"Base Libraries"**
  This feature contains all necessary files for property handling and display. Therefore, it is not selectable.
- **"mvDELTA/mvSIGMA driver"**
  This is also not selectable.
- **"Tools"**
  This feature contains tools for the frame grabber (e.g. to acquire images (**wxPropView** (p. 34))).
- **"mvIMPACT acquire API"**
  The "mvIMPACT acquire API" contains the header for own programming. Additionally you can choose the examples, which installs the sources of wxPropView, mvIPConfigure and various small examples. The project files shipped with the examples have been generated with Visual Studio 2013. However projects and make-files for other compilers can be generated using CMake fairly easy. See CMake section in the C++ manual for additional details.
- **"Documentation"**
  This will install this manual as single HTML help file (.chm).

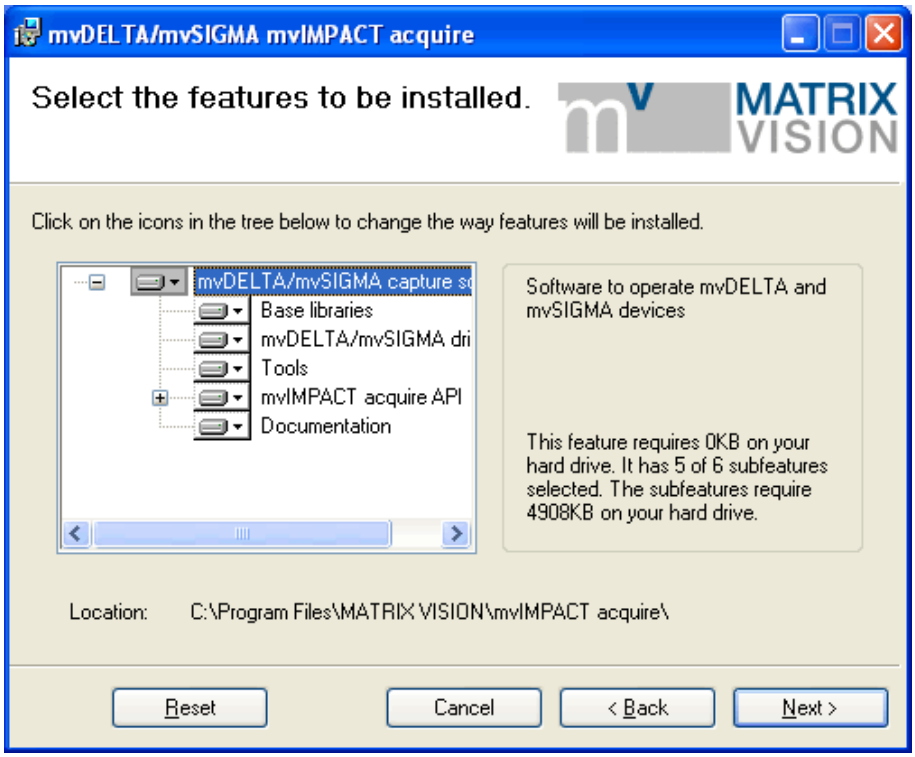


Figure 4: mvDELTA/mvSIGMA installer - Select features

Confirm the installation by clicking **"Next"**.

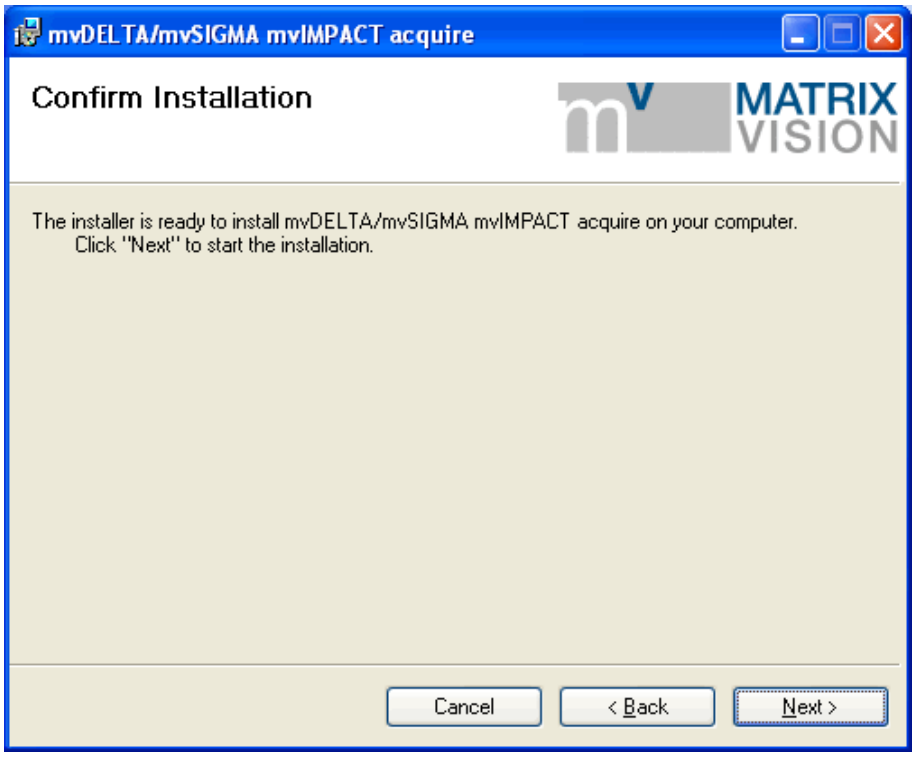


Figure 5: mvDELTA/mvSIGMA installer - Confirm installation

The installation process copies the files to Windows. Then Windows shows two messages to signal that the drivers are not checked through Microsoft. This is only an attempt to make insecure and it is recommended to ignore it.

Press **"Continue Anyway"**...

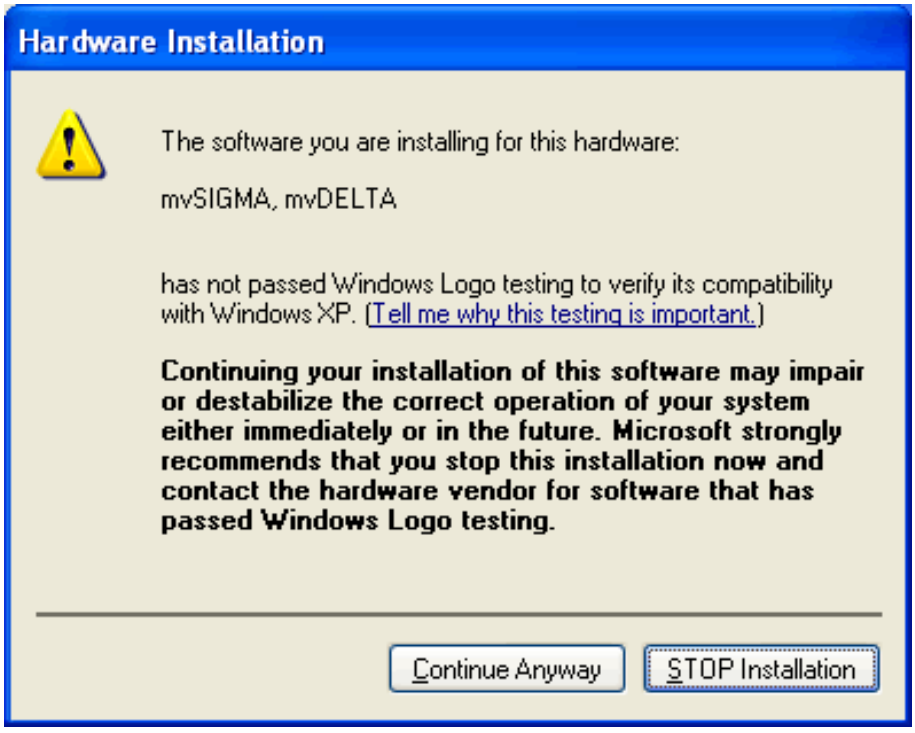


Figure 6: mvDELTA/mvSIGMA installer - Windows logo testing

... and again and finish the driver installation.

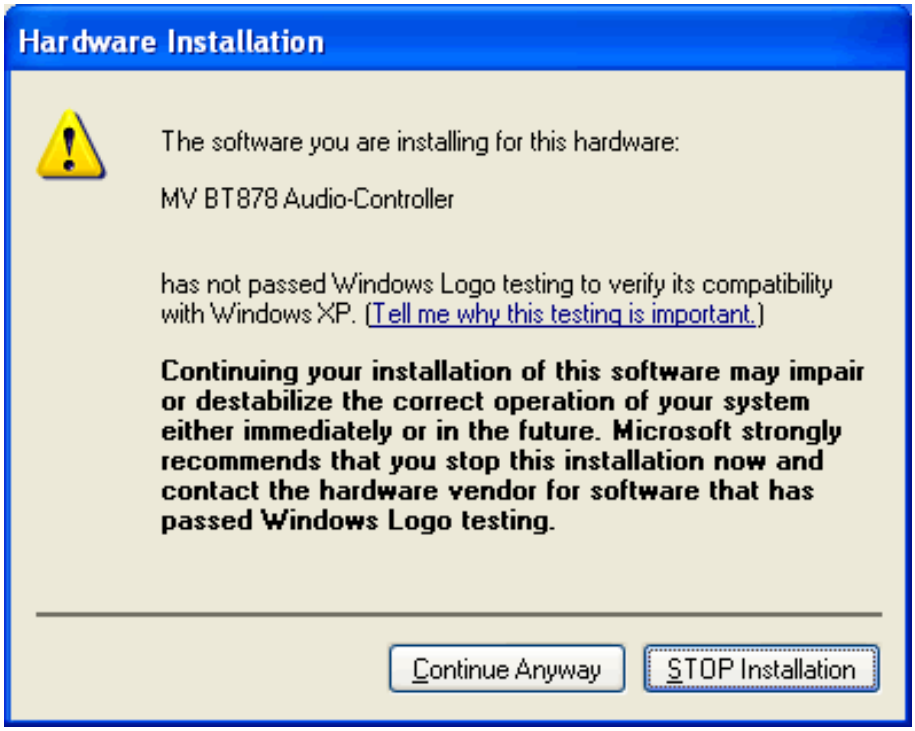


Figure 7: mvDELTA/mvSIGMA installer - Setup of the DMA buffer

Now, the installation process is finished.

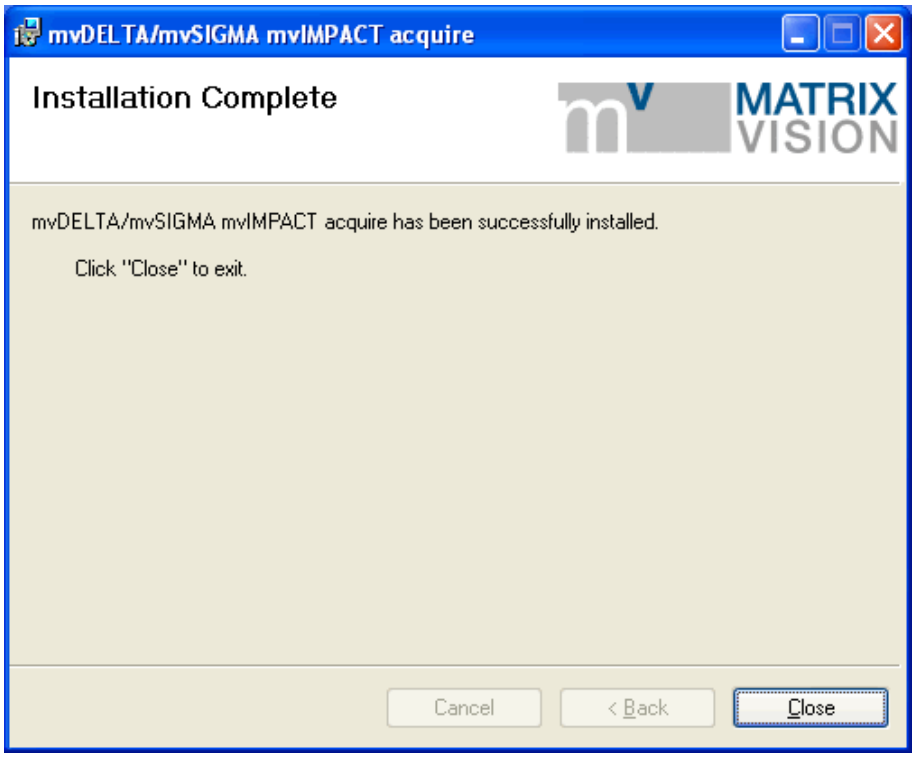


Figure 8: mvDELTA/mvSIGMA installer - Installation complete

After this, you have to restart the system.

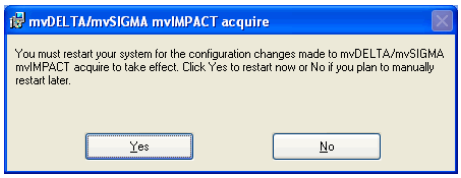


Figure 9: mvDELTA/mvSIGMA installer - Restart the system

After the restart, you can acquire images with the frame grabber. Simply start the application **wxPropView** (p. 34) (wxPropView.exe).

See also

wxPropView (p. 34)

1.7.3 Linux

The standard MATRIX VISION frame grabber drivers have been ported to Linux. A common code base ensures that the names and parameters of all the functions are identical to the Windows / DOS versions. This makes porting your existing Windows applications to Linux especially easy. We are constantly improving and updating so you should always take a look at our website for newer versions (`http://www.matrix-vision.de`). The driver consists of two parts:

- A kernel module called matrixfg.o, sigma.o, xsigma.o or titan.o. (From Linux kernel 2.5 onwards these will be called matrixfg.ko, sigma.ko, xsigma.ko and titan.ko). These modules handle kernel mode memory and I/O access and/or interrupts.
- A library which provides user programs with access to all the MATRIX VISION standard frame grabber functions as defined in the header file mv.h

1.7.3.1 System Requirements

Software requirements

- Linux kernel 2.4.x or 2.6.x running on i386 CPU.

  Note

  It is possible that older kernels will also work.

  You will need to have support for kernel modules turned on in your kernel configuration. (This is the default for all major Linux distributions). For the shared versions of the library, the GNU C runtime library libc6 is required. All recent Linux distributions should have this (e.g. SuSE Linux 9.0, 9.1, 9.2). The libraries have been compiled using glibc 2.3.3. on a SuSE 9.0 installation. If you use an older version of glibc you may experience problems. The libraries have been compiled using gcc version 3.3.1 (SuSE Linux). Using a system based on gcc 2.x is not recommended and may not work correctly. For display functions: a framebuffer, the svga library or for X applications, a correctly installed X system. It is also possible to use a framebuffer device. Check your distribution for a package called svga (or similar) or download and compile the sources for this library if you are not using X and you wish to display images or use a framebuffer (e.g. vesafb, rivafb etc.). To compile the kernel modules you will need a correctly configured GNU compiler and at least the kernel header files and Makefiles installed. It is usually advisable to install all of the kernel sources. The kernel configuration should exactly match the kernel you are using. During installation the kernel modules will be compiled for your system using exactly the same configuration as for the kernel. If you have not compiled your own kernel you will probably have to install the source package for your kernel and retrieve the correct kernel configuration for the running kernel too. This is often stored as file in the /boot directory (Red Hat, Debian, Mandrake) or can be generated from `/proc/config`.gz in the case of SuSE or for 2.6.x kernels which have the appropriate kernel option turned on.

Hardware requirements

- i386 compatible PC. (PowerPC versions of the software are currently being prepared.).

One of the following types of MATRIX VISION frame grabber:

- mvDELTA
- mvSIGMA
- mvGAMMA
- mvTITAN

1.7.3.2 Software Installation

1. Copy the appropriate driver tarball from the CD or web to your computer (e.g. CD-ROM is mounted on `/cdrom` and directory `/mypath` exists):

   ```
   >cp -v /cdrom/linux/dist_titan-030801.tgz /mypath
   ```

2. Unpack the tgz-file using:

   ```
   >tar -xvzf dist_titan-030801.tgz
   ```

3. Make the test programs and kernel modules by changing to the top-level directory:

   ```
   >cd /mypath/dist_titan-030801
   ```

4. and then...

   ```
   > ./configure
   > make
   ```

   It is also possible (and sometimes necessary) to start the configure program with options. See the INSTALL file for full details. Two common options are:

   ```
   > ./configure --with-kernel=<kernel source directory>
   ```

   (to enter an alternative location for the kernel sources. The default location is `/usr/src/linux`).

5. Ensure you are logged on as "`root`" and then install the kernel modules on your computer with:

   ```
   >make install
   ```

Allocator kernel module

Some MATRIX VISION frame grabbers support a linear DMA buffer in Hard Live mode. In order to ensure that we have a linear memory area, we use a modified version of the allocator kernel module from Alessandro Rubini's book Linux Device Drivers (O'Reilly 1998 ISBN: 1-56592-292-1). This originally handled only linear memory blocks above the top of the memory range usually used by the kernel (highmemory). The allocator module from Rubini will not work correctly on systems with more than 1 GB of memory. In fact, on 2.4.x kernels the upper limit is mem=896M. In order to address this problem we have significantly rewritten parts of the allocator module to use a different method and this is now used by default. The two methods are described below:

1. *Method 1 "MATRIX VISION kmalloc method"*:

   This method makes use of the kernel function kmalloc to reserve blocks of memory. Since this function is only capable of reserving small blocks of memory (normally 128 kB) it must be called several times to reserve a block of, say 2 MB. If kmalloc is called early in the boot process before too many applications have used up memory (and possibly returned it to the pool, thereby creating "holes" in the memory map) then it is very likely that successive calls to kmalloc will return successive linear blocks of memory. On a machine with, say, 1GB of RAM the chances are very high that you will be able to get 2MB of continuous memory. The longer the computer runs the less likely is this method to work. Therefore it is essential that the allocator module is loaded as early as possible in the boot process and not unloaded again, as long as it is needed. The way to do this depends on your Linux distribution. Some, like Debian, allow you to enter the name of kernel modules that should be loaded during boot in a file (/etc/modules). Others, like SuSE or United Linux, allow boot scripts in /etc/init.d/boot.d which are capable of loading required modules. If you can find a way to load the allocator module during booting you may use this method and you will not need to pass a mem= parameter to your kernel in the boot loader.

2. *Method 2 "Rubini's Method"*:

   In order to reserve space for allocator you need to start the kernel with an extra parameter which instructs it to use slightly less than all the memory. e.g If your computer contains 256 MB of RAM start the kernel with the parameter:

   ```
   mem=252M
   ```

   The kernel will be able to use 252MB of memory and 4 MB may be used by allocator. A computer with 512 MB of RAM could be started with:

   ```
   mem=508M
   ```

   Note

   Method 1 is only available if you are using a kernel with a version number >= 2.4.20.

   If you do not start your kernel using this parameter the allocator module will still load without an error but linear DMA access will not work correctly. The amount of high memory you should reserve depends on the resolution and colour depth you are going to use and on your application.

   Method 2 will not work correctly on systems with more than 1 GB of memory. In fact, on 2.4.x kernels the upper limit is mem=896M

3. *Parameters:*

   Some kernel module parameters are available to affect the way allocator works:

   Examples:

   If you need to reserve more RAM (e.g. 8MB) then use linsizemb=8.
   If you want a more conservative search (100 MB) then use maxresmb=100
   If you want to force Method 2 use use_kmalloc=0 (only on 2.4.x kernels).
   If you know that you have reserved exactly 8MB using the kernel mem=parameter (Method 2) but you do not want the allocator module to search for this free memory then try use_kmalloc=0 himem=8

4. *Summary:*

   Assuming you have an up-to-date kernel method 1 will be used by default to reserve 4 MB of RAM and you will normally not need to provide any parameters. Up to half of your available RAM will be searched for this 4 MB. You should load allocator during the boot process. If you have a kernel older than 2.4.20 then method 2 will be used and you must provide a mem= parameter to your kernel boot loader and cannot use more than 896 MB of RAM (even if more is installed in your computer). You can load or unload allocator anytime you want.

Frame grabber kernel module

You can create the kernel modules individually, if you want:

1. Login as root.

2. Change directory to the

   ```
   driver/os/linux/allocator
   ```

   and

   ```
   driver/os/linux/matrixfg (or driver/os/linux/titan for e.g. mvTITAN)
   ```

   subdirectories in succession and

3. compile the kernel modules using the following command in each directory:

   ```
   >./configure

   > make
   ```

4. You can install the kernel modules (`allocator.o` and `matrixfg.o` or `titan.o`) by copying them to the appropriate directory on your computer (e.g. "`/lib/modules/2.2.16/misc`") and then typing this command in the appropriate directory:

   ```
   >make install
   ```

   You can now load the kernel modules by calling the script

   ```
   mvload matrixfg (or mvload titan)
   ```

   to be found in the "`bin`" subdirectory. This will also create the appropriate device nodes after loading the modules. The script

   ```
   mvunload matrixfg
   ```

   unloads both modules and deletes the device nodes.

   Note

   Device numbers for MATRIX VISION frame grabbers have not yet been officially assigned so we use a dynamic major device number technique which will usually assign a number of 254. See the script mvmodprobe or Rubini's book for more details. If you want to assign a fixed number (e.g. one of the numbers used for experimental drivers, for example 60) you can load matrixfg with a parameter (see below). Fixed major numbers also allow automatic loading of the kernel modules if you enter the appropriate alias lines in /etc/modules.conf. If everything has worked correctly you should see that both allocator and matrixfg(or titan) have been loaded by using the command:

   ```
   >lsmod
   This will list all the loaded kernel modules like this....
   Module Size Used by
   matrixfg 17212 0 (unused)
   allocator 2104 0 [matrixfg]
   ... etc.
   >
   ```

   A very small part of the driver is supplied in binary form only. If you do not wish to "taint" your kernel then please use the GPL version of the driver. The GPL version contains most of the features of the non-GPL version with the exception of e.g. fast channel switching and triggered capture of images.

5. *Parameters:*

   The kernel modules matrixfg or titan may be loaded using one or more of the following parameters:

VGA card

In order to display live images on your screen the driver attempts to determine the starting address of the linear memory used by the installed VGA card. It is possible to do this if the X-Windows system is running using function calls to an extended library (DGA). However, this library is not installed on all X-Windows systems, so the configure script will attempt to find out if DGA is being used on your system. Sometimes it may be helpful to turn on DGA by altering the configuration file for X-Windows (usually /etc/X11/XF86Config). To do this you need to have "root" access and you need to add a line to the section called "Module" like this:

```
Section "Module"
Load "extmod"
..
etc.
```

Using PCI mapping information the driver can determine which of your cards is the VGA card and which memory areas are mapped. However, it is not always possible to tell which of the mapped areas is the linear image memory. If the VGA card is a common one it should be known by the driver and the VGA detection will work correctly. New cards are constantly appearing on the market and it is not always possible to keep the driver up-to-date. In order to ensure that your VGA card is correctly detected please follow the procedure described below:

You may check whether the VGA card has been correctly detected by looking at your kernel message file (using dmesg or look at the file /var/log/messages). If you find (after some searching) a line like this

```
Sep 10 11:00:45 .... kernel: mv_vga Unknown video memory base address.
```

it means that the VGA card is unknown. If we don't know the VGA card then we can't be sure of the linear display memory address which we need as the destination for our DMA transfers. In this case we need:

1. to find out the address and start the kernel module by hand with an extra parameter and/or
2. add the information to a table in the kernel module sources and recompile.

In fact, the best way is to try it out using (1) and then do (2).

Here's how to do it.: Open the file `driver/os/linux/kmod/vgatable.h` and read the comments in the header area. It tells you how to add the vendor ID and device ID for your graphic card to the driver, Alternatively you can load the matrixfg kernel module using an extra parameter like this:

```
vidmem=0xd8000000 (Example for a NVidia card)
```

Just replace `0xd8000000` with the correct PCI memory address for your card. You can find this out by using `"lspci -v"` or `"cat /proc/pci"`. If your card has more than one memory area you might have to try them all out until, when loading the kernel module, the error message (Unknown video memory base address) disappears and the test program works.

Note

The table in `driver/os/linux/kmod/vgatable.h` shows you which of the base addresses (`PCI_BASE_ADDRESS_0` or `PCI_BASE_ADDRESS_1`) most of the usual cards use. `PCI_BASE_ADDRESS_0` is the first address that `"lspci -v"` or `"cat /proc/pci"` shows and `PCI_BASE_ADDRESS_1` is the second address, Some dual-head cards have 3 addresses. Note that the command lspci is actually deprecated and it may be necessary to examine the PCI cards using a different command or the command `"cat /proc/pci"`.

Example Here is part of the output for Linux computer at MATRIX VISION (`"lspci -v"`):

```
01:00.0 VGA compatible controller: nVidia Corporation GeForce 256 (rev 10)
(prog-if 00 [VGA])
Subsystem: Guillemot Corporation: Unknown device 5022
Flags: bus master, 66Mhz, medium devsel, latency 32, IRQ 11
Memory at e0000000 (32-bit, non-prefetchable) [size=16M]
Memory at d8000000 (32-bit, prefetchable) [size=128M]
Expansion ROM at <unassigned> [disabled] [size=64K]
Capabilities: [60] Power Management version 1
Capabilities: [44] AGP version 2.0
```

And here's the same thing using "`lspci -nv`" (showing numeric IDs). Class 0300 is a VGA card:

```
01:00.0 Class 0300: 10de:0100 (rev 10)
Subsystem: 14af:5022
Flags: bus master, 66Mhz, medium devsel, latency 32, IRQ 11
Memory at e0000000 (32-bit, non-prefetchable) [size=16M]
Memory at d8000000 (32-bit, prefetchable) [size=128M]
Expansion ROM at <unassigned> [disabled] [size=64K]
Capabilities: [60] Power Management version 1
Capabilities: [44] AGP version 2.0
```

Trying out the address `0xe0000000` does not work here. But the second address (0xd8000000) does work. So the following line needs to be added to the table in *vgatable.h* and *matrixfg.o* needs to be recompild, installed and loaded. The vendor ID is `0x10de` and the device ID is `0x0100` (from the output above):

```
{0x10de, 0x0100, "Geforce 256", PCI_BASE_ADDRESS_1},
```

Sometimes the IDs have already been defined in the kernel headers (*/usr/source/linux/include/linux/pci*.h (or pci_↩ ids.h) and you can use sensible names instead of numbers like this (an invented example, doesn't really exist):

```
{PCI_VENDOR_ID_NVIDIA, PCI_DEVICE_ID_NVIDIA_GEFORCE2_256, "Geforce 256",
PCI_BASE_ADDRESS_1}
```

If you are successful you might like to inform MATRIX VISION what you have added to the table so that it can be included in future releases.

Note

Normally all chips from the same family use the same address base. e.g. All current NVidia chips use PCI_BASE_ADDRESS_1 which is the second address shown in lspci -v.

Libraries

The libraries are supplied as shared versions. The libraries are to be found in the subdirectory *lib*. The command

```
make install
```

will copy the appropriate library file for your frame grabber to the standard library path on your computer e.g. */usr/lib* depending on the prefix you passed to

```
./configure
```

Alternatively you may leave the libraries in the *lib* subdirectory or copy the library to any other directory but you will have to set the environment variable *LD_LIBRARY_PATH* to include this directory.

E.g. for the bash shell:

```
>export LD_LIBRARY_PATH=$ LD_LIBRARY_PATH:/mypath/lib
```

Yet another alternative is to change the configuration file for your `ldconfig` command or use the `mvldconfig` script in the MATRIX VISION bin directory. The shared libraries contain the version number within the name. E.g. if the library for the mvSIGMA-SLC is called *libslc.so.3.22.0*, you will find symbolic links to the libraries within the lib directory:

```
>ls -l lib
drwxrwxr-x 3 technik users 1024 Sep 25 17:20 .
drwxrwxrwx 21 technik users 2048 Sep 25 17:26 ..
lrwxrwxrwx 1 technik users 16 Sep 24 09:53
libslc.so -> libslc.so.3.22.0
lrwxrwxrwx 1 technik users 16 Sep 24 09:54
libslc.so.3 -> libslc.so.3.22.0
-rwxrwxr-x 1 technik users 561114 Aug 30 13:11
libslc.so.3.22.0
>
```

The following is a list of the libraries available and their function. The names of the libraries actually supplied may vary depending on the current version/build number (e.g. *libtitan.so.1.0.2* instead of *libtitan.so.1.0.1*).

Note

The term mvTITAN series is used for all mvTITAN and mvGAMMA products

1.7.3.3 Writing your own applications

A number of test applications have been included to show you how to write programs that use the MATRIX VISION frame grabber driver. Some of these test programs may not be contained in the driver version you receive but at least linuxtest should always be available.

linuxtest

The files contained in the subdirectory **linuxtest** can be used to make simple test programs for all the MATRIX VISION frame grabbers. An image will be shown on the screen in hard live or soft live mode. You can make versions without any display at all (nodisp), VGA (vga) , X (x11) or Frame buffer (fb) versions. The top-level configure script will already have made the test programs for you and you will find them in the subdirectories called nodisp, vga, x11 and fb. Alternatively you may change directory to the linuxtest subdirectory and type

```
>./configure
> make
```

to (re)make the programs. The configure script will examine your system and only make test programs according to the libraries installed on your system. I.e. the SVGA versions will only be made if the SVGA library has been installed and can be found by the configure script. Likewise, if you only have the MATRIX VISION libraries for the mvTITAN series, the test programs for mvTITAN will be the only ones created. If you install other display libraries later you will have to repeat the `configure`, `make` process so that they are recognised.

Example for mvTITAN series: The command

```
>xtitantest --help
```

will show a list of possible parameters for the test program. You will also need a file called *titantest.ini* which contains information about the color depth, camera definitions etc. A number of example INI files are supplied within the `linuxtest/examples` subdirectory. You should copy and rename or use symbolic links as appropriate. In general, the INI file required is named after the test program:

See the README file in the linuxtest directory for more details.

sdltest

The files contained in the subdirectory sdltest can be used to make simple SDL based test programs for all the MATRIX VISION frame grabbers. These programs are heavily based on **linuxtest** but use SDL to create and manage a Window. If you are running X and you have the correct SDL libraries installed on your system these programs will be created by the top level configure script. Again, you may change to the subdirectory and call configure and make there if you wish.

See the README file in the sdltest directory for more details.

1.7.3.4 Special notes for Linux

VGA applications

When using the VGA library for display you should add an extra, as yet undocumented function to the INI file used to initialize the frame grabber:

```
UseLibVga 1024 768 8 0
```

The parameters are

- x resolution in pixels for the display
- y resolution in pixels for the display
- color depth in pixels for the display
- set to 0 if using the first 3 parameters

Alternatively you may specify a number representing the VGA mode as the fourth parameter and leave the first three parameters as 0. See the VGA library documentation for details about the modes. E.g.

```
UseLibVga 0 0 0 12
```

will set VGA mode 12.

Note

In order to change VGA modes you need access rights as root.

Frame buffer applications

When using the frame buffer display library (`libmvfb`) for display you will need to be running a valid framebuffer otherwise you will get unpredictable results and/or crashes. The kernel must contain frame buffer support - either `vesafb` (for VESA-compatible cards) or a frame buffer that supports your graphic card directly (e.g. `rivafb`).

1.7.3.5 Copyrights

We have tried our best to adhere to the terms of all software licenses. Please contact us if you feel we have misinterpreted anything.

Some of the original work on the matrixfg, sigma and titan kernel modules was based on the source code for the bttv driver, in particular the routines to detect the VGA card and some memory management routines. Since the original code was released under the terms of the GNU General Public License (GPL), we are happy to supply our kernel module source code and give full credit to the original authors (see header files). Our kernel driver source code is therefore also released under the terms of the GPL, the full text of which may be found in the file GPL/COPYING. Loading these kernel modules will not "taint" your kernel!

The xsigma kernel module is an exception. Not all the source code for this module is supplied and this module is not released under the GPL. It contains no "foreign" GPL code (i.e. copyright not held by MATRIX VISION GmbH) or code derived from "foreign" GPL code. If this is a problem for you please use the sigma kernel module and libslc instead. The GPL version has a slightly reduced functionality.

The allocator module is Copyright © 1998 Alessandro Rubini and is released under the terms of the GPL. We have made some alterations which are documented in the source files.

The tutorial and test program for X are Copyright © 1998- 2005 MATRIX VISION GmbH and are distributed under the terms of the GPL. Please note that KDE/QT programs you develop may require a commercial license from TrollTech unless you also distribute them under the GPL or QPL, See `http://www.trolltech.no` or `http://www.trolltech.com` for more details. The VGA library used by some of the display libraries is free software and may be distributed and modified without restriction. The X libraries appear to be distributed under similar terms. See the header files on your Linux system for details.

The GNU C library is distributed under the terms of the GNU Library General Public License (GLPL). Applications. According to the terms of this license, "work that uses the library" is not restricted by the license. The full text of this license may be found in the file GPL/GLPL.html.

The frame grabber shared libraries for MATRIX VISION frame grabbers are based solely on the original Windows source code developed by us and contain no GPL code. These files are Copyright © 1992-2005 MATRIX VISION GmbH and the source code is not included here. You may distribute the binaries together with your own applications if you include the above Copyright notice in your documentation. Programs that use kernel services and functions are not derived form the kernel and are therefore not covered by the GPL like the kernel sources themselves.

The library libtitan.so.x.x for the mvTITAN series of frame grabbers uses some binary code from the Trimedia® code relocation library (libload.a) supplied by Philips. These functions are © Copyright Philips Semiconductors Trimedia Product Group 1997-2000.

1.7.4 Settings behavior 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.

Now, whenever a device is opened, the driver will execute following procedure:

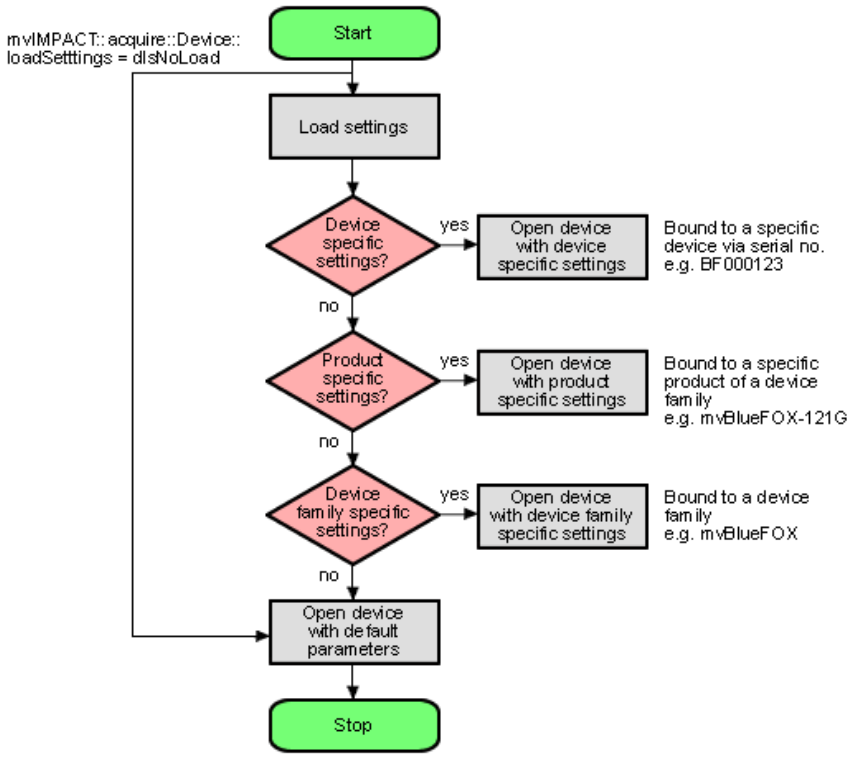


Figure 10: wxPropView - Device setting start procedure

- 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. On 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). So when the user is currently working with a device 'VD000001' belonging to the product group 'VirtualDevice' and there is a setting exclusively for this device storing a product specific setting now will automatically delete the setting for 'VD000001'. 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. 34) 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.
- On 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 by 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 here 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. 34) will be stored in these settings!
- 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. 34) 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. 34) 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.

1.8 Technical Data

1.8.1 mvDELTA

1.8.1.1 Connectors

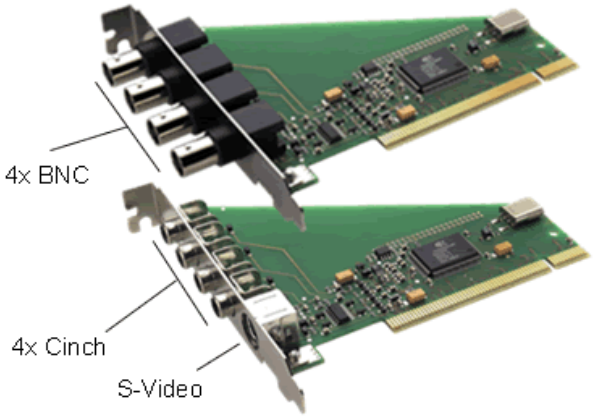


Figure 1: Connectors mvDELTA

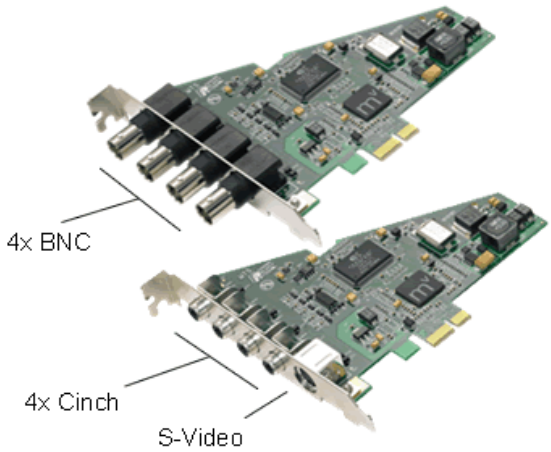


Figure 2: Connectors mvDELTAe

1.8.1.2 Technical specifications

	mvDELTA	mvDELTAe
Video input		
Input signal	Interlaced, monochrome	
	Interlaced, color	
50Hz	CCIR, PAL, SECAM, S-VHS, Y/C	
60 Hz	RS-170, RS-330, NTSC	
Number of video inputs	4	
Pixel clock	17.7 MHz @ 50 Hz	
	14.3 MHz @ 60 Hz	
Resolution		
Digitalisation	768 x 576 pixels (50Hz)	
Storage	0..768 x 0..576 pixels (50Hz)	
Grey scale	8 bit	
Color	24 bit true color	
Memory formats	32 bit RGB, 24 bit RGB, 16 bit RGB, 15 bit RGB, 16 bit YUV packed, 16 bit planar	
Image memory	Main memory of PC or image memory of VGA	
Interface		
Bus	PCI bus	PCI Express x1 bus
Transfer rate	DMA, 0-wait bursts, max. 132 MB/sec	
Intel FX chip set	Approx. 90 MB/sec	
Current consumption		
PCI 5V	max. 0.21A	
PCIe 3.3V		max. 0.15A
PCIe 12V		max. 0.2A
Environmental conditions		
Ambient temperature	0 up to 50 C	
Storage temperature	-20 up to 70 C	
Humidity	10 up to 90 % non-condensing	
Dimensions		
Length	123 mm	
Width	95 mm	

1.8.2 mvSIGMA-SLC

1.8.2.1 Connectors

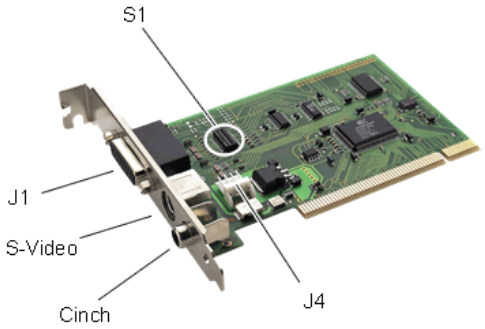


Figure 3: Connectors mvSIGMA-SLC

Default settings are bold.

1.8.2.1.1 J1: D-Sub HD26 digital input / output & power (rev. 5.00 and higher)

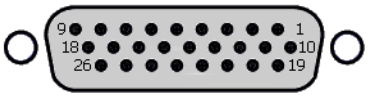


Figure 4: D-Sub HD26 pin numbering (female)

Pin.	Signal	Comment
1	+12V	12V camera power supply (0.7A)
2	Video 1	Video input 1, (with S-VHS mode -> Y1) parallel to cinch jack (cable inside)
3	Video 2	Video input 2, (with S-VHS mode -> Y2) parallel to S-Video jack (pin 3)
4	Video 3	Video input 3, (with S-VHS mode -> Y3)
5	Video 4	Video input 4, (with S-VHS mode -> Y4)
6	GP-OUT 2	Digital TTL output port 2
7	GP-OUT 3	Digital TTL output port 3
8	Trigger-In	External trigger input / digital input
9	GP-IN 4	Digital TTL input port 4
10	Ground camera supply	
11	GP-IN 5	Digital TTL input port 5
12	Ground Video 1	Ground video input 1 and chroma input 1
13	Ground Video 2	Ground video input 2 and chroma input 2
14	Ground Video 3	Ground video input 3 and chroma input 3
15	Ground Video 4	Ground video input 4 and chroma input 4
16	GP-IN 6	Digital TTL input port 6
17	Ground ext Trigger	Ground ext. Trigger / d igital input and output ports 0/1
18	GP-IN 7	Digital TTL input port 7
19	GP-OUT 0	Digital TTL output port 0

20	GP-OUT 1	Digital TTL output port 1
21	Chroma input 1	Chroma input 1 (C1) with S-VHS mode
22	Chroma input 2	Chroma input 2 (C2) with S-VHS mode, parallel to S-Video jack (pin 3)
23	Chroma input 3	Chroma input 3 (C3) with S-VHS mode
24	Chroma input 4	DChroma input 4 (C4) with S-VHS mode
25	Video out	Y out
26	Video out	only in S-VHS mode: C out

1.8.2.1.2 J4: Power supply (floppy) You can connect a free power supply cable for floppy drives on connector J4 to increase the available current on the power supply pins on J1 to 2A.

Figure 5: J4

Pin	Signal
1	+12V
2	Ground
3	Ground
4	Not connected

1.8.2.1.3 Termination of video input With switch S1 the termination of the video inputs can be switched on/off. Default adjustment of switch S1 is **on**.

Meaning of switch S1, 75Ohm terminator

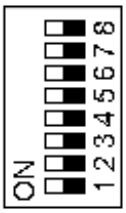


Figure 6: Video

Position	Comment
1	CVBS0 / Y0
2	CVBS1 / Y1
3	CVBS2 / Y2
4	CVBS3 / Y3
5	C0
6	C1
7	C2
8	C3

1.8.2.2 Technical specifications

	mvSIGMA-SLC
Video input	
Input signal	Interlaced, monochrome
	Interlaced, color
50 Hz	CCIR, PAL, SECAM, S-VHS, Y/C
60 Hz	RS-170, RS-330, NTSC
Number of video inputs	4
Sync signal	External source
Pixel clock	17.7 MHz @ 50 Hz
	14.3 MHz @ 60 Hz
Resolution	
Digitalisation	768 x 576 pixels (50 Hz)
Storage	0..768 x 0..576 pixels (50 Hz)
Grey scale	8 bit
Color	24 bit true color
Memory formats	32 bit RGB, 24 bit RGB, 16 bit RGB, 15 bit RGB, 16 bit YUV packed, 16 bit planar
Image memory	Main memory of PC or image memory of VGA
Interface	
Bus	PCI bus
Transfer rate	DMA, 0-wait bursts, max. 132 MB/sec
Intel FX chip set	Approx. 90 MB/sec
Current consumption	
PCI 5V	max. 0.3A
PCI 12V	max. 0.05 A (without camera)
Camera supply	Via PCI: 12V max. 0.7A fused
	Via additional power plug: 12V max. 1.5V fused
Environmental conditions	
Ambient temperature	0 up to 50 C
Storage temperature	-20 up to 70 C
Humidity	10 up to 90 % non-condensing
Dimensions	
Length	125 mm
Width	76 mm

1.8.2.3 Device Feature And Property List

1.8.3 mvSIGMA-SQ/-SQe

1.8.3.1 Connectors

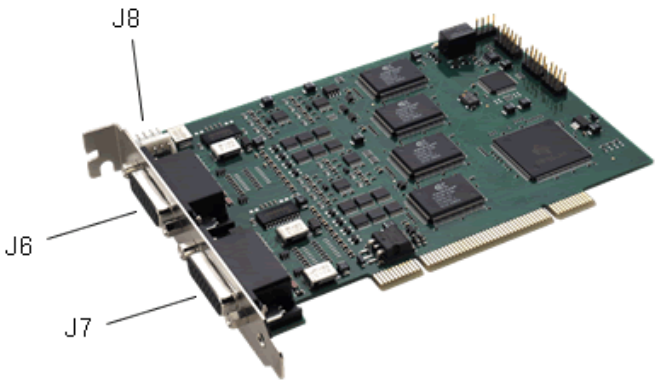


Figure 7: Connectors mvSIGMA-SQ (Rev. 1.0)

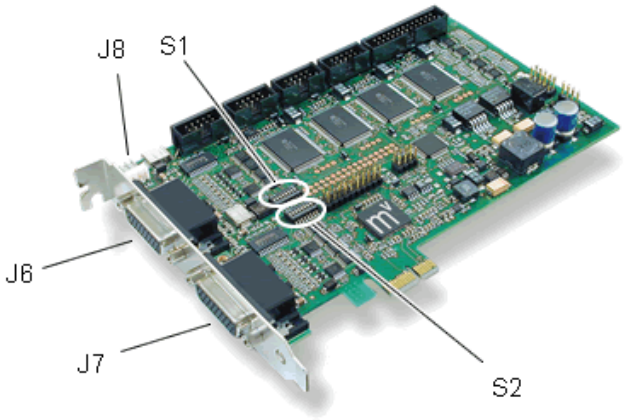


Figure 8: Connectors mvSIGMA-SQe (Rev. 1.0)

Default settings are bold.

1.8.3.1.1 D-Sub HD26BU digital input / output & power (rev. 5.00 and higher)

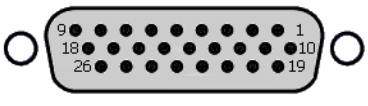


Figure 9: D-Sub HD26BU pin numbering (female)

Pin.	Jack J6	Jack J7
1	12V/0.6A camera power supply	12V/0.6A camera power supply
2	Video 0 / Luminance 0	Video 8 / Luminance 2
3	Video 1 / Luminance 1	Video 9 / Luminance 3
4	Video 2	Video 10
5	Video 3	Video 11
6	DIG-IN 0 / Trigger-In (Decoder 0)	DIG-IN 8 / Trigger-In (Decoder 2)
7	DIG-IN 1 (Decoder 0)	DIG-IN 9 (Decoder 2)
8	DIG-OUT 0 (Decoder 0)	DIG-OUT 4 (Decoder 2)
9	DIG-OUT 1 (Decoder 0)	DIG-OUT 5 (Decoder 2)
10	GND	GND
11	DIG-IN 4 / Trigger-In (Decoder 1)	DIG-IN 12 / Trigger-In (Decoder 3)
12	DIG-IN 5 (Decoder 1)	DIG-IN 13 (Decoder 3)
13	DIG-IN 6 (Decoder 1)	DIG-IN 14 (Decoder 3)
14	DIG-IN 7 (Decoder 1)	DIG-IN 15 (Decoder 3)
15	GND	GND
16	GND	GND
17	GND	GND
18	DigOut VCC ($<$ 24V)	DigOut VCC ($<$ 24V)
19	DIG-IN 2 (Decoder 0)	DIG-IN 10 (Decoder 2)
20	DIG-IN 3 (Decoder 0)	DIG-IN 11 (Decoder 2)
21	Video 4 / Chrominance 0 (Decoder 0)	Video 12 / Chrominance 2 (Decoder 2)
22	Video 5 / Chrominance 1 (Decoder 1)	Video 13 / Chrominance 3 (Decoder 3)
23	Video 6	Video 14
24	Video 7	Video 15
25	DIG-OUT 2 (Decoder 1)	DIG-OUT 3 (Decoder 1)
26	DIG-OUT 6 (Decoder 2)	DIG-OUT 7 (Decoder 2)

1.8.3.1.1.1 Characteristics of the digital I/Os Digital inputs

Input impedance	10 KOhm
Downstream pre-devider	1:4.7
Trigger level	digital adjustable
Max. input voltage	24 V

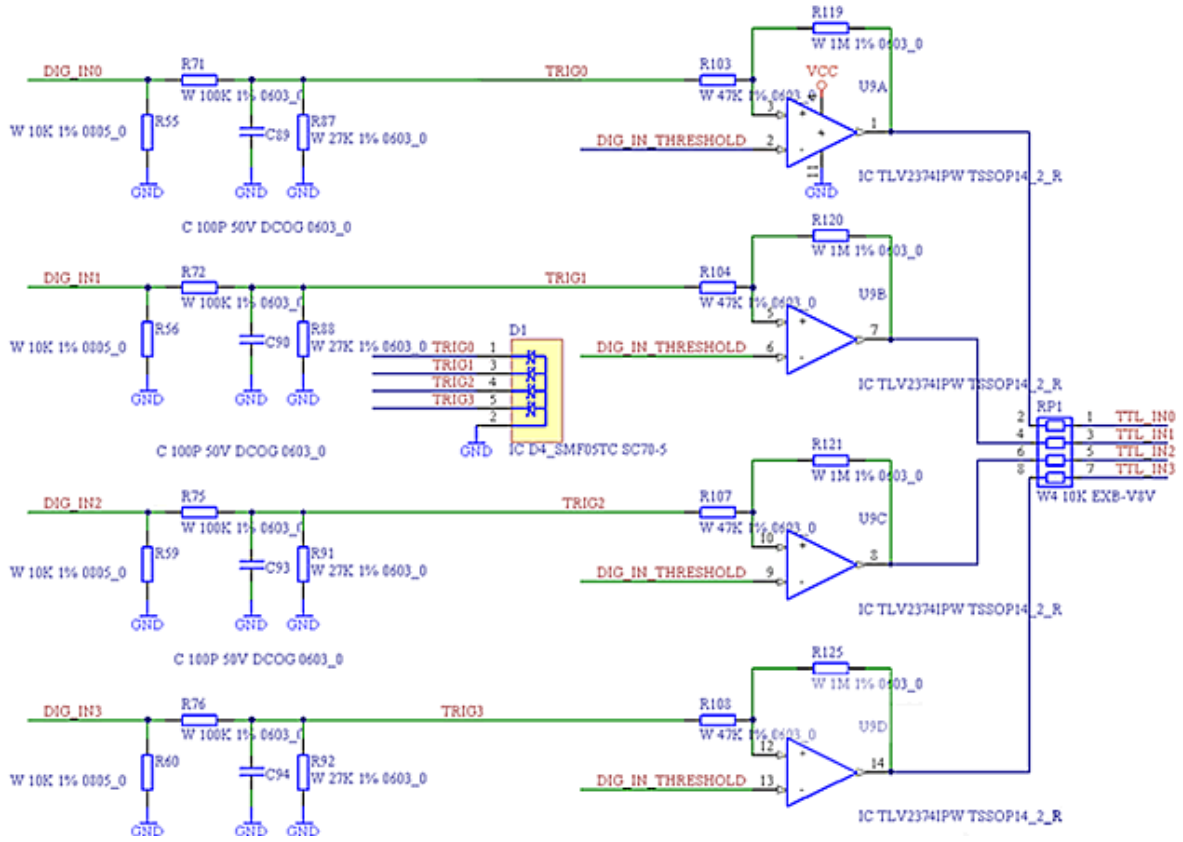


Figure 10: digital input of decoder 0

Digital ouputs

Output voltage	12 V or DigOut VCC
Max. output current	200 mA per pin

The digital outputs are conducted as so called high side drivers. For the default switching voltage the PC's internal supply of 12V is provided. If required this can be increased up to 24V by conducting an external voltage on pin 18.

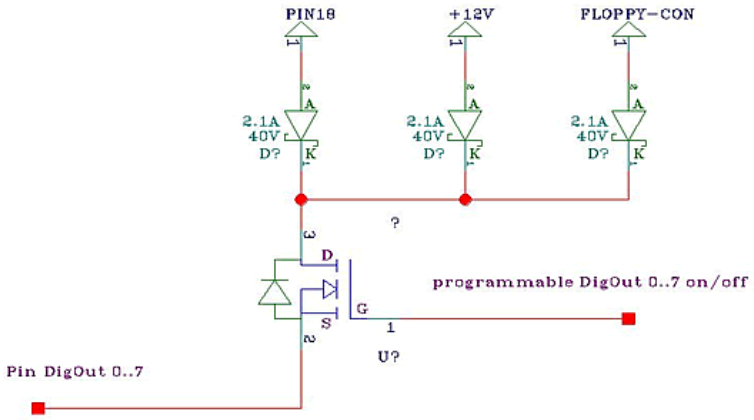


Figure 11: digital output

1.8.3.1.2 J8: Power supply (floppy) You can connect a free power supply cable for floppy drives on connector J8 to increase the available current on the power supply pins on J6 and J7 to 2A.

Figure 11: J8

Pin	Signal
1	+12V
2	Ground
3	Ground
4	Not connected

1.8.3.1.3 Termination of video input (since mvSIGMA-SQ Rev. 2.0 and mvSIGMA-SQe Rev. 1.0)

With switch S1 (above) the termination of the video inputs can be switched on/off. Default adjustment of switch S1 is **on**.

Meaning of switch S1, 75Ohm terminator

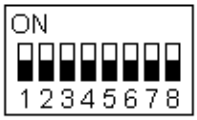


Figure 12: video

Position	Comment
1	Y0
2	Y1
3	Y2
4	Y3
5	Y4/C0
6	Y5/C1
7	Y6
8	Y7

With switch S2 (below) the termination of the video inputs can be switched on/off. Default adjustment of switch S2 is **on**.

Meaning of switch S2, 75Ohm terminator

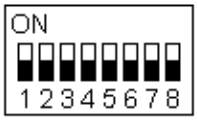


Figure 13: video

Position	Comment
1	Y8
2	Y9
3	Y10
4	Y11
5	Y12/C2
6	Y13/C3
7	Y14
8	Y15

1.8.3.2 Technical specifications

	mvSIGMA-SQ	mvSIGMA-SQe
Video input		
Input signal	Interlaced, monochrome	
	Interlaced, color	
50 Hz	CCIR, PAL, SECAM, S-VHS, Y/C	
60 Hz	RS-170, RS-330, NTSC	
Number of video inputs	16, terminated with 75Ohm by default	
Sync signal	External source	
Resolution		
Digitalisation	768 x 576 (PAL) / 640 x 480 (NTSC)	
Grey scale	8 bit	
Color	24 bit true color	
Memory formats	32 bit RGB, 24 bit RGB, 16 bit RGB, 15 bit RGB, 16 bit YUV packed, 16 bit planar	
Image memory	Host RAM or overlay	
Interface		
Bus	PCI bus	PCI Express x1 bus
Current consumption		
PCI 5V	max. 1.5A	
PCI 12V	max. 0.2A	
PCIe 3.3V		max. 0.15A
PCIe 12V		max. 0.6A
Camera supply	Via PCI: 12V max. 0.7A fused	
	Via additional power plug: 12V max. 1.5V fused	
Environmental conditions		
Ambient temperature	0 up to 50 C	
Storage temperature	-20 up to 70 C	
Humidity	10 up to 90 % non-condensing	
Dimensions		
Length	150 mm	170 mm
Width	106 mm	111 mm

1.9 GUI tools

1.9.1 Introduction

MATRIX VISION provides several convenient tools with graphical user interface to set up and work with their devices. Please find a short list and description below:

1.9.2 wxPropView

With wxPropView it is possible

- to acquire images
- to configure the device, and
- to display and modify the device properties.

1.9.3 mvDeviceConfigure

mvDeviceConfigure can be used to

- to set the device ID
- to update firmware- to disable CPU sleep states(some version of Windows only)

1.9.4 mvIPConfigure

With mvIPConfigure it is possible

- to configure the network behavior of GigE Vision™ devices
- to determine and solve network issues.

1.9.5 mvGigEConfigure

With mvGigEConfigure it is possible

- to install, remove or configure the MATRIX VISION GigE Vision™ capture filter driver.

See also

For further information about the tools, please follow the link to the separate manual describing the GUI tools in great detail on our website: `https://www.matrix-vision.com/manuals/`

1.10 Developing applications using the mvIMPACT Acquire SDK

The `mvIMPACT Acquire SDK` is a comprehensive software library that can be used to develop applications using the devices described in this manual. A wide variety of programming languages is supported.

For C, C++, .NET, Python or Java developers separate API descriptions can be found on the MATRIX VISION website:

- `mvIMPACT Acquire C API`
- `mvIMPACT Acquire C++ API`
- `mvIMPACT Acquire Java API`
- `mvIMPACT Acquire .NET API`
- `mvIMPACT Acquire Python API`

Compiled versions (CHM format) might already be installed on your system. These manuals contain chapters on

- 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 installation packages for Windows and Linux
- a detailed API documentation
- etc.

1.11 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.

1.11.1 Supported interfaces

1.11.1.1 IAMCameraControl

1.11.1.2 IAMDroppedFrames

1.11.1.3 IAMStreamConfig

1.11.1.4 IAMVideoProcAmp

1.11.1.5 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_ACQUIRE_PROPERTYSET

1.11.1.6 ISpecifyPropertyPages

1.11.2 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 or read on how to configure the log-output using **mvDeviceConfigure** in the **"mvIMPACT Acquire GUI Applications"** manual.

1.11.3 Setting up devices for DirectShow usage

In order to be able to access a device through the mvIMPACT Acquire driver stack from an application through DirectShow a registration procedure is needed. This can either be done using mvDeviceConfigure or by a command line tool that is part of the Windows operating system.

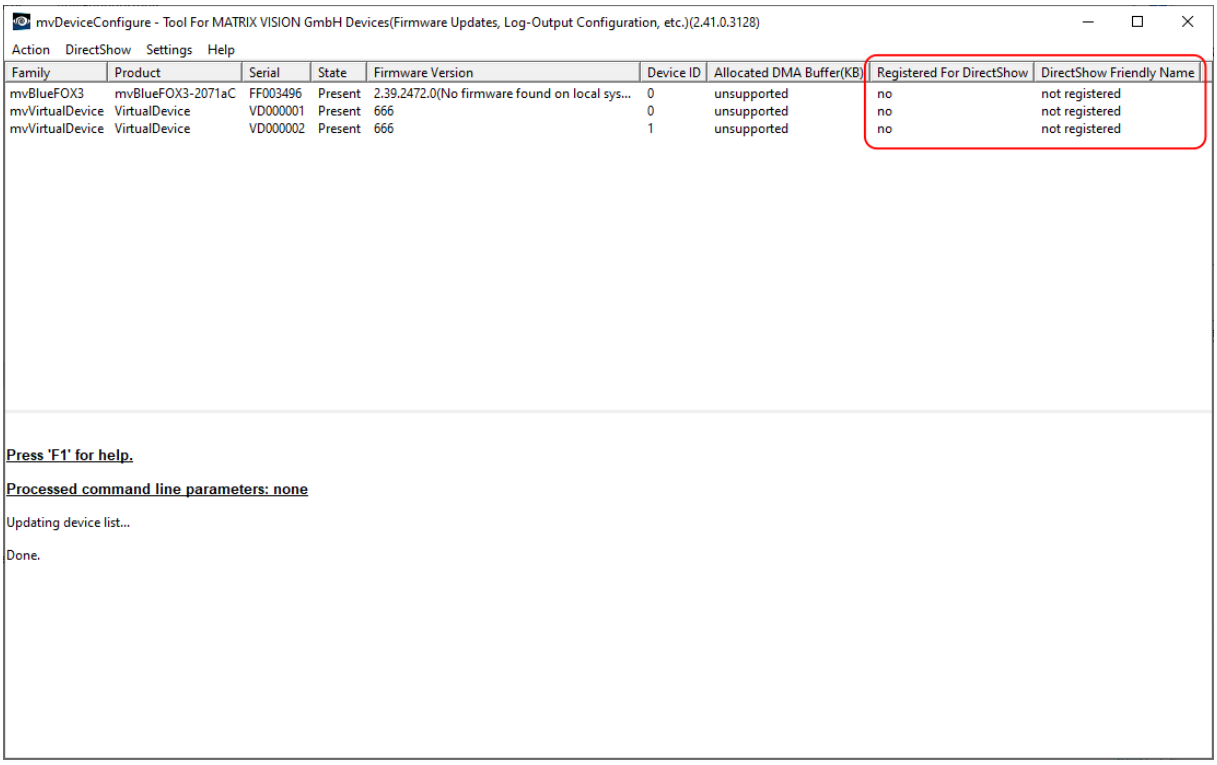
Note

Please be sure to register the MV device for DirectShow with a matching version of **mvDeviceConfigure**. I.e. if you have installed the 32-bit version of the VLC Media Player, Virtual Dub, etc., you have to register devices with the 32-bit version of mvDeviceConfigure (*"C:\Program Files\MATRIX VISION\mvIMPACT Acquire\bin"*, the 64-bit version resides in *"C:\Program Files\MATRIX VISION\mvIMPACT Acquire\bin\x64"*)!

1.11.3.1 Registering devices

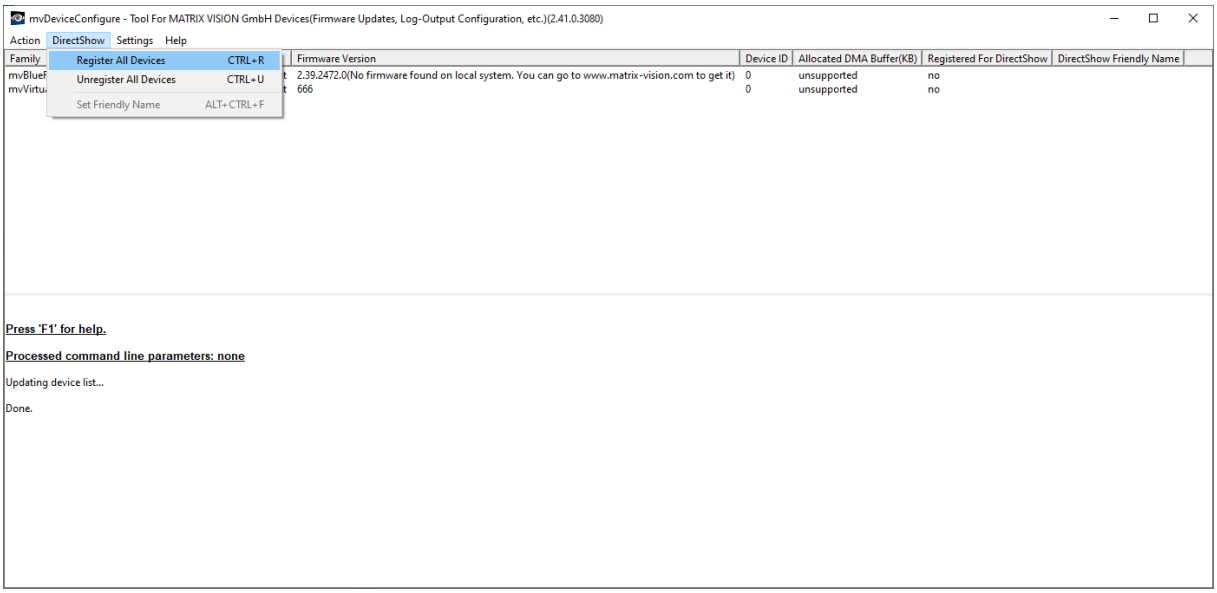
To register all devices currently recognized by the mvIMPACT Acquire driver stack for access with DirectShow the following registration procedure is needed:

1. >mvDeviceConfigure needs to be started(with elevated rights).
   If no device has been registered the application will more or less (depending on the installed devices) look like this.



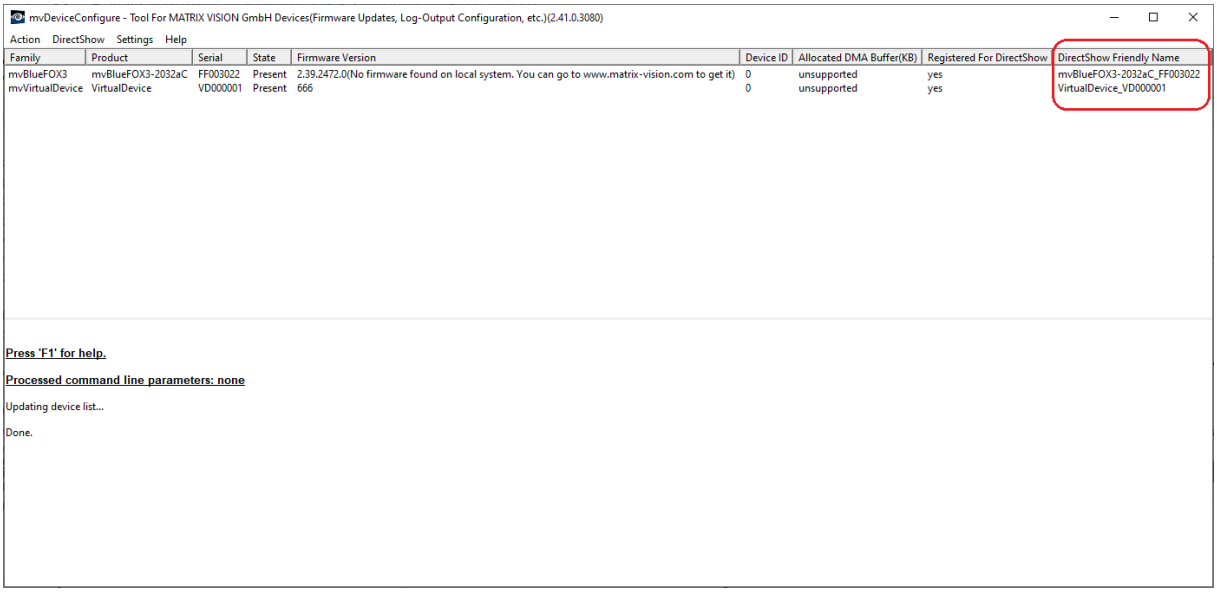
mvDeviceConfigure - After Start

2. To register every installed device for DirectShow access click on the menu item **"DirectShow"** → **"Register All Devices"**.



mvDeviceConfigure - Register All Devices

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 which is displayed in the **"DirectShow Friendly Name"** column.

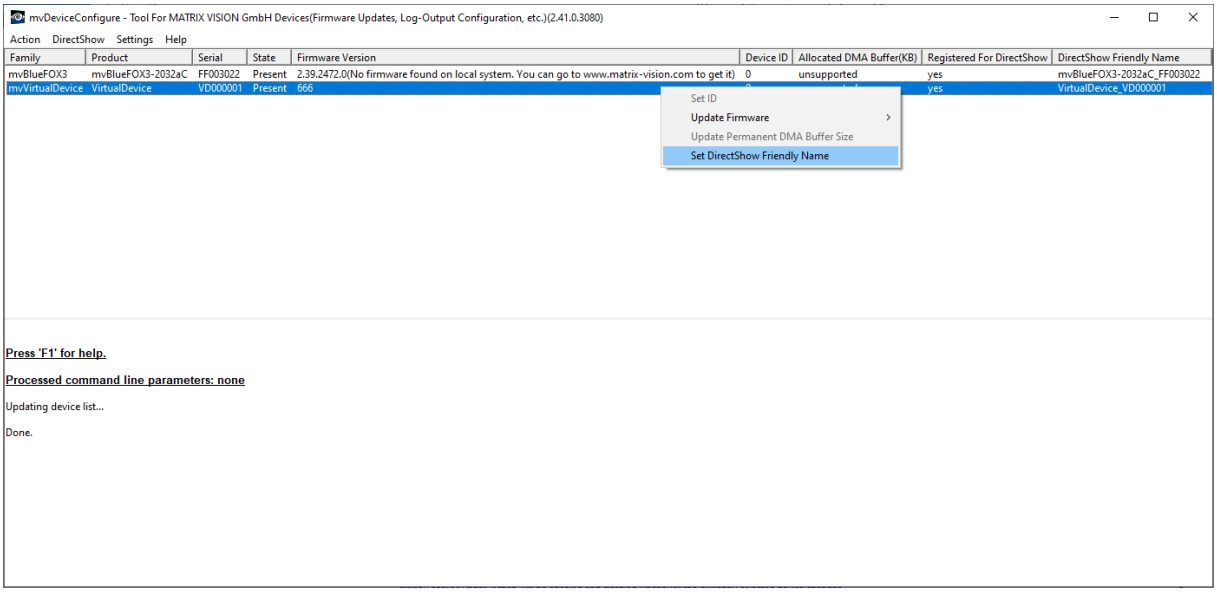


mvDeviceConfigure - All Devices Registered For DirectShow Access

1.11.3.2 Renaming devices

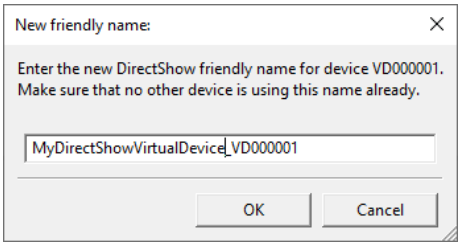
To modify the DirectShow friendly name of a device:

1. mvDeviceConfigure needs to be started(with elevated rights).
2. right-click on the device to rename and select **"Set DirectShow Friendly Name"**:



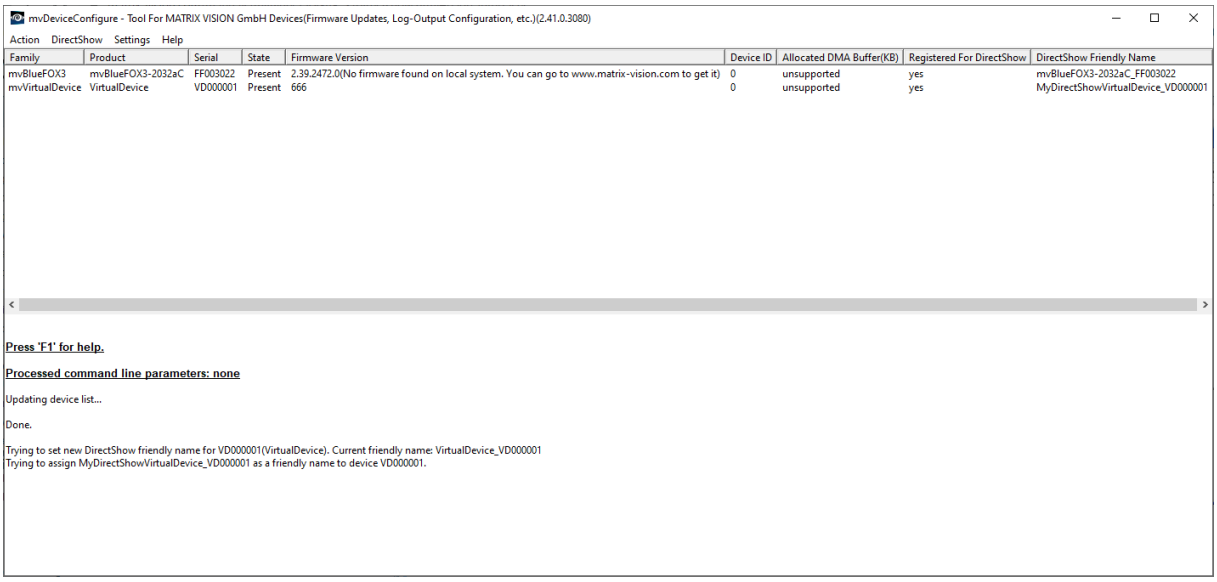
mvDeviceConfigure - Set DirectShow Friendly Name

3. Then, a dialog will appear. Please enter the new name and confirm it with **"OK"**.



mvDeviceConfigure - Dialog For New Name

4. Afterwards the column "DirectShow friendly name" will display the newly assigned friendly name.



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.

1.11.3.3 Using regsvr32

To register all devices currently recognized by the mvIMPACT Acquire driver stack with auto-assigned names, the Windows tool "`regsvr32`" can be used from an elevated command shell.

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) without any user interaction:

```
regsvr32 <path>\DirectShow_acquire.ax /s
```

Unregister ALL devices that have been registered before without any user interaction:

```
regsvr32 <path>\DirectShow_acquire.ax /u /s
```

1.12 Glossary

A/D reference	Upper threshold of video signal to be digitized. All values above this limit value are digitized to 255. Increasing the reference level results in contrast deterioration and vice versa.
ADC	Analog to digital converter (A/D converter)
Resolution	Number of pixels (horizontal x vertical)
Base address	Starting address from which the memory or register are inserted.
Image refresh rate	Number of transferred images per second. Normally specified in Hz (e.g. 70 Hz)
Bpp	Bits per pixel
Bus	A group line via which the various parts of the computer communicate with one another.
CCIR	Comité Consulatif International of the Radio Communications European video standard for 50 Hz gray scale.
Clamp signal	Clamp signal means, that a AC coupled video signal is clamped on the porch to get a signal transfer with less noise and independent from the d.c. voltage portion.
CPU	Central processing unit
DAC	Digital to analog converter (D/A converter)
Defaults	Standard system settings
DIO	Digital inputs and outputs
DIP switch	Dual inline package (housing design)
External trigger	External event used to initiate image capture.
False colors	Colors are assigned to gray scale via a look-up table. This allows even small gray scale differences can be displayed clearly.
Field	All odd lines of a field (odd field) or all even lines of a field (even field) of an interlaced video image.
Frame grabber	Here: PC plug-in card for digitization and storage of video images.
Horizontal sync	The portion of the analog signal which specifies the line end of the video signal.
Host	Here: the PC
Interlaced	Interlacing method; conforming to the television standard, this method involves acquiring two fields in succession (all odd lines, all even lines) and combining them to create a frame. The result is greatly reduced flicker during on-screen display.
Interrupt	Interrupt signal sent to the processor. The program currently running is interrupted and a predefined function is executed.
ISR	Interrupt service routine
IRQ	Interrupt request
Look-up table	Table of assignments. Here, new gray scale or colors are normally assigned to gray scale. Look-up tables can, however, also be used for any other math functions.
LSB	Least significant bit
LUT	Look-up table
Monochrome	A single-color (black and white) image
MSB	Most significant bit
Non-interlaced	Image acquisition and output line by line
NTSC	National Television Standard Code. US video standard for 60 Hz colors.
Overlay	Image memory for outputting text and graphics via the video monitor.
PAL	Phase alteration line; 50 Hz video standard for color.
Pixels	Picture element
PoCL	Power over CameraLink - The cameras are powered over CL cable and therefore need no additional power supply. The mvHYPERIONs which supports PoCL, are **"Switchable PoCL frame grabbers"** as described in the CameraLink™ specification. This means that both camera and cable have to support PoCL otherwise Pin 1 and Pin 26 of the CL connectors act like internal shields.
Pseudo colors	Display of gray scale images in false colors. A corresponding color is assigned to a specific gray scale value.

Square pixels	Square-shaped pixels (height-width ratio 1:1)
RS170	US video standard for 60 Hz b/w colors
TFT display	Thin film transistor display
True color	24-bit true color; 16.7 million colors
Vertical sync	Synchronization pulse in video signal for field end recognition.
Zero signal	The zero signal was needed with the old frame grabbers, to calibrate the analog/digital converter (ADC) (signal and parameter aren't important anymore).