DECTRIS
detecting the future

User Manual

PILATUS3 R/S/X Detector System

Document Version v1.2.0

DECTRIS Ltd.
5405 Baden-Daettwil
Switzerland
www.dectris.com

Document Version v1.2.0
© Copyright 2020
DECTRIS Ltd.

CONTENT

DOCUMENT HISTORY

Current Document

Table 1: Current Version of this Document

Version	Date	Status	Prepared	Checked	Released
v1.2.0	2020-07-09	release	LG	AD	LG

Changes

Table 2: Changes to this Document

Version	Date	Changes
v1.2.0	2020-07-09	New Linux Distribution
v1.1.3	2020-01-16	Low Energy Calibration Option
v1.1.2	2019-12-03	New Server J.
v1.1.1	2019-10-03	New Server O.
v1.1.0	2019-07-09	Update of corporate design of PILATUS R, S and X series.
v1.0.0	2019-06-28	First Release.

1. GENERAL INFORMATION

1.1. Contact and Support

Address:	DECTRIS Ltd. Taefernweg 1 5405 Baden-Daettwil Switzerland
Phone:	+41 56 500 21 02
Fax:	+41 56 500 21 01
Homepage:	http://www.dectris.com/
Email:	support@dectris.com

Should you have questions concerning the system or its use, please contact us via telephone, e-mail or fax.

1.2. Explanation of Symbols

Caution #0

Caution blocks are used to indicate danger or risk to equipment.

Information #0

Information blocks are used to highlight specific information.

1.3. Warranty Information

Should your detector require warranty service, contact DECTRIS for further information. Before shipping the system back, please contact DECTRIS to receive the necessary transport and shipping information. Make sure that the original packaging is used when returning the system.

Caution	#1
Do not ship the system back before you receive the necessary transport and shipping information.	

When returning the detector system for repair, be sure to fill out and include the service form at the back of this document to provide the support division with the necessary information.

1.4. Disclaimer

DECTRIS has carefully compiled the contents of this manual according to the current state of knowledge. Damage and warranty claims arising from missing or incorrect data are excluded.

DECTRIS bears no responsibility or liability for damage of any kind, also for indirect or consequential damage resulting from the use of this system.

DECTRIS is the sole owner of all user rights related to the contents of the manual (in particular information, images or materials), unless otherwise indicated. Without the written permission of DECTRIS it is prohibited to integrate the protected contents in this publication into other programs or other websites or to use them by any other means.

DECTRIS reserves the right, at its own discretion and without liability or prior notice, to modify and/or discontinue this publication in whole or in part at any time, and is not obliged to update the contents of the manual.

2. SAFETY INSTRUCTIONS

Caution	#2
Please read these safety instructions before operating the detector system.	

- Before turning the power supply on, check the supply voltage against the label on the power supply. Using an improper main voltage will destroy the power supply and damage the detector.
- Power down the detector system before connecting or disconnecting any cable.
- Make sure the cables are connected and properly secured.
- Avoid pressure or tension on the cables.
- The detector system should have enough space for proper ventilation. Operating the detector outside the specified ambient conditions could damage the system.
- The detector is not specified to withstand direct beam at a synchrotron. Such exposure will damage the exposed pixels.
- Place the protective cover on the detector when it is not in use to prevent the detector from accidental damage.
- Opening the detector or the power supply housing without explicit instructions from DECTRIS will void the warranty.
- Do not install additional software or change the operating system.
- Do not touch the entrance window of the detector.

3. SYSTEM DESCRIPTION

3.1. Components

The PILATUS3 detector system consists of the following components:

- PILATUS3 detector
- Power supply for the detector[1]
- Detector control unit
- Thermal stabilization unit[1]
- Pilatus Processing Unit (PPU) [1]
- Accessories
- Documentation

3.2. Hybrid Photon Counting (HPC) Technology

3.2.1. Basic Functionality

DECTRIS X-ray detectors provide direct detection of X-rays with optimized solid-state sensors and CMOS readout ASICs in hybrid pixel technology. Well-proven standard technologies are employed independently for both the sensor and the CMOS readout ASIC. The X-ray detectors operate in single-photon counting mode and provide outstanding data quality. They feature very high dynamic range, zero dark signal and zero readout noise and hence achieve optimal signal-to-noise ratio at short readout time and high frame rates. Large-area detectors with dedicated active areas are built of multiple identical modules using a modular system concept.

Key Advantages

- Direct detection of X-rays
- Single-photon counting
- Excellent signal-to-noise ratio and very high dynamic range (zero dark signal, zero noise)
- Low-energy X-ray suppression (energy resolution by single energy threshold)
- Short readout time and high frame rates
- Shutterless operation
- Modular detectors enabling multi-module detectors with large active area

The PILATUS3 hybrid pixel detector is composed of a sensor, a two-dimensional array of pn-diodes processed in a high-resistivity semiconductor, connected to an array of readout channels designed in advanced CMOS technology (figure 3.1. Each readout channel is connected to its corresponding detecting element through a microscopic indium ball, with a typical diameter of 18 µm. This connection process is called ‘bump bonding’.

1 Some systems might be delivered without this component. Please consult the Technical Specifications for more information.

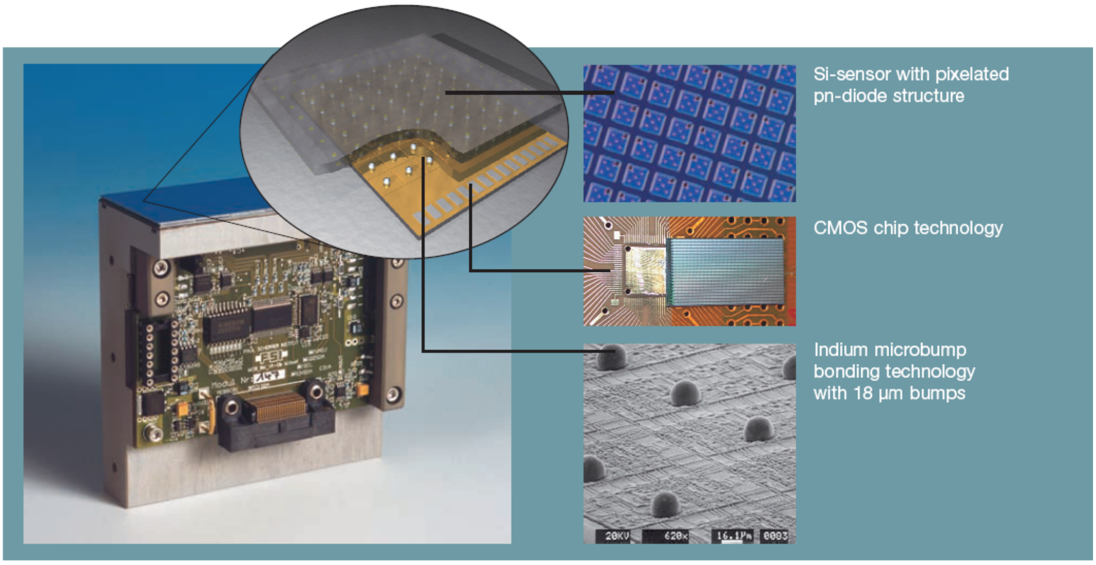

Figure 3.1: The PILATUS3 detector module, the basic element of all DECTRIS Ltd. area detector systems

The quantum efficiency depends on thickness of the silicon sensors, which are available for a wide energy range. The count rate is greater than 1×10^7 photons/s/pixel, enough to perform many experiments using the high flux of modern synchrotron light sources. However, the detector cannot withstand a direct synchrotron beam.

3.2.2. Readout Chip

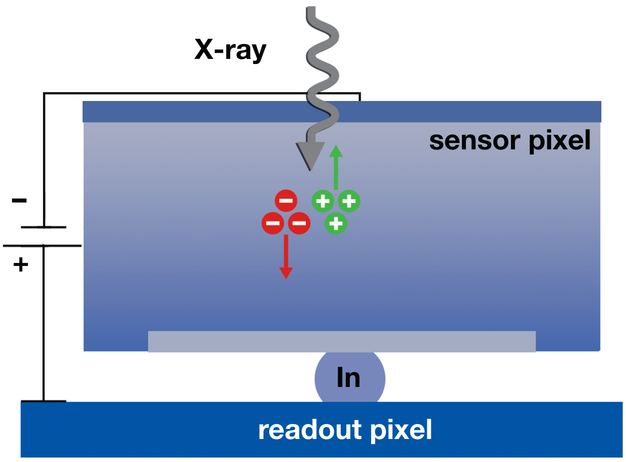

Figure 3.2: Principle of Direct Detection

The great advantage of this approach is that standard technologies are used for both the silicon sensor and the CMOS readout chips, which guarantees highest quality. Both processes are optimized separately, as the best silicon substrates for X-ray detection and for high-speed/high-quality electronics are very different. Moreover, the small size of the pixel and the interconnection results in a very low capacitance, which has the beneficial effect of reducing the noise and power consumption of the pixel readout electronics.
X-ray data collection can be improved with detectors operating in single photon counting mode. A hybrid pixel which features single photon counting comprises a charge-sensitive pre-amplifier (CSA), which amplifies the signal generated in the sensor by the incoming X-ray, and a comparator (Comp), which produces a digital signal if the incoming charge exceeds a pre-defined threshold. The comparator feeds a 20 bit counter, which then leads to completely digital storage and noiseless readout of the number of detected X-rays in each pixel (see figure 3.3).

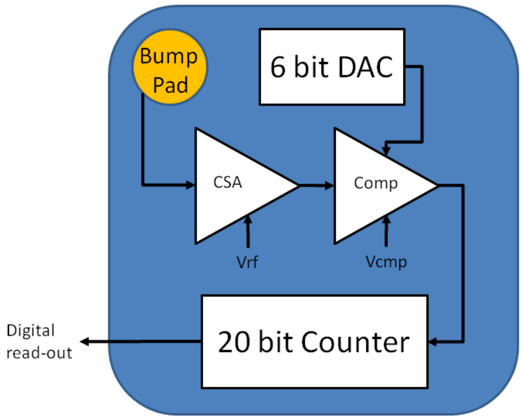

Figure 3.3: Design of the Photon Counting Circuit in each Pixel

The fundamental unit of the DECTRIS detectors, the module, consists of a single fully depleted monolithic silicon sensor with an 8 x 2 array of CMOS readout chips bump-bonded to it. Each sensor is a continuous array of 487 pixel x 195 pixel = 94 965 pixel without dead areas and covers an active area of 33.5 mm x 83.8 mm = 2807.3 mm^2. The readout chips are wire-bonded to the underlying print, which is glued to the mounting bracket. Together with its readout control electronics the sensor with readout chips forms the complete module.

3.2.3. Instant Retrigger

The PILATUS3 X-ray features an instant retrigger capability, improved high-rate counting performance. Instant retrigger capability results in non-paralyzable counting and allows for enhanced count-rate correction. In addition, the PILATUS3 detectors features various enhancements such as counter overflow handling, improved pixel uniformity and reduced crosstalk.

Instant retrigger technology with adjustable dead time is a photon counting imaging method that results in non-paralyzable counting and achieves an improved high-rate counting performance. In a conventional single-photon counting X-ray detector, the charge pulses generated by impinging photons are counted by digital circuits. Simultaneously generated pulses can pile up and as a result, photon counts can be lost. At high photon fluxes, pulse pile-up significantly affects the observed count rate and can lead to complete paralyzation of the counting circuit. In the first generation series of single-photon counting hybrid pixel X-ray detectors "PILATUS3 ", count rate correction is applied in order to compensate for the counting loss at high count rates. However, counter paralyzation limits the maximum usable count rate. In PILATUS3 detectors, the instant retrigger technology re-evaluates the pulse signal after a predetermined dead-time interval after each count and is able to retrigger the counting circuit should a pulse pile-up occur. The dead time interval is adjustable and is equivalent to the width of one single photon pulse. This results in non-paralyzable counting and allows for enhanced count-rate correction so as to achieve improved data quality at high count rates.

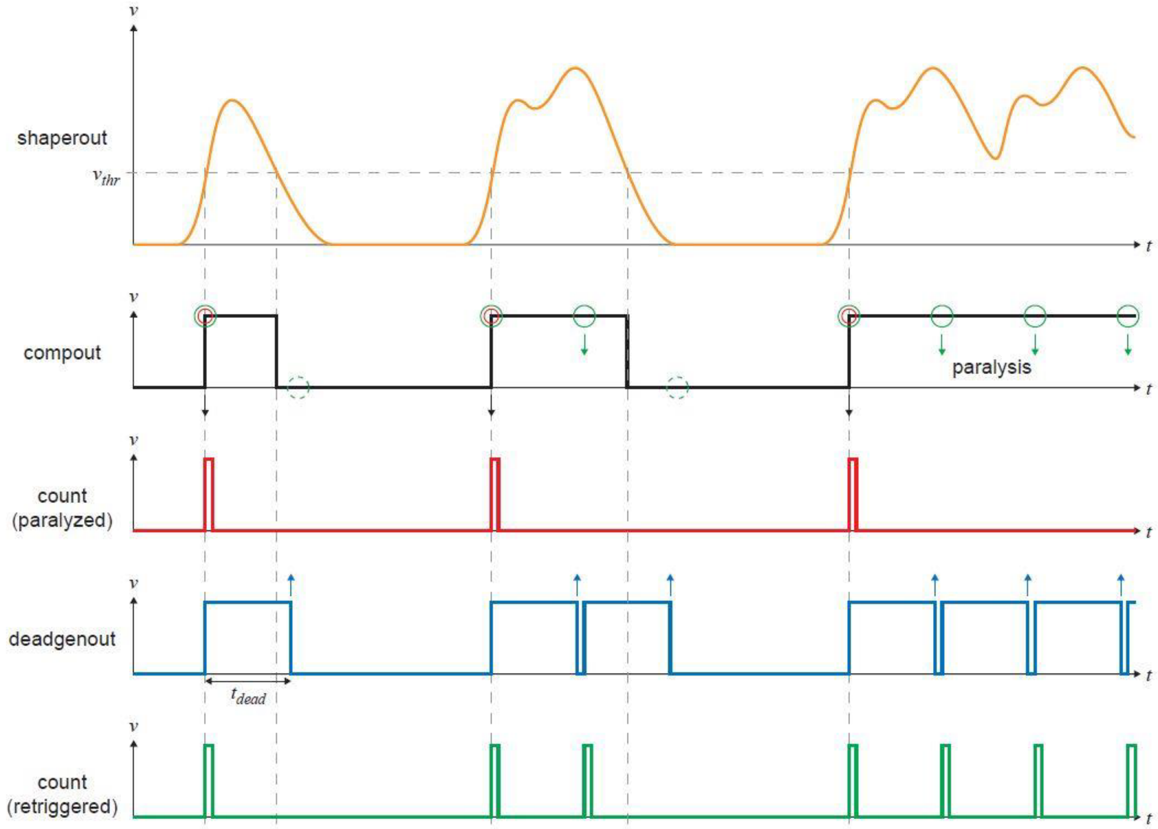

Figure 3.4: Signal Waveforms Illustrating Instant Retrigger Technology

The principle of instant retrigger technology is illustrated in figure 3.4. The first diagram shows the signal pulses generated by impinging photons and the effective discriminator threshold level for single photon counting. The pulse signal includes a series of one single pulse, a pile-up of two pulses and a pile-up of multiple pulses. The second diagram shows the corresponding digital discriminator output signal that triggers the counting circuit. The third diagram shows the corresponding counts being registered by a conventional single photon counting X-ray detector, clearly illustrating that counts are lost in the case of pulse pile-up and that this can lead to paralyzation. The fourth diagram shows the respective dead-time generator output signal provided by a single photon counting X-ray detector with instant retrigger technology. Here, a predefined dead time interval is started whenever a count has been registered. The fifth diagram shows the corresponding counts being registered, including potential retriggering of the counting circuit after the dead-time interval after each count. This clearly illustrates that pulses are counted more accurately in the case of pile-up and that counting is non-paralyzable.

3.3. Software

The PILATUS3 detector system is controlled via Camserver.

3.3.1. Overview of Camserver

Camserver is a freestanding program that controls the PILATUS3 detector and provides a simple user interface for "atomic" (single function) commands. It is intended to provide a minimal, but fully functional, low level interface to camera hardware. On invocation, Camserver takes one optional command-line argument, the path to its resource file, by default called camrc. Camserver will also use the same path to open its debugging file, camdbg.out. For standard operation the argument does not have to be defined.

A major feature of Camserver is to accept TCP socket connections from a high level controller, which can provide high level services to this or other cameras. (see chapter 6 and 13 for more details). The interface is a simple text-based message passing system. Images - the ultimate product of a working area X-ray detector - do not pass through the socket interface, but are written to a configurable location on the detector server's filesystem (e.g., a NFS mount) where any program can access them.

3.3.2. Configuration Directories and Files for Camserver

In the default setup, all necessary data for use of the PILATUS3 detector system is in the directories:

Table 3.1: Camserver Directories

Directory	Description
/usr/local/bin/	Camserver executable
/etc/dectris/	Camserver configuration file
/var/tmp/dectris/camstat/	Camserver status files
/var/log/dectris/	Camserver log files
/var/local/lib/dectris/config/cam_data/	Detector configuration files
/var/local/lib/dectris/config/calibration/	Detector calibration data
/home/det/images	Default image directory

The configuration files from the detector and Camserver:

Table 3.2: Camserver Directories

File Name	Description
/etc/dectris/camrc	Camserver configuration containing all details and links to further configurations files
/var/local/lib/dectris/config/cam_data /camera.def	Detector configuration file containing the information about the detector: serial number, sensor thickness, rate correction.
/var/local/lib/dectris/config/cam_data /pidet.set	Contains the start procedure for Camserver. 1

Information	#1
	Please note that this startup configuration is only valid until a high level controller sets a new one (e.g. see section 3.3.4 and section 5.1).

3.3.3. Overview of TVX

TVX is a free, open source, data acquisition and control software suite tailored to X-ray science. TVX is an attempt to provide a flexible user interface that is easily adapted to control a broad range of 2D X-ray detectors as well as a powerful collection of analysis tools.
The suite operates by distributing the tasks of data analysis and hardware control between two separate programs. TVX contains a user interface and analysis tool suite. TVX communicates over a TCP/IP connection to Camserver.

3.3.4. Configuration Files for TVX

In the default setup, all data for the use of TVX with the PILATUS3 detector system are in the directory /home/det/tvx. Important configuration files are listed in the following table. The directory is given relative to the default path /home/det/tvx/.

Table 3.3: TVX Configuration Files

File	Directory	Description
tvxrc	./	Configuration file for TVX
default.gl	config	Startup glossary of TVX
det_spec.gl	config	Detector specific parameters for TVX
user.gl	config	User short cuts for TVX
startup.gl	config	Final startup glossary executing detector self-tests (see section 5.1 for more details)

3.3.5. Compatibility to Earlier Configurations of Camserver and TVX

Earlier models of PILATUS3 detectors contained all configuration files below the /home/det/p2_det directory. To preserve compatibility the original file structure is mapped with links. The following table shows the relationship between links and physical directories on the filesystem of the detector server. Please note that not all files within the directories are mapped one-to-one.

Table 3.4: Relationship between Link and Physical Directory

Physical Directory	Symbolic Link
/home/det/tvx/	/home/det/p2_det/
/home/det/images	/home/det/p2_det/images/
/var/local/lib/dectris/config/	/home/det/p2_det/config/
/var/tmp/dectris/camstat/	/home/det/p2_det/config/camstat/
/var/log/dectris/	/home/det/p2_det/camdbg_logs/

4. QUICK START GUIDE

Before you turn on the system, make sure you have read this user documentation, the Technical Specifications, and set up the detector accordingly.

- Turn on the detector server.
- Log in procedure: Enter User and password (provided on a separate sheet in your user documentation folder)
- Open a shell.
- Change to the TVX directory. All following paths are given relative to this!
  cd/home/det/tvx
- Run TVX by typing
  ./runtvx

After the initialization (∼ 20 s) you should get the following screen:

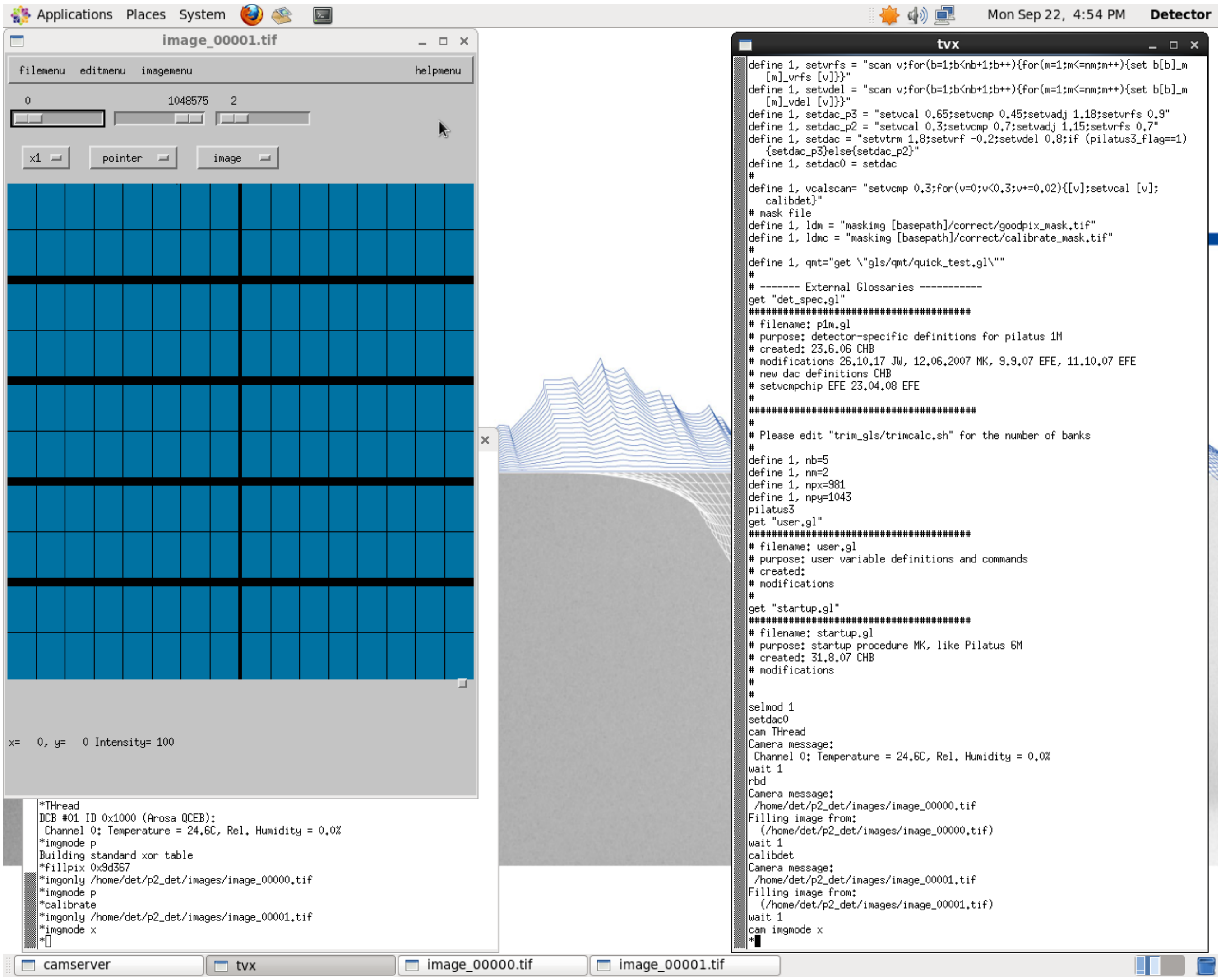

Figure 4.1: Startup Screen after Executing ./runtvx

- The command ./runtvx is a shell script, which starts the program Camserver (window on the left side of figure 4.1) and TVX (window on the right side of figure 4.1) as well as manages all log files.
- During startup the detector sets several parameters defined in the startup scripts (details in section 5.1).

- Once two images (green and blue) appear on the screen the detector is ready to operate. In figure 4.1 only one (blue) image is visible since it is above the other (green) one. Both terminals (TVX and Camserver) have their prompt, indicated by an asterisk symbol (*).
- The detector is now ready for operation with Copper (8 keV) radiation. If you would like to change to other energies, please see chapter 7.
- To take an image you can type expose 10 in the TVX window, where 10 is the exposure time in seconds.

Further information:

- Details on how to control the detector from a specific environment (e.g. at a synchrotron) can be found in section 5.2.
- Details on how to trigger the detector with an external signal can be found in section 6.3.
- For information concerning the dead pixels and gaps please see chapter 8.

5. CONTROL THE DETECTOR

The PILATUS3 detector can be controlled from the delivered detector server or via the socket connection. Below these ways are described in detail.

5.1. From the Detector Server (Stand-Alone Operation)

To use the detector as a stand-alone system follow the instructions in the Quick Start Guide (chapter 4).

The runtvx command is a shell script which starts the actual programs: Camserver and TVX. It also handles the log-files. During the Camserver startup a connection to the detector is established and verified. TVX automatically connects to Camserver and configures the detector via the startup procedure defined in default.gl. This is a so-called glossary file (*.gl), which is a convention used for files which are processed by TVX. This default.gl file (stored in /home/det/tvx/config) makes several definitions (short cuts), which can be used further on in TVX.

For example the exposem command makes an endless loop and writes images in a temporary file (see also chapter 11). This is a simple live-view tool of TVX.

At the end of the startup glossary three more glossaries are called. The first one (det_spec.gl) loads detector specific parameters such as camera size and number of modules. The second one (user.gl) is used to enter user short cuts if you define your own ones. The last one (startup.gl) sets voltages on the actual X-ray sensitive elements and carries out a test of the digital part and the analog part of each pixel. The last two tests are done with the TVX definitions rbd (read back detector) and calibdet, respectively. The results are shown in a green (digital test with 1000 counts loaded into the counter of each pixel) and a blue (analog test with 100 simulated pulses fed into each pixel) image. These two images show up automatically at the end of the startup procedure and verify the full functionality of the detector.

5.2. From a Specific Environment (via Socket Connection)

In the previous section the stand alone operation is shown. However, often there is a need to integrate the detector into an existing environment.

The PILATUS3 detector can be easily integrated into any environment. To do this, one has to send commands through a socket connection to Camserver. Any client can connect to Camserver via a socket connection, and issue plain text commands. However, only the first connection will get full control and can execute commands. All following connections will only have read access. The command syntax (see chapter 13) over the socket is identical to the syntax to be typed directly in the Camserver window. Thus, direct typing is helpful for testing.

The reply from Camserver (acknowledgement) consists of a command index number, followed by a space and either ”OK” or ”ERR”, followed by another space and possibly a message. The acknowledgement arrives after the requested action is completed, typically in 1-2 ms; some commands, such as trimming commands (e.g. SetCu), take longer, especially for a big detector. All acknowledgements end in 0x18 (ASCII ’CAN’) without a newline; there may be internal newlines in long messages. Since there is no terminating linefeed, MS Windows sockets must be opened in binary mode; this is not a consideration for UNIX-like systems.

Because of the socket connection protocol, the camera hardware and server can reside on a different machine from the high-level controller.

Camserver implements a token mechanism to prevent more than one outside process from having control over the hardware. The Camserver window has full control at all times.
There is a debug facility to help with setting up the interface. If you type dbglvl 5, the file camdbg.out will contain many messages, including the exact contents of socket messages. Be sure to set dbglvl 1 (the default) before doing real work, else camdbg.out can grow without limit. If you encounter problems with the detector, a run with dbglvl 6 reproducing the error can be helpful for diagnosis. Simply capture camdbg.out and send it including a description of the issue to DECTRIS support.

The Camserver program of the PILATUS3 detector provides a simple to use interface for either EPICS or SPEC. Several clients for these protocols have been written at the Swiss Light Source (SLS) at the Paul Scherrer Institute (PSI) and by Mark Rivers of the University of Chicago:
http://cars.uchicago.edu/software/epics/areaDetector.html//

5.2.1. Integrating the PILATUS3 Detector into your Environment

1. If needed, change the hostname to be compatible with the local network.
2. Set up the detector on the network, if needed. Note that the detector does not require an external network.
3. Configure Camserver, and the client TVX, as needed. Probably the defaults will be adequate, but many parameters can be adjusted in camrc. For further details see section 3.3.2.
4. Start the detector by ./runtvx in the /home/det/tvx directory.
5. If you are using your own client (see section 5.2.2) for Camserver (e.g., SPEC or EPICS) disconnect TVX (type disconnect in TVX). This can be done automatically in the startup script after the test images are shown. Connect your client and begin issuing commands (see chapter 13).
6. Alternatively to points 4, 5 it also possible to start only Camserver with the command ./camonly in the /home/det/tvx directory.

5.2.2. Testclients

The detector can be controlled from any client, which opens a socket connection to Camserver.

Caution #3

! Make sure TVX is disconnected (type disconnect in TVX) before you connect your own client.

The simplest client perhaps is telnet. It is possible to test it on the detector server itself by executing telnet localhost 41234. Here localhost is the "IP" of the detector and 41234 the port where Camserver is listening to. A basic test client written in C can be found under chapter 14.

For more information please contact DECTRIS at support@dectris.com.

6. HOW TO USE THE PILATUS3 DETECTOR THROUGH CAMSERVER

Camserver is a completely freestanding program that controls the detector and provides a simple user interface for atomic (single function) commands. It is intended to provide a minimal, but fully functional, low level interface to camera hardware.

To get help on the Camserver commands use the help facility of TVX (see chapter 11).

All commands in Camserver (unlike in TVX) can be abbreviated to the minimum number of letters that make the command unambiguous; below we use only the full names for clarity. Commands are not case-insensitive, but pathnames are case-sensitive (same for TVX). A full list of commands can be found in chapter 13.

Information	#2
We recommend that full command names are used for clarity.	

6.1. Main Commands

Table 6.1: Main Commands

Command	Description
Exposure [filename]	Make an exposure with defined variables (e.g. exposure time and exposure period, see 6.1.1 and 13 for more details). The format of the file is determined from the supplied extension. The file is stored relative to the path defined by the imgpath unless an absolute path is given.
ExtTrigger [filename]	Start exposure with defined variables after receiving an external trigger and store images in [filename].
ExtMTrigger [filename]	Start multiple exposures with defined variables after receiving multiple external triggers and store images in [filename] (see section 6.3.2).
ExtEnable [filename]	Start exposure defined by external signal and store images in [filename].
help exposurenaming	Type this in the TVX window for a discussion of how exposure series are named.

6.1.1. Variables

The following variables can be viewed just by typing them; all times are in seconds.

Table 6.2: Main Commands

Variable	Description
NImages [N]	Query or set the number of images in a sequence.
ExpTime [time]	Query or set the exposure time (10^{-6} to 10^{6} s).
ExpPeriod [time]	Query or set the exposure period for serial exposures. The exposure period must be at least 0.95 ms longer than the exposure time.
imgpath [path]	Query or set the default image path.

Table 6.2: Main Commands - continued

Variable	Description
Delay [time]	Query or set the external trigger delay. This is the time to wait after the external trigger before taking the first image. The delay may not be greater than the exposure period.
nexpframe [N]	Number of exposures per frame. This is a so called multi exposure mode. nexpframe sets the number of exposures before the detector is read out e.g. nexpframe 3 exposes the detector 3 times before reading out an image of the 3 combined exposures. See point 6.3.4 for more details.

The usual way is to set all mentioned variables and then execute a command from the section above.

6.2. Image Formats

Due to the high dynamic range of 20 bit of the PILATUS3 detectors, images are stored as 32 bit (signed) integers. These images can be viewed and analyzed with TVX or other image viewers. Many viewers do not support 32-bit TIFF files; however, these images may be read in IDL, MATLAB or with the image viewer ALBULA provided by DECTRIS.

The default image file type for TVX is set in tvxrc; however, any file type can be specified explicitly. Camserver has no default, so the file type must be specified explicitly for each exposure.

TVX supports the following image formats.

Table 6.3: Image Formats

Format	Description
.cbf	Crystallographic binary format
.tif	32-bit signed TIFF files
.edf	ESRF data format
.img	Raw data format
(no extension or misspelled)	Raw data format

Of these four image formats .tif, .edf, and .img are uncompressed and .cbf is the only (lossless) compressed format. The .cbf and .edf headers are pure text and therefore human-readable. The .tif header is encoded, so only its comment section is human-readable.

- The .cbf format is the only compressed image format and is recommended in situations where I/O bandwidth may be limited. The byte-offset-compression algorithm usually reduces the file size to a quarter of the .tif image. The initial header data is followed by a PILATUS3 section. As for the .tif format, this section can be extended by the user with additional items (see chapter 9). Just before the actual compressed data there is additional information for the .cbf format. The full .cbf header can be defined by the user through the commands described in chapter 9. More details to the .cbf format can be found on http://www.bernstein-plus-sons.com/software/CBF/.
- The .tif file contains the standard TIFF header including a PILATUS3 section (comment section) followed by the uncompressed data. The PILATUS3 section contains at least the following items (example):

    # Silicon sensor, thickness 0.320 mm

    # Exposure_time 1.000 000 s

Exposure_period 1.000 950 s
Tau = 0 s
Count_cutoff 1,208,517 counts
Threshold_setting: 5 keV
Gain_setting: autog (vrf = 1.000)
N_excluded_pixels = 7
Excluded_pixels: badpixel_mask.tif
Flat_field: FF_p3-0106-500Hz_E10000_T5000_vrf_m0p100.tif
Trim_file: p3-0106-500Hz_E10000_T5000.bin
Image_path: /home/det/images/
Ratecorr_lut_directory: Continuous
Retrigger_mode: 1

This list can be extended by the user with additional items, as described in chapter 9. The TIFF header is of a fixed length of 4,096 bytes, so it is possible to make a file reader by stripping off the header and reading the row-major block of data.

- The .edf file starts with the header according to the ESRF data format followed by the uncompressed data.
- The .img file contains only the uncompressed data and no header.

6.3. External Triggering

External Triggering can be used to start the exposure of the detector. The trigger should be provided on the detector's ExtIn input (compare Technical Specifications document). The following three different trigger modes are available:

- External Trigger: triggers a predefined series of commands after the detector receives a rising edge (section 6.3.1).
- External Multitrigger: triggers each exposure with an external pulse, but times the exposures using the internal timer (section 6.3.2).
- External Enable: gates the detector's images on the positive signal applied to the external enable input of the detector (section 6.3.3).

The last two modes have the option of the multiple exposure functionality (section 6.3.4).

Information	#3
Make sure that you define ALL settings required for the desired trigger mode as described below.	

There is no timeout after executing a trigger command in Camserver: the detector will wait until a trigger signal arrives. This state can only be interrupted by transmitting a K command to Camserver over the socket connection.

6.3.1. External Trigger Mode

The external trigger mode is exactly the same as an exposure except that an external pulse is used rather than the enter key on the keyboard.

External trigger mode is activated with the Camserver command ExtTrigger imagename.tif where imagename.tif is the name of the images you wish to be taken. The first image name in a series will be imagename_00000.tif unless otherwise specified. If NImages > 1, the image number will be incremented for each image in the series.

Information	#4

The settings required for external triggering are:

- NImages
- ExpTime
- ExpPeriod
- Delay (optional)

After receiving a trigger (rising edge), the detector will wait a period of time defined by Delay, take an exposure of length ExpTime, readout the image and after a period defined by ExpPeriod will repeat the cycle for NImages images.

Information	#5

The image number is only incremented during the exposure series; if you reissue the command ExtTrigger imagename.tif the system will start writing images from imagename_00000.tif and overwrite existing data.

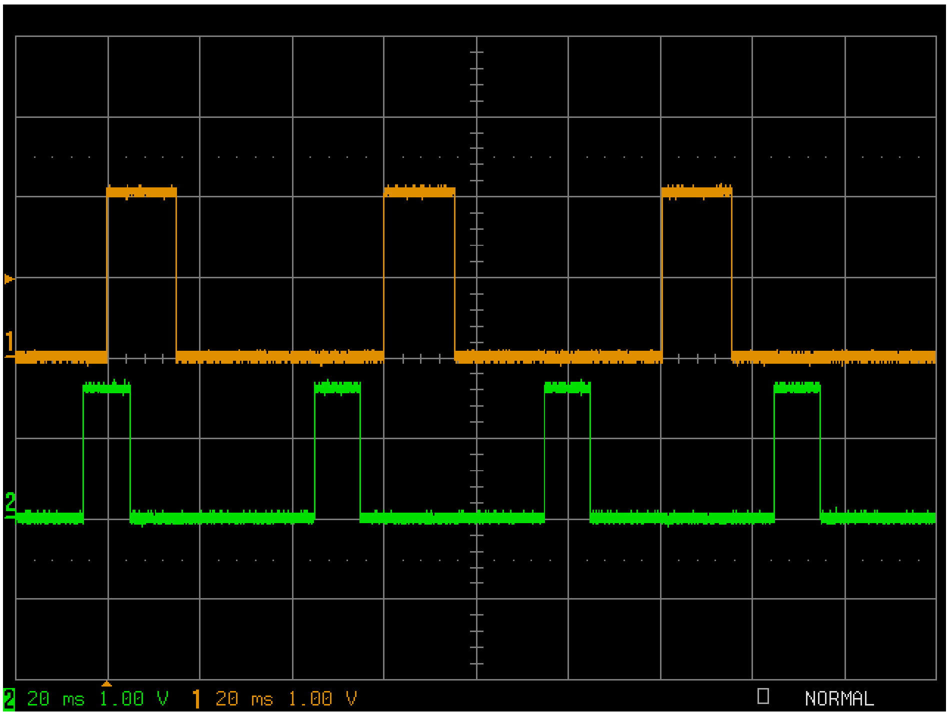

Figure 6.1: Oscilloscope Trace of an External Trigger. (orange) enable signal of the detector; (green) trigger

In figure 6.1 the upper trace is the exposure (enable) signal, the lower trace is from the pulse generator being used as a trigger. For this external trigger, NImages is 3, the Delay is 0.005 s, ExpTime is 0.016 s and ExpPeriod is 0.06 s. Note that, only the first rising edge of the trigger signal is used in this trigger mode.

Because the external trigger relies upon the module's internal clock signal to start the timing of the exposure, there is a delay and jitter between the trigger signal and the start of the first exposure. The maximum jitter is ∼15 ns with an average delay of 177 ns (see figure 6.2).

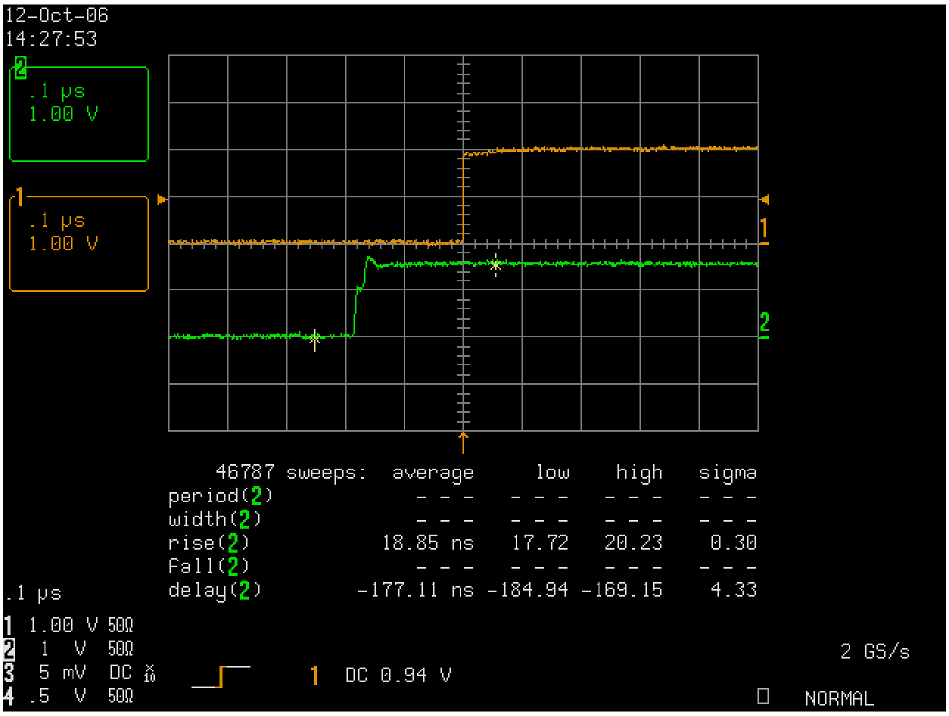

Figure 6.2: Delay and Jitter. (orange) enable signal of the detector; (green) trigger

6.3.2. External Multi Trigger Mode

The external multi trigger mode is started with the Camserver command ExtMTrigger imagename.tif where imagename.tif is the name of the images you wish to be taken. The image name will be imagename_00000.tif unless otherwise specified. If NImages >1 the image number will be incremented for each image in the series.

After issuing the ExtMTrigger imagename.tif command the detector will monitor and take a number of images defined below, on the level of the trigger pulse.

Information	#6

The settings required for external multi triggering are:

- NImages
- ExpTime
- ExpPeriod
- Delay (optional)

After receiving a trigger (rising edge), the detector will wait the period of time defined by Delay, take an exposure as defined by ExpTime, readout the image and will wait for the next trigger edge. This will be repeated NImages times.

Information	#7

The image number is only incremented during an exposure series. If you reissue the command ExtTrigger imagename.tif, it will start writing images from imagename_00000.tif and overwrite existing data.

6.3.3. External Enable Mode

This mode allows for detection of photons during the trigger signal is high (gating), see 6.3.
The external enable mode is started with the Camserver command ExtEnable imagename.tif where imagename.tif is the file name of the images. The first image name will be imagename_00000.tif unless otherwise specified. If NImages >1 the image number will be incremented for each image in the series.

After issuing the ExtEnable imagename.tif command the detector will enable counting when the input signal on the external input is high. In total the number of images pre-defined by NImages can be recorded.

Information	#8

The settings required for external enable mode are:

- NImages
- ExpTime (optional, should be set for an adequate rate correction to the trigger pulse length)
- ExpPeriod (optional, should be set for an adequate rate correction to the trigger pulse period)

Caution	#4

The image number is only incremented during an exposure series; if you reissue the command ExtEnable imagename.tif it will start writing images from imagename_00000.tif and overwrite existing data.

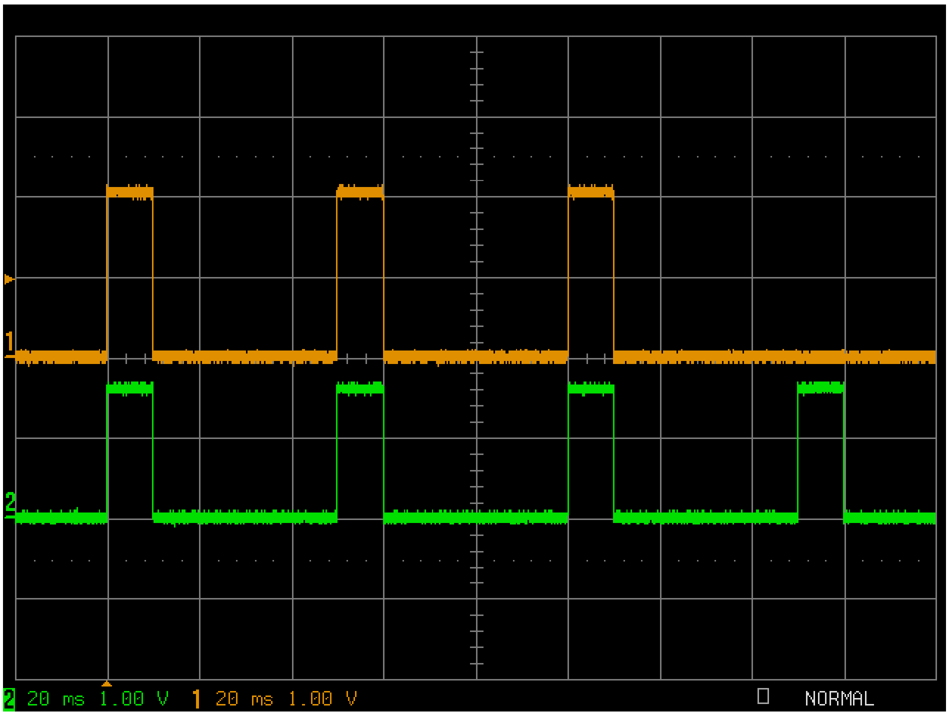

Figure 6.3: Oscilloscope Image of an external enable. (orange) enable signal of the detector; (green) external gate

In this example using external enable, NImages was set to 3.

Because external enable gates the counter directly, it does not rely upon the detector's internal clock. This means that the delay between the enable and start of the exposure is negligible and mostly given by the rise time of the enable signal provided to the detector. This can be seen in the oscilloscope image below (6.4).

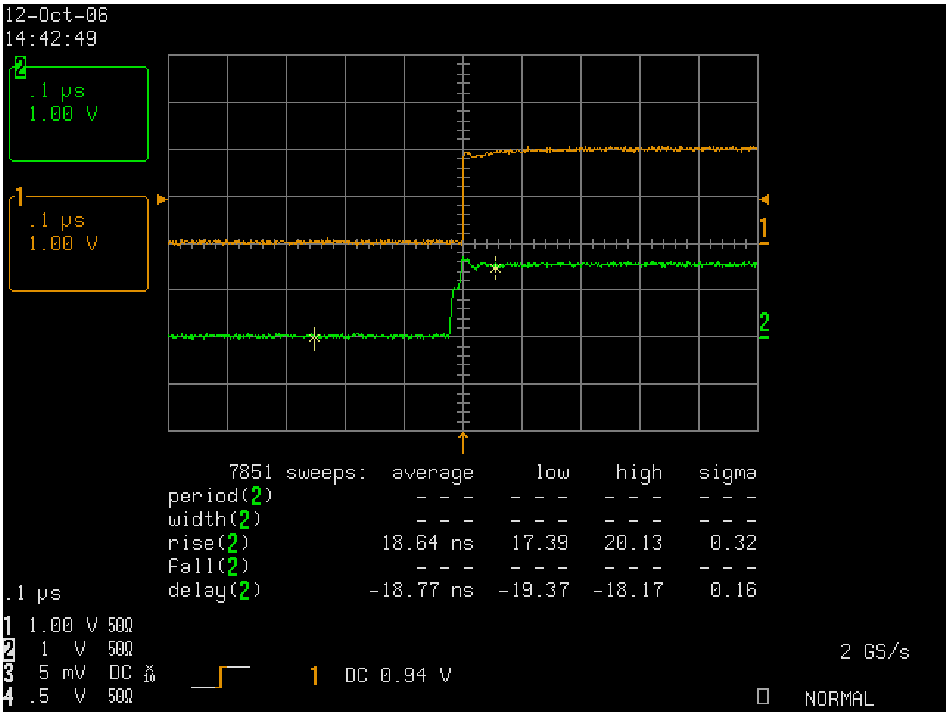

Figure 6.4: Oscilloscope Trace of the Typical Delay Between Enable Signal and Exposure. (orange) enable signal of the detector; (green) trigger

6.3.4. Multiple Exposure Functionality

This functionality can be used only in combination with the external multi trigger and the external enable mode. This functionality is useful to capture data from a rapidly repeating event that generates only a few X-rays per pixel for each event, such as pump-probe experiments. For example, it is possible to synchronize to the bunch structure of a storage ring providing that an appropriate gate is available from the ring control system. The data are accumulated in the pixels and read out after a certain number, e.g. 250,000, of exposures is collected.

To use this mode, set the variable NExpFrame to the desired value. The default value is 1. Please note that the definition of the ExpTime and the ExpPeriod changes as soon as the NExpFrame value is set different from 1. In this case the ExpTime and the ExpPeriod are the time and period, respectively, for each single exposure defined by the gate signal.

Information	#9

The settings required for multiple exposures are:

- NExpFrame (>1)
- ExpTime (is set to the gate pulse length)
- ExpPeriod (is set to the gate pulse period)

If the detector is used in this mode and it is synchronized to the bunch structure of a storage ring (high count rate), the rate correction might be invalid and it is advisable to turn it off (see RateCorrLUTDir in chapter 13).

In the following example NExpFrame is set to 2. The detector will take exposures in the same way as described for external enable, but will additively bundle 2 exposures in each readout. If NImages is defined to be 3 and NExpFrame is defined to be 2, then the detector will take 6 exposures and generate 3 images. It is necessary to provide 6 pulses or rising edges to achieve a successful readout of 3 images.

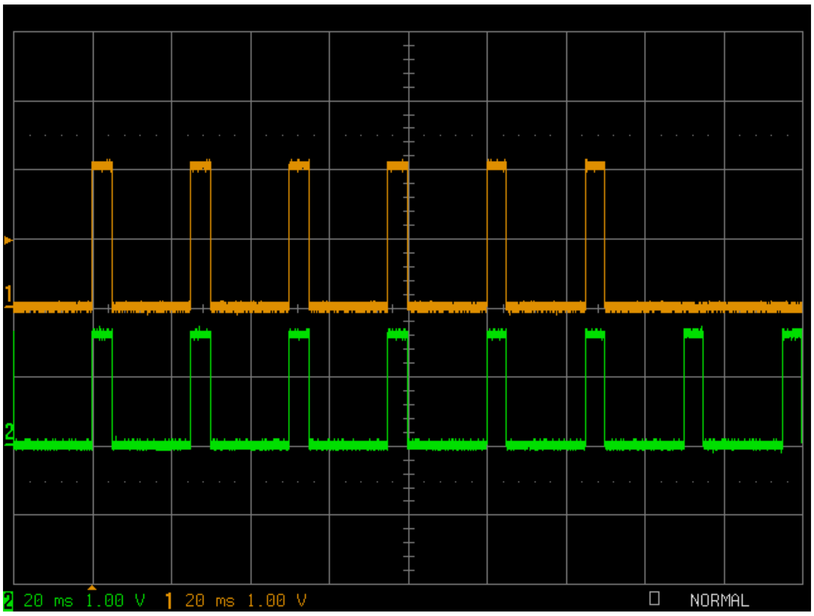

Figure 6.5: Oscilloscope Image of the Multiple Exposures. (orange) enable signal of the detector; (green) external gate

7. TRIMMING THE DETECTOR

7.1. Principle

PILATUS3 detectors possess an adjustable energy threshold, which has to be set due to the working principle of the detector. The comparator voltage of the readout chip controls this threshold. Furthermore, the threshold of every pixel can be individually trimmed with six trim bits (6-bit DACs), which allow $2^6 = 64$ different values. The trim voltage of each chip determines the magnitude of the influence of these trim bits. Since the process of threshold trimming is rather complex it is done during factory calibration. To be able to operate the detector appropriately, the determined trim-parameters have to be applied with one of the methods described below. See also figure 3.3.

Information	#10
Every detector is calibrated in the factory.	

7.2. Trimming Commands

The detector will be automatically trimmed for copper radiation when started. During startup of Camserver the command SetCu is executed. This causes that the detector is trimmed optimal for copper radiation.

To change the default behavior during startup please replace the line SetCu with another trimming command (e.g. SetMo for Molybdenum radiation) in the Camserver startup file (pidet.set, see section 3.3.2).

The trimming of a running detector can be switched by using the corresponding trimming command.

Table 7.1: Trimming Commands

Command	Description
SetCu	Appropriate for operating with Copper radiation
SetMo	Appropriate for operating with Molybdenum radiation
SetCr[1]	Appropriate for operating with Chromium radiation
SetFe[1]	Appropriate for operating with Iron radiation
SetAg[1]	Appropriate for operating with Silver radiation
SetEnergy [energy][1]	Set or query the threshold setting for used X-ray energy in eV

7.3. Set the Threshold with more Control

In order to have more control over the threshold the SetThreshold command can be used.

Background: The detector systems require that an energy threshold is set. This threshold is usually set to 50 % of the incoming X-ray energy for reasons which are explained below. However, it is also possible to set the threshold to an arbitrary energy.

What happens: All X-ray photons with an energy higher than the threshold energy will be counted by the detector whereas photons with an energy below the threshold will be cut off and therefore not counted. This transition is not perfectly sharp and follows an s-shaped curve with a derivative of about 1 keV (FWHM). This means that if you set the threshold energy at exactly the incoming (monochromatic) energy ~50 % of the photons are detected. At higher energies the asymptote of this

[1] depending on availability of energy calibrations

s-shape is not a constant but still slightly increasing.
The optimum of the threshold value is at 50 % of the incoming energy. This arises from the fact that it is possible that an X-ray will be absorbed at the boundary between two pixels and the charge is divided to both of them. If the threshold energy is set below 50 % of the incoming X-ray energy it is possible that one photon is counted twice in two neighboring pixels.

Best usage: We recommend to set the threshold energy for the PILATUS3 detector between 50 % and 80 % of the incoming X-ray energy. The upper limit of 80 % is not a hard criterion but above that the energy resolution gets worse. Moreover, one should not set the threshold much closer than 1 keV to the incoming X-ray energy due to the suppression of the primary incoming energy.
However, it is not a problem to set the threshold e.g. to 60 % of the incoming energy. At this threshold there are only about 3-10 % (depending on the energy) fewer counts as compared to a threshold of 50 %. This decrease is continuous and almost linear until the threshold is $\sim$ 80 % of the incoming energy.

Energy: The X-ray energy has to be set via the SetEnergy command to get an optimized flat field (compare chapter 10) since the flat field depends on the energy and also on the energy-to-threshold ratio. If the SetThreshold command is used and the threshold is not set to 50 % of the incoming energy, it is necessary to specify the correct energy as a parameter in the SetThreshold command for an adequate flat field.

Table 7.2: SetThreshold Command

Command	Description
SetThreshold [energy energy_value] [[gain] threshold][2]	This command allows setting the threshold energy as well the incoming energy for an optimized flat field. • If all parameters are omitted the current settings are shown. • If the key value pair [energy energy_value] is given, the energy_value is set as incident photon energy, else the incident photon energy is assumed to be 2x the threshold energy. This is important for the correct usage of the flat field, since it depends on both, energy and threshold. • gain has no effect on current systems and is only included for backwards compatibility with earlier PILATUS detectors. • threshold is in eV. • SetThreshold 0 turns off (invalidates) the current settings; however nothing is transmitted to the detector. This may be used to force a reload of the setting. Example: SetThreshold 7400 SetThreshold energy 14200 7500 Example (earlier systems): SetThreshold midG 7400 SetThreshold energy 14200 lowG 7500

[2] depending on availability of energy calibrations

8. BAD PIXEL MASK AND MODULE GAPS

8.1. Using the Bad Pixel Mask

During factory calibration the bad pixels (non-responding or noisy) are identified and stored in a tif image. This image contains the value 0 for good pixels and 1 for bad pixels. Once the bad pixel mask is loaded, bad pixels are flagged with -2 in the recorded images. This can be desired or not since once the pixels are flagged, their information is lost. It is also possible to record without bad pixel mask and apply the bad pixel mask when post processing the data.

Information	#11
	The bad pixel mask created during factory tests is automatically applied to each image that you take. The bad pixel mask is loaded every time the detector is trimmed (e.g. with the Camserver command SetCu).

To NOT apply this bad pixel mask, you have to use the command LdBadPixMap with a 0 as argument after a trimming command (e.g. SetCu). For further details please see Camserver command LdBadPixMap in chapter 13.

8.1.1. Adding new Bad Pixels to the Mask

It is possible that one or more noisy pixels can appear over time. To add such a bad pixel to the mask you have to modify the actual (bad pixel) mask and put a one at the corresponding x y position a one. This can be done with TVX or any software able to manipulate 32 bit tif images.

To prevent loss of information make a backup of the old mask. In a shell simply type (as one line) where DDMMYY is the actual date:

[]$_ Shell Code

```
cp /var/local/lib/dectris/config/calibration/Mask/badpix_mask.tif
/var/local/lib/dectris/config/calibration/Mask/badpix_mask_untilDDMMYY
```

Then display the bad pixel mask with TVX:

[]$_ Code in TVX

```
disp /var/local/lib/dectris/config/calibration/Mask/badpix_mask.tif
```

Add the bad pixels, e.g. at the coordinates x: 17, y: 126 by the following command:

[]$_ Code in TVX

```
pixlfill 1 17 126 17 126
```

The coordinates are given twice because it is also possible to define a box, which will be filled with the first value after the pixlfill command (see also section 11.3). To assure the modified mask is saved please execute the following command:

[]$_ Code in TVX

```
deleteallobjs
```

8.1.2. Make a new Bad Pixel Mask from an Uniform Illumination

If you expose the detector with a flat field you can use this image to create a new mask with TVX by defining upper and lower limits. Pixels in the detector that are either dead, too noisy or behave in a non-desirable manner can be masked out.

Example: If you have an image called img_01.tif with an average count (see boxall in section 11.2 for the statistical analysis) of e.g. 1000 counts per pixel stored in the image directory /home/det/images you can use the mkmask command (see section 11.3) in TVX to get a mask image:

[]$_ Code in TVX

```
mkmask img_01.tif goodpixel_mask.tif 600 1400
```

This will give you a file called goodpixel_mask.tif with 0 where the pixel values are outside the mentioned boundaries of 60 % and 140 % of the mean value of the provided image, img_01.tif. To obtain the appropriate format for Camserver please invert the image:

[]$_ Code in TVX

```
move badpixel_mask.tif=1-goodpixel_mask.tif
```

This created badpixel_mask.tif must then be copied to /var/local/lib/dectris/config/calibration/Mask. Please make a backup copy of the old mask.

8.2. Flag the Module Gaps

On the PILATUS3 the gaps between the modules can be filled either with the value 0 or -1. Which number will be filled in, is controlled by the Camserver command called GapFill (see Camserver command at chapter 13).
The default value for GapFill is set to -1. To change this behavior please change the line Gapfill -1 in the Camserver startup file /var/local/lib/dectris/config/cam_data/pidet.set.

9. CRYSTALLOGRAPHY PARAMETERS IN FILE HEADER

The MxSettings command allows the user to store more information (as comments or in the CBF template) inside the image header of TIFF and CBF images. The syntax of the MxSettings command is the following:

[]$_ MxSettings Command

```
MxSettings [parm\_name value]
```

It can be used to enter one of the following parameters parm_name with its value:

Wavelength; Energy_range; Detector_distance; Detector_Voffset; Beam_xy; Beam_x; Beam_y; Flux; Filter_transmission; Start_angle; Angle_increment; Detector_2theta; Polarization; Alpha; Kappa; Phi; Phi_increment; Chi; Chi_increment; Omega; Omega_increment; Oscillation_axis; N_oscillations; Start_position; Position_increment; Shutter_time; CBF_template_file

Information #12

The values of these parameters may alternatively be specified in the CBF header template. In this case, the parameter CBF_template_file gives the full path to the template. Note that the CBF template may be used to set variables even when TIFF images are to be written.

The command with no parameters will print the entire current list. If only one parameter without value is given the corresponding value is printed.

In an automatic sequence, values are automatically incremented by their corresponding increments starting from the initially set value (Start_angle is incremented by Angle_increment, Phi by Phi_increment, Chi by Chi_increment, Omega by Omega_increment, and Position by Position_increment).

MXsettings CBF_template_file 0 can be used to turn off this setting.

More information can be found in the cbf specification available from https://www.dectris.com/.

10. FLAT FIELD IMAGE

10.1. Using the Flat-Field Correction Image in Camserver

Information	#13
Flat-field correction files recorded during factory calibration are by default automatically applied to all acquired images.	

To correct sensitivity variations between different pixels a flat-field correction is applied.

The flat-field is a function of the beam energy and energy to threshold ratio. It is automatically loaded when setting the energy via the trim commands SetCu, SetMo, SetFe, SetAg, SetCr, and SetEnergy.

If the SetThreshold command is used to set the threshold energy also the used X-ray energy has to be specified to obtain an optimized flat-field. (See SetThreshold command in chapter 13 and section 7.3 for more details).

To turn off the flat-field correction you can use the Camserver command LdFlatField 0. To turn it on again, you may use the trim commands in Camserver, which will reload the flat-field.
The flat-field can also be applied while post processing the data. The flat-fields for all used settings are stored as tif files under /var/local/lib/dectris/config/calibration/FF_*E<Energy>_T<Threshold>_*.tif.

Be aware that the built-in flat-field correction can result in discretization errors due to rounding, which can become relevant in particular for regions with only few photons counted per pixel. This is because after flat-field correction the image will be rounded to the nearest integer and saved as data type integer.

11. TVX - DATA AQUISITION AND ANALYSIS SOFTWARE

TVX is a handy tool for data acquisition and analysis. This section describes only the most commonly used commands in TVX. All commands are case-insensitive; however, filenames are case-sensitive.

An object in TVX may be an image or a graph. Many commands, such as move, will work on objects of either kind. Objects can be combined with standard arithmetic operators (+, -, *, /, +=, etc.), logical operators (<, >, <=, >=, |, ||, &, &&) and special operators («, », !, :, «=, etc.) in arbitrarily complex expressions to perform sophisticated analyses and to construct custom scripts. In case of doubt, try it! You can't hurt anything.

Many commands in TVX require an input value or argument. Without the declaration of a value, the currently set value is shown.

Information	#14
Input values are shown in Italic. Macros are shown underlined.	

Table 11.1: Commands and Macros

Command or Macro	Description
menu	Shows all commands. Output is divided in 5 parts • Reserved Words: (do not use as variables) • External Procedures: • User Commands in current directory: All TVX commands; use man command to get a detailed description • Defined variables & strings: This are macros which are created during startup (e.g. in default.gl); use show macroname to see more details • User Variables: assigned variables
help command -or- man command	Displays the help text for the command. Help help is a good way to start.
show macroname	Displays the definition of a macro.
ESC-button	Stop a running task and return to the TVX line interpreter.
CTRL-C	Full reset of TVX. 5
rbd	Read Back Detector. Self-test of the digital part of the detector. Sends a digital pattern to each pixel, reads it back and displays the image. 6 Every pixel shows 1000 counts.
calibdet	Self-test of the detector. Sends 100 calibrate pulses to the analog part of each pixel, reads back the recorded values as an image and displays the result. 6 Every pixel shows 1000 counts.

Table 11.1: Commands and Macros - continued

Command or Macro	Description
setdac	Sets all Digital Analog Converters (DAC) to the predefined values.
imagepath path	Image Path. Without the input of a path it displays the current default path. With a declaration it changes the default path for images. The imagepath command also sets the autoname to the new path.
grafpath path	Display or change the default path for graphs. The keyword grafpath can be used in expressions as [grafpath]
expose exposure_time (in seconds)	expose 1: takes an image with an exposure time of 1 s. Shortest exposure time: >0.000 001 s. Shows the exposed image and its name immediately after completion.
exposem exposure_time	continuous camera mode without saving images. Takes images until any key is pressed. The last image is stored in temp.tif
disp filename	Display an image (see section 11.1). Opens up to 3 windows with successive invocations.
disp1 filename	Displays an image reusing the last window. Useful in loops.
graf filename1[filename2[filename3]]	Graph up to 3 graphs in a window.
Convert [S][D][type]	Change the image data type. S...source image, D...destination image. Type...Char, Short, Int, Float.
define	DEFINE name="instruction1; instruction2;....." Defines macro, user symbol name or value. Example: define tpict="zpict; move imt=im3" defines symbol tpict as a combination of zpict, and the built-in move instruction.
CaptureIm filename	Capture the default image to filename Captures a displayed image (and its zoom) as a .ppm (portable pixmap) file, including coloration and contrast adjustments.
CaptureGR filename	Capture the default graph to filename Captures a displayed graph (and its zoom) as a .ppm (portable pixmap) file.
connect [ip_address]	Connect the socket connection from TVX to the Camserver at [ip_address]. The IP is not necessary if TVX is running on the same computer.
disconnect	Disconnect the socket connection from TVX to Camserver. To e.g, allow a beamline control system like EPICS take control over Camserver.

Caution #5

⚠ Do not use this in Camserver.

Caution	#6
Use this command always after a startup.	

11.1. Description of the Image Display

The TVX command disp allows to display images. The image will be displayed in its own GUI, which contains several options to adjust the contrast and the min-max values. Moreover, it is possible to display the image in different color schemes. This image display will automatically show up after you execute the expose macro in TVX.

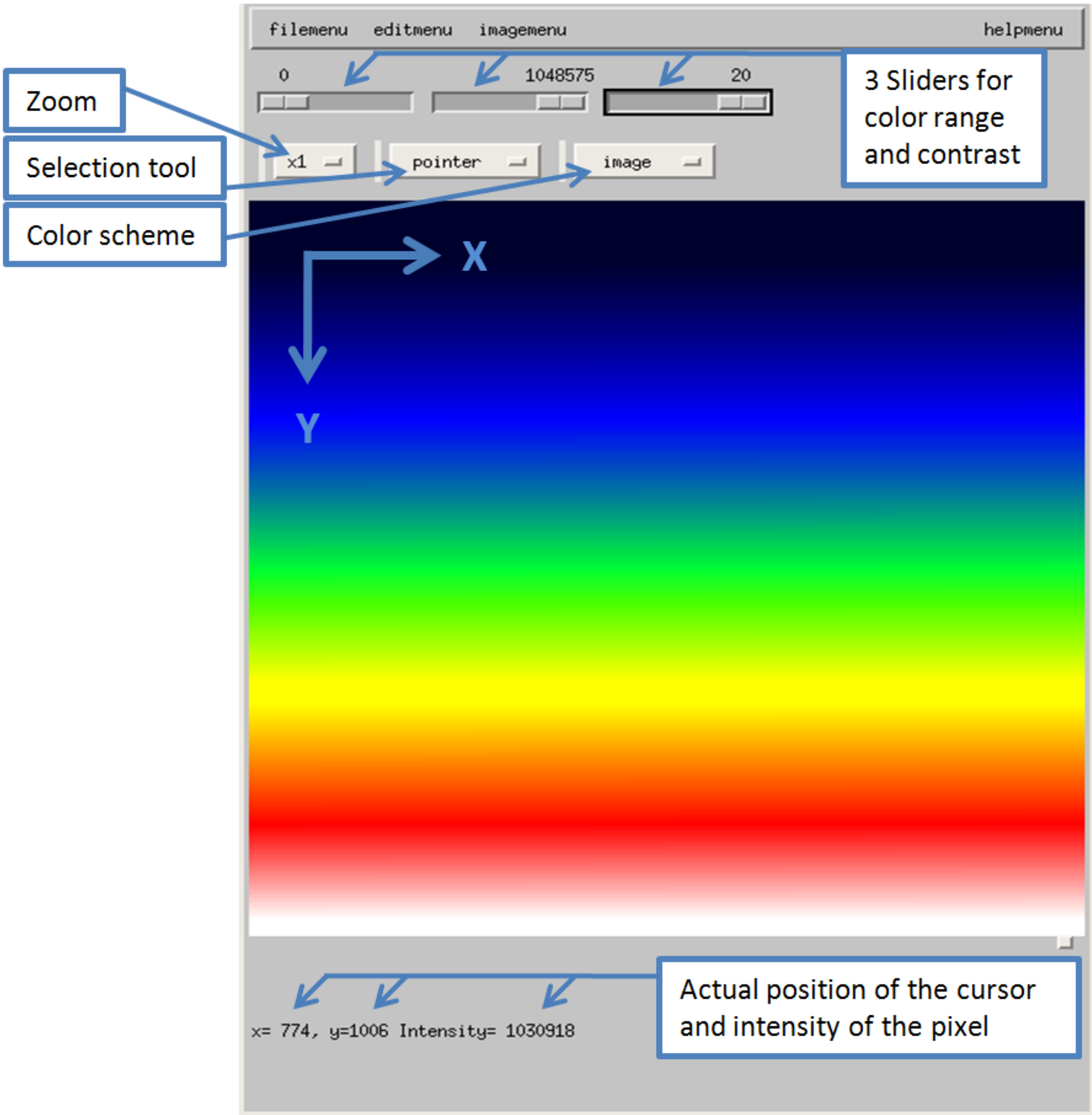

Figure 11.1: Image Display from TVX

Table 11.2: Display Tools

Display Tools	Description
(sliders)	Define the color and the contrast of the image. For every value of a pixel a color from a lookup table will be displayed. With the two left sliders the cut off for the low and high values can be set. Values outside this range are displayed with the same color. The third slider defines the contrast factor. The sliders can be moved with the mouse or by putting the mouse on the slider and adjusting the value with the left and right cursor buttons. They can also be set from the command line using the disp command (use man disp).
zoom	A magnification can be chosen and the enlarged area is shown in a new window. The zoom outline in the main window can be positioned by clicking or dragging with the mouse with the right button depressed.

Table 11.3: Selection Tools

Selection Tools	Description
pointer	Normal pointer.
annulus	Allows analysis of circular areas. The sizes of the circles can be adjusted with the mouse or directly by the setting the values in the image window.
box	Allows analysis of rectangular areas. Move the box with the right mouse or place the center of the box with the left mouse button. The size of the box can be adjusted with the mouse or directly by setting the values in the image window.
butterfly	Allows analysis of special shaped areas. The shape of the area can be adjusted with the mouse or directly by the setting the values in the image window. The circle is only for alignment purposes.
Line	Distance measuring tool. Requires that the correct pixel size is set in setup file (tvxrc)
resolution	Resolution circles for crystallographic patterns. Calculates the resolution of the image. The correct parameters for the detector should be set in the detector setup file or from the command line (det_dist, lamdba and pixel size)

Table 11.4: Display Mode

Display Mode	Description
grays	Color lookup table with gray scale.
spectral	Color lookup table with a spectral distribution (blue and black near zero, red fading to pink and white at the high end)
thermal	Color lookup table going from blue through yellow and red, but no greens.

Table 11.4: Display Mode - continued

Display Mode	Description
decades	The values between Min and Max are displayed linearly, but with the scale wrapping around Scal number of times. Thus, Scal = 1 is linear, Scal = 5 covers the range Min to Max with 5 linear segments going from 0 to 255, 0 to 255, etc. This gives lots of artificial contrast that is good for smoothly-varying SAXS data, but is otherwise rather non-intuitive.

power	The image is displayed between Min and Max using the transfer function:
	
	(# grays)*((value - min)/(max - min))*(Scal/15)
	# grays is usually 256. Thus, a small value of Scal (~3) gives a very steep transfer function at low values, and very little contrast at high values.
	Scal = 15 is a linear transfer function; Scal > 15 is useful only in special cases.

reverse	The values are reversed - X-rays in the image become black rather than white. Useful for crystallographic images.

Several test images and graphs are included in the system.

Information	#15

Try the following:

imagepath examples
disp gray20bit.tif

grafpath examples
grafdemo

Information	#16

Example: Butterfly selection tool

This selection tool is useful for straight line integrations (densitometer traces) and for integrating small angle scattering patterns from either a line or a point X-ray source.

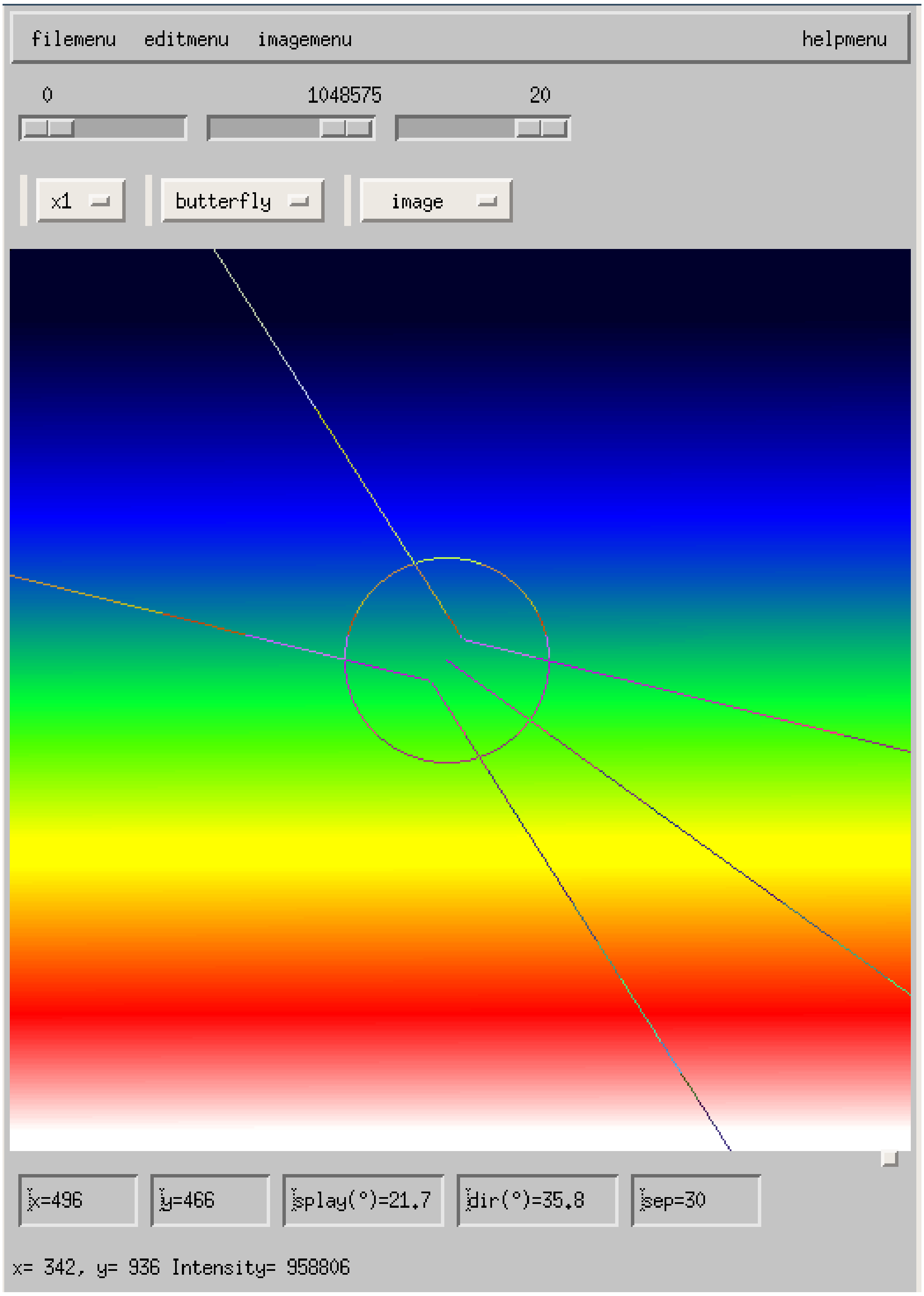

Figure 11.2: Example of the Butterfly Selection Tool

The size and position can be adjusted directly with the mouse or by typing the values directly into the boxes. The circle is used only as a positioning aid. Use the command integrate in the TVX window to display the result.

11.2. Image Analysis and Processing Commands

TVX offers basic image analyzing and processing commands. The most important commands are described in this document. All created numeric data are stored in the directory given by grafpath; image data are stored in the directory given by imagepath.

Table 11.5: Image Analysis and Processing Commands

Command	Description
move filename1=filename2	The basic image manipulation command. In the simple form shown, this copies an image to a new name or directory. However, filename2 can be any arithmetic expression of images and constants.
Integrate	Integrate the pixels selected by the current selection tool - box, butterfly (includes straight-line case), or spot (annulus tool) - and show the resulting graph. Usage: integrate [IM] [graph_name] For the butterfly, the graph name can be given as the second parameter; in this case the image name must be specified. In other cases, the default image is used if no image is specified.
histogram lo hi int	Histogram of the pixels selected by the box tool on the image. Alternatively, specify the image and region-of-interest on the command line. Usage: histogram [IM] lo hi int [x1 y1 x2 y2] [graph_name], where lo is the first value to use, hi is the last value, and int is the interval. If IM is not specified, the default IM is used. [x1 y1 x2 y2] are the coordinates of the box to be histogrammed. If no graph_name is specified, the histogram is placed in file hist[n].dat in the default graph directory, where n rotates through the values 0...5. This file can be then be moved to a permanent file by a command such as move myhist=hist1. The histogram parameters are remembered, so subsequent operations with the same parameters can be obtained by just typing histogram. If the coordinates are specified on the command line, the parameters must also be specified. If the file name is specified, either the image name must also be given, or 3 (or 7) numeric parameters must be specified. In the integral mode, the integral is written to hist[n+1].dat. If the name is specified, it is appended with _i for the integral. See also histset
box	Print statistics from the current box selection tool on the image. Alternatively, specifiy image name and box coordinates on the command line. Usage: box [IM] [x1 y1 x2 y2] If IM is not given, the default IM is used. [x1 y1 x2 y2] are the coordinates of box to be examined. If not given, use the box selection tool on image. If given, the box selection tool is created or updated on the image, if it is displayed. If the box is set with the mouse, box and integrate give the same result. Several system variables are set: counts (total counts in box), area, mean, minimum, maximum, stdev (rms), var (variance), xcen and ycen (centroid), box_x1, box_x2, box_y1 and box_y2 (corners of box).
boxall	Print statistics from the whole (current) image.
format n1[.n2]	Control the number of digits to be printed (n1) or the number of decimal places (n2).

Table 11.5: Image Analysis and Processing Commands - continued

Command	Description
deleteallobjs	Delete the TVX record of all objects – the objects themselves are untouched. Images are stored in the TVX memory up to the limit specified in tvxrc, which can consume significant resources; use this command to free up memory. In addition, one can create files with identical names in various directories. To avoid the necessity of always specifying full path names, use this command to clear the TVX memory.
deleteobj filename	Deletes the specified object from the TVX memory. The file on disk is untouched.
maskimg	Specify a mask image to be used by many TVX commands, such as box and histogram (see below).

11.3. Mask Files

Setting a mask image is useful when you are looking at the statistics of images from the detector. Pixels in the detector that are either dead, too noisy or behave in a non-desirable manner can be masked out. After a pixel has been masked, it will be excluded by the statistical analysis routines in TVX so that your results will not be distorted by image pixels with too high or too low values.
The mask file is an image file that uses only two distinct values for each pixel. Every pixel that is to be masked out is given a value of 0, every other pixel is given a value of 1. You can create a mask file from another image by using the command mkmask.

Table 11.6: Mask File Commands

Command	Description
mkmask	Make a mask from an image between two limits, inclusively. Usage: mkmask [IM [IMout]] low high The result is a mask of 1's and 0's, which can be used to select pixels of an image by multiplication. If no image is supplied, the default is used. Note that a float input object returns a 32-bit integer mask. Because the generated file is a normal image you can use any of the image manipulation tools supplied in TVX to alter your mask image if you wish.
maskimg [IM] -or- maskimg 0	Set or turn off the current mask image. If present, the mask is used to blank out bad pixels in statistical routines such as box, integrate, spot and histogram. All Zeros in the mask are excluded from the analysis whereas non-zeroes are included. With no argument, displays the current mask image name, if any. With numeric argument (e.g. 0), turns off the mask image.
ldm	ldm (short for: load mask) is a macro using the maskimg command to load the factory-produced mask (stored in /home/det/tvx/correct/goodpix_mask.tif).
pixlfill [IM] value	Set pixels in image IM to value using the current box (coordinates) as a template. This permits you to manually alter a mask image based on observations on a different image.

Information	#17
	If the command **deleteallobjs** is used after loading a mask image, the maskimg will be reset. The stored image is untouched.

11.4. User Defined Commands

TVX supports complex C-like commands in the command line.

Example:
To display a series of images as a movie:

[]$_ Shell Code in TVX

```
format 2; for (i=0;i<100;i++){disp1 image_000[i]; wait 0.5}
```

Displays image_00000 to image_00099 and waits 0.5 s between each picture. The brackets [ ] mean to substitute the enclosed argument as text with the number of digits specified by the format.

With **define** one can create custom commands for the current session and eventually save them for reuse.

[]$_ Shell Code in TVX

```
define test1 = "format 2; for (i=0;i<100;i++){disp1 image_000[i]; wait 0.5}"
```

Table 11.7: User Defined Commands

Command	Description
define name="string" **define** name=value	Define a name (value or command), which can be used in the current session. They are not saved when TVX closes.
save "myfile.gl"	Saves the currently defined commands in myfile.gl as text. Such files are called glossaries. Glossaries may also have executable commands edited in following all the definitions; these are preserved when the file is overwritten.
get "myfile.gl"	Load the definitions from myfile.gl, and execute any commands appended after the definitions.

11.5. Glossary Files

When TVX is started, a glossary is automatically started up called
/home/det/tvx/config/default.gl.

In this glossary, the main commands for using the detector are defined. Three other glossaries are called from default.gl (all in config):

Table 11.8: Glossary Files

Glossary	Description
det_spec.gl	Detector specific definitions. In case of multi module detectors number of banks, modules, tools for addressing modules and analyzing module specific data.
user.gl	User specific commands
startup.gl	Commands, which are automatically loaded at startup, e.g. setdac, rbd, calibdet. For usage at the beamline, usually the last command is Disconnect, which allows remote control of Camserver.

11.6. Example

The following line is a simple example of using TVX to create a flat field correction file (corr.tif) out of a high intensity count image (image_00001.tif).

After you recorded the image with adequate statistics and stored it in /home/det/images, you can use the following commands:

- disp image_00001.tif
- ldm
- boxall
- convert image_00001.tif corr_image_float.tif float
- move corr.tif=[mean]/corr_image_float.tif

If the image_00001.tif was an appropriate flat image (see chapter 10 for details) the 5 lines create a proper correction image (corr.tif), which can be used for flat field correction.

12. FACTORY CALIBRATION AND CORRECTION

The PILATUS3 detector systems come fully calibrated. The following calibrations are done at the DECTRIS premises:

1) Energy Calibration
See the system information sheet in your user handbook for more information about the calibrated energies and settings. The discriminator thresholds in the individual pixels are set by an automated procedure (as described in chapter 7).

2) Rate Correction
The counting mechanism introduces a short dead time after each hit, which becomes significant for rates above a few 10^5 counts/s/pixel. The resulting loss in the number of counted photons is corrected by applying corresponding rate-correction factors. If the uncertainty of the correction becomes too large, the correction is cut off at a saturation value. This value is also printed in the header. The correction depends on the energy and the selected threshold setting.

3) Flat Field
An appropriate flat field will correct sensitivity variations within and between different pixels. The flat-field correction image is loaded automatically when trimming the detector (as described in chapter 10).

4) Bad Pixels
The system is supplied with a mask image containing the defective pixels. Camserver uses this mask to flag defective pixels as -2 in the final image. The gaps between the modules are flagged with -1. This default behaviour can be changed with the Camserver commands LdBadPixMap and GapFill (see chapter 13).

13. CAMSERVER COMMANDS

The following list presents the Camserver commands with a short description. For detailed usage of the detector system please see the chapters 6 to 10.

Information	#18

The reply from Camserver consists of a command index number, followed by a space and either "OK" or "ERR", followed by another space and possibly a message (see section 5.2).
returnCode returnState string\0x18

- \0x18 is ASCII for cancel
- Each parameter is separated by a space

Table 13.1: Camserver Commands

Command	Arguments (unit) [default values]	Description	Socket Connection Return Code	Socket Connection Return Text
Exposure	file_base_name + file extension[1]	Start an exposure. ExpTime, ExpPeriod, ImgPath and NImages have to be set beforehand to the desired values. The image is written to the specified file_base_name1 relative to ImgPath, or to an absolute path if given. The format of the image is derived from the filename extension if given (tif, cbf or edf); otherwise a RAW image is written. If an exposure series is set up (NImages >1), an image number is inserted before the extension[1].	at start: 15 at end: 7	at start: starting xxx second background: <date & time> at end: full path name of last image
ExtTrigger	file_base_name + file extension[1]	Arm the detector for an exposure or an exposure series started by one external trigger. Before execution, set timing parameters by the commands ExpTime and ExpPeriod. To specify a delay between the trigger and the start of the exposure a Delay time can be set. The time from arming the system until the arrival of the first trigger is unlimited. Use K transmitted over the socket connection to interrupt this state.	at start: 15 at end: 7	at start: starting externally triggered exposure(s): <date & time> at end: full path name of last image

[1] More information about **file_base_name** see end of the table

DECTRIS detecting the future

Table 13.1: Camserver Commands - continued

Command	Arguments (unit) [default values]	Description	Socket Connection Return Code	Socket Connection Return Text
ExtMtrigger	file_base_name + file extension[1]	Arm the detector for an exposure series where each exposure is started by an external trigger. Set timing parameters by the commands ExpTime and ExpPeriod before execution. Each exposure is trigged by the external trigger, but uses the internal timer for the exposure time. Individual exposures can be added up within one readout/image by specifying NExpFrame prior to execution (see Multiple Exposure Mode 6.3.4). The time from arming the system until the arrival of the first trigger is unlimited. Use K transmitted over the socket connection to interrupt this state.	at start: 15 at end: 7	at start: starting externally multi-triggered exposure(s): <date & time> at end: full path name of last image
ExtEnable	file_base_name + file extension[1]	Make an exposure or an exposure series using an external gate signal. Each exposure is started when the signal changes to high and is finished when the signal changes to low again. The time from arming the system until the arrival of the first trigger is unlimited. Use K transmitted over the socket connection to interrupt this state.	at start: 15 at end: 7	at start: starting externally enabled exposure(s): <date & time> at end: full path name of last image
ExpTime	Time (s) [1.0]	Set the exposure time; time < 60 days.	15	Exposure time set to: xxx sec.
ExpPeriod	Time (s) [1.05]	Set the exposure period ExpPeriod = ExposureTime + Readout time. Time < 60 days. For readout time please refer to the Technical Specifications of the detector.	15	Exposure period set to: xxx sec
ImgPath	Path [/home/det/images]	Change the image path If the directory does not exist, it will be created if possible. A path relative to the current path is accepted; '..' is accepted. If imgpath test is given, and the current directory named is test, a new directory is NOT created. If such a new directory is desired, it may be specified by test/test. If imgpath test1/test2 is given, and the current path is .../test1/test2, a new directory is NOT created.	10	the path
NImages	Number (#)[1]	Set the number of images for an automatic sequence Maximum number of images is 999,999	15	N images set to: nn

DECTRIS detecting the future

Table 13.1: Camserver Commands - continued

Command	Arguments (unit) [default values]	Description	Socket Connection Return Code	Socket Connection Return Text
Delay	Time (s) [0]	Set the delay from the external trigger until start of the first exposure. The time must be shorter than 64 s The delay is reset to 0 for ordinary exposures and external enable	15	Delay time set to: n.n sec
NExpFrame	Number (#)[1]	Set the number of exposures to accumulate per frame/readout. This is a method summing up images within the detector chip. A value >1 is a waste of X-rays except when using external enable or external multi-trigger to synchronously capture an event. If nexpframe >1, the reported measured exposure time applies to the last exposure only. The maximum possible number is 2^{32}-1.	15	Exposures per frame set to: nn
MXsettings	Mxparameters (text)[none]	Set crystallographic parameters reported in the image header mxsettings [parm_name value] [parm_name value] ... Possible: Wavelength, Energy_range, Detector_distance, Detector_Voffset, Beam_xy, Beam_x, Beam_y, Flux, Filter_transmission, Start_angle, Angle_increment, Detector_2theta, Polarization, Alpha, Kappa, Phi, Phi_increment, Chi, Chi_increment, Omega, Omega_increment, Oscillation_axis, N_oscillations, Start_position, Position_increment, Shutter_time, CBF_template_file Not settable with mx_settings, but provided to templates from detector settings: Timestamp, Exposure_period, Exposure_time, Count_cutoff, Compression_type, X_dimension, Y_dimension	15	None set or current settings
SetCu		Command to trim the detector optimal for usage with Copper radiation	15	OK /tmp/setthreshold.cmd
SetMo		Command to trim the detector optimal for usage with Molybdenum radiation	15	OK /tmp/setthreshold.cmd
SetCr		Command to trim the detector optimal for usage with Chromium radiation	15	OK /tmp/setthreshold.cmd
SetFe		Command to trim the detector optimal for usage with Iron radiation	15	OK /tmp/setthreshold.cmd
SetAg		Command to trim the detector optimal for usage with Silver radiation	15	OK /tmp/setthreshold.cmd
SetGa		Command to trim the detector optimal for usage with Gallium radiation	15	OK /tmp/setthreshold.cmd

DECTRIS detecting the future

Table 13.1: Camserver Commands - continued

<table>
<tr><th>Command</th><th>Arguments (unit) [default values]</th><th>Description</th><th>Socket Connection Return Code</th><th>Socket Connection Return Text</th></tr>
<tr><td>SetThreshold</td><td>Threshold parameters (text)[none]</td><td>This command allows setting the threshold energy as well the incoming energy for an optimized flat field.
Usage: SetThreshold [energy energy_value] [[gain] threshold]
• If all parameters are omitted the current settings are shown.
• If the key value pair [energy energy_value] is given, the energy_value is set as incident photon energy, else the incident photon energy is assumed to be 2x the threshold energy. This is important for the correct usage of the flat field, since it depends on both, energy and threshold.
• gain has no effect on current systems and is only included for compatibility with earlier PILATUS detectors.
• threshold is in eV.
• SetThreshold 0 turns off (invalidates) the current remembered settings; however nothing is transmitted to the detector. This may be used to force a reload of the setting.
Example:
SetThreshold 7400
SetThreshold energy 14200 7500
Example (earlier systems):
SetThreshold midG 7400
SetThreshold energy 14200 lowG 7500</td><td>15</td><td>Setting the threshold:
pathname of file

Without argument (once set):
Settings: xxx gain; threshold: xxxx eV; vcmp: x.xxx V Trim file: /path/to/trim/file/abc.bin

Without argument (never set):
Threshold has not been set</td></tr>
</table>

DECTRIS detecting the future

Table 13.1: Camserver Commands - continued

Command	Arguments (unit) [default values]	Description	Socket Connection Return Code	Socket Connection Return Text
SetEnergy	Energy (eV)[none]	Simplified method to set the threshold The requested energy is used to calculate appropriate threshold settings for the detector.	15	Setting the energy: pathname of file Without argument (once set): Energy setting: xxxx eV Settings: xxx gain; threshold: xxxx eV; vcmp: x.xxx V Trim file: /path/to/trim/file/abc.bin Without argument (never set): Threshold has not been set
K		Interrupts an exposure series	13 7	ERR kill full path name of last image
camcmd K		Interrupts a single exposure and an exposure series	13 7	ERR kill full path name of last image
LdBadPixMap[2]	Filename (text)[none]	Load a mask image giving bad pixels to be flagged Filename must be a full pathname. If filename is not given, the current setting is shown. If filename is "0" or "off", the pixel flagging function is turned off. The maximum number of bad pixels is 9000; the flag value is -2.	15	none or pathname
LdFlatField[2]	Filename (text)[none]	Load the flat-field correction file Filename must be a full pathname. File must be a 32-bit floating-point TIFF image. If filename is not given, the current setting is shown. If filename is "0" or "off", the flat field function is turned off. The image is pixel-wise multiplied by the correction file.	15	none or pathname

[2] Automatically set by the trimming command (e.g. SetCu).

DECTRIS detecting the future

Table 13.1: Camserver Commands - continued

Command	Arguments (unit) [default values]	Description	Socket Connection Return Code	Socket Connection Return Text
RateCorrLUTDir[3]	Directory name (text)[pathname]	Sets the directory from which the look-up table for the rate correction is loaded. The detector is shipped with a correction for a continuous X-ray beam. Directory name must be an absolute pathname or relative to /var/local/lib/dectris/config/cam_data/ratecorrection/. When called with the argument "0" or "off", the rate correction is disabled.	15	RateCorrLUTDirectory is pathname or off; Disabling LUT based correction
ReadoutTime		Show detector readout time in milliseconds.	15	Detector readout time [ms]: time in milliseconds
SetRetriggerMode	Number (1,0)[1]	Enables or disables the retrigger mode. Number can only be "1" (enable, default) or "0" (disabled). The current setting is shown when no argument is given.	15	Retrigger mode is set; Retrigger mode is not set
GapFill	Number (0,-1)[0]	Set the value to be used in pixels between modules. Number can only be "0" or "-1". If n is omitted, the current value is printed.	15	Detector gap-fill is: nn
THread	Channel (n)[all]	Read one of the temperature and humidity sensors Channels are numbered 1-6 on the first detector control board, 7-12 on the second. If channel is not specified, # 0 is addressed. If no sensor is connected, -99 is printed.	215	temperature and humidity
SetAckInt	Number (#)[0]	Set the interval for acknowledgements over the socket. This causes Camserver to acknowledge every n-th image. If N is omitted, the current value is shown. At the default (n=0) only the last exposure of a series is acknowledged. The initiating command is always acknowledged, so for 1 or more images, there is an acknowledgement before the start and at the end of a series. There are some restrictions at high frame rate: n cannot be too low.	15	none or current setting
ResetCam		Reset the camera.	15	none
DebTime	Time (s)[0]	Set the contact debounce time for external enable mode. If it is not given, the current setting is echoed. This is useful when the external enable is not "clean", e.g., derived from a mechanical switch. The external enable pulse must be shorter than 85 s.	15	Debounce time set to: n.n sec

[3] Set by any trimming command (e.g. SetCu) to the default value /var/local/lib/dectris/ config/cam_data/ratecorrection/Continuous.

DECTRIS detecting the future

Table 13.1: Camserver Commands - continued

Command	Arguments (unit) [default values]	Description	Socket Connection Return Code	Socket Connection Return Text
HeaderString	String (text)[none]	Give a string to be included in the image header. The maximum length is 68 characters, no formatting permitted. Enclose the text in quotes to transmit non-alpha characters.	15	none
Exit, Quit		Close the socket connection.		none
Df		Show the number of 1 KB blocks available on ImgPath.	5	1K blocks available
ExpEnd		Return the filename with which the exposure ended.	6	full path name of last image
CamSetup		Report camera setup.	2	Camera definition: ; Camera name: ; Camera state: ; Target file: ; Time left: ; Last image: ; Master PID is: ; Controlling PID is: ; Exposure time: ; Last completed image: ; Shutter is:
Telemetry		Report camera telemetry.	18	Image format: ; and additional camera messages
ResetModulePower	Time (s) [1]	Cycles module supply voltages and HV. Argument <time> is the power off time Note: threshold and energy settings will be lost.	at start: 15 at end: 15	Resetting module power, sleeping for <time> seconds >>> Threshold settings no longer valid
Version		Print the TVX/Camserver version.	24	version
ShowPID		Show the PID of the process receiving the command.	16	the pid

[1]file_base_name - Exposure commands take a filename or file basename as their argument. For single images, the filename is used as typed; if a recognized image file format extension is present (.tif, .edf, .cbf), the file will be created in that format. Otherwise, a RAW image is produced. For multiple exposure series (NImages >1), the typed name is used as a basename; again, the extension, if given, is used to set the image format. The following examples show the interpretation of the basename:

DECTRIS detecting the future

basename	files produced
test6.tif	test6_00000.tif, test6_00001.tif, ...
test6_.tif	test6_00000.tif, test6_00001.tif, ...
test6_000.tif	test6_000.tif, test6_001.tif, ...
test6_014.tif	test6_014.tif, test6_015.tif, ...
test6_0008.tif	test6_0008.tif, test6_0009.tif, ...
test6_2_0035.tif	test6_2_0035.tif, test6_2_0036.tif, ...
test6_014B.tif	test6_014B_00000.tif, test6_014B_00001.tif, ...

I.e., the numbers following the last '_' are taken as a format template, and as a start value. The minimum number of digits in the format is 3; there is no maximum; the default is 5. The format is also constrained by the requested number of images.

DECTRIS detecting the future

13.1. Camserver Known Issues and Features

13.1.1. Acquiring Images

Information	#19
More information about the command "NExpFrame" can be found in chapter 13, p. 41.	

Table 13.2: Acquiring images

Camserver Version	Issue
7	By using NExpFrame = 1, the exposure period for single exposure is changed.

13.1.2. Read-only Connection

Table 13.3: Read-only Connection

Camserver Version	Issue
7	If a telnet socket is opened subsequently after closing it, or a socket is closed subsequently after opening it, Camserver will only allow read-only connections, permanently. Restarting Camserver is necessary to recover this issue. Pause 1 s after opening a socket before closing it. Wait 1 s after closing a socket, before opening a new one.

13.1.3. Stopping Acquisitions

Information	#20
More information about the commands "K" and "CamCmd k" can be found in chapter 13, p.43.	

Table 13.4: Stopping Acquisitions

Camserver Version	Issue
7	For ni = 1: The kill command "k" is only accepted if the client sends "CamCmd k".
7	"13 ERR kill" (instead of "7 ERR ok") message occurs after a long waiting time or usage of external trigger (ExtMTrigger) with 50 Hz or lower frame rates when using command "k" or "CamComd k".
7	Camserver returns code 1 instead of 13 ERR or 7 ERR when "k" / "CamCmd k" is called when detector is idling.

13.1.4. Updating Camserver

Table 13.5: Updating Camserver

Camserver Version	Issue
7	Camserver Message: "Import Error: No module named tornado.wsgi". Tornado has to be installed. The Import Error can be ignored. Python Tornado is not needed for Camserver.

14. CAMSERVER TEST CLIENT

[]$_ Camserver Test Client

```
/*****************************************************************************\
Name: cam_client.c
Created by: Sebastian Commichau, May 2009
Modified by: Stefan Brandstetter, Feb 2011
Purpose: Client for Camserver
Compile with: gcc -o cam_client cam_client.c
DECTRIS Ltd.
Taefernweg 1
CH-5405 Baden
www.dectris.com
\*****************************************************************************/

#include <stdio.h>
#include <netdb.h>
#include <signal.h>
#include <unistd.h>
#include <string.h>

#define BUFSIZE 1024

int main() {

  // Change to required IP or hostname
  char server[64] = "localhost";

  // Change to required port
  int port = 41234;

  char buffer[BUFSIZE];

  int s;
  struct sockaddr_in s_addr;
  fd_set rfd;

  // Open socket descriptor
  s = socket(PF_INET, SOCK_STREAM, 0);

  // Resolve hostname and try to connect to server
  struct hostent *hostent = gethostbyname(server);
  s_addr.sin_family = PF_INET;
  s_addr.sin_port = htons((unsigned short) port);
  s_addr.sin_addr = *(struct in_addr*) hostent->h_addr;

  // Connect to socket
  if (connect(s, (struct sockaddr *) &s_addr, sizeof(s_addr)) < 0)
  return;

  printf("\n***** Command line socket interface for Camserver *****\n\n");
  printf("Type 'exit' to quit\n");
```

[]$_ Camserver Test Client

```
  // Main loop processing user input and socket input
  while (1) {

    printf("cam_client> ");
    fflush(STDIN_FILENO);

    // Wait for data either from terminal (stdin) or from socket
    FD_ZERO(&rfd);
    FD_SET(s, &rfd);
    FD_SET(STDIN_FILENO, &rfd);

    if (select(((int) s)+1, &rfd, NULL, NULL, NULL)==-1)
    break;

    // Data from stdin
    if (FD_ISSET(STDIN_FILENO, &rfd)) {

      fgets(buffer, BUFSIZE, stdin);

      if (!strcmp(buffer,"exit\n"))
      break;

      // Replace carriage return by null character
      if (buffer[strlen(buffer)-1] == '\n')
      buffer[strlen(buffer)-1] = '\0';

      // Write to socket
      write(s, buffer, strlen(buffer)+1);

      bzero(buffer, sizeof(buffer));
    }

    // Data from socket
    else if (FD_ISSET(s, &rfd)) {

      // Read from socket
      if (read(s, buffer, BUFSIZE)==0) {
        printf("server not existing anymore, exiting...\n");
        break;
      }

      printf("%s\n",buffer);

      bzero(buffer, sizeof(buffer));
    }
  }

  close(s);
  return 0;
}
```