DECTRIS®

detecting the future

EIGER

Detector Manual

Version: V1.1

DECTRIS Ltd.
5400 Baden
Switzerland
www.dectris.com

Table of Contents

1. Document History

1.1. Current document

Version	Date	status	prepared	checked	released
1.1	26.03.2015	In Work	SB/AM/MS/MH/BL/TD/GT		

1.2. Changes

Version	Date	Changes
1.0	17.12.2014	First version
1.1	26.03.2015	Additional information

2. *How to use this Manual*

Before you start to operate the EIGER detector system please read this manual thoroughly.

Address and Support:

DECTRIS Ltd.
Neuenhoferstrasse 107
5400 Baden
Switzerland
Phone: +41 56 500 21 02
Fax: +41 56 500 21 01

Website:

- www.dectris.com → Support → Technical Notes → EIGER
- www.dectris.com → Support → FAQ
- www.dectris.com → Support → Problem Report

Email:

- support@dectris.com

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

⚠ **Do not attempt to ship the system back before receiving the necessary transport and shipping information from DECTRIS.**

2.1. Disclaimer

DECTRIS has carefully compiled the contents on 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 published in these applications into other programs or other Web sites 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 application in whole or in part at any time, and is not obliged to update the contents of the manual.

3. Warnings

Please read these warnings carefully before operating the detector

- Before switching on the power supply, check if the supply voltage printed on the detector is in accordance to the power supply.
- Power down the detector system before connecting or disconnecting any cable.
- Make sure all required cables are connected and properly secured.
- Avoid any pressure or tension on the cables.
- Operating the detector outside the specified ambient conditions will damage the system.
- The detector is not specified to withstand direct beam at a synchrotron. The direct beam will damage the exposed pixels.
- Place the protective cover on the detector when it is not in use.
- Opening the detector or the power supply housing will void the warranty.
- DO NOT INTERACT WITH THE DETETOR SERVER OTHER THAN VIA THE EIGER API OR THE WEB INTERFACE – THERE IS NO CONSOLE ACCES TO THE DETECTOR SERVER.
- DO NOT TOUCH THE MYLAR ENTRANCE WINDOW OF THE DETECTOR! The modules will be damaged.

The Technical Specifications document contains important additional information for your specific detector in addition to this manual. Please read it before installing the detector.

4. System Description

The EIGER detector system consists of the following components:

- EIGER detector
- Power supply for detector
- Detector control unit (a Dell PowerEdge system)
- Water chiller
- Data cable(s) (10G Ethernet, Cat6A only)
- Coolant tubing with couplings and hose clamps
- Documentation: this manual, technical specifications, EIGER API

4.1. Hardware

The EIGER X-ray detector operates in "single photon counting" mode and is based on the hybrid pixel technology. X-rays are directly converted into electric charge and processed in the CMOS readout chips. This design has no dark current or readout noise, whilst providing a short readout time, and excellent spatial resolution (a point spread function of 1 pixel). As the quantum efficiency depends on the silicon sensor thickness, the detectors can be used for energies of up to 30 keV, in some applications even higher. The counting rate is greater than $2x10^6$ counts/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.

The EIGER hybrid pixel detector is composed of a silicon sensor, which is a two-dimensional array of pn-diodes processed in high-resistivity silicon, connected to an array of readout channels designed in advanced CMOS technology.

4.2. Software

The EIGER detector system is controlled via the EIGER API, which relies on a http/REST interface. The API documentation is provided as a separate document "EIGER API".

The detector's web interface (chapter 7) gives access to fundamental settings and status parameters and also enables a first test to see if the detector system has been set up properly (after Installation and start up as described in chapters 5 and 6).

The EIGER detector writes images in the HDF5 file format (see chapter 10). DECTRIS provides the image viewer ALBULA, which has the possibility to handle the HDF5 images, with the primary aim to display them. ALBULA is available free of charge for the platforms Linux, Windows and Mac (for download please go to http://www.dectris.com). The ALBULA version for Linux comes with a Python API for handling the HDF5 files that allows performing arithmetic operations on image data, as well as, basic analysis. Furthermore, the API enables seamless integration of the viewer into a beamline infrastructure. More information on HDF5 and ALBULA is given in chapter 10.

5. Installation

5.1. Mounting

The detector has to be mounted as indicated in the technical specifications of the EIGER detector. The chiller has to be installed in a place with adequate ventilation and set up according to its user manual. The detector control unit (DCU) should be mounted in an appropriate 19” computer rack.

5.2. Cables and Tubing

5.2.1. Detector

The detector has to be connected to the external power supply. Please make sure the power switch at the back of the detector is OFF.

Connect the detector's Ethernet ports marked “DATA” ports to the corresponding ports of the detector control unit with the supplied Cat6A network cables only. The number of ports/cables depends on the detector type (X series: 4 ports, R series: 1 port) as shown in the technical specifications document. If more than one cable is required, please pay attention that “DATA1” is connected to Port 1 and so on.

⚠ Proper detector function can only be guaranteed for the original Ethernet cables supplied with the detector.

Finally, plug in the dry air or nitrogen supply. For pipe diameter, flow rate and medium requirements, please see the technical specifications.

5.2.2. Chiller

The tubing needs to be fixed to the chiller and the couplings using the hose clamps supplied. Plug in the couplings in the detector and check the flow direction (i.e. the chiller outlet should be connected to the detector cooling inlet and vice versa). Once the tubing is installed, the chiller has to be filled with coolant and connected to power. For coolant requirements please see the technical specifications.

5.2.3. Detector Control Unit (DCU)

The detector control unit needs to be connected to power, to the detector (as specified above) and to the customer's workstation meant to receive the acquired data via a network connection. See the technical specifications for further information on cables and connectors. Please refer to section 5.3 and 7.2 of this manual for accessing and configuring the network connection of the detector control unit.

Do not connect a mouse, keyboard or screen to the detector control unit, as it is solely controlled via the network interface and all actions are carried out either via the EIGER API (see separate document) or tested via the web interface (see below).

5.3. Accessing the Detector Control Unit

The EIGER detector is controlled via the network interface of the detector control unit. Hence, the IP network address of the detector control unit has to be known to be able to connect to the interface. Depending on the network structure, there are several ways of determining the IP network address, which are described below.

- See the technical specifications for the default network port configuration of your detector control unit.
- The default network port configuration may be changed through the detector's web interface (see section 7.2).

5.3.1. Using DHCP

If there is a DHCP server available on the network, plug the network cable into a port of the detector control unit pre-configured for DHCP. See the technical specifications for the default network port configuration of your detector control unit.

If your detector control unit has a LCD panel in the front (EIGER X systems), the IP network address can be retrieved from this panel.

Alternatively, or if your detector control unit has no LCD panel (E.g. EIGER R systems), the IP network address can be retrieved by searching for the MAC address on the network. For network safety reasons, please ask the network administrator for assistance in obtaining the IP address. If you are the network administrator, and/or have the permission the following linux command can be used to retrieve the IP network address:

- `sudo nmap -sP xxx.xxx.xxx.xxx/24 | awk '/^Nmap/{ip=$NF}/yy:yy:yy:yy:yy:zz/{print ip}'`

where `xxx.xxx.xxx.xxx/24` is network address range to be scanned (e.g. `192.168.0.1/24`) and `yy:yy:yy:yy:yy:zz` is the MAC address of the DHCP network port in the back of the detector control unit. The MAC address of the first network port can be found on the bottom of the service tag label (pull-out label in the front of the detector control unit). The MAC addresses of the second, third and fourth ports are the same as the first one, but with the last two digits incremented by `zz+2`, `zz+4` and `zz+5`, respectively. (e.g. if the first port is `01:23:45:67:89:ab`, then the second port is `01:23:45:67:89:ad`.)

5.3.2. Using a fixed IP

If you want to access the detector control unit using a fixed IP network address, plug the network cable into a port of your detector control unit pre-configured for a fixed IP. See the technical specifications for the network port configuration of your detector control unit and configure your network accordingly.

If you use e.g. a laptop to access the detector control unit directly for the initial configuration, you can use the following network settings on the laptop:

IP address:	`10.42.42.42`
Subnet mask:	`255.255.0.0`
Default gateway:	not required

6. Getting Started

Please make sure the detector, the chiller and the detector control unit are properly mounted and connected according to Chapter 5 and the technical specifications. Before operating the detector make sure you have read the previous chapters of this document as well as the technical specifications.

6.1. Startup Procedure

- Turn on the nitrogen or dry air flow at least 30 minutes before turning on the detector.
- Turn on the chiller and set the operation temperature specified in the technical specifications. Please read the chiller manual, as some models require to be powered and additionally activated in order to operate properly.
- Turn on the power switch at the back of the detector and press the power button in the front of the detector control unit.
- To quickly verify correct operation of the complete detector system, it is possible to record an image through the detector's Web Interface, as described in section 7.3.

7. *Web Interface*

The EIGER web interface provides simple but limited access to the detector system for installation, testing and system update. For productive operation of the detector, please refer to the EIGER API documentation.

7.1. Accessing the Web Interface

To access the detector's web interface please enter the detector's IP address into a browser of a PC which is on the same network as the detector control unit. To determine the detector's IP address, see Section 5.3. The browser will show you the web interface of the detector as shown below.

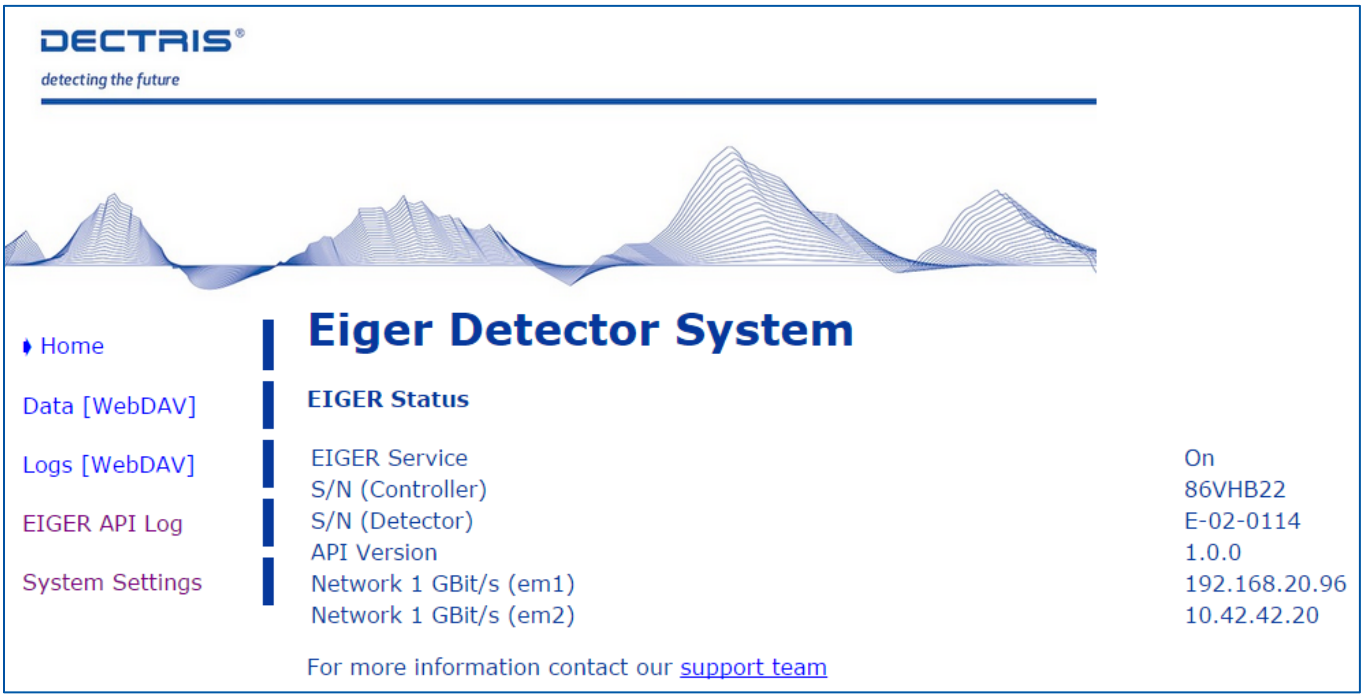

Figure 1: Screenshot of the home page of the detector web interface showing the EIGER status.

The following information is available through the of the web interface menu:

- Home: Status of EIGER service (more detailed through system settings), controller and detector serial numbers, API version and network configuration
- Data [WebDAV]: file listing of the /data/ directory
- Logs [WebDAV]: file listing of the /logs/ directory
- EIGER API log: RestAPI live log
- System Settings: to access the detector control unit system settings (see Section 7.2)

7.2. System Settings

The system settings are password protected to prevent unauthorized access:

User: eiger
Password: #EIGER_Detector#

Once logged in, you will see the web interface for the detector system settings similar to the one shown in Figure 2.

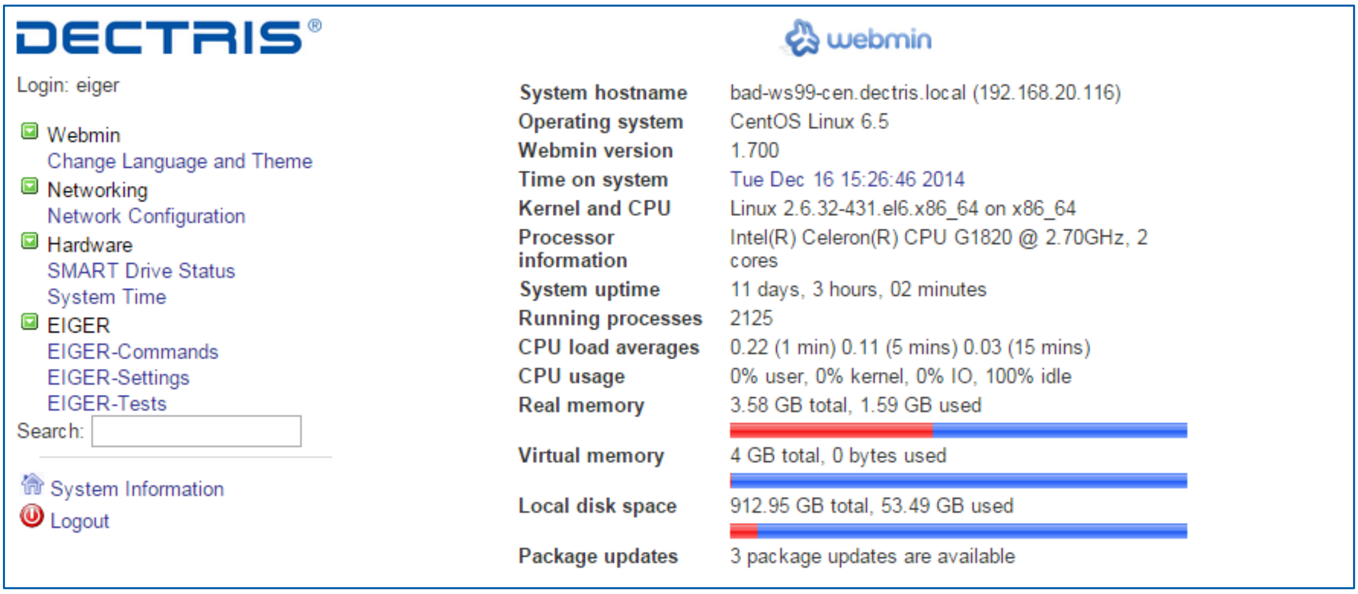

Figure 2: Screenshot of the system settings page showing the menu structure (left) and detailed detector control unit status information (right)

The following menu items are available for configuration/testing:

- Webmin → Change Language and Theme: Configure language and login password
- Networking → Network Configuration: Configure network interfaces, routing and gateways, hostname and DNS client and host addresses
- Hardware → SMART Drive Status: Retrieve hard drive status information
  → System Time: Configure system and hardware time, timezone and time server sync
- EIGER → EIGER-Commands: Detector commands (initialize, arm, trigger, disarm, cancel and abort)
  → EIGER-Settings: Detector control unit commands (poweroff, reboot, upgrade, cable check, restart DAQ)
  → EIGER-Tests: Basic detector tests (EIGER image capture test and EIGER components status)

7.3. Test Image using the Web Interface

In the EIGER-Tests section please execute the command EIGER image capture test. This will test will initialize the connected detector and then record a half second image. After this image is recorded it is possible to download the image by the direct links or from the Data directory (see section 7.1). The images can be displayed using the ALBULA viewer (see section 10).

8. General Usage of the Detector System

8.1. Detector Control and Output

The EIGER detector system is controlled through the EIGER API, an interface to the detector that is based on the http protocol and implemented on the detector control unit. The EIGER API documentation supplied with the system describes this interface in detail and allows to integrate the detector control into its own software easily. Please refer to the EIGER API documentation for details.

The images recorded by the detector can be accessed in two different ways. Images can be stored by the file writer on the detector control unit (RAM disk) as HDF5 container files (for details see section 10), which include meta-data in a NeXus-compatible format. These files have to be subsequently copied to a permanent storage by the user (to prevent an overflow of the available RAM disk, which could overwrite existing images and result in data loss). Alternatively, the images can be directly obtained through the http interface via the detector's monitor interface. Both ways are described in detail in the EIGER API documentation.

8.2. Recording images

To record images or image series the following steps need to be performed through the EIGER API (see the EIGER API documentation for details).

- Initialize the detector (detector and detector control unit must be running; only required once).

- Program the detector parameters for recording (exposure time, exposure period, energy threshold, trigger method, number of images, …) and specify the desired output (File Writer and/or Monitor Interface, bit depth, …).

- Arm the detector

- Send trigger(s) to record the programmed image or image series. Do this once or multiple times (creating auto-named files).

- Disarm the detector (to assure files will be finalized and closed).

- To change settings for further measurements, continue with Program the detector parameters.

- Download the recorded image data: Copy the HDF5 image containers from the server before they are overwritten. OR: Get the images through the Monitor Interface before the monitor buffer overflows.

8.3. Control of the detector from a specific environment

Integrating the detector to a specific environment requires understanding the necessary detector functions. The API reference will list all possible commands and its features, but it does not give an explanation of the required functionality. The main as well as further parameters are explained below.

8.3.1. Main configuration parameters

Following this section allows controlling the detector and record images. The files are recorded, but whether they are written or streamed depends on the configuration of the filewriter, the stream or the monitor. To record images only the following parameters need to be adjusted.

- nimgage
- count_time
- framte_time
- photon_energy

whereas the photon energy needs to be set to the used X-ray energy of the experiment. The timing parameters, frame_time and count time need to be separate at least by the detector_readout_time, which can be read back from the API. The count_time is the time the detector counts photons and the frame time is the time the next image starts. The nimages is the number of times this is repeated. A good way of understanding this is that the detector “thinks” in series of n images. This series is started after a trigger command or an electronic pulse on the external trigger input. To change between the trigger modes one can use the

- trigger_mode

which allows to change between the modes (see section 9 for details). There is a convenience function for photon_energy, called

- element

which will accept as argument the chemical symbol of elements e.g. "Cu", "Mo", ... and set the photon energy to the $K_{\alpha1}$ emission line of the element. From version 1.1 onwards it will be possible to utilize the parameter

- ntrigger

with values >1 which allow several triggers commands or external pulses per arm, disarm sequence. This mode allows recording several series with the same parameters, which can be an advantage for some experiments. If the filewriter is enabled the recorded files are written in a hdf5 file. The filewriter has the following important configuration parameters

- name_pattern
- nimages_per_file
- compression_enabled

where name_pattern sets the name of the file (using the pattern $id adds at the place a sequence number). The parameter sets the number of images per data container. For the parameters nimgaes_per_file please see the www.dectris.com/hdf5 after registration and sign in. A value of 1000 (default) causes that every 1000th image a data file will appear after recording on the data directory. If for example 1800 images are planed to record the arm, trigger, disarm sequence will cause that after the arm the master file will be available on the data directory. The trigger will start the image series and after 1000 recorded images one data container will appear. This will state will not change until the disarm command which finishes the series and closes the second open data container. This will make the data container appear on the data directory.

Please note that recorded image data can also be retrieved in different ways. For details please see the API documentation.

8.3.2. Additional configuration parameters

The following parameters are for special conditions and should not be set with care understanding the consequences. The parameters can clearly negatively influence the data quality!

- threshold_energy (will be set automatically by photon energy to 50% of the possible value, this should only be changed if the suppression of fluorescence is necessary for the experiment: can be used between 50% - 80%, there are no limits from API.)

Corrections can be turned off using the following, commands. This should most certainly not be done by any standard experiment since it requires post treatment of the data. There are very few experiments where this can be an advantage but it can easily be a negative influence.

- flatfield_correction_applied
- pixel_mask_applied (possible in future versions)

Further settings such as the following allow to enter beamline and experiment relevant parameters.

- beam_center_x
- beam_center_y
- detector_distance

- detector_orientation
- detector_translation
- wavelength (but corresponds and acts like photon energy)

Further parameters and their function are described in the API reference manual.

8.3.3. Dependences of parameters

The following parameters are either directly affected by the detector calibration or they affect other parameters that stem from the calibration: flatfield, pixel mask, element, photon energy, wavelength and threshold energy. Changing one of these parameters may let the EIGER API change other parameters as well in order to keep the detector configuration consistent.

- photon_energy: Changing photon energy in generally sets element to an empty string and wavelength to its equivalent. threshold energy is set to photon energy/2. This implies that threshold energy should be set after photon energy or wavelength. flatfield is recalculated.
- element: Setting the element is equivalent to setting the photon energy to the Kα line of the element. Hence photon energy, wavelength and all parameters that depend on photon energy are changed accordingly.
- wavelength: Changing wavelength in generally sets element to an empty string and photon energy to its equivalent. threshold energy is set to photon energy/2. This implies that threshold energy should be set after photon energy or wavelength. flatfields are recalculated.
- threshold_energy: flatfield is recalculated whenever threshold energy changes.
- flatfield: The flatfield has been measured during calibration as a function of photon energy and threshold energy. It may be set explicitly to some value via the API. This affects number of excluded pixels.
- pixel mask: The pixel mask has been measured during calibration. It may be set explicitly to some value via the API. This has no impact on any other parameter.

9. Trigger Usage

9.1. Introduction

In order to record an image or a series of images, the EIGER detector has to be configured, armed and the exposure(s) started by a trigger signal. The detector can be triggered through software (internal trigger) or by an externally applied TTL signal (external trigger). Three different trigger modes are available and described below.

Notes:

For setting the trigger parameters (config/…), the detector has to be initialized first. The sequence of commands, as described in the "EIGER API" documentation, has to be maintained.
See the document "EIGER API" for a description of the configuration parameters.

Depending on the type of EIGER System, the valid values for the trigger parameters differ. Please consult the technical specifications of your system. The values presented in the examples below are on the low side of performance and will work on every EIGER system. If the settings or the external trigger/enable pulses applied are outside the specifications, acquisitions will not be performed, and subsequently crucial information may be missed.

All values used in the example are for demonstrational purposes only and shall be adapted to meet the requirements of your application.

9.2. INTS - Internal (software) triggering

An exposure (series) can be triggered by using a software trigger. This is the default mode.

Example of a detector configuration for an internally (software) triggered exposure series:

config/trigger_mode = ints

config/frame_time = 1

config/count_time = 0.7

config/nimages = 10

Note:

For setting these parameters, the detector has to be initialized first. The sequence of commands, as described in EIGER API documentation, has to be maintained.

See API: Configuration for a description of the configuration parameters.

The detector will start the first exposure after a command is processed. All subsequent frames will be triggered according to the information previously configured by the “frame_time” and “count_time” parameters. The detector will record “nimages” frames.

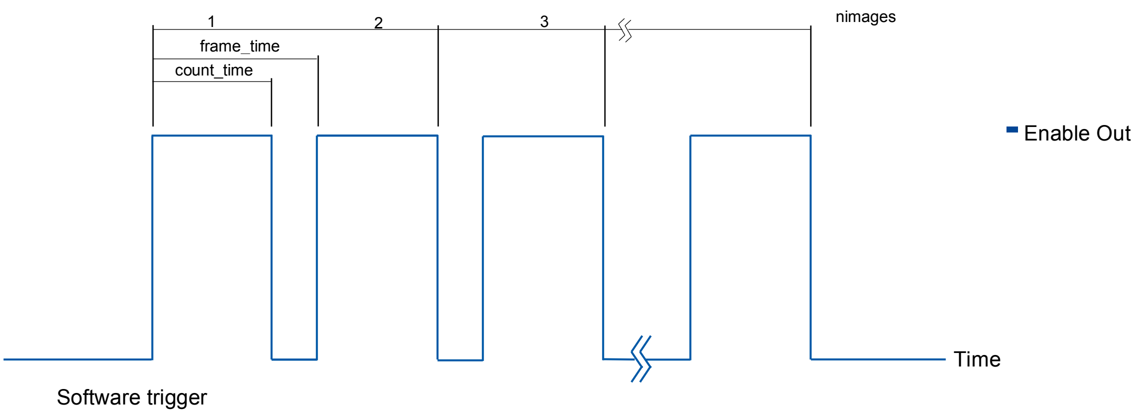

Figure 3: Series of exposures, defined by "frame_time", "count_time" and "nimages", triggered by a software trigger.

9.3. EXTS - Externally triggered exposure

9.3.1. EXTS exposure series with multiple frames, activated by one single trigger pulse

An exposure (series) can be triggered by using an external trigger source.

Below is an example of a detector configuration for an externally triggered exposure series:

config/trigger_mode = exts

config/frame_time = 1

config/count_time = 0.7

config/nimages = 10

Note:

For setting these parameters, the detector has to be initialized first. The sequence of commands, as described in EIGER API documentation, has to be maintained.

See API: Configuration for a description of the configuration parameters.

After the detector has been armed, the acquisition can be triggered by a single external trigger pulse. See the technical specifications for further information on the specifics of the allowed trigger signal.

The detector will start exposing after the trigger signal. All subsequent frames will be internally triggered according to the information previously configured by the “frame_time” and “count_time” parameter. The detector will record “nimages” frames.

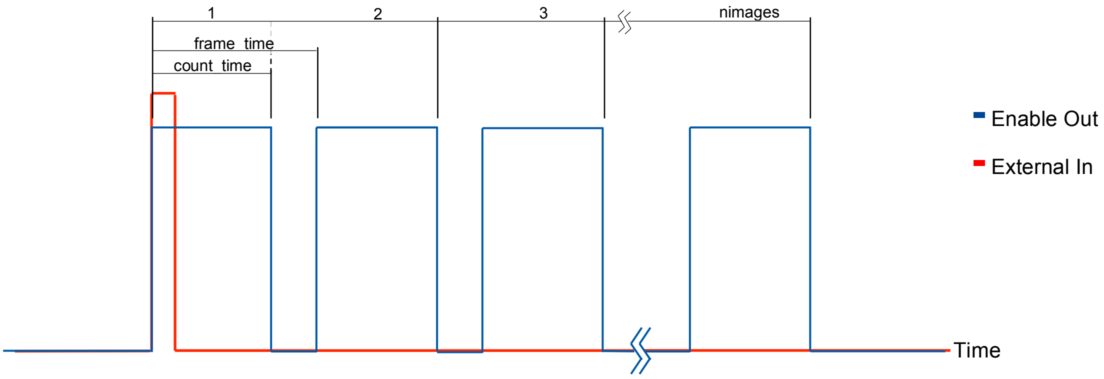

Figure 4: Exposure series defined by “frame_time”, “count_time” and “nimages”, triggered by a single external trigger pulse. Periods not drawn to scale in the time domain.

9.4. EXTE - Externally enabled exposure series with multiple frames activated by multiple trigger pulses

The EIGER detector system also allows externally enabled exposure series

config/trigger_mode = exte

config/count_time (It is advised to set count_time to a value as close as possible to the real enable time.)

config/nimages = 10

Note:

For setting these parameters, the detector has to be initialized first. The sequence of commands, as described in EIGER API documentation, has to be maintained.

See API: Configuration for a description of the configuration parameters.

After arming the detector, the acquisition can be enabled by an external signal. See the technical specifications for further information on the specifics of the allowed enable signal.

The detector will start exposing the first image after the rising edge and stop after the falling edge. In the same matter all subsequent frames are externally enabled. The count time and period are determined by the external enable signal. The detector will record as many frames as valid (according to the specifications) enable pulses are received.

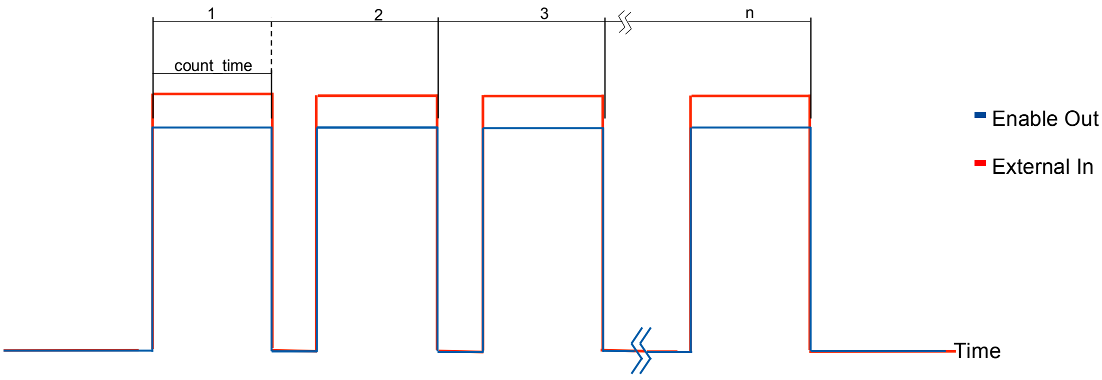

Figure 5: Exposures defined by external enable

The acquisition is controlled by the shape of the external signal. The detector will wait for pulses until the detector has been disarmed by issuing the disarm command.

10. HDF5 and ALBULA

10.1. HDF5 with using ALBULA

ALBULA is DECTRIS' cross-platform image viewer. The Linux version also provides an image library for the Python language.

ALBULA can be downloaded for free at www.dectris.com (after registration). Scripts written in Python using ALBULA can be used to read, display and store data taken by the EIGER (and also PILATUS) detectors.

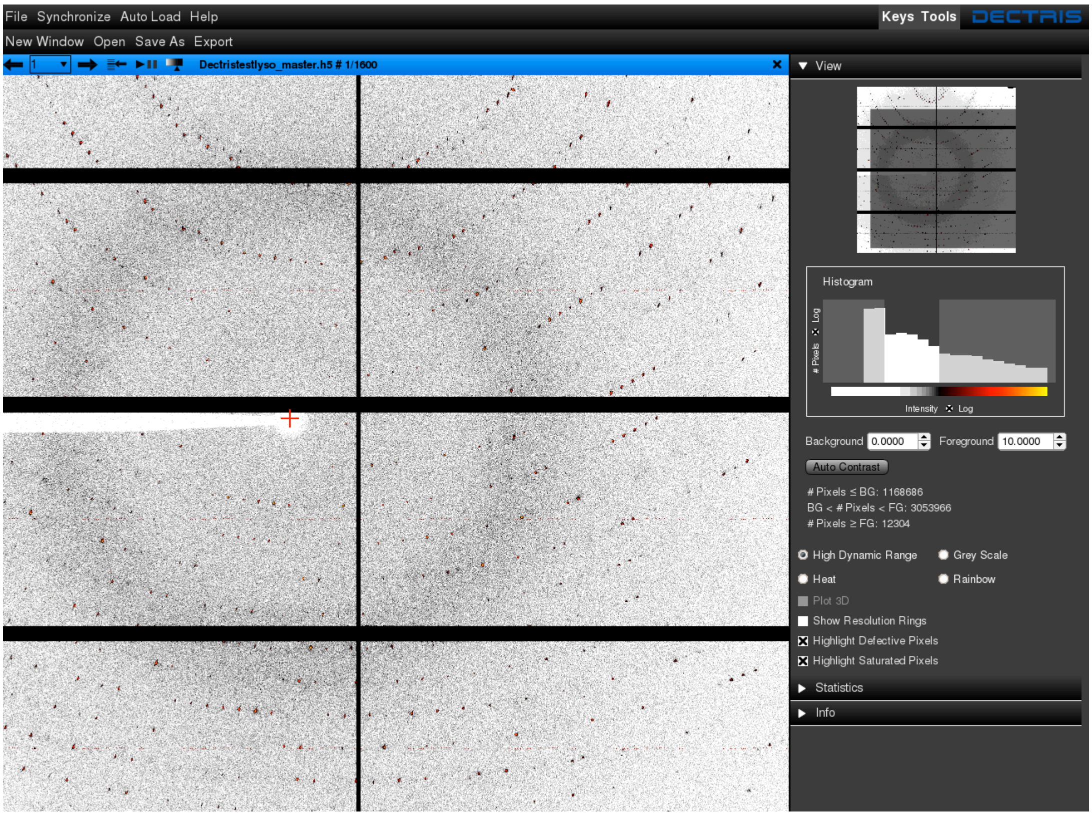

Figure 6: Screenshot ALBULA showing crystallographic data of an EIGER X 4M

10.2. ALBULA HDF5 Python Library (Linux only)

The following examples illustrated how the data stored in HDF5 files by the EIGER detector can be manipulated with ALBULA.

10.2.1. Get Started

```
#!/usr/bin/python

### import the dectris.albula image library ###
import sys
sys.path.insert(0,"/usr/local/dectris/python")
import dectris.albula as albula


def iterateChildren(node,nodeList=[]):
    """ iterates over the children of a neXus node """
    if node.type() == albula.GROUP:
        for kid in node.children():
            nodeList = iterateChildren(kid,nodeList)
    else:
        nodeList.append(node)
    return nodeList
```

```
### open the albula viewer ###
m = albula.openMainFrame()
s = m.openSubFrame()
```

10.2.2. Read the Data

```
### read the compressed (or uncompressed) container through the master file ###
h5cont = albula.DHDF5IntContainer("series_16_master.h5")

### loop over the frames and display them in albula ###
for i in range(h5cont.size()):
    #s.loadImage(img)
    img = h5cont[i]
    ### read header items using convenience functions ###
    optData = img.optionalData()
    ## e.g. wavelength ##
    wavelength = optData.wavelength()
    ## threshold energy ##
    threshold_energy = optData.threshold_energy()


### Read the header item directly without convenience functions ###
neXusHeader = h5cont.neXus()
### print all header item names with path ###
neXusRoot = neXusHeader.root()

for kid in iterateChildren(neXusRoot):
    print kid.neXusPath()

## extract wavelength ##
wavelength = neXusRoot.childElement('/entry/instrument/monochromator/wavelength')
## print value ##
print "wavelength value: ",wavelength.value()
## print attributes ##
for attr in wavelength.attributes():
    print attr.name(), attr.value()

## extract threshold ##
threshold_energy = neXusRoot.childElement('/entry/instrument/detector/threshold_energy')
## print value ##
print "threshold_energy value: ",threshold_energy.value()
## print attributes ##
for attr in threshold_energy.attributes():
    print attr.name(), attr.value()
```

10.2.3. Write the Data

```
### write the (uncompressed) images and the neXus header to a new HDF5 file ###
HDF5Writer = albula.DHDF5Writer("testContainer.h5",1000, neXusHeader)
for i in range(h5cont.size()):
    img = h5cont[i]
    HDF5Writer.write(img)
### flushing closes the master and the data files ###
HDF5Writer.flush()


### write the images in the cbf format. Careful: Information from the header will be lost!
###
for i in range(h5cont.size()):
    img = h5cont[i]
    albula.DImageWriter.write(img, "testImage_{0:05d}.cbf".format(i),
albula.DImageWriter.CBF)

### write the images in the tif format. Careful: Information from the header will be lost!
###
for i in range(h5cont.size()):
    img = h5cont[i]
    albula.DImageWriter.write(img, "testImage_{0:05d}.tif".format(i),
albula.DImageWriter.TIF)
```

10.3. HDF5 without using ALBULA

The EIGER HDF5 data can also be read with programs using the HDF5 library directly. The EIGER data is by default compressed using the LZ4[1] algorithm. In order to decompress the data, the HDF5 plugin filter[2] can be used, see https://github.com/dectris/HDF5Plugin. By setting the environment variable HDF5_PLUGIN_PATH to the path where the compiled plugin filter can be found, the HDF5 library will decompress the data compressed with LZ4 by itself. In case the user wants to use proprietary software like Matlab, IDL or similar, he has to make sure that the HDF5 library version used by this software is at least v1.8.11 in order for the plugin mechanism to work.

For developers using C++, example code can be found on the Dectris website, here after registration and login.

[1] See https://code.google.com/p/lz4/

[2] In order to use the filter plugin mechanism, HDF5 v1.8.11 or greater must be used. See also http://www.hdfgroup.org/HDF5/doc/Advanced/DynamicallyLoadedFilters/HDF5DynamicallyLoadedFilters.pdf