DIAMOND LIGHT SOURCE LIMITED

Merlin: Medipix3RX Quad Readout

Manual and User Guide

Dr. Richard Plackett – Diamond Detector Group; David Omar - Diamond Detector Group

11th November 2015
Version 1.0

This Document provides a reference for operating the MERLN Medipix3 readout developed by the Detector Group at Diamond Light Source Ltd.

Merlin: Medipix3RX Quad Readout, Manual and User Guide

Revision History

Version	Date	Changes	Author
Version 0.1	28/6/2012	Initial Version	Richard Plackett, Detector Group
Version 0.2	29/5/2013	Updated specifications for Medipix3 RX Added Hardware section Added Medipix Overview Added EPICS Interface guide Updated LabView Interface guide Added Equalisation guide Added Step-by-Step guides Added Measurements Section	Richard Plackett, Detector Group
Version 0.3	01/10/2014	Updated to cover Quad software Incorporated IP commands list. Software operation section rewritten. Calibration section rewritten. Remote command section added.	Richard Plackett, Detector Group
Version 0.4	6/10/2014	Fix Endian specification in file format. Added Trigger Out.	David Omar, Detector Group
Version 0.5	20/10/2014	Add Triggering set-up for continuous external control.	David Omar, Detector Group
Version 0.6	19/1/2015	Fix TCP port numbers. TCP Command list updated. Extra clarification in file format. General corrections and enhancements. Added section describing the use of ImageJ to read data files. Some updates are only valid from software version 0.64 onwards.	David Omar, Detector Group
Version 0.7	30/1/2015	Fix TCP data channel description. Fix TCP GET,SOFTWAREVERSION command. Added Acquisition Header file description. Added missing UI information. Added sensor gap description. Reorganised sections to bring some related sections together.	David Omar, Detector Group
Version 0.8	27/5/2015	Added Sensor Image Orientation. Improved Image Data format presentation. Fixed TriggerInTTLDelay and TriggerInLVDSDelay commands. Fixed cross reference and typographical errors.	David Omar, Detector Group
Version 0.9	13/7/2015	Updated for version 0.65 onwards. Added TCP commands for Flat Field and Multi-Gate Triggering.	David Omar, Detector Group
Version 1.0	11/11/2015	Update for version 0.65.9 Improve information on DACs, Equalisation, Calibration, Menus and EPICS. Fix error in POLARITY command.	David Omar, Detector Group

Contents

1. Introduction/Overview

Medipix3 is a photon counting solid state pixel detector typically used with a 300µm or 500µm thick silicon sensor. It has a 256 by 256 array of 55µm pitch pixels and a configurable counter depth, most commonly used in 12 or 24 bit modes. After an exposure the chip takes 820µs to read out, when used with a Merlin system (or 1.64ms in 24 bit mode) after which time another exposure can be taken. There are a number of other modes which give the system greater energy resolving or timing ability and these will be described later in this document.

The Medipix chip is controlled by the Merlin readout system (developed at Diamond) which is based on a National instruments FlexRIO PXI FPGA system, it incorporates an embedded PC and provides a local graphical user interface and a TCP/IP interface to an external control program such as Matlab, TANGO or EPICS. The Merlin system provides an interface to the chip; manages the chips configuration and interprets threshold set in keV to digital units used by the detector. It will either save captured images to a local disk or stream them over the TCP/IP link. The system is capable of taking images at up to 1 kHz repetition rate, however only 1200 frames can be taken at this speed. The maximum continuous frame rate is of the order 100 fps depending on the load on the systems CPU. The Merlin system can perform an exposure (or series of exposures referred to here as an **acquisition**) either immediately as requested, or in response to an external trigger signal. The system is capable of running up to four chips in parallel to allow larger area sensors to be controlled. These are typically arranged in a two by two square configuration or a four by one rectangular configuration and consist of multiple readout ASICs bonded to a single silicon sensor.

2. Technical Specifications

2.1. Detector Specifications

Pixel array (Monochrome)	256 by 256 pixels (single chip detector) 512 by 512 pixels (quad chip detector)
Pixel size (Monochrome)	55µm by 55µm square pixels
Pixel array (Colour)	128 by 128 pixels (single chip detector) 256 by 256 pixels (quad chip detector)
Pixel size (Colour)	110µm by 110µm square pixels
Sensor area	14mm by 14mm (single chip detector) 28mm by 28mm (quad chip detector)
Sensor thickness	300µm or 500µm
Sensor type	Reverse biased hybrid silicon diode array
Nominal sensor bias	80V (120V for 500µm sensors)
Nominal sensor leakage current*	1µA
Exposure time	minimum 1us, no practical maximum
Dynamic Range	12 bit (0 – 4094) or 24 bit (0 – 281,474,976,710,654) configurable
Readout time 12 bit	820µs
Readout time 24 bit	1.64ms
Maximum frame rate (1200 frames)§	1kHz
Maximum frame rate (sustained)*§	100Hz
Threshold range*	5 keV upwards
Threshold Resolution*	240 eV (see section 14)
Pixel Threshold Dispersion (standard deviation)*	645 eV (see section 14)
Pixel Threshold Dispersion (FWHM)*	1.51keV (see section 14)
Point spread function	1 pixel
Maximum trigger response jitter	20ns
Cooling	air cooled‡
Detector power consumption*	1W per Medipix chip
Operating Temperature	10 – 50 C
Detector head dimensions	7 by 5 by 2 cm
PXI chassis dimensions	25 by 20 by 18 cm
PXI Chassis weight	10kg
Communication cable type	68 pin VHDCI to VHDCI Shielded twisted pair
Communication cable length	1m to 10m

* Typical values for information.
§ Sustained frame rate is for 12bit operation on a PXIe-8135 controller with Windows 7 64bit and a solid state drive, provided that sufficient storage is available. The same rate is possible over TCP provided that local storage is not also used and the receiving client can accommodate the data. Merlin can exceed the sustainable rate for short periods due to local storage within the readout hardware and CPU DRAM. See section 8.16 for more details.
‡ The sensor case acts as a heat sink and may be additionally cooled by external conduction.

2.2. Ratings and Inputs

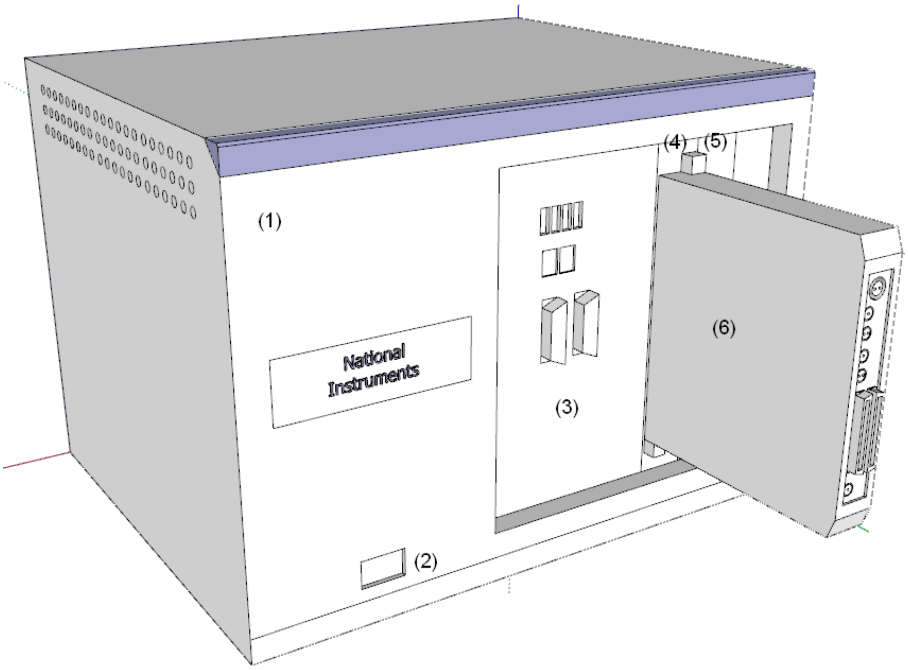
Chassis power input	IEC320/C14	240V (standard pc power lead)
External trigger input (TTL)	Lemo00	3.3V or 5V TTL (rising or falling programmable)
External trigger input (LVDS)	Lemo00 2pin	LVDS (rising or falling programmable)
External trigger output (TTL)	Lemo00	5V rising or falling signal into high impedance
External trigger output (LVDS)	Lemo00 2pin	LVDS rising or falling signal into high impedance
External sensor bias input (Via adaptor)	Lemo00	+/-120V
External sensor bias input (Direct to sensor, when fitted)	HV Lemo	+/-500V

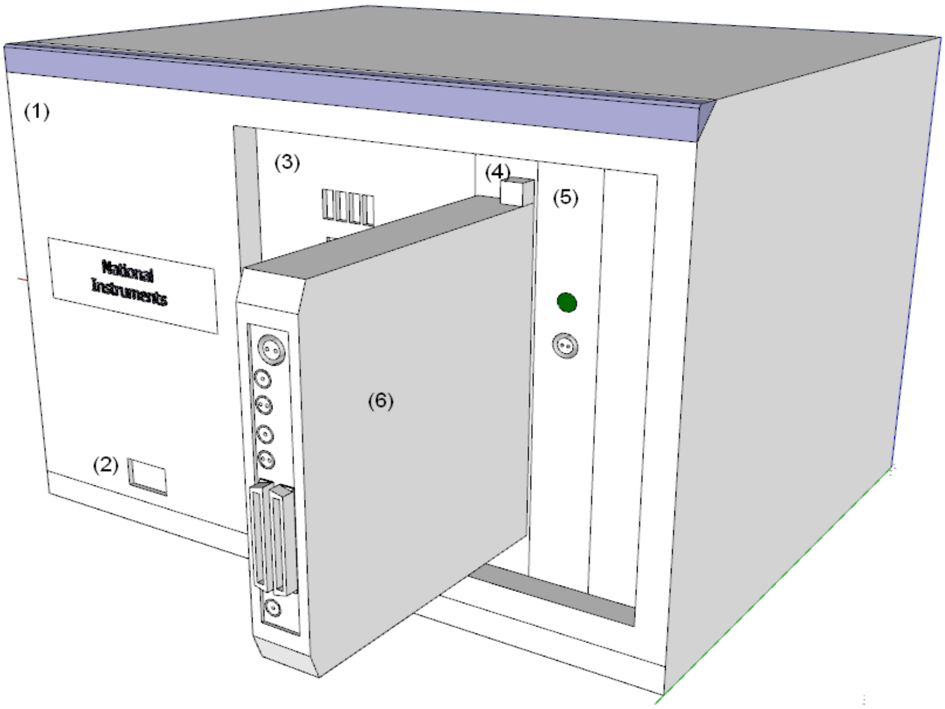
2.3. Radiation Tolerance

The response of the Medipix can be altered by significant exposure to x-rays, if the device is exposed to more than approximately 10^{10} cm^{-2} x-ray photons it could need recalibrating. To avoid this it is recommended to avoid exposing the detector to direct beam or excessive beam reflections. The symptom of requiring a recalibration is an increase in noisy pixels in the general region of the irradiation.

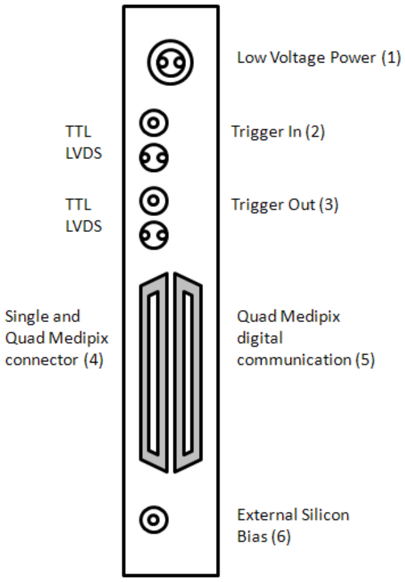
3. Hardware

The system consists of the following components shown in the diagrams below. (1) A National Instruments PXI express chassis with (2) power control for the system. (3) A control PC mounted in the chassis (referred to as the **host PC**). (4) An NI 7962 FPGA flexRIO card mounted in the chassis, (5) A power card that provides a fused-output for the detectors low voltage and (6) an adapter card that converts the digital flexRIO signals into levels and voltages to control the Medipix detector.





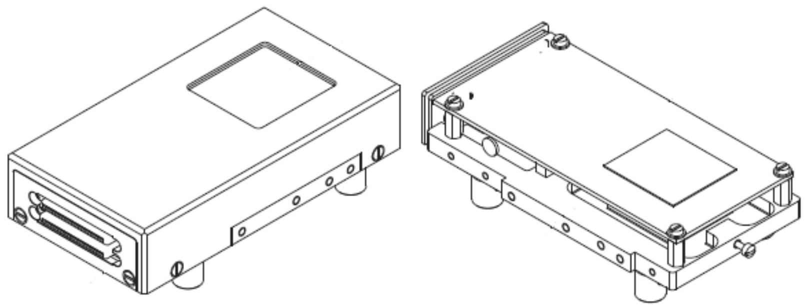
3.1. Adapter Card front Panel



(1) Low voltage power input supplied from the power card
(2) Trigger in via LVDS or TTL, can be programmed to react to rising or falling edges.
(3) Programmable Trigger out to trigger other devices as necessary.
(4) Primary 68 pin VHDCI connector – connects to the single Medipix chip detectors and is the primary connection for the Quad detectors.
(5) Secondary 68 pin VHDCI connector – provides additional digital communication for Quad detector readout. This is left unused with single chip detectors.
(6) External Silicon bias input, enabled via an internal jumper.

3.2. Detector Head Mounting

The Medipix assembly is mounted on a carrier PCB, which is housed in the detector head box. The diagram shows the components and how they are assembled. The chip is mounted centrally with respect to the window and the alignment mark, with its surface approximately 3 mm behind the opening.



IMPORTANT: When attaching the detector head to the Merlin Adapter Card connect the upper of the VHDCI pair on the detector head to the left of the pair on the Adapter card as it faces you. If these are connected the other way round the system will be damaged and may be left completely inoperable.
Some sensors have an additional high voltage Limo connector for connecting external bias. The bias voltage and polarity applied will depend on the details of the sensor. Please refer to the specific documentation for your sensor.

3.3. Heat Sinking

The system should be operated with a thermal contact through the back of the Merlin Detector Head housing to a larger metal mounting structure that will assist in dissipating this heat load. If this is large enough, then the Merlin Detector Head does not require active cooling and can operate in a standalone manner without it. However this does result in the detector head heating up and may result in a slight increase in the minimum threshold. The system can also be run with a small level of active air cooling such a small CPU heat sink and fan arrangement. As this will be strongly influenced by a users' installation this has been left up to the users themselves to implement.

3.4. Internal and External Bias Voltage

The standard Merlin bias voltage can be configured up to 120V and is for use with positively biased (hole collecting) silicon sensors up to 500µm thick. An alternate power card featuring higher voltage and positive and / or negative polarity has been developed for use with CdTe, or other sensors requiring higher bias voltages, or negative polarity. This version of the power card is an option and can be configured with a range of power modules, so the exact specification will vary with application.

When using the alternate power card, or any external source that may be used above 100V, the bias voltage should be configured to feed directly into the sensor, not via the adaptor card and control cable. The control cable connectors are not rated for higher voltages and may break down.

4. Medipix3 Overview

The Medipix3RX pixel readout ASIC (Application Specific integrated Circuit) is the most significant part of the detector system described in this manual. Its functionality sets the envelope for the performance of the system. The Medipix3 ASICs were designed by the Medipix group at CERN, supported by the Medipix3 collaboration and fabricated by IBM in their 130nm CMOS process.

The ASICs are most commonly used bump bonded to a sensor. For the purposes of this document it will be generally assumed the ASIC is coupled to a reverse biased 500 µm silicon pixel sensor with a positive bias applied to the sensitive side. Other sensor bias polarities, thicknesses or semiconductor materials have been used with these ASICs, as well as the related Timepix chips being used with no sensor to directly detect electrons produced by gas amplification chambers or micro-channel plates.

The Medipix3 has three versions currently in existence, 3.0, 3.1 and 3RX. 3.0 and 3.1 are functionally identical with small changes to reduce the performance variation with radiation damage and temperature. Medipix3RX is a significant change to the design that further mitigates these problems and improves the charge summing method to reduce the effect of threshold variation. The current version of the Merlin software only supports Medipix3RX, however older versions can be used to communicate with single 3.0 and 3.1 chips.

Medipix3 is designed as an imaging chip with additional spectroscopic features. In common with its predecessors, Medipix2 and Timepix it provides photon counting with a 55µm pixel size over its 256 by 256 pixel matrix, however in addition it has a number of more advanced features. The increase in transistor density allowed by the move to the 130nm fabrication process has allowed each pixel to contain a significantly greater level of functionality. Specifically each pixel now contains two threshold levels, two discriminators, two 12 bit counters and analogue charge summing circuitry. By combining these resources in a number of different ways a large number of operating modes can be achieved.

4.1. Fine Pitch Sensor Modes

These modes are normally used with 55µm pitch sensors.

Single Pixel Mode

In this mode two thresholds write counts into two 12 bit counters with a defined active shutter open period. By reading only a single counter a simple number of events over threshold counting mode is achieved. At 12 bits the pixel counter will saturate at 4095, after which any further events will not be registered. Six and one bit counter depth will saturate at 63 and 1 respectively. These lower counter depths can be used to speed up the burst frame rate. Merlin defaults to reading one counter, but can be configured to read both.

24bit Single Pixel Mode

By cascading the two counters together one threshold can write into a double depth counter. 24bit counter operation is compatible with other modes with the exception of Continuous Read Write. At 24 bits the pixel counter will saturate at 281,474,976,710,655 after which any further events will not be registered.

Continuous Read Write Mode

By overlapping the reading and writing of both 12 bit counters a mode with no readout dead time is produced. The minimum acquisition time available in this mode is the minimum readout time for a single counter. A single threshold is used for both counters. All counter depths, apart from 24bit can be used.

Charge Summing Mode

With small pixel counting detectors a significant limit on their spectroscopic performance is the effect of charge sharing, where photons or particles falling at the pixel boundaries register as two hits each having a lower energy as the charge is shared between two or more pixels. To overcome this and enhance the spectroscopic performance of the chip Medipix3 has a reconstruction algorithm that operates in the analogue part of the pixel circuitry before the threshold is applied. This charge summing logic compares the charge deposited in neighbouring pixels and allocates all the charge deposited to the pixel with the most charge initially. This significantly enhances the counting accuracy and spectroscopic ability of the detector for a very small degradation in spatial resolution. Two thresholds are used: threshold 0 is used to determine that a pixel has the necessary minimum individual charge to be the centre of reconstruction and threshold 1 gates the final reconstructed charge.

4.2. 110µm Pitch Spectroscopic Sensor Modes

These modes are normally used with 110µm pitch sensors.

Colour Mode

To provide a significant approximation of a full spectroscopic detector, Medipix3 can be operated in colour mode that aggregates the utilities available in 4 pixel electronics to give 8 thresholds and counters in a matrix of 110µm pitch. These thresholds are adjustable so they can be placed above or below energy edges as desired. This allows eight photon counting images to be recorded with different thresholds from a single acquisition.
IMPORTANT: if a 55µm pixel pitch sensor is used, then 3 out of 4 sensor pixels will not be active in the counting process. The electronics from all 4 pixels will be used, but the pixel sensor elements are not integrated. This will have an impact on the count rate for a given flux.

Charge Summing Colour Mode

In this mode the 110µm pixels described above also gain the ability to remove the charge sharing tails from their spectra, with the limitation that this reduces them to only four thresholds.
IMPORTANT: if a 55µm pixel pitch sensor is used, then 3 out of 4 sensor pixels will not be active in the counting process. Charge sharing may not occur as expected and for this reason, colour charge summing should only be used with 110µm pitch sensors.

24bit Colour Mode

This mode gives a 24bit pixel depth, but with only four colour thresholds. 24bit counter operation is compatible with other modes with the exception of Continuous Read Write.

Continuous Read Write Mode

This mode gives no readout dead time, but with only four colour thresholds. All counter depths, apart from 24bit can be used.

5. Starting the System

Generally the Merlin system should start automatically on the chassis being booted and the default user logs in, with the Merlin program running automatically from the start up menu. If this is not set or does not occur for some other reason it may be necessary to start the system manually.

To do this, turn on the PXI chassis with the detector head connected, log in with an appropriate user account and run the Merlin program from the provided shortcut or C:/Program Files/Merlin Quad/Merlin Quad.exe. On a 64 bit system the application will be in C:/Program Files (x86)/Merlin Quad/Merlin Quad.exe.

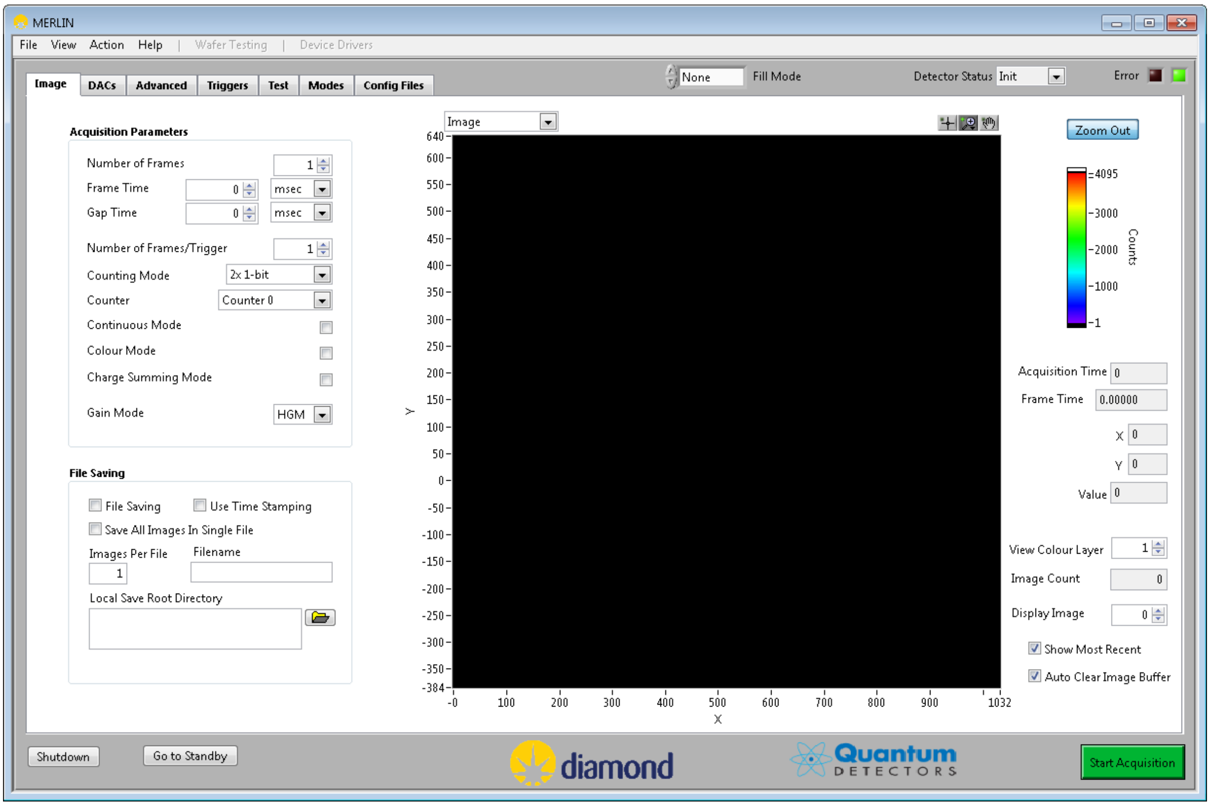
If the system is controlled via a remote system such as EPICS, the remote control system should be (re) started after the Merlin program has been launched and has finished its initialisation.

6. Merlin GUI Interface

This section of the documentation provides instructions for running the system from the Merlin GUI. This method provides the most complete access to all the Merlin functions and routines.

When the system is operated remotely over the TCP/IP link this program is still used and the values set by the remote operation will be displayed on the appropriate front panel. This can be useful for debugging purposes.

6.1. Main / Image Panel



Acquisition Parameters

Several of the acquisition parameters interact and some combinations are invalid. When any of these parameters are changed it is possible that any incompatible parameters will be altered to reflect the last requested mode change. It is suggested that all parameters are reviewed after making changes to ensure that the detector is configured as expected.

Number of Frames – The number of frames that will be taken in this acquisition. If set to 0 the system will acquire frames indefinitely in "Live View" mode, this should not be used in a high frame rate situation. It is not possible to update acquisition parameters while running Live View. It will be necessary to stop the acquisition first.

Frame Time – This sets the time the detector is sensitive for in each frame of the acquisition. This is overridden when the system is set to be externally gated from the Trigger Tab. In this case this is instead used as a de-bounce timer to stop multiple fast triggers form a noisy trigger line. The value set will be FPGA controlled and is accurate to 20ns

Gap Time – This sets the time between frames in an acquisition. Its minimum is the readout time of 820µs per counter read. When external triggers are used this is instead used as a de-bounce timer to stop multiple fast triggers form a noisy trigger line. The value set will be FPGA controlled and is accurate to 20ns. The software will range check the minimum value, depending on the mode selected. In any case the readout will always complete successfully.

Number of Frames Per Trigger – This sets the number of frames that will be taken after a trigger is received. This allows multiple short pulses of frames to be acquired after a trigger signal within a single acquisition. The total number of frames acquired will also be set by 'Number of Frames', even if 'Number of Frames per Trigger' is greater. This parameter also sets the number of frames in a continuous sequence when using Continuous Mode.

Counting Mode – This sets the bit depth of the counters to be read back. This depth affects the readout dead time. It can be set to 2x1, 2x6, 2x12 or 1x24. The default setting is 2x12bit where both counters are available individually (although this does not require both to be read out). If 24 bit mode is set a different internal structure is used so only a single deep counter will be presented.

Counter – This is where the counters to be read are set. It is possible to read counter0, counter1 or both counter0 & counter1. This parameter does not normally need to be set as it is automatically configured by other mode selections. If you need to change this selection, then it should be done last. When changing manually from dual counter to a single counter, normally counter 0 is required, unless using charge summing when counter 1 should be used.

Continuous Mode – This enables the continuous-read-write dead time free readout mode where the two counters alternate acquiring and reading. The minimum acquisition time is the readout time. Only 1bit, 6 bit or 12 bit counter depths are available. The frames per trigger should be set to the number of frames required as if it is set to '1' a series of non-continuous images will be recorded as if each had been triggered independently.

Colour Mode – This switches the chip into 'colour mode' which combines groups of 4 pixel electronics into super pixels each with 8 threshold levels. Checking this box will load the default colour mode configuration files automatically. Four images are acquired with each counter. It is not possible to have all 8 thresholds and have 24bit or continuous read write modes enabled as the second counter is in use. Note that colour mode does not add in the events on the 3 out of 4 sensor pixel elements that have had their electronics borrowed, when used on a 55µm sensor. This mode is best used with a 110µm sensor which will collect events across the whole sensor array.

Charge Summing Mode – This switches the chip into 'charge summing mode' that sums the analogue signals in neighbouring pixels before choosing which to allocate them to. Checking this box loads the default configuration files for this mode. In this mode Threshold 0 performs an arbitrator function and Threshold 1 acts on the summed charge. Threshold 1 should be set above Threshold 0.

Gain Mode – This selects which of the four gain modes the pixels operate in.
SLGM – Super Low Gain Mode
LGM – Low Gain Mode
HGM – High Gain Mode
SHGM – Super High Gain Mode (this can be unstable)

Fill mode - This determines the action to be taken with the large pixels that can be found in the centre of quad sensors where there is a gap between readout chips. The pixels can either be set to zero, have no action taken not inserting the virtual pixels, have the value split across virtual pixels or have the values interpolated across the virtual pixels.

File Saving

File Saving – This checkbox saves images taken in an acquisition to a local Merlin Image Binary (.mib) file on the chassis hard drive in the location specified below.

Use Time Stamping – This option, when enabled, creates an additional folder structure layer that uses a date and time string as its name. This means it is not possible to overwrite data as each measurement will be saved in a new folder even if the filename is not changed.

Save all Frames in Single File – This option concatenates all the frames into a single binary file. This is useful to stop the system writing a very large number of small files.

Images Per File – This sets the number of frames to be concatenated into each binary file. If this is set to 1 each frame has its own file.

Filename – This sets the filename to be used. Where a series of files are to be written a number is appended to separate them. If a new file extension other than .mib is used the system will use this, however it will not change the format of the data saved.

Local Save Root Directory – The folder into which the data will be saved. Time-stamped folders or DAC scans will create subfolders. If the folder does not exist it will be created automatically.

Main Image Display

This normally displays the image currently set in the internal buffer (usually the last image taken). It can also be used to view the mask, trim or test bits, or to view an integral image of all the frames in the buffer. The image will resize depending on the number of chips, their mode and their arrangement. You are able to zoom and change the colour scale. Right clicking allows you to set mask and test bits on individual pixels interactively.

It is also possible to save single frames after the data has been taken even if the file saving checkbox on the image tab was not ticked. To save images in this way use the display image control on the image tab to select the image from the image buffer that you wish to save and then go to **file>save image**. This will prompt you for a file name and then save the file in the binary format described above. The other image views can also be saved in this way.

Image Buffer Controls

Image Count – This shows the number of frames taken. The buffer is limited to the last 1000 frames, but this does not affect file saving.

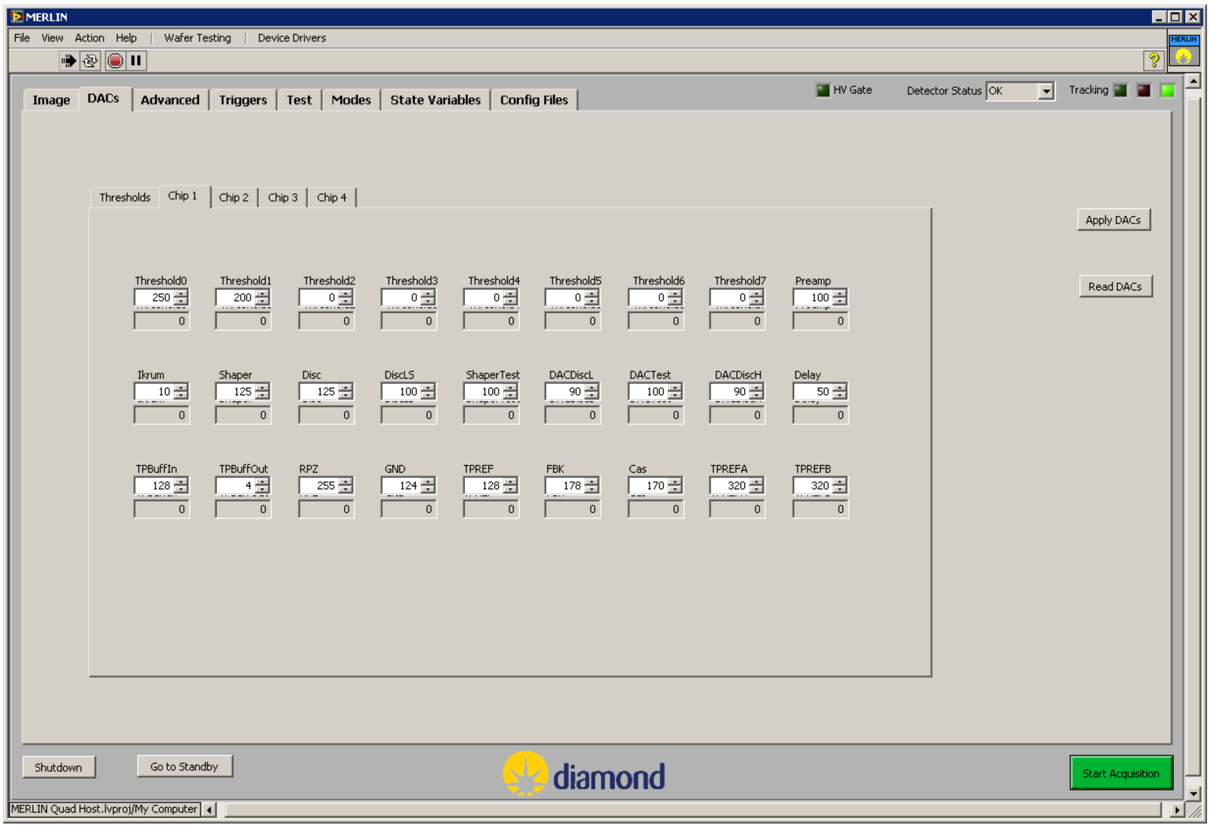
Display Image – This allows you to select which frame from the buffer you wish to show in the main display.

View Colour Layer – In colour mode this allows you to scroll through 8 colour layers per frame, each corresponding to a different counter.

Show Most Recent – This jumps Display Image to the latest image taken.

Auto Clear Image Buffer- This clears the image buffer before each acquisition, so only frames from the most recent acquisition will be available.

6.2. DAC Control Tab

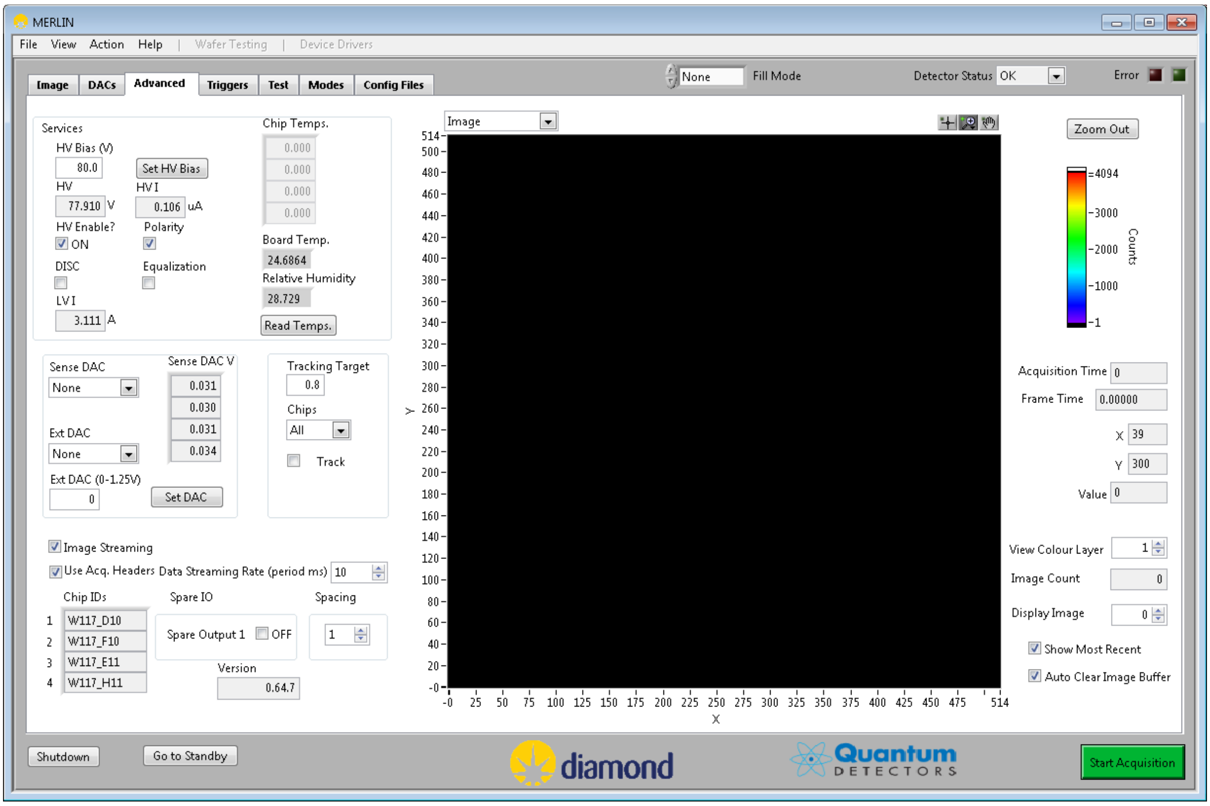


Thresholds Tab
This allows the 8 thresholds to be set across multiple chips at the same time in keV. To set the values entered click Apply Thresholds on versions 0.63 and earlier.

Chip 1-4 tabs
These tabs allow you to individually set all the global chip (ie not pixel) parameters available for each chip. In general this should not be used in normal operation although the thresholds can be set in this way in DAC codes. This is an arbitrary unit that is related to keV in each mode by a threshold mapping calibration file. The values of the DACs currently set in the chips can be read back by clicking Read DACs and new values uploaded by clicking apply DACs on versions 0.63 and earlier.

WARNING: incorrect DAC settings can cause the chip to operate in an unstable manner or cease to function. It is recommended that the system be operated on the default DAC settings supplied, and these only be adjusted by an expert operator. Some DACs will affect equalization and calibration of the sensor.

6.3. Advanced Tab



Services

HV Bias (V)[1] – This sets the target bias voltage to apply to the sensor. This shouldn't exceed 130V. For sensors or adaptor cards with external bias configured, this control is not used and should be set to its minimum value. This parameter may be set in the .cmd file accompanying the sensor (see section 7.5 Command File, below).
Set HV Bias – Applies the target voltage to the sensor.
HV – reads back the voltage applied to the sensor
HV I – reads the current drawn by the sensor.
HV enable – turns on or off the HV module.

Polarity – this changes the charge collection polarity on the chips. When the box is checked the ASIC is in hole collection mode, when unchecked it is in electron collection mode. This does not affect the sensor HV. This should match the sensor bias polarity, hole collection for positively biased sensors and electron collection for negatively biased sensors. This parameter may be set in the .cmd file accompanying the sensor (see section 7.5 Command File, below).

DISC – This changes the DISC_SPM_CSM set on the chips. This sets the routing of the data from counter1 to counter0. This is generally managed automatically by the modes on the main panel.

Equalisation – This changes the equalisation bit on the chip that enables the second counter to be completely independent of the first. This is generally managed automatically by the modes on the main panel.

---

[1] These controls apply to the internal HV Bias supply on the Merlin Adaptor card. They do not apply if external bias is used, although in this case the internal supply should be disabled.

LV I – This reports the low voltage current drawn by the detector head.

Read Temps – This initiates a read operation for the various temperature and humidity sensors. This should not be used during an acquisition sequence as it will attempt to communicate with the readout ASICs and disrupt the acquisition.
Board Temp – This reads an independent sensor coupled to the chip carrier board thermal plane. This is considered to be the most reliable temperature reading.
Relative Humidity – This reads the humidity from the same sensor.
Chip Temps – These temps are calculated from a sensor within the readout ASICs, although they are in a more advantageous position, they require greater calibration and so should not be relied upon at this stage.

Sense and External DAC controls

Sense DAC – This selects the DAC, internal to the ASIC to read the analogue voltage of.
Sense DAC V – This reports the analogue voltages of the set DAC for all four chips.

Tracking Target – This sets the target value to which the "**Sense DAC**" should be set.
Chips – This sets the chips to which the tracking procedure should be applied.
Track – This runs the tracking procedure, automatically adjusting the digital input to the DACs until the output matches the target set. When the operation is complete the box will uncheck.

Ext DAC – This is the DAC you wish to override with the external input.
Ext DAC Input – This is the analogue value you wish to apply to the DAC selected above.
Set DAC – This applies the value entered in Ext DAC Input

Other Controls and Information

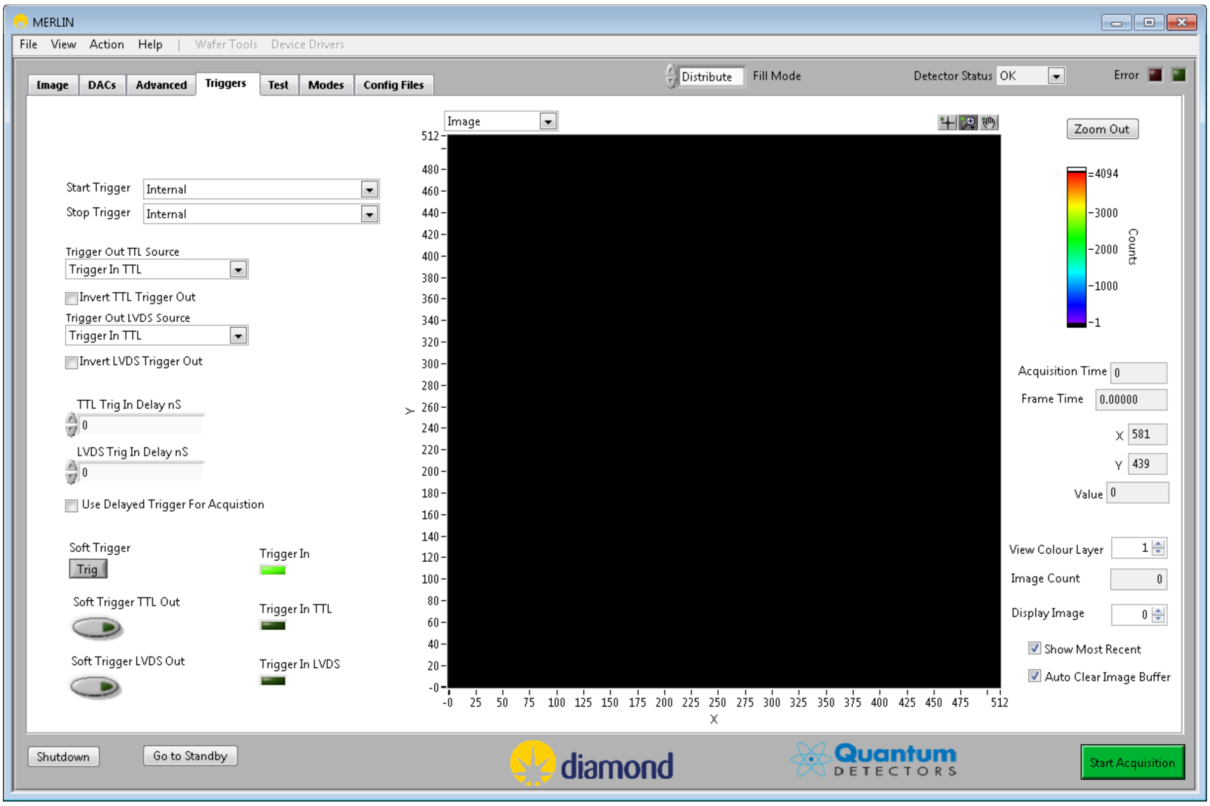
Image Streaming – allows you to turn image streaming over TCP/IP on and off.

Data Streaming Rate– This sets the maximum rate at which images should be streamed over the TCP/IP link, this is useful if the system is causing overload on networking equipment downstream.

Use Acq. Headers – Turns on and off the generation of text acquisition header files.
Chip IDs – Displays the decoded ID codes read back from the chips.

Spacing – Takes images with a fraction of the image masked and then summed back together.
Spacing 1 = one image at 100% active
Spacing 2 = four images at 25% active
Spacing 3 = nine images at 100/9% active etc
This allows the matrix to be run with a less intense hit rate for debugging purposes. Use of this mode will limit the detector to single frame acquisitions using internal triggering. Other combinations will not work. Set spacing back to 1 for normal operation.

6.4. Triggers Tab



Start Trigger

Internal – The frame or sequence will start as soon as the Start Acquisition button is pressed or as soon as the previous has finished.
Rising/Falling Edge (LVDS) – The frame or sequence starts when a rising or falling edge is detected on the selected line (TTL In, or LVDS In).
Soft Trigger – The frame or sequence starts when a **Soft Trigger** message is received on TCP. Note that **Start Acquisition** configures the readout system before the acquisition is started. **Soft Trigger** is used once the system is set up so it has lower latency.
Multi Trigger Rising/Falling Edge (LVDS), or Soft Trigger – The frame starts when a rising or falling edge is detected on the selected line (TTL In, or LVDS In, or Soft Trigger). A frame is accumulated over multiple triggers. [2]

Stop Trigger

Internal – The frame or sequence will stop when it reaches the frame time set on the main panel.
Rising/Falling Edge (LVDS) – The frame or sequence will stop when a rising or falling edge is detected on the selected line (TTL In, or LVDS In).
Soft Trigger – Same as **Internal** for **Stop Trigger**.[3]

[2] This option is available from version 0.65 onwards.
[3] This option has been removed from 0.65 onwards.

Trigger Out TTL/LVDS Source

Trigger In TTL/LVDS (Delayed) – The Trigger Out signals follows the input Trigger signal, with an optional delay set by **TTL/LVDS Trig In Delay nS**.
Follow Shutter – The Trigger Out signals follows the Shutter.
One Per Acq Burst – The Trigger Out signals follows the burst defined by **Number of Frames Per Trigger**.
Shutter & Sensor Readout – The Trigger Out signals follows the Shutter, but does not go inactive until the frame data is read from the Medipix3 chip.
Busy – The Trigger Out signals follows the Shutter, but does not go inactive until all data has cleared the FPGA buffer. Note that data may still be stored in the host PC and so this signal does not indicate that all data has been written to disk, or sent to TCP.
Soft Signal – The Trigger Out signals follow the Soft Trigger Out controls. [4]

Invert TTL/LVDS Trigger Out

The Trigger Out signal is normally low with a high going pulse. This control inverts it. Please note that changing this control will lead to a change in the level of the Trigger Out signal, so it should be set before any external equipment is sensitive to the trigger.

TTL/LVDS Trig In Delay nS

A trigger delay value which will be rounded to 10nS intervals.

Use Delayed Trigger for Acquisition

When selected the appropriate **Trigger In Delay** is used to delay the acquisition from the trigger signal selected by **Start Trigger**. This has no effect on **Internal**, or **Soft Trigger**.

Soft Trigger

This button initiates a Soft Trigger and only has any function if the **Start Trigger** is configured to use it.[4]

Soft Trigger TTL/LVDS Out

These buttons controls the state of the Trigger Out TTL/LVDS Signals if the source is configured as **Soft Signal**. The on and off signal levels are subject to the output inversion control. [4]

Trigger In

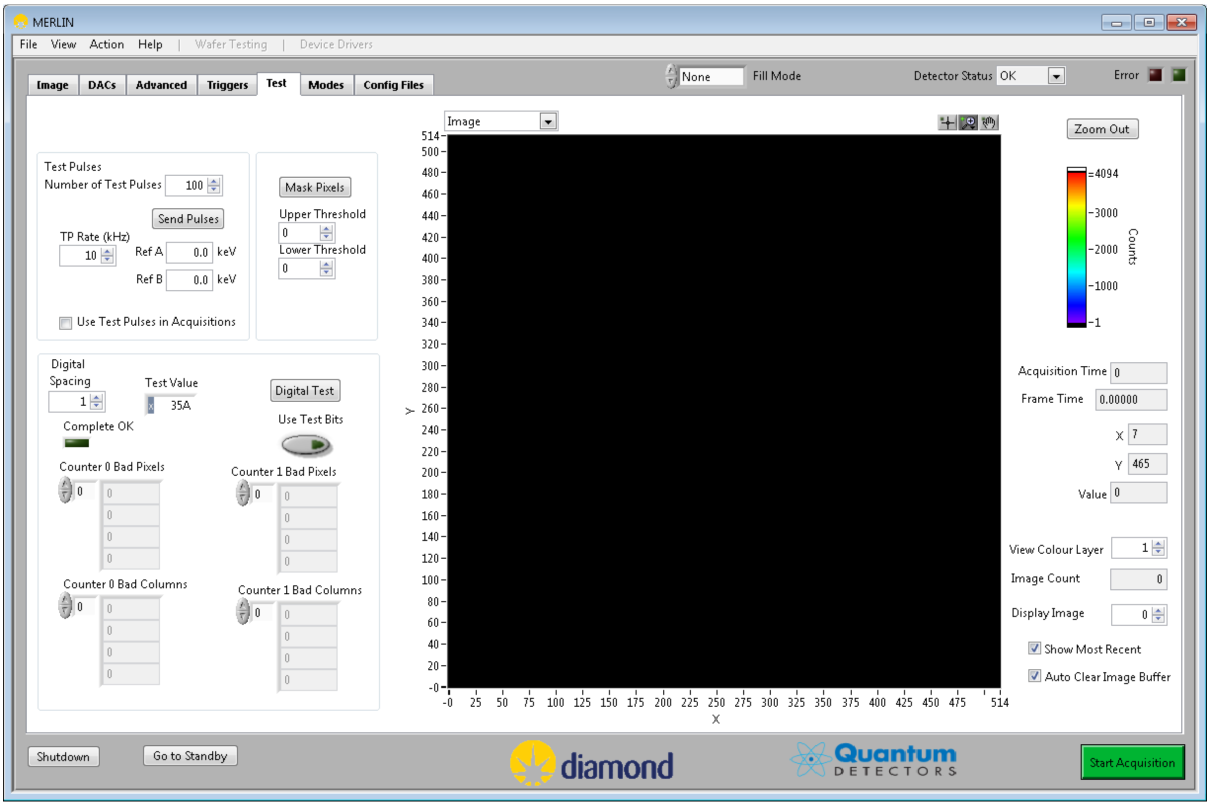
This indicator is active when the selected **Start Trigger** is received. [4]

Trigger In TTL/LVDS

These indicators show the raw state of the respective Trigger In signal. Pulsed inputs will be subject to under sampling. [4]

[4] This option is available from version 0.65 onwards.

6.5. Test Tab



Test Pulses

Test pulses are a method of injecting charge pulses into the front end of selected pixels to test their analogue functionality.

Number of Pulses – This sets the number of pulses to be sent in the image.
Send Pulses – This immediately takes an image with the pulses in.
TP Rate – This sets the rate at which the pulses occur.
Ref A – This sets the (very approximate) keV value of the amplitude of the 'A' population test pulses.
Ref B - This sets the (very approximate) keV value of the amplitude of the 'B' population test pulses.
Use Test Pulses in Acquisitions – This sets test pulses to appear in normal acquisitions and DAC scans.
Test Pulse Pattern – for pulses to work a pattern must be set to define which pixels are active. This can either be done from menu->File->load config and a .tst file loaded, or two shortcut patterns: menu->file –>load logo test pattern and menu->file->load quad test pattern.

Mask pixels

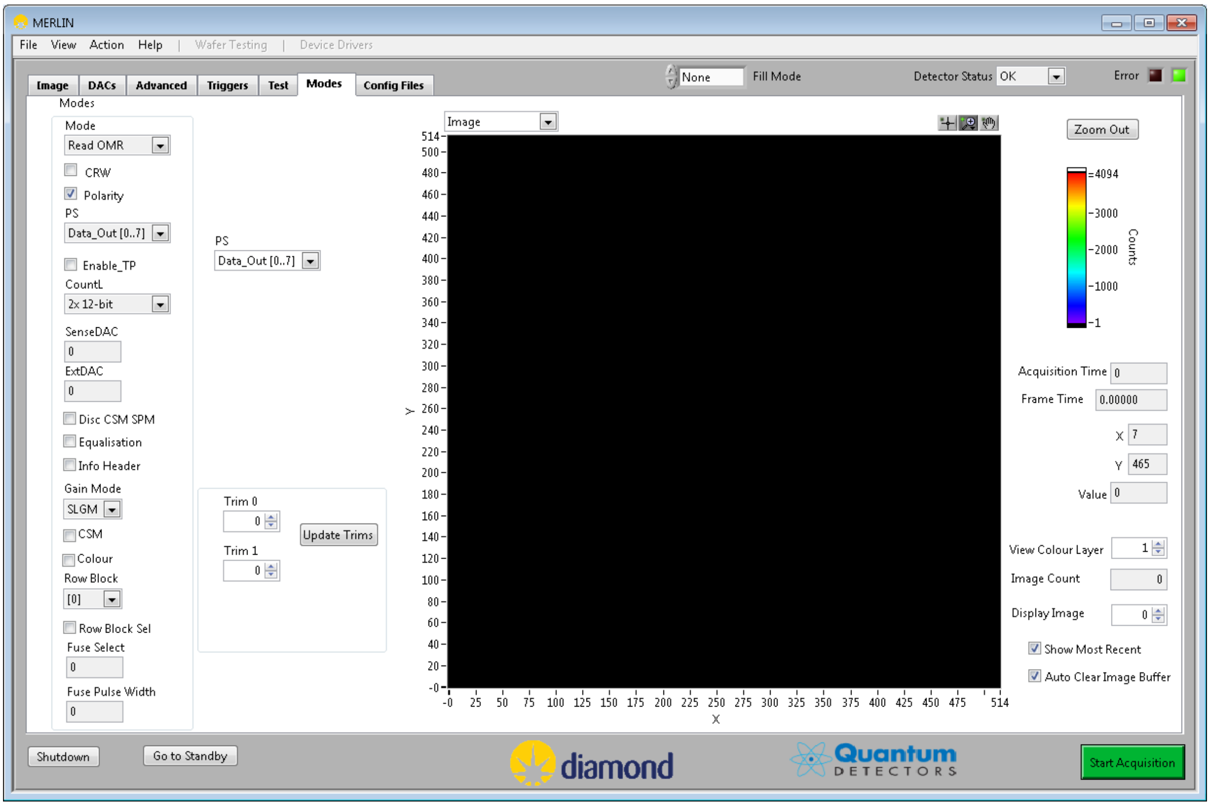
Mask Pixels – masks pixels with more counts in the current image than the **Upper Threshold** and fewer than the **Lower Threshold**, e.g.
With both set to zero, any pixels with counts in will be masked.
With the **Lower Threshold** set to zero and the **Upper Threshold** set to a value above which the pixel should be considered as "noisy" or hot, those pixels will be masked.
With the **Lower Threshold** set to one and the **Upper Threshold** set to the maximum count in the image (see the Image Viewer for image statistics) or use 4095 for 12 bit images, then dead pixels with zero counts will be masked.

With both **UPPER** and **Lower Thresholds** set appropriately both "dead" and "noisy" pixels may be masked in one operation.
While the Pixel Configuration is now set for the current operation and mode, it needs to be saved for any future use or for different modes of operation.

Digital Test

Digital test writes and reads back a pattern of bits to test the functionality of the digital part of the chip. This can also be a useful test of the installation cabling.
Test Value – If this is non zero, then this value is written to the counters in the pixel matrix and read back.
Digital Spacing – If **Test Value** is non zero, then this selects the number of consecutive bits holding the same value.
Use Test Bits – uses the same test pattern as may be used with the analogue test pulses described above.
Counter 0/1 Bad Pixels/Columns – This section returns the results summary of the digital test. A bad column is determined by the number of bad pixels in a chip column being above a fixed threshold.
The digital test will put six images into the display buffer and these can be reviewed if necessary. These are three pairs of counter 1 and counter 0: test pattern; inverse test pattern and match result. The match images will show zero counts for a match, i.e. a blank image.

6.6. Mode Tab

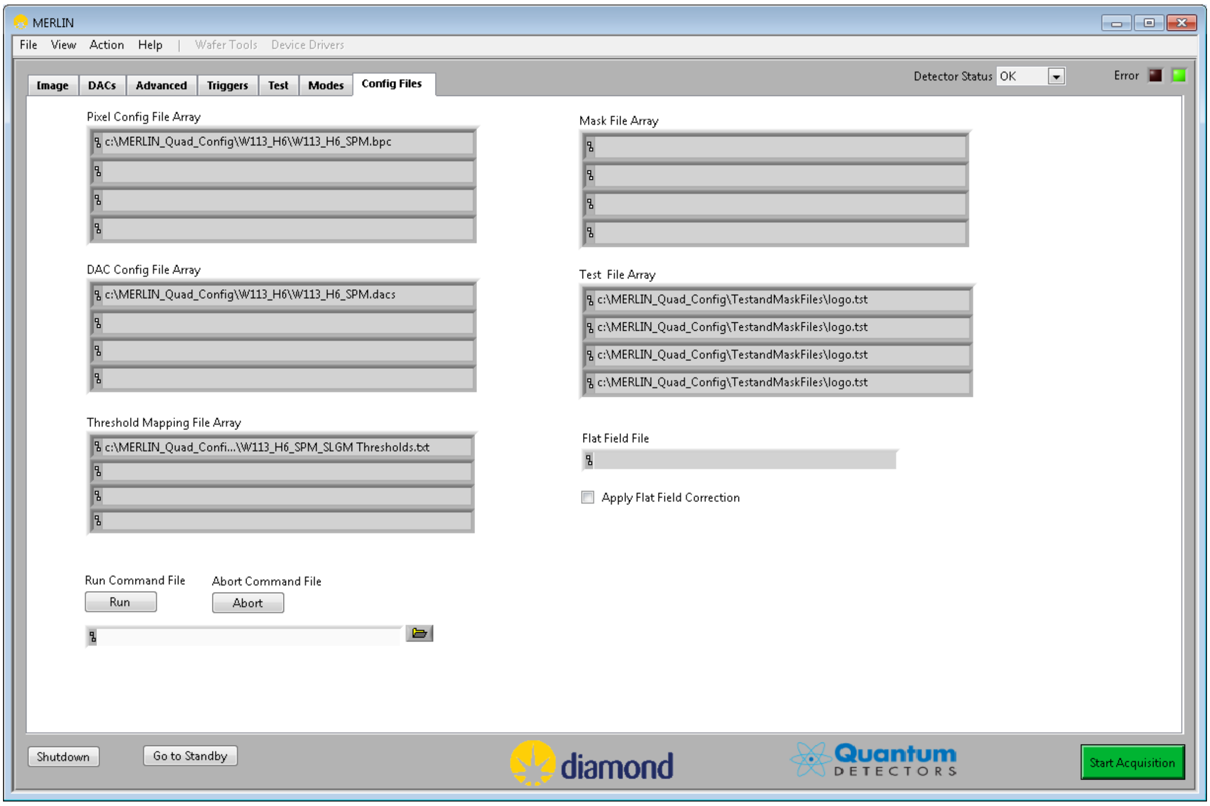


The majority of this tab is to give low level information about the mode being set. It should not be used to control the mode of the chip, which should be done by the controls on the main image panel.

PS – This control sets the chip configuration for the number of data channels the chips communicate on. This is used if a custom interface with less than then full 8 data lanes is used.

Update Trims – This manually sets all the trim bits in Threshold0 and Threshold1 to a specified value. This is used for debugging purposes. However using this option will override the preset equalisation values and these will need to be reloaded for the chips to operate in a useful manner afterwards.

6.7. Configuration Files Tab

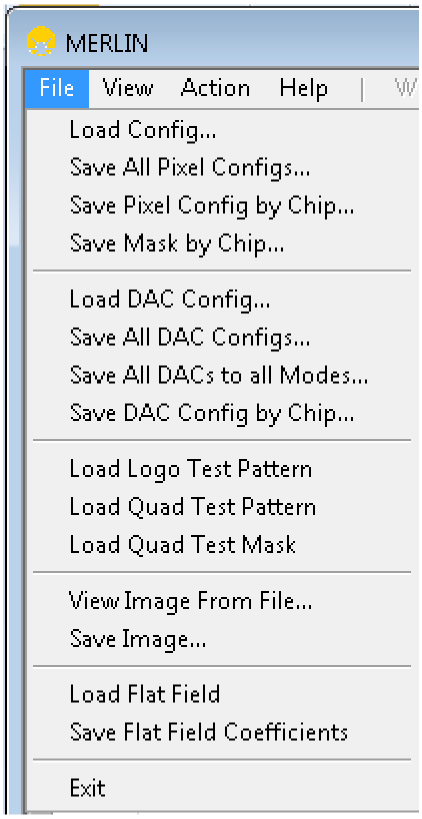


This tab displays the currently loaded configuration files for all the chips being controlled. Each front end mode has a .bpc file, a .dacs file and four threshold mapping files, one for each gain mode. Full descriptions of the configuration files are given in section 7 Operational Modes and Configuration File Management below.

Run Command File
This allows a custom command file to be run at any point, not just at start-up. The command format is the same as that defined for TCP in section 13 TCP/IP Remote Command Reference below.

Apply Flat Field Correction
This control only has any effect if the Flat Field File is selected from the File Menu.

6.8. File Menu



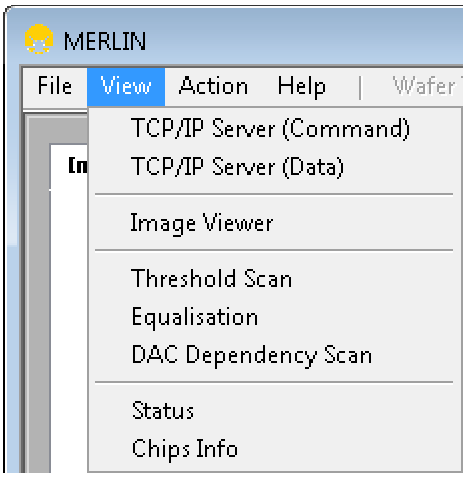
Full descriptions of the configuration file menu items, including Pixel, DACs, Test and Mask load and save commands are given in section 7 Operational Modes and Configuration File Management below.

View Image From File – loads a Merlin MIB image into the buffer which is then available to the Image Viewer.

Save Image – saves the currently displayed image on the Main Image Panel.

Exit – shuts down the system.

6.9. View Menu



TCP/IP Server (Command) – opens the TCP/IP Server Status Window see section 6.11 below.

TCP/IP Server (Data) – opens the TCP/IP Data Channel Server Status Window.

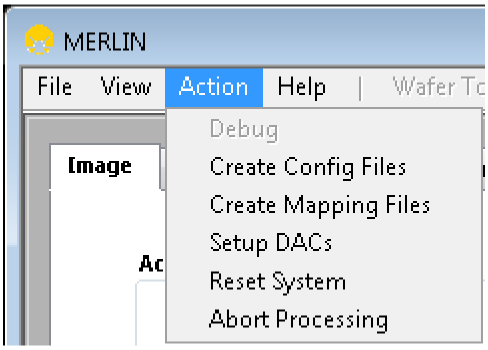
Image Viewer – opens the Image Viewer, see section 6.14 below.

Threshold Scan – opens the Threshold and DAC scan window, see section 10.2 Equalisation below.

Equalisation – opens the Equalisation scan window, see section 10.2 Equalisation below.

Status – opens the Status Window see section 6.13 below.

6.10. Action Menu



Create Config Files – will create a set of configuration files for your sensor. A full description is given in section 7.7 Saving Configuration Files below.

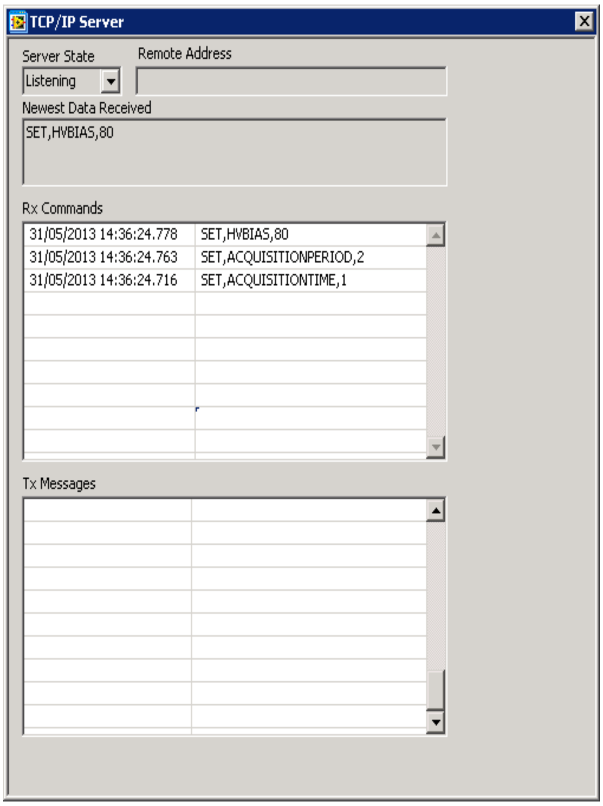
Create Mapping Files – Edit threshold mapping. A full description is given in section 10.7 Threshold Calibration below.

Setup DACs – this will setup GND, FBK and CAS DACS to preset levels. A full description is given in section 9.1 Setting Voltage Levels for GND, FBK and CAS below.

Reset System – will reinitialise the system hardware and reload the configuration files.

Abort Processing – this will terminate the current acquisition. Unlike the normal stop acquisition, which will return all already captured data; Abort Processing will discard any data that has not already been saved.

6.11. TCP/IP Server Status Window



This window is accessed from the menu: **View –> 'TCP/IP Server (Command)'**. It is a good place to begin debugging communication problems from remote scripts.

Server Status
'Connected' means a session is in progress, 'Listening' means the server is waiting for a connection to be established.

Newest Data Received
A copy of the most recent command received.

Rx Commands
A list of received commands and the times at which they were received.

Tx Messages
A list of responses transmitted and the times at which they were sent.

Remote Client Address
The address of the remote client connected.

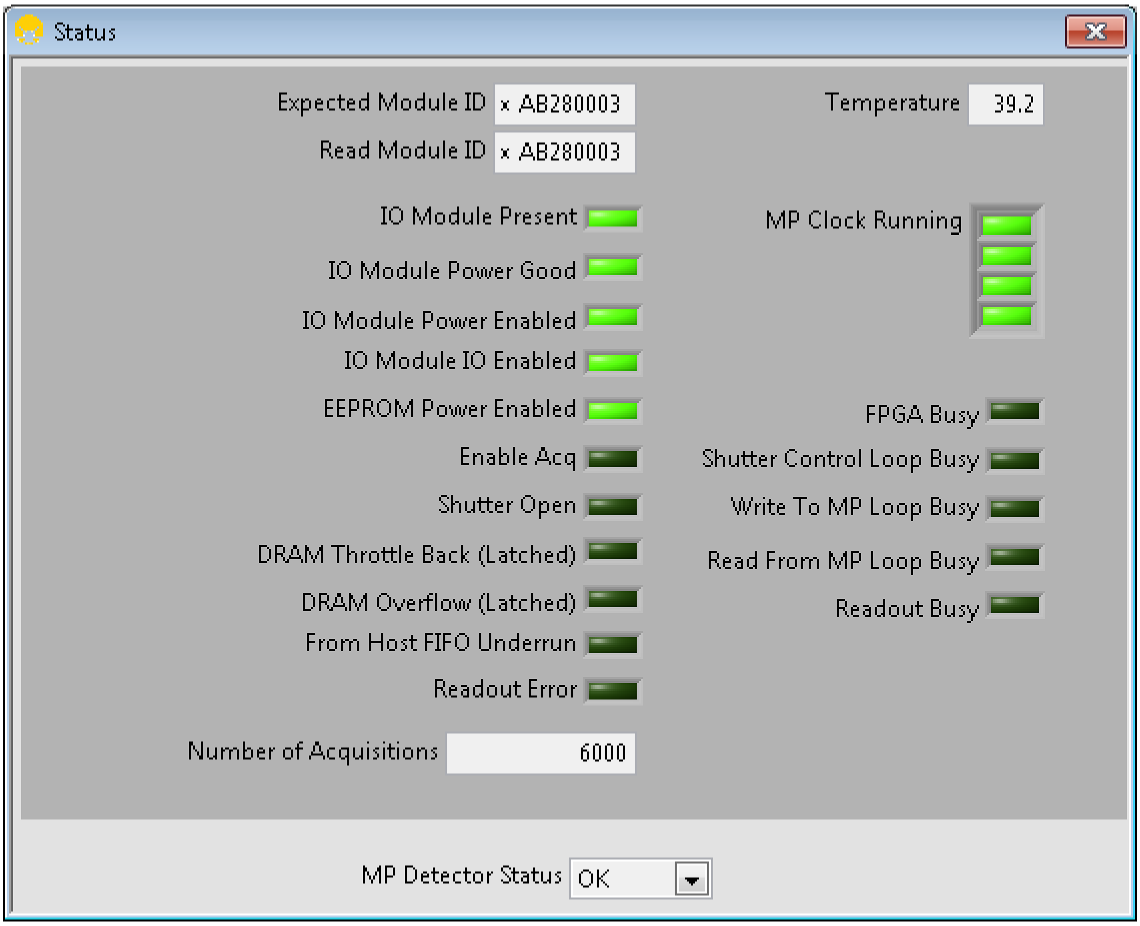
6.12. TCP/IP Data Server Status Window

This window is accessed from the menu: **View –> 'TCP/IP Server (Data)'**. It shows the status of the data connection.

Server Status
'Connected' means a session is in progress, 'Listening' means the server is waiting for a data connection to be established.

Remote Client Address
The address of the remote client connected.

6.13. Status Window



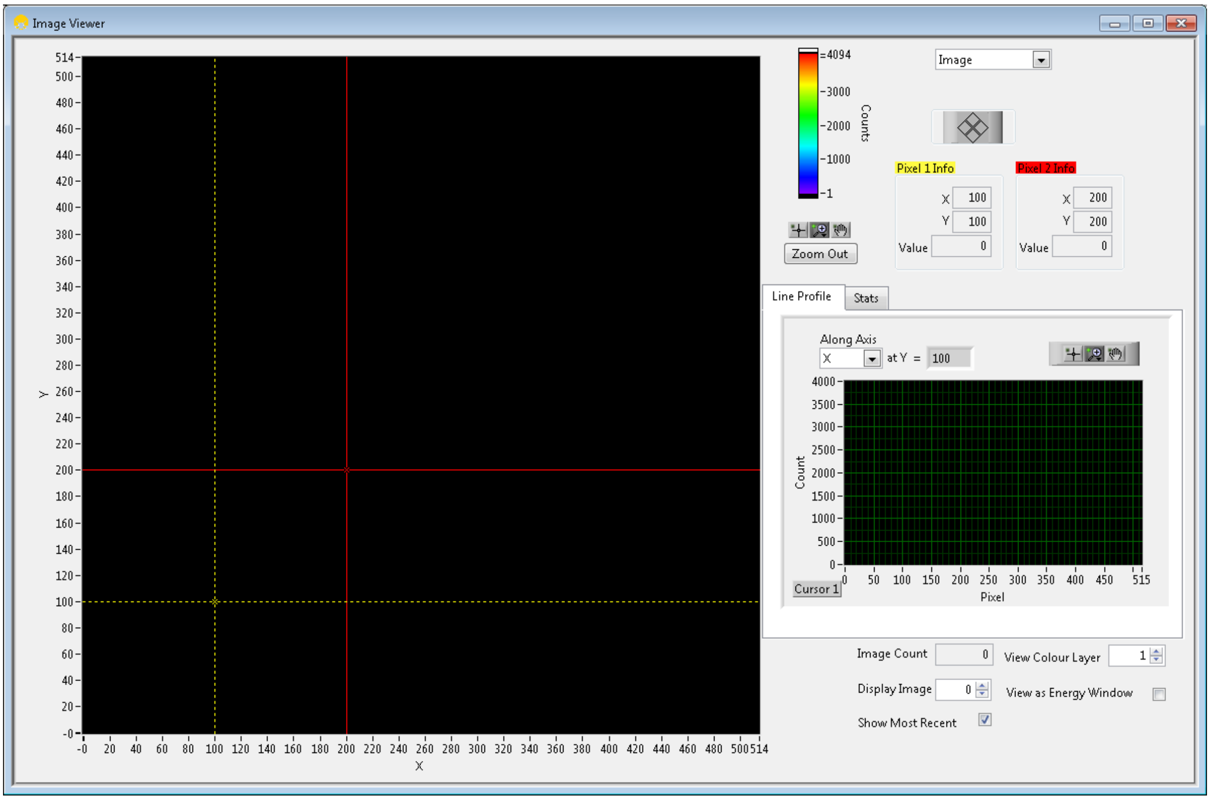
The Status Window provides a quick health check on the system. It is accessed from the menu: **View –> Status**. The normal idle status of a system with a quad sensor is shown above. **Enable Acq**, **Shutter Open** and the **Busy** status indicators, are all expected during normal operation. The status is sampled and may show the effects of under sampling. This is normal, these are indicators only.

DRAM Throttle Back (Latched) indicates that the current acquisition has exceeded the system performance limits. See section 8.16 Acquisition Performance Limits below for more details.

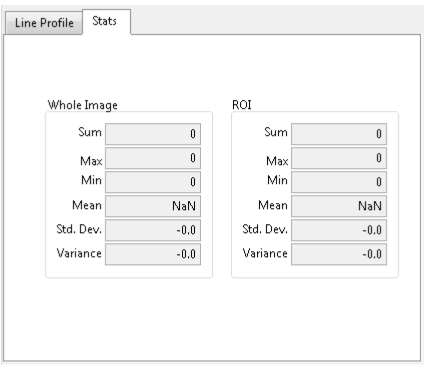
DRAM Overflow (Latched), **From Host FIFO Underrun** and **Readout Error** should not occur and may indicate either a hardware, or installation fault.

MP Clock Running indicates the sensor chips are connected. If these lights are out then there may be a connection problem. Not that only one light will be on for a single chip sensor.

6.14. Image Viewer

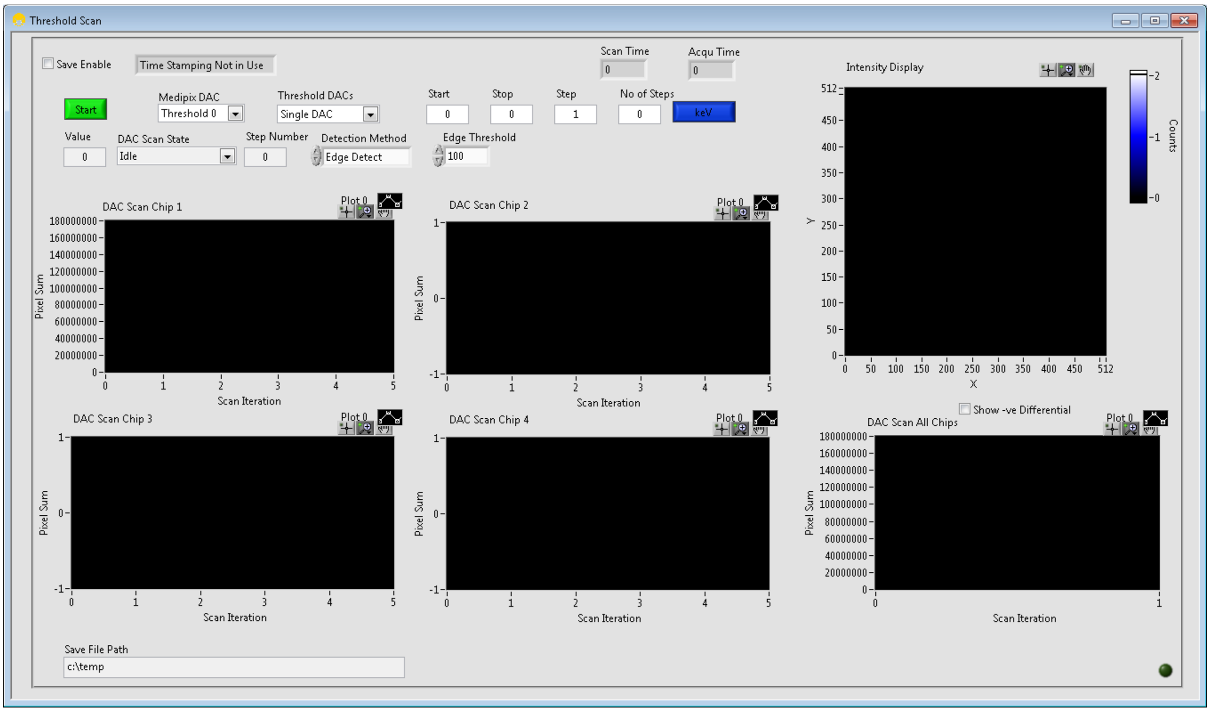


This window is accessed from the menu: **View –> 'Image Viewer'**. It is similar to the image display on the main control panel, but has the addition of the Line Profile and Statistics for simple online analysis. This analysis is not a comprehensive suite, but a tool to help ensure that the data being captured meets the requirements of the experiment.



Two cross hair cursors are used to define a rectangular region of interest, summarised in the stats tab, and either of the cursor 1 lines may be used for the Line Profile.

6.15. Threshold Scan Window



This window is accessed from the menu: **View –> 'Threshold Scan'**. The Threshold Scan window can be used to take a sequence of acquisitions with a series of different threshold settings. This will yield a spectrum summed over the whole image. It is also possible to use this display to locate the noise edge of the sensor and help choose a safe operating point. While threshold scans will normally be configured in terms of keV, the scan can also be configured in DAC steps and can operate on any of the Medipix DACs.

The scan will operate in the mode and with the acquisition settings as configure on the main panel.

Threshold Scan Displays

Intensity Display – this is similar to the main panel image and is here to avoid the need to switch back and forwards too often.

DAC Scan All Chips – this will show the integral of the counts, or the differential which will show a spectral type display. It is possible to export the results by right clicking on the graph.

DAC Scan Chip 1-4 – as **DAC Scan All Chips** but showing the result for each chip individually.

Show –ve Differential – changes the display from threshold sum to differential (spectral). Note that this should only be changes when a scan has completed.

Scan Setup

Medipix DAC – selects the threshold, or DAC to scan.

Threshold DACs – selects single or multiple thresholds, for colour mode.

Start – the starting value in keV, or DAC steps. The scan will always go from start to stop. It is possible to scan up, or down, by swapping the start and stop values.

Stop – the stop value in keV, or DAC steps.

Step – the step value in keV, or DAC steps. This is always positive, regardless of the direction of scanning.

No of Steps – sets, or indicates the number of acquisitions in the sequence.

keV / DAC – selects the start, stop and step units. Units are as shown.

Start – initiates the scan.

File Saving

Save Enable – turn on scan file saving. Note that the main panel file saving should be turned off. The scan initiates a sequence of single acquisitions which would lead to undesirable results, hence the separate control.

Save File Path – for scan image files.

Scan Progress Indicators

Value – step value.

DAC Scan State – shows the operating state.

Step Number – in the scan sequence.

Scan Time – elapsed.

Acqu Time – the acquisition time for each scan step.

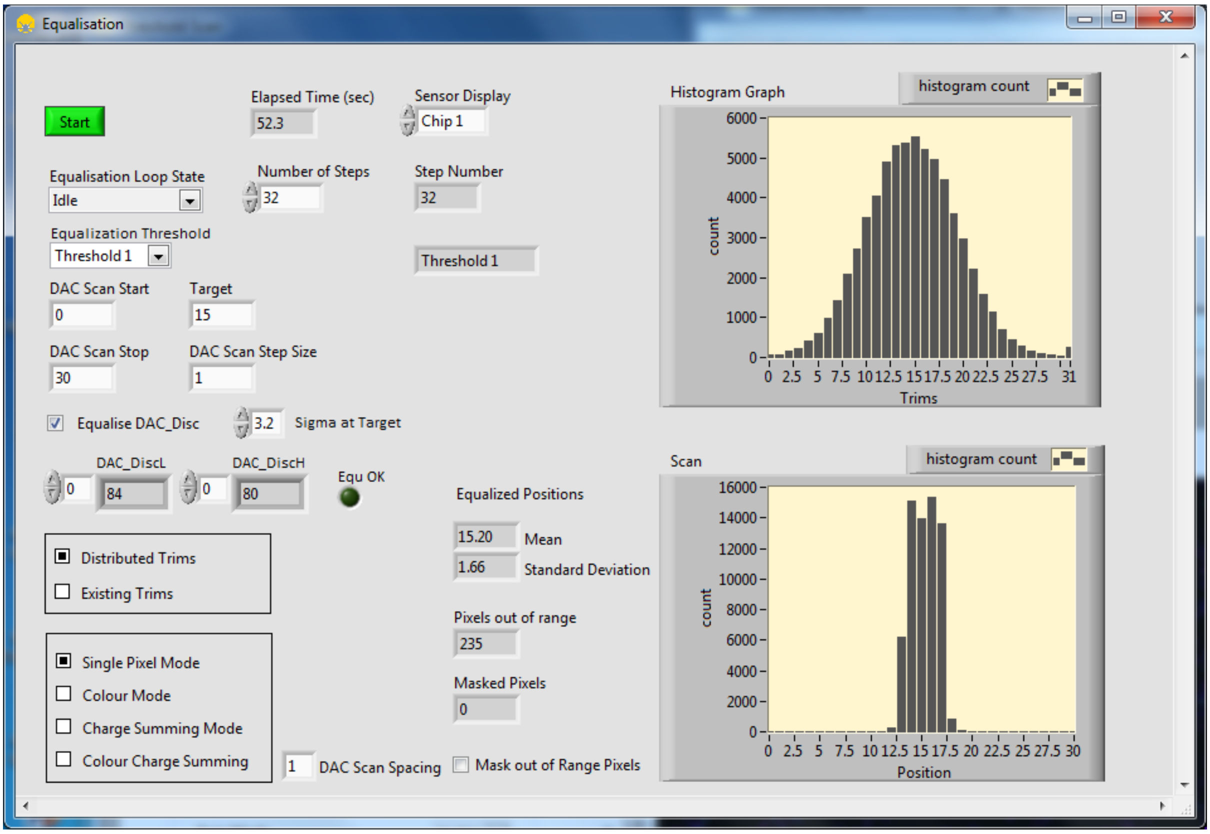
Equalisation Controls

These controls need to be set when using the Equalisation procedure, but don't have any effect when the Threshold Scan is use alone.

Detection Method – set as per equalisation requirements.

Edge Threshold – set as per equalisation requirements.

6.16. Equalisation Window



The Equalisation window is launched through menu: **View->Equalisation**. Equalising is a key step in setting up and calibrating the detector. The full procedure is described in section 10 Calibrating the Medipix3 Detector below.

Trim Display

Histogram Graph – of the pixel trim distribution. After equalisation the graph should show a normal distribution spread across the 32 trim steps. Before equalisation all trims are set to 31, so a small residual spike on this value will indicate a number of pixels could not be pulled into range.

Scan – shows the noise edge position in threshold DAC steps.

Sensor Display – allows the display to be switched between individual chips, or the whole sensor.

Equalised Positions: Mean – threshold noise edge position.

Equalised Positions: Standard Deviation – of threshold noise edge position.

Pixels Out of Range – the number that cannot be equalised with the current settings.

Masked Pixels – the number that have been disabled.

Equalisation Setup

Mode selection: **Single Pixel Mode**; **Colour Mode**; **Charge Summing Mode**; **Colour Charge Summing** – duplicates the main panel mode selection for convenience.

Distributed Trims; **Existing Trims** – selects whether the equalisation will start fresh, or carry on from existing settings.

Equalisation Threshold – to equalise.

DAC Scan Start/Stop/Step Size – set the scan range to be used for equalisation.

Target – threshold DAC value to position the noise edge.

Number of Steps – to iterate equalisation.

Equalise DAC_Disc – enable optimisation of the trim strength.

DAC_DiscL/H – indicates the optimised value of the trim strength DAC.

Sigma at Target – sets the offset from the DAC_Disc mean to the threshold target.

Mask out of Range Pixels – automatically after equalisation.

Start – initiates the equalisation.

Scan Progress Indicators

Equ OK – DAC_Disc has returned a sane value.

Step Number – of current Scan.

Elapsed Time (sec) – of the process.

7. Operational Modes and Configuration File Management

Whilst the Medipix3 can be operated in a large number of modes there are four groups of settings that require different pixel configurations to be loaded to allow the chip to operate correctly. In addition the four gain modes each have a different mapping between the energy of detected photons and the threshold DAC codes.

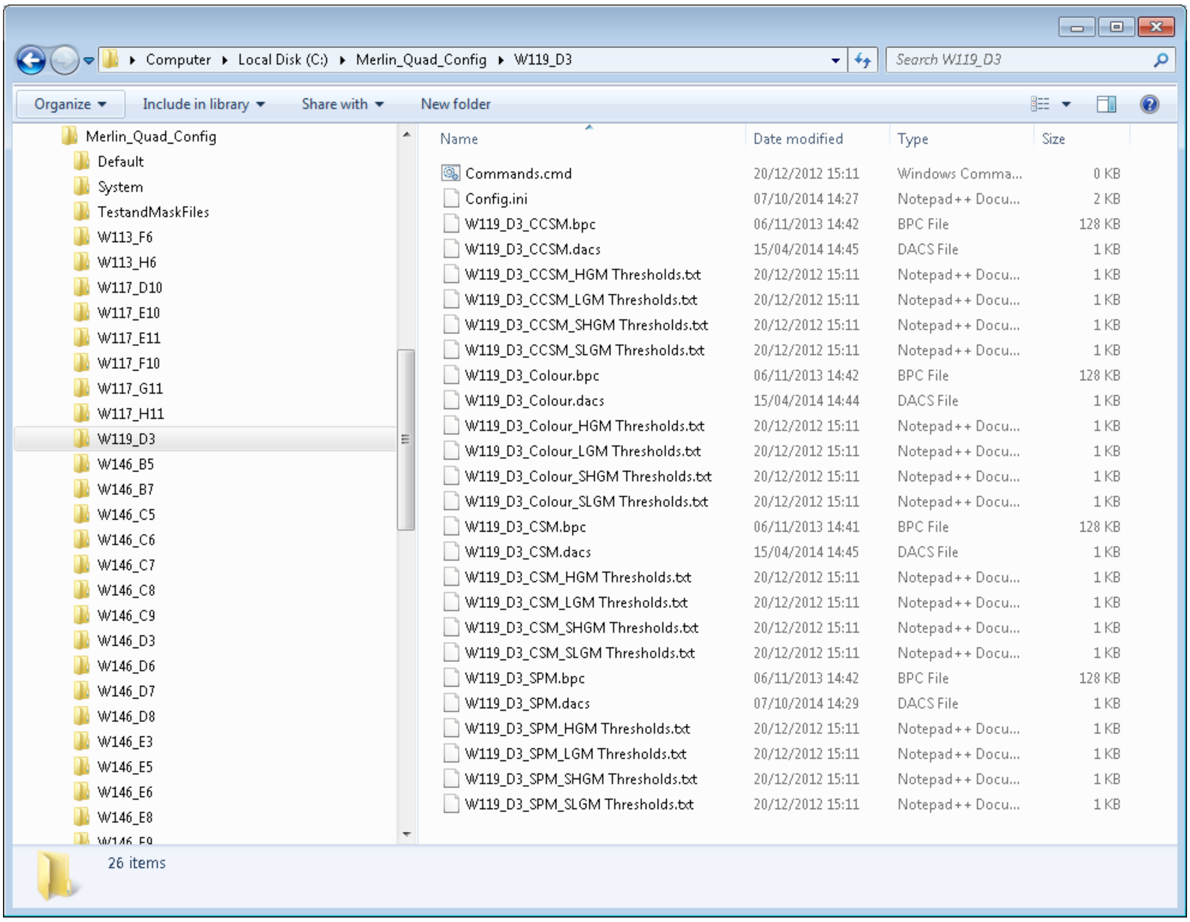
The Merlin system manages switching between these four modes and sixteen gain mappings by using a configuration file list set for each detector chip and defined in a config.ini file in the configuration folder of each chip. The correct folder is selected by reading the chip's identification and looking for a folder with the corresponding name. The chip in position 1 is used to store information on the layout of the sensor (quad, 4by1, pixel gaps etc)

7.1. Operation Modes

The four distinct operational modes that require new pixel configurations are Single Pixel Mode (SPM), Colour Mode, Charge Summing Mode (CSM) and Colour Charge Summing Mode (CCSM). The changes in the front end part of the pixel caused by switching between these modes require a different equalisation trim to be set for the pixels. As the system is moved into one of these modes, it loads the config files associated with it. The currently loaded config files can be seen on the config files tab.

7.2. Configuration Directory and File Layout

The complete sets of files for a chip are shown in the figure below.



The default location for the configuration is **C:\Merlin_Quad_Config**. Under this directory there will be one directory per sensor chip and other additional directories. The **Default** directory contains a template for configuring new sensors and should not be altered. **TestandMaskFiles**, as the name suggests, contains a series of test files that are not specific to any one chip. Inside each chip configuration directory is a **config.ini** file ad a series of files with a structured naming convention. The format is Serial-Number_Mode_Gain[Threshold].ext with extensions of bpc, dacs or txt. The bpc and dacs formats are not described here as they are automatically handled by the Merlin software. The dacs files contain the per chip DAC configuration and the bpc files contain the pixel equalisation trims and a mask should there be any noisy pixels. Note that while the dacs files do contain threshold values for historic reasons, these will most likely be overridden by thresholds set in keV.

7.3. Threshold Gain Mapping

The threshold gain mapping allows users to set the Merlin's threshold in keV without needing knowledge of the Medipix3's DAC codes system. In addition it can be used to overcome chip to chip variations in global gain and provide a uniform response from a multi-chip system like a quad. For each mode and each gain setting there is an individual text file that maps from keV to DAC codes. These files are manually generated during the calibration process. Usually a small number of points are entered and the system interpolates a desired keV setting to the nearest appropriate DAC code. The files essentially provide the base of these look up tables.

7.4. Config.ini File format

The config.ini files contain the locations of all the pixel configuration files and threshold mapping files as well as some information about the detector head. The file for the chip W146_G1 is reproduced below as an example.

```
[Config Paths]
DAC File = "W146_G1_SPM.dacs"
BPC File = "W146_G1_SPM.bpc"
SLGM Threshold Mapping File = "W146_G1_SPM_SLGM Thresholds.txt"
LGM Threshold Mapping File = "W146_G1_SPM_LGM Thresholds.txt"
HGM Threshold Mapping File = "W146_G1_SPM_HGM Thresholds.txt"
SHGM Threshold Mapping File = "W146_G1_SPM_SHGM Thresholds.txt"

[Colour Config Paths]
DAC File = "W146_G1_Colour.dacs"
BPC File = "W146_G1_Colour.bpc"
SLGM Threshold Mapping File = "W146_G1_Colour_SLGM Thresholds.txt"
LGM Threshold Mapping File = "W146_G1_Colour_LGM Thresholds.txt"
HGM Threshold Mapping File = "W146_G1_Colour_HGM Thresholds.txt"
SHGM Threshold Mapping File = "W146_G1_Colour_SHGM Thresholds.txt"

[CSM Config Paths]
DAC File = "W146_G1_CSM.dacs"
BPC File = "W146_G1_CSM.bpc"
SLGM Threshold Mapping File = "W146_G1_CSM_SLGM Thresholds.txt"
LGM Threshold Mapping File = "W146_G1_CSM_LGM Thresholds.txt"
HGM Threshold Mapping File = "W146_G1_CSM_HGM Thresholds.txt"
SHGM Threshold Mapping File = "W146_G1_CSM_SHGM Thresholds.txt"

[CCSM Config Paths]
DAC File = "W146_G1_CCSM.dacs"
BPC File = "W146_G1_CCSM.bpc"
SLGM Threshold Mapping File = "W146_G1_CCSM_SLGM Thresholds.txt"
LGM Threshold Mapping File = "W146_G1_CCSM_LGM Thresholds.txt"
HGM Threshold Mapping File = "W146_G1_CCSM_HGM Thresholds.txt"
SHGM Threshold Mapping File = "W146_G1_CCSM_SHGM Thresholds.txt"
```

```
[Commands Path]
Commands File = "Commands.cmd"

[Chip Config]
Revision = "3RX"
Pixel Size = "55"

[Multi Chip Config]
Layout = "2x2"
Gap X = "0"
Gap Y = "0"
```

The configuration file names are structured with the sensor chip serial number and operating mode. It is not recommended that the naming format is changed. The “Chip Config” and “Multi Chip Config” sections detail the physical configuration of the sensor assembly. However, it is possible to alter the layout value to indicate whether the image should be reconstructed with inter chip gaps, or not. Valid values are “1x1”, “Nx1”, “Nx1G”, “2x2”, or “2x2G”. The “G” suffix indicates that the Gap is inserted into the image. The Gap X and Y indicate how many extra 55µm pixel elements are present between the sensor chips. Note that this value must be in 55µm steps, even for 110µm sensors. Note that the “Multi Chip Config” section is only used for the first chip of a multi-chip sensor. It is ignored for the others. Details of the impact of these parameters can be found in section 8.2 Sensor Geometry and Gap Mode below.
The command file name can be safely changed.

7.5. Command File

Command files can be used to set up default parameters, or make a series of predetermined configuration changes. The command file specified in the sensor Config.ini file, section 7.4 above, is loaded and run during initialisation. The command file is the only way to set up certain default parameters. Subsequently any command file can be loaded and run from the **Config Files** tab.
A typical command file may look like this:

```
SET,ACQUISITIONTIME,1
SET,HVBIAS,80
SET, POLARITY,0
SET,THRESHOLD0,2.5
SET,THRESHOLD1,5
SET,THRESHOLD2,10
SET,THRESHOLD3,15
SET,THRESHOLD4,20
SET,THRESHOLD5,25
SET,THRESHOLD6,30
SET,THRESHOLD7,35
```

Definitions for all available commands can be found in Section 13 TCP/IP Remote Command Reference, below.

7.6. Loading Configuration Files

In normal operation all configuration files are loaded automatically. No user action is required. If circumstances do require loading files manually, then there are a number of options on the File menu. Some load options will present you with the select chip dialogue. Unless you have a very special reason to do otherwise, it is strongly suggested that you select **ALL**. The **ALL** option does assume that the configuration files for your system are laid out in the standard configuration as described above. You will select the file for the first chip in the system and the others will be automatically loaded from the same set.

Load Config – This option will load BPC, TST, MSK and TRM files. BPC files contain a combination of trim, mask and test configuration in a single binary file, so loading a BPC will wipe out any overlaid mask. While it will also overwrite the test image, the test file list is recorded and automatically reloaded when needed. The BPC file normally contains the sensor equalisation configuration data.
Loading a mask MSK file will merge the mask image from the file with the already active mask. If this is not what you want, you can clear the existing mask by using the context menu on the image display (main, or viewer).
Loading test TST, or trim TRM files will overwrite that portion of the configuration.

Load DAC Config – This option will load DACS files.

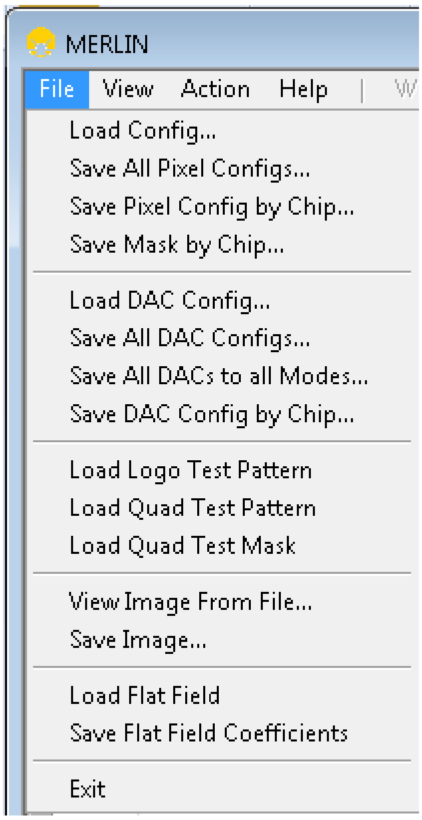
Load Logo Test Pattern – This option loads the same test image to all chips. As this does not take into account the chip orientation, the image will appear rotated on some chips for certain sensor geometries. This is normal.

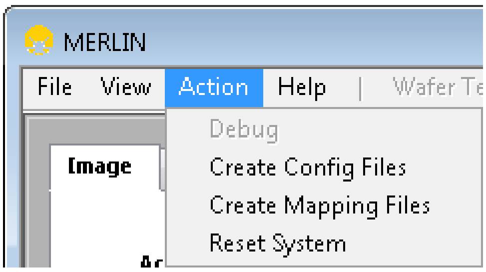
Load Quad Test Pattern – This option loads different test images to each chip. This test pattern can be used to identify the order of the individual chips within the sensor.

Load Quad Test Mask – This option loads different mask images to each chip. This is actually the same image as the Quad Test Pattern.

Load Flat Field – This option loads a flat field image, or pre-processed flat field coefficients.

If you change operating mode, the Merlin system will automatically reload the correct settings for the new mode and override any configuration that you have manually loaded.





7.7. Saving Configuration Files

IMPORTANT: **back up your configuration data for all chips**. Loading the wrong configuration may cause temporary inconvenience, but is easily undone. Overwriting your configuration will affect your calibration and may cause the system to become unusable.

IMPORTANT: **edit and save the configuration to the files for your sensor chips, not the default directory**. The default directory is the baseline configuration for creating configurations for new sensors. It will not be used once a sensor has been calibrated and has its own configuration files.

Calibrating a sensor is probably the main reason for needing to write to configuration files. Your system should be calibrated when you receive it, so be sure that you actually need to do this before proceeding. Some save options will present you with the select chip dialogue. Unless you have a very special reason to do otherwise, it is strongly suggested that you select **ALL**. The **ALL** option does assume that the configuration files for your system are laid out in the standard configuration as described above. You will select the file for the first chip in the system and the others will be automatically saved to the same set.

Create Config Files – This option is only normally used with a new sensor that has not been calibrated. It will create a complete set of configuration files based on the default set and your sensor chip IDs. Once you have a set of configuration files, you can proceed to setting up the DACs, equalising and calibrating your sensor in all required modes.

Save All Pixel Configs – This option will save the currently loaded set of BPC files that contain the sensor equalisation to the files listed on the **Config Files** tab.

Save Pixel Config by Chip – This option will save the current BPC configuration that contains the sensor equalisation to the files that you specify.

Save All DAC Configs – This option will save the currently loaded set of DACS files that contain the sensor operating points to the files listed on the **Config Files** tab.

Save All DACs to all Modes – This option will save the current DACs configuration and save copies to all other modes. This option may be used during the calibration procedure to set the same basic operating point, before the individual setup for each mode is done.

Save DAC Config by Chip – This option will save the current DACs configuration that contains the sensor operating points to the files that you specify.

Save Mask by Chip – This option will save the current Mask that to the files that you specify. The separate mask files are not automatically loaded.

Save Flat Field Coefficients – This option saves flat field coefficients pre-processed from a flat field image.

8. Taking Acquisitions

In all cases, it is advisable to set the counting, continuous, colour and charge summing modes before moving on to the other parameters. Selecting invalid combinations will result in Merlin overriding other parameters. The minimum gap is enforced, depending on the selected mode parameters.

8.1. Single Pixel Mode

To acquire a standard 12 bit image in the default single pixel mode:

On the main image tab:
Set **Counting Mode** to 2x12 bit
Uncheck **Continuous Mode**, **Colour Mode** and **Charge Summing Mode**
Set the number of frames you wish to acquire in **Number of Frames**
Set the length of time you wish each frame to be active for in **Frame Time**
Set the inactive time between frames in **Gap Time**
Set the **Number of Frames per Trigger** to '1'
Set **Counter** to Counter0
Select the **Gain Mode** you wish to use, usually LGM or HGM for x-rays

If you wish to save the files to disk:
Tick the **File Saving** checkbox.
Set your desired **Filename**
Set an appropriate **Local Save Root Directory.**

To set your threshold:
On the DACs Tab, in the Thresholds tab:
Set **Threshold0** to a level above the electronic noise of the system usually ~6keV or half the energy you are operating at whichever is higher.
Set **Threshold1** to a level higher than Theshold0. As this is not normally used in Single Pixel Mode this could be above the operating energy.
Click the **Apply Thresholds** button.

Return to the main Image Tab:
Click the green **Start Acquisition** button to launch the acquisition.

The Acquisition Time on the main panel will count up for the full length of the acquisition and the start acquisition button will go red.
If you wish to abort the acquisition once it is in progress click the red Stop Acquisition button.

When the Acquisition is finished the button will return to green. Note that if a large number of frames have been captured, the readout may lag and will need to catch up before the system will allow you to run another acquisition.

8.2. 24 bit readout

To run an acquisition with 24 bit depth, follow the same procedure as the Single Pixel Mode above, but set **Counting Mode** to 1x24 bit.

8.3. Reading Two Counters

To read information from both counters, follow the same procedure as the Single Pixel Mode acquisition above but set **Counter** to 'Counter0 & Counter1'.

You should also make sure that Threshold1 is set higher than Theshold0.

This will display and save two images for every frame you have set in **Number of Frames** and present them as an alternating sequence, conter1, counter0, counter1, counter0 etc. As the two counter images are taken from the same analogue information they can be directly subtracted to produce an energy window image.

8.4. Continuous Read Write Mode

To acquire a series of frames in the dead time free continuous read write mode set the system as for a Single Pixel Mode Acquisition with the following changes:
Set **Number of frames per Trigger** to be the same as **Number of Frames.**
Tick the **Continuous Mode** checkbox**.**

In this mode **Gap Time** is ignored and the **Frame Time** must be set to more than the 12 bit readout time of 820µs. This is to allow the alternating counters time to read out whilst the system acquires the other frame. As the memory of the system is limited if the system is set to run over 100Hz, only 1200 frames (or 4800 frames) should be acquired in one acquisition for a Quad detector (or Single detector).

8.5. Charge Summing Mode

Two thresholds are used: threshold 0 is used to determine that a pixel has the necessary minimum individual charge to be the centre of reconstruction and threshold 1 gates the final reconstructed charge. To run an acquisition with Charge Summing, follow the same procedure as the Single Pixel Mode above, but tick the **Charge Summing Mode** checkbox.

Set **Threshold0** to a level above the electronic noise of the system usually ~6keV or a quarter the energy you are operating at whichever is higher.
Set **Threshold1** to a level higher than Theshold0, but lower than the operating energy.

Note that this mode **cannot** be used in conjunction with Reading Two Counters to obtain an energy window. Charge summing will return the image in counter 1. If counter 0 is also selected, this will return the arbitrated single pixel counts. This is not normally of interest.

8.6. Colour Mode (with 8 colour layers)

To acquire images in colour mode set the system as for a standard single pixel mode image with the following alterations:

Tick the Colour Mode Checkbox.
Set **Counter** to 'Counter0 & Counter1'
Ensure all eight thresholds are set to the required values on the thresholds tab.

This will result in Eight images being acquired for every frame set in **Number of Frames.**
By default these will be displayed in the **Colour Layer** index of the image and will be saved in concatenated files.

8.7. Hardware Triggering

To use hardware triggering to define the active period of the detector you have a number of options. These can be used to define either single frames or multiple frames in an acquisition, or to launch a burst of frames after the receipt of a hardware trigger. The trigger input can be either TTL or LVDS. The jitter on the trigger response is in the region of 10ns as the shutter is controlled by a 100MHz clock.

As the trigger settings are partly dependant on the acquisition mode, it is advisable to set the mode first, and then configure the triggers. If the mode is changed after the trigger set-up, then recheck the trigger settings. In all cases **Start Acquisition** arms the system and must be received before any trigger is recognised. **Stop Acquisition** may be used to end any sequence.

The following sections describe some of the more common configurations. Please use these as a guide if your exact requirement is not mentioned.

8.8. Trigger Start of Frame (Shutter Open)

To run the system in a state where an external pulse triggers start of a frame set the system as though it were taking the desired type of acquisition (Single Pixel Mode, Two Threshold, 24 bit, Colour Mode etc.)

On the Triggers Tab set **Trigger Start** to rising or falling edge as required and **Trigger Stop** to Internal.

Set the **Number of Frames** as required for the overall acquisition and the **Frame Time** as needed. **Frames per Trigger** should be set to 1.

Click **Start Acquisition** to arm the system. On receipt of the trigger the system will start the frame and when the frame time has elapsed will automatically end the frame and be ready to receive the next trigger. The frame time programmed is accurate to 10ns.

8.9. Trigger End of Frame (Shutter Close)

To set the system to run with the shutter held open until a trigger is received set the system in the same manner as described for the starting trigger, but set the **Trigger Start** to internal and the **Trigger Stop** to rising or falling.

The system will start acquiring immediately when the **Start Acquisition** is pressed, but will not end the frame until a trigger is received. When the frame is ended if **Number of Frames** has been set to more than one, another frame will be started after the **Gap Time** has elapsed. **Frames per Trigger** should be set to 1.

8.10. Trigger Start and End (Gate, Shutter Follows Trigger)

In this mode the shutter is completely controlled by the external trigger with both the open and close set. This can either be in the form of a 'gate' where a rising edge starts the frame and a falling edge end it, or as a pair of pulses where one starts and the next stops. These are enabled by having **Trigger Start** and **Trigger Stop** both set to either rising or falling as required.

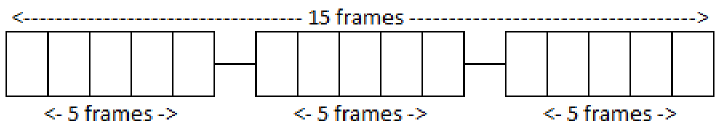
As the system can react very fast a de-bounce mechanism is used to avoid unwanted short multiple frames. The **Frame Time** should be set to a value smaller than your minimum desired acquisition time as the shutter close will be inhibited for this length of time. **Frames per Trigger** should be set to 1.

8.11. Trigger a Burst of Acquisitions

To launch a burst of acquisitions after a received trigger set the system as if you were triggering the start of a frame as described above, but set the **Gap Time** as desired for the burst and **Frames Per Trigger** for the number of frames you want for the burst.

This can also be used in Continuous read write mode. In this case the **Gap Time** is ignored and the **Continuous Mode** checkbox should be ticked.

As before the system will arm on clicking **Start Acquisition** and on receipt of the external trigger signal will record the number of frames set to **Frames Per Trigger**. The system will stop when the total **Number of Frames** is reached even if this is not a multiple of **Frames Per Trigger**.

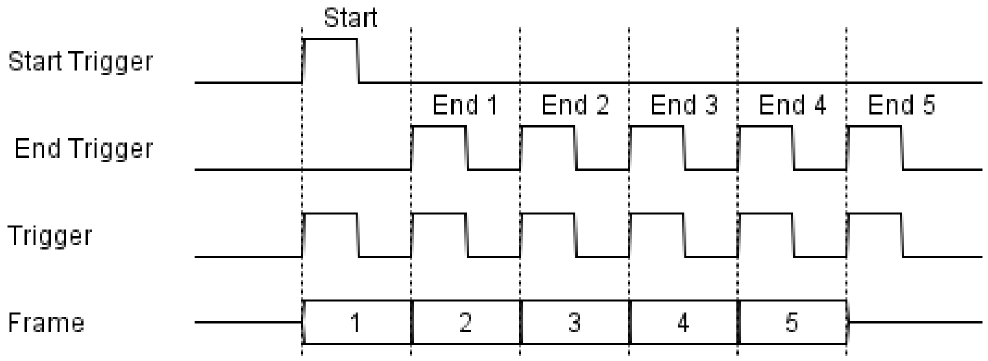


In this example Merlin is configured with **Continuous Mode**, **Number of Frames** = 15 and **Frames Per Trigger** = 5. Each trigger will start a burst of five frames, and then the acquisition will pause and wait for the next trigger. After three triggers, the final burst will complete and the acquisition will end.

8.12. Trigger a Continuous Burst of Acquisitions

If you want to trigger one continuous burst with one Start Trigger pulse, then see section 8.11 Trigger a Burst of Acquisitions above. This section will describe the case of controlling the individual frame times within a continuous burst using external triggers. In this mode, one trigger signal can be used to start a burst; switch frames and end the acquisition.

To understand the configuration, it is necessary to understand how the **Frames per Trigger** parameter works for continuous mode. This parameter sets the number of frames that will occur after a start trigger signal. It also sets the number of continuous frames that will be captured in a single burst. There is therefore just one start trigger per burst. Internally, the Merlin system handles the switch from one continuous frame to the next as a stop acquisition event, so by adding a **Stop Trigger** we can control this event. The total number of trigger pulses will be one more than the number of frames: one start plus one stop per frame. There is no way to distinguish the final trigger pulse, in hardware. This is configured purely by the **Number of Frames per Trigger** parameter.



In the diagram "Start Trigger" and "Stop Trigger" are pseudo signals to show how Merlin interprets the sequence of Trigger pulses.

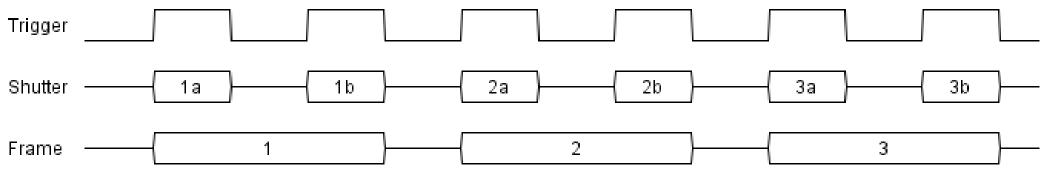
Connect external trigger;
Turn **Continuous Mode** on;
Set **Number of Frames**;
Set **Number of Frames per Trigger** (this is the number of continuous frames in one burst);
Set **Start Trigger** (e.g. Rising Edge);
Set **Stop Trigger** to the same as **Start Trigger**;
Double check all the settings;
Start Acquisition.

As the system can react very fast a de-bounce mechanism is used to avoid unwanted short multiple frames. The **Frame Time** should be set to a value smaller than your minimum desired acquisition time as the shutter close will be inhibited for this length of time. In continuous mode, the minimum actual frame time is also limited by the readout speed as one counter must be fully read out before switching to the other counter. If the trigger pulses come too fasts, then the system will ignore them. Also, as the triggers are edge sensitive, the system will not see the early edge and will wait for the trigger to cycle round and produce the next edge.

Variations are possible, using two trigger signals, or using the opposite edge for start and stop. Triggering multiple bursts is possible, but this would be best achieved using different trigger signals for start and stop, or by using the trigger output as feedback.

8.13. Multi Triggering Frame Capture

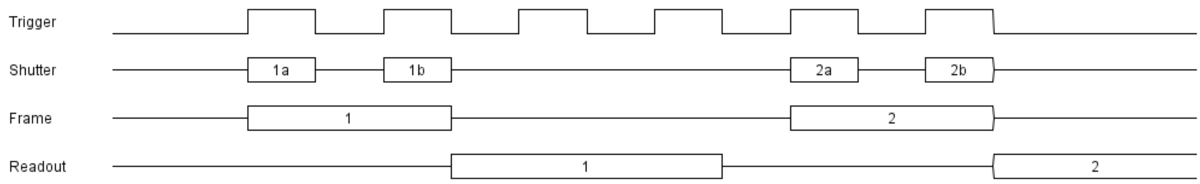
When it is necessary to build up an acquisition over time, but it is undesirable to leave the shutter open the whole time, an image can be built up using multi-gate triggering. Using this mode, the shutter can be opened and closed at a much higher rate than possible with conventional single image readout. The faster rate is possible because the detector does not need to wait for the readout time before the next shutter opening. One application is repeatedly isolating x-rays from a single electron bunch as it makes multiple passes through the storage ring. Multi-gate triggering is available from Version 0.65 onwards.



This example shows Multi-gate triggering for **Number of Frames** = 3 and **Number of Frames per Trigger** = 2.

Connect external trigger;
Turn **Continuous Mode** *off*;
Set **Number of Frames**;
Set **Number of Frames per Trigger** (the number of individually triggered gates, that will make up one frame);
Set **Start Trigger** (e.g. Multi-Gate Rising Edge);
Set **Stop Trigger** as required (probably Internal or Falling Edge);
Double check all the settings;
Start Acquisition.

As the system can react very fast a de-bounce mechanism is used to avoid closing the shutter too early. Assuming that **Start Trigger** is Multi-Gate Rising Edge and **Stop Trigger** is Falling Edge, then the **Frame Time** should be set to a value *smaller* than your minimum desired acquisition time as the shutter close will be inhibited for this length of time. If you want the shutter open for a fixed duration starting with each trigger pulse the **Stop Trigger** can be set to Internal and the **Frame Time** set as normal. The total acquisition time will be the sum of the individual gates. Using the internal frame timer will give more consistent frame opening times, especially for very short gate periods.



Once each frame has been acquired (built up from the required number of shutter gates), Merlin will read out the frame. During this time, further triggers will be ignored. As there are multiple shutter openings per acquired frame, the shutter timestamp and acquisition period in the image header will be inaccurate.

8.14. Trigger Output

It is possible to use the trigger output to drive another piece of equipment, say a motor stage. It is also possible to use the trigger input and the trigger output to provide feedback. This way, when a motor stage has finished moving, it can trigger a capture and the trigger output can indicate when the capture has finished and the motor can move again. In this example, set the Trigger Out to follow the acquisition burst and the falling edge indicates the end of acquisition. The trigger output can be inverted if necessary.

If Trigger Out is selected to show “Busy”, the signal will be active until the hardware readout is complete for all acquisitions, i.e. after the **Number of Frames** is reached.

8.15. Software Trigger Control

For further flexibility when Merlin is under TCP control, the trigger outputs can be set by TCP commands. Select the trigger out to be a soft trigger out and use the SET SoftTriggerOutTTL/LVDS command to control the state. It is also possible to interrogate the Trigger input state with GET TriggerInTTL/LVDS. These controls and indicators are also available on the Trigger tab of the GUI. They can be used for diagnostic purposes while setting up Merlin to work with other equipment.

8.16. Acquisition Performance Limits

The basic performance limits are set out in the Technical Specifications, section 2 above. These assume a standard installation on Windows 7 64bit with no other applications running on the readout PC. It is recommended that other applications are not run on the readout PC, however, the actual conditions of use are down to the user and the user must determine if any additional loading still leaves the Merlin system with adequate performance.

It is possible that the acquisition parameters are set beyond the limits set out, with, or without additional software running on the readout PC. In this case, the actual performance seen may not match the parameters set. This section discusses the **Counting Mode**, **Frame (Exposure) Time**, **Gap (Readout) Time** and **Frame Rate**, both burst and continuous as they apply to the performance limits.

The **Gap Time** is limited by the speed that data can be read from the Medipix3. This **Readout Time** is a hardware limit and not affected by software loading. The minimum value is dependent on the counting mode and varies in proportion to the counter depth. No matter how low the Gap Time is set, the hardware will always complete the readout, with the result that the Gap Time may be longer than requested.

The **Frame (Exposure) Time** is mainly limited by the need to get enough light in, or the need to avoid saturation in the counter. In **Continuous Mode**, this parameter defines the **Burst Frame Rate**. For all modes, it defines the **Frame Rate** in conjunction with the **Gap Time**. Also in **Continuous Mode**, the minimum **Frame (Exposure) Time** is limited by the **Readout Time**, as this happens within the **Frame (Exposure) Time**.

Altering the **Counting Mode** can impact the **Burst Frame Rate**, via the **Gap Time** and **Frame (Exposure) Time**, but will not give any real improvement to the **Continuous Frame Rate**[5].

The **Continuous Frame Rate** is limited by the system's ability to process data and store it to disk, or transmit the data via TCP. It does not matter how the frame rate is made up of the various factors like **Frame (Exposure) Time**, or **Readout Time**, or if the capture rate is steady, or in bursts. Data is buffered into system memory and written out to disk, or TCP as fast as possible. The buffer will smooth out variations in the frame rate to some extent. If data starts arriving too fast, the system memory buffer will fill up and the hardware readout buffer will be unable to empty. The hardware memory can store about 1200 12bit quad chip frames and when it reaches its pre-set limit, it will pause acquisition. In all non-continuous modes, the pause will extend the **Gap Time** before taking the next frame. In **Continuous Mode**, the pause does not have a gap to extend, so one frame is extended. In both cases, the pause will extend until a lower, nearly empty, threshold is reached and the acquisition will resume. This process will be repeated if data continues to arrive faster than it can

[5] The "Continuous Frame Rate" is the sustainable rate at which Merlin can process data and is not related to "Continuous Mode".

be written out. An indication that the data output has been throttled back is given on the Status Window, see section 6.13 above.

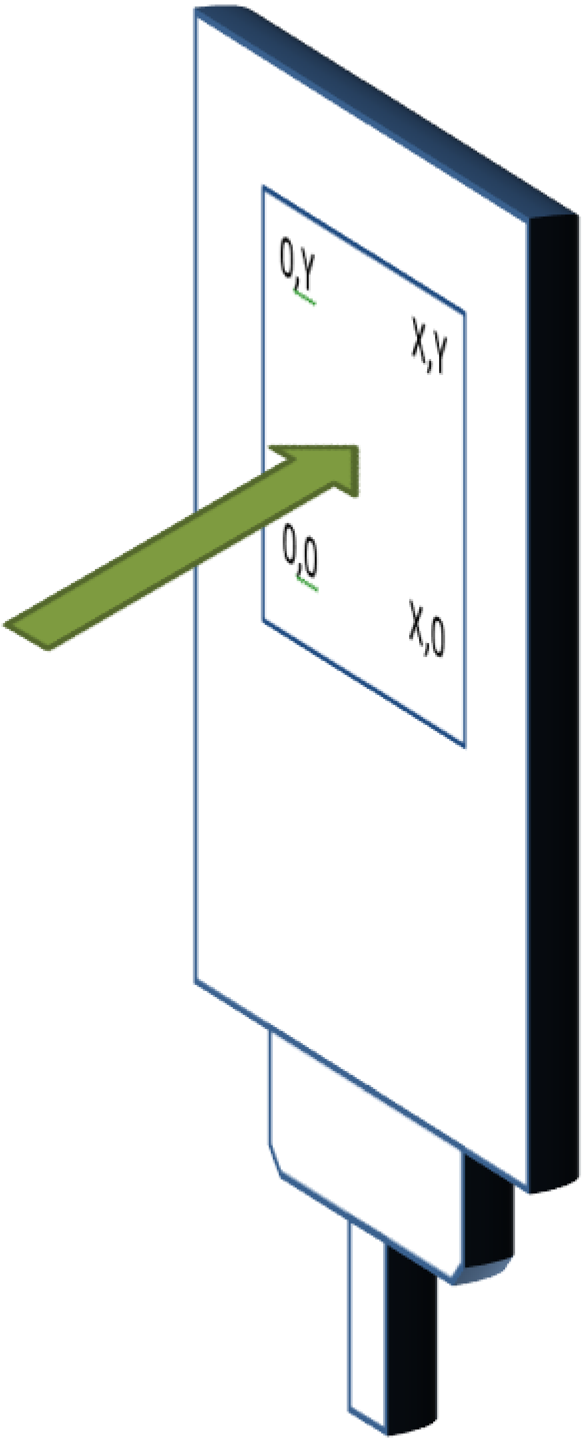
All data will continue to be saved by the Merlin System. The image header records the hardware timestamp of the start of the frame and the exposure time (elapsed acquisition time). This information can be used to identify the point at which any **Gap Time** or **Frame (Exposure) Time** has been extended by a hardware readout pause.

The performance limits are still the same if the acquisition is controlled by external triggers. Exceeding the performance limits may result in some triggers being missed. If Merlin is used in closed loop control with a motor stage using both Trigger In and Trigger Out, an optimal acquisition sequence could be achieved, by allowing the motor to move on one Trigger Out set to **One Per Acq Burst** and the other Trigger out set to **Busy** to hold off the start of the next frame.

The length of an acquisition burst is limited by a combination of factors. The Burst Frame Rate is important, because even as data is added, data is being transferred to disk, or via TCP. If the Burst Frame Rate is only slightly higher than the **Continuous Frame Rate** then quite long bursts may be possible. Faster bursts will fill the memory faster. Burst lengths of 4000 to 5000 frames are possible.

In conclusion, the **Burst Frame Rate** is limited either by the **Frame (Exposure) Time** and the need to get enough light in, or the **Readout Time**, or both. It is a hard limit. The **Continuous Frame Rate** is limited by the system's ability to store data to disk, or transmit the data via TCP. The Burst Length is limited by the combination of hardware memory and host PC system buffer memory.

8.1. Sensor Image Orientation

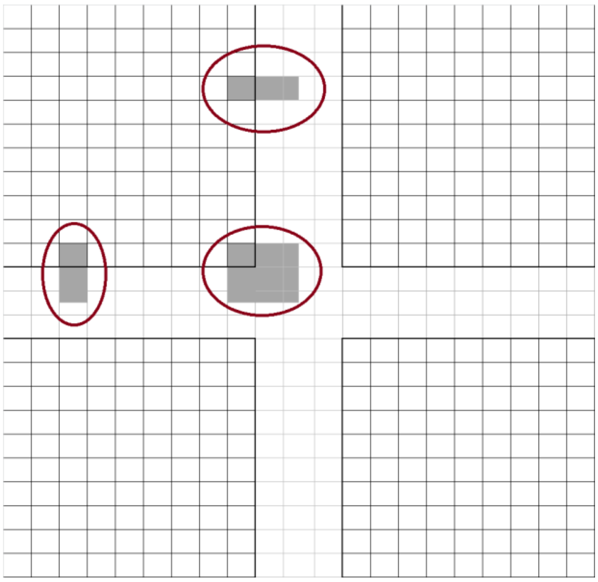


When the detector head is mounted with the cable down, incident photons strike the sensor with pixel 0,0 at the bottom left hand corner as shown. This is the orientation shown on the Merlin control panel.

8.2. Sensor Geometry and Gap Mode

This applies to all multi chip sensors. These are constructed by bonding several readout ASICs to one large sensor. When this is done, it is necessary to allow a clearance around each readout ASIC. This in turn leads to a discontinuity in the pixel pattern with a gap between ASICs. In practice the pixels at the edge of the ASIC, but in the centre of the sensor extend half way across the gap.
Merlin allows alternate methods of dealing with the gaps, in order to accommodate different measurement requirements.

Quad Sensor, Centre Region with Three Pixel Gap

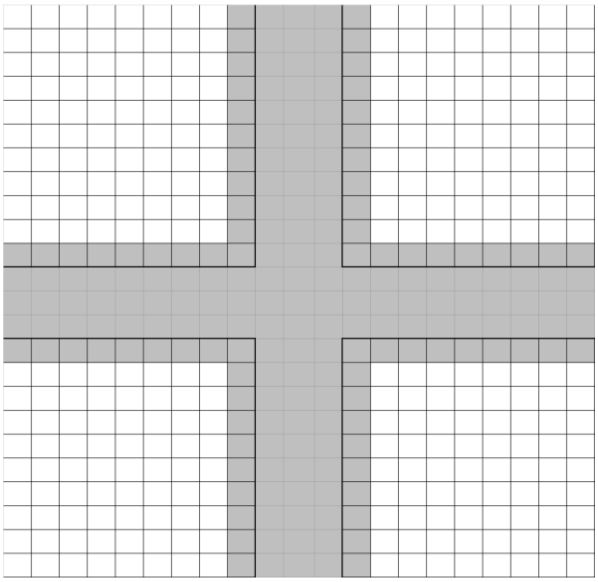


The highlighted regions show where sample pixels extend half way across the gap. It is clear from this diagram how the charge collection area for these pixels is larger than the others. This will lead to increased counts and earlier saturation for these pixels.

Raw Data, or Geometrically Accurate?

Merlin can process the data to provide a geometrically accurate image and correct for the count rate distortion of the larger pixels adjacent to the gap. Part of this reconstruction is controlled in the Sensor specific configuration; see section **[Multi Chip Config]** within 7.4 Config.ini File format above for more details. These parameters allow the choice of a geometrically accurate image, or an unmodified raw data set.

Fill Region and Modes



The diagram shows the area that may be affected by gap filling operations. There are four modes:

None – No action, the gap area will have zero counts. The gap edge pixels will not be adjusted.

Zero Fill – The whole of the shaded region, including the gap edge pixels will be zeroed. This will remove the pixels that over counted.

Distribute – Takes the counts in the gap edge pixels and divides them amongst the pixels in the gap.

Interpolate – Takes the counts in the gap edge pixels, performs a count correction, then uses the corrected values either side of the gap to interpolate values for the gap pixels. This may give a better looking image, but may not be as accurate in terms of the overall sensor photon count as **Distribute**.

9. DAC Summary

This table summarises the DAC names numbers and effects.

DAC No.	RX Name	RX Effect	Suggested Default Values
1-8	Threshold 0-7	Global Thresholds	Set via Threshold Tab.
9	Preamp	Preamplifier Gain	175
10	Ikrum	Pulse decay rate	10 (100 for higher counting speed)
11	Shaper	Amplifier Time constant	200 (SPM), 150 (CSM)
12	Disc	Discriminator Bias	128
13	Disc_LS	Discriminator Bias	100
14	Shaper Test	Test DAC	0
15	DAC Disc L	Low trim bits strength[6]	See section 10.2 below.
16	Delay	Comparator timing adjust	30
17	TP Buffer In	Bias for Test Pulses	128
18	TP Buffer Out	Bias for Test Pulses	4
19	RPZ	Pole zero compensation	255
20	GND	Amplifier virtual earth	0.65V, see section 9.1 below.
21	TP Ref	Test Pulse reference point	128
22	FBK	Amp operating point	0.9V, see section 9.1 below.
23	Cas	Amp operating point	0.85V, see section 9.1 below.
24	TP Ref A	Test Pulse Level group A	As required, or 0 when not used.
25	TP Ref B	Test Pulse Level group B	As required, or 0 when not used.
26	Band Gap*	Read Silicon Band Gap	
27	BG Temp*	Read Temperature Sensor	
28	DAC Bias*	Read DAC Bias	
29	Cas Bias*	Read Cascode Bias	
30	DAC Test	Read Test DAC	0
31	DAC Disc H	High trim bits strength[6]	See section 10.2 below.
	*Read only DACs		

9.1. Setting Voltage Levels for GND, FBK and CAS

The optimal values of GND, FBK and CAS DACS need to be set to specific values in mV due to variations between chips caused by the fabrication process. The **Sense DAC** functions on the advanced panel should be used in conjunction with the DAC panel to set these accordingly. The target values are GND = 650mV, FBK = 900mV and CAS = 850mV. These values can be set up using the tracking feature. Individual DACs can be set on the Advanced tab, or all three DACs can be set in one go using the menu: **Action->Setup DACs**.

IMPORTANT: Merlin stores separate DAC setting files for each mode, see section 7.1 Operation Modes above. Switching modes will load the DAC settings for the new mode, so any DAC changes will not carry across. It is possible to copy DAC settings from one mode to another using the menu: **File-> Save DAC Config by Chip**. It is best to select ALL as this will allow you to overwrite all the DAC settings for a particular mode. This operation will only normally be done prior to a full sensor calibration. Fine tuning the DACs for each mode can follow this. Also note that some of the DACs will directly affect the chip equalisation and threshold calibration.

[6] From Version 0.65 onwards, DAC Disc L and DAC Disc H can be automatically set during Equalisation.

10. Calibrating the Medipix3 Detector

There are three elements of calibrating the Merlin detector. Firstly the DACS must be set up as in section 9 DAC Summary above. Secondly equalisation, which sets individual pixel adjustment bits to ensure that the whole matrix within a chip responds uniformly to a global threshold setting. There may be some iteration between some DAC setting and equalisation, if the Trim distribution does not fit in the range available, or is too compressed. Finally, the threshold energy calibration, which is dependent on the equalisation for its starting point and sets the relationship between the internal threshold DAC and a real energy value. This second calibration also allows multiple chip systems like quads to respond in the same way to a given energy signal.
Equalisation and calibration require that the sensor is biased with the correct voltage and the sensor is configured for hole or electron collection as appropriate for the sensor polarity (see section 6.3 Advanced Tab, above).
IMPORTANT: *Before beginning either of these processes, it is strongly recommended that you back up your existing configuration. It is also suggested that you take backups after the final calibration to ensure that you can return to known good settings.*

10.1. Outline Procedure

It should only be necessary to run through the whole procedure the first time that a sensor is calibrated. It is important that certain steps are carried out in sequence, so while it is possible to skip some steps if they have been done previously, following steps should be completed due to the dependencies of the settings.

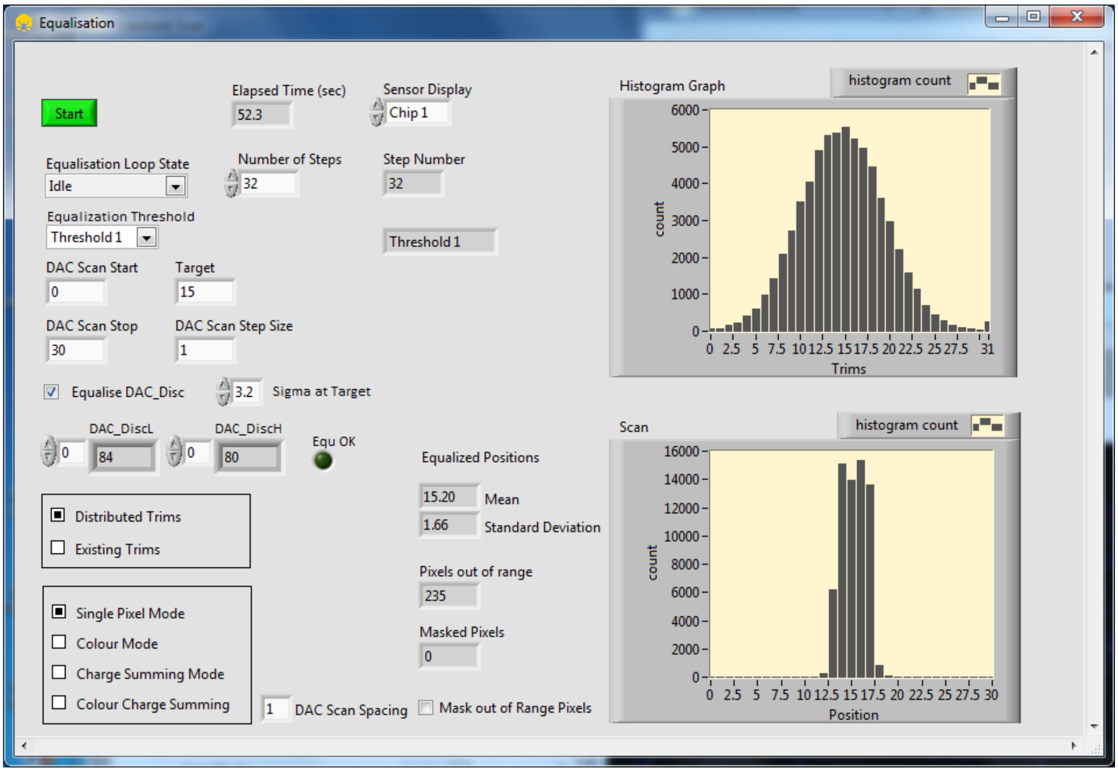
- Create Config Files
- Per Mode Settings:
  - Track GND, FBK and CAS DACS
  - Set other DACS, if not using defaults
  - Save All DAC Configs
  - Equalisation for Threshold 0 and 1, or odd and even in required mode
    - Save All DAC Configs
    - Save All Pixel Configs
    - DAC Scan to locate noise edge
    - Check for noisy pixels above noise edge and mask as necessary
  - Repeat Equalisation steps if necessary (noisy pixels can cause non-optimal equalisation)
  - Threshold Calibration
  - Save Threshold Mapping

10.2. Equalisation

Equalisation is the process of setting the pixel threshold adjustment bits so that all the pixels respond in a uniform way to the detectors global threshold setting. Each threshold in each pixel has 5 bits of adjustment that need to be set. This is required to correct for variations in the fabrication process that affect the pixel amplifiers and cause a spread in thresholds between pixels. Below the

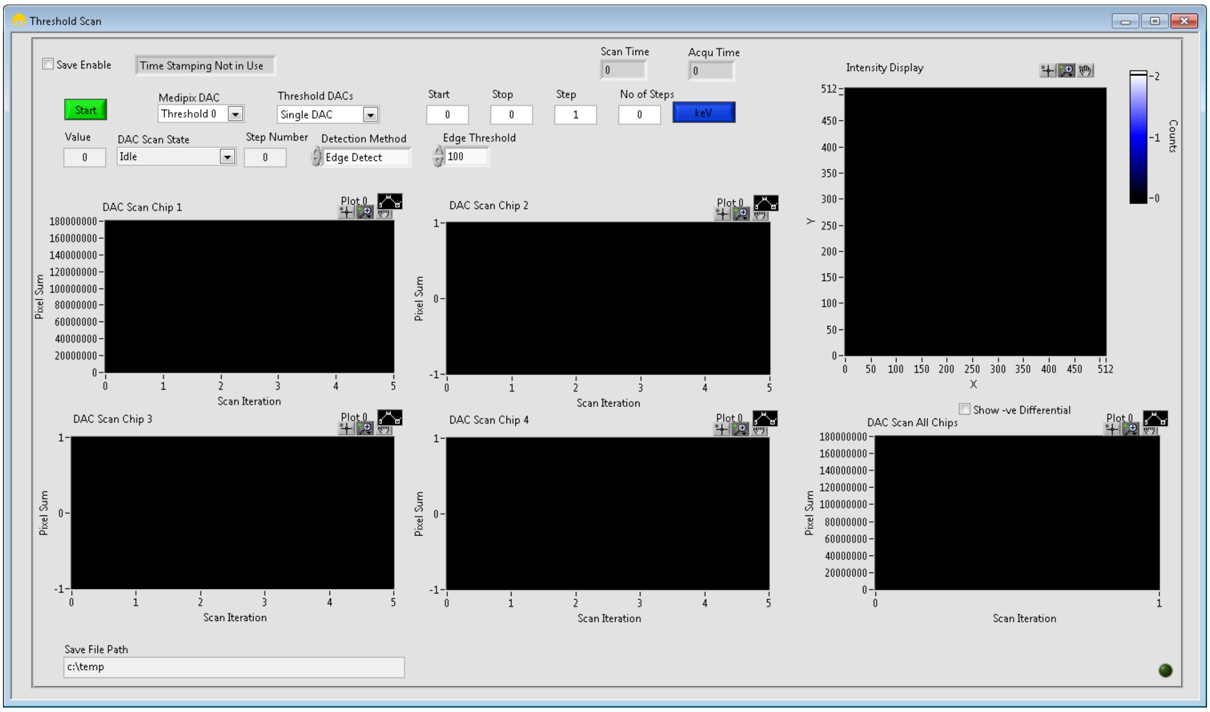
Equalisation and DAC scan panels are shown followed by instructions for running the equalisation in various ways.

Equalisation Panel



The Equalisation panel is launched through menu: **View->Equalisation**.

DAC & Threshold Scan Panel



The DAC Scan Panel is launched through menu: **View->Threshold scan** (versions from 0.64 onwards), or through menu: **View->DAC scan** (versions up to 0.63). From version 0.64 onwards, the DAC and Threshold scan control panels have been combined. The main difference was that the DAC Scan operated in DAC steps and the Threshold Scan operated in keV using the threshold calibration. The new panel can operate in either mode. You can check and change the units using the **keV DACS** button to the right of the **Start**, **Stop**, **Step** and **Number of Steps** controls. The equalisation process will set the correct mode automatically.

10.3. Single Pixel Mode Noise Edge Equalisation

- On the Main Image Tab set the system to do a 12bit SPM acquisition with
  - **Number of Frames** set to 1
  - **Frame Time** set to 1ms
  - **Counter** set to Counter0
  - **Gain Mode** set to SLGM

- On the DAC scan panel
  - Set **Detection Method** to Edge
  - Set **Edge Threshold** to 100

- On the Equalisation panel
  - **Medipix DAC** = Threshold0
  - **DAC Scan Start** = 0
  - **DAC Scan Stop** = 30
  - **DAC Scan Step** = 1
  - Use **Distributed Trims** Set to ticked
  - **Target** set to be 10
  - Select **Single Pixel Mode**
  - Select **Equalise_DAC_Disc**
  - **Sigma at Target** = 3.2

  - Then click Start

The system will run through and take about a minute to complete the equalisation in this mode, or longer with **Eqlise_DAC_Disc** set. At the end it will set the bits to the best positions. Check the **Standard Deviation**; a value in the range of 1-2 indicates a good spread. After this manually running a DAC scan will display the full width of the noise peak.

To save the calibration use menu->**File**->**Save All Pixel Configs** and menu->**File**->**Save All DAC Configs**.
IMPORTANT: changing modes will reload the saved configuration. If you do this before saving, you will lose your new equalisation.

10.4. Equalising Threshold1 on Noise

To Equalise the upper threshold follow the procedure for the lower threshold described above but on the Equalisation panel set Medipix DAC = Threshold1. All the other settings and operations are the same.

10.5. Colour Mode Equalisation on Noise

To equalise the system in colour mode, initially set the system into **Colour Mode** on the Main Image Tab, or on the Equalisation window[7].

Then follow the procedure above for SPM equalisation with both Threshold0 and Threshold1 as before. Threshold0 will equalise all the Even thresholds at once and Theshold1 will equalise the Odd thresholds.

10.6. Charge Summing Monochrome and Colour Modes on Noise

The method is essentially the same as the above methods with the appropriate mode selection made. Note that in charge summing, it is Threshold1, or the Odd thresholds that are most important. You should only calibrate the charge summing mode that matches your sensor configuration: 55µm, or 110µm.

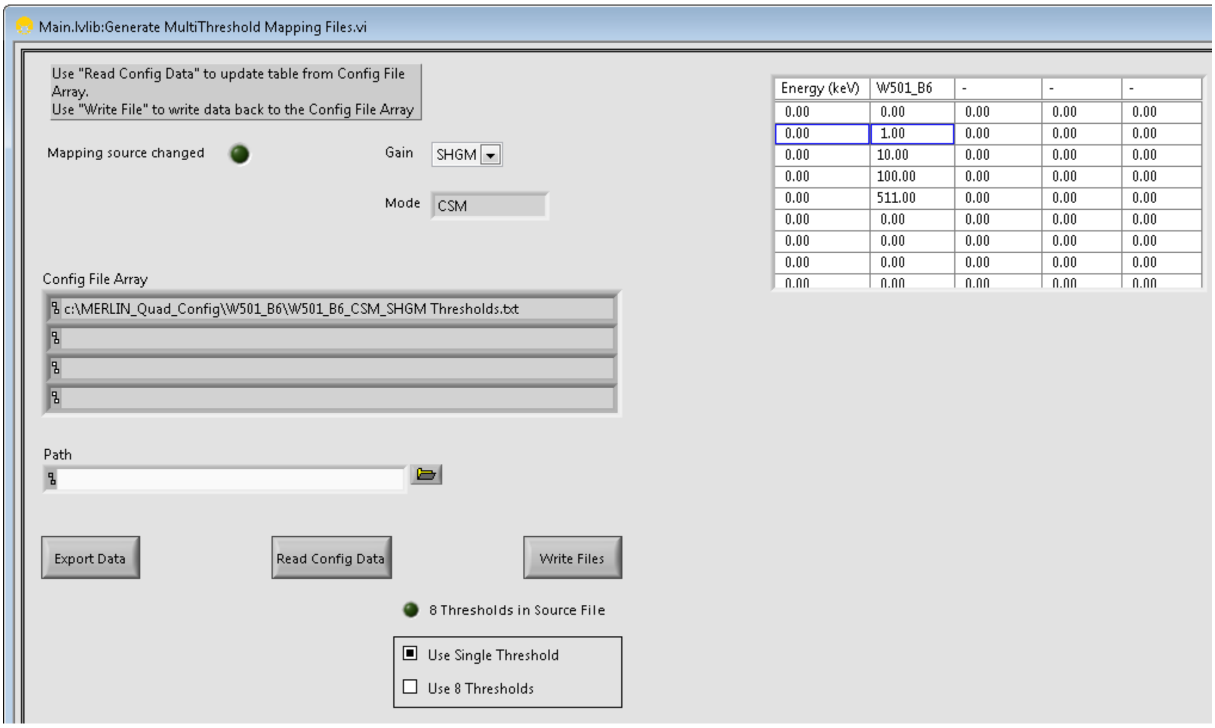
10.7. Threshold Calibration

To confidently set the value of the threshold in keV a look-up table is used as described in the Section 7.3 above. To determine the DAC values which correspond to particularly energies it is necessary to perform a calibration. This can be done with any reasonably mono-energetic source of x-rays (or other particles) of known energy. Equalisation must be completed before attempting to calibrate the threshold mapping.

The detector should be set in the desired state for operation (readout mode, gain mode, equalisation, sensor bias etc.) and a threshold scan performed with the detector exposed to the photons. A fit to the corresponding differential peak should then be performed to find the peak of the distribution in DAC codes. This will then correspond to the energy value in the table. This should be repeated for a number of energies in the working range if possible. The resultant calibration should be linear. As the DAC codes are integer and the peak would have occurred at a fractional value, a linear fit of the measured data might be required.

An editing window is available to avoid the need to select and edit individual configuration files.

[7] On older versions of the software, it was necessary to set the mode in both places.



Open the Threshold Mapping Files window using the menu **Action->'Create Mapping Files'**. Ensure that the correct mapping files have been loaded. The energy levels will depend on the X-ray sources that you have used. For each energy level add the corresponding DAC value for each chip present. Select **Write Files** when you have finished, to save the threshold mapping. Any changes can be discarded by selecting **Read Config Data**.

11. Data File Formats

11.1. Image Data

The Merlin system saves image data in a binary format (by default with a .mib extension) with the file name and position as specified on the main image tab. The file consists of a variable length ASCII string followed by an array of binary numbers that matches the dimensions of the detector being used. The counts will be either UINT16 or UINT32 depending on whether the system was running in 12 or 24 bit mode. 6 bit, or 1 bit mode images will be saved as UINT8.[8]
If the system is set to save more than one frame per file then the file structure will remain as above and they are simply concatenated together. This produces a binary file containing a series of two dimensional unsigned integer 8 bit, 16 bit or 32 bit big-endian data arrays interspersed by ASCII strings.
The frame header string contains a human readable ASCII text string formatted as comma separated values as shown in the following tables. The “Data Type” gives an indication of the source data format.

Merlin Quad Version 1 Frame Header.

Data Type	Field Name	Description (MQ1)
String	Header ID	Identifies the Frame Header start and allows version identification should any fields change in future. This also allows Merlin Single data headers to be distinguished from Merlin Quad. Quad frames may contain data for one, or many chips. MQ1: Merlin Quad (version 1).
U32	Acquisition Sequence Number	This is the number of a frame within an acquisition sequence. This number will reset with each start acquisition.
U16	Data Offset	The offset from the beginning of the header to the beginning of the image data. The same as the total header length. This will also be the offset from the beginning of the file for the first image, or if there is only one per file.
U8	Number of Chips	The number of readout ASICs used to build up the sensor. This may be used with the sensor layout and chip select fields to determine the overall sensor and data configuration.
U32	Pixel Dimension X	The image size in the X (width) direction.
U32	Pixel Dimension Y	The image size in the Y (height) direction.
String	Pixel Depth in File	For ease of processing, the pixel size is rounded to a 2^N multiple number of byes. The only exception is single bit mode where eight pixels are packed in a byte. This field represents the pixel depth as a U8 number of bits, with “U” pre-pended. Valid values include: U01, U08, U16, U32 and U64.
String	Sensor Layout	Text string padded with leading spaces, 2x2, Nx1, 2x2G, Nx1G

[8] Versions prior to 0.65 save UINT16 as the smallest format.

Data Type	Field Name	Description (MQ1)
Hex U8	Chip Select	This is a bit field representation in hexadecimal of the chips that were active during the capture of the frame. Chip 1 is the least significant bit, chip 2, the next bit etc.
	Time Stamp	Date and time to nearest uS. Format: yyyy-mm-dd hh:mm:ss.ssssss Eg 2013-09-17 13:01:53.744951 The time stamp gives accurate relative timings between frames in the same sequence.
Double	Acquisition Shutter Time	Floating point representation of the shutter opening time in seconds.
U8	Counter	Counter 0, or 1. In colour mode, this will represent the colour layer, equivalent to Threshold 0 to 7.
U8	Colour Mode	0 = Single threshold (monochrome) image, 1 = Multi threshold (colour, or counter 0&1) image
U8	Gain Mode	0 = SLGM, 1 = LGM, 2 = HGM, 3 = SHGM.
Single	Threshold (0..7)	The values of the thresholds in keV.
	DACs	One section per chip.
	Extension(s)	Optional extended information.
String	Padding	Null.

Merlin Quad Version 1 Frame Header DAC Section.

Data Type	Field Name	Description (MQ1 DACs)
String	DAC Format	Medipix 3.0, or RX DAC layout. MQ1 only supports RX, so only the RX layout is given here. Current possible values “3RX”, “3.0
U16	Threshold 0	9 bit DAC.
U16	Threshold 1	9 bit DAC.
U16	Threshold 2	9 bit DAC.
U16	Threshold 3	9 bit DAC.
U16	Threshold 4	9 bit DAC.
U16	Threshold 5	9 bit DAC.
U16	Threshold 6	9 bit DAC.
U16	Threshold 7	9 bit DAC.
U8	Preamp	
U8	Ikrum	
U8	Shaper	
U8	Disc	
U8	Disc LS	
U8	Shaper Test	
U8	DAC Disc L	
U8	DAC Test	
U8	DAC DISC H	
U8	Delay	
U8	TP Buff In	
U8	TP Buff Out	
U8	RPZ	
U8	GND	

Data Type	Field Name	Description (MQ1 DACs)
U8	TP Ref	
U8	FBK	
U8	Cas	
U16	TP Ref A	9 bit DAC.
U16	TP Ref B	9 bit DAC.

Merlin Quad Version 1A Header Extension.

Data Type	Field Name	Description (MQ1A)
String	Header ID	Identifies the Header extension and allows version identification should any fields change in future. MQ1A: Merlin Quad (version 1, extension A).
	Extended Time Stamp	UTC date and time in nS. Format: yyyy-mm-ddThh:mm:ss.sssssssssZ Eg 2015-06-24T13:23:38.482272452Z The time stamp gives accurate relative timings between frames in the same sequence.
U64+'nS'	Acquisition Shutter Time in nS	U64 representation of the shutter opening time in nanoseconds with 'nS' suffix.
U8	Counter Depth	The original acquisition counter depth, as opposed to the file representation.

The padding should not be used as a delimiter. If additional fields are added to the header in future they will be added between the last documented field and the padding. This should allow any application that does not understand the new fields to continue working unmodified.

An example quad chip header is shown below.

```
MQ1,000001,00768,04,0512,0512,U16,2x2,0F,2014-09-23 09:54:12.000752,
0.001000,0,0,0,9.000000E+0,0.000000E+0,0.000000E+0,0.000000E+0,
0.000000E+0,0.000000E+0,0.000000E+0,0.000000E+0,3RX,000,000,000,000,
000,000,000,000,100,010,125,125,100,100,090,100,090,050,128,004,255,
154,128,206,192,256,256,3RX,000,000,000,000,000,000,000,000,100,010,
125,125,100,100,090,100,090,050,128,004,255,141,128,196,182,256,256,
3RX,000,000,000,000,000,000,000,000,100,010,125,125,100,100,090,100,
090,050,128,004,255,143,128,198,186,256,256,3RX,000,000,000,000,000,
000,000,000,100,010,125,125,100,100,090,100,090,050,128,004,255,143,
128,196,187,256,256,
```

Following the header is a binary block with the image data as specified in the header. If multiple frames were saved per file there will then be a new header section followed by the next frame of data.

Colour images are saved as a series of colour layer images. Each layer has a header and image data in the standard format. The number of layers may be four, or eight, depending on the counter mode. Four for single counter operation, eight for dual counter operation.

It is also possible to save mask, test and pixel trim data in this format for analysis purposes. When this is done a lot of the meta data associated with images will not be populated with meaningful information, but the essential fields that identify the formatting will.

11.2. Acquisition Header

This provides a summary of the detector configuration in text format.

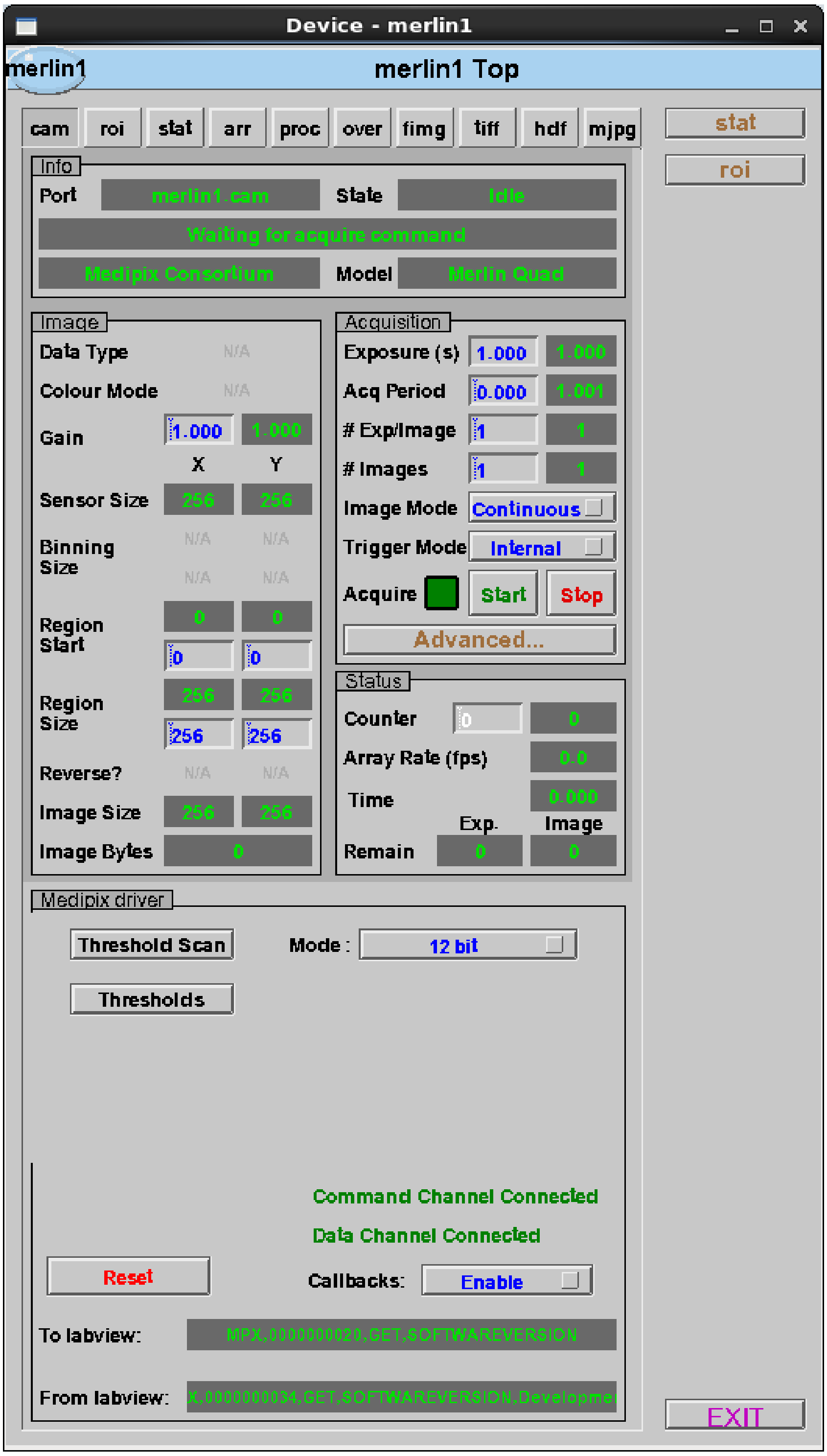
```
HDR,
Time and Date Stamp (yr, mnth, day, hr, min, s):        30/01/2015
    11:47:53
Chip ID:        W117_D10,W117_F10,W117_E11,W117_H11
Chip Type (Medipix 3.0, Medipix 3.1, Medipix 3RX):      Medipix 3RX
Assembly Size (1X1, 2X2):          2x2G
Chip Mode  (SPM, CSM, CM, CSCM):        SPM
Counter Depth (number): 12
Gain:   SLGM
Active Counters:        Counter 0
Thresholds (keV):
    1.000000E+1,1.000000E+1,1.000000E+1,1.000000E+1,1.000000E+1,1.000
    000E+1,1.000000E+1,1.000000E+1
DACs:
    056,056,056,056,056,056,056,056,100,010,125,125,100,100,120,100,0
    90,050,128,004,255,137,128,187,187,320,320;
    056,056,056,056,056,056,056,056,100,010,125,125,100,100,120,100,0
    90,050,128,004,255,158,128,214,201,320,320;
    056,056,056,056,056,056,056,056,100,010,125,125,100,100,120,100,0
    90,050,128,004,255,148,128,205,196,320,320;
    056,056,056,056,056,056,056,056,100,010,125,125,100,100,120,100,0
    90,050,128,004,255,144,128,194,188,320,320
bpc File:
    c:\MERLIN_Quad_Config\W117_D10\W117_D10_SPM.bpc,c:\MERLIN_Quad_Co
    nfig\W117_F10\W117_F10_SPM.bpc,c:\MERLIN_Quad_Config\W117_E11\W11
    7_E11_SPM.bpc,c:\MERLIN_Quad_Config\W117_H11\W117_H11_SPM.bpc
DAC File:
    c:\MERLIN_Quad_Config\W117_D10\W117_D10_SPM.dacs,c:\MERLIN_Quad_C
    onfig\W117_F10\W117_F10_SPM.dacs,c:\MERLIN_Quad_Config\W117_E11\W
    117_E11_SPM.dacs,c:\MERLIN_Quad_Config\W117_H11\W117_H11_SPM.dacs
Gap Fill Mode:  None
Flat Field File:        Dummy (C:\<NUL>\Temp.ffc)
Dead Time File: Dummy (C:\<NUL>\Temp.dtc)
Acquisition Type (Normal, Th_scan, Config):     Normal
Frames in Acquisition (Number): 1
```

```
Frames per Trigger (Number):    1
Trigger Start (Positive, Negative, Internal):   Internal
Trigger Stop (Positive, Negative, Internal):    Internal
Sensor Bias (V, µA)     80 V
Sensor Polarity (Positive, Negative):   Positive
Temperature (C):        FPGA Temp 41.500000 Deg C
Medipix Clock (MHz):    120MHz
Readout System: Merlin Quad
Software Version:       0.64.7
End
```

12. Operating Merlin via EPICS/GDA

This section provides instructions for running the Merlin via the EPICS/GDA interface.

The EPICS Merlin interface is built on the Area Detector Record software, and for non Medipix specific actions such as file saving image processing and the live preview the user should refer to the documentation for this suite. Via the EPIC interface GDA scripts can perform acquisitions in the usual way for Area Detector Record devices.



The Merlin EPICS front panel is shown opposite. The CAM tab panels that provide Merlin specific functions are *Acquisition* and *Medipix Driver.*

12.1. Acquisition

Exposure: This sets the time in seconds that the shutter will be open for. As with all these variables, the value is set in the blue and grey box and read back from Merlin in the green and grey box.

Acq Period: The period with which multiple acquisitions are repeated in multiple image mode. The minimum that can be set is the exposure time added to the minimum readout time. If a period value is set too low the Merlin system will enforce a minimum possible readout time. Care should be taken to reduce this when the exposure time is reduced as it is not adjusted automatically.

#Exp/Image: This has no function and should remain set at 1

#Images: In multiple image mode this sets the number of images to be taken. **CAUTION** If a high frame rate has been selected it is up to the user to ensure that this does not overflow the 1200 frames at 1kHz limit on the local hardware memory.

Image Mode: This allows you to select a number of acquisition scenarios.

- **Single:** where only one image is ever taken.
- **Multiple:** where the number of images taken is equal to the value set in **#Images**.
- **Continuous:** where the system will take images at the defined rate indefinitely. **CAUTION** In continuous mode it is possible to rapidly overflow the internal buffer and or to generate a very substantial quantity of data which could cause other problems with EPICS/GDA, if the acquisition period is not long enough – use with care. This mode is normally used for experimental set-up and is not to be confused with Continuous RW.
- **Threshold:** Run a threshold scan.
- **Background:** See EPICS documentation.

Trigger Mode: This allows you to select whether the system will respond to an external trigger or not.

- **Internal:** uses the Merlin clock to drive the shutter and does not accept external triggers.
- **Trigger Enable:** allows the **TTL Trigger In** signal to open and close the shutter as a gating signal.
- **Trigger start rising:** allows the **TTL Trigger In** signal to open the shutter on the rising edge, but the exposure is internally timed.
- **Trigger start falling:** allows the **TTL Trigger In** signal to open the shutter on the falling edge, but the exposure is internally timed.
- **Trigger both rising:** allows the **TTL Trigger In** signal to open the shutter on one rising edge, then close the shutter on the next rising edge.
- **Software:** This is mislabelled. It actually allows the **LVDS Trigger In** signal to open the shutter on the rising edge, but the exposure is internally timed. Confirm the operation on your own installation before use.

Start/Stop: These buttons start an acquisition of one or more images, or cancel a run in progress. The indicator 'light' will show when the acquisition is being taken.

Advanced: This opens a new panel that gives access to a number of general Area Detector Record features that are not specific to Merlin.

12.2. Medipix Driver

Mode: This allows you to select a number of predefined mode setting combinations.

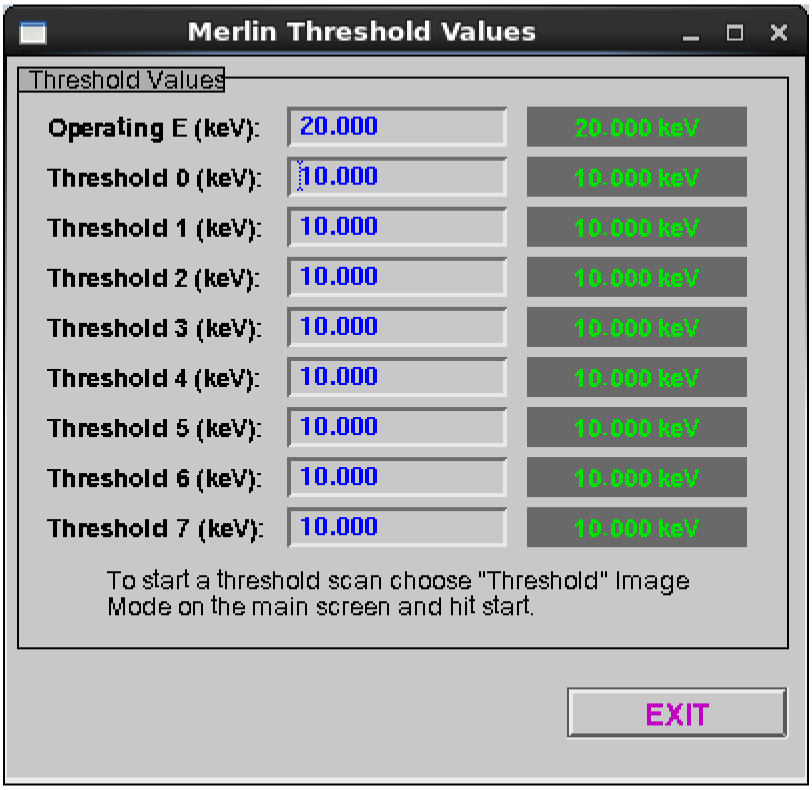
- **12 bit:** This allows the system to be set in 24 bit mode.
- **24 bit:** This allows the system to be set in 24 bit mode. 24 bit mode has a double readout length of 1.7ms.

- **Two Threshold:** When this setting is enabled each shutter open period produces two images with separate thresholds. The upper threshold is set by the Threshold 1 setting.
- **Continuous RW:** This mode allows multiple images to be taken with no readout gap between them. The Acq Period setting is ignored and the Exposure time is used. The exposure time cannot be reduced below the minimum readout time of 820µs in this mode.
- **Colour:** This allows the system to be set in eight colour mode for 110µm pitch spectroscopic sensors.
- **Charge Summing:** This allows the system to be set in charge summing mode.

Thresholds: This opens a new panel that gives access to individual threshold settings.

Threshold Scan: This opens a new panel that gives access to the threshold scan settings.

12.3. Merlin Threshold Values



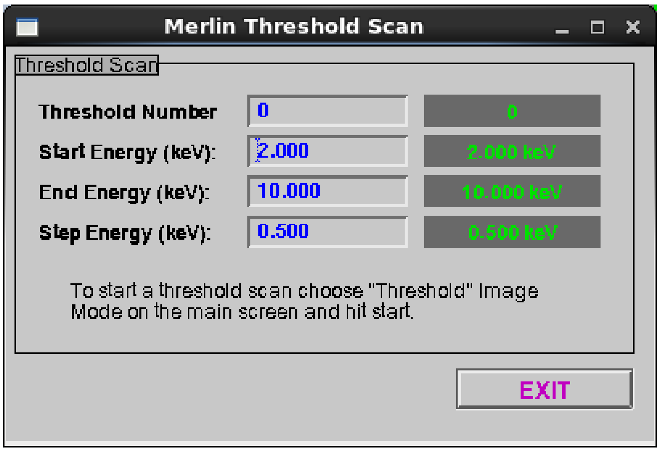
Threshold 0: This sets the energy in keV of the lower threshold of the chip. This is the threshold generally used in the normal (single pixel / single counter) modes of operation. The effect of the threshold has to be calibrated when the detector is set up to allow a reliable conversion from keV to units compatible with the chip's operation. A look-up table described in the Chip Configuration Files section is used.

Threshold 1-7: Similar to Threshold 0, Threshold 1 sets the upper threshold for acquisitions using the dual energy mode when Counter 1 is enabled. Threshold 2-7 are used for colour mode imaging. They uses the same lookup table as Threshold 0.

Operating Energy: This can be set instead of setting the Thresholds. It automatically sets the thresholds to half its value. This will override any value set in the Threshold boxes, however any subsequent value set in a threshold will re assert itself.[9]

[9] On versions of Merlin prior to 0.65.7 setting operating energy alone may result is sub optimal operation. On all versions to date, this control only works for fine pitch single pixel operation. For CSM and Colour modes, the thresholds must be set individually. In all cases, setting individual thresholds is recommended over using set operating energy.

12.4. Merlin Threshold Scan



Threshold Scan: This opens the threshold scan popup shown below that allows a series of acquisitions to be taken with incrementing threshold. Be aware that due to the quantisation of the threshold DACs and the scaling between keV and DAC codes the number of images recorded may be less than you would expect, especially if you are using a step size approaching the minimum energy resolution of the Threshold.

Reset: This resets the link to the Merlin system and may be used in case of a communication problem.

Callbacks: This setting is a generic Area Detector Record setting, by enabling it you allow the data to be passed to other parts of the control system such as file writing. If it is disabled the images recorded will not be sent beyond the initial panel.

12.5. Merlin Local Configuration with EPICS

There are certain items that should be set up in the sensor command file (see section 7.5 Command File above): HVBIAS and POLARITY. These are required regardless of whether or not EPICS is in use.

Trigger Out and Gain commands may be set in the command file as they are not controlled by EPICS. Future versions of EPICS might add support so it would be wise to check. This can be done by examining EPICS GUI for the addition of any Trigger Out, or Gain control, or by monitoring the commands sent by EPICS to Merlin, see section 6.11 TCP/IP Server Status Window above.

Other commands should be avoided. At best you may be confused as to what defaults are set when EPICS overrides the command file settings, or at worst, there might be a race and the system settings will be unreliable.

For the same reason, you should avoid using the Merlin GUI when operating Merlin from EPICS, or any other system using the Merlin TCP interface. A key exception is set-up or script debugging, where monitoring the Merlin GUI can confirm correct operation and help to find issues.

12.6. GDA Scripts

GDA can access all the EPICS process variables, so it can set up any aspect of Merlin that is accessible to EPICS.

13. TCP/IP Remote Command Reference

A key feature of the Merlin system is that it is controllable via a remote command system based on passing TCP/IP text messages. This allows the system to be easily driven by a wider experimental control system or controlled by a custom program or scripting language.

The communication of the system is split between two channels, a command channel and a data channel. The command channel is used to pass short ASCII messages in both directions, with the client program sending commands to the Merlin system and the Merlin responding immediately. The Data channel is unidirectional and is used to send image data from the Merlin to the client program. The channels have been separated in this way to avoid heavy use on the data channel flooding the command channel and risking a system lock.

The command channel is capable of receiving two connections. One of these is required to be free in order to use the internal command file option. Merlin listens on ports 6341 and 6342 until an external control system connects on these ports. If the connection is dropped for any reason, the Merlin will re-enter the listening state until the external system reinitiates the connection.

13.1. Command Channel Communication

The command channel listens on port 6341 and waits for a TCP/IP connection. Once this is made command strings may be sent to the system, these need to be of the form:

`MPX,0000000025,SET,NUMFRAMESTOACQUIRE,1`

Where the element `MPX` identifies the string as a Merlin command.

Followed by the ten character ASCII length of the remainder of the message for example `0000000025,`

This is followed by one of three action statements `SET`, `GET`, or `CMD`. These define the nature of the operation to be performed. SET – sets the value of an internal variable, GET returns the currently stored value and CMD commands the system to take an action.

This is followed by the name of the command or the variable, in the example above `NUMFRAMESTOACQUIRE` sets the number of frames per acquisition.

This is followed by a parameter if the system is setting a variable.

When the system receives a command that starts with a valid MPX identifier it generates an immediate response. The response if of the form:

`MPX,0000000025,GET,NUMFRAMESTOACQUIRE,5,0`

Which is an echo of the body of the received command, with the last parameter giving a code number that signifies whether the command has been recognised.
0 denotes understood
1 denotes the system is busy

2 denotes that the command has not been recognised.
3 denotes parameter out of range.

The last but one parameter denotes a value returned from a GET command, if a GET command was sent. In the case of a SET or CMD statement this value is not present.
The response is sent as an acknowledgement that the command has been received, not as an indication that the command has been fully processed. More commands can be sent immediately, but care is needed when switching from configuration, SET, command and action, CMD, commands. Configuration must be allowed to complete before beginning an acquisition sequence, and vice versa.

13.2. Data Channel Communication

The data channel communicates on Port 6342 and sends the data in the same format as a series of frames in exactly the same manner as the binary data file described in section 11 Data File Formats above with the MPX identifier and data length prefix. Optionally there may be an Acquisition Header frame preceding the data frames. Again, this follows the Acquisition Header file format with the MPX identifier and data length prefix.

On receipt of a start acquisition command:

```
MPX,<length>,CMD,STARTACQUISITION
```

Merlin will immediately respond with the following on the command channel:

```
MPX,<length>,CMD,STARTACQUISITION,0
```

Then immediately on the data channel (assuming the command can proceed) an Acquisition header will be sent to record details about the system settings for this acquisition. The header will formatted as follows:

```
MPX,<length>,HDR,<ASCII Acquisition Header>
```

<length> = Length of following data in bytes, i.e. length of “HDR,<ASCII Acquisition Header>”. Encoded as 10 chars of ASCII decimal representation with leading zeros.

Merlin Quad data frame header begins MQn, where n is a format version identifier. Currently only MQ1 is defined (see section 11 Data File Formats, above).

```
MPX,<length>,MQ1,<ASCII Header><pixel data>
```

It is important to note that data is only streamed to the TCP data channel while the TCP data port is in the connected state. The receiver must initiate this connection before beginning an acquisition, or the data will not be available for streaming.

13.3. TCP/IP Commands List

A list of commands used to control the detector over TCP/IP.

Driver Variables

Name	Type	Notes	Format / Range
SOFTWAREVERSION	GET	String	#.##.#.#, but the major revision part may be interpreted as Floating Point ##.##

Execute Commands (CMD type)

Name	Type	Notes
STARTACQUISITION	CMD	Start an acquisition. If the start trigger mode is internal, the acquisition will begin immediately; otherwise the detector will arm and wait for a trigger.
STOPACQUISITION	CMD	Stop any running acquisition, finishes sending frame
SOFTTRIGGER	CMD	Triggers an acquisition that has been configured for soft trigger and started.
THSCAN	CMD	Perform a threshold scan
RESET	CMD	Restores default configuration values

Acquisition Modes

Name	Type	Units	Notes	Format Range
COLOURMODE	SET/GET		0=Monochrome 1=Colour	0 – 1
CHARGESUMMING	SET/GET		0=Off, 1=ON	0 – 1
GAIN	SET/GET		0=Super Low 1=Low 2=High 3=Super High	0 – 3
CONTINUOUSRW	SET/GET		0=Off, 1=ON	0 – 1
ENABLECOUNTER1	SET/GET		0=Counter 0 1=Counter 1 2-Counters 0 & 1	0 – 2
THRESHOLD0	SET/GET	keV	Single	0-999.99
THRESHOLD1	SET/GET	keV	Single	0-999.99
THRESHOLD2	SET/GET	keV	Single	0-999.99
THRESHOLD3	SET/GET	keV	Single	0-999.99
THRESHOLD4	SET/GET	keV	Single	0-999.99
THRESHOLD5	SET/GET	keV	Single	0-999.99

Name	Type	Units	Notes	Format Range
THRESHOLD6	SET/GET	keV	Single	0-999.99
THRESHOLD7	SET/GET	keV	Single	0-999.99
OPERATINGENERGY	SET/GET	keV	Single	0-999.99
COUNTERDEPTH	SET/GET	bits	Integer (1,6,12,24).	1, 6, 12 or 24
FILLMODE	SET/GET		Gap Fill 0=None 1=Zero Fill 2=Distribute 3=Interpolate [10]	0 – 3
FLATFIELDCORRECTION	SET/GET		0=Off, 1=ON (if path is set) [11]	0 – 1
FLATFIELDFILE	SET/GET	File Path	MIB image, or coefficient file. [11]	

[10] FILLMODE command added in Version 0.65.
[11] FLATFIELDCORRECTION and FLATFIELDFILE commands added in version 0.65.

Acquisition & Trigger Control

Name	Type	Units	Notes	Format / Range
NUMFRAMESTOACQUIRE	SET/GET	Frames	Integer. The number of frames to acquire when the next STARTACQUISITION command is sent, or hardware trigger is received. If the value is zero, acquisition should continue until a STOPACQUISITION command.	0 – 100,000
ACQUISITIONTIME	SET/GET	Milliseconds	Single. The time over which to acquire a frame in milliseconds.	Scientific format
ACQUISITIONPERIOD	SET/GET	Milliseconds	Single. The time from the start of one acquisition to the start of the next, in milliseconds.	Scientific format

Name	Type	Units	Notes	Format / Range
TRIGGERSTART	SET/GET		0=Internal (default), 1=Rising Edge (TTL), 2=Falling Edge (TTL), 3=Rising Edge (LVDS), 4=Falling Edge (LVDS), 5=Soft Trigger, 6=Multi-Gate Rising Edge (TTL), 7= Multi-Gate Falling Edge (TTL), 8= Multi-Gate Rising Edge (LVDS), 9= Multi-Gate Falling Edge (LVDS), 10= Multi-Gate Soft Trigger [12]	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
TRIGGERSTOP	SET/GET		0=Internal (default), 1=Rising Edge (TTL), 2=Falling Edge (TTL), 3=Rising Edge (LVDS), 4=Falling Edge (LVDS)	0, 1, 2, 3, 4
NUMFRAMESPERTRIGGER	SET/GET	Frames	Integer. Number of frames to acquire per trigger pulse.	1 – 100,000
TriggerOutTTL	SET/GET		0=Trigger In TTL, 1=Trigger In LVDS, 2=Trigger In TTL Delayed, 3=Trigger In LVDS Delayed, 4=Follow Shutter, 5=One per Acq Burst, 6=Shutter and Sensor Readout, 7=Busy, 8=Soft Trigger Out [13]	0 – 7
TriggerOutLVDS	SET/GET		0=Trigger In TTL, 1=Trigger In LVDS, 2=Trigger In TTL Delayed, 3=Trigger In LVDS Delayed, 4=Follow Shutter, 5=One per Acq Burst, 6=Shutter and Sensor Readout, 7=Busy, 8=Soft Trigger Out [13]	0 – 7
TriggerOutTTLInvert	SET/GET		0=Normal 1=Inverted	0 – 1
TriggerOutLVDSInvert	SET/GET		0=Normal 1=Inverted	0 – 1
TriggerInTTLDelay	SET/GET	nS	Long Integer (10nS increments)	0 – 42949672950
TriggerInLVDSDelay	SET/GET	nS	Long Integer (10nS increments)	0 – 42949672950

[12] TRIGGERSTART Multi-Gate parameters 6-10 added in Version 0.65.
[13] TriggerOutTTL and TriggerOutLVDS Soft Trigger Out option added in Version 0.65.

Name	Type	Units	Notes	Format / Range
TriggerUseDelay	SET/GET		0=No Delay 1=Apply delay to start and stop	0 – 1
SoftTriggerOutTTL	SET/GET		0=Trigger Off 1=Trigger Active [14]	0 – 1
SoftTriggerOutLVDS	SET/GET		0=Trigger Off 1=Trigger Active [14]	0 – 1
TriggerInTTL	GET		0=Trigger Off 1=Trigger Active [14]	0 – 1
TriggerInLVDS	GET		0=Trigger Off 1=Trigger Active [14]	0 – 1

[14] SoftTriggerOutTTL, SoftTriggerOutLVDS, TriggerInTTL and TriggerInLVDS were added in Version 0.65.

Threshold Scan Control

Name	Type	Units	Notes	Format / Range
THSCAN	SET/GET		Integer 0-7.	0 - 7
THSTART	SET/GET	keV	Single Start energy of the scan.	0 – 999.99
THSTOP	SET/GET	keV	Single Stop energy of the scan.	0 – 999.99
THSTEP	SET/GET	keV	Single Scan step size.	0 – 999.99
THNUMSTEPS	SET/GET		Integer Fixes the number of steps to avoid ambiguities around end limit.	0 – 511

Local File Saving Control

Name	Type	Units	Notes	Format / Range
FILEDIRECTORY	SET/GET		String. 256 chars max.	Windows File Path Format
FILENAME	SET/GET		String. 256 chars max.	Windows File Name Format
FILEENABLE	SET/GET		0=Disable, 1=Enable. When enabled the Merlin software saves each frame to the local disk.	0 – 1

Status

Name	Type	Units	Notes	Format / Range
DETECTORSTATUS	GET		0=IDLE, 1=BUSY, 2=Standby	0 – 2
TEMPERATURE	GET	degrees C	Single	-100 – 200

Local Configuration

Name	Type	Units	Notes	Format / Range
POLARITY	SET/GET		0=Negative (electron collection), 1=Positive (hole collection)	0 – 1
HVBIAS	SET/GET		Volts (internal generator only)	0 – 120

Local configuration commands are intended for use in configuration command files (see section 7.5 Command File, above) rather than by TCP control, however, they are listed here as they share the same command interpreter.

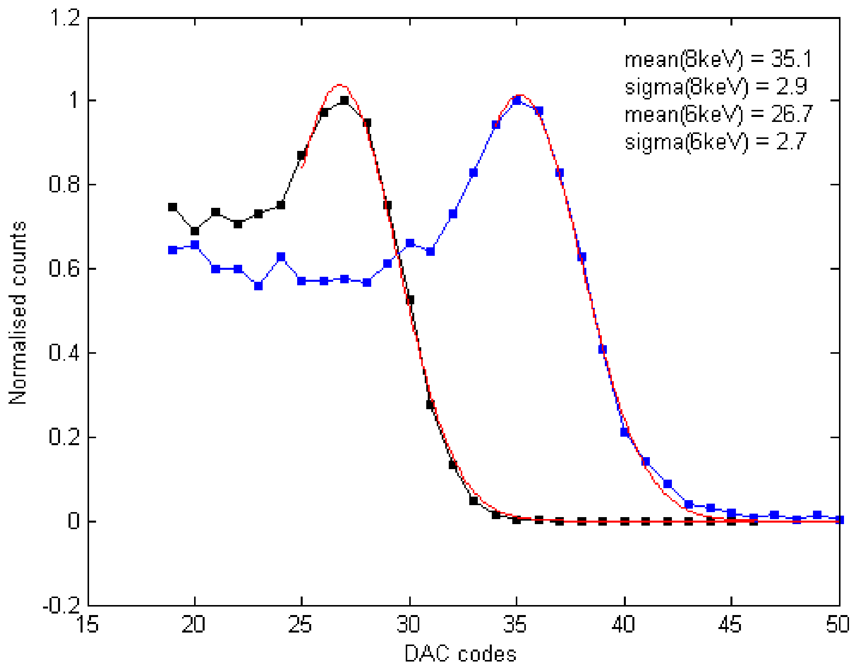
14. Measurements of Performance

14.1. Measurement of Threshold Resolution and Pixel Energy Dispersion

Two measurements were taken in High Gain Mode to determine the Threshold resolution and Pixel energy Dispersion. These were threshold scans with an Fe55 radioactive source and a copper fluorescence target bombarded by an x-ray tube.

In both cases the same noise edge Trim Scan equalisation was used with the chip in HGM.

A differential plot of the mean counts each scan was made and normalised to take account of the differing fluxes. Fits were performed to the appropriate part of scan as shown in the figure below.

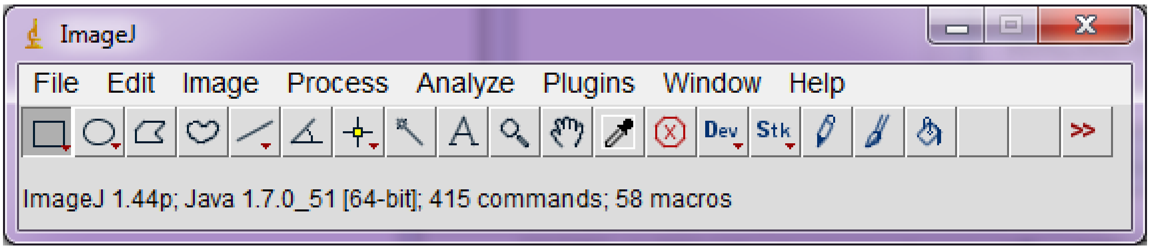


As the distance between the two energies is known to be 2keV the separation of the two mean values in DAC codes allows us to conclude that each DAC code corresponds to 238eV.

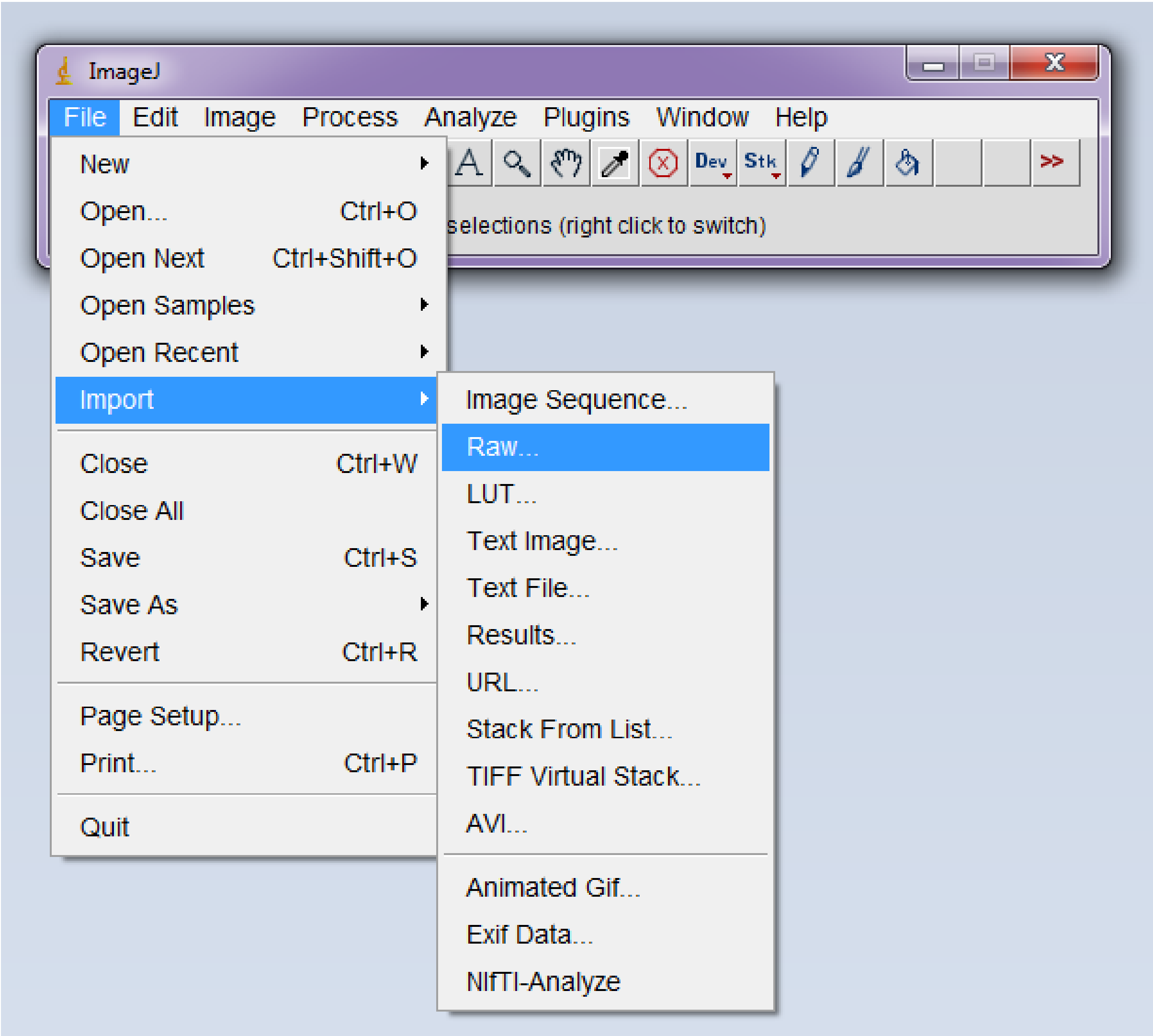
As the width of the iron peak is 2.7 DAC codes we conclude this is 643eV which corresponds to 1.51keV FWHM.

15. Using ImageJ to Read Merlin Data Files

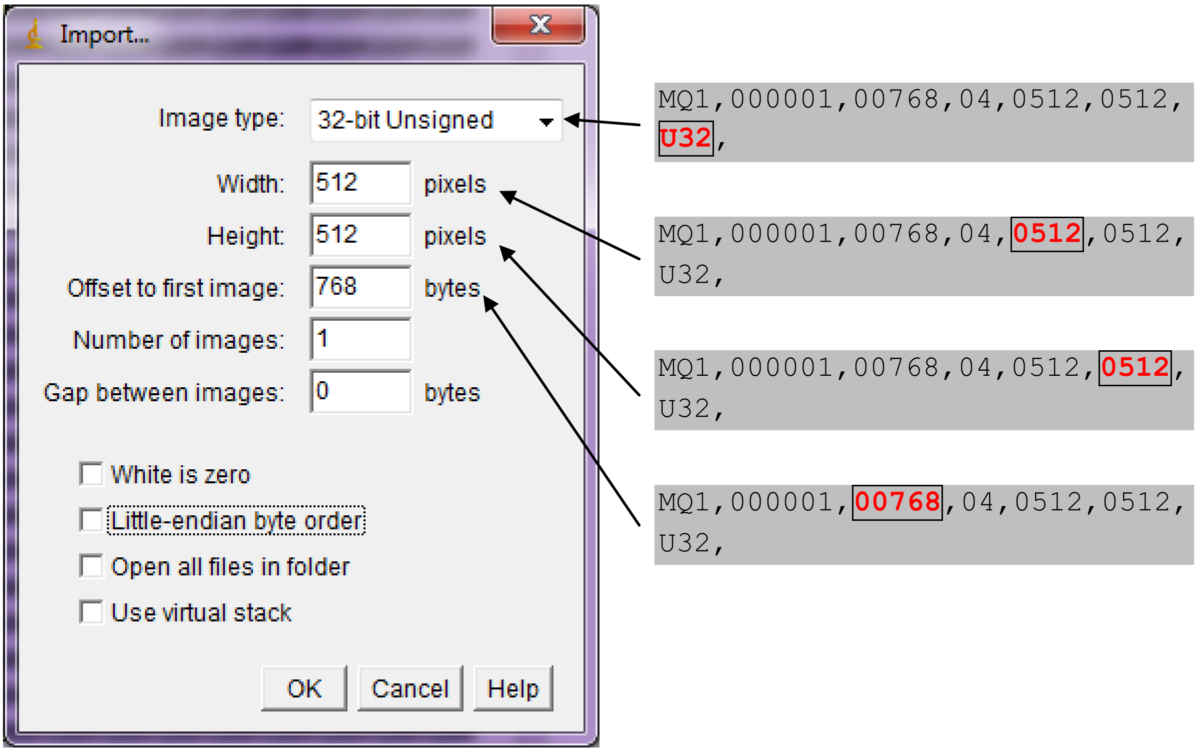
Merlin data files are saved in a raw binary format as described in section 11.1 above. Most image viewers will not be able to open this format, but these files can be read with the open source ImageJ program. Here are some brief instructions:



Merlin Image files must be imported, not opened. Use the File-Import-Raw menu option.



ImageJ cannot read the header information from the file, so this must be entered manually. You can pull out the first part of the image header using a text editor to check the relevant parameters, for example: "`MQ1,000001,00768,04,0512,0512,U32,`"



Uncheck the tick boxes as show. It is possible to use the "Open all files in folder" option, if there are not too many files. Exactly how many is too many will depend on your computer specifications.

Please see the ImageJ documentation for further aspects of this program.

16. References and Further Reading

Medipix3 Manual – Available on request from the Medipix3 Collaboration

Medipix3RX Manual – Available on request from the Medipix3 Collaboration

R. Ballabriga *et al,* "*Characterization of the Medipix3 Pixel Readout Chip*", 2011 *JINST* **6** C01052

R. Ballabriga *et al,* "*Medipix3: A 64 k pixel detector readout chip working in single photon counting mode with improved spectrometric performance*" , NIM A 633 S15 – S18, 2011.

R. Plackett et al, "*Merlin: a fast versatile readout system for Medipix3*" ,R Plackett *et al* 2013 *JINST* **8** C01038.

Ballabriga, Rafael, et al., et al. The Medipix3RX: A high resolution, zero dead-time pixel detector readout chip allowing spectroscopic imaging. IWORID. 2012.

Pennicard, David. Simulations of Medipix3; charge summing and dispersion. s.l. : DESY, CERN Medipix 3 Collaboration, 09 2010. Vols. 2010-09.

Simulations of charge summing and threshold dispersion effects in Medipix3. **Pennicard, D**. s.l. : Elsevier, 2011.

The Medipix3RX: a high resolution, zero dead-time pixel detector readout chip allowing spectroscopic imaging. **Ballabriga, R**., et al., et al. s.l. : JINST / IOP PUBLISHING FOR SISSA MEDIALAB, 2013. 14th INTERNATIONAL WORKSHOP ON RADIATION IMAGING DETECTORS.

Count rate linearity and spectral response of the Medipix3RX chip coupled to a 300mm silicon sensor under high flux conditions. **Frojdh, E**., et al., et al. Paris : JINST / IOP PUBLISHING FOR SISSA MEDIALAB, 2014. 15th INTERNATIONAL WORKSHOP ON RADIATION IMAGING DETECTORS.