Single **Chip Medipix Software Design**

Contents

[1. Revision History 2](#_Toc313958185)

[2. Software Overview 2](#_Toc313958186)

[3. Command Format 3](#_Toc313958187)

[3.1. Format for SET 3](#_Toc313958188)

[3.2. Request Format for CMD/GET 4](#_Toc313958189)

[3.3. Response Format for SET/CMD 4](#_Toc313958190)

[3.4. Response Format for GET 4](#_Toc313958191)

[3.5. Error Codes 5](#_Toc313958192)

[4. Command Examples 5](#_Toc313958193)

[5. Full List Of Commands (used by EPICS) 6](#_Toc313958194)

[5.1. Driver Variables 6](#_Toc313958195)

[5.2. Execute Commands (CMD type) 6](#_Toc313958196)

[5.3. Medipix3 Modes 7](#_Toc313958197)

[5.4. Acquisition & Trigger Control 7](#_Toc313958198)

[5.5. Threshold Scan Control 8](#_Toc313958199)

[5.6. Local File Saving Control 8](#_Toc313958200)

[5.7. Image Correction 9](#_Toc313958201)

[5.8. File and Status 9](#_Toc313958202)

[6. Data Frame Format 9](#_Toc313958203)

[6.1. Overview 9](#_Toc313958204)

[6.2. Data Frame Header 11](#_Toc313958205)

[6.3. Acquisition Header 12](#_Toc313958206)

# **Revision History**

|  |  |  |
| --- | --- | --- |
| Issue | Changes | Authors |
| 1.0 | Initial Draft | Matt Pearson |
| 1.1 | Addition of Data channel message format | Richard Plackett  Giles Knap |
| 1.2 | Revisions after review meeting 09/01/2012 | Giles Knap |
| 1.3 | Minor changes to header protocol | Giles Knap |

# Software Overview

The interface library is a set of functions handling socket connection, set and get commands and reading the data.

The interface library will have two threads:

* One for sending commands and receiving responses. This will not block, and will be run in the same context as the areaDetector command thread. Commands must be able to be sent while acquiring data (eg. Stopping acquisition, or reading acquisition status).
* A thread to read the data frames from a dedicated socket. This will block, waiting for data to be ready in the socket buffer.

Every data acquisition will produce a data frame, or a series of data frames. All frames will be sent to the interface library via the data socket. Files can, in addition, be saved locally to disk.

There will be the following functions in the interface library:

* Set command function
* Get command function
* Read data frame (blocks until data is ready)
* Connect to the detector (creates sockets, starts data read thread, issues connections).
* Disconnect from the detector (issues a close on the sockets, destroys data thread).

Connection status will be fed back to the areaDetector software in an error code.

The set and get functions will take the Medipix command and send it to the Labview program.

# Command Format

## Format for SET

<header>,<length>,<type>,<name>,<value>

* <header> = MPX
* <length> = Length of following data in bytes
  + i.e. length of “,<type>,<name>,<value>”.
  + Encoded as 10 chars of ASCII decimal representation with leading zeros.
* <type> = SET
* <name> = name of the variable
* <value>= the value to set

All requests are in ASCII format (upper case).

## Request Format for CMD/GET

<header>,<length>,<type>,<name>

* <header> = MPX
* <length> = Length of following data in bytes
  + i.e. length of “,<type>,<name>”
  + Encoded as 10 chars of ASCII decimal representation with leading zeros.
* <type> = GET or CMD
* <name> = name of the function or variable

All requests are in ASCII format (upper case).

## Response Format for SET/CMD

After each SET or CMD command, an acknowledgement must be returned:

<header>,<length>,<type>,<name>,<error code>

* <header> = MPX
* <length> = Length of following data in bytes
  + i.e. length of “,<type>,<name>,<error code>”
  + Encoded as 10 chars of ASCII decimal representation with leading zeros.
* <type> = SET or CMD
* <name> = name of the function or variable
* < error code > = 0 for no error, or non-zero integer for error.

All responses are in ASCII format (upper case).

## Response Format for GET

For GET commands, the value and error code are both returned as follows:

<header>,<length>,<type>,<name>,<value>,<error code>

* <header> = MPX
* <length> = Length of following data in bytes
  + i.e. length of “,<type>,<name>,<value>,<error code>”
  + Encoded as 10 chars of ASCII decimal representation with leading zeros.
* <type> = GET
* <name> = name of the variable
* <value> = the value of the variable
* <error code> = 0 for no error, or non-zero integer for error.

If error is non zero then value should be ignored.

All responses are in ASCII format (upper case).

## Error Codes

Error numbers are:

0 = No Error / OK

1 = Unknown Error

2 = Unknown Command

3 = Param Out Of Range

>4 TBD

# Command Examples

All fields are ASCII.

**A set command example:**

Request

MPX,0000000025,SET,NUMFRAMESTOACQUIRE,1

Response

MPX,0000000025,SET,NUMFRAMESTOACQUIRE,0

**A get command example:**

Request:

MPX,0000000023,GET,NUMFRAMESTOACQUIRE

Response:

MPX,0000000027,GET,NUMFRAMESTOACQUIRE,1,0

**A CMD command example:**

Request

MPX,0000000021,CMD,STARTACQUISITION

Response

MPX,0000000023,CMD,STARTACQUISITION,0

# Full List Of Commands (used by EPICS)

## Driver Variables

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Notes** | **Format / Range** |
| GETSOFTWAREREVISION | GET | Floating Point | ##.## |

## Execute Commands (CMD type)

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Notes** |
| STARTACQUISITION | CMD | Start an acquisition |
| STOPACQUISITION | CMD | Stop any running acquisition, finishes sending frame |
| ABORT | CMD | Aborts any acquisition or process, terminates data stream |
| SELFTEST | CMD | Perform a self test. Report will be stored locally with unique file name, default option for EPICS |
| NOISEEQUALISATION | CMD | Performs equalisation on noise |
| FLATFIELDCORRECTION | CMD | Perform a flat field correction |
| THSCAN | CMD | Perform a threshold scan |
| RESET | CMD | Restores default configuration values |

## Medipix3 Modes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Units** | **Notes** | **Format Range** |
| COLOURMODE | SET/GET |  | Colour mode: 0=Monochrome 1=Colour | 0 – 1 |
| CHARGESUMMING | SET/GET |  | 0=Off, 1=ON | 0 – 1 |
| HIGHGAIN | SET/GET |  | 0=Off, 1=ON | 0 – 1 |
| CONTINUOUSRW | SET/GET |  | 0=Off, 1=ON | 0 – 1 |
| ENABLECOUNTER1 | SET/GET |  | 0=Off, 1=ON | 0 – 1 |
| 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 |
| 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 |  | Integer (1, 4, 12 or 24). Maybe 6 for vs 3.2 | 1, 4, 12 or 24, 6 don't check |
| TEMPERATURE | GET | degrees C | Single | -100 - 200 |

## Acquisition & Trigger Control

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Units** | **Notes** | **Format / Range** |
| NUMFRAMESTOACQUIRE | SET/GET |  | 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 |
| TRIGGERSTART | SET/GET |  | 0=internal (default), 1=rising edge, 2=falling edge | 0, 1, 2 |
| TRIGGERSTOP | SET/GET |  | 0=internal (default), 1=rising edge, 2=falling edge | 0, 1, 2 |
| NUMFRAMESPERTRIGGER | SET/GET |  | Integer. Number of frames to acquire per trigger if multiple external triggers. | 1 - 100,000 |

## Threshold Scan Control

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Units** | **Notes** | **Format / Range** |
| THSCAN | SET/GET |  | Integer 0-7. | 0 - 7 |
| THWINDOWMODE | SET/GET |  | 0=Off, 1=ON | 0 – 1 |
| THWINDOWSIZE | SET/GET | keV | Single The set size between counter 0 and 1. | 0-999.99 |
| 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 |

## 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 |
| FILEFORMAT | SET/GET |  | 0=Binary, 1=ASCII | 0 -1 |
| FILECOUNTER | SET/GET |  | Integer. Set the file number to start at. | 1 - 100,000 |
| FILEENABLE | SET/GET |  | 0=Disable, 1=Enable. When enabled the Labview software saves each frame to the local disk. | 0 - 1 |

## Image Correction

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Units** | **Notes** | **Format / Range** |
| FLATFIELDCORRECTION | SET/GET |  | 0=OFF, 1=ON | 0 - 1 |
| DEADTIMECORRECTION | SET/GET |  | 0=OFF, 1=ON | 0 - 1 |
| MASKINDATA | SET/GET |  | 0=OFF, 1=ON | 0 - 1 |

## File and Status

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Units** | **Notes** | **Format / Range** |
| DACFILE | SET/GET |  | String (full path and filename). 256 chars max. | Windows File Path Format |
| PIXELMATRIXFILE | SET/GET |  | String (full path and filename). 256 chars max. | Windows File Name Format |
| DETECTORSTATUS | GET |  | 0=IDLE, 1=BUSY | 0 - 1 |

# Data Frame Format

## Overview

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.

The <ASCII Acquisition Header> is always 2044 bytes long for a total ok 2K bytes including the HDR field.

All data channel responses begin with the MPX identifier followed by length and then by one of the following:

* HDR – data acquisition header
* 12B – a data frame with 12 bit pixels
* 24B – a data frame with 24 bit pixels

Immediately following the completion of the first frame the data frame will be sent on the data channel with an embedded frame header (see below

MPX,<length>,12B,<ASCII Header><pixel data 65536 x 16bit>

This will be repeated after every frame until the acquisition is complete.

<ASCII Header> is always 252 bytes long for a total of 256 bytes including the ‘12B,’ header.

The pixel data will be LSB first Pixels are sent in rows from left to right, starting with bottom row and proceeding sequentially to the top row (each rows is 256 pixels and there are 256 rows).

In the case of two counters being used independently (with two independent thresholds or threshold Window Mode), two frame communications will be made sequentially after each frame. The data frame will then look like this:

MPX,<length>,12B,<ASCII Header>,<pixel data 65536 x 16bit>

In the case of two counters being used together, a single frame communication will be made after each frame. The data frame will then look like this:

MPX,<length>,24B,<ASCII Header>,<pixel data 65536 x 32bit>

In this case of the pixel data component will be 65536 x 32bit values. These represent the 24 bit pixels and the top byte is padded with zeros.

## Data Frame Header

The 252 character frame header will be a comma separated list of ASCII values as follows. It is intended to record settings that could change from frame to frame. The Header is space padded to 252 bytes.

<frame\_number>,<counter\_number>,<start\_time>,<duration>,<Th0>,<Th1>,<DAC001>,<DAC002> … <DAC100>

The meaning of each value is as follows:-

<frame\_number> The number of the frame within the acquisition <counter\_number> Counter0 or counter1 data

<start\_time> The time the shutter opened

<duration> The length of time the shutter was open

<Th0> The value of Th0 in keV

<Th1> The value of Th1 in keV

<DACs> The values of the DACs

The format of each value is as follows

* <frame number> = unsigned integer (6 characters)
* <counter number> = 0 or 1
* <start time> = formatted as 2012-01-06 11:26:00.000
* <duration> = number of milliseconds represented as floating point in scientific notation, single precision 6 characters.
* <Th0> = floating point, single precision.
* <Th1> = floating point, single precision.
* <DACs>= integer values (max 3 digits)

Notes:

* The frame number is reset by a new acquisition
* Data frame should support up to 128KB approx. At 100Hz this is 12.8MB/s or 100Mbits/s.

## Acquisition Header

The acquisition header is 2kB (less 4 bytes) of ASCII in a verbose human readable format as follows in the below example. This may be expanded if other data is required. This is included to allow system parameters to be incorporated into the hd5/nexus format as required.

Chip ID:

W4\_F6

Chip Type (Medipix3.0, Medipix3.1, Medipix3.2):

Medipix3.1

Assembly Size (single, 2by1, 3by1, 4by1, quad):

single

Chip Mode (SPM, CSM, CM, CSCM):

SPM

Gain (high low):

high

Active Counters (number):

2

Thresholds (in keV):

0 0

DACs:

000 000 256 256 256 256 256 256 100 020 140 150 200 038 121 128 128 050 255 085 128 139 128 255 256

bpc File:

C:\Data\Example\test.bpc

Flat Field File:

C:\Data\Example\test.ffc

Dead Time File:

C:\Data\Example\test.dtc

Acquisition Type (normal, th\_scan, config\_readback):

normal

Frames in Acquisition (number):

1000

Trigger Start (positive, negative, internal):

positive

Trigger Stop (positive, negative, internal):

internal

Trigger Burst (number):

10

Time and Date Stamp (year month day hour minute second millisecond):

2011 12 01 09 16 49 003

Sensor Bias (if available V uA):

120 0.9

Sensor Polarity (positive, negative):

positive

Temperature (number in degrees C):

24.45

Medipix Clock (number in MHz):

120

Readout System Identifier:

Merlin 1

Software Version:

0.0.0