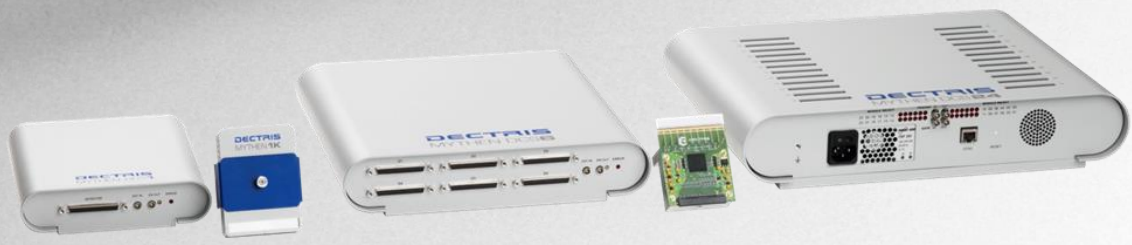
DECTRIS®

detecting the future

Socket Interface Specification

MYTHEN Detector System

Version: 3.0.0
Document Version: 2



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

Table of Contents

1. Document History

1.1. Current document

Version	Date	status	prepared	checked	released
3.0.0 DV2	04.02.2016	ready for release	DJ	**AM**	**PT**

1.2. Changes

Version	Date	Changes
3.0.0 DV2	04.02.2016	New postal address
3.0.0 DV1.1	11.06.2013	Adapted CD
3.0.0 DV1.0	21.09.2012	Simplified operation of the detector system Better data quality due to optimized calibration Restricted threshold and energy ranges By default all commands affect all active modules

2. How to use this guide

Before you start to operate the MYTHEN detector system please read the User Manual and the Technical Documentation included in the documentation package carefully.

This document has been designed for the MYTHEN detector systems.

2.1. Address and Support

DECTRIS Ltd.
Taefernweg 1
5405 Baden-Daettwil
Switzerland
Phone: +41 56 500 21 02
Fax: + 41 56 500 21 01

Website:

- www.dectris.com → support → Technical Notes → MYTHEN
- www.dectris.com → support → FAQ
- www.dectris.com → support → Problem Report

Email:

- support@dectris.com

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

Do not ship the system back before you receive the necessary transport and shipping information!

2.2. Explanation of Symbols

Symbol	Description
	Important or helpful notice
	Caution. Please follow the instructions carefully to prevent equipment damage or personal injury.
	DC-current
	AC-current
	Ground
	Functional earth

2.3. Explanation of Terms

Term	Description
GUI	**G**raphical **U**ser **I**nterface
DCS1	**D**etector **C**ontrol **S**ystem for **1** detector module
DCS6	**D**etector **C**ontrol **S**ystem for up to **6** detector modules
DCS24	**D**etector **C**ontrol **S**ystem for up to **24** detector modules
Detector module	The smallest fully functional unit of the detector (1280 channels).
MCB	**M**odule **C**ontrol **B**oard
FIFO	Storage working according to the **F**irst **I**n - **F**irst **O**ut principle
keV	**K**ilo **e**lectron **V**olt

2.4. Disclaimer

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

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

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

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

3. Warnings

Please read these warnings before operating the detector system.

- DO NOT TOUCH THE ENTRANCE WINDOW OF THE DETECTOR.
- Place the protective cover on the entrance window of the detector when it is not in use.
- The detector is not specified to withstand direct beam at a synchrotron. Such exposure will damage the exposed channels.
- The detector system should have enough space for proper ventilation. Operating the detector outside the specified ambient conditions could damage the system.
- The air inlets and outlets of the detector control system should not be blocked.
- Power down the detector system before connecting or disconnecting any cable.
- Before connecting the power supply to the mains-supply, check the supply voltage with the label on the power supply. Using an improper main voltage will destroy the power supply and could damage the detector.
- Make sure the cables are connected and properly secured.
- Avoid pressure or tension on the cables.
- Opening the detector, the detector control system or the power supply housing without explicit instructions from DECTRIS Ltd. will void the warranty.
- The embedded Linux operating system on the detector control system has customized software for controlling the MYTHEN detector system. Do not make any changes to the Linux operating system without explicit instructions from DECTRIS Ltd.

4. Introduction

The detector control system (DCS) hosts an embedded Linux system, on which the MYTHEN Socket Server is running. The DCS also stores the calibration files for the detector as well as some network settings. A client can open a socket connection to the server and control the MYTHEN detector by sending the commands listed in Chapter 7. Example applications in C++ for Linux and Windows are provided on the system memory stick. A *spec* implementation based on this socket interface is available from http://www.certif.com.

To take advantage of new functionality and bug fixes, it is possible to update to a newer version of the MYTHEN Socket Server within the same major release. The configuration files and the network settings stored on the DCS will not be affected by the update. The required files can be downloaded from the DECTRIS Ltd. website http://www.dectris.com.

An upgrade to a new major release usually requires a recalibration of the detector by DECTRIS Ltd. and is therefore not possible to be performed by the customer. If you are interested in a major upgrade please contact sales@dectris.com.

5. Operation

5.1. Connecting to the server

As a first step, the client has to open a socket connection to the server. The user can choose whether he wants to use UDP or TCP as protocol. While TCP is more reliable, UDP is faster and allows for higher frame rates.

To open the connection, the user has to specify the IP of the DCS, which is by default 192.168.0.90. The port of the socket server is 1030 for the UDP and 1031 for the TCP protocol.

The commands sent to the server are ASCII strings. The response is a byte array, which has to be interpreted as string, integer array or float array, depending on the command. All commands return responses of well defined lengths. Since the response might be fragmented on its way through the network, the client might have to read several packages, until the expected response length is reached.

In case of an error, the server will respond with a negative number. The meaning of the error codes can be found in section 7.2.

5.2. Initialization

First of all it should be checked, that the socket server has the expected version. This is done with help of the command "-get version".

Afterwards the detector system should be initialized by invoking the "-reset" command, which takes up to two seconds per module. This command loads settings for Cu X-rays and enables the flatfield correction and the interpolation for bad channels (more details on the default values after initialization can be found in section 7).

For proper operation all delivered modules have to be connected to the DCS. Nonetheless it is possible to read out only the first few modules. The number of active modules to be used can be specified using the “-nmodules” command. The MYTHEN system can also be operated with more than the delivered modules if the corresponding calibration files are available on the DCS. When increasing the number of active modules, all modules are set back to default settings.

To check, whether the communication with all modules is working, the module serial numbers can be read out ("-get modnum").

By default the commands affect all active modules. Some commands can be invoked on single modules. This can be achieved by selecting a specific module with the “-module” command. The same command can also be used, to restore the default behavior by selecting all active modules.

5.3. Settings

5.3.1. Predefined Settings

The simplest way to operate the detector is to load predefined settings for a copper or molybdenum tube with the “-settings" command. Depending on the calibration of your system, predefined settings for chromium and silver X-rays are also available.

5.3.2. Setting Threshold and Energy

The predefined settings can be customized by explicitly specifying the energy threshold and the X-ray energy with the command “-kthreshenergy”. The allowed ranges for these parameters depend on the calibration of your system.

5.3.3. High frame rates

To maximize the frame rate, the user should consider the following options:

1. Minimizing the number of bits which are read out with the "-nbits" command.
2. Disabling the server side rate correction with the command "-ratecorrection 0".
3. Disabling the server side flatfield correction with the command “-flatfieldcorrection 0".
4. Using a UDP connection instead of the slower TCP connection.
5. Using the “-readoutraw” command instead of the slower “-readout” command.

5.4. Acquisition control

Before starting an acquisition, the number of frames and the duration of the frames have to be programmed. The former can be done with the "-frames" command, the latter with the "-time" command. Since the duration is specified in units of 100ns, the argument can be a large number. Therefore the user should make sure to handle these numbers correctly. For example in the C programming language, it will be necessary to use the data type *long long*, which comprises 64 bits.

An acquisition is started with the "-start" command. Afterwards each frame is read out with a "-readout" command. A readout command that is sent while the corresponding frame is still acquired will return when the data collection for this frame has finished. Since the DCS can only buffer four frames, it is important to read out the frames as soon as possible. An acquisition can be interrupted with the "-stop" command, which results in the loss of the frames, which have not yet been read out.

The status of the acquisition can be checked with the "-status" command. Sending commands other than "-readout", "-status" or "-stop" during an acquisition can result in misbehavior of the detector.

5.5. Data corrections

For channels marked as bad the server will replace the number of counts with the average number of counts of the next lower and the next upper working channels. With the "-badchannelinterpolation" command the user can turn off this correction. In this case the number of counts will equal to -2 for these channels.

The server automatically applies a flat-field correction to the data. This behavior can be disabled with the "-flatfieldcorrection" command.

For high incoming X-ray fluxes, the rate correction can be turned on with the "-ratecorrection" command.

5.6. Triggering and gating

The standard and the continuous trigger modes can be enabled by the "-trigen" and "-conttrigen" commands. The timing of the measurements can be adjusted with the "-delbef" and "-delafter" commands.

The gating mode is turned on by the "-gateen" command. The number of gates can be programmed with the "-gates" command.

The polarity of the input and output signals can be defined with the help of the "-inpol" and "-outpol" commands.

5.7. Debugging

After sending the command "-log start" the socket server will log its activities into a file on the DCS until a "-log stop" command is sent. The content of the file is sent to the user in response to the "-log read" command. Since there is only limited disk and memory space on the DCS, the logging functionality should only be turned on for short periods of time.

For debugging problems related to the transmission of the data packages over the socket, the freely available network protocol analyzer WireShark can be used.

6. Example Applications

As a starting point a Unix and a Windows C application with some comments are provided on the system memory stick. They set up a socket connection to the DCS and read back the module serial numbers. Afterwards a short acquisition is performed, and the result is printed on the command prompt. The Windows version was developed with the freely available Microsoft Visual C++ 2008 Express Edition.

7. Commands

7.1. Command list

Synopsis: Specifies the format of the command. Values in angle brackets have to be replaced by the user: *b* stands for a boolean (*i.e*. a 0 or 1), *n* for an integer, *f* for a float, *time* for a 64-bit integer (*long long* or int64_t, used for the timing commands).

Arguments: Gives the meaning of the arguments. Default values after initialisation are set in *italic* type.

Description: Describes the effect of the command.

Return Type: Specifies the data type of the returned byte array. If no return type is indicated, the command returns an integer: a zero indicating successful execution of the command or a negative number in case of an error (see section Error Codes). If the return value is of type float, the error code is returned as a float number, and as an integer otherwise.

Return Size: Specifies the number of data elements in the returned byte array. For example a response of return type *int* and return size 1280 is 5120 bytes long (since one integer uses 4 bytes). N_{MOD} stands for the number of active modules (set by "-nmodules").

7.1.1. General commands

Table 1 General Commands.

Synopsis	Argument	Description	Return Type	Return Size
-get assemblydate		Returns the assembly date of the system as character array.	char	50
-get badchannels		Returns the status of each channel. A value of 0 means that the channel is working, a value of 1 means that it is defective.	int	N_{MOD}*1280
-get commandid		Returns the current command identifier. The command identifier increases by one whenever a client sends a command to the server. Can be used to make sure that no other clients changed the state of the system.	int	1
-get modnum		Returns the serial numbers of the modules, which have to be interpreted as hex-numbers.	int	N_{MOD} *
-get module		Returns either the position of the selected module or 65535 when all modules are selected.	int	1
-get nmaxmodules		Returns the maximal number of modules which can be connected to this MYTHEN system.	int	1
-get nmodules		Returns the number of active modules.	int	1
-get sensormaterial		Returns an identifier for the sensor material. 0: Silicon	int	1
-get sensorthickness		Returns the nominal sensor thickness in µm.	int	1
-get systemnum		Returns the serial number of the MYTHEN system.	int	1
-get version		Returns the software version of the socket server, which is of the form "M3.0.0". The returned string is terminated by a Null character.	char	7
-module <n>	Module position $0 \le n < N_{MOD}$ | *65535*	Selects the module at the given position as the target of module specific commands. An argument of 65535 selects all active modules. After initialization all modules are selected.		
-nmodules <n>	Number of modules	Sets the number of active modules (N_{MOD}). When the number of active modules is increased, all modules are set back to default settings. After		

	initialization the number of active modules is set to the number of delivered modules.
-reset	Sets the detector back to default settings. This command takes about two seconds per module.

7.1.2. Acquisition control

Table 2 Acquisition control commands.

Synopsis	Argument	Description	Return Type	Return Size
-delafter <time>	Delay in units of 100 ns. Default is 0 ns.	Sets the delay between two subsequent frames (“delay after frame”).		
-frames <n>	Number of frames. Default is *1*.	Sets the number of frames within an acquisition.		
-get delafter		Returns the delay between two subsequent frames.	long long	1
-get frames		Returns the number of frames.	int	1
-get nbits		Returns the number of bits to be read out.	int	1
-get status		Returns the status of the DCS as a bit pattern: Bit 0: Acquisition running Bit 3: Waiting for trigger Bit 16: No data available for readout (empty buffer)	int	1
-get time		Returns the programmed exposure time of one frame in units of 100ns.	long long	1
-nbits <n>	n = 4 | 8 | 16 | *24*	Sets the number of bits to be read out, thereby determining the dynamic range and the maximum frame rate. After initialization 24 bits are read out.		
-readout		The DCS can internally buffer the last four acquired frames. This command returns the data of the oldest frame in the buffer (*i.e*. the frames are returned in the same order as they were acquired). If the buffer is empty and there is an ongoing measurement, the command returns the data after the measurement has finished. If the readout fails for some reason, all count values are set to -1.	int	N_{MOD}*1280

<table>
<tr><td>-readoutraw</td><td></td><td>Returns the data of the oldest frame in the buffer in a raw, compressed format. The DCS does not apply any correction to the data, thereby allowing for maximal frame rate. The example code in section 0 shows how to decode the raw data.</td><td>int</td><td>N_B*N_{MOD}
<table><tr><th>N_{BITS}</th><th>N_B</th></tr><tr><td>24</td><td>1280</td></tr><tr><td>16</td><td>640</td></tr><tr><td>8</td><td>320</td></tr><tr><td>4</td><td>160</td></tr></table></td></tr>
<tr><td>-start</td><td></td><td>Starts an acquisition with the programmed number of frames or gates.</td><td></td><td></td></tr>
<tr><td>-stop</td><td></td><td>Stops the current acquisition. The data of the ongoing frame is discarded.</td><td></td><td></td></tr>
<tr><td>-time <time></td><td>Exposure time in units of 100ns. Default is 1s.</td><td>Sets the exposure time of one frame.</td><td></td><td></td></tr>
</table>

7.1.3. Detector settings

Table 3 Detector setting controlling commands.

Synopsis	Argument	Description	Return Type	Return Size
-energy <f>	$E_{min} \leq f \leq E_{max}$ in units of keV. Default is *8.05* keV.	Sets the X-ray energy for the selected modules while keeping the current threshold energy. The energy is used to provide an optimal flatfield correction. The supported energy range depends on the calibration of the system. This command takes about two seconds per module.		
-get energy		Returns the X-ray energies of the selected modules.	float	N_{MOD}*
-get energymax		Returns the maximal supported X-ray energy in units of keV.	float	1
-get energymin		Returns the minimal supported X-ray energy in units of keV.	float	1
-get kthresh		Returns the threshold of the modules in keV.	float	N_{MOD}*
-get kthreshmax		Returns the maximal supported threshold energy in units of keV.	float	1
-get kthreshmin		Returns the minimal supported threshold energy in units of keV.	float	1
-kthresh <f>	$Thr_{min} \leq f \leq Thr_{max}$ in units of keV. Default is *6.4* keV.	Sets the energy threshold for the selected modules while keeping the current X-ray energy. The supported threshold range depends on the calibration of the system. This command takes about two seconds per module.		
-kthreshenergy <f1> <f2>	$Thr_{min} \leq f1 \leq Thr_{max}$, $E_{min} \leq f2 \leq E_{max}$ in units of keV. Defaults are f1 = *6.4* keV and f2 = *8.05* keV.	Sets the energy threshold and the X-ray energy for the selected modules. The supported ranges depend on the calibration of the system. This command takes about two seconds per module.		

<table>
<tr>
<td>-settings <name></td>
<td>name = Cu | Mo | Cr | Ag</td>
<td>Loads predefined settings for the selected modules and the specified X-ray radiation. The energy threshold is set to a suitable value. The availability of Cr and Ag settings depends on the calibration of your system. This command takes about two seconds per module. After initialization settings for Cu X-rays are loaded.

The following table shows, how this commands relates to –kthreshenergy

<table>
<tr><th></th><th>is equivalent to</th></tr>
<tr><td>-settings Cr</td><td>-kthreshenergy 4.8 5.41</td></tr>
<tr><td>-settings Cu</td><td>-kthreshenergy 6.4 8.05</td></tr>
<tr><td>-settings Mo</td><td>-kthreshenergy 8.74 17.48</td></tr>
<tr><td>-settings Ag</td><td>-kthreshenergy 11.08 22.16</td></tr>
</table>
</td>
</tr>
</table>

7.1.4. Data correction

Table 4 Data correction commands.

Synopsis	Argument	Description	Return Type	Return Size
-badchannelinterpolation <b>	0: Disable *1: Enable*	Turns on or off the interpolation routine for bad channels. When enabled, the number of counts for a bad channel is set to the average number of counts of the next lower and the next upper working channels. If there is no next lower (upper) working channel, the number of counts of the next upper (lower) working channel is used. When disabled the number of counts for a bad channel is set to -2.		
-flatfieldcorrection <b>	0: Disable *1: Enable*	Enables or disables the flatfield correction. After initialization the flatfield correction is enabled.		
-get badchannelinterpolation		Returns whether the bad channel interpolation is enabled (1) or not (0).	int	1
-get cutoff		Returns the maximal count value before flatfield correction.	int	1
-get flatfield		Returns the currently loaded flatfield calibration of all modules.	int	N_{MOD}*1280
-get flatfieldcorrection		Returns whether the flatfield correction is enabled (1) or not (0).	int	1
-get ratecorrection		Returns whether the rate correction is enabled (1) or not (0).	int	1
-get tau		Returns the dead time constant for all active modules in units of ns.	float	N_{MOD}*
-ratecorrection <b>	*0: Disable* 1: Enable	Enables or disables the rate correction. After initialization the rate correction is disabled.		
-tau <f>	f > 0 or f = *-1.0* in units of ns	Sets the dead time constant (used by the rate correction) for the selected modules. If the argument is equal to -1.0 the system uses predefined values appropriate for the loaded settings.		

7.1.5. Trigger / Gate

Table 5 Trigger and Gating control commands.

Synopsis	Argument	Description	Return Type	Return Size
-conttrigen <b>	*0: Disable* 1: Enable	Enables or disables the continuous trigger mode. In this mode, each frame requires a trigger signal.		
-delbef <time>	Delay in units of 100ns. Default is *0ns.*	Sets the delay between a trigger signal and the start of the measurement (“delay before frame”).		
-gateen <b>	*0: Disable* 1: Enable	Enables or disables the gated measurement mode.		
-gates <n>	Number of gates. Default is *1.*	Sets the number of gate signals within one frame.		
-get conttrig		Returns whether the continuous trigger mode is enabled (1) or not (0).	int	1
-get delbef		Returns the delay between a trigger signal and the start of the measurement.	long long	1
-get gate		Returns whether the gated mode is enabled (1) or not (0).	int	1
-get gates		Returns the number of gates.	int	1
-get inpol		Returns the input polarity for the trigger and gate signals. 0: rising edge / active high, 1: falling edge / active low.	int	1
-get outpol		Returns the polarity of the enable out signal. 0: active high, 1: active low	int	1
-get trig		Returns whether the trigger mode is enabled (1) or not (0).	int	1
-inpol <b>	*0: rising edge / active high* 1: falling edge / active low	Sets the input polarity for trigger (rising or falling edge) and gate signals (active high or low).		
-outpol <b>	*0: active high* 1: active low	Sets the polarity of the enable out signal.		

-trigen <b>	*0: Disable* 1: Enable	Enables or disables the trigger mode. The acquisition starts after the trigger signal. Subsequent frames are started automatically.

7.1.6. Debugging

Table 6 Debugging commands.

Synopsis	Argument	Description	Return Type	Return Size
-log start		Instructs the socket server to log its activities into a file on the DCS. Do not enable logging except for debugging purposes, since it will slow down the server and because there is only limited space for the log file on the DCS.		
-log stop		Stops the logging functionality of the socket server. The command returns the size of the log file.	int	1
-log read		Instructs the socket server to send back the content of the log file. The length of the server response has to be inferred from the return value of the "-log stop" command.	char	
-testpattern		Returns a test data set with the number of counts for each channel equal to the channel number. Can be used to verify the client implementation of the readout mechanism.	int	N_{MOD}*1280

*In the legacy or legacy2 mode, the return size is 1 instead of N_{MOD}.
[1]In the legacy mode the first frame is read out automatically if the acquisition time is below the automatic readout time. In this case the return type and size are as stated for the readout command.
[2]For a bad edge channel, the return value is twice the value of its neighbor minus the value of its neighbor after next

7.1.7. Deprecated commands

Table 7 Depreceated commands.

Synopsis	Argument	Description	Return Type	Return Size
-autoreadouttime <time>	time > 0 or time = -1 in units of 100ns	Sets the automatic readout time. If the acquisition time is below the automatic readout time, the first frame is read out automatically after the "-start" command. The automatic readout mechanism is disabled if the argument is equal to -1.		
-autosettings <f>	f > 0 in units of keV	Sets the threshold to the specified value and loads suitable settings for the selected modules. This command is identical to the “-kthresh” command and should no longer be used.		
-flipchannels <b>	*0: Disable* 1: Enable	Enables or disables the flipping of the channel numbering. In versions older than 2.0 the numbering was flipped compared to the MythenGUI. After initialization the flipping of the channels is turned off.		
-get autoreadouttime		Returns the programmed automatic readout time in units of 100ns.	long long	1
-get flipchannels		Returns whether the channel flipping is enabled (1) or not (0).	int	1
-get settings		Returns the settings of the modules 0: Standard, 1: Highgain, 2: Fast, 3: Unknown	int	N_{MOD}^{*}
-get settingsmode		Returns the last loaded settings. The result is of the form "auto <thr> <e>" (with <thr> being the threshold and <e> the energy) or “manual” if not all modules have the same settings. In legacy2 mode the result string is of the form "predefined StdCu or "auto <thr>.	char	50

-legacy		Enables the legacy mode for backwards compatibility with socket servers below version 2. This means that the auto readout time is set to 2s, flatfield correction is turned off and channel flipping is turned on. The effects of the legacy2 mode are also active.		
-legacy2		Enables the legacy mode for backwards compatibility with socket servers of major version 2. In this mode • the possibility to select all modules is disabled • the return size of some commands is 1 instead of N_{MOD} • the “-get settingsmode” command works as before		
-read		Deprecated version of the "-get flatfield" command.	int	N_{MOD}*1280
-settings <name>	name = *StdCu* | StdMo | HgCr | HgCu | FastCu | FastMo	Deprecated arguments for the “-settings” command. Settings for the corresponding X-ray energies are loaded. Appropriate gain values are automatically selected by the server.		
-trimfile <name>	name = TrimCr | TrimCu | TrimMo	Deprecated version of the "-settings" command. The arguments TrimCr, TrimCu, TrimMo corresponde to the arguments HgCr, StdCu, StdMo of the "-settings" command.		

7.2. Error Codes

If the response of the server consists of a single, negative number, an error has occured during the execution of the command. The following table lists the corresponding error codes.

Table 8 List of error codes.

Error code	Description
-1	Unknown command
-2	Invalid argument
-3	Unknown settings
-4	Out of memory
-5	Module calibration files not found
-6	Readout failed
-10	Flatfield file not found
-11	Bad channel file not found
-12	Energy calibration file not found
-13	Noise file not found
-14	Trimbit file not found
-15	Invalid format of the flatfield file
-16	Invalid format of the bad channel file
-17	Invalid format of the energy calibration file
-18	Invalid format of the noise file
-19	Invalid format of the trimbit file
-20	Version file not found
-21	Invalid format of the version file
-22	Gain calibration file not found
-23	Invalid format of the gain calibration file
-24	Dead time file not found
-25	Invalid format of the dead time file
-30	Could not create log file
-31	Could not close log file
-32	Could not read log file

7.3. Decoding the raw data

The example code below shows how to decode the compressed raw data returned by the "-readoutraw" command. The usage of this command is only necessary, when very high frame rates (> 300 Hz) are required. All corrections (flatfield-, rate-, bad channel-corrections) have to be applied afterwards by the client.

```
// Input
// nmods: number of active modules
// nbits: number of bits, which were read out
// data: response of the -readoutraw command
//
// Output
// result: array of size 1280*nmods with the number of counts of all
//         channels

void decodeRawReadout(int nmods, int nbits, int *data, int *result)
{
  int chanperline = 1;     // default for 24 bits
  int mask=0xffffff;       // default for 24 bits
  if (nbits == 16)
  {
    chanperline = 2;
    mask=0xffff;
  }
  if (nbits == 8)
  {
    chanperline = 4;
    mask=0xff;
  }
  if (nbits == 4)
  {
    chanperline = 8;
    mask=0xf;
  }

  int size = 1280/chanperline*nmods;
  u_int32_t tmpArray[size];                  // the data has to interpreted
  memcpy(tmpArray, data, size*sizeof(int));  // unsigned int32

  for (int j = 0; j < chanperline; j++)
  {
    int shift = nbits*j;
    int shiftedMask = mask<<shift;
    for (int i = 0; i < size; i++)
    {
       result[i*chanperline+j]=((tmpArray[i]&shiftedMask)>>shift)&mask;
    }
  }
}
```

8. Release Notes

8.1. Version 3.0.0

The main objective of the new release is a substantially improved detector calibration in parallel with a considerably simplified detector usage. The user has no longer to worry about choosing suitable gain settings, this task is now entirely handled by the detector. Depending on the energy threshold, the server continuously adjusts the gain settings to ensure noiseless measurements with the smallest possible detector dead-time.

8.1.1. Modifications

- Predefined settings are loaded by simply specifying the X-ray energy in the form “-settings Cu”. No gain has to be specified.
- For optimal flatfield correction, the server has to be informed about the X-ray energy. It is recommended to either use predefined settings or to set the energy threshold and the energy with help of the new “-kthreshenergy” command. The energy can also be set with the new command “-energy”.
- Improved precision of the rate correction.
- Many new commands to retrieve the state of the system: “-get conttrig”, “-get cutoff”, “-get delafter”, “-get delbef”, “ -get frames”, “-get gate”, “-get gates”, “-get inpol”, “-get module“ , “-get nmaxmodules”, “-get outpol”, “-get trig”.
- When increasing the number of active modules with the “-nmodules” command, all modules are set back to default settings.
- Multi-module systems can be controlled without looping over individual modules:
  - Commands retrieving module specific information return by default the values for all modules.
  - Commands setting module specific parameters affect by default all modules.
- Instead of additional gain settings, extended threshold ranges can be requested by the customer. Therefore the commands to set the energy or threshold only accept arguments within the supported range. The upper and lower limits can be retrieved by “-get energymin”, “-get energymax”, “-get kthreshmin”, “-get kthreshmax”.
- New command “-get commandid” to retrieve a command identifier. This can be used to make sure that no other clients changed the state of the system.
- New command “-testpattern” for debugging purposes.
- Legacy2 mode for backwards compatibility with version 2.

8.1.2. Migration notes

If you migrate from version 2 take care of the following changes:

- For optimal flatfield correction, the X-ray energy has to be specified.
- After initialization all modules are selected
- Commands retrieving module specific information return the values for all values.
- The “-kthresh” commands only accepts arguments within the range supported by the specific MYTHEN system.
- Since the gain automatically changes with the threshold energy, the number of bad channels can change with the specified threshold energy.
- The “-autosettings” command is deprecated. Please use “-kthreshenergy” instead.
- The commands “-get settings” and “-get settingsmode” are deprecated.
- The arguments “StdCu”, “HgCr”, … of the “-settings”command are deprecated.

8.1.3. Legacy2 mode

If for some reasons you are not going to migrate, you can enable the Legacy2 mode with the "-legacy2" command. After having sent this command, the interface to the server is identical with version 2 and your software will still run. Although the interface is the same, keep in mind that the calibration of the detector will be different than for a system delivered with version 2. For example you will not see any differences in data acquired with “StdCu” and “FastCu” settings, because these deprecated arguments are internally mapped to the same detector settings. Furthermore there are upper and lower limits on the allowed threshold and energy ranges.

8.2. Version 2.1.0

- The bad channel interpolation now supports neighboring bad channels. In this case the interpolated value is equal to the average number of counts of the next lower and the next upper working channels. If there is no next lower (upper) working channel, the number of counts of the next upper (lower) working channel is used.
- Different modules can now have different dead time constants for the rate correction. This fixes the first known issue of version 2.0.5.
- Increasing the number of active modules now works flawlessly, fixing the second known issue of version 2.0.5.
- The rate correction is now applied before the flatfield correction.

8.3. Version 2.0.5

New command to get the assembly date of the system („-get assemblydate“).

8.4. Version 2.0.4

New commands (“-get sensorthickness”, “-get sensormaterial”, “-get systemnum”) to retrieve information about the sensor and the system. If the information can not be found, one of the new error codes -20 and -21 is returned. If you upgrade an older system to version 2.0.4, the sensor information will not be available.

Several improvements and bug fixes for multi-module systems (especially for the MYTHEN 24K system) have been implemented:

- Faster execution of the reset command.
- After execution of the reset command the first module (module 0) is selected (instead of the last one).
- Correct sending of data if its length is a multiple of the UDP/TCP packet size.
- The initialization sets the number of modules to the number of delivered modules. For systems which are delivered with less modules than the DCS maximally supports (*e.g*. a MYTHEN 3K with 3 modules on a DCS6), the user no longer needs to specify the number of modules. The command to set the number of active modules is only necessary for running a system with less (or more) than the number of delivered modules. In contrast to earlier versions (2.0.2 and 2.0.3), the corresponding command has to be executed after the reset command, since a reset now sets the number of modules to the number of delivered modules.

8.5. Version 2.0.3

The server is now able to handle optimized high gain settings. This change is purely internal and has no consequences for the user.

8.6. Version 2.0.2

The reset command does not change the number of active modules any more. This prevents the issuing of error messages by the reset command when working with a not fully equipped multi-module system.

8.7. Version 2.0.1

Two bugs in version 2.0.0 have been fixed:

- Due to a type conversion problem, the server-side flatfield correction truncated the correction factors to integers. As a consequence many channels returned zero counts in all readouts. The problem only occurred when the server-side flatfield correction was turned on.
- The result of the” -get badchannels” did not have the format as described in this specification.

8.8. Version 2.0.0

Version 2.0.0 is a complete redesign of the socket server and the user is encouraged to adapt his older implementation to the new version, since it offers better stability and more functionality.

8.8.1. New Functionalities

- The server can be controlled over a TCP socket
- Support for up to 24 modules
- Fast settings are available to the user
- Built-in support for flat-field correction
- Built-in support for rate correction
- Automatic loading of optimal settings based on the required threshold
- Many new commands to read out the current settings and parameters
- Logging functionality

8.8.2. Migration notes

If you migrate from version 1.3.0, take care of the following changes:

- By default the automatic readout of the first frame in an acquisition is disabled.
- The flipping of the channels is turned off after initializtion.
- By default the flat-field correction is performed directly on the DCS.
- The "-read" command is deprecated and should be replaced by the "-get flatfield" command.
- The "-trimfile" command is deprecated and should be replaced by the "-settings" command.

8.8.3. Legacy Mode

If for some reasons you are not going to migrate, you can enable the legacy mode with the "-legacy" command. After having sent this command, the server should behave identically to version 1.3.0 up to bug fixes and the more detailed error responses.

9. Appendix

9.1. Table of Tables

9.2. Referenced Documents

All the following documents are available through DECTRIS Ltd. homepage.

https://www.dectris.com/technical_mythen.html#main_head_navigation

Term	Description
Firmware Update Guide	Firmware_Update-MYTHEN-V1_6.pdf
Module Handling Instructions	Handling_Instructions-MYTHEN-V2_1.pdf
Linux Installation Guide	Linux_Installation-HowTo_MYTHEN-V1_3.pdf
Network Settings	Network_Settings-MYTHEN-V1_3.pdf
Quick Start Guide	QuickStart-MYTHEN-V2_1.pdf
Socket Interface Specifications	Socket_Interface_Spec-MYTHEN-V3.0.0-V1_1.pdf
Technical Documentation	Technical_Documentation-MYTHEN-V1_1.pdf
Trigger Notes	Trigger_Note-MYTHEN-V2_1.pdf
User Manual	User_Manual-MYTHEN-V2_1.pdf