MAGIC-TDAS 04-01
040520/FGoebel

Users Guide and Reference Manual for the *Data Acquisition* and the *FADC* system of the MAGIC telescope

F. Goebel
<fgoebel@mppmu.mpg.de>

July 17, 2005

Abstract

This document describes the operation of the data acquisition (DAQ) and the readout electronics (FADCs and receiver boards) of the MAGIC telescope. It is divided in 2 parts: The **Users Guide** section should be read by the shift crew before any data taking. It describes the steps that have to be followed before and after data taking. It also contains information on how to handle the most frequent problems that may occur during data taking. The **Reference Manual** is a more complete description of the functionality of the DAQ and the readout electronics. This is primarily meant as a reference for experts.

Contents

1 Users Guide

1.1 Operation during normal conditions

1.1.1 Procedure before Start of Data Taking

To start the DAQ program log in as `operator` on pc2 and type `start_readout`. It is advisable to start the DAQ remotely from a console in the control room, because you should have a look at the DAQ program from time to time (see section:1.2.5). You will get a prompt `magic_readout>` which tells you that the DAQ is running. You can type in some commands. However, under normal conditions the data taking should be handled only from the central control (Arehucas).

Before you can take data you have to turn on the 4 FADC racks and the receiver board rack in the electronics room. The switches are located on the wall behind the racks. The FADC racks are powered by four 220 V plugs with yellow switches. The receiver board rack needs 380 V. You have to switch on the big red switch on the wall.

In order to avoid overheating of the electronics, switch on the airconditioning and the fan for the extraction of the hot air. The airconditioning and the fan are turned on by connecting the 380 V plugs on the wall behind the FADC racks. Currently, only one of the aircontitionings (the left one) can be switched on. The second 380 V plug is used to connect the powerful fan. The 380 V power supply cable for the fan is hanging from the ceiling of the electronics room. Please connect it to the closest 380 V plug.

The airconditioning of the electronics room will only work if all the doors to the electronics room are closed. Therefore, **please keep all doors to the electronics room closed while the electronics and the airconditioning are on!**

1.1.2 Procedure after Data Taking

The readout electronics consumes a lot of power and should therefore be switched off whenever it is not needed. Please switch off the 4 FADC racks, the receiver board rack, the fan and the airconditioning.

It is advisable, though not mandatory, to stop the DAQ program. Just type `QUIT` inside the DAQ program (after the `magic_readout>` prompt).

1.2 Trouble Shouting

Below you find some steps you may follow if you encounter certain kinds of errors while operating the MAGIC data acquisition chain. If this does not help contact the experts:

Florian Goebel	`fgoebel@mppmu.mpg.de`	Tel: +49 89 32354 445, +49 89 41310729
Thomas Schweizer	`tschweiz@physik.hu-berlin.de`	Tel: +49 30 2093 7797

In any case notify the above experts of any problems you have encountered and the measures you have taken.

1.2.1 The electronics racks can not be switched on or went off by themselves

You can tell that the FADC power is off by looking at the green LEDs on the FADC fans (they should be on when the FADC rack is on).

If the power is unexpectedly off, most likely some fuses have blown. If only some racks do not get power the fuses on the wall behind the electronics racks (left side) may have blown. In that case, switch off the affected racks, switch on the fuse and switch on the racks again.

If all racks are off, the fuse behind the big UPS in the garage may have blown. Open the right most (out of three) fuse boxes on the wall behind the UPS and check for blown fuses. Again, switch off the electronics before switching on the fuse.

The FADCs have an overheat protection, with relays powered by 220 V. Whenever there is an overheating or the 220 V are disconnected the relays will be released and the power supply of the FADCs is disconnected. Therefore, after a complete power cut, you have to switch on the relays again. Before doing so, switch off the racks. Then, press the red buttons at the backside of the FADC racks and finally you may switch on the racks.

If no fuses have blown and the overheat relays are out, the electronics might have overheated. Switch off the FADC rack and check that no serious damage has happened. Leave the FADC rack off for some minutes to cool it down. Then you may switch on the relays again, before switching on the plug on the wall for the FADC rack (make sure it was off when you push the red relays button).

1.2.2 The receiver boards beep

This is most likely the overheat alarm of the receiver boards. You may temporarily press the `Alarm Off` button to stop the beeping. If the overheat problem does not go away the beeping will start again after some time.

Please, stop data taking and switch off the electronics for some minutes. Check whether the electronics room is not sufficiently cooled (temperature should be below 20^o) and try to find the reason (doors closed, airconditioning and fans on?).

If you cannot find any obvious way how to cool down the temperature in the electronics room, you may open the backside of the receiver board rack, which helps to cool down the receiver electronics.

Switch the receiver boards on again after some minutes and restart data taking.

1.2.3 The DAQ crashed

Unfortunately, the DAQ program crashes from time to time (due to some problems with the driver for the MicroEnable card). Normally you will notice that the DAQ crashed because Arehucas suddenly tells you that the DAQ is unavailable.

When you got to the terminal where you started the DAQ (with `start_readout`) you will most likely see that the DAQ stopped. Just try to start it again (with `start_readout`). If that doesn't help or you realize that pc2 froze, you have to reboot pc2. In the worst case you might have to push the red reset button of this machine.

The DAQ computer pc2 is located inside the third (from the left) rack in the electronics room. It is controlled by the monitor and keyboard inside the rack, which are connected to the 4 port switch close to the monitor.

1.2.4 The trigger sends triggers but the DAQ rate is 0

The most likely reason is that the master FADC rack (rack A, 2^{nd} rack from the left) is off.

If this is not the case, try to stop the trigger and start it again. You may also stop the DAQ run and start a new one.

1.2.5 Continuous error messages on the DAQ terminal and DAQ aborts (and restarts) runs after 1 or 2 events

This is usually an indication that something is wrong with the hardware of the DAQ or the FADCs.

In the simplest case one or more of the FADC racks is off (not switched on or blown fuse). Try to switch them on.

If this is not the case the situation is more complicated. Probably, a cable or even an FADC board is faulty or broken. The diagnosis is not trivial. Try to reach the experts.

If the experts are not reachable you may try the following:

- If the error message (see 2.2.3) concerns basically only one board you may switch off the corresponding rack and pull out the affected board (white labels on top of FADC board) by a few cm before pushing it back in again and switching the electronics on again. You may also check the cables connecting the affected board.

- In case of Integrity Error 4 of one or very few modules, you may turn off the corresponding FADC rack, pull out the FADC board and take off the affected FADC module (see 2.2.3 of explanation of error messages). Put everything back together (careful, not to damage any cables) and hope that the error goes away. It sounds a bit like *black magic* but it usually helps. The reason is probably a bad electrical connection.

  If this does not help, you may also exchange the FADC module with a neighboring module. Only if this does not help either, exchange the module by a spare module (should be in the right gray cabinet in the electronics room).

- If you get many error messages affecting a whole FADC rack, check which is the first affected board. If this is the first board of a rack, check if the rack is switched on. If everything looks ok and the error does not go away, one of the power supplies may be broken. You can only check this by opening the back of the FADC rack. It should be sufficient to unscrew the lower board. You can measure the voltage with a voltmeter at the horizontal copper bars at the back of each FADC crate. The lower bar should be ground. Relative to the ground you should measure the following voltages: $+5V, -5.2V, +3.3V, -2.0V$. These are the nominal values. Deviations by 0.3V are still ok (actually usually the voltage is set a bit higher than the nominal values). If you find one voltage close to $0V$ the corresponding power supply (black boxes inside FADC rack) is probably broken. There should be spares in the left gray cabinet. Before replacing the power supplies contact the experts.

2 Reference Manual

2.1 Introduction: General Overview of the Data Acquisition Chain

The data acquisition chain of MAGIC records the data from the 577 PMTs in the camera and saves them to disk for further analysis.

The analog PMT signals are transmitted to the counting house via 162 m long optical fibers using Vertical Cavity Surface Emitting Laser drivers (VCSELs, $\lambda = 850$ nm). In the electronics room the fibers are connected to the back of the receiver board rack (8 fibers per receiver board).

In the **Receiver Boards** the signal is split. One branch goes to a discriminator that generates a signal for the trigger system. The discriminator threshold can be set individually for each pixel from the Central Control. The signal in the second branch is amplified, stretched to $\sim$6 nsec FWHM and split into a high and a low gain channel in order to increase the dynamic range. The high gain signal is further amplified by a factor 10 while the low gain signal is delayed by $\sim$55 nsec. If the signal is above a preset threshold the delayed low gain signal is combined with the high gain signal using a fast GaAs switch before it is sent to FADC.

The analog signals of the 8 channels in each receiver board are transmitted to the corresponding **FADC board** through the white 3M cables. The 8 bit FADCs continuously digitize the signal with 300 MHz. The digitized samples are stored in 32 kByte long ringbuffers. When a trigger signal arrives the pointer in the ringbuffer is set back buy a predefined value to compensate for the trigger latency and 30 slices are read out per pixel. If the receiver board switched to the low gain, the first approximately 15 slices correspond to the high gain signal and the remaining 15 correspond to the low gain signal. The 30 slices are written to a 512 kBytes long FiFo buffer.

The FADCs for the 577 pixels are contained in 4 racks with 5 crates. Each crate contains 4 motherboards equipped with 8 FADC cards each. The data is multiplexed and sent to one PC by a hierarchical structure of interface boards. The GPS clock time, the trigger and the calibration information for each event is recorded by dedicated digital modules (inside the last FADC board) which are read out together with the other FADC modules.

The complete readout is controlled by the DAQ computer (pc2) via the so called MicroEnable card, a PCI card with a FPGA (Xilinx) Chip. It reads out the FADC data stored in the FiFo buffers sequentially through the hierarchical structure of interface boards.

The **DAQ program** (`magic_readout`) is a multithreaded readout program written in C++. It reads out the MicroEnable card, reorganizes the data into raw data format and writes it over a GBit Ethernet connection to the RAID5 disk system on the fileserver (pc15).

2.2 DAQ

As stated in section: 1.1.1 the DAQ program can be started using the script `start_readout` by any user from any directory.

The DAQ computer pc2 is located inside the third (from the left) rack in the electronics room. It is controlled by the monitor and keyboard inside the rack, which are connected to the 4 port switch close to the monitor. The switch is also used for the trigger and the file server. The password for `daq` can be found under the keyboard of pc2.

The DAQ program itself is called `magic_readout` located in `daq@pc2:MagicReadout/`. The source can be found in `daq@pc2:MagicReadout/src/` (please don't touch it without explicit permission from the DAQ expert).

2.2.1 Commands

The DAQ program can be controlled from the Central Control or from the console directly. In both cases a certain number of commands can be sent. You can get a listing of the possible commands typing “`?`“ or “`HELP`”. The currently available commands are the following:

HELP or ? :
print listing of command options

STATE :
print the current status of the readout

CONF :
print the values of the configuration variables

DTEST :
switch to `daq_test` (take data but don't write it to disk)

DARUN `<numtriggers>` `<ProjectName>` `<SourceName>` `<SourceRaHour>` `<SourceRaMin>` `<SourceRaSec>` `<SourceDecDeg>` `<SourceDecMin>` `<SourceDecSec>` `<SourceEpochChar>` `<SourceEpochDate>`:
switch to `data_run` i.e. take cosmics data and write it to disk. `<numtriggers>` is the maximum number of events before switching automatically back to `standby` mode. If `<numtriggers>` is larger than the maximum number of events per file (see `MaxFileSize` in configuration file `magic_daq.config`) the DAQ will automatically start new runs until the total number of `<numtriggers>` events is reached. `<numtriggers>` = 0 stands for infinit number of events.

CARUN `<numtriggers>` ... same parameters as DARUN :
switch to `calibration_run` i.e. take calibration data and write it to disk

PERUN `<numtriggers>` ... same parameters as DARUN :
switch to `pedestal_run` i.e. take pedestal data and write it to disk

TERUN `<numtriggers>` :
same as DARUN filling the rest of the parameters like `<ProjectName>` etc with dummy variables

PREAD `<runnum>` :
read the pedestal values as obtained from (pedestal-)run `<runnum>` and use them for the calculation of the charges.

LOCK! :
Not used

ULOCK :
Not used

HALT! :
Stop reading data from FADCs. If not in `daq_test` mode the data in the Menable buffer is written to the run file before the run is closed.

STOP! :
Stop reading data from FADCs and close run immediadely. Data in the Menable will be lost.

QUIT! :
Exit the DAQ program `magic_readout`. Does only work if in `standby` mode

QUIT! force :
Exit the DAQ program `magic_readout` even is not in `standby` mode. Runs may not be closed propperly and data may be lost.

The interpretation of the commands is case sensitive i.e. they have to be typed in capital letters. It is mandatory to provide the required number of parameters. Else the command will be ignored and you get an error message accordingly.

2.2.2 Configuration Files

The behaviour of the DAQ program is steered by some configuration files. The DAQ needs to find these configuration files in the directory from where it is executed. In all configuration files lines starting with "`*`" are comments. Do not make any changes to the configuration files without explicit authorization from the experts since they affect the functionality of the DAQ.

magic_daq.config

This configuration file is the main configuration file containing all the fundamental parameters for the DAQ. The parameter name is followed by the parameter value. If a parameter is not set, default values are assumed (see: `run_descriptor.c` for default values). The parameters are explained in the following. Numbers in bracket indicate currently used values (not necessarily default values).

MuxStart (8) :
: First Muxer Number (possible values: 0..63) to be read in each Rack (there are 2 Muxers per board i.e. a rack fully equipped with 20 boards has 40 Muxers. The first Muxer Number is typically set to 8 the last Muxer Number is thus 47). `MuxEnd - MuxStart` should always be a muliple of 2 (always read complete boards). Make sure that the Master Board (currently muxer 8+9 in rack A) is being read out.

MuxEnd (47) :
: Last Muxer Number (possible values: 0..63) to be read in each Rack.

MuxEndLastRack (37) :
: The last Muxer Number in the last rack can be set independently (e.g. because the last rack is not fully equipped with boards). If `MuxEndLastRack` is not specified `MuxEnd` is taken.

RackStart (0) :
: Number of first rack (possible values: 0..3) to be read out. 0 stands for rack A (currently second rack from the left), 1 stands for rack B, etc.

RackEnd (3) :
: Last Rack Number (possible values: 0..3) to be read out.

MuxDefault (8) :
: Mux-Address while readout is not active. (very technical)

RackDefault (0) Rack-Address while readout is not active.

FadcBaseline (10) :
: The baseline of the FADCs. The mean pedestal value should then be close to `FadcBaseline` ADC counts.

FadcOffset (72) :
: The offset (in units of 6.6 nsec i.e. 2 samples) between the position in the ringbuffer when the trigger arrives and the position where the pulse is. Change this parameter to adjust the position of the pulse inside the time window of the FADC.

HfPolarity (1) :
: Polarity of the "HalfFull Flag" produced by the FiFo in the Master board. Currently the following settings are required:

1 : if AllmostEmptyFlag selected by switch inside Master Board
0 : if HalfFullFlag selected

LatticeClkFlag (1) :
Determines the clock speed on the Lattice chip on the FADC motherboards. It affects the dead time introduced by the readout of the ringbuffer. It also determines the clock speed of the internal ClockTickCounter which recorded in the raw data for each board.
0 : clk on Lattice = clk on mother board (currently: 18.75 MHz)
1 : clk on Lattice = clk on mother board / 2 (currently: 9.375 MHz)
2 : clk on Lattice = clk on mother board / 4 (currently: 4.6875 MHz)
3 : clk on Lattice = clk on mother board / 8 (currently: 2.34375 MHz)

CounterReset (1) :
0 : reset clocktick every second
1 : reset clocktick with Reset signal

SelectFifoClk (0) :
Selects the clock (coming form the MicroEnable card) to be used for the readout of the FiFos on the FADC mother boards.
0 : normal clock from Menable card
1 : use Sout0 signal as clock signal

PlxClk (30) :
Clock speed (in MHz) of PCI DMA transfer (from MicroEnable card to pc).

FpgaClk (30) :
Clock speed (in MHz) of FrontEnd readout (from FiFos on FADC boards to MicroEnable card).

GapBetweenMux (4) :
Number of clock cycles between the readout of consecutive Muxers.

LoGainNumSamples (15) :
Number of high gain Samples.

HiGainNumSamples (15) :
Number of low gain Samples

NumCheckSamples (6) :
Number of additional identically repeated FADC samples read out after the normal FADC samples. They are used to check the data consistency

NumDigiSamples (4) :
Number of additional bytes per event with digital information coming along with the FrontEnd data stream.

NumPixInMux (4) :
Number of channels per muxer

NumPixInBoard (8) :
Number of channels per board

NumPixInCamera (577) :
Number of Pixels in the camera

InvertData (1) :
If `InvertData != 0` the FADC data is inverted (inverted ADC = 255 - ADC).

ResetReadout (0) :
Determines the behaviour of the DAQ when the maximum number of events (determined by `MaxFileSize`) is reached.

0 : do not reset the FrontEnd readout
1 : reset the FrontEnd readout (FADCs)

UsePedRun (1) :
If `UsePedRun > 1` use result of the latest pedestal run for calculation of charges (which are e.g. sent to the central control).

WriteLastEvent (0) :
When the readout stops because of a data integrity error this flag decides whether the corrupted event will be written as the last event of the run or not.

0 : do not write corrupted event
1 : write corrupted event as last event of run

CheckTime (1) :

1 : Perform consistency checks between GPS time, FADCClockTick and PC time
0 : don't do time consistency checks

GpsClckTickNSecDiffTolerance (10000000) :
If the time difference between the GPS Clock and the FADCClockTick is larger than *GpsClckTickNSecDiffTolerance* nsec print out a warning message.

GpsPCSecDiffTolerance (2) :
If the GPS time of the first event and the PC time at start of a run differ by more than *GpsPCSecDiffTolerance* sec print out a warning message.

MaxFileSize (1000000000) :
Maximum File size in Bytes

PixAssignFile (`fadc_pixels.config`) :
Filename of configuration file containing the mapping of FrontEnd Pixel numbers to spiral Camera Pixel numbers (see format description below).

BoardStatusFile (`board_status.config`) :
Filename of configuration file with board status flags (see format description below).

RawDataPath (`/remote/filesrv/local/rawdata/`) :
Path where to write the raw data files

LogPath (`log`) :
Path where to write the log files

StatPath (`stat`) :
Path where to write the raw data files

Parameters for DAQ Online analysis 2.3:

ExtractorFirstSlice (0) :
First slice of extractor search window

ExtractorLastSlice (19) :
Last slice of extractor search window

ExtractorWinSize (4) :
Number of slices added up by the extractor

PedestalFirstSlice (18) :
First slice for pedestal extraction

PedestalFirstSlice (29) :
Last slice for pedestal extraction

PedestalMaxDiff (50) :
Maximum difference between highest and lowest slice to classify signal as pedestal signal

ThresCharge (100.0) :
Minimum charge to classify signal as good signal

MinGoodPixels (500) :
Minimum number of pixels required to have a signal above threshold to classify event as calibration event

magic_tcp.config

This configuration file contains the TCP/IP settings of the DAQ program.

CC_Hostname (pc2.magic.iac.es) :
Name of pc where the Central Control (Arehucas) is running.

GUI_Hostname (pc3.magic.iac.es) :
Name of pc where the Online Analysis (Mona) is running.

CC2DAQ_Port (7011) :
Port for commands received from the Central Control.

DAQ2CC_Port (7012) :
Port for reports sent to the Central Control.

GUI2DAQ_Port (7021) :
Port for commands received from the GUI (not used).

DAQ2GUI_Port (7000) :
Port for reports sent to the Online Analysis (Mona).

fadc_pixel.config

This configuration file contains the mapping of FrontEnd pixel numbers to spiral camera pixel numbers. It contains one line for each pixel with the following three numbers:
`<FADC board number> <FADC module number in board> <spiral pixel number>`

The `<FADC board number>` starts with 0 for the first board and counts the boards in the order in which they are read out. Currently, the boards are numbered from 0 to 74 starting from the upper left board (A8) in rack A (currently second FADC rack from the left until board D228 in the lower right part of rack D.

The file can be generated automatically using the "`pixel_numbering`" scripts (see section: **??**).

board_status.config

This configuration file contains flags for each board defining the board status and which data integrity checks will be performed. For each board there is one line with the following three numbers:
`<FADC board number> <board status> <check flag>`

`<FADC board number>` is defined as in the `fadc_pixel.config` file.

`<board status>` can have the following values:

0 :	non existing or connected board (default)
1 :	FADC board
2 :	digital board

`<check flag>` determines the data consistency checks to be performed for this board. There are currently 4 possible checks:

1 :	DAQBoardNumber :	check for correct board number
2 :	FADCEvtNumber :	check if the internal event counter of the board increased by 1 compared to previous event
3 :	FADCClockTick :	check if the internal clock counter increases from event to event by the same amount for all boards
4 :	CheckBytes :	check if the (2*3) check bytes after the FADC data bytes are correct

The checks to be performed are encoded in a hex number where the first bit stands for test 1 etc.

bit = 0 :	do not performed specified check for this board
1 :	perform check for this board and abort run after faulty event if data inconsistency is found

examples:	0x02 :	perform only test 2)
	0x0f :	perform tests 1) - 4)

2.2.3 Data integrity checks

The DAQ automatically performs some data integrity checks. Currently there are 4 different checks (soure code: `fadc2raw.cpp`). If any of the checks fails, the DAQ will close the run rejecting the faulty event (compare configuration parameter: *WriteLastEvent*) and all following events, reset the readout and start a new run. An error message is written to the logfile and to stdout. The error message may look like:

ERROR: Data Integrity Error: 1 Wrong BoardNum: 0 (should be: 110) Event: 1

or

ERROR: Data Integrity Error: 4 in Channel: 0 BoardId: 18 Event: 97981

The board ID or Num refers to the white label on top of each FADC board (not the yellow labels). The channel number refers to the FADC module within each board. Channel number 0 refers to the innermost module, while channel number 7 is the first module when pulling out the FADC board.

Which checks are performed for a given board is defined in the `board_status.config` configuration file. The following checks can be performed:

DAQBoardNumber Every board writes out its board number (see section 2.4). In the data stream the board data blocks should be ordered. The DAQ checks if the board number in the data stream corresponds to the expected board number.

FADCEvtNumber Each board has an internal event counter. The DAQ checks for each board independently if the event counter increased by 1 compared to the previous event. Note: it is not required that all boards have the same event number. In some cases a problem with the initialization of the event counter has been observed which did not affect the data integrity.

FADCClockTick Each board has an internal clock counter. The frequency is defined by `LatticeClkFlag` in the `magic_daq.config` file. The DAQ checks if the clock counter difference to the previous event is the same for all boards (± 5 clock ticks). Again, it is not required that the clock counter value itself is the same for all boards. Only the difference to the previous event matters.

CheckBytes While the previous 3 conditions check if the correct board data block was read out this condition checks the consistency within the board data block. The 30 (actually:`LoGainNumSamples` + `HiGainNumSamples` in `magic_daq.config`) FADC bytes are followed by 3 identical copies of 2 (arbitrary) bytes. The DAQ checks if this additional bytes are really indentical copies.

2.2.4 Logfiles

The logfiles are usually written to */remote/disc01/pc15/daqlog/mreadout_<date>.log*.

2.3 Online Analysis

The DAQ performs a very simple online analysis. The online analysis determines a average pedestal ("from low gain") and the signal area and arrival time for each event and each channel. The MEAN and RMS values are send to the central control (every second) and at the end of each run they are written to an ASCII-file usually in */remote/home/pc2/daq/MagicReadout/stat/dcheck_<date>.txt*.

2.4 FADC system

2.5 Receiver Boards