

User's Guide for the
REGARD MuonTomograph
RaspberryPi Based
Data Acquisition System
and its Graphical User Interface

Gergő Hamar
for the REGARD Group

Wigner Research Centre for Physics, Budapest
Dept. High Energy Physics
REGARD Detector Physics Group

Budapest, 2015.

Contents

1	Introduction	1
2	The RaspberryPi	2
3	The MtRD Board	3
3.1	MtRD Board Connections	4
3.2	Board hardware settings	5
4	The REGARD MuonTomograph Setup	6
5	MtRD Software	7
5.1	Setting up the RaspberryPi	7
5.2	Installation of the MtRpiDaq Software	8
5.3	Starting a run	8
5.4	Initialization files	9
5.5	Description of the switches	10
5.6	Data format	11
5.7	Examples	12
5.8	Stopping a run	13
6	MtRDG, the graphical user interface	14
6.1	Installation	15
6.2	MtRDG functions	16
7	MTS4 Setup, and its GUI	18
7.1	MTS4 Setup	18
7.2	Analysis routines	19
7.3	Graphical User Interface	20
	Abbreviations	22
	References	24

1 Introduction

This User's Guide gives detailed information and instructions for the usage of the "MtRD", which is the RaspberryPi based data acquisition system of the REGARD MuonTomographs. The MtRD can be used to handle both digital and ADC data thus became a standard DAQ in the laboratory of the REGARD Group for several applications and setups. MtRD is used for the new MuonTomograph setups for laboratory and portable applications, for MWPC based cosmic tracking, and for GEM quality assurance tests.

The two main hardware elements of the MtRD data acquisition system, the RaspberryPi mini computer and the MtRD Board will be introduced in sections 2. and 3. respectively.

To achieve better understanding of the MtRD the standard MT setup is described in section 4.

The installation and usage of the DAQ software is detailed in section 5. with several examples. While in section 6. the graphical user interface of the named DAQ software is described.

Section 7. is dedicated to the REGARD Group's special MuonTomograph setup, called "MTS4", its main parameters, and its graphical user interface.

2 The RaspberryPi

The RaspberryPi [1] (RPi) is a mini computer with a 700 MHz processor and all the main standard computer interfaces and additionally several GPIO pins. These features allow us to build a compact tool for the whole data acquisition system.

As the RaspberryPi is a computer, it runs an operating system, called Raspbian [2], which is a Debian based Linux for the RaspberryPi. The basic data storage unit is a standard SD card, where the OS is located on as well. (For list of usable SD card types check [4].)

The DAQ board uses the GPIO pins, while the USB outputs are used for WiFi connection and control of the HV modules.

The RaspberryPi needs a 5V power ($\approx 1A$) through its microUSB connector. Either its standard power supply or custom designed can be used, however noise level should be checked for non-high-performance units.

Note, there are several models of RaspberryPi at 2015, changing the number of ethernet and USB ports, and varying the GPIO pin configuration. Recently a new version, called Raspberry Pi 2 has been declared, with a quad core CPU, thus enhancing its computation power, with full compatibility with older variant. The MtrD Board was designed to fit onto the GPIO pins of the advanced and popular model : **RaspberryPi Model-B Rev-2**. Possible connection for other models can be checked in the Developer's Guide section of this document.



Figure 1: Photo of the *RaspberryPi Model-B*. On the left the SD card can be seen, while on the other side there is a USB WiFi receiver and a USB flash storage unit.

3 The MtRD Board

The data acquisition board is a two-sided printed circuit board, designed by the REGARD Group in the Wigner R.C.P. (Budapest).

The MtRD Board is designed to work with 10 MT chambers, thus has 10 parallel trigger and data input lines. It has three 20-pin IDC sockets called "Trigger", "Data", and "Control"; directly connectible to chambers with baseplate type of REGARD MT-10 (50cm) or REGARD MT-15 (25cm). The Board has a 26-pin IDC socket as well, to communicate with the RaspberryPi via its GPIO pins.

A photo of the MtRD Board can be seen on Figure 1. with marks for all the mentioned main elements. A working setup with several MT chambers, power supply, high voltage feed, and the MtRD data acquisition system can be seen on Figure 3.

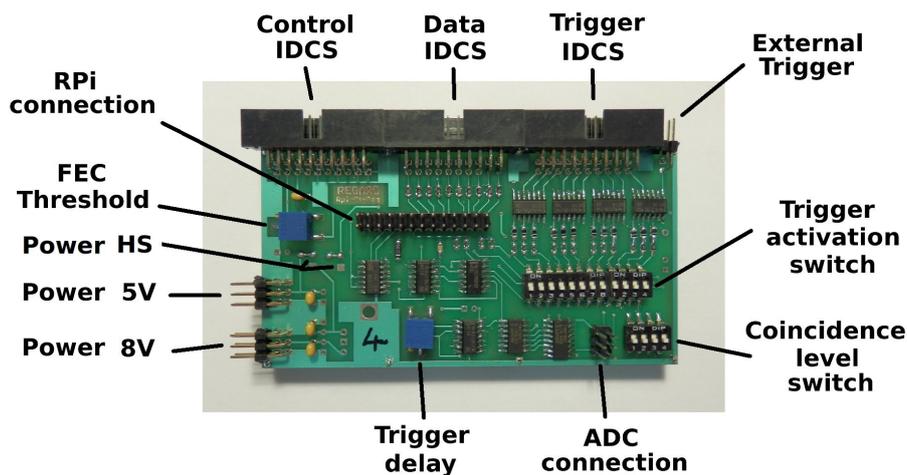


Figure 2: Photo of the Mt RaspberryPi Daq (MtRD) Board, with labels. Detailed description of the connections and switches can be found in section 3.1. and 3.2.

3.1 MtRD Board Connections

Specification of the connectors on the MtRD Board:

- Grounding: Eight single female pinhead legs are used for grounding.
- Power for the board, stabilized 5V : 3x2 pinhead.
Middle ones for the power, other four for ground;
- Power for the board, non-stabilized : 3x2 pinhead.
Middle ones for the power, other four for ground; input voltage should be between 7-10 V (recommended is 8 V).
- Trigger input: 20-pin IDC socket (for ribbon cable)
10 TTL signals and 10 grounds, odd and even pins respectively.
- External trigger input: double pinhead
One TTL signal and the common ground.
- Data input: 20-pin IDC socket (for ribbon cable)
10 TTL signals and 10 grounds, odd and even pins respectively.
- Control output: 20-pin IDC socket (for ribbon cable)
Responsible for the communication and the power distribution for the FECs and the trigger electronics. The pins are in the following order:
G,G,Th,G,5,5,5,G,G,G,Tr,G,G,G,C,G,G,G,G,6;
with meaning of: Th = FEC Threshold, 5 = FEC Power of 5V,
Tr = Trigger, C = Clock, 6 = Trigger card power 5V, G = Ground.
- RPi connection: 26-pin IDC socket (for ribbon cable)
Connection with the RPi GPIO pins. Pin-1 is marked like on the RPi.
- ADC connection: 6-pin IDC socket (for ribbon cable)
Control and data flow for the connected ADCs.

3.2 Board hardware settings

Specification of the switches and potentiometers of the MtRD Board

- Trigger activation switch.
From the 10 trigger input channels, a 100 Hz oscillator, and the external trigger input any can be switched on/off for the trigger conditions.
- Coincidence level switch.
Coincidence level can be set to : OR gated, double-, triple-, quad-coincidence. The switch for the selected level should be on, all the others should be in off state.
- Trigger delay potentiometer.
The proper delay for the trigger's leading edge can be tuned to maximize efficiency for tracking or analog measurements.
- Digital threshold potentiometer.
The threshold for the DIN416 FEC can be set.
- Power input selector hardware-switch
To select the feed for the needed 5V power for the Board and the FECs via soldering a zero resistance for the stabilized 5 V (left) or the 8 V (lower) input.

4 The REGARD MuonTomograph Setup

As the MtRpiDaq is the dedicated data acquisition system for the REGARD MuonTomograph Setup, it is necessary to introduce the setup itself. The standard cosmic setup consists of several layers of multiwire chambers, with 1+1 dimensional readout, several digital electronics of DIN416, and connections compatible with the MtRpiDaq Board eg. baseplate of Regard-Mt10 or Regard-Mt15. In case of a standard setup, the connection and usage of the MtRD Board is straightforward. A photo of a working setup can be seen on Figure 3.

The MtRpiDaq system is flexible thus it can be used for several different detector systems. Several tests with MWPCs, and GEM based detectors were done.

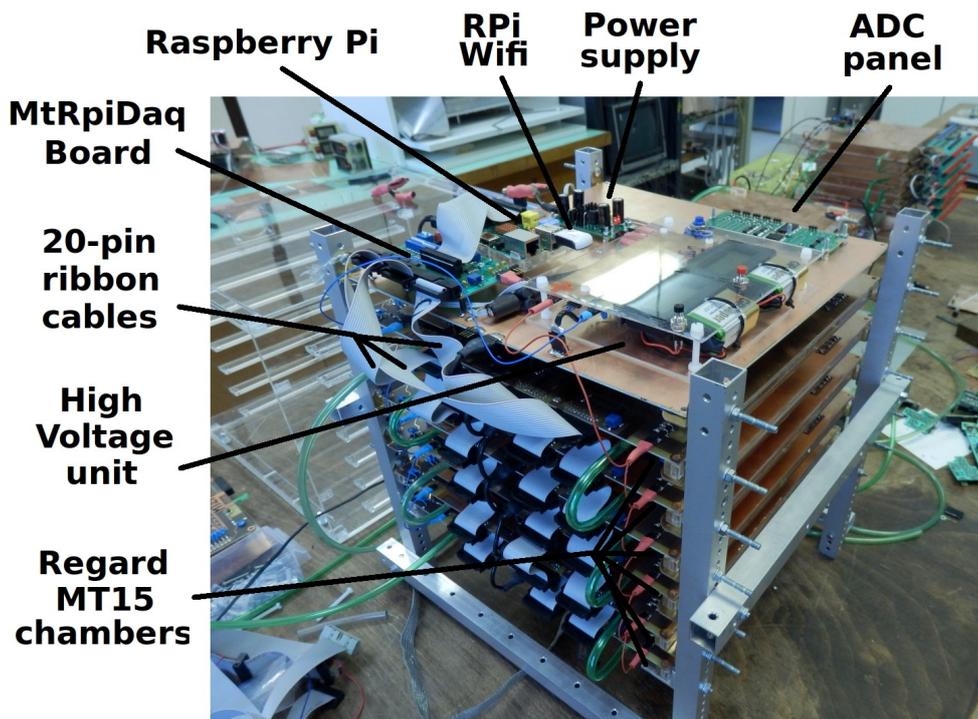


Figure 3: Photo of a REGARD MuonTomograph setup with six Regard-MT15 type chambers at Budapest in February 2015.

5 MtRD Software

The REGARD MuonTomograph RPi DAQ Software is written in *C/C++*, to ensure high speed running, modularity, compatibility, and programmer-friendly upgrade possibility. The DAQ Software runs on the RPi, and is responsible for the communication with the MtRD Board and storage of the data.

5.1 Setting up the RaspberryPi

First of all the operating system of the Raspberry Pi should be settled. Instructions for setting up the SD card with a Raspbian OS can be found in [2]. Tests and software verifications were done with *Raspbian-Wheezy-2013-09-25*, however all Raspbian should be suitable as well.

The default user for the Raspbian system is "pi" with the password "raspberry", therefore this manual will refer to these as defaults.

The RaspberryPi will be used as a distant machine, therefore the proper net connection should be set up; and there are several possibilities. The RPi can be used with LAN cables, however this is not recommended due to the introduced ground loops, which will increase the noise level. However the RPi does not have built-in WiFi, a simple USB WiFi adapter could be used. The WiFi should be configured to connect automatically to the chosen local network. This is the most recommended usage. (Instructions for this can be found for example in [3].) The RPi can work as a WiFi access point as well, this is only preferred if there are no local wireless network options.

From now on the IP address of the RPi will be named RPIIP.
Please check that the RPi is reachable and usable via logging in.
ssh pi@RPIIP ls

Note, as the usage of the GPIO pins need root permissions, most of the running commands will start with the *sudo* command. The user who will run the MtRpiDaq program should have sudo rights, for the default user "pi" this is granted by default.

5.2 Installation of the MtRpiDaq Software

Copy the "MtRpiDaq" directory to the home of the used RaspberryPi user.

```
scp -r MtRpiDaq/ pi@RPIIP:.
```

All source files can be found in the REGARD Budapest Group (WignerRCP) DataServer, for further details ask us at "regard@lists.kfki.hu".

To achieve maximal compatibility with the current Raspbian used, one should recompile the MtRpiDaq Software on the actual RPi; this can be done via the compiling script.

```
ssh pi@RPIIP
cd MtRpiDaq
source CompileMtRpiDaq
```

After successful compilation the MtRpiDaq Software is ready to use.

5.3 Starting a run

The MtRpiDaq Software is a command line driven software. One should start it on the RPi from its directory:

```
cd MtRpiDaq
```

MtRpiDaq has some necessary files and several switches to set parameters for the run. In general the syntax is:

```
sudo ./MtRpiDaq [switch [parameter]]
```

Each run produces two files, one contains the raw event-by-event data (*.ebe*), while the other stores the run-specific settings (*.sett*) like an offline logbook.

There are several initialization files for the basic/default settings, while switches can be used to overwrite these settings.

The detailed list and description of the initialization files and the available switches can be found in section 5.4 and 5.5 respectively. The structure of the raw data output can be found in section 5.6.

5.4 Initialization files

There are three initialization files for the MtRpiDaq Software to be loaded. The existence of these files are mandatory, however their content can be changed any time respecting the need of the usage of the system.

- **RunNumber.ini**
This file stores the run number, contains only one positive integer number. It is increased automatically for any successful start of the MtRpiDaq Software. In case it is needed, it can be changed simply by overwriting it, like:
echo 1025 > RunNumber.ini
- **SettFile.ini**
This file stores the hardware settings and comments to the next run, and will be added to the settings file of the run (as an offline logbook for later use). It should be edited/updated by the user every time before the start of the run, if needed. Any kind of text editor can be used, eg.
vim Settings.ini
- **Filenaming.ini**
The file contains the path and the naming of the coming runs, eg. `../Measurements/Run`". For the actual output files the run number and the extension will be added.
- **ReadoutSettings.ini**
This file stores the default parameters for the readout process. Each line holds a keyword (case sensitive!) and the values to be used as default. All these values can be overwritten by the switches. The keywords with the corresponding switches are: Statistics (-e), NChambers (-c), Channels (-c), ChamberSize (-s), NAdcValues (-a), Prevalence (-p), ReorderBits (-r); their description can be found in the section of the switches: section 5.5.

5.5 Description of the switches

All the switches have a short and a long name with equal response. Most of the switches need an additional input eg. numbers; these are written in capital letters in the following list.

The list of switches for the MtRpiDaq Software are the following:

- -h, --help
Shows the help, where these switches are listed with their short descriptions.
- -e, --events NUMBER
Sets the number of events to be recorded. Minus one (-1) means a semi-infinite run.
- -c, --channels NUMBEROFCHANNELS CHID0 CHID1 ...
One can specify how many channels should be used from the 10 lines, and which are those channels.
- -s, --size SIZEX SIZEY
Sets the size of the chambers in use. The two integer parameters are for the number of FECs (16 channels) used in X and Y. Thus altogether $(SIZEX+SIZEY)*16$ digital bits will be read out.
- -a, --adc-values NUMBER
From the additional ADC connector NUMBER values will be read out.
- -r, --reorder-bits [yes/no]
In case of Regard-Mt10 or Regard-Mt15 baseplates a reordering of bits should be asked.
- -f, --force
Starts the measurement and overwrites the files without question. Useful for starting in the background or with the *nohup* command.
- -p, --prevalence NUMBER
Sets the prevalence for writing info to the screen, thus after every NUMBER of events it gives feedback about the data collection process, and the last event.
- -b, --break-line NUMBER
Artificial line-break after every NUMBER of 16 bits for screen info.

5.6 Data format

The settings file (.sett) contains the starting date and time of the run, the run number, all applied readout settings (from file or from the used switches), and the content of the edited SettFile.ini file for notes (see section 5.4).

The basic element of the collected data is the event; each event is in a new line in the file. The data file contains information on time, hits, triggers and ADC values for all the events. The stored numbers for the events with proper order and description are listed below.

- Event number
Simple counter for the events; can be efficiently used for offline crosschecks and data integrity.
- Time information
Counts the elapsed time in microseconds between this and the previous event. Sum of these gives the total time the run took.
- Hits
For all channels (chambers) the non-zero bits are counted (NHits) and their position stored (HitPos0,..). For all the channels in growing id, these numbers are listed.
- Number of missed triggers
During the readout of data new triggers cannot be accepted, but are counted and stored.
- Crosscheck
Predefined bitpattern are read out for every event to ensure data quality. In case this crosscheck was passed it is 1, otherwise 0.
- Trigger patten binary code
For the 10+1+1 channels the trigger states are store in a 12 digit binary number, written out its corresponding decimal value.
- ADC values
All the read ADC values in order of connection.

5.7 Examples

Let us have a setup of four Regard-Mt15 chambers with 2+2 FECs, with channels 0,1,5,6. We would like to record their pulse height as well with four ADCs; Let us collect 10k event, and let us see every hundredth event on the screen. For this the "ReadoutSettings.ini" should look like this:

```
NChambers 4
Channels 0 1 5 6
ChamberSize 2 2
NAdcValues 4
Prevalence 100
Statistics 10000
ReorderBits Yes
```

One should save the hardware settings in the "SettFile.ini" file eg.:

```
Chambers: four MT15 type
PositionsMm 0 50 100 150
Gas Ar:CO2 80:20
Test measurement
```

Before the first use one should check the "Filenaming.ini" to point to an existing directory with the desired naming pretag, eg:

```
../Measurements/MtRun
```

Before the start of the run one should check the run number as well, to avoid overwriting of some old valuable data:

```
cat RunNumber.ini
```

If everything is ok, we can start the run simply with the command:

```
sudo ./MtRpiDaq
```

Starting from the previous files let us try different configurations. Let us change the run number to 288:

```
echo 288 > RunNumber.ini
```

In case we do not want to read out ADC, and have an semi-infinite run:

```
sudo ./MtRpiDaq -events -1 -adc-values 0
```

Or let us start the process in the background with no hangups, thus one can log out from the computer without disturbing the data taking. For this we need to start it without any crosschecking questioning; and it is recommended to use a file as standard output.

```
nohup sudo ./MtRpiDaq -e -1 -a 0 -f &> StdDump &
```

5.8 Stopping a run

The termination of a running MtRpiDaq Software should be done via the SIGINT or SIGTERM signal to ensure closure of open files and active connections. In case the command line is active one can use the standard break of unix: "Control-C". In case of detached process killall command can be used:

```
sudo killall MtRpiDaq.run
```

If the RaspberryPi has been switched off during the DAQ process, the file called "Running" should be deleted from the "MtRpiDaq" directory before the start of the next run.

6 MtRDG, the graphical user interface

This section is about the Regard MuonTomograph RaspberryPi based Data Acquisition System's Graphical User Interface (MtRDG).

In the previous sections the MtRD system has been introduced, where the DAQ software runs on the RaspberryPi machine. However, the simplest way to control it is locally from the RPi itself, the most convenient is to log into the RPi as a remote PC via eg. ssh, and control the data acquisition. For the later case an additional graphical user interface (MtRDG) have been developed, where the GUI runs on the local machine, and send the commands via ethernet connection.

The GUI only sends the commands to the RPi, thus the GUI can be closed at any time during data taking; while reopening the GUI does not affect the current data taking. Even multiple GUIs can be used for the same MtRD (however it is not recommended to use them at the same time to avoid confusion.)

The recorded data files will be stored on the RPi, with path and naming given by the "Filenamng.ini" file of the MtRpiDaq, see section 5.4.

6.1 Installation

The source of the MtRDG is written in *WxWidgets* [5], which is a C++ based cross-platform programming language for development of GUIs.

Copy the "MtRpiDaqGui" directory to the desired location onto the computer to be used for the remote access. All source files can be found in the REGARD Budapest Group (WignerRCP) DataServer, for further details ask us at "regard@lists.kfki.hu".

To achieve maximal compatibility with the operating system in use, one should recompile the MtRpiDaqGui on the actual local machine. For this one has to have the necessary *WxWidgets* libraries, which can be downloaded from the standard linux repositories or from the homepage of *WxWidgets*. As the MtRDG uses standard linux commands, it is mandatory to use unix based OS. The cross-platform feature of *WxWidgets* allows us to compile it on several different platforms, for details see [5].

The compilation successfully was tested on *Ubuntu 10.04* with *WxWidgets 2.6* and *WxWidgets 2.8* as well. The compilation script should be called from its directory:

```
source CompileMtRpiDaqGui
```

As the MtRDG communicates with the RPi it is necessary to free this data flow from password protection. This can be done with the linux command *ssh-keygen* (for further information on this see its manual). A tiny script has been written to simplify this process, call it with the IP address of the RPi as parameter:

```
source RpiLogin.sh RPIIP
```

In case an ssh-key exists on the local machine, it will ask about overwriting the key with a new one, the recommended answer is 'no'. This script must not be called multiple times from the same machine and user, thus the multiple log of the ssh-key will result in denying the access onto the RPi. This can be cured with removing the referred lines or the entire file : "pi@RPIIP:~/.ssh/authorized_keys".

6.2 MtRDG functions

The MtRDG has only one initialization file : "GuiSettings.ini". This contains the most important parameter, the IP address of the RPi. It can have one other parameter, the number of chambers to be shown, what will be saved to the .sett file for the offline logbook. The keywords are "IpAddress" and "NChambers" respectively.

A screenshot from the MtRDG can be seen on figure 4. The GUI has one window, which can be separated into three main parts.

All the parameters of the MtRD Software (see section 5.) for the readout can be set on the upper part of the panel. The naming and meaning of the parameters are done like in the section 5.

The middle part, called "Settings" will be saved in the .sett file for the offline logbook (together with the main parameters from the readout part).

The lower part contains the status of the MtRD, and can be made recent with the "Update" button. The run can be started with the "Start" button, and can be stopped any time with the "Stop" button. The button "Quit" closes the GUI, however has no effect on the data taking and on the MtRD itself.

In case the run was interrupted by a power cut on the RPi, than before the next run the "Stop" button should be used to reset the indicator of the running condition on the MtRD.

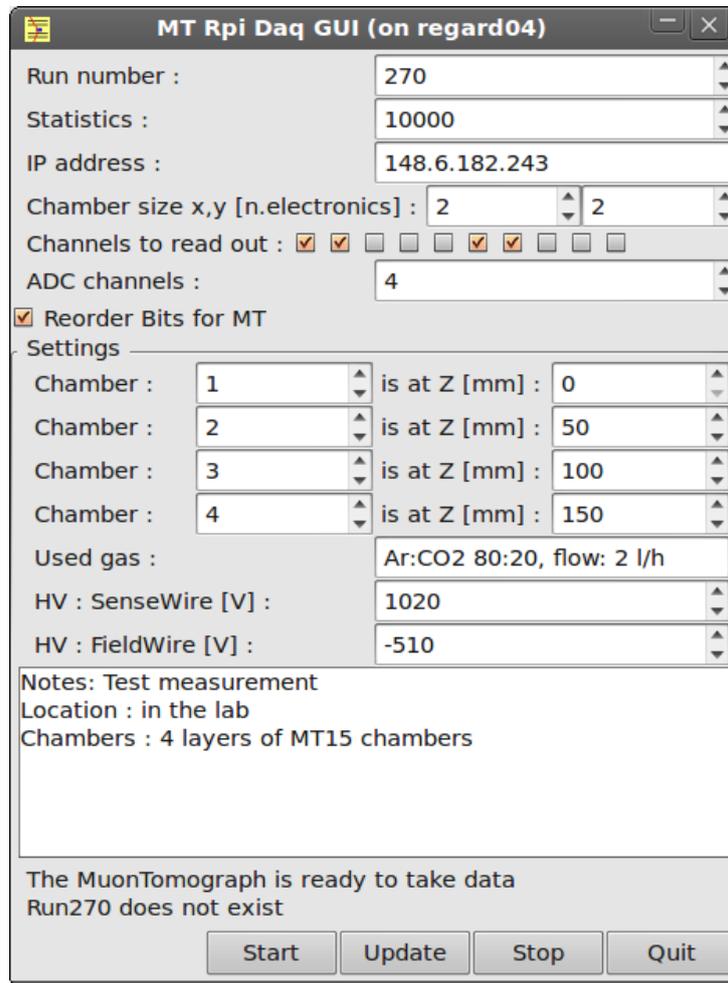


Figure 4: Screenshot from the MtrD Graphical User Interface. Parameters are set as for the first example in section 5.7.

7 MTS4 Setup, and its GUI

This section is dedicated to the REGARD Group's special MuonTomograph setup, called "MTS4", its main parameters, and its graphical user interface.

7.1 MTS4 Setup

The REGARD MTS4 Setup consists of six layers of Regard-MT15 type Close Cathode Chamber [6] [7], a small high voltage unit, the MtRpiDaq system with all its components, and a single input power supply. The support and enclosing structure of the setup is a plexiglas box, what is used as an environmental isolator as well. Similar setups were used for muon tomography in underground natural and artificial caves [8] [9]. A photo of the Regard-MTS4 Setup can be seen on figure 5.

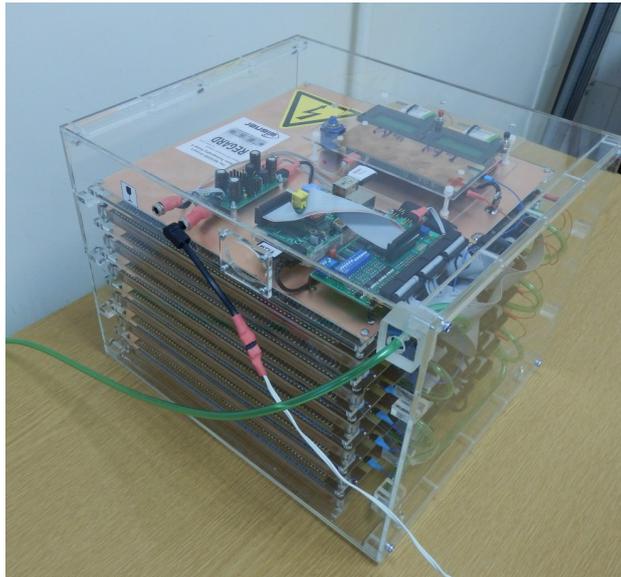


Figure 5: The Regard-MTS4 Setup.

The connection with the RPi is done via wireless networking, where the RPi is the host and gives IP via DHCP on its network with SSID: Mts4Rpi and password: RegardMts4. The IP of the RPi is 192.168.0.1.

7.2 Analysis routines

As a cosmic muon passes through the chambers it leaves hits, from which the track can be reconstructed. Tracking algorithms (like in particle physics experiments) can be used to find the tracklets, and thus compute the cosmic flux and check the performance of the detectors. (Like in [8] [9]). Analysis routines for the known configuration of the MTS4 setup has been implemented, the compiled analysis program (Mts4Analysis) has been put onto the RPi of the DAQ. This way the basic results and standard plots can be generated for any of the recorded runs.

The analysis code should be called from its directory "Mts4Analysis/", with the command:

```
./Mts4Analysis.run
```

The Mts4Analysis code needs three input variables in the file "Mts4Analysis.sett" (located in the code's directory) with keywords and corresponding integer values. The identification number of the run (RunNumber), the binsize for the calculation of cosmic flux (FluxBinSize), and the binsize for the time dependent parameter crosschecks in minutes (TimeBinSize).

The analysis code produces two files, into the directory of the runs "/home/pi/Measurements/" with the same prefix ("MtRun") and run number as the data to be analyzed. The results file (.res) contains the one- and tow-dimensional histograms of the computed values for cosmic flux, hits, and efficiencies for diagnostics. All these set of values are labeled in the previous line with human readable hashtag comments, while the numbers are written with standard ASCII characters. The analysis code produces a set of standard plots form the mentioned results file, which plots are exported as a .pdf file.

The analysis code has been written by László Oláh (olah.laszlo@wigner.mta.hu) of the REGARD Group in the WignerRCP Budapest.

7.3 Graphical User Interface

The graphical user interface of the MTS4 setup (Mts4Gui) is based on the MtrRDG, thus compilation and requirements as the same as for the MtrRDG (see section 6.).

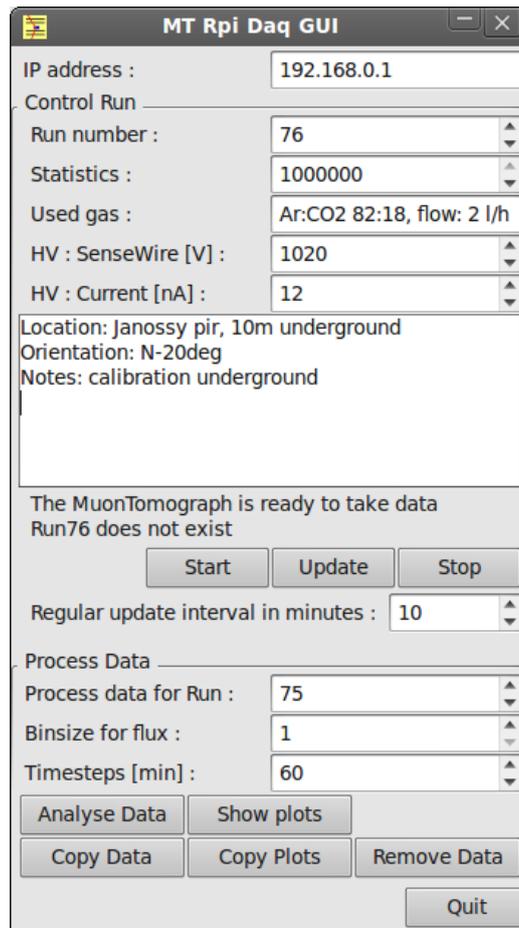


Figure 6: Screenshot from the Regard-MTS4 Graphical User Interface. ...

The fix parameters of the setup has been removed from the panels, and several new useful buttons have been added to call the analysis code (mentioned in the previous section) and do simple file handling. A screenshot of the Mts4Gui can be seen on figure 6.

The description of the upper part of the Mts4Gui can be found in section 6. In this version, the updates can be set up be made on a regular basis; change of interval will take action after pushing the "Update" button.

The special section of the MTS4 setup is located on the lower part of the window, in the panel "Process Data". Parameters of the analysis code of

section 7.2. can be set, and the code can be called via the button "Analyse Data". The program will notify the user the estimated time of the analysis process (which may be even minutes!), what shall be waited for to elapse. Note, as the RPi is a low performance machine, running the analysis besides the data taking can slower both processes, thus it is highly not recommended.

Data from the runs are recommended to be saved on an other (secured) machine, therefore copy buttons have been added into the GUI. The produced files of a run, all of them or only the plots, can be copied via the "Copy Data" and "Copy Plots" buttons respectively onto the local machine of the GUI into the directory "../Measurements/". The pdf file of the copied plots can be checked with any pdf viewer, and if *evince* is installed it can be called via the "Shot plots" button.

However the storage space on the RPi is moderately large (MTS4 uses 16GB SD cards), it is certainly limited; thus occasional storage space shall be freed via removing the (previously saved!) data files. This can be done directly after logging into the RPi, or run-by-run with the button "Remove Data", which removes all raw and analysis output files from the RPi. Do not forget to save all these files before removing them.

Abbreviations

ADC - Analog-to-Digital Converter

Electronic device that converts the the continous analog voltage to a digital number at a defined time.

CCC - Close Cathode Chamber

Asymmetric MWPC with low material budget and high tolerance on mechanical imperfections; designed by the REGARD Group. This is the standard chamber for the REGARD MT. Further details can be found in [6] [7].

DIN416 -

Digital FEC with 16 channels, designed by the REGARD Group. The standard FEC for the REGARD MT chambers.

FEC - Front-End Card

Electronic boards on the detector side, typically with the functions of signal preamplification, shaping, and digitalization.

GEM - Gas Electron Multiplier

The most used element of the novel micropattern gaseous detector family in instrumentation of particle physics.

GPIO - General-Purpuse Input/Output

GPIO is a generic pin on an integrated circuit which can be set to behave wether as an input or as an output.

GUI - Graphical User Interface

A human computer interface where users can interact with the electronics via graphical elements like buttons and icons.

IDC - Insulation-Displacement Contact

IDC is the standard connector type for ribbon cables.

IDCS - IDC Socket

PCB mountable standard socket for IDC connectors.

MT - Muontomograph

MT10,MT15 -

Standard baseplated for the 50cm and 25cm wide REGARD MuonTomograph chambers.

MtRD - MuonTomograph RaspberryPi based Data acquisition

The data acquisition system for the REGARD Muontomographs.

MtRDG - MtRD Graphical user interface

The graphical user interface of the MtRD runs on a remote computer, and can access and control the MtRD via communication through local ethernet connection.

MTS4 - REGARD Muontomograph Setup v4

A special fully operational fix MT setup of the REGARD Group. The setup consists of six layers of MT15 chambers with DAQ, HV, and LV modules included.

MWPC - Multi-Wire Proportional Chamber

Gaseous detector, typically used for tracking of charged particles.

PCB - Printed Circuit Board

REGARD -

Gaseous detector research and development group in the WignerRCP in Budapest.

RPi - RaspberryPi

The RaspberryPi is a miniature single-board computer, for details see section 2. and [1].

wxWidgets -

The wxWidgets is a C++ based programming language for graphical user interfaces with cross-platform support.

References

- [1] Raspberry Pi webpage
www.raspberrypi.org
- [2] Raspbian Operating System
www.raspbian.org
- [3] Setting up wifi connection for a RPi
<http://weworkweplay.com/play/automatically-connect-a-raspberry-pi-to-a-wifi-network/>
- [4] List of usable SD card for the RaspberryPi
http://elinux.org/RPi_SD_cards
- [5] WxWidgets programming language
<http://www.wxwidgets.org/>
- [6] D.Varga, G.Hamar, G.Kiss
Asymmetric multi-wire proportional chamber with reduced requirements to mechanical precision
Nucl.Instrum.Meth. A 648 (2011) 163-167
- [7] D.Varga, G.Hamar, G.Bencédi, G.Kiss
Close Cathode Chamber: Low material budget MWPC
Nucl.Instrum.Meth. A 698 (2013) 11
- [8] G.G.Barnaföldi, D.Varga, L.Oláh, G.Hamar, H.G.Meleg, G.Surányi
Portable Cosmic Muon Telescope for Environmental Applications
Nucl.Instrum.Meth. A 689 (2012) 60
- [9] L.Oláh, G.G.Barnaföldi, G.Hamar, H.G.Meleg, G.Surányi, D.Varga
CCC-based Muon Telescope for Examination of Natural Caves
Geosci. Instrum. Method. Data Syst. 1 (2012) 229