Muonic - Documentation

svn Revision 92, Nov. 2011

Contents

1 Disclaimer

This software is still under developement. This means, that some features do not work properly. Data loss or severe system damage might occur.

The software is distributeted under the terms of Gnu Public License (GPL), so the authors are not resposible for any damage, which is caused by using this software. By installing this software, the user accepts the terms of the GPL `http://www.gnu.org/licenses/`.

2 Start of the software

The software is written completely in `python`, so it should run under every operating system. However, in this version, it is highly recommended to run it under `LINUX`. On a `LINUX` box, it can be started via command line

```
python daq.py -f FILENAME [-options]
```

after one changed to the `muonic` cirectory with

```
cd muonic
```

The file FILENAME will then be an ASCI file, which makes it possible to reproduce the plot, which is shown in the tab `Muon Rates`. In the following the further options:

-s A simulation mode is started, which can be used for testing and demonstration. It allows to launch the program without an attached DAQ card. The shown data is completely arbitrary.

-d Produces more logging information in the terminal window.

-t TIMEWINDOW Allows to change the timewindow for which the muon rate is calculated to a value of TIMEWINDOW seconds. Default is 5 seconds. If not explicitely needed, this should not be changed.

-p Creates a file with the individual pulsetimes.

-i INFILE Has no effect in this revision.

-h Shows the options explained above.

In most cases, the program is then launched with a simple:

```
python daq.py -f FILEINAME
```

3 General

Autamatically created files by the software (like `*_decays`) should end up in the subfolder `data`.

4 Menu

4.1 File

Just for exiting the program

4.2 Settings

Coincidence kriteria and threshold settings for the individual channels can be selected here. Also some more advanced options can be selceted. By clicking the `Ok` button in the individual windows, the commands of choice will be issued to the DAQ card, and the settings will be stored in the CPLD register of the card until the card is switched of.

4.2.1 Channel Configuration

Use Channel Activate the different channels. Only channels with checked boxes are activated, all others will be turned off.

Coincidence Choose a coincidence level which defines the trigger criterion
Singles: No trigger
Twofold: Trigger, if at least two channels see a signal
Threefold: Trigger, if at least three channels see a signal
Fourfold: Trigger, if all four channels see a signal

Veto : One of the channels can be defined as veto. This means, if this channel detects some signal, no trigger is recorded. Choosing channel 0 as veto is not possible. If one does not like to choose a veto, one has to select always `None`.

4.2.2 Thresholds

The thresholds for the individual channels can be chosen. If possible, the current threshold is displayed above the input line.
The thresholds have to be chosen in such a way that for a rate measuerement all channels see about the same rate and the changes in the rate look

correlated, if the attached scintilation counters are situated on top of each other. The expected rate for one of the scintilation counters in the silver metal boxes is about 6-7 Hz. The thresholds are also stored in the DAQ card's CPLD register.

4.2.3 Options

Use DAQ CPLD clock rather than software clock for rate calculation This feature does not work properly at the moment. It should allow the user to select the CPLD trigger time stamp from the DAQ card as time measurement for the muon rate calculation instead of the software clock.

Emulate Chan3 Software Veto For cases, in which the hardware veto does not work, a software veto can be established for a muon decay measurement. This can only be done for channel 3.

4.3 Help

By clicking `DAQ Commands`, a window with a text taken from a Fermilab *.pdf document with a listing of DAQ commands pops up.

5 The tabbed interface

5.1 DAQ Output

In the large text window everything which comes and goes to the DAQ card is displayed. Commands can be issued via `Command`. By clicking on `Save to File` it is possible to store the raw data in a file. This file has to be created first. The button `Periodic Call` allows to periodically sent a command to the DAQ card.

5.2 Muon Rates

Propably the most important tab. A plot of the individual channel and trigger rates which is updated every 5 seconds is shown. At the right handside, the average rates since the start of the measurement are shown, as well as the total number of events. Further below, the start time of the measurement and the maximum of the measured rates is displayed. If the maximum rate is very high, this might indicate a problem with the software.

Also a `Stop` button can be found, which allows to hold the plot. By hitting `Restart` the plot is cleared and a new plot is started.

5.3 Muon Lifetime

If the box `Look for decaying muons` is checked, the coincidence and veto criteria are selected in such a way that for three scintilators on top of each other two adjacent triggers in an intervall of 20 microseconds are counted as a decayed muon. The histogram underneath helps to check signal purity, since the peak should be at 2.2 microseconds. Also an exponential decay should be visible. The decay information is stored in a file with the ending `*_decays`. This file can then be analyzed further.

5.4 Pulse Analyzer

By checking `Show the last triggered Pulses in the time intervall` the software shows the last triggered pulses in the 5 second time intervall. This is only for demonstration and has no effect.

6 Measurements with Muonic

6.1 Calibration for a rate measurement

After starting the program with `python daq.py -f FILENAME` the channel configuration window has to be opened. One needs to activate the channels which should be used. Now every single channel has to be calibrated, which means that one has to find a suitable threshold. To do so, one uses the plot which one can find in the tab `Muon Rates`. To change the threshold settings, one uses the `Threshold Settings` dialogue.

6.2 Muon telescope

If the calibration is succesful, one needs two of the scintilation counters, which one puts on top of each other in a distance of about 40cm. A twofold coincidence must be chosen. The rate of triggers now provides a criterion how many muons might have passed the scintillators. Now one can change for example the angle of the two scintilators to investigate the angular dependence of the observed muons, or one can observe at different weather conditions, etc.

6.3 Muon-decay

You will need at least 3 scintillation counters!
A measurement will take a long time to collect enough data (days!)
The muon decay measurement with `muonic` is at the moment still under developement, but for those who want to try it, there are two ways to do it:

Offline One puts 3 scintilation counters on top of each other, the order of the channels must be 0-2-1 with 0 at the top. Then one records the raw DAQ data (this can be done in the tab `DAQ Output`).
The raw data than can be fed into the shell script `create_mudecay_fit.sh`. To do so, one changes with `cd` into the subfolder `analysis` and executes `sh create_mudecay_fit RAWDATAFILENAME`. If the data is located in another folder than in the `analysis` folder, one has to provide the complete path, like this:

```
sh create_mudecay_fit /home/cosmic/DATA/RAWDATAFILENAME
```

After a succsesful execution of the script, a window with an exponential fit to the data pops up, and the calculated decay is displayed as one of the fit parameters.

Semi-Offline One chooses the tab `Muon Lifetime` and checks the checkbox. Then one puts at least 3 scintilation counters on top of each other, channel 3 must be the downmost.
In the subfolder `data` a file `FILENAME_decays` is now created, this can be analysed with the scripts in the `analysis` subfolder: To do so, one executes in the given order

```
python get_numbers.py ../data/DATEINAME_decays > output
python fit.py output
```

Afger that, on can delete the automatically created file `output` with

```
rm output
```

The created exponential fit will be stored as *.png graphics in the file `fit.png` and can be opened by simply double clicking it in the file browser.