

Muon rate display

Pedro Piçarra and Ricardo Amadeu

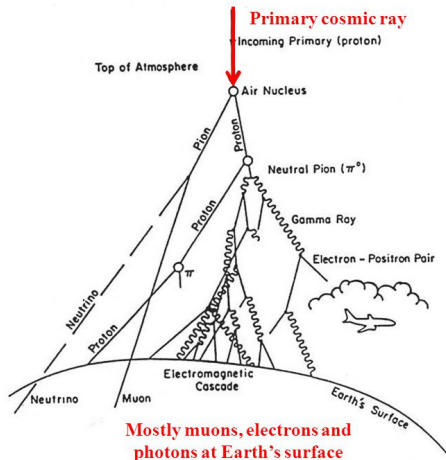
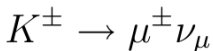
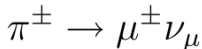
Supervisor: Fernando Barão

September 5, 2018

- **Muons** are produced in the **interaction between cosmic rays and the atmosphere** and can be detected with scintillation counters.

Introduction

- **Muons** are produced in the **interaction between cosmic rays and the atmosphere** and can be detected with scintillation counters.
- Muons are the result of Meson's decay like pions which arise from the collision between a proton and atmospheric particles.



Muon rate

- The surface **muon flux presents periodic variations** caused by many physical processes (which occur both in space and in the Earth's atmosphere).

Muon rate

- The surface **muon flux presents periodic variations** caused by many physical processes (which occur both in space and in the Earth's atmosphere).
- The **atmosphere's density and temperature fluctuation** throughout the year is the **cause of the variation of muon rates**. In the Summer, the atmosphere is warmer and less dense therefore the pions are less likely to interact and more likely to decay to muons.

Muon rate

- The surface **muon flux presents periodic variations** caused by many physical processes (which occur both in space and in the Earth's atmosphere).
- The **atmosphere's density and temperature fluctuation** throughout the year is the **cause of the variation of muon rates**. In the Summer, the atmosphere is warmer and less dense therefore the pions are less likely to interact and more likely to decay to muons.

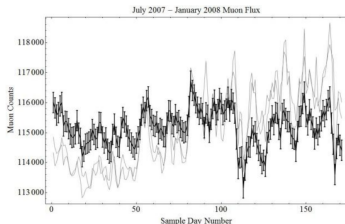


Figure: The darkest lines are the temperature and pressure while the lighter lines are the muon counts

Project main goals

- **Record events triggered by muon detection** in scintillation counters, with their respective timestamps

Project main goals

- **Record events triggered by muon detection** in scintillation counters, with their respective timestamps
- **Organize data of events** recorded to simplify its analysis, e.g. graphics

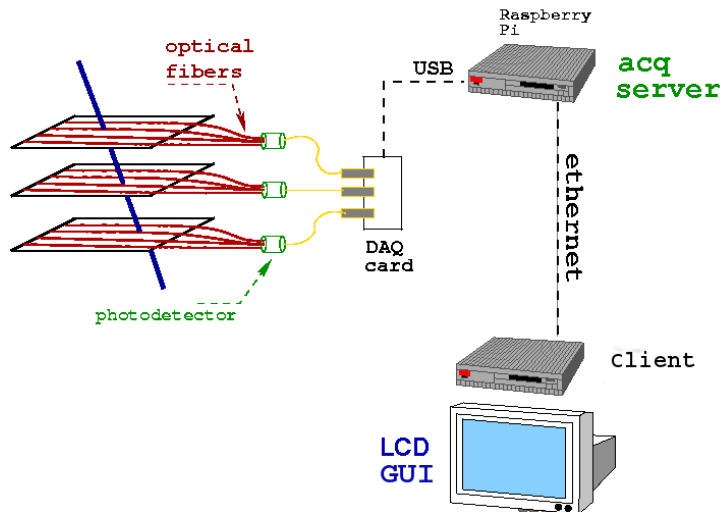
Project main goals

- **Record events triggered by muon detection** in scintillation counters, with their respective timestamps
- **Organize data of events** recorded to simplify its analysis, e.g. graphics
- **Remotely control** the instruments used in **the experiment** and their condition (Slow Control) and **live monitor the muon detection rates**

Project main goals

- **Record events triggered by muon detection** in scintillation counters, with their respective timestamps
- **Organize data of events** recorded to simplify its analysis, e.g. graphics
- **Remotely control** the instruments used in **the experiment** and their condition (Slow Control) and **live monitor the muon detection rates**
- Provide a **high level of abstraction** to facilitate the user's interaction with the program

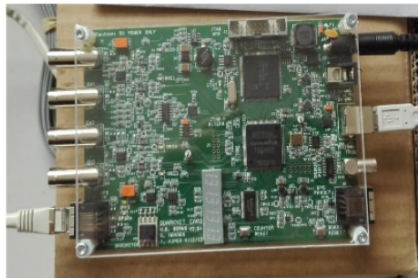
Muon telescope



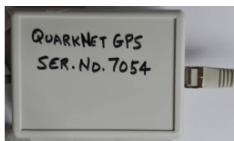
Main components of Project



Raspberry Pi



QN card



QN GPS

Project tasks

- Develop a **Server** to be launched in the **Raspberry Pi**.

Project tasks

- Develop a **Server** to be launched in the **Raspberry Pi**.
- Develop a **Client**, which is launched in the **user's computer**, to remotely control the data acquisition.

- Develop a **Server** to be launched in the **Raspberry Pi**.
- Develop a **Client**, which is launched in the **user's computer**, to remotely control the data acquisition.
- Develop a **Graphical User Interface** for live monitoring of detection muon rates and manage the communications between client and server

Project tasks

- Develop a **Server** to be launched in the **Raspberry Pi**.
- Develop a **Client**, which is launched in the **user's computer**, to remotely control the data acquisition.
- Develop a **Graphical User Interface** for live monitoring of detection muon rates and manage the communications between client and server
- The whole project is written in **C++** programming language.

Project tasks

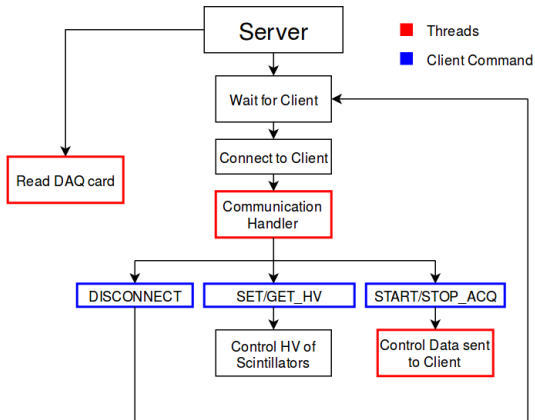
- Develop a **Server** to be launched in the **Raspberry Pi**.
- Develop a **Client**, which is launched in the **user's computer**, to remotely control the data acquisition.
- Develop a **Graphical User Interface** for live monitoring of detection muon rates and manage the communications between client and server
- The whole project is written in **C++** programming language.
- The GUI was written with **gtkmm**, which is the official C++ interface for the popular GUI library GTK+.

Project tasks

- Develop a **Server** to be launched in the **Raspberry Pi**.
- Develop a **Client**, which is launched in the **user's computer**, to remotely control the data acquisition.
- Develop a **Graphical User Interface** for live monitoring of detection muon rates and manage the communications between client and server
- The whole project is written in **C++** programming language.
- The GUI was written with **gtkmm**, which is the official C++ interface for the popular GUI library GTK+.

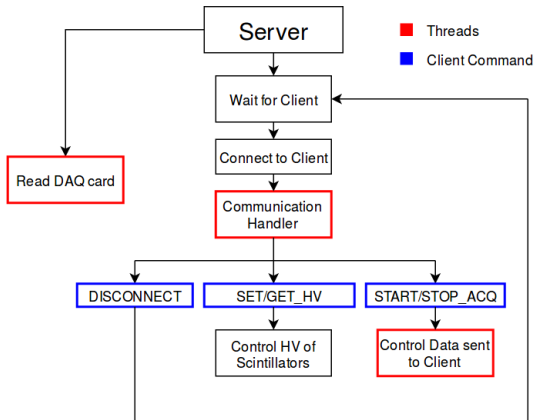
Server tasks

- Continuously **listen to client connections**



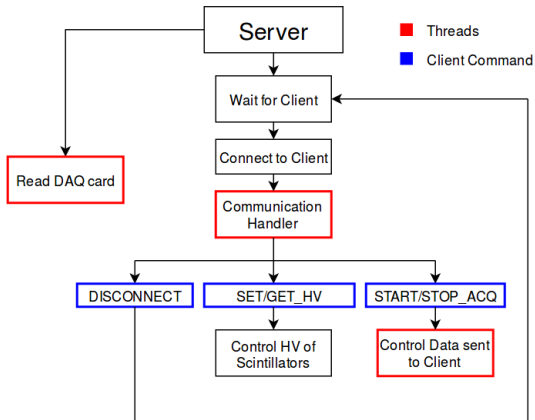
Server tasks

- Continuously **listen to client connections**
- **Store data events** into a queue (FIFO container) of strings.



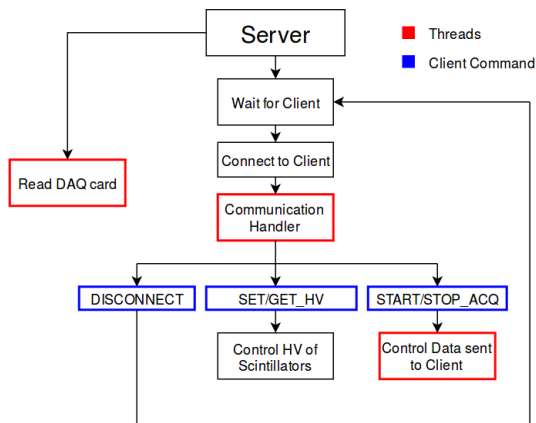
Server tasks

- Continuously **listen to client connections**
- **Store data events** into a queue (FIFO container) of strings.
- **Receive commands** from the Client



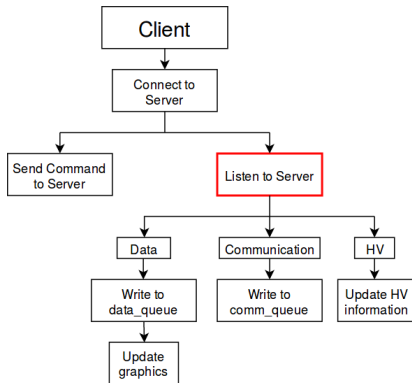
Server tasks

- Continuously **listen to client connections**
- **Store data events** into a queue (FIFO container) of strings.
- **Receive commands** from the Client
- Read event_queue and **send data to Client** (strings of information with time stamps)



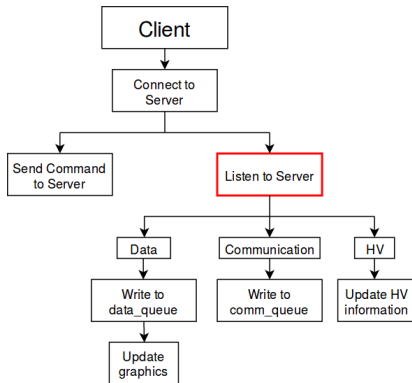
Client tasks

- Connects to Server during its initialisation



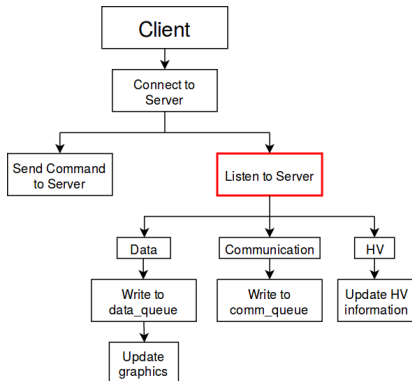
Client tasks

- Connects to Server during its initialisation
- **Sends commands** to Server



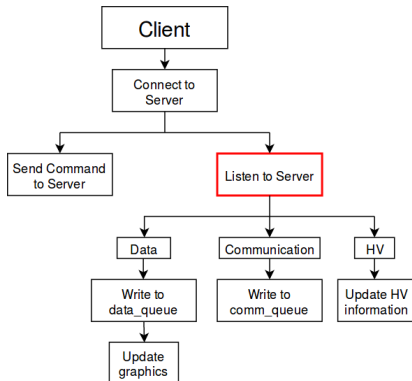
Client tasks

- Connects to Server during its initialisation
- **Sends commands** to Server
- Launches a thread that is continuously listening to Server



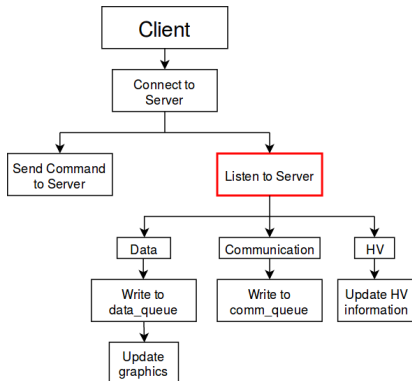
Client tasks

- Connects to Server during its initialisation
- **Sends commands** to Server
- Launches a thread that is continuously listening to Server
- **Stores communication info and data acquired** from server in separate queues

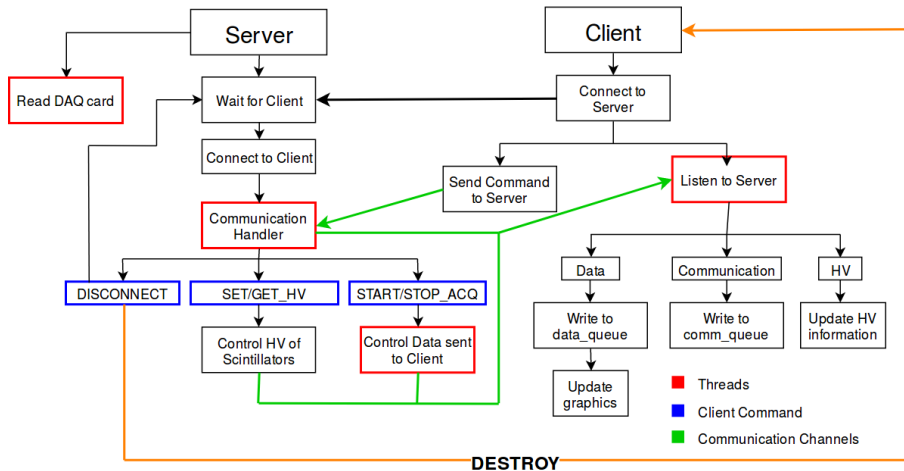


Client tasks

- Connects to Server during its initialisation
- **Sends commands** to Server
- Launches a thread that is continuously listening to Server
- **Stores communication info and data acquired** from server in separate queues
- Creates **graphics displaying data events** received



Server-Client interaction



LabRC GUI - Muon Telescope Monitor (Jul 2018)

Server:
 IP:
 Port:

Control:
 Acquisition:
 HV:

Graph:
 MUON HV
 July 2007 - January 2008 Muon Flux

Console:

```

Wed Sep 5 02:07:01 2018: [LRCQncard::ServerCommandHandler] Communication Terminated
Wed Sep 5 02:07:01 2018: [LRCQnclient::SetCommand] Command sent... EXIT
Wed Sep 5 02:06:59 2018: [LRCQncard::ServerCommandHandler] acquisition stopped...!!!!
Wed Sep 5 02:06:59 2018: [LRCQnclient::SetCommand] Command sent... STOP_ACQ
Wed Sep 5 02:06:21 2018: [LRCQncard::ServerCommandHandler] acquisition started...
Wed Sep 5 02:06:21 2018: [LRCQnclient::SetCommand] Command sent... START_ACQ
Wed Sep 5 02:06:18 2018: [LRCQnclient::LRCQnclient] Connection status: success
Wed Sep 5 02:06:18 2018: [LRCQnclient::LRCQnclient] Valid Address
Wed Sep 5 02:06:18 2018: [LRCQnclient::LRCQnclient] Socket open...
Wed Sep 5 02:06:18 2018: [LRCQnclient::LRCQnclient] The port number is: 4000
  
```

Next steps

- Conclude the **High-voltage control**

Next steps

- Conclude the **High-voltage control**
- Conclude the **graphic display**

Next steps

- Conclude the **High-voltage control**
- Conclude the **graphic display**
- Study the **correlation between the muon's rate and the weather conditions**, namely temperature and atmospheric pressure.

Next steps

- Conclude the **High-voltage control**
- Conclude the **graphic display**
- Study the **correlation between the muon's rate and the weather conditions**, namely temperature and atmospheric pressure.
- **Display the results** in the 'LIP Control Room' in IST