



TÉCNICO
LISBOA



LABORATÓRIO DE INSTRUMENTAÇÃO
E FÍSICA EXPERIMENTAL DE PARTÍCULAS

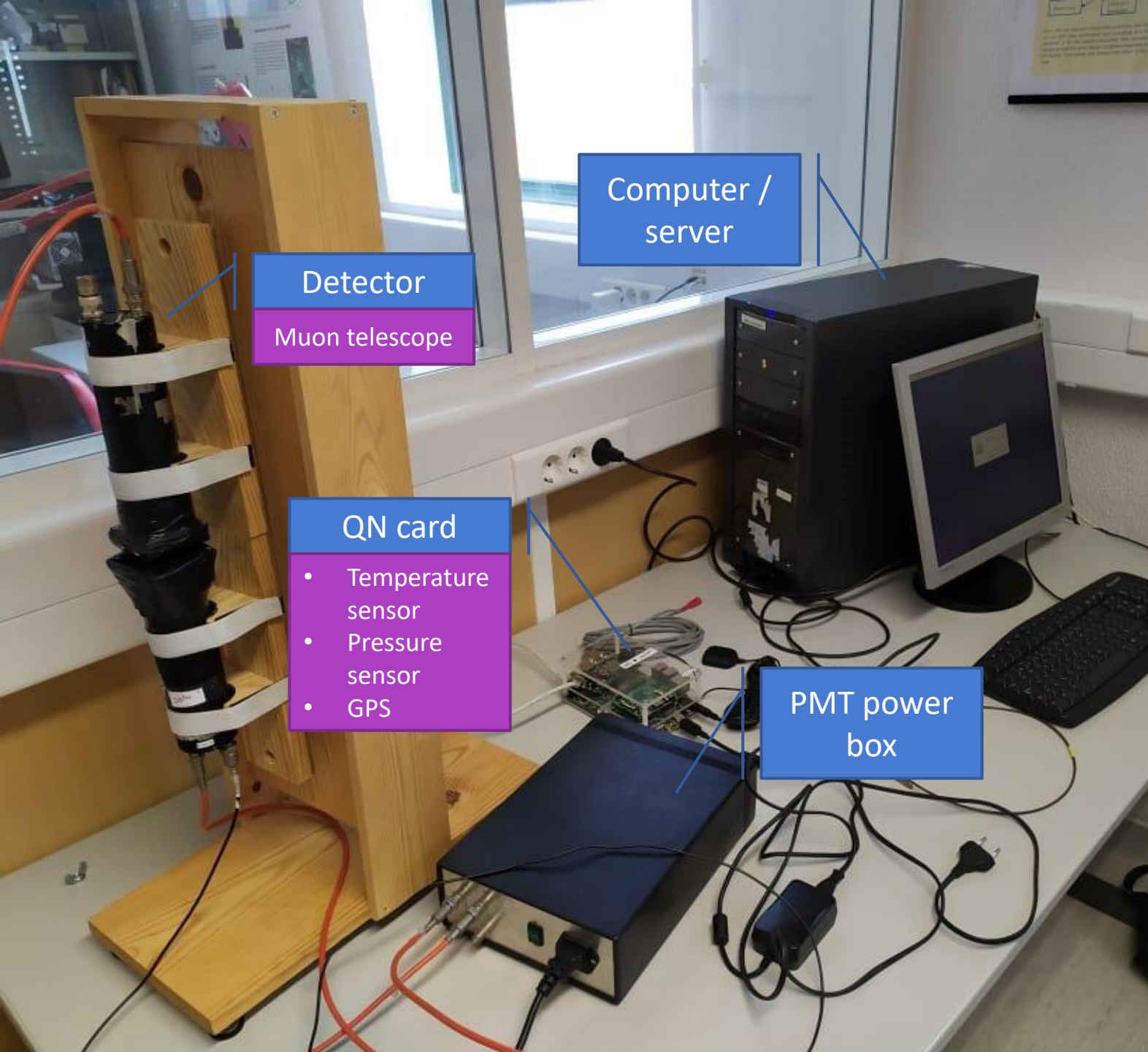
LabRC: Muon Telescope Acquisition

Final presentation

Ana Sofia Sousa

Francisco Rodrigues

Professor Fernando Barão



Detector

Muon telescope

Computer /
server

QN card

- Temperature sensor
- Pressure sensor
- GPS

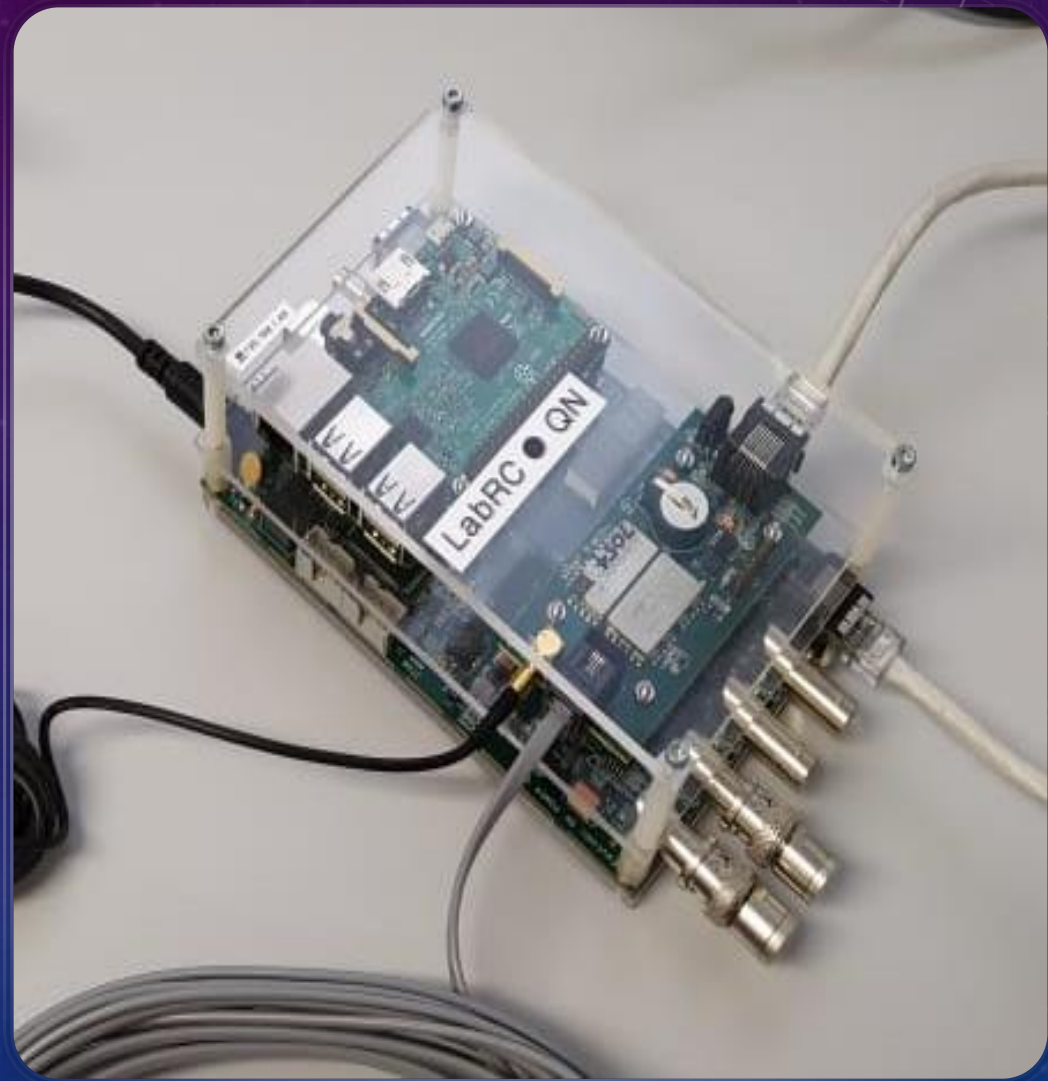
PMT power
box

OUR EXPERIMENTAL SETUP

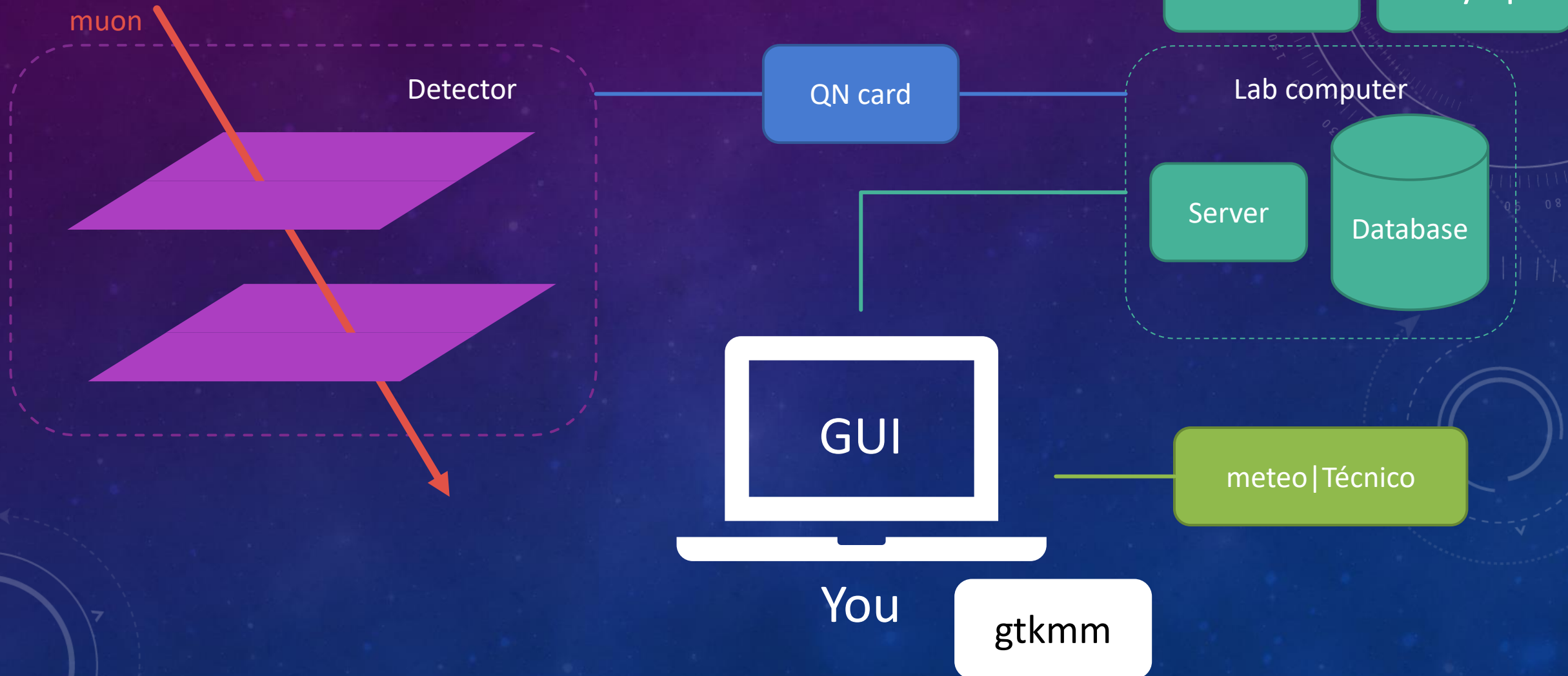
- ❑ Server running 24/7
- ❑ Accessible from any computer through a ssh tunnel
- ❑ Interact through Graphics User Interface or the console
- ❑ User can switch between live streaming and/or saving the events in the database
- ❑ Past events can be requested from database
- ❑ Configurations saved for run identification

NEEDS & GOALS

- Control the acquisition card (QN card)
- Control the PMT High Voltage
- Handle events
- Store and/or stream events
- Store configurations used in each run
- User friendly
- Available anywhere (remote access to the setup)



EVENT ACQUISITION SCHEME



The background is a dark blue gradient with several faint, light blue circular UI elements. These include concentric circles, dashed lines, and scales with numerical values (e.g., 150, 160, 170, 180, 190, 200, 210, 220, 230, 240, 250, 260). Some circles have arrows indicating direction, and others have partial segments, suggesting interactive or dynamic components.

GRAPHICAL USER INTERFACE

CLIENT PROGRAMS – GUI AND CONSOLE



GRAPHICAL USER INTERFACE

Window mode

The screenshot shows the LabRC Muon Telescope Acquisition GUI in window mode. The interface is dark-themed with a blue header bar. The main content area is divided into several sections:

- Header:** Help, About, QN card settings, Server settings.
- Server:** IP address: 127.0.0.1, Status: **not connected**, QN card status: **unknown**. Includes a "Connect:" button with a globe icon.
- Acquisition status:** Detector: **unknown**, GPS position: **unknown**, Temperature: **unknown**, Pressure: **unknown**.
- Detector:** Start: [Play button], Stop: [Stop button]. Includes a dropdown menu.
- Save settings:** DB (server): , JSON (local): .
- Muon rate / Meteorology:** A large dark area with a faint diagram of a detector structure.
- Request data:** [Up/Down arrow icon].
- Plot settings:** Current selection: Nothing yet..., Clear selection: [Clear icon], Show dots: , Show line: , Change main colour: [Color picker].

The screenshot shows the LabRC Muon Telescope Acquisition GUI in console mode. The interface is dark-themed with a blue header bar. The main content area is a large text area displaying the following text:

```
QN card: connected & not running Enable auto scroll: 
```

```
TL L0=30 L1=30 L2=20 L3=30
DC
DC C0=13 C1=70 C2=91 C3=00
DT
DT T0=00 T1=99 T2=9B T3=00
VG
VG=0
TH
TH TH=27.5
BA
BA 1520
Adjust this count 1520 to calibrate sensor in mBar.
mBar now reads = 1027.9 (use cmd 'SA' when done)
TL
TL L0=30 L1=30 L2=20 L3=30
DC
DC C0=13 C1=70 C2=91 C3=00
DT
DT T0=00 T1=99 T2=9B T3=00
VG
VG=0
```

The text area is scrollable, and the bottom of the window shows a command prompt with the text `\exit`.

Console mode

WINDOW MODE – LIVE ACQUISITION

The screenshot shows the LabRC Muon Telescope Acquisition software interface. The window title is "LabRC: Muon Telescope Acquisition". The interface is divided into several sections:

- Navigation:** Help, About, QN card settings, Server settings.
- LabRC Logo:** A stylized logo with the text "LabRC" and a graphic of lines radiating from a point.
- Mode Selection:** Two buttons: "Live mode" (highlighted) and "Database mode".
- Server Information:**
 - IP address: 127.0.0.1
 - Status: **connected**
 - QN card status: **connected**
- Acquisition Status:**
 - Detector: **not running**
 - GPS position: **unknown**
 - Temperature: **27,60 C**
 - Pressure: **1027,70 mbar**
- Detector Control:**
 - Start:
 - Stop:
 - Custom:
- Save settings:**
 - DB (server):
 - JSON (local):
- Plotting:**
 - Muon rate:
 - Meteorology:
 - Request data:
 - Plot settings:
 - Current selection: Nothing yet...
 - Clear selection:
 - Show dots:
 - Show line:
 - Change main colour:

DETECTOR CONFIGURATIONS

Pre-set sensor modes:

- Muon telescope
- Muon decay
- Cosmic shower

QN card settings

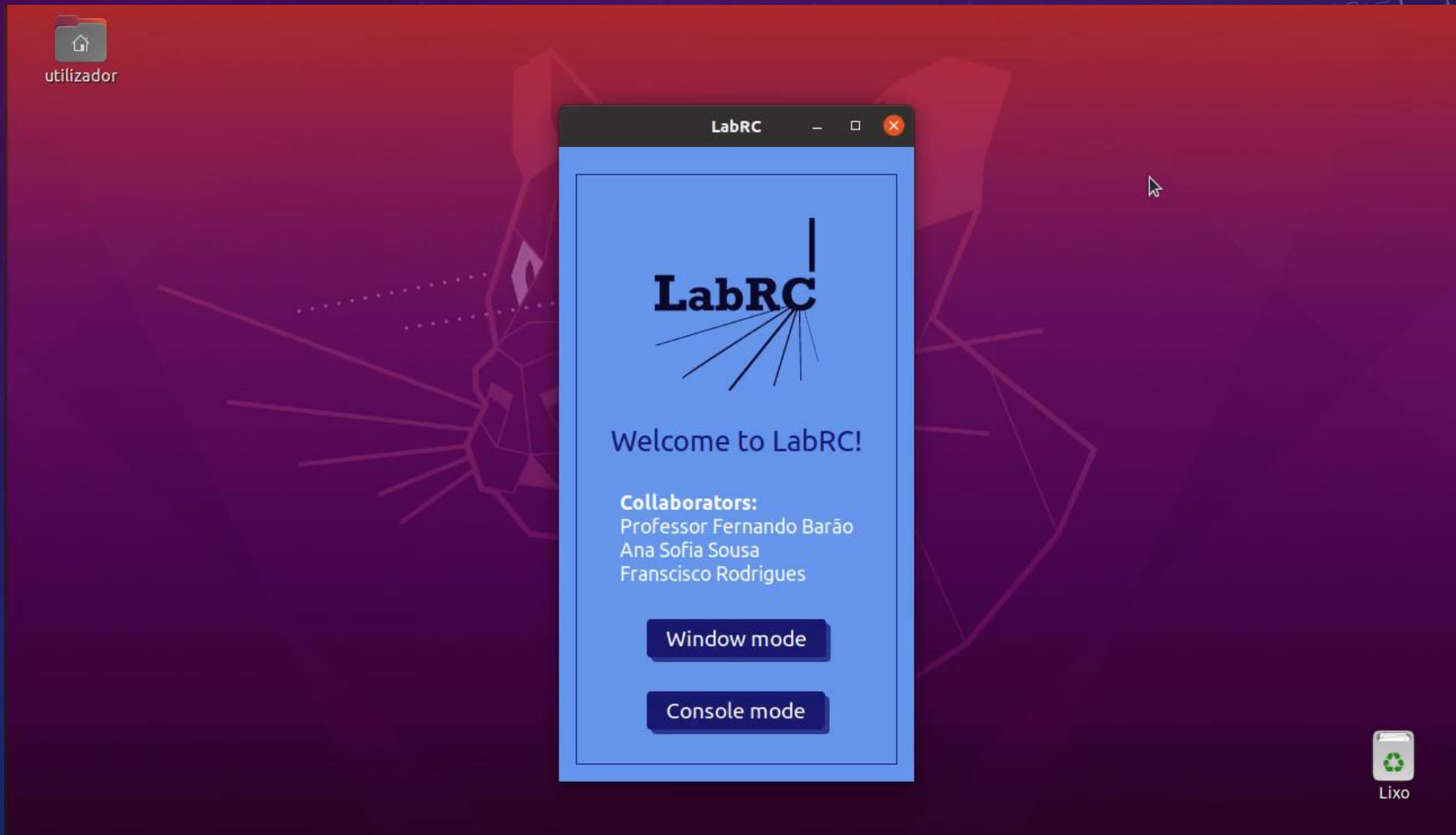
Settings name: Muon telescope

	Channel 1	Channel 2	Channel 3	Channel 4	
Threshold (mV)	500 - + <input checked="" type="checkbox"/>	500 - + <input checked="" type="checkbox"/>	500 - + <input type="checkbox"/>	500 - + <input type="checkbox"/>	
Time delay (x10 ns):	2 - +				
Time width (x10 ns):	1 - +				
Trigger mode:	2-fold				
Veto:	<input checked="" type="radio"/> None	<input type="radio"/> Channel 1	<input type="radio"/> Channel 2	<input type="radio"/> Channel 3	<input type="radio"/> Channel 4

Status updates: Disabled

Cancel OK

LIVE MODE DEMONSTRATION



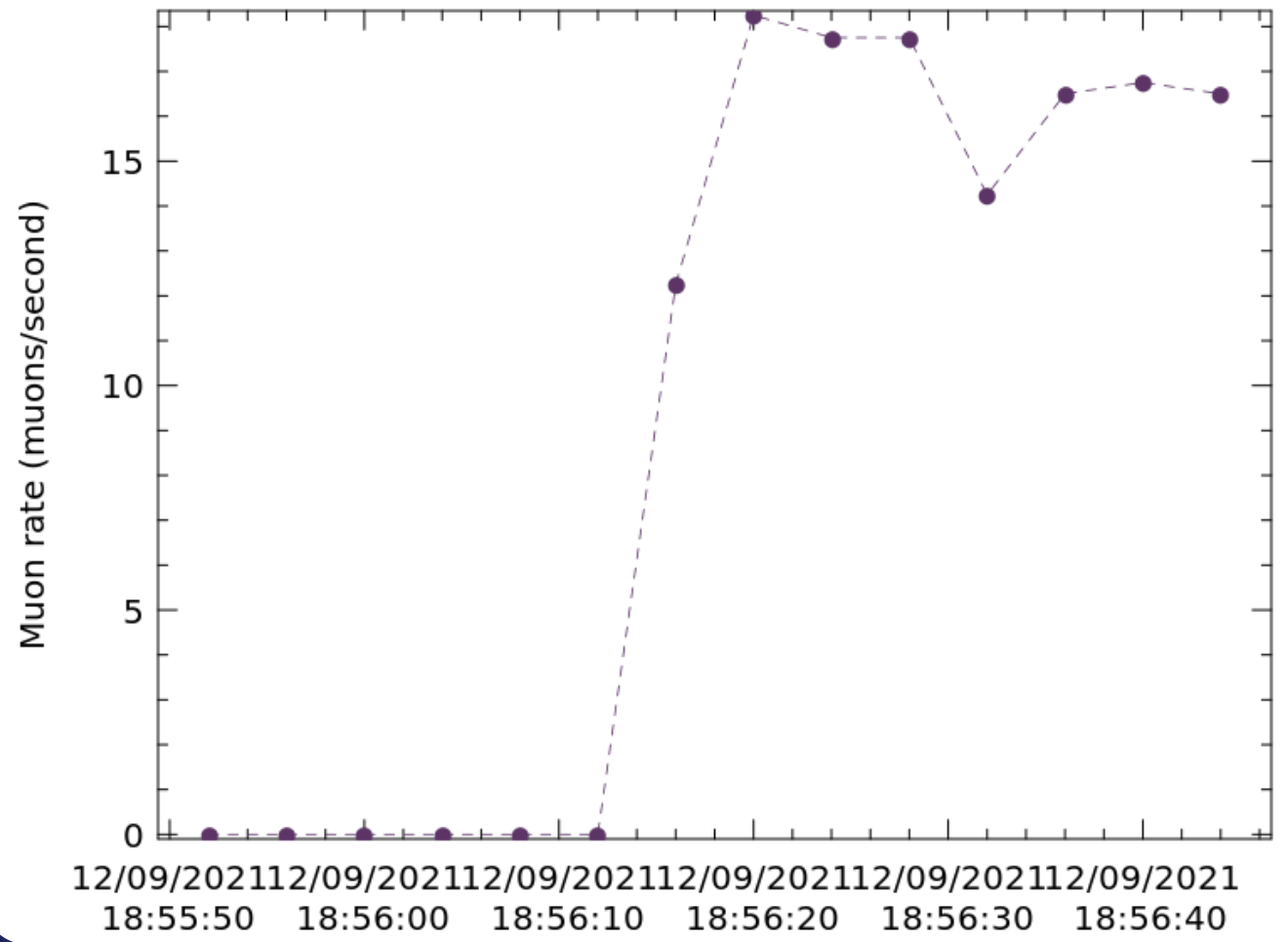
MUON RATE

Export options include:

- png (as pictured)
- eps
- pdf

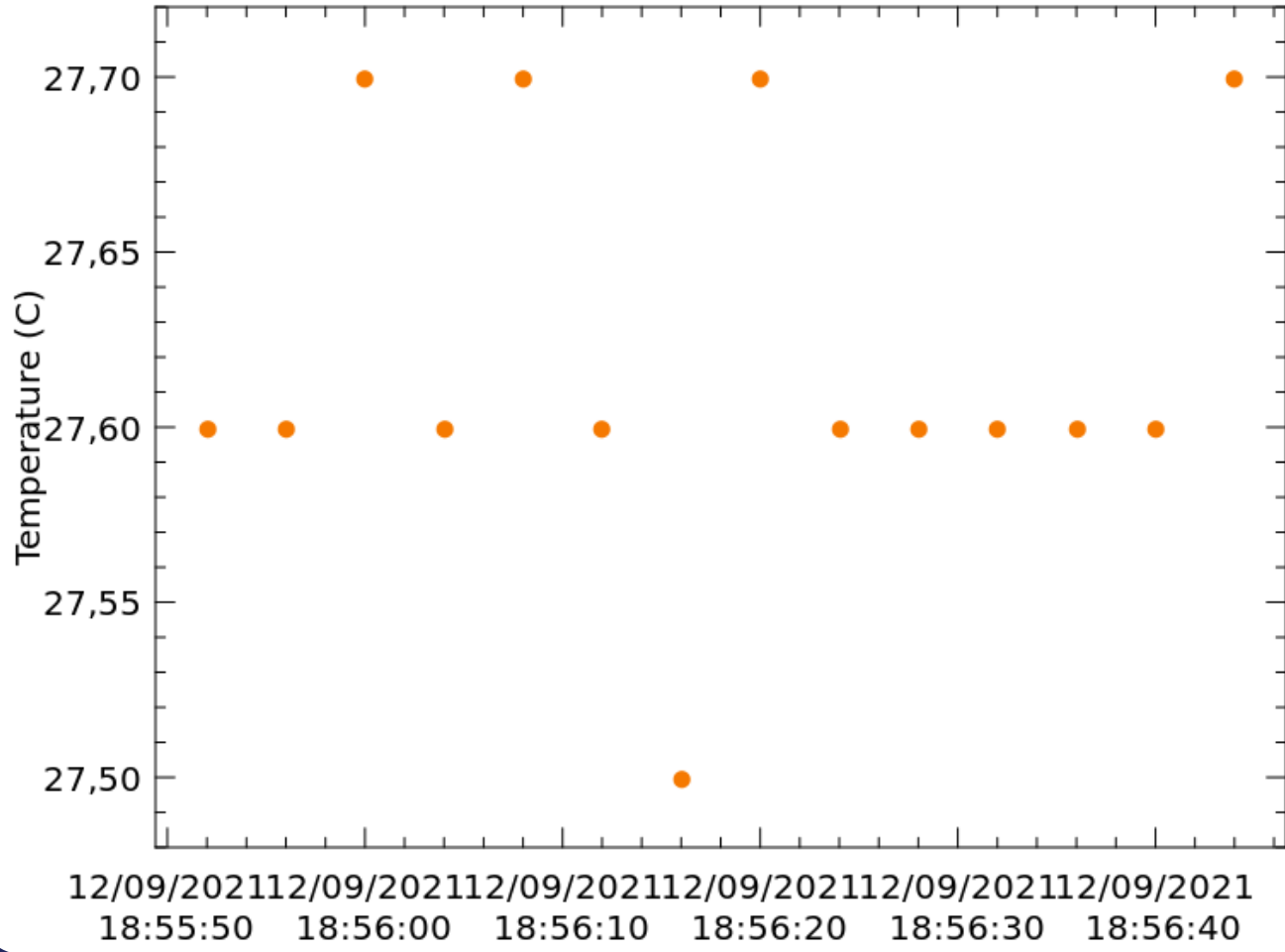
Printing is also supported

Muon rate as a function of time



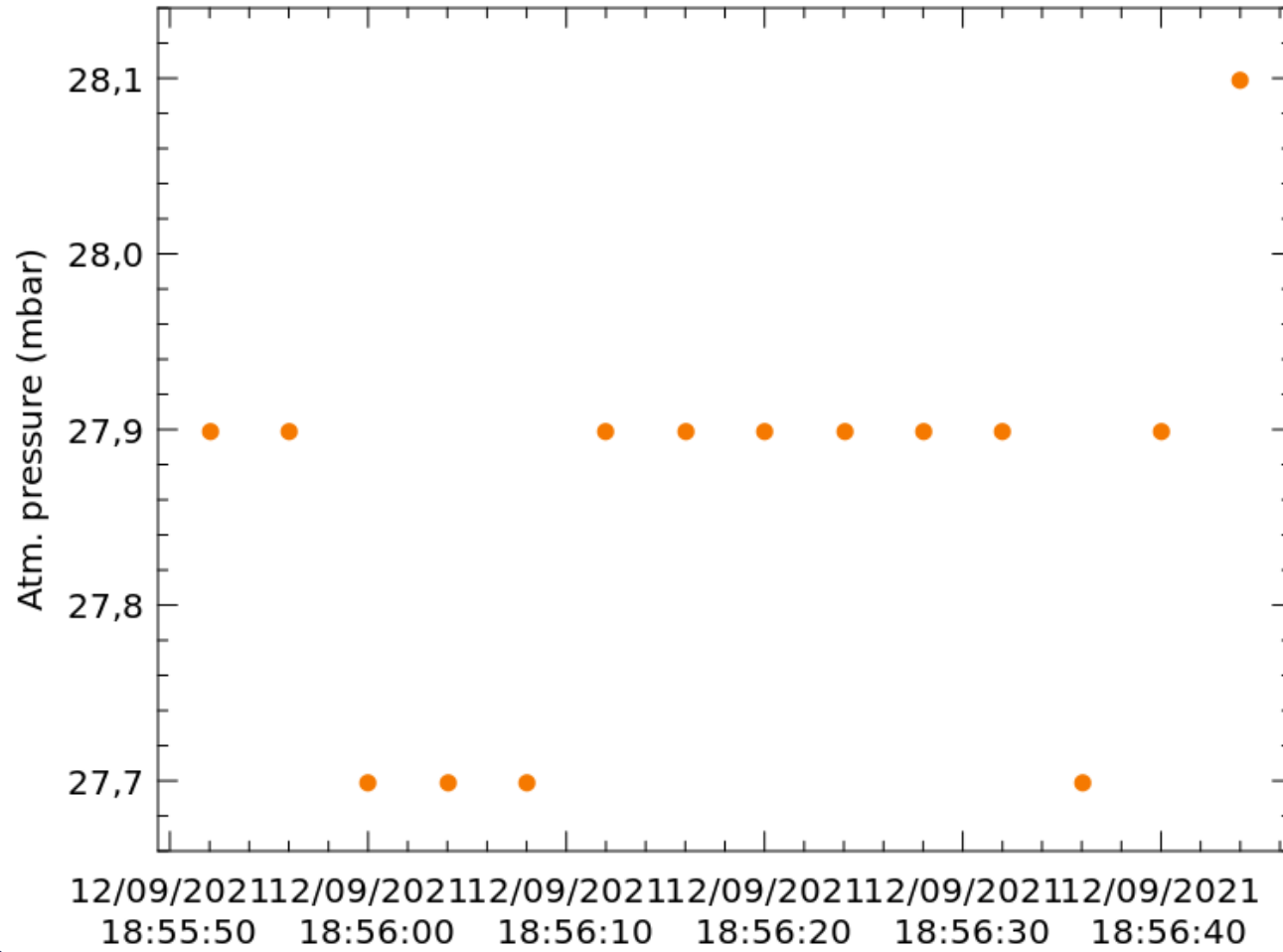
TEMPERATURE

Temperature as a function of time



PRESSURE

Pressure as a function of time (-1000 mbar)



JSON DATA OUTPUT

```
1  {
2    "object_type": "configuration",
3    "values": [
4      {
5        "time_width": 145,
6        "time_delay": 2,
7        "status_mode": 0,
8        "status_time": 1,
9        "trigger": 0,
10       "veto": 0,
11       "thresholds": [
12         30,
13         30,
14         20,
15         30
16       ],
17       "channel_states": [
18         1,
19         1,
20         0,
21         0
22       ]
23     }
24   ]
25 }
```

```
27  {
28    "object_type": "event",
29    "values": [
30      {
31        "timestamp": 1631472975,
32        "nanoseconds": 343486931,
33        "id": 0,
34        "has_rising_start": [
35          false,
36          true,
37          false,
38          false
39        ],
40        "edges": [
41          [],
42          [
43            3,
44            36.75
45          ],
46          [],
47          []
48        ]
49      }
50    ]
51 }
```

DATABASE MODE – RUN SELECTION

LabRC: Muon Telescope Acquisition

Help About QN card settings Server settings

LabRC

Live mode
Database mode

Server
IP address: 127.0.0.1
Status: **connected**
QN card status: **connected**

Muon rate Meteorology Comparison

Run selection parameters

Start date: End date:

Start date: 01/09/2021, 00:00
End date: 12/09/2021, 00:00

All

Cancel

Request data:

Plot settings

Current selection
Nothing yet...

Clear selection:

Show dots:
Show line:
Change main colour:

Run selection

Showing: All

Name		Duration (s)	Selected
Muon telescope	<input type="button" value="i"/>	3364	<input type="checkbox"/>
Muon decay	<input type="button" value="i"/>	8610	<input type="checkbox"/>
Cosmic shower	<input type="button" value="i"/>	2787	<input checked="" type="checkbox"/>
Fernando Pessoa	<input type="button" value="i"/>	2336	<input type="checkbox"/>
Muon decay	<input type="button" value="i"/>	21451	<input type="checkbox"/>
Experiment 14	<input type="button" value="i"/>	4687	<input checked="" type="checkbox"/>
Muon telescope	<input type="button" value="i"/>	11015	<input type="checkbox"/>

Cancel OK

DATABASE MODE – RUN SELECTION

LabRC: Muon Telescope Acquisition

Help About QN card settings Server settings

LabRC

Live mode
Database mode

Server
IP address: 127.0.0.1
Status: **connected**
QN card status: **connected**

Muon rate Meteorology Comparison

Run selection parameters

Start date: End date:
Start date: 01/09/2021, 00:00
End date: 12/09/2021, 00:00

Cosmic shower

Cancel

Request data:

Plot settings

Current selection
Nothing yet...

Clear selection:

Show dots:
Show line:
Change main colour:

Run selection

Showing: Cosmic shower

Name	Duration (s)	Selected
Cosmic shower	5215	<input type="checkbox"/>
Experiment 14	2947	<input type="checkbox"/>
Broccoli 754	17269	<input type="checkbox"/>

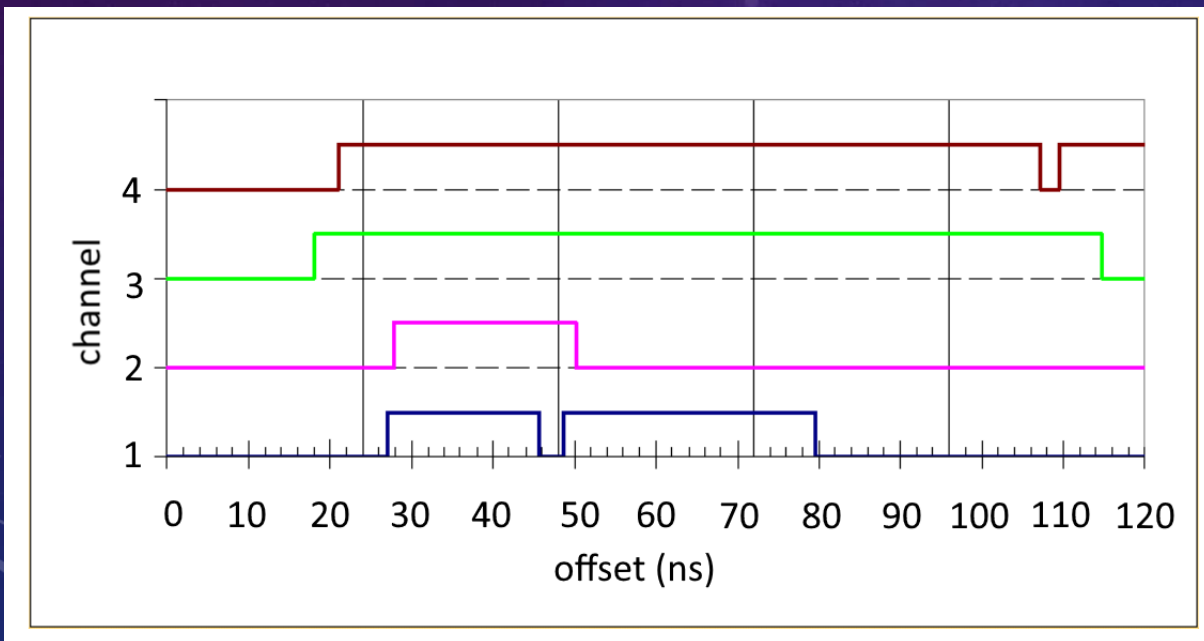
Cancel OK

A server rack with blue lighting and digital overlays. The image shows a stack of server units with blue perforated front panels. The background is dark blue with bokeh light effects in shades of yellow and green. Overlaid on the image are various digital graphics: white circles, dashed lines, and numbers (140, 150, 160, 170, 180, 190, 200, 210, 220, 230, 240, 250, 260) arranged in a circular pattern on the left side. The text 'SERVER & DATABASE' is centered in white, sans-serif font.

SERVER & DATABASE

WHAT IS AN EVENT?

An event consists of an initial "trigger" timestamp, 4 booleans describing whether the input starts with a rising or a falling edge, and 4 vectors of offsets corresponding to the rising and falling edges of each channel.



This would be represented as:

Timestamp: [unix timestamp]

Rising-Start: {true, true, true, true}
(all inputs start with a RE)

Edges: {

{27, 45.75, 48.75, 79.5},

(channel 1)

{27.75, 50.25},

(channel 2)

{18, 114.75},

(channel 3)

{21, 107.5, 109.5}

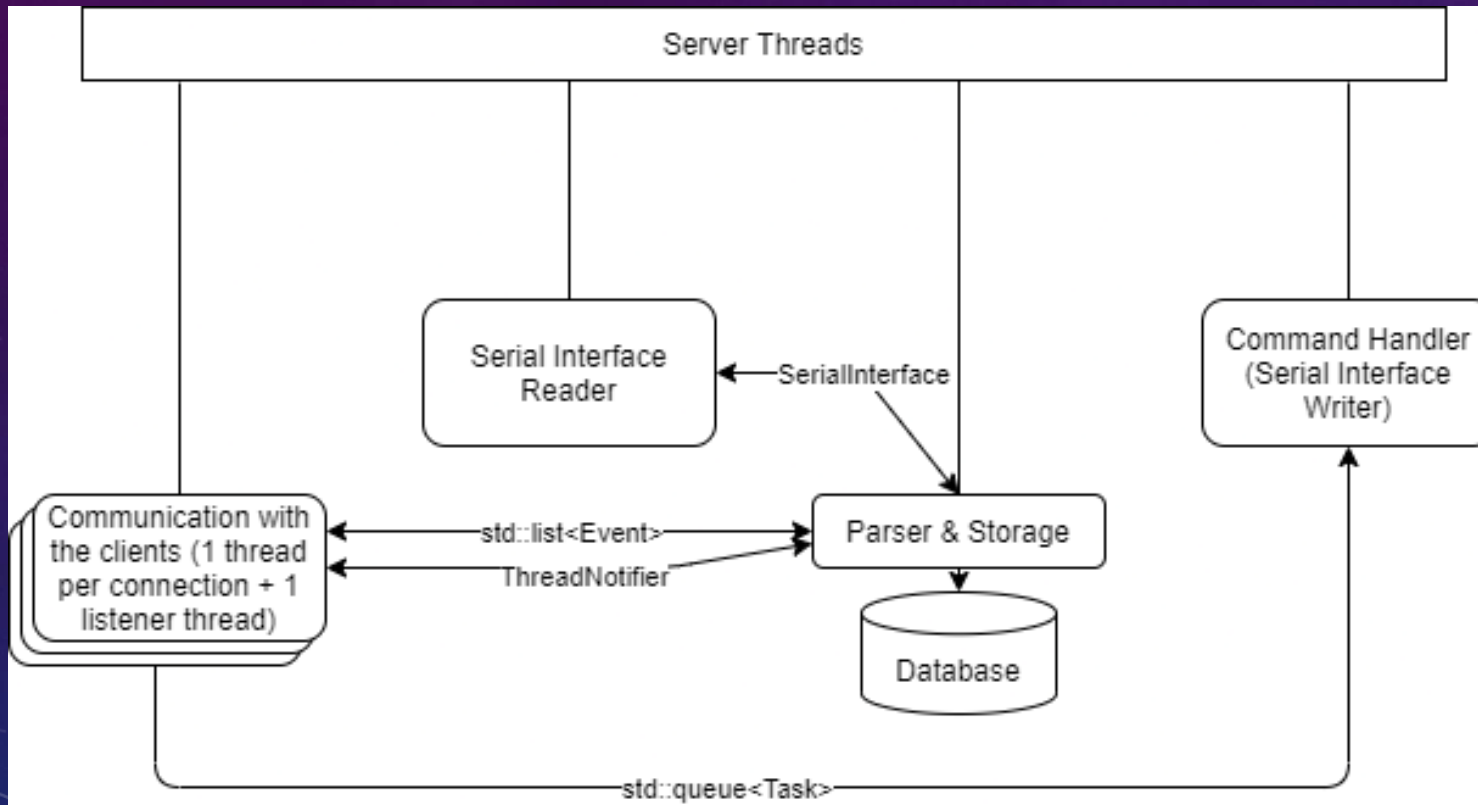
(channel 4)

}

green - rising edge

red - falling edge

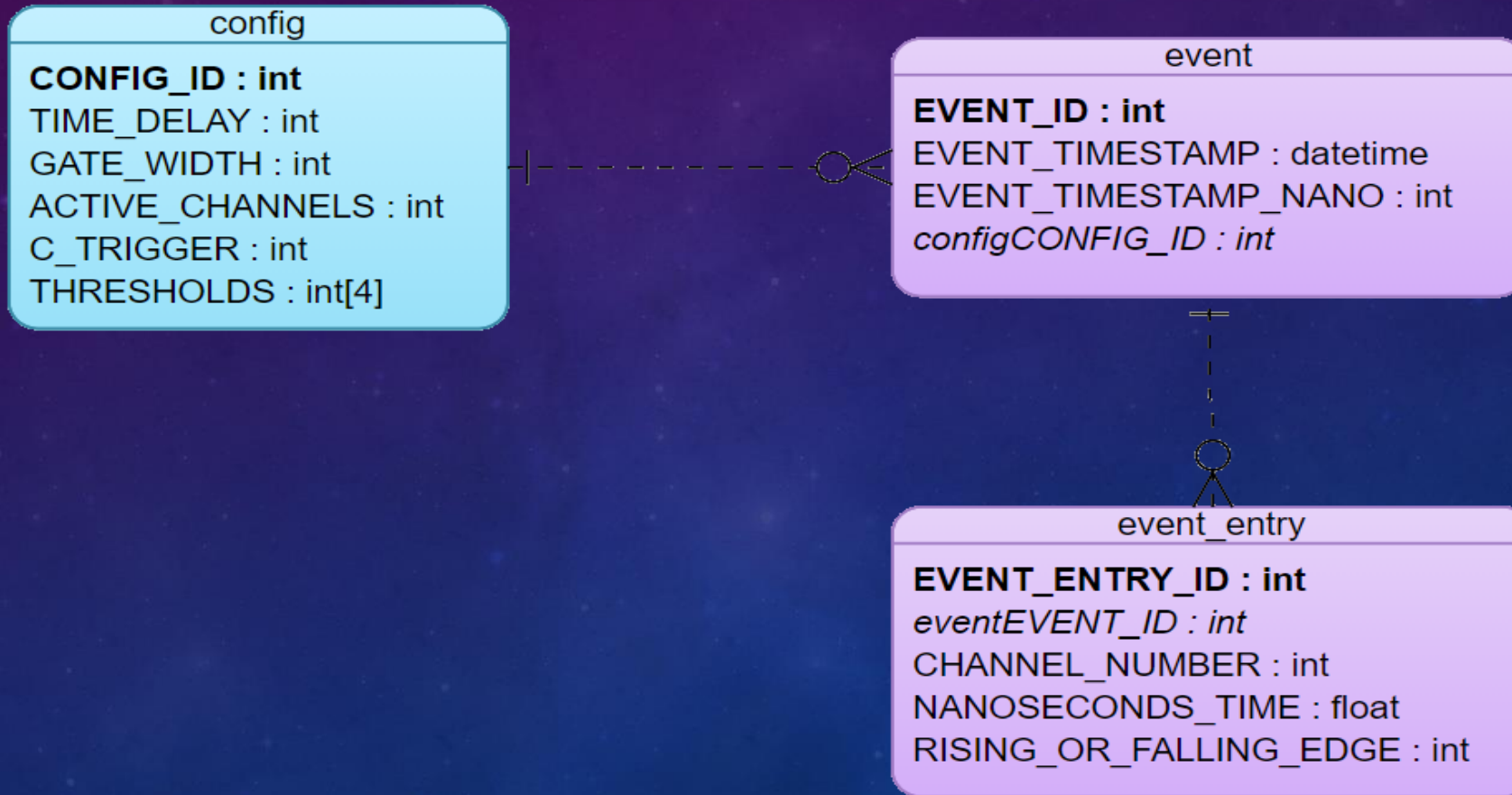
SERVER DETAILS



The server supports:

- ❑ Arbitrary number of clients
- ❑ LIVE/DATABASE mode
- ❑ Whitelists
- ❑ Client limiting

DATABASE STRUCTURE



SERVER OUTPUT

The server outputs its logs and debug information to *stdout*, but by using a command such as "rotatelogs" we can pipe that to a file.

```
Ircuser@labrc-lippad:~/LabRC/I x + v - □ X
[2021-09-12 18:26:08.585] (parser_thread) BA
[2021-09-12 18:26:08.597] (parser_thread) BA 1520
[2021-09-12 18:26:08.598] (parser_thread) Adjust this count 1520 to calibrate sensor in mBar.
[2021-09-12 18:26:08.598] (parser_thread) mBar now reads = 1027.9 (use cmd 'SA' when done)
[2021-09-12 18:26:08.979] (parser_thread) 62A3C5DD AF 00 00 00 00 00 00 00 6253A02C 172545.010 270102 V 00 0 +0044
[2021-09-12 18:26:12.368] (parser_thread) 67B0D179 BD 00 00 00 00 00 00 00 66CC08EC 172548.002 270102 V 00 0 +0036
[2021-09-12 18:26:13.552] (192.168.1.103:41234-client-thread) Adding task...
[2021-09-12 18:26:13.552] (192.168.1.103:41234-client-thread) Added!
[2021-09-12 18:26:13.552] (192.168.1.103:41234-client-thread) Channel: 7/8
[2021-09-12 18:26:13.552] (consumer-thread) Got task
[2021-09-12 18:26:13.552] (consumer-thread) Got task!
[2021-09-12 18:26:13.552] (consumer-thread) [SERIAL] TH
[2021-09-12 18:26:13.554] (parser_thread) TH
[2021-09-12 18:26:13.555] (parser_thread) TH TH=27.5
[2021-09-12 18:26:13.634] (192.168.1.103:41234-client-thread) Adding task...
[2021-09-12 18:26:13.634] (192.168.1.103:41234-client-thread) Added!
[2021-09-12 18:26:13.634] (192.168.1.103:41234-client-thread) Channel: 6/8
[2021-09-12 18:26:13.634] (consumer-thread) Got task
[2021-09-12 18:26:13.634] (consumer-thread) Got task!
[2021-09-12 18:26:13.634] (consumer-thread) [SERIAL] BA
[2021-09-12 18:26:13.636] (parser_thread) BA
[2021-09-12 18:26:13.647] (parser_thread) BA 1520
[2021-09-12 18:26:13.648] (parser_thread) Adjust this count 1520 to calibrate sensor in mBar.
[2021-09-12 18:26:13.648] (parser_thread) mBar now reads = 1027.7 (use cmd 'SA' when done)
[2021-09-12 18:26:14.073] (192.168.1.103:41236-client-thread) Lost connection with 192.168.1.103:41236!
[2021-09-12 18:26:14.073] (192.168.1.103:41234-client-thread) Lost connection with 192.168.1.103:41234!
[2021-09-12 18:26:15.338] (parser_thread) 6C1B3092 80 3C 00 00 00 00 00 00 6B4471AC 172551.010 270102 V 00 0 +0044
[2021-09-12 18:26:15.339] (parser_thread) 6C1B3095 80 25 00 00 00 00 00 00 6B4471AC 172551.010 270102 V 00 0 +0044
[2021-09-12 18:26:18.065] (parser_thread) 702E0B6B A7 00 00 00 00 00 00 00 6FBCDA6C 172554.002 270102 V 00 0 +0036
[2021-09-12 18:26:18.415] (parser_thread) 70B360B0 80 3E 00 00 00 00 00 00 6FBCDA6C 172554.002 270102 V 00 0 +0036
```

DOXYGEN DOCUMENTATION

The screenshot shows the Doxygen documentation interface for LabRC. The top navigation bar includes the LabRC logo, a search box, and links for Main Page, Related Pages, Namespaces, and Classes. A left sidebar lists various project components, with 'Muon telescope: development of acquisition and analysis full chain' selected. The main content area features a title, an introduction section, and three paragraphs of text describing the project's goals and the physical processes involved in muon detection.

LabRC

Main Page Related Pages Namespaces Classes

Search

LabRC

Muon telescope: development of acquisition and analysis full chain

- Introduction
- Muon telescope: overall description
- Acquisition and analysis full chain
- Tasks and tools
- References
- Database Storage
- Data format
- GUI documentation
- Networking
- QuarkNet Card
- Serial Interface
- Software
- Namespaces
- Classes

Muon telescope: development of acquisition and analysis full chain

Introduction

Cosmic muons are produced in the upper atmosphere. The interaction of primary cosmic rays, comprised essentially of protons, with air molecules gives rise to both pions and kaons (mesons), both of which have very short life times, and subsequently decay.

The purpose of this project is to operate a portable muon telescope and monitor the time variability of the detected cosmic muons. This variability is due to variations in the primary cosmic rays that arrive to Earth.

Since the production of mesons, which are the particles from which the muons stem from, happens at altitudes of around 10 kilometres (that is, the lower stratosphere), the conditions of the terrestrial atmosphere will have an impact on the muons' arrival at Earth. The upper atmosphere will have an effect on the meson production through the mean free path of primary cosmic rays.

After the muons are generated, their arrival to Earth can be affected by meteorological parameters such as atmospheric pressure or temperature. Thus, these meteorological parameters will be monitored in parallel with the muon detection rate, so as to evaluate any correlation.

Generated by [doxygen](#) 1.8.17

FURTHER USES

- ❑ The source programs (header files and C++ files) were constructed in a way to allow for easy interpretation and re-implementation.
- ❑ There are both broad classes and niche classes, so a new client program can be constructed without starting from scratch and taking advantage of one of different levels of implementation.
- ❑ The networking related classes (server and client) are 100% independent from the GUI.
- ❑ Great care was taken to ensure proper documentation of every file, class and function, as well as how to install and run the programs developed. There are also several small test programs where implementation can be easily understood and quick tests in communication are performed.

What follows is a very simple example of communication with the server using our “LRCClient” class.

```
#include "LRCClient.h"  
#include "Common.h"  
#include <iostream>
```

Includes

```
int main() {
```

```
LRC::LRCClient client;
```

```
try {
```

```
    client.reconnect("127.0.0.1", "8000");
```

```
}
```

```
catch (Networking::network_exception &exception)
```

```
{
```

```
    std::cout << "Unable to connect to server." << std::endl;
```

```
    return 1;
```

```
}
```

Connect to the
server

```
LRC::Result result;
```

```
auto v1 = client.get_voltage(LRC::HVChannel::CHANNEL1, &result);
```

```
if (result != LRC::Result::SUCCESS || !v1.has_value()) {  
    std::cout << "Error getting voltage for channel 1" << std::endl;  
    return 1;  
}
```

Get channel 1 voltage

```
auto v2 = client.get_voltage(LRC::HVChannel::CHANNEL2, &result);
```

```
if (result != LRC::Result::SUCCESS || !v2.has_value()) {  
    std::cout << "Error getting voltage for channel 2" << std::endl;  
    return 1;  
}
```

Get channel 2 voltage

```
std::cout << "Channel 1 voltage: " << v1.value() << " V" << std::endl;
```

```
std::cout << "Channel 2 voltage: " << v2.value() << " V" << std::endl;
```

Output results

```
}
```


DOXYGEN DOCUMENTATION

LRCWindow Class Reference

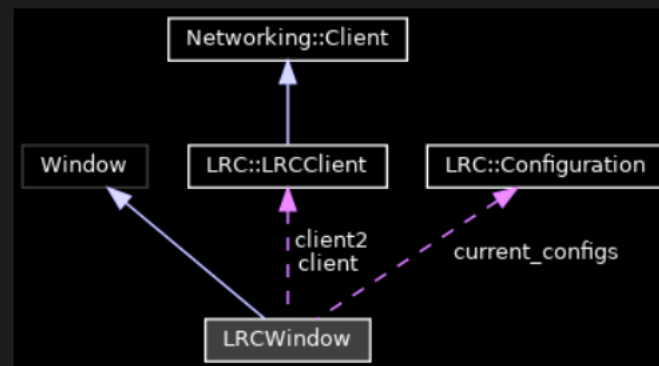
```
#include <LRCWindow.h>
```

Inheritance diagram for LRCWindow:



[legend]

Collaboration diagram for LRCWindow:



[legend]

◆ get_string_delim()

```
std::string SerialInterface::get_string_delim ( const std::string & delim = "\n" ,  
                                              ssize_t      maxsize = -1  
                                              )
```

Read bytes from the buffer until a 'delimiter' is hit, or the number of bytes read is greater than maxsize.

Warning

This operation blocks until enough data is available.

Parameters

delim is the delimiter - this function stops reading more bytes after it encounters this sequence.
maxsize is the maximum number of bytes to be read. If maxsize=-1, then there is no upper limit.

Returns

This function returns a string with the bytes read from the internal buffer, up to but not including the delimiter.

Note

As an example, if delim="\n" and next bytes stored in the internal buffer are "hello world\n welcome\n", this will return "hello world"



About

This program was developed as part of an internship conducted at LIP, Técnico, University of Lisbon, during the summer of 2021.

Professor Fernando Barão, Técnico Physics Dept.

Students:

Ana Sofia Sousa, MEFT

Francisco Rodrigues, MEEC



LABORATÓRIO DE INSTRUMENTAÇÃO
E FÍSICA EXPERIMENTAL DE PARTÍCULAS



TÉCNICO
LISBOA

OK

THANK YOU

REFERENCES

- <https://quarknet.org>
- <https://www.beej.us/guide/bgnet/html/>
- <https://www.technical-recipes.com/2014/getting-started-with-client-server-applications-in-c/>
- <https://www.boost.org/>
- <https://developer.gnome.org/gtkmm-tutorial/3.24/>
- <https://tschoonj.github.io/gtkmm-plplot/index.html>
- <http://caboruivo.tecnico.ulisboa.pt:64104/api>