

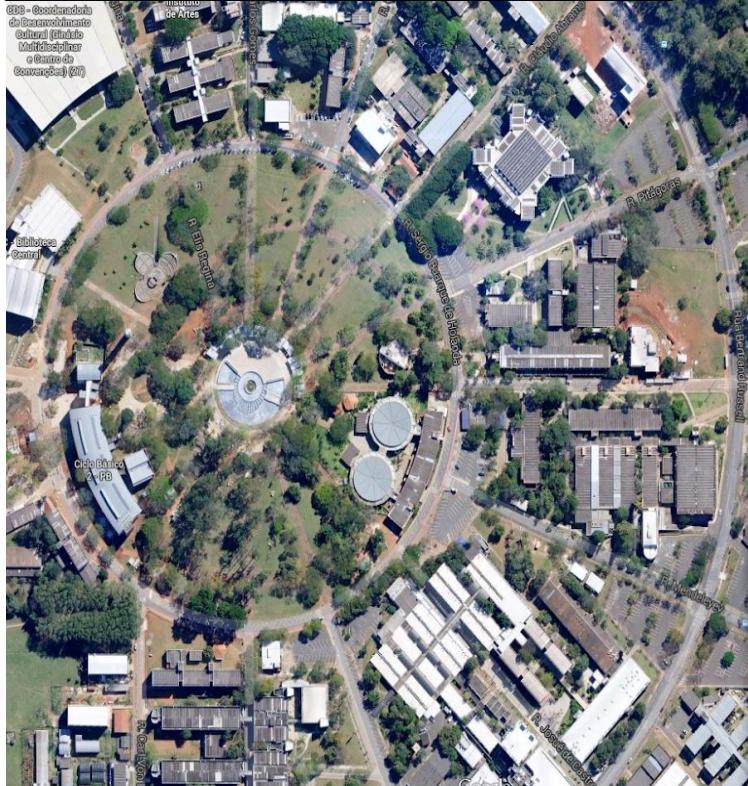
# The new data acquisition system of the TANCA detector

**Anderson Campos Fauth**

**Instituto de Física Gleb Wataghin - Unicamp**

# The Tanca at Unicamp

University of Campinas



Tanca

Tanca is a nickname for the TANk detector of CAMpinas

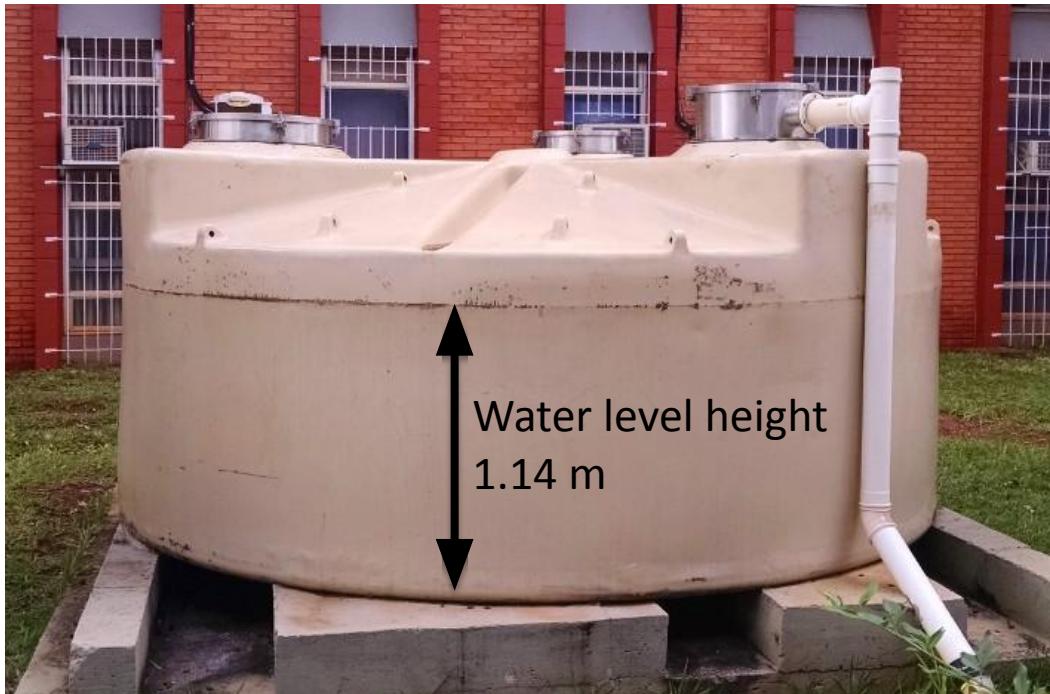
Campinas, SP, Brazil  
22° 54' S, 47° 03 N  
640 meters a.s.l.



# The TANCA goals

- Uses the flux of galactic cosmic rays modulated by transient solar ejecta (CME) and changes in the global structure of the heliospheric and terrestrial magnetic fields to study space weather
- Short-term modulations are associated with interplanetary transient perturbations, causing phenomena like Forbush Decrease
- Long-term modulations in the cosmic ray flux are associated with the eleven years solar cycle
- Academics
  - Muon decay, experiment for undergraduate and graduate students
  - Detector physics
  - Interaction of radiation with matter
  - Construction and characterization of particle detectors
  - Monte Carlo simulations of detectores and extensive air showers

# The Tanca detector



Photonis XP1805

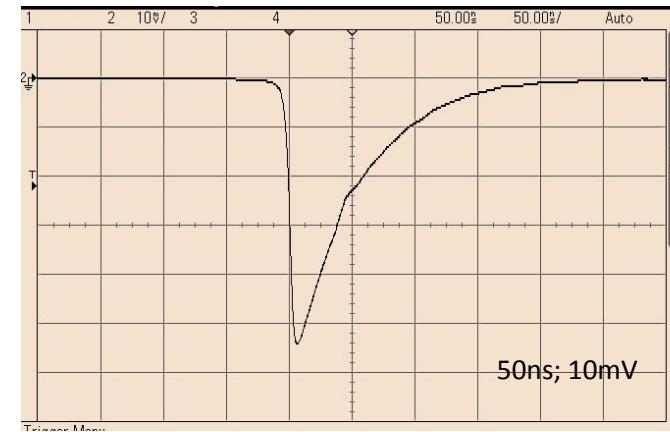


Voltage-divider, HV power supply

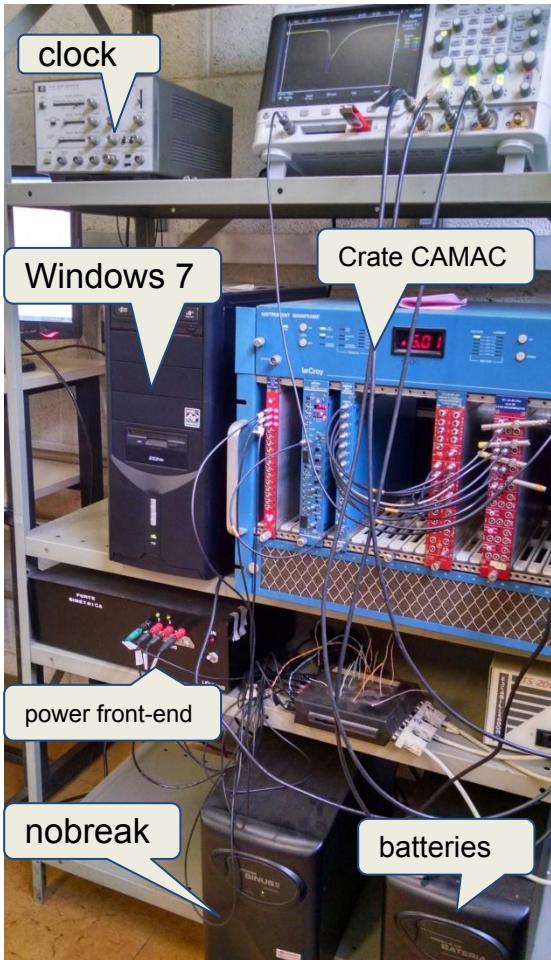


- The detector is a cylindrical polyethylene tank
- 3 PMT 9" Photonis XP1805
- Liner Tyvek
- 11400 liters of pure water
- Cherenkov light
- $\text{Area}_{\text{vertical}} = 10 \text{ m}^2$

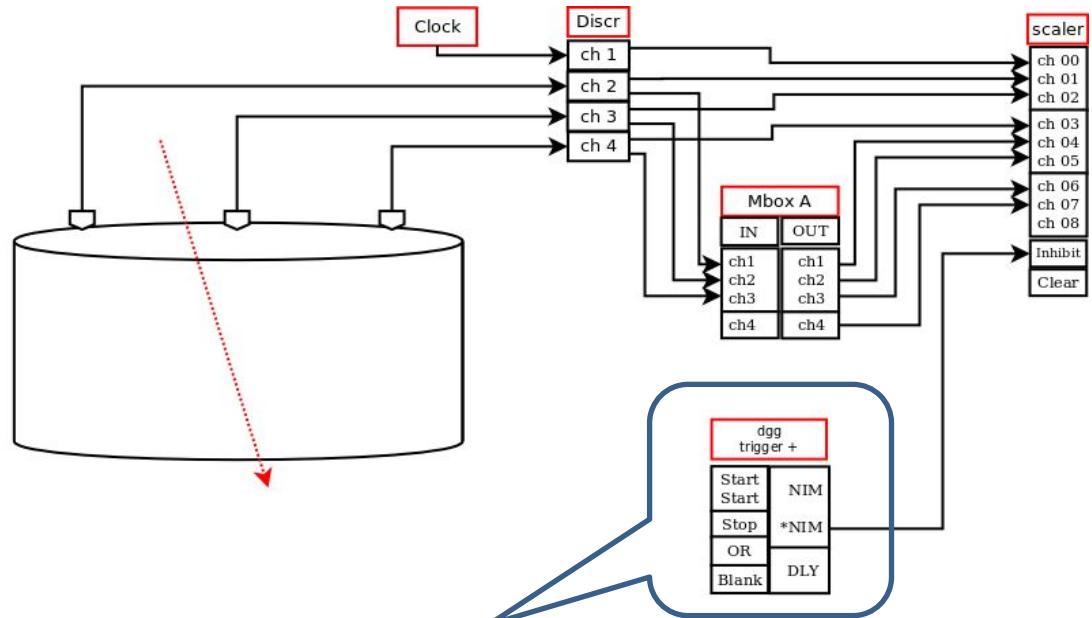
Muon pulse



# The old Tanca DAq



CAMAC/NIM modules: LeCroy Scaler 2551 (100 MHz); Logic Unit CAEN; Discriminator CAEN; Dual Gate Generator LeCroy 2323A; DAC CAEN



Precise (0.02sec) gate for the 1 sec time interval of the counts,  
future GLE detection?

HV1=1075V HV2=1125V; HV3=1250V; Limiares: L=-22mV; TC=300ns

ms CK S1 S2 S3 D12 D13 D23 TRI

2016 08 30 00 00 00 039 1472515200039

M 2016 08 29 23 56 1 2016 08 29 20 50 941 23.1 48.8 0.0 23.9 135 0.0 01

1472515201125 4996 2136 2174 2372 1860 1911 1924 1813

1472515202170 4996 2024 2055 2263 1727 1768 1817 1682

1472515203215 4996 2143 2199 2404 1882 1902 1965 1830

# The new DAq using a CAEN DT5720B digitizer



4 Channel

2 Vpp single ended input range

Resolution: **12 bit**

Sampling rate: 31.25 to **250 MS/s simultaneously** on each channel

Programmable event size and pre-post trigger adjustment

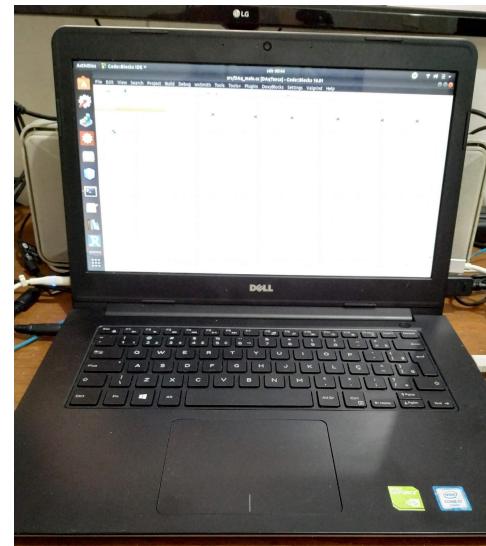
USB2.0 compliant up to **30 MB/s transfer** rate

Trigger Time Stamp 31-bit counter – 16 ns resolution – 17 s range

Common Trigger

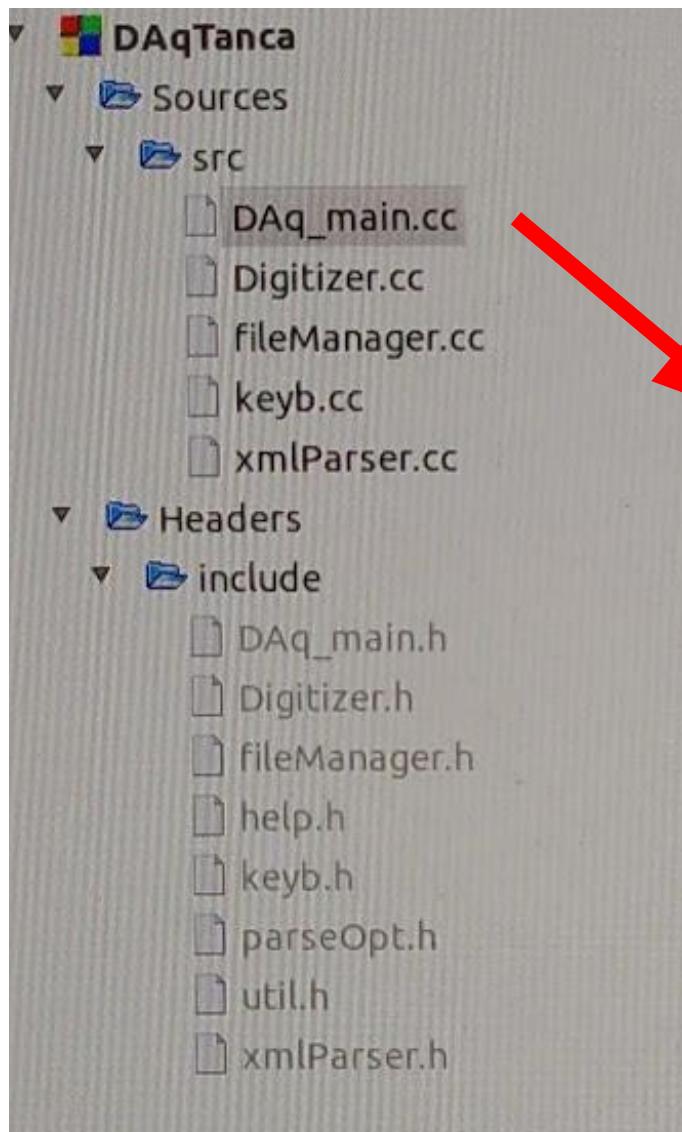
- External (signal on TRG\_IN)
- Software (by USB or Optical Link)
- **Self trigger** (internal threshold self-trigger)

# Experimental setup at home



# C++ DAq Tanca on Ubuntu 18 LTS

I modified the code provided by Samuel de Jong, from University of Victoria, CA for a CAEN V1730 digitizer (<https://github.com/samdejong86/CAEN-v1730-DAQ>)



```
1 //includes do C/C++
2 #include <iostream>
3 #include <getopt.h>
4
5 //includes deste código
6 #include "DAq_main.h"
7 #include "Digitizer.h"
8 #include "xmlParser.h"
9 #include "help.h"
10 #include "parseOpt.h"
11
12 volatile sig_atomic_t flag=0;
13
14 void my_function(int sig){ // can be called asynchronously
15     flag = 1; // set flag
16 }
17
18
19
20 int main(int argc, char *argv[]){
21     cout<<"======"<<endl;
22     cout<<"DAq using digitizer CAEN DT5720B"<<endl;
23     cout<<"Usage: DAqTanca [-v] -x Tanca_v01.xml"<<endl;
24     cout<<"======"<<endl;
25
26     //Linux signal()
27     signal(SIGINT, my_function); // SIGINT=Interrupt from keyboard
28
29     XmlParser settings = getOpt(argc, argv);
30
31     Digitizer digi(settings);
32
33     if(verbose){
34         cout<<digi;
35         digi.setVerbose(verbose);
36     }
37     bool digiOpen = digi.OpenDigitizer();
38
39     if(digiOpen){
40         digi.Readout();
41         digi.CloseDigitizer();
42     }
43
44 }
45
```

# Parser digitizer configuration

```
<xml>
<Active>
  <duration>
    0
  </duration>
  <reclen>
    70
  </reclen>
  <posttrigger>
    50
  </posttrigger>
  <ch0>
    1
  </ch0>
  <DCoffset0>
    56000
  </DCoffset0>
  <polarity0>
    NEGATIVE
  </polarity0>
  <threshold0>
    180
  </threshold0>
  <trslope0>
    NEGATIVE
  </trslope0>
  <ch1>
    1
  </ch1>
  <DCoffset1>
    56000
  </DCoffset1>
  <polarity1>
    NEGATIVE
  </polarity1>
  <threshold1>
    180
  </threshold1>
  <trslope1>
    NEGATIVE
  </trslope1>
  <ch2>
    1
  </ch2>
  <DCoffset2>
    56000
  </DCoffset2>
  <polarity2>
    NEGATIVE
  </polarity2>
  <threshold2>
    180
  </threshold2>
  <trslope2>
    NEGATIVE
  </trslope2>
  <triggermode>
    Maj1
  </triggermode>
</Active>
</xml>
```

## File XML

- duration:
  - 0 = **continuous, new file every full hour**
  - time **hh:mm:ss**
  - **number** triggers
- record length
- post trigger
- chanel ON
  - DC offset
  - Polarity
  - **Threshold**
  - **Threshold slope**
- **Trigger mode:** OR or **Maj1** (doubles and triple coincidence)
- **output file ROOT** CERN format **TTree**
  - **YYYY\_MM\_DD\_HH\_MMtanca.root**

# C++ Digitizer class

```
#include "Digitizer.h"
#include <iostream>
#include <bitset>

static CAEN_DGTZ IRQMode_t INTERRUPT_MODE = CAEN_DGTZ_IRQ_MODE_ROAK;
extern volatile sig_atomic_t flag;

+ Digitizer::Digitizer(XmlParser settings){}

+ void Digitizer::DefaultSettings(){}

+ bool Digitizer::OpenDigitizer(){}

+ void Digitizer::Readout(){}

+ void Digitizer::CloseDigitizer(){}

+ CAEN_DGTZ_ErrorCode Digitizer::ProgramDigitizer(){}

// Setting trigger Majority = 1 (any majority response coincidence)
+ CAEN_DGTZ_ErrorCode Digitizer::Maj1Programmer(){}

    // setup for an 'OR' trigger
+ CAEN_DGTZ_ErrorCode Digitizer::OrProgrammer{}

+ CAEN_DGTZ_ErrorCode Digitizer::WriteRegisterBitmask(uint32_t address, uint32_t data, uint32_t mask)

+ CAEN_DGTZ_ErrorCode Digitizer::Calibrate_DC_Offset()

+ CAEN_DGTZ_ErrorCode Digitizer::SetCorrectThreshold()

+ void Digitizer::CheckKeyboardCommands(){}

+ void Digitizer::startAcq(){}

+ void Digitizer::printOn(ostream & out) const{
    // overload of << operator
+ ostream& operator<<(ostream& os, const Digitizer& r) {
```

# C++ Filemanager class

```
1 #include "fileManager.h"
2 #include <sstream>
3 #include <stdint.h>
4 #include <sys/stat.h>
5
6 #include <algorithm>
7
8 +void fileManager::init(string filename="DT5720B.root", uint16_t EnableMask=0, int saveInt=100){
47
48
49 +void fileManager::OpenFile(){}
93
94
95 +void fileManager::CloseFile(){}
165
166
167 +void fileManager::addEvent(CAEN DGTZ EventInfo t *EventInfo, CAEN DGTZ UINT16 EVENT t *Event16){}
224
225
226 +void fileManager::OpenNewFile(){}
264
265
266 +void fileManager::DeleteDir(){}
```

# Terminal output with verbose on

usage: **DAqTanca -v -x Tanca\_v01.xml**

```
=====  
DAQ using digitizer CAEN DT5720B  
Usage: DAqTanca [-v] -x Tanca_v01.xml  
=====  
Parsing ./xml/Tanca_v01.xml  
Duration of run          Continuous data taking  
=====  
Digitizer Settings:  
LinkType:           USB  
LinkNum:            0  
ConetNode:          0  
BaseAddress:        0  
Nch:                4  
Nbit:               12  
Ts:                 4  
RecordLength:       70  
PostTrigger:        50  
InterruptNumEvents: 0  
TestPattern:        0  
FPIOtype:           0  
ExtTriggerMode:    Acq only  
Trigger Mode:      Maj1  
      Window:        10  
EnableMask:         0111  
=====  
Channel 0  
  TriggerMode:      Acq only  
  PulsePolarity:   Negative  
  Trigger Polarity: Negative  
  DCOffset:         56000  
  Threshold:        180  
Channel 1  
  TriggerMode:      Acq only  
  PulsePolarity:   Negative  
  Trigger Polarity: Negative  
  DCOffset:         56000  
  Threshold:        180  
Channel 2  
  TriggerMode:      Acq only  
  PulsePolarity:   Negative  
  Trigger Polarity: Negative  
  DCOffset:         56000  
  Threshold:        180  
Channel 3 Disabled  
=====  
Digitizer: Opening digitizer  
FileManager: Opening ./2022_02_13/2022_02_13_13_39tanca.root  
FileManager: Save interval: 10000000000  
=====  
Connected to CAEN Digitizer Model DT5720B  
ROC FPGA Release is 04.16 - Build 1720  
AMC FPGA Release is 00.15 - Build 1906  
=====  
Digitizer: Preprocessing data
```

```
Digitizer: Programming digitizer
Digitizer: Trigger enabled:
    Channel 0      Enabled: 1
    Channel 1      Enabled: 1
    Channel 2      Enabled: 1
    Channel 3      Enabled: 0
Global Trigger Mask 0xB10C address = 000000011010000000000000000000000000000111
=====
Digitizer: Acquisition started
Digitizer: Start time: 1644759586.49813
[q] to quit, [s] to stop/start run
Reading at 1.15 MB/s (Trg Rate: 2681.63 Hz, 26819 events)
fileManager: number of events so far = 36264
FileManager: OpenNewFile: 40
Reading at 1.15 MB/s (Trg Rate: 2695.03 Hz, 53772 events)
Reading at 1.15 MB/s (Trg Rate: 2695.13 Hz, 80726 events)
Reading at 1.15 MB/s (Trg Rate: 2694.93 Hz, 107678 events)
Reading at 1.15 MB/s (Trg Rate: 2694.93 Hz, 134630 events)
Reading at 1.15 MB/s (Trg Rate: 2694.83 Hz, 161581 events)
Reading at 1.15 MB/s (Trg Rate: 2694.83 Hz, 188532 events)
fileManager: number of events so far = 197959
FileManager: OpenNewFile: 41
Reading at 1.15 MB/s (Trg Rate: 2694.83 Hz, 215483 events)
Reading at 1.15 MB/s (Trg Rate: 2694.73 Hz, 242433 events)
Reading at 1.15 MB/s (Trg Rate: 2694.63 Hz, 269382 events)
Reading at 1.15 MB/s (Trg Rate: 2694.63 Hz, 296331 events)
Reading at 1.15 MB/s (Trg Rate: 2694.73 Hz, 323281 events)
Reading at 1.15 MB/s (Trg Rate: 2694.63 Hz, 350230 events)
fileManager: number of events so far = 359640
FileManager: OpenNewFile: 42
Reading at 1.15 MB/s (Trg Rate: 2694.63 Hz, 377179 events)
Reading at 1.15 MB/s (Trg Rate: 2694.73 Hz, 404129 events)
Reading at 1.15 MB/s (Trg Rate: 2694.63 Hz, 431078 events)
Reading at 1.15 MB/s (Trg Rate: 2694.63 Hz, 458027 events)
Reading at 1.15 MB/s (Trg Rate: 2694.63 Hz, 484926 events)
```

USB2.0 compliant up to 30 MB/s transfer rate  
2690 Hz => 1.15 MB/s

```
Reading at 1.15 MB/s (Trg Rate: 2694.53 Hz, 619718 events)
Reading at 1.15 MB/s (Trg Rate: 2694.63 Hz, 646667 events)
Reading at 1.15 MB/s (Trg Rate: 2694.63 Hz, 673616 events)
fileManager: number of events so far = 682995
FileManager: OpenNewFile: 44
Reading at 1.15 MB/s (Trg Rate: 2694.63 Hz, 700565 events)
Reading at 1.15 MB/s (Trg Rate: 2694.53 Hz, 727513 events)
■ Acquisition stopped
```

# Reading out at about 10 kHz

(3 ch, record length=70)



Less than 10kHz

Busy LED



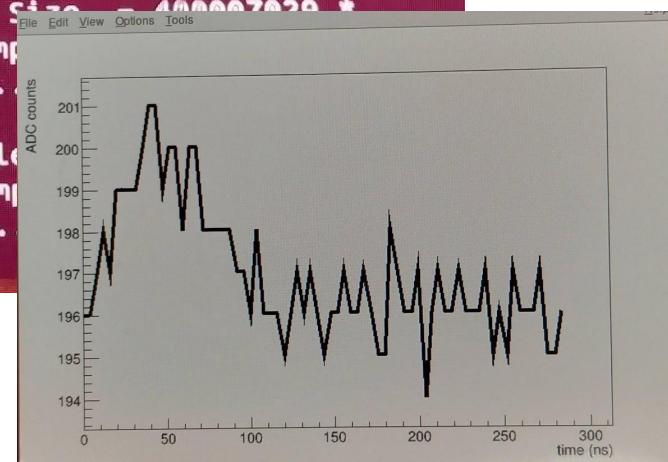
More than 10kHz

One hour tanking data at 2100 Hz generated one file of 1GB  
One hard disk with **1TB** will record about **40 days of data (3 PMTs)**

# TTree data

```
File Edit View Search Terminal Help
*root [5]
*root [5] data->Print()
*****
*Tree :data      : Waveform data
*Entries : 5259720 : Total =    9352755749 bytes File Size = 1119874726 *
*          : Tree compression factor = 8.35
*****
*Br  0 :ch0      : vector<double>
*Entries :5259720 : Total Size= 3103535569 bytes File Size = 384013318 *
*Baskets : 3066 : Basket Size= 5299712 bytes Compression= 8.08
*.....
*Br  1 :ch1      : vector<double>
*Entries :5259720 : Total Size= 3103535569 bytes File Size = 315173595 *
*Baskets : 3066 : Basket Size= 5299712 bytes Compression= 9.85
*.....
*Br  2 :ch2      : vector<double>
*Entries :5259720 : Total Size= 3103535569 bytes File Size = 400007020 *
*Baskets : 3066 : Basket Size= 5299712 bytes Compression= 8.08
*.....
*Br  3 :time     : time/D
*Entries : 5259720 : Total Size= 42148503 bytes File Size = 42148503 *
*Baskets : 774 : Basket Size= 57856 bytes Compression= 8.08
*.....
root [6] □
```

pulse generator;  
ch0; waveform=2



cho

ch1

ch2

time

# Summary

- It was developed a code to read the new DAq of the TANCA detector using C++ and ROOT
- The DAq is capable of recording data from three PMTs at rates up to 10000 Hz without significant dead time.
- The output file is .root format containing a TTree with ADC channel branches and the time stamp.
- This upgrade will allow studies of solar events with a better data quality and new kind of data analysis.

# Thank you for your attention

Anderson Fauth  
[fauth@unicamp.br](mailto:fauth@unicamp.br)

I thank the LAGO Workshop organization and Samuel de Jong, from Department of Physics & Astronomy - University of Victoria, CA for providing the CAEN V1730 digitizer code used as the basis for this work.

This work is supported by FAPESP, CNPq and Unicamp.

