

# muX analysis meeting

## 20.09.2021

Software & DAQ status, ongoing developments  
Future plans

Frederik Wauters

Johannes Gutenberg University Mainz

# Our MIDAS DAQ

This is an update from the [previous meeting](#), where we discussed

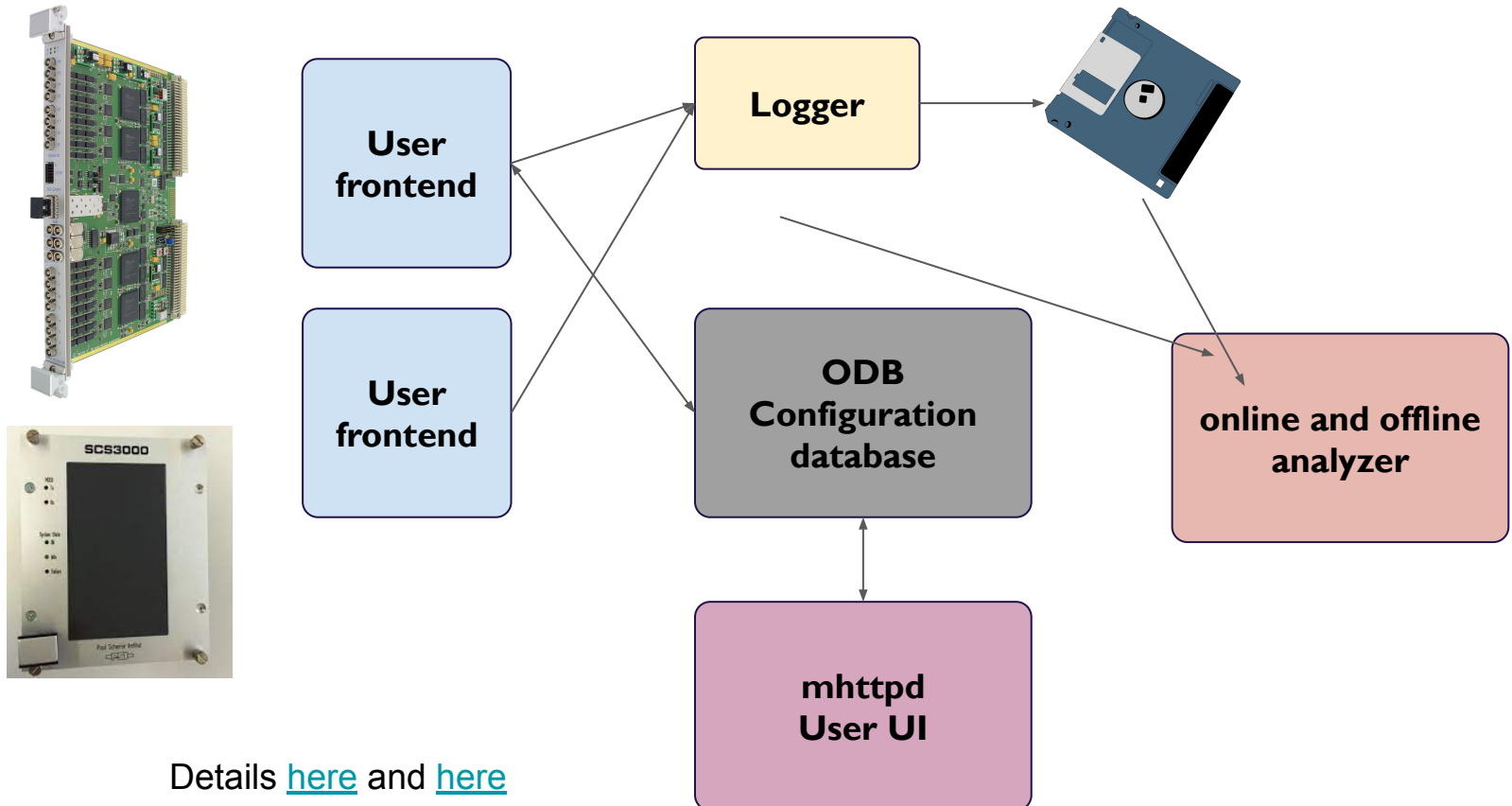
- Clustering
- Template fitting
- Improved ELET algorithm
- Machine learning ideas
- DAQ updates

# Our MIDAS DAQ

## Brief muX MIDAS DAQ intro

- Midas: <https://bitbucket.org/tmidas/midas/src/develop/> and <https://en.wikipedia.org/wiki/Midas>
- Our DAQ: <https://bitbucket.org/muxpsi/mux-daq/src/master/>

Detectors etc., aka a physics experiment.



Details [here](#) and [here](#)

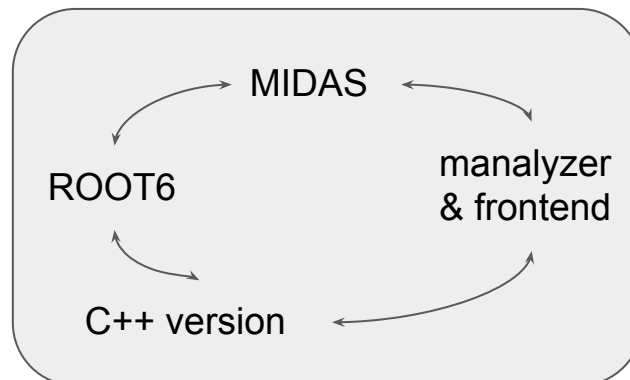
# Our MIDAS DAQ

## **Brief muX MIDAS DAQ intro**

- Midas: <https://bitbucket.org/tmidas/midas/src/develop/> and <https://en.wikipedia.org/wiki/Midas>
- Our DAQ: <https://bitbucket.org/muxpsi/mux-daq/src/master/>
- Midas has moved on:
  - C to C++
  - JSON like ODB interface
  - Everything via mhttpd web interface, HTML5, Javascript interface
  - manalyzer is the recommended/supported analysis framework (not mana.cpp)
- ROOT6 is now standard
- C++ has moved on to (I propose minimum C++14)
- Our DAQ machine has SL6, gcc 4.7 with dev package, ....



Update needed, but everything is tied together.



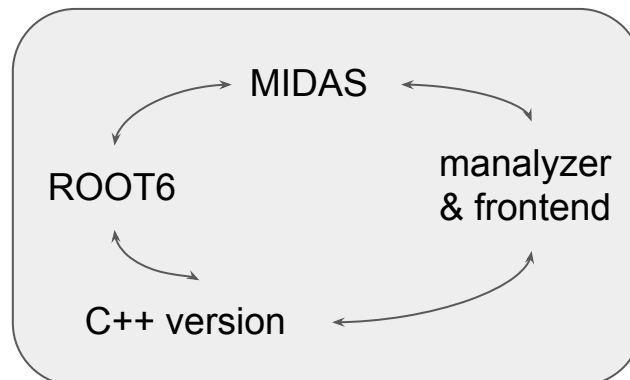
# Our MIDAS DAQ

## **Brief muX MIDAS DAQ intro**

- Midas: <https://bitbucket.org/tmidas/midas/src/develop/> and <https://en.wikipedia.org/wiki/Midas>
- Our DAQ: <https://bitbucket.org/muxpsi/mux-daq/src/master/>
- Midas has moved on:
  - C to C++
  - JSON like ODB interface
  - Everything via mhttpd web interface, HTML5, Javascript interface
  - manalyzer is the recommended/supported analysis framework (not mana.cpp)
- ROOT6 is now standard
- C++ has moved on to (I propose minimum C++14)
- Our DAQ machine has SL6, gcc 4.7 with dev package, ....



Update needed, but everything is tied together.



# Digitizer

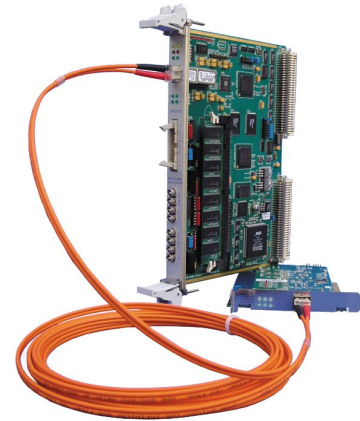
<https://www.struck.de/sis3316.html>

<https://www.struck.de/linux1100.htm>



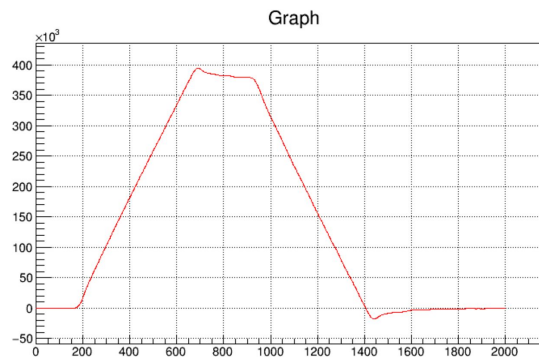
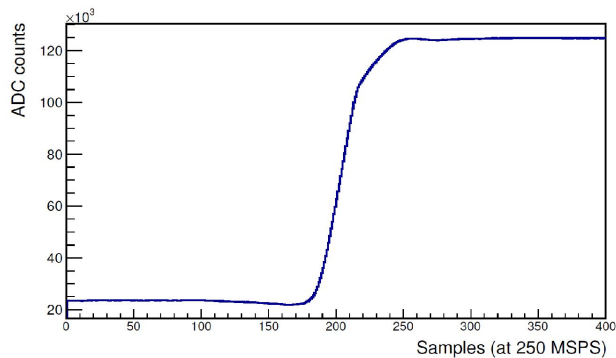
X 3

- Perfect for HPGe
- Ok'ish for scintillators
- 250 MSPS



Got it running for:

- Linux 5.4
- Ubuntu 20.4 LTS



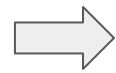
# Frontend



T5 etc.)

ments (Hyp

hing



```
//-----
// Parameters in sub-folder
sprintf(sub tree,"Data Format & Memory");
status = db find key(hdb, hkey, sub tree, &hsubkey);
if (status != DB SUCCESS){
    cm msg(MERROR, func , "%s/%s not found.", tree name,sub tree);
    abort();
}

size=N ADC*sizeof(sis3316 settings[iadc].maxEventsBank);
status = db get value(hdb,hsubkey,"Max events per bank",
    &size,TID DWORD,0);

//-----
// Parameters in sub-folder
sprintf(sub tree,"Pileup");
status = db find key(hdb, hkey, sub tree, &hsubkey);
if (status != DB SUCCESS){
    cm msg(MERROR, func , "%s/%s not found.", tree name,sub tree);
    abort();
}

size=N ADC*sizeof(sis3316 settings[iadc].pileupLength);
status = db get value(hdb,hsubkey,"Pileup length",
    &size,TID DWORD,0);

size=N ADC*sizeof(sis3316 settings[iadc].prePileupLength);
status = db get value(hdb,hsubkey,"PrePileup length",
    &size,TID DWORD,0);

//-----
// Parameters in sub-folder
sprintf(sub tree,"MAW Trace Buffer");
status = db find key(hdb, hkey, sub tree, &hsubkey);
if (status != DB SUCCESS){
    cm msg(MERROR, func , "%s/%s not found.", tree name,sub tree);
    abort();
}

size=N ADC*sizeof(sis3316 settings[iadc].dataFormatBit5);
status = db get value(hdb,hsubkey,"Triqger or energy MAW",
    &(sis3316 settings[iadc].dataFormatBit5),
```

```
//define the odb structure, not modify or read anything
midas::odb default settings{
    {"Names", std::array<std::string,32>() },
    {"Module Address", 805306368 },
    {"Info & Status", {
        {"VME FPGA Firmware", 0 },
        {"ADC 1-4 FPGA Firmware", 0},
        {"ADC 5-8 FPGA Firmware", 0},
        {"ADC 9-12 FPGA Firmware", 0},
        {"ADC 13-16 FPGA Firmware", 0},
        {"Serial Number", 0},
        {"Board Temperature", 99.9},
        {"Triqger Gate Length", std::array<uint32 t,4>(0) },
        {"Header Length", std::array<uint32 t,4>(0) },
        {"Event Length", std::array<uint32 t,4>(0) },
        {"Events Per Bank", std::array<uint32 t,4>(5000) } } },
    {"Data Format & Memory", {
        {"Max events per bank", std::array<uint32 t,4>(5000) }
    }},
    {"Pileup", {
        {"Pileup Length", std::array<uint32 t,4>(3500)},
        {"PrePileup Length", std::array<uint32 t,4>(500) } } },
    {"MAW Trace Buffer", {
        {"Triqger or energy MAW", std::array<uint32 t,4>(1) },
        {"MAW Buffer Length", std::array<uint32 t,4>(0) },
        {"MAW Buffer Delay", std::array<uint32 t,4>(0) },
        {"MAW Buffer Start Index", std::array<uint32 t,16>(0) },
        {"MAW Energy Pickup Index", std::array<uint32 t,16>(0) } } },
    {"FIR Energy", {
        {"Energy Gap Value", std::array<uint32 t,16>(10) },
        {"Energy Peaking Value", std::array<uint32 t,16>(10) },
        {"Energy Tau Table", std::array<uint32 t,16>(0) },
        {"Energy Tau Table", std::array<uint32 t,16>(0) } } },
    {"FIR Triqger", {
        {"Triqger Gap Value", std::array<uint32 t,16>(10) },
        {"Triqger Peaking Value", std::array<uint32 t,16>(10) },
        {"Triqger Threshold", std::array<uint32 t,16>(100) },
        {"Triqger Mode", std::array<uint32 t,16>(0) },
        {"Triqger Enable", std::array<uint32 t,16>(0) } } },
    {"Channel Configuration",{
        {"DAC offset", std::array<uint32 t,16>(32768) },
        {"Negative Signal", std::array<uint32 t,16>(0) },
        {"Disable 50 Ohm Termination", std::array<uint32 t,16>(0) },
        {"Enable 2 V Range", std::array<uint32 t,16>(0) },
        {"ADC Range", std::array<uint32 t,8>(1) } } }
}

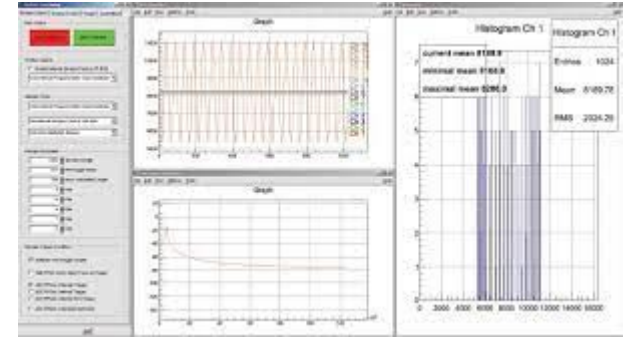
default settings.connect("/Equipment/Trigger/Settings/sis3316/"+str(module id));
settings.connect("/Equipment/Trigger/Settings/sis3316/"+str(module id));
```

Module Address	805306368 (0x30000000)	[12] 40 (0x28)	[13] 40 (0x28)
		[14] 20 (0x14)	[14] 20 (0x14)
		[15] 20 (0x14)	[15] 20 (0x14)

...  
...

# Frontend

- ❑ There is a Struck test GUI for testing  
→ Useful, not well maintained (Needs ROOT5 etc.)
- ❑ We have a frontend.cpp
  - ❑ 3 synchronized modules
  - ❑ Written by AK, RP, FW
  - ❑ Used by muX and other PIEI experiments (HyperMu, muonium, ... )
- ❑ A bit messy
- ❑ Needs analyzer running to see something

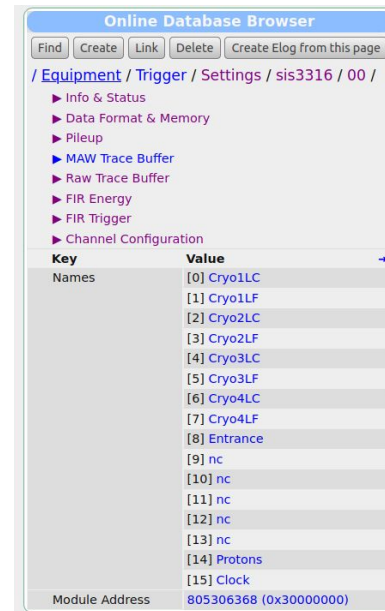


- ❑ Frontend improvements
  - ❑ Clean up code
  - ❑ Faster?
  - ❑ ODBxx interface
  - ❑ Test mode would be nice:
    - ❑ See waveforms
    - ❑ MAW waveform
    - ❑ A basic histogram

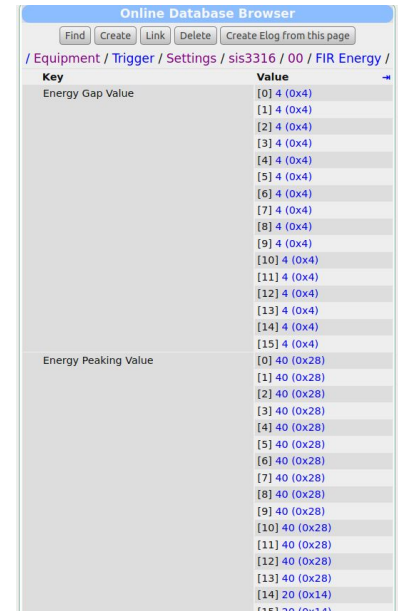
→ Nice, but priority?

...

...



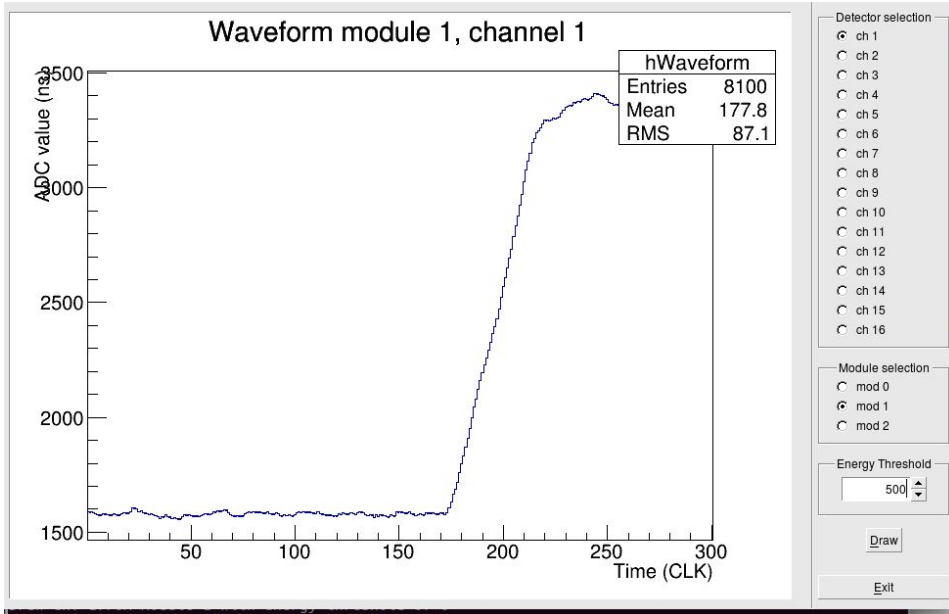
Key	Value
Names	[0] Cryo1LC
	[1] Cryo1LF
	[2] Cryo2LC
	[3] Cryo2LF
	[4] Cryo3LC
	[5] Cryo3LF
	[6] Cryo4LC
	[7] Cryo4LF
	[8] Entrance
	[9] nc
	[10] nc
	[11] nc
	[12] nc
	[13] nc
	[14] Protons
	[15] Clock
Module Address	805306368 (0x30000000)



Key	Value
Energy Gap Value	[0] 4 (0x4)
	[1] 4 (0x4)
	[2] 4 (0x4)
	[3] 4 (0x4)
	[4] 4 (0x4)
	[5] 4 (0x4)
	[6] 4 (0x4)
	[7] 4 (0x4)
	[8] 4 (0x4)
	[9] 4 (0x4)
	[10] 4 (0x4)
	[11] 4 (0x4)
	[12] 4 (0x4)
	[13] 4 (0x4)
	[14] 4 (0x4)
	[15] 4 (0x4)
Energy Peaking Value	[0] 40 (0x28)
	[1] 40 (0x28)
	[2] 40 (0x28)
	[3] 40 (0x28)
	[4] 40 (0x28)
	[5] 40 (0x28)
	[6] 40 (0x28)
	[7] 40 (0x28)
	[8] 40 (0x28)
	[9] 40 (0x28)
	[10] 40 (0x28)
	[11] 40 (0x28)
	[12] 40 (0x28)
	[13] 40 (0x28)
	[14] 20 (0x14)
	[15] 20 (0x14)



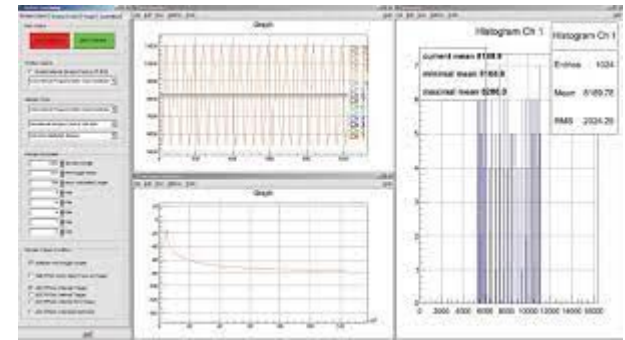
# Frontend



```

Draw ch: 1from module 1 with energy threshold of 0
Draw ch: 1from module 1 with energy threshold of 0
Draw ch: 1from module 1 with energy threshold of 0
Draw ch: 1from module 1 with energy threshold of 0
Draw ch: 1from module 1 with energy threshold of 0
Draw ch: 1from module 1 with energy threshold of 400
Draw ch: 1from module 1 with energy threshold of 400
Draw ch: 1from module 1 with energy threshold of 402
Draw ch: 1from module 1 with energy threshold of 402
Draw ch: 1from module 1 with energy threshold of 402
Draw ch: 1from module 1 with energy threshold of 402
Draw ch: 1from module 1 with energy threshold of 402
Draw ch: 1from module 1 with energy threshold of 402
Draw ch: 1from module 1 with energy threshold of 402
Draw ch: 1from module 1 with energy threshold of 402
Draw ch: 1from module 1 with energy threshold of 402
Draw ch: 1from module 1 with energy threshold of 402
Draw ch: 1from module 1 with energy threshold of 402
Draw ch: 1from module 1 with energy threshold of 402
Draw ch: 1from module 1 with energy threshold of 402
Draw ch: 1from module 1 with energy threshold of 402

```



HyperMu, muonium, ... )

Online Database Browser

Find Create Link Delete Create Elog from this page

/ Equipment / Trigger / Settings / sis3316 / 00 /

- ▶ Info & Status
- ▶ Data Format & Memory
- ▶ Pileup
- ▶ MAW Trace Buffer
- ▶ Raw Trace Buffer
- ▶ FIR Energy
- ▶ FIR Trigger
- ▶ Channel Configuration

Key	Value
Names	[0] Cryo1LC
	[1] Cryo1LF
	[2] Cryo2LC
	[3] Cryo2LF
	[4] Cryo3LC
	[5] Cryo3LF
	[6] Cryo4LC
	[7] Cryo4LF
	[8] Entrance
	[9] nc
	[10] nc
	[11] nc
	[12] nc
	[13] nc
	[14] Protons
	[15] Clock
Module Address	805306368 (0x30000000)

Online Database Browser

Find Create Link Delete Create Elog from this page

/ Equipment / Trigger / Settings / sis3316 / 00 / FIR Energy /

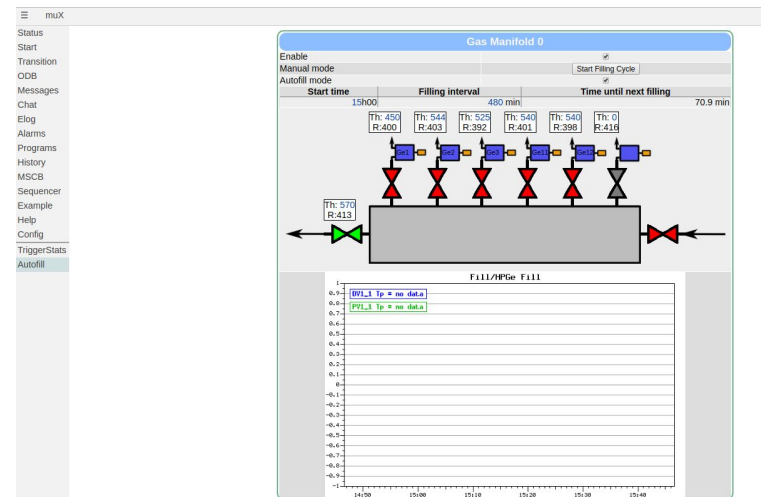
Key	Value
Energy Gap Value	[0] 4 (0x4)
	[1] 4 (0x4)
	[2] 4 (0x4)
	[3] 4 (0x4)
	[4] 4 (0x4)
	[5] 4 (0x4)
	[6] 4 (0x4)
	[7] 4 (0x4)
	[8] 4 (0x4)
	[9] 4 (0x4)
	[10] 4 (0x4)
	[11] 4 (0x4)
	[12] 4 (0x4)
	[13] 4 (0x4)
	[14] 4 (0x4)
Energy Peaking Value	[0] 40 (0x28)
	[1] 40 (0x28)
	[2] 40 (0x28)
	[3] 40 (0x28)
	[4] 40 (0x28)
	[5] 40 (0x28)
	[6] 40 (0x28)
	[7] 40 (0x28)
	[8] 40 (0x28)
	[9] 40 (0x28)
	[10] 40 (0x28)
	[11] 40 (0x28)
	[12] 40 (0x28)
	[13] 40 (0x28)
	[14] 20 (0x14)
	[15] 20 (0x14)

There is a waveform display, but only works after first analysis pass with custom setting:

<https://muon.npl.washington.edu/elog/neutralcurrents/Run2019/131>

# Midas

- ❑ Midas is being used for several running and upcoming experiments (Mu3e, g-2, MEGII, ... )  
→ Actively developed
  - ❑ [ODBxx](#)
  - ❑ [History system](#)
  - ❑ Custom/user webpages → Put everything in the web interface
  
- ❑ See keep an eye on
  - ❑ <https://midas.triumf.ca/eelog/Midas/>
  - ❑ [https://midas.triumf.ca/MidasWiki/index.php/Main\\_Page](https://midas.triumf.ca/MidasWiki/index.php/Main_Page)
  - ❑ <https://bitbucket.org/tmidas/midas/src/develop/>

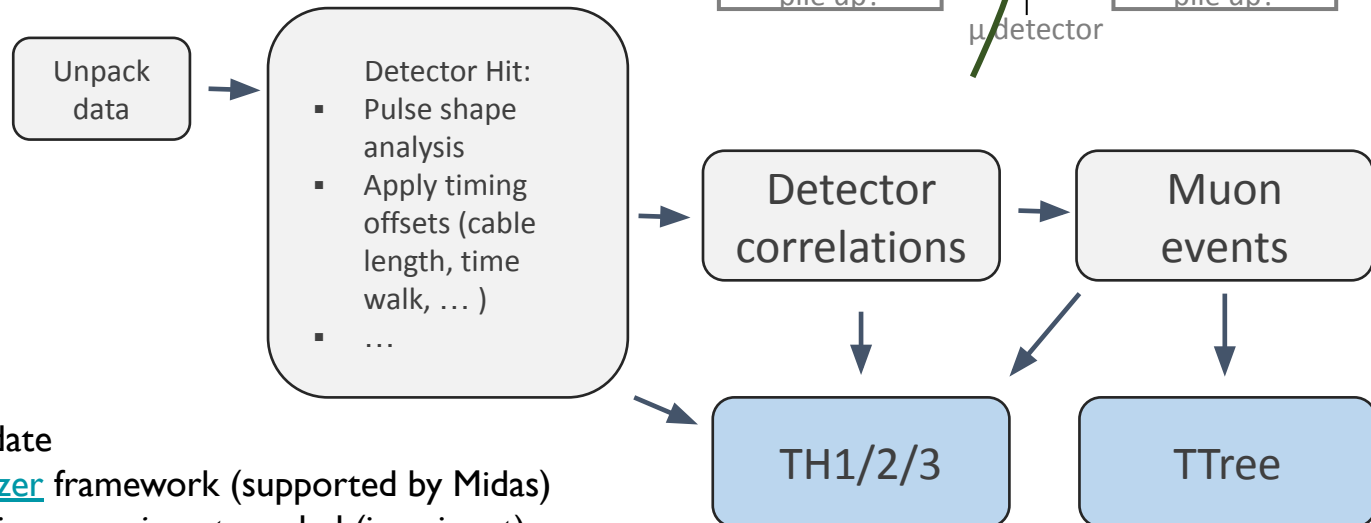


# Analyzer

- ❑ We use a hacked version of mana.cpp, inherited MuCap/MuSun (2000's)
  - ❑ Not supported anymore
  - ❑ Solid C++ framework for muX on top: [mux\\_analyzer](#)

## C++ analyzer with MIDAS interface:

- \*.mid file unpacker
- ODB interface for settings
- Sequential analysis modules (mandatory + user)
- First stage analyzer = online analyzer
- ROOT hist and TTree output



- ❑ Needs an update
  - ❑ [manalyzer](#) framework (supported by Midas)
  - ❑ No active *experiment* needed (json input)
  - ❑ Modern C++

# Analyzer

- ❑ We use a hacked version of mana.cpp
  - ❑ Not supported anymore
  - ❑ Solid C++ framework for muX on top: [mux\\_analyzer](#)
- ❑ Needs an update
  - ❑ [manalyzer](#) framework (supported by Midas)
  - ❑ No active *experiment* needed (json input)
  - ❑ Modern C++
- ❑ We have one! [https://bitbucket.org/muxpsi/new\\_daq/src/master/analyzer/](https://bitbucket.org/muxpsi/new_daq/src/master/analyzer/)
  - ❑ Working (but still with the old rootana package, needs to be fixed to work with the main Midas lib)
  - ❑ Keeps analysis structure
  - ❑ JSON input
  - ❑ Root server compatible with JSROOT
  - ❑ Bring it online when?

# Online display

- ❑ Have an old histogram viewer
  - ❑ Really nice to format and group histograms for monitoring
  - ❑ 100 % online mode brokes (I run delay)
- ❑ Nowadays folks use [JSROOT](#)
- ❑ Supported by Midas/[MAnalyzer](#)
  
- ❑ A prototype exist! <https://muon.npl.washington.edu/elog/neutralcurrents/Analysis2019/40>
  - ❑ Needs *pages* to see more plots
  - ❑ Needs histogram formatting

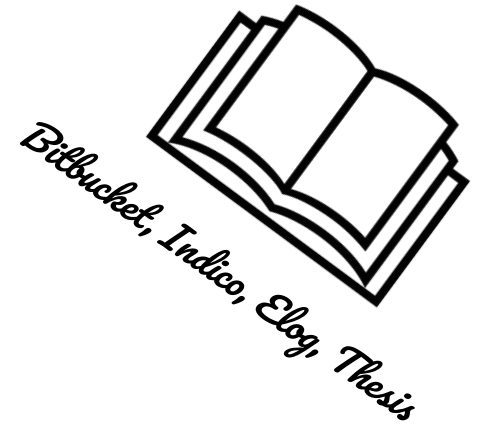


# Online display

- ❑ Have an old histogram viewer
  - ❑ Really nice to format and group histograms for monitoring
  - ❑ 100 % online mode brokes (I run delay)
- ❑ Nowadays folks use [JSROOT](#)
- ❑ Supported by Midas/[MAnalyzer](#)
  
- ❑ A prototype exist! <https://muon.npl.washington.edu/elog/neutralcurrents/Analysis2019/40>
  - ❑ Needs *pages* to see more plots
  - ❑ Needs histogram formatting



# Analyzer



THUS

Conclusions of 10 months ago

## TODO's

- Clean up master branch

## Motivating upgrades:

- Core muX late in experimental cycle
- Working software package
- + slowly more and more outdated
- + FE & analyzer used by
  - muX
  - Dubna
  - Muonium (2020)
  - Hypermu (2019)
  - MIXE (2019/2020)

Have little time in Q1/2 2021, happy to help students with development

# Analyzer

THUS

Time for some updates

- New DAQ machine with modern linux & g++ compiler
  - ROOT6
  - C++14
  - Latest Midas
  - Manalyzer
  - ? more DAQ gear ? ....
- Our online analyzer is also our 1st stage offline analyzer
- We have a working updated:
  - Analyzer
- In progress:
  - Event display
  - Frontend software

Analysis progress not discussed here, but worth mentioning:

- ELET and Machine Learning waveform processing (AS, FW, MK)
- Baseline correction (AS, FW)
- Clustering (FW, SV)
- Coincidence analysis (ND, ... )