

# DAPHNE Decoder

---

Jake Calcutt

9/5/23

# Introduction

Began work on a raw decoder for DAPHNE data from the DAQ

Module found [here](#)

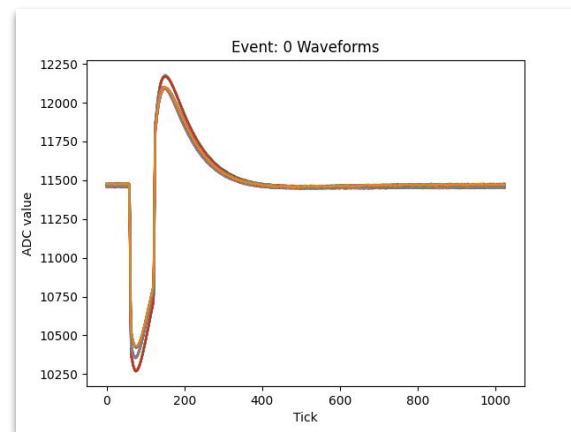
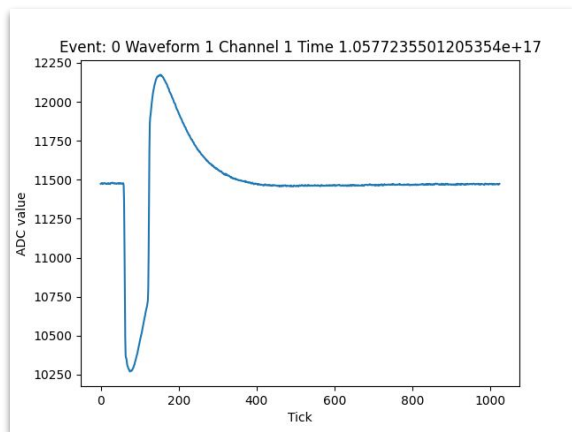
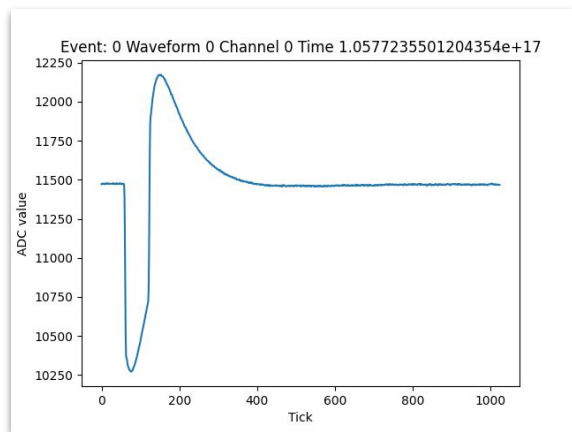
- Note: in a feature branch built off of duneprototypes v09\_78\_01d00
- Still needs some work

Tested on files from `samweb list-files run\_type hd-protodune and file\_type detector and run\_number 22628`

- Filled with pulser data

# Example waveforms

File: np04\_hd\_run022628\_0000\_dataflow0\_datawriter\_0\_20230818T112308.hdf5



# Questions/Next Steps

What should be used for the OpDetWaveform channel?

DAPHNEFrame::get\_channel seems to return nothing?

```
133
134     size_t n_frames = (frag_size - frag_header_size)/sizeof(DAPHNEFrame);
135     std::cout << "NFrames: " << n_frames << " Header TS: " <<
136             frag->get_header().trigger_timestamp << std::endl;
137     for (size_t i = 0; i < n_frames; ++i) {
138         auto frame = reinterpret_cast<DAPHNEFrame*>{
139             static_cast<uint8_t*>(frag->get_data()) + i*sizeof(DAPHNEFrame)};
140
141         std::cout << i << " " << frame->get_channel() << " " <<
142             frame->get_timestamp() << " " << frame->s_num_adcs <<
143             std::endl;
144         raw::OpDetWaveform waveform(frame->get_timestamp(i, frame->s_num_adcs);
145         for (size_t j = 0; j < static_cast<size_t>(frame->s_num_adcs); ++j) {
146             //std::cout << "\t" << frame->get_adc(j) << std::endl;
147             waveform.push_back(frame->get_adc(j));
148         }
149         opdet_waveforms.emplace_back(waveform);
150     }
151 }
152 }
```

# Questions/Next Steps

Which for the waveform timestamp?

- Frame timestamp or trigger timestamp?
- Below: [SSPRawdecoder](#) for PDSP used trigger timestamp apparently
- Also what units?

```
133
134     size_t n_frames = (frag_size - frag_header_size)/sizeof(DAPHNEFrame);
135     std::cout << "NFrames: " << n_frames << " Header TS: " <<
136         frag->get_header().trigger_timestamp << std::endl;
137     for (size_t i = 0; i < n_frames; ++i) {
138         auto frame = reinterpret_cast<DAPHNEFrame*>{
139             static_cast<uint8_t*>(frag->get_data()) + i*sizeof(DAPHNEFrame)};
140
141         std::cout << i << " " << frame->get_channel() << " " <<
142             frame->get_timestamp() << " " << frame->s_num_adcs <<
143             std::endl;
144         raw::OpDetWaveform waveform(frame->get_timestamp(), frame->s_num_adcs);
145         for (size_t j = 0; j < static_cast<size_t>(frame->s_num_adcs); ++j) {
146             //std::cout << "\t" << frame->get_adc(j) << std::endl;
147             waveform.push_back(frame->get_adc(j));
148         }
149         opdet_waveforms.emplace_back(waveform);
150     }
151 }
152 }
```

```
487     /// time
488     ///long double time = trig.timestamp_nova*clockData.OpticalClock().TickPeriod(); //in experiment microseconds
489     // DO NOT USE clockData.OpticalClock().TickPeriod()!!!! It is not precise enough
490     // use OpticalClock().Frequency, and do the division yourself with high precision.
491     double time = double(trig.timestamp_nova % 1000000000 ) / double(clockData.OpticalClock().Frequency());
492     //true time truncated by 10 digits in order to make sure the math works correctly
493     //std::cout << time << std::endl;
```

# Questions/Next Steps

Want to save recob::OpHits?

- [SSPRawdecoder](#) included some light reconstruction to create these