SSD online calib status

Name 1*, Name 2, Name 3

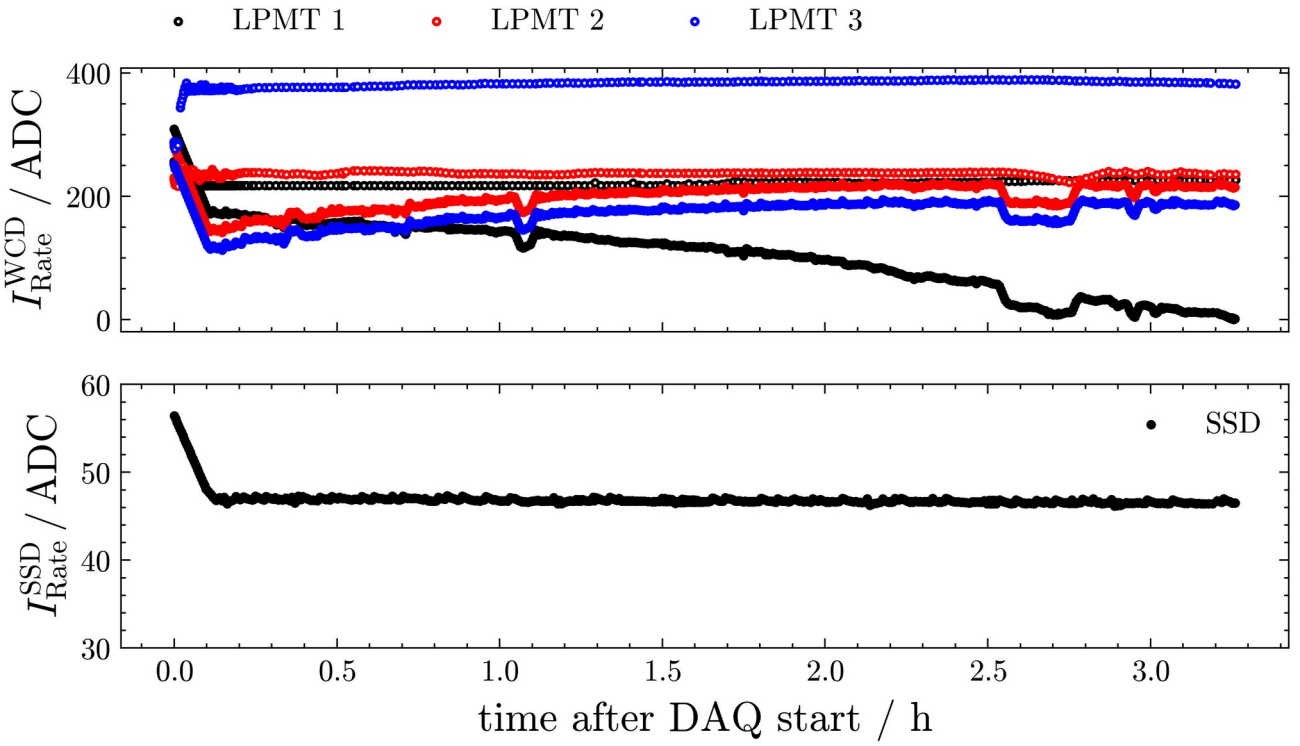
Outline

- Introduction
- First point
  - Discussion
- Second point
- Summary and outlook

First results of integration test

- SSD stable
- Realistic?
- too stable?

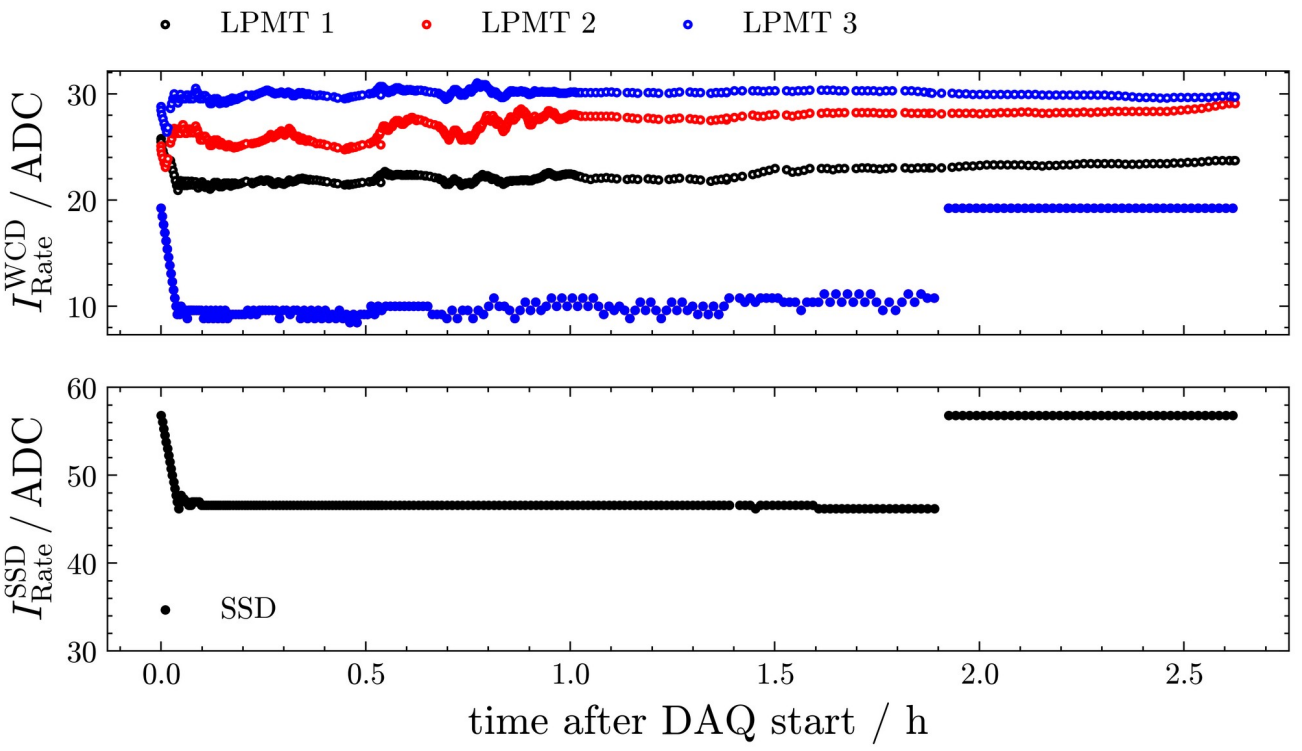
- Some issues:
  - WCD calib values
  - DAQ restarts?



First results of integration test

- SSD stable
- Realistic?
- too stable?

- Some issues:
  - WCD calib values
  - DAQ restarts?



Open questions: trigger bits

- Muonbuffer

Bits	Description
5:0	Muon triggers and SIPM calibration flag for this burst
6	PMT0 above threshold, instance 1
7	PMT1 above threshold, instance 1
8	PMT2 above threshold, instance 1
9	SSD PMT above threshold, instance 1
10	PMT0 above threshold, instance 2
11	PMT1 above threshold, instance 2
12	PMT2 above threshold, instance 2
31	SSD PMT above threshold. instance 2

```
713   int pmt, bin;
714   int thT1_multiplicity;
715   if(8096-NBBIN-1 < index){
716     return(1);
717   }
718   flags->ttag     = buff1[index];
719   flags->trig_type= buff2[index];
720
721   if((((flags->ttag      >> 31) & 0x1) !=1) ||
722      (((flags->trig_type >> 31) & 0x1) !=1)    ){
723     return(2);
724   }
725
726   // how to apply PMT trigger masks, known at FPGA lvl?
727   flags->ttag      &= 0x7FFFFFFF;
728   flags->is_calib_channel =  ( flags->trig_type >> 5 ) & 0x1;
729   flags->is_wcd_thT1 = 0;
730
731   uint32_t isSSD = ((flags->trig_type >> 9) & 0x1) || ((flags->trig_type >> 31) & 0x1);
732   uint32_t isWCD = ((flags->trig_type >> 6) & 0x7) || ((flags->trig_type >> 10) & 0x7);
733   flags->is_ssd_only = !isWCD && isSSD;
734   // flags->is_ssd_only = ( flags->trig_type >> 5 ) & 0x1;          // for integration tests
735   flags->wcd_pmt_mask = 7;                                        // for integration tests
736   flags->trig_type &= 0xF;
```

Open questions: trigger bits

- Muonbuffer

Bits	Description
5:0	Muon triggers and SIPM calibration flag for this burst
6	PMT0 above threshold, instance 1
7	PMT1 above threshold, instance 1
8	PMT2 above threshold, instance 1
9	SSD PMT above threshold, instance 1
10	PMT0 above threshold, instance 2
11	PMT1 above threshold, instance 2
12	PMT2 above threshold, instance 2
31	SSD PMT above threshold. instance 2

```
713   int pmt, bin;
714   int thT1_multiplicity;
715   if(8096-NBBIN-1 < index){
716     return(1);
717   }
718   flags->ttag     = buff1[index];
719   flags->trig_type= buff2[index];
720
721   if((((flags->ttag      >> 31) & 0x1) !=1) ||
722      (((flags->trig_type >> 31) & 0x1) !=1)    ){
723     return(2);
724   }
725
726   // how to apply PMT trigger masks, known at FPGA lvl?
727   flags->ttag      &= 0x7FFFFFFF;
728   flags->is_calib_channel =  ( flags->trig_type >> 5 ) & 0x1;
729   flags->is_wcd_thT1 = 0;
730
731   uint32_t isSSD = ((flags->trig_type >> 9) & 0x1) || ((flags->trig_type >> 31) & 0x1);
732   uint32_t isWCD = ((flags->trig_type >> 6) & 0x7) || ((flags->trig_type >> 10) & 0x7);
733   flags->is_ssd_only = !isWCD && isSSD;
734   // flags->is_ssd_only = ( flags->trig_type >> 5 ) & 0x1;          // for integration tests
735   flags->wcd_pmt_mask = 7;                                        // for integration tests
736   flags->trig_type &= 0xF;
```

Open questions: trigger bits

- Muonbuffer

Bits	Description
5:0	Muon triggers and SIPM calibration flag for this burst
6	PMT0 above threshold, instance 1
7	PMT1 above threshold, instance 1
8	PMT2 above threshold, instance 1
9	SSD PMT above threshold, instance 1
10	PMT0 above threshold, instance 2
11	PMT1 above threshold, instance 2
12	PMT2 above threshold, instance 2
31	SSD PMT above threshold. instance 2

```
713   int pmt, bin;
714   int thT1_multiplicity;
715   if(8096-NBBIN-1 < index){
716     return(1);
717   }
718   flags->ttag     = buff1[index];
719   flags->trig_type= buff2[index];
720
721   if((((flags->ttag      >> 31) & 0x1) !=1) ||
722      (((flags->trig_type >> 31) & 0x1) !=1)    ){
723     return(2);
724   }
725
726   // how to apply PMT trigger masks, known at FPGA lvl?
727   flags->ttag      &= 0x7FFFFFFF;
728   flags->is_calib_channel =  ( flags->trig_type >> 5 ) & 0x1;
729   flags->is_wcd_thT1 = 0;
730
731   uint32_t isSSD = ((flags->trig_type >> 9) & 0x1) || ((flags->trig_type >> 31) & 0x1);
732   uint32_t isWCD = ((flags->trig_type >> 6) & 0x7) || ((flags->trig_type >> 10) & 0x7);
733   flags->is_ssd_only = !isWCD && isSSD;
734   // flags->is_ssd_only = ( flags->trig_type >> 5 ) & 0x1;          // for integration tests
735   flags->wcd_pmt_mask = 7;                                        // for integration tests
736   flags->trig_type &= 0xF;
```

Open questions: trigger bits

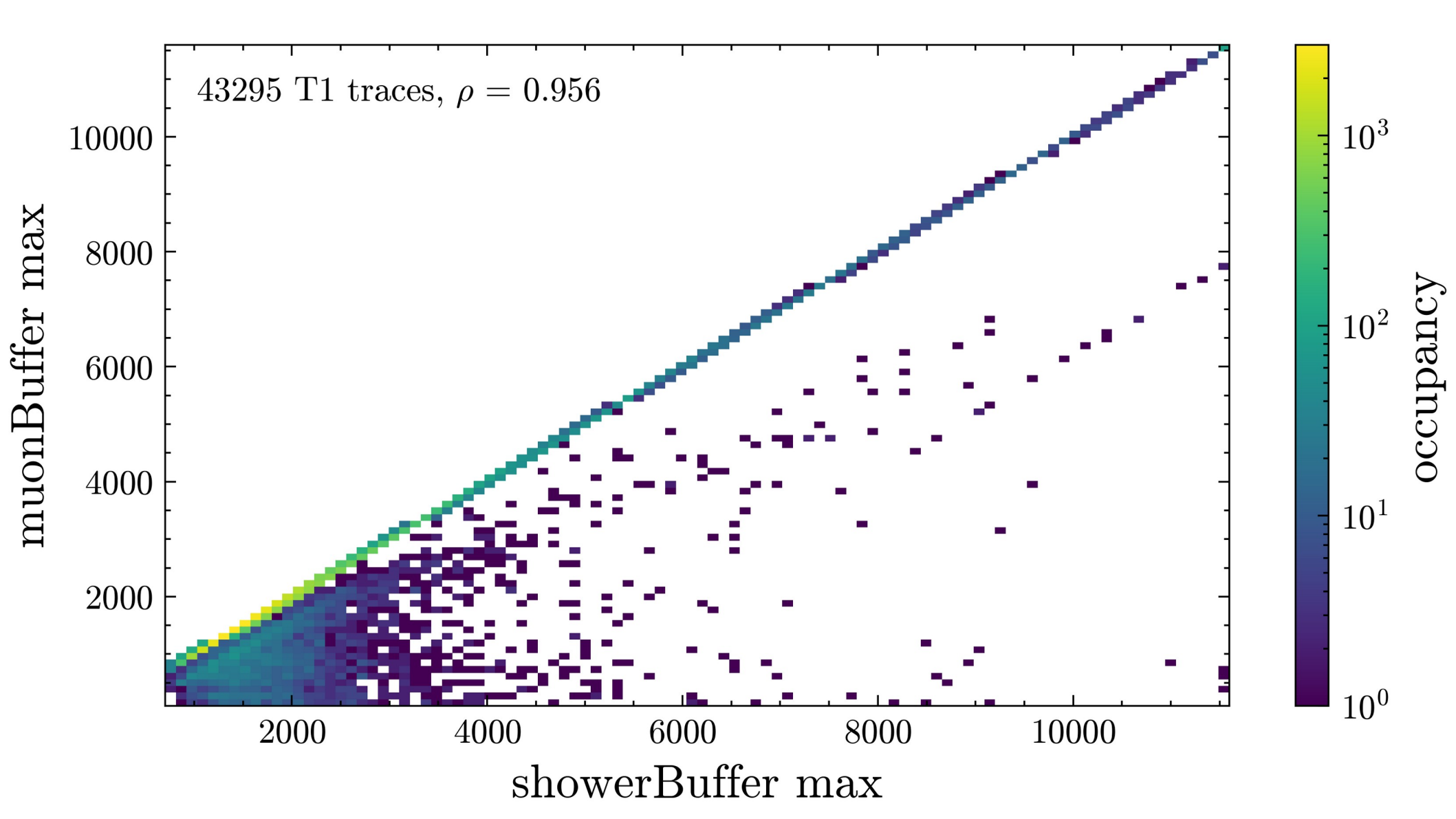
- Muonbuffer

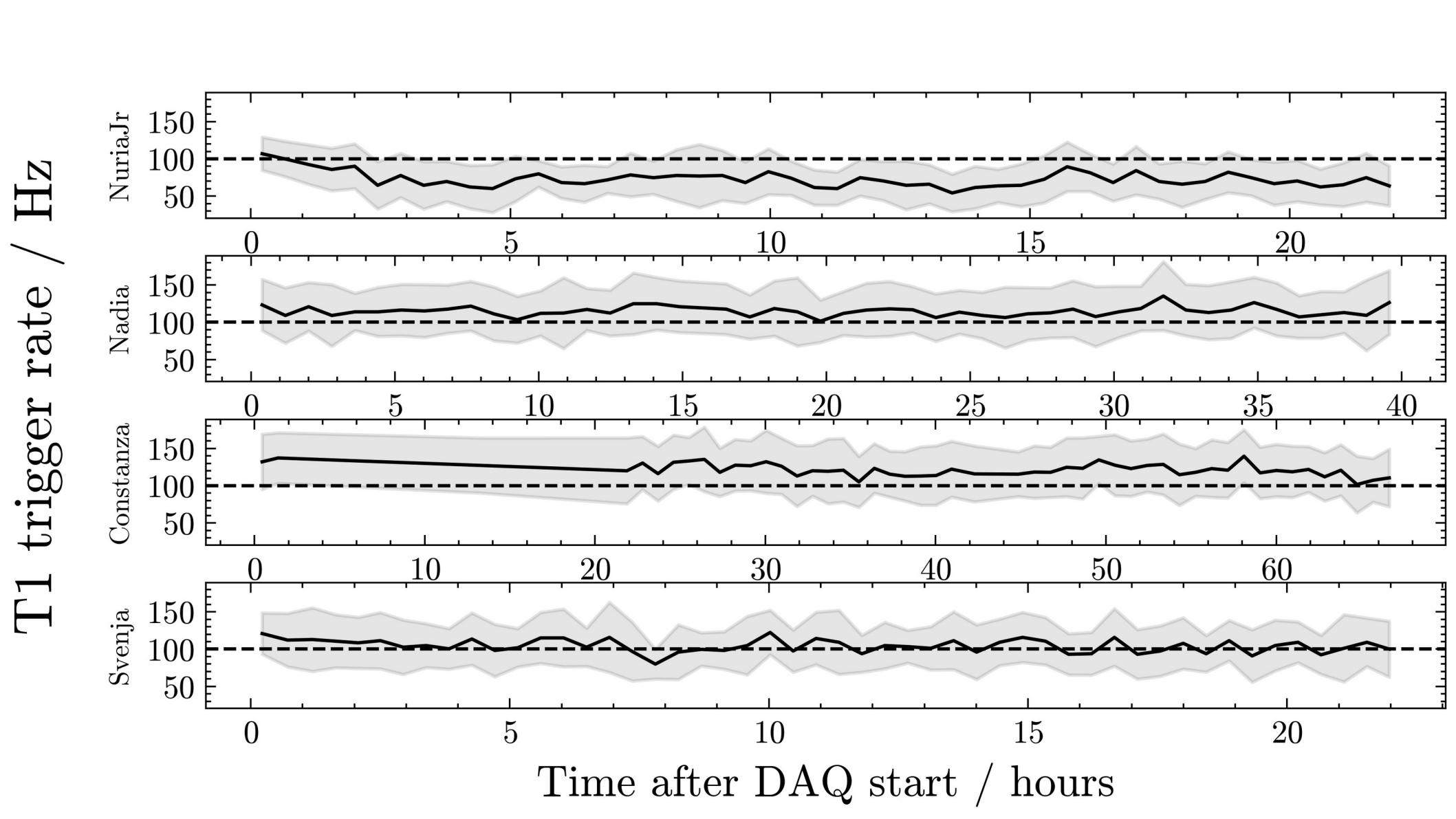
Bits	Description
5:0	Muon triggers and SIPM calibration flag for this burst
6	PMT0 above threshold, instance 1
7	PMT1 above threshold, instance 1
8	PMT2 above threshold, instance 1
9	SSD PMT above threshold, instance 1
10	PMT0 above threshold, instance 2
11	PMT1 above threshold, instance 2
12	PMT2 above threshold, instance 2
31	SSD PMT above threshold. instance 2

- Showerbuffer
  - Tube mask needed
  - which bits to read?
  - How to access?

```
713   int pmt, bin;
714   int thT1_multiplicity;
715   if(8096-NBBIN-1 < index){
716     return(1);
717   }
718   flags->ttag     = buff1[index];
719   flags->trig_type= buff2[index];
720
721   if((((flags->ttag      >> 31) & 0x1) !=1) ||
722      (((flags->trig_type >> 31) & 0x1) !=1)    ){
723     return(2);
724   }
725
726   // how to apply PMT trigger masks, known at FPGA lvl?
727   flags->ttag      &= 0x7FFFFFFF;
728   flags->is_calib_channel =  ( flags->trig_type >> 5 ) & 0x1;
729   flags->is_wcd_thT1 = 0;
730
731   uint32_t isSSD = ((flags->trig_type >> 9) & 0x1) || ((flags->trig_type >> 31) & 0x1);
732   uint32_t isWCD = ((flags->trig_type >> 6) & 0x7) || ((flags->trig_type >> 10) & 0x7);
733   flags->is_ssd_only = !isWCD && isSSD;
734   // flags->is_ssd_only = ( flags->trig_type >> 5 ) & 0x1;          // for integration tests
735   flags->wcd_pmt_mask = 7;                                        // for integration tests
736   flags->trig_type &= 0xF;
```

Backup


43295 T1 traces, $\rho = 0.956$
muonBuffer max
showerBuffer max
occupancy
10000
8000
6000
4000
2000
2000
4000
6000
8000
10000
10^3
10^2
10^1
10^0


T1 trigger rate / Hz
NuriaJr
Nadia
Constanza
Svenja
150
100
50
0
5
10
15
20
25
30
35
40
60
Time after DAQ start / hours

```
991   /* check if histogram ACQ time is greater than target time */
992   if ( ((muonbuffer->ttag_sec - CurrentHisto->histo.StartSecond) >
993         dt_online) || force_wr )
994   {
995     /* perform online calibration and (re)set thresholds */
996     int all_within_limits = 0;
997     for (i=0; i<4; i++)
998     {
999       int delta = CurrentHisto->extra.countsT70[i] / dt_online - TThresh;
1000      int sign = (delta > 0) - (delta < 0);
1001      delta = abs(delta);
1002
1003      if (delta <= 2)                                    // have 70 +- 2 Hz rate
1004      {                                                  // ---------------------------
1005        if (calib_adjust[i] > 1) --calib_adjust[i];      // decrement calib_adjust by ~0.1 ADC
1006        all_within_limits += 1;
1007      }
1008      else                                               // don't have 70 +- 2 Hz rate
1009      {                                                  // ---------------------------
1010        dt_online = 15;                                  // (re)set online calib interval to 10s
1011        if (delta > 20) calib_adjust[i] = 10;            // 70 +- >20 Hz -> calib increment = ~1 ADC
1012        else if (delta > 10) calib_adjust[i] = 5;        // 70 +- >10 Hz -> calib increment = ~0.5 ADC
1013        else if (delta > 5) calib_adjust[i] = 2;         // 70 +- > 5 Hz -> calib increment = ~0.2 ADC
1014      }
1015
1016      if !(CurrentHisto->extra.peak_threshold[i] < -sign * calib_adjust[i])
1017      {
1018        CurrentHisto->extra.peak_threshold[i] += sign * calib_adjust[i];
1019      }
1020
1021      if (all_within_limits == 4) dt_online += 5;
1022      if (dt_online > gl.integrationInterval) dt_online = gl.integrationInterval;
1023    }
```