Global TDAQ status

Riccardo Ridolfi, Mauro Villa, Silvia Biondi, Sofia Colombi

University of Bologna and INFN Bologna

25 May 2021 – X FOOT Collaboration Meeting

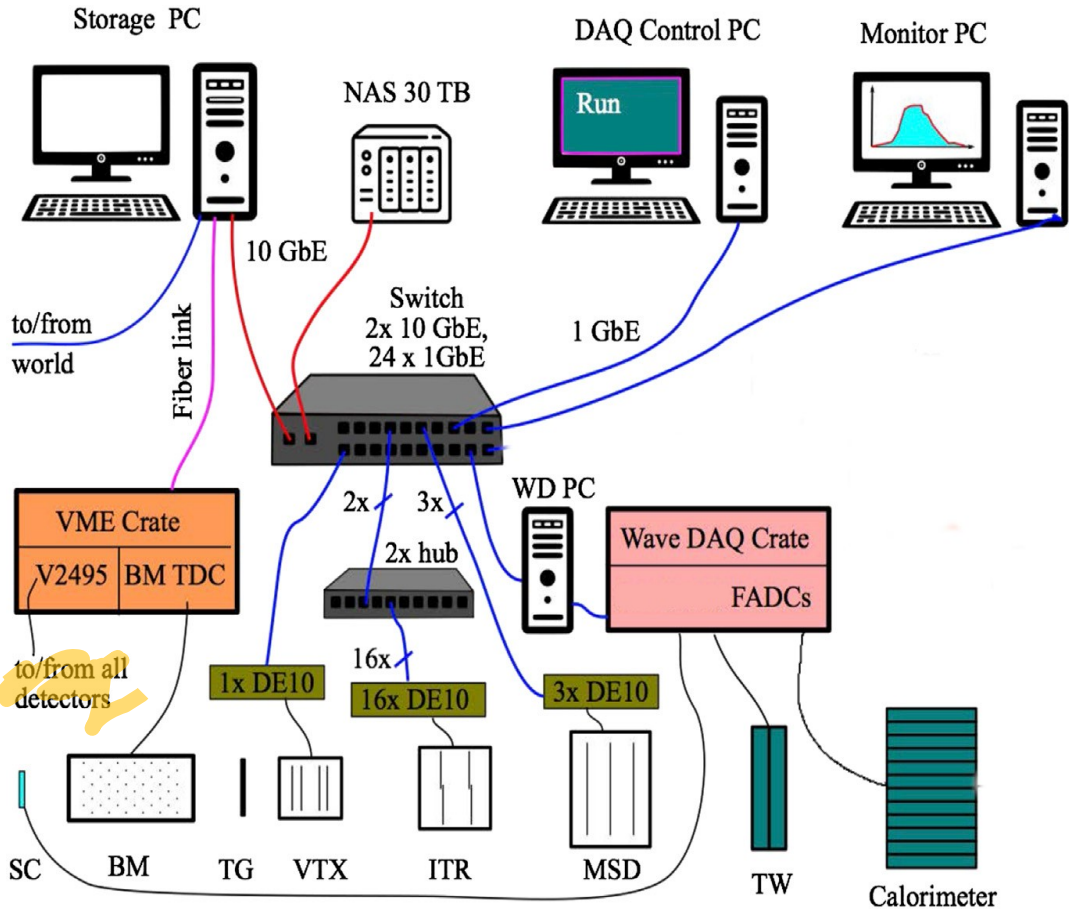
DAQ logical scheme

-**VME crate**:
Trigger and BM boards (TDC & discriminators)

-**3 DAQ PCs**:
Storage, control and monitoring

-**remote detectors**:
SC, VTX, IT, MSD, TW and CALO

-some of them have **intermediate PCs** performing some operations



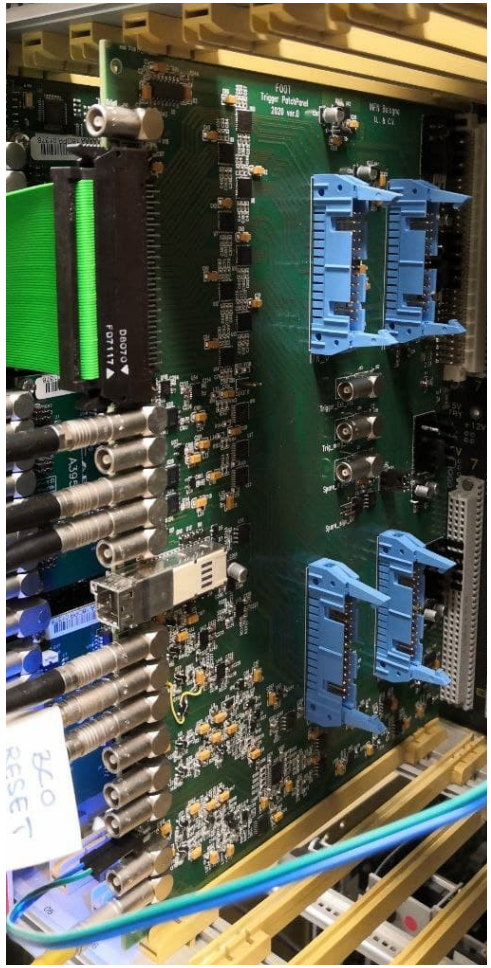
FOOT Trigger Patch Panel(s)

-**Motivation**:
Distribution of triggers, timestamp and busy signal from/ to detectors in a compact way

-**different types of signals** are handled for different detectors

-**highly integrated**, designed as a VME 6U board

-IT and MSD are treated in the same way (blue connectors), **interface boards** to receive and send signals will be needed



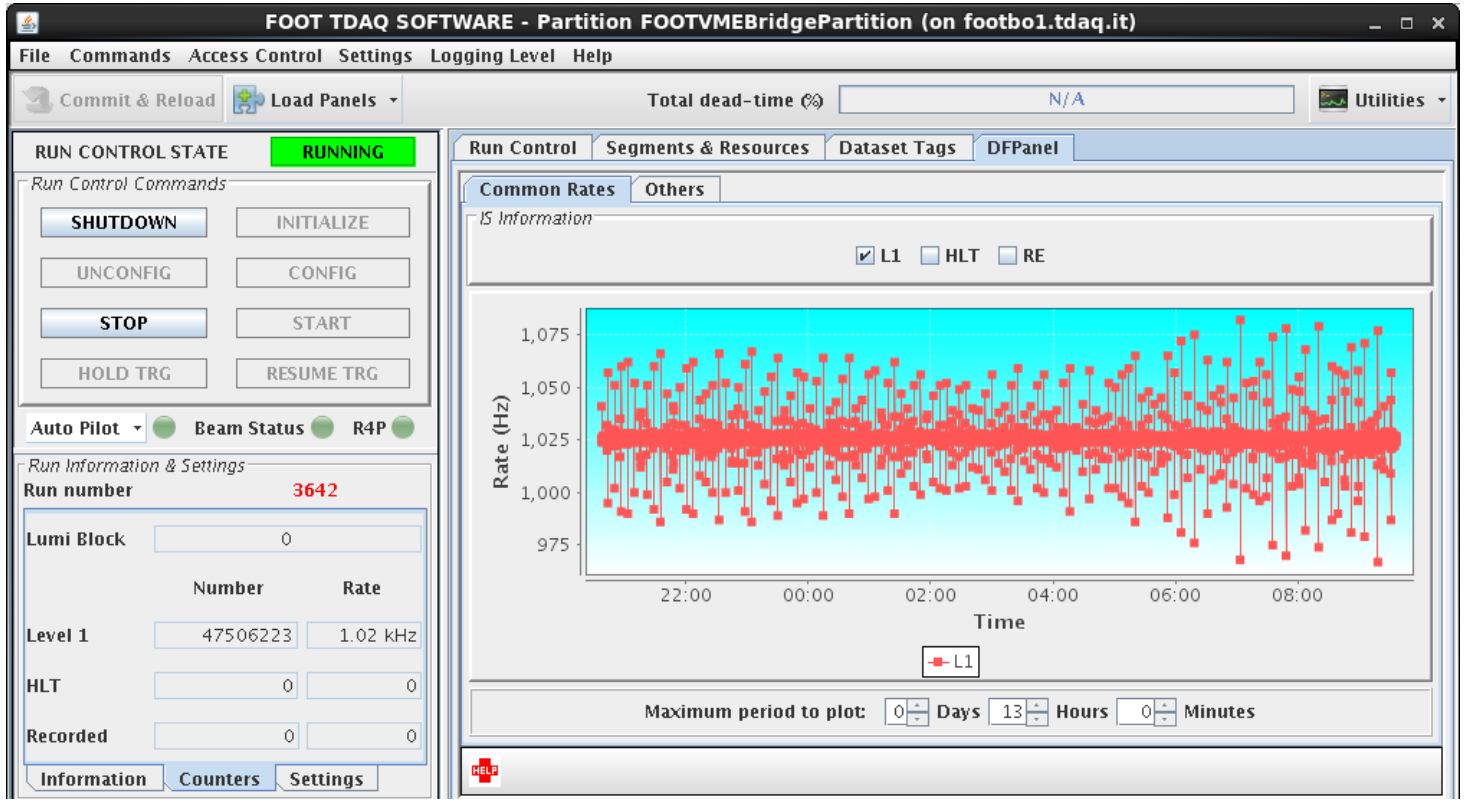
WaveDAQ test campaign

-**WaveDAQ: from Pisa to Bologna**
WaveDAQ system (Wavedream boards and PC) spent 10 days in Bologna this month

-**a lot of tests** were performed in view of GSI and other data takings

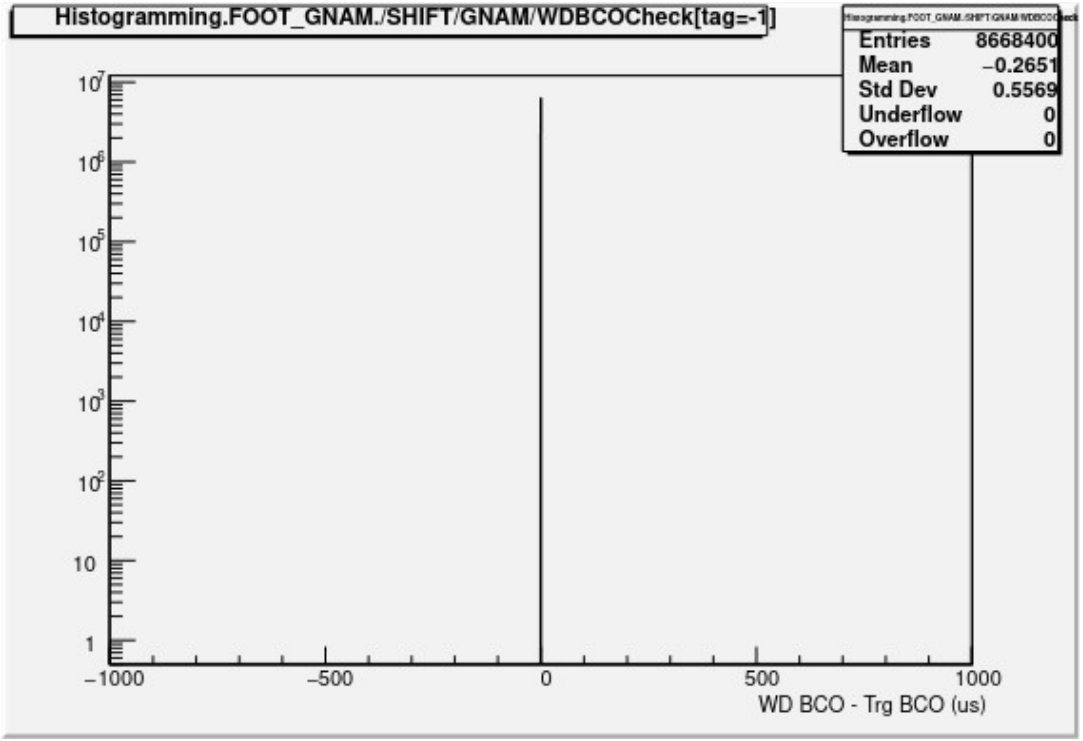
-**what we achieved**:
Long runs w/o errors (one in 10 hours), very good time alignment also with beam simulator, check of important signals, working data transmission, online monitoring improved, 2 kHz max rate with 40 kB events

WaveDAQ test campaign



Event size: 60 kB, 1 kHz stable rate
Long run with **48 million** events!

WaveDAQ test campaign



Very good time alignment between **trigger board** and **Wavedream!**

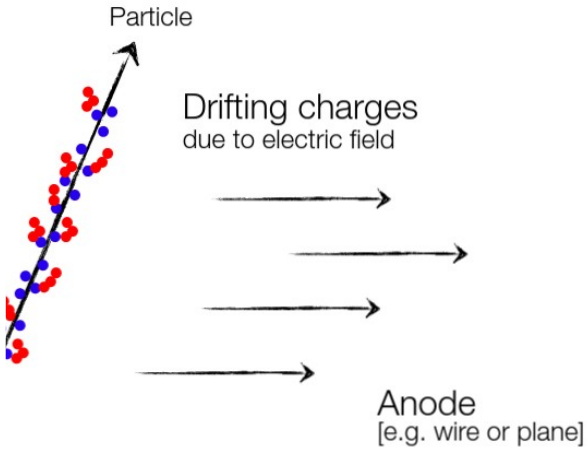
Evaluation of “t0” for beam monitor

-**Motivation**:
Precise measurement of direction of the track in BM needs a good **knowledge of the starting time** of the drift (~ns)

-TDC measurements are performed wrt to trigger signal

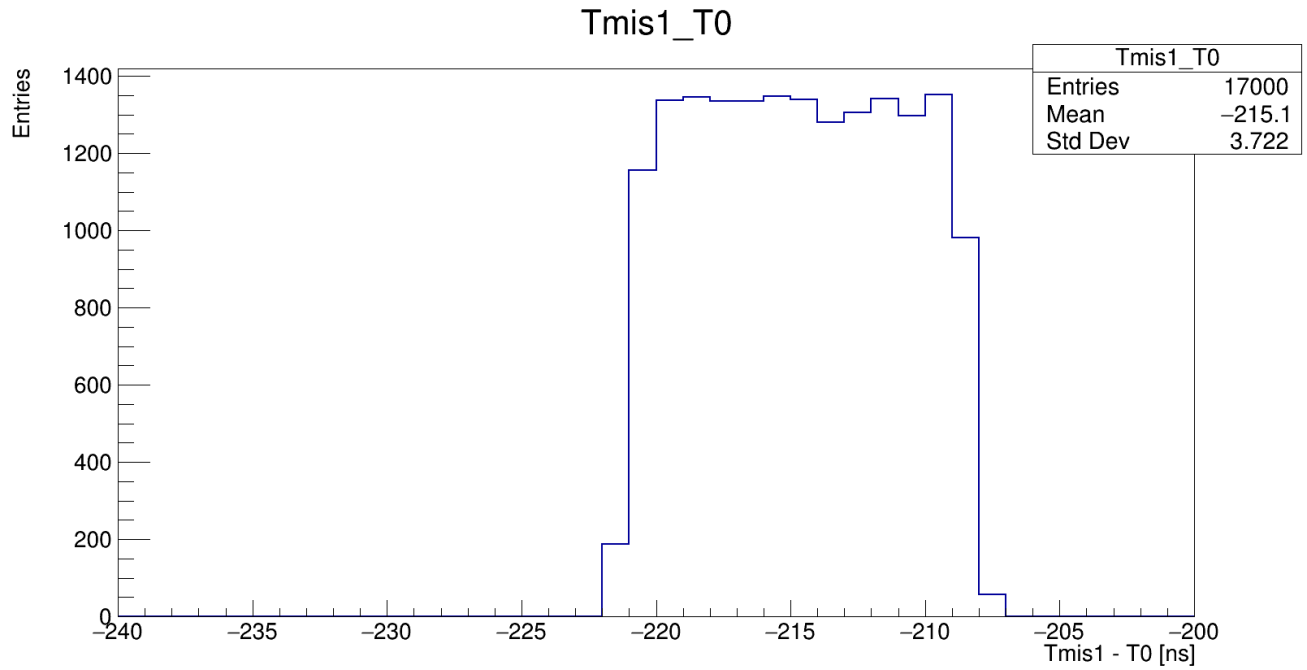
-trigger signal is forwarded to BM after passing through both the Wavedream and the central trigger board → jitters (~10 ns at GSI 2019)

-a “WD-only” trigger signal is now available in the patch panel but all jitters have to be characterised



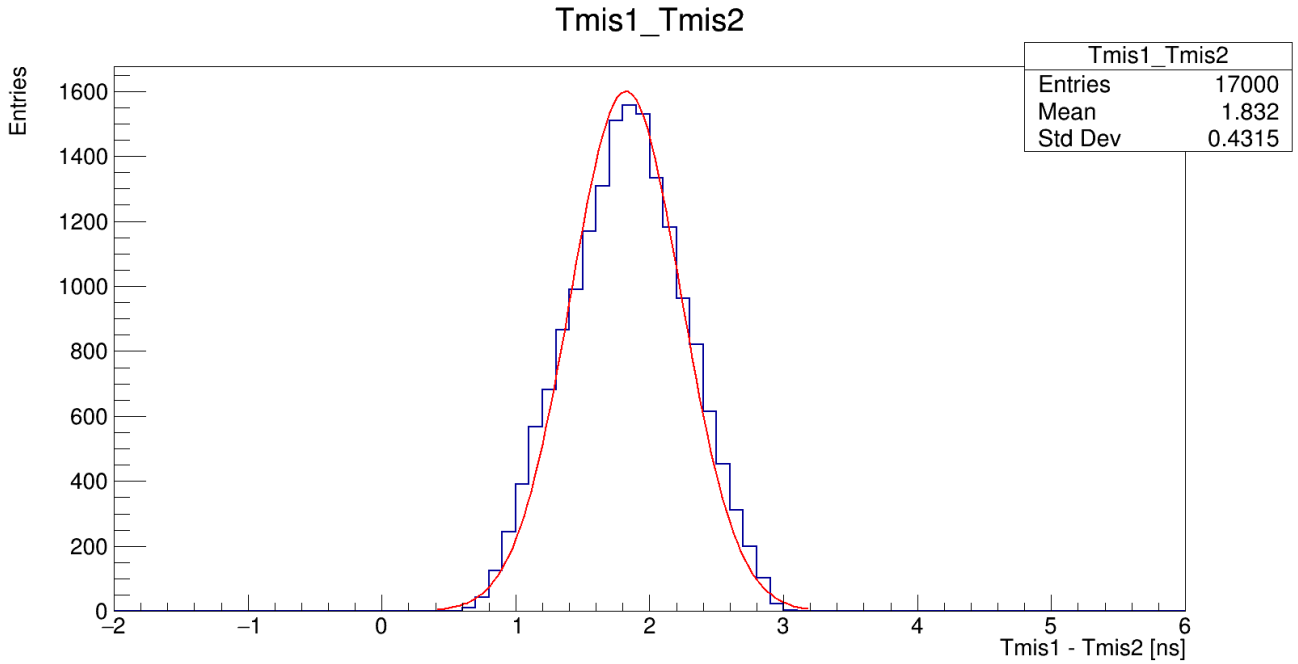
Measurement of jitter

With TDC only (new configuration)...



Measurement of jitter

...with TDC and WD



Thanks to recent design improvements, it will be possible to evaluate a T0 with a jitter **<0.5 ns**

MSD DAQ

-strong collaboration since last general meeting

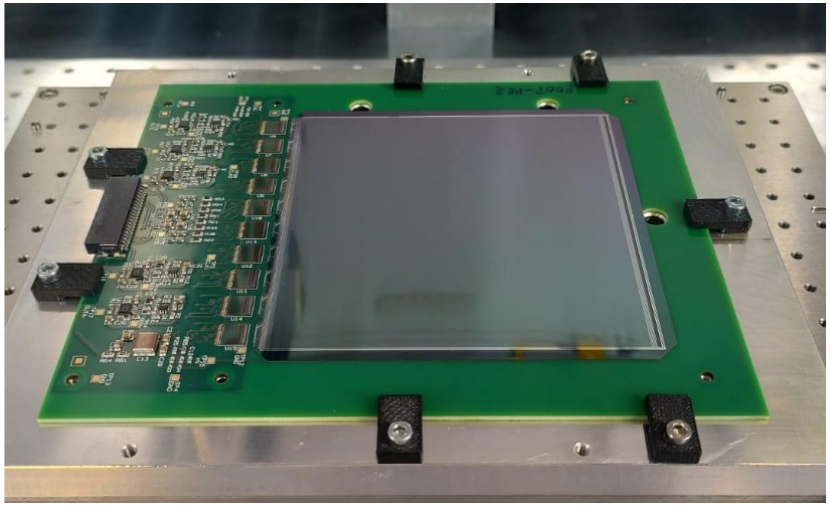
-MSD integration is going fast, first data with sensors were taken using FOOT general DAQ

-online monitoring is about to be defined

- **a joint MSD-TDAQ test beam in Trento is scheduled for 3-4-5 June**

- we plan to spot (and solve!) problems before GSI

-for details, see Mattia's talk later



VTX/IT DAQ

-a remote integration was attempted but not feasible (network limitations)

-a joint VTX-TDAQ lab test in Frascati is scheduled for the 3rd week of June

-online monitoring is to be defined

- we plan to spot (and solve!) problems before GSI

Reading raw data in SHOE

-The **EventReader class** fits the modular structure of the binary file

-The code provides a **C++ pointer** to each detector and it is possible to access information event by event

-The **TDAQ-SHOE interface** is in TAGdaqApi directory

-Several changes occurred since GSI and the interface **has to be updated** before July

-We will **circulate** soon some **binary files** to be tested with SHOE

```
aa aa 34 12 08 00 00 00  06 00 00 00 01 00 00 00  |..4.............|
75 3a 41 01 55 98 01 00  00 00 00 00 00 00 00 00  |u:A.U...........|
bb aa 34 12 08 00 00 00  46 4f 4f 54 2d 52 43 44  |..4.....FOOT-RCD|
39 00 00 00 64 61 74 61  5f 74 65 73 74 2e 30 30  |9...data_test.00|
30 30 33 37 35 30 2e 70  68 79 73 69 63 73 5f 66  |003750.physics_f|
6f 6f 74 2e 64 61 71 2e  52 41 57 2e 5f 6c 62 30  |oot.daq.RAW._lb0|
30 30 30 2e 5f 46 4f 4f  54 2d 52 43 44 20 20 20  |000._FOOT-RCD   |
bc aa 34 12 04 00 00 00  29 00 00 00 47 55 49 44  |..4.....)...GUID|
3d 43 36 35 45 38 42 45  37 2d 31 30 42 41 2d 45  |=C65E8BE7-10BA-E|
42 31 31 2d 42 30 34 39  2d 33 43 46 44 46 45 44  |B11-B049-3CFDFED|
30 31 34 34 38 20 20 20  13 00 00 00 53 74 72 65  |01448   ....Stre|
61 6d 3d 70 68 79 73 69  63 73 5f 66 6f 6f 74 20  |am=physics_foot |
11 00 00 00 50 72 6f 6a  65 63 74 3d 64 61 74 61  |....Project=data|
5f 74 65 73 74 20 20 20  0b 00 00 00 4c 75 6d 69  |_test    ....Lumi|
42 6c 6f 63 6b 3d 30 20  bb bb 34 12 0c 00 00 00  |Block=0 ..4.....|
a6 0e 00 00 00 00 00 00  01 00 00 00 00 00 00 00  |................|
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00 00 00 00 00 00 00 00  cc cc 34 12 04 00 00 00  |..........4.....|
01 00 00 00 d8 08 00 00  ee 34 12 ee 09 00 00 00  |.........4......|
00 00 01 03 30 52 46 00  a6 0e 00 00 00 00 00 00  |....0RF.........|
00 00 00 00 00 00 00 00  00 00 00 00 30 31 46 00  |............01F.|
4a 73 a7 60 44 3b 00 00  00 00 00 00 1f 00 00 40  |Js.`D;.........@|
c2 0e 00 08 02 00 00 18  c2 0e 00 09 02 00 00 19  |................|
c2 0e 00 0a ec 0f 38 02  ec 0f b8 02 04 00 00 1a  |......8.........|
c2 0e 00 0b 02 00 00 1b  2e 35 56 8c bf 01 00 80  |.........5V.....|
30 31 f0 e0 30 35 46 00  4a 73 a7 60 44 3b 00 00  |01..05F.Js.`D;..|
00 00 00 00 04 02 00 a0  01 00 00 00 00 00 00 00  |................|
e4 f7 de 03 4b 0f 4a 0f  48 0f 49 0f 4b 0f 48 0f  |....K.J.H.I.K.H.|
49 0f 49 0f 4b 0f 4b 0f  4b 0f 4b 0f 4b 0f 4e 0f  |I.I.K.K.K.K.K.N.|
4f 0f 51 0f 4e 0f 4e 0f  4d 0f 51 0f 51 0f 4c 0f  |O.Q.N.N.M.Q.Q.L.|
4e 0f 4f 0f 50 0f 50 0f  51 0f 53 0f 53 0f 54 0f  |N.O.P.P.Q.S.S.T.|
56 0f 5b 0f 54 0f 53 0f  55 0f 54 0f 55 0f 55 0f  |V.[.T.S.U.T.U.U.|
```

Reading raw data in SHOE

RemoteEvent.hh 511 Bytes

```
#ifndef REMOTEEVENT_HH
#define REMOTEEVENT_HH

#include "reader/BaseFragment.hh"
#include <vector>

class RemoteEvent : public BaseFragment {

  public:
    u_int time_sec;
    u_int time_usec;
    u_int eventNumber;

    u_int evtSize;
    std::vector<u_int> values;

    virtual ~RemoteEvent();

    virtual void readData(unsigned int **p);
    virtual void printData() const;
    virtual bool check() const { return true;};

    virtual std::string classType() const {return "RemoteEvent";};

};

#endif
```

WDEvent.hh 619 Bytes

```
#ifndef WDEVENT_HH
#define WDEVENT_HH

#include "reader/RemoteEvent.hh"

class WDEvent : public RemoteEvent {

public:
  //firmware data
  u_int detectorHeader;
  u_int boardHeader;
  u_int numberWords;
  u_int hardwareEventNumber;
  u_int BCOofTrigger;
  u_int numBoards;
  u_int TWChans;
  u_int CaloChans;
  u_int trigType;
  uint64_t TriggerPattern;
  uint64_t TriggerGenerationBin[32];


  virtual ~WDEvent();

  virtual void readData(unsigned int **p);
  virtual void printData() const;
  virtual bool check() const{return true;};

  virtual std::string classType() const {return "WDEvent";};

};

#endif
```

Reading raw data in SHOE

RemoteEvent.hh 511 Bytes

```
#ifndef REMOTEEVENT_HH
#define REMOTEEVENT_HH

#include "reader/BaseFragment.hh"
#include <vector>

class RemoteEvent : public BaseFragment {

  public:
    u_int time_sec;
    u_int time_usec;
    u_int eventNumber;

    u_int evtSize;
    std::vector<u_int> values;
```

```
// Global event information
//InfoEvent* evInfo = daqFileReader.getInfoEvent();
// Trigger data
TrgEvent*  evTrg  = checkWD.getTriggerEvent();

// TDC # 0 and # 1
//const TDCEvent* evTDC0 =                                    // tdc # 0
//static_cast<const TDCEvent*>(daqFileReader.getFragmentID(dataV1190 | 0x30));
//const TDCEvent* evTDC1 =                                    // tdc # 1
    //static_cast<const TDCEvent*>(daqFileReader.getFragmentID(dataV1190 | 0x31));

//if( evTDC0!=NULL ) evTDC0->printData();  // example how to use
//if( evTDC1!=NULL ) evTDC1->printData();

const WDEvent* evWD = static_cast<const WDEvent*>(checkWD.getFragmentID(dataWD | 0x30));
```

WDEvent.hh 619 Bytes

```
#ifndef WDEVENT_HH
#define WDEVENT_HH

#include "reader/RemoteEvent.hh"

class WDEvent : public RemoteEvent {

public:
  //firmware data
  u_int detectorHeader;
  u_int boardHeader;
  u_int numberWords;
  u_int hardwareEventNumber;
  u_int BCOofTrigger;
  u_int numBoards;
  u_int TWChans;
  u_int CaloChans;
  u_int trigType;
  uint64_t TriggerPattern;
  uint64_t TriggerGenerationBin[32];


  virtual ~WDEvent();

  virtual void readData(unsigned int **p);
  virtual void printData() const;
  virtual bool check() const{return true;};

  virtual std::string classType() const {return "WDEvent";};

};

#endif
```

Monitoring tools

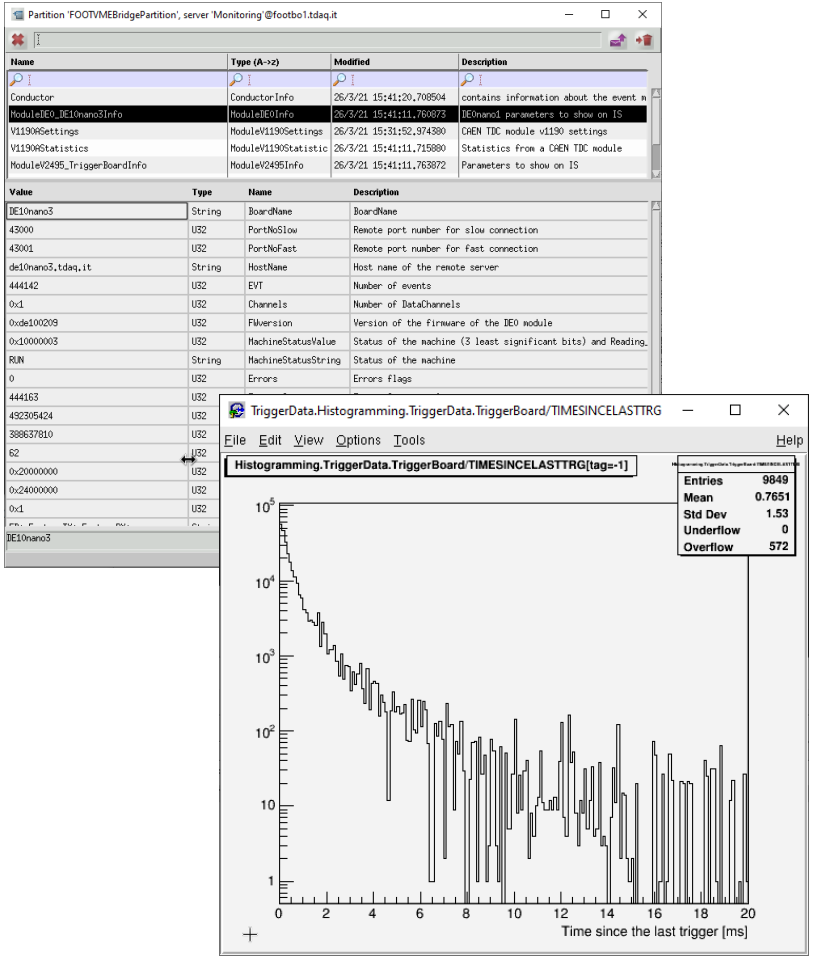
-In FOOT we have three tools for **online monitoring**: Information Service (IS), Online Histogram (OH) and GNAM

-**IS**: publish or read information on the run, boards, system, trigger rates

-**OH**: histograms with information of one detector, filled in each event (need to select only useful information)

-**GNAM**: full-event monitoring sampler

-**Database** (offline): it contains configuration of all boards and information about the run → some infos still to be added!



GNAM@FOOT

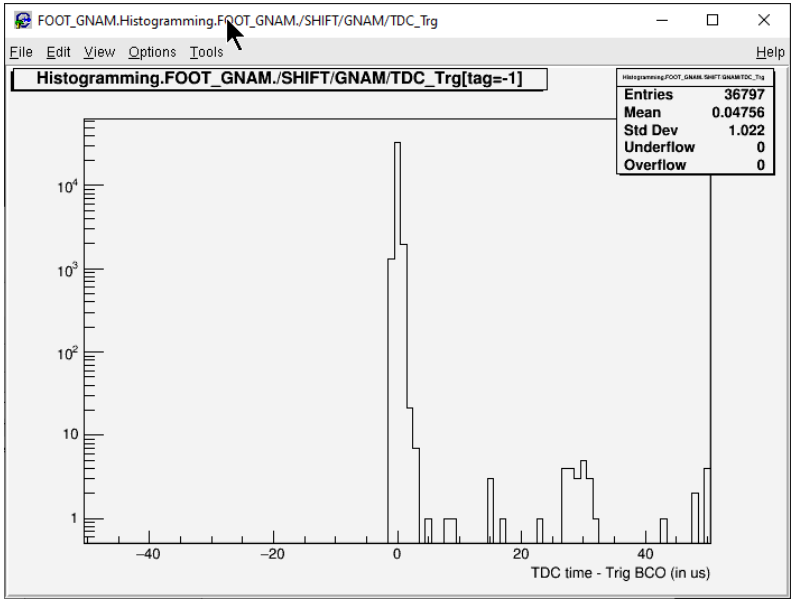
-Monitoring tool, not time critical

-It is a **sampler of events**, but it sees the full FOOT event

-We can monitor deeply inside a detector and **correlate** also different **detectors**

-Useful at least to check for time correlations and for **beam shape**

-In principle we can search for **tracks,** interactions, fragmentations and we can make a full event reconstruction



TDC time tag – TS time

Event size

from CDR

Detector	Board(s)	DAQ channels	max event rate (kHz)	Event size (bytes)
Trigger	V2495	1	10	40 B
Start Counter	DreamWave	4	1	8.2 kB
Beam Monitor	TDC	36	5	0.1 kB
Vertex detector	SoC on DEx	$4 \cdot 10^6$	2	0.9 kB
Inner tracker	SoC on DEx	$28 \cdot 10^6$	2	2.1 kB
Outer tracker	Custom	$6 \cdot 10^3$	2	0.5 kB
$\Delta E/\Delta x$	DreamWave	80	1	8.4 kB
Calorimeter	QDC	400	2	1.7 kB
Total DAQ	Storage PC	-	1	22 kB

Actual event size:
- Trigger + BM + VTX → ~2kB/event
- SC + TW + CALO → ~60 kB/event
- MSD → 8 kB/event

---

70 kB/event

@1kHz → **70 MB/s**!!

Ready for GSI (conclusions)

SC + TW + CALO:
joint lab test in Bologna **already done**
Some minor changes will be needed, overall status OK

VTX:
joint lab test in Frascati **to be done** (3rd week of June)

MSD:
joint test beam in Trento **to be done** (3-4-5 June)

Thanks for your attention!

Description of the setup