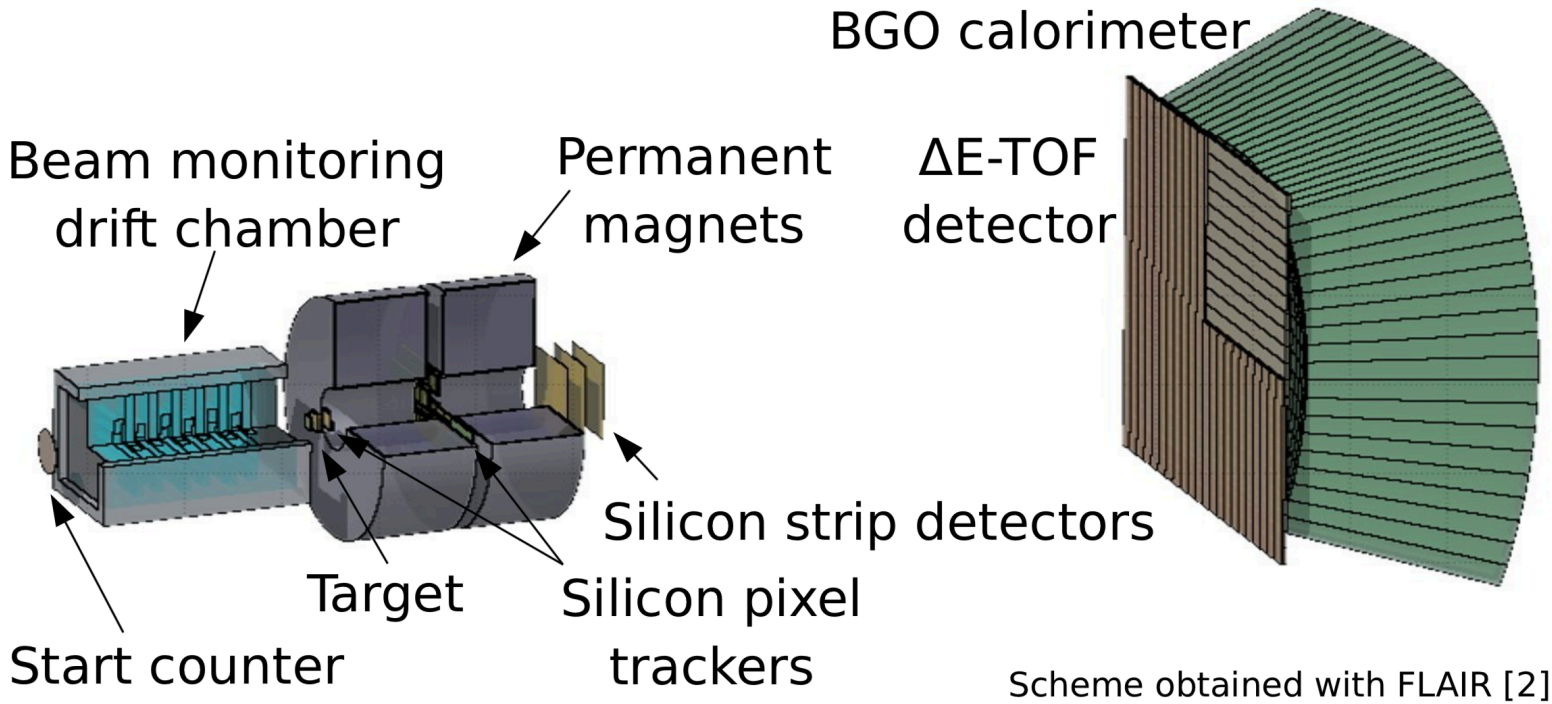
ΔE-TOF detector and software update

INFN and University of Pisa group

Outline

- Introduction
- Summary of data acquisition and data structure
- Simulation status (short)
- Status SHOE software implementation
- Outlook and conclusions

Introduction

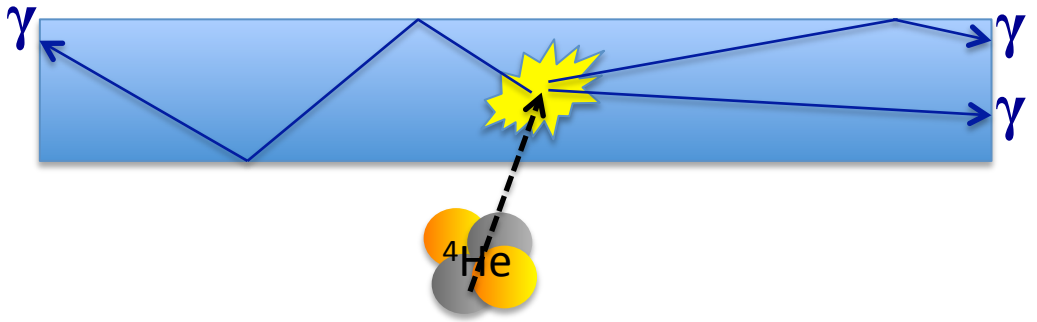


Scheme obtained with FLAIR [2]

- 2 layers of **plastic scintillators**, 20+20 bars, 44 x 2 x 0.3 cm^3
- Double-side **SiPM** read-out

Introduction

- ΔE-TOF detector: based on scintillator bars

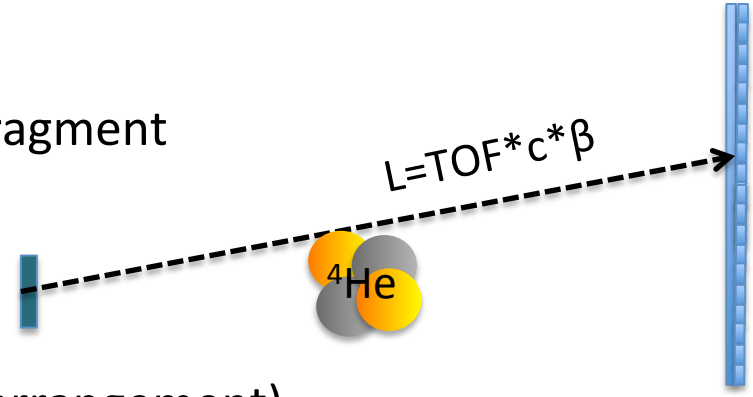


- ΔE-TOF detector provides
  - **Time-Of-Flight** of fragments
    - $TOF=t_{det}-t_{trig}=(t_l+t_r)/2 - t_{trig}$ → velocity **β** **QUESTION**
  - **Energy** deposited (from charge-energy calibration)

$$-\frac{\Delta E}{dx} = Kz^2 \frac{Z}{A}\frac{1}{\beta^2}[\frac{1}{2}\ln\frac{2m_e c^2\beta^2\gamma^2 W_{\max}}{I^2}$$

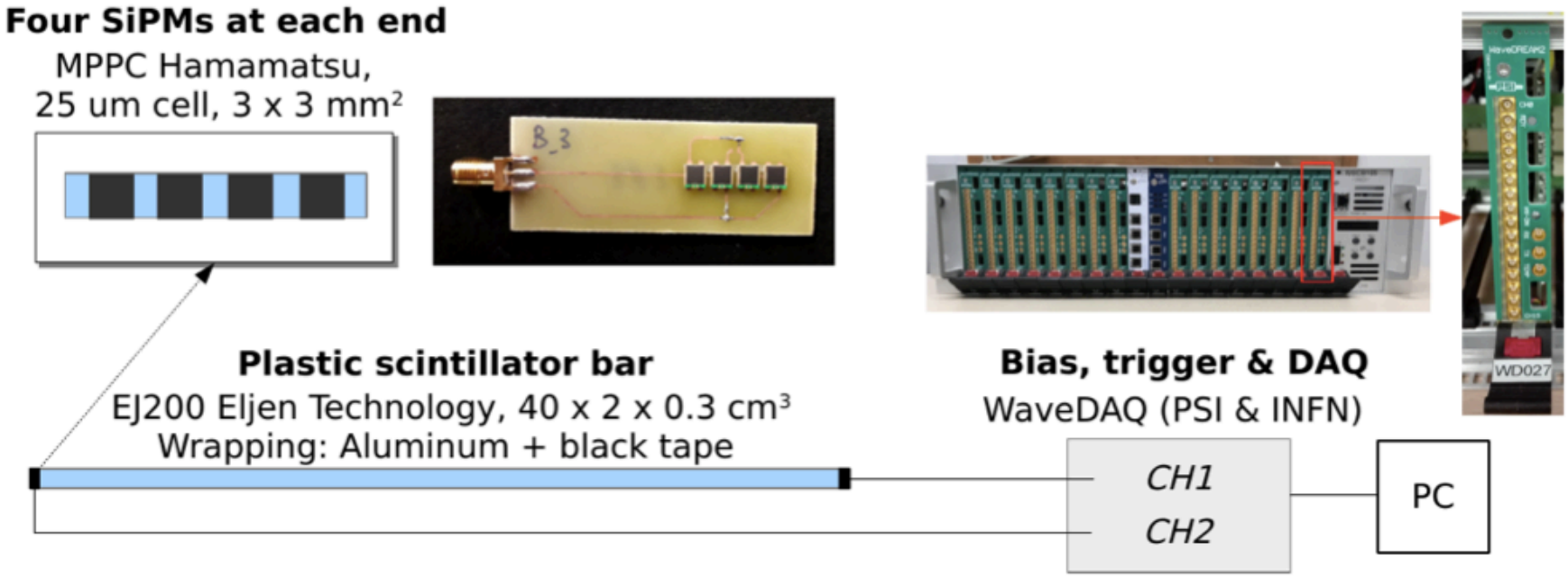
$$\rightarrow z = \beta\sqrt{\frac{E_{bar}}{d}\frac{1}{K\frac{Z}{A}[\frac{1}{2}\ln\frac{2m_e c^2\beta^2\gamma^2 W_{\max}}{I^2}]}})$$

→**Z** of fragment



  - **Position** of deposit (2D by orthogonal arrangement)
    - By determining $T_{lr}=t_l-t_r$
    - By ratio of collected energies $\ln(E_l/E_r)$

Data acquisition



- SiPM signal amplified and sent to fast digitizer (5 Gsamples/s)
- DAQ system (WaveDAQ: PSI-INFN):
  - Based on DRS-ASIC developed at PSI (Stefan Ritt)
  - Channels from each bar connected to custom board WaveDREAM (WDB)
  - Can readout 16 (+2 synch)channels (22 bars in one plane, so 44 channels → 88 channels for 2 planes in FOOT) [**So 6 boards needed**]
  - Switchable gain amplifiers from 0.5 up to 100 in steps
  - Can provide power for SiPMs
- Connected to trigger board

→ Output is binary file

Data structure

- C++ macro to decode the signal to waveforms in root format
  - to be copied and adjusted in SHOE

```
#include <string.h>
#include <stdio.h>
#include "TFile.h"
#include "TTree.h"
#include "TString.h"
#include "TGraph.h"
#include "TCanvas.h"
#include "Getline.h"

typedef struct {
   char           tag[3];
   char           version;
} FHEADER;

typedef struct {
   char           time_header[4];
} THEADER;

typedef struct {
   char           bn[2];
   unsigned short board_serial_number;
} BHEADER;

typedef struct {
   char           event_header[4];
   unsigned int   event_serial_number;
   unsigned short year;
   unsigned short month;
   unsigned short day;
   unsigned short hour;
   unsigned short minute;
. . . .
```

Luca Galli, Stefan Ritt

PAUL SCHERRER INSTITUT

uation Board User's Manual

Word	Byte 0	Byte 1	Byte 2	Byte 3	Contents
0	'D'	'R'	'S'	'2'	File header, Byte 3 = version
1	'T'	'I'	'M'	'E'	Time Header
2	'B'	'#'	Board number		Board serial number
3	'C'	'0'	'0'	'1'	Channel 1 header
4	Time Bin Width #0				Effective time bin width in ns for channel 1 encoded in 4-Byte floating point format
5	Time Bin Width #1				
…	…				
1027	Time Bin Width #1023				
1028	'C'	'0'	'0'	'2'	Channel 2 header
1029	Time Bin Width #0				Effective time bin width in ns for channel 2 encoded in 4-Byte floating point format
1030	Time Bin Width #1				
…	…				
2052	Time Bin Width #1023				
2053	'E'	'H'	'D'	'R'	Event Header
2054	Event Serial Number				Serial number starting with 1
2055	Year		Month		Event date/time 16-bit values
2056	Day		Hour		
2057	Minute		Second		
2058	Millisecond		Range		Range center (RC) in mV
2059	'B'	'#'	Board number		Board serial number
2060	'T'	'#'	Trigger cell		Number of first readout cell
2061	'C'	'0'	'0'	'1'	Channel 1 header
2062	Scaler #1				Scaler for channel 1 in Hz
2063	Voltage Bin #0		Voltage Bin #1		Channel 1 waveform data encoded in 2-Byte integers. 0=RC-0.5V and 65535=RC+0.5V. RC see header.
2064	Voltage Bin #2		Voltage Bin #3		
…	…		…		
2574	Voltage Bin #1022		Voltage Bin #1023		
2575	'C'	'0'	'0'	'2'	Channel 2 header
2576	Scaler #2				Scaler for channel 2 in Hz
2577	Voltage Bin #0		Voltage Bin #1		Channel 2 waveform data encoded in 2-Byte integers. 0=RC-0.5V and 65535=RC+0.5V. RC see header.
2578	Voltage Bin #2		Voltage Bin #3		
…	…		…		
3088	Voltage Bin #1022		Voltage Bin #1023		
3089	'E'	'H'	'D'	'R'	Next Event Header
…					

Data structure

- C++ macro to decode the signal to waveforms in root format
  - to be copied and adjusted in SHOE

```
#include <string.h>
#include <stdio.h>
#include "TFile.h"
#include "TTree.h"
#include "TString.h"
#include "TGraph.h"
#include "TCanvas.h"
#include "Getline.h"

typedef struct {
   char           tag[3];
   char           version;
} FHEADER;

typedef struct {
   char           time_header[4];
} THEADER;

typedef struct {
   char           bn[2];
   unsigned short board_serial_number;
} BHEADER;

typedef struct {
   char           event_header[4];
   unsigned int   event_serial_number;
   unsigned short year;
   unsigned short month;
   unsigned short day;
   unsigned short hour;
   unsigned short minute;
. . . .
```

Luca Galli, Stefan Ritt

PAUL SCHERRER INSTITUT

uation Board User's Manual

Word	Byte 0	Byte 1	Byte 2	Byte 3	Contents
0	'D'	'R'			File header, Byte 3 = version
1	'T'	'I'	'M'	'E'	Time Header
2	'B'	'#'	Board number		Board serial number
3	'C'	'0'	'0'	'1'	Channel 1 header
4	Time Bin Width #0				Effective time bin width in ns for channel 1 encoded in 4-Byte floating point format
5	Time Bin Width #1				
…	…				
1027	Time Bin Width #1023				
1028	'C'	'0'	'0'	'2'	Channel 2 header
1029	Time Bin Width #0				Effective time bin width in ns for channel 2 encoded in 4-Byte floating point format
1030	Time Bin Width #1				
…	…				
2052	Time Bin Width #1023				
2053	'E'	'H'	'D'	'R'	Event Header
2054	Event Serial Number				Serial number starting with 1
2055	Year		Month		Event date/time 16-bit values
2056	Day		Hour		
2057	Minute		Second		
2058	Millisecond		Range		Range center (RC) in mV
2059	'B'	'#'	Board number		Board serial number
2060	'T'	'#'	Trigger cell		Number of first readout cell
2061	'C'	'0'	'0'	'1'	Channel 1 header
2062	Scaler #1				Scaler for channel 1 in Hz
2063	Voltage Bin #0		Voltage Bin #1		Channel 1 waveform data encoded in 2-Byte integers. 0=RC-0.5V and 65535=RC+0.5V. RC see header.
2064	Voltage Bin #2		Voltage Bin #3		
…	…		…		
2574	Voltage Bin #1022		Voltage Bin #1023		
2575	'C'	'0'	'0'	'2'	Channel 2 header
2576	Scaler #2				Scaler for channel 2 in Hz
2577	Voltage Bin #0		Voltage Bin #1		Channel 2 waveform data encoded in 2-Byte integers. 0=RC-0.5V and 65535=RC+0.5V. RC see header.
2578	Voltage Bin #2		Voltage Bin #3		
…	…		…		
3088	Voltage Bin #1022		Voltage Bin #1023		
3089	'E'	'H'	'D'	'R'	Next Event Header
…					

File header

Data structure

- C++ macro to decode the signal to waveforms in root format
  - to be copied and adjusted in SHOE

```
#include <string.h>
#include <stdio.h>
#include "TFile.h"
#include "TTree.h"
#include "TString.h"
#include "TGraph.h"
#include "TCanvas.h"
#include "Getline.h"

typedef struct {
   char           tag[3];
   char           version;
} FHEADER;

typedef struct {
   char           time_header[4];
} THEADER;

typedef struct {
   char           bn[2];
   unsigned short board_serial_number;
} BHEADER;

typedef struct {
   char           event_header[4];
   unsigned int   event_serial_number;
   unsigned short year;
   unsigned short month;
   unsigned short day;
   unsigned short hour;
   unsigned short minute;
. . . .
```

Luca Galli, Stefan Ritt

PAUL SCHERRER INSTITUT

uation Board User's Manual

Word	Byte 0	Byte 1	Byte 2	Byte 3	Contents
0	'D'	'R'	'S'	'2'	File header, Byte 3 = version
1	'T'	'I'	'M'	'2'	Time header
2	'B'	'#'	Board number		Board serial number
3	'C'	'0'	'0'	'1'	Channel 1 header
4	Time Bin Width #0				Effective time bin width in ns for channel 1 encoded in 4-Byte floating point format
5	Time Bin Width #1				
…	…				
1027	Time Bin Width #1023				
1028	'C'	'0'	'0'	'2'	Channel 2 header
1029	Time Bin Width #0				Effective time bin width in ns for channel 2 encoded in 4-Byte floating point format
1030	Time Bin Width #1				
…	…				
2052	Time Bin Width #1023				
2053	'E'	'H'	'D'	'R'	Event Header
2054	Event Serial Number				Serial number starting with 1
2055	Year		Month		Event date/time 16-bit values
2056	Day		Hour		
2057	Minute		Second		
2058	Millisecond		Range		Range center (RC) in mV
2059	'B'	'#'	Board number		Board serial number
2060	'T'	'#'	Trigger cell		Number of first readout cell
2061	'C'	'0'	'0'	'1'	Channel 1 header
2062	Scaler #1				Scaler for channel 1 in Hz
2063	Voltage Bin #0		Voltage Bin #1		Channel 1 waveform data encoded in 2-Byte integers. 0=RC-0.5V and 65535=RC+0.5V. RC see header.
2064	Voltage Bin #2		Voltage Bin #3		
…	…		…		
2574	Voltage Bin #1022		Voltage Bin #1023		
2575	'C'	'0'	'0'	'2'	Channel 2 header
2576	Scaler #2				Scaler for channel 2 in Hz
2577	Voltage Bin #0		Voltage Bin #1		Channel 2 waveform data encoded in 2-Byte integers. 0=RC-0.5V and 65535=RC+0.5V. RC see header.
2578	Voltage Bin #2		Voltage Bin #3		
…	…		…		
3088	Voltage Bin #1022		Voltage Bin #1023		
3089	'E'	'H'	'D'	'R'	Next Event Header
…					

Data structure

- C++ macro to decode the signal to waveforms in root format
  - to be copied and adjusted in SHOE

```
#include <string.h>
#include <stdio.h>
#include "TFile.h"
#include "TTree.h"
#include "TString.h"
#include "TGraph.h"
#include "TCanvas.h"
#include "Getline.h"

typedef struct {
   char           tag[3];
   char           version;
} FHEADER;

typedef struct {
   char           time_header[4];
} THEADER;

typedef struct {
   char           bn[2];
   unsigned short board_serial_number;
} BHEADER;

typedef struct {
   char           event_header[4];
   unsigned int   event_serial_number;
   unsigned short year;
   unsigned short month;
   unsigned short day;
   unsigned short hour;
   unsigned short minute;
. . . .
```

Luca Galli, Stefan Ritt

PAUL SCHERRER INSTITUT

uation Board User's Manual

Board header

Word	Byte 0	Byte 1	Byte 2	Byte 3	Contents
0	'D'	'R'	'S'	'2'	File header, Byte 3 = version
1	'T'	'I'	'M'	'E'	Time Header
2	'B'	'#'	Board number		Board serial number
3	'C'	'0'	'0'	'1'	Channel 1 header
4	Time Bin Width #0				Effective time bin width in ns for channel 1 encoded in 4-Byte floating point format
5	Time Bin Width #1				
…	…				
1027	Time Bin Width #1023				
1028	'C'	'0'	'0'	'2'	Channel 2 header
1029	Time Bin Width #0				Effective time bin width in ns for channel 2 encoded in 4-Byte floating point format
1030	Time Bin Width #1				
…	…				
2052	Time Bin Width #1023				
2053	'E'	'H'	'D'	'R'	Event Header
2054	Event Serial Number				Serial number starting with 1
2055	Year		Month		Event date/time 16-bit values
2056	Day		Hour		
2057	Minute		Second		
2058	Millisecond		Range		Range center (RC) in mV
2059	'B'	'#'	Board number		Board serial number
2060	'T'	'#'	Trigger cell		Number of first readout cell
2061	'C'	'0'	'0'	'1'	Channel 1 header
2062	Scaler #1				Scaler for channel 1 in Hz
2063	Voltage Bin #0		Voltage Bin #1		Channel 1 waveform data encoded in 2-Byte integers. 0=RC-0.5V and 65535=RC+0.5V. RC see header.
2064	Voltage Bin #2		Voltage Bin #3		
…	…		…		
2574	Voltage Bin #1022		Voltage Bin #1023		
2575	'C'	'0'	'0'	'2'	Channel 2 header
2576	Scaler #2				Scaler for channel 2 in Hz
2577	Voltage Bin #0		Voltage Bin #1		Channel 2 waveform data encoded in 2-Byte integers. 0=RC-0.5V and 65535=RC+0.5V. RC see header.
2578	Voltage Bin #2		Voltage Bin #3		
…	…		…		
3088	Voltage Bin #1022		Voltage Bin #1023		
3089	'E'	'H'	'D'	'R'	Next Event Header
…					

Data structure

- C++ macro to decode the signal to waveforms in root format
  - to be copied and adjusted in SHOE

```
#include <string.h>
#include <stdio.h>
#include "TFile.h"
#include "TTree.h"
#include "TString.h"
#include "TGraph.h"
#include "TCanvas.h"
#include "Getline.h"

typedef struct {
   char           tag[3];
   char           version;
} FHEADER;

typedef struct {
   char           time_header[4];
} THEADER;

typedef struct {
   char           bn[2];
   unsigned short board_serial_number;
} BHEADER;

typedef struct {
   char           event_header[4];
   unsigned int   event_serial_number;
   unsigned short year;
   unsigned short month;
   unsigned short day;
   unsigned short hour;
   unsigned short minute;
. . . .
```

Luca Galli, Stefan Ritt

PAUL SCHERRER INSTITUT

uation Board User's Manual

Word	Byte 0	Byte 1	Byte 2	Byte 3	Contents
0	'D'	'R'	'S'	'2'	File header, Byte 3 = version
1	'T'	'I'	'M'	'E'	Time Header
2	'B'	'#'	Board number		Board serial number
3	'C'	'0'	'0'	'1'	Channel 1 header
4	Time Bin Width #0				Effective time bin width in ns for channel 1 encoded in 4-Byte floating point format
5	Time Bin Width #1				
…	…				
1027	Time Bin Width #1023				
1028	'C'	'0'	'0'	'2'	Channel 2 header
1029	Time Bin Width #0				Effective time bin width in ns for channel 2 encoded in 4-Byte floating point format
1030	Time Bin Width #1				
…	…				
2052	Time Bin Width #1023				
2053	'E'	'H'			Event Header
2054	Event Serial Number				Serial number starting with 1
2055	Year		Month		Event date/time 16-bit values
2056	Day		Hour		
2057	Minute		Second		
2058	Millisecond		Range		Range center (RC) in mV
2059	'B'	'#'	Board number		Board serial number
2060	'T'	'#'	Trigger cell		Number of first readout cell
2061	'C'	'0'	'0'	'1'	Channel 1 header
2062	Scaler #1				Scaler for channel 1 in Hz
2063	Voltage Bin #0		Voltage Bin #1		Channel 1 waveform data encoded in 2-Byte integers. 0=RC-0.5V and 65535=RC+0.5V. RC see header.
2064	Voltage Bin #2		Voltage Bin #3		
…	…		…		
2574	Voltage Bin #1022		Voltage Bin #1023		
2575	'C'	'0'	'0'	'2'	Channel 2 header
2576	Scaler #2				Scaler for channel 2 in Hz
2577	Voltage Bin #0		Voltage Bin #1		Channel 2 waveform data encoded in 2-Byte integers. 0=RC-0.5V and 65535=RC+0.5V. RC see header.
2578	Voltage Bin #2		Voltage Bin #3		
…	…		…		
3088	Voltage Bin #1022		Voltage Bin #1023		
3089	'E'	'H'	'D'	'R'	Next Event Header
…					

Event header

Data structure

- C++ macro to decode the signal to waveforms in root format
  - to be copied and adjusted in SHOE

```
#include <string.h>
#include <stdio.h>
#include "TFile.h"
#include "TTree.h"
#include "TString.h"
#include "TGraph.h"
#include "TCanvas.h"
#include "Getline.h"

typedef struct {
   char           tag[3];
   char           version;
} FHEADER;

typedef struct {
   char           time_header[4];
} THEADER;

typedef struct {
   char           bn[2];
   unsigned short board_serial_number;
} BHEADER;

typedef struct {
   char           event_header[4];
   unsigned int   event_serial_number;
   unsigned short year;
   unsigned short month;
   unsigned short day;
   unsigned short hour;
   unsigned short minute;
. . . .
```

Luca Galli, Stefan Ritt

PAUL SCHERRER INSTITUT

uation Board User's Manual

Word	Byte 0	Byte 1	Byte 2	Byte 3	Contents
0	'D'	'R'	'S'	'2'	File header, Byte 3 = version
1	'T'	'I'	'M'	'E'	Time Header
2	'B'	'#'	Board number		Board serial number
3	'C'	'0'	'0'	'1'	Channel 1 header
4	Time Bin Width #0				Effective time bin width in ns for channel 1 encoded in 4-Byte floating point format
5	Time Bin Width #1				
…	…				
1027	Time Bin Width #1023				
1028	'C'	'0'	'0'	'2'	Channel 2 header
1029	Time Bin Width #0				Effective time bin width in ns for channel 2 encoded in 4-Byte floating point format
1030	Time Bin Width #1				
…	…				
2052	Time Bin Width #1023				
2053	'E'	'H'	'D'	'R'	Event Header
[illegible]	Event Serial Number				Serial number starting with 1
2055	Year		Month		Event date/time 16-bit values
2056	Day		Hour		
2057	Minute		Second		
2058	Millisecond		[illegible]		Range center (RC) in mV
2059	'B'	'#'	Board number		Board serial number
2060	'T'	'#'	[illegible]		Number of last readout cell
2061	'C'	'0'	'0'	'1'	Channel 1 header
2062	[illegible]				Scaler for channel 1 in Hz
2063	Voltage Bin #0		Voltage Bin #1		Channel 1 waveform data encoded in 2-Byte integers. 0=RC-0.5V and 65535=RC+0.5V. RC see header.
2064	Voltage Bin #2		Voltage Bin #3		
…	…		…		
2574	Voltage Bin #1022		Voltage Bin #1023		
2575	'C'	'0'	'0'	'2'	Channel 2 header
2576	Scaler #2				Scaler for channel 2 in Hz
2577	Voltage Bin #0		Voltage Bin #1		Channel 2 waveform data encoded in 2-Byte integers. 0=RC-0.5V and 65535=RC+0.5V. RC see header.
2578	Voltage Bin #2		Voltage Bin #3		
…	…		…		
3088	Voltage Bin #1022		Voltage Bin #1023		
3089	'E'	'H'	'D'	'R'	Next Event Header
…					

Event info (board header, board data, channel info)

Data structure

- C++ macro to decode the signal to waveforms in root format
  - to be copied and adjusted in SHOE

```
#include <string.h>
#include <stdio.h>
#include "TFile.h"
#include "TTree.h"
#include "TString.h"
#include "TGraph.h"
#include "TCanvas.h"
#include "Getline.h"

typedef struct {
   char           tag[3];
   char           version;
} FHEADER;

typedef struct {
   char           time_header[4];
} THEADER;

typedef struct {
   char           bn[2];
   unsigned short board_serial_number;
} BHEADER;

typedef struct {
   char           event_header[4];
   unsigned int   event_serial_number;
   unsigned short year;
   unsigned short month;
   unsigned short day;
   unsigned short hour;
   unsigned short minute;
. . . .
```

Luca Galli, Stefan Ritt

PAUL SCHERRER INSTITUT

uation Board User's Manual

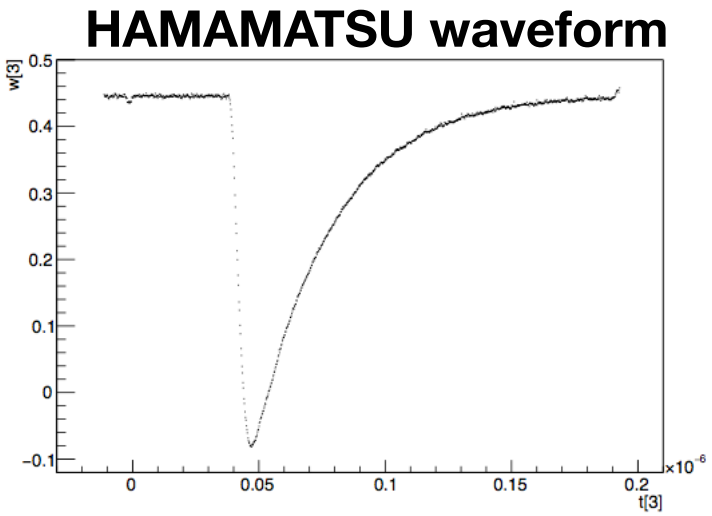
Word	Byte 0	Byte 1	Byte 2	Byte 3	Contents
0	'D'	'R'	'S'	'2'	File header, Byte 3 = version
1	'T'	'I'	'M'	'E'	Time Header
2	'B'	'#'	Board number		Board serial number
3	'C'	'0'	'0'	'1'	Channel 1 header
4	Time Bin Width #0				Effective time bin width in ns for channel 1 encoded in 4-Byte floating point format
5	Time Bin Width #1				
…	…				
1027	Time Bin Width #1023				
1028	'C'	'0'	'0'	'2'	Channel 2 header
1029	Time Bin Width #0				Effective time bin width in ns for channel 2 encoded in 4-Byte floating point format
1030	Time Bin Width #1				
…	…				
2052	Time Bin Width #1023				
2053	'E'	'H'	'D'	'R'	Event Header
2054	Event Serial Number				Serial number starting with 1
2055	Year		Month		Event date/time 16-bit values
2056	Day		Hour		
2057	Minute		Second		
2058	Millisecond		Range		Range center (RC) in mV
2059	'B'	'#'	Board number		Board serial number
2060	'T'	'#'	Trigger cell		Number of first readout cell
2061	'C'	'0'	'0'	'1'	Channel 1 header
2062	Scaler #1				Scaler for channel 1 in Hz
2063	Voltage Bin #0		Voltage Bin #1		Channel 1 waveform data encoded in 2-Byte integers. 0=RC-0.5V and 65535=RC+0.5V. RC see header.
2064	Voltage Bin #2		Voltage Bin #3		
…	…		…		
2574	Voltage Bin #1022		Voltage Bin #1023		
2575	'C'	'0'	'0'	'2'	Channel 2 header
2576	Scaler #2				Scaler for channel 2 in Hz
2577	Voltage Bin #0		Voltage Bin #1		Channel 2 waveform data encoded in 2-Byte integers. 0=RC-0.5V and 65535=RC+0.5V. RC see header.
2578	Voltage Bin #2		Voltage Bin #3		
…	…		…		
3088	Voltage Bin #1022		Voltage Bin #1023		
3089	'E'	'H'	'D'	'R'	Next Event Header
	…				

Next event

Waveform processing

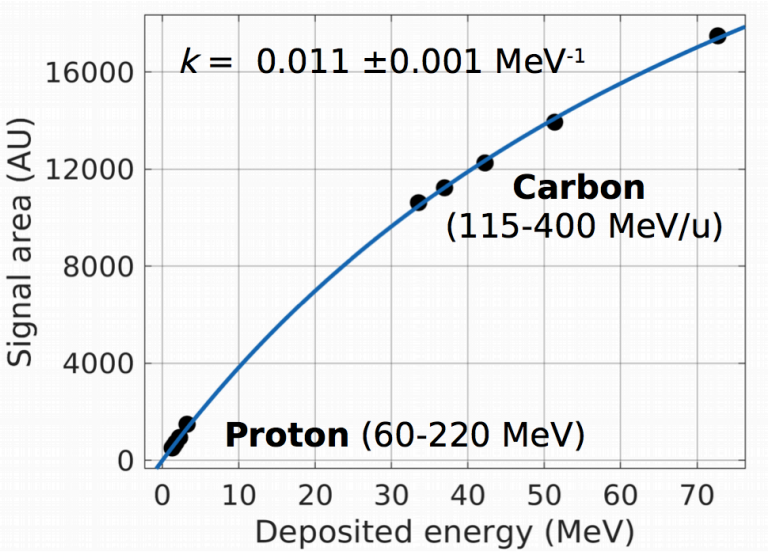
- A C++ macro (“readbinary.cc) will give as output a root file with
  - N (N_{max}=88+11) time arrays: vector[1024]
  - N (N_{max}=88+11) amplitudes (voltages): vector[1024]

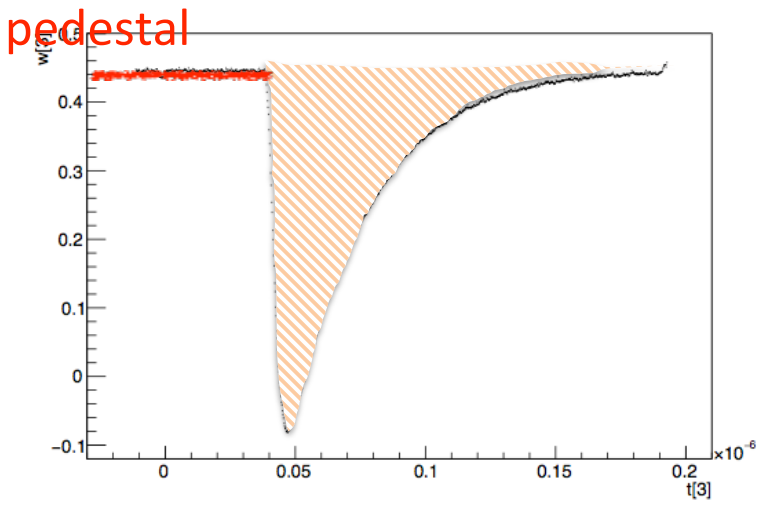
  → Max 44*2=88 waveforms (firmware based zero suppression for channels without hits)
- Charge and times from WFM processing (next)
- Position from
  - time difference $t=t_{left}-tr_{ight}$
  - Charge ratio
  - Bar crossing (orthogonal planes)

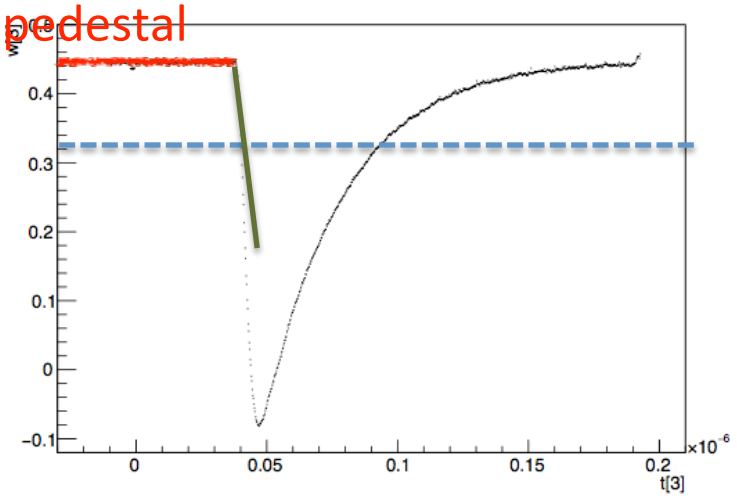


Waveform processing

- **Charge** obtained by
  - Determining pedestal
  - Integrate waveform (surface)
  - Calibrate to MC/NIST deposited value with Birks law → get **energy**



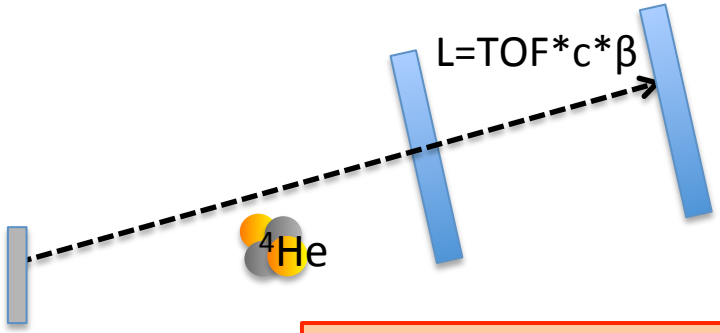




- **Time** obtained by performing linear fit around crossing point
- Position: see previous slide

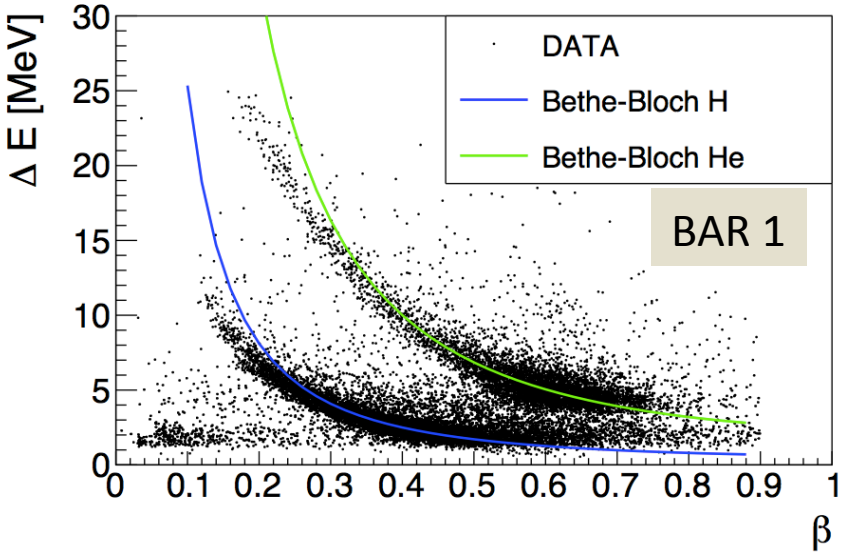
Current detector and reco status

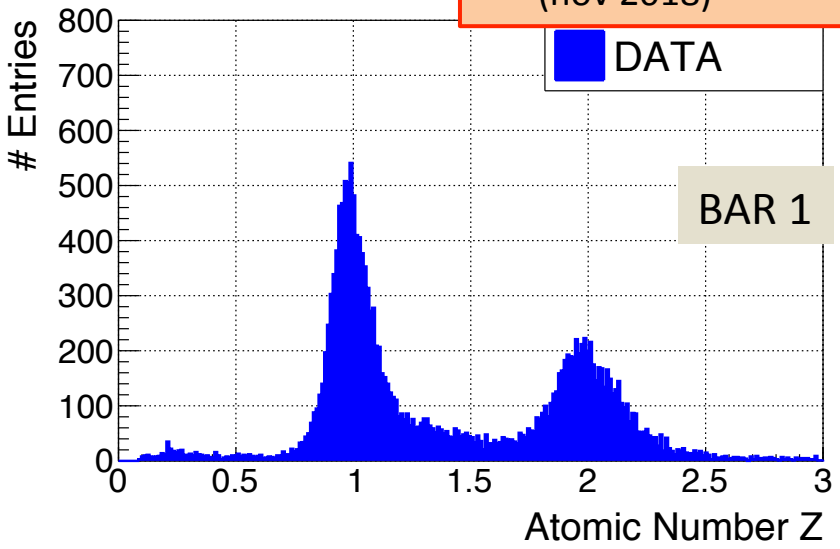
- ✓ **Stand alone DAQ software for the WDAQ readout developed and working!**
- ✓ June 2018: test-beam at CNAO with 2 bars (4 channels)
- ✓ Binary output decoded as before to obtain:
  - ✓ Energy
  - ✓ TOF
- ✓ Performances analyzed



Summary of performance and fragment identification results see

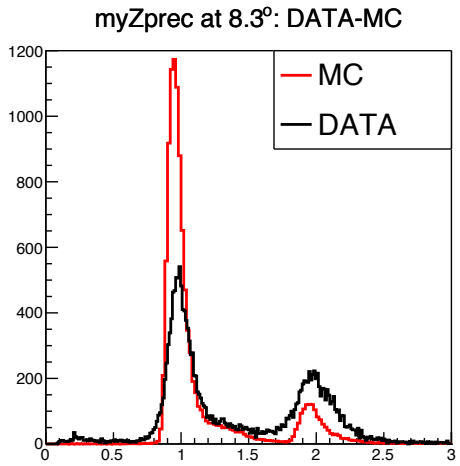
- Paper NIMA
- Presentation by E. Carriocchi at IEEE (nov 2018)

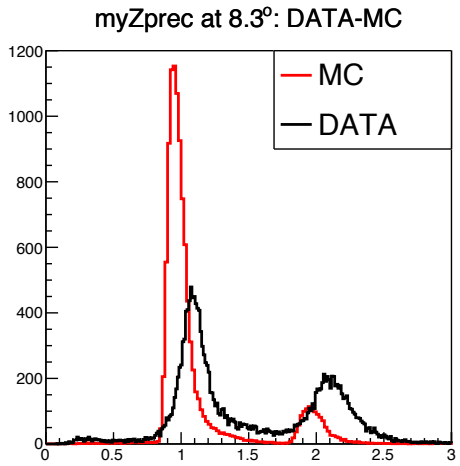


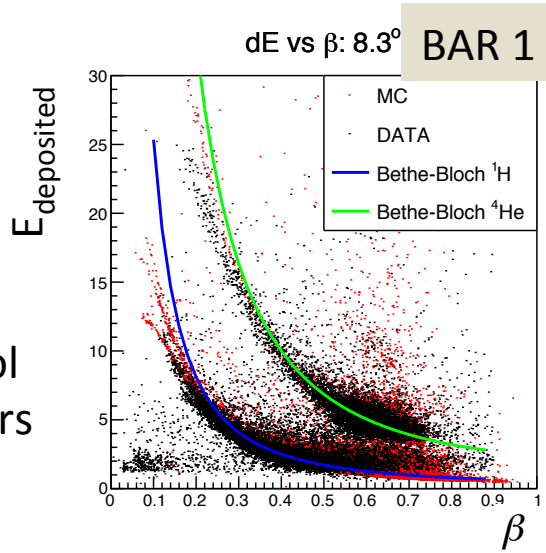


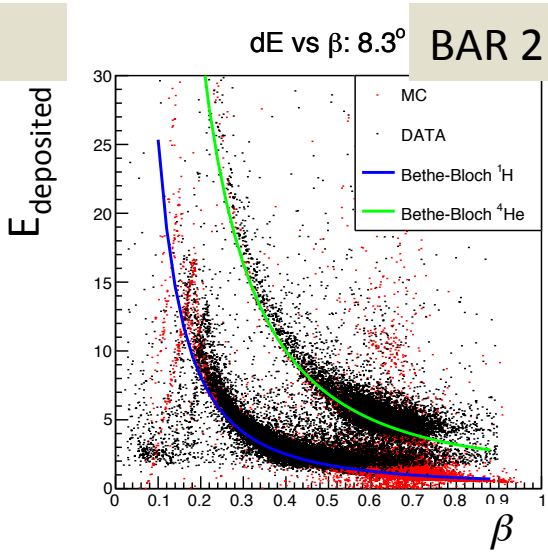
Simulation status

- ✓ Fragmentation in bars studied in simulations studied for various fragments as function of
  - ✓ Energy
  - ✓ Thickness of bar
  - ✓ Z of fragments
- ✓ Comparison performed with experimental data
  - Energy deposit
  - Beta
  - TOF
  - Z
  - …
- Work ongoing to understand
  - Isotopes
  - Energy deposits
  - Various aspects not fully understood and under control
  - How to account for loss in bars (relevant for low beta)









Status SHOE software implementation

- ✓ Few attemps to install SHOE on pc → abandoned
- ✓ Got account on Bologna TIER 3
- • Tried to compile but fails…

Conclusions

- Data structure defined and working (for prototype)
- Some help needed to run SHOE