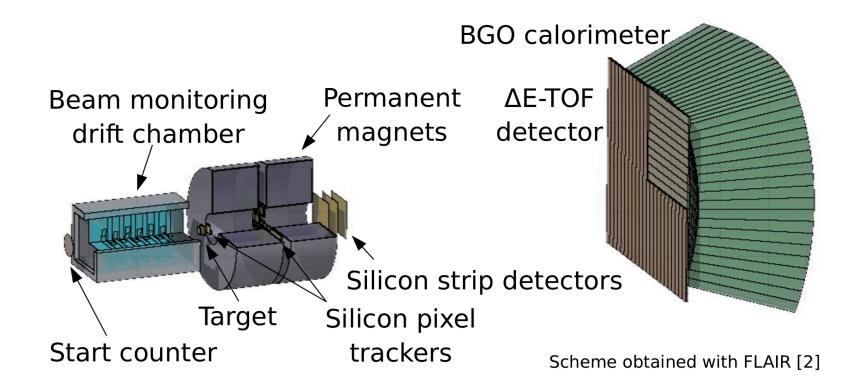
## **ΔE-TOF detector and software update**

**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

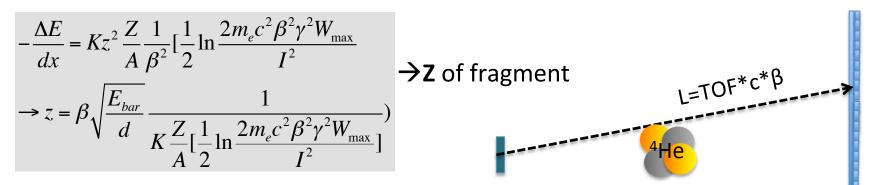
## Introduction



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

## Introduction

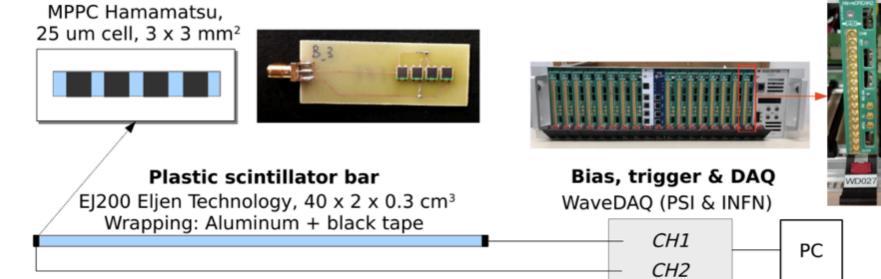
- ΔE-TOF detector: based on scintillator bars
- ΔE-TOF detector provides
  - Time-Of-Flight of fragments
    - TOF= $t_{det}$ - $t_{trig}$ = $(t_1+t_r)/2 t_{trig} \rightarrow velocity \beta QUESTION$
  - Energy deposited (from charge-energy calibration)



- Position of deposit (2D by orthogonal arrangement)
  - By determining T<sub>Ir</sub>=t<sub>I</sub>-t<sub>r</sub>
  - By ratio of collected energies  $ln(E_l/E_r)$

## Data acquisition

#### Four SiPMs at each end



- 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

#### $\rightarrow$ Output is binary file

Connected to trigger board

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

<pre>#include <string. #include <stdio.h: #include "TFile.h" #include "TTree.h" #include "TString #include "TGraph. #include "TCanvas #include "Getline</stdio.h: </string. </pre>	> ' . h'' 1'' . h''
<pre>typedef struct {     char     char } FHEADER;</pre>	<pre>tag[3]; version;</pre>
<pre>typedef struct {     char } THEADER;</pre>	<pre>time_header[4];</pre>
<pre>typedef struct {     char     unsigned short } BHEADER;</pre>	<pre>bn[2]; board_serial_number;</pre>
<pre>typedef struct {     char     unsigned int     unsigned short     unsigned short     unsigned short     unsigned short     unsigned short </pre>	month; day; hour;

	Luca (	Galli, S	Stefar	n Ritt	PAUL SCHERRER INSTITU uation Board User's Manua
Word	Byte 0	Byte 1	Byte 2	Byte 3	Contents
0	`D'	`R'	`s'	`2'	File header, Byte 3 = version
1	`T'	`ī'	`M'	`E'	Time Header
2	'B'	`#'	Board	number	Board serial number
3	`C'	`0 <i>'</i>	<b>`</b> 0′	`1'	Channel 1 header
4		Time Bin	Width #0		
5		Time Bin	Width #1		Effective time bin width in ns for channel 1 encoded in 4-Byte
					floating point format
1027		Time Bin W	idth #1023		-
1028	`C'	<b>`</b> 0′	`0 <i>'</i>	`2'	Channel 2 header
1029		Time Bin	Width #0		
1030		Time Bin	Width #1		Effective time bin width in ns for channel 2 encoded in 4-Byte
					floating point format
2052		Time Bin W	idth #1023		-
2053	'E'	`H'	'D'	'R'	Event Header
2054		Event Ser	ial Number	Serial number starting with 1	
2055	Year Month				Event date/time 16-bit values
2056	Da	ау	Но	ur	
2057	Min	ute	Sec	ond	1
2058	Millis	second	Rai	nge	Range center (RC) in mV
2059	`B'	`# <i>'</i>	Board	number	Board serial number
2060	`T'	`# <i>'</i>	Trigge	r cell	Number of first readout cell
2061	`C'	<b>`</b> 0′	`0 <i>'</i>	`1'	Channel 1 header
2062		Scale	er #1		Scaler for channel 1 in Hz
2063	Voltage Bin #0		Voltage	Bin #1	
2064	Voltage Bin #2		Voltage	Bin #3	Channel 1 waveform data encoded in 2-Byte integers. 0=RC-0.5V and
					65535=RC+0.5V. RC see header.
2574	Voltage H	3in #1022	Voltage I	3in #1023	
2575	`C'	`0 <i>'</i>	<b>`</b> 0′	<b>`</b> 2′	Channel 2 header
2576		Scal	er #2		Scaler for channel 2 in Hz
2577	Voltage	Bin #0	Voltage	Bin #1	
2578	Voltage	Bin #2	Voltage	Bin #3	Channel 2 waveform data encoded
					in 2-Byte integers. 0=RC-0.5V and 65535=RC+0.5V. RC see header.
3088	Voltage H	3in #1022	Voltage I	3in #1023	
		`H'	'D'	'R'	Next Event Header

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

<pre>#include <string. "getline<="" "tcanvas="" "tfile.h="" "tgraph.="" "tstring="" "ttree.h="" #include="" <stdio.h="" pre=""></string.></pre>	> " .h" h" .h"
<pre>typedef struct {     char     char } FHEADER;</pre>	tag[3]; version;
<pre>typedef struct {     char } THEADER;</pre>	<pre>time_header[4];</pre>
<pre>typedef struct {     char     unsigned short } BHEADER;</pre>	<pre>bn[2]; board_serial_number;</pre>
<pre>typedef struct {     char     unsigned int     unsigned short     unsigned short     unsigned short     unsigned short     unsigned short</pre>	month; day; hour;

Contents	Byte 3	Byte 2	Byte 1	Byte 0	Word
File header, Byte 3 = version	head	File	`R'	`D'	0
Time Header	`E'	'M'	`I'	`T'	1
Board serial number	number	Board	`# <i>'</i>	`B'	2
Channel 1 header	`1'	·0'	`0 <i>'</i>	`C'	3
		Width #0	Time Bin		4
Effective time bin width in ns for channel 1 encoded in 4-Byte		Width #1	Time Bin		5
floating point format					
		idth #1023	Time Bin W		1027
Channel 2 header	`2'	<b>`</b> 0′	10'	`C'	1028
		Width #0	Time Bin		1029
Effective time bin width in ns for channel 2 encoded in 4-Byte		Width #1	Time Bin		1030
floating point format					
		idth #1023	Time Bin W		2052
Event Header	'R'	'D'	`H'	`E'	2053
Serial number starting with 1	ial Number		Event Seri		2054
Event date/time 16-bit values	nth	Year Month		Ye	2055
	ur	Но	ау	Da	2056
	ond	Sec	ute	Min	2057
Range center (RC) in mV	nge	Rar	Millisecond		2058
Board serial number	number	Board	`# <i>'</i>	`B'	2059
Number of first readout cell	r cell	Trigge	`# <i>'</i>	`T'	2060
Channel 1 header	`1'	<b>`</b> 0′	`0 <i>'</i>	`C'	2061
Scaler for channel 1 in Hz		er #1	Scale		2062
	Bin #1	Voltage	Voltage Bin #0		2063
Channel 1 waveform data encode in 2-Byte integers. 0=RC-0.5V ar	Voltage Bin #3		Voltage Bin #2		2064
65535=RC+0.5V. RC see header.					
	3in #1023	Voltage B	Bin #1022	Voltage H	2574
Channel 2 header	`2'	<b>`</b> 0′	<b>`</b> 0′	`C'	2575
Scaler for channel 2 in Hz		er #2	Scale		2576
	Bin #1	Voltage	Bin #0	Voltage	2577
Channel 2 waveform data encode in 2-Byte integers. 0=RC-0.5V ar	Bin #3	Voltage	Bin #2	Voltage	2578
65535=RC+0.5V. RC see header.					
	3in #1023	Voltage B	3in #1022	Voltage H	3088
Next Event Header	'R'	<b>י</b> ם'	`H'	YE'	3089

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

<pre>#include <string. #include <stdio.h: #include "TFile.h" #include "TTree.h" #include "TString #include "TGraph. #include "TCanvas #include "Getline</stdio.h: </string. </pre>	> ' h" h"
<pre>typedef struct {     char     char } FHEADER;</pre>	<pre>tag[3]; version;</pre>
<pre>typedef struct {     char } THEADER;</pre>	<pre>time_header[4];</pre>
<pre>typedef struct {     char     unsigned short } BHEADER;</pre>	<pre>bn[2]; board_serial_number;</pre>
<pre>typedef struct {     char     unsigned int     unsigned short     unsigned short     unsigned short     unsigned short     unsigned short </pre>	month; day; hour;

					uation Board User's Manua
Word	Byte 0	Byte 1	Byte 2	Byte 3	Contents
0	'D'	`R'	`s'	<b>`</b> 2′	File header, Byte 3 = version
1	`T'	`I'	Tim	e hea	Ce feader
2	`B'	`# <i>'</i>	Board	number	Board serial number
3	`C'	`0 <i>'</i>	<b>`</b> 0′	`1'	Channel 1 header
4		Time Bin	Width #0		
5		Time Bin	Width #1		Effective time bin width in ns for channel 1 encoded in 4-Byte
					floating point format
1027		Time Bin W	idth #1023		
1028	`C'	`0'	<b>`</b> 0′	`2'	Channel 2 header
1029		Time Bin	Width #0		
1030		Time Bin	Width #1		Effective time bin width in ns for
			<ul> <li>channel 2 encoded in 4-Byte floating point format</li> </ul>		
2052		Time Bin W	idth #1023		
2053	`E'	`H'	'D'	`R'	Event Header
2054		Event Ser:	ial Number		Serial number starting with 1
2055	Ye	ar	Mor	th	Event date/time 16-bit values
2056	Da	ау	Но	ur	
2057	Min	ute	Sec	ond	
2058	Milli	second	Rar	ige	Range center (RC) in mV
2059	`B'	`# <i>'</i>	Board	number	Board serial number
2060	`T'	`# <i>'</i>	Trigge	r cell	Number of first readout cell
2061	`C'	`0 <i>'</i>	<b>`</b> 0 <i>'</i>	`1'	Channel 1 header
2062		Scale	er #1		Scaler for channel 1 in Hz
2063	Voltage	Voltage Bin #0		Bin #1	
2064	Voltage Bin #2		Voltage Bin #3		Channel 1 waveform data encoded
					in 2-Byte integers. 0=RC-0.5V and 65535=RC+0.5V. RC see header.
2574	Voltage I	3in #1022	Voltage E	in #1023	
2575	`C'	<b>`</b> 0′	<b>`</b> 0′	`2'	Channel 2 header
2576		Scale	er #2		Scaler for channel 2 in Hz
2577	Voltage	Bin #0	Voltage	Bin #1	
2578	Voltage	Bin #2	Voltage	Bin #3	Channel 2 waveform data encoded
					in 2-Byte integers. 0=RC-0.5V and 65535=RC+0.5V. RC see header.
3088	Voltage I	Bin #1022	Voltage E	in #1023	1
3089	NE/	`H'	<b>`</b> D <b>′</b>	'R'	Next Event Header

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

<pre>#include <string. "getline<="" "tcanvas="" "tfile.h="" "tgraph.="" "tstring="" "ttree.h="" #include="" <stdio.h="" pre=""></string.></pre>	> " .h" h" .h"
<pre>typedef struct {     char     char } FHEADER;</pre>	<pre>tag[3]; version;</pre>
<pre>typedef struct {     char } THEADER;</pre>	<pre>time_header[4];</pre>
<pre>typedef struct {     char     unsigned short } BHEADER;</pre>	<pre>bn[2]; board_serial_number;</pre>
<pre>typedef struct {     char     unsigned int     unsigned short     unsigned short     unsigned short     unsigned short     unsigned short </pre>	month; day; hour;

	Luca (	Galli, S	Stefar	n Ritt	PAUL SCHERRER INSTITUT uation Board User's Manual
Word	Byte 0	Byte 1	Byte 2	Byte 3	Contents
0	`D'	`R'	`s'	121	File header, Byte 3 = version
1	`T'	`ī'	`M'	`E'	Time Header
2	`B'	`# <i>'</i>	Boa	n <b>re</b> r he	ac er <sup>number</sup>
- J	`C'	<b>`</b> 0 <b>′</b>	·0/	`1'	Channel 1 header
4		Time Bin	Width #0		
5		Time Bin	Width #1		Effective time bin width in ns for channel 1 encoded in 4-Byte
					floating point format
1027		Time Bin W	idth #1023		
1028	`C'	<b>`</b> 0′	`0 <i>'</i>	`2'	Channel 2 header
1029		Time Bin	Width #0		
1030		Time Bin	Width #1		Effective time bin width in ns for
					<ul> <li>channel 2 encoded in 4-Byte floating point format</li> </ul>
2052		Time Bin W			
2053	`E'	`H'	'D'	`R'	Event Header
2054		Event Ser:	ial Number		Serial number starting with 1
2055	Ye	ar	Month		Event date/time 16-bit values
2056	Da	ay	Но	ur	
2057	Min	ute	Sec	ond	
2058	Millis	second	Rar	nge	Range center (RC) in mV
2059	`B'	`# <i>'</i>	Board	number	Board serial number
2060	`T'	`# <i>'</i>	Trigge	r cell	Number of first readout cell
2061	`C'	<b>`</b> 0′	`0 <i>'</i>	`1'	Channel 1 header
2062		Scale			Scaler for channel 1 in Hz
2063	Voltage Bin #0		Voltage	Bin #1	
2064	Voltage	Bin #2	Voltage	Bin #3	Channel 1 waveform data encoded in 2-Byte integers. 0=RC-0.5V and
					65535=RC+0.5V. RC see header.
2574	Voltage H	3in #1022	Voltage H	3in #1023	
2575	`C'	<b>`</b> 0′	<b>`</b> 0′	`2'	Channel 2 header
2576		Scale	er #2		Scaler for channel 2 in Hz
2577	Voltage	Bin #0	Voltage	Bin #1	
2578	Voltage	Bin #2	Voltage	Bin #3	Channel 2 waveform data encoded
					in 2-Byte integers. 0=RC-0.5V and 65535=RC+0.5V. RC see header.
2000	Voltage F	3in #1022	Voltage H	3in #1023	]
3088	vortuge i				

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

<pre>#include <string. "getline<="" "tcanvas="" "tfile.h="" "tgraph.="" "tstring="" "ttree.h'="" #include="" <stdio.h:="" pre=""></string.></pre>	> " .h" h" .h"
<pre>typedef struct {     char     char } FHEADER;</pre>	<pre>tag[3]; version;</pre>
<pre>typedef struct {     char } THEADER;</pre>	<pre>time_header[4];</pre>
<pre>typedef struct {     char     unsigned short } BHEADER;</pre>	<pre>bn[2]; board_serial_number;</pre>
<pre>typedef struct {     char     unsigned int     unsigned short     unsigned short     unsigned short     unsigned short     unsigned short </pre>	month; day; hour;

	Luca	cum, .			uation Board User's Manua	
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'	`# <i>'</i>	Board number		Board serial number	
3	`C'	<b>`</b> 0′	<b>`</b> 0′	<b>`</b> 1'	Channel 1 header	
4		Time Bin	Width #0			
5		Time Bin	Width #1		Effective time bin width in ns for channel 1 encoded in 4-Byte	
					floating point format	
1027		Time Bin W	idth #1023			
1028	`C'	`0 <i>'</i>	`0 <i>'</i>	<b>`</b> 2′	Channel 2 header	
1029		Time Bin	Width #0			
1030		Time Bin	Width #1		Effective time bin width in ns for channel 2 encoded in 4-Byte	
					floating point format	
2052		Time Din W	114th #1022		-	
2053	`E'	`H' <b>F</b> \	ent h	eade	Event Header	
2054		Litene ber		cuuc	Serial number starting with 1	
2055	Year Month				Event date/time 16-bit values	
2056	Da	ау	Но	ur	-	
2057	Min	ute	Sec	ond	-	
2058	Milli	second	Rar	nge	Range center (RC) in mV	
2059	`B'	`# <i>'</i>	Board	number	Board serial number	
	`T'	`# <i>'</i>	Trigge	r cell	Number of first readout cell	
2060	`C'	<b>`</b> 0′	`0 <i>'</i>	`1'	Channel 1 header	
2060 2061			a.u. #1			
		Scale	er #1		Scaler for channel 1 in Hz	
2061	Voltage	Scale Bin #0	voltage	Bin #1	Scaler for channel 1 in Hz	
2061 2062			1		Channel 1 waveform data encoded	
2061 2062 2063	Voltage	Bin #0	Voltage	Bin #3	Channel 1 waveform data encoded	
2061 2062 2063 2064	Voltage	Bin #0 Bin #2	Voltage Voltage	Bin #3	Channel 1 waveform data encoded in 2-Byte integers. 0=RC-0.5V an	
2061 2062 2063 2064 	Voltage	Bin #0 Bin #2	Voltage Voltage	Bin #3	Channel 1 waveform data encoded in 2-Byte integers. 0=RC-0.5V and	
2061 2062 2063 2064  2574	Voltage 1 Voltage 1	Bin #0 Bin #2  Bin #1022	Voltage Voltage Voltage H	Bin #3 3in #1023	Channel 1 waveform data encode in 2-Byte integers. 0=RC-0.5V an 65535=RC+0.5V. RC see header.	
2061 2062 2063 2064  2574 2575	Voltage 1 Voltage 1	Bin #0 Bin #2  Bin #1022	Voltage Voltage Voltage H '0'	Bin #3 	Channel 1 waveform data encodee in 2-Byte integers. 0=RC-0.5V an 65535=RC+0.5V. RC see header. Channel 2 header	
2061 2062 2063 2064  2574 2575 2576	Voltage Voltage Voltage	E Bin #0 E Bin #2  Bin #1022 `O' Scale	Voltage Voltage Voltage I '0' er #2	Bin #3 	Channel 1 waveform data encoded in 2-Byte integers. 0=RC-0.5V an 65535=RC+0.5V. RC see header. Channel 2 header Scaler for channel 2 in Hz Channel 2 waveform data encoded	
2061 2062 2063 2064  2574 2575 2576 2577	Voltage Voltage Voltage	Bin #0 Bin #2  Bin #1022 `0' Scale Bin #0	Voltage Voltage Voltage H vor er #2 Voltage	Bin #3 	Channel 1 waveform data encoded in 2-Byte integers. 0=RC-0.5V an 65535=RC+0.5V. RC see header. Channel 2 header Scaler for channel 2 in Hz Channel 2 waveform data encoded in 2-Byte integers. 0=RC-0.5V an	
2061 2062 2063 2064  2574 2575 2576 2577 2578	Voltage Voltage 1 'C' Voltage Voltage	Bin #0 Bin #2  Bin #1022 `0' Scale Bin #0	Voltage Voltage Voltage H vor er #2 Voltage	Bin #3 Bin #1023 '2' Bin #1 Bin #3	Channel 1 waveform data encode in 2-Byte integers. 0=RC-0.5V an 65535=RC+0.5V. RC see header. Channel 2 header Scaler for channel 2 in Hz Channel 2 waveform data encodee	

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

<pre>#include <string. "getline<="" "tcanvas="" "tfile.h"="" "tgraph.="" "tstring="" #include="" <stdio.h="" pre=""></string.></pre>	> " .h" h" .h"
<pre>typedef struct {     char     char } FHEADER;</pre>	tag[3]; version;
<pre>typedef struct {     char } THEADER;</pre>	<pre>time_header[4];</pre>
<pre>typedef struct {     char     unsigned short } BHEADER;</pre>	<pre>bn[2]; board_serial_number;</pre>
<pre>typedef struct {     char     unsigned int     unsigned short     unsigned short     unsigned short     unsigned short     unsigned short </pre>	month; day; hour;

		/	Stefar		uation Board User's Manua	
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'	`# <i>'</i>	Board	number	Board serial number	
3	`C'	`0 <i>'</i>	`0 <i>'</i>	<b>`</b> 1'	Channel 1 header	
4		Time Bin	Width #0			
5		Time Bin	Width #1		Effective time bin width in ns for channel 1 encoded in 4-Byte	
					floating point format	
1027		Time Bin W	idth #1023		-	
1028	`C'	`0 <i>'</i>	`0 <i>'</i>	<b>`</b> 2'	Channel 2 header	
1029		Time Bin	Width #0			
1030		Time Bin	Width #1		Effective time bin width in ns for	
					<ul> <li>channel 2 encoded in 4-Byte floating point format</li> </ul>	
2052					-	
2052			idth #1023		Event Header	
2055	`E'	`H'	'D'	`R'		
2055		Event Ser.	Serial number starting with 1 Event date/time 16-bit values			
2055		ar	Mor		Event date/ume 16-bit values	
		ау		ur	_	
2057		ute	-	ond	_	
2058	Millis	secon EVE	ent in	f0	Range center (RC) in mV	
2059	`B'	`#′		number	Board serial number	
2060	`T'	(b)	ardh	eadei	Nun Cart Gdout cell	
2061	`C'	`0 <i>'</i>	`0'	`1 <b>'</b>	Channel 1 header	
2062		dat	a,ŗıcha	innel	Scille 🔂 channel 1 in Hz	
2063	Voltage	Bin #0	Voltage	Bin #1		
2064	Voltage	Bin #2	Voltage	Bin #3	Channel 1 waveform data encode	
					in 2-Byte integers. 0=RC-0.5V ar 65535=RC+0.5V. RC see header.	
2574	Voltage H	Bin #1022	Voltage H	3in #1023		
2575	`C'	<b>`</b> 0′	`0 <i>'</i>	<b>`</b> 2'	Channel 2 header	
2576		Scale	er #2		Scaler for channel 2 in Hz	
2577	Voltage	Bin #0	Voltage	Bin #1		
2578	Voltage	Bin #2	Voltage	Bin #3	Channel 2 waveform data encode	
					in 2-Byte integers. 0=RC-0.5V an 65535=RC+0.5V. RC see header.	
		Din #1022	Voltago	4n #1022		
3088	Voltago					

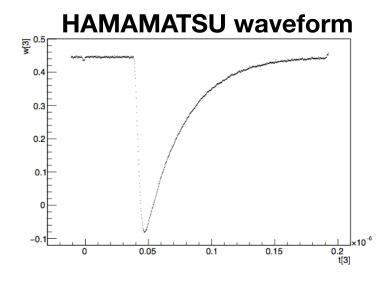
- C++ macro to decode the signal to waveforms in root format
  - to be copied and adjusted in SHOE

<pre>#include <string. "getline<="" "tcanvas="" "tfile.h="" "tgraph.="" "tstring="" "ttree.h="" #include="" <stdio.h="" pre=""></string.></pre>	> " .h" h" .h"
<pre>typedef struct {     char     char } FHEADER;</pre>	tag[3]; version;
<pre>typedef struct {     char } THEADER;</pre>	<pre>time_header[4];</pre>
<pre>typedef struct {     char     unsigned short } BHEADER;</pre>	<pre>bn[2]; board_serial_number;</pre>
<pre>typedef struct {     char     unsigned int     unsigned short     unsigned short     unsigned short     unsigned short     unsigned short </pre>	month; day;

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'	<b>`</b> 0′	`0 <i>'</i>	<b>`</b> 1'	Channel 1 header
4	Time Bin Width #0				
5		Time Bin	Width #1		Effective time bin width in ns for channel 1 encoded in 4-Byte floating point format
1027		Time Bin W	idth #1023		-
1028	`C'	`0 <i>'</i>	`0 <i>'</i>	`2'	Channel 2 header
1029		Time Bin	Bin Width #0		
1030		Time Bin	Width #1  Nidth #1023		Effective time bin width in ns for channel 2 encoded in 4-Byte floating point format
2052		Time Bin W			
2053	`E'	`H'	'D'	'R'	Event Header
2054		Event Ser	ial Number		Serial number starting with 1
2055	Year Month		Event date/time 16-bit values		
2056	Day		Hour		1
2057	Min	ute	Second		-
2058	Milli	second	Range		Range center (RC) in mV
2059	`B'	`# <i>'</i>	Board	number	Board serial number
2060	`T'	`# <i>'</i>	Trigger cell		Number of first readout cell
2061	`C'	`0 <i>'</i>	`0 <i>'</i>	<b>`</b> 1'	Channel 1 header
2062		Scale	er #1		Scaler for channel 1 in Hz
2063	Voltage	Bin #0	Voltage Bin #1		
2064	Voltage	Bin #2	Voltage Bin #3		Channel 1 waveform data encoder in 2-Byte integers. 0=RC-0.5V an 65535=RC+0.5V. RC see header.
2574	Voltage H	Bin #1022	Voltage Bin #1023		
2575	`C'	<b>`</b> 0′	<b>`</b> 0′	<b>`</b> 2'	Channel 2 header
2576		Scale	er #2		Scaler for channel 2 in Hz
2577	Voltage	Bin #0	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 I	3in #1022	Voltage Bin #1023		
3089	`E'	OVT O	vont	`R'	Next Event Header

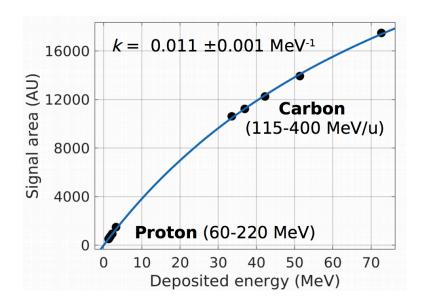
## Waveform processing

- A C++ macro ("readbinary.cc) will give as output a root file with
  - N (N<sub>max</sub>=88+11) time arrays: vector[1024]
  - N (N<sub>max</sub>=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<sub>left</sub>-tr<sub>ight</sub>
  - Charge ratio
  - Bar crossing (orthogonal planes)

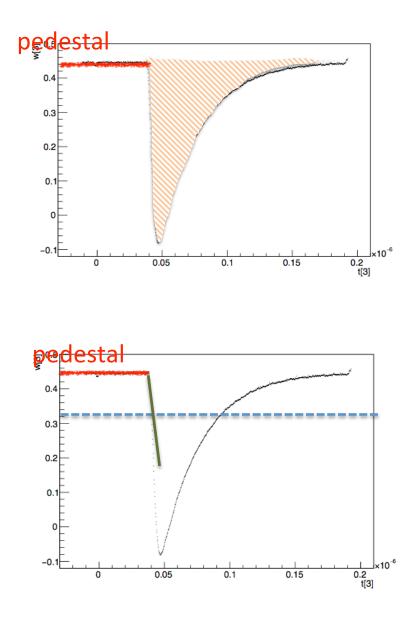


## Waveform processing

- 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

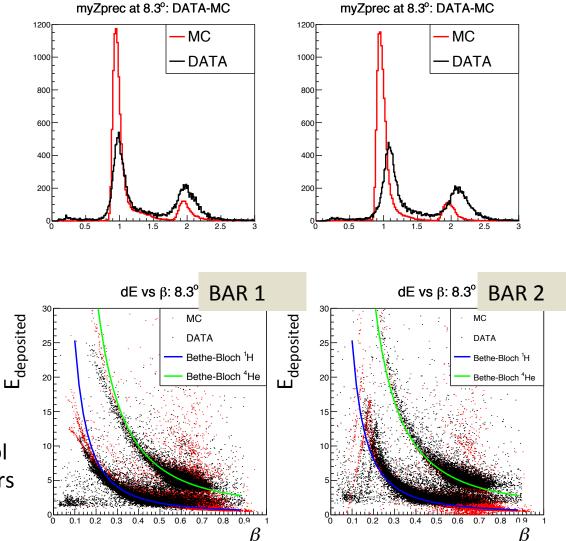


## Current detector and reco status

✓ Stand alone DAQ software for the WDAQ  $L=TOF*c*\beta$ readout developed and working! June 2018: test-beam at CNAO with 2 bars (4 channels) Binary output decoded as before to obtain: Summary of ✓ Energy performance and fragment identification ✓ TOF results see ✓ Performances analyzed **Paper NIMA** Presentation by E. Carriocchi at IEEE (nov 2018) 30 800 A E [MeV] # Entries DATA DATA 700 25 Bethe-Bloch H 600 **Bethe-Bloch He** 20 500 BAR 1 BAR 1 15 400 300 10 200 100 00 0.8 0.5 1.5 2 2.5 0.9 1 .6 0 Atomic Number Z

## 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