GBP biweekly meeting – CLEAR data analysis Current status

25/10/2022

Università degli Studi di Padova

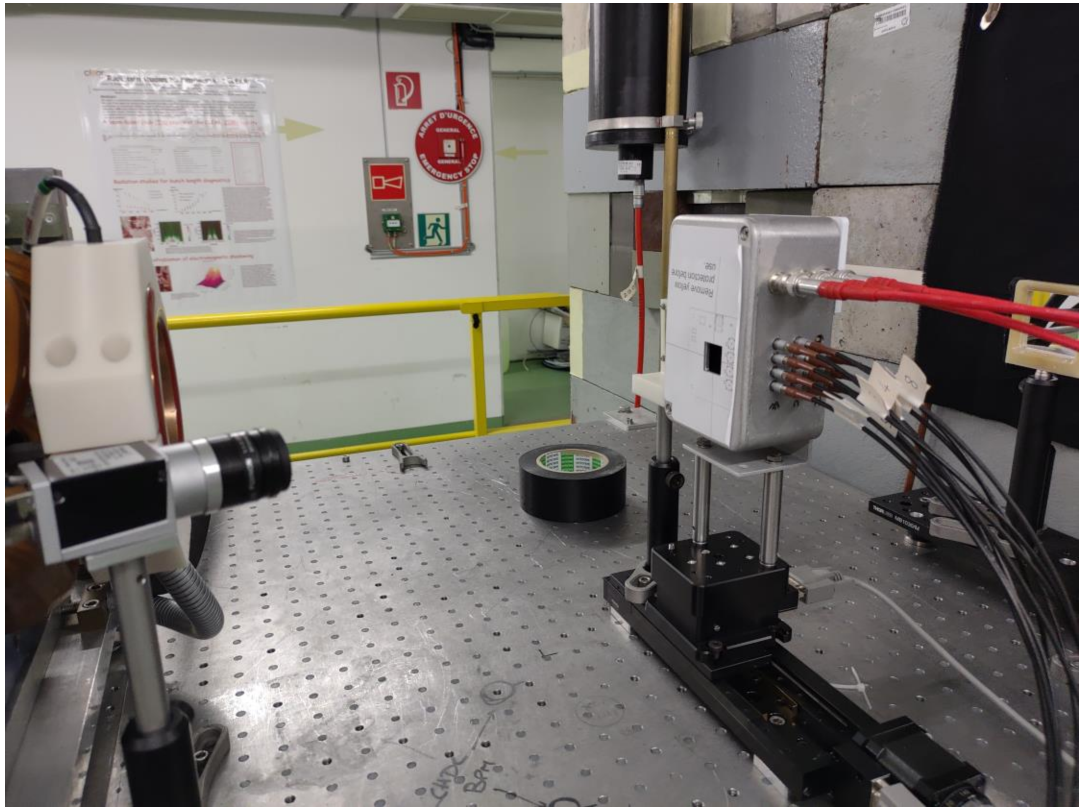
INFN

Table of contents

Recap of CLEAR test

Detectors + equipment

- A stack of three 4-strip sapphire sensors of three different companies
- Digitizers 8-bit 1GS/s 1MHz connected directly with 30m cables
- Power supply CAEN N1471H



Nb. channels	4	4	3 (1 not working)
Thickness	110um	150um	150um
Manufacturer	Wuppertal	University	M-type
Beam intercept	first	second	third
Digitizer channel associations (s	5 (1) 11 (2) 12 (3) 6 (4)	7 (1) 3 (2) 4 (3) 8 (4)	9 (1) 1 (2) 2 (3) 10 (4) - not working

Recap of CLEAR test

Measure types

- Bare signals as a function of the beam charge
- Scan in the horizontal position (along strip) for strip response uniformity
- Scan in the vertical position (to find beam center)
- Irradiation of the sample up to 15MGy
- Scan in HV to measure CCE in both irradiated/non-irr. area
- Scan in beam charge to investigate detector response in both irradiated/non-irr. areas

Data recorded and formats

Data recorded

1. Digitized waveforms at 10Hz
2. Beam charge 10Hz during irradiation + 1Hz otherwise
3. PSU current/voltage monitors (current rate 1Hz, voltage rate 0.2Hz)

Data format + size, respectively

1. Matlab files (binary .mat compressed) - 1.1GB
2. Tabular ASCII files (text .dat uncompressed) - 22.2MB
3. Non-tabular ASCII files (text .txt uncompressed) - 40.2MB

Data structure. Digitizers

Typical matlab file structure

- File name has generic info about measure type
- Within the file, ONE date is recorded with the time the file was closed and saved in the host PC

- 12 arrays in with double precision values, containing each 600 triggers each with 100 samples for each trigger

- Vertical scale + offset already in data
- **NO** information about the horizontal time scale -> <u>need to check CLEAR log at that time of the day!</u>

CCE_Luxe_50pC_213mm08-Sep-2022 19:55:25.mat
CCE_Luxe_50pC_213mm08-Sep-2022 19:56:26.mat
CCE_Luxe_50pC_213mm08-Sep-2022 19:57:28.mat
CCE_Luxe_50pC_213mm08-Sep-2022 19:58:29.mat
CCE_Luxe_50pC_213mm08-Sep-2022 19:59:30.mat
CCE_Luxe_50pC_213mm08-Sep-2022 20:00:31.mat
CCE_Luxe_50pC_213mm08-Sep-2022 20:01:32.mat
CCE_Luxe_50pC_213mm08-Sep-2022 20:02:34.mat

Workspace

Name	Value
date_value	'08-Sep-2022 19:59:30'
Signal_01_all	600x100 double
Signal_02_all	600x100 double
Signal_03_all	600x100 double
Signal_04_all	600x100 double
Signal_05_all	600x100 double
Signal_06_all	600x100 double
Signal_07_all	600x100 double
Signal_08_all	600x100 double
Signal_09_all	600x100 double
Signal_10_all	600x100 double
Signal_11_all	600x100 double
Signal_12_all	600x100 double

Data structure. Bergoz

Typical bergoz file structure

- File name contains information about the horizontal position of the table (211mm) and train charge (2700pC)
- A file contains (# shots) lines of 10Hz chg. data. Some file with 20k shots, but this number isn't always the same

```
Irradition_160_2700pC_211mm_2.dat
Irradition_160_2700pC_211mm_3.dat
Irradition_160_2700pC_211mm_4.dat
Irradition_160_2700pC_211mm_5.dat
Irradition_160_2700pC_211mm_6.dat
Irradition_160_2700pC_211mm_7.dat
Irradition_160_2700pC_211mm.dat
```

Row structure

- Shot index
- Unixtime of the shot (PC host time)
- Instantaneous beam charge
- Accumulated charge from the beginning of the file

```
# shotIdx unixtime charge[pC] accumulatedCharge[pC]
1 1662562833.6899738 1334.21 1334.21
2 1662562833.7899253 1340.58 2674.79
3 1662562833.8898566 1329.16 4003.95
4 1662562833.9897985 1360.37 5364.32
5 1662562834.0897367 1375.52 6739.84
6 1662562834.1896489 1331.43 8071.27
7 1662562834.2895718 1322.02 9393.30
8 1662562834.3895104 1259.34 10652.64
9 1662562834.4894056 1234.31 11886.95
10 1662562834.5893195 1237.09 13124.05
11 1662562834.6892455 1239.73 14363.78
12 1662562834.7891932 1266.11 15629.89
13 1662562834.8891146 1212.94 16842.83
14 1662562834.989024 1205.88 18048.71
15 1662562835.0889482 1237.86 19286.57
16 1662562835.1888812 1238.89 20525.46
17 1662562835.288829 1269.48 21794.94
18 1662562835.3887708 1346.61 23141.55
```

Data structure. PSU

Typical psu file structure – day 6

```
06_09_2022 14:40:09 + 0.5s/reading
Current: 0.0 0.0 0.0
Voltage: 0.0 0.0 0.0
06_09_2022 14:40:21 + 0.5s/reading
Current: 0.0 0.0 0.0
06_09_2022 14:40:28 + 0.5s/reading
Current: 0.0 0.0 0.0
06_09_2022 14:40:34 + 0.5s/reading
Current: 0.0 0.0 0.0
06_09_2022 14:40:40 + 0.5s/reading
Current: 0.0 0.0 0.0
06_09_2022 14:40:47 + 0.5s/reading
06_09_2022 14:40:53 + 0.5s/reading
06_09_2022 14:40:53 + 0.5s/reading
Voltage: 0.0 0.0 0.0
06_09_2022 15:08:27 + 0.5s/reading
Current: 0.0 0.0 0.0
Voltage: 0.0 0.0 0.0
06_09_2022 15:08:39 + 0.5s/reading
```

Non uniform time difference between readings

Three channel readings for current and voltage

Every 5 current a voltage reading but sometimes *missing hits* are present

Host PC time of the subprocess call to communicate with PSU

Data structure. PSU

Typical psu file structure – days after 7

```
08_09_2022 08:58:58 + 0.001s/reading
Current: 0.00055 0.00055 0.0
Voltage: 199.2 199.6 199.8
08_09_2022 08:58:58 + 0.001s/reading
Current: 0.00055 0.00055 0.0
08_09_2022 08:58:58 + 0.001s/reading
Current: 0.0008 0.0007 0.0
08_09_2022 08:58:58 + 0.001s/reading
Current: 0.0008 0.0007 0.0
```

Since 7th, the acquisition rate was improved to \sim 4Hz
-> more robust acquisition code from 7th

```
09_09_2022 09:48:21 + 0.01s/reading
Current: 0.0 0.0044 0.0
09_09_2022 09:48:21 + 0.01s/reading
Current: 0.0 0.0044 0.0
09_09_2022 09:48:21 + 0.01s/reading
Voltage: 199.2 199.7 199.9 0.0
09_09_2022 10:01:33 + 0.001s/reading
Current: 0.0 0.0 0.0 0.0
09_09_2022 10:01:34 + 0.001s/reading
Current: 0.0 0.0 0.0 0.0
Voltage: 199.2 199.7 199.9 0.0
09_09_2022 10:01:34 + 0.001s/reading
Current: 0.0 0.0 0.0 0.0
```

In the same file 9th Sept, the acquisition format of the file changes to record all the four channels

Challenges in data analysis

Heterogeneous data sources means

- **Data formats**
  - 10 Hz Bergoz is tabular ascii
  - Timber data is tabular ascii in CSV format
  - PSU data in ascii but it has its own non-standard format - which is also not consistent over time (slide 8)
  - Digitizers data saved in Matlab binary format
- **Different acquisition rates**
  - 10Hz digitizers,
  - 10Hz Bergoz, if available. Otherwise 1Hz (Timber)
  - Non-periodic PSU data for days 6-7, then pseudo 4Hz data
- **Synchronization**: all data recorded with local PC time
  - no universal triggering time

Challenges in data analysis

A preliminary step of data **pre-processing** consisting in

- **Data formats**
  - Writing MATLAB script to convert .mat into ascii format (CSV)
  - Writing (python) callables to read
    - 10 Hz Bergoz
    - 1 Hz Timber file
    - 10 Hz CSV digitizers
    - PSU data
- **Synchronization**
  - Correlation algorithm to find the O(2sec) time shift between 10Hz Bergoz/digitizers
- **Different acquisition rates**
  - The REFERENCE is digitizers data stream. When 10Hz data is available, synchronization is performed. If there is no data available with such rate, the 'missing' data is created by using the last known data point.

    For example, from digitizer we have data at A: t=1.0 and B: t=1.1 while for PSU at C: t=0.5 and D: t=2.5
    To points A,B it is copied PSU data at C.

    ROOT/CSV file contains Boolean flags telling the user this occurred.

MATLAB conversion script MAT -> CSV

Editor - /home/pietro/work/Clear/Digitizer/Data/matlab/exportToCSV.m

Read_MTV.m | Luxe_acquisition_main.m | plot_folder_progression.m | Read_BCM.m | Read_BPI.m | Read_pulse_picker.m | Luxe_acquire_function.m | exportToCSV.m

```matlab
files = struct2table(dir("*.mat")).name;
disp(files)
for i=1:length(files)
    fname = convertCharsToStrings(files(i));
    data = load(fname);
    %concatData = [data.Signal_01_all,";", data.Signal_02_all,";", data.Signal_03_all,";", data.Signal_04_all,";", data.Signal_05_all,";", data.Signal_06_all,";", data.Signal_07_all,
    concatData = [data.Signal_01_all, data.Signal_02_all, data.Signal_03_all, data.Signal_04_all, data.Signal_05_all, data.Signal_06_all, data.Signal_07_all, data.Signal_08_all, data
    fname_out = replace(fname, ".mat", ".csv");
    disp(fname_out)
    writematrix(concatData, fname_out);
end
```

1. Take all the .mat files in the current directory and
2. Loop over, by performing the operations
3. Load file
4. Take the 12 channels and concatenate into a row
5. Save this in csv
6. Repeat for all the others .mat file in dir

MATLAB conversion script MAT -> CSV

Example

File structure of CSV contains

- 600 rows (+1 empty at the end with cr)

- A row contains 12 x 100 = 1200 columns separated by commas, with values of the digitized wav.
  - The first 100 values pertains to channel 1
  - From column 101 to 201 ch2 data
  - …

Workspace

Name	Value
date_value	'08-Sep-2022 19:59:30'
Signal_01_all	600x100 double
Signal_02_all	600x100 double
Signal_03_all	600x100 double
Signal_04_all	600x100 double
Signal_05_all	600x100 double
Signal_06_all	600x100 double
Signal_07_all	600x100 double
Signal_08_all	600x100 double
Signal_09_all	600x100 double
Signal_10_all	600x100 double
Signal_11_all	600x100 double
Signal_12_all	600x100 double

File repository

Source files are available at Pierre Korysko CERNBOX:
https://cernbox.cern.ch/index.php/s/hv4N3C5rZhARzqj

A backup of the whole directory can be found on the **Confluence GBP**
https://confluence.desy.de/display/GBP/Measures+test+@CLEAR

The 100GB volume **clearTestbeam** can be mounted on **CloudVeneto**, containing

- an uncompressed backup of Pierre's directory
- analysis scripts
- binary dump of pre-processed data
    - meaning memory dumps of integrated waveforms, allowing to run queries and analysis in a faster way than just performing by means of callable
- CSV files with converted digitizers data (>35GB uncompressed : 1GB compr.)

clearTestbeam volume on CloudVeneto

It is 100GB volume that already contains both data and analysis scripts.

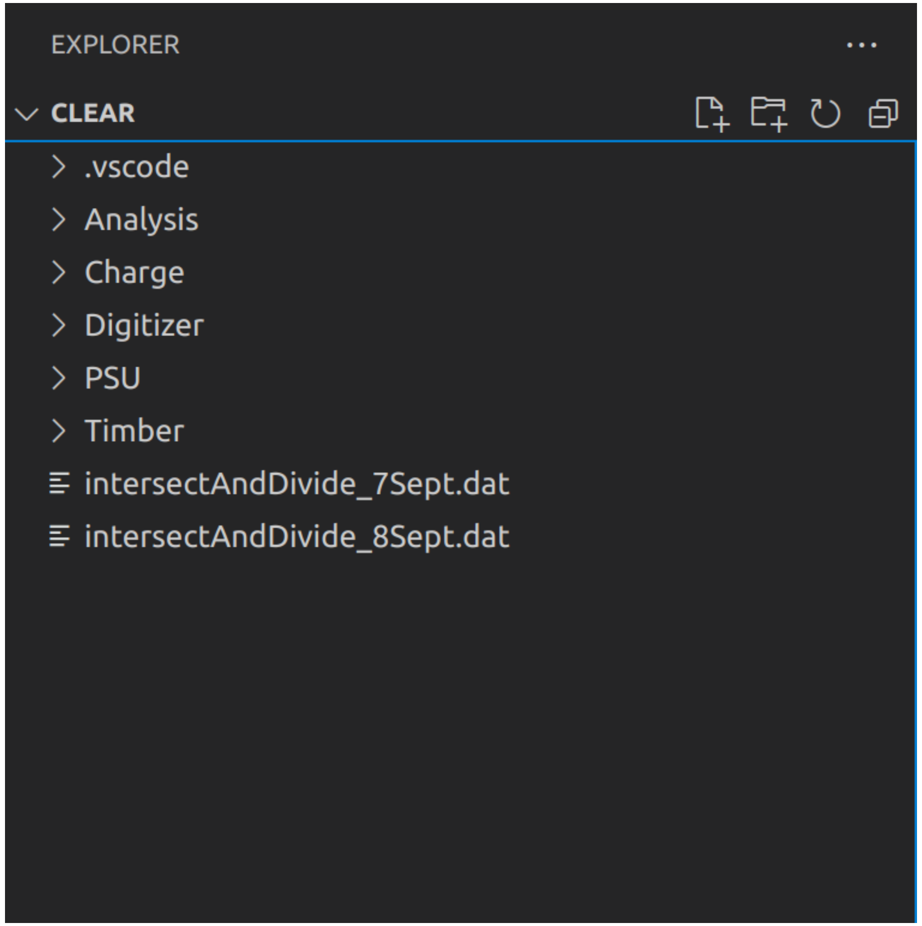
The directory structure is shown on the right.

Problem

A volume can be attached to 1 VM at a time

Possible solutions

- Use NFS (how-to)
- Duplicate volume and transfer the ownership to you



Analysis tools

```
OUTLINE
_core_attachTimestamp
_core_baselineValue
_core_dateFromFile
_core_getCSVList
_core_getCSVMeasureDay
_core_getDateFromFilename
_core_getDATList
_core_getDATMeasureDay
_core_getIntegratedCharge
_core_getMeasureStartEndTime
_core_getPSUList
_core_getPSUMeasureDay
_core_getTimestep
_core_getWaveformsFromFile
_core_integrateWaveform
_core_readDATFile
_core_readPSUFile
_core_readTimberFile
analyze
chg_concat
fileExplorer
inspectBaseline
inspectBergoz
inspectFile
inspectWaveform
intersectAndDivide
irradiation
makeCSV
makeROOT6Sept
makeROOT7Sept_scanX
offsets
```

```
plot_Irradiation
plotAllDay
plotAllDay_Bergoz
plotAllDay_Digitizers
plotAllDay_PwrSupply
plotBergozTimber
plotMotors
plotPSU
plotPSUCurrent
plotPSUVoltage
plotTimber
plotTogether
plotTogether_synch
plotTogether_synchInspect
psu_concat
synch_offset
synch_studyOffset
syncWaveform
wav_concat
wav_concat_synch
```

The >2500 lines of code are well commented. Most of the callables have default arguments, allowing to call them also in case some of the parameters are not specified.

Pre-processing and analysis

Also, data **pre-processing** include

– **Waveform integration,** with baseline subtraction.

- Baseline algorithm in short
  - Calculate first derivative
    - » Smooth waveform with a N=10 samples window (by convoluting with constant function)
    - » Calculate first derivative with finite difference with nearest neighbour
    - » Normalize the first derivative – e.g. to have absolute threshold definition – and
  - Look for the presence of a peak by checking for points in first derivative over 'threshold' (default 3.0 works fine)
  - Baseline value is the average of the samples before the peak position

For more info check here…

```
# Return the value of the integral of the waveform
def integrateWaveform(data: list, baseline = "auto", pedestal = 0., threshold = 3.0) -> float:…

# Return the constant value of the baseline
def baselineValue(data:list, mode = "auto", threshold = 3.0):…
```

- Integration is (sum of points * timestep) and gives (V*sec) quantity
  - Timestep must be read from CLEAR logbook by looking at the digitizer horizontal scale

Diagnostic

There are several functions to check what is going on in the pre-processing script.
For example, they include a callable:

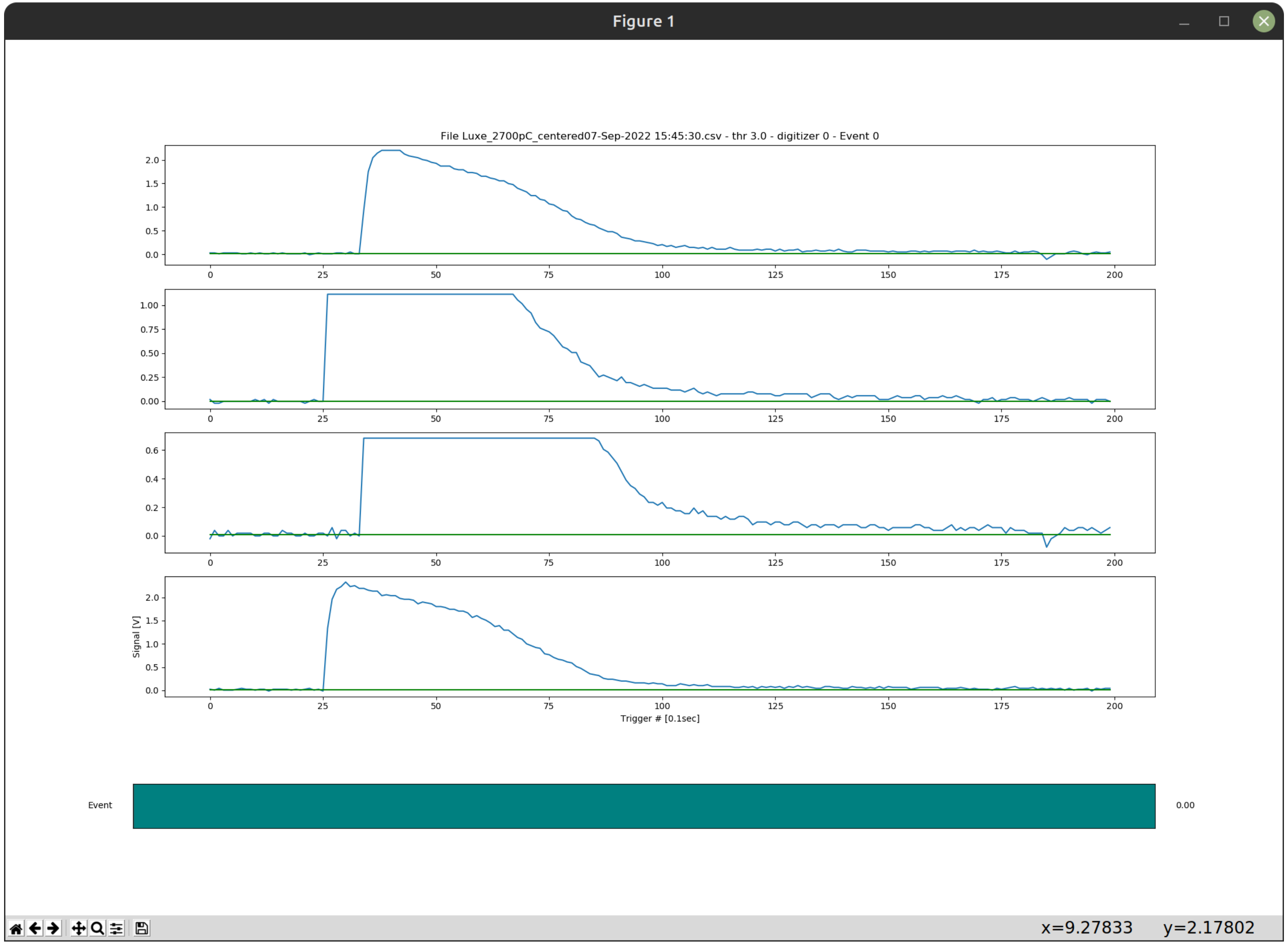
1. to inspect files
   - inspectFile,
   - fileExplorer,
   - inspectBergoz,
   - plotMotors,
   - plotBergozTimber,
   - PlotTimber,
   - plotPSU,
   - [...]
2. to check if waveform integration is doing fine
   - fileExplorer
   - inspectWaveform
3. to inspect baseline algorithm (e.g. inspectBaseline).

...and so on.

All the documentation is provided in the source code with **#comments**
Examples follow...to skip go to slide 24

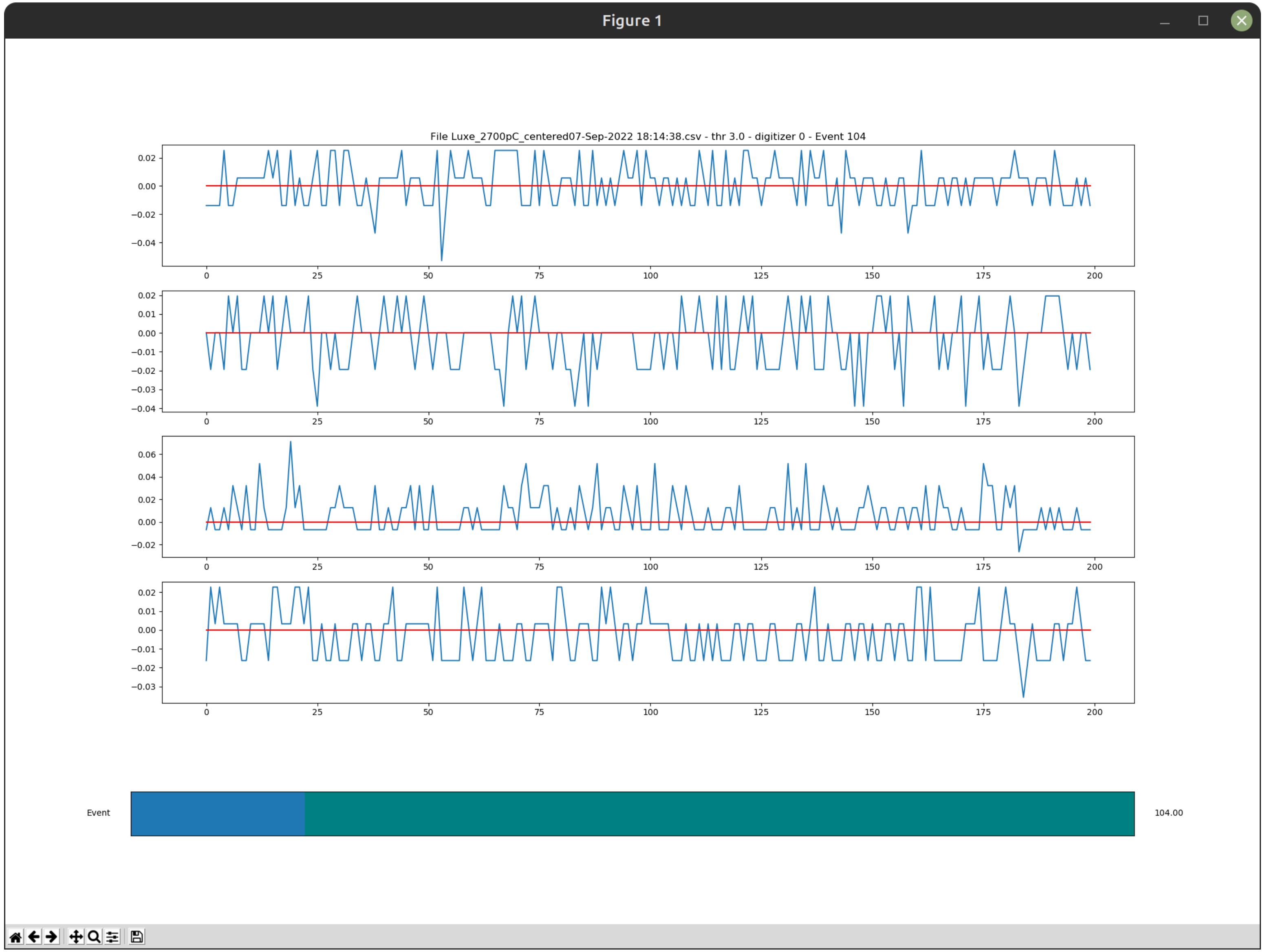
inspectWaveform(fname, digitizer, baselineAlgorithm, thres)

This function opens a window with a plot of the waveforms vs trigger number for a certain file. A slider in the lower part of the window can be used to browse the files in the Digitizer/Data folder (ascending time sorted). The day variable restricts file browsing to measures took in the Nth day (by default 7th Sept.)



inspectWaveform(fname, digitizer, baselineAlgorithm, thres)

This function opens a window with a plot of the waveforms vs trigger number for a certain file. A slider in the lower part of the window can be used to browse the files in the Digitizer/Data folder (ascending time sorted). The day variable restricts file browsing to measures took in the Nth day (by default 7th Sept.)

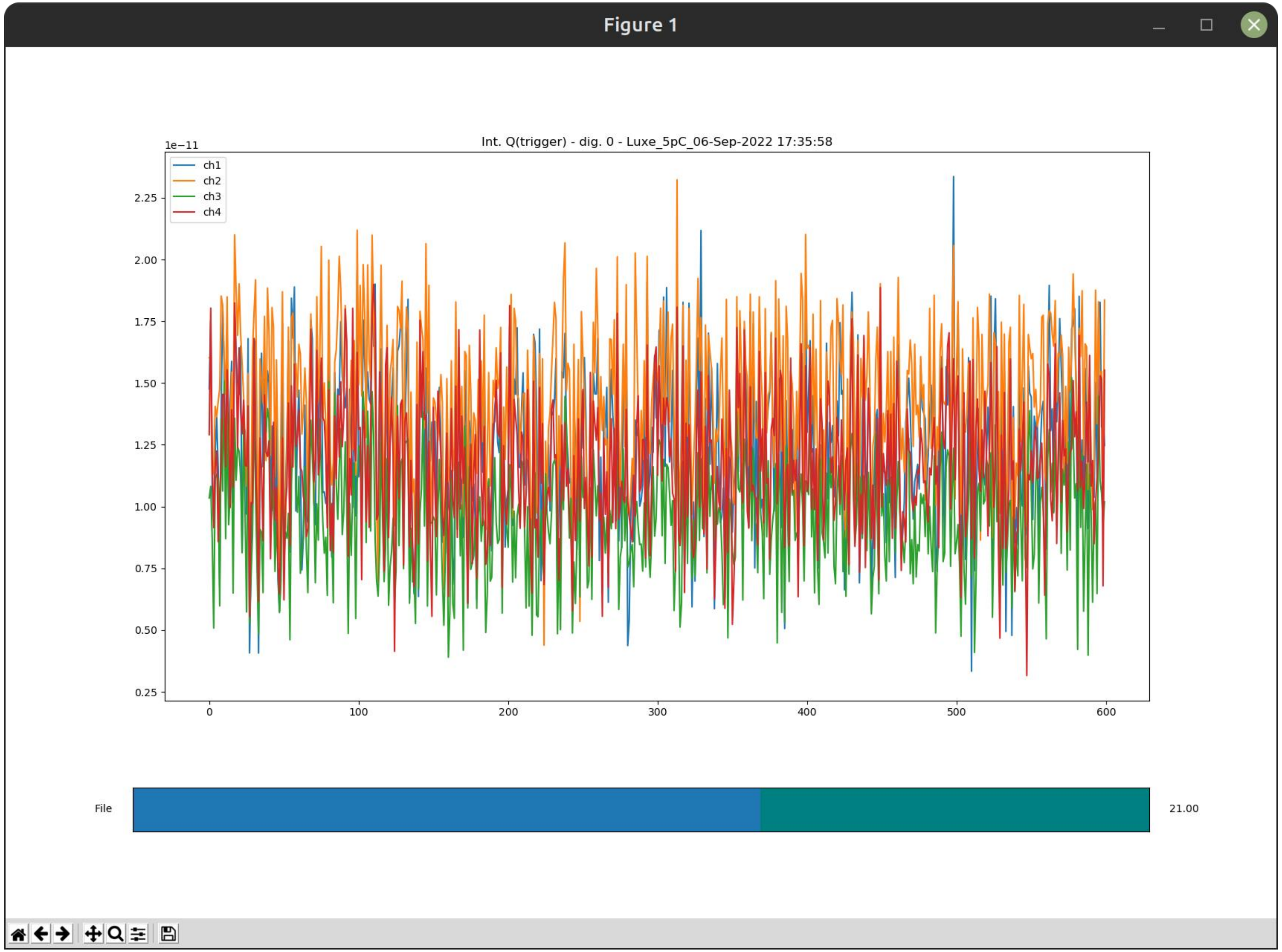


fileExplorer(day)

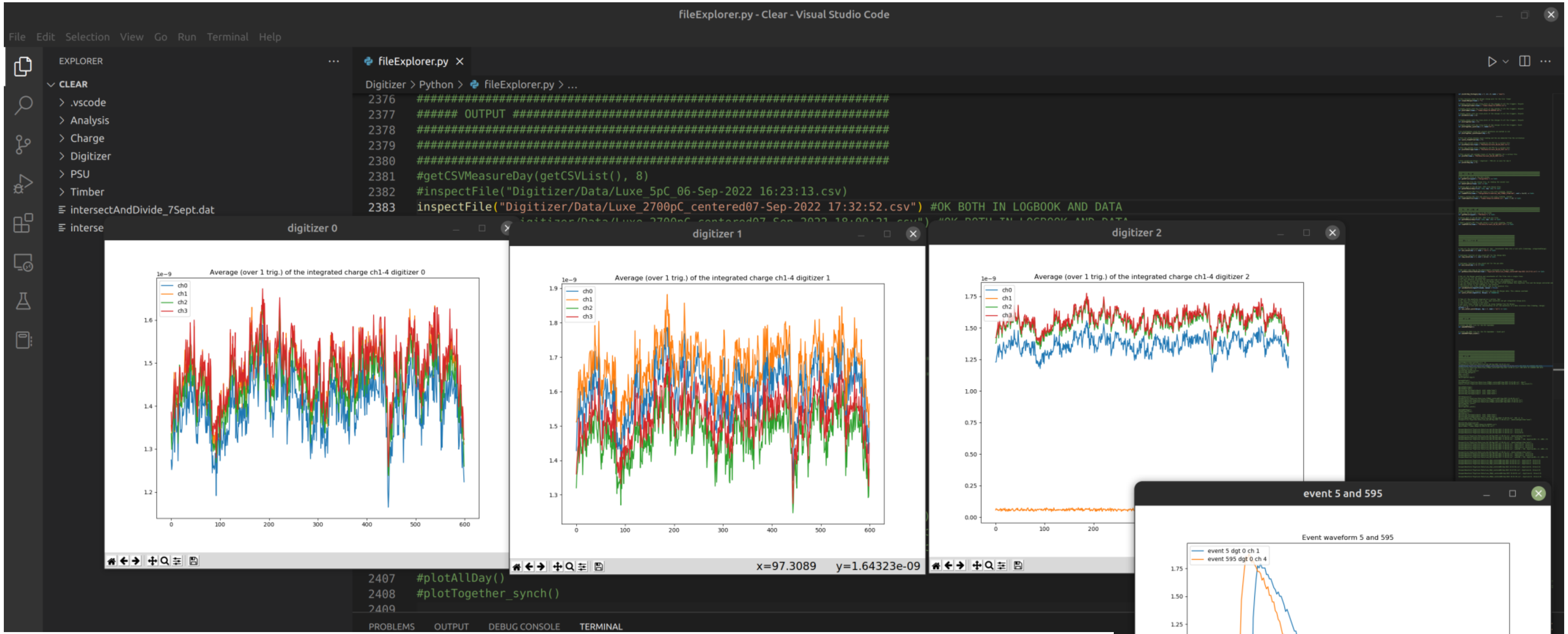
This function opens a window with a plot of the integrated charge vs trigger number for a certain file.

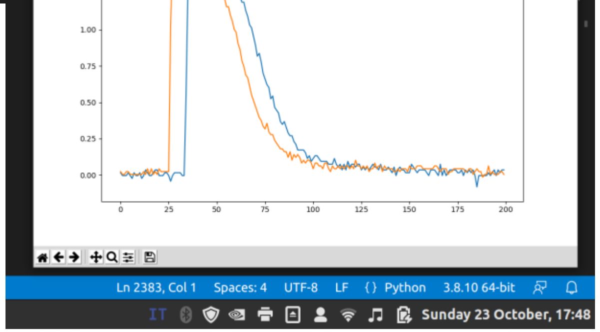
A slider in the lower part of the window can be used to browse the files in the Digitizer/Data folder (ascending time sorted).

The day variable restricts file browsing to measures took in the Nth day (by default 7th Sept.)



inspectFile

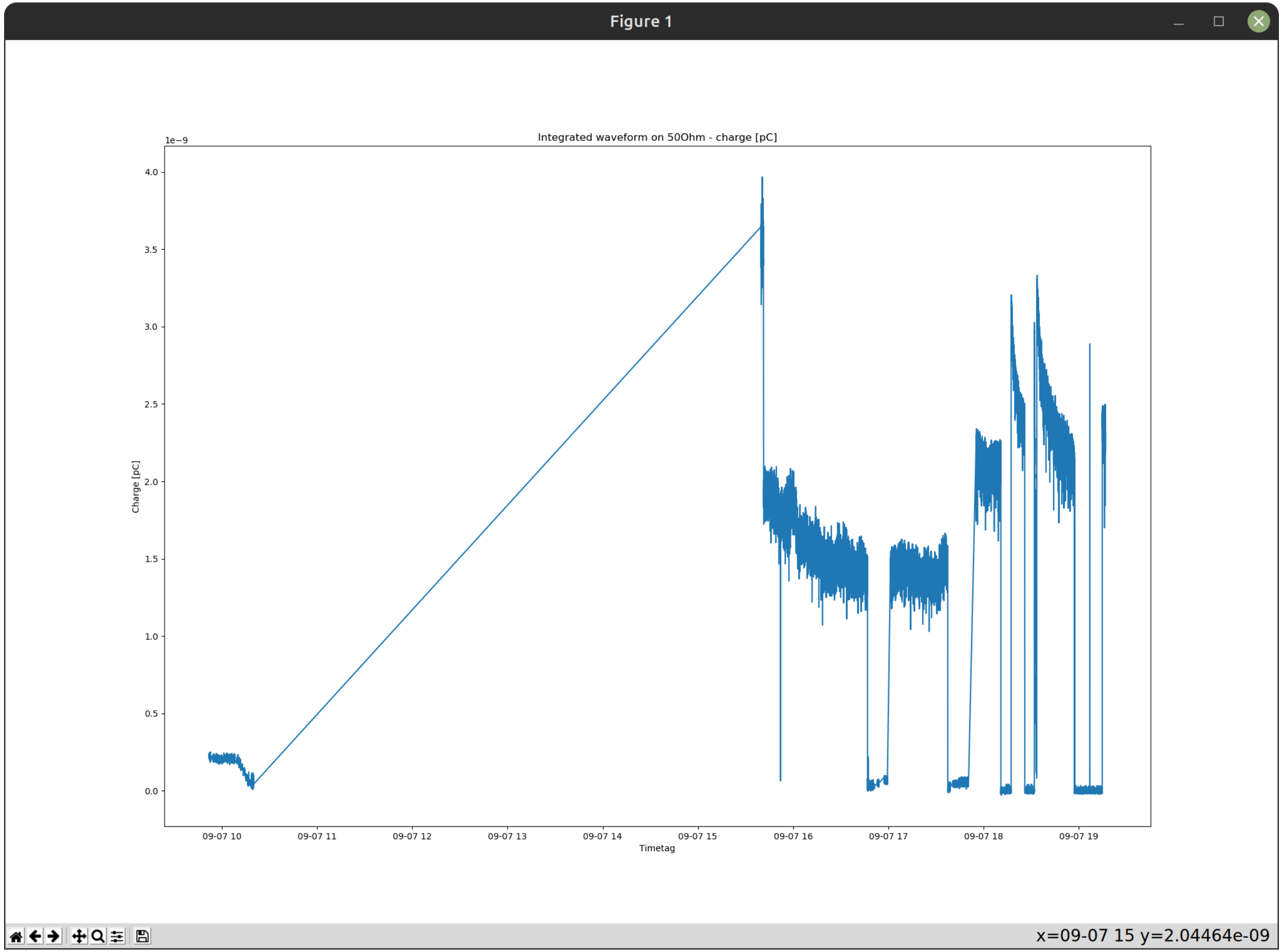




Each digitizer gets a window and integrated charge per each shot (60 here) is shown. In the example, the file "luxe_5pC ..." was recorded when there wasn't beam. **Baseline ON**

plotAllDay_Digitizers(day)

Analogous to the function fileExplorer. The plot this time shows the integrated charge for all the measures in the 'day' day



Status of the analysis

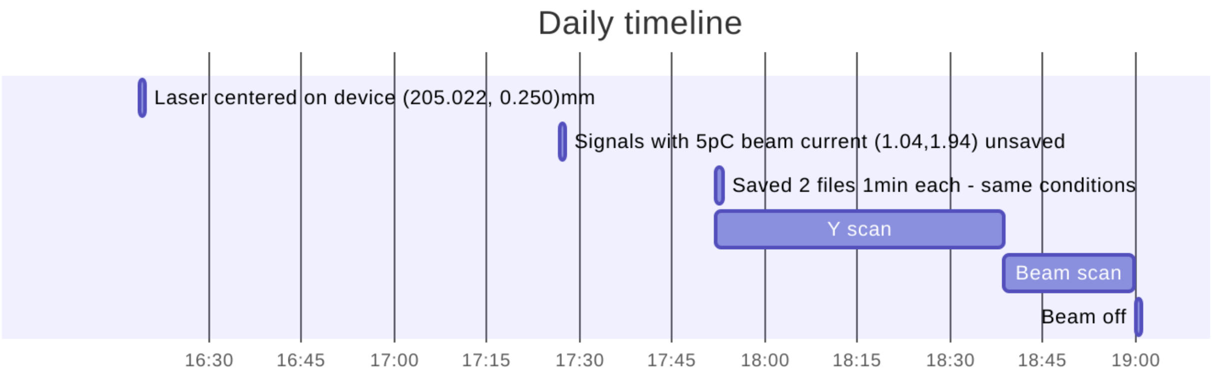
Experimental measures cover the dates 6, 7, 8, 9, 12, 13

- Analysis software fully developed.
- Measures of day 6-7 have been analysed: preliminary results (6-7) concerning horizontal-scan and irradiation (7) are present.
- So far, most of the efforts gone into the preliminary development of the software infrastructure for the analysis: **Measures of days 8-… to be analysed yet!**
- A detailed picture of what happened has been achieved for days 5-9 (check Hedgedoc logbook)

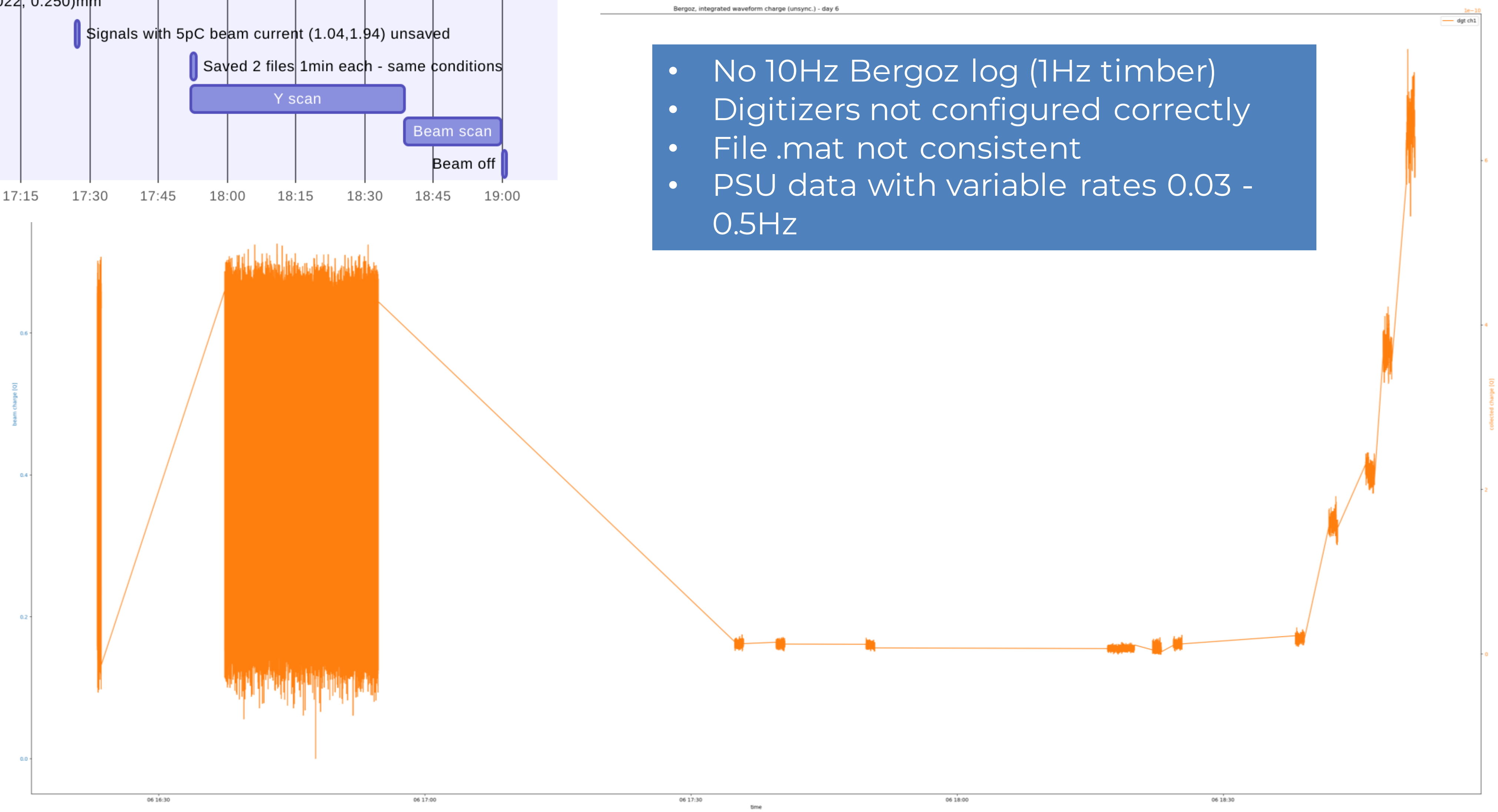
Issues

- A few acquisition from first day are *cursed:* different samples nb. between channels !
- It seems that non-linear response of the detector to beam charge is observed.

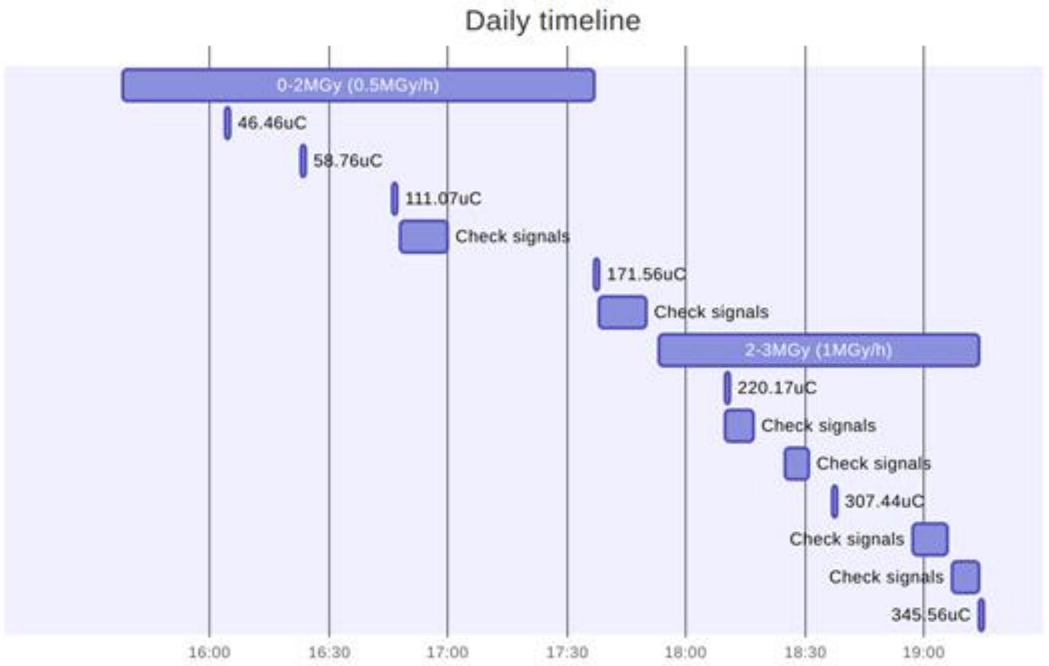
6th Sept.



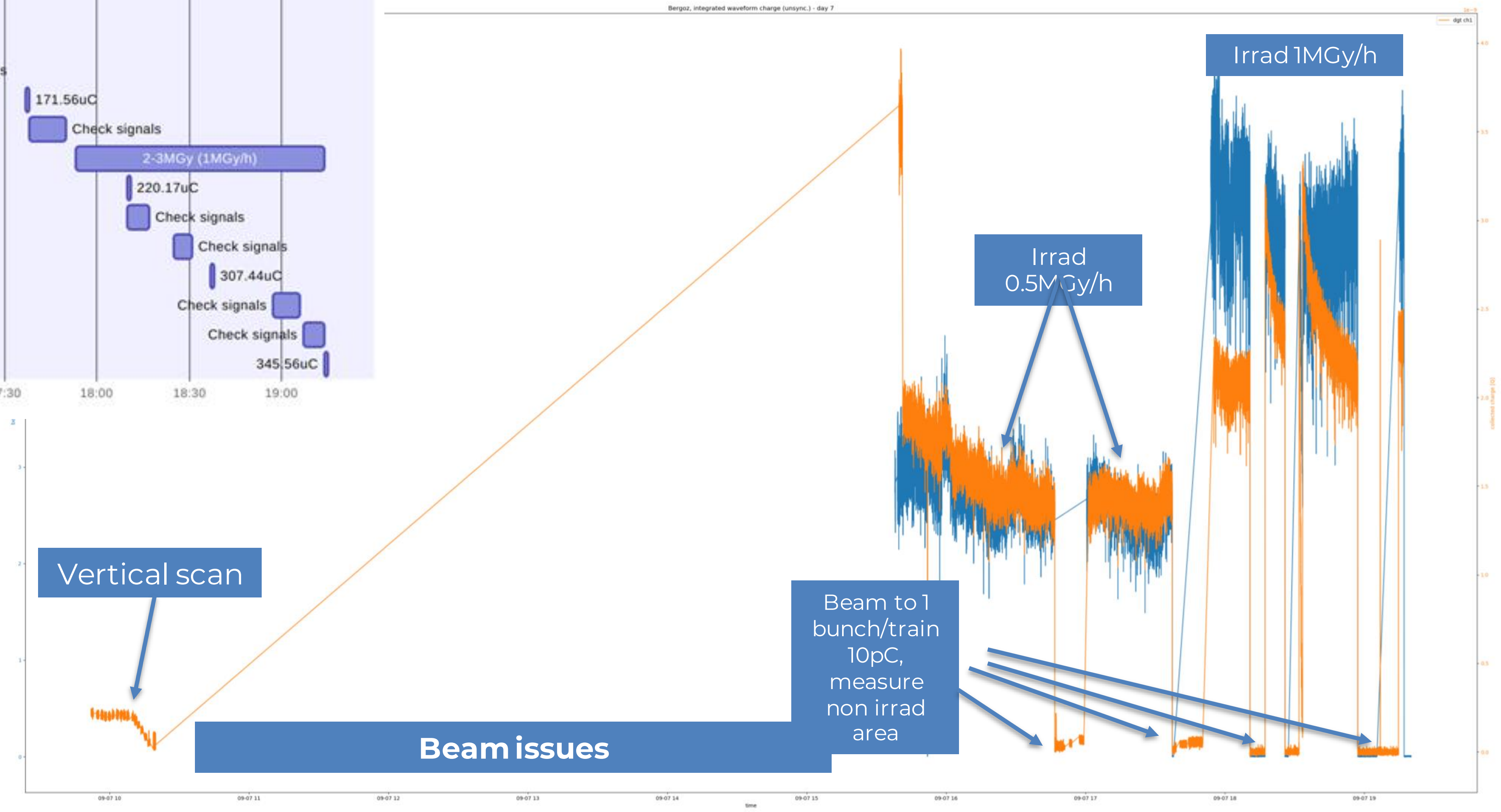
- No 10Hz Bergoz log (1Hz timber)
- Digitizers not configured correctly
- File .mat not consistent
- PSU data with variable rates 0.03 - 0.5Hz



7th Sept.

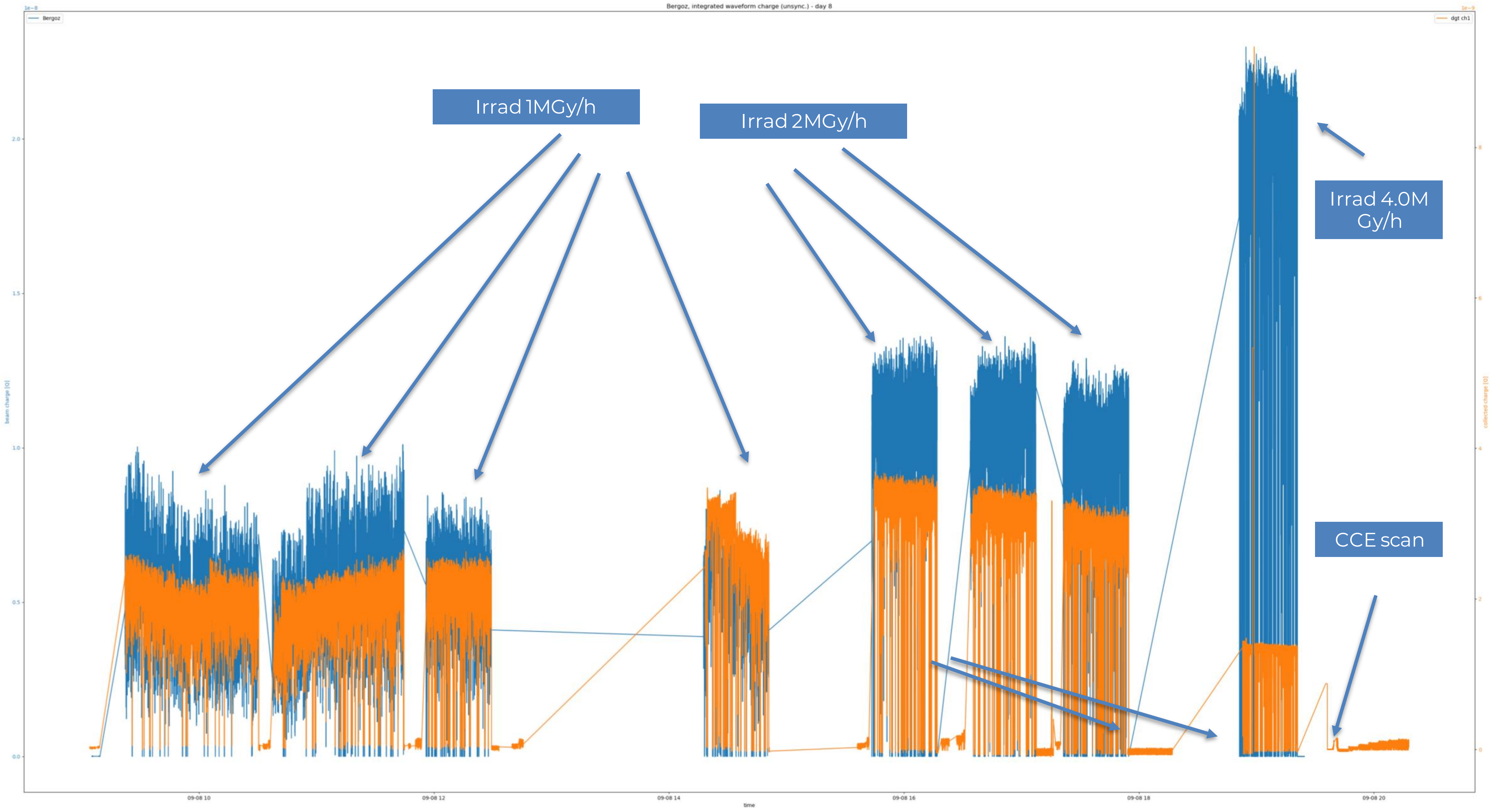


Daily timeline
0-2MGy (0.5MGy/h)
46.46uC
58.76uC
111.07uC
Check signals
171.56uC
Check signals
2-3MGy (1MGy/h)
220.17uC
Check signals
Check signals
307.44uC
Check signals
Check signals
345.56uC
16:00
16:30
17:00
17:30
18:00
18:30
19:00



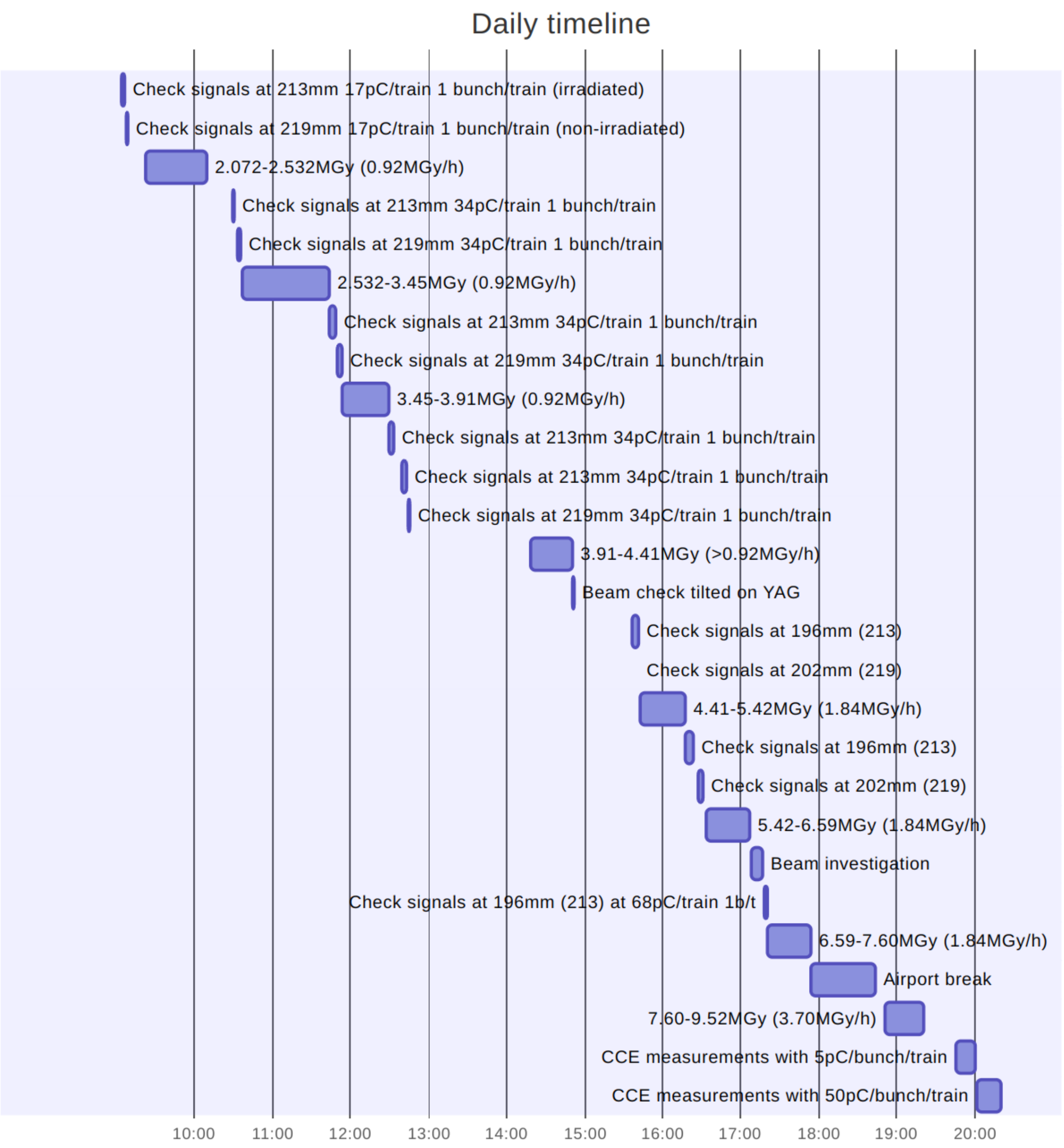
Irrad 1MGy/h
Irrad 0.5MGy/h
Vertical scan
Beam issues
Beam to 1 bunch/train 10pC, measure non irrad area

8th Sept.



Bergoz, integrated waveform charge (unsync.) - day 8
Bergoz
dgt ch1
beam charge [Q]
collected charge [Q]
time
09-08 10
09-08 12
09-08 14
09-08 16
09-08 18
09-08 20
Irrad 1MGy/h
Irrad 2MGy/h
Irrad 4.0M Gy/h
CCE scan

8th Sept.



Daily timeline
Check signals at 213mm 17pC/train 1 bunch/train (irradiated)
Check signals at 219mm 17pC/train 1 bunch/train (non-irradiated)
2.072-2.532MGy (0.92MGy/h)
Check signals at 213mm 34pC/train 1 bunch/train
Check signals at 219mm 34pC/train 1 bunch/train
2.532-3.45MGy (0.92MGy/h)
Check signals at 213mm 34pC/train 1 bunch/train
Check signals at 219mm 34pC/train 1 bunch/train
3.45-3.91MGy (0.92MGy/h)
Check signals at 213mm 34pC/train 1 bunch/train
Check signals at 213mm 34pC/train 1 bunch/train
Check signals at 219mm 34pC/train 1 bunch/train
3.91-4.41MGy (>0.92MGy/h)
Beam check tilted on YAG
Check signals at 196mm (213)
Check signals at 202mm (219)
4.41-5.42MGy (1.84MGy/h)
Check signals at 196mm (213)
Check signals at 202mm (219)
5.42-6.59MGy (1.84MGy/h)
Beam investigation
Check signals at 196mm (213) at 68pC/train 1b/t
6.59-7.60MGy (1.84MGy/h)
Airport break
7.60-9.52MGy (3.70MGy/h)
CCE measurements with 5pC/bunch/train
CCE measurements with 50pC/bunch/train
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
18:00
19:00
20:00