

# 1 Описание

Программа decoder декодирует результаты измерений программой Digitizer 1.1.1, сохраненные в бинарном формате. Программа позволяет сохранять результаты в формате .root и .txt и гистограммы в формате .root и .png.

Исходный код программы находится в репозитории: [GitHub](#).

# 2 Зависимости

- ROOT
- cmake and make
- C++17

# 3 Инструкция по установке

## 3.1 Linux

Установка зависимостей (для Ubuntu/Linux Mint):

```
1 sudo apt install gcc cmake
```

Установка программы:

```
1 git clone https://github.com/kuzmenkoas/decoder
2 cd decoder
3 mkdir build
4 cd build
5 cmake ..
6 make
```

## 3.2 Windows

1. Установить [Visual Studio](#).
2. Установить [CMake](#).
3. Скомпилировать проект с помощью утилиты CMake (использовать компилятор MSVC).
4. Собрать проект с помощью Visual Studio (кнопка decoder.exe).
5. Добавить путь к исполняемому файлу (путь/decoder/out/build/x64-Debug/) в переменную окружения PATH (Win+R, вызвать sysdm.cpl).

# 4 Запуск

Для запуска необходимо указать путь к исполняемому файлу (в Linux build/decoder), в Windows, если добавить путь в переменную окружения PATH, то из любой директории можно вызвать decoder.exe.

Программа позволяет использовать два варианта задания конфигурации декодирования (с помощью командной строки и с помощью конфигурационного файла).

Так же у программы предусмотрено различное поведение в зависимости от количества и типа входных бинарных файлов (PSD и Waveform).

Очередность передаваемых аргументов программе строго типизирована. Первым, если нужен, передается конфигурационный файл, далее идут бинарные файлы, если передавать два бинарных файла, то сначала PSD, после Waveform.

Далее, для примера будут использованы конфигурационные файлы, которые лежат в директории decoder/cfg и бинарные файлы в директории decoder/data.

Декодировать PSD и Waveform с помощью конфигурационного файла:

```
1 decoder.exe cfg/psdWaveform.cfg data/  
    raw_data_psd_wavetest_2025_11_06__13_28_09.bin data/  
    raw_waveform_psd_wavetest_2025_11_06__13_28_09.bin
```

Декодировать PSD и Waveform с помощью командной строки:

```
1 decoder.exe data/raw_data_psd_wavetest_2025_11_06__13_28_09.bin data/  
    raw_waveform_psd_wavetest_2025_11_06__13_28_09.bin
```

Декодировать PSD с помощью конфигурационного файла:

```
1 decoder.exe cfg/psd.cfg data/raw_data_psd_wavetest_2025_11_06__13_28_09.  
    bin
```

Декодировать PSD с помощью командной строки:

```
1 decoder.exe data/raw_data_psd_wavetest_2025_11_06__13_28_09.bin
```

Декодировать Waveform с помощью конфигурационного файла:

```
1 decoder.exe cfg/waveform.cfg data/  
    raw_waveform_psd_wavetest_2025_11_06__13_28_09.bin
```

Декодировать Waveform с помощью командной строки:

```
1 decoder.exe data/raw_waveform_psd_wavetest_2025_11_06__13_28_09.bin
```

## 5 Конфигурация декодера

Есть два парсера для конфигурации декодера – через командную строку и через конфигурационный файл.

### 5.1 Командная строка

После запуска программы, в зависимости от количества бинарных файлов и их типов, различные конфигурационные модули запустятся.

#### 1. Тип бинарного файла

Если передать 1 бинарный файл в программу, то выведется следующее сообщение:

```
1 What file is it?  
2 (1) PSD  
3 (2) Waveform
```

Ведите 1 для типа PSD и 2, если бинарный файл содержит форму сигнала (waveform). Это влияет на алгоритм декодирования.

#### 2. Формат сохранения результатов

Следующий вопрос появится вне зависимости от количества переданных файлов и их типов.

```
1 Choose format to save
2   (1) Root
3   (2) Txt
4   (3) Root and Txt
```

Необходимо ввести 1, 2 или 3 для выбора формата сохранения декодированных данных.

### 3. Какие данные находятся в бинарном файле PSD

Данный текст появится, если есть входной бинарный файл PSD. Он говорит программе, какие данные сохранены в бинарном файле (необходимо для корректного декодирования). Зная параметры, программа в определенной последовательности декодирует данные (зная, сколько байт уходит на тот или иной параметр).

```
1 What is encode in binary PSD file.
2 Choose everything that is stored in bin file
3 (decoder will not be able to work correctly if this parameters will be
   incorrect)!
4 For multiple write as 123 - for qShort, qLong and cfd_y1.
5 If in binary file stored all parameters choose 0 (ALL):
6   (1) qShort
7   (2) qLong
8   (3) cfd_y1
9   (4) cfd_y2
10  (5) baseline
11  (6) height
12  (7) eventCounter
13  (8) eventCounterPSD
14  (9) psdValue
15  (0) ALL
```

Введите цифры выбранных параметров непрерывно. Например, 126 для qShort, qLong и height, 157 для qShort, baseline и eventCounter. Если в бинарном файле сохранены все параметры, то достаточно ввести только 0.

### 4. Перевернуть сигнал или нет

Этот модуль появится если:

1. бинарный файл PSD содержит qShort, qLong или baseline.
2. для переданного бинарного файла waveform модуль появится в любом случае.
3. если есть оба бинарных файла – модуль появится.

```
1 Reverse integral?
2   (1) Yes
3   (2) Not
```

Если введете 1, то результат baseline, qShort и qLong будет перевернут (умножен на -1).

### 5. Конфигурация формы сигнала

Модуль появится, если в входных файлах есть бинарный файл waveform.

```
1 The number of baseline points?
1 The number of qShort points?
1 The number of qLong points?
```

Для всех них необходимо, чтобы пользователь ввел целое число (integer). Для одного события у формы сигнала есть определенная длина (настраивается в программе Digitizer). Baseline points необходимо для определения шума (baseline), определяется как

среднее значение в диапазоне (первые точки сигнала от 0 до введенного пользователем). Далее, от baseline points до qShort points определяется значение интеграла qShort. И последнее, от baseline points до qLong points определяется значение интеграла qLong.

Данное сообщение появится, если входным бинарным файлом будет только waveform.

#### 1 The number of waveform points?

Необходимо ввести целое число, характеризующую длину форму сигнала (задается в программе Digitizer, обязаны совпасть при декодировании, иначе программа некорректно сработает).

### 6. Конфигурация гистограмм

Данное сообщение появится независимо от входных файлов. Здесь настраивается гистограмма по декодированным данным (для PSD и Waveform шаблон одинаковый, количество вариантов данных – разное).

```
1 Choose parameter to plot (multiply input, example: 123 for 3 parameters to
   plot)
2 (0) exit
3 (1) qShort
4 (2) qLong
5 (3) cfd_y1
6 (4) cfd_y2
7 (5) baseline
8 (6) height
9 (7) eventCounter
```

Необходимо непрерывно ввести все варианты, для которых необходима гистограмма. Если гистограмма не нужна, то достаточно ввести значение 0.

Далее, если необходима гистограмма для каждого параметра появятся выводы с настройкой.

```
1 Configure a plot from PSD data for parameter qShort
2
3 Enter number of bins
```

Настраивается количество бинов – целое число.

```
1 Enter min value of histogram
```

Необходимо ввести минимальное значение для гистограммы (целое число, может быть отрицательным).

```
1 Enter max value of histogram
```

Необходимо ввести максимальное значение для гистограммы (целое число, может быть отрицательным).

## 5.2 Конфигурационный файл

Можно все настройки для декодера написать в конфигурационном файле и скомпилировать программе.

Структура файла:

```
1 Output
2 Reverse
3 DataPSD
4 DataWaveform
5 WaveformConfig
6 Histogram
```

Для любого файла обязательными полями являются: Output.

Необязательные: Reverse, Histogram.

Для PSD бинарного файла необходимо заполнить поле DataPSD.

Для Waveform бинарного файла необходимо заполнить поля DataWaveform и WaveformConfig.

В Output настраивается формат вывода (txt и root): RootNtuple и TxtNtuple.

Выглядит как:

```
1 Output
2 + RootNtuple
3 + TxtNtuple
```

В результате все выводится в root файл и txt файлы.

Если необходимо сохранить только в root файле (т.к. стоит учитывать, что вывод в txt формате долгий), то:

```
1 Output
2 + RootNtuple
```

Значения baseline, qShort и qLong (интегралы) можно реверсировать (умножить на -1). Делается с помощью параметра Reverse:

```
1 Reverse true
```

или

```
1 Reverse false
```

При значении true – умножит эти значения (для PSD и Waveform данных) на -1, при значении false – не будет умножать.

В DataPSD необходимо указать все переменные, которые закодированы в бинарном файле (если некорректно указать, то результат декодирования будет неверным).

Все возможные переменные указаны ниже:

```
1 DataPSD
2 + qShort
3 + qLong
4 + cfd_y1
5 + cfd_y2
6 + baseline
7 + height
8 + eventCounter
9 + eventCounterPSD
10 + psdValue
```

В DataWaveform указывается, какие переменные необходимо получить. Есть 4 параметра:

```
1 DataWaveform
2 + qShort
3 + qLong
4 + baseline
5 + entries
```

Из нового тут только параметр entries – сохраняет в отдельном ntuple значения формы сигнала как по одному событию. Есть значения id – номер события, t – момент времени, когда получено значение сигнала, wave – значение сигнала.

Далее, в WaveformConfig задаются значения для декодирования: точки отсчета для baseline, qShort и qLong (как в программе Digitizer, при этом необязательно задавать такие же) и длина сигнала (wavelength).

```

1 WaveformConfig
2 + baseline 100
3 + qShort 120
4 + qLong 380
5 + wavelength 1000

```

При этом, значение wavelength обязано быть таким же, как при измерениях, и его необязательно указывать, если программа получает два входных бинарных файла – PSD и Waveform.

Для настройки гистограмм необходимо указать: тип файла, параметр, количество бинов, минимальное значение, максимальное значение. Именно в такой последовательности!

Пример:

```

1 Histogram
2 + PSD qShort 1000 -20000 200000
3 + Waveform qShort 1000 -20000 200000

```

Пример конфигурационного файла для декодирования PSD и Waveform:

```

1 Output
2 + RootNtuple
3 + TxtNtuple
4
5 Reverse false
6
7 DataPSD
8 + qShort
9 + qLong
10 + cfd_y1
11 + cfd_y2
12 + baseline
13 + height
14 + eventCounter
15 + eventCounterPSD
16 + psdValue
17
18 DataWaveform
19 + qShort
20 + qLong
21 + baseline
22 + entries
23
24 WaveformConfig
25 + baseline 100
26 + qShort 120
27 + qLong 380
28
29 Histogram
30 + PSD qShort 1000 -20000 200000
31 + Waveform qShort 1000 -20000 200000

```

Пример конфигурационного файла для декодирования только PSD:

```

1 Output
2 + RootNtuple
3 + TxtNtuple
4
5 Reverse false
6
7 DataPSD
8 + qShort

```

```
9 + qLong
10 + cfd_y1
11 + cfd_y2
12 + baseline
13 + height
14 + eventCounter
15 + eventCounterPSD
16 + psdValue
17
18 Histogram
19 + PSD qShort 1000 -20000 200000
```

Пример конфигурационного файла для декодирования только Waveform:

```
1 Output
2 + RootNtuple
3
4 Reverse false
5
6 DataWaveform
7 + qShort
8 + qLong
9 + baseline
10 + entries
11
12 WaveformConfig
13 + baseline 100
14 + qShort 120
15 + qLong 380
16 + wavelength 1000
17
18 Histogram
19 + Waveform qShort 1000 -20000 200000
```