

Автоматизированный измерительный стенд

Андрей Козий

13 сентября 2021 г.

1 Обзор стенда

Автоматизированный Измерительный Стенд (АИС) предназначен для измерения параметров Детектора Одиночных Фотонов (ДОФ), с последующей программной обработкой полученных данных и формирования отчетов по эксплуатационным характеристикам устройств.

Принципиальная схема стенда представлена на рисунке 1. Чёрные стрелки обозначают контакт двух устройств с использованием проводов, желтые – оптоволокна.

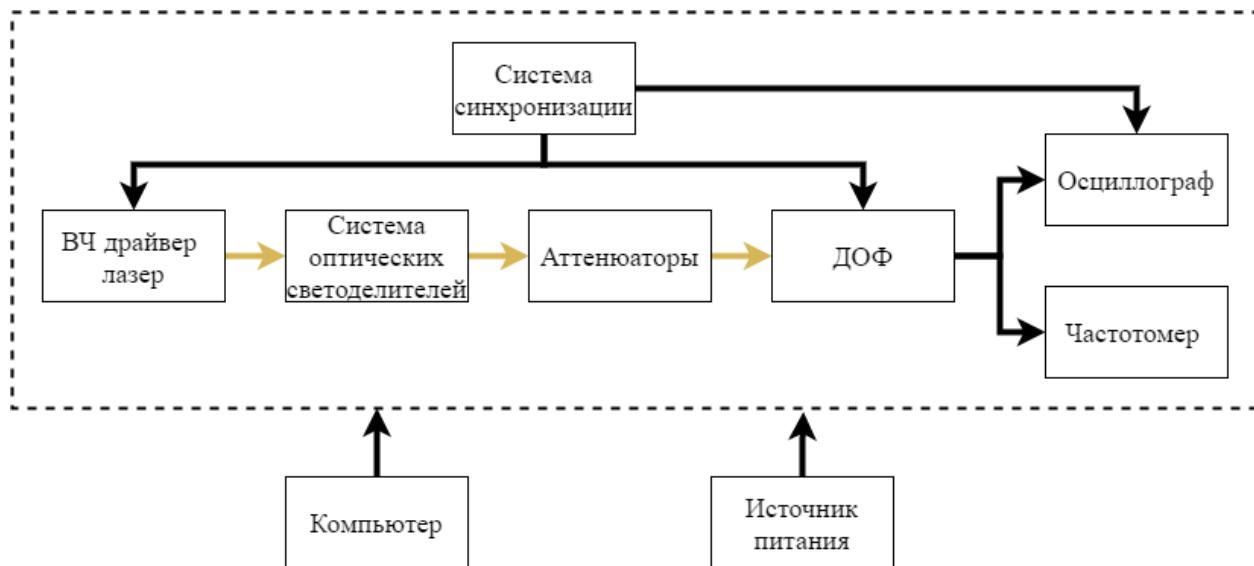


Рис. 1: Принципиальная схема автоматизированного измерительного стенда.

На рисунке 2 представлена фотография измерительного стенда.

На рисунке 3 представлены фотографии отдельных узлов измерительного стенда.

Система синхронизации позволяет генерировать лазерные импульсы и стробы в ДОФ в одной сетке с частотой 312.5 МГц. Синхро-импульсы передаются на три блока – ВЧ драйвер лазера, ДОФ и осциллограф. Сигнал, приходящий на ВЧ драйвер и на осциллограф имеет одинаковую форму, но отличается только по амплитуде.

ВЧ драйвер лазера управляет формой лазерного импульса и периодом его следования в частотной сетке, определенной системой синхронизации. Используемая частота следования лазерных импульсов: 10 кГц, 100 кГц, 156.25 МГц, 312.5 МГц (другие на практике не используются).

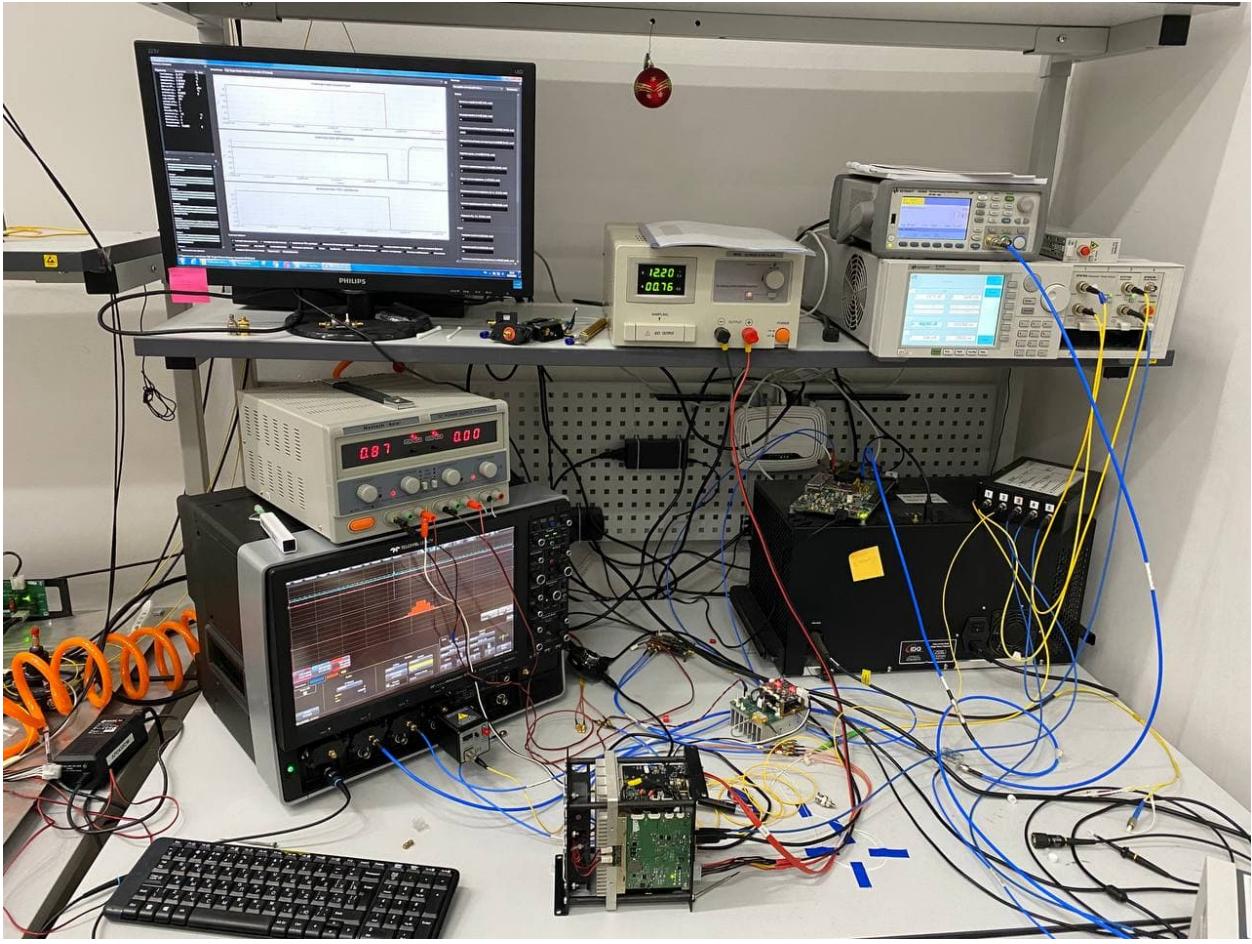


Рис. 2: Фотография измерительного стенда.

Лазерные импульсы передаются в систему оптических светоделителей, где одно из выходных направлений используется для последующей передачи оптического излучения на аттенюаторы.

Для ослабления лазерного излучения используется система из двух последовательно соединенных аттенюаторов. Первый аттенюатор обладает встроенным измерителем мощности на выходе. Это позволяет поддерживать и точно контролировать мощность выходного оптического излучения. Второй аттенюатор настроен на постоянное поглощение в 60 dB.

После системы аттенюаторов сильно ослабленное световое излучение (со средним числом фотонов на импульс 0.1 – 1, в зависимости от типа проводимых измерений) передается на детектор одиночных фотонов, где и происходит регистрация.

Электрический сигнал с ДОФ передается на частотомер и осциллограф. По частотометру определяется скорость счета и благодаря простым математическим преобразованиям вероятность детектирования фотона PDE . На осциллографе, используя сигнал с ДОФ как опорный, и регистрируя импульсы системы синхронизации, можно получать гистограммы распределения послеимпульсов и временного разрешения.

Источник питания и компьютер, которые вынесены за пунктирный прямоугольник,

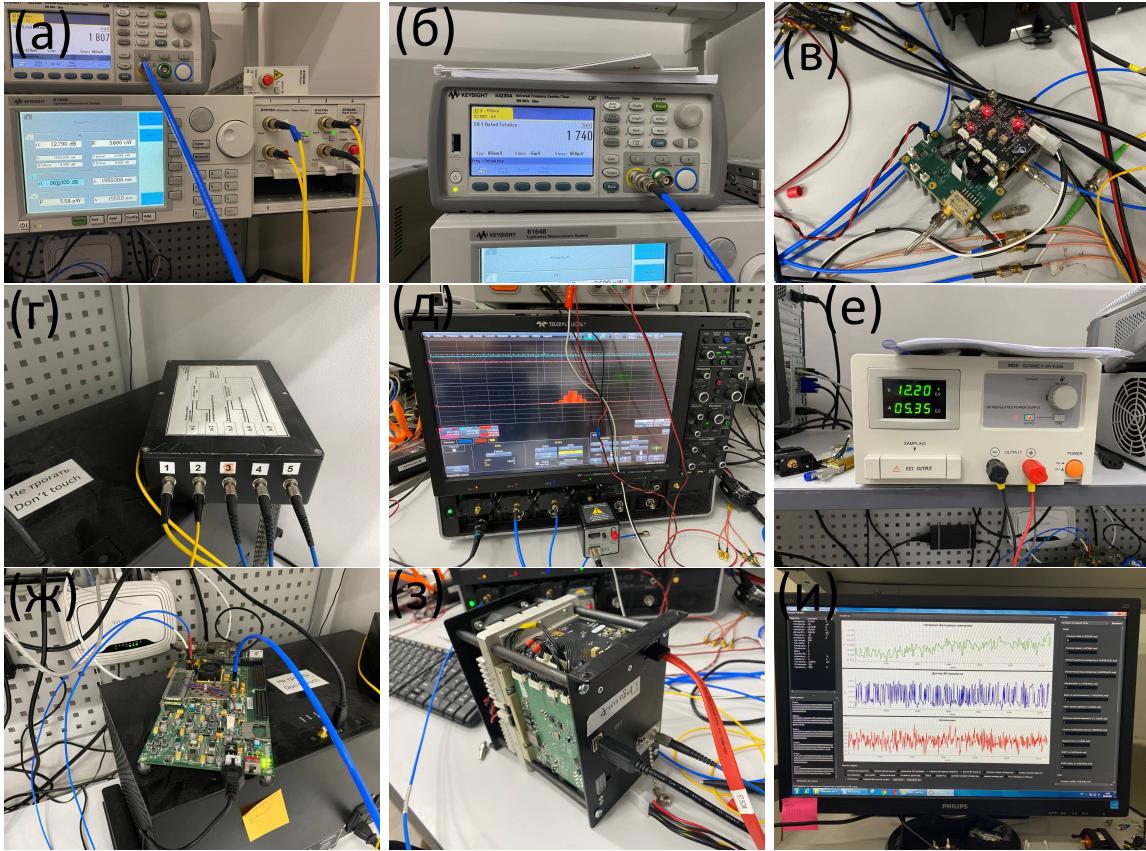


Рис. 3: Фотографии отдельных узлов измерительного стенда. а) блок с аттенюаторами; б) частотомер; в) ВЧ драйвер лазера; г) система оптических светоделителей; д) осциллограф; е) источник питания; ж) схема синхронизации; з) детектор одиночных фотонов; и) компьютер.

подключены к каждому элементу данной системы, исключая систему оптических светоделителей. Компьютер используется для управления каждым блоком данной системы, и для обработки приходящих с них данных.

Далее будет представлено 2 раздела, описывающих работу автоматизированного стенда. В разделе "Алгоритм и настройка измерений" будет подробно описан алгоритм снятия параметров и управления используемых узлов, а также структура файлов настройки параметров измерения стенда. В разделе "Описание программы обработки" будет описана структура скриптов для работы GUI и формирования отчетов.

2 Алгоритм и настройка измерений

Настройка измерений задается с помощью трех файлов:

"Equipment.txt" – файл с основными параметрами, в котором определяются подключаемые блоки; "SPDConfig_strobe.txt" – файл с параметрами измерений для детектора стробированного типа; "SPDConfig_freerun.txt" – файл с параметрами измерений для детектора типа фриран.

2.1 Файл Equipment

В данном файле устанавливается информация об адресах подключаемого оборудования в локальной сети. Сам детектор `DevBAlias` подключается через порт `COM3`, частотомер `FreqCAlias` имеет адрес `KeySight_53230A`, блок с аттенюаторами `8164BIPAddress` имеет адрес `192.168.1.101`, осциллограф `ScopeAlias` имеет адрес `LeCroy`.

```
1 [Equipment]
2 DevBAlias = COM3
3 FreqCAlias = KeySight_53230A
4 8164BIPAddress = 192.168.1.101
5 ScopeAlias = LeCroy
```

2.2 Файл SPDConfig-strobe

0. Тест отсутствия импульса. Проводится как первый этап тестирования сразу после того, как детектор вышел на режим и пользователем была нажата кнопка на проведение первого тестирования. `HV_Bias` задает напряжение смещения на детекторе. Код строба выставляется в 0. `FreqC_GateTime` определяет время проведения данного тестирования в секундах.

```
1 [Test00_NoPulses]
2 HV_Bias = 40
3 FreqC_GateTime = 1
```

1. Тест наличия срабатываний детектора. Проводится автоматически после нулевого этапа. Код строба устанавливается 0. Начиная с напряжения `HV_Bias_Quick_Start` и до напряжения `HV_Bias_Quick_Stop`, изменяется напряжение на детекторе с шагом `HV_Bias_Quick_Step`. Каждый шаг проводится измерения в течение времени `FreqC_Quick_GateTime` в секундах. Когда грубым шагом удалось получить значения темнового счета `DCR` больше величины `PulsesCount_min`, шаг заменяется на более малый `HV_Bias_Precise_Step`, а время измерения заменяется на `FreqC_Precise_GateTime`. Таким образом, напряжение будет изменяться на малый шаг как в сторону уменьшения, так и увеличения, пока не будет получено значение темнового счета `PulsesCount_max`.

```
1 [Test01_HV_bias]
2 FS = 100
3 TS = 2
4 HV_Bias_Quick_Start = 50
5 HV_Bias_Quick_Stop = 80
6 HV_Bias_Quick_Step = 0,5
7 FreqC_Quick_GateTime = 0,5
8 HV_Bias_Precise_Step = 0,05
9 FreqC_Precise_GateTime = 0,5
10 PulsesCount_max = 300
11 PulsesCount_min = 200
```

2. Тест настройки фазы. Проводится автоматически после первого этапа. Проводится только для стробированного детектора. Устанавливается частота следования лазерных импульсов `FS` в кГц, и команда для формирования нужной ширины лазерного импульса `TS`. Длина волны излучения лазера, используемая во внутренних вычислениях `Laser_WaveLength`. Параметр `P_units = 1` показывает, что далее устанавливается

мощность в размерности Вт. На выходе первого аттенюатора контролируется мощность `P_set`. Установленное затухание на втором аттенюаторе `Att`. Параметры выбираются таким образом, чтобы средняя энергия на импульс равнялась одному фотону. Начальная фаза синхронизации между лазерным импульсом и стробами детектора устанавливается равной `Phase_Quick_Start`. После чего алгоритм начинает смещать фазу на величину `Phase_Quick_Step`, вплоть до значения `Phase_Quick_Stop`. На каждой фазе измерение скорости счета выполняется в течение `FreqC_Quick_Gate`. После того, как указанный диапазон фазы был пройден грубым шагом, устанавливается то значение фазы, при котором была получена максимальная скорость счета. Затем, с шагом `Phase_Step_Precise` происходит смещение как в сторону меньшей, так и в сторону большей фазы, на число шагов, равное 10(5). В каждой точке скорость счета определяется за время `FreqC_Precise_Gate`.

```

1 [ Test02_Strobe_Phase]
2 FS = 100
3 TS = 2
4 Laser_WaveLength = 1550
5 P_units = 1
6 P_set = 3,6E-9
7 Att = 50
8 Phase_Quick_Start = 0
9 Phase_Quick_Stop = 300
10 Phase_Quick_Step = 10
11 FreqC_Quick_Gate = 0,2
12 Phase_Step_Precise = 1
13 FreqC_Precise_Gate = 1

```

3. Измерение сетки. После завершения тестов, можно перейти на следующее окно, в котором необходимо задать параметры последующих измерений. В числе данных параметров следующие: начальная и конечная температура, шаг измерения. В случае, если исследуется стробируемый детектор, то можно выставить начальный и конечный код строба, шаг измерения. Для запуска измерений необходимо однократно нажать на соответствующую кнопку. После этого устанавливается заданная стартовая температура и начинаются измерения. Первым этапом в непосредственных измерениях является измерение сетки. Параметр `HV_Grid_Start` устанавливает начальное значение напряжения смещения, параметр `HV_Grid_Stop` устанавливает максимальное напряжение смещения, выше которого напряжение устанавливаться не будет. Если напряжение смещения преодолело максимально доступное, и при этом скорость темнового счета не стала больше параметра `HV_Grid_Pulse_Count_max`, то строка с данным кодом строба в файле "Grid_HV_bias.csv" заполняется величинами `NULL`. Начальный шаг по напряжению смещения устанавливается `HV_Grid_Initial_Step`, каждое измерение проводится `HV_Grid_FreqC_Num_Of_Meas` раз, каждое в течение `HV_Grid_FreqC_Gate_Time`. Как только грубым шагом удалось дойти до такого режима работы детектора, при котором темновой счет больше `HV_Grid_Pulse_Count_min`, включается режим с точным шагом. Данную точку напряжения назовем точкой разветвления. Сначала осуществляется поиск напряжения, при котором значение темнового счета максимально приближено к `HV_Grid_Pulse_Count_min`. Таким образом, одна ветвь идет в сторону уменьшения напряжения. Новый шаг равен грубому шагу деленному пополам,

и напряжение уменьшается на значение нового шага, вплоть до момента, пока скорость счета не станет меньше HV_Grid_Pulse_Count_min. После этого значение шага снова уменьшается в два раза. После чего напряжение начинает увеличиваться на величину нового шага, пока не станет больше HV_Grid_Pulse_Count_max. Затем шаг снова уменьшается и алгоритм повторяется, пока значение шага не станет равным HV_Grid_Initial_Step / 32. Затем, с точки разветвления начинается поиск верхней границы напряжения по аналогичному алгоритму, только теперь приближение скорости счета производится до величины HV_Grid_Pulse_Count_max. Таким образом, определяются две точки напряжения – верхнее и нижнее. Если разница между данными значениями меньше HV_Grid_Minimum_delta_V, то считается, что доминирует эффект захвата зарядов (charge persistence), в результате чего такой режим работы детектора не может являться оптимальным, и его измерение, и измерение всех больших кодов строба не имеет смысла. В массив сетки на строки соответствующих кодов строба заносится значение NULL, и алгоритм измерения переходит к следующему шагу. В случае, если разница между напряжениями больше величины HV_Grid_Minimum_delta_V, то формируется одномерная линейная сетка напряжений для данного кода строба, с числом точек, равное HV_Grid_HV_bias_Number_Of_Intervals. Аналогичный алгоритм выполняется для следующего кода строба. Таким образом, для детектора стробированного режима для выбранной температуры, сетка напряжений представляется в виде двумерного массива. Для детектора типа фриран – в виде одномерного массива.

```

1 [Test03_Measurments_Grid]
2 HV_Grid_FS = 100
3 HV_Grid_TS = 2
4 HV_Grid_Start = 50
5 HV_Grid_Stop = 80
6 HV_Grid_Initial_Step = 1
7 HV_Grid_FreqC_Gate_Time = 0,1
8 HV_Grid_FreqC_Num_Of_Meas = 10
9 HV_Grid_Pulse_Count_min = 30
10 HV_Grid_Pulse_Count_max = 2000
11 HV_Grid_Minimum_delta_V = 0,4
12 HV_Grid_HV_bias_Number_Of_Intervals = 10

```

4. Измерение DCR. Для каждой точки сетки напряжений и кодов строба выполняется измерение темнового счета – всего DCR_FreqC_Num_Of_Meas измерений, каждое в течение DCR_FreqC_Gate_Time. Данный подход необходим для вычисления значения среднеквадратического отклонения. В результате чего создается два файла – "DCR.csv" и "DCRSigma.csv".

```

1 [Test04_Measurments_DCR]
2 DCR_FS = 100
3 DCR_TS = 2
4 DCR_FreqC_Gate_Time = 0,1
5 DCR_FreqC_Num_Of_Meas = 30

```

5. Измерение скорости счета со включенным излучением и временного разрешения. Для каждой точки сетки напряжений и кодов строба выполняется алгоритм настройки фазы, аналогичный описанному в пункте 2 – тестирование настройки фазы. Только в данных измерениях затухание на втором аттенюаторе устанавливается равным

`QE_ATT3`, чтобы среднее число фотонов на импульс равнялось 0.1. Измерения в процессе поиска фазы, и непосредственно измерение после настройки фазы, выполняется `QE_FreqC_Num_Of_Meas` раз, по `QE_FreqC_Gate_Time` секунд каждое. Это позволяет определить среднеквадратическое отклонение в скорости счета. В результате чего создается два файла – "QE.csv" и "QESigma.csv".

Далее, после проведения измерения скорости счета, проводится измерение временного разрешения с установленной настройкой фазы, в случае, если индекс точки в составленной сетке напряжений при заданном коде строба больше или равно `TR_Start_Meas_index`. Это обусловлено тем, что в первых точках сетки *PDE* достаточно мало, и статистика на гистограмме собирается слишком долго. Для измерения статистики используется осциллограф, на котором предварительно загружается настройка с именем `TR_Scope_Recall_File`. Количество собранных измерений на осциллографе задается величиной `TR_Histo_Size`. Данный алгоритм проводится для каждой точки сетки.

```

1 [ Test05_Measurments_QE ]
2 QE_FreqC_Gate_Time = 0,1
3 QE_FreqC_Num_Of_Meas = 30
4 QE_ATT3 = 60
5 TR_Start_Meas_index = 5
6 TR_Scope_Recall_File = spd-jitter-setup--00003.ls
7 TR_Histo_Size = 10000

```

6. Измерение послеимпульса. Режим работы лазера переключается на частоту следования импульсов `AP_FS` кГц. Затухание на постоянном аттенюаторе устанавливается `AP_ATT3`, чтобы количество фотонов на импульс составляло 1, с целью увеличения скорости сбора статистики. На осциллографе загружается файл настройки `AP_Scope_Recall_File`. Следующий алгоритм выполняется только для двух точек сетки по каждому коду строба – самой правой и центральной. Для каждой выбранной точки сначала устанавливается фаза, по описанному в пункте 2 алгоритму. При поиске фазы выполняется `AP_FreqC_Num_Of_Meas` измерений по `AP_FreqC_Gate_Time` секунд, в каждой точке фазы. После чего запускается сбор статистики. Остановка происходит по достижению количества собранных срабатываний `AP_Histo_Size`.

```

1 [ Test06_Measurments_AP ]
2 AP_FS = 10
3 AP_TS = 2
4 AP_ATT3 = 60
5 AP_FreqC_Gate_Time = 0,1
6 AP_FreqC_Num_Of_Meas = 10
7 AP_Scope_Recall_File = spd-ap-setup-100khz--00000.ls
8 AP_Histo_Size = 10000

```

7. Измерение временного разрешения на высокой частоте. Измерения производятся для каждого кода строба, для точек сетки, чей индекс больше `TR_bif_Start_Meas_index`. Это обусловлено тем, что при меньших индексах малая вероятность детектирования фотона, и статистика собирается в течение нерелевантно большого времени. Режим работы лазера переключается на частоту `TR_bif_FS`, затухание на втором аттенюаторе `TR_bif_Att3`. В данном случае тоже необходимо выполнить алгоритм настройки фазы, описанный в пункте 2. Для этого контролируется выходная мощность на первом

аттенюаторе TR_bif_P_set_phase_tune. Измерение на каждой точке фазы выполняется TR_bif_FreqC_Num_Of_Meas раз, каждое в течение TR_bif_FreqC_Gate_Time. После настройки фазы контролируемая мощность на выходе первого аттенюатора устанавливается TR_bif_P_set_meas Вт. На осциллографе загружается настройка с именем TR_bif_Scope_Recall_File. Начинается сбор статистики, который заканчивается по достижению количества точек на гистограмме TR_bif_Histo_Size.

```

1 [ Test07_Measurments_TR_bifurcation ]
2 TR_bif_FS = 156250
3 TR_bif_TS = 2
4 TR_bif_Att3 = 60
5 TR_bif_P_set_meas = 5,6E-6
6 TR_bif_P_set_phase_tune = 500E-9
7 TR_bif_FreqC_Gate_Time = 0,1
8 TR_bif_FreqC_Num_Of_Meas = 10
9 TR_bif_Scope_Recall_File = spd-jitter-setup--00003.lss
10 TR_bif_Histo_Size = 100000
11 TR_bif_Start_Meas_index = 5

```

2.3 Файл SPDConfig-freerun

Файл конфигурации для детектора типа фриран имеет ту же самую структуру, что и файл конфигурации для стробированного режима. Он обладает полностью идентичными полями (за исключением шага 7, на котором измеряется временное разрешение на высокой частоте), однако некоторые поля нечитываются программой и не участвуют в алгоритме измерений. Для данного типа детектора не нужно использовать алгоритм настройки фазы.

0. Тест отсутствия импульсов. Выполняется аналогично стробированному режиму.
1. Тест наличия срабатываний детектора. Выполняется аналогично стробированному режиму.
2. Тест настройки фазы. Отключен при тестировании детектора типа фриран.
3. Измерение сетки. Отличие от стробированного режима заключается в том, что в данном случае отключено изменение кода строба. Для построения всех массивов считает, что измерения проводятся с единственным кодом строба, равным 0. Таким образом, алгоритм настройки повторяется, как будто для единственного строба.
4. Измерение *DCR*. Алгоритм проводится аналогично алгоритму для стробированного режима, только для одного кода строба.
5. Измерение скорости счета со включенным излучением и временного разрешения. Алгоритм проведения аналогичен алгоритму для стробированного детектора, только для одного кода строба, и с отключенной функцией подстройки фазы.
6. Измерение послеимпульса. Проводится аналогично алгоритму для стробированного детектора, только для одного кода строба, и с отключенной функцией подстройки фазы.

3 Описание программы обработки

Репозиторий программы обработки доступен по ссылке Автоматизированный Измерительный Стенд.

Данный репозиторий обладает следующей структурой:

- Папка "!SPDStandResults" содержит папки с измерениями для различных детекторов. Папка с именем детектора содержит папки с датой проведенных измерений. Папка с датой измерения содержит папки "report", папки формата "T_-*", файлы "ErrorLog.txt" с логом возникающих при измерениях ошибок, "SPDConfig_.txt" являющийся копией файла настройки измерений, "Grid_Temperature.csv" с измеряемой сеткой по температуре, и в случае измерения стробируемого детектора файлы "Grid_Stb_Code.csv" с измеряемой сеткой по кодам строба, "gate_voltage.xlsx" с таблицей перевода кода строба в амплитуду строба для конкретного детектора.

Папка формата "T_-*" содержит папки с кодом строба в шестнадцатиричном формате типа "0x*", файлы "Grid_HV_bias.csv" с двумерной сеткой напряжений смещений, в котором каждая строка соответствует конкретному значению кода строба, "DCR.csv" с измеренным темновым счетом в точках сетки, определенных файлом "Grid_HV_bias.csv" (далее в точках сетки напряжений); "DCRSigma.csv", "QE.csv", "QESigma.csv" со среднеквадратическим отклонением (СКО) для скорости темнового счета, скорости счета при включенном излучении и СКО для скорости счета при включенном излучении соответственно.

Папка формата "0x*" содержит три типа файлов в формате "AP_*", "TR_*" и "TR_Bif_*", в названии которого содержится значение напряжения смещения, при котором данное измерение было проведено. В файле "AP_*" представлена статистика с осциллографа для определения постимпульса, в файле "TR_*" представлена статистика с осциллографа для определения временного разрешения (джиттера) при частоте лазерных импульсов 100 кГц, в файле "TR_Bif_*" представлена статистика с осциллографа для определения временного разрешения и забросов срабатываний детектора в строб, в который не приходит лазерное излучение, при частоте лазерных импульсов 156.25 МГц.

Папка "report" содержит папки "QE" и "SNR", в которых в качестве оптимальных параметров представлены параметры, полученные при фиксированном *PDE* с минимальным *DCR*, и при минимальном соотношении сигнал-шум *SNR* соответственно. Каждая папка содержит графики зависимости различных параметров детектора, а также сформированные отчеты в формате pdf. Отчет, имеющий формат "*_CD.pdf" содержит графики, в которых используется код строба, а отчет формата "*_VG.pdf" содержит графики, в которых используется амплитуда строба.

- Файл "afterpulse.py" содержит функцию Ap_analyze(), которая определяет мертвое время *DT* и вероятность постимпульса *AP* по файлам формата "AP_*".
- Файл "design5.py" содержит представление GUI дизайна "design5.ui", переведенное в код python. Данная конвертация может быть выполнена командой

`$ pyuic5 path/to/design.ui -o output/path/to/design.py` в консоли. Более подробно это описано в статье про python GUI. В файле "design5.py" определяются такие объекты, как label, line, combobox, slider, button, Widget (окна для отрисовки графиков), задаются координаты их размещения.

- Файл "generate_pdf.py" отвечает за генерацию отчетов в формате pdf. В нем используется библиотека fpdf. Кастомизация колонтитулов происходит благодаря переопределению некоторых функций класса `class CustomPDF(FPDF)`. Функция `create_pdf()` размещает на странице pdf уже сгенерированные изображения, в зависимости от типа детектора и используемого критерия. Функция `simple_table()` отвечает за представление таблиц.
- Файл "graphics_generation.py" отвечает за генерацию различных графиков и их последующее сохранение. Функция `get_sub_slice(qe_sat)` возвращает массивы срезы различных физических величин, полученные для заданного значения `qe_sat`. Функция `get_slices()` формирует набор массивов-срезов для различных параметров `qe_sat`, благодаря вызову функции `get_sub_slice(qe_sat)`. Функция `plot_dcr_qe_fr()` генерирует график $DCR(V_b)$ и $PDE(V_b)$ для ДОФ типа фриран (freerun). Функция `plot_dcr()` генерирует график $DCR(V_g)$ для стробированного (gated) ДОФ. Функция `plot_ap_fr()` генерирует график $AP(V_b)$ для ДОФ типа фриран. Функция `plot_ap()` генерирует график $AP(V_g)$ для стробированного ДОФ. Функция `plot_snr_fr()` генерирует график $SNR(V_b)$ для ДОФ типа фриран. Функция `choose_optimum_fr()` осуществляет поиск оптимальных параметров для ДОФ типа фриран по критерию минимального SNR (`criterion = 'SNR'`) и по критерию минимального DCR при заданном PDE (`criterion = 'QE'`). Функция `TR_filtration(cnt)` производит сглаживание джиттера `cnt` с помощью низкочастотного фильтра. Функция `choose_optimum()` осуществляет поиск оптимальных значения для стробированного детектора по критерию минимального SNR и по критерию минимального DCR при заданном PDE . Функция `find_bif(hist)` осуществляет обработку гистограммы из файла "TR_Bif_*", благодаря чему определяется относительное количество забросов кликов детектора в соседний строб. Функция `plot_TR()` генерирует график временного разрешения $TR(t)$ для ДОФ двух типов и для различных критериев оптимальности. Функция `plot_heatmap_bif()` генерирует тепловую карту для зависимости $BIF(V_g, V_b)$. Функция `plot_heatmap()` генерирует тепловую карту для зависимости $SNR(V_g, V_b)$. Функция `generate_pdf()` вызывает описанные функции генерации графиков, описанные выше, а также вызывает функцию `create_pdf()` из файла "generate_pdf.py".
- Файл "GUI5.py" отвечает за взаимодействие в GUI, а именно класс `class ProgramGUI()`, который имеет наследование от классов `QtWidgets.QMainWindow` и `design5.Ui_MainWindow`. Функция `init_ui()` определяет надписи в лэйблах, и описывает связи событий на различных интерактивных элементах с созданными функциями. Функция `create_report()` вызывает функцию `generate_pdf()` из файла "graphics_generation.py". Функция `directory_view()` реализует просмотр папок с детекторами и их отображение

в комбобоксе. Функция `date_view(index)` реализует просмотр папок с датами для выбранного имени детектора. Функция `download_spd()` реализует создание пути к экспериментальным данным, на основе выбранных имен в комбобоксах и вызывает функцию `download_files()`. Функция `download_files()` загружает обработанные функцией `SPD()` (из файла "spd_analyze.py") данные по детектору в объект `SPD`. Функция `shift_slider2_newplot()` вызывается при взаимодействии с одним из слайдеров и вызывает перерисовку графических объектов. Функция `shift_slider2()` вызывается при взаимодействии с одним из слайдеров, вычисляет новые массивы-срезы, которые необходимо отобразить. Функция `shift_slider()` работает аналогично функции `shift_slider2()`. Функции `shift_slider()` и `shift_slider2()` работают в паре, для оптимизации количества перерисовываемых объектов. Функция `fill_labels()` заполняет лэйблы в GUI в соответствии с выбранной на графиках точкой. Функция `plot2d_click()` Реализует отрисовку 2-d графиков в соответствии с выбранным `plot_setup`. Функция `plot3d_click()` реализует отрисовку 3-d графиков в соответствии с выбранным `plot_setup`. В функции `main()` инициализируется GUI.

- Файл "module_fwhm.py" содержит функции `fwhm(array)` и `fwtm(array)`, которые вычисляют полную ширину на половине высоты и на одной десятой высоты от массива соответственно.
- Файл "module_pde.py" содержит функцию `pde()`, которая вычисляет вероятность детектирования фотона, исходя из собранных статистических данных.
- Файл "spd_analyze.py" содержит класс для работы с параметрами ДОФ. В инициализации происходит загрузка массива температуры, а по файловой структуре определяется принадлежность устройства к классу фриран или стробированному. Функция `func_exp()` определена для аппроксимации полученных параметров по послеимпульсу из двух точек, на все точки сетки напряжений. Функция `load_freerun()` реализует загрузку всех данных для детектора типа фриран. Вначале для каждой температуры из папки "0x000" загружаются гистограммы для послеимпульса, по которым определяются параметры вероятности послеимпульса AP и мертвое время DT . В том же цикле загружаются гистограммы для временного разрешения. Далее собранные данные послеимпульса в двух точках аппроксимируются на все точки сетки. Далее загружаются параметры DCR , σ_{dcr} , $QE(== R)$. Из данных параметров вычисляется значение PDE . Функция `load_gated()` реализует загрузку данных для детектора стробированного режима. Вначале анализируются названия папок, исходя из которых определяются принадлежность находящихся внутри данных конкретному коду строба. Далее для каждой температуры загружается сетка напряжений. Далее для каждой температуры и каждой амплитуды строба загружаются гистограммы для послеимпульса и временного разрешения, и анализируются аналогично функции `load_freerun()`. Дальнейшая обработка данных совпадает с обработкой данных для детектора типа фриран, только теперь вместо одного кода строба анализируется некоторое их количество, таким образом добавляя дополнительный цикл.

Функция `plot_section(choosed_T, choosed_graph)` используется для отображения конкретно выбранного типа графика (`choosed_graph`) и для конкретной температуры (`choosed_T`). В текущей версии проекта не используется и применяется для отладки. Функция `slice_return(plot_index, tempind, ind)` используется для возвращения среза массивов для определенного типа запрашиваемого графика (`plot_index`), для определенной температуры (`tempind`) и для определенного кода строба (`ind`). Функция `timeres_return(tempind, gateind, Vb)` возвращает гистограмму для временного разрешения, если она существует, для конкретной температуры `tempind`, когда строба `gateind` и напряжения смещения `Vb`.

- Файл "`test_graphic_generation.py`" используется для тестирования системы генерации отчетов, позволяя это делать не через графический интерфейс, а непосредственно с помощью запуска скрипта.
- Файл "`threedsurfacegraphwindow.py`" содержит класс `ThreeDSurfaceGraphWindow()` для представления трехмерных графиков на одном из виджетов в GUI. Функция инициализации определяет объект, который будет передан в виджет – трехмерное полотно. Функция `draw_graph()` реализует отрисовку непосредственно трехмерного графика. Функция `clear_graphics()` очищает используемое полотно от всех объектов. Функция `plot_slice()` реализует отрисовку двумерной линии на трехмерном графике, позволяющую ориентироваться в выбранном коде строба. Функция `plot_point()` реализует отрисовку точки, параметры которой в настоящий момент отображаются в лэйблах в левой части GUI.
- Файл "`twodgraphicwindow.py`" содержит класс `TwoDGraphicWindow()` для представления двумерных объектов на одном из виджетов в GUI. Функция инициализации определяет объект, который будет передан в виджет – двумерное полотно. Функция `draw_graph()` отрисовывает двумерный график. Проверка на размер массива `slice` реализован для определения типа графика, который передается в функцию. Функция `plot_slice()` отрисовывает вертикальную пунктирную линию, которая показывает рассматриваемую точку, параметры которой в настоящий момент отображаются в лэйблах в левой части GUI.
- Файл "`time_resolution.py`" содержит функцию `time_resolution(filename)`, которая открывает файл гистограммы временного разрешения и переводит его в массив.
- Файл "`start_SPD.bat`" является исполняемым файлом и содержит команду для быстрого запуска файла "`GUI5.py`", содержащий GUI.

На рисунке 4 представлена принципиальная схема работы программы для обработки данных, и взаимодействие файлов друг с другом.

На рисунке 5 представлен пример работы программы для обработки данных. Запуск возможен двойным нажатием на файл "`start_SPD.bat`" или путем запуска скрипта в файле "`GUI5.py`". На данном рисунке красными прямоугольниками и нумерацией обозначены объекты, функции которых будут описаны ниже.

Функции программы для обработки данных:

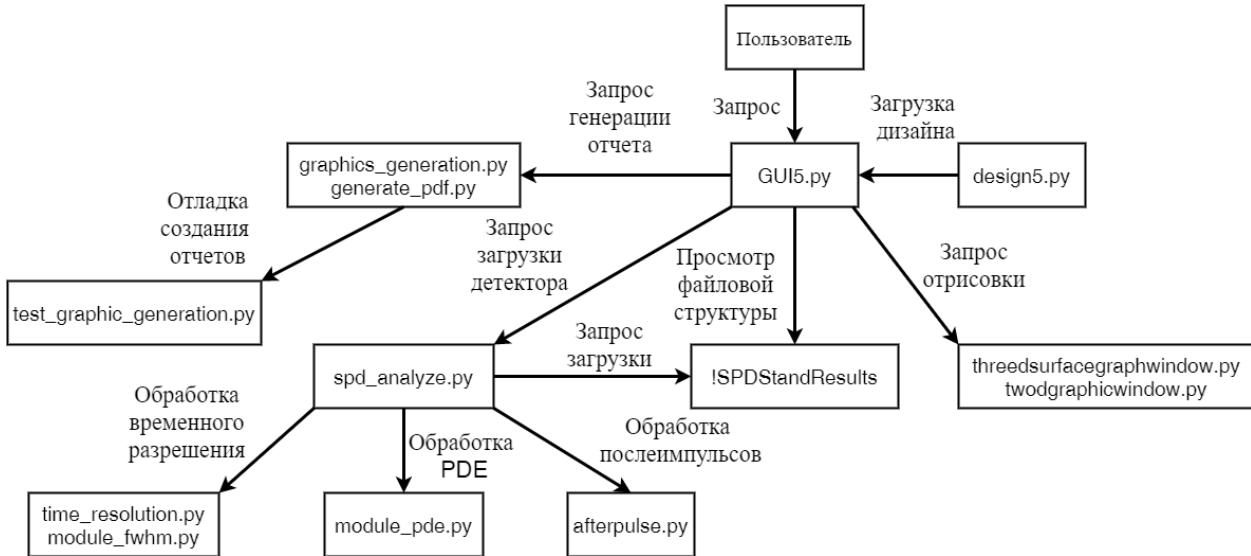


Рис. 4: Принципиальная схема работы программы обработки данных.

1. SPD name. Выбор модели детектора. Каждый детектор носит свое имя, а не просто номер. На примере детектора "Тимофей".
2. Measurement date. Выбор даты измерения детектора. Измерения, проведенные в разное время, могут иметь разные результаты.
3. Temperature. Выбор температуры, для которых будут отображаться все графики.
4. Plot setup. Выбор отображаемой характеристики на графиках в 3-Д окне (10) и в 2-Д окне (11).
5. V_g . Выбор среза по коду строба – выбор кода строба. Показан черной кривой на 3-Д графике. Именно при данной величине кода строба строится 2-Д график в окне (11).
6. V_b . Выбор точки в срезе по коду строба – напряжения смещения. Показан фиолетовой точкой на 3-Д графике и фиолетовой вертикальной пунктирной линией на 2-Д графике. Характеристики детектора в данной точке представлены в окне (8).
7. SPD type. Показывает тип исследуемого детектора. Либо freerun (фриран), либо gated (стробируемый).
8. Efficiency and Technical parameters. Данные параметры определяются в указанной фиолетовым точке на 3-Д графике. Параметры эффективности имеют эксплуатационную значимость для последующей интеграции устройства в систему квантового распределения ключа (КРК). Технические параметры нужны для настройки режима работы детектора.
9. Create report. Нажатие на кнопку формирует отчет по выбранной модели детектора и дате его измерения. Файл отчета сохраняется в папку `report` в корне папки с данными детектора.

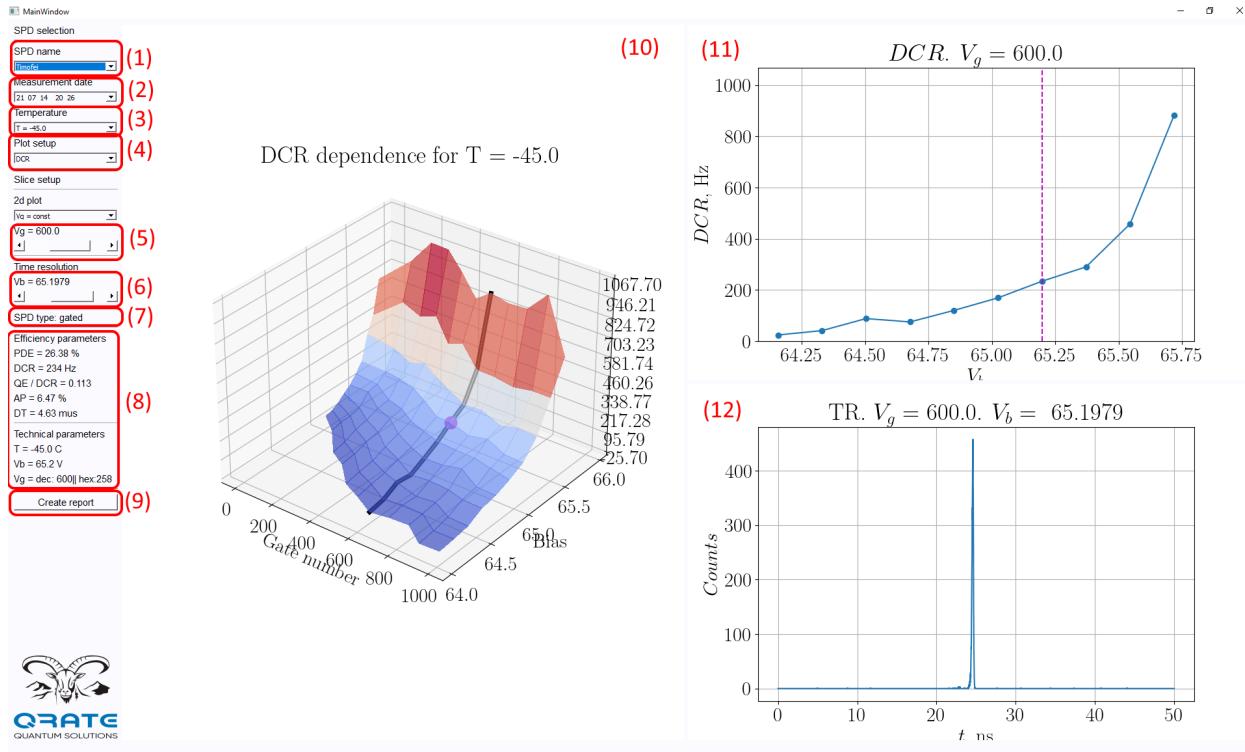


Рис. 5: Пример окна приложения для обработки данных.

10. Окно 3-Д графика. На нем выводится тип графика, указанный в поле (4).
11. Окно 2-Д графика. На нем выводится тип графика, указанный в поле (4), соответствующий срезу в поле (5).
12. Окно 2-Д графика. На нем выводится график временного разрешения в точке, соответствующий фиолетовой точке на 3-Д графике.

Для ДОФ типа фриран 3-Д график отсутствует в связи с отсутствием кода строба как варьируемого параметра. Таким образом, можно изменять только напряжение смещения в окне (6).

Список литературы

- [1] Ссылка на гитхаб с проектом: https://github.com/akoziy98/Automated_Stand
- [2] Ссылка на ресурс по основам работы в PyQt: <https://tproger.ru/translations/python-gui-pyqt/>