

Описание программных компонентов

Описание программы загрузки градиентной импульсной последовательности	2
Описание тестирования программы загрузки градиентной импульсной последовательности	3
Описание программы загрузки синхронизирующей последовательности	6
Описание программы загрузки радиочастотной последовательности для Hask RF	8
Описание тестирования программы загрузки синхронизирующей последовательности	10
Описание программы задания параметров Picoscope	15
Описание тестирования программы задания параметров Picoscope	15

Описание программы загрузки градиентной импульсной последовательности

Программа загрузки импульсной последовательности в градиентный усилитель основана на библиотеке `clsocket` (язык C++, Ссылка на репозиторий <https://github.com/DFHack/clsocket>) и предназначена для отправки и приема UDP пакетов между управляющим устройством и платой градиентного усилителя.

Программа загрузки импульсной последовательности содержит инициализацию активного сокета, а также следующие функциональные компоненты:

- Запрос версии API (функция `get_API_version`).
- Запрос версии прошивки (функция `get_sw_revision`).
- Запрос состояния градиентного усилителя (функция `get_gru_state`).
- Чтение точек перегиба траектории из txt файла (функция `get nodes`). TXT файл состоит из двух столбцов: первый из которых представляет собой количество временных отсчетов по 10 мкс относительно начала импульсной последовательности, второй столбец – амплитуда сигнала на выходе градиентного усилителя в диапазоне от [-32 768; 32 767].
- Загрузка точек траектории (функция `upload_traj`). Поскольку существует ограничение на максимально возможную длину UDP пакета, то целая траектория разделяется на сегменты по 200 точек (см. Протокол обмена с ВУ v.03).
- Выгрузка точек траектории (функция `download_traject`) для получения точек траектории, которые были интерполированы градиентным усилителем.
- Получение участков траектории, которые не были загружены (функция `get unloaded num`)
- Закрытие сокета (функция `close socket`)

Описание тестирования программы загрузки градиентной импульсной последовательности

Для тестирования необходимо подключить плату градиентного усилителя к управляющему устройству. Для подключения питания платы градиентного усилителя используется USB кабель, для передачи данных используется Ethernet. При подключении платы необходимо проверить, что загорелись светодиоды питания и сети Ethernet на плате. Для начала тестирования необходимо выставить настройки сети Ethernet управляющего устройства: IP: 192.168.100.2, маска сети 255.255.255.0, а также в свойствах Ethernet подключения необходимо отключить все подключения кроме IPV4 и Ncap драйвера. Пример представлен на рисунке 1.

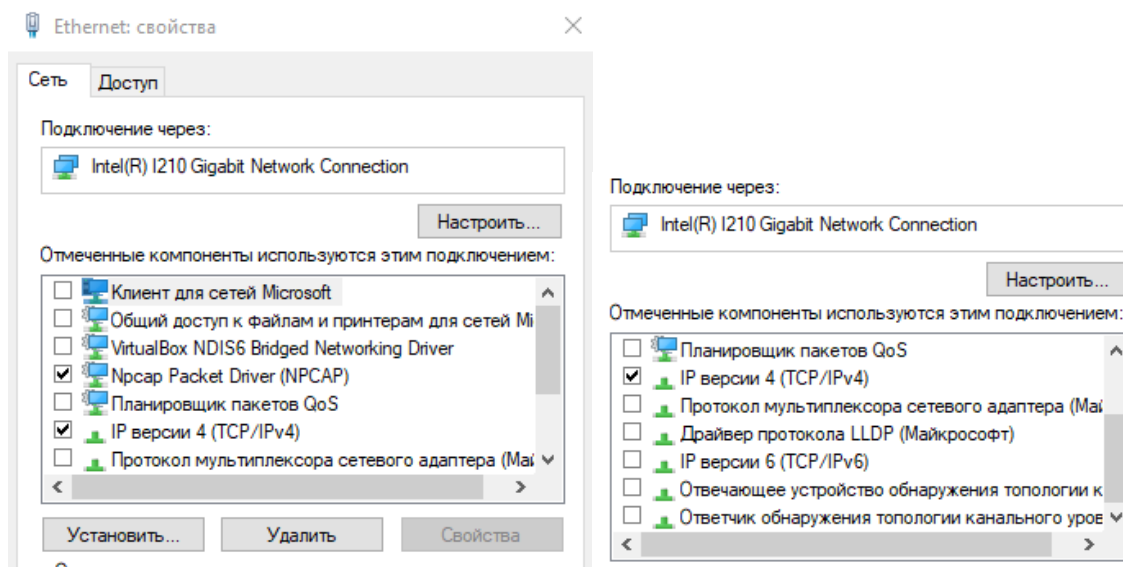


Рисунок 1 – Установка параметров сети Ethernet

После выставления параметров сети Ethernet необходимо убедиться в получении беззапросных пакетов данных от платы градиентного усилителя, для этого необходимо запустить программу Wireshark от имени администратора, выбрать Ethernet протокол, для выделения значимых пакетов данных рекомендуется применить фильтр:

```
arp || icmp || udp.port == 5001 || udp.port == 5002 && !(icmp >= "Destination unreachable")
```

При успешном подключении можно наблюдать отправку беззапросных пакетов данных с порта 5002 на порт 6001 с интервалом в одну секунду. Пример успешного подключения представлен на рисунке 2. Если после подключения не наблюдаются беззапросные пакеты данных необходимо перезагрузить плату градиентного усилителя с помощью отключения и подключения USB кабеля, также необходимо проверить параметры сети.

arp icmp udp.port == 5001 udp.port == 5002 && icmp >= "Destination unreachable"						
No.	Time	Source	Destination	Protocol	Length	Info
963	43.116593	192.168.100.3	192.168.100.2	UDP	246	5002 → 6001 Len=204
964	43.116611	192.168.100.2	192.168.100.3	ICMP	274	Destination unreachable (Port unreachable)
965	43.872737	GigaByteTech_e0:7f:: Broadcast	ARP	42	Who has 192.168.100.2? (ARP Probe)	
975	44.116713	192.168.100.3	192.168.100.2	UDP	246	5002 → 6001 Len=204
976	44.116770	192.168.100.2	192.168.100.3	ICMP	274	Destination unreachable (Port unreachable)
979	44.868228	GigaByteTech_e0:7f:: Broadcast	ARP	42	ARP Announcement for 192.168.100.2	
994	45.117733	192.168.100.3	192.168.100.2	UDP	246	5002 → 6001 Len=204
995	45.117801	192.168.100.2	192.168.100.3	ICMP	274	Destination unreachable (Port unreachable)
1010	46.118598	192.168.100.3	192.168.100.2	UDP	246	5002 → 6001 Len=204
1011	46.118658	192.168.100.2	192.168.100.3	ICMP	274	Destination unreachable (Port unreachable)
1015	46.870976	GigaByteTech_e0:7f:: Broadcast	ARP	42	ARP Announcement for 192.168.100.2	
1025	47.119575	192.168.100.3	192.168.100.2	UDP	246	5002 → 6001 Len=204
1026	47.119681	192.168.100.2	192.168.100.3	ICMP	274	Destination unreachable (Port unreachable)
1030	47.868414	GigaByteTech_e0:7f:: NS-NUB-PhysServer-3.. ARP	42	Who has 192.168.100.3? Tell 192.168.100.2		
1031	47.869378	NS-NUB-PhysServer-3.. GigaByteTech_e0:7f:: ARP	60	192.168.100.3 is at 02:23:45:67:89:ab		
1041	48.120604	192.168.100.3	192.168.100.2	UDP	246	5002 → 6001 Len=204
1042	48.120640	192.168.100.2	192.168.100.3	ICMP	274	Destination unreachable (Port unreachable)
1059	49.121576	192.168.100.3	192.168.100.2	UDP	246	5002 → 6001 Len=204
1060	49.121640	192.168.100.2	192.168.100.3	ICMP	274	Destination unreachable (Port unreachable)
1076	50.122460	192.168.100.3	192.168.100.2	UDP	246	5002 → 6001 Len=204
1077	50.122465	192.168.100.2	192.168.100.3	ICMP	274	Destination unreachable (Port unreachable)
1091	51.123434	192.168.100.3	192.168.100.2	UDP	246	5002 → 6001 Len=204
1092	51.123460	192.168.100.2	192.168.100.3	ICMP	274	Destination unreachable (Port unreachable)
1104	52.124409	192.168.100.3	192.168.100.2	UDP	246	5002 → 6001 Len=204
1105	52.124439	192.168.100.2	192.168.100.3	ICMP	274	Destination unreachable (Port unreachable)
.....						
> Frame 2: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{49A5F2F6-8C1D-4D59-9F63-FD48972F48C7}, 1						
> Ethernet II, Src: Xerox_7a:c5:b5 (9c:93:4e:7a:c5:b5), Dst: Broadcast (ff:ff:ff:ff:ff:ff)						
> Address Resolution Protocol (request)						
				0000	ff ff ff ff ff ff 9c 93 4e 7a c5 b5 00 00 01 N2:.....
				0010	00 00 00 00 00 01 9c 93 4e 7a c5 b5 ac 10 16 b0 N2:.....
				0020	00 00 00 00 00 00 ac 10 16 dc 00 00 00 00 00
				0030	00 00 00 00 00 00 00 00 00 00 00 00 00 00

Рисунок 2 – Пример успешного подключения платы градиентного усилителя

Для проверки обмена простыми пакетами UDP между управляющим устройством и платой градиентного усилителя необходимо запустить проект simple_requests.cbp с помощью программы Codeblocks. Данная программа состоит из запросов версии прошивки градиентного усилителя, запроса состояния градиентного усилителя, а также запроса версии API и проверки ответа платы градиентного усилителя на корректность полученных пакетов. После запуска программы simple_requests открывается окно консоли, пример консоли успешно отработавшей программы представлен на рисунке 3.

```
SOCKET INITIALIZE AND OPEN

GET API VERSION
API VERSION OK

GET SW REVISION
SW REVISION VERSION OK

GET GRU STATE
GRU STATE OK

SOCKET CLOSE
closed

Process returned 1 (0x1)   execution time : 0.010 s
Press any key to continue.
|
```

Рисунок 3 – Изображение консоли при успешном обмене простыми пакетами данных

После проверки корректности простых пакетов данных необходимо протестировать загрузку/выгрузку траектории; для этого необходимо запустить проект GRU_traject.cbp. Данный пример Пример успешного прохождения проверок загрузки траектории представлен на рисунке 4.

Описание программы загрузки синхронизирующей последовательности

Программа загрузки синхронизирующей последовательности основана на библиотеке `Due pulse programmer` (ссылка на ресурс <https://phas.ubc.ca/~michal/DuePulseProgrammer/>). Данная библиотека обеспечивает задание синхронизирующей импульсной последовательности с шагом 20 нс. Импульсная программа состоит из односимвольной команды “D”, после которой двумя байтами указывается старшая часть количества событий, затем двумя байтами указывается младшая часть количества событий и далее передаются сами события, подробное описание представлено в файле `Synchronisation\due-pp-docs.pdf`. События, представленные в библиотеке, можно разделить на:

- События выполняющиеся безусловно (простые события).
- События, выполняющиеся с приходом внешнего управляющего сигнала **на цифровой пин 22 Arduino Due.**

Для инициализации синхронизирующей программы необходимо выполнить команду `due_init_program`, в качестве аргументов необходимо передать ссылку на предварительно созданную синхронизирующую программу, имеющую тип `due_prog_t`. Для добавления в синхронизирующую программу простого события необходимо выполнить команду `due_prog_t(&program1, outputs1, ticks1)`, в качестве параметров необходимо передать ссылку на импульсную программу, количество тактов по 20нс, в течение которых необходимо поддерживать определенное состояние пинов, далее необходимо передать непосредственно состояние пинов (например `0x00FFFFFF`).

Для добавления в синхронизирующую программу события, выполняющегося с приходом внешнего управляющего сигнала необходимо сначала добавить как минимум одно простое событие в синхронизирующую последовательность и далее добавить событие по триггеру с помощью

команды `due_wait_for_trigger`, формат входных параметров совпадает с параметрами для обычного события.

После загрузки всех событий необходимо загрузить синхронизирующую последовательность на плату, для этого используется функция `due_upload_trajectory(serial, &program1)`, в параметры которой необходимо передать объект типа `serial`, представляющий собой последовательный порт, к которому подключена плата, также в параметры функции передается созданная синхронизирующая программа.

Загруженная импульсная последовательность может быть запущена с помощью отправки символа “e” по последовательному порту. При перезагрузке платы синхронизирующая программа стирается и ее необходимо снова загружать.

Описание программы загрузки радиочастотной последовательности для Hack RF

Программа загрузки радиочастотной последовательности для Hack RF основана на утилите Hack RF transfer (ссылка на репозиторий https://github.com/greatscottgadgets/hackrf/blob/master/host/hackrf-tools/src/hackrf_transfer.c), адаптированную под язык программирования C++. Перед запуском программы необходимо убедиться в установленном драйвере WINUSB для коммуникации с Hack RF. На рисунке 4 представлен вид приложения Zadig при правильной установке драйвера. При необходимости необходимо установить драйвер WINUSB с помощью приложения Zadig.

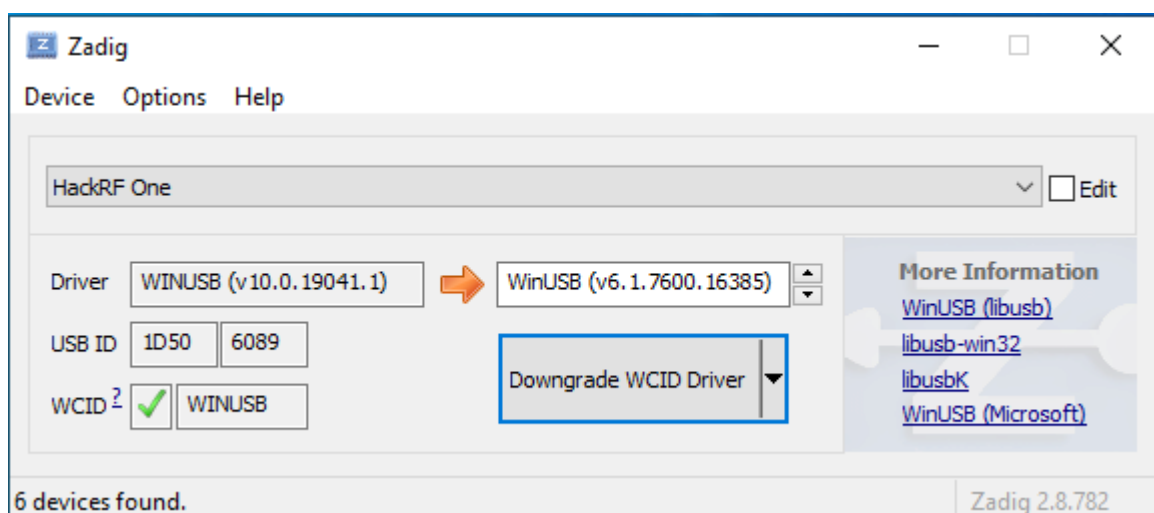


Рисунок 4 – Установка драйвера для Hack RF

Для проверки правильной инициализации Hack RF можно воспользоваться утилитой `hackrf info`. Для запуска скомпилированной программы загрузки радиочастотной последовательности необходимо с помощью командной строки перейти в каталог с файлом `exe` и указать название `exe` файла. После названия `exe` файла необходимо указать параметры (источник https://manpages.debian.org/unstable/hackrf/hackrf_transfer.1.en.html) через пробел:

- **-r <filename> # Сохранить данные в файл.**
- **-t <filename> # Передать данные из файла.**

- -w # получить данные в файл с WAV заголовком и автоматическим названием.
- -f <set_freq_hz> Задать частоту в Гц
- -a <set_amp> Параметры усилителя 1=Активен, 0=Неактивен.
- -l <gain_db> Задать lna усиление, диапазон 0-40dB, шаг 8dB.
- -i <gain_db> Задать vga(if) усиление, диапазон 0-62dB, шаг 2dB.
- -x <gain_db> Задать TX vga усиление, 0-47dB, шаг 1dB.
- -s <sample_rate_hz> Задать число выборок в секунду (доступные значения 2/8/10/12.5/16/20МГц)
- -n <num_samples> Задать количество отсчетов для передачи.
- -b <baseband_filter_bw_hz> Задать полосу пропускания фильтра в МГц.

Пример успешного задания сигнала с переносом частоты на 3 МГц, скоростью выборки 2 МГц, включенным усилителем и 40дБ усилением сигнала передачи: hackrftrans00.exe -t prog_29_05_24.bin -f 3000000 -s 2000000 -a 1 -x 40.

Описание тестирования программы загрузки синхронизирующей последовательности

Для тестирования необходимо подключить плату Arduino Due к управляющему устройству по USB, используя USB Native port (рисунок 4.1) на плате.

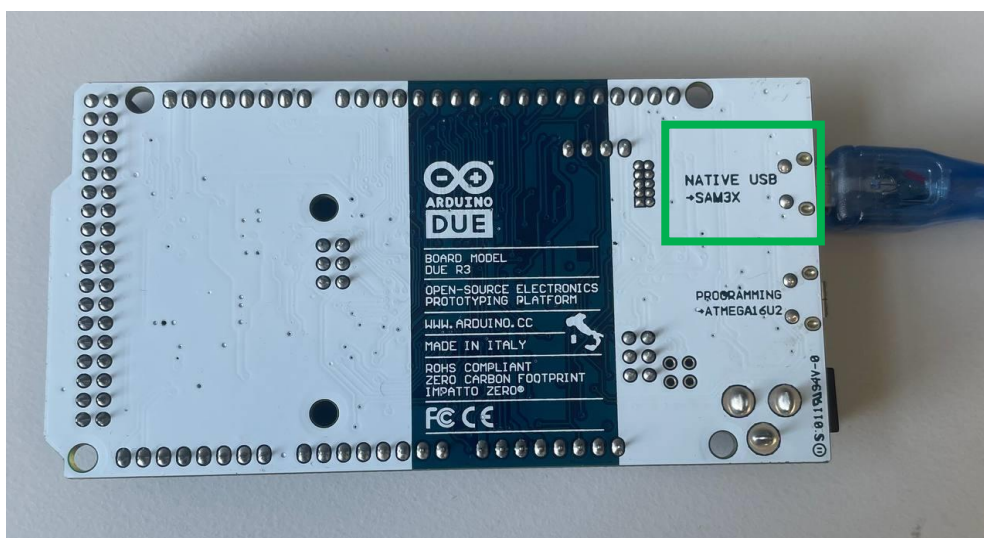


Рисунок 4.1 – Подключение к Native port Arduino

Сразу после подключения к компьютеру рекомендуется выполнить сброс платы с помощью кнопки на панели платы. Кроме подключения платы Arduino Due требуется также подключить к компьютеру по USB Hack RF и Picoscope серии 4000A. Радиочастотный выход Hack RF необходимо подключить к входу Picoscope через аттенюатор 10дБ, а также необходимо использовать параллельное подключение 50 Ом поскольку вход picoscope имеет высокое входное сопротивление (~МОм). TTL выводы hack rf необходимо подключить к портам 22 и GND на плате Arduino Due. Порт 8 Arduino Due необходимо подключить ко входу Picoscope. На рисунке 5 представлена экспериментальная установка.



Рисунок 5 – Экспериментальная установка

После подключения запустить программу Codeblocks, открыть проект Sync.cbr. Далее необходимо выбрать Com порт, к которому подключена плата Arduino Due, узнать это можно с помощью диспетчера устройств в соответствующем разделе. Пример определения Com порта представлен на рисунке 6

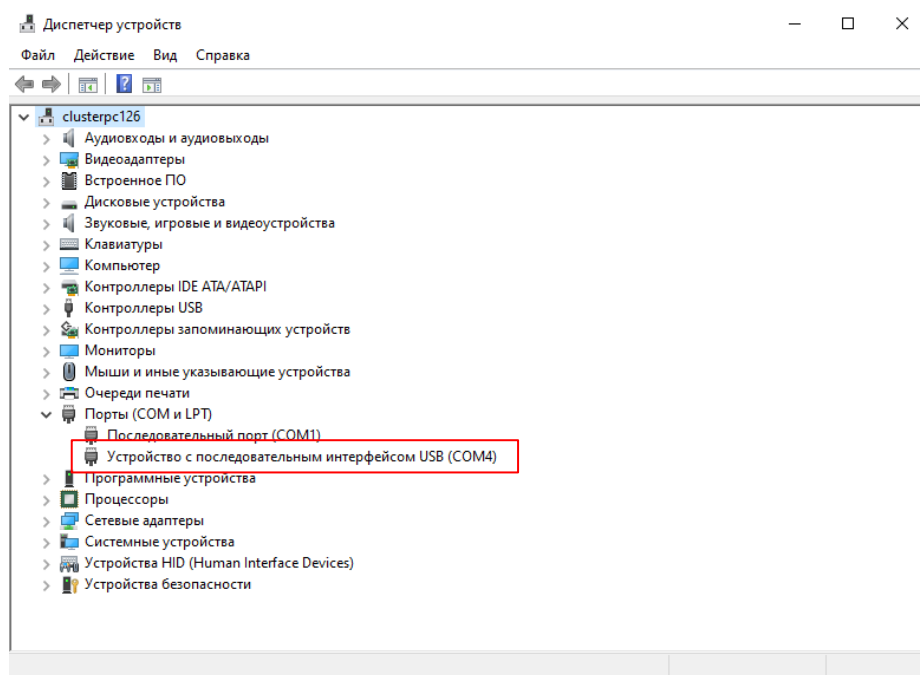


Рисунок 6 – Определение Com-порта, к которому подключена плата Arduino due

Далее необходимо прописать используемый Com порт в файле main.cpp проекта Sync.cbp, пример представлен на рисунке 5.

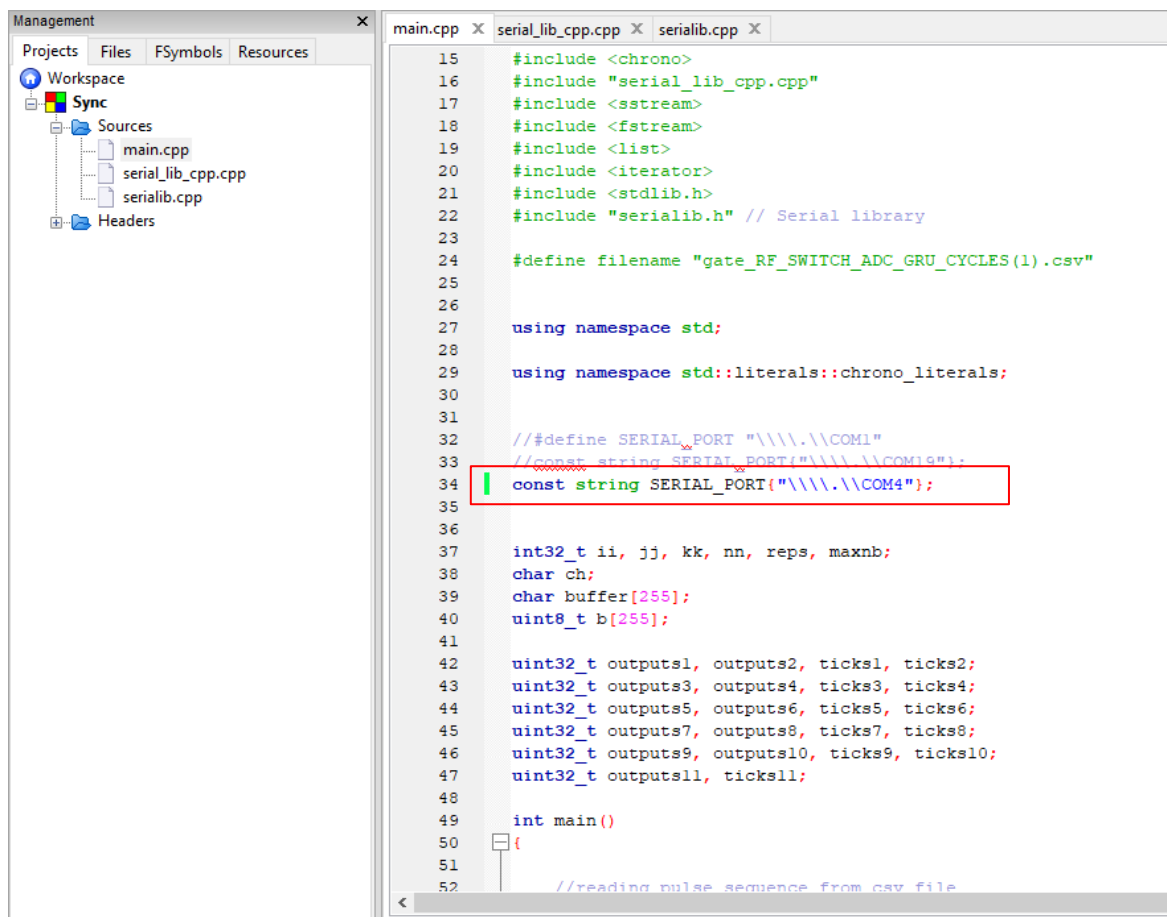


Рисунок 7 – Изменение последовательного порта

После изменения номера последовательного порта необходимо сохранить проект и скомпилировать. После этого тестовая программа может быть запущена. После запуска программы, импульсная последовательность загружается на плату Arduino Due, для старта импульсной программы необходимо отправить односимвольную команду “e” (означает “execute” – немедленное выполнение), после отправки односимвольной команды “e” плата будет ожидать прихода синхронизирующего **однополярного положительного сигнала амплитудой 3,3В на порт 22 (подача сигнала амплитудой 5В недопустима!)** Arduino Due. Для запуска синхронизирующей программы будем использовать начало передачи радиочастотной импульсной программы с Hack rf. Для запуска программы с hack rf необходимо с использованием командной строки перейти в директорию, в которой расположен exe файл, содержащий утилиту hackrf transfer. Для запуска

радиочастотной импульсной последовательности необходимо после имени exe файла указать требуемые параметры. Пример действий в консоли представлен на рисунке 8.

```
Командная строка
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.
C:\Users\michael.murzin>cd ../../
C:\>D:
D:\>cd D:\Low_field_MRI\Program_components\Radiofrequency_sequence\hack_rf\hackrftrans00\bin\Debug
D:\Low_field_MRI\Program_components\Radiofrequency_sequence\hack_rf\hackrftrans00\bin\Debug>hackrftrans00.exe -t prog_29_05_24.bin -f 3000000 -s 2000000 -a 1 -x 40
call hackrf_set_sample_rate(2000000 Hz/2.000 MHz)
call hackrf_set_hw_sync_mode(0)
call hackrf_set_freq(3000000 Hz/3.000 MHz)
call hackrf_set_amp_enable(1)
Stop with Ctrl-C
3.9 MiB / 1.002 sec = 3.9 MiB/second, average power -20.4 dBfs
3.9 MiB / 1.007 sec = 3.9 MiB/second, average power -19.9 dBfs
3.9 MiB / 0.993 sec = 4.0 MiB/second, average power -19.9 dBfs
4.2 MiB / 1.009 sec = 4.2 MiB/second, average power -20.2 dBfs
```

Рисунок 8 – Запуск утилиты hack rf из консоли

Для визуализации импульсной последовательности можно использовать любой осциллограф, рассмотрим визуализацию импульсной последовательности с использованием цифрового осциллографа Picoscope 4424A. На рисунке 9 представлена осциллограмма, полученная в результате теста.

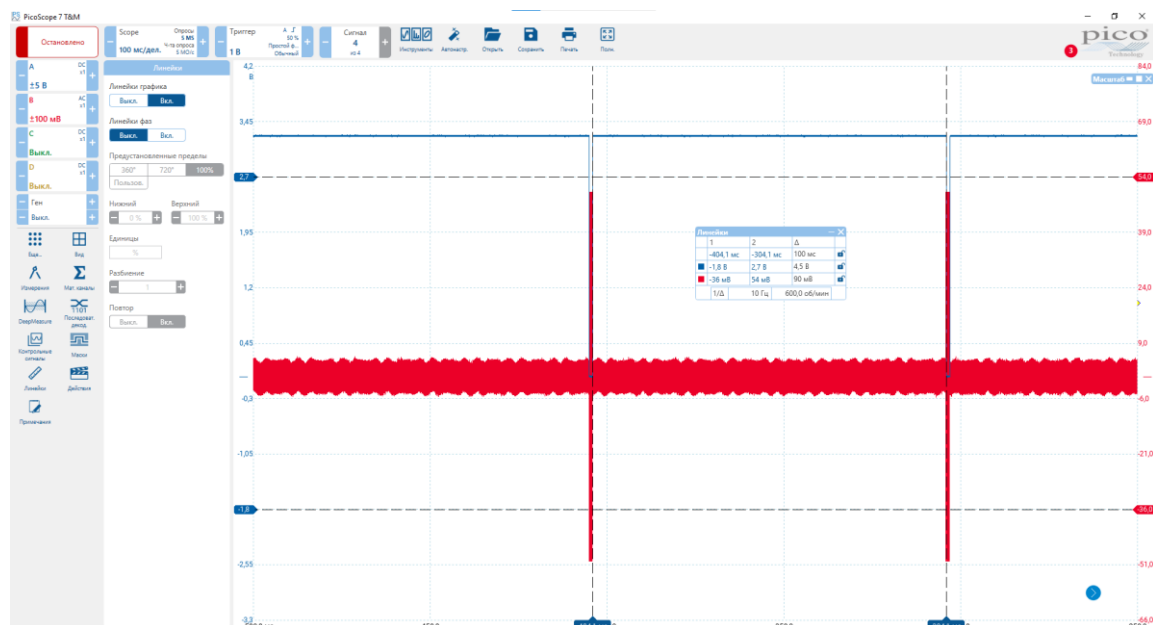


Рисунок 9 – Осциллограмма в результате тестирования

Красным цветом на осциллограмме показана последовательность радиочастотных импульсов, синим цветом показана последовательность синхронизирующих импульсов.

Описание программы задания параметров Picoscope

Программа задания параметров Picoscope составлена на основе описания Picoscope API programmers guide (источник: <https://www.picotech.com/download/manuals/picoscope-4000-series-a-api-programmers-guide.pdf>). Программа задания параметров определяет временные, амплитудные параметры развертки сигнала, параметры триггера, число точек для сбора

Описание тестирования программы задания параметров Picoscope

Для тестирования требуется следующее оборудование: Hack RF, Picoscope серии 4000A, управляющий компьютер. Для запуска теста необходимо подключить по USB Hack RF и Picoscope к управляющему ПК, а также соединить выход ANT через аттенюатор 10дБ к Picoscope. При подключении к Picoscope также требуется использовать ответвитель с подключенным 50 Ом, поскольку входы Picoscope имеют высокое сопротивление. Вид тестового стенда представлен на рисунке 10.



Рисунок 10 – Тестовый стенд для Picoscope API

Для тестирования необходимо запустить из командной строки файл `hackrf_trans00.exe` в директории `Hackrf\hackrftrans00\bin\Debug` со следующими параметрами:

```
hackrf_trans00.exe -c 127-f 3000000 -s 2000000 -a 1 -x 40
```

Для запуска программы задания параметров Picoscope необходимо запустить файл `pico_test_00_second_copy.exe`, находящийся в директории `Picoscope_API\pico_test_00_second_copy\bin\Debug`

После отработки программы задания параметров Picoscope API необходимо завершить передачу радиочастотного импульса Hack RF в консоли с помощью команды `Cntrl+C`. Далее необходимо запустить скрипт `plot_data.py` на языке Python для визуализации радиочастотного импульса. В результате запуска должно появиться изображение радиочастотного импульса, имеющего вид, представленный на рисунке 11.

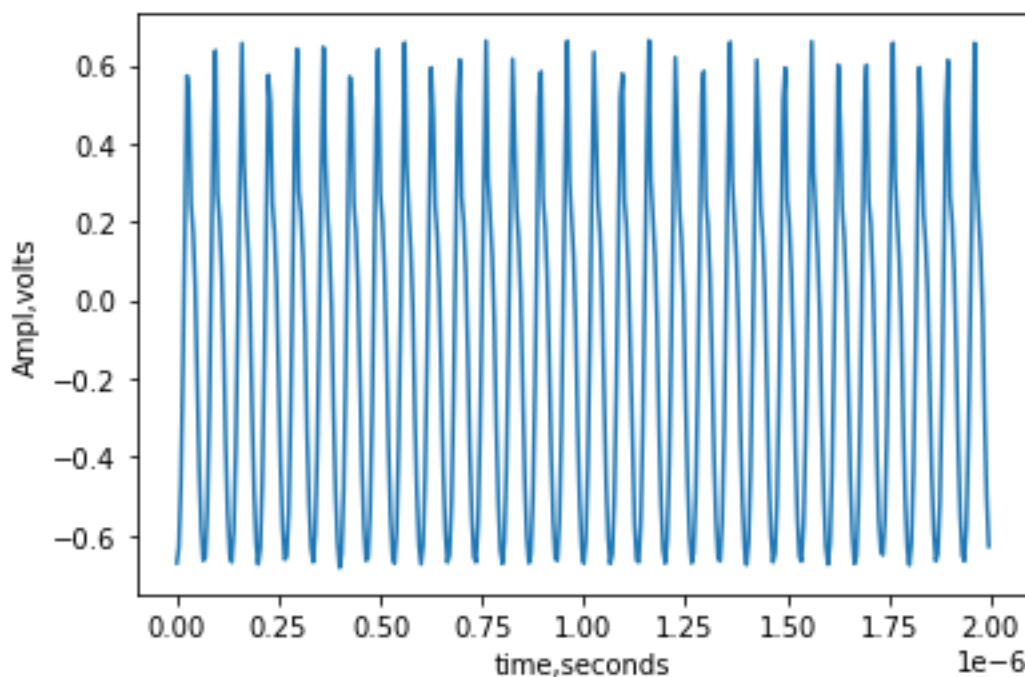


Рисунок 11 – Результат тестирования программы задания параметров Picoscope