

Краткое руководство к использованию **drsspectrum 2.3**

1. Установка

1.1. Ubuntu

Для установки в OS Ubuntu (и других Linux), использовать утилиту **make**. Запуск **make** без параметров скомпилирует код и соберёт программу в папку **./bin**.
make install скопирует программу в пользовательскую директорию **\$HOME/bin** если **\$HOME/bin** прописан в **\$PATH**, программу можно будет запускать из любого места, в противном случае запускать только из директории где находится сборка с указанием директории (к примеру **./bindrsspectrum** если запускать из той же директории где находитесь)

При сборке новой версии или изменении кода **make install** повторно

make debug Сборка отладочной версии. Использовать только при отладке кода.

1.2. Возможные проблемы

Используются заголовочные файлы стандартной библиотеки, поэтому в этой части проблем не должно быть, если используется GCC 4.7 и выше.

Также используются библиотеки ROOT. При установке может жаловаться на отсутствие некоторых рутовых библиотек, установка оных решает проблему.

~~Также можно поправить мэйкфайл, где используется **ROOTLIBS = \$(shell root-config --glibs)**. В данном случае прописываются все стандартные рутовые библиотеки. Закомментировав эту строчку (**#** перед строкой), и раскомментировав следующую, (**#ROOTLIBS = -L/usr/lib/x86_64-linux-gnu -lCore -lRIO -lHist -lGpad -lTree -lIm -lIdl**) Включаются только необходимые библиотеки.~~

По умолчанию в коде заголовочные файлы рута прописаны как **#include <root/TFile.h>**, чтобы использовать запись типа **#include <TFile.h>** добавить в мэйкфайле **-DDEBROOT** в **COM_FLAGS**. Также возможны другие проблемы как непосредственно с ROOT, так и другие (наверное только с рут) но это уже выходит за рамки данного руководства.

2. Использование

2.1. Параметры запуска

Запуск программы (имя программы может быть другое)

bindrsspectrum input_file

где **input_file**, файл данных, который нужно обработать. Либо

bindrsspectrum input_file noise_min noise_max signal_min signal_max.

где noise_min noise_max signal_min signal_max, локализация шума и сигнала.

Перечень параметров и краткое описание можно получить запустив программу с опцией **-h** или **--help**

-d, --only-detect Вычислить noise_min noise_max signal_min signal_max и выйти. Тут вроде всё понятно. Если нужно только узнать локализацию сигнала и шума.

-m, --channel-mode <channels> Выбрать рабочие каналы из **используемых** [4ch 3ch 2ch 1ch]. Записывать десятичное число. Пример: -m 4 для 3 канала (0b0100) По умолчанию используется режим 0b0001, то есть первый канал. Обращаю внимание, что здесь говорится об **используемых** каналах. То есть если используются 2, к примеру 3й и 4й, то для третьего 0b0001 для четвертого соответственно 0b0010 (2). Ну и для обоих 0b0011 (3).

-r, --without-root-app Запустить без root application. То есть не выводить гистограммы на экран, а выполнить программу и выйти.

-a, --a-factor <factor> установить множитель. По умолчанию 1
-b, --b-shift <shift> установить сдвиг. По умолчанию 0

-o, --out-png Вывести гистограммы в png. Гистограммы будут выведены в директорию res, в директории с данными.

-k, --amplitude Запустить в амплитудном режиме.

-n, --outdir <dir> Имя новой output директории. Новая директория будет в директории res.

-s, --safety Безопасный режим. Если во время набора менялось число каналов, в этом случае в обычном режиме программа даст ошибку. В безопасном режиме программа работает медленнее, так как используются все каналы, и идёт пересчёт событий. Поэтому если количество каналов не менялось во время набора лучше использовать режим по умолчанию.

-c, --max-num <num> установить максимальное число возможных каналов. По умолчанию 4. Это для случаев если каналов у девайса больше 4х. Используется только в безопасном режиме, так как в обычном идёт автоматический расчёт числа каналов.

-v, --volt-mode Изменить volt mode. По умолчанию диапазон от -0,5 до 0,5 V, изменить на 0 - 1 V. В целом эта опция ничего не делает, только меняет вид графика среднего сигнала.

-e, --online <sec> Использовать режим “онлайн”. Набор данных с обновлением гистограмм. **<sec>** выставляет частоту обновления в секундах. Минимум одна секунда. Возможны проблемы в отображении гистограмм. Если гистограммы не отображаются (пустой canvas), можно пересобрать программу изменив makefile.

Снимите комментарий со строки

#USETHREADAPP = -DUSETHREADAPP

и пересоберите, запустив **make clean make.**

Здесь используется «костыль», работа которого не до конца изучена, и возможны выбросы, но на тестах проблем не возникало.

2.2. Использование скриптов

Параметры можно использовать совместно, например **-r -s -o**, будет значить «запустить без root application в безопасном режиме, гистограммы сохранить в png в папку res». Если вас не устраивают параметры по умолчанию или лень каждый раз вводить параметры для однообразной задачи можно написать скрипт, который изменит поведение по умолчанию программы, и запускать уже скрипт.

Пример для скрипта из примера выше: script.sh

```
#!/bin/bash
```

```
./bindrssppectrum $@ -r -s -o
```

Затем выполнить **chmod +x script.sh** чтоб файл стал исполняемым и запускать его как обычную программу из терминала.

Здесь \$1 первый аргумент при запуске скрипта, для передачи его в нашу программу. Соответственно запуск программы происходит как **./script.sh namefile**

Можно разместить скрипт в пользовательскую папку \$HOME/bin (если папка прописана в переменной окружения), и запускать без указания директории из любого места.

3. Прочее

Возможны проблемы при обработке файлов в режиме autodetect. Кускоффский код расчётов я пока оставил как есть, а он иногда творит чудеса. В этом случае можно ввести границы шума и сигналов самостоятельно в виде **drsspectrum input_file noise_min noise_max signal_min signal_max**. Также иногда полезно вводить эти параметры и в обычных случаях, посмотрев вид среднего сигнала на гистограммах. Чтоб вывести гистограммы запустите программу с параметром **-o** Пока границы задаются одинаковыми для всех каналов.