

Начало

Так, парни, инструкция.

Все основные шаги описаны тут, но на языке вероятного противника

<https://www.expresslrs.org/software/toolchain-install/>

Посуди, всё что я напишу ниже, есть вольный перевод этой страницы

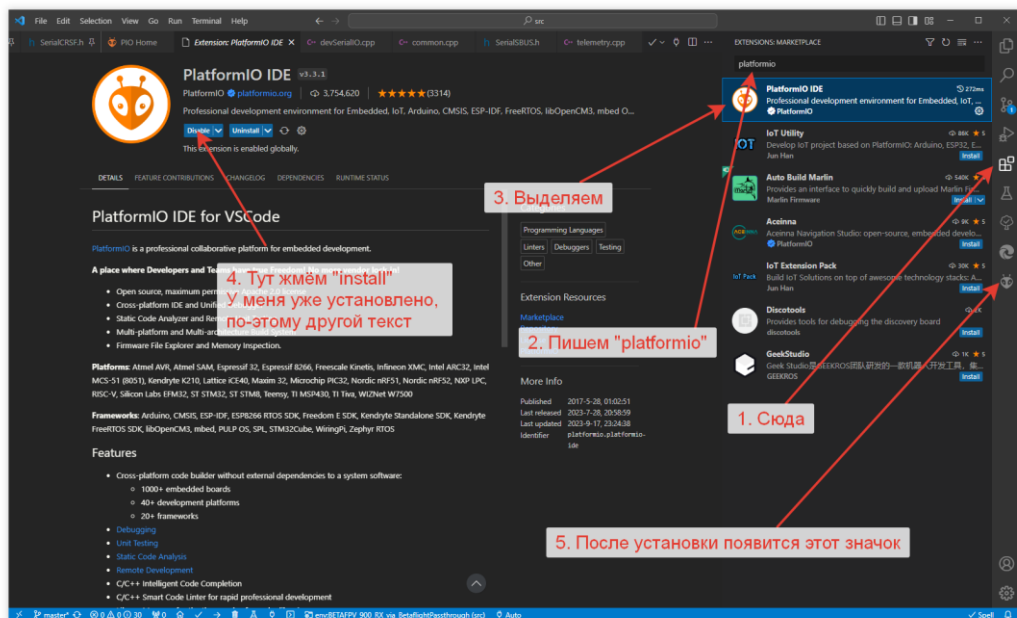
Рекомендую так же ознакомиться

<https://expresslrs.ru>

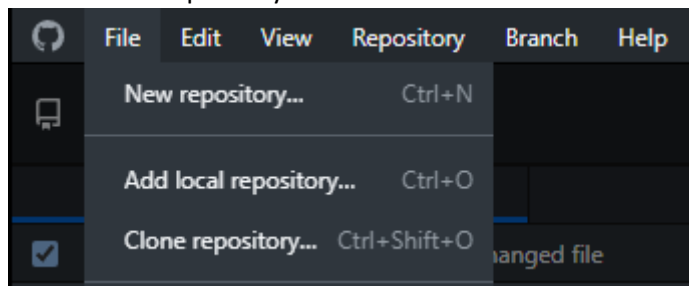
<https://habr.com/en/articles/744810/>

Настройка среды разработки

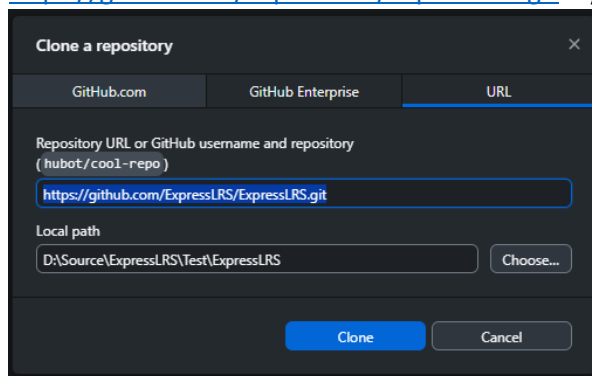
1. Надо поставить Visual Studio Code
2. Второе, ставим расширение PlatformIO. Тыкаем значок расширений, пишем в строке поиска "platformio".
Смотри картинку внизу.



3. Ставим git.
 - a. Ставим сам git <https://git-scm.com/download/win>
 - b. Рекомендую поставить GitHub Desktop. Очень дружелюбная софтина для управления репозиториями, отличный интерфейс, всё как вы любим, ссылка <https://desktop.github.com>
4. Клонировем репозиторий
 - a. File -> Clone repository...



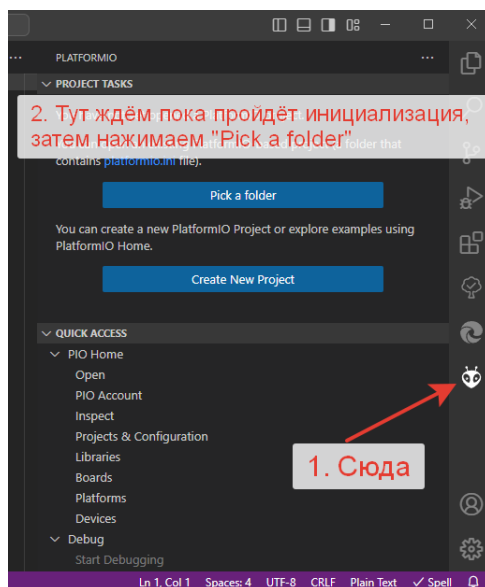
- b. В открывшемся окне выбираем вкладку "URL", вставляем путь к репозиторию <https://github.com/ExpressLRS/ExpressLRS.git> и указываем куда. Смотри скрин ниже



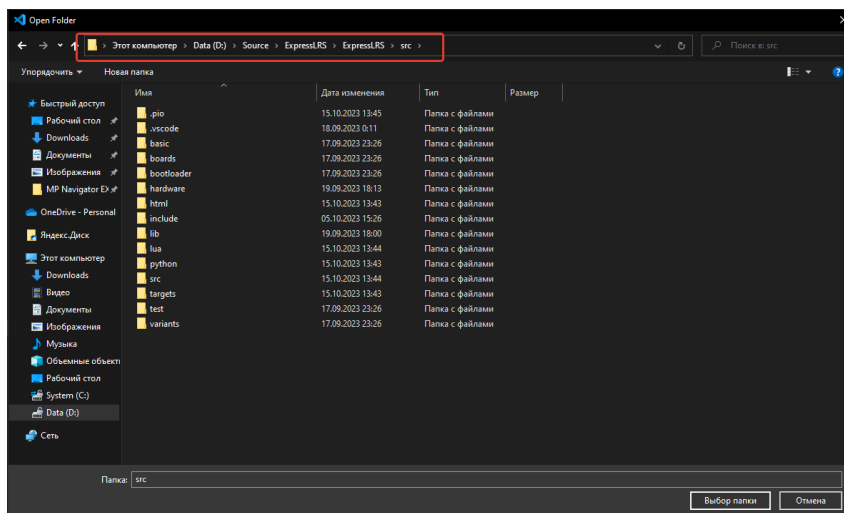
- c. Жмём «Clone» и ждём скачивания.

Открываем файлы на редактирование

Открываем VS Code, жмём иконку PlatformIO, затем выбираем папку с файлами.

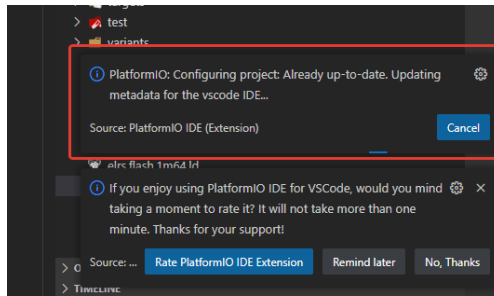


ВАЖНО, надо выбрать папку «src», смотри скрин ниже. В красном прямоугольнике выделен путь к нужной папке. Сами понимаете, начало может быть разным, но конец только один, это папка «src»



Дальше жмём «Выбор папки» и наблюдаем как PlatformIO начнёт инициализацию проекта.

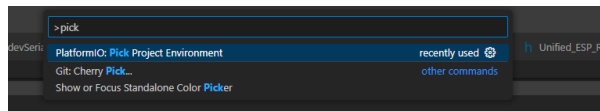
Если всё сделали правильно, то в правом нижнем углу увидите сообщение, что идёт инициализация. Если нет, значит что-то вы сделали не так. Подумайте что.



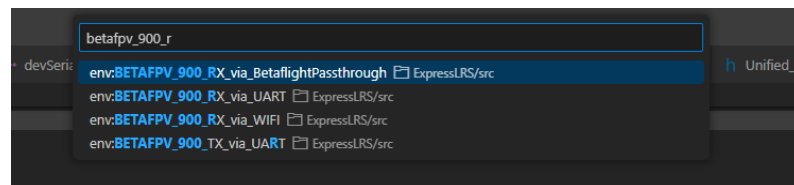
Редактирование и сборка

Для того, чтобы VSCode правильно отображала сам код и знала для чего мы разрабатываем, нам надо указать «платформу».

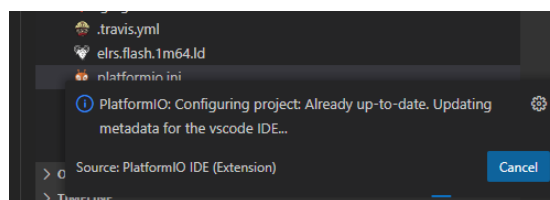
Жмём Ctrl + Shift + P – откроется палитра команд. Там ищем «pick project environment» и жмём Enter.



Откроется список возможных (известных) платформ. Для примера я выберу «BETAFPV_900_RX» для разработки под приёмник BETAFPV. Жмём Enter и смотрим как VSCode настраивает среду разработки под эту платформу.



Вот так.



Всё, мы готовы писать под этот приёмник.

Подготовка файлов перед сборкой

Надо подредактировать файл «user_defines.txt», там куча параметров, вот небольшая выжимка, которая позволит быстро стартовать, этих 4-х строк достаточно:

```
-DMY_BINDING_PHRASE="myPassPhrase"  
-DRegulatory_Domain_FCC_915  
-DAUTO_WIFI_ON_INTERVAL="60"  
-DUART_INVERTED
```

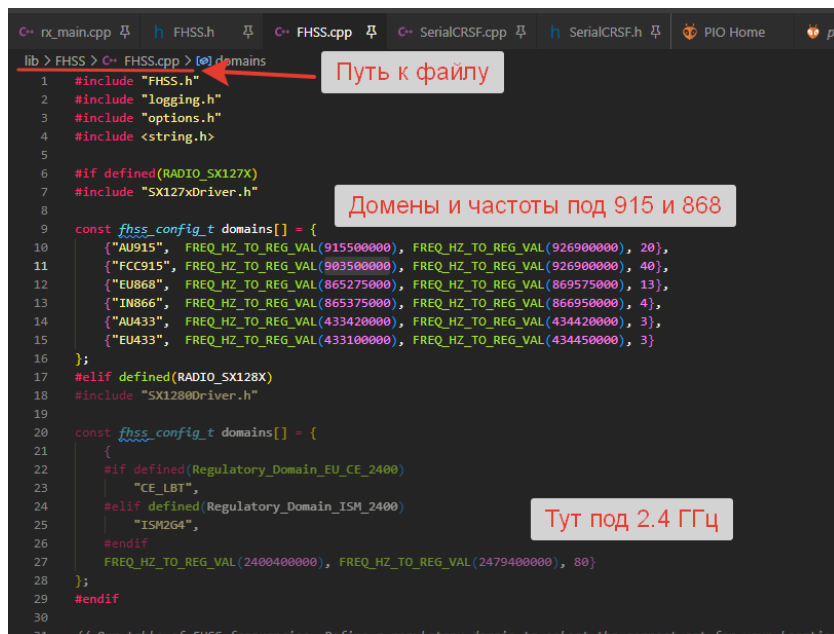
Как сместить частоты

Список доменов и их частот располагается в файле FSHN.cpp, смотри скрин ниже.

Здесь инициализируется структура из 4-х полей: Имя домена, начальная частота в интервале, конечная частота и количество каналов. Далее по коду интервал будет делиться на количество каналов и тем самым будет выбираться частота для коммуникации с передатчиком

Для примера, тест проводился на китайском BAYCK 915, таргет у него как раз BETAFPV_900_RX. В своём тесте я уменьшал частоты на 100 МГц. SDR анализатор показал что оно работает.

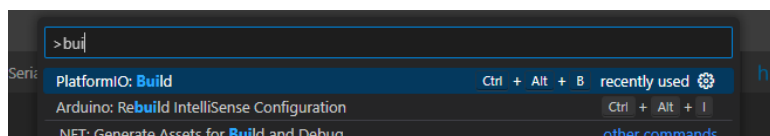
Важно, надо проводить тестирование приёмников передатчиков на предмет того, на какой частоте они могут работать после модификации.



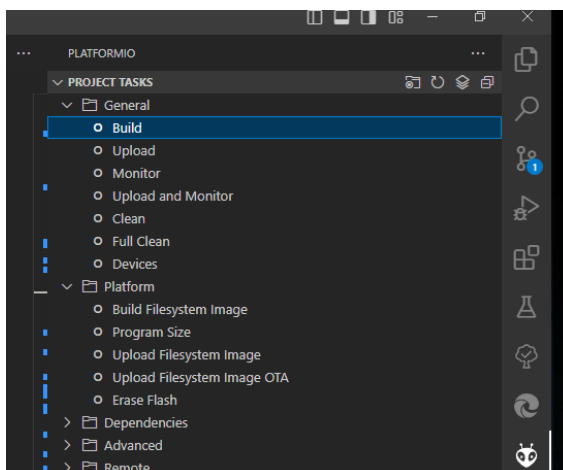
```
lib > FHSS > C++ FHSS.cpp > [?] domains
1 #include "FHSS.h"
2 #include "logging.h"
3 #include "options.h"
4 #include <string.h>
5
6 #if defined(RADIO_SX127X)
7 #include "SX127xDriver.h"
8
9 const fhss_config_t domains[] = {
10 {"AU915", FREQ_HZ_TO_REG_VAL(915500000), FREQ_HZ_TO_REG_VAL(926900000), 20},
11 {"FCC915", FREQ_HZ_TO_REG_VAL(903500000), FREQ_HZ_TO_REG_VAL(926900000), 40},
12 {"EU868", FREQ_HZ_TO_REG_VAL(865275000), FREQ_HZ_TO_REG_VAL(869575000), 13},
13 {"IN866", FREQ_HZ_TO_REG_VAL(865375000), FREQ_HZ_TO_REG_VAL(869500000), 4},
14 {"AU433", FREQ_HZ_TO_REG_VAL(433420000), FREQ_HZ_TO_REG_VAL(434420000), 3},
15 {"EU433", FREQ_HZ_TO_REG_VAL(433100000), FREQ_HZ_TO_REG_VAL(434450000), 3}
16 };
17 #elif defined(RADIO_SX128X)
18 #include "SX1280Driver.h"
19
20 const fhss_config_t domains[] = {
21 {
22 #if defined(Regulatory_Domain_EU_CE_2400)
23 "CE_LBT",
24 #elif defined(Regulatory_Domain_ISM_2400)
25 "ISM2G4",
26 #endif
27 FREQ_HZ_TO_REG_VAL(2400400000), FREQ_HZ_TO_REG_VAL(2479400000), 80
28 };
29 #endif
30
31 // Our table of FHSS frequencies. Define a regulatory domain to select the correct set for your location
```

Так, частоты поменяли, собираем!

Жмём "Ctrl + Shift + P" и в палитре команд выбираем «PlatformIO: Build»



Так же можно нажать на значок PlatformIO и выбрать сборку проекта, смотри скрин ниже.



Начнётся сборка. Для сборки важно наличие интернета, так как PlatformIO может закачивать необходимые для сборки библиотеки.

После сборки у вас будет файл с прошивкой, который можно прошить в приёмник.

```
Retrieving maximum program size .pio\build\BETAFPV_900_RX_via_BetaflightPassthrough\firmware.elf
Checking size .pio\build\BETAFPV_900_RX_via_BetaflightPassthrough\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [=====] 57.2% (used 46868 bytes from 81920 bytes)
Flash: [=====] 51.9% (used 514324 bytes from 991216 bytes)
Building .pio\build\BETAFPV_900_RX_via_BetaflightPassthrough\firmware.bin
Creating BIN file ".pio\build\BETAFPV_900_RX_via_BetaflightPassthrough\firmware.bin" using
appendConfiguration([".pio\build\BETAFPV_900_RX_via_BetaflightPassthrough\firmware.bin"],
esp8285
```

Ну или можно сразу прошить, если выбрать «Upload». Ну вот и готово!

Как быстро шить из под Express LRS Configurator

Когда вы поправили файлы под ваши частоты, то можно собирать прошивки локально и заливать в устройства намного быстрее.

Открываем configurator и смотрим скрин ниже

Версия прошивки

ОФИЦИАЛЬНЫЕ ВЕРСИИ ВЕТКА GIT КОММИТ GIT ЛОКАЛЬНЫЙ ПУЛЛ РЕКВЕСТ GIT

ОПАСНАЯ ЗОНА
Используйте эти исходники, только если вы знаете, что вы делаете или были проинструктированы разработчиками проекта

Локальный путь
D:\Source\ExpressLRS\ExpressLRS\src 1. Указываем папку с исходниками

ВЫБЕРИТЕ ПАПКУ

Устройство

Категория устройства
BETAFPV 900 MHz 2. Указываем устройство

Устройство
BETAFPV 900MHz RX

Метод прошивки
☒ BetaflightPassthrough ? ☐ UART ? ☐ WIFI ? 3. Остальные параметры и шьёмся! Это где-то там внизу :)