# No description available.

# STM32 (Основной) Быстрый старт

1. Подключите USB-C к порту USB-C
2. Откройте проект Arduino IDE для STM32H747, целевая плата — Arduino Giga R1
   1. Исходный код:<https://github.com/Time-Appliances-Project/Incubation-Projects/tree/master/Software/WiWi/SDR/V4/WiWi_SDR>
3. Загрузите это в COM-порт платы.
   1. Если COM-порт не найден, переведите в режим загрузки. Удерживайте кнопку Boot (SW2, кнопка дальше от USB-C) и нажмите кнопку Reset (SW1, кнопка ближе к USB-C), и отпустите SW2. Это переведет в режим загрузки STM32 по умолчанию
   2. Если вы также хотите перепрошить загрузчик Arduino, используйте процедуру программирования загрузчика Arduino, ее не следует повторять более одного раза на плату.
      1. https://support.arduino.cc/hc/en-us/articles/7991505977116-Burn-the-bootloader-on-GIGA-R1-WiFi
4. Откройте Tera term на COM-порте платы, теперь вы должны увидеть CLI для STM32.
   1. CLI похож на однослойную файловую систему. Например, вы можете получить доступ к вещам, связанным с dpll, набрав dpll . Чтобы выйти из системы меню dpll, просто используйте cd ..
5. Теперь у вас есть исходный код и доступ к STM32, главному контроллеру платы.

# ESP32S3 Быстрый старт

1. доступ ESP32S3
   1. При подключенном USB-C (питание подается через USB-C) подключите его к порту Micro-USB рядом с портами антенны SMA.
   2. Перейдите в режим загрузчика, если вы никогда ранее не прошивали или хотите перепрошить с нуля:
      1. Удерживайте SW3 (GPIO0, кнопка ближе к Micro USB) и переключите SW4 (сброс, кнопка дальше от Micro USB), затем отпустите SW3
2. Программа с примером Arduino IDE ESP32S3 скетч, параметры Arduino приведены ниже в разделе ESP32S3
   1. Источник:<https://github.com/Time-Appliances-Project/Incubation-Projects/tree/master/Software/WiWi/SDR/V4/ESP32S3_SDR>
3. Откройте Tera Term на последовательном порту ESP32S3, затем используйте CLI для доступа к функциям ESP32S3 и исходному коду

# SDR (RX) Быстрый старт

1. Начните подачу радиочастотного сигнала в субгигагерцовый тракт, антенна SDR — J8, верхний левый SMA
   1. Я инжектировал с помощью LimeSDR синусоиду частотой 100 кГц с несущей частотой 900 МГц.
2. На STM32 приостановите поток IQ (он будет запущен по умолчанию при загрузке) и распечатайте его.
   1. В Tera Term начните запись в текстовый файл, затем выполните эти команды
      1. cd ..
      2. spi\_iq
      3. subg\_pause
      4. channel\_print\_samples 0 5000
3. Теперь у вас будет 5000 I/Q-выборок в виде 16-битных значений со знаком, напечатанных в шестнадцатеричном формате.

# Аппаратные компоненты

## Микроконтроллер STM32H747

### IDE / Программирование

* Arduino IDE (Arduino Giga R1 как плата) -<https://docs.arduino.cc/tutorials/giga-r1-wifi/giga-getting-started/>
* Последовательный порт USB для программирования/отладки — это разъем USB-C рядом с STM32H747
* Кнопка загрузки -> SW2, кнопка дальше от USB-C
* Кнопка RST -> SW1, кнопка ближе к USB-C
* Справочный код использует загрузчик Arduino Giga R1 и IDE, но код использует весь код STM32 HAL
* Код STM32 HAL -><https://github.com/STMicroelectronics/stm32h7xx-hal-driver>
* Мой исходный код: https://github.com/Time-Appliances-Project/Incubation-Projects/tree/master/Software/WiWi/SDR/V4/WiWi\_SDR

### CLI

* Реализовано с использованием встроенной библиотеки cli -><https://github.com/funbiscuit/embedded-cli>
* В настоящее время реализовано однослойное меню, разное оборудование/системы имеют свои собственные меню. Посмотрите разделы CLI для других компонентов в этом руководстве по использованию, чтобы увидеть некоторые примеры и то, что реализовано

### Техническая спецификация

https://www.st.com/resource/en/datasheet/stm32h747ag.pdf

### Справочное руководство

https://www.st.com/resource/en/reference\_manual/rm0399-stm32h745755-and-stm32h747757-advanced-armbased-32bit-mcus-stmicroelectronics.pdf

### Еще не проверено

* USB High speed через USB3320, должно быть как у Arduino Giga
* 100M Ethernet через LAN8472 phy. Должно быть то же самое, что и у Arduino Portenta? Код lan8472.c из библиотеки STM32 включен в libmbed.a, которая включена в компиляцию Arduino Giga, но не проверял никакой функциональности

## 8A34001 ДФАПЧ

### Цель

* Настраиваемый источник частоты с чрезвычайно высоким разрешением для всех компонентов, кроме тактового генератора ядра STM32
* Временная метка с чрезвычайно высоким разрешением
* Выравнивание и измерение времени и частоты с внешними разъемами U.FL для измерения и ввода источников времени и/или частоты

### графический интерфейс

* Документация 8A34001 -> https://www.renesas.com/en/products/clocks-timing/application-specific-clocks/network-synchronization/ieee-1588-and-synchronous-ethernet-clocks/8a34001-system-synchronizer-ieee-1588-eight-channels?srsltid=AfmBOooG1SNjJe9RvMbQirHlspVYP4Z1xA2yP6WvAmFFQsMtCW\_i4wAy#design\_development
* Renesas Timing Commander -> https://www.renesas.com/en/software-tool/timing-commander
* Нужны файлы личности для 8A34001 -><https://www.renesas.com/en/document/swo/timing-commander-personality-file-clockmatrix-8a34001-v10130-fw488-fw498?r=455866>
* GUI генерирует файл .tcs, который является текущей конфигурацией
* Для целей программирования GUI генерирует файл EEPROM .hex

### CLI

* Из главного меню, dpll
  + Чтобы обновить EEPROM (только для новых плат), прошейте с помощью Xmodem
    - Xmodem\_update
    - Tera term -> Файл -> Передача -> Xmodem -> Отправить, выберите .hex-файл, созданный с помощью графического интерфейса EEPROM Timing Commander

## SiT5501 TCXO

### Цель

* Высокостабильный (10ppb) источник частоты для платы
* Также настраивается I2C (зависит от спецификации)

### Техническая спецификация

* https://www.sitime.com/support/resource-library/datasheets/sit5501-datasheet

### CLI

## ПЛИС LFE5U-25F

### Цель

* Преобразовать поток данных IQ Sub-GHz и 2.4GHz из AT86RF215IQ в формат, используемый микроконтроллером STM32H747. По умолчанию используется несколько потоков SPI
* Сигналы выбора управления на плате, например, управление радиочастотным переключателем
* Необязательно: обработка потоков данных IQ перед отправкой в ​​STM32H747, в настоящее время не реализовано
* Дополнительно: отправка сигнала TX из AT86RF215IQ, в настоящее время не реализовано

### IDE

* Решётчатый алмаз:<https://www.latticesemi.com/latticediamond>
* Исходный код: https://github.com/Time-Appliances-Project/Incubation-Projects/blob/master/Software/WiWi/SDR/V4/ECP5\_Test.7z

### Техническая спецификация

* <https://www.latticesemi.com/Products/FPGAandCPLD/ECP5>
* Техническая спецификация:<https://www.latticesemi.com/view_document?document_id=50461>

### CLI

* С верхнего уровня: fpga
* Обновить изображение (мой пример с использованием термина tera)
  + обновление\_xmodem
  + Tera term -> Файл -> Передача -> Xmodem -> Отправить, выберите файл .bit, созданный Diamond
* доступ I2C
  + read\_reg
  + write\_reg

## AT86RF215IQ SDR

### Цель

* Преобразователь RF Front-end в цифровой поток данных IQ
* Параллельные приемопередающие тракты для Sub-GHz и 2.4GHz, оба работают одновременно

### Техническая спецификация

* https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42415-WIRELESS-AT86RF215\_Datasheet.pdf

### Справочный код

* <https://github.com/cariboulabs/cariboulite/tree/main/software/libcariboulite/src/at86rf215>
* https://github.com/uw-x/tinysdr/tree/main/firmware/at86rf215

### CLI

* С главного, at86
  + Read\_reg
  + Write\_reg

## Приемопередатчик SX1276 LoRA / GFSK

### Цель

* Стандартный приемопередатчик LoRA/GFSK Sub-GHz, настраиваемый через SPI, имеет контакты прерывания для пакетной работы (TX done / RX done / другие)

### Техническая спецификация

* https://www.mouser.com/datasheet/2/761/sx1276-1278113.pdf?srsltid=AfmBOoroVoWihTFWYKwrhVO9QEcf5YYK2sFgTSESLSAekwR4-rMPhS78

### CLI

* Из главного, sx1276
  + Read\_reg
  + Write\_reg

## Микроконтроллер ESP32S3 и WiFi-приемопередатчик

### IDE / Программирование

* Используйте Arduino IDE с установленным менеджером плат ESP32
  + <https://docs.espressif.com/projects/arduino-esp32/en/latest/installing.html>
* Для программирования подключите к порту Micro USB рядом с разъемами SMA на плате.
  + SW3 — это GPIO0, кнопка ближе к Micro USB
  + SW4 сброшен, кнопка дальше от Micro USB
  + Чтобы войти в режим загрузки (в первый раз или при возникновении неполадки), удерживайте кнопку SW3 и переключите кнопку SW4, затем отпустите кнопку SW3.
* Это автономный микроконтроллер, мой эталонный код использует ту же библиотеку CLI.
  + Исходный код: https://github.com/Time-Appliances-Project/Incubation-Projects/tree/master/Software/WiWi/SDR/V4/ESP32S3\_SDR

A screenshot of a computer

AI-generated content may be incorrect.

### CLI

* Из главного меню, esp32
* Для доступа к ESP32 UART через STM32 (использует UART0 ESP32S3 на плате)
  + проходной

### Техническая спецификация

* https://www.espressif.com/sites/default/files/documentation/esp32-s3\_datasheet\_en.pdf

### Справочное руководство

* https://www.espressif.com/sites/default/files/documentation/esp32-s3\_technical\_reference\_manual\_en.pdf