

Добавление платы в Marlin.

1. Добавляем файл определения пинов
..\Marlin\src\pins\stm32f4\pins_STM32F401CCU6_UNI.h.
2. Добавляем в файл ..\Marlin\src\pins\pins.h:

```
#elif MB(STM32F401CCU6_UNI)
  #include "stm32f4/pins_STM32F401CCU6_UNI.h"          // STM32F4
  env:blackpill_f401cc_uni env:blackpill_f401cc_uni_bootloader
  env:blackpill_f401cc_uni_nobootloader
```

Важно: в закомментированной части строки есть информация о необходимом окружении, она парсится скриптом во время подготовки к компиляции, поэтому должна соответствовать имени вашего окружения на следующем шаге.

3. Добавляем в файл ..\Marlin\src\core\boards.h:

```
#define BOARD_STM32F401CCU6_UNI          5301 // STM32F401CCU6 BLACKPILL BOARD UNI
```

Номер после названия платы может быть любым (при выходе свежих версий система нумерации периодически изменяется, а уже существующие номера сдвигаются), главное чтобы он был уникальным.

4. Добавляем/редактируем в файле ..\Marlin\Configuration.h следующую информацию:

```
#ifndef MOTHERBOARD
  #define MOTHERBOARD BOARD_STM32F401CCU6_UNI
#endif

#define SERIAL_PORT          -1
#define BAUDRATE              250000
```

5. Добавляем информацию об окружении для платы в файл ..\ini\stm32f4.ini:

```
#
# blackpill_f401cc
#
[env:blackpill_f401cc_uni]
extends              = stm32_variant
platform             = ststm32@~14.1.0
platform_packages    = framework-arduino~4.20600.231001
                    toolchain-gccarmnoneabi@1.100301.220327
board               = genericSTM32F401CC
build_flags         = ${stm32_variant.build_flags} -DHSE_VALUE=25000000U -O0

[env:blackpill_f401cc_uni_bootloader]
extends             = env:blackpill_f401cc_uni
board_build.offset  = 0x8000

[env:blackpill_f401cc_uni_nobootloader]
extends             = env:blackpill_f401cc_uni
board_build.offset  = 0x0000
upload_protocol     = stlink
```

6. Компилируем прошивку в PlatformIO выбрав окружение `[blackpill_f401cc_uni_nobootloader]` если не будет использоваться bootloader, или `[blackpill_f401cc_uni_bootloader]` если будет использоваться bootloader.
- Важно:** в файле `..\Marlin\src\pins\stm32f4\pins_STM32F401CCU6.h` необходимо настроить симуляцию EEPROM, оставив раскомментированной только **ОДНУ** строку:
- `#define SDCARD_EEPROM_EMULATION` – при использовании прошивки без bootloader’а
- `#define FLASH_EEPROM_EMULATION` – при использовании прошивки с bootloader’ом.

Следующая таблица поможет вам выбрать вариант установки прошивки:

С использованием bootloader’а	Без использования bootloader’а
Необходимо предварительно загрузить bootloader в память МК (1 раз).	Можно сразу заливать прошивку в МК.
Можно загружать/обновлять прошивку используя SD карту.	Загружать/обновлять прошивку можно только при помощи программатора.
EEPROM симулируется во FLASH памяти МК.	EEPROM симулируется на внешней SD карте, которая должна быть вставлена в кардридер до включения принтера для корректной работы EEPROM.
Максимальный размер прошивки: 224 Кбайт.	Максимальный размер прошивки: 256 Кбайт.

7. При использовании bootloader’а, первоначально необходимо залить сам bootloader. Используйте файл `BOOTLOADER_F401CC_UNI_SPI_SD.hex` из репозитория. Предварительно выполните полную очистку FLASH. После загрузите файл bootloader’а при помощи ST-Link и STM32CubeProgrammer. Затем вы можете заливать/обновлять прошивку просто записывая файл прошивки Marlin с именем **firmware.bin** на SD карту. При включении принтера bootloader проверит SD карту на наличие файла **firmware.bin** и в случае его обнаружения запишет его во FLASH память МК, а после удачной верификации переименует файл в **firmware.cur** и запустит основную прошивку.
8. В случае, если вы решите не использовать bootloader, вы можете сразу загрузить прошивку во FLASH память МК при помощи ST-Link’а из VSC с установленным PlatformIO или из STM32CubeProgrammer.

Добавление своих терморезисторов в Marlin.

1. Создаем header-файл с калибровкой терморезистора и именем `..\Marlin\src\module\thermistor\thermistor_59.h`:

```
#pragma once

constexpr temptable_entry_t temptable_59[] PROGMEM = {
  { OV(91), 300 },
  { OV(106), 290 },
  { OV(121), 280 },
  { OV(140), 270 },
  { OV(161), 260 },
  { OV(186), 250 },
  { OV(217), 240 },
  { OV(248), 230 },
  { OV(287), 220 },
  { OV(330), 210 },
  { OV(374), 200 },
  { OV(426), 190 },
  { OV(478), 180 },
}
```

```

    { OV(536), 170 },
    { OV(594), 160 },
    { OV(654), 150 },
    { OV(709), 140 },
    { OV(760), 130 },
    { OV(807), 120 },
    { OV(850), 110 },
    { OV(887), 100 },
    { OV(919), 90 },
    { OV(944), 80 },
    { OV(964), 70 },
    { OV(980), 60 },
    { OV(993), 50 },
    { OV(1001), 40 },
    { OV(1006), 30 },
    { OV(1007), 25 },
    { OV(1017), 0 }
};

```

- В файл записываем свою таблицу калибровки, показания АЦП (чтобы их видеть в консоли, необходимо раскомментировать параметр **#define SHOW_TEMP_ADC_VALUES** в файле ..\Marlin\Configuration_adv.h) и показания реальной температуры, полученной с помощью своей термопары или иным заведомо точным способом. Значения АЦП записываются в 10-битном виде, поэтому если у вас АЦП имеет 12 бит, то значения стоит делить на 4 (2 в степени битность вашего АЦП минус 10). Чем чаще шаг по температуре, тем лучше, но чаще 10 смысла делать нет, верхний предел записываем тот, который нам необходим (очевидно, нет смысла калибровать стол до 270 градусов).
- В файл ..\Marlin\src\module\thermistor\thermistors.h добавляем строки:

```

#if ANY_THERMISTOR_IS(59)
  #include "thermistor_59.h"
#endif

```

- В файле ..\Marlin\Configuration.h определяем как используемый наш резистор:

```

#define TEMP_SENSOR_0 59

```