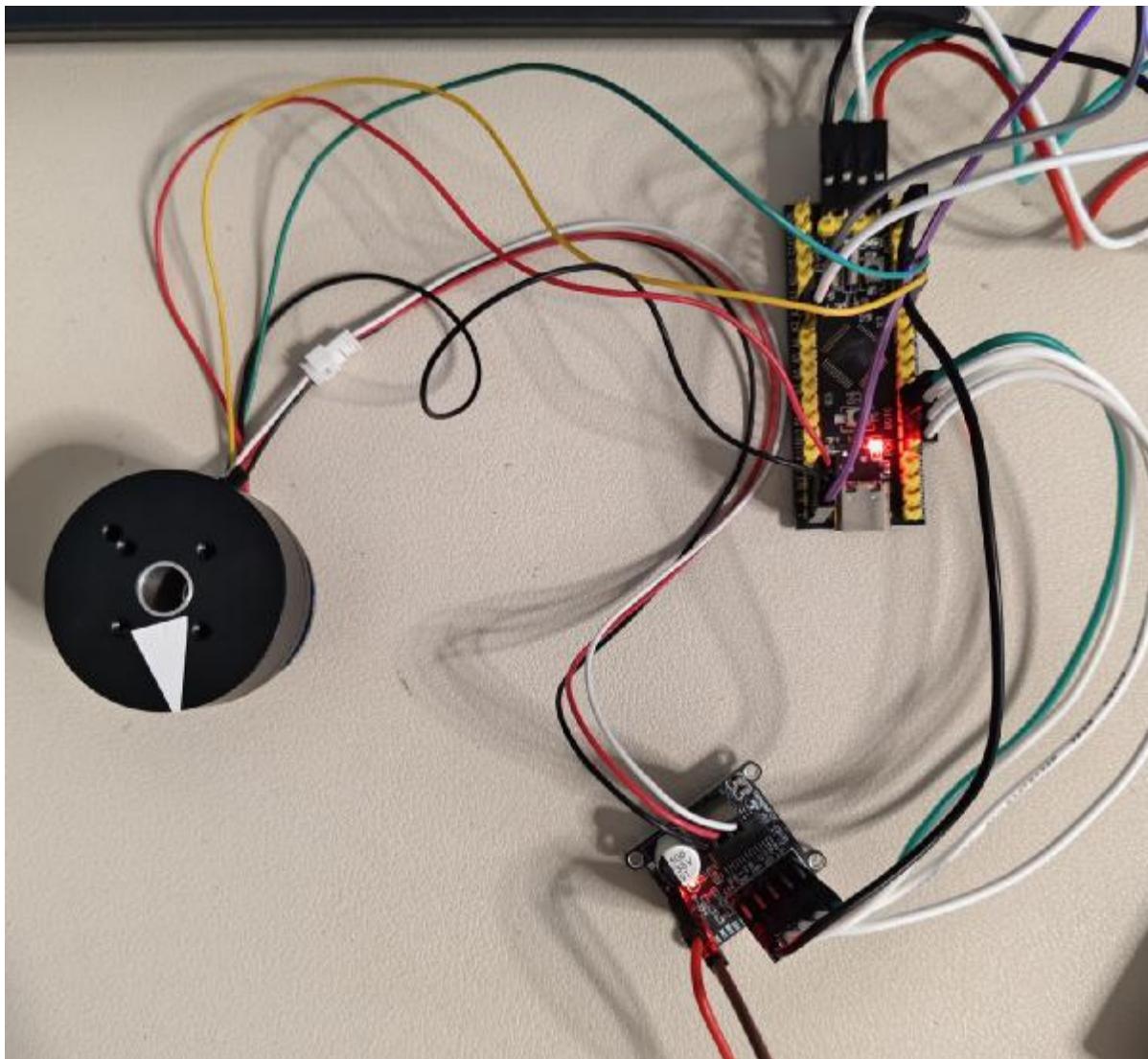


STM32 引用 SimpleFOC 库闭环驱动直流无刷电机

M 创动工坊提供 mcdgf.taobao.com

一、 硬件准备



STM32 核心板，2804 电机和 SimpleFOC mini 驱动板（M 创动工坊提供），ST-link，12V 直流电源，USB 线等

二、 软件准备

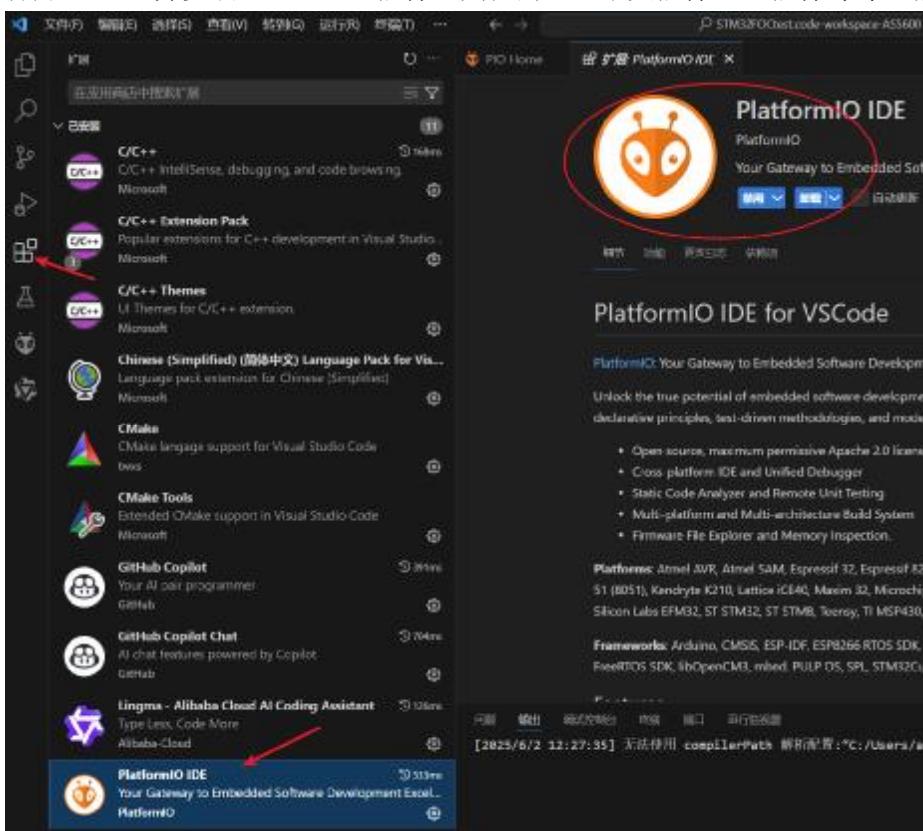
1. 安装微软的 Vscode，网上很多教程，且有说明书



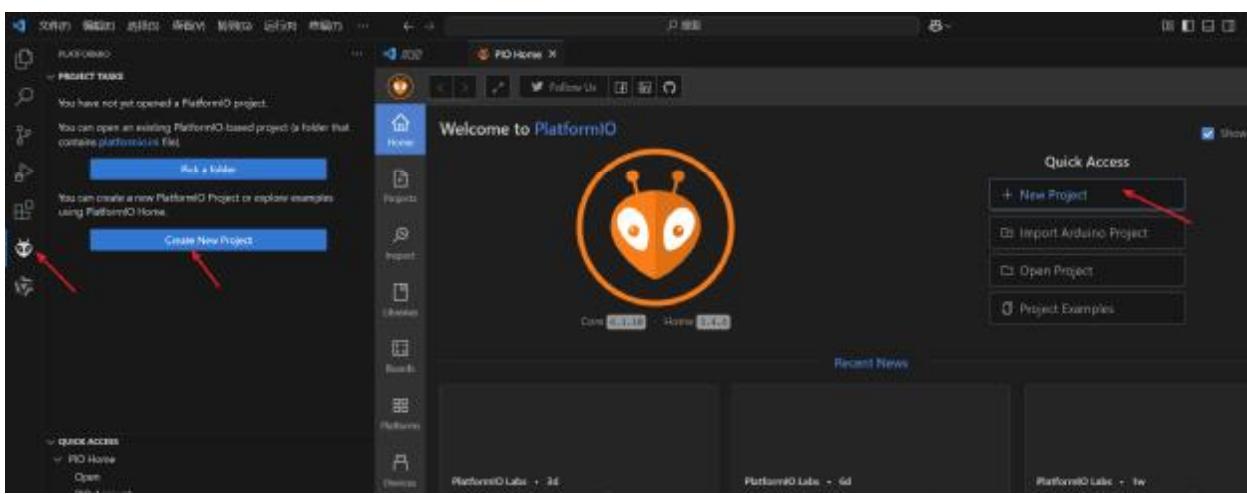
安装好 stlink 驱动



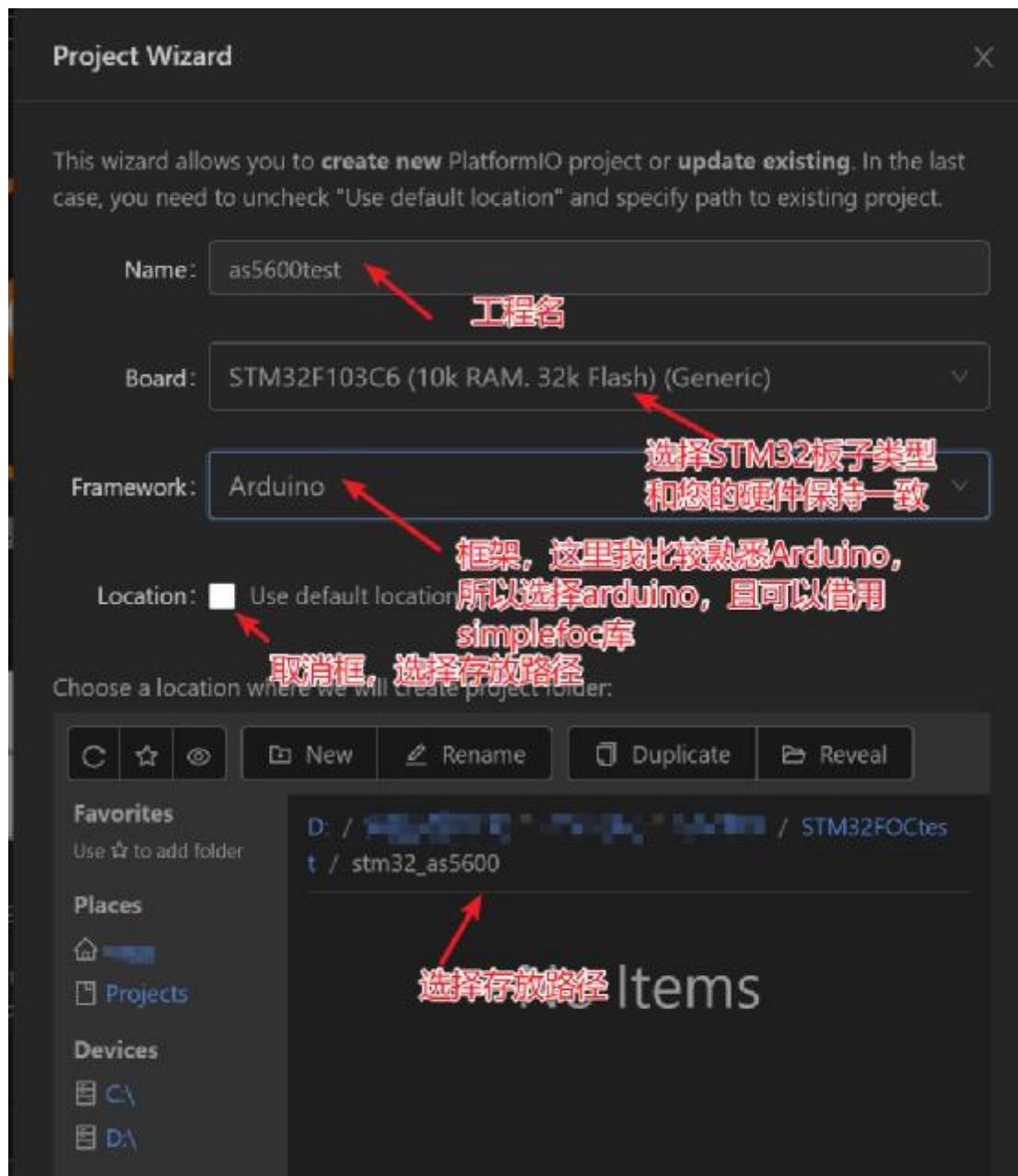
2. 打开 vscode 并安装 PlatformIO 插件。其他的一些中文插件、AI 插件等等，根据需要安装



3. 打开插件，新建工程



4. 重要的设置



5. Ini 文件设置, 这是关键

```

platformio.ini
1 ; PlatformIO Project Configuration File
2 ;
3 ; Build options: build flags, source filter
4 ; Upload options: custom upload port, speed and extra flags
5 ; Library options: dependencies, extra library storages
6 ; Advanced options: extra scripting
7 ;
8 ; Please visit documentation for the other options and examples
9 ; https://docs.platformio.org/page/projectconf.html
10
11 [env:genericSTM32F103C6]
12 platform = ststm32
13 board = genericSTM32F103C6
14 framework = arduino
15

```

Ini文件设置

```

platformio.ini
1 ; PlatformIO Project Configuration file
2 ;
3 ; Build options: build flags, source filter
4 ; Upload options: custom upload port, speed and extra flags
5 ; Library options: dependencies, extra library storages
6 ; Advanced options: extra scripting
7 ;
8 ; Please visit documentation for the other options and examples
9 ; https://docs.platformio.org/page/projectconf.html
10
11 [env:genericSTM32F103C8]
12 platform = ststm32
13 board = genericSTM32F103C8
14 framework = arduino
15
16 lib_deps =
17     askuric/Simple FOC@ 2.3.4
18 board_build.mcu = stm32f103c8t6
19 board_upload.maximum_size = 32768
20 upload_protocol = stlink
21 debug_tool = stlink
22 build_flags = -Os
23

```

1. 将SimpleFOC库
2. 定义子
3. 定义使用stlink协议

```

[env:genericSTM32F103C8]
platform = ststm32
board = genericSTM32F103C8
framework = arduino
lib_deps = askuric/Simple FOC@ 2.3.4
board_build.mcu = stm32f103c8t6

```

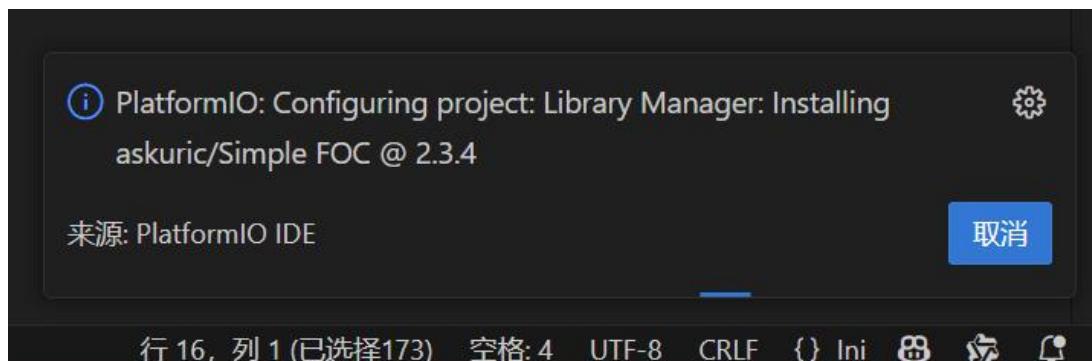
; 烧录工具配置（根据使用的工具选择）

upload_protocol = stlink ; 使用 ST-Link 烧录

; 调试端口配置（ST-Link）

debug_tool = stlink

build_flags = -Os ; 优化代码体积写好后，记得点保存，就开始自动下载库，右下角



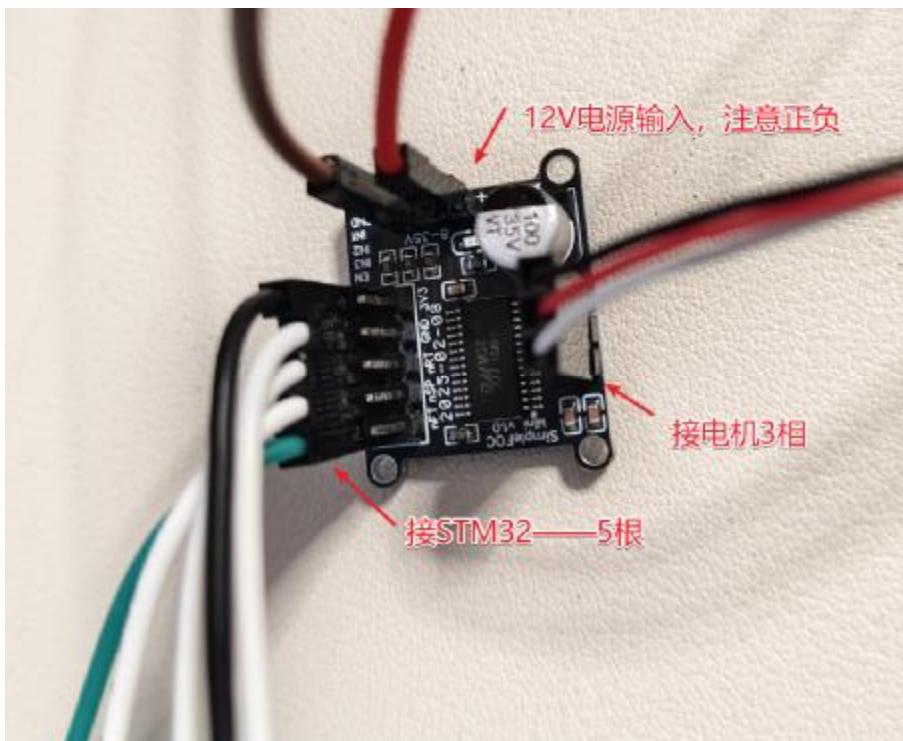
设置到这里就结束了。

三、接线

根据程序定义，接线

Simplefoc mini 板与 STM32 接线：

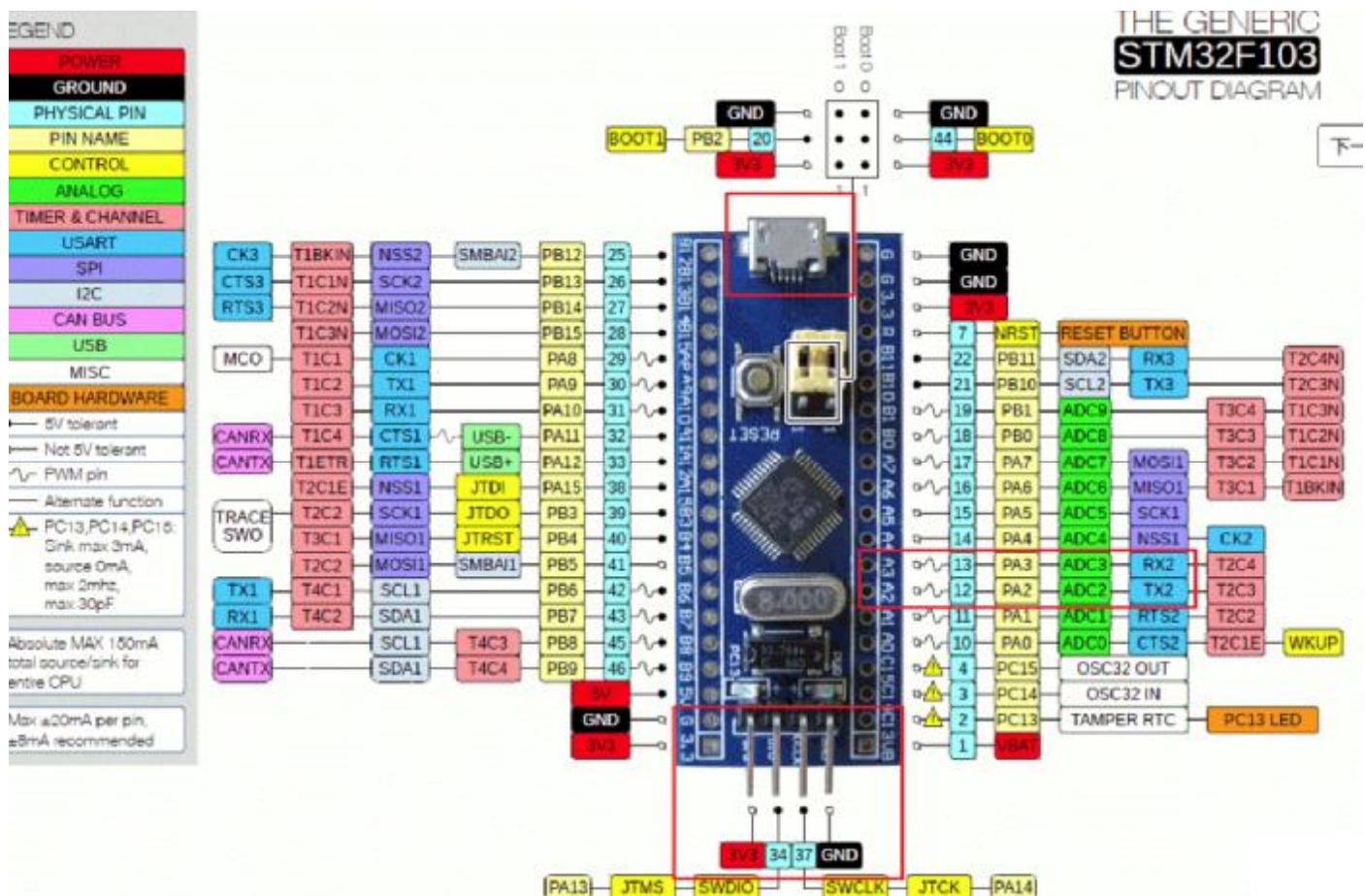
1. In1-2-3 分别接 STM32 板子的 A8,A9,A10，这是三个 PWM 引脚
2. en 接 STM32 板子的 A11，这是使能引脚
3. GND 接 GND



STM32 与 ST-Link 接线

按 STM32 和 ST-link 的丝印接即可

STM32 与 USB 转串口模块接线, TX 接 RX, RX 接 TX



STM32 与编码器接线:

参照程序定义

```
#define I2C_SCL PB6 // SCL 引脚为 PB6
#define I2C_SDA PB7 // SDA 引脚为 PB7
MagneticSensorI2C sensor = MagneticSensorI2C(0x36, 12, 0X0E, 4);
```

SDA 接 STM32 的 PB7

SCL 接 PB6

GND 接 G

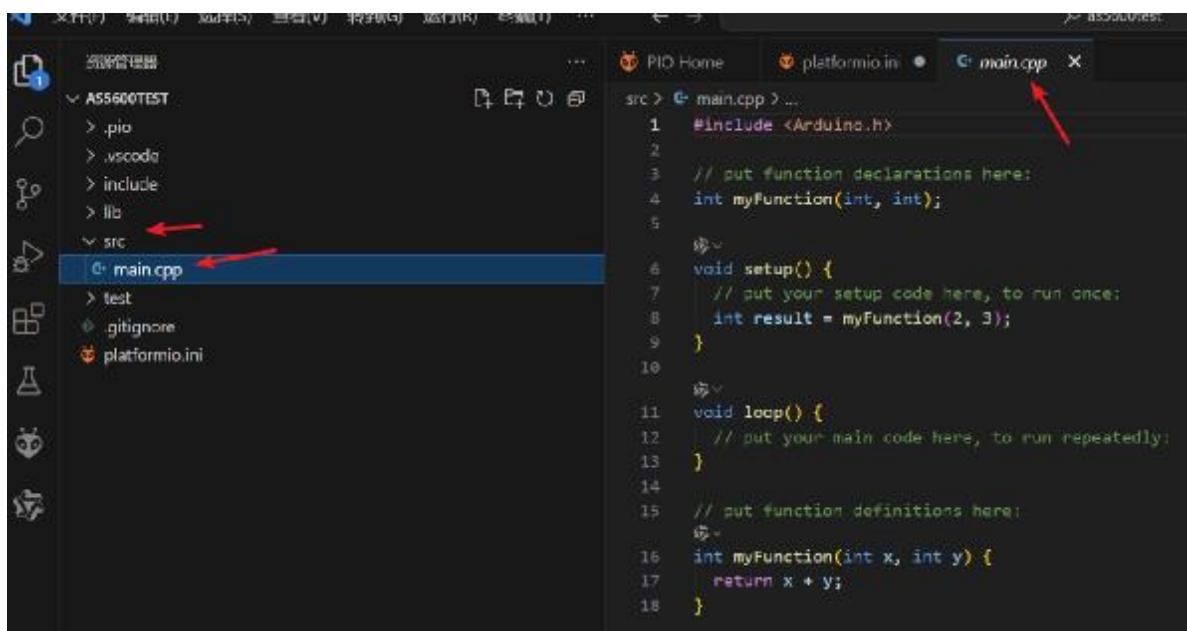
VCC 接 3V3

到此，线已接完。

四、 编程

1. 开始编程

打开默认是这样的



- 按 arduino 格式 C++语言，写入以下代码，代码就不做注释了，可以直接复制粘贴到 AI，如 DEEPSEEK 上，让它帮忙逐行解析。

```
3. #include <SimpleFOC.h>
4.
5. // SPI Magnetic sensor instance (AS5047U example)
6. // MISO PA7
7. // MOSI PA6
8. // SCK PA5
9. // MagneticSensorSPI sensor = MagneticSensorSPI(PA4, 14, 0xFFFF);
10.
11. // I2C Magnetic sensor instance (AS5600 example)
12. // make sure to use the pull-ups!!
13. // SDA PB7
14. // SCL PB6
15. #define I2C_SCL PB6 // SCL 引脚为 PB6
16. #define I2C_SDA PB7 // SDA 引脚为 PB7
17. MagneticSensorI2C sensor = MagneticSensorI2C(0x36, 12, 0X0E, 4);
18.
19. // Motor instance
20. BLDCMotor motor = BLDCMotor(7);
21. // BLDCDriver3PWM(IN1, IN2, IN3, enable(optional))
22. BLDCDriver3PWM driver = BLDCDriver3PWM(PA8, PA9, PA10, PA11);
```

```
23. // BLDCDriver6PWM(IN1_H, IN1_L, IN2_H, IN2_L, IN3_H, IN3_L, enable(optional))
24. // BLDCDriver6PWM driver = BLDCDriver6PWM(PA8, PB13, PA9, PB14, PA10, PB15, PB12);
25.
26. // angle set point variable
27. float target_angle = 0;
28. // instantiate the commander
29. Commander command = Commander(Serial2);
30. void doTarget(char *cmd) { command.scalar(&target_angle, cmd); }
31.
32. void setup()
33. {
34.
35.     Serial2.begin(115200);
36.     // initialise magnetic sensor hardware
37.     // 初始化 I2C, 指定 SCL 和 SDA 引脚
38.     Wire.setSCL(I2C_SCL);
39.     Wire.setSDA(I2C_SDA);
40.     Wire.begin();
41.
42.     sensor.init();
43.     // 等待传感器初始化完成
44.     delay(1000);
45.
46.     Serial2.println("AS5600 initialized successfully!");
47.     // link the motor to the sensor
48.     motor.linkSensor(&sensor);
49.
50.     // driver config
51.     // power supply voltage [V]
52.     driver.voltage_power_supply = 12;
53.     driver.pwm_frequency = 30000; // 30kHz PWM frequency
54.     driver.init();
55.     // link the motor and the driver
56.     motor.linkDriver(&driver);
57.
58.     // choose FOC modulation (optional)
59.     motor.foc_modulation = FOCModulationType::SpaceVectorPWM;
60.
61.     // set motion control loop to be used
62.     motor.controller = MotionControlType::velocity;//速度模式
63.
64.     // controller configuration
65.     // default parameters in defaults.h
66.
67.     // velocity PI controller parameters
68.     motor.PID_velocity.P = 0.1f;
69.     motor.PID_velocity.I = 0.005f;
70.     // maximal voltage to be set to the motor
71.     motor.voltage_limit = 6;
```

```

72.
73. // velocity low pass filtering time constant
74. // the lower the less filtered
75. motor.LPF_velocity.Tf = 0.1f;
76.
77. // angle P controller
78. motor.P_angle.P = 1;
79. // maximal velocity of the position control
80. motor.velocity_limit = 40;
81.
82. // use monitoring with serial
83. Serial2.begin(115200);
84. // comment out if not needed
85. motor.useMonitoring(Serial2);
86.
87. // initialize motor
88. motor.init();
89. // align sensor and start FOC
90. motor.initFOC();
91.
92. // add target command T
93. command.add('T', doTarget, "target angle");
94.
95. Serial.println(F("Motor ready."));
96. Serial.println(F("Set the target angle using serial terminal:"));
97. _delay(1000);
98. }
99.
100. void loop()
101. {
102.     motor.loopFOC();
103.     motor.move(target_angle);
104.     command.run();
105. }
```

就这么多！

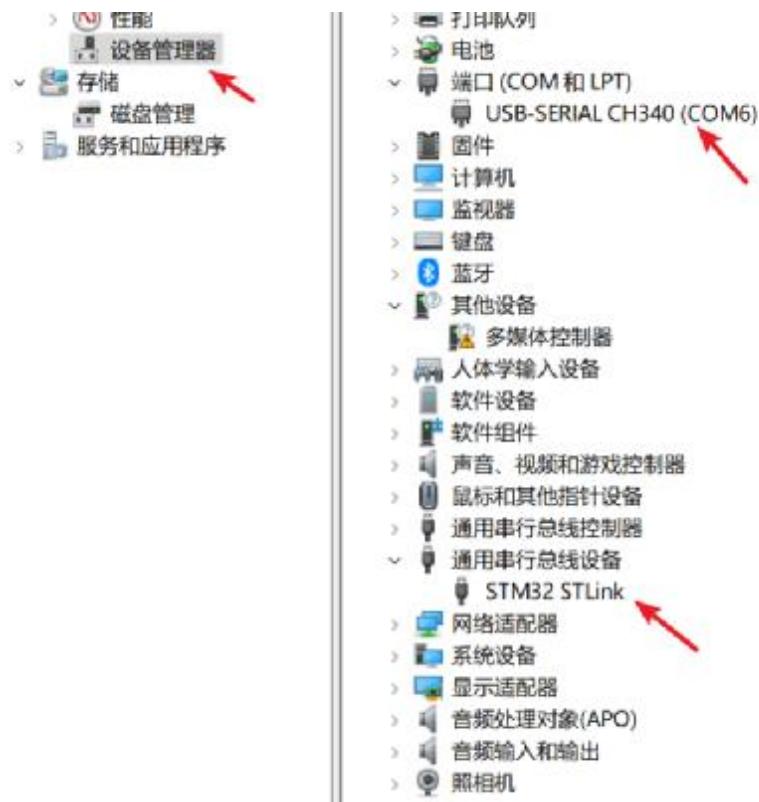
五、 实现

1. 编译烧录

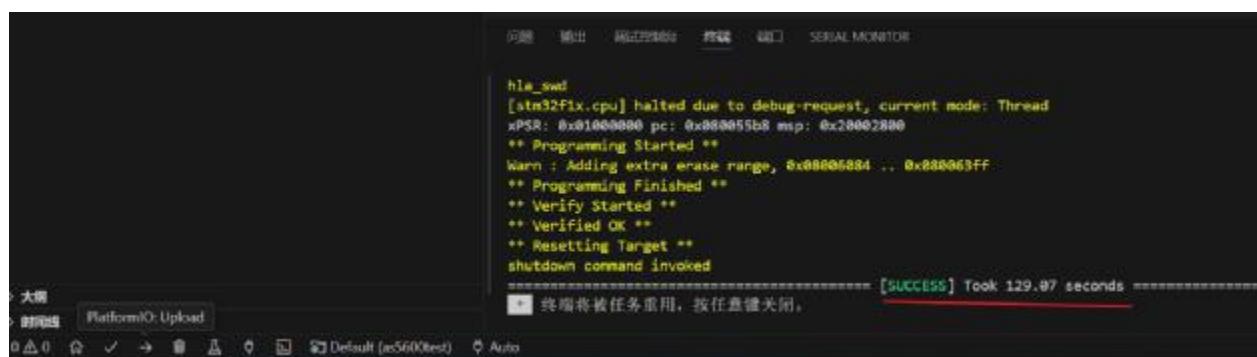


2. 插上 USB, 开始烧录

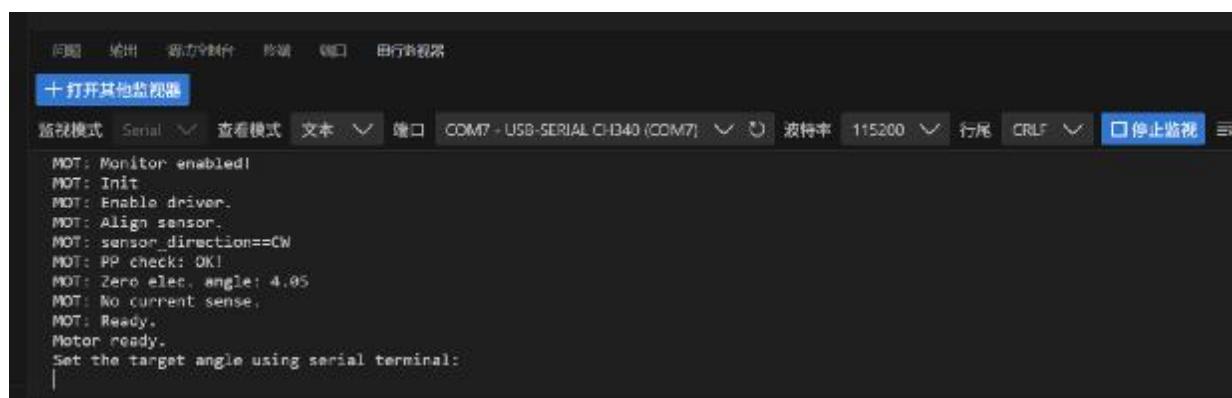
确定驱动安装正确，设备管理器中，可以看到这两个。



3. 烧录成功



打开串口，显示初始化成功
电机会来回摆动校准零位偏差



```
Motor ready.  
Set the target angle using serial terminal:  
---- 已发送 utf8 编码消息: "T10\r\n" ----  
Warn: \r detected!  
10.000  
|  
T15
```

Test - as5600 velocity Auto

输入命令 T+数字

不过这样电机会有尖锐的声音**异响！**这里只是演示一下，验证硬件是否有故障，接线是否正确。

这不是硬件故障的问题，是控制的缘故。

死区未设置，STM32 的默认 PWM 太低等等

改用 ESP32 可以避免此问题，也欢迎有解决方案的朋友留言告诉我，非常感谢！

到此，测试结束！本文档主要针对 M 创动工坊淘宝店提供硬件, mcdgf.taobao.com