

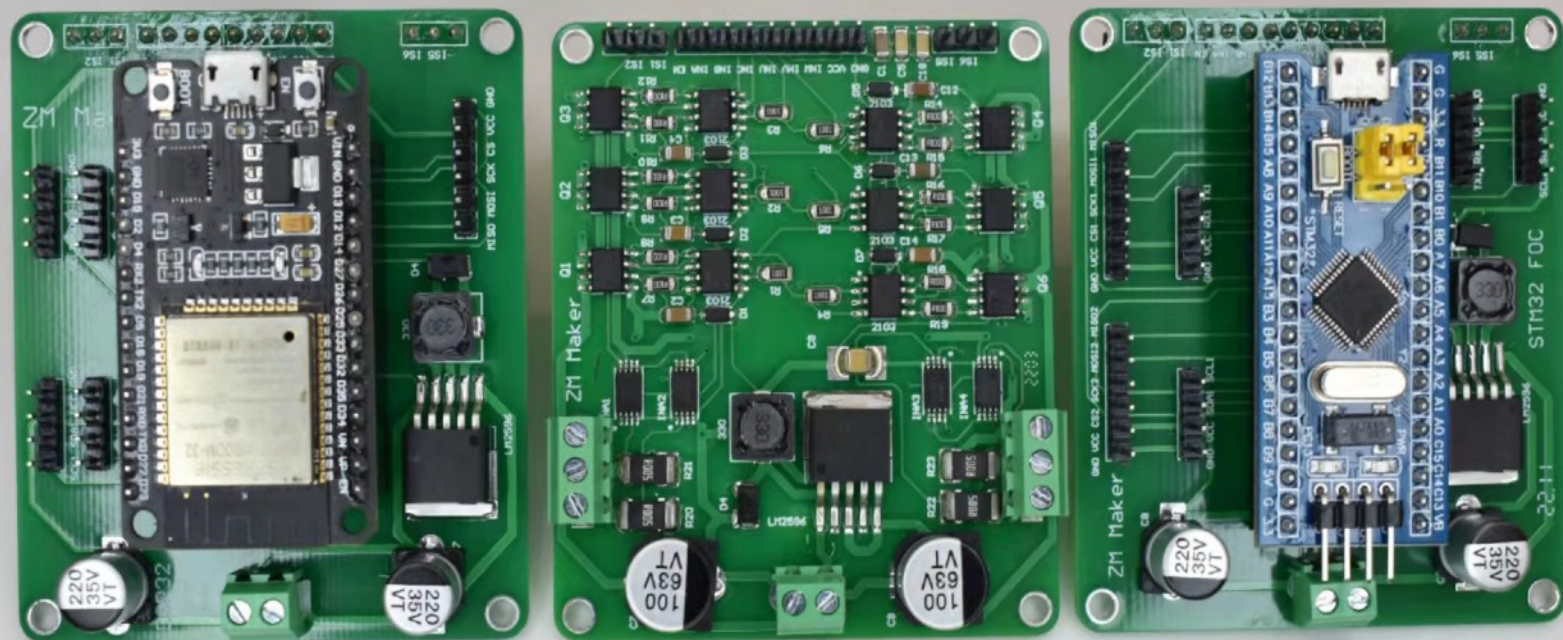
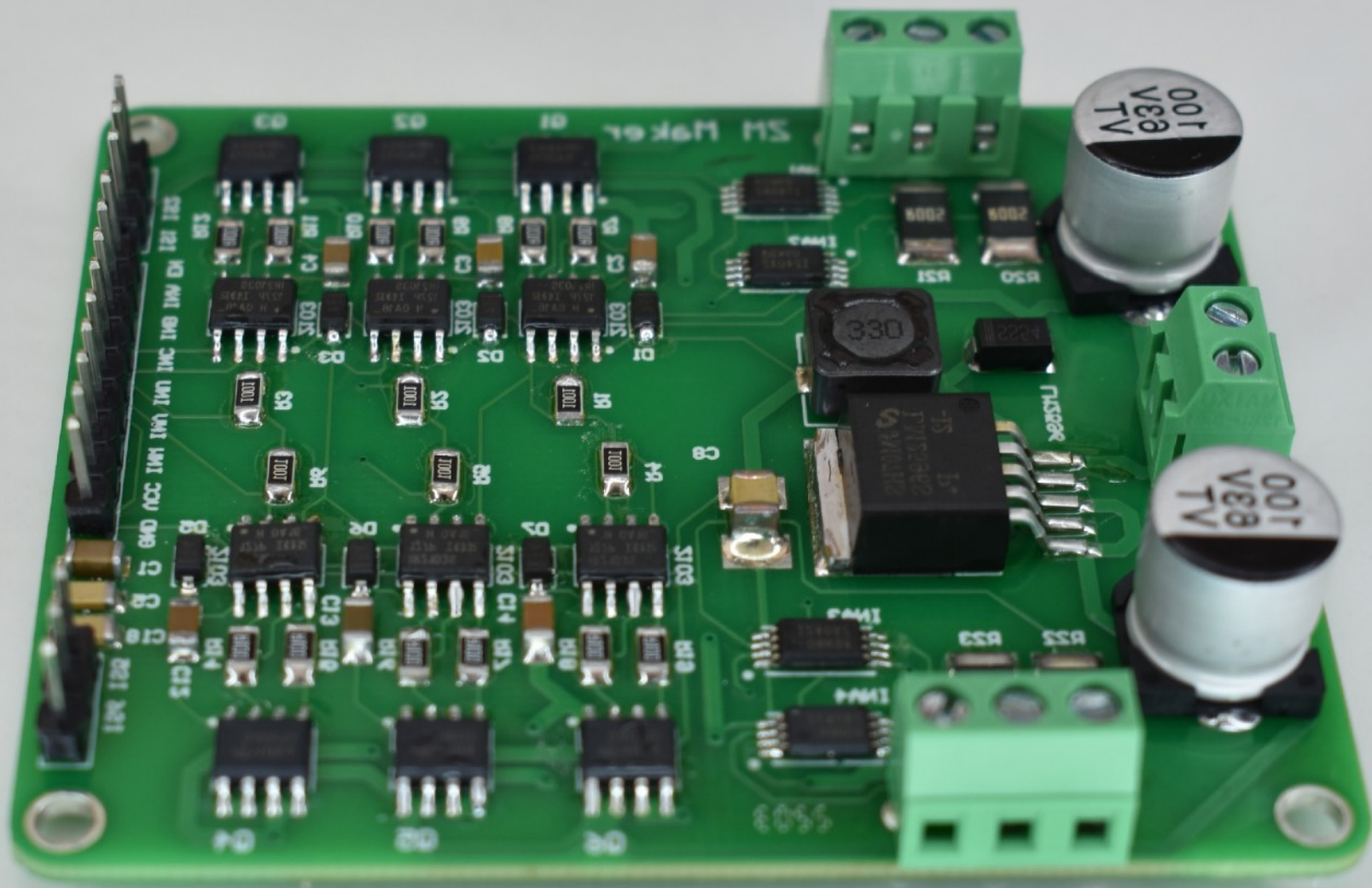
全N-MOS 双路FOC驱动板

支持多种主控：stm32、
esp32、arduino

支持三环：位置环、速度
环、电流环

- 输入电源：12 ~ 36V
- 输出电流：0 ~ 5V
- VCC：3.3 ~ 5V
- 电机KV值：不高于300
- 支持主控：STM32、ESP32、Arduino
- 电流环：双路 + INA240在线电流监测

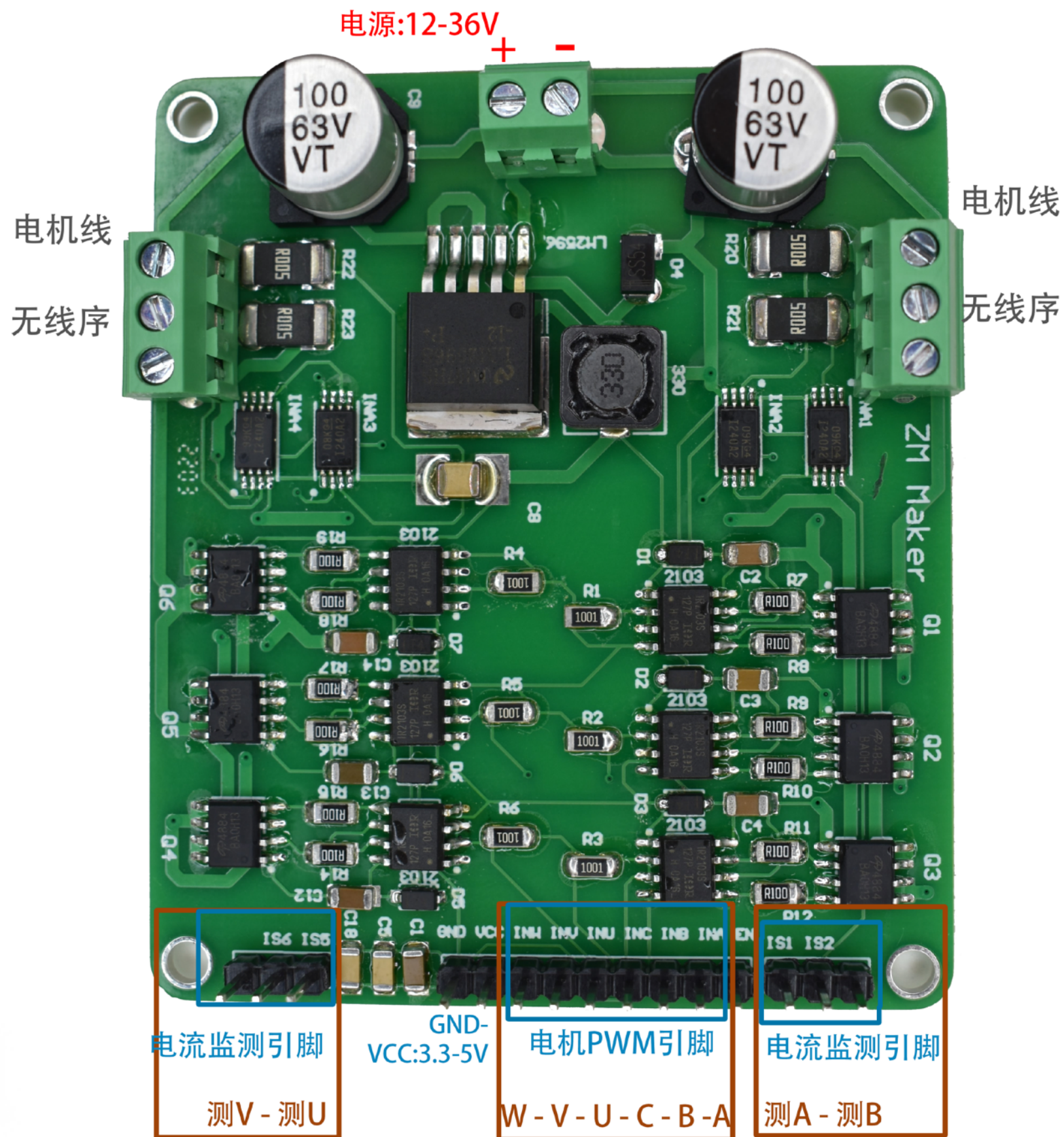
【淘宝店铺：智木创客工作



驱动版的介绍

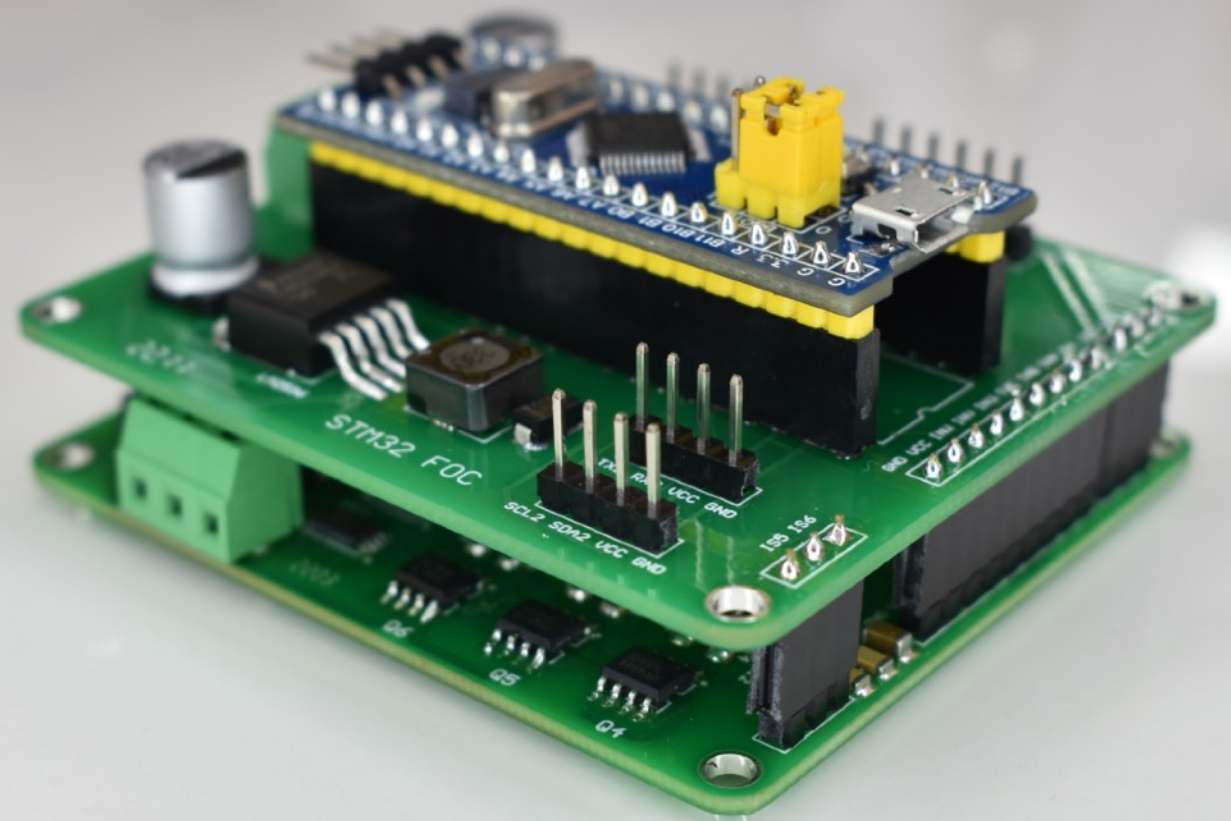
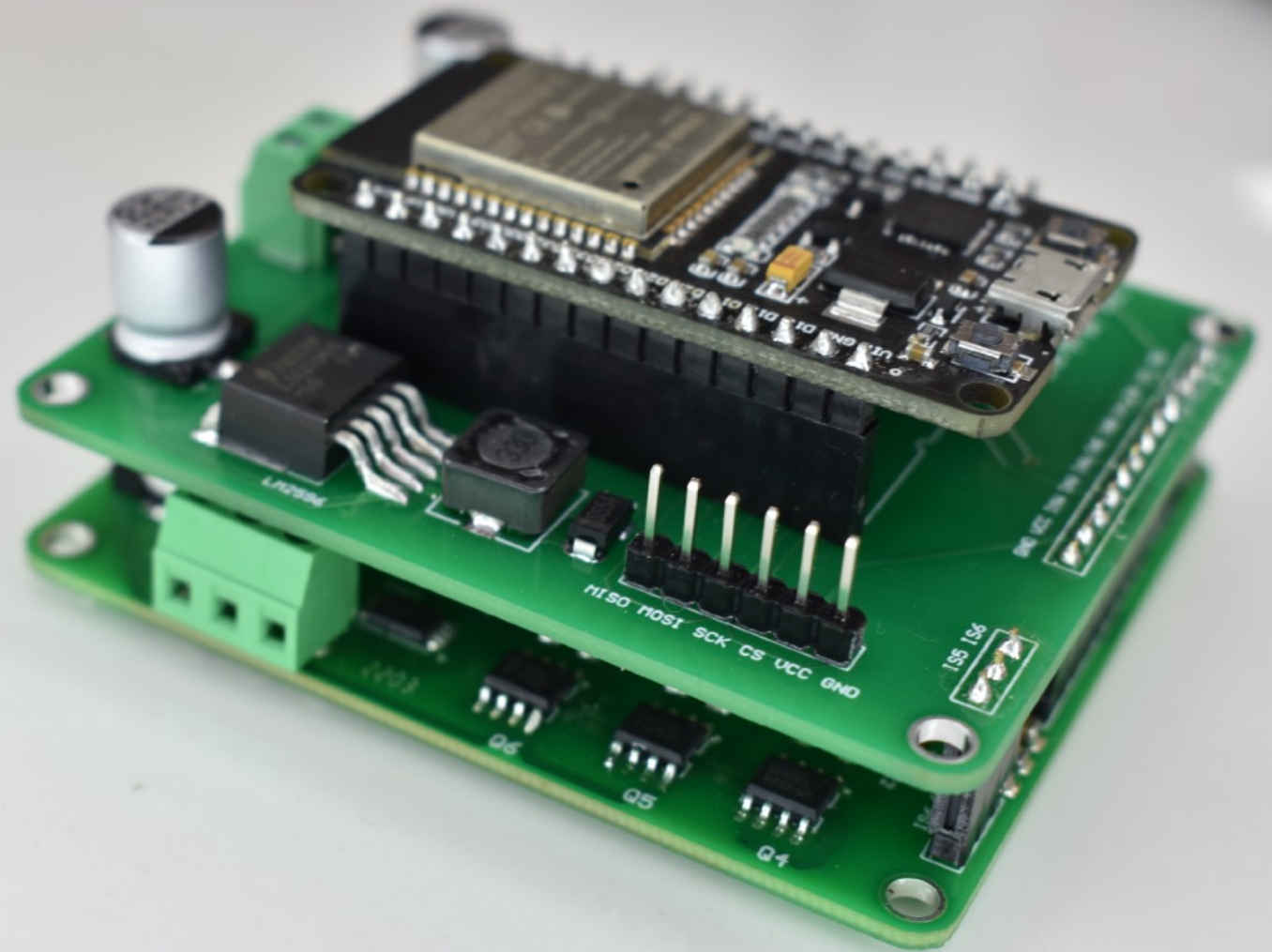
- 第一路电机PWM引脚分别是：INA, INAB, INC
- 第二路电机PWM引脚分别是：INU, INV, INW
- IS1 是 INA 的电流监测引脚
- IS2 是 INB 的电流监测引脚
- IS5 是 INU 的电流监测引脚
- IS6 是 INV的电流监测引脚
- 电机PWM引脚ABC/UVW与程序设计上的ABC/UVW接线——对应
- VCC: 3.3V 或 5V

【淘宝店铺：智木创客工作



主控转接板

- Esp32 (DEVKIT V1)
主控转接板: 2路IIC + 1路
SPI + 12V降压电路
- Stm32 (F103C8T6)
主控转接板: 2路IIC + 2路
SPI + 12V降压电路
- 提供stm32, esp32在
arduinoIDE上的配置教程
以及位置环、速度环、电流
环控制示例例程



arduino及其库的安装

SIMPLEFOC

- 1.打开arduino IDE并通过单击启动
- 2.点击 工具> 管理库...
- 3.搜索“simpleFOC”库并安装。
注意：需要使用我们提供的示例程序，
需要安装2.2.1版本！！
- 4.重新打开IDE，找到库示例：File > Examples > Simple FOC 就可以看到simplefoc的众多示例

Arduino官网：<https://www.arduino.cc>

SimpleFOC库学习文档：

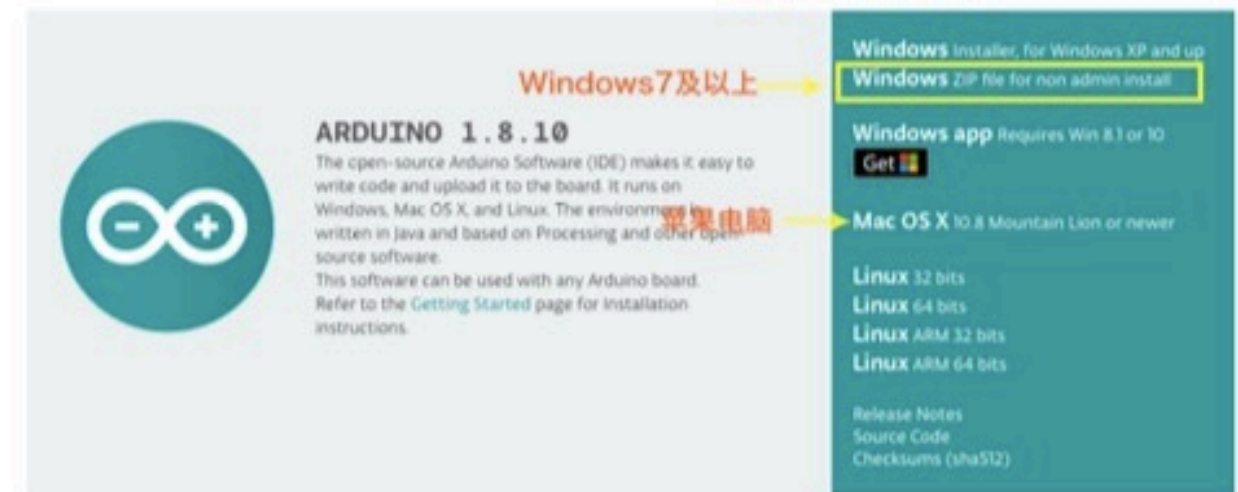
<https://docs.simplefoc.com/>

【淘宝店铺：智木创客工作

IDE的安装

从官网下载 IDE开发环境 (<https://www.arduino.cc/en/Main/Software>)，根据电脑系统选择相应的IDE包。

Download the Arduino IDE



再点击如下图所示的“JUST DOWNLOAD”下载

Contribute to the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). Learn more on how your contribution will be used.



载完毕后解压文件夹，双击**arduino.exe**运行软件。Arduino IDE依赖Java开发环境，需要PC机安装Java的JDK并进行变量配置。如果双击启动失败，可能是PC机无JDK支持。



```
magnetic_sensor_analog_example | Arduino 1.8.13 (Windows Store 1.8.42.0)
文件 编辑 项目 工具 帮助

#include <SimpleFOC.h>

/**
 * Magnetic sensor reading analog voltage on pin A1. This voltage is proportional to rotation position.
 * Tested on AS5600 magnetic sensor running in 'analog mode'. Note AS5600 works better in 'i2C mode' (1
 *
 * MagneticSensorAnalog(uint8_t _pinAnalog, int _min, int _max)
 * - pinAnalog      - the pin that is reading the pwm from magnetic sensor
 * - min_raw_count  - the smallest expected reading. Whilst you might expect it to be 0 it is often ~15
 * - max_raw_count  - the largest value read. whilst you might expect it to be 2^10 = 1023 it is often
 */
MagneticSensorAnalog sensor = MagneticSensorAnalog(A1, 14, 1020);

void setup() {
  // monitoring port
  Serial.begin(115200);

  // initialise magnetic sensor hardware
  sensor.init();

  Serial.println("Sensor ready");
  _delay(1000);
}

void loop() {
  // display the angle and the angular velocity to the terminal
  Serial.print(sensor.getAngle());
  Serial.print("\t");
  Serial.println(sensor.getVelocity());
}
```

以 ESP32+AS5600_IIC通信 为例： 读取编码器信息

1.打开示例： 文件 -> 示例 ->

Simple FOC -> utils -> sensor_test ->
magnetic_sensors -> magnetic_sensor_i2c->
magnetic_sensor_i2c_dual_bus_examples -
>esp32_i2c_dual_bus_example

打开后的示例如左图所示

2.我们对照驱动版简介，将编码器线
VCC,GND,SDA,SCL 正确接线接线

3.确认接线之后，点击： 工具 -> 开发版 ->
ESP32 Arduino -> DOIT ESP32 DEVKIT
V1； 端口 ->选择相应com口



设置好后，将示例代码上传

4.上传成功之后，检查接线，无误后接上电源
线，点击： 工具 -> 串口监视器

将波特率改与程序一致，示例为 115200

转动电机，观察串口监视器输出的弧度值数据

【淘宝店铺： 智木创客工作

双路板-simplefoc-编码器读值：

- 1.在开发的主控扩展板上，我们可以将ESP32做为主控。提供了 双路 IIC 读值方式，也单独引出来了一路SPI引脚：
SCL1-23, SDA1-19 ; SCL2-22, SDA2-21 。
- 2.在开发的主控扩展板上，我们可以将STM32C8T6做为主控。提供了双路 IIC 读值方式，双路 SPI 读值方式。支持AS5047, AS5600：
SCL1-PB6, SDA1-PB7; SCL2-PB10, SDA2-PB11;
MISO1-PB14, MOSI1-PB15, SCK1-PB13, CS1-PB12;
MISO2-PB4, MOSI2-PB5, SCK2-PB3, CS2-PA15
串口下载TX1-PA9, RX1-PA10
串口通信 TX2-PA2, RX2-PA3

程序示例中，有提供esp32, stm32的各个读值例程，请查阅

双路板-simplefoc-开环控制：

- 开环控制容易发烫，只适合用来测试电机。
- 1.在主控转接板上，我们将ESP32做为主控。第一路电机控制引脚（25，26，27）；第二路电机控制引脚（32，33，4）。
- 2.在主控转接板上，我们将STM32C8T6做为主控。第一路电机控制引脚（PA6，PA7，PB8）；第二路电机控制引脚（PA0，PA1，PB9）。

程序示例中，有提供nano，esp32，stm32的各个电机例程，
请查阅

若使用官方例程，需更自己更改电机引脚，磁极对数（7）（我们提供的电机磁极对数为7），来与我们的板子对应上哈。

双路板-simplefoc-闭环控制：

- 闭环控制上在编码器读值和开环控制基础上实现的。
- 假若闭环控制失败，则去检查开环控制电机是否正常，读取编码器是否正常。若电机正常，编码器读值正常，那么检查电机和编码器在程序上、硬件接线上 是否对应正确。
- 程序示例中，有提供nano，esp32，stm32的各个电机例程，请查阅
- 注意下：串口控制时，需要在弧度值前加“T”，格式：T6.28
- 串口监视器右下角设置： 换行符 — 115200 波特率

若使用官方例程，需更自己更改电机引脚、编码器引脚，磁极对数 来与我们的板子对应上哈。除此，还要更改PID参数，将I值和D值 设置为0.电机运行无误后，再去调整PID参数。


```

// velocity PI controller parameters
motor.PID_velocity.P = 0.09;
motor.PID_velocity.I = 1.70;
motor.PID_velocity.D = 0.0005;
// maximal voltage to be set to the motor
motor.voltage_limit = 6;
// velocity low pass filtering time constant
// the lower the less filtered
motor.LPF_velocity.Tf = 0.04;

// angle P controller
motor.P_angle.P = 20;
// maximal velocity of the position control
motor.velocity_limit = 20;

// use monitoring with serial
Serial.begin(115200);
// comment out if not needed
motor.useMonitoring(Serial);

```

【淘宝店铺：智木创客工作

电机抖动，调整PID值

1.先将motor.PID_velocity.P=0； motor.PID_velocity.I=0；
motor.PID_velocity.D=0；

2.将 P 慢慢增大，找到最佳响应最佳状态。

3.再调整I值，使电机的转动更加顺滑，无抖动。调整效果比较好时，可无需调整D值。

比例控制有个缺点，就是会产生余差，要克服余差就必须引入积分作用。比例控制有个缺点，就是会产生余差，要克服余差就必须引入积分作用。

积分时间小，表示积分速度大，积分作用就强；反之，积分时间大，则积分作用就弱。如果积分时间无穷大，表示没有积分作用，控制器就成为纯比例控制器。只有偏差不存在时，积分才会停止。

微分作用主要是用来克服被控对象的滞后，常用于温度控制系统

更加详细的资料就请去上网找资料学习吧；

PID值的调节需要经验，慢慢调整找最佳状态

PID常用口诀

参数整定找最佳，从小到大顺序查，
先是比例后积分，最后再把微分加，
曲线振荡很频繁，比例度盘要放大，
曲线漂浮绕大湾，比例度盘往小扳，
曲线偏离回复慢，积分时间往下降，
曲线波动周期长，积分时间再加长，
曲线振荡频率快，先把微分降下来，
动差大来波动慢，微分时间应加长，