

深圳市必趣科技有限公司  
**BIG TREE TECH**

---

**BIGTREETECH**  
**42 步进电机闭环驱动控制板**  
**使用说明书**

# 目录

一、产品简介.....	3
二、产品参数.....	3
三、产品特点.....	3
四、接口说明.....	4
五、按键功能.....	5
六、参数调节.....	5
1.OLED 按键调节方法.....	5
2.命令行调节方法.....	7
2.1 驱动异常解决方法.....	8
2.2 串口调试指南.....	9
七、常见问题解答.....	19
八、注意事项.....	20
九、所有资料下载地址.....	21

## 一、产品简介

步进电机闭环驱动使用了闭环控制技术。以 ATSAM D 21G18A 为主控制器，A4954 作为电机驱动器，AS5047D 为磁性编码器。闭环驱动是通过把步进电机的转动角度，反馈给控制板，把需要转动的距离和实际转动的距离进行比较，计算出误差值，然后进行补偿，从而达到防止多步和丢步的问题。该闭环驱动可以彻底克服开环步进电机的丢步问题，同时也能明显的提升电机高速工作时的性能，从而提升机器的加工速度和精度。电机在长时间工作造成失步时，主控芯片可以通过磁性编码器检测出步进电机的失步情况，从而进行补偿，使机器可以继续正常工作。

## 二、产品参数

电机电源 ( VM ) : 12V---24V

电机默认工作电流 : 800mA

电机默认待机电流 : 500mA

驱动细分默认 : 16 细分

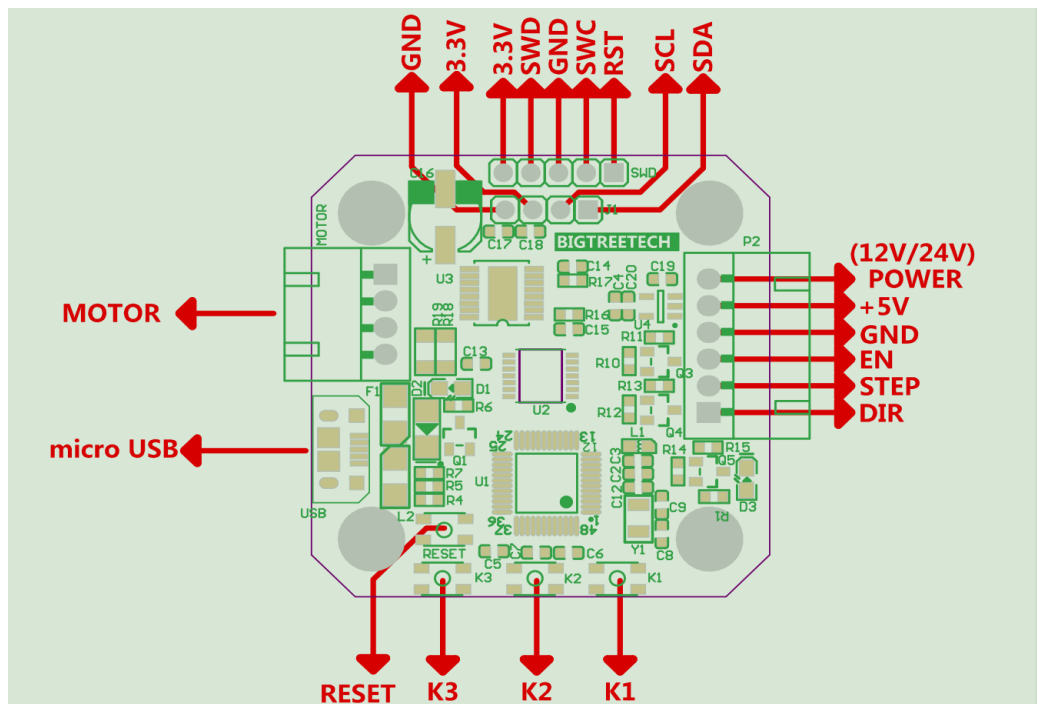
细分步数 : 1   2   4   8   16   32   64   128   256

最大电流 : 2000mA

### 三、产品特点

- 1、产品整套出售，省去繁杂的组装和调试，即插即用。
- 2、可以在高速打印下不丢步。
- 3、发热量低，运行效率高。
- 4、安装方便，易于接线。
- 5、和普通驱动相比，电机运行更加稳定。
- 6、有 OLED 的情况下，可通过按键修改电机的电流和细分，操作简单，也可使用 USB 进行参数调节。
- 7、无 OLED 的情况下，可用 USB 连接电脑，通过命令行进行参数调节。

### 四、接口说明



## 五、按键功能

RESET：复位按键

K1：确认选择（仅在设置界面有效）

## K2：切换设置界面和状态界面

K3：选中下一行选项（仅在设置界面有效）

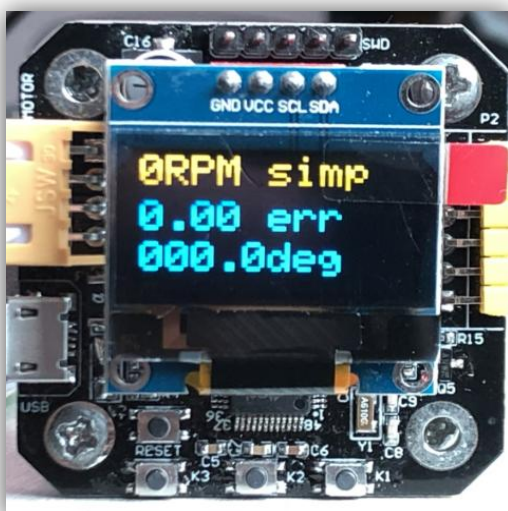
## 六、参数调节

注：42 步进电机闭环驱动控制板有两种参数调节方法

## 1、OLED 按键调节方法

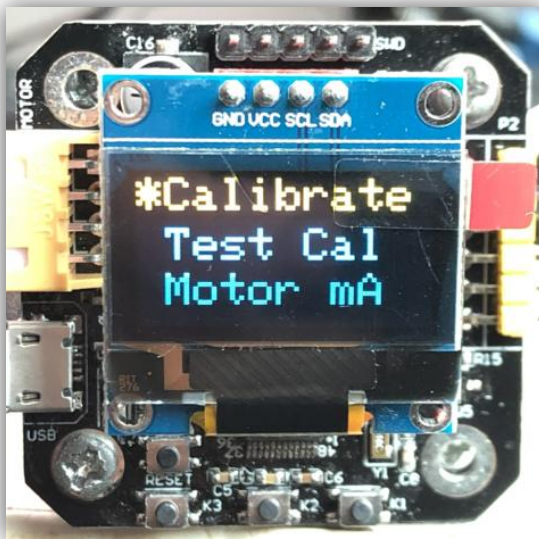
优点：不需要连接电脑，通过 OLED 显示，按键设置，操作简单直观。

## 状态界面



- 1) .第一行 0 RPM , 代表当前转数 0 转/分钟。
- 2) .第二行 0.00 err , 代表电机实际位置和目标位置的误差为 0°
- 3) 第三行 000.0deg , 代表电机当前目标位置为 0°

## 设置界面



设置界面总共有 7 个选项

### **Calibrate:**

电机转动 200 次 ( 每次 1.8° ) 来对编码器进行校准 , 出厂首次上电后 , 必须要校准参数。

### **Test Cal:**

测试 calibrate 并以度为单位报告最大误差

### **Motor mA**

设置电机运行时的电流，默认 800MA

## Hold mA

设置电机待机时的电流，默认 500MA

## Microstep,

设置细分，默认 16 细分

## EnablePin

设置使能信号的逻辑电平，Enable 代表高电平使能，!Enable 代表低电平使能，默认 !Enable 低电平使能

## DirPin

设置 dir 信号的逻辑电平，修改此选项可以更改电机的转动方向

## 按键

RESET: 复位键

K2: 切换设置界面和状态界面

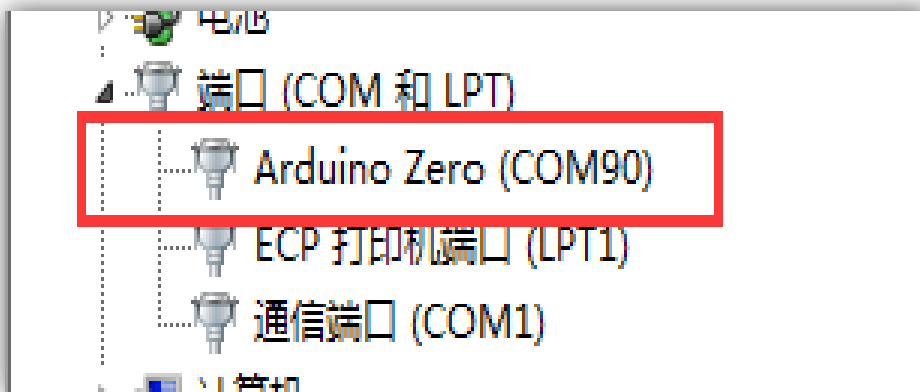
K3: 选中下一行选项(仅在设置界面有效)

K1: 确认选择(仅在设置界面有效)

## 2、命令行调节

优点：可设置的参数更详细

连接电脑后，查看驱动是否安装就绪



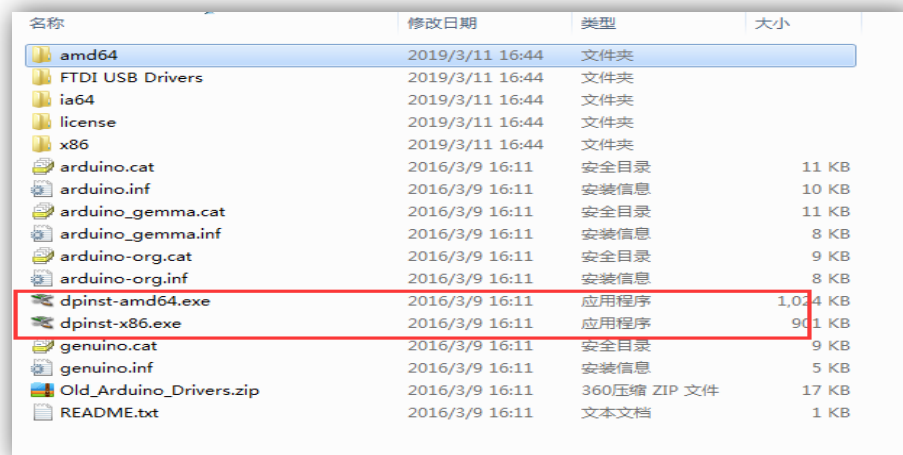
如果显示正常，则不需要安装以下驱动。

### 2.1、驱动异常解决方法

驱动下载地址：

<https://github.com/bigtreotech/BIGTREETECH-SKR-V1.3>

根据电脑的系统选择安装驱动程序。



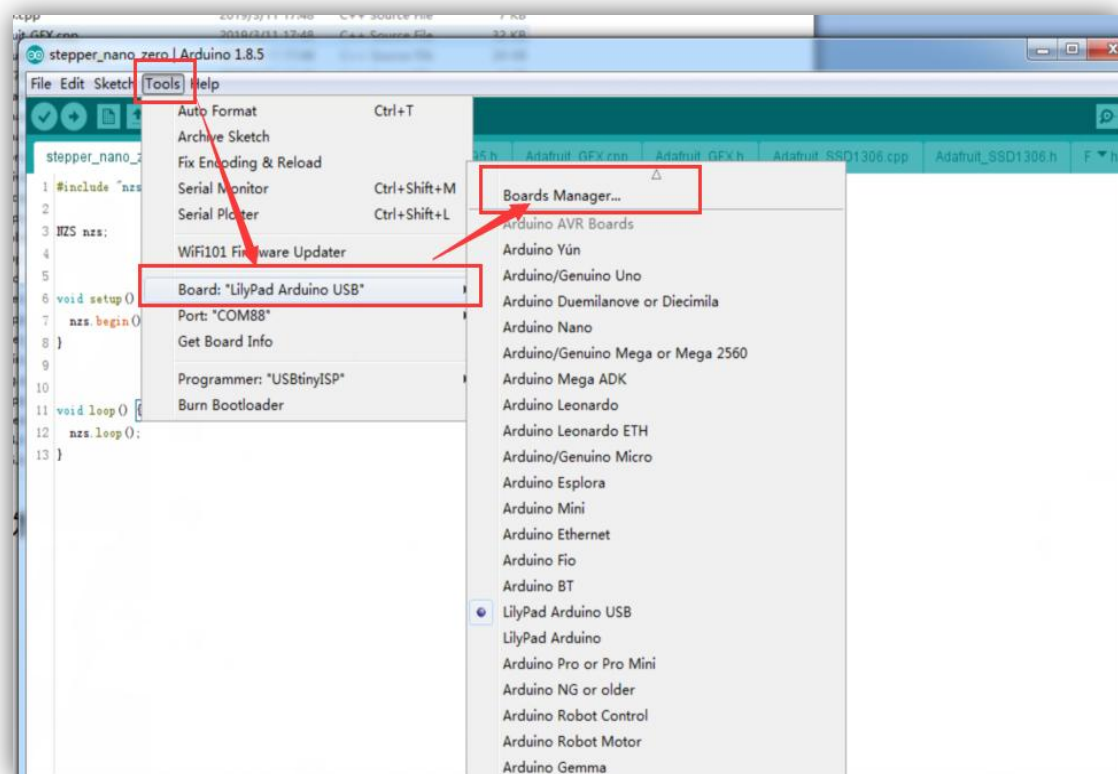


## 2.2 串口调试指南

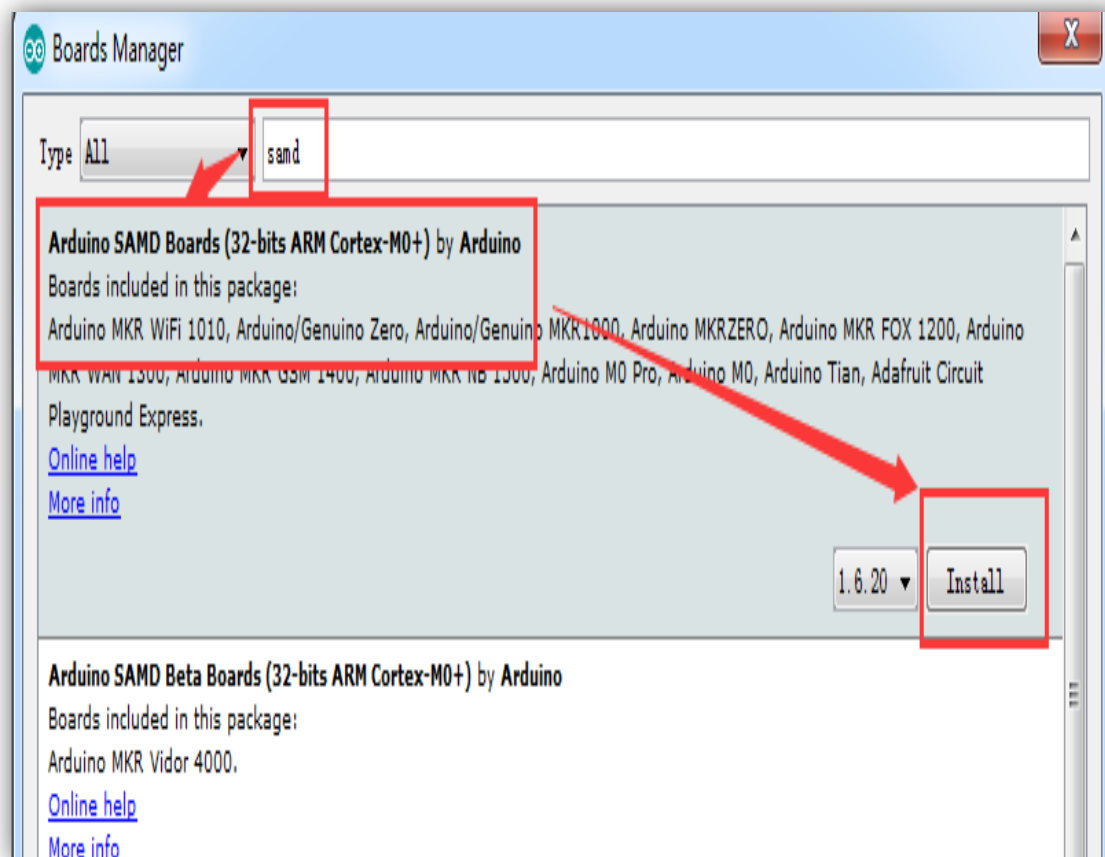
### ①驱动准备就绪后，用 arduino IDE 打开固件

名称	修改日期	类型	大小
stepper_nano_zero.ino	2019/3/11 17:48	Arduino file	1 KB
glcdfont.c	2019/3/11 17:48	C Source File	9 KB
A4954.cpp	2019/3/11 17:48	C++ Source File	10 KB
A5995.cpp	2019/3/11 17:48	C++ Source File	7 KB
Adafruit_GFX.cpp	2019/3/11 17:48	C++ Source File	32 KB
Adafruit_SSD1306.cpp	2019/3/11 17:48	C++ Source File	24 KB
as5047d.cpp	2019/3/11 17:48	C++ Source File	8 KB
calibration.cpp	2019/3/11 17:48	C++ Source File	13 KB
command.cpp	2019/3/11 17:48	C++ Source File	9 KB
commands.cpp	2019/3/11 17:48	C++ Source File	37 KB
eeprom.cpp	2019/3/11 17:48	C++ Source File	8 KB
fet_driver.cpp	2019/3/11 17:48	C++ Source File	36 KB
Flash.cpp	2019/3/11 17:48	C++ Source File	5 KB
ftoa.cpp	2019/3/11 17:48	C++ Source File	4 KB
nonvolatile.cpp	2019/3/11 17:48	C++ Source File	5 KB
nzs.cpp	2019/3/11 17:48	C++ Source File	19 KB
nzs_lcd.cpp	2019/3/11 17:48	C++ Source File	11 KB
planner.cpp	2019/3/11 17:48	C++ Source File	5 KB
sine.cpp	2019/3/11 17:48	C++ Source File	12 KB
stepper_controller.cpp	2019/3/11 17:48	C++ Source File	40 KB
steppin.cpp	2019/3/11 17:48	C++ Source File	7 KB
syslog.cpp	2019/3/11 17:48	C++ Source File	6 KB
utils.cpp	2019/3/11 17:48	C++ Source File	2 KB
.cproject	2019/3/11 17:48	CPROJECT 文件	13 KB
A4954.h	2019/3/11 17:48	H 文件	3 KB
A5995.h	2019/3/11 17:48	H 文件	2 KB

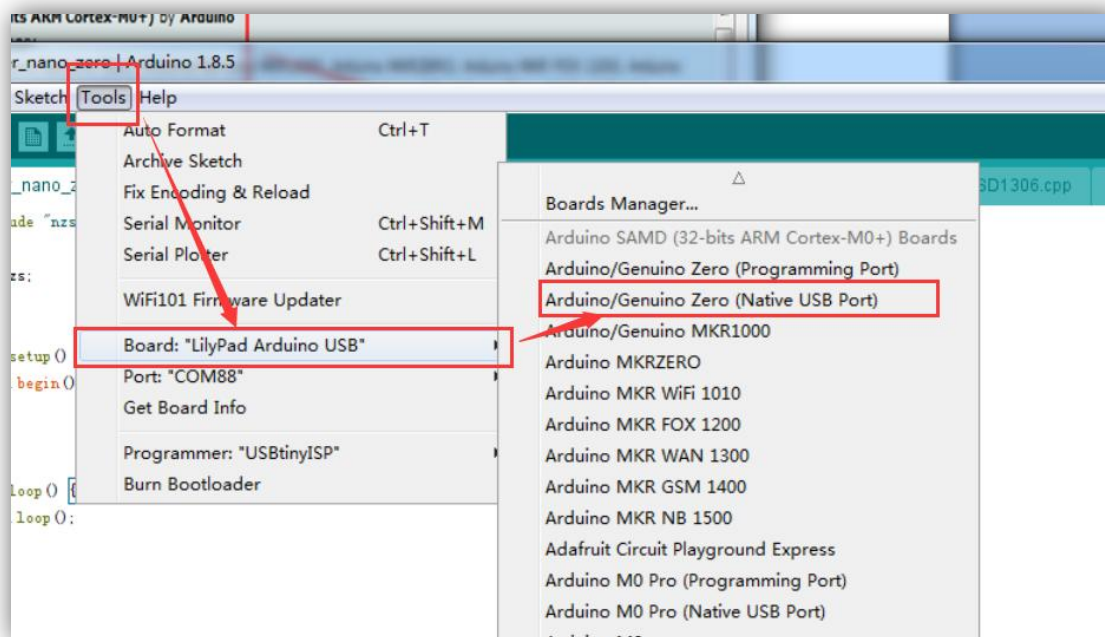
### ②安装开发板



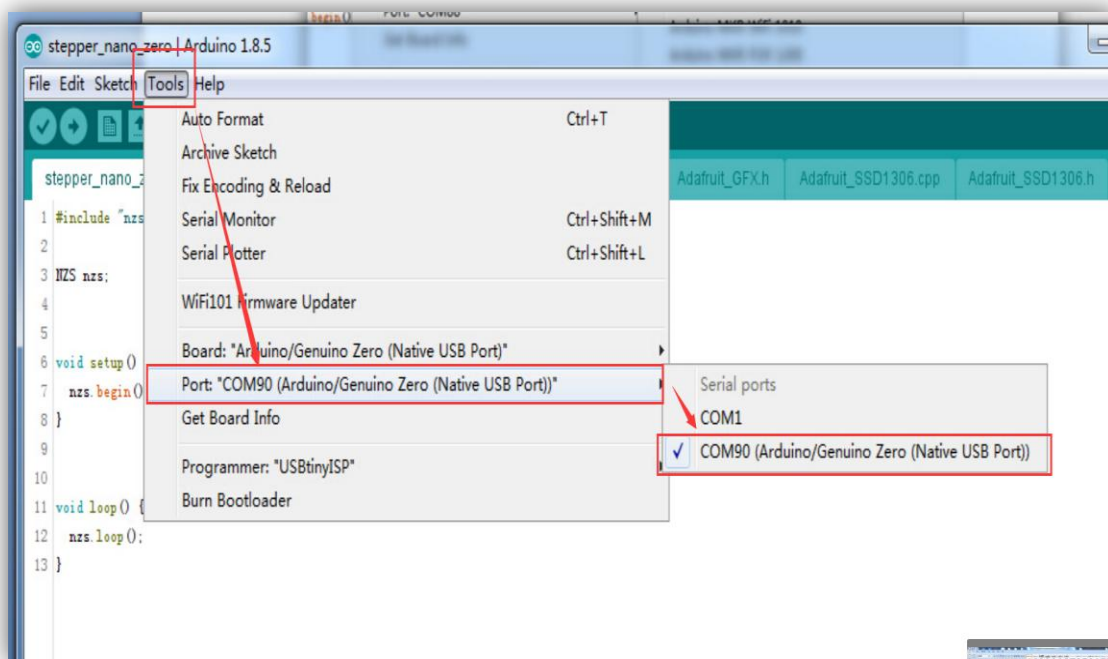
### ③输入 “samd”



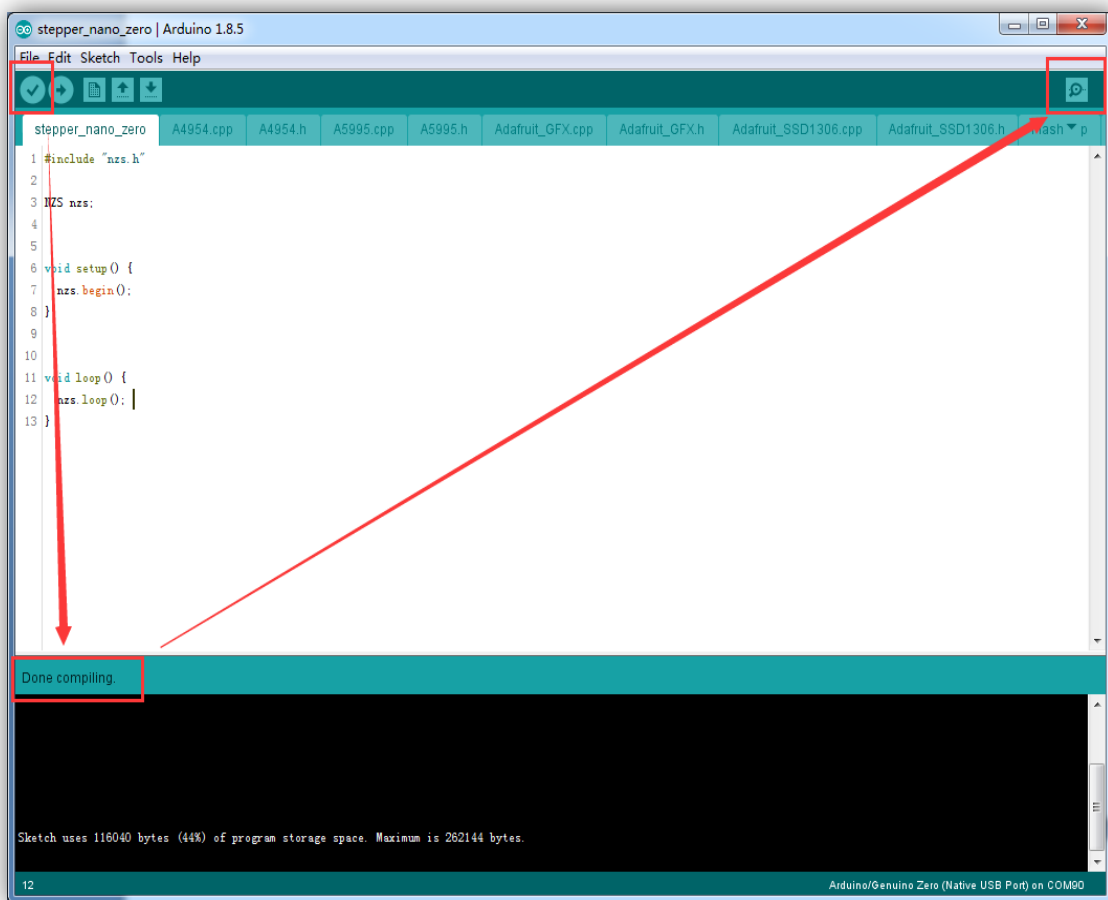
### ④选择开发板



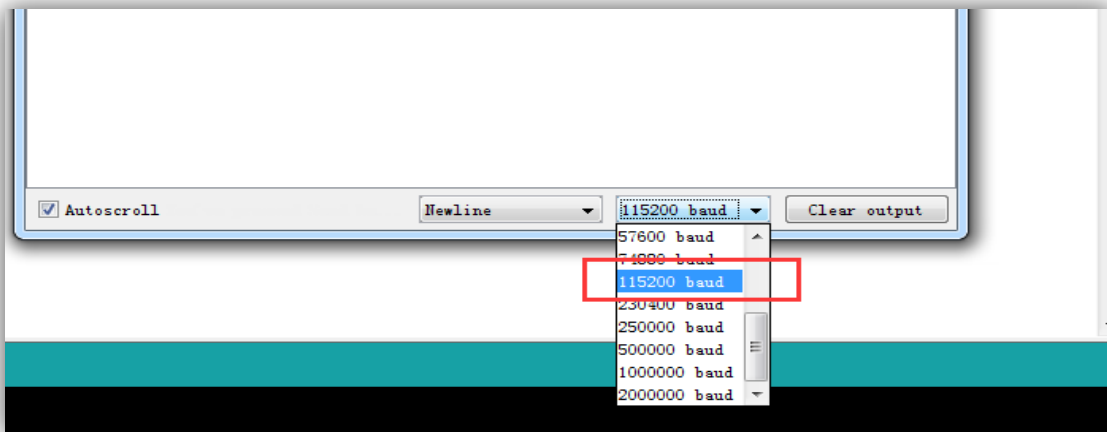
## ⑤选择端口



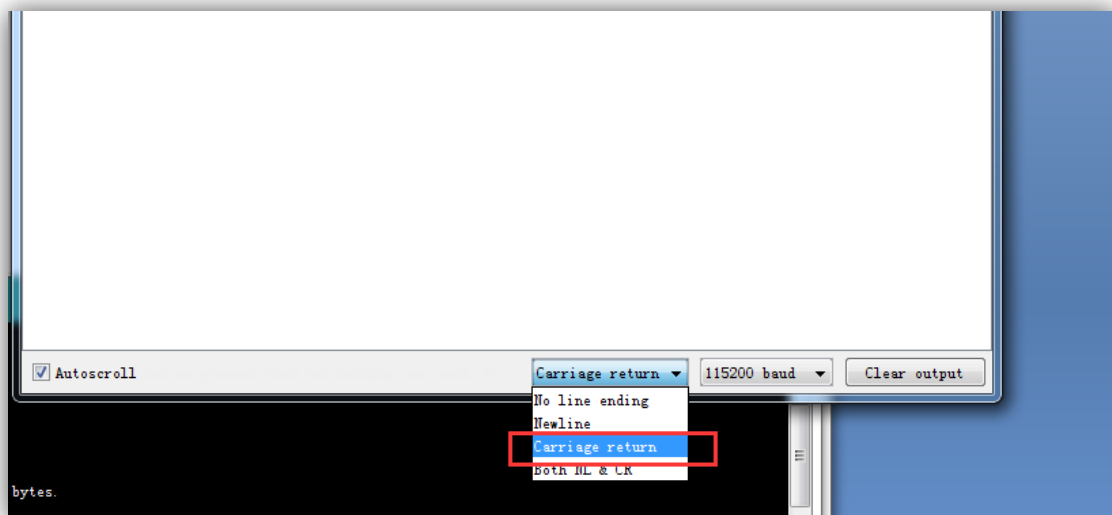
## ⑥编译无误-&gt;打开串口监视器



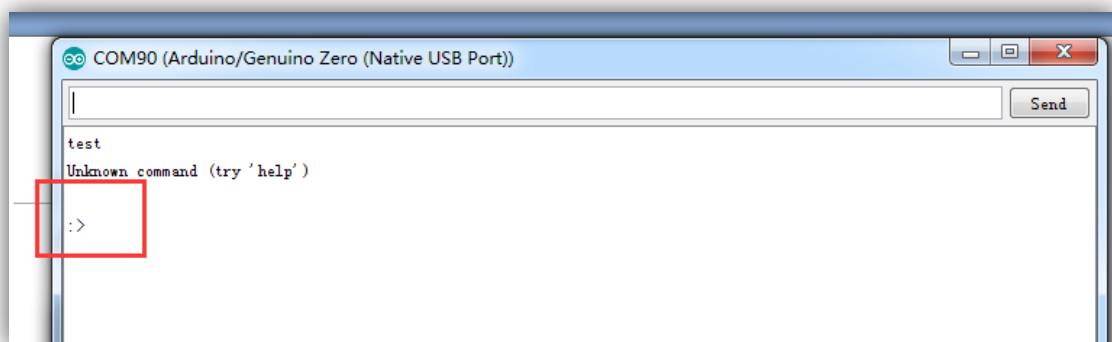
### ⑦将波特率设置为 115200



### ⑧选择 “carriage return”



### ⑨随便发送一个命令例如 “test”，当返回 “:>” 时，代表现在可以发送新命令



## ⑩发送 “help” 将返回所有命令列表

```

test
Unknown command (try 'help')

:>help
help - Displays this message
calibrate - Calibrates the encoder, should be done with motor disconnected from machine
getcal - Prints the calibration table
testcal - tests the calibration of the encoder
microsteps - gets/sets the microstep size, example 'microsteps 16'
step - Steps motor one step, optionally direction can be set is 'step 1' for reverse
feedback - enable or disable feedback controller, 'feedback 0' - disables, 'feedback 1' - enables
readpos - reads the current angle as 16bit number, applies calibration if valid
encoderdiag - Prints encoder diagnostic
spid - with no arguments prints SIMPLE PID parameters, with arguments sets PID 'sPID Kp Ki Kd' Where Kp,Ki,Kd are floating point numbers
vpid - with no arguments prints VELOCITY PID parameters, with arguments sets PID 'sPID Kp Ki Kd' Where Kp,Ki,Kd are floating point numbers
ppid - with no arguments prints POSITIONAL PID parameters, with arguments sets PID 'sPID Kp Ki Kd' Where Kp,Ki,Kd are floating point numbers
dirpin - with no arguments read dirpin setting, with argument sets direction pin rotation
enablepinmode - gets/sets the functionality of the enable pin
errorlimit - gets/set the error limit which will assert error pin (when error pin is set for error output)
ctrlmode - gets/set the feedback controller mode of operation
maxcurrent - gets/set the maximum motor current allowed in milliAmps
holdcurrent - gets/set the motor holding current in milliAmps, only used in the simple positional PID mode
homecurrent - gets/set the motor moving and holding currents that will be used when pin A3 is low
motorwiring - gets/set the motor wiring direction, should only be used by experts
stepsperrotation - gets/set the motor steps per rotation, should only be used by experts
boot - Enters the bootloader
move - moves encoder to absolute angle in degrees 'move 400.1'
velocity - gets/set velocity in RPMs
factoryreset - resets board to factory defaults
stop - stops the motion planner
setzero - set the reference angle to zero
data - enables/disables binary data output
looptime - returns the control loop processing time
eepromerror - returns error in degrees from eeprom at power up relative to current encoder
eepromloc - returns location in degrees eeprom on power up
eepromwrite - forces write of location to eeprom
setpos - sets the current angle in degrees
reboot - reboots the unit
eepromsetloc - sets the device angle based on EEPROM last reading, compenstates for error
homepin - sets the pin used to drop to homing current
homeangledelay - sets the angle delay in dropping to homing current
home - moves the motor until home switch (enable pin) is pulled low. example 'home 360 0.5' move up to 360 degrees at 0.5 RPM
pinread - reads pins as binary (bit 0-step, bit 1 - Dir, bit 2 - Enable, bit 3 - Error, bit 4 - A3, bit 5- IX, bit 6 - RX
errorpin - Sets the logic level of error pin
geterror - gets current error
getsteps - returns number of steps seen
debug - enables debug commands out USB

:>

```

如果出现以上数据，则表示连接成功，可以发送下面的命令修改参数。

## help

help 命令将返回支持的所有命令列表。

## getcal

该命令将打印出 200 点校准表。如果您正在进行固件开发并且不希望每次更新固件时都进行校准，这将非常有用。您可以将此表复制到 nonvolatile.cpp 文件中

## calibrate

电机转动 200 次(每次  $1.8^\circ$ )来对编码器进行校准

## testcal

测试 calibrate 并以度为单位报告最大误差

## step

这将使电机顺时针移动一步，转动的角度基于当前的细分设置。使用 “step 0 16” 可以使电机顺时针移动 16 步，使用 “step 1 16” 可以使电机逆时针移动 16 步。

## feedback

此命令用来设置控制板处于开环还是闭环状态。

## Readpos

以度数为单位报告电机当前位置

## **encoderdiag**

报告 AS5047D 编码器的内部寄存器，便于检查编码器是否有问题。

## **microsteps**

此命令获取/设置当前的细分

## **spid**

此命令用于设置 simple positional PID 模式下的 Kp，Ki 和 Kd 参数。

## **ppid**

此命令用于设置 positional PID 模式下的 Kp，Ki 和 Kd 参数。

## **vpid**

此命令用于设置 velocity PID 模式下的 Kp，Ki 和 Kd 参数。

## **velocity**

此命令用于设置 velocity PID 模式下电机旋转的速度

## **boot**

该命令将使微处理器进入引导加载程序模式。(也可以通过双击 RESET 按键来完成)

## **factoryreset**

这将擦除校准和设置的所有参数，重置为出厂设置。执行此命令后，需要再次对电机进行校准。

## **Dirpin**

“ dirpin 0” 将设置电机在 dir 引脚为高电平时顺时针旋转  
“dirpin 1” 将设置电机在 dir 引脚为高电平时逆时针旋转

## **errorlimit**

获取/设置可接受的最大错误度数，当误差超过这个度数时 errorpin 就会输出错误标志

## **ctrlmode**

获取/设置控制器操作模式。参数 0-4 用来设置当前的控制模式。

0: Controller off 代表此控制器不使用

1: Open-Loop 没有反馈的开环模式

2: Simple PID 出厂默认的模式

3: Positional PID



## 4: Velocity PID

如果您不确定您在做什么，请设置为 Simple PID。

### **maxcurrent**

设置电机的最大电流，以 MA 为单位，“maxcurrent 2000”  
会设置电机的最大电流为 2000MA

### **holdcurrent**

Simple PID 模式下，最小电流（即没有位置误差的电流）就是 holdcurrent。Holdcurrent 越高，电动机的温度越高，噪声越大，扭力也越大。

### **motorwiring**

步进电机可以接线反插时转动方向就会改变，固件将使用编码器检测电机接线方向，并且自动修正接线反插时的校准补偿值。出厂首次上电时会检测电机转动的方向。如果之后接线发生变化，您可以使用此命令进行补偿。但是，如果接线反插，最好恢复出厂设置并重新校准电机。

### **stepsperrotation**

固件将在出厂首次上电时检测步进电机每圈所需要的步数并存储在 flash 中。此命令将从 flash 中读取此参数，并允许用户在

更改电机时更改此参数。但是，如果更换电机，最好恢复出厂设置并重新校准电机。

## **move**

此命令指定电机转动到绝对角度位置。而且用户可以指定移动的速度( RPM )。例如，如果当前电机角度为 0，发送 “move 3600 20”，电机将以 20RPM 的速率旋转 10 圈到 3600°。如果再次发送 “move 3600”，此时电机已经处于 3600°就不会产生任何效果。

## **stop**

如果发送的 move 命令需要很长时间才能执行完，但是我想要中断 move 命令现在就停下来，就可以发出 stop 命令，该命令将停止 move 操作。

## **setzero**

此命令会将当前电机位置设置为绝对角度 0°。请注意，如果您正在执行 move 命令，setzero 命令会立即生效，因此建议在发送 setzero 命令之前 stop 当前的 move 命令或者等待当前 move 命令完成。

## 七、常见问题解答 ( FQA )

Q : 上电后为什么出现 “Waiting MOTO POWER” 的界面？

A : 出现该界面后的处理方法和原因：

### 1、处理方法

( 1 ) 断开电源后重新上电。

( 2 ) 拆下电机后面的控制板，查看磁铁是否在正中心的位置

### 2、可能存在的原因

( 1 ) 上电后人为的转动电机。

( 2 ) 驱动转接板接到的主板中没有固件。

Q : 设置好参数后，怎么没有变化？

A : 参数设置好后需要等待 2-5s 的时间，然后按下闭环驱动板的复位按键。

Q : 有固件，磁铁是在正中心，重新上电后，为什么还是会出现 “Waiting MOTO POWER” 的界面？

A：拔掉转接板上面的线，等待 1 分钟左右后再接上。

Q：用 USB 连接电脑后，为什么无法与电脑通讯？

A：请检查电脑是否安装好驱动（[如上](#)）。若驱动没问题，还是无法通讯，请检查调试工具的 COM 口是否与驱动一致，并检查波特率是否为 115200。

## 八、注意事项

- 1、插转接板时，请注意转接板的方向，一定不要接反。
- 2、首次使用时，该闭环驱动板必须接到有固件的主板上，再对编码器进行校准（[参见上面的校准教程](#)）校准时间需要 1-2 分钟，请耐心等待。
- 3、用 USB 连接电脑时检查驱动是否安装就绪
- 4、显示屏或串口显示的数据不正常时，请拆下闭环驱动板，查看电机轴上的磁铁是否在正中心的位置。如若不在，请调回正中心即可。
- 5、上电后不要立即进行设置，等初始化完成（3-5s）再进行参数设置。
- 6、上电后请不要人为转动电机。

## 九、所有资料下载地址

<https://github.com/bigtreotech/BIGTREETECH-SKR-V1.3>