

**BIG TREE TECH**

---

**BIGTREETECH**  
**42 Stepper Motor Closed Loop**  
**Driver Board**  
**Manual**

# Contents

<a href="#">一、 Product introduction.....</a>	<a href="#">3</a>
<a href="#">二、 Product parameters.....</a>	<a href="#">3</a>
<a href="#">三、 Features.....</a>	<a href="#">3</a>
<a href="#">四、 Interface Description.....</a>	<a href="#">4</a>
<a href="#">五、 Key Function.....</a>	<a href="#">5</a>
<a href="#">六、 Parameter adjustment.....</a>	<a href="#">5</a>
<a href="#">1.OLEDButton adjustment method.....</a>	<a href="#">5</a>
<a href="#">2.Command line adjustment method.....</a>	<a href="#">7</a>
<a href="#">2.1Solutions for abnormal driver.....</a>	<a href="#">8</a>
<a href="#">2.2 Serial Port Debugging Guide.....</a>	<a href="#">9</a>
<a href="#">七、 Frequently Asked Questions.....</a>	<a href="#">13</a>
<a href="#">八、 Precautions.....</a>	<a href="#">14</a>

## 一 . product introduction

The closed-loop drive of the stepper motor uses closed-loop control technology. The ATSAM21G18A is the main controller, the A4954 is the motor driver, and the AS5047D is the magnetic encoder. The closed-loop drive is to feedback the rotation angle of the stepping motor to the control panel, compare the distance that needs to be rotated with the distance of the actual rotation, calculate the error value, and then compensate, so as to prevent the problem of multi-step and lost step. The closed-loop drive can completely overcome the lost step of the open-loop stepping motor, and can also significantly improve the performance of the motor at high speed, thereby improving the processing speed and accuracy of the machine. When the motor is out of step for a long time, the main control chip can detect the step-out condition of the stepping motor through the magnetic encoder, thereby compensating, so that the machine can continue to work normally.

## 二.the product parameters

Motor Power Supply (VM): 12V---24V

Motor default operating current: 800mA

Motor default standby current: 500mA

Drive segmentation default: 16 segments

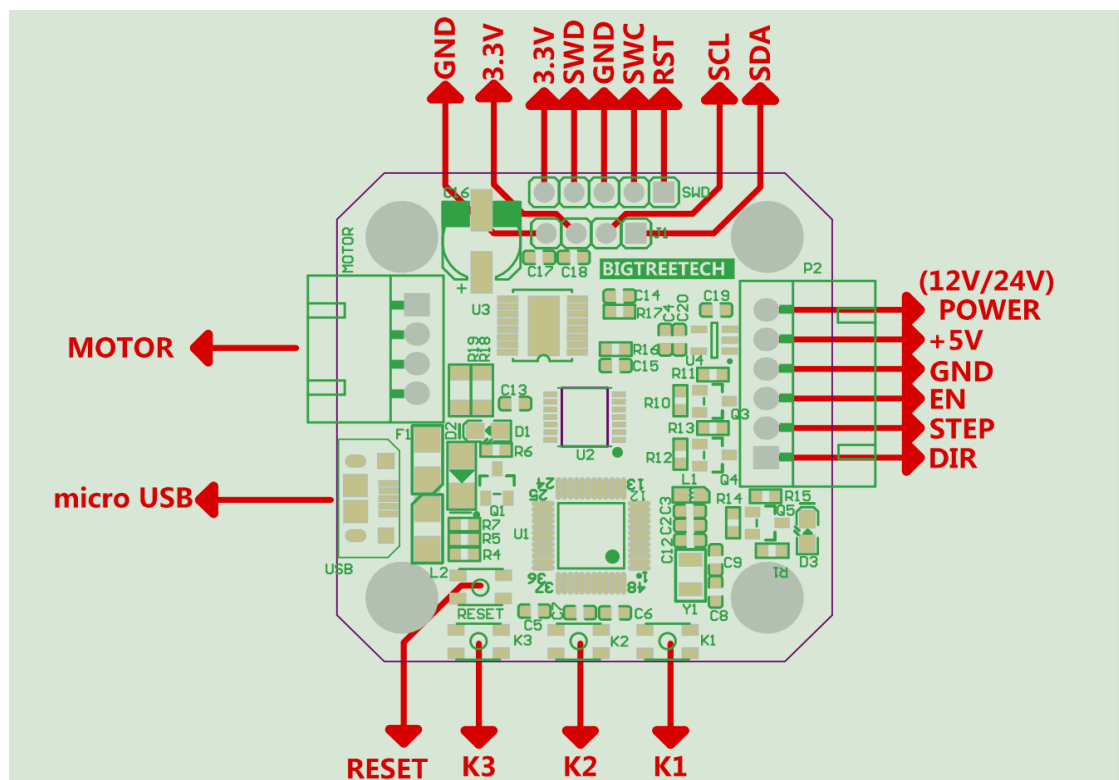
Subdivision steps: 1 2 4 8 16 32 64 128 256

Maximum current: 2000mA

### ≡.product features

1. The whole set of products is sold, eliminating the complicated assembly and debugging, and plug and play.
- 2, can not lose steps in high-speed printing.
3. Low heat generation and high operating efficiency.
- 4, easy to install, easy to wire.
5. Compared with ordinary drives, the motor runs more stably.
6. In the case of OLED, the current and subdivision of the motor can be modified by pressing the button. The operation is simple, and the parameter adjustment can also be performed using USB.
7. In the case of no OLED, you can use USB to connect to the computer and adjust the parameters by command.

## 四、 the interface description



## 五、 Key Function

**RESET:** reset button

**K1:** Confirm selection (only valid in the setting interface)

**K2:** Switch settings interface and status interface

**K3:** Select the next line option (only valid in the settings interface)

## 六、 parameter adjustment

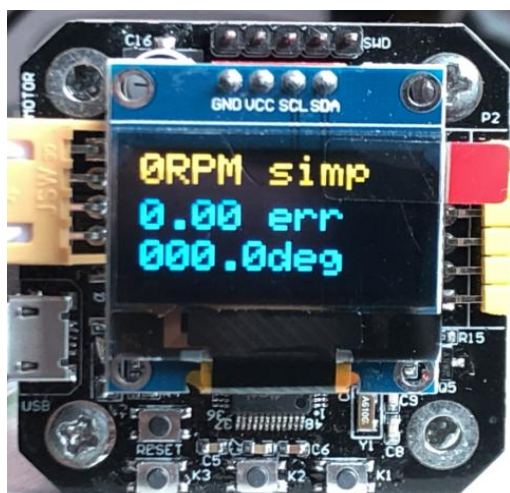
Note: 42 stepping motor closed-loop drive control board has two parameter adjustment methods

1、

OLED button adjustment method

Advantages: no need to connect to a computer, through OLED display, button settings, simple and intuitive operation.

### Status interface

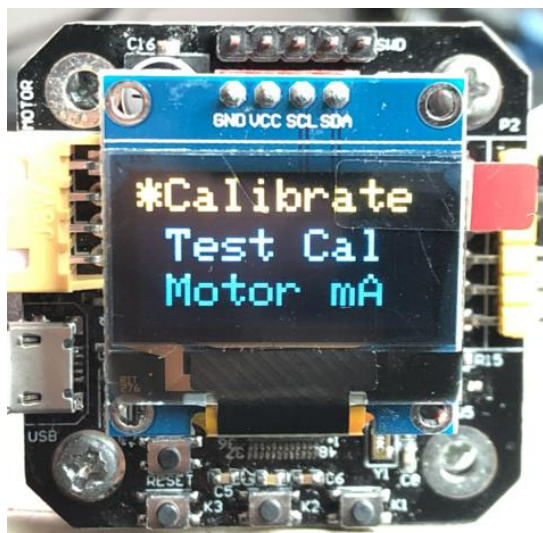


1 ) .The first line, 0 RPM, represents the current number of revolutions per revolution.

2 ) .The second line, 0.00 err, represents the error between the actual position and the target position of the motor.

3 ) The third line, 000.0deg, represents the current target position of the motor is 0 °.

4) Setting interface



There are a total of 7 options in the settings interface.

### **Calibrate:**

The encoder is rotated 200 times ( $1.8^{\circ}$  each time) to calibrate the encoder. After the first power-on, the parameters must be calibrated..

### **Test Cal:**

Test calibrate and report the maximum error in degrees

### **Motor mA**

Set the current when the motor is running, default 800MA

### **Hold mA**

Set the current when the motor is in standby, the default is 500MA

## **Microstep,**

Set segmentation, default 16 segments

## **EnablePin**

Set the logic level of the enable signal, Enable represents the high level enable, !Enable represents the low level enable, default !Enable low enable

## **DirPin**

Set the logic level of the dir signal. Modify this option to change the direction of rotation of the motor.

## **Key**

RESET: resetting key

K2: Switch settings interface and status interface

K3: Select the next line option (only valid in the settings interface)

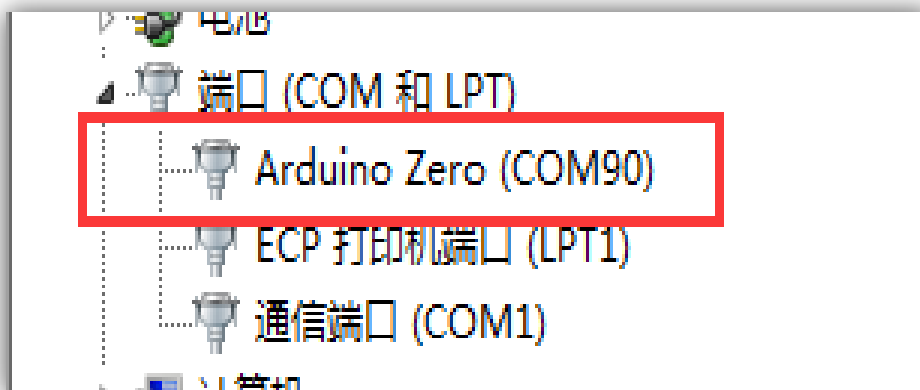
K1: Confirm selection (only valid in the setting interface)

## **2、 Command line adjustment**

Advantages: the parameters that can be set are more detailed



After connecting the computer, check if the driver is installed.



If the display is normal, you do not need to install the following drivers

## 2.1 Solutions for abnormal driver

Driver download website:

<https://github.com/bigtreotech/BIGTREETECH-SERVO42A-V1.0>

According computer system to install driver

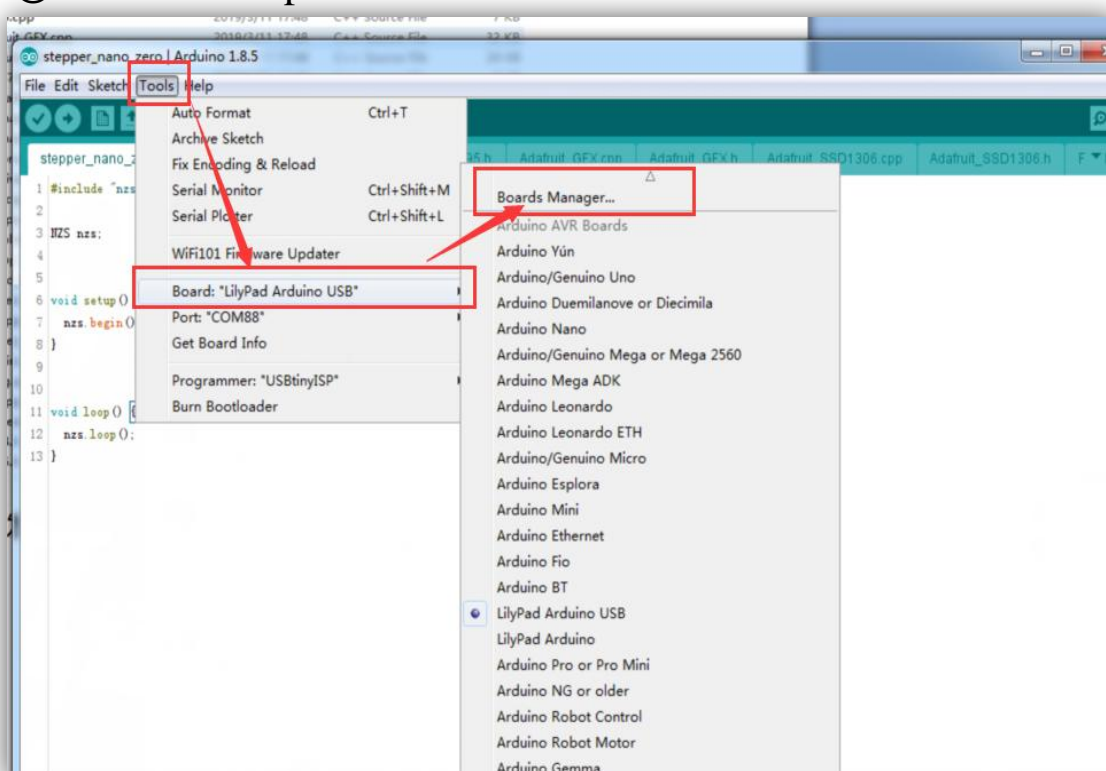
名称	修改日期	类型	大小
amd64	2019/3/11 16:44	文件夹	
FTDI USB Drivers	2019/3/11 16:44	文件夹	
ia64	2019/3/11 16:44	文件夹	
license	2019/3/11 16:44	文件夹	
x86	2019/3/11 16:44	文件夹	
arduino.cat	2016/3/9 16:11	安全目录	11 KB
arduino.inf	2016/3/9 16:11	安装信息	10 KB
arduino_gemma.cat	2016/3/9 16:11	安全目录	11 KB
arduino_gemma.inf	2016/3/9 16:11	安装信息	8 KB
arduino-org.cat	2016/3/9 16:11	安全目录	9 KB
arduino-org.inf	2016/3/9 16:11	安装信息	8 KB
dpinst-amd64.exe	2016/3/9 16:11	应用程序	1,024 KB
dpinst-x86.exe	2016/3/9 16:11	应用程序	901 KB
genuino.cat	2016/3/9 16:11	安全目录	9 KB
genuino.inf	2016/3/9 16:11	安装信息	5 KB
Old_Arduino_Drivers.zip	2016/3/9 16:11	360压缩 ZIP 文件	17 KB
README.txt	2016/3/9 16:11	文本文档	1 KB

## 2.2 Serial Port Debugging Guide

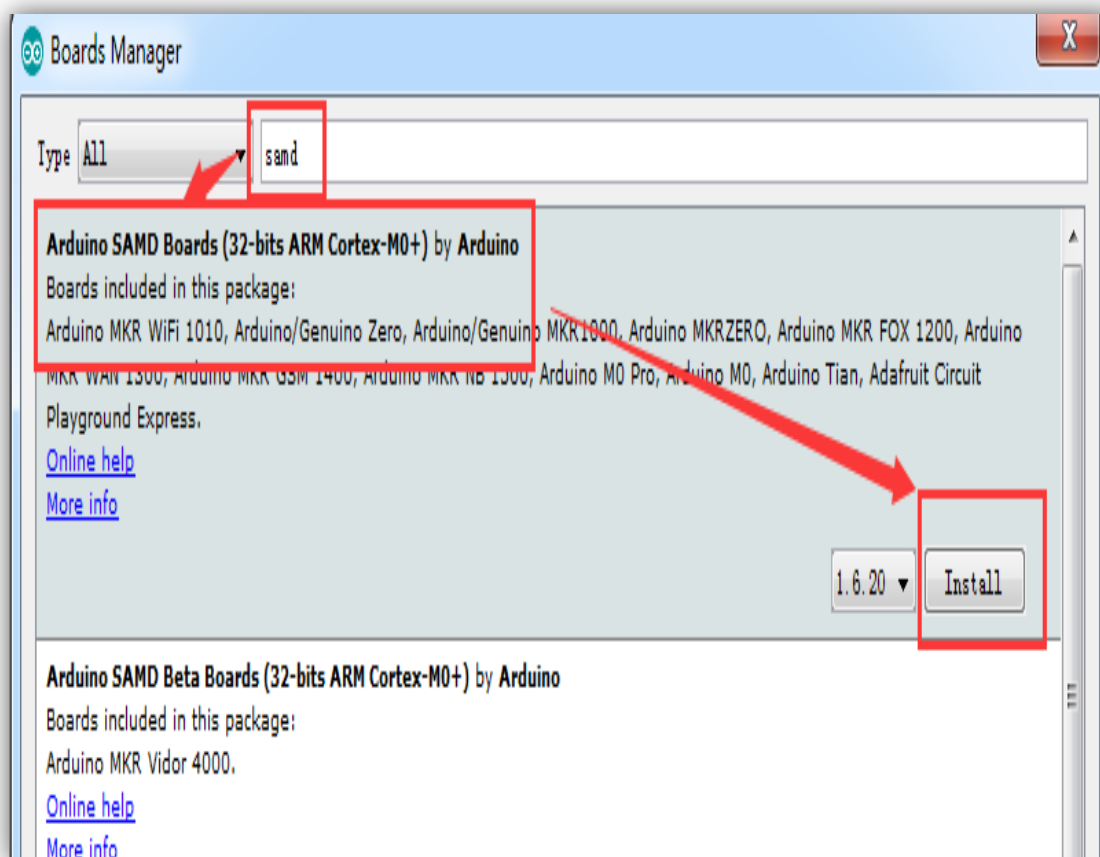
①After the driver is ready, open the firmware with the arduino IDE

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

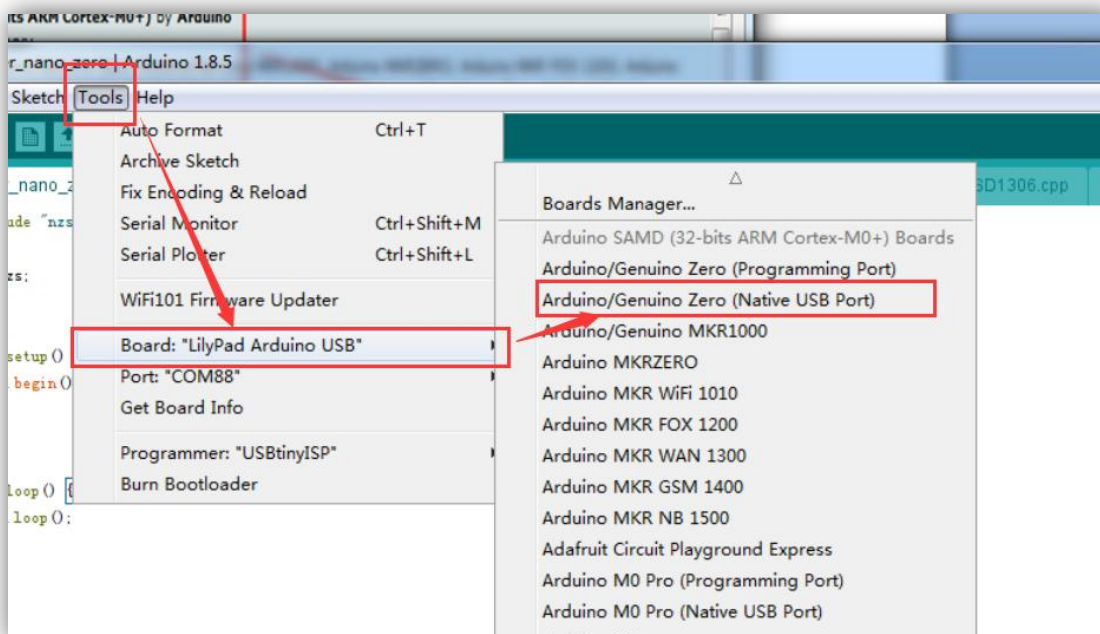
②Install development board



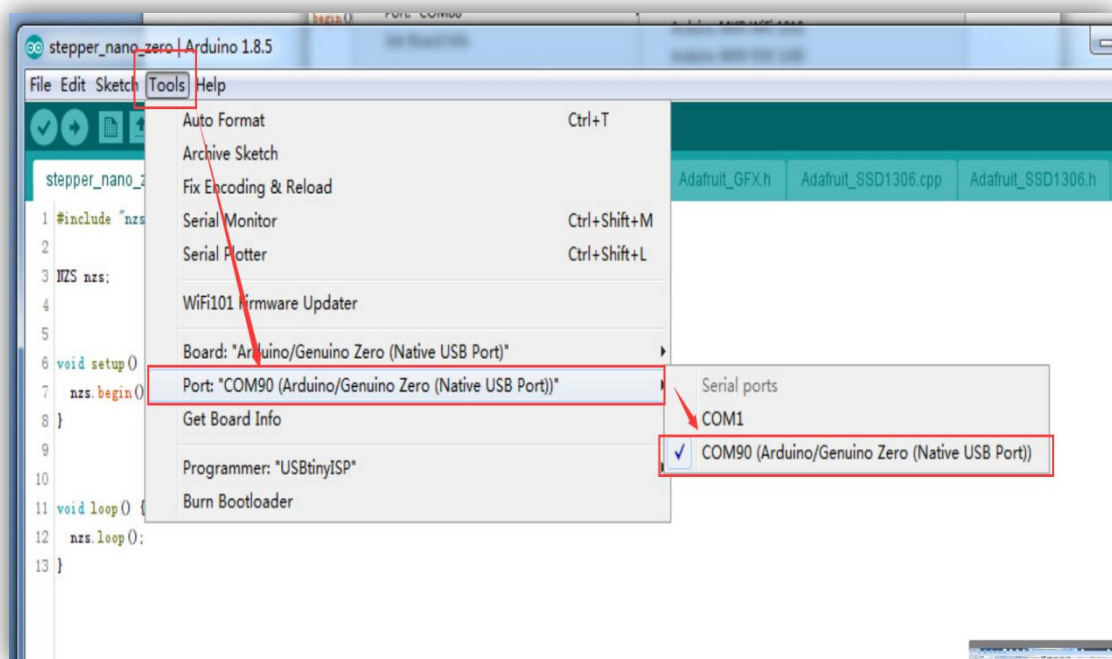
### ③Input “samd”



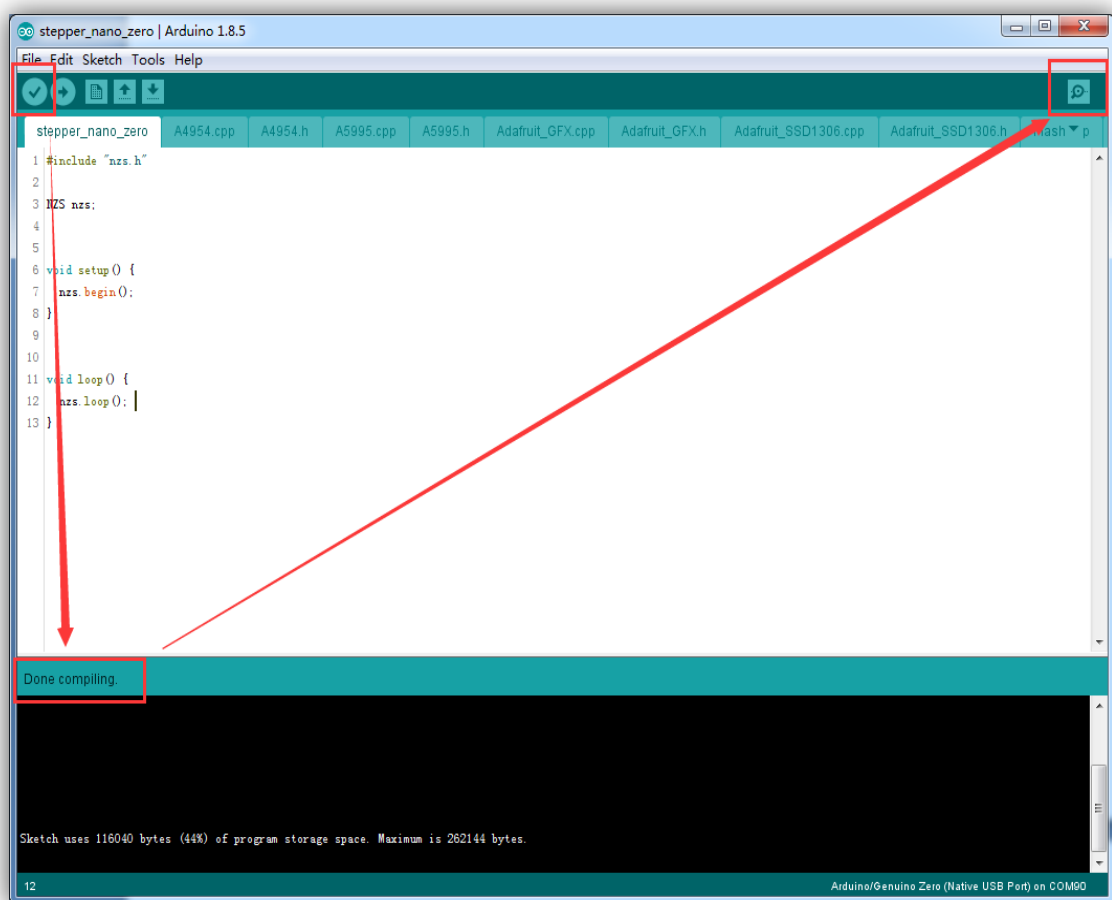
### ④Select development board



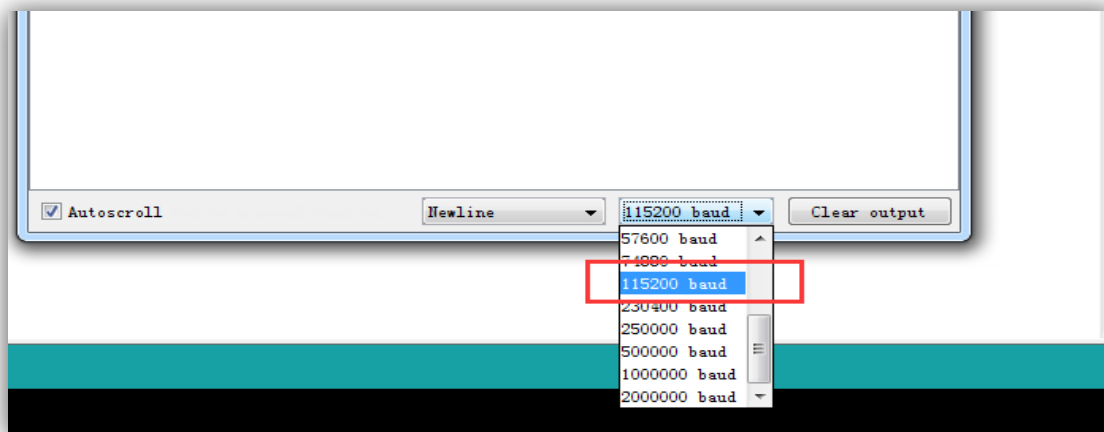
## ⑤ Choose a port



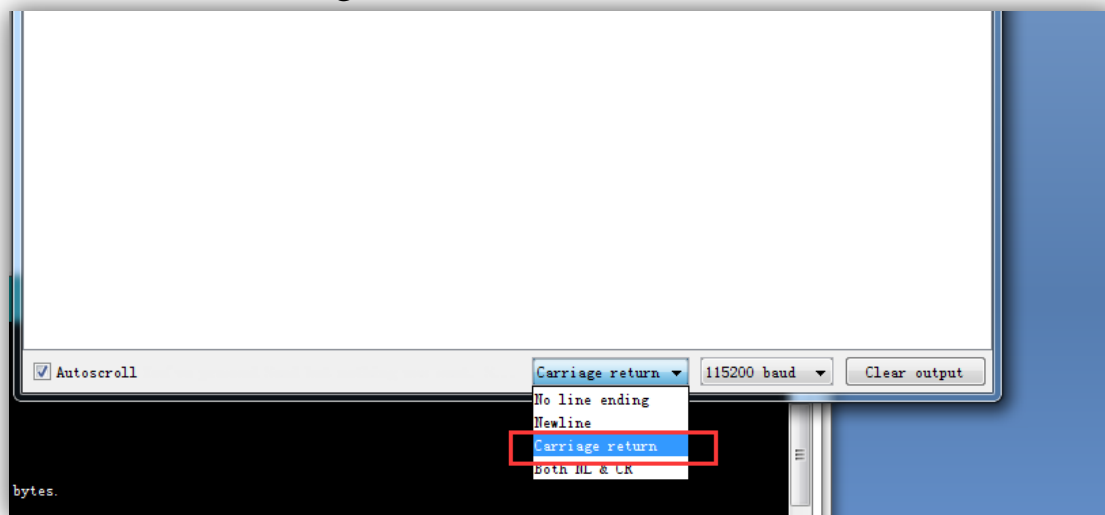
## ⑥ Compiler without error -> opens the serial monitor



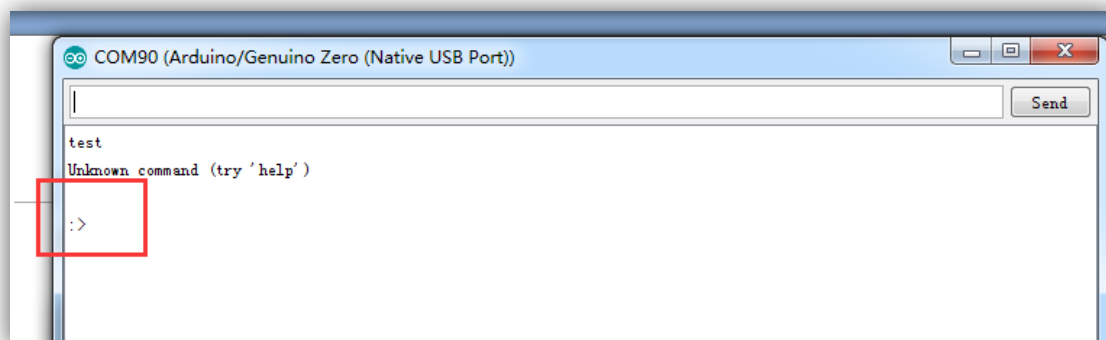
⑦ Set the baud rate to 115200



⑧ Choose "carriage return"



⑨ Send a command like "test" , When returning to " :> " ,  
Representatives can send new orders now



## ⑩ Sending "help" will return a list of all commands

```

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
stepsperrrotation - 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- TX, 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

:>

```

If the above data appears, it mean that the connection is successful, you can send the following command to modify the parameters.

## **help**

The “help”command will return a list of all supported commands.

## **getcal**

This command will print out a 200-point calibration table. This is useful if you are doing firmware development and don't want to calibrate each time you update the firmware. You can copy this table into a nonvolatile.cpp file.

## **calibrate**

The encoder is calibrated by rotating the encoder 200 times ( $1.8^{\circ}$  each time)

## **testcal**

Test calibrate and report the maximum error in degrees

## **step**

This will move the motor one step clockwise, based on the current subdivision setting. Use “step 0 16” to move the motor 16 steps clockwise and “step 1 16” to move the motor counterclockwise by 16 steps.

**feedback**

This command is used to set whether the control board is in open loop or closed loop.

**Readpos**

Report the current position of the motor in degrees

**encoderdiag**

Reports the internal registers of the AS5047D encoder to make it easy to check the encoder for problems.

**microsteps**

This command gets/sets the current segment

**spid**

This command is used to set the Kp, Ki and Kd parameters in the simple positional PID mode.

**ppid**

This command is used to set the Kp, Ki and Kd parameters in positional PID mode.

**vpid**



This command is used to set the Kp, Ki and Kd parameters in Velocity PID mode.

### **velocity**

This command is used to set the speed of motor rotation in velocity PID mode.

### **boot**

This command will put the microprocessor into bootloader mode. (can also be done by double-clicking the RESET button)

### **factoryreset**

This will erase all parameters of the calibration and settings and reset to factory settings. After executing this command, the motor needs to be calibrated again.

### **Dirpin**

"dirpin 0" will set the motor to rotate clockwise when the dir pin is high.

"dirpin 1" will set the motor to rotate counterclockwise when the dir pin is high.

### **errorlimit**

Get/set the maximum number of acceptable errors, errorpin will output an error flag when the error exceeds this degree

### **ctrlmode**

Gets/sets the controller operating mode. Parameter 0-4 is used to set the current control mode.

0: Controller off :Represents this controller not used

1: Open-Loop :Open loop mode without feedback

2: Simple PID :Factory default mode

3: Positional PID

4: Velocity PID

If you are not sure what you are doing, set it to Simple PID.

### **maxcurrent**

Set the maximum current of the motor in MA. “maxcurrent 2000” will set the maximum current of the motor to 2000MA.

### **holdcurrent**

In Simple PID mode, the minimum current (ie, the current without position error) is holdcurrent. The higher the

Holdcurrent, the higher the temperature of the motor, the greater the noise and the greater the torque.

### **motorwiring**

When the stepper motor can be reversed, the direction of rotation will change. The firmware will use the encoder to detect the motor wiring direction and automatically correct the calibration compensation value when the cable is reversely inserted. The direction in which the motor rotates is detected when the factory is first powered up. If the wiring changes afterwards, you can use this command to compensate. However, if the wiring is reversed, it is best to restore the factory settings and recalibrate the motor.

### **stepsperrotation**

The firmware will detect the number of steps required per step of the stepper motor at the factory power-on and store it in the flash. This command reads this parameter from flash and allows the user to change this parameter when changing the motor. However, if you replace the motor, it is best to restore the factory settings and recalibrate the motor.

### **move**

This command specifies that the motor is rotated to an absolute angular position. And the user can specify the speed of movement (RPM). For example, if the current motor angle is 0, send "move 3600 20" and the motor will rotate 10 turns to 3600° at 20RPM. If you send "move 360 0" again, the motor is already at 3600° and it will have no effect.

### **stop**

If the move command sent takes a long time to execute, but I want to interrupt the move command and stop now, I can issue the stop command, which will stop the move operation.

### **setzero**

This command sets the current motor position to an absolute angle of 0°. Note that if you are executing the move command, the setzero command will take effect immediately, so it is recommended to stop the current move command or wait for the current move command to complete before sending the setzero command.

## **七、 ( FQA )**

Q: Why does the “Waiting MOTO POWER” interface appear after power-on?

A: The processing method and reason after the interface appears:

1, processing method

(1) Re-power on after disconnecting the power.

(2) Remove the control panel behind the motor and check if the magnet is in the center.

2, possible reasons

(1) Artificially rotating the motor after power-on.

(2) There is no firmware in the motherboard connected to the drive adapter board.

Q: After setting the parameters, how did it change?

A: After setting the parameters, you need to wait for 2-5s, then press the reset button of the closed loop drive.

Q: There is firmware, the magnet is in the center. After re-powering, why is the “Waiting MOTO POWER” interface still appearing?

A: Unplug the line above the adapter plate and wait for about 1

minute before connecting.

Q: After connecting to the computer with USB, why can't I communicate with the computer?

A: Please check if the driver is installed on the computer (as above). If the driver is ok, it is still not possible to communicate. Please check if the COM port of the debugging tool is consistent with the drive and check if the baud rate is 115200.

## 八、 matters needing attention

1. When inserting the adapter board, please pay attention to the direction of the adapter board, and do not reverse it.
2. When using for the first time, the closed-loop driver board must be connected to the motherboard with firmware, and then the encoder is calibrated (see the calibration tutorial above). The calibration time takes 1-2 minutes. Please be patient.
3. Check if the driver is installed when connecting to the computer with USB.
4. When the data displayed on the display or serial port is abnormal, please remove the closed-loop drive board and check

if the magnet on the motor shaft is in the center position. If not, please return to the center.

5. Do not set it immediately after power-on, and wait for initialization (3-5s) before setting parameters.

6. Please do not manually turn the motor after power-on.