Waveshare ST3215 High Precision and Torque User Manual

Contents [hide

WAVESHARE
share awesome hardware

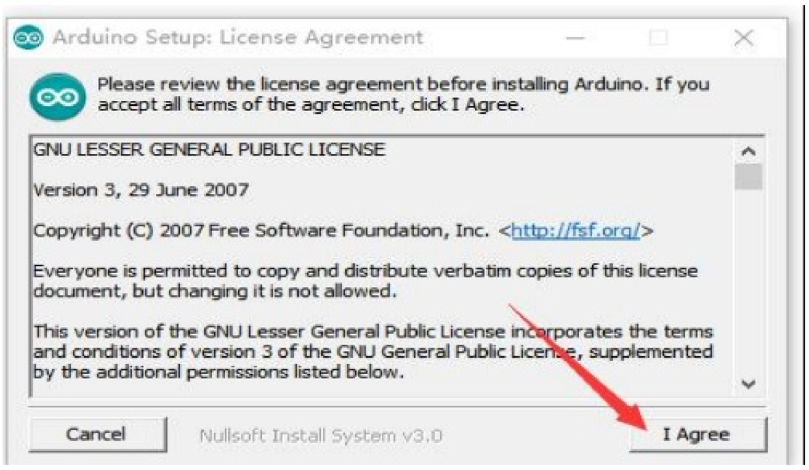
Waveshare ST3215 High Precision and Torque

SERVO DRIVER WITH ESP32

HOW TO INSTALL ARDUINO IDE

You can download the Arduino IDE from Arduino. cc.

1. Choose the Arduino IDE suitable for your system.
2. Run the installation package.
3. Click on I Agree.



Please review the license agreement before installing Arduino. If you accept all terms of the agreement, click I Agree.

GNU LESSER GENERAL PUBLIC LICENSE

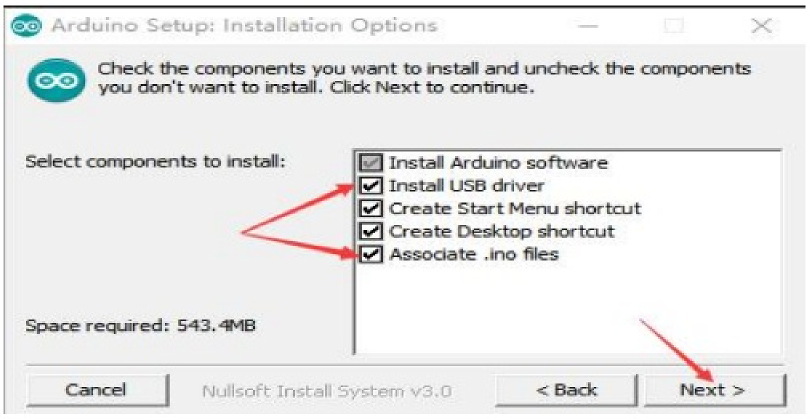
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

4. Check Install USB driver and Associate. in files, then click Next.


Arduino Setup: Installation Options

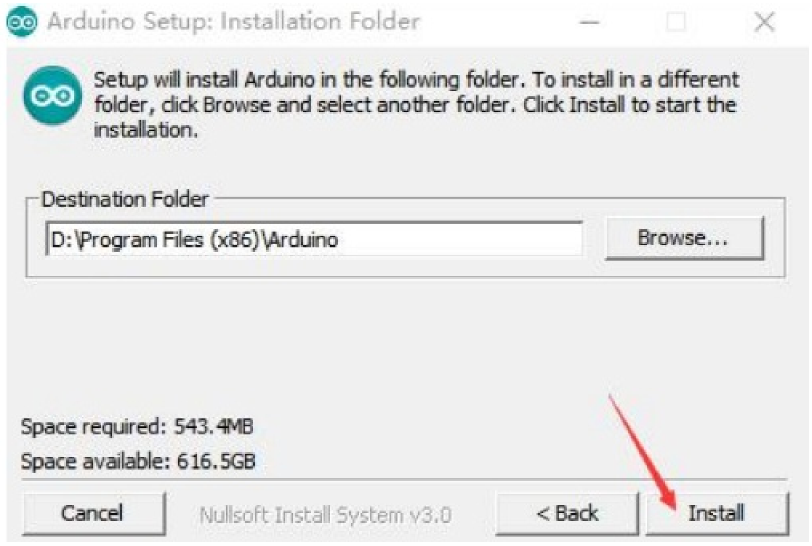
Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:

- [✓] Install Arduino software
- [✓] Install USB driver
- [✓] Create Start Menu shortcut
- [✓] Create Desktop shortcut
- [✓] Associate .ino files

Space required: 543.4MB

5. Select the installation path of Arduino IDE, then click Install.

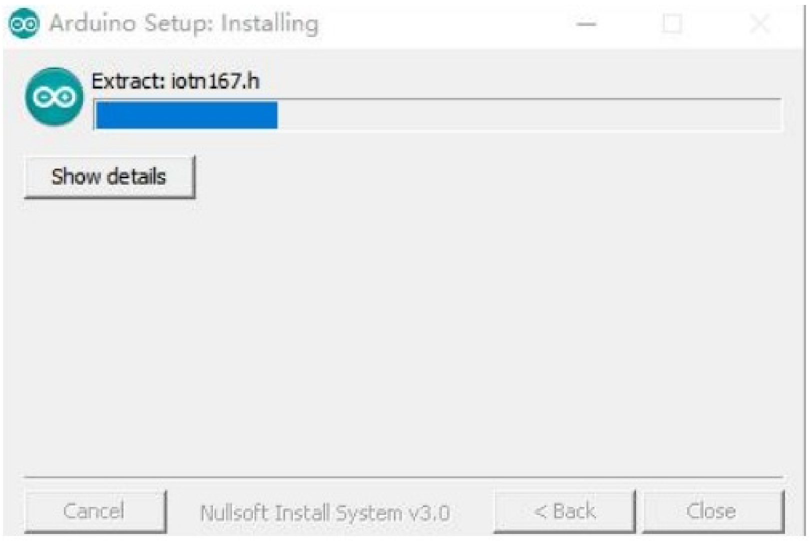

Arduino Setup: Installation Folder

Setup will install Arduino in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

Destination Folder: D:\Program Files (x86)\Arduino

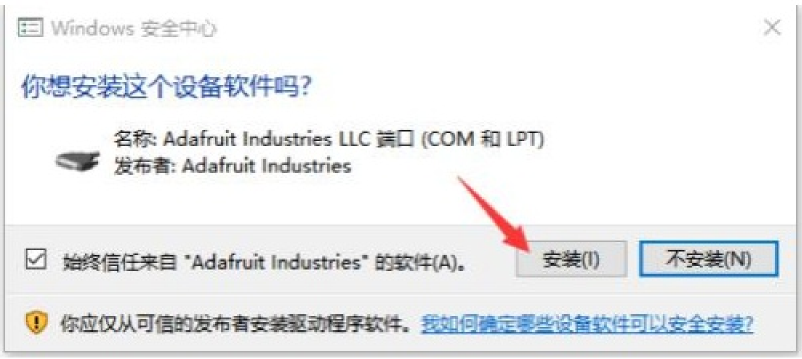
Space required: 543.4MB
Space available: 616.5GB

6. Wait for the installation to complete.


Arduino Setup: Installing

Extract: iotn167.h

7. Install the required drivers.
8. Click to install.


Windows 安全中心

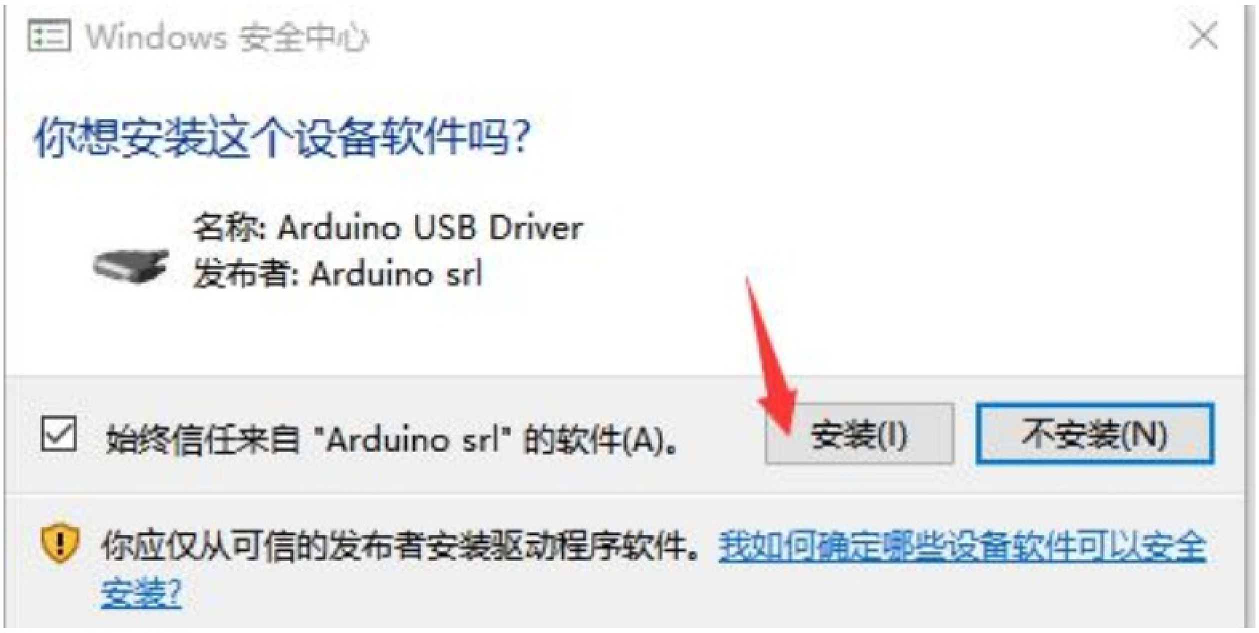
你想安装这个设备软件吗?

名称: Adafruit Industries LLC 端口 (COM 和 LPT)
发布者: Adafruit Industries

始终信任来自 "Adafruit Industries" 的软件(A)。

你应仅从可信的发布者安装驱动程序软件。我如何确定哪些设备软件可以安全安装?

9. Click to install.


Windows 安全中心

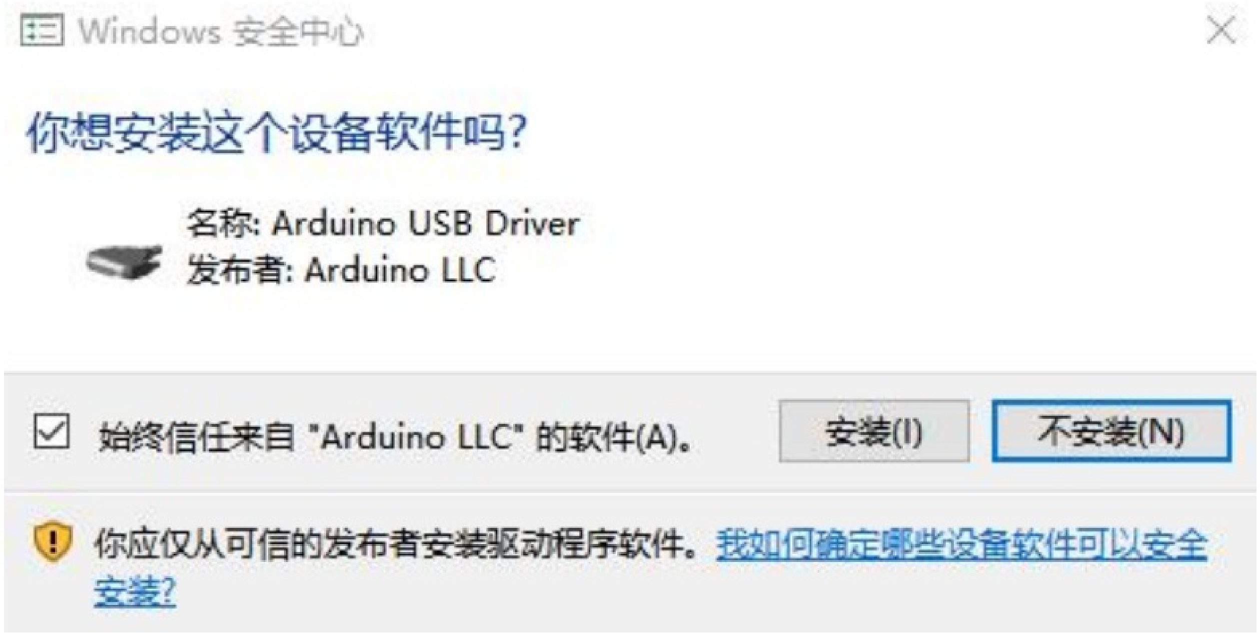
你想安装这个设备软件吗?

名称: Arduino USB Driver
发布者: Arduino srl

☑ 始终信任来自 "Arduino srl" 的软件(A)。

你应仅从可信的发布者安装驱动程序软件。我如何确定哪些设备软件可以安全安装?

10. Click to install.


Windows 安全中心

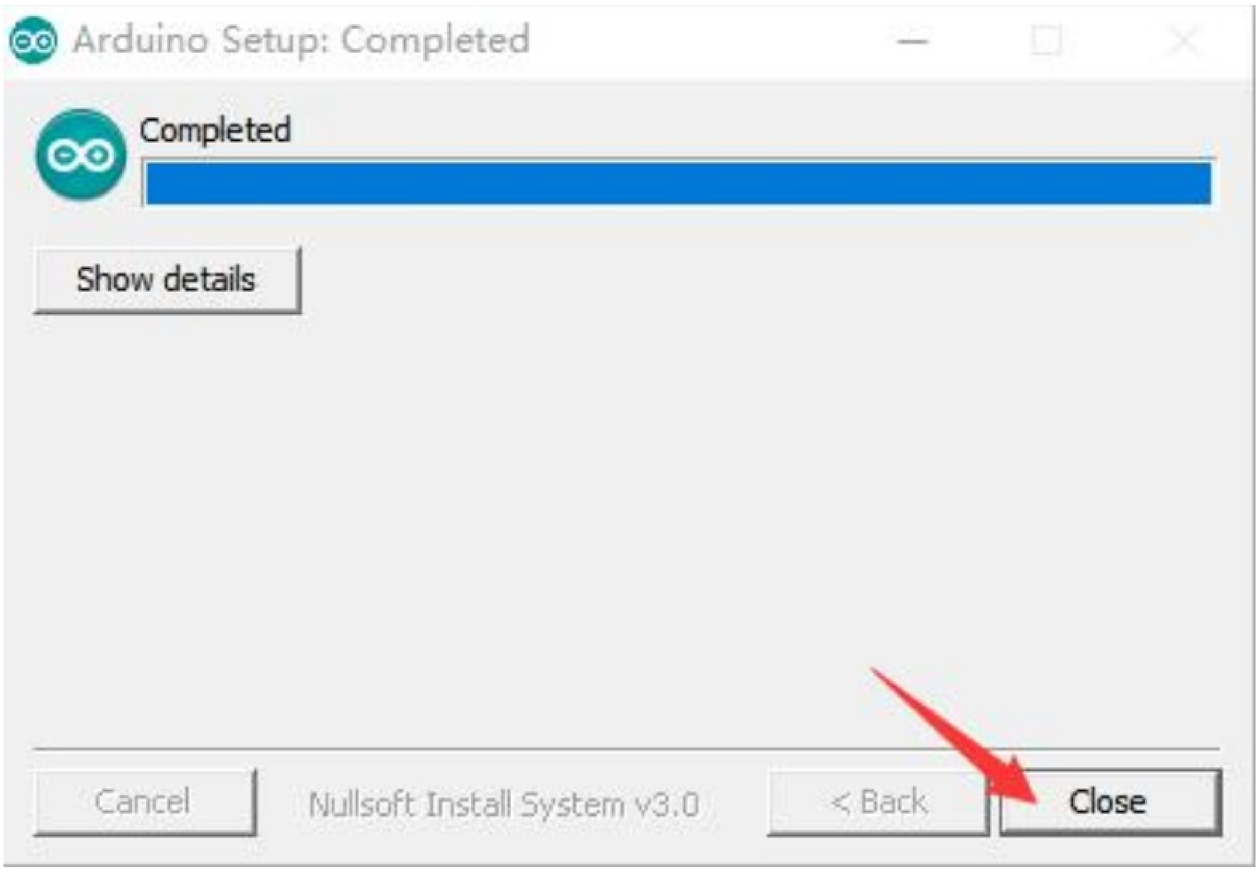
你想安装这个设备软件吗?

名称: Arduino USB Driver
发布者: Arduino LLC

☑ 始终信任来自 "Arduino LLC" 的软件(A)。

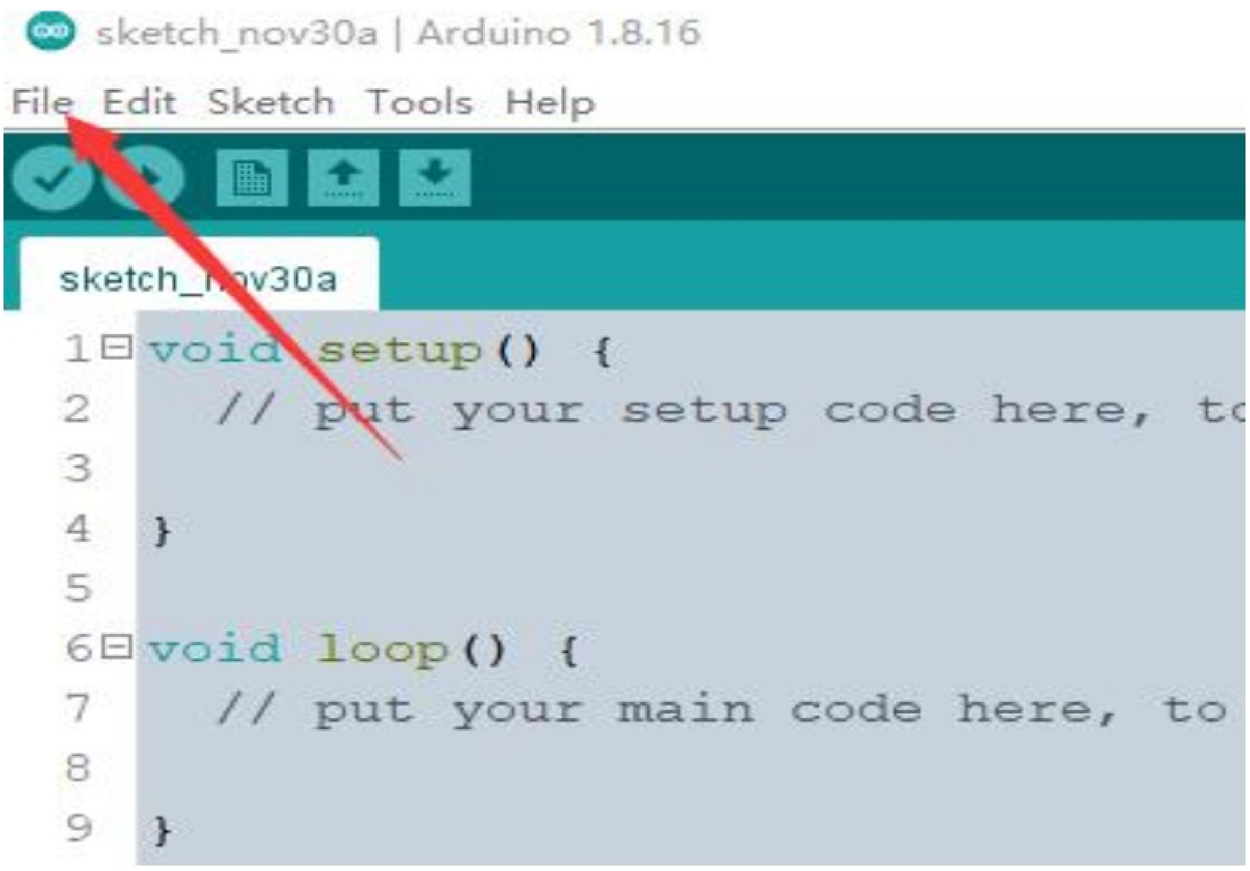
你应仅从可信的发布者安装驱动程序软件。我如何确定哪些设备软件可以安全安装?

11. Arduino IDE installation is complete.



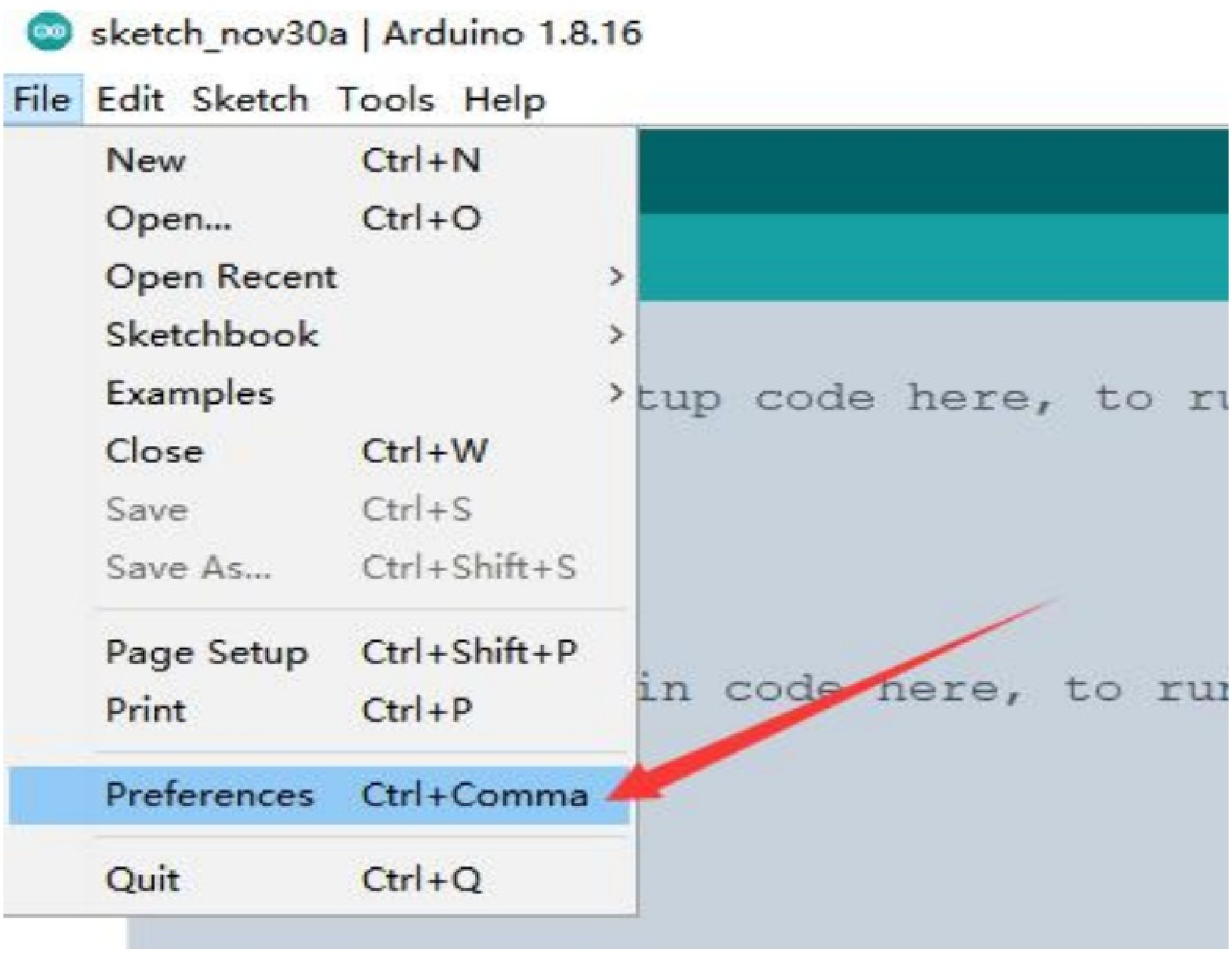
INSTALL ARDUINO CORE FOR THE ESP32

1. Run Arduino IDE, and click File.

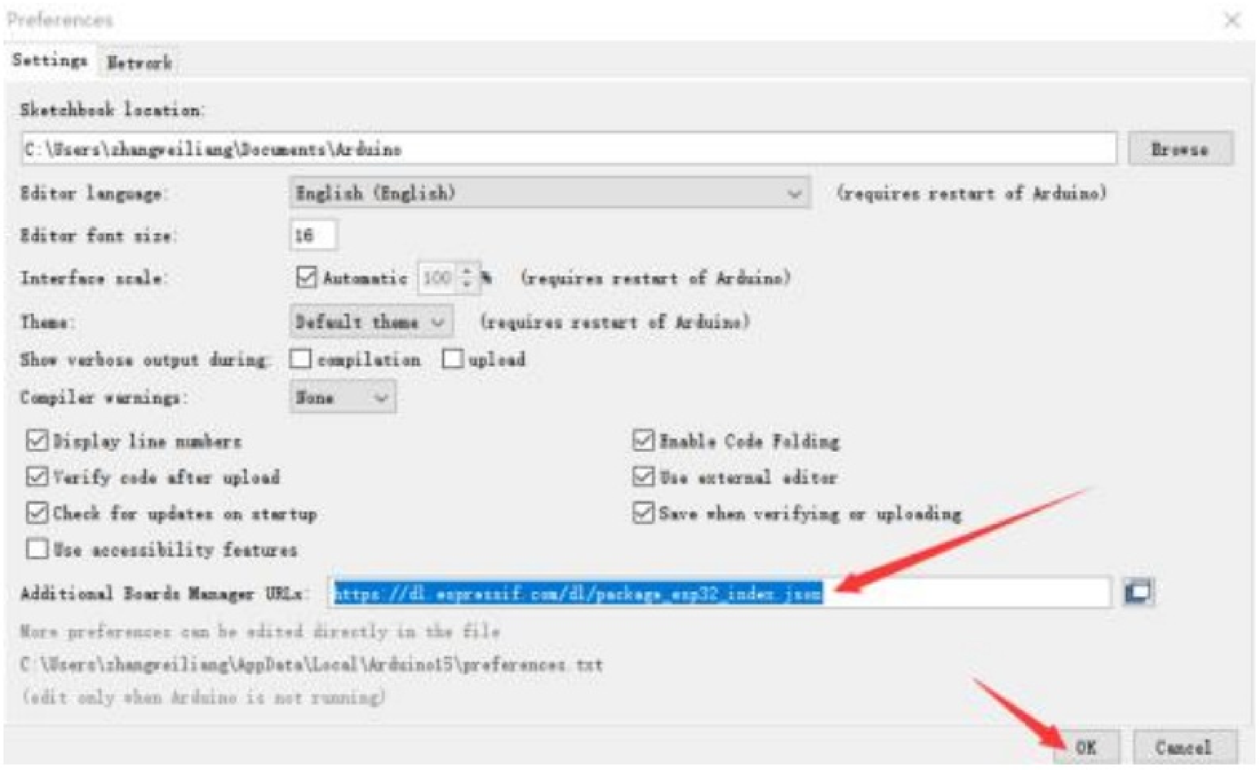


```
void setup() {
  // put your setup code here, to

}

void loop() {
  // put your main code here, to

}
```

2. Click on Preferences.



3. Fill https://dl.espressif.com/dl/package_esp32_index.json in Additional Boards Manager URLs and click OK.


Preferences

Settings | Network

Sketchbook location:
C:\Users\zhangweiliang\Documents\Arduino [Browse]

Editor language: English (English) (requires restart of Arduino)

Editor font size: 16

Interface scale: ☑ Automatic 100 % (requires restart of Arduino)

Theme: Default theme (requires restart of Arduino)

Show verbose output during: ☐ compilation ☐ upload

Compiler warnings: None

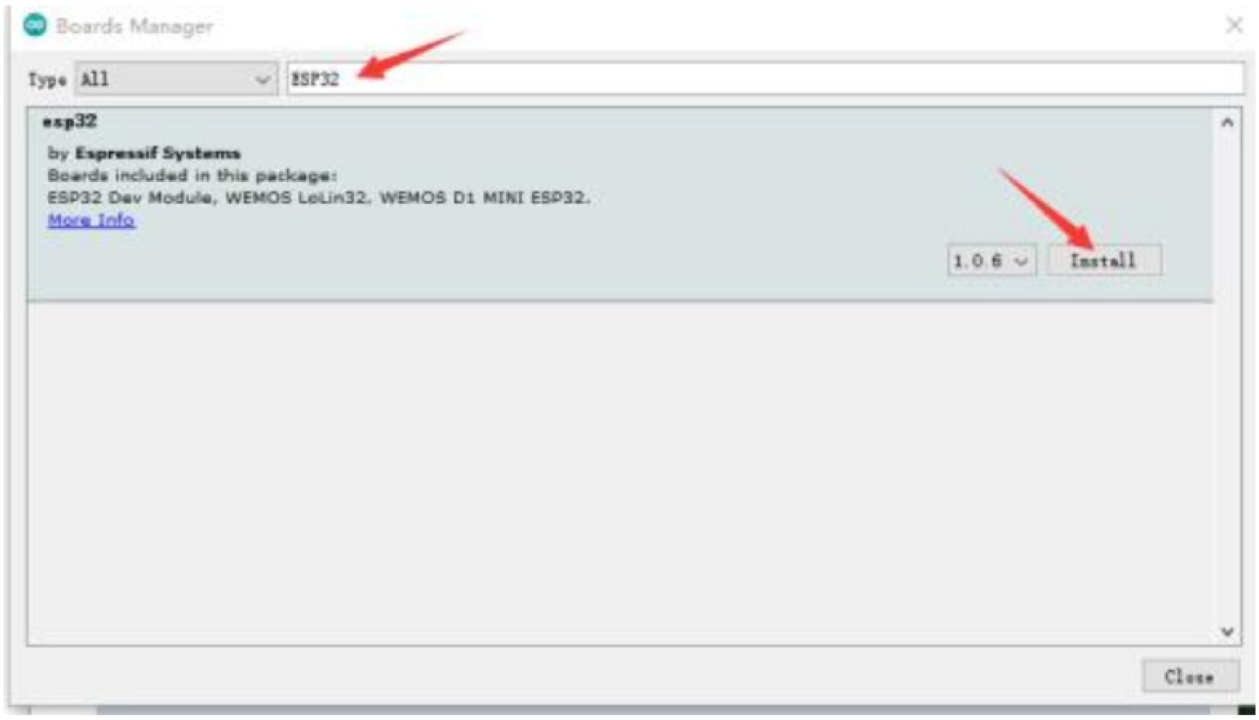
☑ Display line numbers
☑ Verify code after upload
☑ Check for updates on startup
☐ Use accessibility features
☑ Enable Code Folding
☑ Use external editor
☑ Save when verifying or uploading

Additional Boards Manager URLs: https://dl.espressif.com/dl/package_esp32_index.json

More preferences can be edited directly in the file
C:\Users\zhangweiliang\AppData\Local\Arduino15\preferences.txt
(edit only when Arduino is not running)

[OK] [Cancel]

4. Restart the IDE, and click Tools > Board > Boards Manager to open the Boards Manager.
5. Fill in ESP32 and click Install. [Note: ESP32 version 1.0.6 needs to be installed here.]


Boards Manager

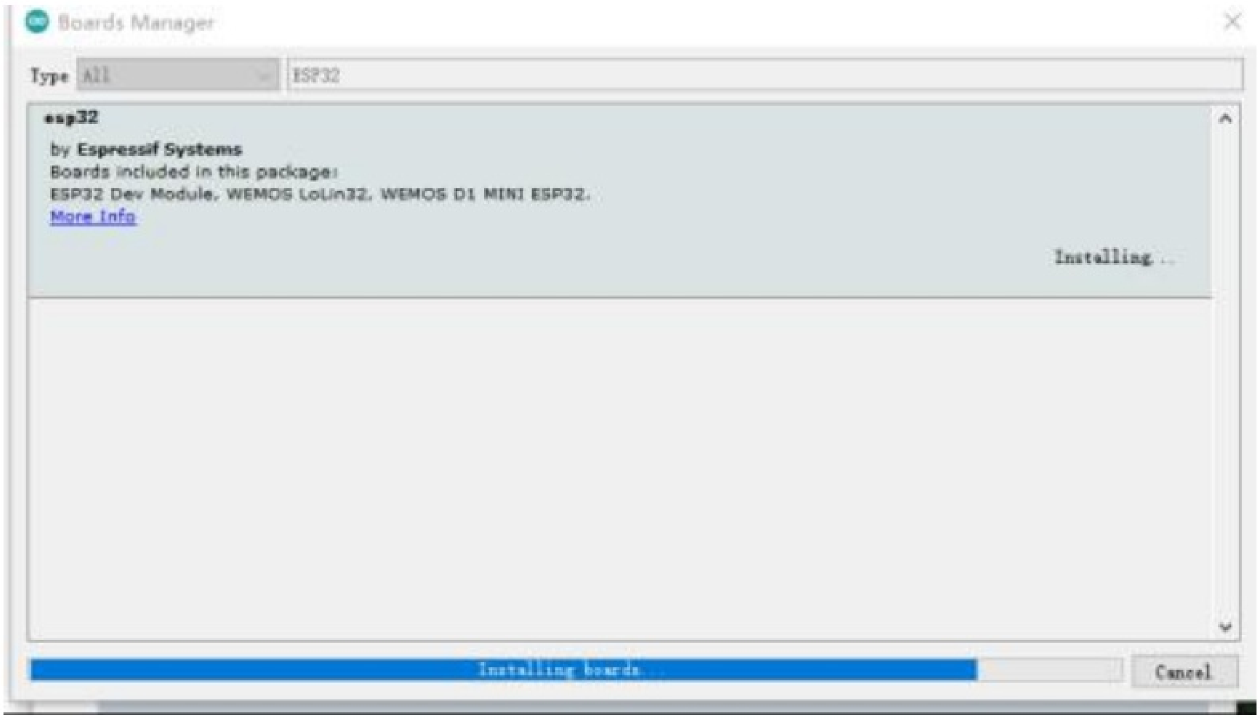
Type All ESP32

esp32
by Espressif Systems
Boards included in this package:
ESP32 Dev Module, WEMOS LoLin32, WEMOS D1 MINI ESP32.
More Info

1.0.6 Install

Close

6. Wait for the installation to complete.


Boards Manager

Type All ESP32

esp32
by Espressif Systems
Boards included in this package:
ESP32 Dev Module, WEMOS LoLin32, WEMOS D1 MINI ESP32.
More Info

Installing...

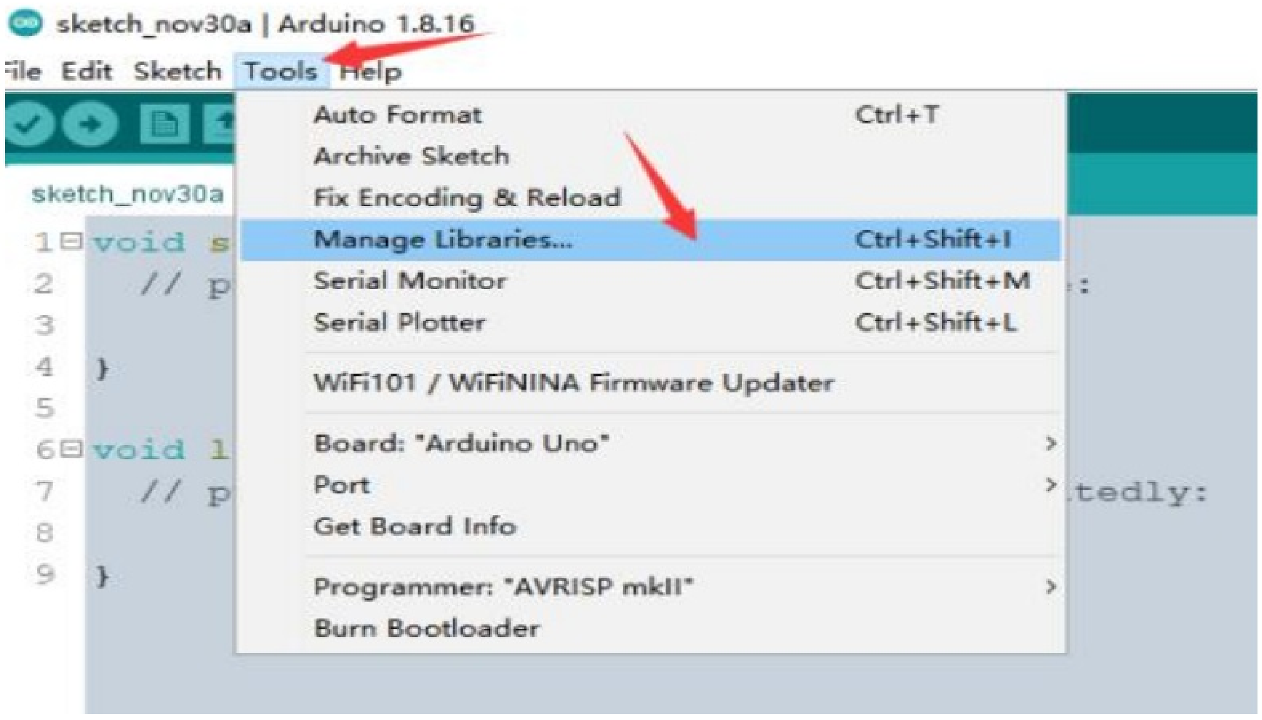
Installing boards...

Cancel

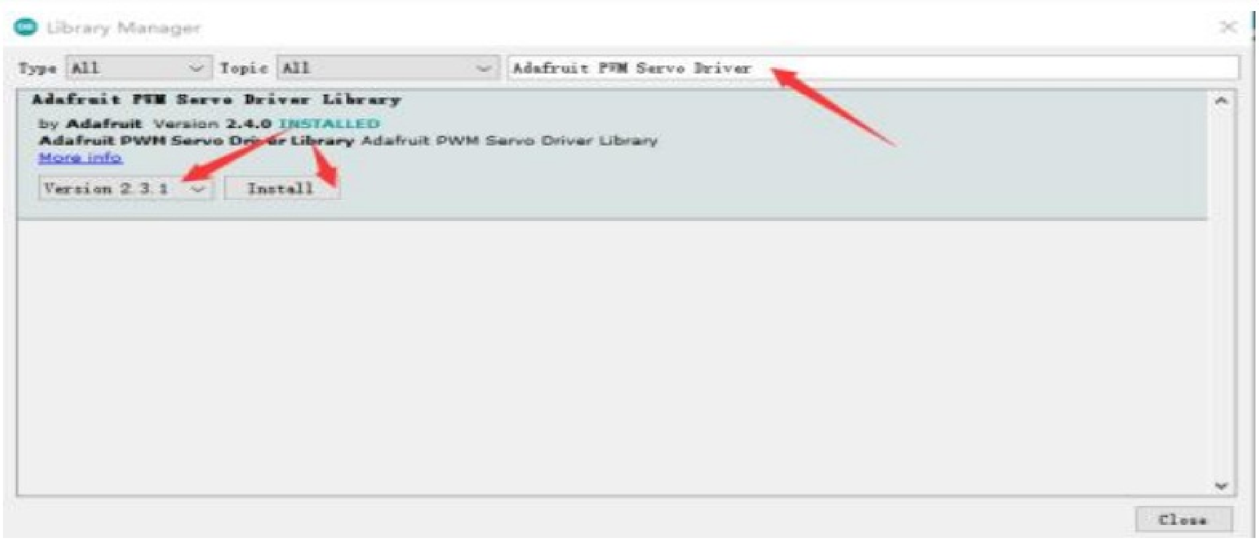
7. After installation, you can use Arduino IDE to develop ESP32.

DOWNLOAD PROGRAM & INSTALL DEPENDENCY LIBRARIES

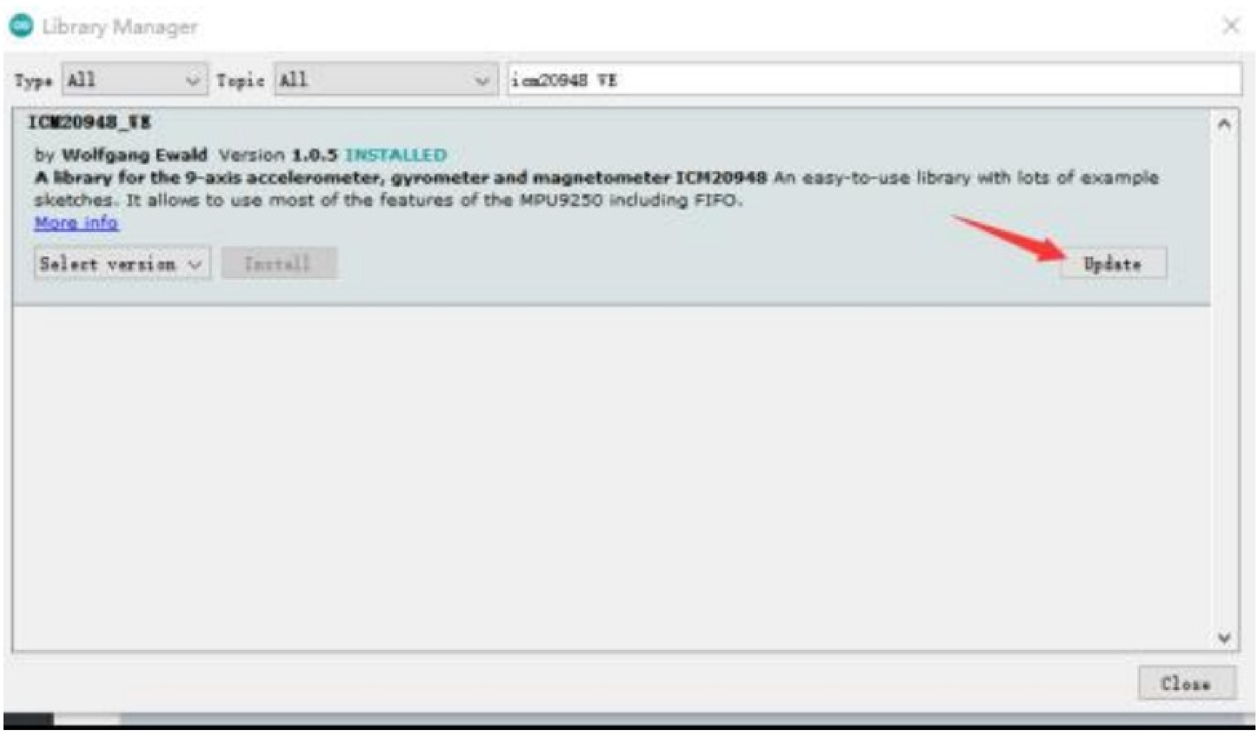
1. Please install the following libraries through Manage Libraries: Adafruit SSD1306, Adafruit NeoPixel. Library installed by copying files is SCServo.


sketch_nov30a | Arduino 1.8.16
File Edit Sketch Tools Help
sketch_nov30a
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino Uno"
Port
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader

2. Click Tools > Manage Libraries to open the library manager.

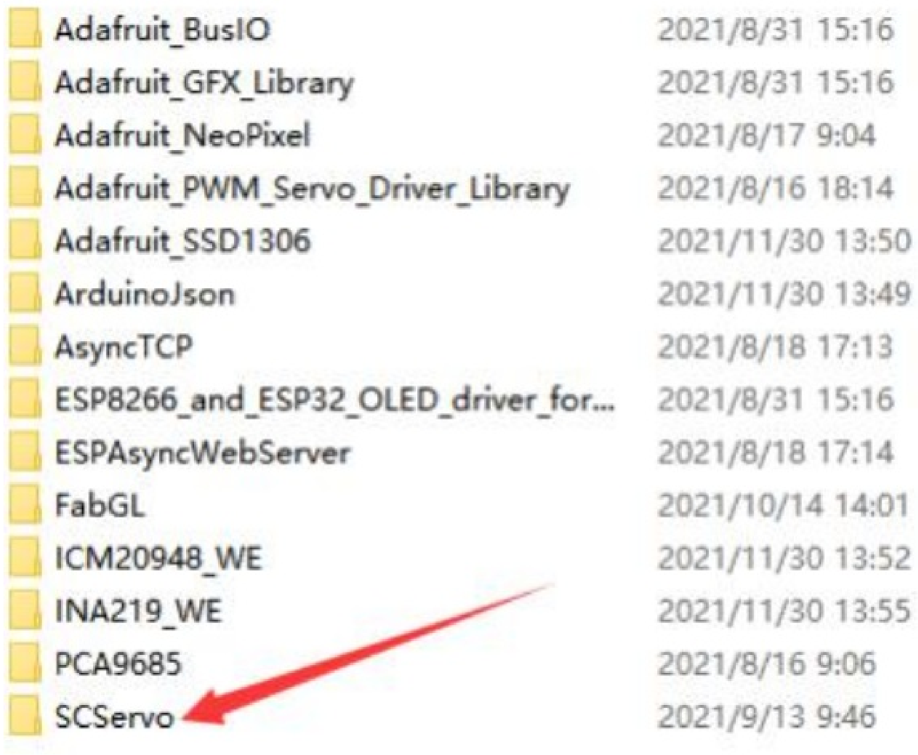


Adafruit PWM Servo Driver Library
by **Adafruit** Version **2.4.0** INSTALLED
Adafruit PWM Servo Driver Library Adafruit PWM Servo Driver Library
More info



ICM20948_WE
by **Wolfgang Ewald** Version **1.0.5** INSTALLED
A library for the 9-axis accelerometer, gyrometer and magnetometer ICM20948 An easy-to-use library with lots of example sketches. It allows to use most of the features of the MPU9250 including FIFO.
More info

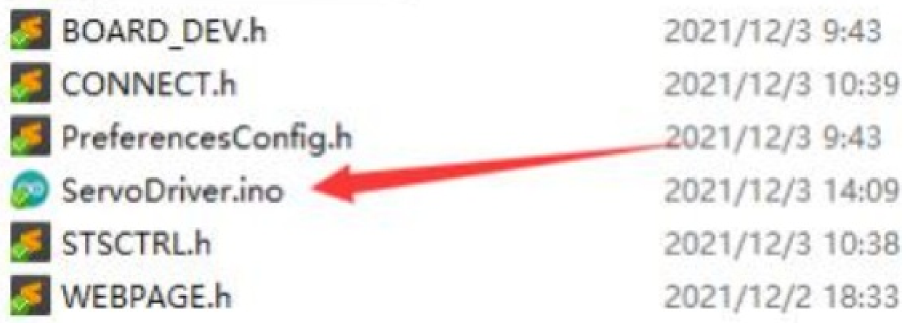
3. Enter the libraries that need to be installed through the library manager in the search box, and install: Adafruit SSD1306, Adafruit NeoPixel
4. Click “Update”, and if there is no “Update”, please click “Install” for the newest version.
5. Click Github link to download the program, or you can go to waveshape to download it.
6. Copy SCServo to \Documents\Arduino\libraries.



Adafruit_BusIO	2021/8/31 15:16
Adafruit_GFX_Library	2021/8/31 15:16
Adafruit_NeoPixel	2021/8/17 9:04
Adafruit_PWM_Servo_Driver_Library	2021/8/16 18:14
Adafruit_SSD1306	2021/11/30 13:50
ArduinoJson	2021/11/30 13:49
AsyncTCP	2021/8/18 17:13
ESP8266_and_ESP32_OLED_driver_for...	2021/8/31 15:16
ESPAsyncWebServer	2021/8/18 17:14
FabGL	2021/10/14 14:01
ICM20948_WE	2021/11/30 13:52
INA219_WE	2021/11/30 13:55
PCA9685	2021/8/16 9:06
SCServo	2021/9/13 9:46

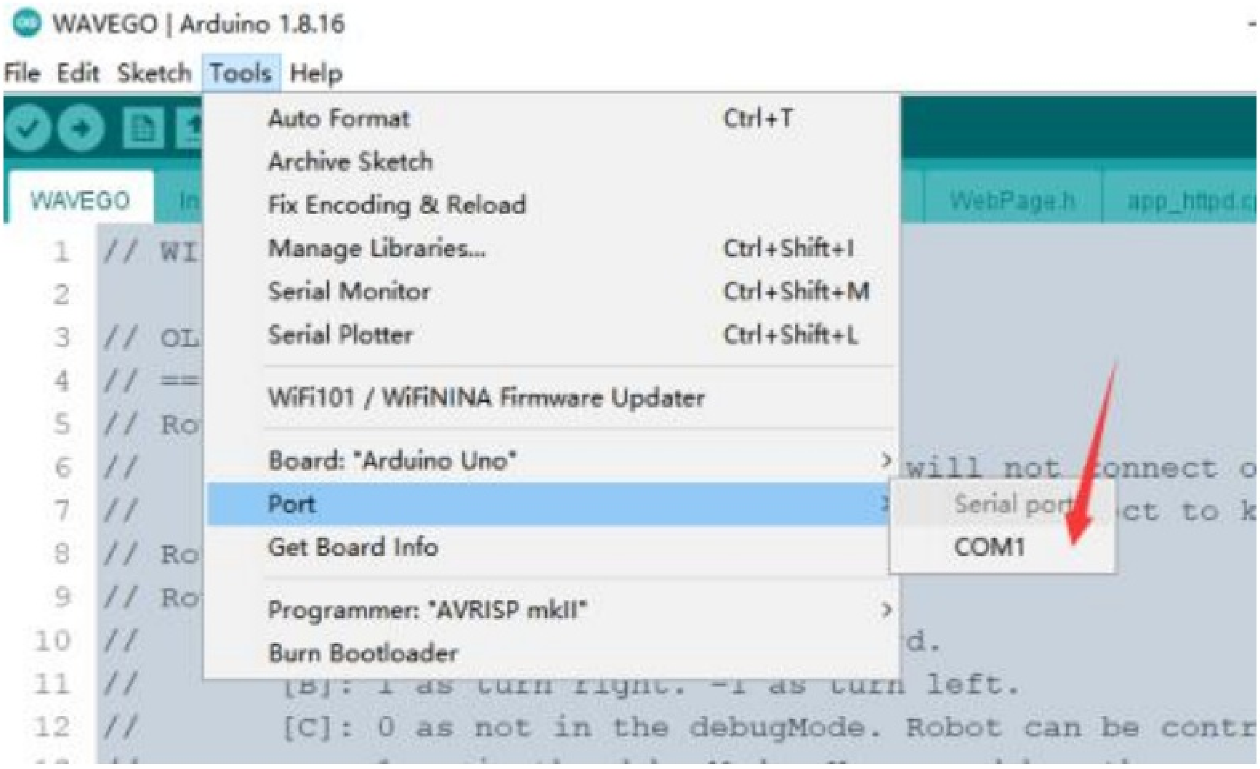
UPLOAD THE PROGRAM TO THE ROBOT

1. Double-click to run SERVO DRIVER WITH ESP32 ST/ServoDriverST/ServoDriverST.ino.

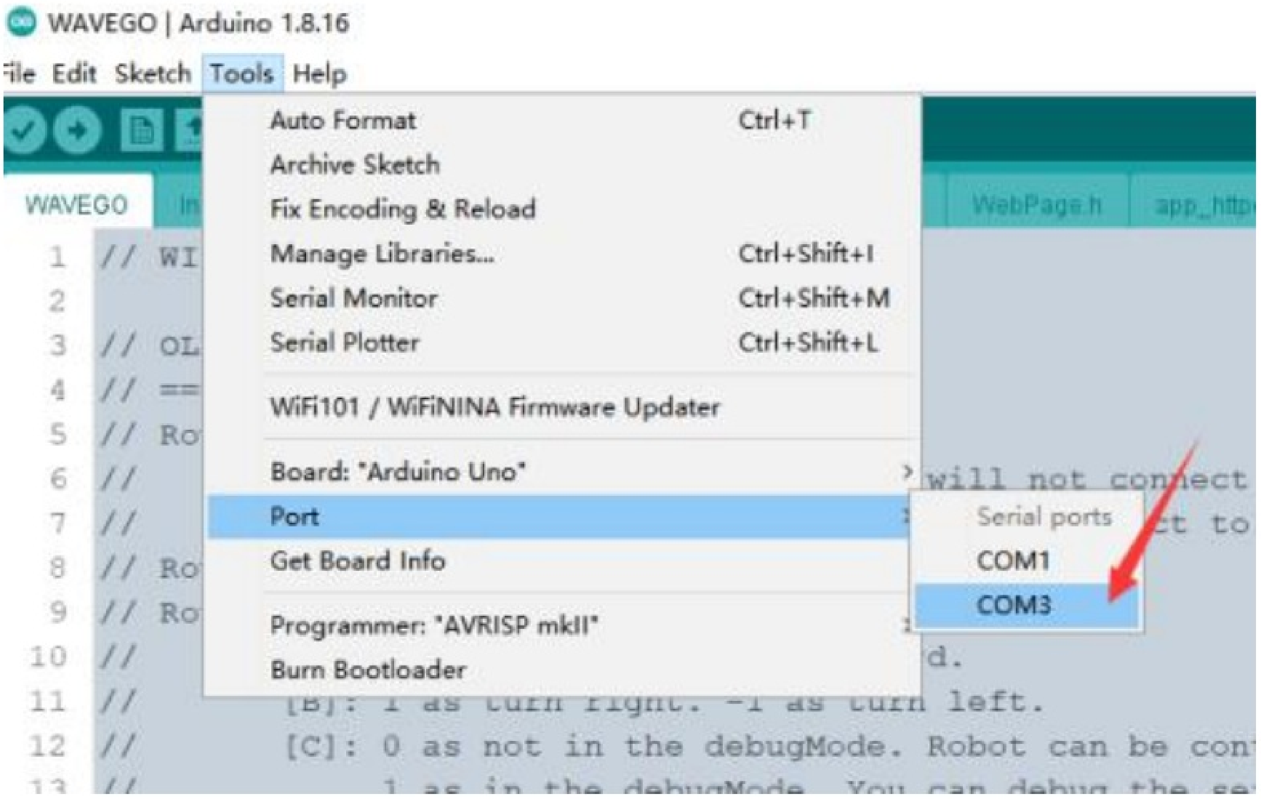


BOARD_DEV.h	2021/12/3 9:43
CONNECT.h	2021/12/3 10:39
PreferencesConfig.h	2021/12/3 9:43
ServoDriver.ino	2021/12/3 14:09
STSCTRL.h	2021/12/3 10:38
WEBPAGE.h	2021/12/2 18:33

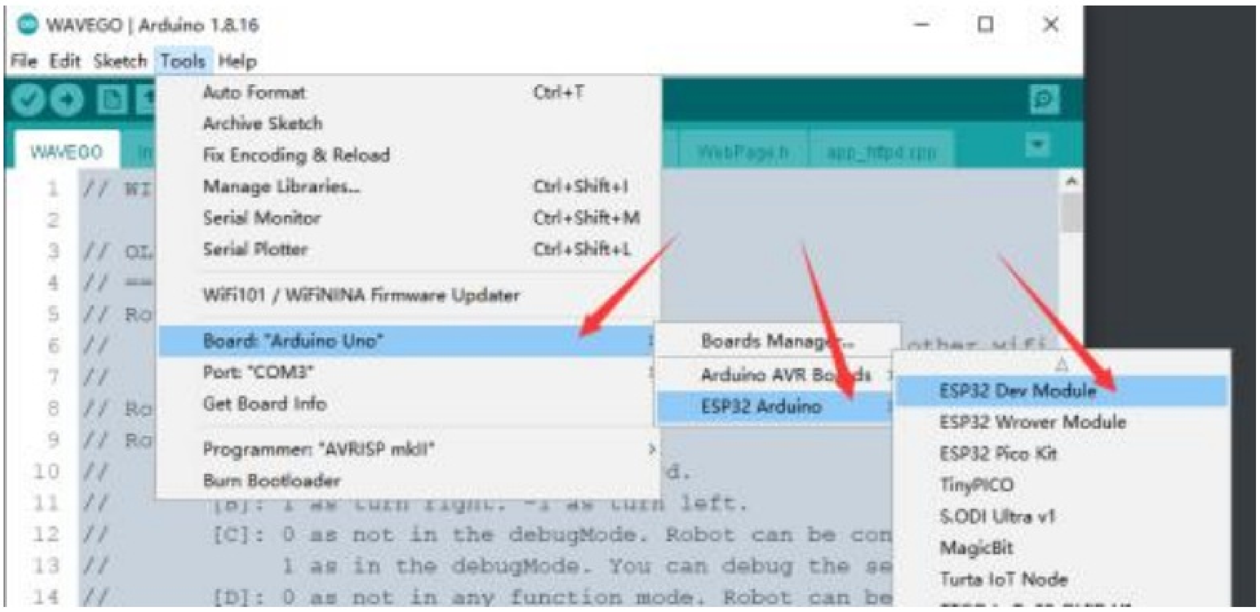
2. Click Tools > Port to remember the existing COM Port, no need to click it.
3. The COM displayed here is different for different computers, remember this COM.


WAVEGO | Arduino 1.8.16
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino Uno"
Port
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Serial ports
COM1

4. Connect the driver board and computer.
5. Click Tools > Port, and click on this new COM.
6. The newly-appearing COM is different for different computers, click it.


WAVEGO | Arduino 1.8.16
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino Uno"
Port
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Serial ports
COM1
COM3

7. Click Tools > Boards: > ESP32 Arduino > ESP32 Dev Module and select ESP32 Dev Module.
8. Other settings are as follows:

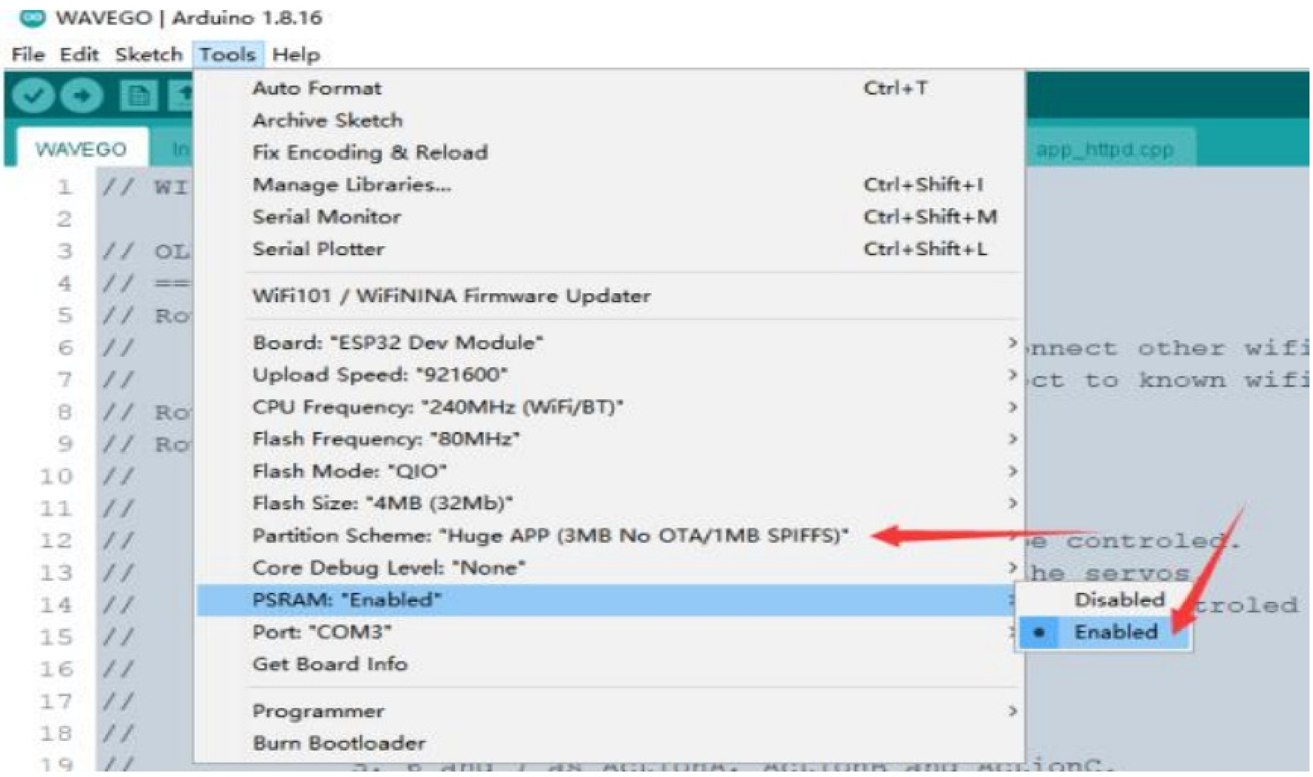


```
//                [B]: 1 as turn right. -1 as turn left.
//                [C]: 0 as not in the debugMode. Robot can be con
//                     1 as in the debugMode. You can debug the se
//                [D]: 0 as not in any function mode. Robot can be
```

9. Click “Upload” in the upper left corner to upload the program.

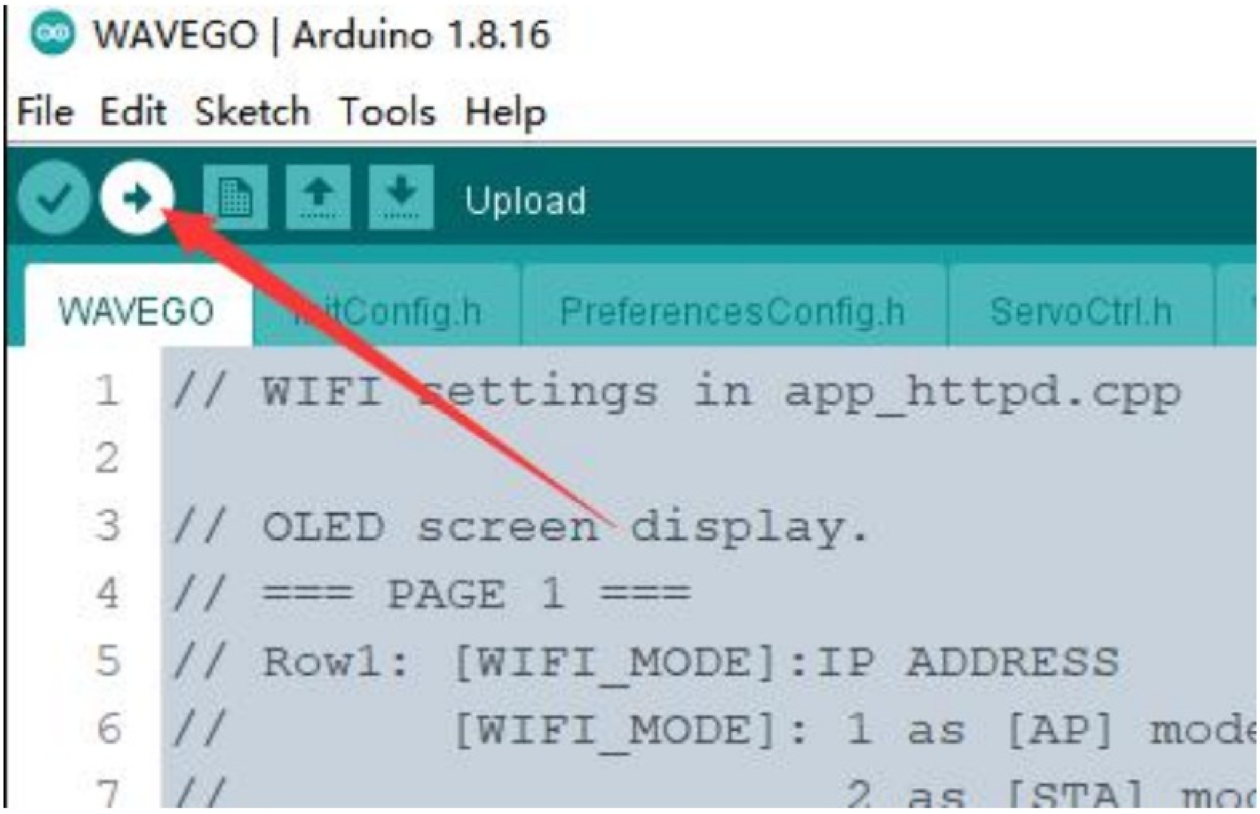
```
Upload Speed: "921600"

CPU Frequency: "240MHz(WiFi/BT)"

Flash Frequency: "80MHz"

Flash Mode: "QIO"

Flash Size: "4MB(32Mb)"

'''Partition Scheme: "Huge APP(3MB No OTA/1MB SPIFFS)"

PSRAM: "Disabled"'''
```

10. Note that the automatic downloading circuit will fail when WAVEGO uses the serial port to connect with the Raspberry Pi. You need to disconnect the serial port from the Raspberry Pi before uploading the program to WAVEGO.



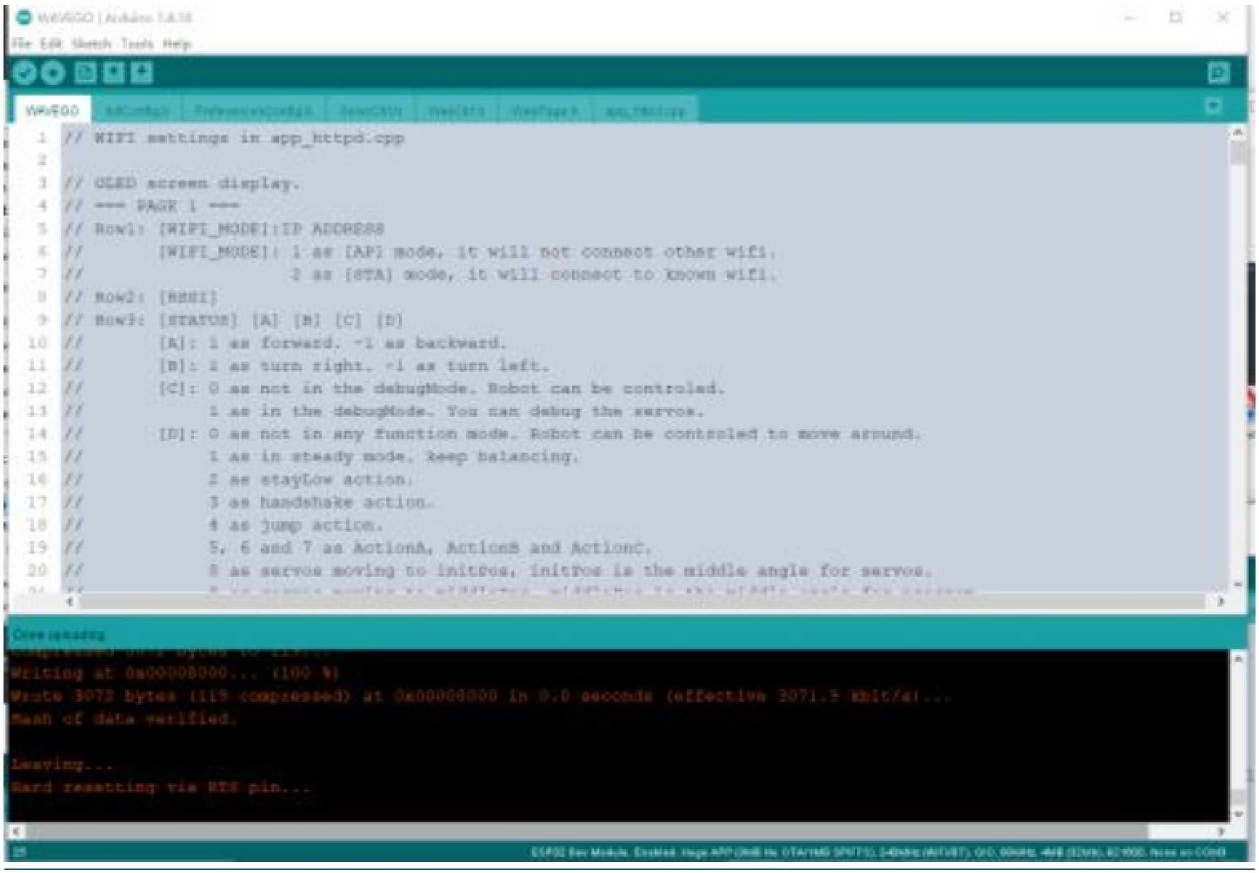
```
Board: "ESP32 Dev Module"
Upload Speed: "921600"
CPU Frequency: "240MHz (WiFi/BT)"
Flash Frequency: "80MHz"
Flash Mode: "QIO"
Flash Size: "4MB (32Mb)"
Partition Scheme: "Huge APP (3MB No OTA/1MB SPIFFS)"
Core Debug Level: "None"
PSRAM: "Enabled"
Port: "COM3"
```

11. Wait for uploading the program.



```
// WIFI settings in app_httpd.cpp

// OLED screen display.
// === PAGE 1 ===
// Row1: [WIFI_MODE]:IP ADDRESS
//          [WIFI_MODE]: 1 as [AP] mode
```

12. After displaying “Leaving… Hard resetting via RTS pin…”, it means that the upload has been successful.

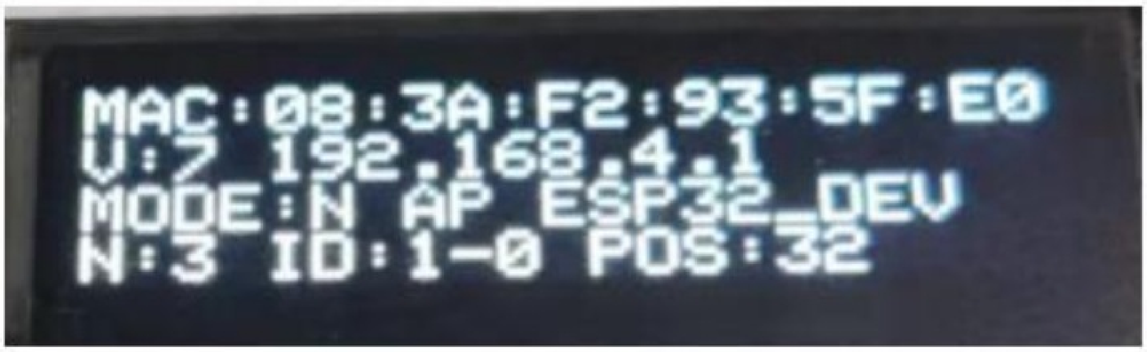


```
// WIFI settings in app_httpd.cpp

// OLED screen display.
// ---- PAGE 1 ----
// Row1: [WIFI_MODE]:IP ADDRESS
//       [WIFI_MODE]: 1 as [AP] mode, it will not connect other wifi.
//                    2 as [STA] mode, it will connect to known wifi.
// Row2: [RSSI]
// Row3: [STATUS] [A] [B] [C] [D]
//       [A]: 1 as forward. -1 as backward.
//       [B]: 1 as turn right. -1 as turn left.
//       [C]: 0 as not in the debugMode. Robot can be controled.
//            1 as in the debugMode. You can debug the servos.
//       [D]: 0 as not in any function mode. Robot can be controled to move around.
//            1 as in steady mode. keep balancing.
//            2 as stayLow action.
//            3 as handshake action.
//            4 as jump action.
//            5, 6 and 7 as ActionA, ActionB and ActionC.
//            8 as servos moving to initPos, initPos is the middle angle for servos.
```

```
Done uploading
Writing at 0x00008000... (100 %)
Wrote 3072 bytes (119 compressed) at 0x00008000 in 0.0 seconds (effective 3071.5 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

HOW TO USE THE DRIVER BOARD

1. Connect the SC15 Servo to the driver board. There are two 3pin servo ports on the driver board, which are connected, so you can press any one of them.
2. Use 6-8.4V DC power supply, the driver board will automatically turn on after plugging in the power supply.

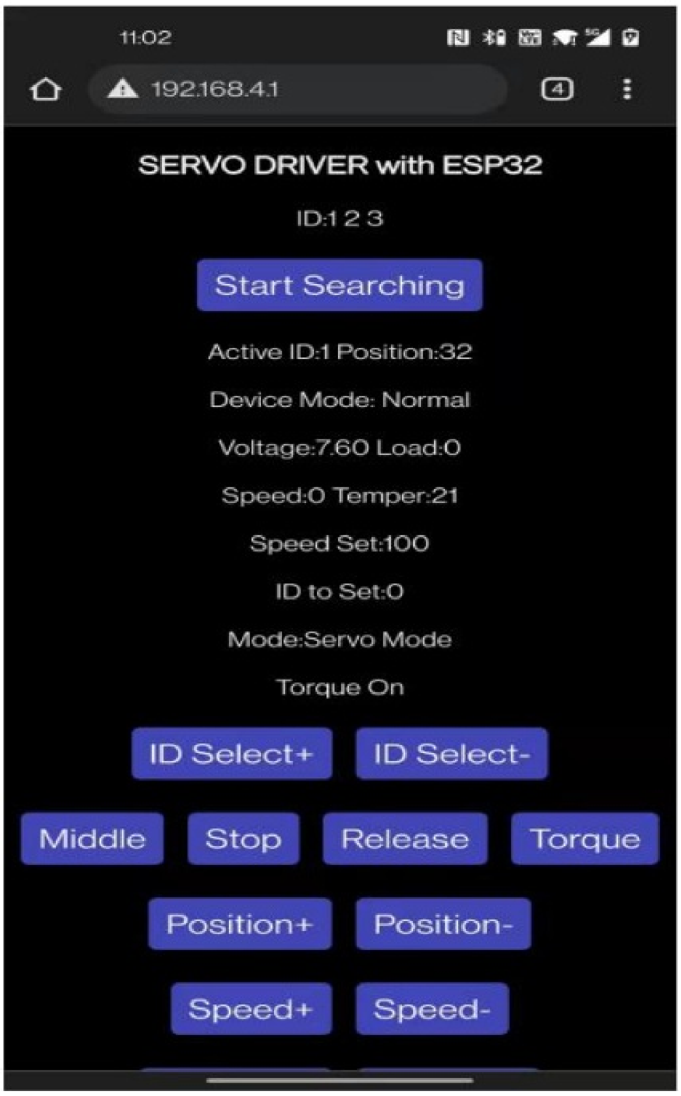


```
MAC:08:3A:F2:93:5F:E0
V:7 192.168.4.1
MODE:N AP ESP32_DEV
N:3 ID:1-0 POS:32
```

- When powered on, the RGB blue light indicates that the driver board starts to initialize WIFI and establish a WebServer. After completion, the driver board will automatically scan the connected servos. The RGB light is green during scanning, and then the RGB will turn off.
- After power on, the first line of the OLED is the MAC address of the device, which is unique and used for ESP-NOW communication.
- The second row V: is the measured voltage. After the next is the device IP address.
- The third line MODE: is the role of the device, N is Normal, ESP-NOW related functions are turned off; L is Leader, ESP-NOW will send the current Active Servo ID and location information to the Follower, and control the servo with the same ID connected to the Follower. F is Follower, the device will receive commands from Leader through ESP-NOW.
- The role of the device is followed by AP or STA. In AP mode, the device will establish a WIFI hotspot; in STA mode, the device will connect to a known WIFI.

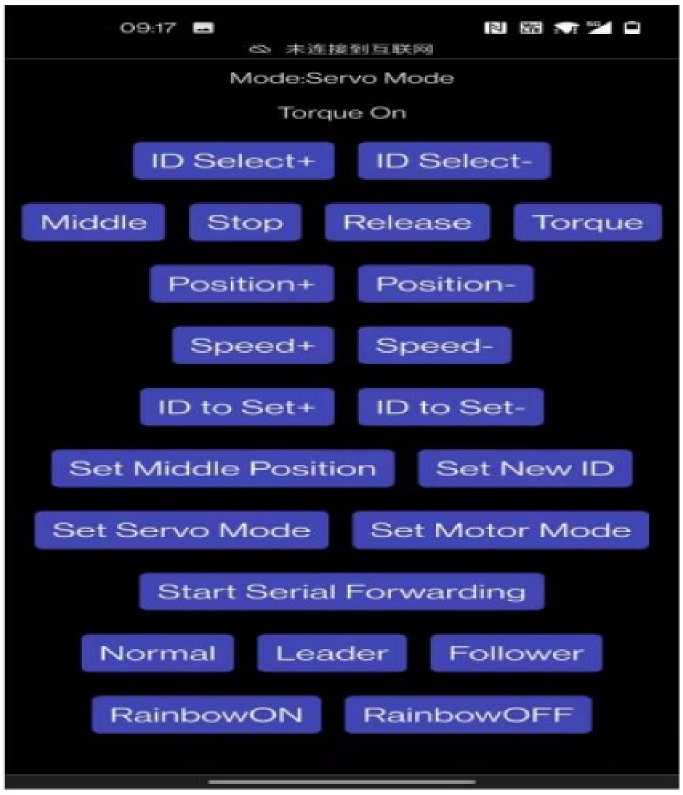
- In AP mode, the AP is followed by the SSID of the device; in STA mode, the STA is followed by the signal strength RSSI.
- The first line N: is Number, which shows the number of servos connected in the last scan; ID: is the currently selected servo (Active Servo), the -0 behind the servo indicates that the servo is in servo mode; – 3 Indicates that the servo is in motor mode. POS: is the current position of the servo.

3. Download and install the Chrome browser on your phone, or other browsers with the same kernel.
4. In the default program, after the driver board is powered on, a WIFI hotspot will be created automatically, and the mobile phone will be used to search for the WIFI hotspot.
5. In the default program, the WIFI name is ESP32_DEV and the WIFI password is 12345678 .
6. After the connection is successful, open the browser and enter 192.168.4.1 in the address bar to open the ESP32 web interface.
   - Generally, if the servo is connected first and then powered on, the driver board will scan the connected servo. If you turn on the servo first and then connect to the servo, you need to manually click Start Searching on the WEB interface of the mobile phone, and then wait for a while for the servo to scan, including re-scanning the servo after resetting the servo ID.


11:02
192.168.4.1
SERVO DRIVER with ESP32
ID:1 2 3
Active ID:1 Position:32
Device Mode: Normal
Voltage:7.60 Load:0
Speed:0 Temper:21
Speed Set:100
ID to Set:0
Mode:Servo Mode
Torque On

7. If you only connect one servo, the ID Select+ and ID Select- buttons have no effect; if you connect multiple servos with different IDs, you can use these two buttons to select the servo, the currently selected servo ID is displayed after ID. If you connect multiple servos with the same ID, it is very likely that the ping will fail, resulting in no servo ID to choose from the driver board. Hence, you need to disconnect the redundant connections. Only connect one servo, after changing the ID, connect it in series with the servos with other IDs.
8. “Middle” button, after pressing the servo, the servo will rotate to the middle position of the servo swing range.

9. “Stop” button, the servo will stop turning after pressing it.
10. “Release” button, press the servo to turn off the torque lock, and people can turn the servo by hand (due to the existence of multi-stage reduction gears, it may be very ifficult to rotate the servo between them, and a lever arm can be used to rotate).
11. “Torque” button, press it to open the torque lock, when receiving external force, the servo will try to keep it in one position.
12. “Position+” , the servo position value increases after pressing, the servo starts to rotate clockwise, and the rotation stops when released.
13. “Position-” , the servo position value decreases after pressing, the servo starts to rotate counterclockwise, and the rotation stops when released.
14. “Speed+ Speed-” , press to increase/decrease the speed of the servo, the changed speed is displayed after “Speed Set”, the moving speed of the servo will use this speed.
15. “ID to Set+ ID to Set- Set New ID”, used to select the servo ID to be changed, displayed after ID to Set:. For example, when you need to change the servo with ID 3 to the servo with ID 9, you need to click “ID to Select” first to change the display area.ID: to 3 (provided that the scanned servo has an ID of 3), and then use the ID to Set button to change the number behind to 9, then click Set New ID, click OK in the pop-up dialog box, so that the ID3 servo is set to ID9 Right, now you need to click Start Searching again to find the servos.
16. “Set Middle Position”, dedicated to ST series, SC series servos do not have this function, this button is used to set the currently selected servo position as the middle position of the servo.


09:17
未连接到互联网
Mode:Servo Mode
Torque On

17. “Set Servo Mode”, set the current servo as the servo mode, which is also the initial default mode of the servo and can be rotated to an absolute position.
18. “Set Motor Mode”, set the current servo to motor mode. In this mode, the servo can rotate continuously and control the relative position of each rotation. It can be used as a geared motor (SC series servo) or a stepper motor (ST series steering gear).
19. “Start Serial Forwarding”, after the servo is turned on, the servo will start the serial port forwarding, and forward the data of the Type-C UART0 and the UART1 used to control the servo, so that the FD tool on the computer

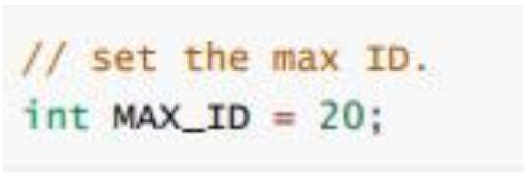
can be used to debug the servo. The default baud rate of UART0 is 115200.

20. “Normal”, set the current device to Normal, the related functions of ESP-NOW will not be enabled in this mode.
21. “Leader”, set the current device as Leader , in this mode, the device will send the current Active Servo’s ID and Position to Follower through ESP-NOW.
22. “Follower”, set the current device as Leader, in this mode, the device will receive commands from Leader through ESP-NOW.
23. “RainbowON RainbowOFF”, used to turn on and off the RGB rainbow light, you can expand more WS2812 RGB LEDs on the RGB light interface.

AT COMMAND SET

SET THE LARGEST SERVO ID

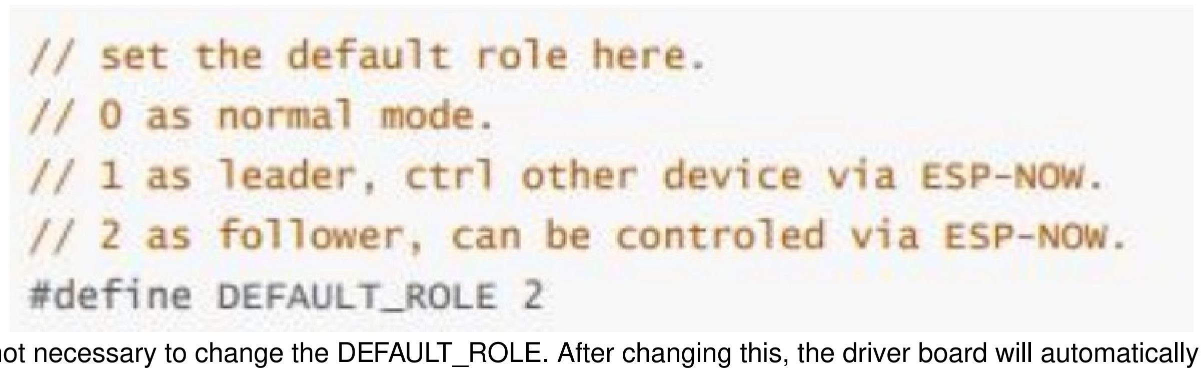
The method of scanning the currently connected servos each time is to ping each ID. Ping through a certain ID means that the servo with this ID is connected to the driver board, so if the number of IDs to Ping is too many, it will take a long time. The maximum ID of the scanning servo, MAX_ID, is defined in ServoDriver.ino. If you control many servos or the servo ID value is relatively large, you can adjust the number of IDs to be pinged in each scan by changing this value. The maximum is 253, the default is 20.



```
// set the max ID.
int MAX_ID = 20;
```

SET MAC ADDRESS

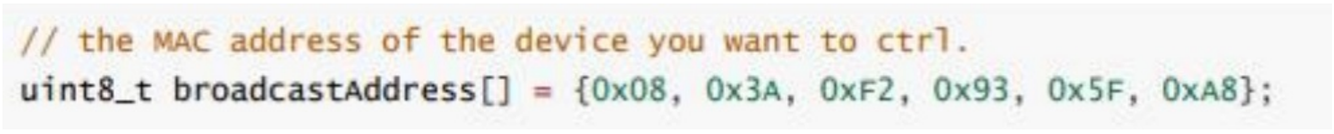
- ESP-NOW is used to communicate among the driver boards and the default program supports one-to-one control. You can follow the tutorials later in this section.
  One-to-many or many-to-one communication.
- The premise of ESP-NOW communication is to get the MAC address of the receiver. After the driver board is powered on, it will automatically obtain the MAC address of the device and display it in the first row of the OLED screen.
- The working mode of the driver board can be set to Normal Leader Follower, when you use one driver board to control another, the control terminal of the driver board is the Leader, and the other controlled driver board is the Follower. The specific operation steps are as follows:
- Connect the Follower to the computer and note the Follower’s MAC address.
- Open ServoDriver.ino, change DEFAULT_ROLE to 2, and upload it to the Follower driver board.



```
// set the default role here.
// 0 as normal mode.
// 1 as leader, ctrl other device via ESP-NOW.
// 2 as follower, can be controled via ESP-NOW.
#define DEFAULT_ROLE 2
```

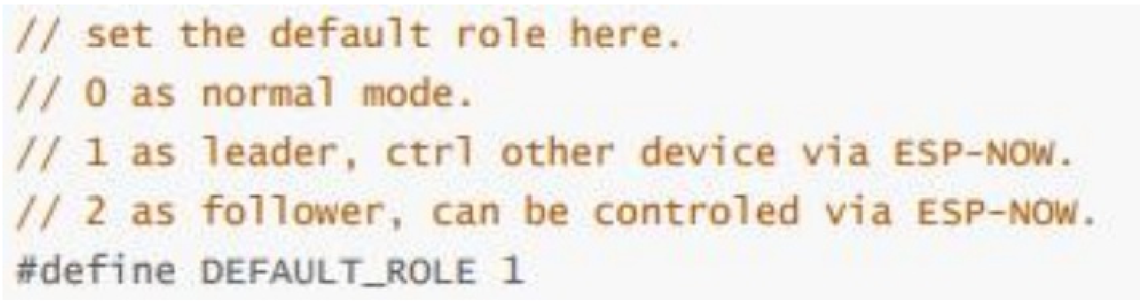
- It is not necessary to change the DEFAULT_ROLE. After changing this, the driver board will automatically be in the corresponding working mode after booting. If you do not change this, you need to manually set it on the WEB page. When the driver boards with the same SSID are running in AP mode at the same time, you may not be able to distinguish which name corresponds to the driver board. For WIFI settings, please refer to the WIFI: AP and STA Mode chapter.

- Connect the Leader driver board to the computer, open ServoDriver.ino , change broadcastAddress[] to Follower's MAC address, and remember to add 0x before each value.



```
// the MAC address of the device you want to ctrl.
uint8_t broadcastAddress[] = {0x08, 0x3A, 0xF2, 0x93, 0x5F, 0xA8};
```

- The above is an example, in practice the MAC address of each device is different.
- Change DEFAULT_ROLE to 1 and upload it to the Follower driver board.



```
// set the default role here.
// 0 as normal mode.
// 1 as leader, ctrl other device via ESP-NOW.
// 2 as follower, can be controled via ESP-NOW.
#define DEFAULT_ROLE 1
```

- After the two devices are connected to the servos with the same ID, the ID and Position of the Leader's Active Servo will be sent to the Followers to control the rotation of the servos connected to them.
- The demo is only suitable for the servo to work in the servo mode. If it works in the motor mode, it does not work. The SC series servo cannot control the angle in the continuous rotation mode.
- When connecting the mobile phone to the driver board, pay attention to whether the SSIDs of the two driver boards have the same name. If they are the same name, it may lead to the connection to the wrong driver board.
- You can refer to the WIFI: AP and STA Mode section to change the SSID.
- **Related links for ESP-NOW:**
- ESP-NOW Two-Way Communication Between ESP32 Boards
- ESP-NOW with ESP32: Send Data to Multiple Boards (one-to-many)
- ESP-NOW with ESP32: Receive Data from Multiple Boards (many to one)

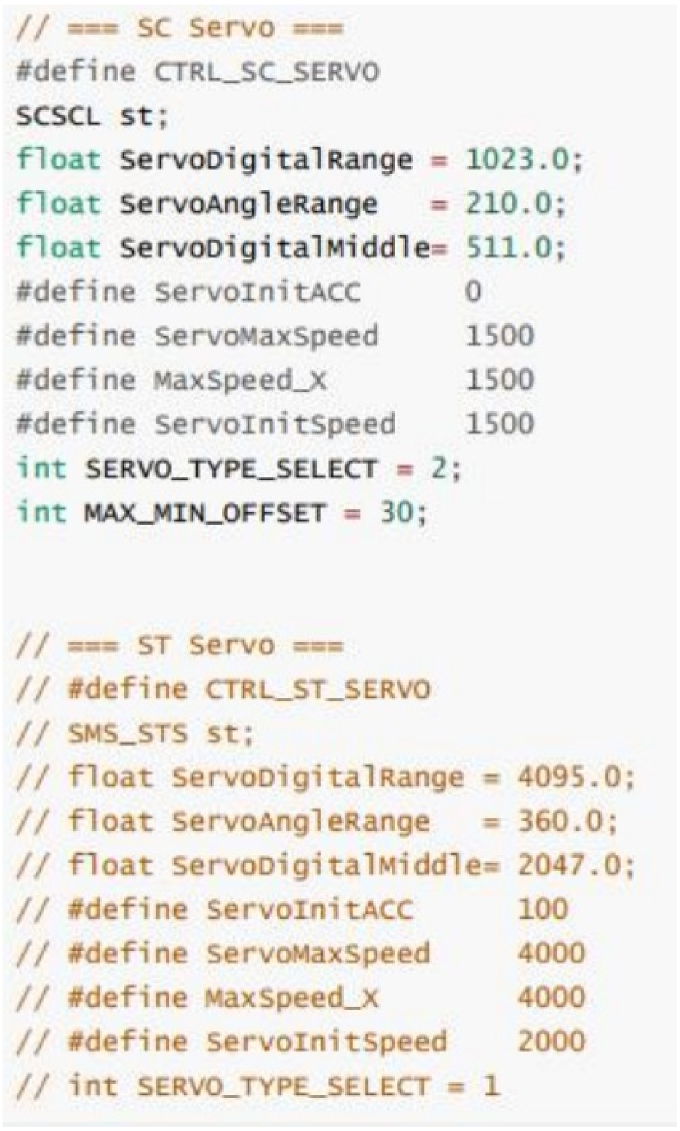
SERVO TYPES

The driver board can control the servo of the SC and ST series. The main differences between the two series of servo parameters are as follows:

	SC	ST
Digital Signal Range	0-1023	0-4095
Servo Rotation Angle	200°	360°
Voltage Range	6-8.4V	6-12V

- For other differences, please refer to the Datasheet on the servo sales page. The voltage is determined by the power supply of the DC port, and the DC port directly supplies power to the servo.
- In the program, you need to adjust the program according to your own servo type. The original program is suitable for SC series servos. If you are using SC series servos, you do not need to change. The relevant code is in STSCTRL.h, and SC is supported by default. The program of the series servo is as follows:

- If you need to control ST series servos, you need to comment out the part of SC servos and uncomment the part of ST series servos. The changed code is as follows:



```
// === SC Servo ===
#define CTRL_SC_SERVO
SCSCL st;
float ServoDigitalRange = 1023.0;
float ServoAngleRange   = 210.0;
float ServoDigitalMiddle= 511.0;
#define ServoInitACC      0
#define ServoMaxSpeed     1500
#define MaxSpeed_X        1500
#define ServoInitSpeed    1500
int SERVO_TYPE_SELECT = 2;
int MAX_MIN_OFFSET = 30;


// === ST Servo ===
// #define CTRL_ST_SERVO
// SMS_STS st;
// float ServoDigitalRange = 4095.0;
// float ServoAngleRange   = 360.0;
// float ServoDigitalMiddle= 2047.0;
// #define ServoInitACC      100
// #define ServoMaxSpeed     4000
// #define MaxSpeed_X        4000
// #define ServoInitSpeed    2000
// int SERVO_TYPE_SELECT = 1
```

WIFI: AP & STA MODE

- In AP mode, the device will establish a WIFI hotspot, which can be connected through the mobile phone to control/debug the servo.
- In STA mode, the device will connect to a known WIFI hotspot.
- You can customize the SSID and password of the WIFI hotspot established by the driver board in AP mode by changing the code in ServoDriver.in:

```
// === SC Servo ===
// #define CTRL_SC_SERVO
// SCSCL st;
// float ServoDigitalRange = 1023.0;
// float ServoAngleRange   = 210.0;
// float ServoDigitalMiddle= 511.0;
// #define ServoInitACC      0
// #define ServoMaxSpeed     1500
// #define MaxSpeed_X        1500
// #define ServoInitSpeed    1500
// int SERVO_TYPE_SELECT = 2;
// int MAX_MIN_OFFSET = 30;
```

```
// === ST Servo ===
#define CTRL_ST_SERVO
SMS_STS st;
float ServoDigitalRange = 4095.0;
float ServoAngleRange   = 360.0;
float ServoDigitalMiddle= 2047.0;
#define ServoInitACC      100
#define ServoMaxSpeed     4000
#define MaxSpeed_X        4000
#define ServoInitSpeed    2000
int SERVO_TYPE_SELECT = 1
```

- You can customize the SSID and password of the WIFI that the driver board will connect to in STA mode by changing the code in ServoDriver.in:
- Set AP mode, the default is AP mode in device code, change DEFAULT_WIFI_ROLE in ServoDriver.ino to 1.

```
// WIFI_STA settings.
const char* STA_SSID = "OnePlus 8";
const char* STA_PWD  = "40963840";
```

- To set STA mode, change DEFAULT_WIFI_ROLE in ServoDriver.ino to 2 .

```
// WIFI_AP settings.
const char* AP_SSID = "ESP32_DEV";
const char* AP_PWD  = "12345678";
```

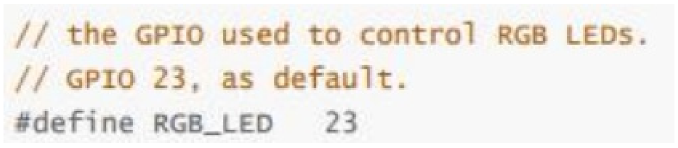
EXPAND MORE RGB-LED

There is a PH1.25-3P main seat on the back of the driver board, and there are silkscreens next to the three pins to mark the function of each pin, respectively 5V GND OUT. OUT is connected with other WS2812 5V IN, which is used to expand more RGB-LED lamp beads.
There are two on-board RGB LEDs on the driver board, and the code that controls them is in RGB_CTRL.h, numbered 0 and 1 respectively, if you expand more lamp beads, you need to modify the value of NUMPIXELS in ServoDriver.h.

```
#define NUMPIXELS 10
```

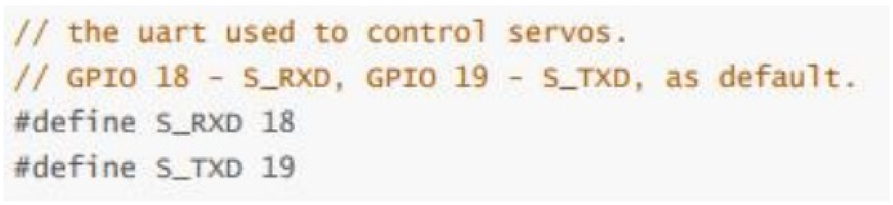
ESP32 PIN FUNCTION

- GPIO 23 is used to control RGB LEDs.



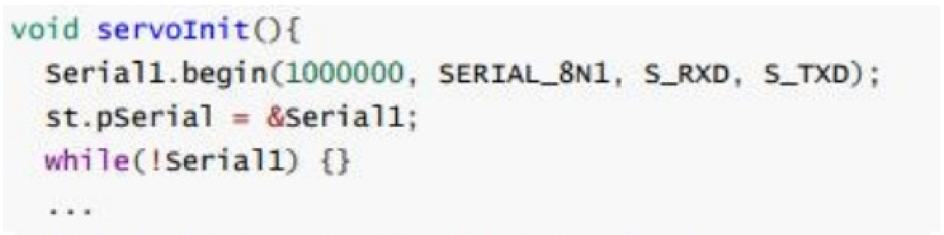
```
// the GPIO used to control RGB LEDs.
// GPIO 23, as default.
#define RGB_LED   23
```

- The GPIO 18 acts as RX and the GPIO 19 acts as TX, communicating with the servo control circuit.
- The baud rate for the UART controlled by the servo is 1,000,000

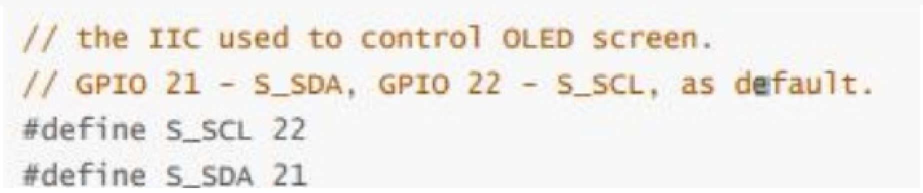


```
// the uart used to control servos.
// GPIO 18 - S_RXD, GPIO 19 - S_TXD, as default.
#define S_RXD 18
#define S_TXD 19
```

- The baud rate of the UART used for the download circuit and host computer communication is 115200.
- Used GPIO 21 as SDA, and GPIO 22 as SCL for controlling OLED screens.



```
void servoInit(){
  Serial1.begin(1000000, SERIAL_8N1, S_RXD, S_TXD);
  st.pSerial = &Serial1;
  while(!Serial1) {}
  ...
```



```
// the IIC used to control OLED screen.
// GPIO 21 - S_SDA, GPIO 22 - S_SCL, as default.
#define S_SCL 22
#define S_SDA 21
```

SECOND DEVELOPMENT

- The Example folder contains the demos for Arduino IDE, Jetson (Python language), and Raspberry Pi (Python language).
- You can use the Arduino IDE demo for secondary development directly on the driver board.
- If you want to use Jetson or Raspberry Pi to control the driver board, you have two options:

1. Control the driver board directly by Jetson or Raspberry Pi, we have provided a demo for this method.
2. Use the driver board as a lower computer, in which the inverse solution of the connecting rod is performed, and the upper computer is responsible for decision-making operations. We will provide relevant demos in specific products in the future.

ARDUINO IDE DEMO

- The demos for the Arduino IDE are in examples\arduinoIDE\ , all tested on SERVO DRIVER with ESP32, no need to change any code.
- The power supply of the servo is from the DC interface, so the DC interface should be connected to the power supply during the test.
- The demos of SC series servos are in the SCSCL folder.
- The demos of ST series servos are in the STSCL folder.

- SCSCL\Broadcast: broadcast to control all the servos. During the test, the servos will swing in a wide range at the maximum speed. Do not connect the servos with other structures.
- SCSCL\Ping: used to test whether the servo is ready, set the servo ID to be viewed by changing TEST_ID, and open the serial monitor to check the servo status during the test.

```
int TEST_ID = 3;
```

- SCSCL\ProgramEprom: Used to change the ID of the servo, change ID_ChangeFrom to ID_Changeto.

```
ID_ChangeFrom = 1;
ID_Changeto   = 2;
```

- SCSCL\RegWritePos: A demo for asynchronous writing, first set the target position and speed of each servo, and then call the following functions to start the movement uniformly.

```
RegWriteAction();
```

- SCSCL\SyncWritePos: A demo for synchronous writing to control multiple servos to move together.

```
st.CalibrationOfs(ID);
```

- SCSCL\WritePos: Used to control the movement of a single servo.
- SCSCL\FeedBack: used to obtain the position, speed, load, voltage, temperature, and movement status of the servo.
- STSCL\CalibrationOfs: The demo used to set the center position of the servo. After calling the following function, the current position of the servo will be used as the center position of the servo.
- STSCL\FeedBack: Used to obtain the position, speed, load, voltage, temperature, and movement status of the servo.
- STSCL\Ping: Used to test whether the servo is ready, set the servo ID to be viewed by changing TEST_ID, and open the serial monitor to check the servo status during the test.

```
int TEST_ID = 3;
```

- STSCL\ProgramEprom: Used to change the ID of the servo, change ID_ChangeFrom to ID_Changeto.

```
ID_ChangeFrom = 1;
ID_Changeto   = 2;
```

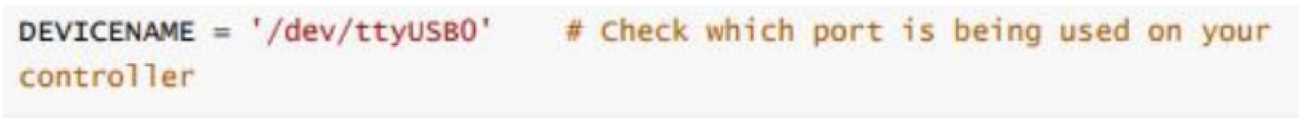
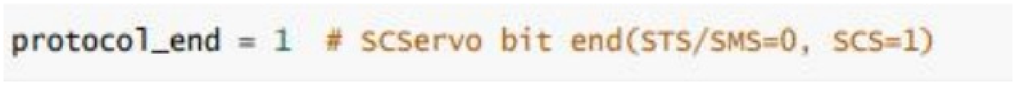
- STSCL\RegWritePos: A demo for asynchronous writing, first set the target position and speed of each servo, and then call the following functions to start the movement uniformly.

```
RegWriteAction();
```

- STSCL\SyncWritePos: A demo for synchronous writing to control multiple servos to move together.
- STSCL\WritePos: Used to control the movement of a single servo.

RASPBERRY PI/JETSON/PC HOST COMPUTER DEMO(PYTHON)

- Servo control demos in Python are in examples\python\.
- When you use this demo to directly control the servos, you need to enable the serial port transparent transmission mode of the driver board. You can operate through the
- WEB terminal and click the Start Serial Forwarding button to enable serial port transparent transmission. For convenience, you can also change the serial port in ServoDriver.ino. The SERIAL_FORWARDING is true, so the driver board will automatically enter the serial port transparent transmission mode after booting.

- When the driver board is in the serial port transparent transmission mode, use the baud rate of 115200 to communicate with the driver board.
- **ping.py:** ping command example
- **read_write.py:** common read and write example
- **sync_write.py:** Synchronous write example
- **sync_read_write.py:** Example of synchronous write and synchronous read
- When using the demo, you need to rewrite some codes according to your actual situation, such as the USB device name. For example Windows: "COM1", Linux: "/dev/ttyUSB0".
  
  ```
  DEVICENAME = '/dev/ttyUSB0'    # Check which port is being used on your
  controller
  ```
- protocol_end is used to select the servo type, 0 for ST series servos and 1 for SC series servos.
  
  ```
  protocol_end = 1  # SCServo bit end(STS/SMS=0, SCS=1)
  ```

FUNCTION TEST

- Double click to run ServoDriver.ino .
- Connect the driver board and click the arrow in the upper left corner of the Arduino IDE to upload the program to the driver board.
- After the upload is successful, disconnect the driver board, connect a servo, and use the DC port for power supply (power supply in the range of 6-8.4V can be used).
- The screen lights up, and the RGB lights emit light green light while scanning the servos.
- Use your phone to search for WIFI, name ESP32_DEV , password 12345678 .
- Open the Chrome browser on your phone (or other browsers with Chrome kernel) and visit 192.168.4.1 .
- Press Position+ or Position- to rotate the servo, if the servo rotates, the test is successful.
- *The content tested in the above steps includes: power supply, communication, steering gear control, OLED, RGB LED, and automatic download circuit.
  It is necessary to ensure that only one driver board is turned on near the mobile phone, otherwise it is not easy to determine which driver board is connected with the same WIFI name.
- **www.waveshare.com/wiki**

Documents / Resources

ST3215 Servo User Manual	**Waveshare ST3215 High Precision and Torque** [pdf] User Manual ST3215 High Precision and Torque, ST3215, High Precision and Torque, Precision and Torque

References

- **Arduino - Home**
- **Waveshare Wiki**

- dl.espressif.com/dl/package_esp32_index.json