AD174

ESP32 WiFi Camera 4WD Smart Robot Car Kit

V2.0

Preface

Our Company

KUONGSHUN Electronic Company is a supplier and manufacturer of electronic components, it is committed to board and starter kit for Arduino, Raspberry PI, Smart Robot Car, 3D printer. It is also a collection of scientific research, design, production, maintenance and sales of high-tech enterprises, in the field of automation with professional standards and mature technology, we rapid rise in the field of foreign trade.

Relying on technology and development, continuing to provide users with high-tech products, is our constant pursuit. Fully introduction of foreign advanced technology to enhance the value of our products.

Company gains users' praise for supplying first-class quality product and superb technical services, has now become the first choice of domestic and international procurement company.

Official Website: https://www.kuongshun.com

Our Tutorial

This course and learning kit is designed for 8+ children and teenagers to Arduino-compatible boards, shields, sensors, and components. The purpose of this tutorial is to introduce learners to how to program a trolley and how to manipulate a programmed trolley. As a common STEM educational tool, carts not only develop learners' hands-on skills, but also help them understand the basic concepts and logic of programming.

Customer Service

As a continuous and fast growing technology company we keep striving our best to offer you excellent products and quality service as to meet your expectation and you can reach out to us by simply drop a line at support@kuongshun.cn We look forward to hearing from you and any of your critical comment or suggestion would be much valuable to us.

And any of problems and questions you have with our products will be promptly replied by our experienced engineers within 12 hours (24hrs during holiday)

we pursue the policy of "progressive, Truth, Rigorous and Unity", keeping innovation, paying attention of technology as the core, committing to quality and putting customer's satisfaction on the priority, dedicated to provide you with the most cost-effective high-tech products and attentive service.

Packing list

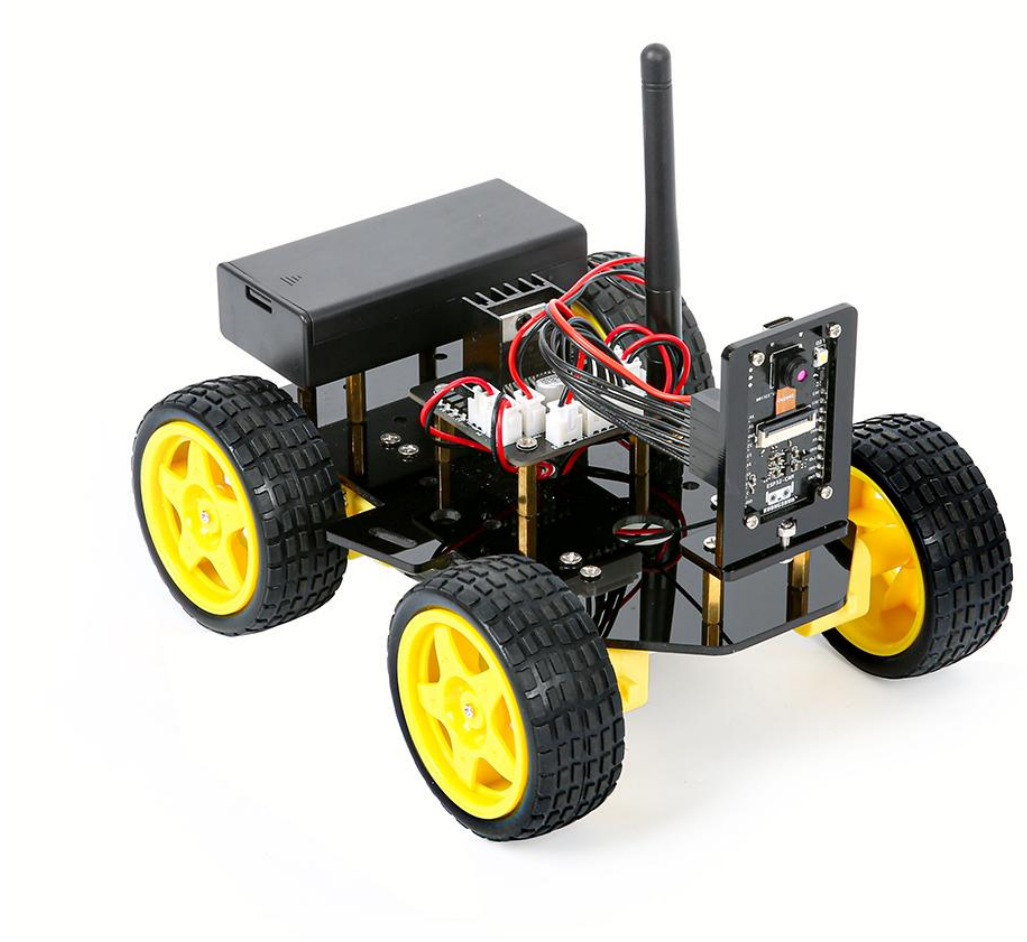
Content

Preparation 5
Lesson 1 Get The Car Moving 8
Lesson 2 WIFI Camera Car Code for Web 21
Lesson 3 WIFI Camera Car Code for APP 31
Frequently Asked Questions 41

Preparation

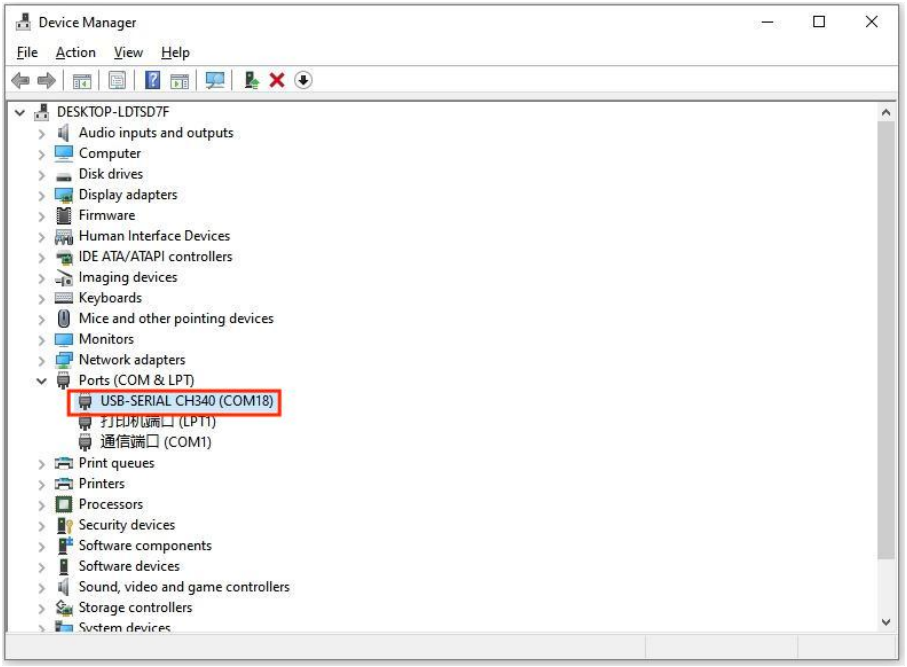
Before proceeding with this tutorial, please make sure you have completed the following five steps:

1.Complete the model assembly as described in "1_Assembly Guide," as shown in the image below: (If you purchased the version without batteries, please prepare or purchase two 18650 batteries separately.)



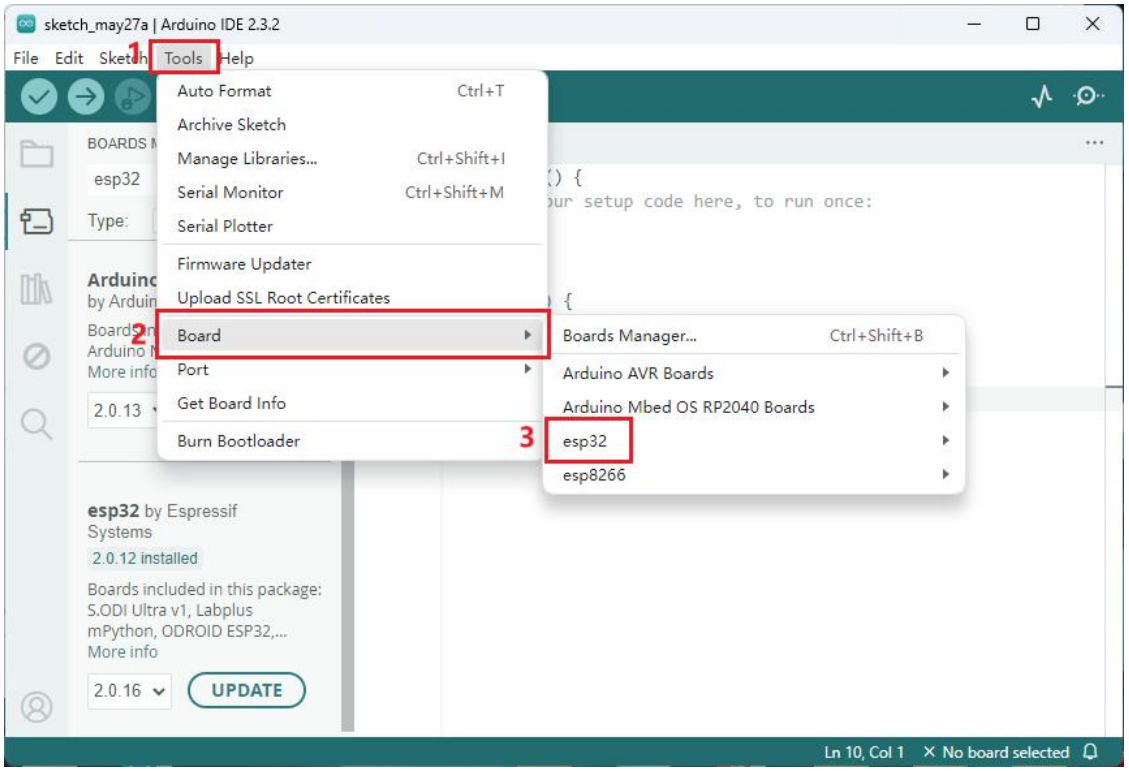
2.Install the Arduino software mentioned in "2_Programming Preparation," as shown in the image below.

3.Install the CH340 driver mentioned in "2_Programming Preparation," as shown in the image below.



4.Install the ESP32 board mentioned in "2_Programming Preparation," as shown in the image below.

Note: Please install version 2.0.12 (do not install versions above 3.0, as this may cause errors when uploading the program).



If any step has not been completed, refer to "1_Assembly_Guide or 2_Programming_Preparation".

Lesson 1 Get The Car Moving

In this project, we will use the ESP32 board and L298N driver board to enable the movement of the car. The car features a two-wheel drive system with an omni-wheel for motion control.

The car is powered by two independent drive wheels, which serve as the main power source, ensuring basic movements like forward and backward. These drive wheels rotate synchronously or asynchronously through precise motor control to meet the car's needs in different movement states.

The omni-wheel at the rear follows the movement of the drive wheels on both sides, providing balance and enabling full-range movement of the car.

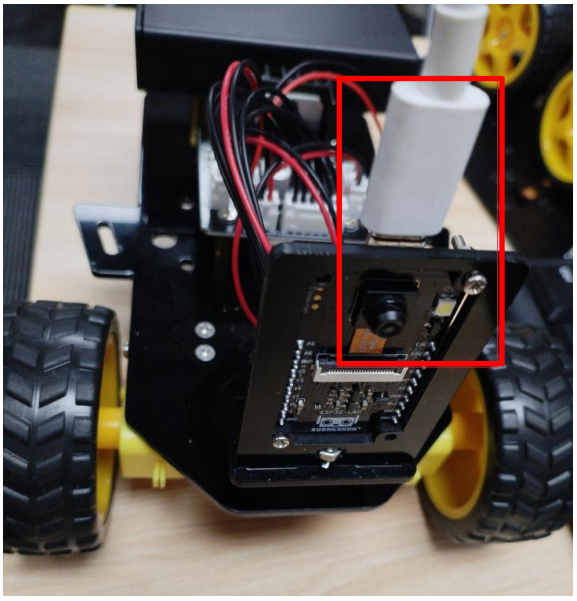
We will first learn how to upload the program to the car, and then later we will study the principles of motion and the program logic.

①Open the 'AUTO_GO.ino' program in 'KUONGSHUN AD174 ESP32 WiFi Camera 4WD Smart Robot Car Kit\3_Code_and_Tutorial\Make the Car Move\AUTO_GO'.

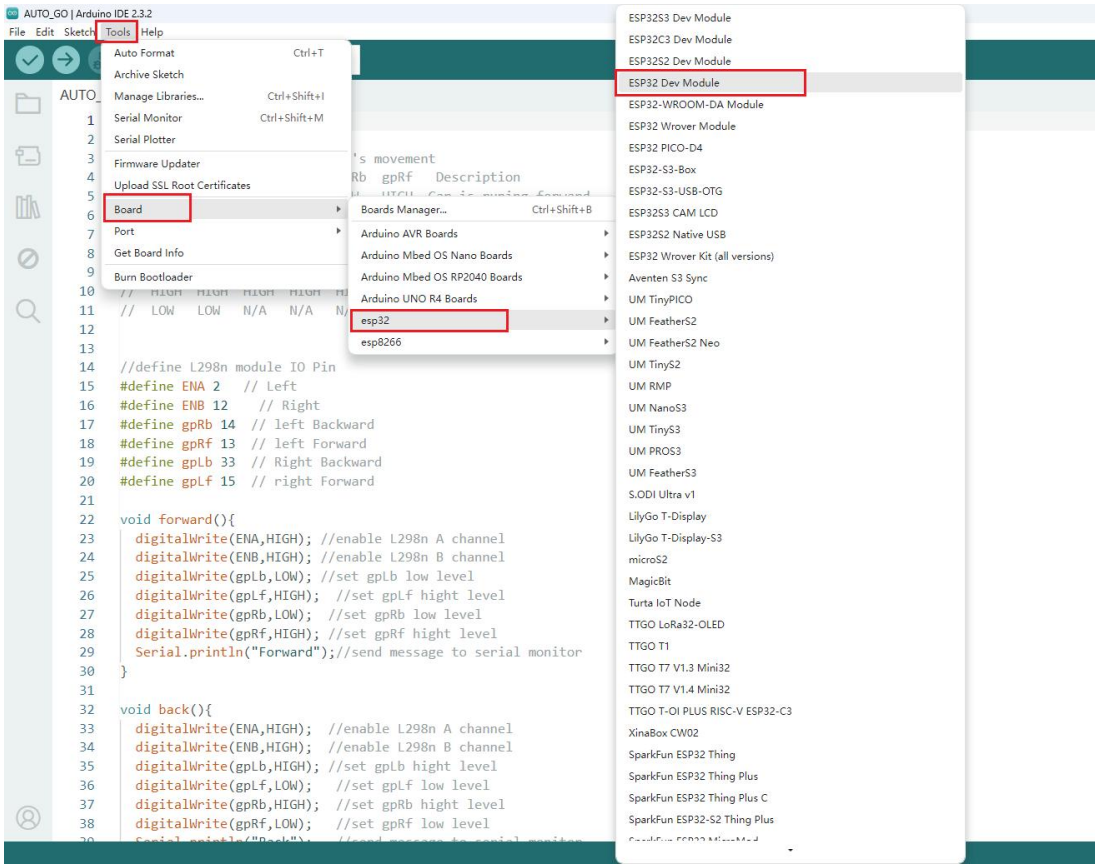
AUTO_GO.ino

```
//www.kuongshun.com

//    The direction of the car's movement
//  ENA   ENB   gpLb  gpLf  gpRb  gpRf   Description
//  HIGH  HIGH  LOW   HIGH  LOW   HIGH  Car is runing forward
//  HIGH  HIGH  HIGH  LOW   HIGH  LOW   Car is runing back
//  HIGH  HIGH  HIGH  LOW   LOW   HIGH  Car is turning left
//  HIGH  HIGH  LOW   HIGH  HIGH  LOW   Car is turning right
//  HIGH  HIGH  LOW   LOW   LOW   LOW   Car is stoped
//  HIGH  HIGH  HIGH  HIGH  HIGH  HIGH  Car is stoped
//  LOW   LOW   N/A   N/A   N/A   N/A   Car is stoped


//define L298n module IO Pin
#define ENA 2   // Left
#define ENB 12    // Right
#define gpRb 14  // left Backward
#define gpRf 13  // left Forward
#define gpLb 33  // Right Backward
#define gpLf 15  // right Forward

void forward(){
  digitalWrite(ENA,HIGH); //enable L298n A channel
  digitalWrite(ENB,HIGH); //enable L298n B channel
  digitalWrite(gpLb,LOW); //set gpLb low level
  digitalWrite(gpLf,HIGH);  //set gpLf hight level
  digitalWrite(gpRb,LOW);  //set gpRb low level
  digitalWrite(gpRf,HIGH); //set gpRf hight level
  Serial.println("Forward");//send message to serial monitor
}

void back(){
  digitalWrite(ENA,HIGH);  //enable L298n A channel
  digitalWrite(ENB,HIGH);  //enable L298n B channel
  digitalWrite(gpLb,HIGH); //set gpLb hight level
  digitalWrite(gpLf,LOW);   //set gpLf low level
  digitalWrite(gpRb,HIGH);  //set gpRb hight level
  digitalWrite(gpRf,LOW);   //set gpRf low level
```

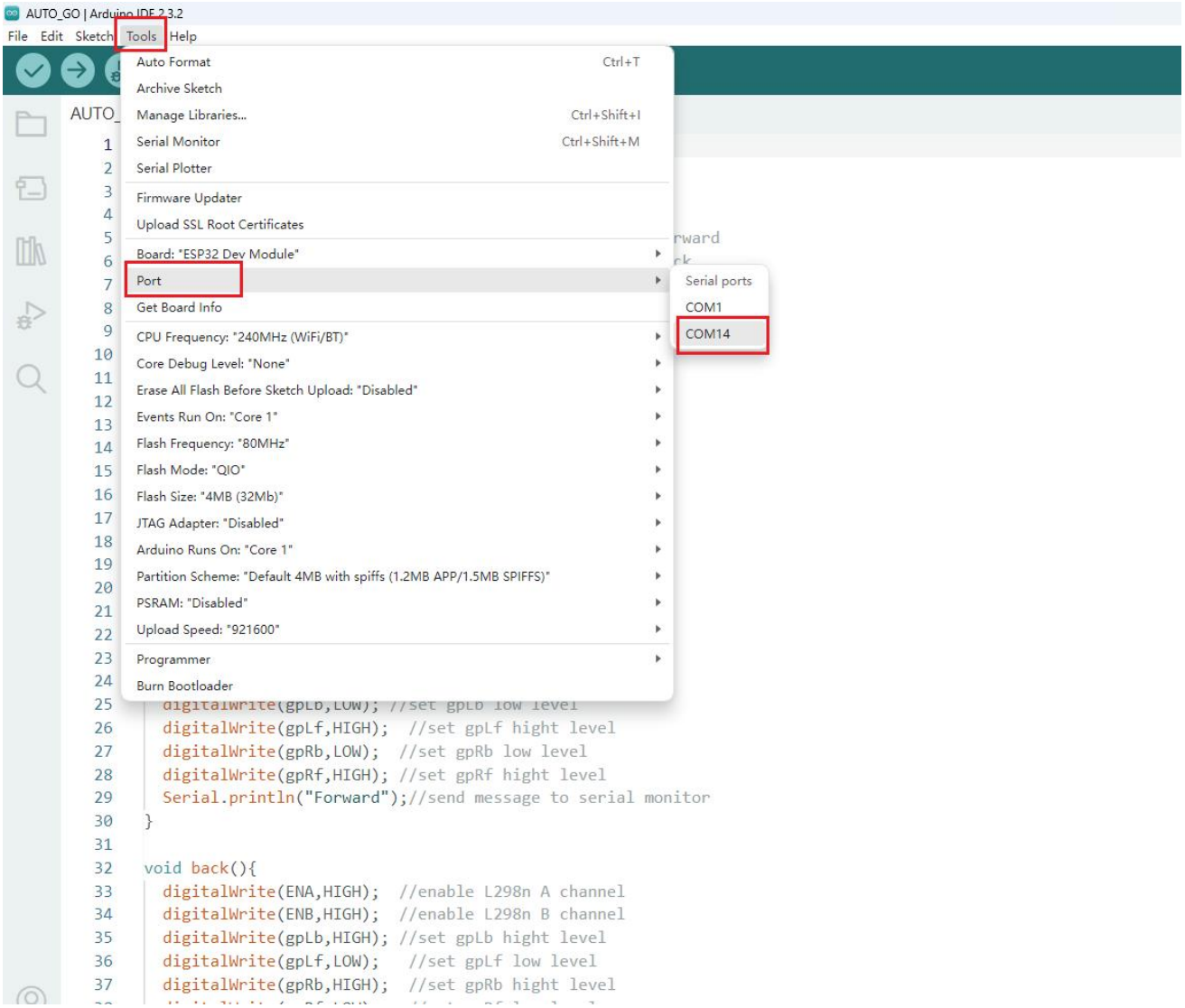
②Connect the ESP32-CAM controller board to your computer with a TYPE-C cable.



③ Select your board: Tools > Board > esp32 > ESP32 Dev Module (If you don't see esp32 in the Arduino IDE, you need to follow the steps in '2_Programming_Preparation' to download and add the ESP32 board.)



④ Select your port: Tools > Port >>>>> COMxx (The COM port number may vary on each computer.The port that disappears when unplugged and reappears when plugged in is your correct port.) (If you don't see any COM ports other than COM1 in Arduino IDE, you need to follow the steps in '2_Programming Preparation' to install the CH340 driver.)



KUONGSHUN AD175 ESP32 WiFi Camera 2WD Smart Robot Car Kit\2_Programming_Preparation

名称	修改日期	类型	大小
1. Installation of Arduino IDE and CH340 dri...	2025/1/6 22:19	文件夹	
2. Install the ESP32 plugin in the Arduino IDE	2025/1/6 22:20	文件夹	

⑤Upload the program to the ESP32 board.

AUTO_GO | Arduino IDE 2.3.2

File Edit Sketch Tools Help

ESP32 Dev Module

AUTO_GO.ino

```
//www.kuongshun.com

//    The direction of the car's movement
//  ENA   ENB   gpLb  gpLf  gpRb  gpRf   Description
//  HIGH  HIGH  LOW   HIGH  LOW   HIGH  Car is runing forward
//  HIGH  HIGH  HIGH  LOW   HIGH  LOW   Car is runing back
//  HIGH  HIGH  HIGH  LOW   LOW   HIGH  Car is turning left
//  HIGH  HIGH  LOW   HIGH  HIGH  LOW   Car is turning right
//  HIGH  HIGH  LOW   LOW   LOW   LOW   Car is stoped
//  HIGH  HIGH  HIGH  HIGH  HIGH  HIGH  Car is stoped
//  LOW   LOW   N/A   N/A   N/A   N/A   Car is stoped


//define L298n module IO Pin
#define ENA 2   // Left
#define ENB 12    // Right
#define gpRb 14  // left Backward
#define gpRf 13  // left Forward
#define gpLb 33  // Right Backward
#define gpLf 15  // right Forward

void forward(){
  digitalWrite(ENA,HIGH); //enable L298n A channel
  digitalWrite(ENB,HIGH); //enable L298n B channel
  digitalWrite(gpLb,LOW); //set gpLb low level
  digitalWrite(gpLf,HIGH);  //set gpLf hight level
  digitalWrite(gpRb,LOW);  //set gpRb low level
  digitalWrite(gpRf,HIGH); //set gpRf hight level
  Serial.println("Forward");//send message to serial monitor
}

void back(){
  digitalWrite(ENA,HIGH);  //enable L298n A channel
  digitalWrite(ENB,HIGH);  //enable L298n B channel
  digitalWrite(gpLb,HIGH); //set gpLb hight level
  digitalWrite(gpLf,LOW);   //set gpLf low level
  digitalWrite(gpRb,HIGH);  //set gpRb hight level
  digitalWrite(gpRf,LOW);   //set gpRf low level
```

Output

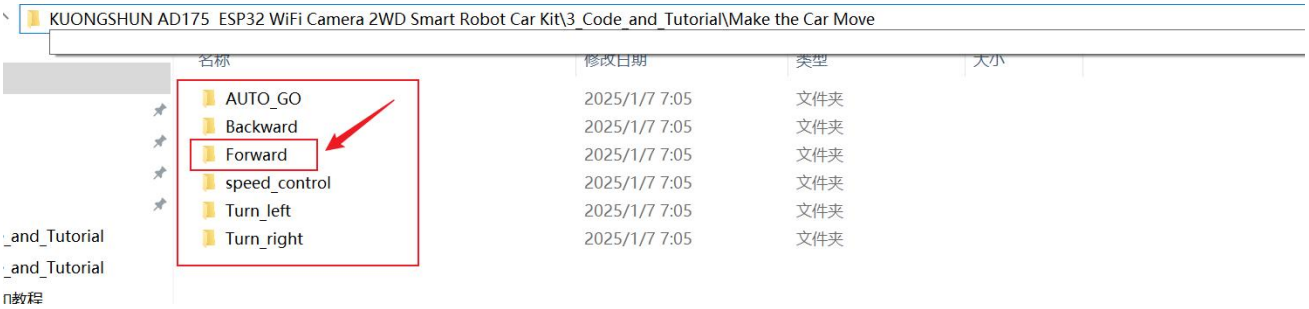
```
Writing at 0x0002958e... (50 %)
Writing at 0x0002ebcd... (62 %)
Writing at 0x0003712c... (75 %)
Writing at 0x0003f185... (87 %)
Writing at 0x00044818... (100 %)
Wrote 237232 bytes (130470 compressed) at 0x00010000 in 2.0 seconds (effective 968.3 kbit/s)...
Hash of data verified.

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

Ln 9, Col 26 ESP32 Dev Module on COM14 2

The picture above shows that it is uploaded successfully.

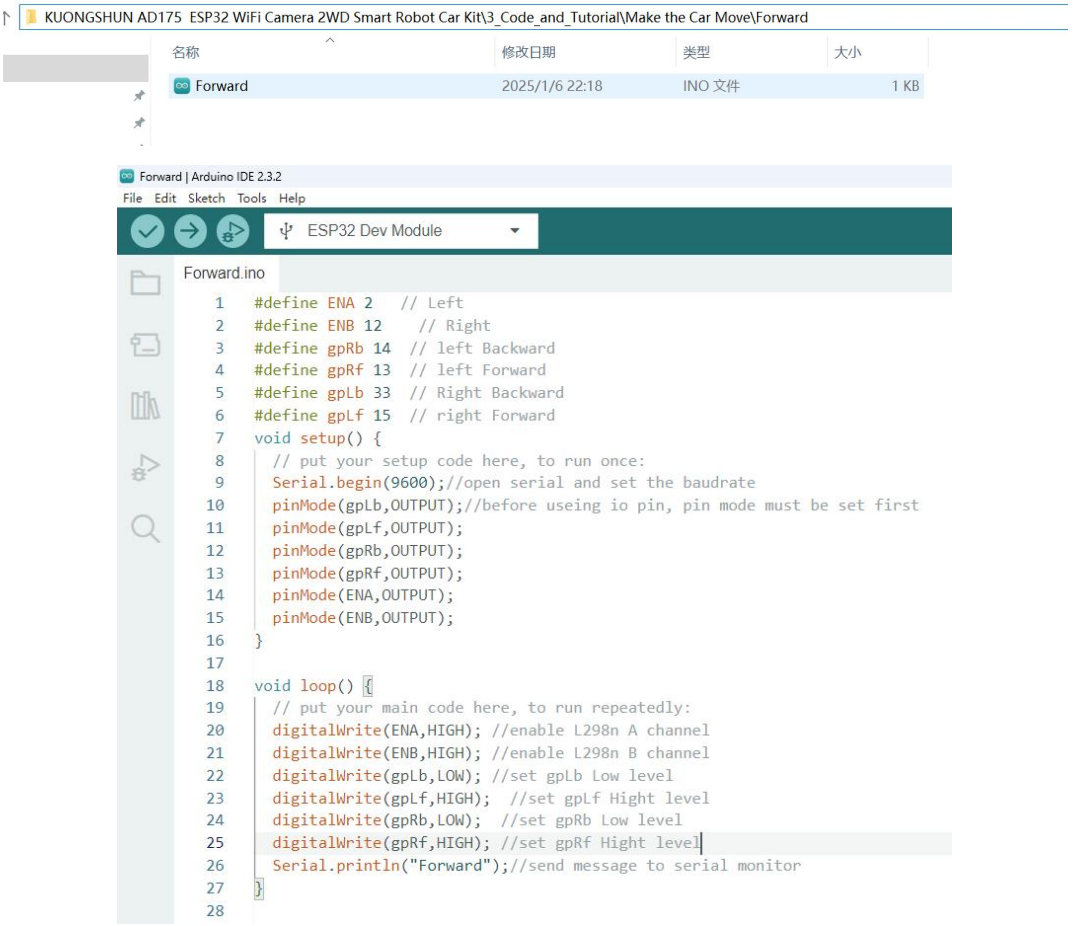
Disconnect the connection with the computer, then power on the car. The car will move forward, backward, turn left, and turn right for 1 second each in a loop. If the car does not move, please press the RESET button on the ESP32 board.

Next, let's upload another program to see the effect.



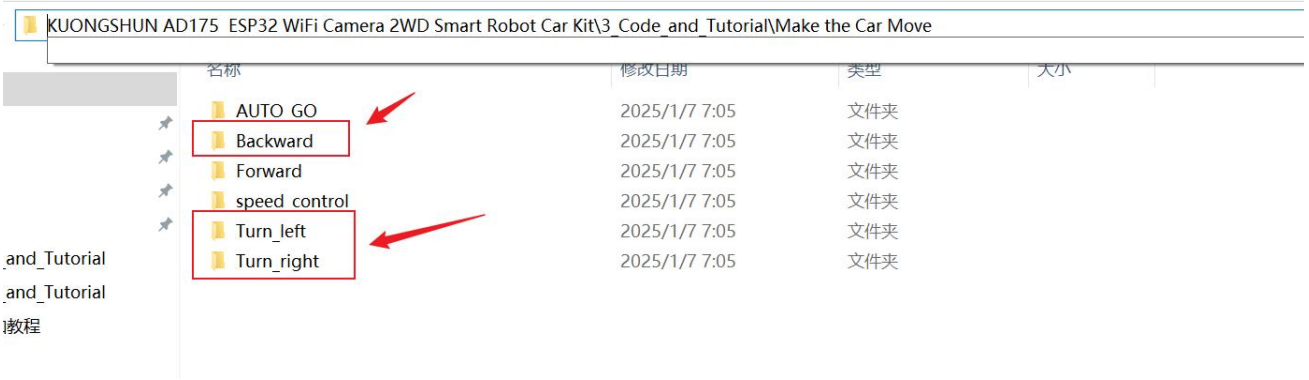
Go forward

Connect the ESP32-CAM controller board to the computer and open the '[Forward.ino]' program in '[KUONGSHUN AD174 ESP32 WiFi Camera 4WD Smart Robot Car Kit\3_Code_and_Tutorial\Make the Car Move\Forward]'. Upload the program to the ESP32 board (repeat the upload steps for the AUTO_GO program).



Disconnect the connection with the computer, then power on the car. The car will move forward continuously.

Similarly, we can upload the programs from the 'Backward', 'Turn_left', and 'Turn_right' directories in the path to the ESP32 board and check if the car moves backward, turns left, and turns right as expected.



Car speed control

Connect the ESP32-CAM controller board to the computer and open the 'Speed_control.ino' program in 'KUONGSHUN AD174 ESP32 WiFi Camera 4WD Smart Robot Car Kit\3_Code_and_Tutorial\Make the Car Move\Speed_control'. Upload the program to the ESP32 board (repeat the upload steps for the AUTO_GO program).

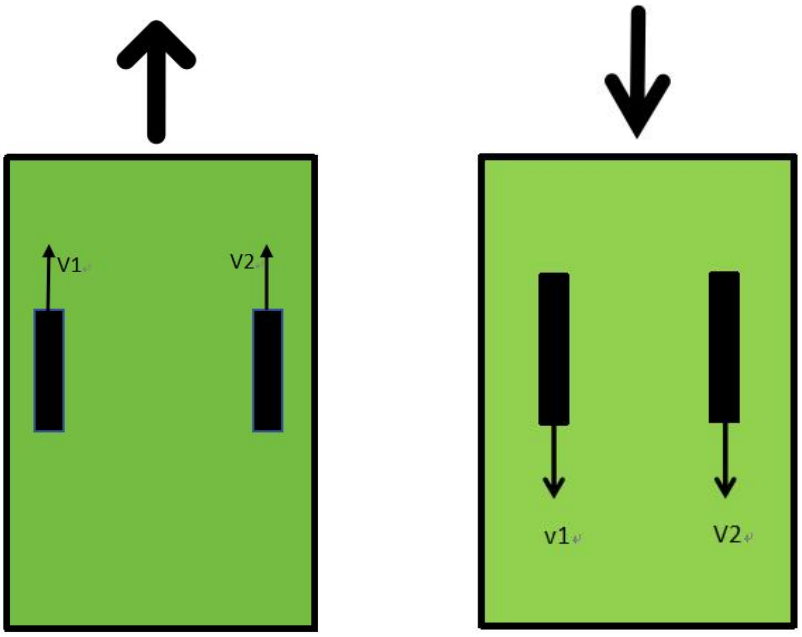
Speed_control.ino

```
#define ENA 2   // Left
#define ENB 12    // Right
#define gpRb 14  // left Backward
#define gpRf 13  // left Forward
#define gpLb 33  // Right Backward
#define gpLf 15  // right Forward
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);//open serial and set the baudrate
  pinMode(gpLb,OUTPUT);//before useing io pin, pin mode must be set first
  pinMode(gpLf,OUTPUT);
  pinMode(gpRb,OUTPUT);
  pinMode(gpRf,OUTPUT);
  pinMode(ENA,OUTPUT);
  pinMode(ENB,OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(gpLb,LOW); //set gpLb Low level
  digitalWrite(gpLf,HIGH);  //set gpLf Hight level
  digitalWrite(gpRb,LOW);  //set gpRb Low level
  digitalWrite(gpRf,HIGH); //set gpRf Hight level
  //accelerate
  for(int i = 0; i <= 255; i++){
  analogWrite(ENB,i);
  analogWrite(ENA,i);
  delay(20);
  }
  delay(1000);
  //Reduce speed
  for(int i = 255; i >= 0; i--){
  analogWrite(ENB,i);
  analogWrite(ENA,i);
  delay(20);
  }
  delay(1000);
}
```

Disconnect the connection with the computer, then power on the car. Starting from a standstill, the car will gradually accelerate until it reaches maximum speed and maintain that speed for 1 second. Then, it will begin to decelerate until it comes to a complete stop, and the process will repeat in a loop.

Principle of movement

Forward and backward movement: By synchronizing the speed of the two drive wheels, the car can move in a straight line either forward or backward. If both wheels rotate forward at the same speed, the robot will move straight forward; if both wheels rotate backward at the same speed, the robot will move backward.

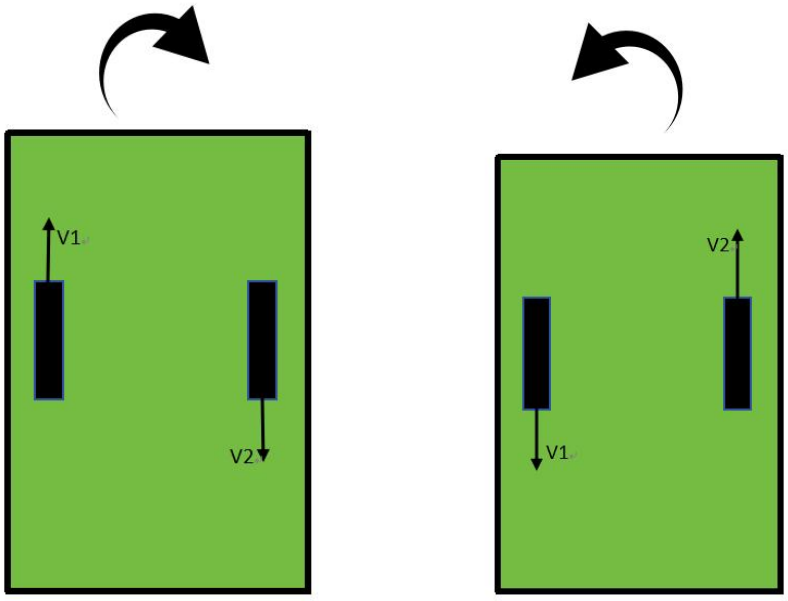


Forward **Backward**

Left and right rotation: By combining the two-wheel drive with the omni-wheel, the robot can rotate in place through unsynchronized driving (one wheel rotates forward while the other rotates in reverse).

When the left wheel moves forward and the right wheel moves backward, the car will rotate to the right in place.

When the left wheel moves backward and the right wheel moves forward, the car will rotate to the left in place.



Turn right **Turn left**

Program logic

1. Left and right wheel control

First, let's take a look at the motor connections on the L298N board. We will use the ESP32 pins 2, 12, 13, 14, 15, and 33 to control the car. Pins 13 and 14 control the right wheel, and pin 12 controls the speed of the right wheel. Pins 15 and 33 control the left wheel, and pin 2 controls the speed of the left wheel. Therefore, the connections are as follows:

L298N	ESP32 controller board
ENA	12
IN1	13
IN2	14
IN3	15
IN4	33
ENB	2

Left and right wheel control commands:

Right wheel:

ENA	gpRf(IN1)	gpRb（IN2）	Right Wheel
HIGH	HIGH	LOW	FORWARD
HIGH	LOW	HIGH	BACKWARD

Left wheel:

ENB	gpLf（IN3）	gpLb(IN4)	Left Wheel
HIGH	HIGH	LOW	FORWARD
HIGH	LOW	HIGH	BACKWARD

Right wheel control program explanation:

Define the pins 12, 13, and 14, which control the right wheel, with more understandable names: ENA (the pin controlling the right wheel's speed), gpRf (the pin controlling the forward direction of the right wheel motor), and gpRb (the pin controlling the reverse direction of the right wheel motor).

Note: Here, pin 13, which is set to HIGH to control the motor's forward rotation, is defined as gpRf, and pin 14, which is set to HIGH to control the motor's reverse rotation, is defined as gpRb.

```
#define ENB 12
#define gpRb 14   //Right Backward
#define gpRf 13  //Right Forward
```

Set the three pins controlling the right wheel to output mode (initialization).

```
pinMode(ENB,OUTPUT);
pinMode(gpRb,OUTPUT);
pinMode(gpRf,OUTPUT);
```

Right wheel forward

```
digitalWrite(ENB,HIGH);
digitalWrite(gpRb,LOW);
digitalWrite(gpRf,HIGH);
```

Right wheel backward

```
digitalWrite(ENB,HIGH);
digitalWrite(gpRb,HIGH);
digitalWrite(gpRf,LOW);
```

Note: To control the wheel's forward and backward movement, simply set the corresponding pin to HIGH and the other pin to LOW, creating a voltage difference that generates current, thereby controlling the motor's rotation.

Left wheel control program explanation:

Define the pins 2, 15, and 33, which control the left wheel, with more understandable names: ENB (the pin controlling the left wheel's speed), gpLf (the pin controlling the forward direction of the left wheel motor), and gpLb (the pin controlling the reverse direction of the left wheel motor).

Note: Here, pin 15, which is set to HIGH to control the motor's forward rotation, is defined as gpLf, and pin 33, which is set to HIGH to control the motor's reverse rotation, is defined as gpLb.

```
#define ENA 2
#define gpLb 33  // Left Backward
#define gpLf 15  // Left Forward
```

Set the three pins controlling the right wheel to output mode (initialization).

```
pinMode(ENA,OUTPUT);
pinMode(gpLb,OUTPUT);
pinMode(gpLf,OUTPUT);
```

Left wheel forward

```
digitalWrite(ENA,HIGH);
digitalWrite(gpLb,LOW);
digitalWrite(gpLf,HIGH);
```

Left wheel backward

```
digitalWrite(ENA,HIGH);
digitalWrite(gpLb,HIGH);
digitalWrite(gpLf,LOW);
```

Note: To control the wheel's forward and backward movement, simply set the corresponding pin to HIGH and the other pin to LOW, creating a voltage difference that generates current, thereby controlling the motor's rotation.

2. Car movement control

Car	Left wheel	Right wheel
Forward	Forward	Forward
Backward	Backward	Backward
Left	Backward	Forward
Right	Forward	Backward
Stop	Stop	Stop

Go forward

```
void forward(){
  digitalWrite(ENA,HIGH); //enable L298n A channel
  digitalWrite(ENB,HIGH); //enable L298n B channel
  digitalWrite(gpLb,LOW); //set gpLb low level
  digitalWrite(gpLf,HIGH);  //set gpLf hight level
  digitalWrite(gpRb,LOW);  //set gpRb low level
  digitalWrite(gpRf,HIGH); //set gpRf hight level
  Serial.println("Forward");//send message to serial monitor
}
```

Go backward

```
void back(){
  digitalWrite(ENA,HIGH);  //enable L298n A channel
  digitalWrite(ENB,HIGH);  //enable L298n B channel
  digitalWrite(gpLb,HIGH); //set gpLb hight level
  digitalWrite(gpLf,LOW);   //set gpLf low level
  digitalWrite(gpRb,HIGH);  //set gpRb hight level
  digitalWrite(gpRf,LOW);   //set gpRf low level
  Serial.println("Back");   //send message to serial monitor
}
```

Turn left

```
void left(){
  digitalWrite(ENA,HIGH);  //enable L298n A channel
  digitalWrite(ENB,HIGH);  //enable L298n B channel
  digitalWrite(gpLb,HIGH); //set gpLb hight level
  digitalWrite(gpLf,LOW); //set gpLf low level
  digitalWrite(gpRb,LOW); //set gpRb low level
  digitalWrite(gpRf,HIGH); //set gpRf hight level
  Serial.println("Left"); //send message to serial monitor
}
```

Turn right

```
void right(){
  digitalWrite(ENA,HIGH); //enable L298n A channel
  digitalWrite(ENB,HIGH); //enable L298n B channel
  digitalWrite(gpLb,LOW); //set gpLb low level
  digitalWrite(gpLf,HIGH); //set gpLf hight level
  digitalWrite(gpRb,HIGH); //set gpRb hight level
  digitalWrite(gpRf,LOW); //set gpRf low level
  Serial.println("Right"); //send message to serial monitor
}
```

3. Car speed control

In the previous program, we use digitalWrite(ENA, HIGH); and digitalWrite(ENB, HIGH); to control the left and right wheels of the car to rotate at full speed. Similarly, digitalWrite(ENA, LOW); and digitalWrite(ENB, LOW); are used to stop the left and right wheels of the car.

To control the speed of the car, we use analogWrite(ENA, speed); and analogWrite(ENB, speed); to adjust the speed of the left and right wheels, respectively. The speed value represents the motor speed, with a range from 0 to 255. The higher the value, the faster the speed. When the speed is 0, the car will stop, and when the speed is 255, the car will move at maximum speed.

Moving forward with acceleration

```
digitalWrite(gpLb,LOW); //set gpLb low level
digitalWrite(gpLf,HIGH);  //set gpLf hight level
digitalWrite(gpRb,LOW);  //set gpRb low level
digitalWrite(gpRf,HIGH); //set gpRf hight level
//accelerate
for(int i = 0; i <= 255; i++){
analogWrite(ENB,i);
analogWrite(ENA,i);
delay(20);
}
```

The car's speed increases from 0 to 255, achieving an acceleration effect.

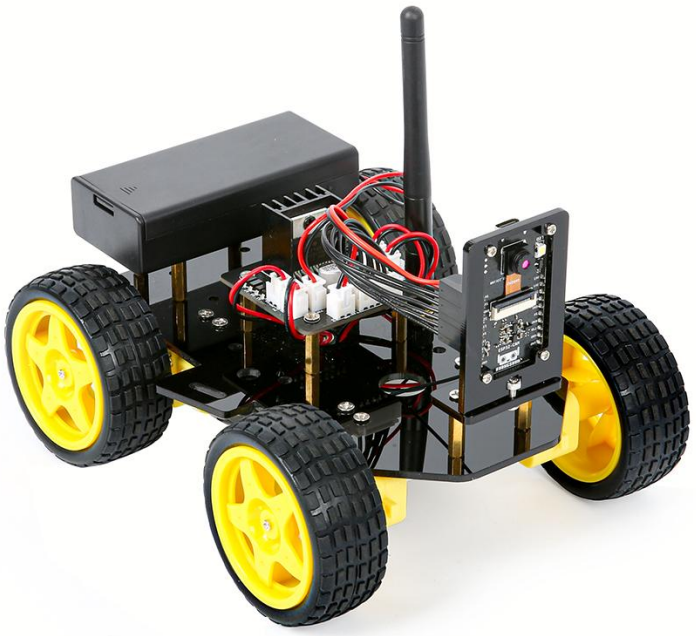
Moving forward at reduced speed

```
digitalWrite(gpLb,LOW); //set gpLb low level
digitalWrite(gpLf,HIGH);  //set gpLf hight level
digitalWrite(gpRb,LOW);  //set gpRb low level
digitalWrite(gpRf,HIGH); //set gpRf hight level
//reduce the speed
for(int i = 255; i >= 0; i--){
analogWrite(ENB,i);
analogWrite(ENA,i);
delay(20);
}
```

The car's speed decreases from 255 to 0, achieving a deceleration effect.

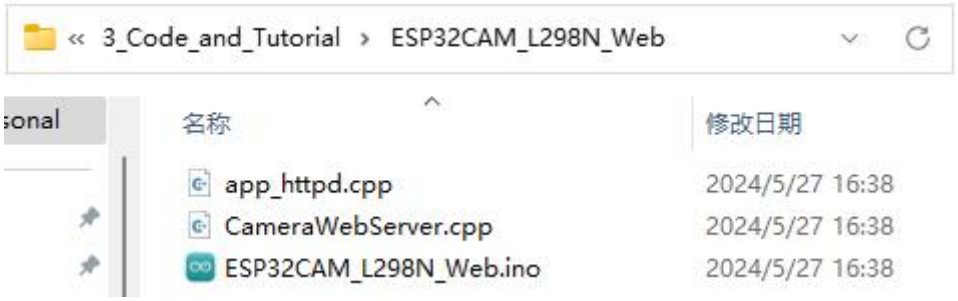
Lesson 2 WIFI Camera Car Code for Web

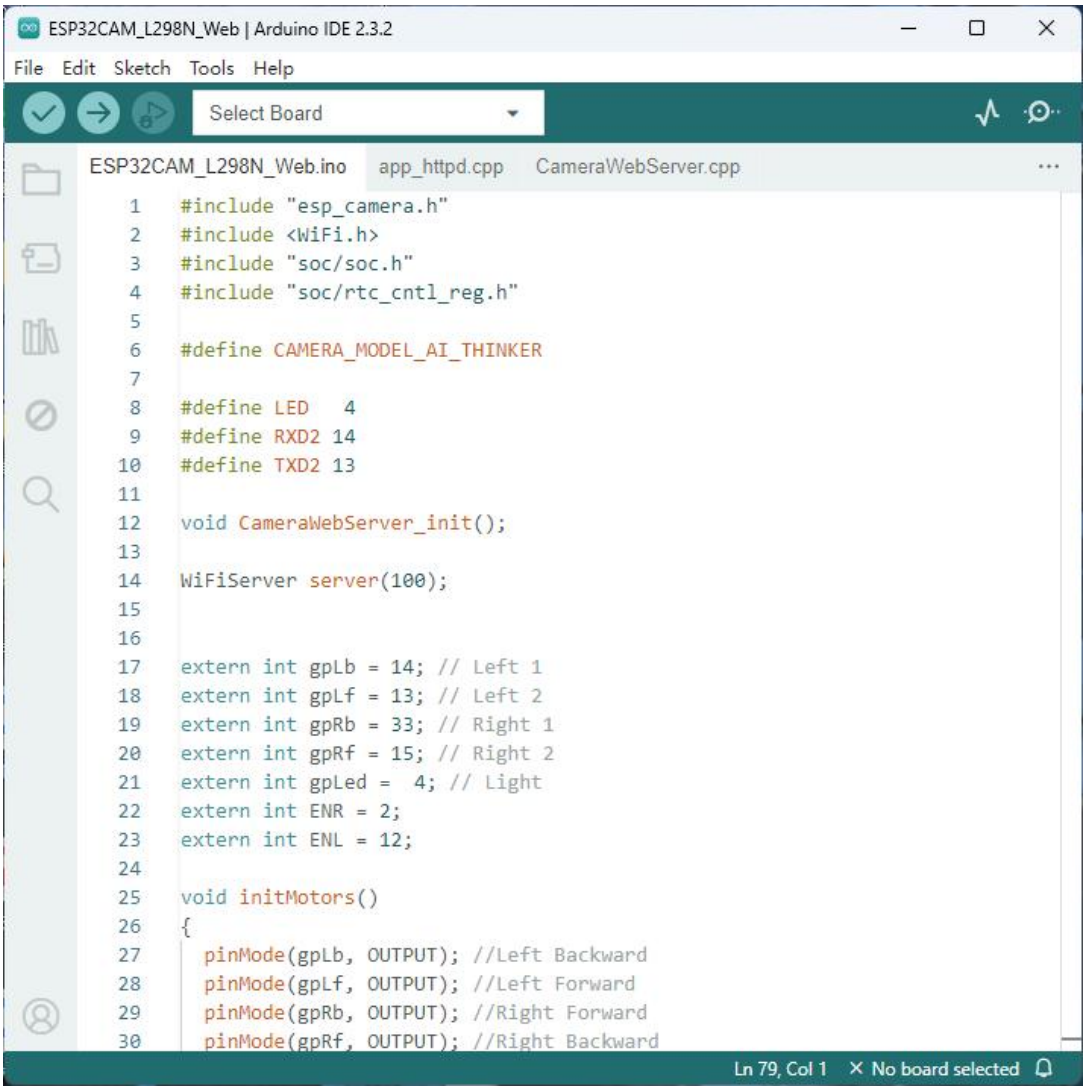
In this project, we will use ESP32-CAM module combined with L298N motor drive board to realize WiFi remote control and real-time monitoring function. The ESP32-CAM module runs a program to build a network server connected by a WiFi hotspot. As long as we connect to the WIFI hotspot, we can watch the monitoring in real time from the network server, and at the same time control the car by the button.



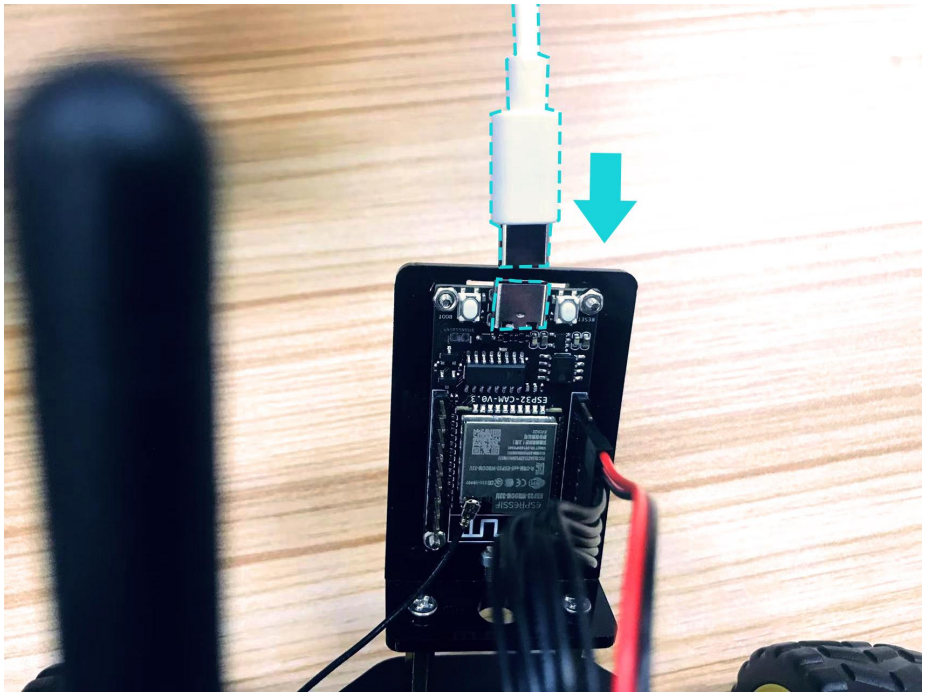
Next, how to upload this program to the car, which can build a server to control the car and watch the surveillance video：

①Open the 'ESP32CAM_L298N_Web.ino' program in 'KUONGSHUN AD175 ESP32 WiFi Camera 2WD Smart Robot Car Kit\3_Code_and_Tutorial\ESP32CAM_L298N_Web'.

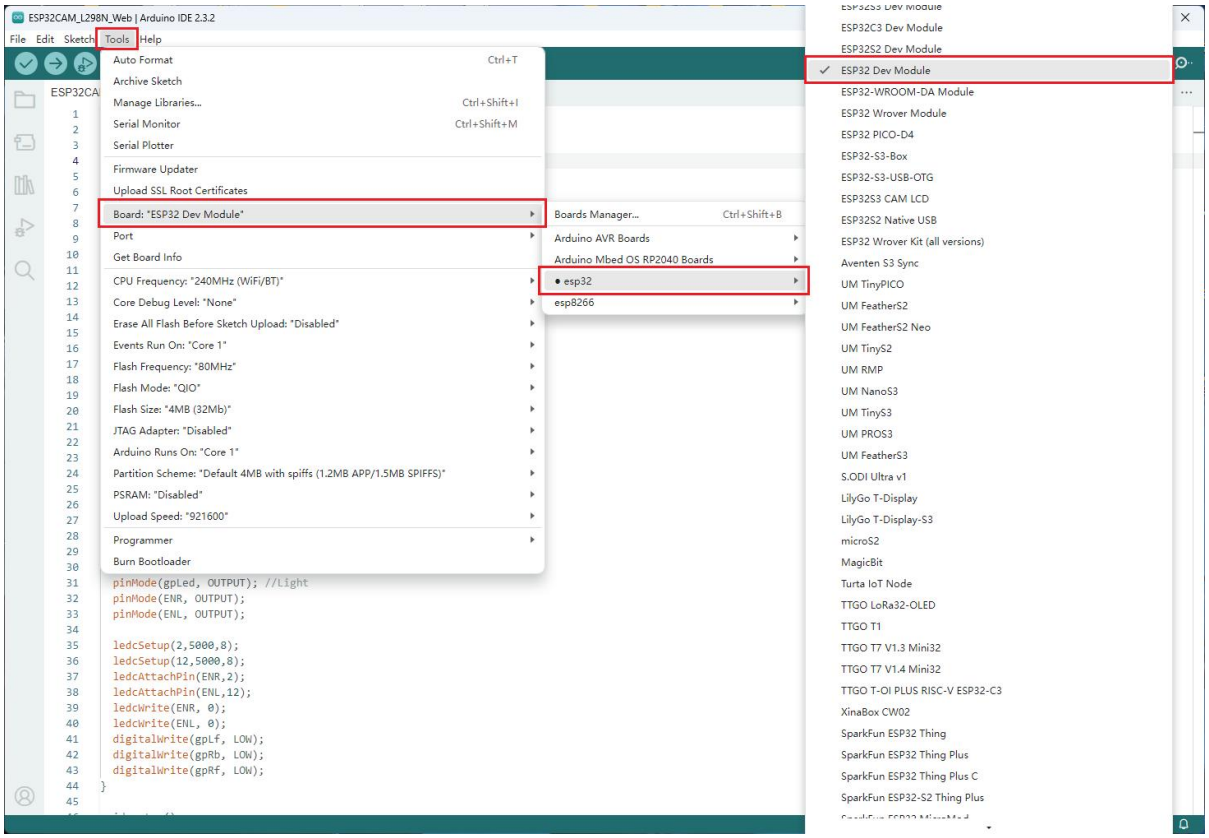




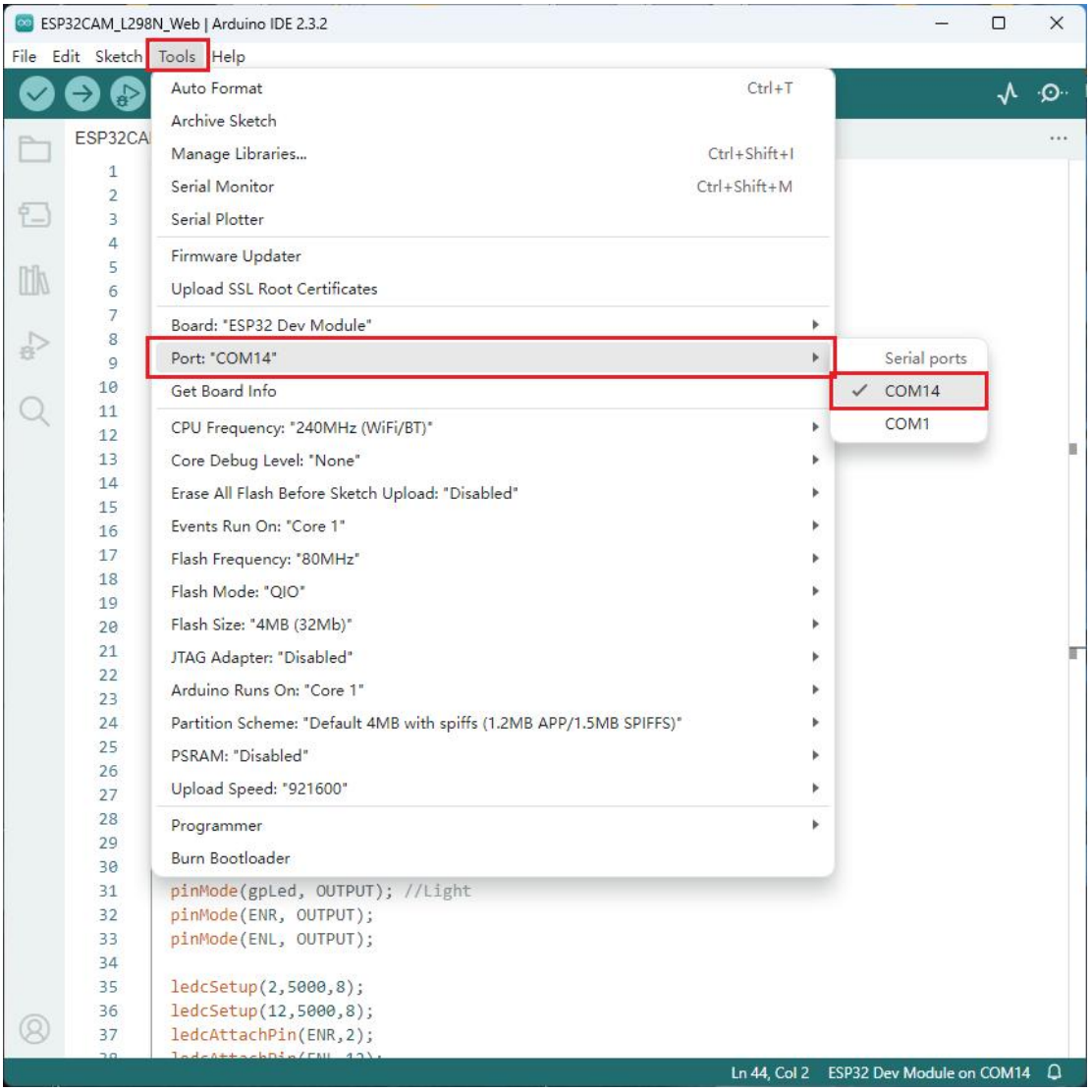
②Connect the ESP32 board to your computer with a TYPE-C cable



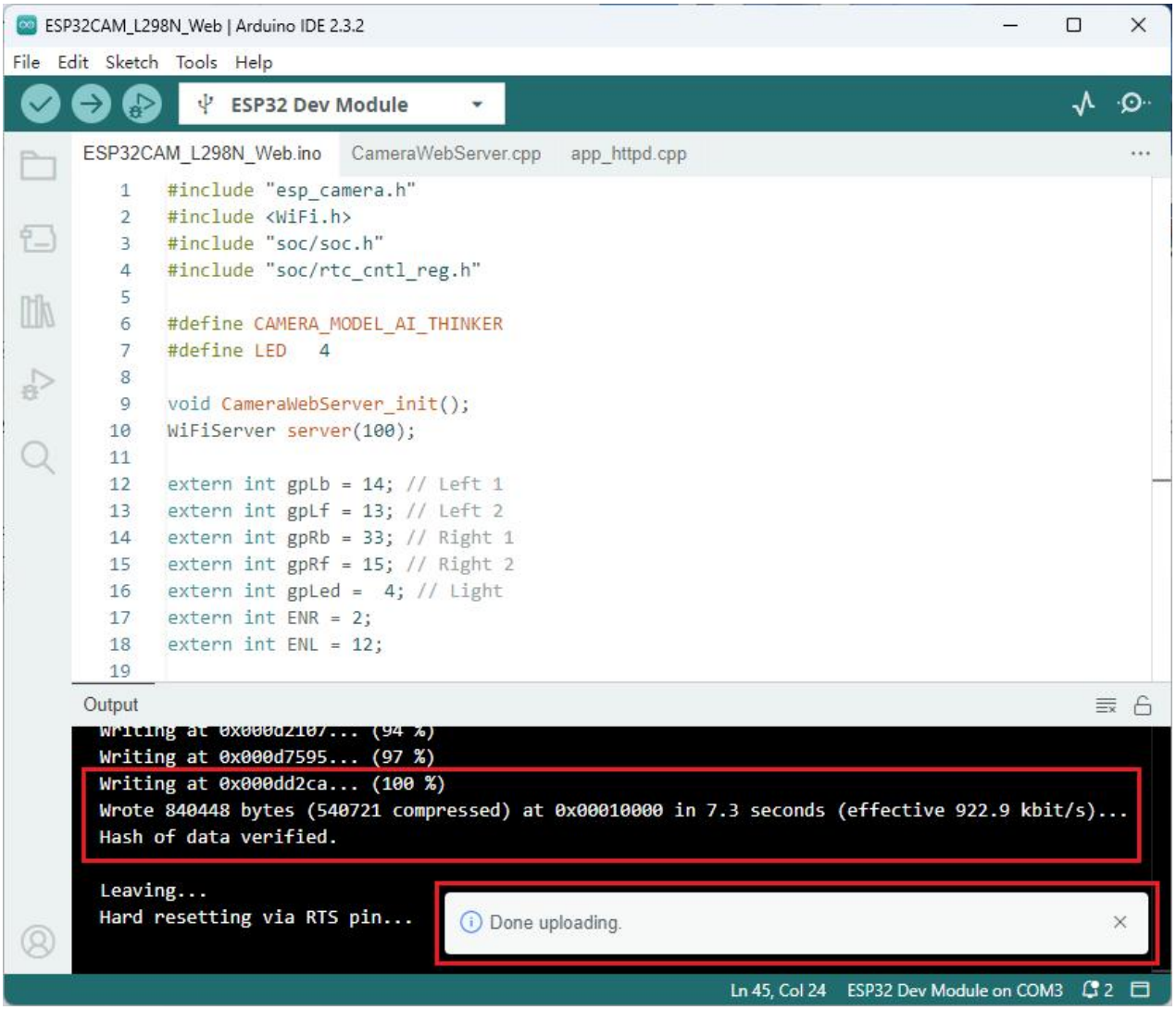
③ Select your Board in Tools > Board >>esp32>>>ESP32 Dev Module



④Select the Port： Tools >Port >>>>>COMxx(The COM port number of each computer may be different, and the port number that appears is your port number) (If you do not see COM ports other than COM1 in Arduino IDE, you need to refer to the installation of CH340 driver in "2_Programming_Preparation" and install it)

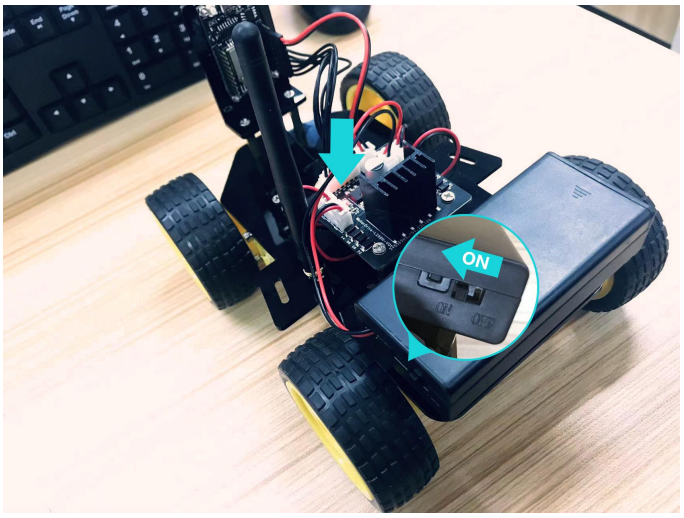


⑤Upload the program to the ESP32-CAM controller board.



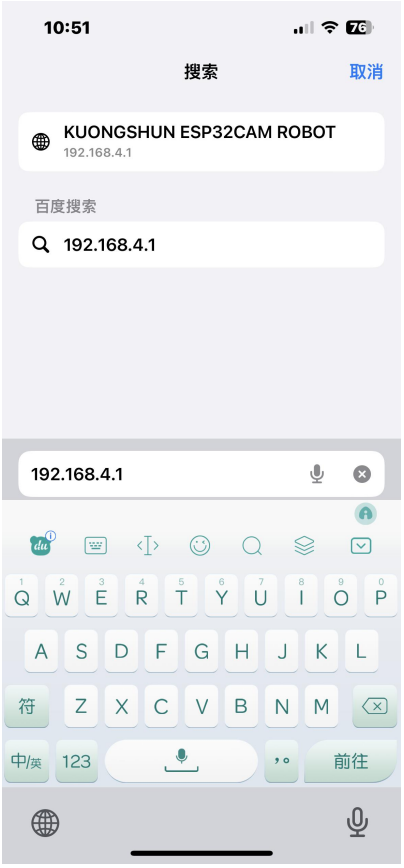
The picture above shows that it is uploaded successfully.

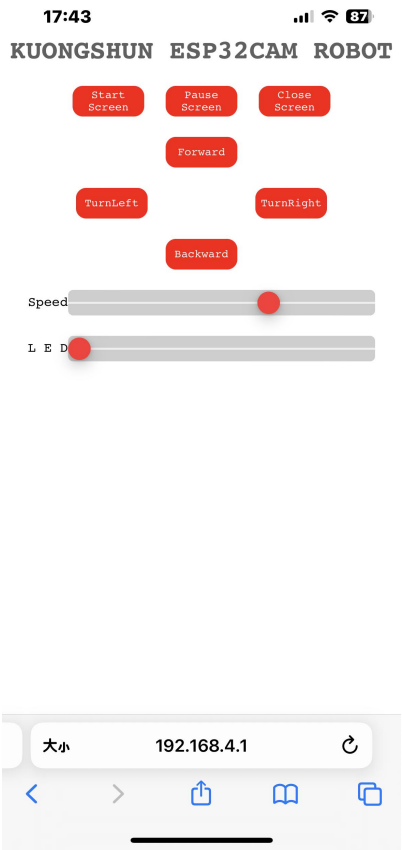
⑥After uploading the code successfully. Disconnect the data line, put the car on the ground, and load 2 18650 batteries. (If you buy the battery-free version, please prepare 2 18650 batteries.) Then turn on the switch.



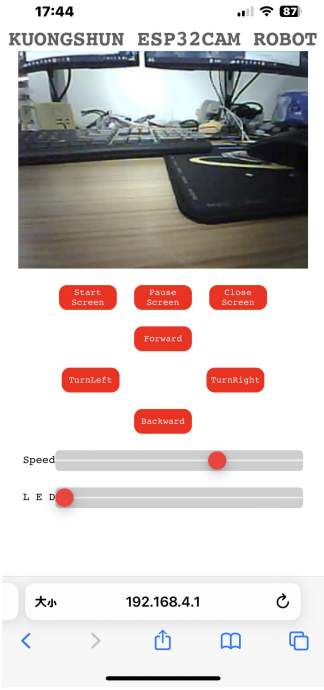
⑦ Then open the phone, turn on the WiFi switch, find the "KUONGSHUN-AD174", enter the password "12345678" to connect to the WIFI. If reminded there is no network, please keep connected.（If the battery switch is turned on and the module lights up, but the phone cannot search for "KUONGSHUN-AD174", please check whether the wiring is correct and whether the battery is powered.）

⑧ Open the mobile browser, enter “192.168.4.1”, jump to the screen below





⑨Introduction of the interface：



：Enable image receiving and display.When you press it：

：Screenshot and display

：Turn off image reception

：Control the cart forward while holding

：Control the cart back while holding

：Control the car to turn left when holding

：Control the car to turn right when holding

：Adjust the speed of the car

：Switch headway LED lights and adjust brightness

Now, when you start the car again, you don't need to download the program again.You just need to start from ⑦.

Program Explanation:

WiFi Name and Password Modification

In this project, you can modify the WiFi name and password by simply changing the values of *ssid and *password. Here, *ssid refers to the WiFi name, and *password refers to the password.

Note: Here, we are modifying the WiFi name and password broadcasted by the car.

ESP32CAM_L298N_Web.ino | CameraWebServer.cpp | app_httpd.cpp

```
#include "esp_system.h"
#include "esp_camera.h"
#include <WiFi.h>

const char *ssid = "KUONGSHUN-AD175";
const char *password = "12345678";
String wifi_name;

#define PWDN_GPIO_NUM     32
#define RESET_GPIO_NUM    -1
#define XCLK_GPIO_NUM      0
#define SIOD_GPIO_NUM     26
#define SIOC_GPIO_NUM     27
#define Y9_GPIO_NUM       35
#define Y8_GPIO_NUM       34
#define Y7_GPIO_NUM       39
#define Y6_GPIO_NUM       36
#define Y5_GPIO_NUM       21
#define Y4_GPIO_NUM       19
#define Y3_GPIO_NUM       18
#define Y2_GPIO_NUM        5
#define VSYNC_GPIO_NUM    25
#define HREF_GPIO_NUM     23
#define PCLK_GPIO_NUM     22

void startCameraServer();
void CameraWebServer_init()
{
  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
```

Change the Video Clarity

By selecting the byte length sample value, you can change the image clarity. There are three clarity options available, and you can only choose one. To do so, comment out the

other two options. Generally, the higher the image quality, the more likely there will be a delay in the image display.

```
//drop down frame size for higher initial frame rate
//s->set_framesize(s, FRAMESIZE_SXGA); //Video sample rate:60000          #9 (High image quality)  1280x1024
s->set_framesize(s, FRAMESIZE_SVGA); //Video sample rate:40000           #7 (Medium image quality)  800x600
//s->set_framesize(s, FRAMESIZE_QVGA); //Video sample rate:10000          #4 (Low image quality)  320x240
```

ESP32CAM_L298N_Web.ino | CameraWebServer.cpp | app_httpd.cpp

```
64      } else {
65        // Limit the frame size when PSRAM is not available
66        config.frame_size = FRAMESIZE_SVGA;
67        config.fb_location = CAMERA_FB_IN_DRAM;
68      }
69    } else {
70      // Best option for face detection/recognition
71      config.frame_size = FRAMESIZE_240X240;
72  #if CONFIG_IDF_TARGET_ESP32S3
73      config.fb_count = 2;
74  #endif
75    }
76    // camera init
77    esp_err_t err = esp_camera_init(&config);
78    if (err != ESP_OK)
79    {
80      Serial.printf("Camera init failed with error 0x%x", err);
81      return;
82    }
83    sensor_t *s = esp_camera_sensor_get();
84    //drop down frame size for higher initial frame rate
85    //s->set_framesize(s, FRAMESIZE_SXGA); //Video sample rate:60000          #9 (High image quality)  1280x1024
86    //s->set_framesize(s, FRAMESIZE_SVGA); //Video sample rate:40000           #7 (Medium image quality)  800x600
87    s->set_framesize(s, FRAMESIZE_QVGA); //Video sample rate:10000          #4 (Low image quality)  320x240
88
89    s->set_vflip(s, 1);   //Image Direction Settings(Up/Down)
90    s->set_hmirror(s, 1); //Image Direction Setting(Left/Right)
91
92    WiFi.setTxPower(WIFI_POWER_19_5dBm);
93    WiFi.mode(WIFI_AP);
94    WiFi.softAP(ssid, password, 5);
95    startCameraServer();
96    // WiFi.softAP(ssid, password);
97    IPAddress miIP = WiFi.softAPIP();
98    Serial.print("AP IP address: ");
99    Serial.println(miIP); //probar 192.168.4.1
100
```

Change the image orientation

```
91    s->set_vflip(s, 1);   //Image Direction Settings(Up/Down)
92    s->set_hmirror(s, 0); //Image Direction Setting(Left/Right)
93
```

If the image displayed on the web interface is flipped vertically, simply change the original s->set_vflip(s, 1); to s->set_vflip(s, 0);. After completing this step, the image will be displayed correctly.

Similarly, if the image displayed on the web interface is flipped horizontally, simply change the original s->set_hmirror(s, 0); to s->set_hmirror(s, 1);. After completing this step, the image will be displayed correctly.

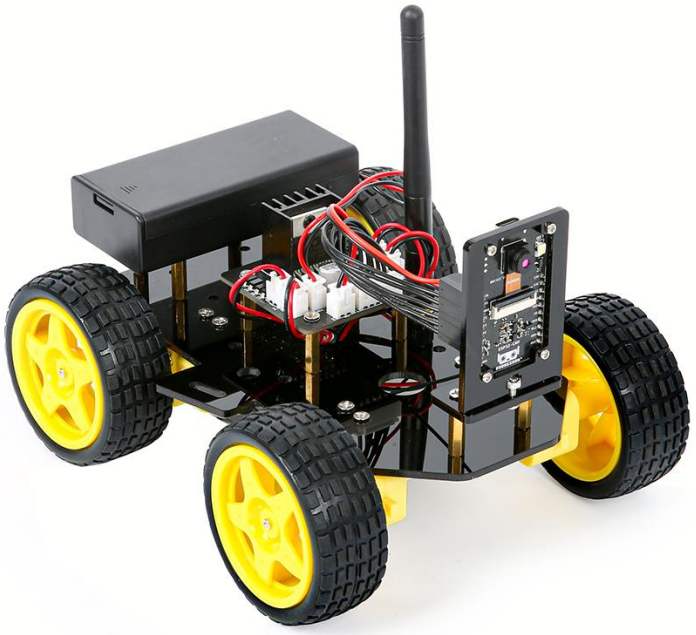
Note: The same operation applies in the APP-controlled program.

ESP32CAM_L298N_Web.ino | CameraWebServer.cpp | app_httpd.cpp

```
    }
  } else {
    // Best option for face detection/recognition
    config.frame_size = FRAMESIZE_240X240;
#if CONFIG_IDF_TARGET_ESP32S3
    config.fb_count = 2;
#endif
  }
  // camera init
  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK)
  {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
  }
  sensor_t *s = esp_camera_sensor_get();
  //drop down frame size for higher initial frame rate
  //s->set_framesize(s, FRAMESIZE_SXGA); //Video sample rate:60000          #9 (High image quality)  1280x1024
  //s->set_framesize(s, FRAMESIZE_SVGA); //Video sample rate:40000          #7 (Medium image quality)  800x600
  s->set_framesize(s, FRAMESIZE_QVGA); //Video sample rate:10000          #4 (Low image quality)  320x240

  s->set_vflip(s, 1);   //Image Direction Settings(Up/Down)
  s->set_hmirror(s, 1); //Image Direction Setting(Left/Right)

  WiFi.setTxPower(WIFI_POWER_19_5dBm);
  WiFi.mode(WIFI_AP);
  WiFi.softAP(ssid, password, 5);
  startCameraServer();
  // WiFi.softAP(ssid, password);
```

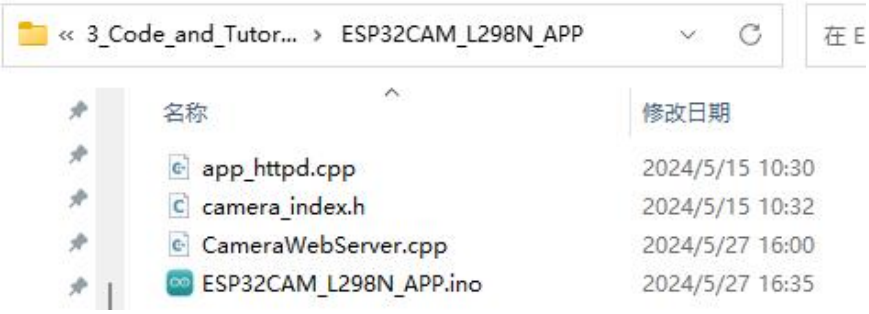
Lesson 3 WIFI Camera Car Code for APP

In this project, we will use ESP32-CAM module combined with L298N motor drive board to realize WiFi remote control and real-time monitoring function. The ESP32-CAM module runs a program to build a network server connected by a WiFi hotspot. As long as we connect to this WIFI hotspot, we can use the APP to watch the surveillance in real time, and at the same time, we can control the trolley to move through the buttons of the APP.



The next step is to explore how to upload this program that can also build a server and control the cart and watch videos through an app:

①Open the 'ESP32CAM_L298N_APP.ino' program in 'KUONGSHUN AD175 ESP32 WiFi Camera 2WD Smart Robot Car Kit\3_Code_and_Tutorial\ESP32CAM_L298N_APP'.



ESP32CAM_L298N_APP | Arduino IDE 2.3.2

File Edit Sketch Tools Help

ESP32 Dev Module

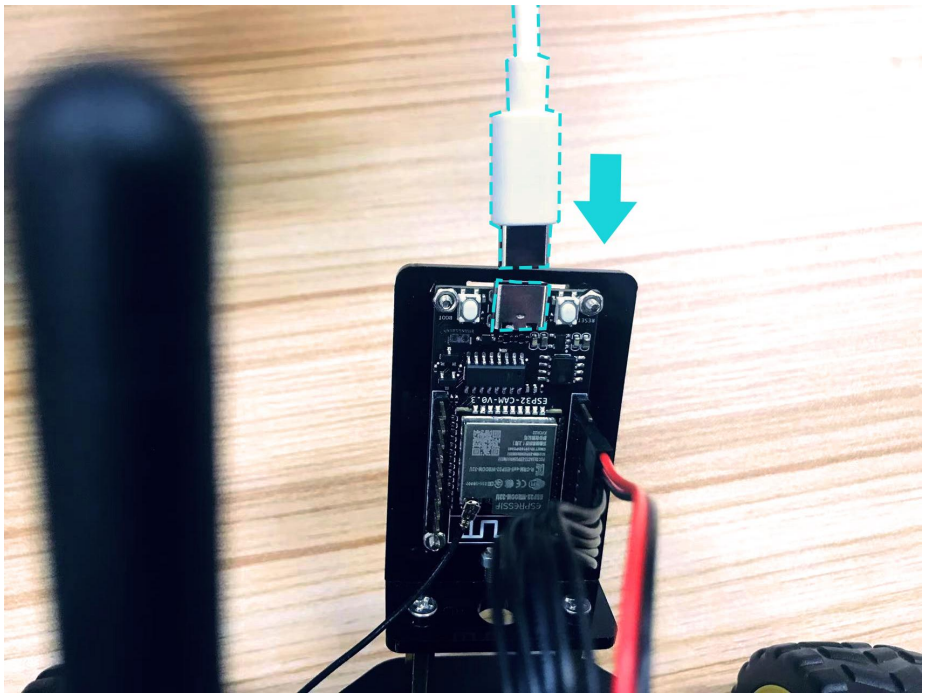
ESP32CAM_L298N_APP.ino CameraWebServer.cpp app_httpd.cpp camera_index.h

```
#include <WiFi.h>
#include "esp_camera.h"

WiFiServer server(100);

byte Stop[17]={0xff,0x55,0x0a,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x0C,0x00,0x00};
char buffer[52];
byte RX_package[17] = {0};
byte dataLen, index_a = 0;
unsigned char prevc=0;
int dataIndex = 0;
int LED_mode=0,LED_val=0;
int speed=225;
String sendBuff;
String Version = "Firmware Version is 0.12.21";
bool ED_client = true;
bool WA_en = false;
bool isStart = false;

#define gpLED  4      // Light

void CameraWebServer_init();
extern int gpLb = 14; // Left 1
extern int gpLf = 13; // Left 2
extern int gpRb = 33; // Right 1
extern int gpRf = 15; // Right 2
```

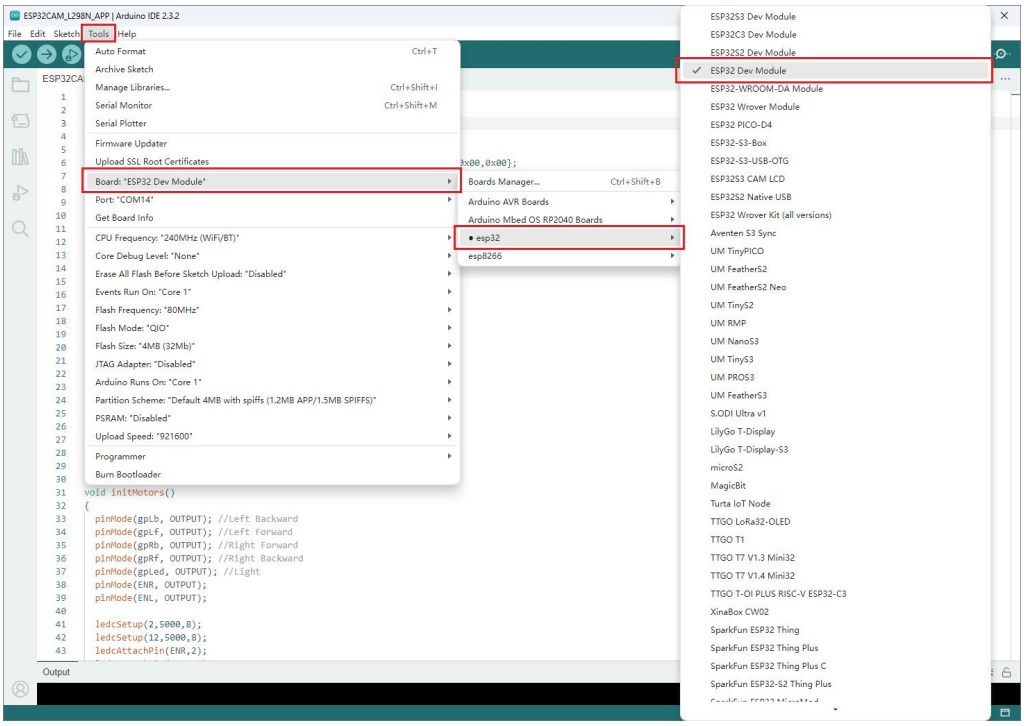
Output

Ln 51, Col 1 ESP32 Dev Module on COM14 [not connected]

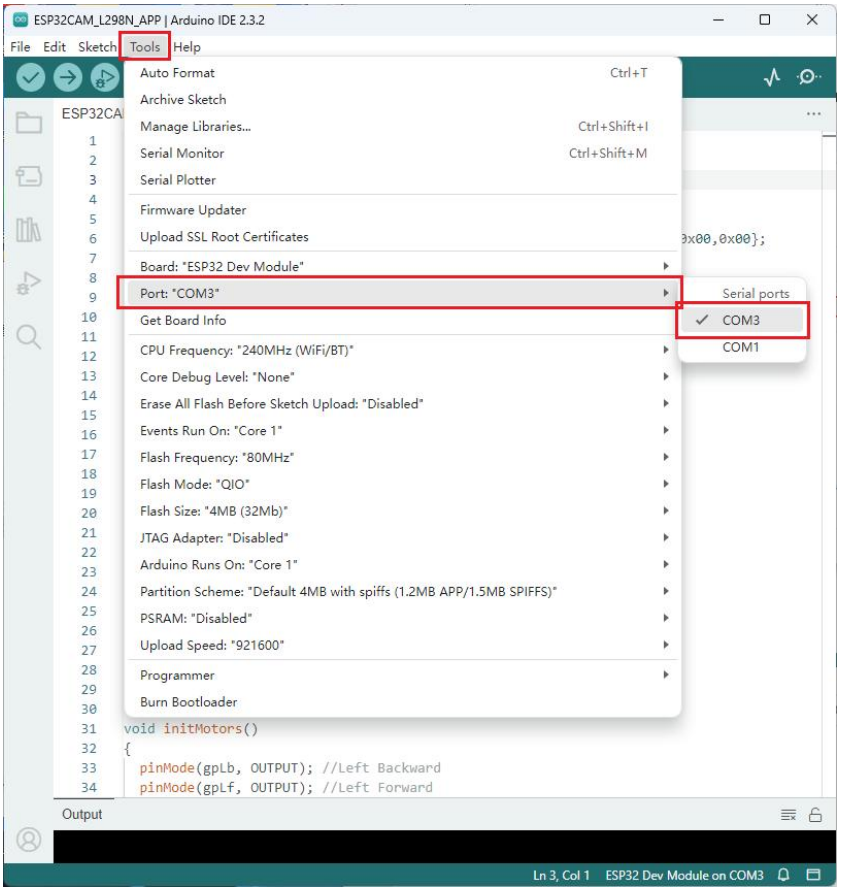
②Connect the ESP32 board to your computer with a TYPE-C cable



③ Select your Board in Tools > Board >>esp32>>>ESP32 Dev Module

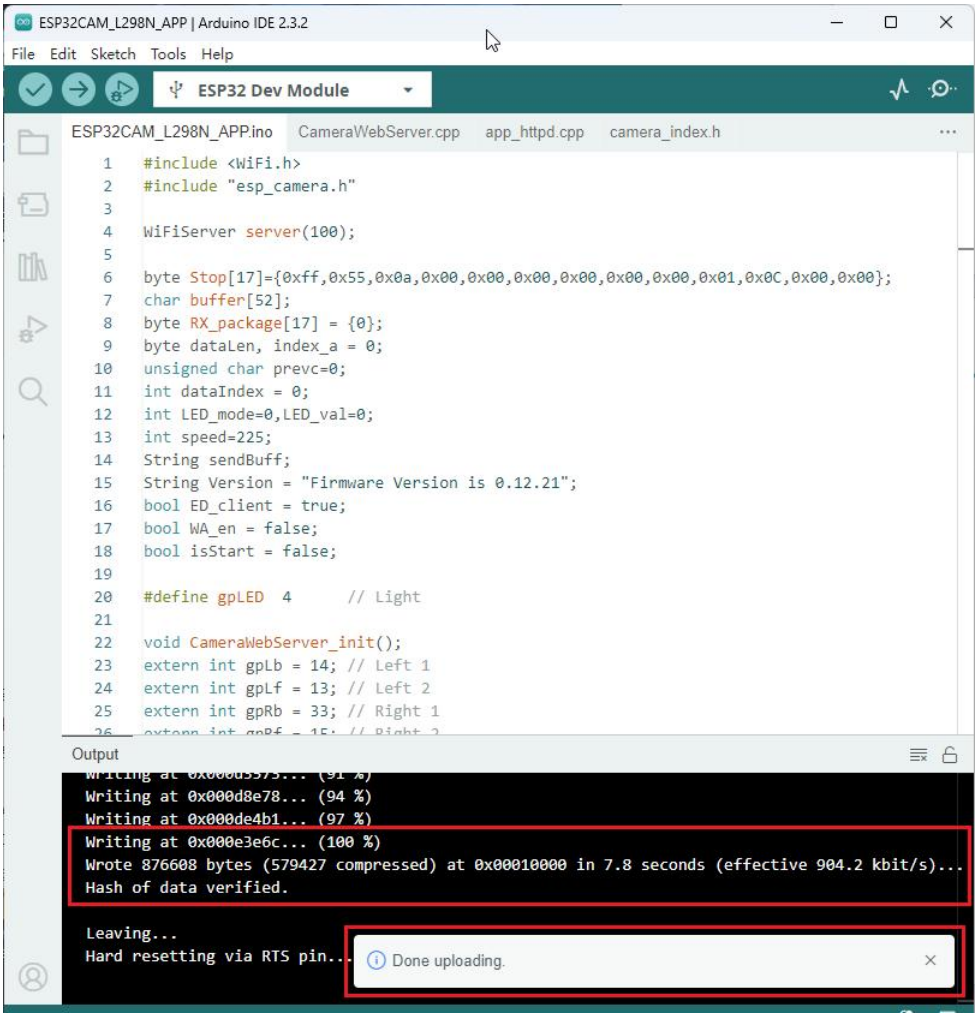


④Select the Port： Tools >Port >>>>>COMxx(The COM port number of each computer may be different, and the port number that appears is your port number)



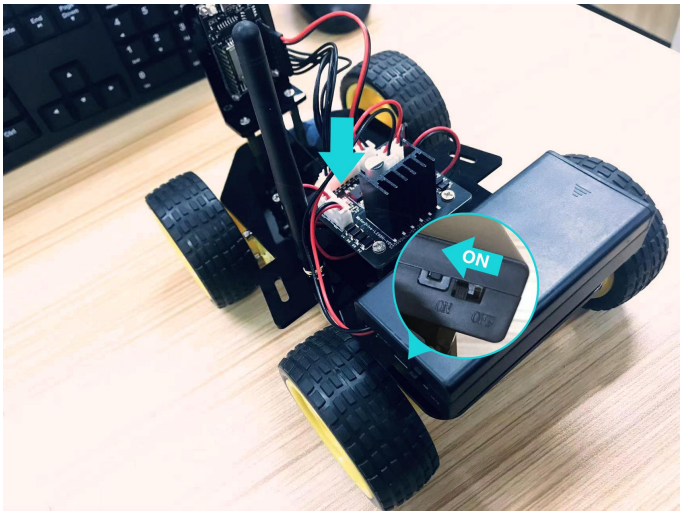
If you do not see COM ports other than COM1 in Arduino IDE, you need to refer to the installation of CH340 driver in "2_Programming_Preparation" and install it.

⑤Upload the program to the ESP32-CAM controller board.



The picture above shows that it is uploaded successfully.

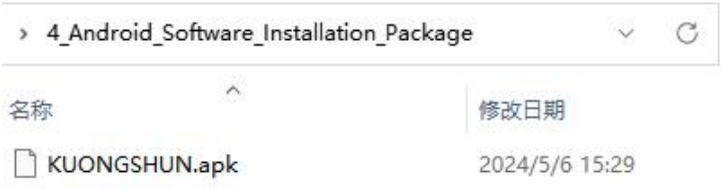
⑥After uploading the code successfully. Disconnect the data line, put the car on the ground, and load 2 18650 batteries. (If you buy the battery-free version, please prepare 2 18650 batteries.) Then turn on the switch.



⑦ To install the application

For Android:

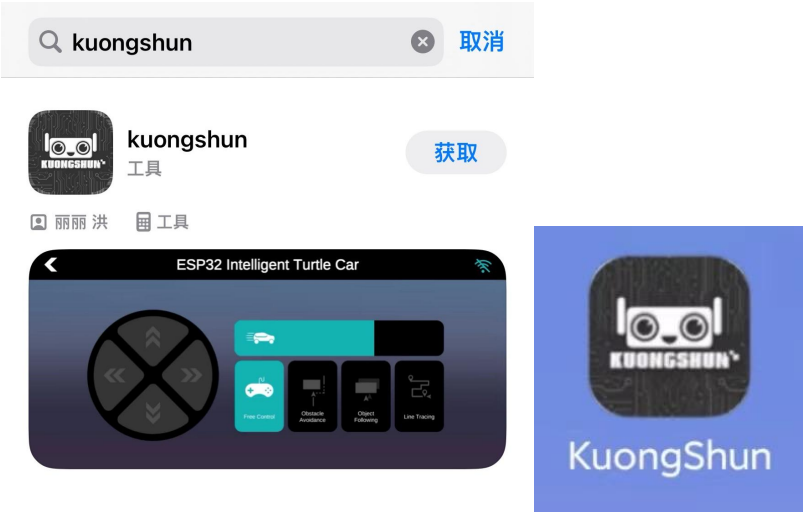
Open the profile folder, copy the apk file in the 4_Android_Software_Installation_Package folder to your Android phone or tablet computer , and then install it later. (You can send it to your phone or tablet via communication software and open and install it directly, or save it to your phone's memory and find and open it in file management)



Once the installation is complete, the following icon will appear on your phone or tablet:

For the Apple system:

Enter kuongshun in the APP Store search bar, click to get the download and install, install as shown in the right picture.

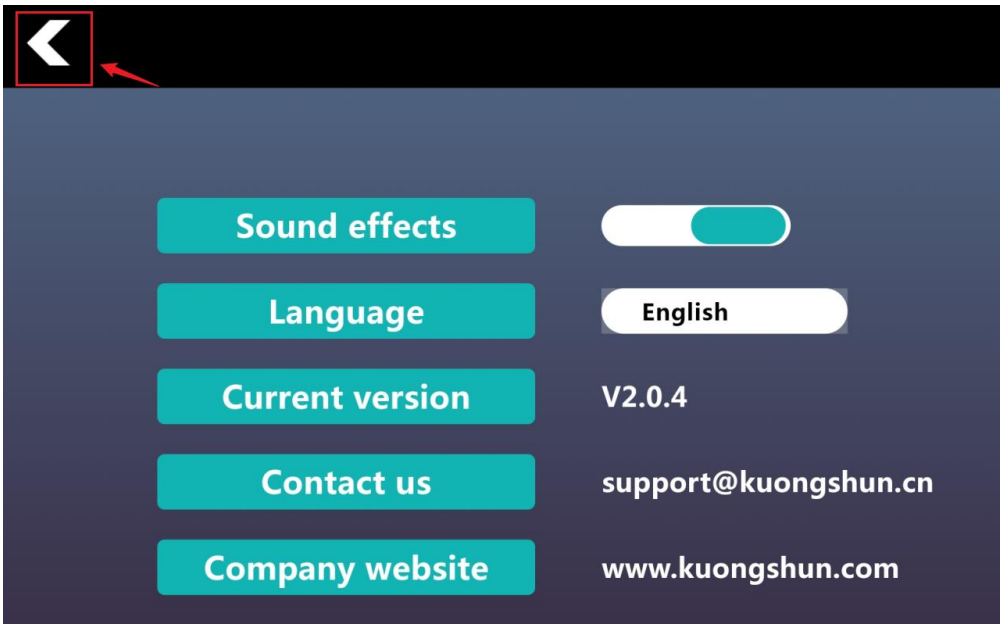


⑧ Then open the phone, turn on the WiFi switch, find the "KUONGSHUN-AD174", enter the password "12345678" to connect to the WIFI. If reminded there is no network, please keep connected.（If the battery switch is turned on and the module lights up, but the phone cannot search for "KUONGSHUN-AD174", please check whether the wiring is correct and whether the battery is powered.）

⑨Open the APP

Set language

Back to home



Select ESP32-CAM Two-wheel Drive.

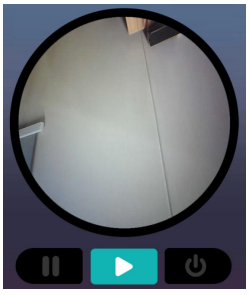
Click the icon in the top-right corner to connect to the ESP32-CAM Two-wheel Drive.

When the "Connected Successfully" message appears, it indicates that the connection was successful. Once connected, the live camera feed will be displayed, and you can start controlling the car as desired.

Note: If the "Connected Fail" message appears, please check if the Wi-Fi connection has been disconnected. After reconnecting to the Wi-Fi, try clicking the icon again to reconnect.

⑩Introduction of the interface：

：The car arrow keys, contains forward and backward left and right. Press and hold will light up and the car will go. Release will go out and the car will stop.

 ：The pause button freezes the screen, the play button opens the video screen, and the close button closes the video screen.

：Slide to adjust the brightness of the light.

：Slide to adjust the speed of the cart.

：Display the connection status, if you are connected to WiFi but connected to the cart you can tap here to connect.

Now, when you start the car again, you don't need to download the program again.You just need to start from ⑧.

Frequently Asked Questions

Q: WIFI signal cannot be searched

A: The power supply may be insufficient. Please check if the battery connection is normal? Does the motor driver board light up normally? Is the wiring correct?

Q: The screen image is inverted

A: If your camera module is "ESP32-CAM-V0.3", in order to display the image correctly, you need to modify the instruction in the **CameraWebServer.cpp** file. Change the command that sets the image flip direction from the original:

s->set_vflip(s, 1); to : s->set_vflip(s, 0);

This change should be applied in both the Web version and the App version.

ESP32-CAM-V0.3

ESP32CAM_L298N_APP.ino | CameraWebServer.cpp | app_httpd.cpp | camera_index.h

```
    }
  } else {
    // Best option for face detection/recognition
    config.frame_size = FRAMESIZE_240X240;
#if CONFIG_IDF_TARGET_ESP32S3
    config.fb_count = 2;
#endif
  }
  // camera init
  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK)
  {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
  }
  sensor_t *s = esp_camera_sensor_get();
  //drop down frame size for higher initial frame rate
  //s->set_framesize(s, FRAMESIZE_SXGA); //Video sample rate:60000          #9 (High image quality)  1280x1024
  s->set_framesize(s, FRAMESIZE_SVGA); //Video sample rate:40000          #7 (Medium image quality)  800x600
  //s->set_framesize(s, FRAMESIZE_QVGA); //Video sample rate:10000         #4 (Low image quality)  320x240

  s->set_vflip(s, 0);  //Image Direction Settings(Up/Down)
  s->set_hmirror(s, 0); //Image Direction Setting(Left/Right)

  // WiFi.setTxPower(WIFI_POWER_19_5dBm);
  WiFi.mode(WIFI_AP);
  WiFi.softAP(ssid, password, 5);
  // WiFi.softAP(ssid, password);
  Serial.print("\r\n");
  startCameraServer();
  Serial.print("\r\n");
  Serial.print("Camera Ready! Use 'http://");
  Serial.print(WiFi.softAPIP());
  Serial.println("' to connect");
```

ESP32-CAM-V2.0

ESP32CAM_L298N_APP.ino | CameraWebServer.cpp | app_httpd.cpp | camera_index.h

```
    }
  } else {
    // Best option for face detection/recognition
    config.frame_size = FRAMESIZE_240X240;
#if CONFIG_IDF_TARGET_ESP32S3
    config.fb_count = 2;
#endif
  }
  // camera init
  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK)
  {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
  }
  sensor_t *s = esp_camera_sensor_get();
  //drop down frame size for higher initial frame rate
  //s->set_framesize(s, FRAMESIZE_SXGA); //Video sample rate:60000          #9 (High image quality)  1280x1024
  s->set_framesize(s, FRAMESIZE_SVGA); //Video sample rate:40000          #7 (Medium image quality)  800x600
  //s->set_framesize(s, FRAMESIZE_QVGA); //Video sample rate:10000         #4 (Low image quality)  320x240

  s->set_vflip(s, 1);  //Image Direction Settings(Up/Down)
  s->set_hmirror(s, 0); //Image Direction Setting(Left/Right)

  // WiFi.setTxPower(WIFI_POWER_19_5dBm);
  WiFi.mode(WIFI_AP);
  WiFi.softAP(ssid, password, 5);
  // WiFi.softAP(ssid, password);
  Serial.print("\r\n");
  startCameraServer();
  Serial.print("\r\n");
  Serial.print("Camera Ready! Use 'http://");
  Serial.print(WiFi.softAPIP());
  Serial.println("' to connect");
}
```

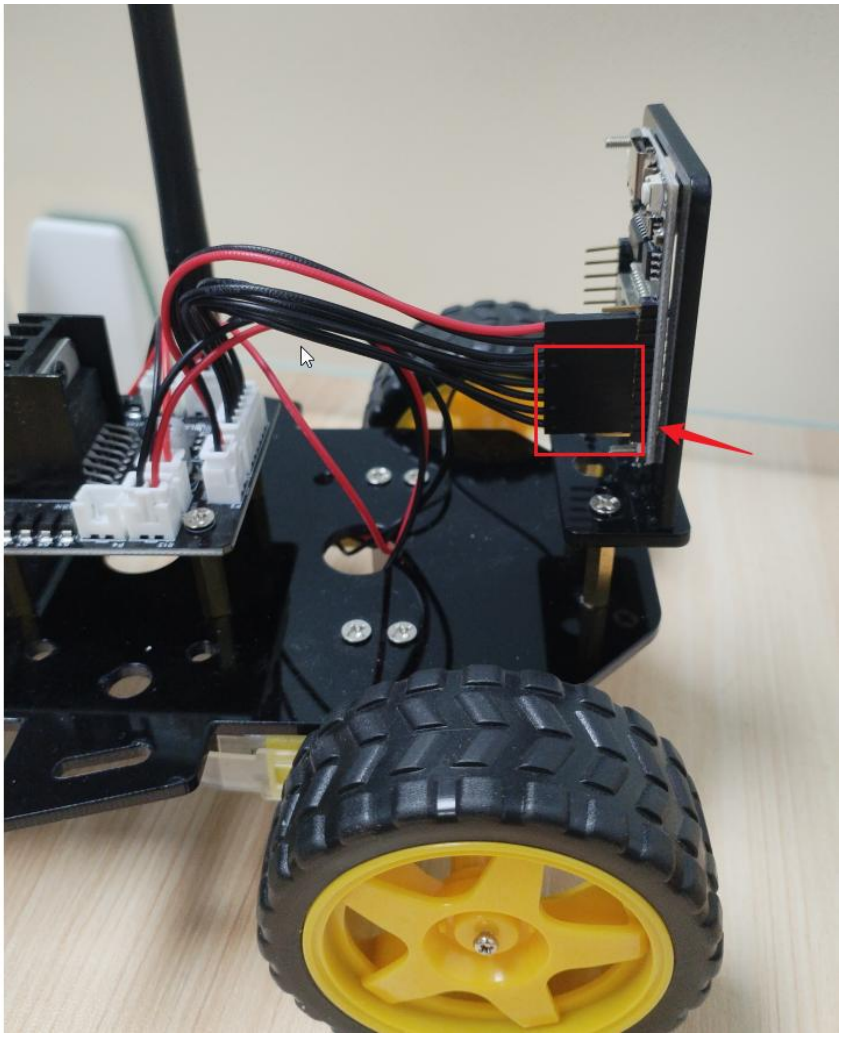
Q: How to adjust the image clarity

A: You can change the image clarity by selecting the byte length sample value. There are three levels of clarity to choose from, and you can select only one. To apply the selected clarity, simply comment out the other two lines. Generally, the higher the image quality, the more likely there will be a delay in the image display.

```
//drop down frame size for higher initial frame rate
//s->set_framesize(s, FRAMESIZE_SXGA); //Video sample rate:60000          #9 (High image quality)  1280x1024
s->set_framesize(s, FRAMESIZE_SVGA); //Video sample rate:40000           #7 (Medium image quality)  800x600
//s->set_framesize(s, FRAMESIZE_QVGA); //Video sample rate:10000          #4 (Low image quality)  320x240
```

ESP32CAM_L298N_APP.ino | CameraWebServer.cpp | app_httpd.cpp | camera_index.h

```
    }
  } else {
    // Best option for face detection/recognition
    config.frame_size = FRAMESIZE_240X240;
#if CONFIG_IDF_TARGET_ESP32S3
    config.fb_count = 2;
#endif
  }
  // camera init
  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK)
  {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
  }
  sensor_t *s = esp_camera_sensor_get();
  //drop down frame size for higher initial frame rate
  //s->set_framesize(s, FRAMESIZE_SXGA); //Video sample rate:60000          #9 (High image quality)  1280x1024
  s->set_framesize(s, FRAMESIZE_SVGA); //Video sample rate:40000           #7 (Medium image quality)  800x600
  //s->set_framesize(s, FRAMESIZE_QVGA); //Video sample rate:10000          #4 (Low image quality)  320x240

  s->set_vflip(s, 1);   //Image Direction Settings(Up/Down)
  s->set_hmirror(s, 0); //Image Direction Setting(Left/Right)

  // WiFi.setTxPower(WIFI_POWER_19_5dBm);
  WiFi.mode(WIFI_AP);
  WiFi.softAP(ssid, password, 5);
  // WiFi.softAP(ssid, password);
  Serial.print("\r\n");
  startCameraServer();
  Serial.print("\r\n");
  Serial.print("Camera Ready! Use 'http://");
  Serial.print(WiFi.softAPIP());
  Serial.println("' to connect");
}
```

Q: When controlling the car to move forward, the car moves backward, and when controlling the car to move backward, it moves forward.

A: It is possible that the connection between the car's motor driver board and the ESP32 control board is reversed.