

AD176

**ESP32 WiFi Camera Mecanum 4WD
Smart Robot Car Kit**

V1.0.24.1.18

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 children and teens ages 8 and up to learn about Arduino-compatible boards, actuators, sensors and related components, and programming.

This course is designed to introduce learners to how to program the cart and how to control the programmed cart. The cart comes with WIFI control, video mapping and mode switching. As a common STEM educational tool, it not only develops learners' hands-on skills, but also helps 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 info@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

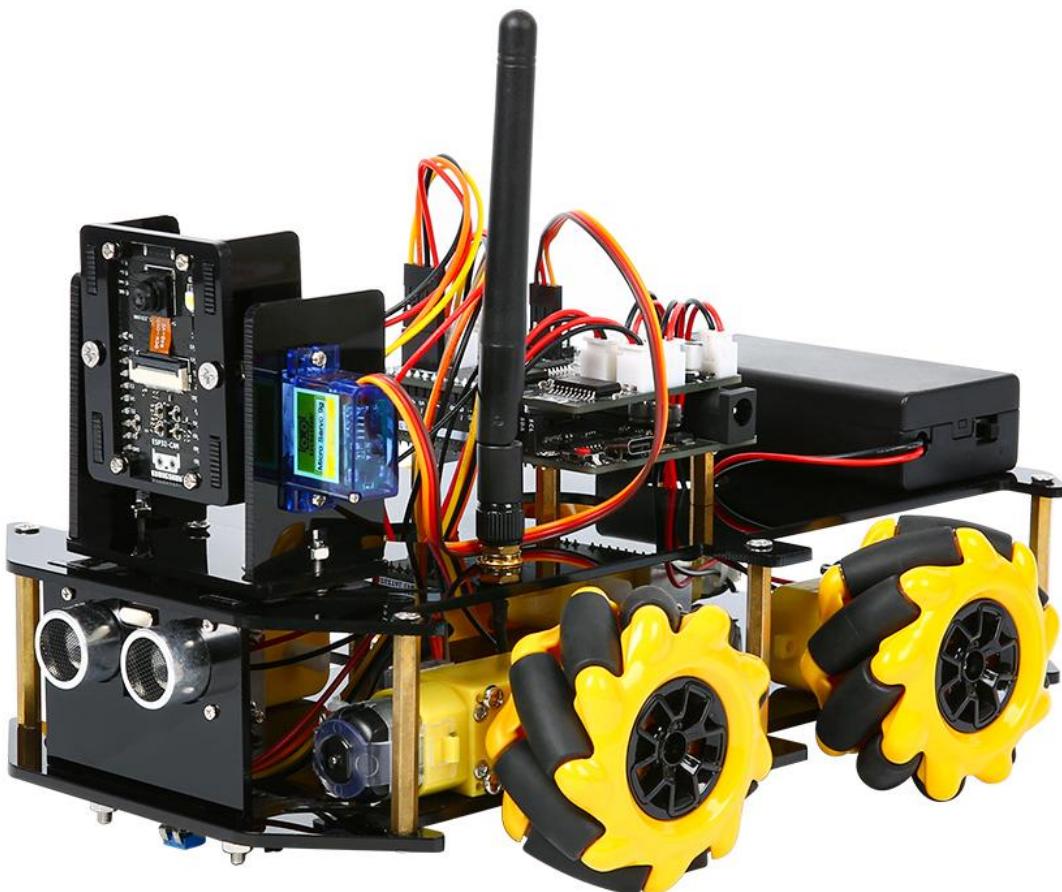


Content

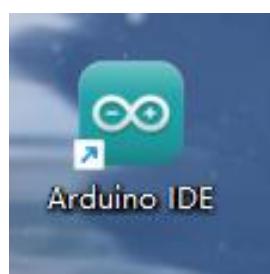
Lesson 0 Preparation	5
Lesson 1 Get The Car Moving	8
Lesson 2 Servo Pan-Tilt	18
Lesson 3 Ultrasonic Sensor Module	26
Lesson 4 Three Infrared Line Tracing	34
Lesson 5 WIFI Camera Multi-Function Program for Web	42
Lesson 6 WIFI Camera Multi-Function Program for APP	54
Frequently Asked Questions	67

Lesson 0 Preparation

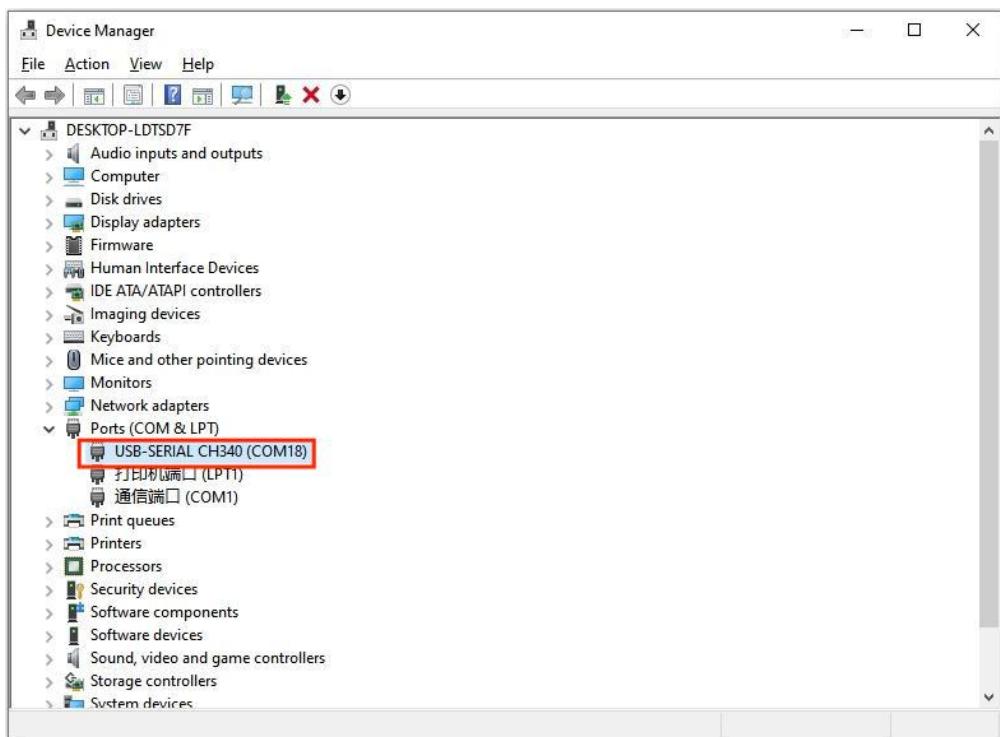
Before proceeding with this tutorial, make sure you have finished putting the model together in “1_Assembly_Guide”.



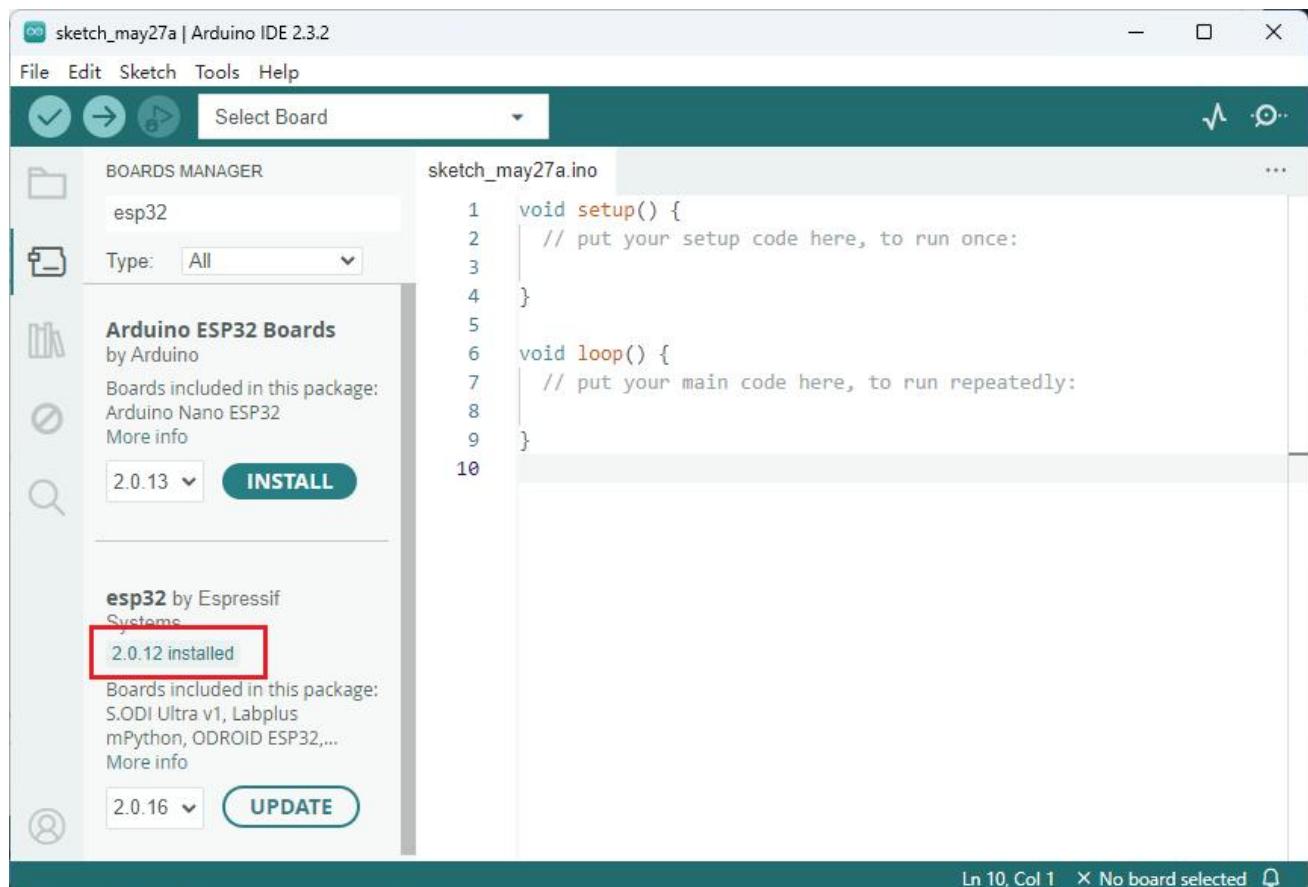
Please also complete the ARDUINO software download:

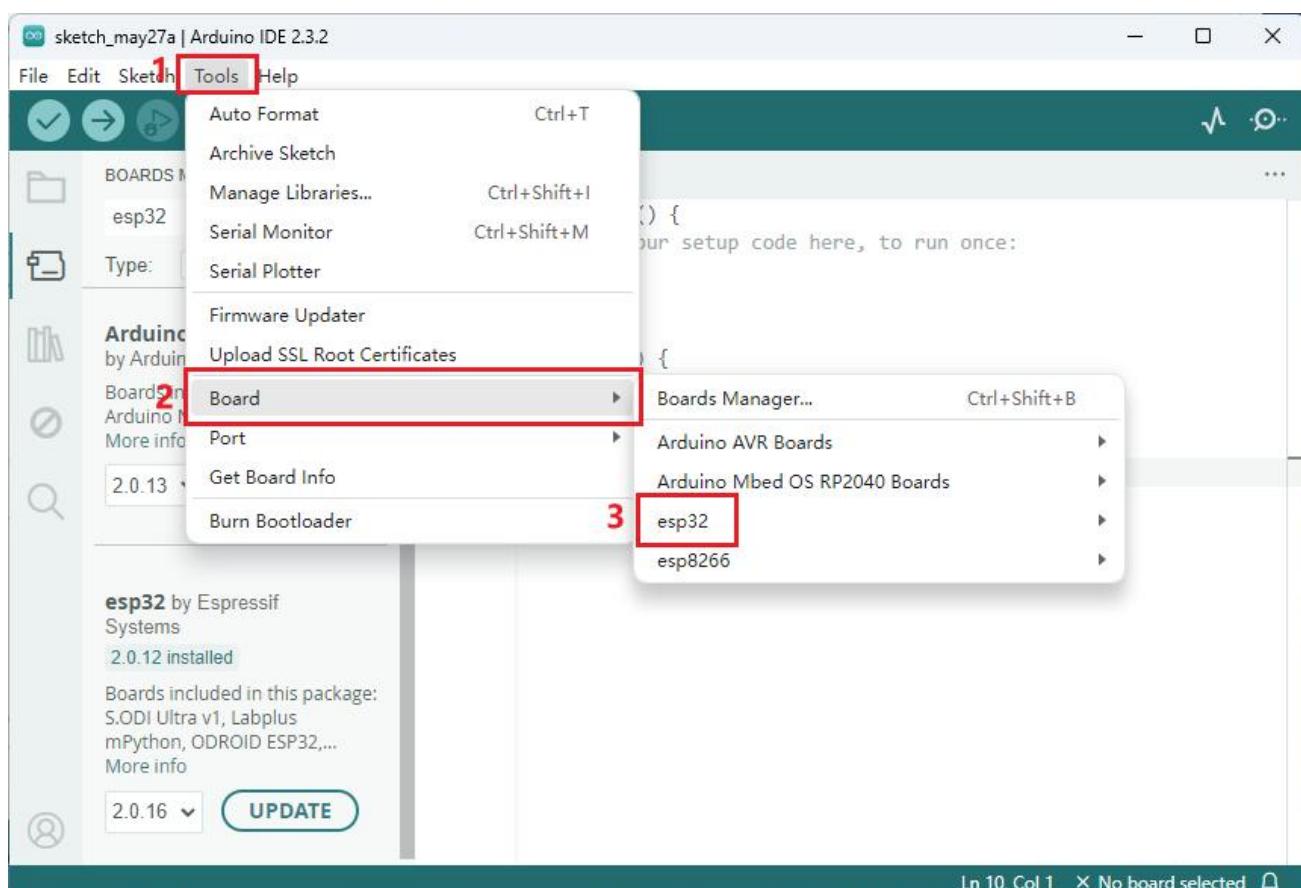


CH340 driver installation:



Install the ESP32 plugin:





And add the library:



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

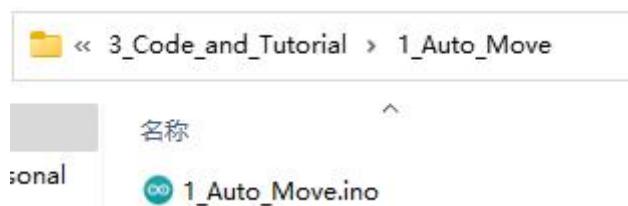
Lesson 1 Get The Car Moving

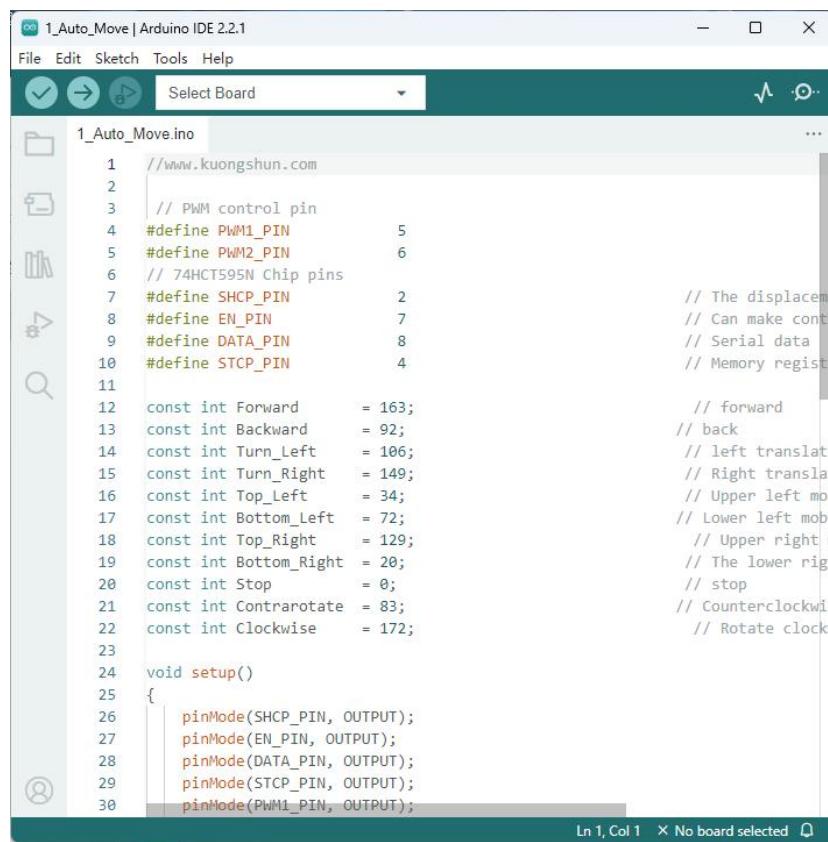
In this project, we will use Arduino compatible board combined with L293D extension board to realize the walking function of the car. This car adopts a four-wheeled Mecanum wheel structure, which can realize multi-directional translation and steering.

The Mecanum wheel is a form of tireless wheel, with a series of rubberized external rollers obliquely attached to the whole circumference of its rim. These rollers typically each have an axis of rotation at 45° to the wheel plane and at 45° to the axle line. Each Mecanum wheel is an independent non-steering drive wheel with its own power train, and when spinning generates a propelling force perpendicular to the roller axle, which can be vectored into a longitudinal and a transverse component in relation to the vehicle. Part of the force is canceled out when all four wheels move at the same time, so the car can move in a specified direction without changing direction.

We'll learn how to upload a program to a car first, and we'll learn how Mecanum wheels work later:

- ① Go to the following path and open the program





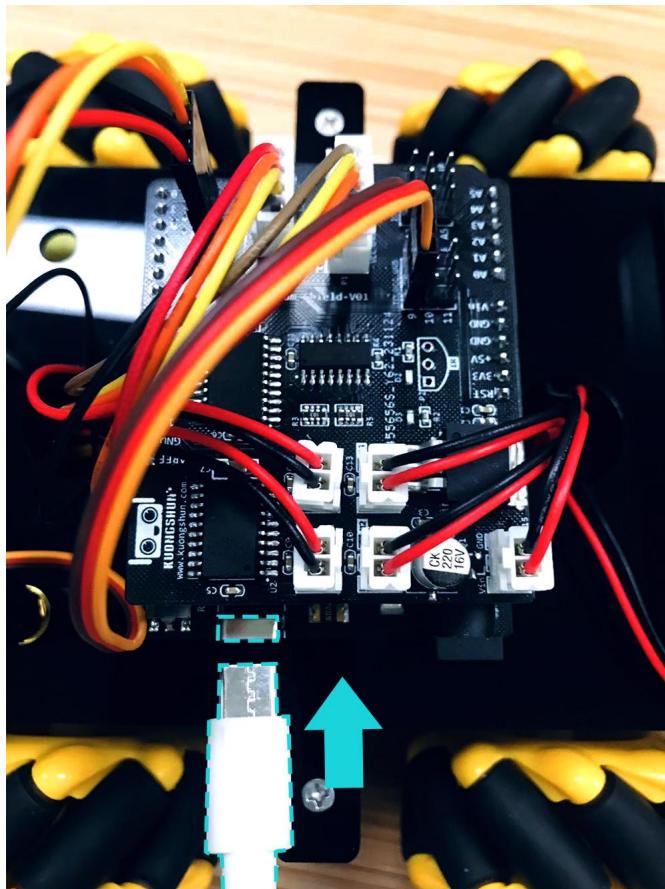
The screenshot shows the Arduino IDE interface with the file '1_Auto_Move.ino' open. The code defines pins for PWM control, chip pins, and various movement constants. It includes a setup function that initializes the pins as outputs.

```
// www.kuongshun.com
// PWM control pin
#define PWM1_PIN 5
#define PWM2_PIN 6
// 74HCT595N Chip pins
#define SHCP_PIN 2 // The displacement
#define EN_PIN 7 // Can make contr
#define DATA_PIN 8 // Serial data
#define STCP_PIN 4 // Memory regist
const int Forward = 163; // forward
const int Backward = 92; // back
const int Turn_Left = 106; // left translation
const int Turn_Right = 149; // Right translation
const int Top_Left = 34; // Upper left mobility
const int Bottom_Left = 72; // Lower left mobility
const int Top_Right = 129; // Upper right mobility
const int Bottom_Right = 20; // The lower right mobility
const int Stop = 0; // stop
const int Contrarotate = 83; // Counterclockwise
const int Clockwise = 172; // Rotate clockwise

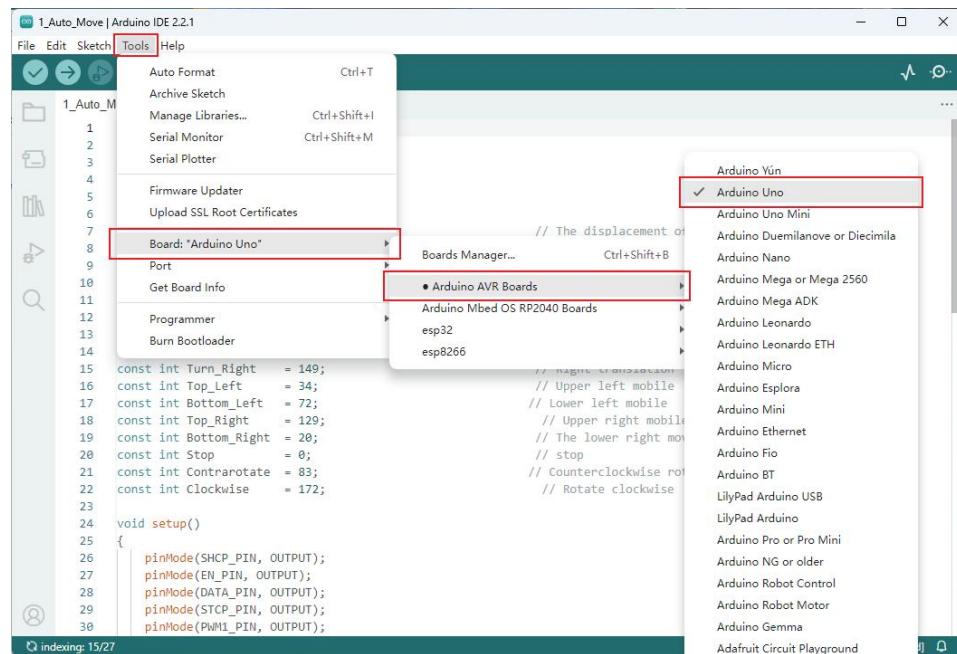
void setup()
{
    pinMode(SHCP_PIN, OUTPUT);
    pinMode(EN_PIN, OUTPUT);
    pinMode(DATA_PIN, OUTPUT);
    pinMode(STCP_PIN, OUTPUT);
    pinMode(PWM1_PIN, OUTPUT);
}
```

In 1, Col 1 × No board selected

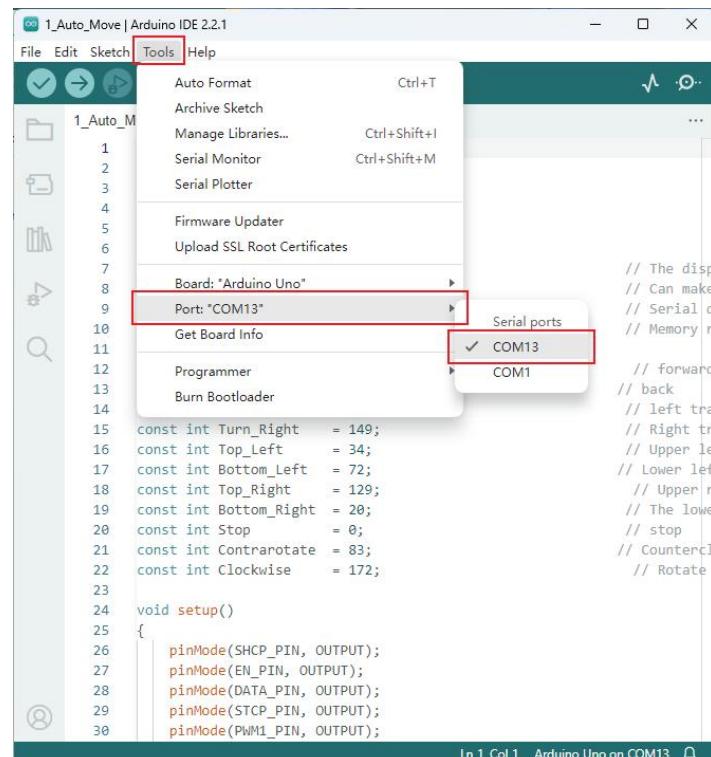
② Connect the Arduino board to your computer with a TYPE-C cable



③ Select your Board in Tools > Board >>Arduino AVR Board>>>Arduino UNO

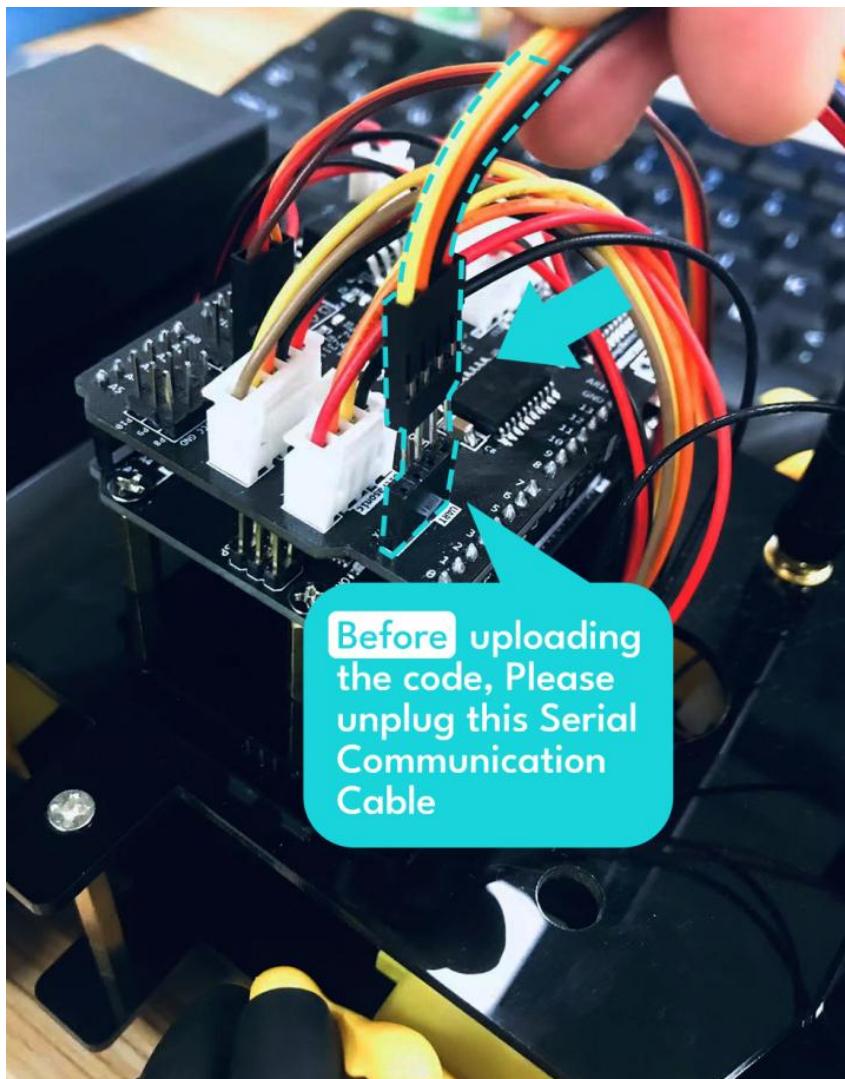


④Select the Port: Tools >Port >>>>COMxx(The COM port number of each computer may be different, and the port number that appears is your port number) (If you do not see COM ports other than COM1 in Arduino IDE, you need to refer to the installation of CH340 driver in "2_Programming_Preparation" and install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

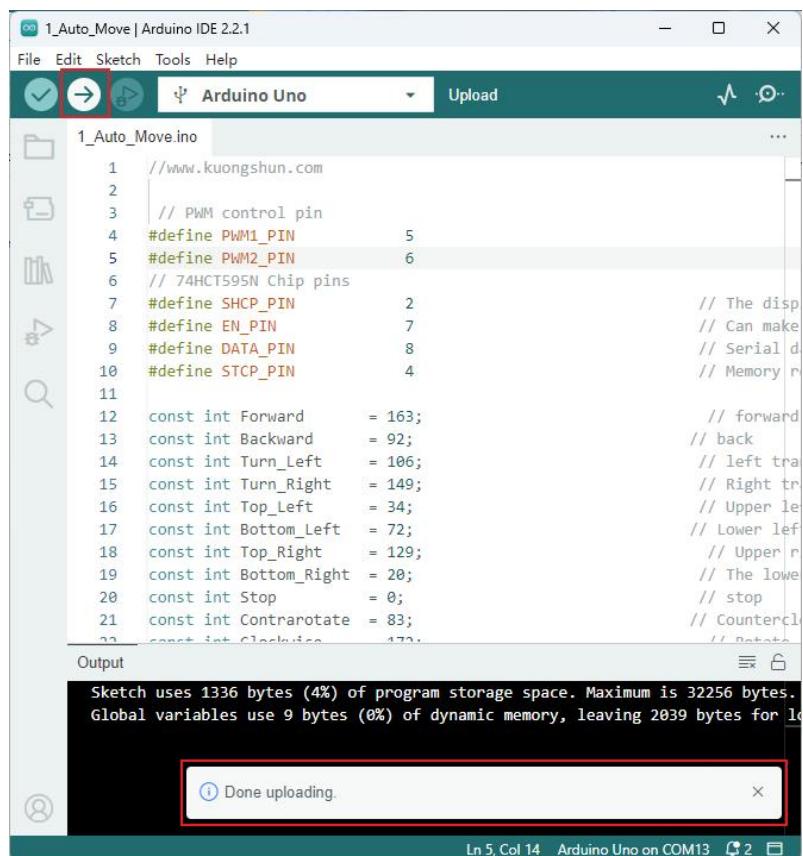


2_Programming_Preparation > 1. Installation of Arduino IDE and CH340 driver			
名称	修改日期	类型	大小
CH340 Serial Driver	2023/12/18 17:50	文件夹	
Mac OS Tutorial for Arduino IDE Installation...	2023/12/8 18:26	DOCX 文档	4,563 KB
Windows Tutorial for Arduino IDE Installatio...	2023/12/18 11:16	DOCX 文档	982 KB

⑤Unplug the serial port transmission cable between the Arduino board and the ESP32-CAM board.(This cable will not be connected back during all uploading procedures, until finally all procedures are uploaded and the car is controlled by WIFI.)



⑥Upload the program to the UNO controller board.



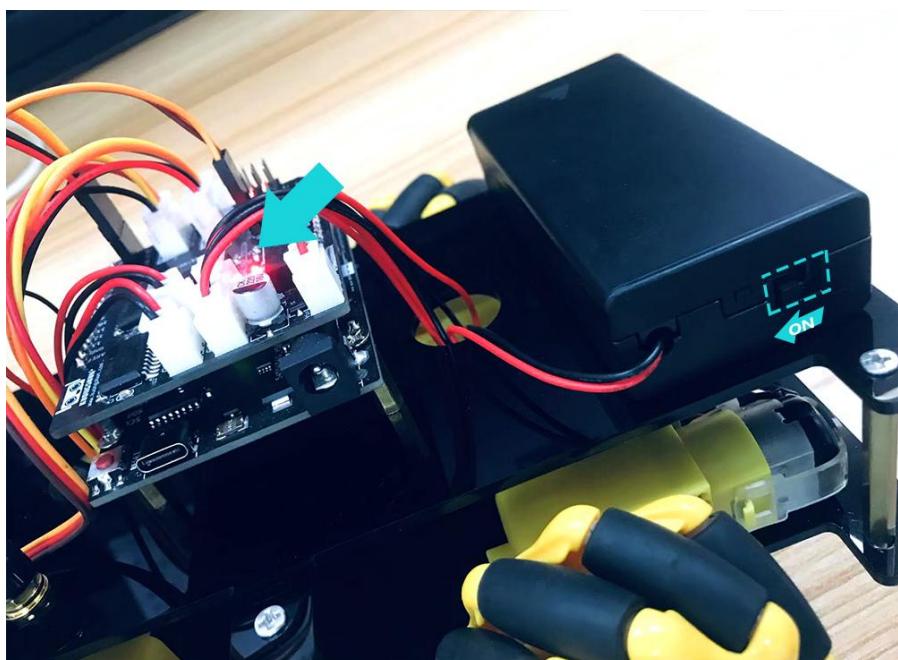
```

1_Auto_Move | Arduino IDE 2.2.1
File Edit Sketch Tools Help
Upload
1_Auto_Move.ino
1 //www.kuongshun.com
2
3 // PWM control pin
4 #define PWM1_PIN      5
5 #define PWM2_PIN      6
6 // 74HCT595N Chip pins
7 #define SHCP_PIN       2           // The disp
8 #define EN_PIN         7           // Can make
9 #define DATA_PIN       8           // Serial d
10 #define STCP_PIN       4           // Memory r
11
12 const int Forward     = 163;      // forward
13 const int Backward    = 92;       // back
14 const int Turn_Left   = 106;      // left tra
15 const int Turn_Right  = 149;      // Right tr
16 const int Top_Left    = 34;       // Upper le
17 const int Bottom_Left = 72;       // Lower lef
18 const int Top_Right   = 129;      // Upper r
19 const int Bottom_Right= 20;       // The lowe
20 const int Stop        = 0;        // stop
21 const int Contrarotate= 83;      // Countercl
22 const int Clockwise   = 172;     // Rotatio
Output
Sketch uses 1336 bytes (4%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables.
Ln 5, Col 14  Arduino Uno on COM13  2  2
Done uploading.

```

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 two 18650 batteries.) Then turn on the switch.

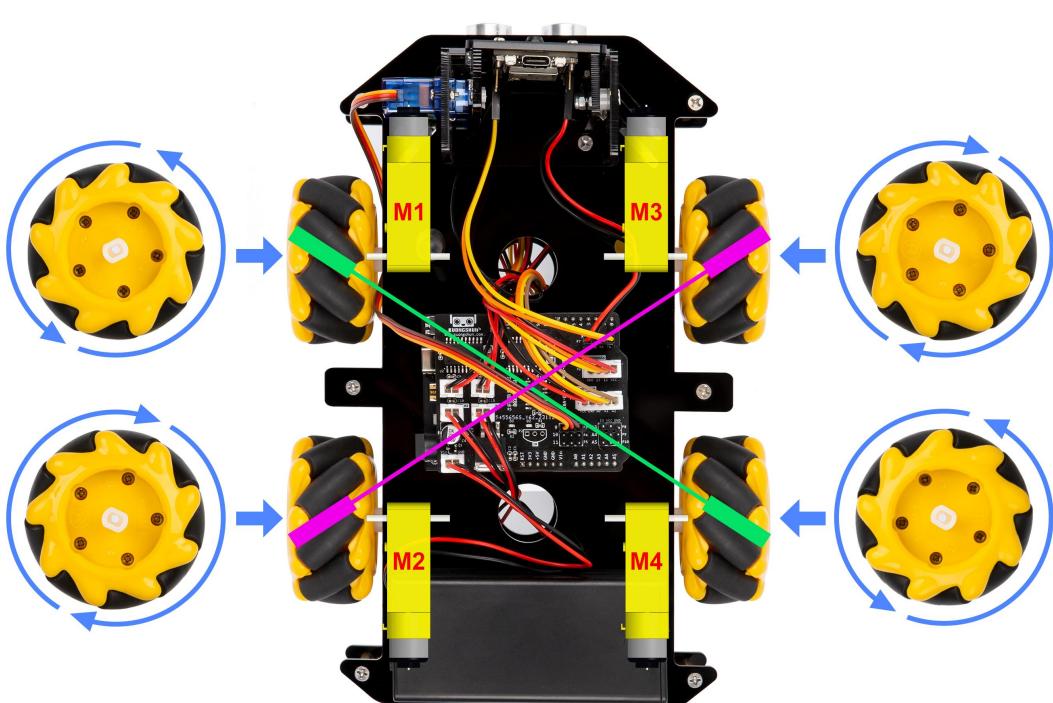


At this point, the car will go forward - backward - go left - go right - go up left - go down right - go down left - go up right - rotate clockwise - rotate counterclockwise - stop, and these 11 states will run for 2 seconds in turn and cycle.

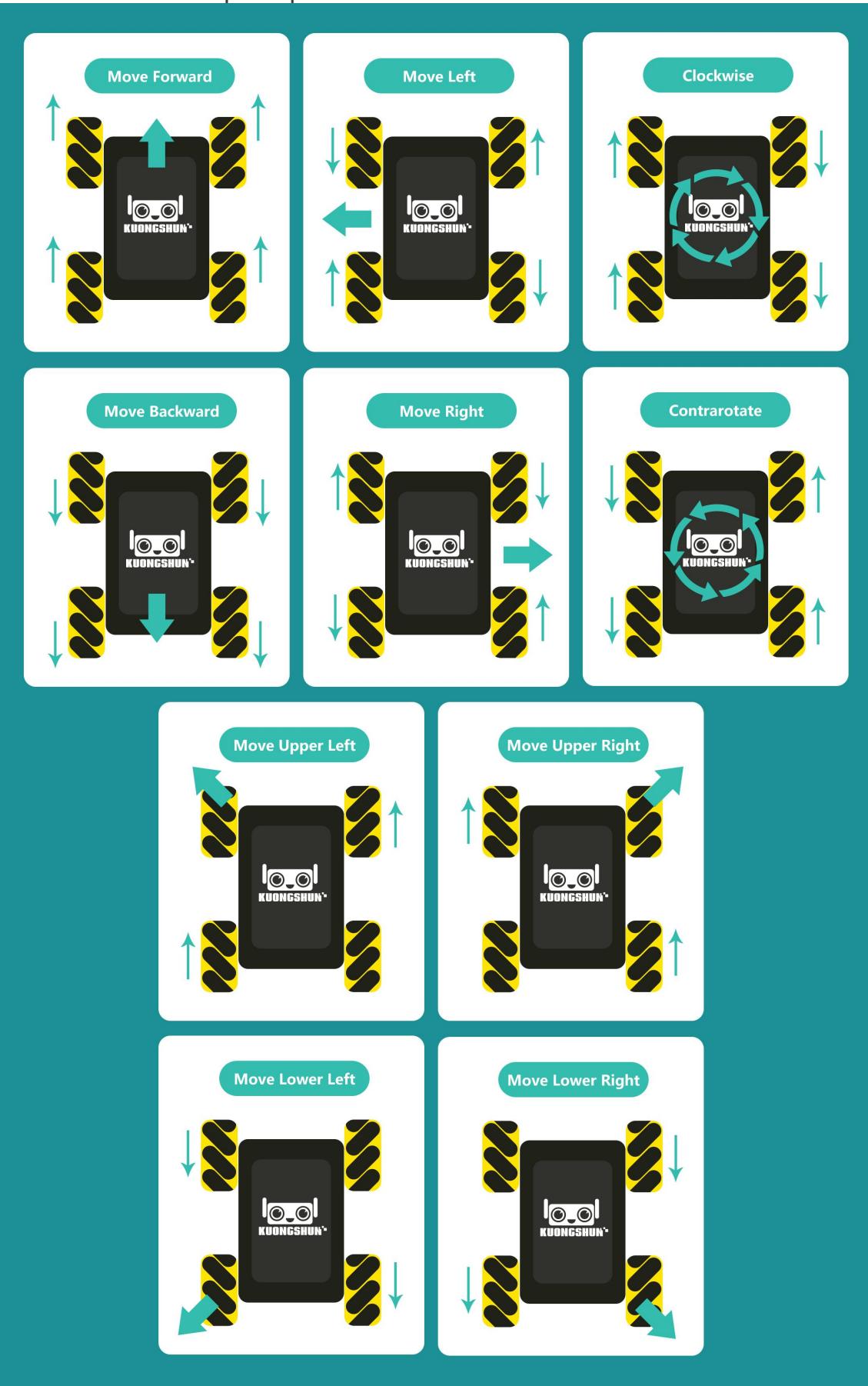
Let's see the Mecanum wheel. Here are the wheels in two directions:



Almost all 4WD Mecanum wheels are installed with this mounting:::



Motor drive and motion principle:

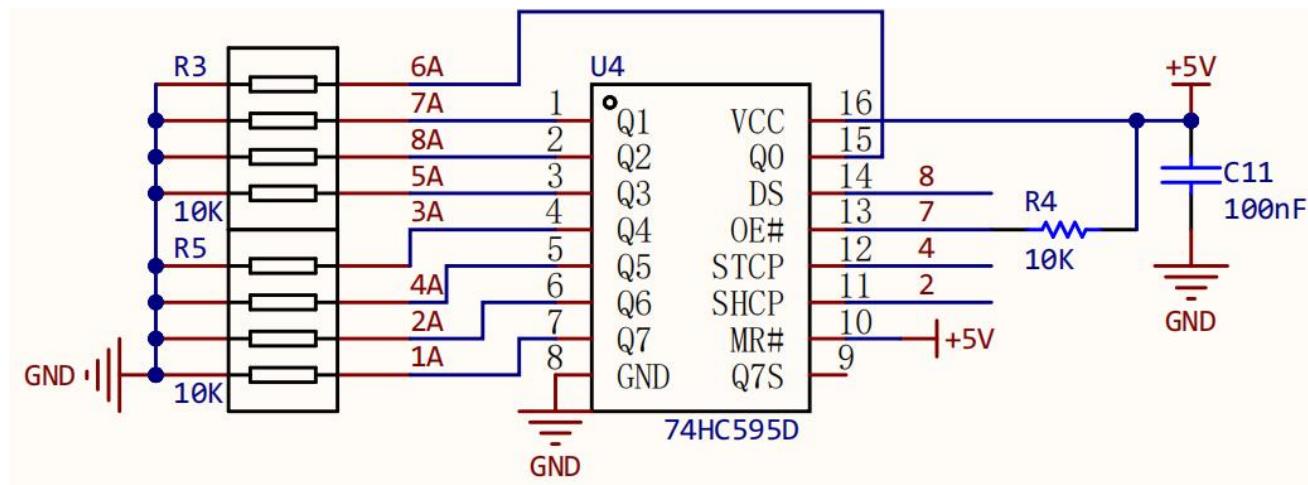


In the code, PWM pins 5,6 control the power (speed) of the wheel, and then through the 74HC595 chip pins 2,7,8,4 control the rotation direction of each motor.

```
// PWM control pin
#define PWM1_PIN 5
#define PWM2_PIN 6
// 74HCT595N Chip pins
#define SHCP_PIN 2
#define EN_PIN 7
#define DATA_PIN 8
#define STCP_PIN 4
```

We can see that the 74HC595D signal port is connected to only four, and a motor generally requires two signal ports to control the direction. Let's learn how the 74HC595D works

Below is the chip wiring diagram of 74HC595D

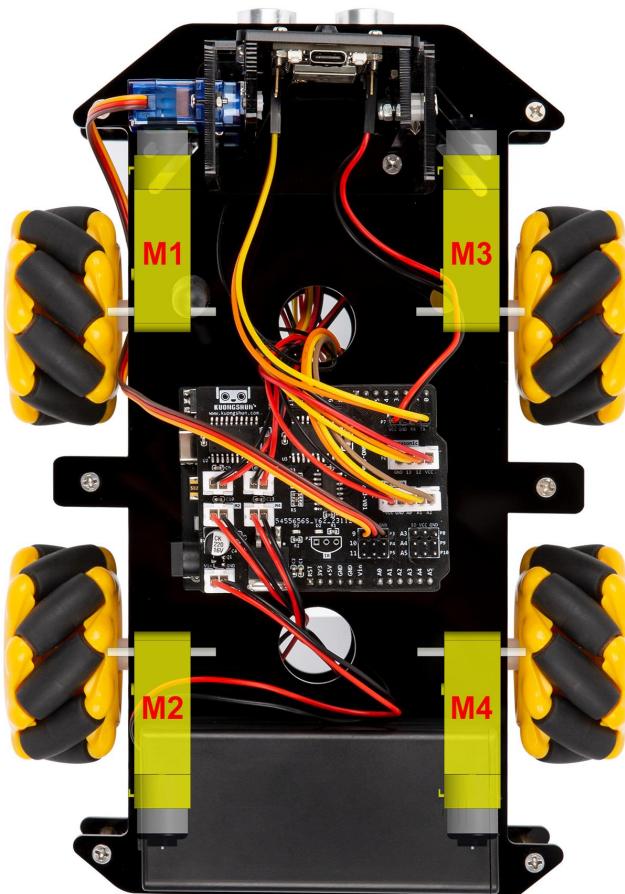


In the core operation program of the motor, the shiftOut function is the main one. The high and low levels of each pin of the motor are controlled by decimal numbers converted from 8-bit binary numbers in the range 0-255.

The shiftOut function has a total of four parameters. The first three parameters are defined and configured at the beginning, and we only need to change the value of Dir. At this time, the system converts the decimal number into an 8-bit binary number to control the high and low levels of the motor.

```
void Motor(int Dir, int Speed)
{
    digitalWrite(EN_PIN, LOW);
    analogWrite(PWM1_PIN, Speed);
    analogWrite(PWM2_PIN, Speed);
    digitalWrite(STCP_PIN, LOW);
    shiftOut(DATA_PIN, SHCP_PIN, MSBFIRST, Dir);
    digitalWrite(STCP_PIN, HIGH);
}
```

For example, if you want a car to move forward, you have to keep all four wheels moving forward. The corresponding 8-bit code is 10100011, which converts to the decimal value 163. If you want the car to go backwards, you have to keep all four wheels turning backwards. The corresponding 8-bit code is 01011100, which converts to the decimal value 92.



Coding value table corresponding to different motion states

(When M1+ is 1 and M1- is 0, M1 advances,

When M1+ is 0, M1- is 1, M1 backs off, and the other motors are similar)

State Of The Car	M1+	M1-	M2+	M2-	M4-	M3-	M3+	M4+	8 bit binary	Decimal system
Move Forward	1	0	1	0	0	0	1	1	10100011	163
Move Backward	0	1	0	1	1	1	0	0	01011100	92
Move Left	0	1	1	0	1	0	1	0	01101010	106
Move Right	1	0	0	1	0	1	0	1	10010101	149
Move Upper Left	0	0	1	0	0	0	1	0	00100010	34
Move Lower Left	0	1	0	0	1	0	0	0	01001000	72
Move Upper Right	1	0	0	0	0	0	0	1	10000001	129
Move Lower Right	0	0	0	1	0	1	0	0	00010100	20
Stop	0	0	0	0	0	0	0	0	00000000	0
Contrarotate	0	1	0	1	0	0	1	1	01010011	83
Clockwise	1	0	1	0	1	1	0	0	10101100	172

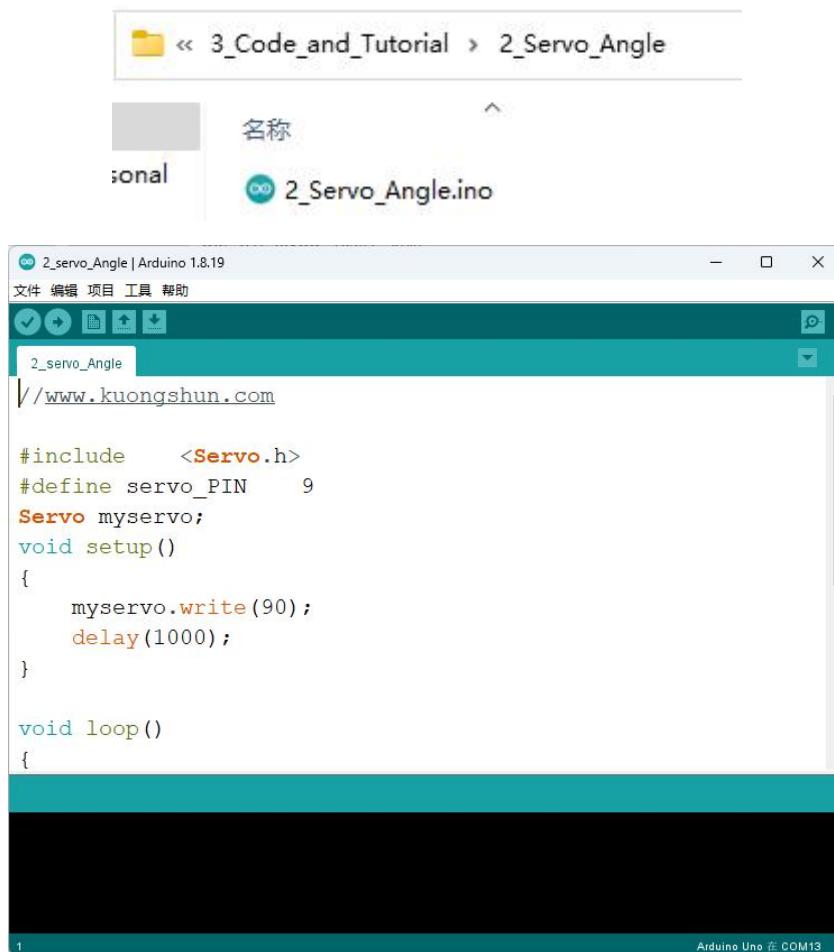
Set a variable for each motion state that stores the decimal code corresponding to the car's motion state:

```
const int Forward      = 163;
const int Backward    = 92;
const int Turn_Left   = 106;
const int Turn_Right  = 149;
const int Top_Left    = 34;
const int Bottom_Left = 72;
const int Top_Right   = 129;
const int Bottom_Right= 20;
const int Stop         = 0;
const int Contrarotate= 83;
const int Clockwise   = 172;
```

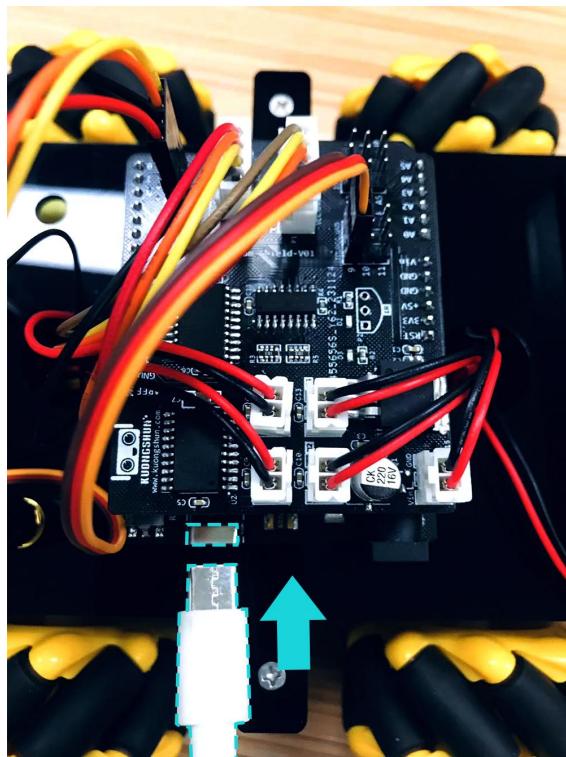
Lesson 2 Servo Pan-Tilt

In this project, we will learn the use of steering gear. It also tests and corrects the Angle of the servo motor. The first thing to do is to upload the servo Angle test program to the Arduino board:

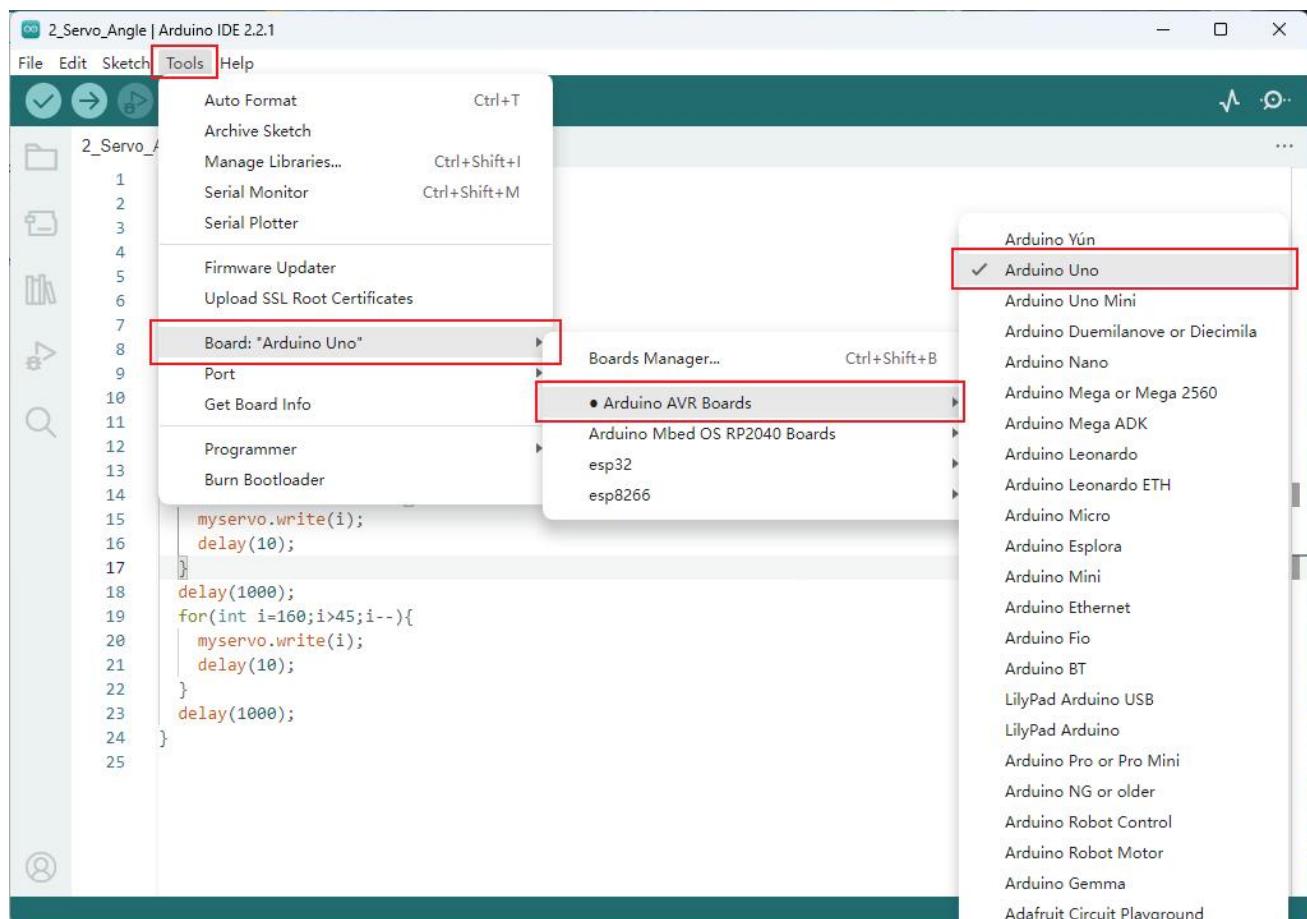
- ① Start by going to the following path and opening the program:



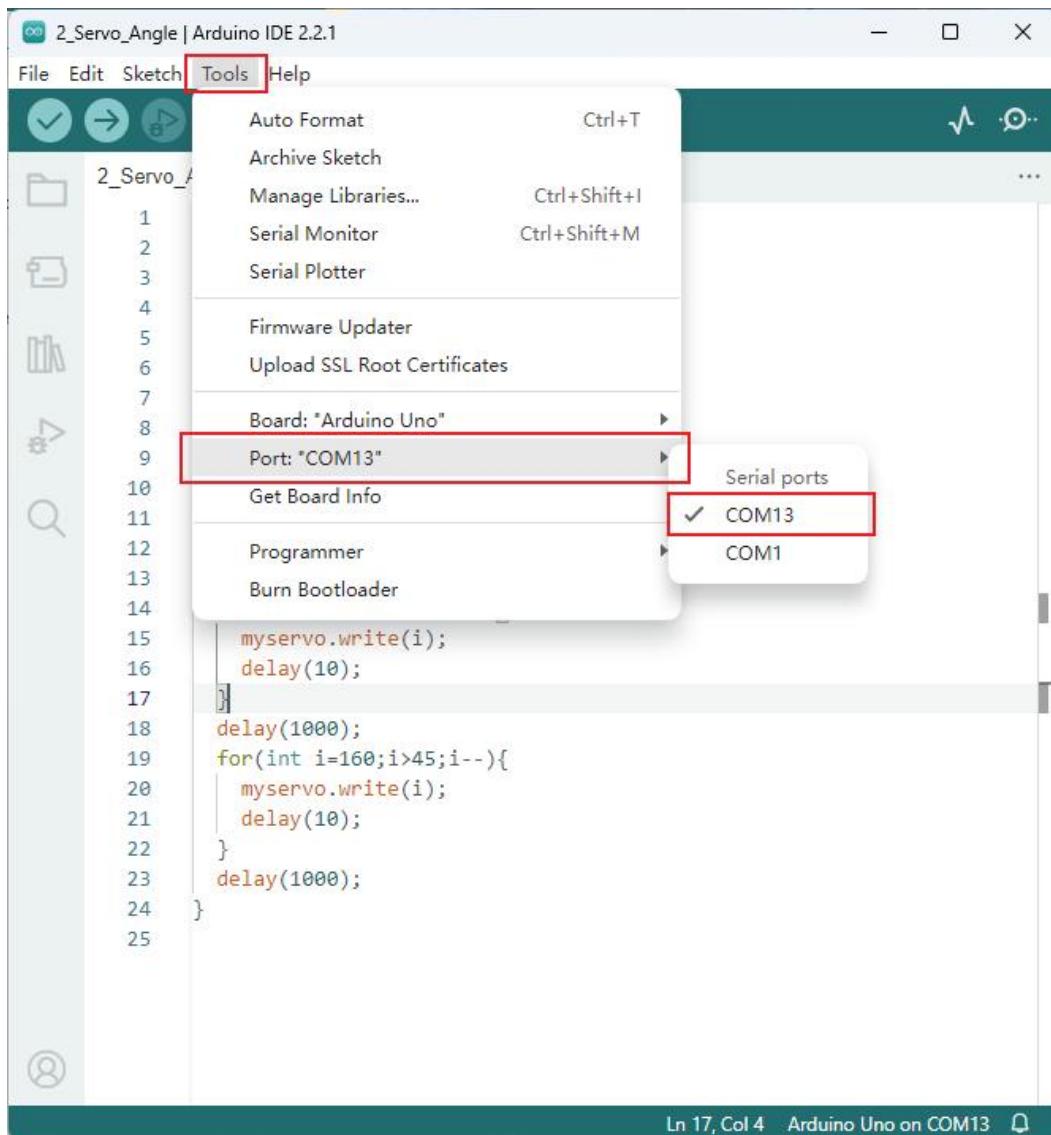
- ② Connect the Arduino board to the computer with TYPE-C data cable



③ Select your control panel: Tools>Development board >> Arduino AVR Board>>> Arduino UNO



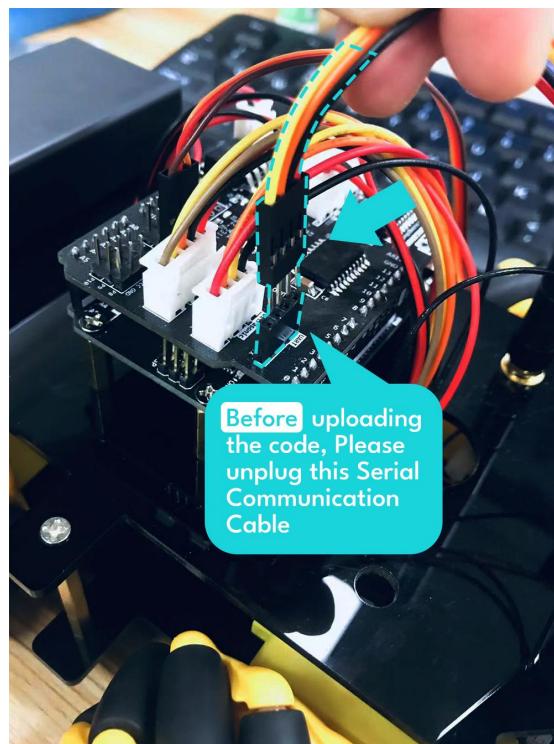
④Select the Port: Tools >Port >>>>COMxx(The COM port number of each computer may be different, and the port number that appears is your port number) (If you do not see COM ports other than COM1 in Arduino IDE, you need to refer to the installation of CH340 driver in "2_Programming_Preparation" and install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



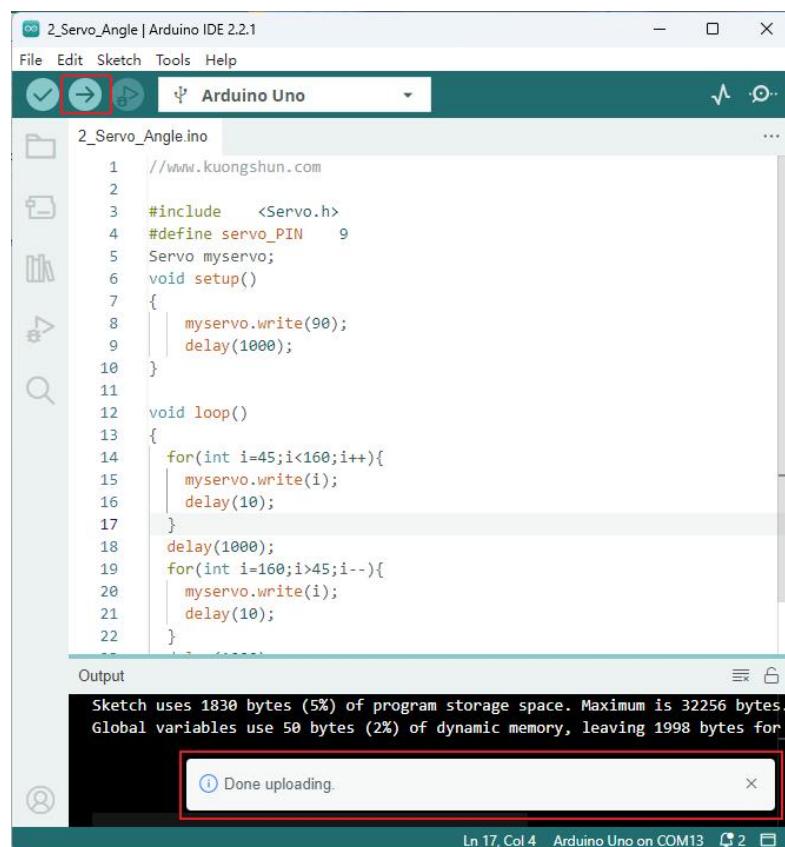
2_Programming_Preparation > 1. Installation of Arduino IDE and CH340 driver				
名称	修改日期	类型	大小	操作
CH340 Serial Driver	2023/12/18 17:50	文件夹		
Mac OS Tutorial for Arduino IDE Installation...	2023/12/8 18:26	DOCX 文档	4,563 KB	
Windows Tutorial for Arduino IDE Installatio...	2023/12/18 11:16	DOCX 文档	982 KB	

⑤Unplug the serial port transmission cable between the Arduino board and the

ESP32-CAM board.(This cable will not be connected back during all uploading procedures, until finally all procedures are uploaded and the car is controlled by WIFI.)



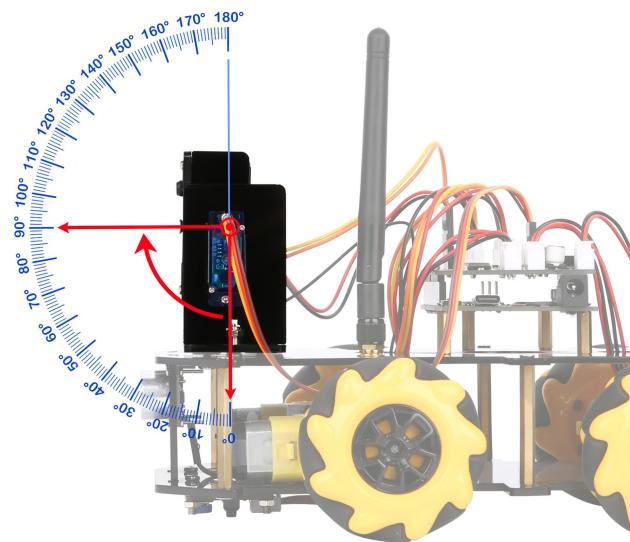
⑥Upload the program to the UNO controller board.



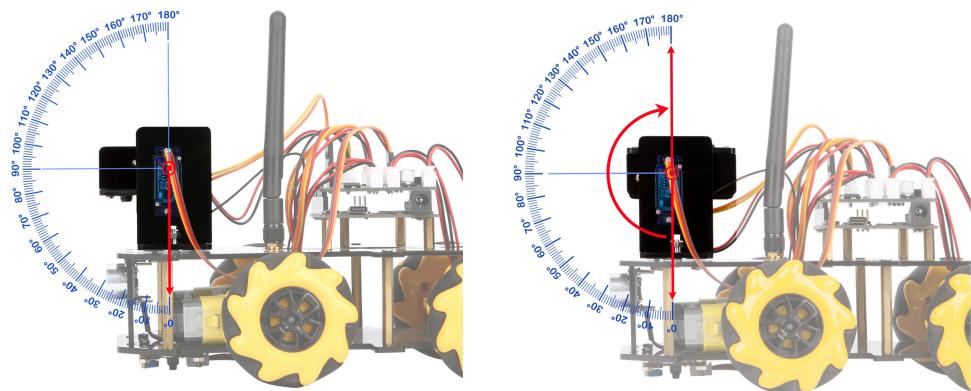
```
2_Servo_Angle | Arduino IDE 2.2.1
File Edit Sketch Tools Help
  ↗ → ↘ Arduino Uno
2_Servo_Angle.ino
1 //www.kuongshun.com
2
3 #include <Servo.h>
4 #define servo_PIN 9
5 Servo myservo;
6 void setup()
7 {
8     myservo.write(90);
9     delay(1000);
10 }
11
12 void loop()
13 {
14     for(int i=45;i<160;i++){
15         myservo.write(i);
16         delay(10);
17     }
18     delay(1000);
19     for(int i=160;i>45;i--){
20         myservo.write(i);
21         delay(10);
22     }
}
Output
Sketch uses 1830 bytes (5%) of program storage space. Maximum is 32256 bytes.
Global variables use 50 bytes (2%) of dynamic memory, leaving 1998 bytes for local variables.
Done uploading.
```

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 two 18650 batteries.) Then turn on the switch.The servo motor of the locomotive will turn to the 90 degree position, stay for one second and then swing up and down. We specify that the orientation is straight forward at 90 degrees, as shown below:



The amplitude of the back-and-forth swing is shown below:

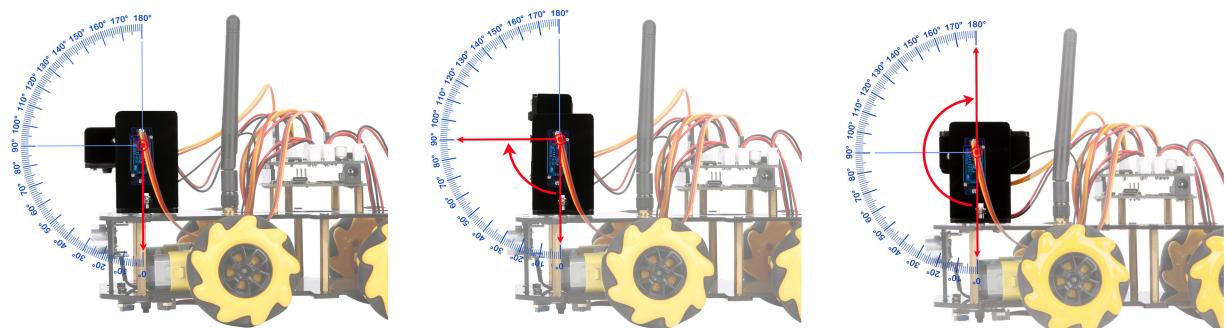


If the swing amplitude of the car is not consistent with the figure above, it shows that your servo placement is not very accurate. We need to press the reset button of the Arduino board to bring the servo back to 90 degrees. Then unplug the data cable to stop the steering gear. At this time, separate the servomotor rocker arm from the ESP32-CAM support bracket as shown in the figure.Straightening the whole structure

and then buckle back.



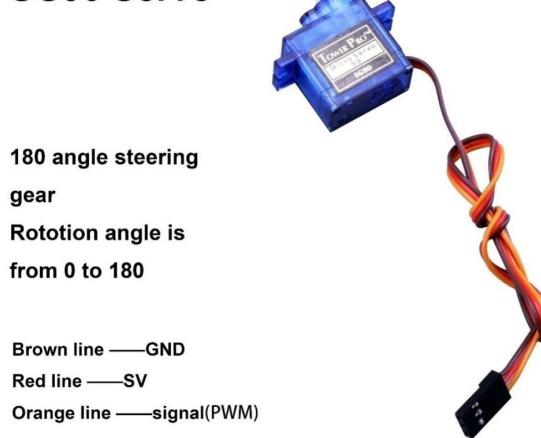
Here's the swing:



Principle Introduction:

SG90 servo motor

SG90 Servo



Classification: 180 steering gear

Usually the servo has three control lines: power line, ground line and signal line.

Definition of servo pin: brown line --GND, red line --5V, orange line -- signal.

How the servo motor works:

The signal modulation chip in the servo motor receives the signal from the control board, and then the servo motor will obtain the basic DC voltage. There is also a reference circuit inside the servo motor, which will generate a standard voltage. These two voltages are compared with each other and then the difference is output. The motor chip will then receive the difference and decide on the speed, direction, and angle. The servo motor stops when there is no difference between the two voltages.

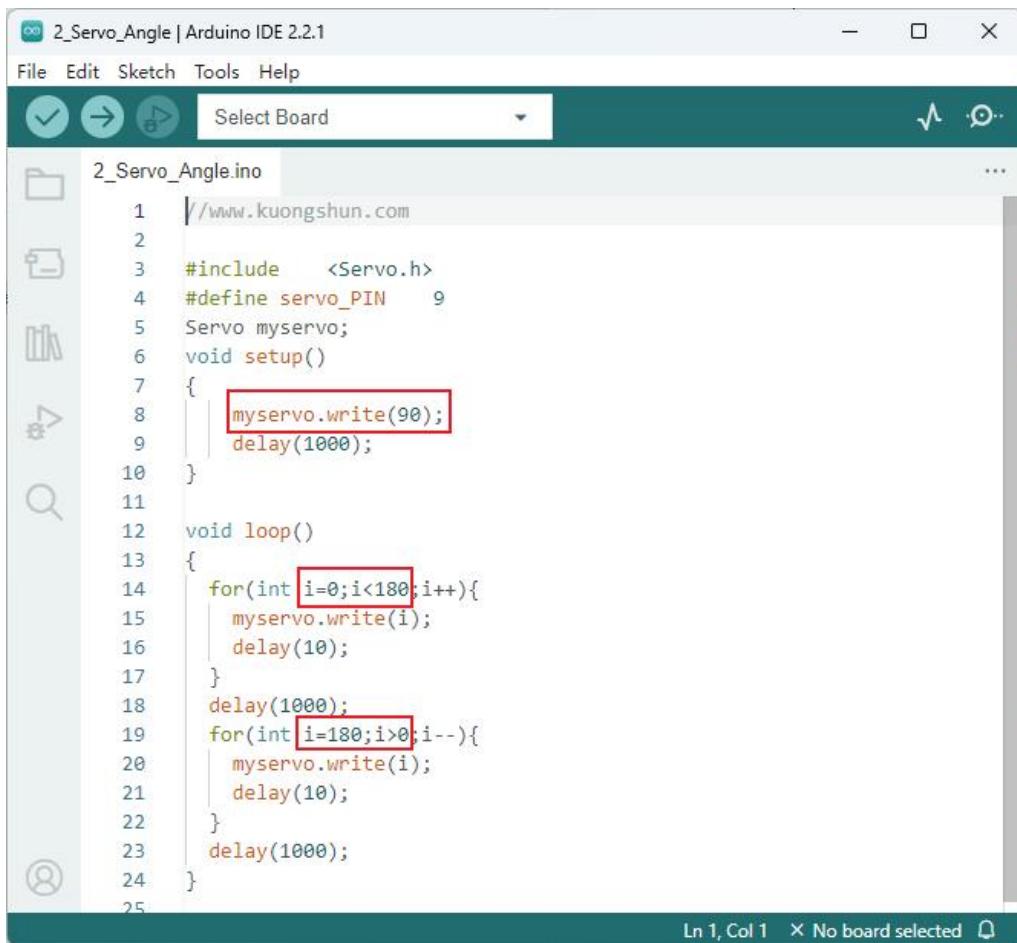
Servo motor control principle:

To control the servo rotation, it is necessary to control the time pulse at about 20ms and the high level pulse width at about 0.5ms to 2.5ms, which is consistent with the angular limit of the servo.

Taking 180 Angle servo as an example, the corresponding control relationship is as follows:

0.5ms	0 度
1.0ms	45 度
1.5ms	90 度
2.0ms	135 度
2.5ms	180 度

At the marked position of the program in the following figure, we can modify the initial Angle and rotation range of the servo. Follow the same steps to upload the program once the changes are made. Then we can see the result of the change.



The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** 2_Servo_Angle | Arduino IDE 2.2.1
- Menu Bar:** File Edit Sketch Tools Help
- Toolbar:** Includes icons for Save, Load, Open, and Select Board.
- Select Board:** A dropdown menu currently set to "Select Board".
- Code Area:** Displays the sketch named "2_Servo_Angle.ino".
- Code Content:** The code is as follows:

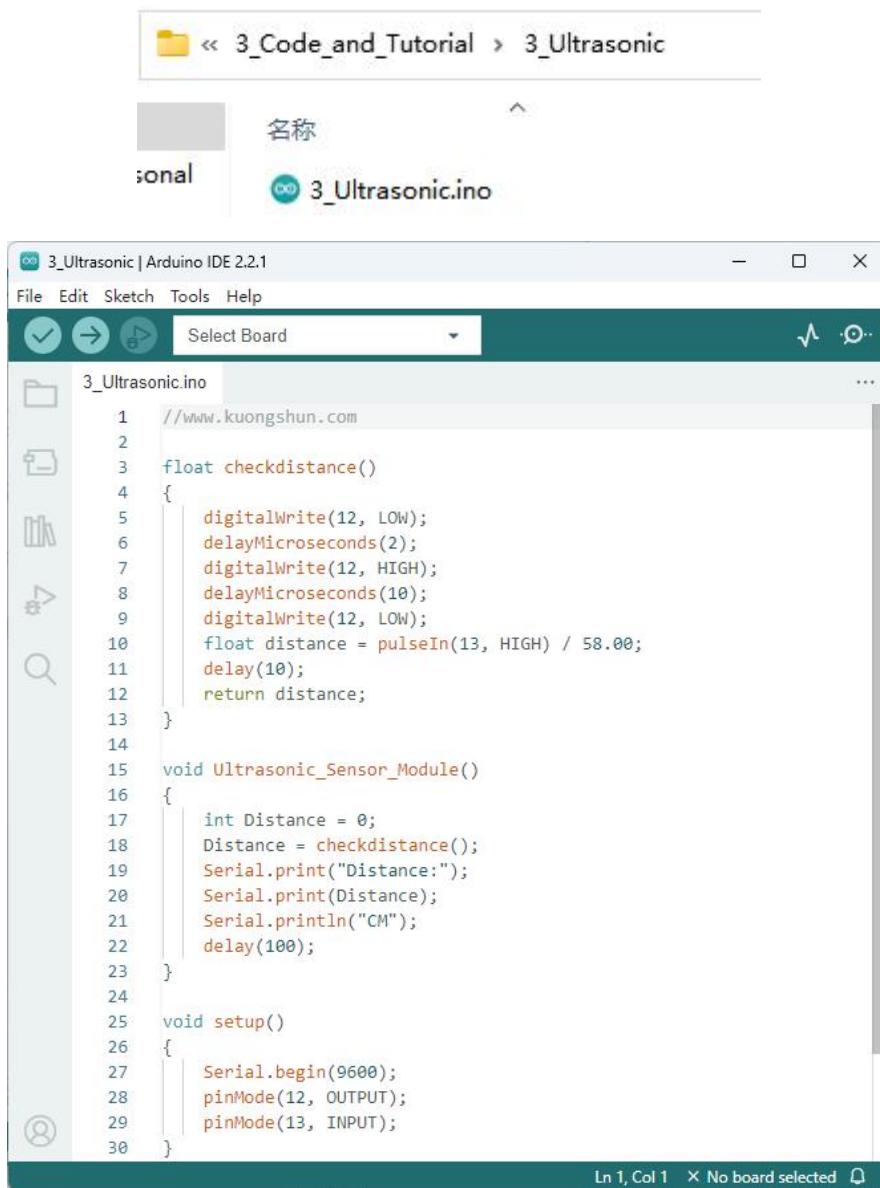
```
1 //www.kuongshun.com
2
3 #include <Servo.h>
4 #define servo_PIN 9
5 Servo myservo;
6 void setup()
7 {
8     myservo.write(90);
9     delay(1000);
10 }
11
12 void loop()
13 {
14     for(int i=0;i<180;i++){
15         myservo.write(i);
16         delay(10);
17     }
18     delay(1000);
19     for(int i=180;i>0;i--){
20         myservo.write(i);
21         delay(10);
22     }
23     delay(1000);
24 }
```

The code uses a servo connected to pin 9 to move between 0 and 180 degrees, with intermediate steps every 10 degrees. It includes a 1-second delay between cycles.

Lesson 3 Ultrasonic Sensor Module

In this project, we will learn how to use the ultrasound module and how to use the serial port monitor to view the distance data returned by the ultrasound. Before we do this, we need to upload the program to the car:

- ① Start by going to the following path and opening the program:



The screenshot shows the Arduino IDE interface. At the top, there's a file browser window titled "3_Ultrasonic" with a folder icon. Below it, the main workspace shows the Arduino sketch "3_Ultrasonic.ino". The code itself is as follows:

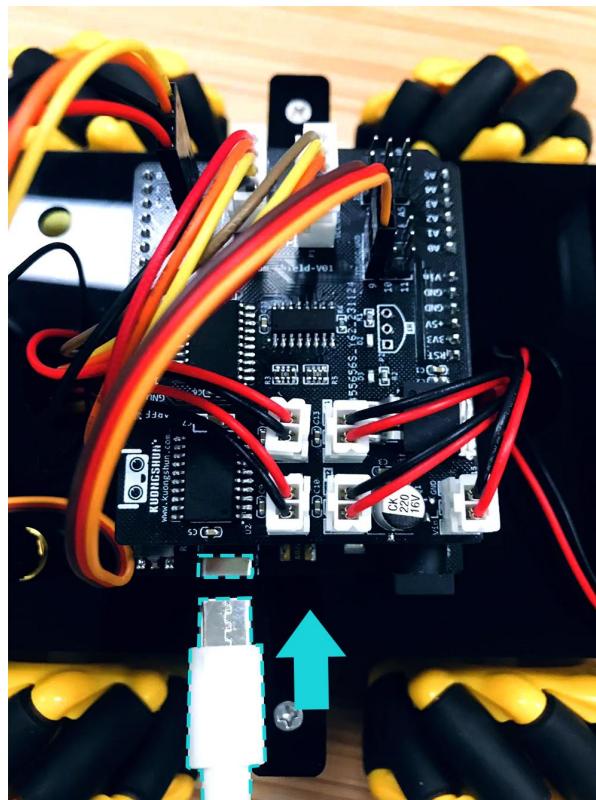
```
//www.kuongshun.com
float checkdistance()
{
    digitalWrite(12, LOW);
    delayMicroseconds(2);
    digitalWrite(12, HIGH);
    delayMicroseconds(10);
    digitalWrite(12, LOW);
    float distance = pulseIn(13, HIGH) / 58.00;
    delay(10);
    return distance;
}

void Ultrasonic_Sensor_Module()
{
    int Distance = 0;
    Distance = checkdistance();
    Serial.print("Distance:");
    Serial.print(Distance);
    Serial.println("CM");
    delay(100);
}

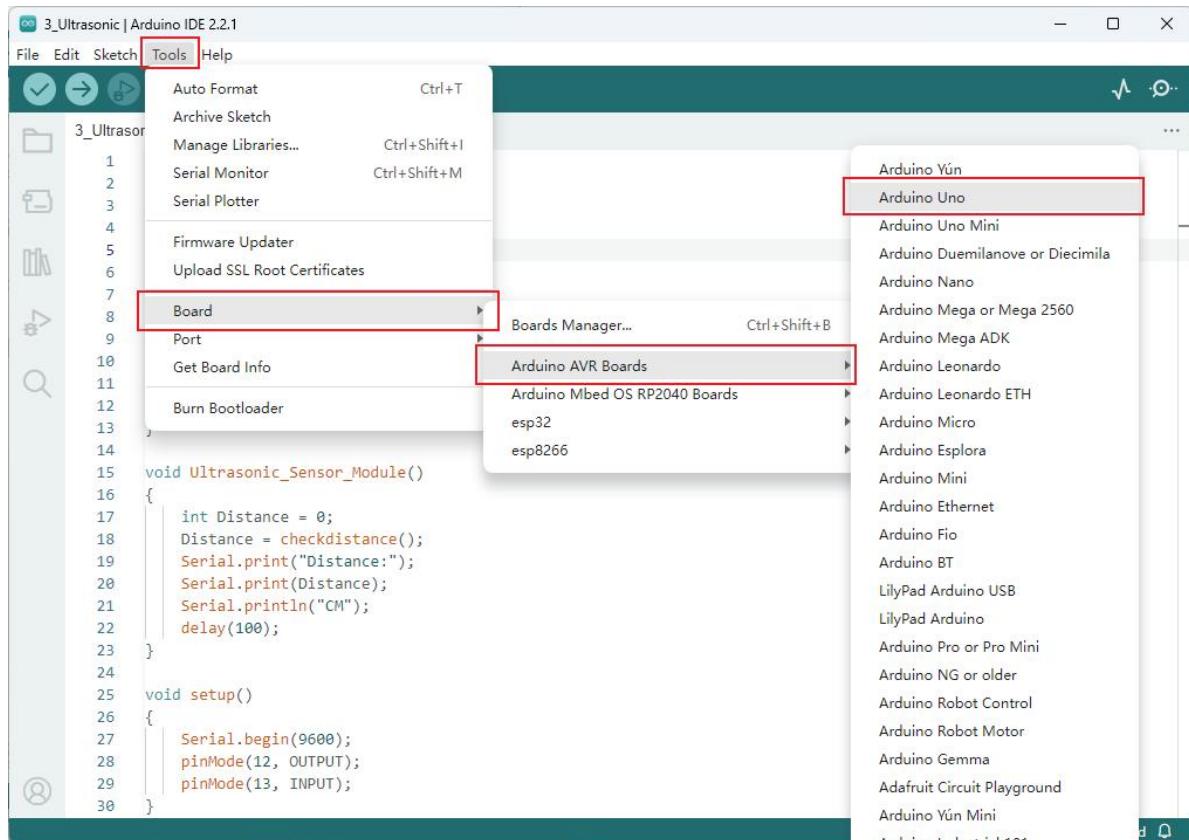
void setup()
{
    Serial.begin(9600);
    pinMode(12, OUTPUT);
    pinMode(13, INPUT);
}
```

At the bottom right of the IDE, status messages say "Ln 1, Col 1" and "No board selected".

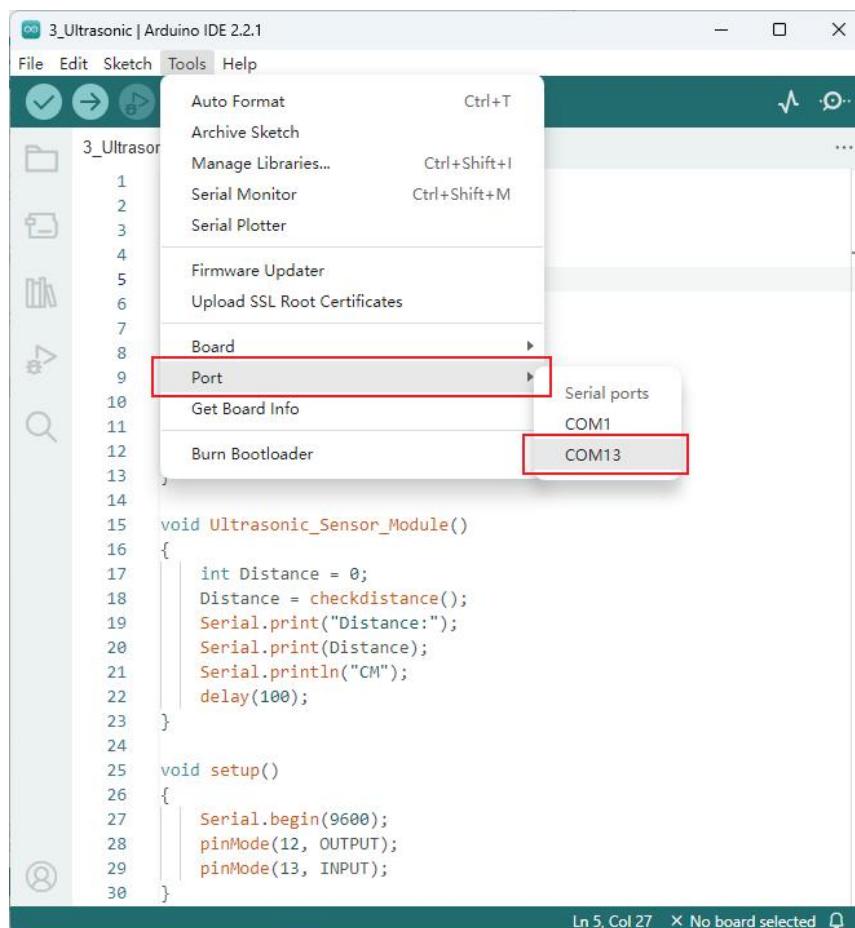
- ② Connect the Arduino board to the computer with TYPE-C data cable



③ Select your control board: Tools > Board >> Arduino AVR Board >>> Arduino UNO

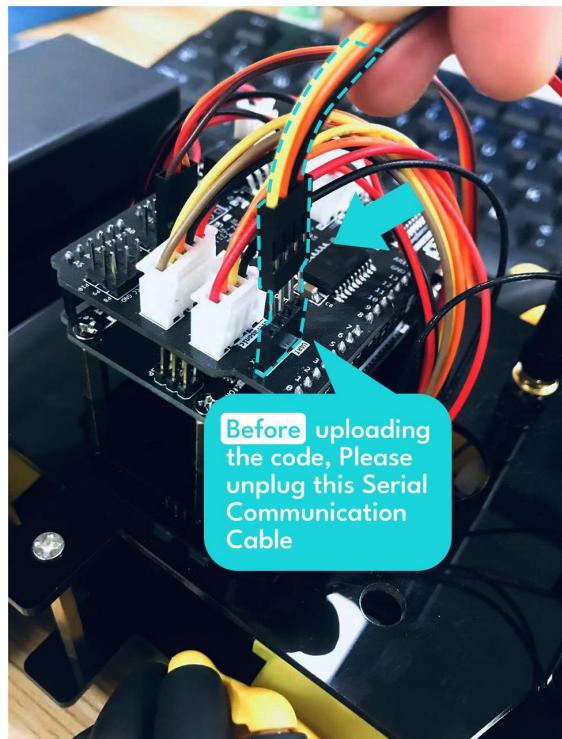


④Select the Port: Tools >Port >>>>COMxx(The COM port number of each computer may be different, and the port number that appears is your port number) (If you do not see COM ports other than COM1 in Arduino IDE, you need to refer to the installation of CH340 driver in "2_Programming_Preparation" and install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

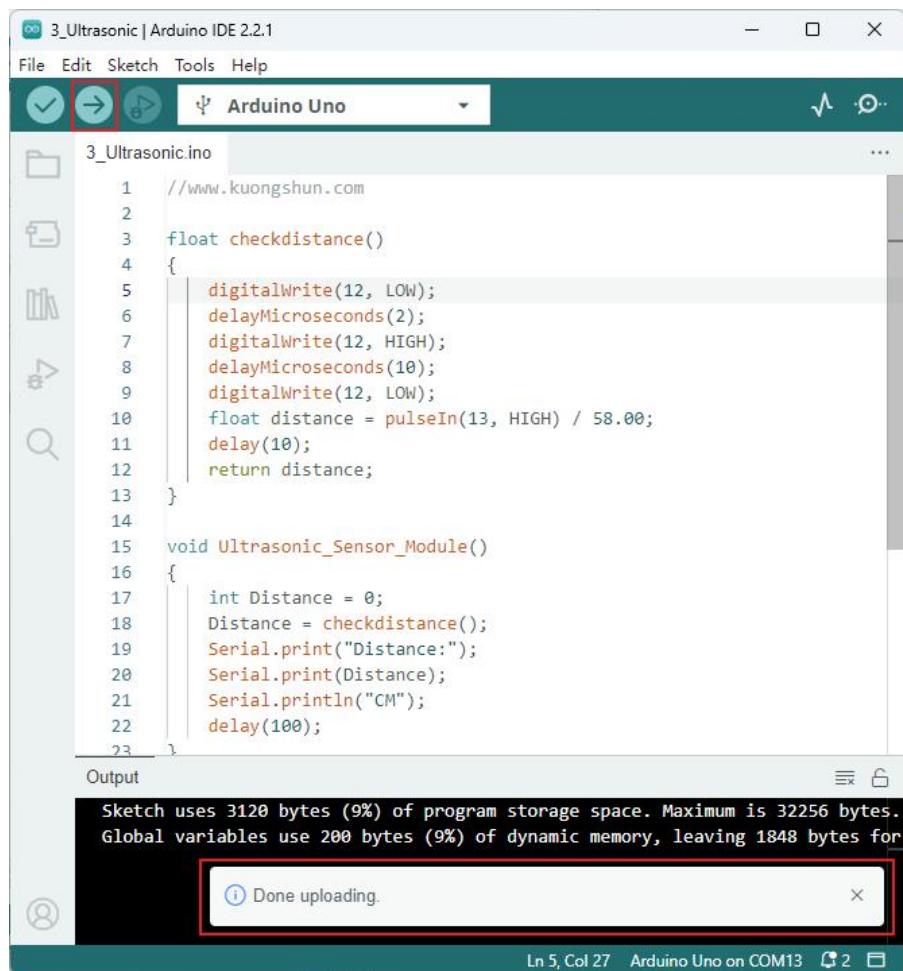


2_Programming_Preparation > 1. Installation of Arduino IDE and CH340 driver >			
名称	修改日期	类型	大小
CH340 Serial Driver	2023/12/18 17:50	文件夹	
Mac OS Tutorial for Arduino IDE Installation...	2023/12/8 18:26	DOCX 文档	4,563 KB
Windows Tutorial for Arduino IDE Installatio...	2023/12/18 11:16	DOCX 文档	982 KB

⑤Unplug the serial port transmission cable between the Arduino board and the ESP32-CAM board.(This cable will not be connected back during all uploading procedures, until finally all procedures are uploaded and the car is controlled by WIFI.)



⑥Upload the program to the UNO controller board.



```
3_Ultrasonic | Arduino IDE 2.2.1
File Edit Sketch Tools Help
Upload (highlighted) Arduino Uno ...
3_Ultrasonic.ino
1 //www.kuongshun.com
2
3 float checkdistance()
4 {
5     digitalWrite(12, LOW);
6     delayMicroseconds(2);
7     digitalWrite(12, HIGH);
8     delayMicroseconds(10);
9     digitalWrite(12, LOW);
10    float distance = pulseIn(13, HIGH) / 58.00;
11    delay(10);
12    return distance;
13 }
14
15 void Ultrasonic_Sensor_Module()
16 {
17     int Distance = 0;
18     Distance = checkdistance();
19     Serial.print("Distance:");
20     Serial.print(Distance);
21     Serial.println("CM");
22     delay(100);
23 }
```

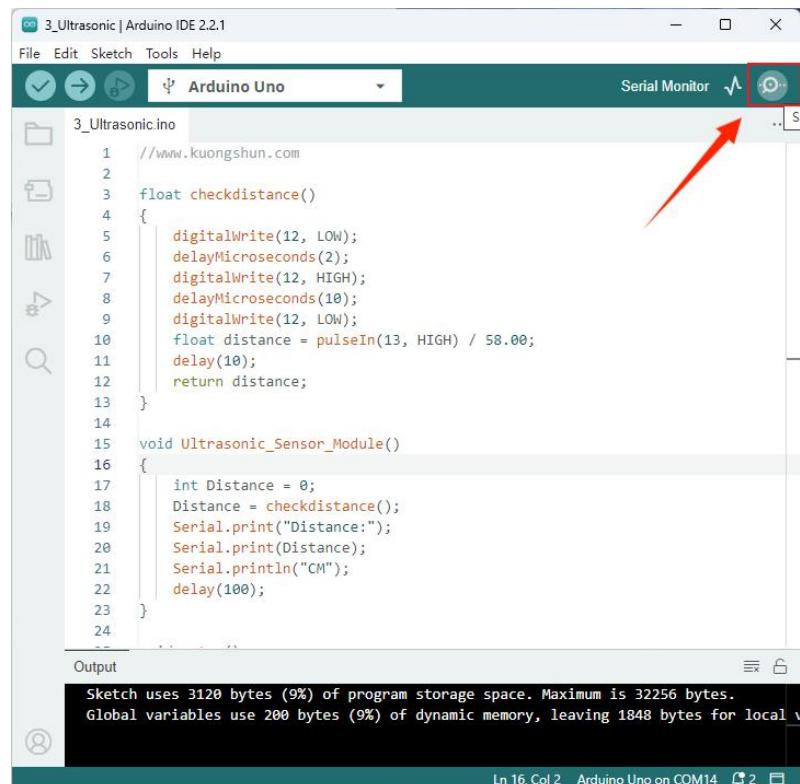
Sketch uses 3120 bytes (9%) of program storage space. Maximum is 32256 bytes.
Global variables use 200 bytes (9%) of dynamic memory, leaving 1848 bytes for

Done uploading.

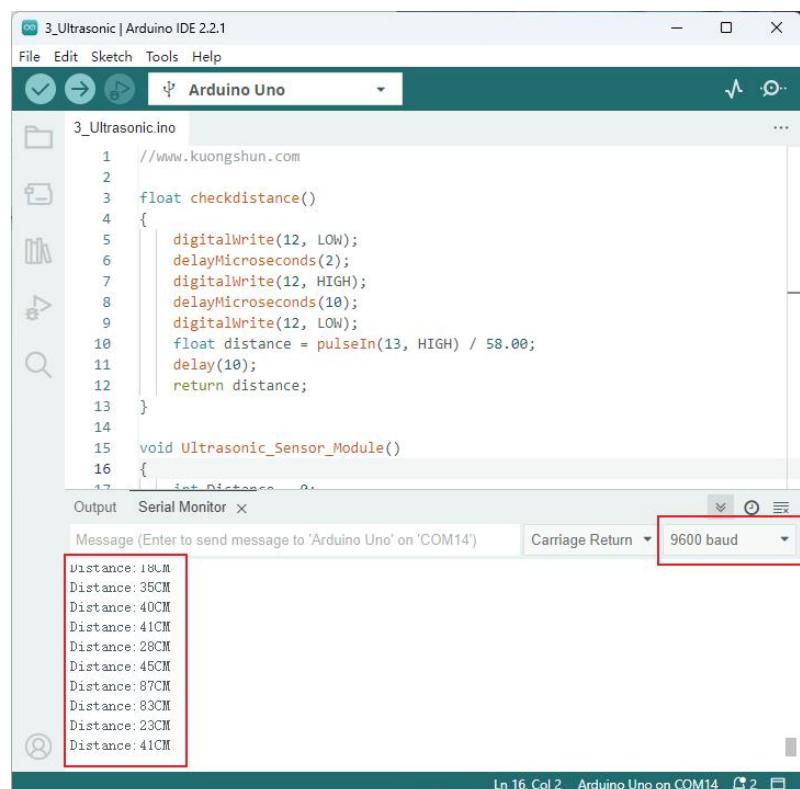
Ln 5, Col 27 Arduino Uno on COM13 2

The picture above shows that it is uploaded successfully.

⑥After uploading the code successfully. Open the serial port monitor and confirm that the baud rate is 9600. You can see the distance information of the ultrasonic wave appearing on the monitor.



The screenshot shows the Arduino IDE interface with the sketch named "3_Ultrasonic.ino". The code defines a function `float checkdistance()` that sends a pulse and measures the time until it's reflected back. It then calculates the distance using the formula $Distance = \frac{time}{2} \times 343$. The main loop calls this function and prints the result to the Serial Monitor. A red arrow points to the "Serial Monitor" tab in the top menu bar.



The screenshot shows the Serial Monitor window with the baud rate set to 9600. The output window displays a series of distance measurements in centimeters:

```
Distance: 18CM
Distance: 35CM
Distance: 40CM
Distance: 41CM
Distance: 28CM
Distance: 45CM
Distance: 87CM
Distance: 83CM
Distance: 23CM
Distance: 41CM
```

Principle Introduction:



Module Features:

Module features: test distance, high precision module.

Applications: Obstacle avoidance, object testing distance, liquid testing, public safety, parking lot testing.

Main technical parameters:

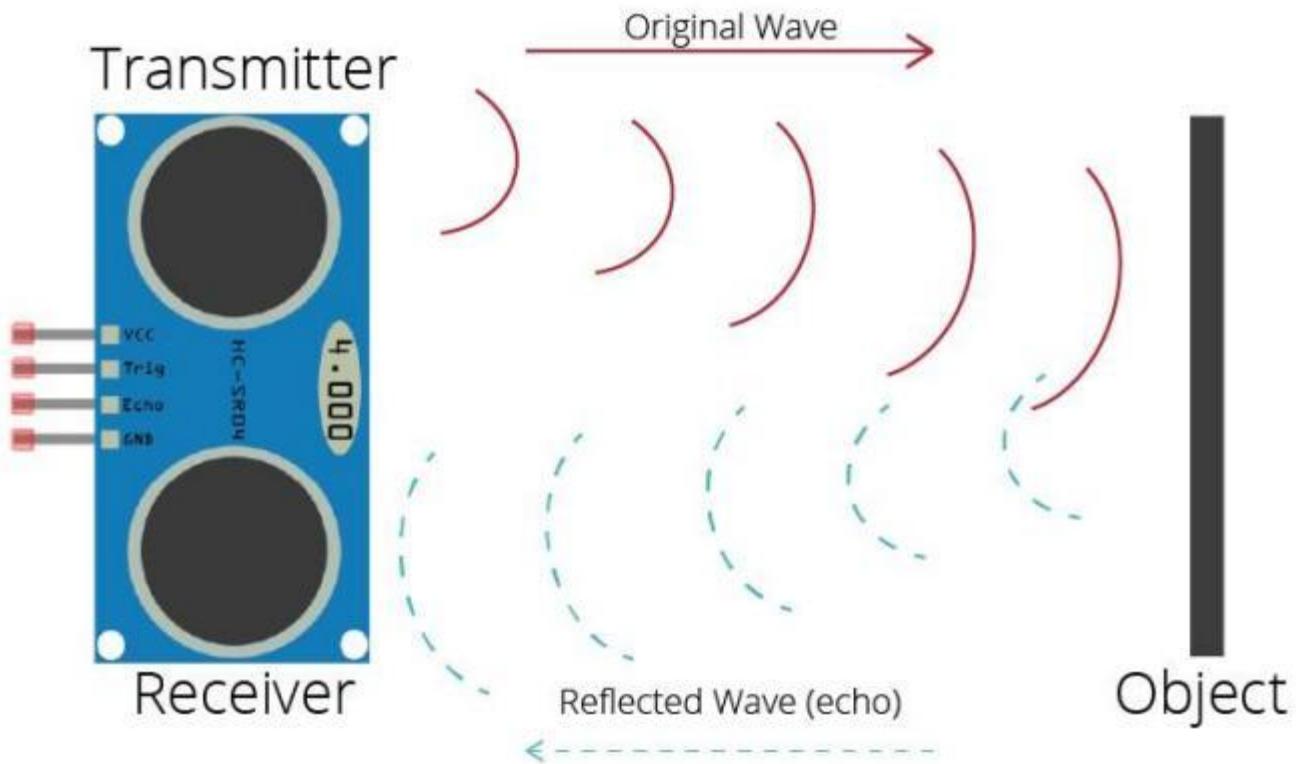
- (1) : Use voltage: DC--5V
- (2) : Static current: less than 2mA
- (3) : Level output: higher than 5V
- (4) : Level output: lower than 0
- (5) : Detection Angle: no more than 15 degrees
- (6) : Detection distance: 2cm-450cm
- (7) : High precision: up to 0.2 cm

Line connection methods: VCC, trig (control end), echo (receiver end), GND

How modules work:

- (1) The TRIG IO port is used to trigger ranging, giving a high level signal, at least 10us a time;
- (2) The module automatically sends 8 square waves of 40kHz to automatically test whether there is a signal returned;

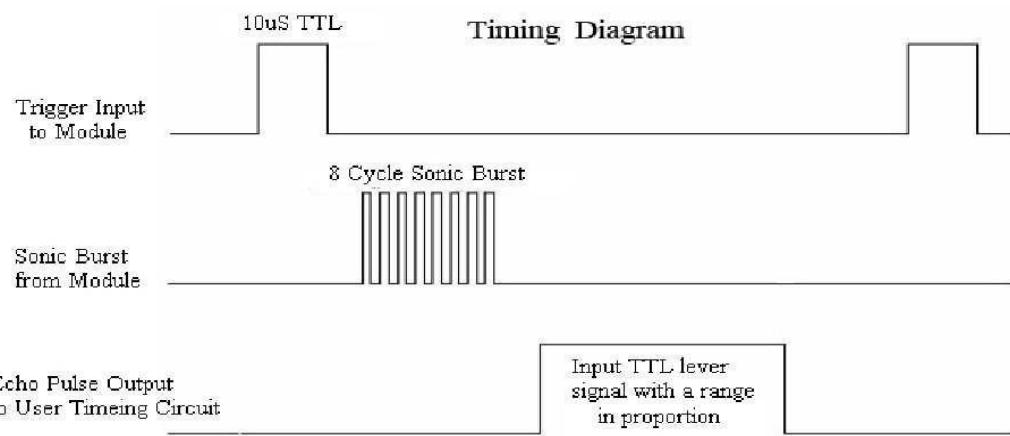
(3) If the signal is received, the module will output a high level pulse through the ECHO's IO port. The duration of the high level pulse is the time between the wave transmission and reception. Therefore, the module can learn the distance based on time.



Test distance = $(\text{high level time} * \text{sound speed (340M/S)})/2;$

In action:

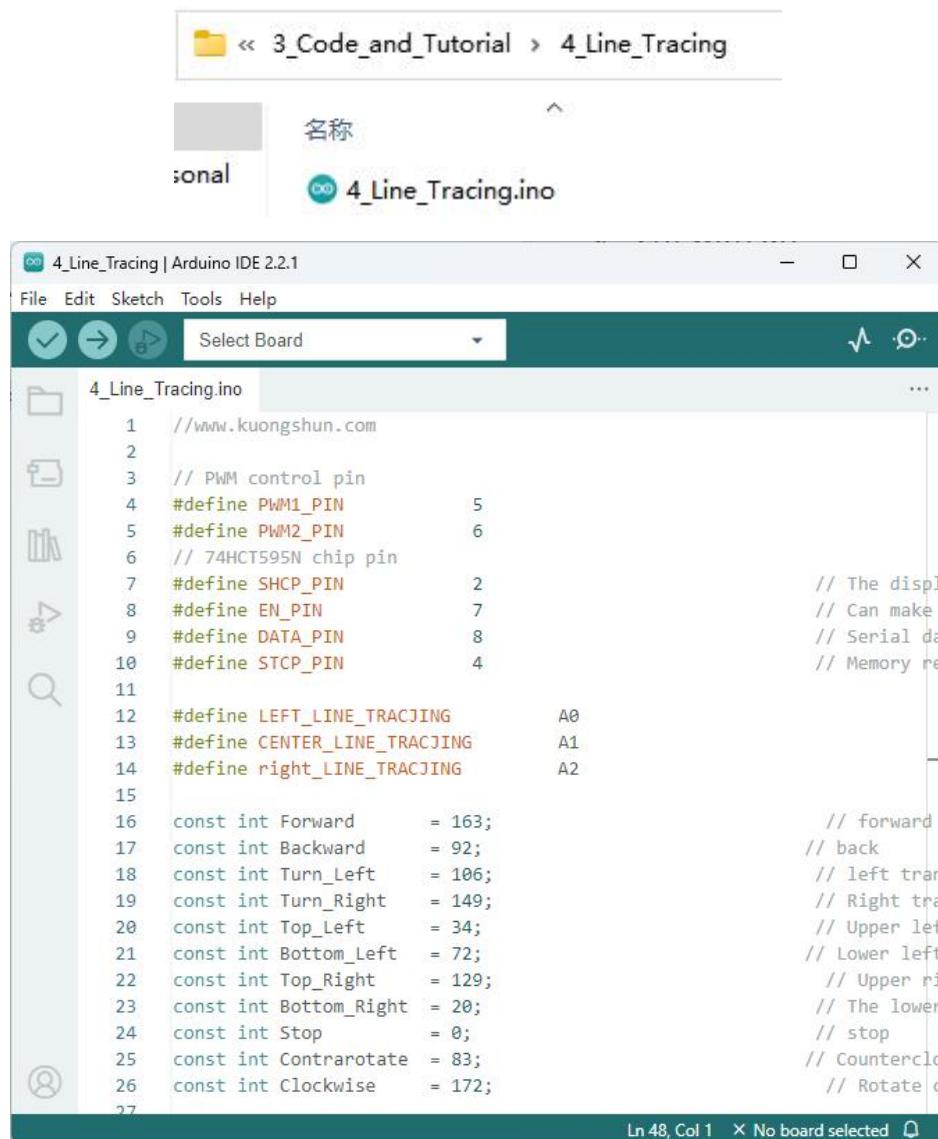
The timing diagram is shown below. Only a short pulse of 10uS is provided to the trigger input to start the ranging, and then the module will emit an ultrasonic pulse train of 8 cycles at 40 kHz and generate an echo. An echo is a range object whose pulse width is proportional to the ranging, which you can calculate by the time interval between sending the trigger signal and receiving the echo signal. Formula: $\mu\text{s}/58 = \text{centimeters}$ or $\mu\text{s}/148 = \text{inches}$; Or: $\text{range} = \text{high level time} * \text{speed (340M/S)} / 2$; We propose to use a measurement period of more than 60ms to prevent the influence of the trigger signal on the echo signal.



Lesson 4 Three Infrared Line Tracing

In this project, we will learn the use of three-way infrared line patrol module and how to make the car complete a complete line patrol. Start by uploading the program to the car.

- ① Start by going to the following path and opening the program:

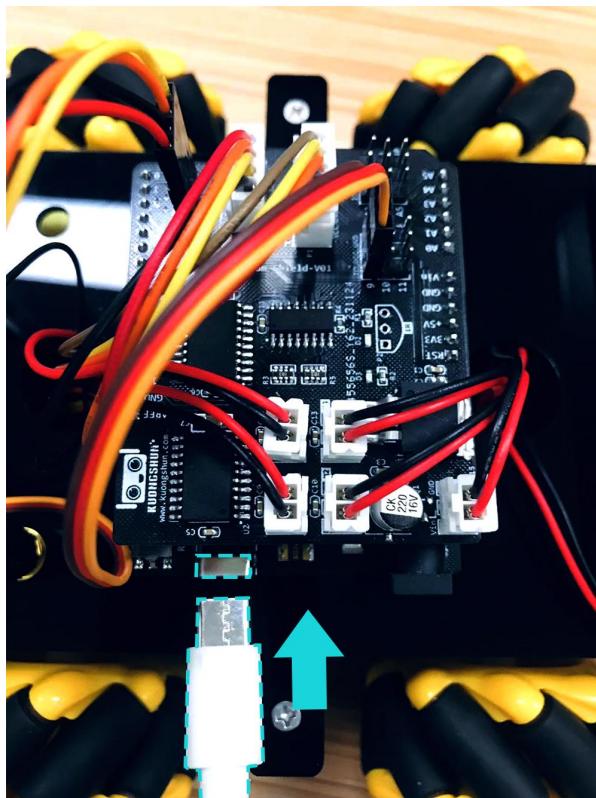


The screenshot shows the Arduino IDE interface. At the top, there's a file browser window titled "3_Code_and_Tutorial > 4_Line_Tracing" containing a single file named "4_Line_Tracing.ino". Below this, the main Arduino IDE window is open, showing the code for "4_Line_Tracing.ino". The code is as follows:

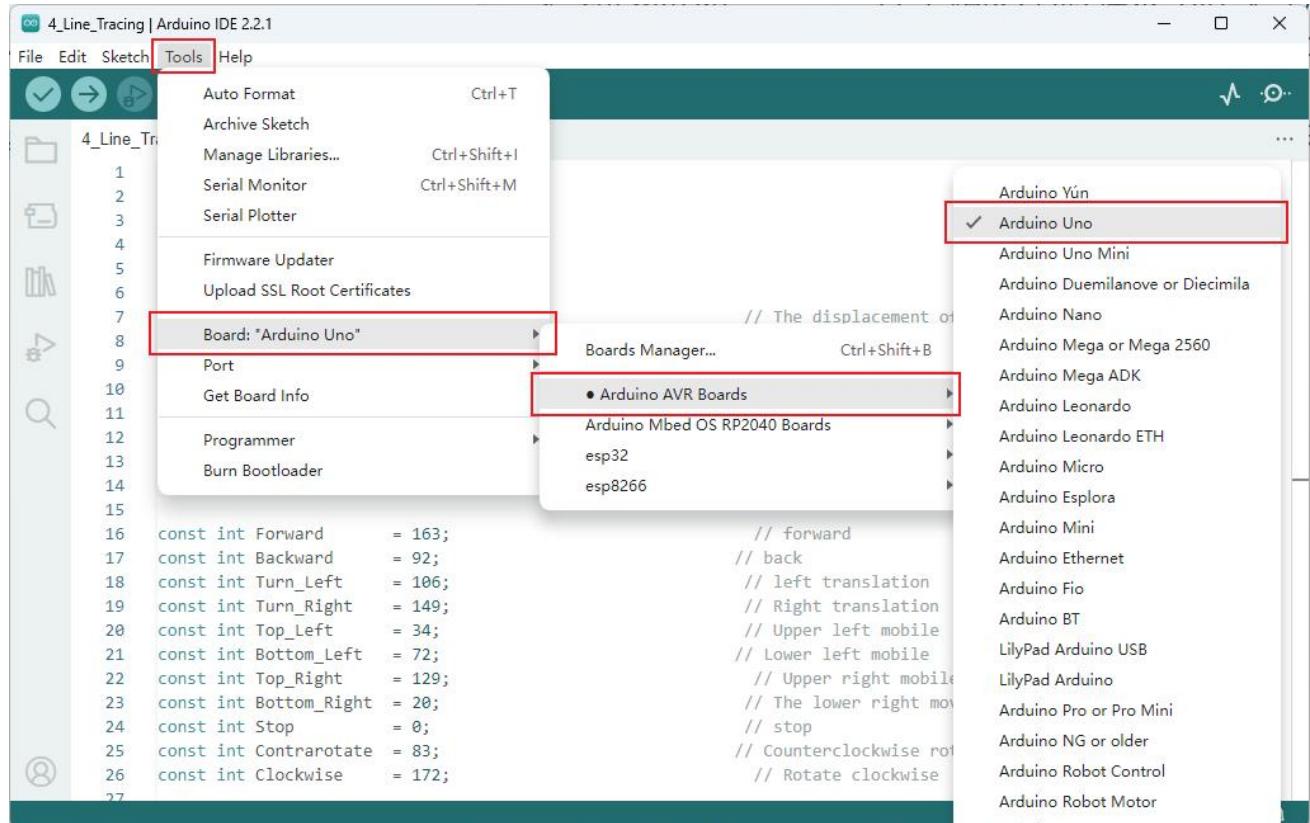
```
1 //www.kuongshun.com
2
3 // PWM control pin
4 #define PWM1_PIN 5
5 #define PWM2_PIN 6
6 // 74HCT595N chip pin
7 #define SHCP_PIN 2
8 #define EN_PIN 7
9 #define DATA_PIN 8
10 #define STCP_PIN 4
11
12 #define LEFT_LINE_TRACING A0
13 #define CENTER_LINE_TRACING A1
14 #define right_LINE_TRACING A2
15
16 const int Forward = 163; // forward
17 const int Backward = 92; // back
18 const int Turn_Left = 106; // left turn
19 const int Turn_Right = 149; // Right tra
20 const int Top_Left = 34; // Upper lef
21 const int Bottom_Left = 72; // Lower left
22 const int Top_Right = 129; // Upper ri
23 const int Bottom_Right = 20; // The lower
24 const int Stop = 0; // stop
25 const int Contrarotate = 83; // Counterclo
26 const int Clockwise = 172; // Rotate c
27
```

At the bottom of the code editor, it says "Ln 48, Col 1" and "No board selected".

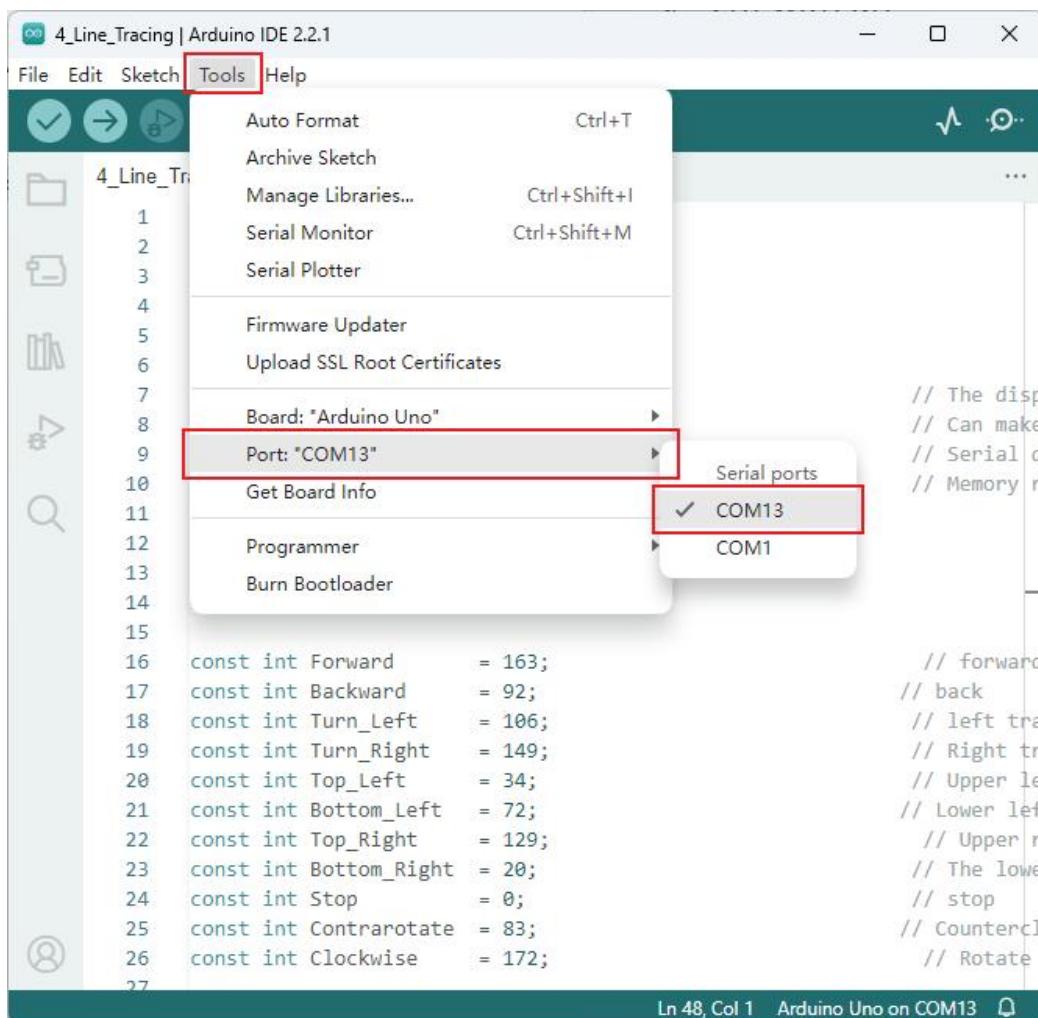
- ② Connect the Arduino board to the computer with TYPE-C data cable



③ Select your control board: Tools > Board >> Arduino AVR Board >>> Arduino UNO

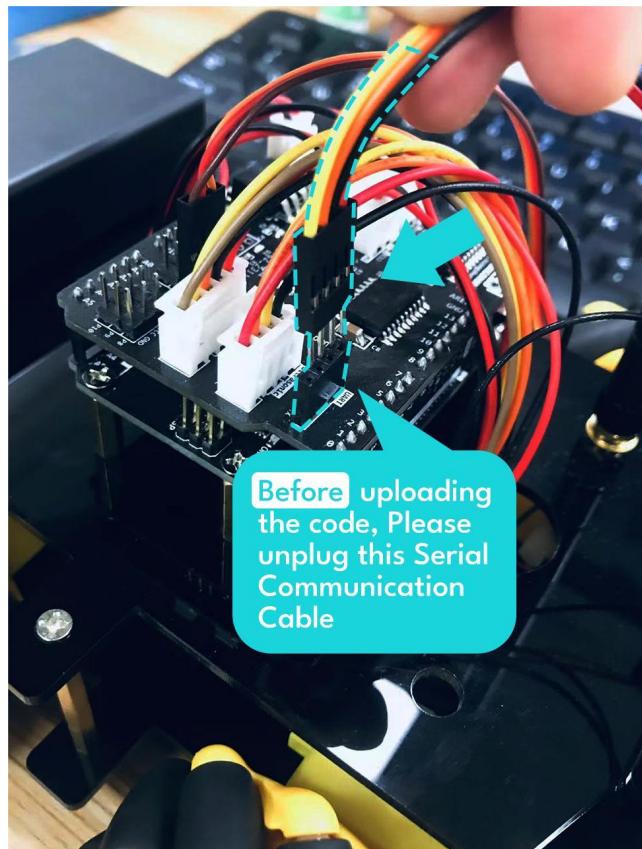


④Select the Port: Tools >Port >>>>COMxx(The COM port number of each computer may be different, and the port number that appears is your port number) (If you do not see COM ports other than COM1 in Arduino IDE, you need to refer to the installation of CH340 driver in "2_Programming_Preparation" and install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

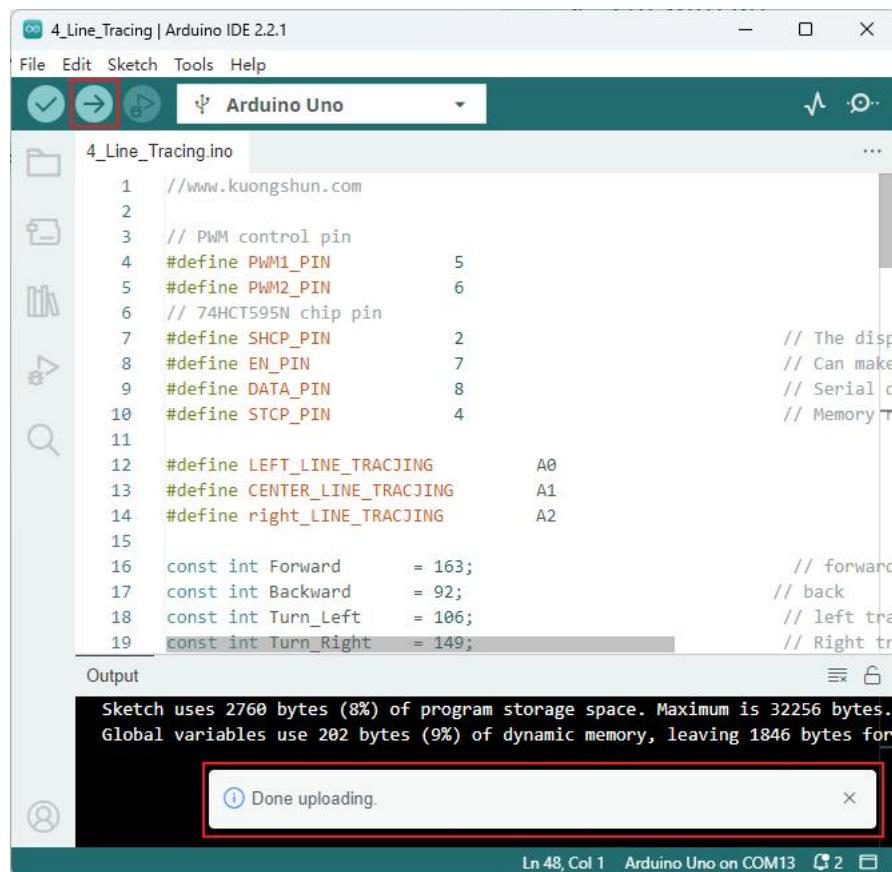


名称	修改日期	类型	大小
CH340 Serial Driver	2023/12/18 17:50	文件夹	
Mac OS Tutorial for Arduino IDE Installation...	2023/12/8 18:26	DOCX 文档	4,563 KB
Windows Tutorial for Arduino IDE Installatio...	2023/12/18 11:16	DOCX 文档	982 KB

⑤Unplug the serial port transmission cable between the Arduino board and the ESP32-CAM board.(This cable will not be connected back during all uploading procedures, until finally all procedures are uploaded and the car is controlled by WIFI.)



⑥Upload the program to the UNO controller board.



```
4_Line_Tracing | Arduino IDE 2.2.1
File Edit Sketch Tools Help
  ↗  ➔  ↘  Arduino Uno
4_Line_Tracing.ino
1 //www.kuongshun.com
2
3 // PWM control pin
4 #define PWM1_PIN      5
5 #define PWM2_PIN      6
6 // 74HCT595N chip pin
7 #define SHCP_PIN       2
8 #define EN_PIN          7
9 #define DATA_PIN        8
10 #define STCP_PIN        4
11
12 #define LEFT_LINE_TRACING   A0
13 #define CENTER_LINE_TRACING A1
14 #define right_LINE_TRACING  A2
15
16 const int Forward     = 163;           // forward
17 const int Backward    = 92;            // back
18 const int Turn_Left   = 106;           // left tra
19 const int Turn_Right  = 149;           // Right tr
Sketch uses 2760 bytes (8%) of program storage space. Maximum is 32256 bytes.
Global variables use 202 bytes (9%) of dynamic memory, leaving 1846 bytes for
Done uploading.
Ln 48, Col 1 Arduino Uno on COM13 2
```

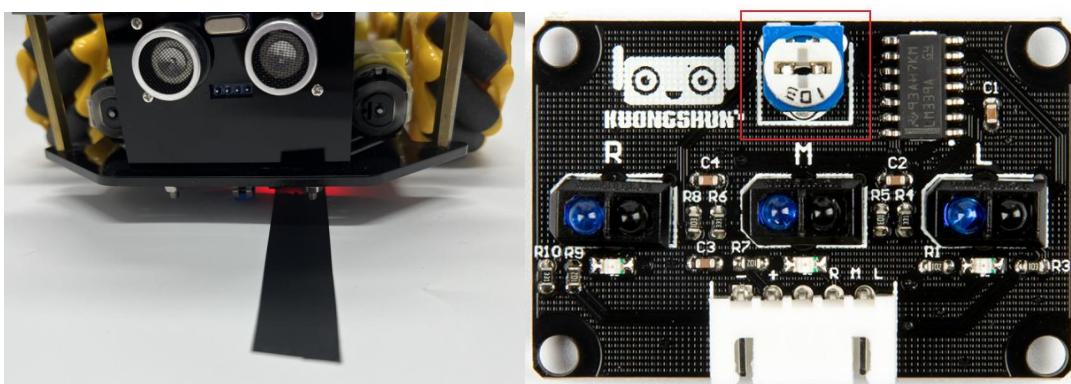
The picture above shows that it is uploaded successfully.

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 two 18650 batteries.) Then turn on the switch. We need to find a light or white surface, put black tape on it, and test whether the car can tell the difference between the black tape and the light background. The steps are as follows:

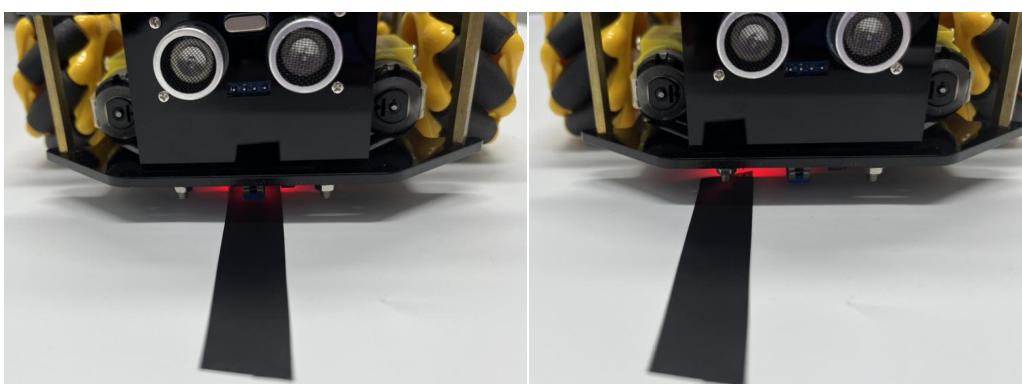
- Attach the next section with black tape on the appropriate plane



- Put the powered trolley on, aim at one of the probes, and you can see the corresponding indicator light lights up separately. If it is not lit, please turn the sliding rheostat knob on the module to turn on the light against the black line and off the light against the white background.



- The same goes for the other two way



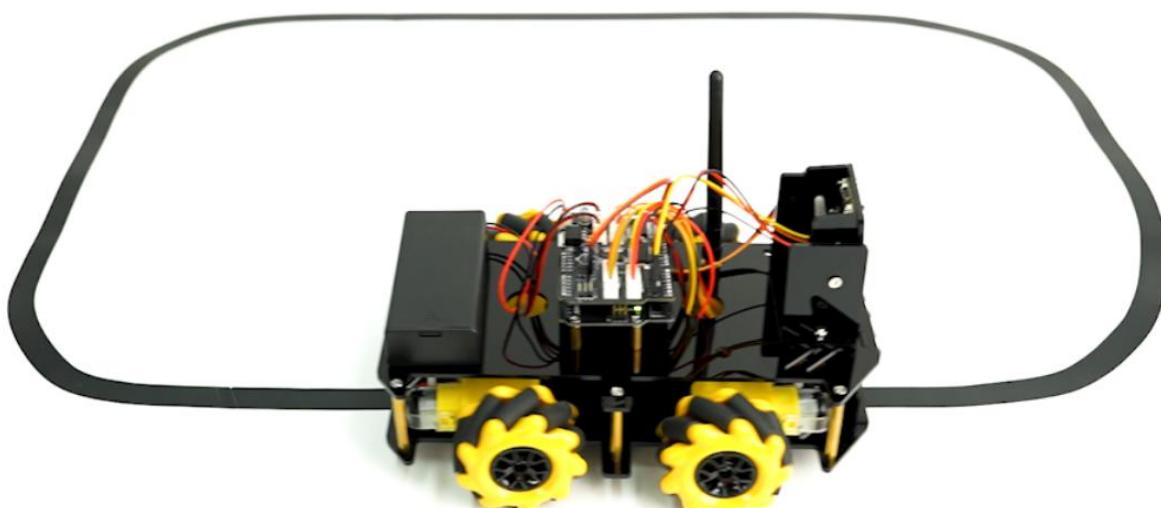
d. All three lights are off against a white background



e. The three lights are on against the black line



If the above steps are correct, you can make a map and start your line patrol



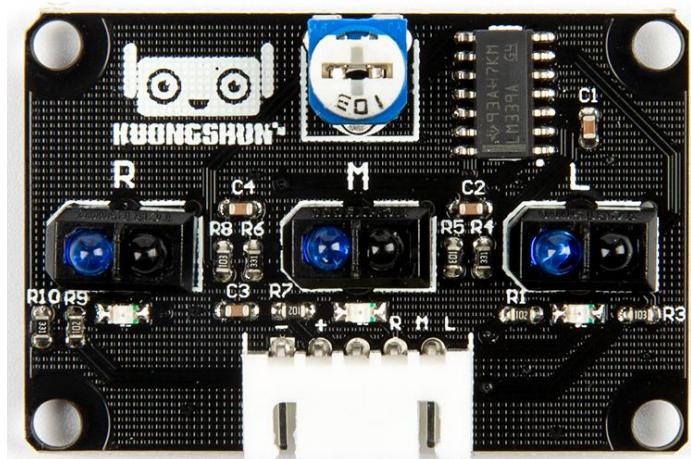
Turn on the switch and the car will walk along the line



The logic and code of the walk are as follows: the left test black to the left, the right test black to the right, only the middle test black when forward

```
Left_Tra_Value = analogRead(LEFT_LINE_TRACING);
Center_Tra_Value = analogRead(CENTER_LINE_TRACING);
Right_Tra_Value = analogRead(right_LINE_TRACING);
if (Left_Tra_Value > Black_Line && Center_Tra_Value <= Black_Line && Right_Tra_Value > Black_Line)
{
| Motor(Forward, 180);
}
else if (Left_Tra_Value < Black_Line && Center_Tra_Value < Black_Line && Right_Tra_Value < Black_Line)
{
| Motor(Forward, 130);
}
else if (Left_Tra_Value >= Black_Line && Center_Tra_Value >= Black_Line && Right_Tra_Value < Black_Line)
{
| Motor(Clockwise, 220);
}
else if (Left_Tra_Value >= Black_Line && Center_Tra_Value < Black_Line && Right_Tra_Value < Black_Line)
{
| Motor(Clockwise, 160);
}
else if (Left_Tra_Value < Black_Line && Center_Tra_Value < Black_Line && Right_Tra_Value >= Black_Line)
{
| Motor(Contrarotate, 160);
}
else if (Left_Tra_Value < Black_Line && Center_Tra_Value >= Black_Line && Right_Tra_Value >= Black_Line)
{
| Motor(Contrarotate, 220);
}
else if (Left_Tra_Value >= Black_Line && Center_Tra_Value >= Black_Line && Right_Tra_Value >= Black_Line)
{
| Motor(Forward, 130);
}
```

Three Infrared Line Tracing



The inspection sensor is located under the front direction of the bottom plate of the car. Each route of the patrol sensor consists of an infrared transmitter tube and an infrared receiver tube. The former is a light-emitting diode that can emit infrared light, and the latter is a photoresistor that is only responsible for receiving infrared light. Black surfaces have different light reflectance than white surfaces. Therefore, the intensity of the infrared reflected light received by the car on the black road surface is different from that on the white road surface, and the resistance value changes accordingly. According to the principle of series resistance voltage division, the color of the road under the car can be inferred from the voltage of the sensor, so as to determine the motion path.

tips

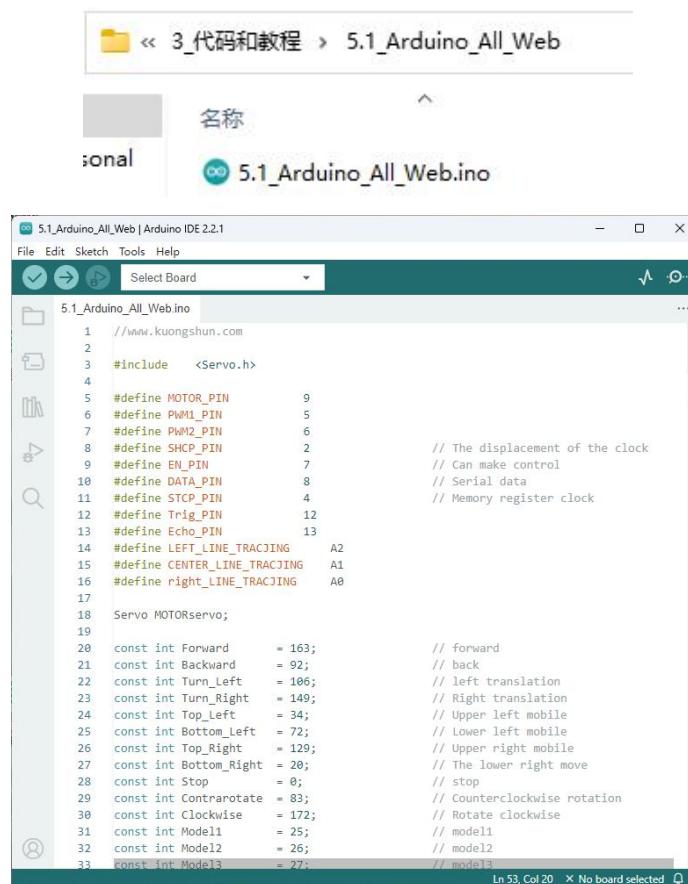
- (1) The bending part of the line should be as smooth as possible. If the turning radius is too small, the car is likely to overshoot the runway.
- (2) Straight line tracking scenes can be made with black and white tape or any color paper to distinguish them from paths.
- (3) In addition to line tracking, we can use our imagination to develop other programs, such as programs that confine a car to a certain area regardless of its motion.

Lesson 5 WIFI Camera Multi-Function Program for Web

In this project, we will use Arduino board combined with L293D extension board to realize the functions of walking, obstacle avoidance, line inspection and route of the car. The ESP32-CAM is used to realize WiFi connection and control the car through the serial port line. At the same time, ESP32-CAM also transmits the image of the front camera to the hotspot server in real time. The mobile phone input the hotspot can view the camera image of the car and control the walking of the car, as well as adjust the operation mode and the speed of the car.

Before presenting the multi-function program of the cart, we first upload the program to the cart. The car has two control boards, and we need to upload two programs to each board. Here are the steps to upload an Arduino board and program:

- ① Start by going to the following path and opening the program:



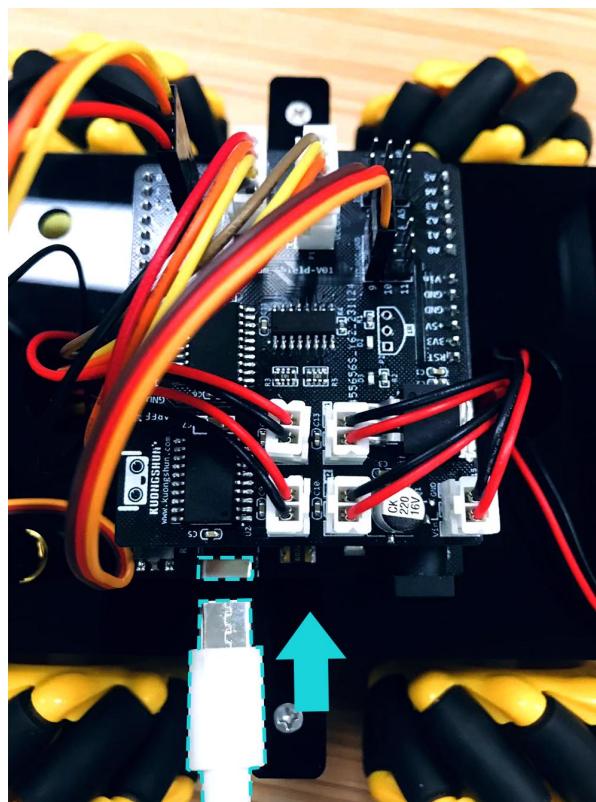
The screenshot shows the Arduino IDE interface with the following details:

- File Path:** 3_代码和教程 > 5.1_Arduino_All_Web
- File Name:** 5.1_Arduino_All_Web.ino
- Code Preview:**

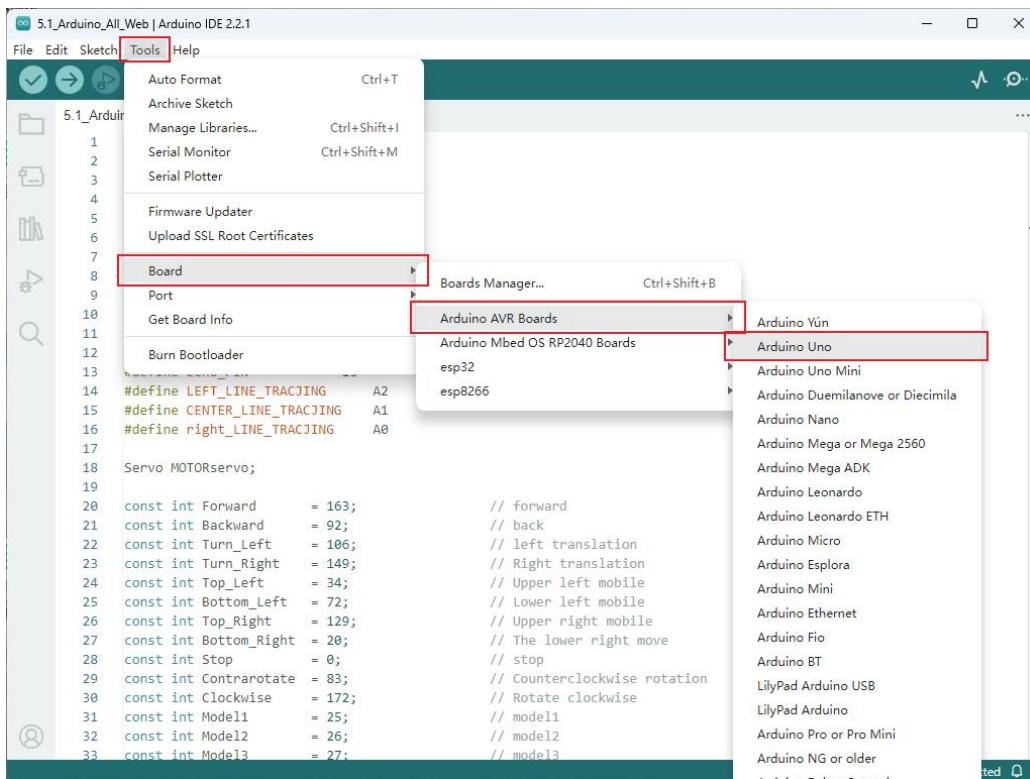
```

1 //www.kuongshun.com
2
3 #include <Servo.h>
4
5 #define MOTOR_PIN 9
6 #define PWM1_PIN 5
7 #define PWM2_PIN 6
8 #define SHCP_PIN 2
9 #define EN_PIN 7
10 #define DATA_PIN 8
11 #define STCP_PIN 4
12 #define Trig_PIN 12
13 #define Echo_PIN 13
14 #define LEFT_LINE_TRACING A2
15 #define CENTER_LINE_TRACING A1
16 #define right_LINE_TRACING A0
17
18 Servo MOTOrservo;
19
20 const int Forward = 163; // forward
21 const int Backward = 92; // back
22 const int Turn_Left = 186; // left translation
23 const int Turn_Right = 149; // Right translation
24 const int Top_Left = 34; // Upper left mobile
25 const int Bottom_Left = 72; // Lower left mobile
26 const int Top_Right = 129; // Upper right mobile
27 const int Bottom_Right = 20; // The lower right move
28 const int Stop = 8; // stop
29 const int Contrarotate = 83; // Counterclockwise rotation
30 const int Clockwise = 172; // Rotate clockwise
31 const int Model1 = 25; // model1
32 const int Model2 = 26; // model2
33 const int Model3 = 27; // model3
    
```
- IDE Status Bar:** Ln 53, Col 20 × No board selected

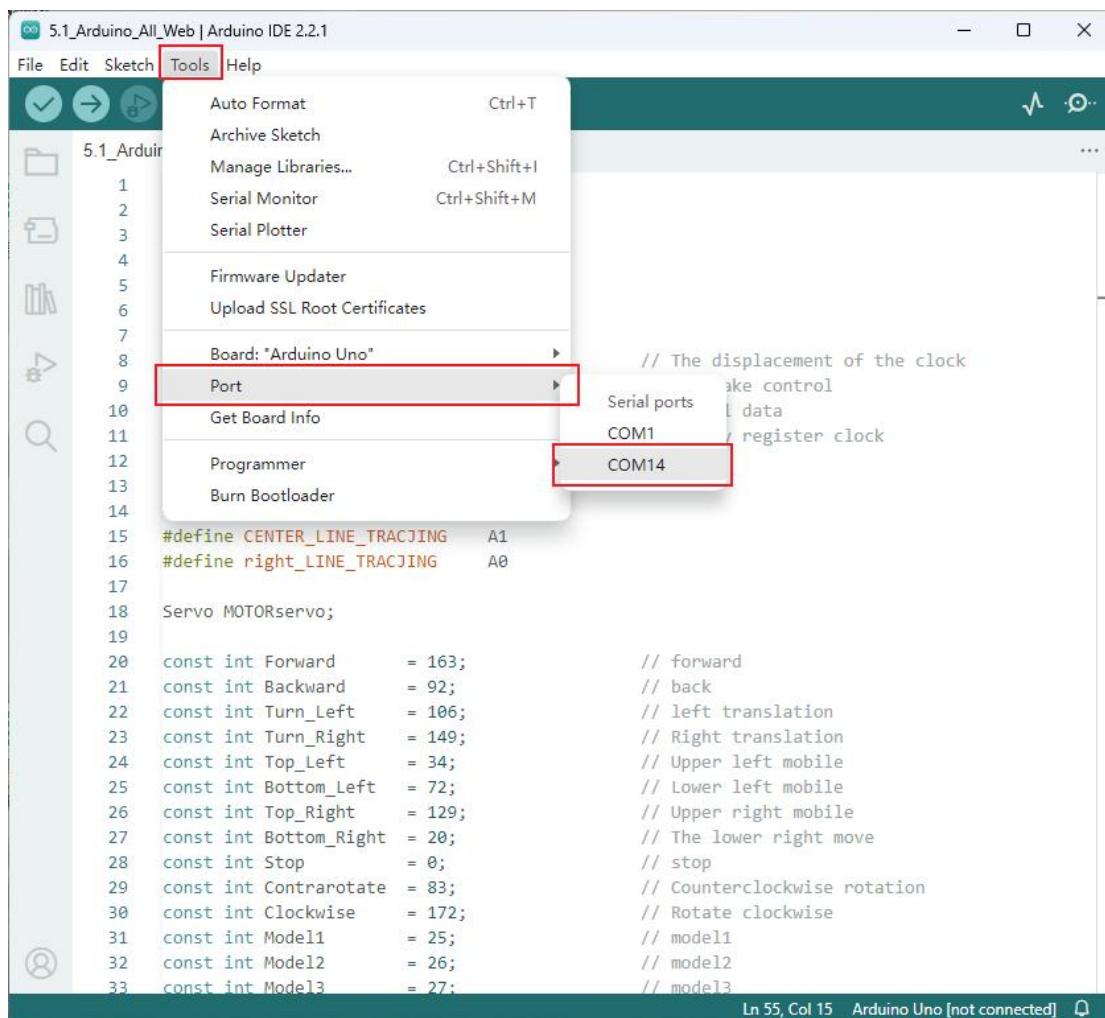
② Connect the Arduino board to the computer with TYPE-C data cable



③ Select your control board: Tools > Board >> Arduino AVR Board >>> Arduino UNO

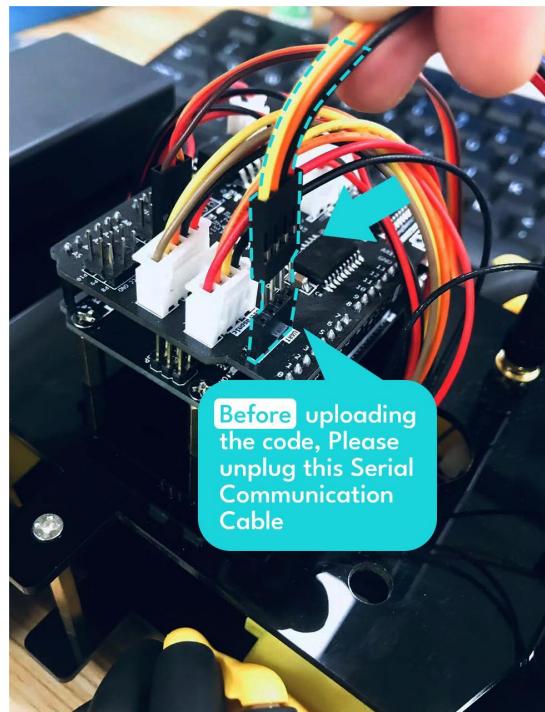


④Select the Port: Tools >Port >>>>COMxx(The COM port number of each computer may be different, and the port number that appears is your port number) (If you do not see COM ports other than COM1 in Arduino IDE, you need to refer to the installation of CH340 driver in "2_Programming_Preparation" and install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

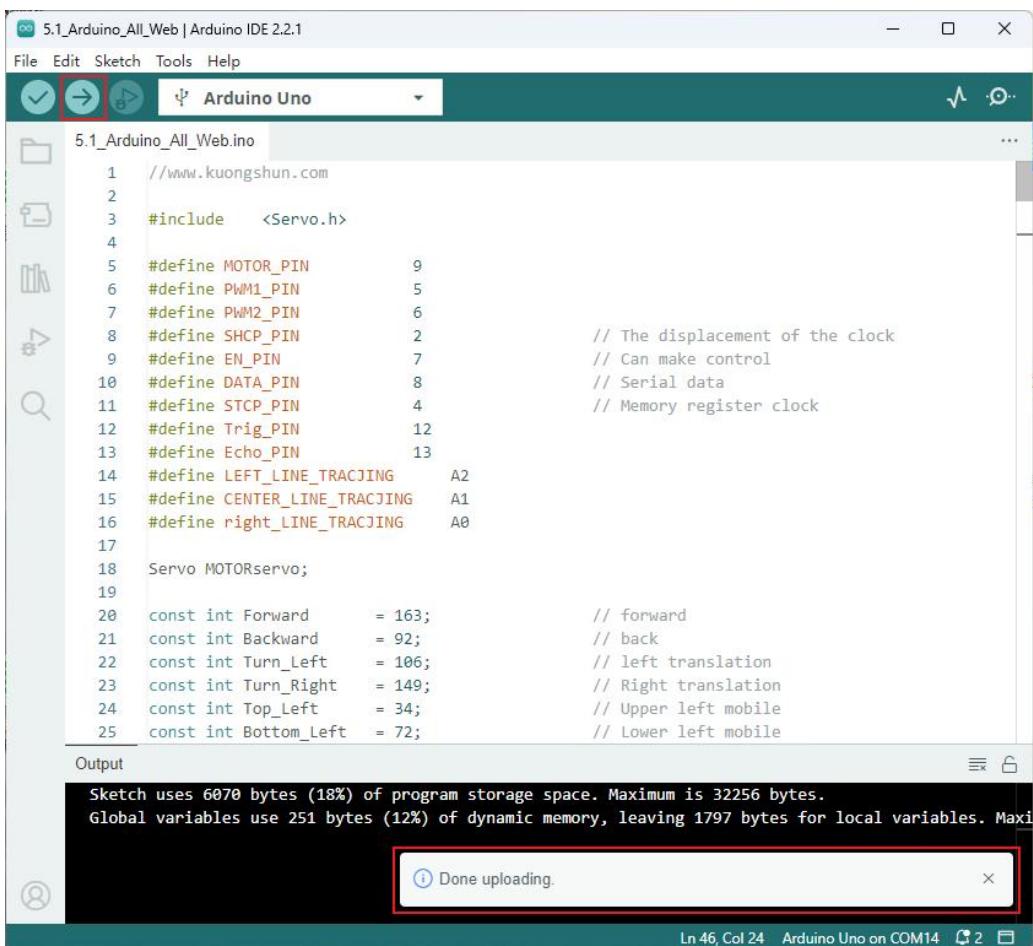


名称	修改日期	类型	大小
CH340 Serial Driver	2023/12/18 17:50	文件夹	
Mac OS Tutorial for Arduino IDE Installation...	2023/12/8 18:26	DOCX 文档	4,563 KB
Windows Tutorial for Arduino IDE Installatio...	2023/12/18 11:16	DOCX 文档	982 KB

⑤Unplug the serial port transmission cable between the Arduino board and the ESP32-CAM board.(This cable will not be connected back during all uploading procedures, until finally all procedures are uploaded and the car is controlled by WIFI.)



⑥Upload the program to the UNO controller board.



```
//www.kuongshun.com
#include <Servo.h>
#define MOTOR_PIN 9
#define PWM1_PIN 5
#define PWM2_PIN 6
#define SHCP_PIN 2           // The displacement of the clock
#define EN_PIN 7             // Can make control
#define DATA_PIN 8           // Serial data
#define STCP_PIN 4           // Memory register clock
#define Trig_PIN 12
#define Echo_PIN 13
#define LEFT_LINE_TRACING A2
#define CENTER_LINE_TRACING A1
#define right_LINE_TRACING A0
Servo MOTORservo;
const int Forward = 163;           // forward
const int Backward = 92;           // back
const int Turn_Left = 106;          // left translation
const int Turn_Right = 149;         // Right translation
const int Top_Left = 34;            // Upper left mobile
const int Bottom_Left = 72;          // Lower left mobile
```

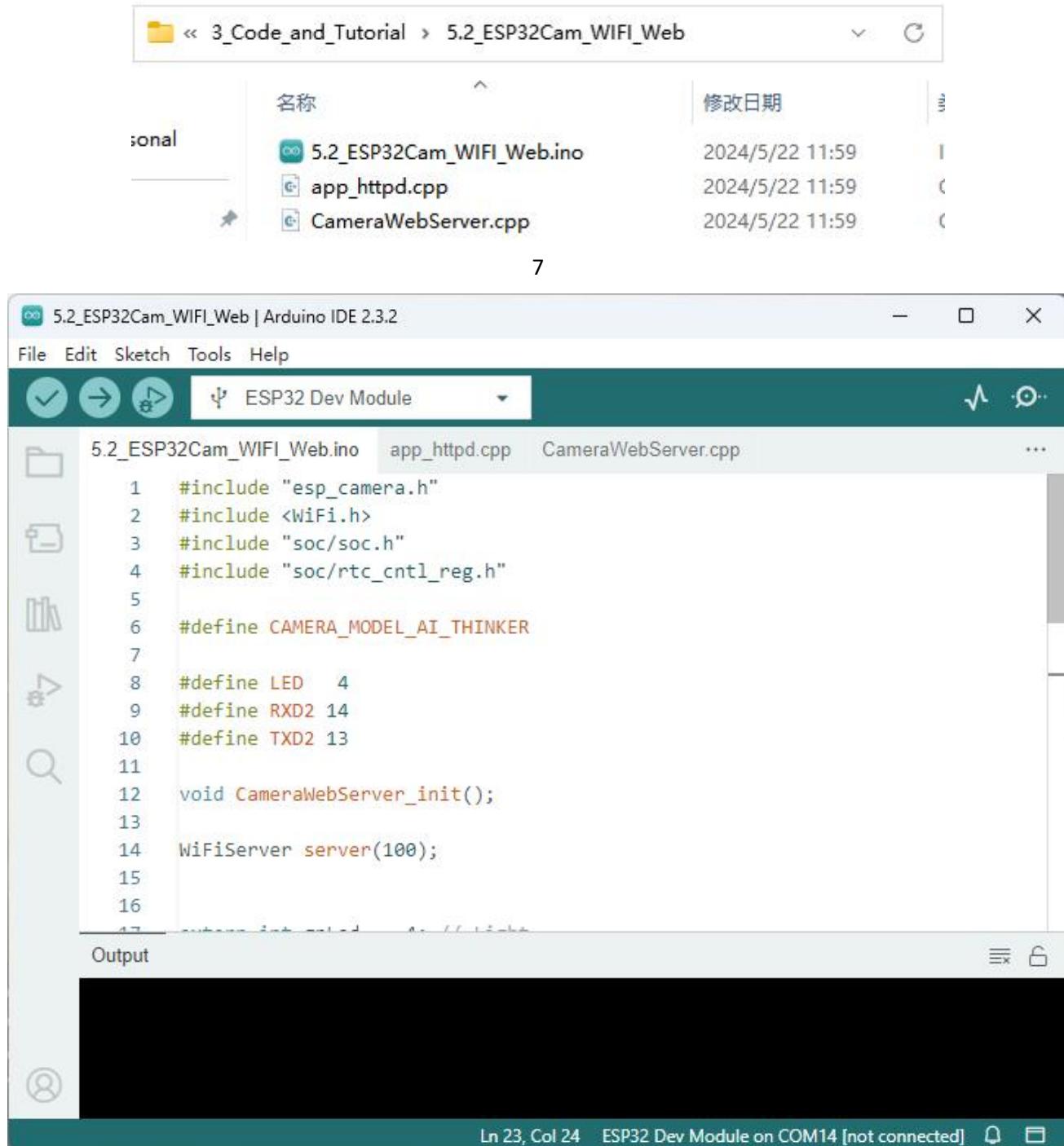
Sketch uses 6070 bytes (18%) of program storage space. Maximum is 32256 bytes.
Global variables use 251 bytes (12%) of dynamic memory, leaving 1797 bytes for local variables. Maxi

Done uploading.

Ln 46, Col 24 Arduino Uno on COM14 2

The picture above shows that it is uploaded successfully.

⑦ After uploading the code successfully. We also need to upload the program to the **ESP32-CAM board** of the locomotive. Steps and code needed to open the following path:



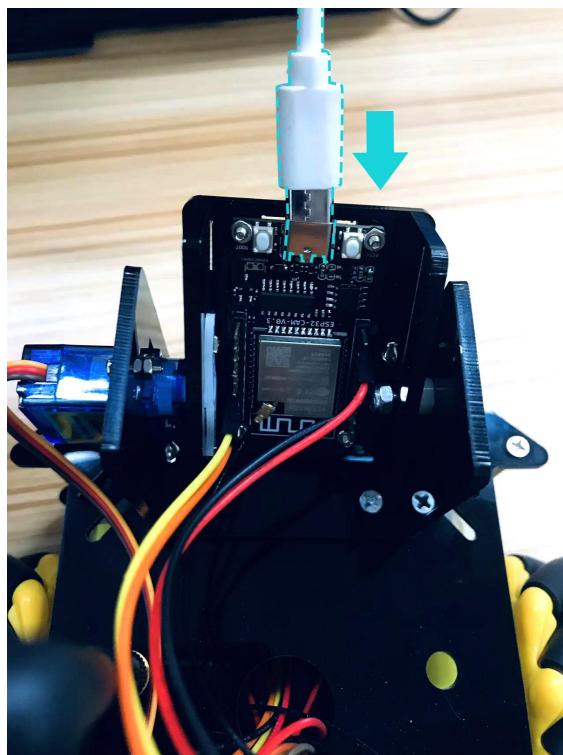
The screenshot shows the Arduino IDE interface with the following details:

- File Explorer:** Shows the project structure: 3_Code_and_Tutorial > 5.2_ESP32Cam_WIFI_Web. The folder contains three files: 5.2_ESP32Cam_WIFI_Web.ino, app_httpd.cpp, and CameraWebServer.cpp.
- Code Editor:** Displays the content of 5.2_ESP32Cam_WIFI_Web.ino. The code includes definitions for camera model, LED pin, RXD2 pin, TXD2 pin, and initializes a WiFi server.

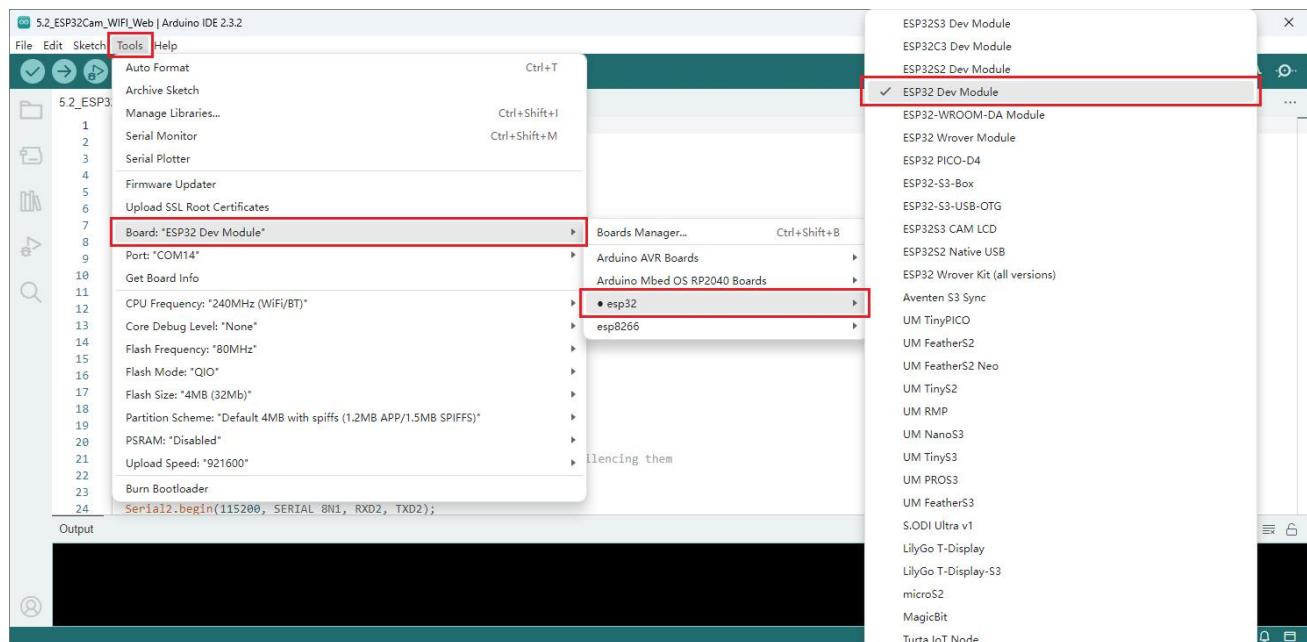
```
1 #include "esp_camera.h"
2 #include <WiFi.h>
3 #include "soc/soc.h"
4 #include "soc/rtc_cntl_reg.h"
5
6 #define CAMERA_MODEL_AI_THINKER
7
8 #define LED 4
9 #define RXD2 14
10 #define TXD2 13
11
12 void CameraWebServer_init();
13
14 WiFiServer server(100);
15
16
17
```

- Status Bar:** Shows the message "Ln 23, Col 24 ESP32 Dev Module on COM14 [not connected]".

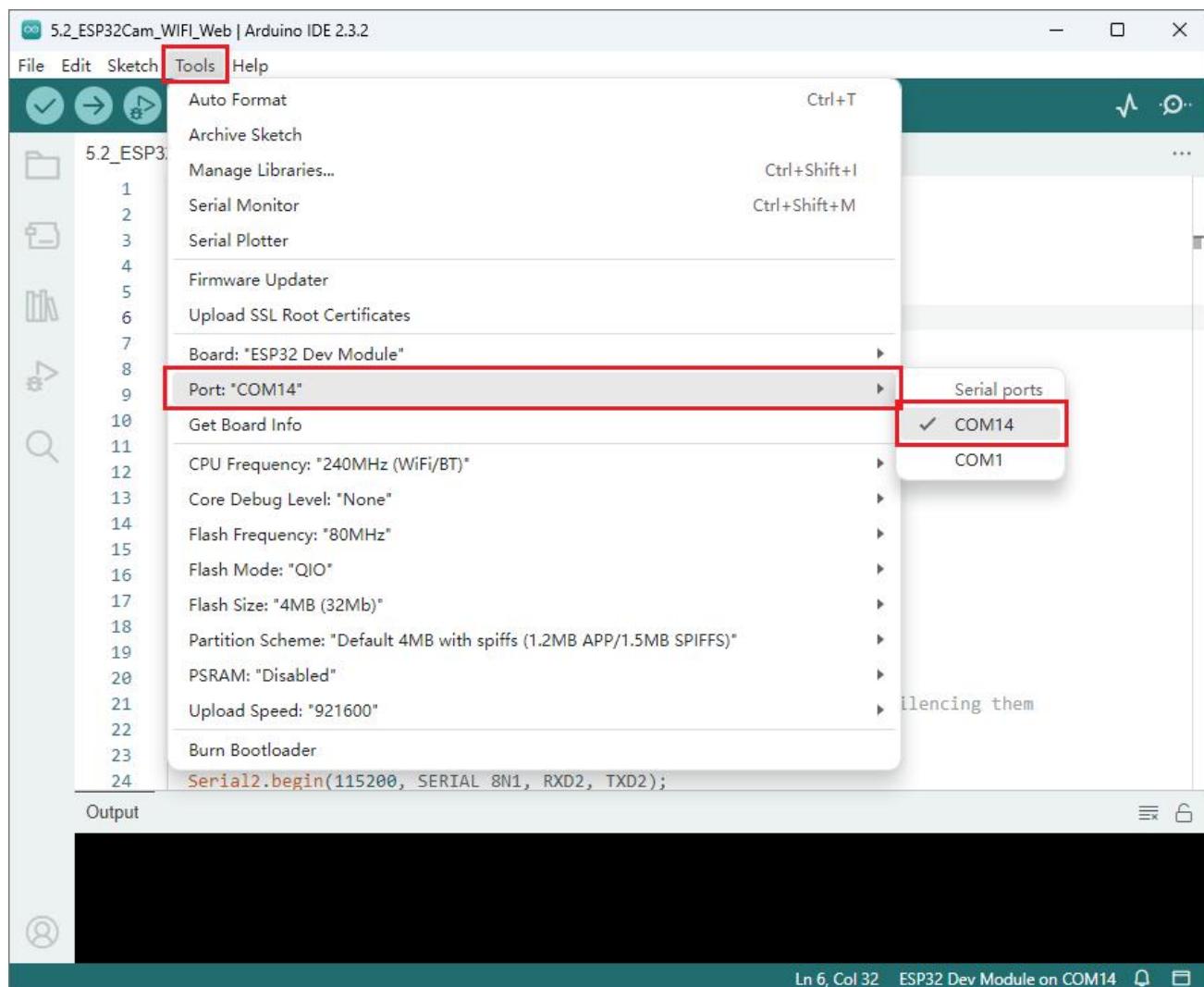
⑧ Connect the **ESP32-CAM board** to the computer with TYPE-C cable



⑨ Select your board: Tools > Board >>esp32>>>ESP32 Dev Module

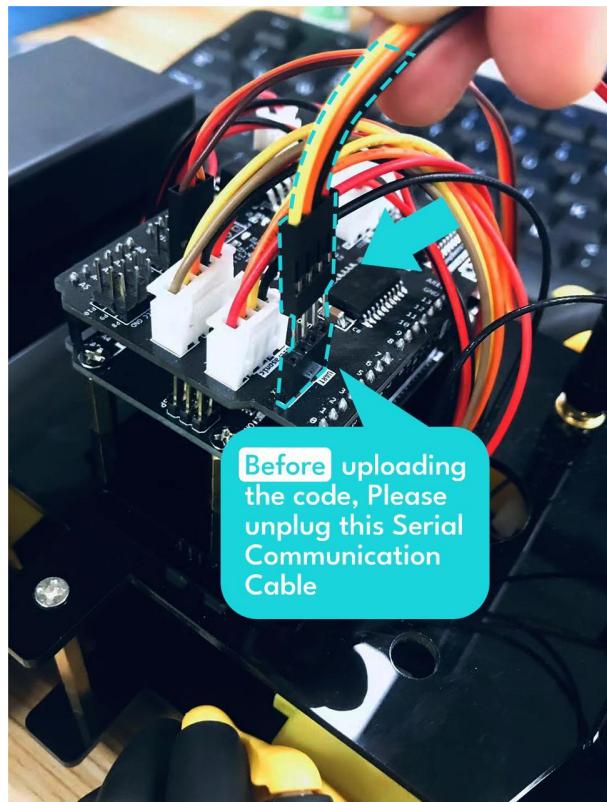


⑩Select the Port: Tools >Port >>>COMxx(The COM port number of each computer may be different, and the port number that appears is your port number) (If you do not see COM ports other than COM1 in Arduino IDE, you need to refer to the installation of CH340 driver in "2_Programming_Preparation" and install it)

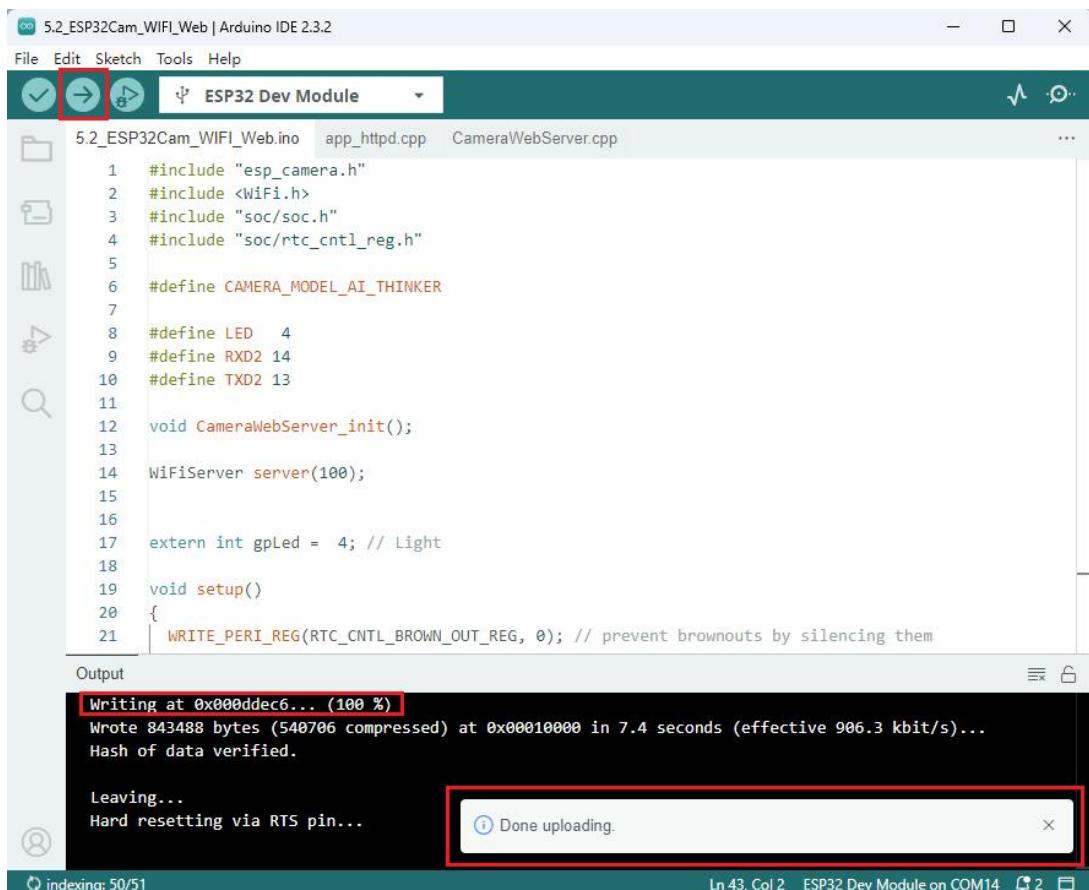


2_Programming_Preparation > 1. Installation of Arduino IDE and CH340 driver >			
名称	修改日期	类型	大小
CH340 Serial Driver	2023/12/18 17:50	文件夹	
Mac OS Tutorial for Arduino IDE Installation...	2023/12/8 18:26	DOCX 文档	4,563 KB
Windows Tutorial for Arduino IDE Installatio...	2023/12/18 11:16	DOCX 文档	982 KB

⑪ unplug the serial port transmission cable between the Arduino control board and the ESP32-CAM control board. (This line can be returned after the upload of this program)



⑫Upload the program to the ESP32-CAM control board.



5.2_ESP32Cam_WIFI_Web | Arduino IDE 2.3.2

File Edit Sketch Tools Help

ESP32 Dev Module

5.2_ESP32Cam_WIFI_Web.ino app_httpd.cpp CameraWebServer.cpp

```
1 #include "esp_camera.h"
2 #include <WiFi.h>
3 #include "soc/soc.h"
4 #include "soc/rtc_cntl_reg.h"
5
6 #define CAMERA_MODEL_AI_THINKER
7
8 #define LED 4
9 #define RXD2 14
10 #define TXD2 13
11
12 void CameraWebServer_init();
13
14 WiFiServer server(100);
15
16
17 extern int gpLed = 4; // Light
18
19 void setup()
20 {
21     WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0); // prevent brownouts by silencing them
```

Output

```
Writing at 0x000ddec6... (100 %)
Wrote 843488 bytes (540706 compressed) at 0x00010000 in 7.4 seconds (effective 906.3 kbit/s)...
Hash of data verified.

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

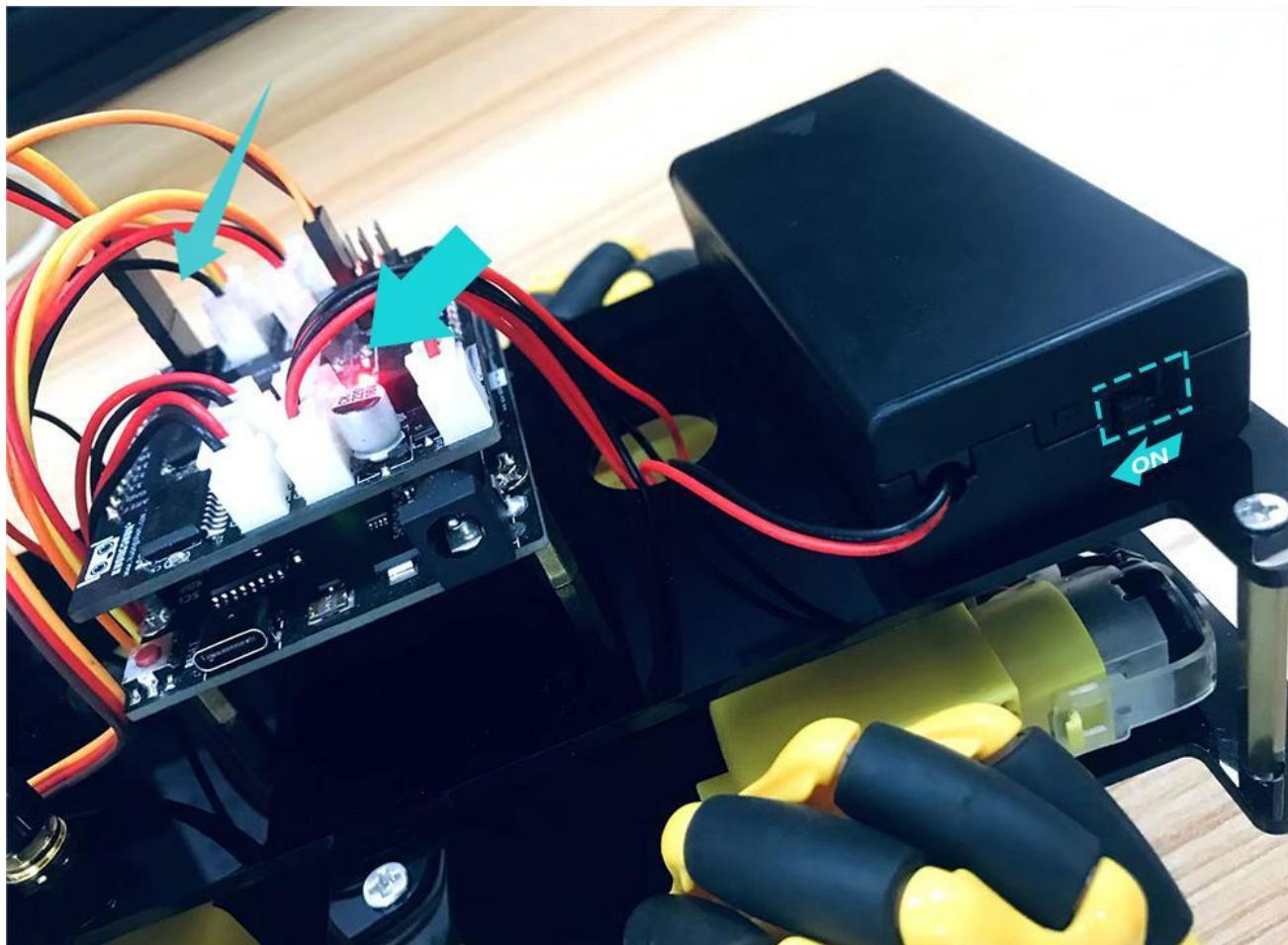
Done uploading.

indexing: 50/51

Ln 43, Col 2 ESP32 Dev Module on COM14 2 2

The picture above shows that it is uploaded successfully.

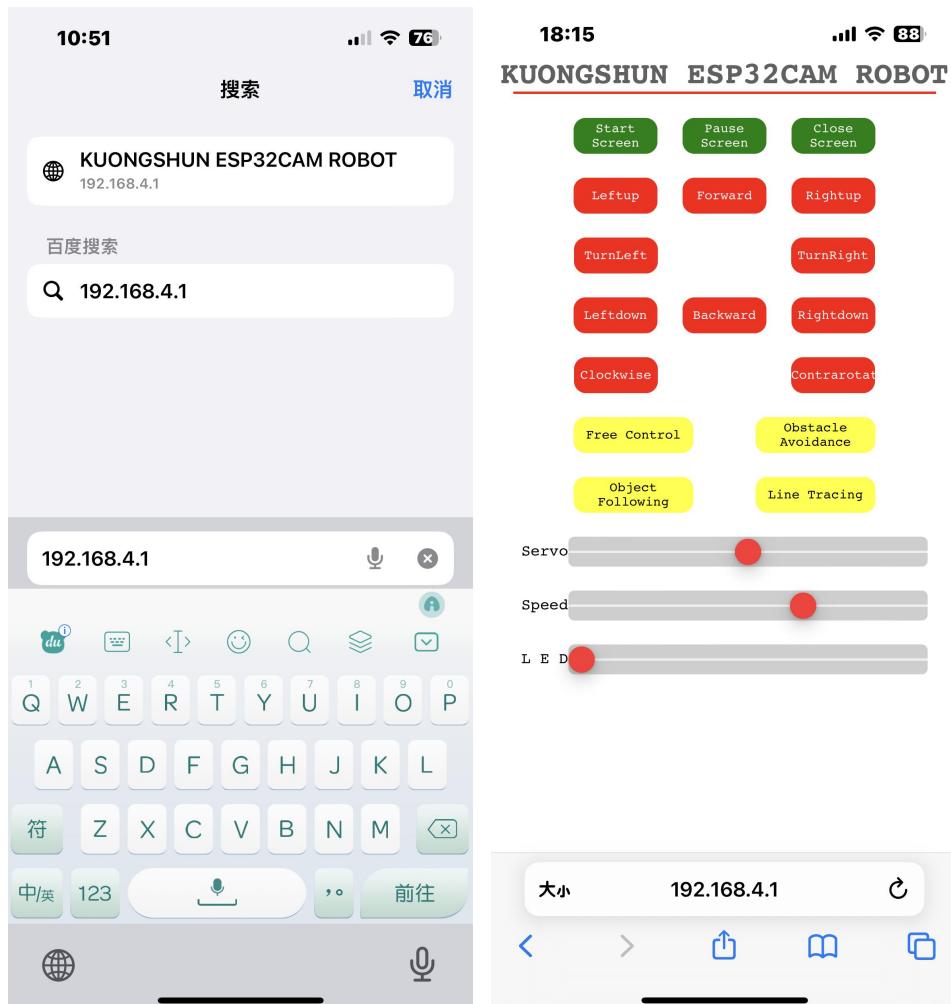
⑬ After uploading the code successfully. Disconnect the data line, put the car on the ground, and load 2 18650 batteries. (If you buy the battery-free version, please prepare two 18650 batteries.) Then turn on the switch.



⑭ Then pick up your phone, turn on the WiFi switch, find "KUONGSHUN", enter the password "12345678" and connect to the hotspot. If you are reminded that there is no network, please keep connected. (If the switch is turned on and the module lights up but the phone cannot search for "KUONGSHUN", please check whether the wiring is correct and whether the battery is powered.)



⑯ Open the mobile browser, input 192.168.4.1, jump to the following screen



⑯ Interface Introduction

Start Screen

: Enable image receiving and display. When pressed, the image will



appear

Pause Screen

: Screenshot and display

Close Screen

: Disable image reception

Forward

: Hold to control the cart to move forward

Backward

: Hold to control the cart to translate backwards

TurnLeft

: Hold to control the car to pan to the left

TurnRight

: Hold to control the car to pan to the right

Leftup

: Hold to control the car to pan left and up

Rightup

: Hold to control the car to pan up right

Leftdown

: Hold to control the car to pan left and down

Rightdown

: Hold to control the car to pan down right

Free Control

: Enter Free Control mode

**Obstacle
Avoidance**

: Enter Obstacle Avoidance mode

**Object
Following**

: Enter Object Following mode

Line Tracing

: Enter Line Tracing mode

Servo

:

Adjust the servo pitch angle of the front of the car

Speed

:

Adjust the speed of the car

L E D

:

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 start in step ⑯

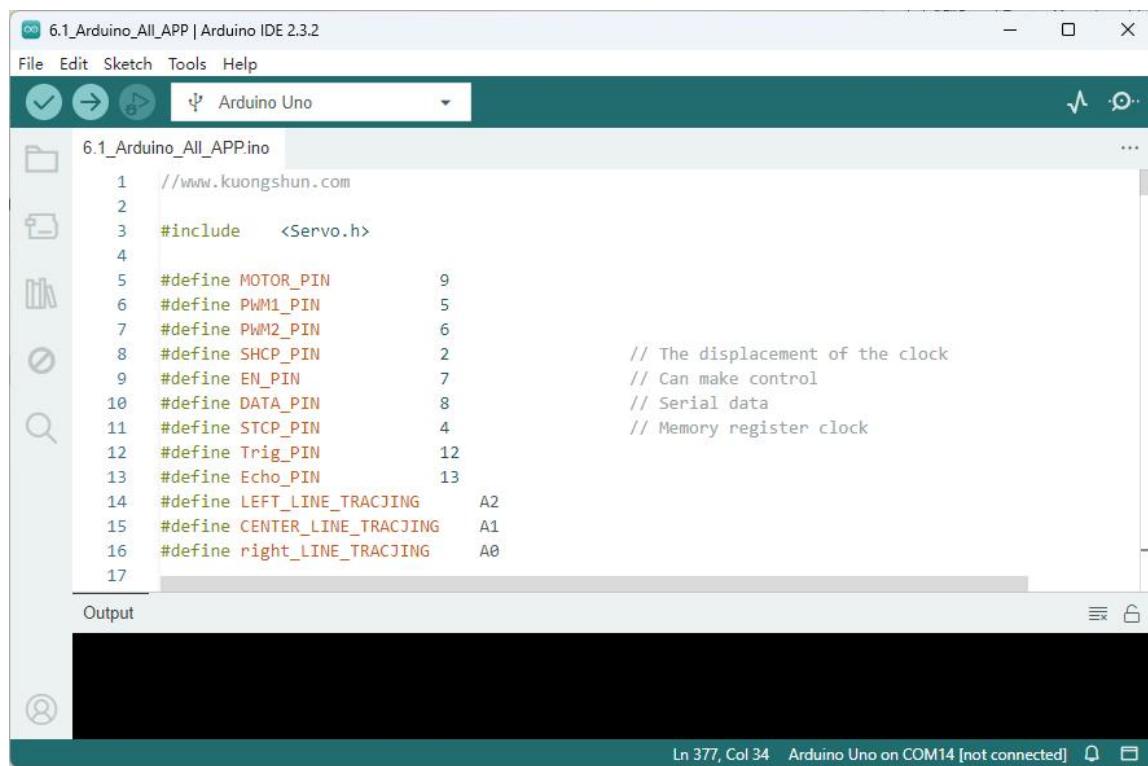
Lesson 6 WIFI Camera Multi-Function Program for APP

In this project, we will use the Arduino board combined with the L293D extension board to realize the functions of walking remote control, obstacle avoidance, patrolling and following of the cart. ESP32-CAM is used to realize WiFi connection and control the cart through serial line. At the same time, ESP32-CAM also transmits the images from the camera at the front of the cart to the web server connected to the hotspot in real time. As long as we connect to this WIFI hotspot, we can use the APP to watch the monitoring in real time, and at the same time, we can control the traveling of the cart through the buttons of the APP, as well as adjusting the mode of operation and the speed of the cart, and so on.

Before presenting the multi-function program of the cart, we first upload the program to the cart. The car has two control boards, and we need to upload two programs to each board. Here are the steps to upload an Arduino board and program:

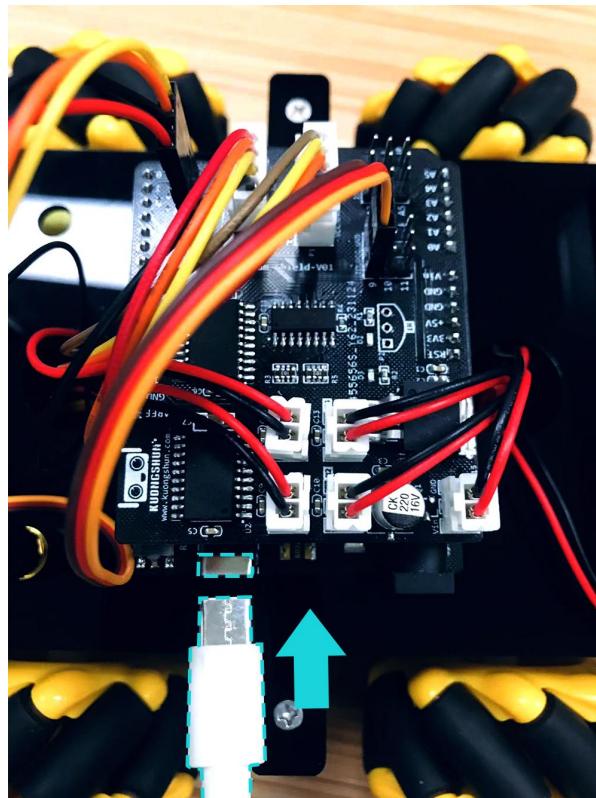
- ① Start by going to the following path and opening the program:



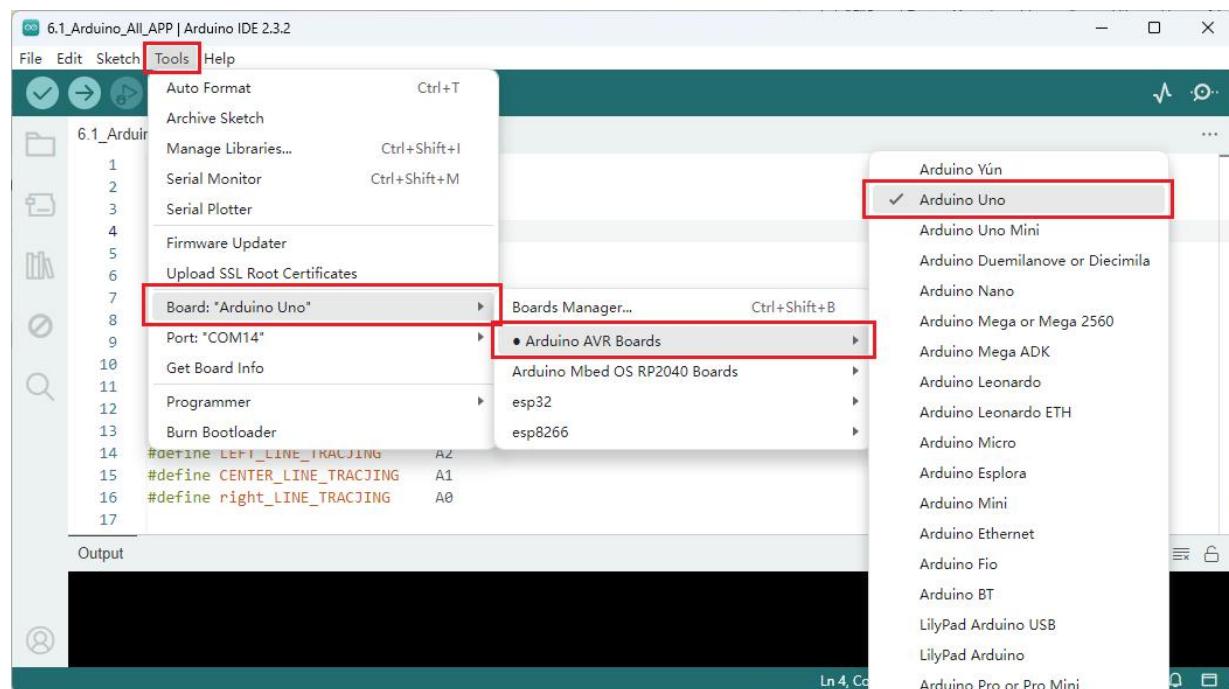


```
6.1_Arduino_All_APP | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Uno
6.1_Arduino_All_APP.ino
1 //www.kuongshun.com
2
3 #include <Servo.h>
4
5 #define MOTOR_PIN 9
6 #define PWM1_PIN 5
7 #define PWM2_PIN 6
8 #define SHCP_PIN 2 // The displacement of the clock
9 #define EN_PIN 7 // Can make control
10 #define DATA_PIN 8 // Serial data
11 #define STCP_PIN 4 // Memory register clock
12 #define Trig_PIN 12
13 #define Echo_PIN 13
14 #define LEFT_LINE_TRACING A2
15 #define CENTER_LINE_TRACING A1
16 #define right_LINE_TRACING A0
17
Output
Ln 377, Col 34 Arduino Uno on COM14 [not connected]
```

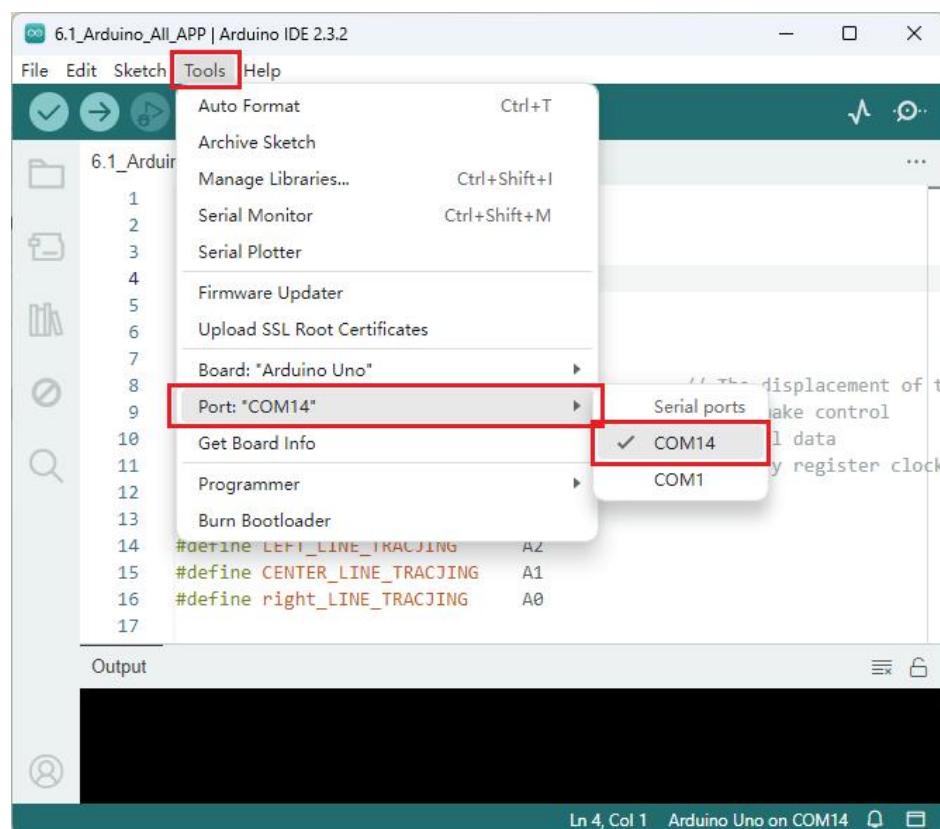
② Connect the Arduino board to the computer with TYPE-C data cable



③ Select your control board: Tools > Board >> Arduino AVR Board >>> Arduino UNO

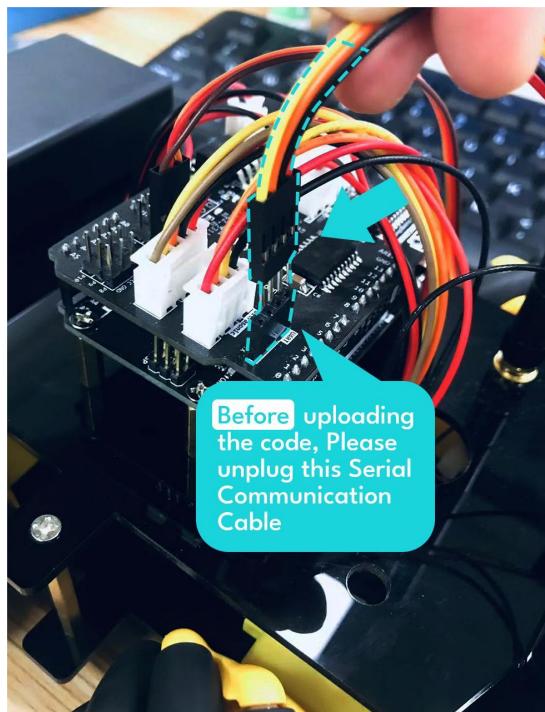


④Select the Port: Tools >Port >>>>COMxx(The COM port number of each computer may be different, and the port number that appears is your port number) (If you do not see COM ports other than COM1 in Arduino IDE, you need to refer to the installation of CH340 driver in "2_Programming_Preparation" and install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

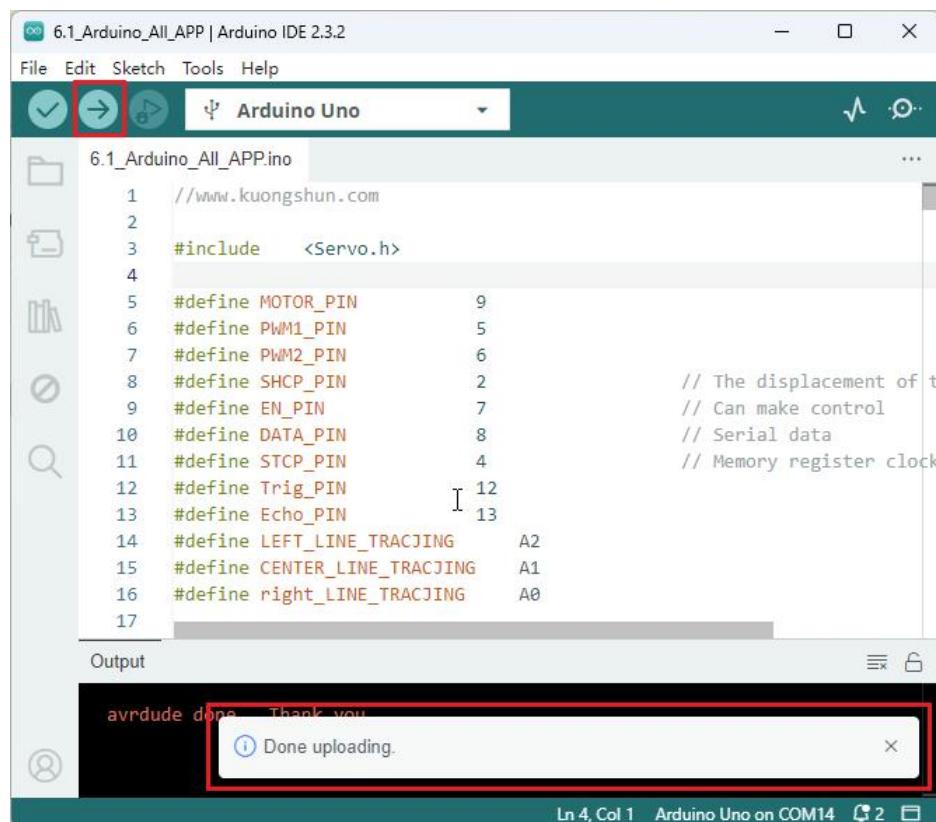


2_Programming_Preparation > 1. Installation of Arduino IDE and CH340 driver			
名称	修改日期	类型	大小
CH340 Serial Driver	2023/12/18 17:50	文件夹	
Mac OS Tutorial for Arduino IDE Installation...	2023/12/8 18:26	DOCX 文档	4,563 KB
Windows Tutorial for Arduino IDE Installatio...	2023/12/18 11:16	DOCX 文档	982 KB

⑤Unplug the serial port transmission cable between the Arduino board and the ESP32-CAM board.(This cable will not be connected back during all uploading procedures, until finally all procedures are uploaded and the car is controlled by WIFI.)



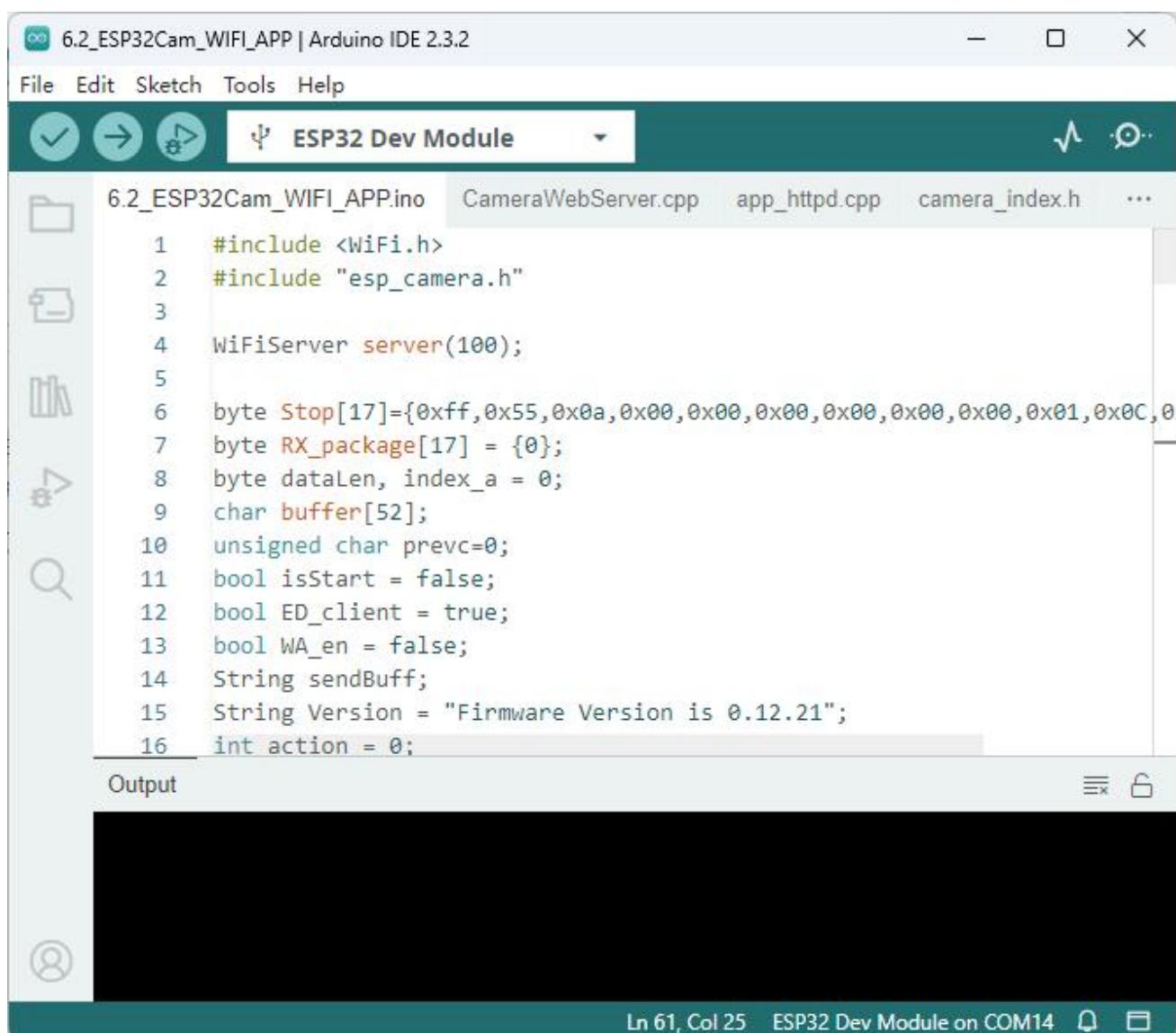
⑥Upload the program to the UNO controller board.



The picture above shows that it is uploaded successfully.

⑦ After uploading the code successfully. We also need to upload the program to the **ESP32-CAM board** of the locomotive. Steps and code needed to open the following path:

名称		修改日期
sonal	6.2_ESP32Cam_WIFI_APP.ino	2024/5/15 18:16
	app_httpd.cpp	2024/5/15 14:07
	camera_index.h	2024/5/15 14:07
	CameraWebServer.cpp	2024/5/21 14:59



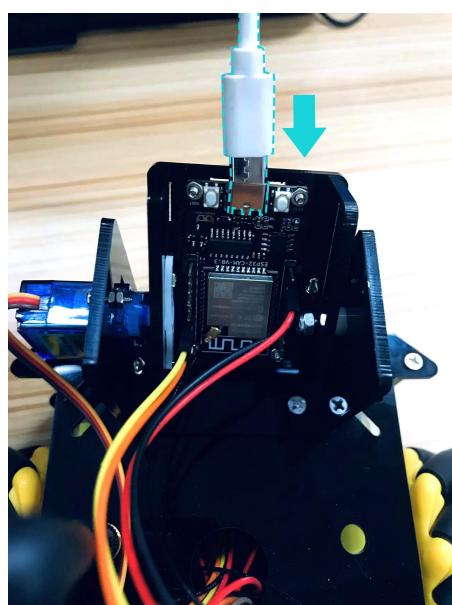
The screenshot shows the Arduino IDE interface with the following details:

- Project title: 6.2_ESP32Cam_WIFI_APP | Arduino IDE 2.3.2
- Board: ESP32 Dev Module
- Sketch content (6.2_ESP32Cam_WIFI_APP.ino):

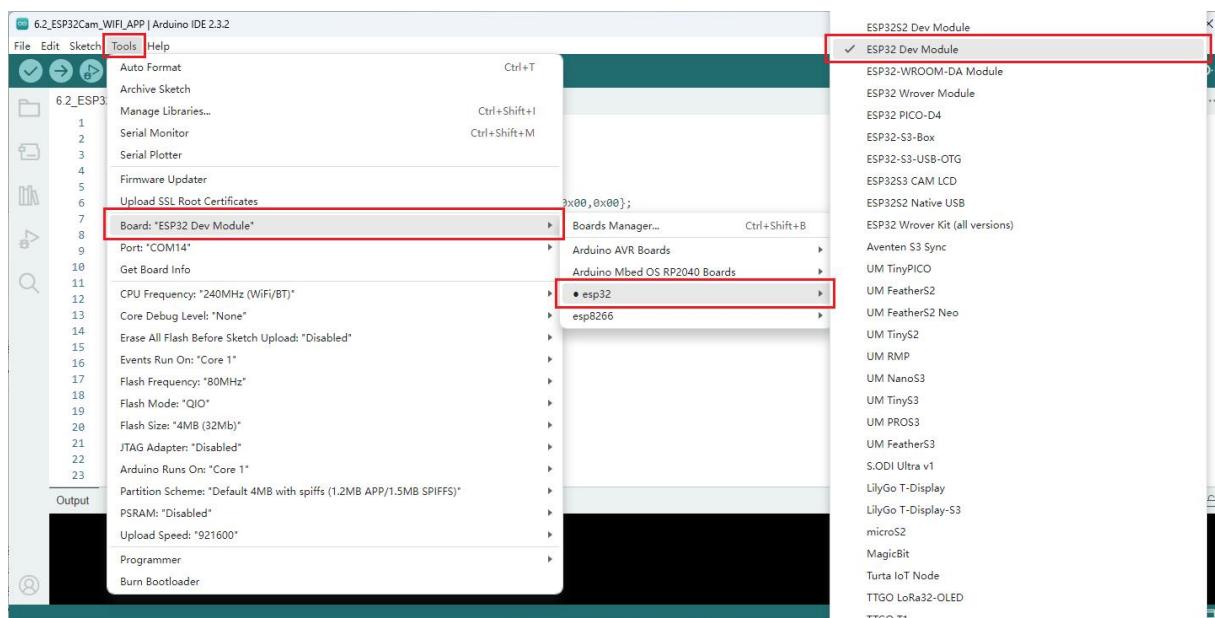
```
1 #include <WiFi.h>
2 #include "esp_camera.h"
3
4 WiFiServer server(100);
5
6 byte Stop[17]={0xff,0x55,0x0a,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x0C,0
7 byte RX_package[17] = {0};
8 byte dataLen, index_a = 0;
9 char buffer[52];
10 unsigned char prevc=0;
11 bool isStart = false;
12 bool ED_client = true;
13 bool WA_en = false;
14 String sendBuff;
15 String Version = "Firmware Version is 0.12.21";
16 int action = 0;
```

The code is a WiFi server setup for an ESP32-CAM, including camera-related headers and variables.

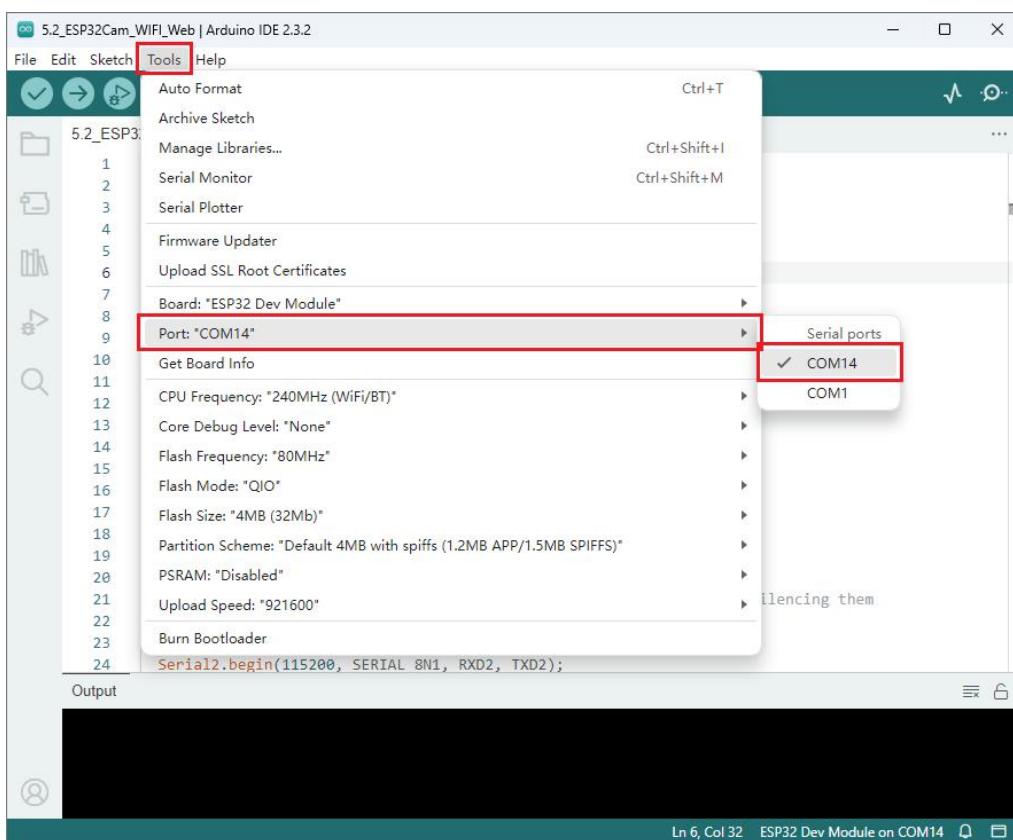
⑧Connect the **ESP32-CAM board** to the computer with TYPE-C cable



⑨ Select your board: Tools > Board >>esp32>>>ESP32 Dev Module

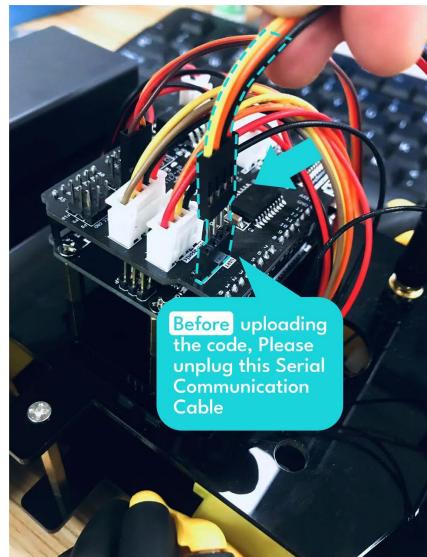


⑩ Select the Port: Tools >Port >>>>COMxx(The COM port number of each computer may be different, and the port number that appears is your port number) (If you do not see COM ports other than COM1 in Arduino IDE, you need to refer to the installation of CH340 driver in "2_Programming_Preparation" and install it)



2_Programming_Preparation > 1. Installation of Arduino IDE and CH340 driver			
名称	修改日期	类型	大小
CH340 Serial Driver	2023/12/18 17:50	文件夹	
Mac OS Tutorial for Arduino IDE Installation...	2023/12/8 18:26	DOCX 文档	4,563 KB
Windows Tutorial for Arduino IDE Installatio...	2023/12/18 11:16	DOCX 文档	982 KB

⑪ unplug the serial port transmission cable between the Arduino control board and the ESP32-CAM control board. (This line can be returned after the upload of this program)



⑫ Upload the program to the ESP32-CAM control board.

```

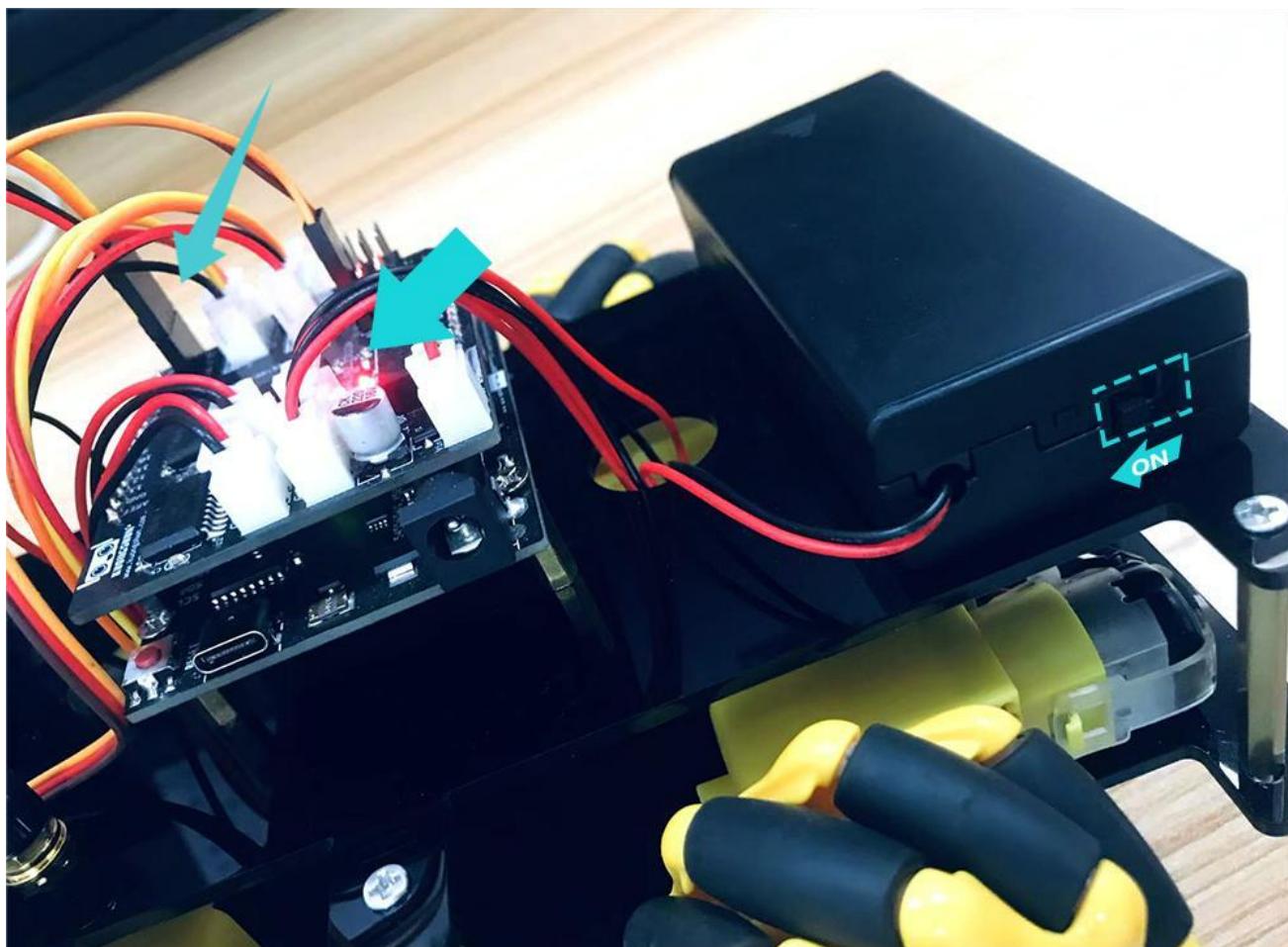
6.2_ESP32Cam_WIFIAPP | Arduino IDE 2.3.2
File Edit Sketch Tools Help
ESP32 Dev Module
6.2_ESP32Cam_WIFIAPP.ino CameraWebServer.cpp app_httpd.cpp camera_index.h ...
1 #include <WiFi.h>
2 #include "esp_camera.h"
3
4 WiFiServer server(100);
5
6 byte Stop[17]={0xff,0x55,0x0a,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x0C,0x00,0x00}
7 byte RX_package[17] = {0};
8 byte datalen, index_a = 0;
9 char buffer[52];
10 unsigned char prevc=0;
11 bool isStart = false;
12 bool ED_client = true;
13 bool WA_en = false;
14 String sendBuff;
15 String Version = "Firmware Version is 0.12.21";
16 int action = 0;
17 char device = 0;
18 byte val = 0;
19 #define _BV(n) (1<<n) // Right
Output
Writing at 0x000d3533... (91 %)
Writing at 0x000d8e73... (94 %)
Writing at 0x000de462... (97 %)
Writing at 0x000e3e17... (100 %)
Wrote 876192 bytes (579153 compressed) at 0x00010000 in 8.0 seconds (effective 881.5 k
Hash of data verified.

Leaving...
Hard resetting via
Done uploading.

```

The picture above shows that it is uploaded successfully.

⑬ After 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 two 18650 batteries.) Then turn on the switch.



⑭ 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)

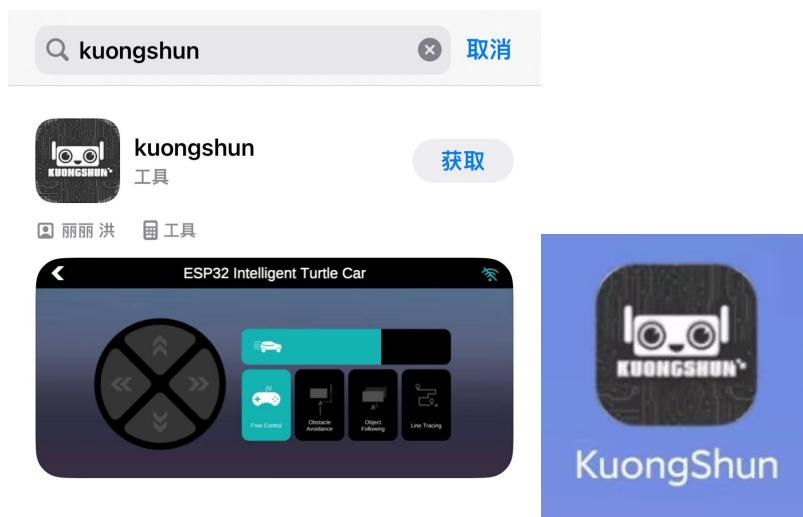
4_Android_Software_Installation_Package		C
名称	修改日期	
KUONGSHUN.apk	2024/5/6 15:29	

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



For the Apple system:

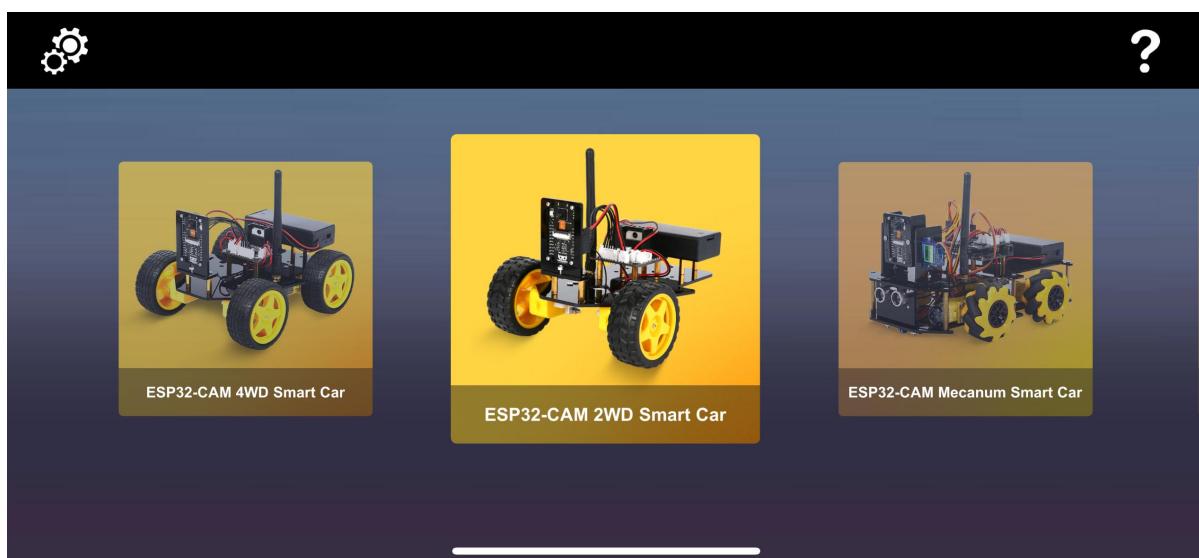
Enter kuongshun in the APP Store search bar, click to get the download and install, install as shown in the right picture.



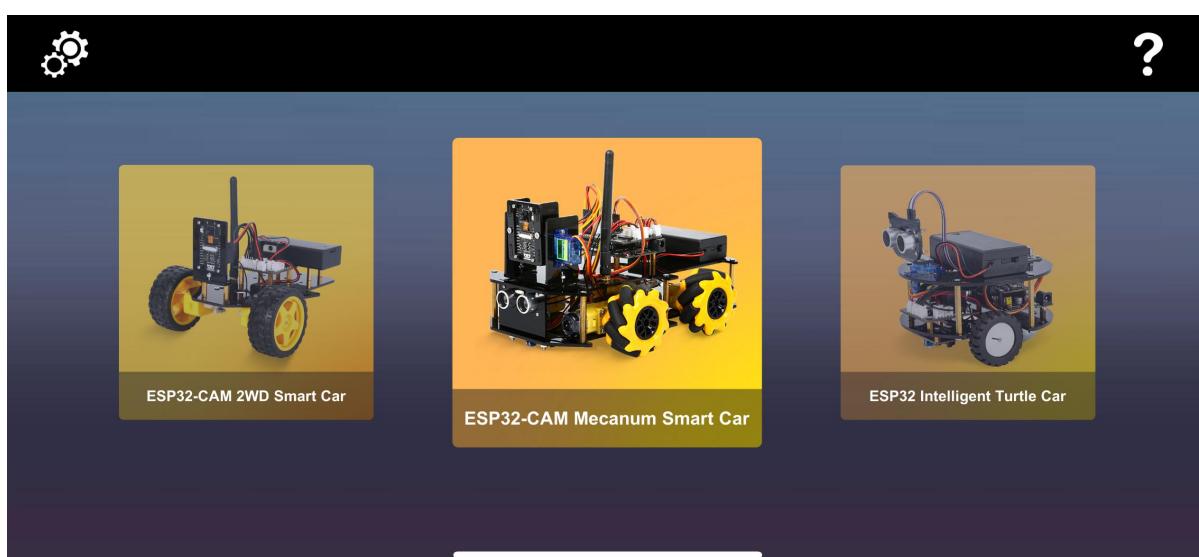
⑯ Then pick up your phone, turn on the WiFi switch, find "KUONGSHUN-AD176", enter the password "12345678" and connect to the hotspot. If you are reminded that there is no network, please keep connected. (If the switch is turned on and the module lights up but the phone cannot search for "KUONGSHUN", please check whether the wiring is correct and whether the battery is powered.)



⑯Open the APP



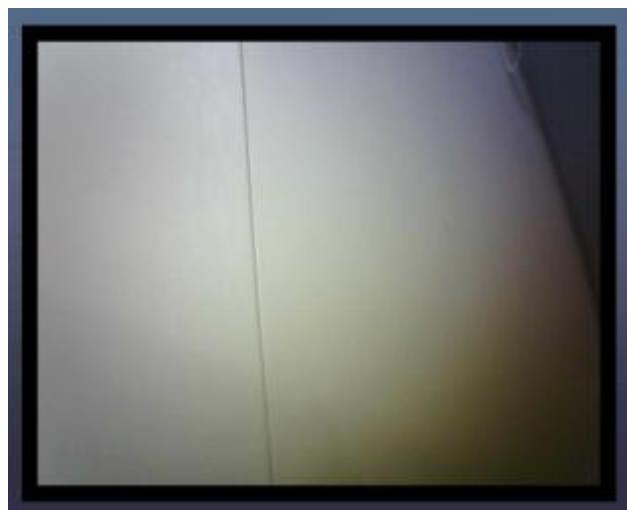
Select ESP32-CAM Mecanum Smart Car



Click to connect automatically



⑯ Interface Introduction



: Video screen display area



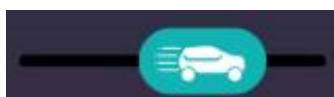
: The pause button freezes the screen, the play button opens the video screen, and the close button closes the video screen



: Tap to switch to the corresponding sport mode



: Slide to adjust headlight brightness



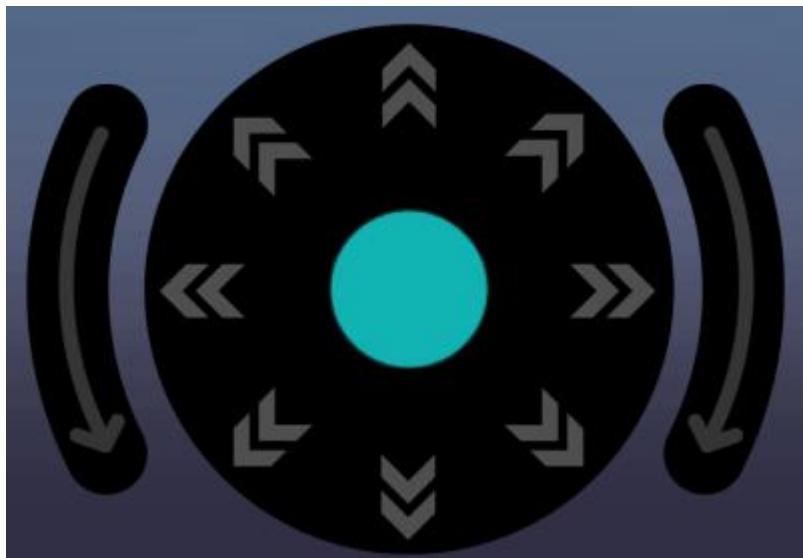
: Slide to adjust the speed of the cart



: Displays the connection status, if WiFi is connected but not connected to the cart, you can tap here to connect



: Adjustable head camera tilt angle



: Press or drag the ball to control the car to move in the corresponding direction, press and hold the rotate button to make the car rotate in place
Now when you start the car again, you don't need to download the program again. You just start in step ⑯

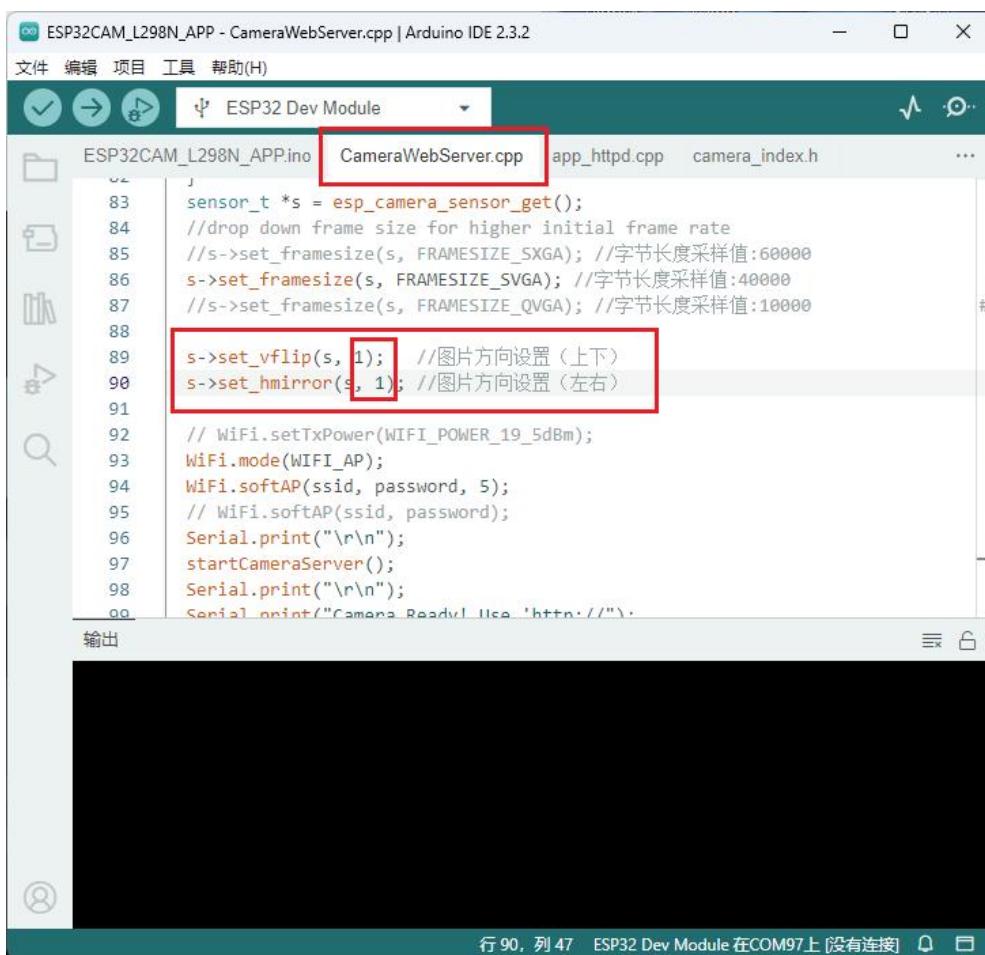
Frequently Asked Questions

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: Maybe the direction changed before and after the camera update, you can find the code in the file box below and change 1 to 0.(Web and App versions are modified here)



The screenshot shows the Arduino IDE interface with the project 'ESP32CAM_L298N_APP' open. The 'CameraWebServer.cpp' file is the active tab, displayed in the code editor. A red box highlights the following lines of code, which are responsible for setting the camera's orientation:

```
sensor_t *s = esp_camera_sensor_get();
//drop down frame size for higher initial frame rate
//s->set_framesize(s, FRAMESIZE_SXGA); //字节长度采样值:60000
s->set_framesize(s, FRAMESIZE_SVGA); //字节长度采样值:40000
//s->set_framesize(s, FRAMESIZE_QVGA); //字节长度采样值:10000
s->set_vflip(s, 1); //图片方向设置（上下）
s->set_hmirror(s, 1); //图片方向设置（左右）
```

The code editor also shows other files in the project: 'ESP32CAM_L298N_APP.ino', 'app_httpd.cpp', and 'camera_index.h'. The status bar at the bottom indicates '行 90, 列 47 ESP32 Dev Module 在COM97上 [没有连接]'.