

Lesson 16 How to use APP to Control the Car

In this tutorial, we learn how to use the **Android APP** to control the car.

16.1 Introduction of ESP8266



ESP01/ESP01S is a type of ESP8266. ESP8266 is a low-cost, high-performance Wi-Fi module developed by Espressif, a company based in China. It has become a prominent choice in Internet of Things (IoT) and wireless communication projects due to its reliable wireless connectivity, suitable for a range of applications from simple sensors to complex smart devices.

Key Features:

Wi-Fi Connectivity: ESP8266 supports 802.11 b/g/n Wi-Fi connections, allowing easy wireless network access for remote communication and control.

Affordability: The cost of ESP8266 is relatively low, making it suitable for mass IoT projects even with budget constraints.

High Performance: Despite its small size, ESP8266 boasts satisfactory computing and communication performance.

GPIO Pins: ESP8266 features multiple GPIO pins, which can be used to connect various external devices like sensors, actuators, and more.

UART Communication: It supports UART communication, enabling serial communication with other devices.

Embedded Development: Programming and development can be carried out using Arduino IDE or other suitable environments for ESP8266.

Flash Storage: Built-in flash storage allows for storing programs and data.

OTA (Over-the-Air) Updates: Supports remote firmware updates over Wi-Fi, eliminating the need for USB or other cables.

Open Source: Most software libraries and SDKs for ESP8266 are open source, allowing for customization and expansion.

16.2 Introduction of WiFi Car

Before programming, we need to have a basic understanding of WiFi.

Station mode

When ESP8266 selects Station mode, it acts as a WiFi client. It can connect to the router network and communicate with other devices on the router via WiFi connection.

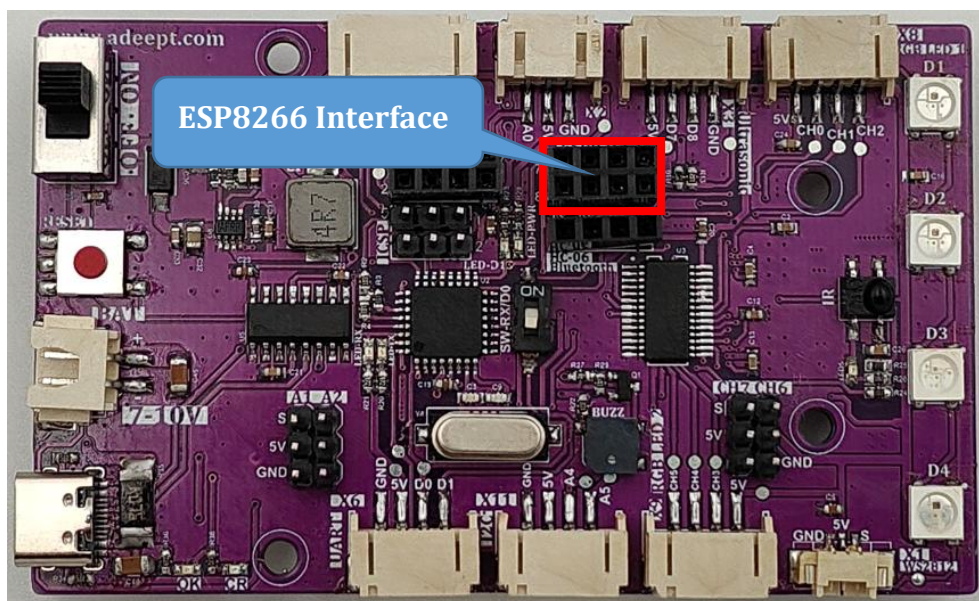
AP mode

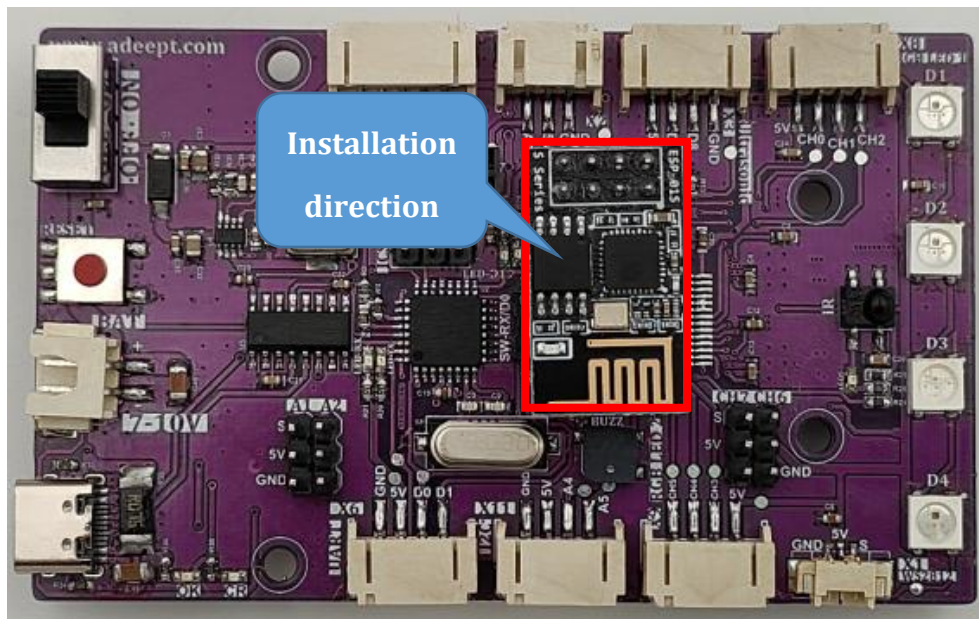
When ESP8266 selects AP mode, it creates a Hotspot network that is separate from the Internet and waits for other WiFi devices to connect. ESP8266 is used as a Hotspot. If a mobile phone or PC wants to communicate with ESP8266, it must be connected to the Hotspot of ESP8266. Only after a connection is established with ESP8266 can they communicate. This is used in the tutorial.

AP+Station mode

In addition to AP mode and station mode, ESP8266 can also use AP mode and station mode at the same time.

16.3 Install of ESP8266 module





16.4 Introduction of the Adeept APP

Install APP

Download the ZIP from the link below and extract it to your Android phone. The installation method is the same as that of the ordinary mobile phone APP.

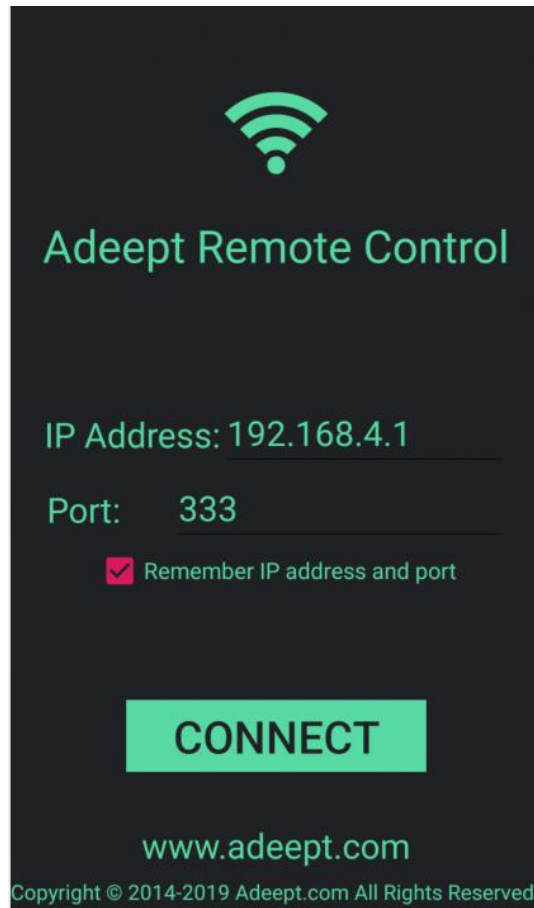
Link: <https://www.adeept.com/learn/detail-41.html>

Icon after successful installation:



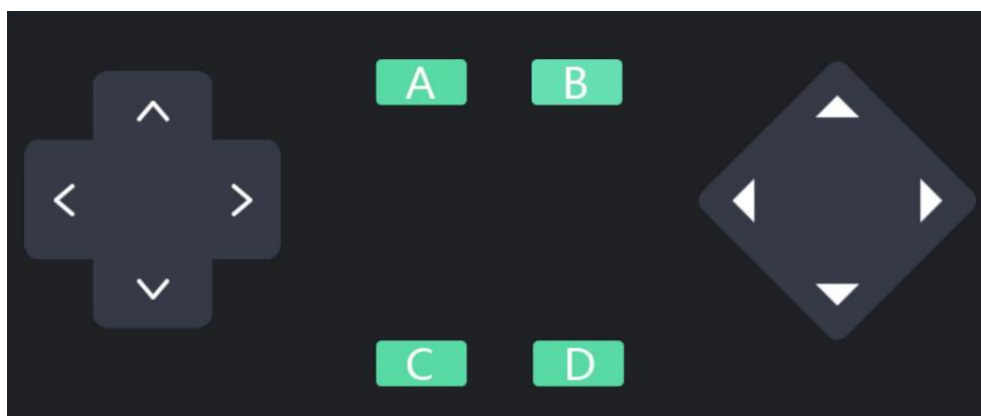
Connect to WiFi. Open the mobile app, enter the IP address of the ESP8266 in the IP address field of the mobile app, and enter port number. Click Connect.

Login interface



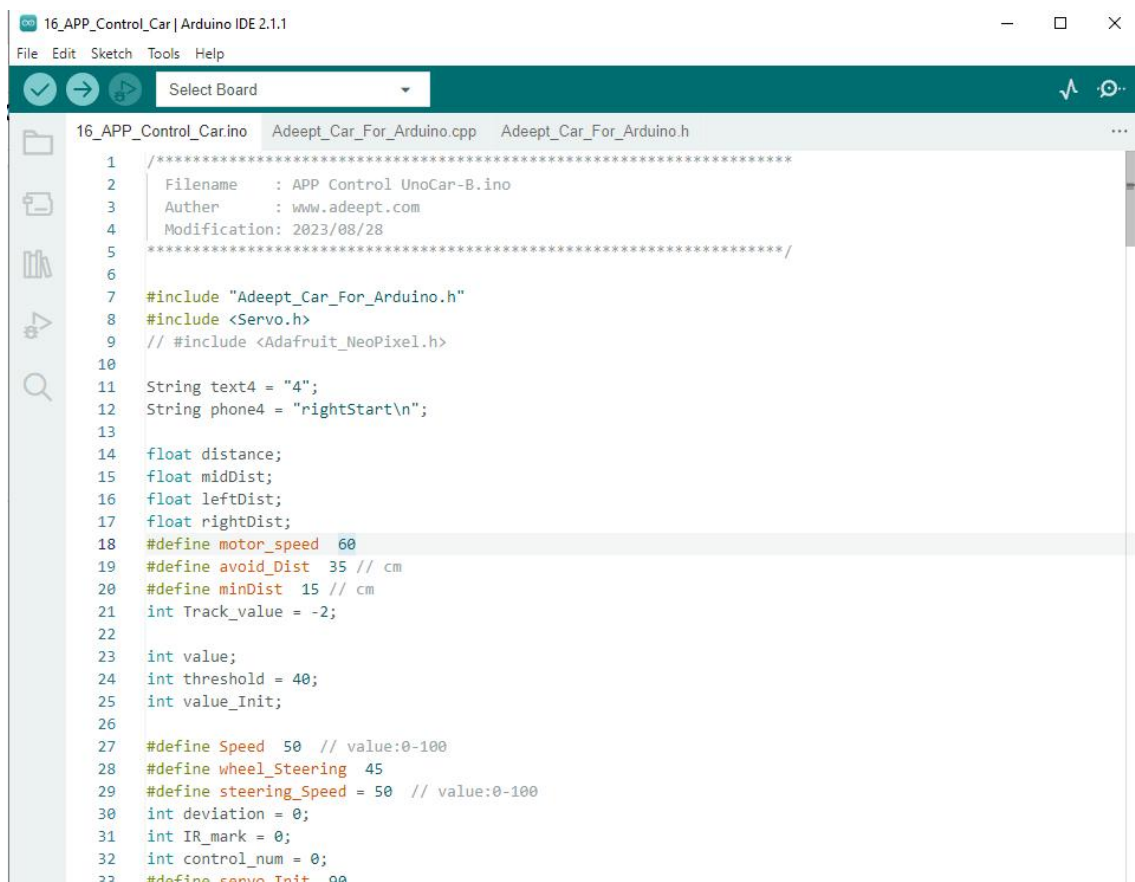
The login interface of the Adeept Remote Control app. It features a dark background with a green Wi-Fi icon at the top. Below the icon, the text "Adeept Remote Control" is displayed in green. There are two input fields: "IP Address: 192.168.4.1" and "Port: 333". A checkbox labeled "Remember IP address and port" is checked. A large green "CONNECT" button is centered below the input fields. At the bottom, the website "www.adeept.com" and a copyright notice "Copyright © 2014-2019 Adeept.com All Rights Reserved." are visible.

Operation interface(Before the ESP8266 is configured successfully, you cannot enter this interface)



16.3 Control the car in AP mode

1. Connect your computer and Adeept Robot Control Board with a USB cable.
2. Open “16_APP_Control_car” folder in “[Adeept_UnoCar-B/Code](#)”, double-click “[16_APP_Control_car.ino](#)”.



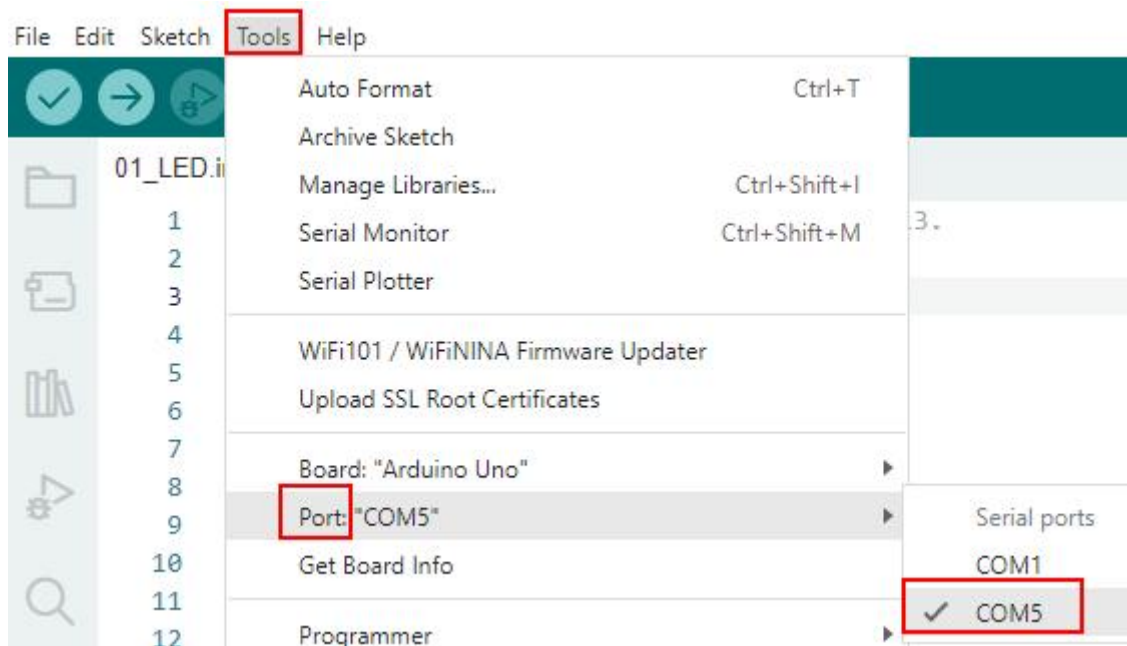
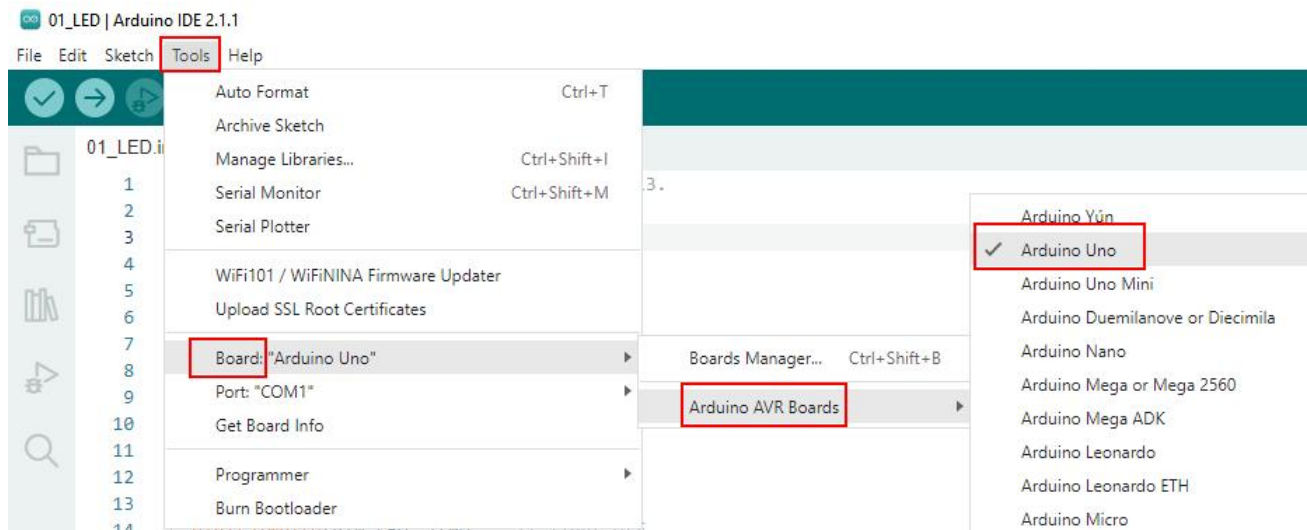
```
1  /*****
2  Filename   : APP Control UnoCar-B.ino
3  Author    : www.adeept.com
4  Modification: 2023/08/28
5  *****/
6
7  #include "Adeept_Car_For_Arduino.h"
8  #include <Servo.h>
9  // #include <Adafruit_NeoPixel.h>
10
11  String text4 = "4";
12  String phone4 = "rightStart\n";
13
14  float distance;
15  float midDist;
16  float leftDist;
17  float rightDist;
18  #define motor_speed 60
19  #define avoid_Dist 35 // cm
20  #define minDist 15 // cm
21  int Track_value = -2;
22
23  int value;
24  int threshold = 40;
25  int value_Init;
26
27  #define Speed 50 // value:0-100
28  #define wheel_Steering 45
29  #define steering_Speed = 50 // value:0-100
30  int deviation = 0;
31  int IR_mark = 0;
32  int control_num = 0;
33  #define servo_Init aa
```


3. Select development board and serial port.

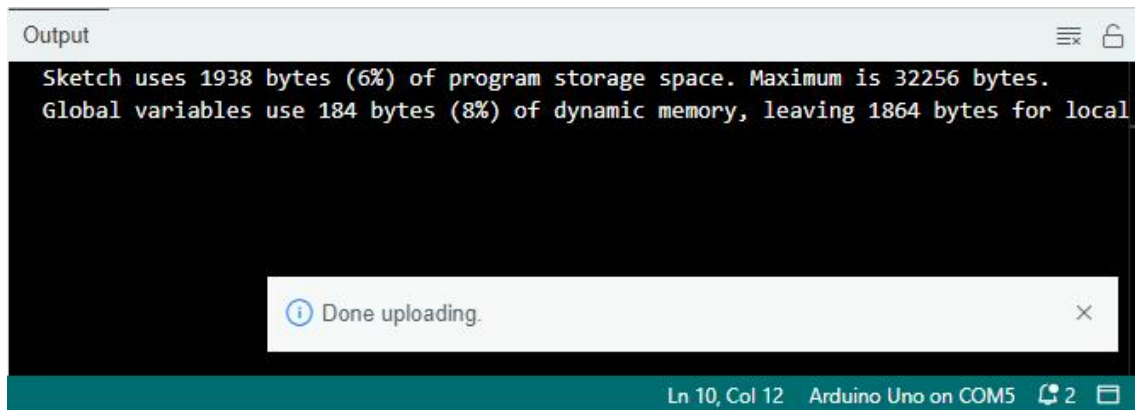
Board: [Tools](#)---->[Board](#)---->[Arduino AVR Boards](#)---->[Arduino Uno](#)

Port: [Tools](#) ---->[Port](#)---->[COMx](#)

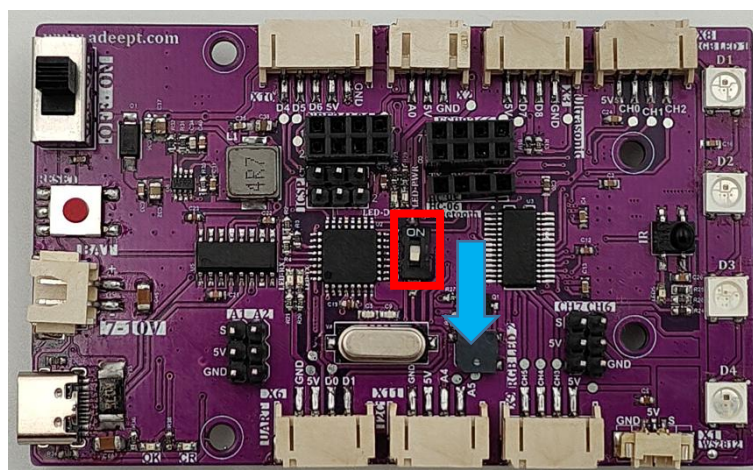
Note: The port number will be different in different computers.



4. After opening, click  to upload the code program to the Arduino. If there is no error warning in the console below, it means that the Upload is successful.



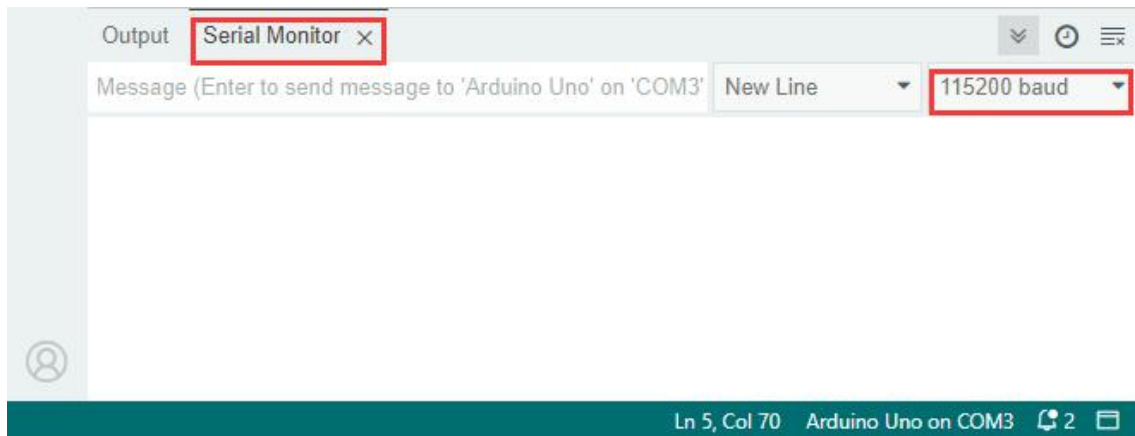
Note: If the upload program fails, a red error message appears. Please confirm whether the RX/D0 switch is in the correct position. Toggle down the paddle (white) of the switch in the picture below.



After assembling the car, please use the 18650 battery to provide power when uploading the program, otherwise the program may not be uploaded successfully due to excessive load.

- click Serial Monitor, Set the baud rate as 115200.

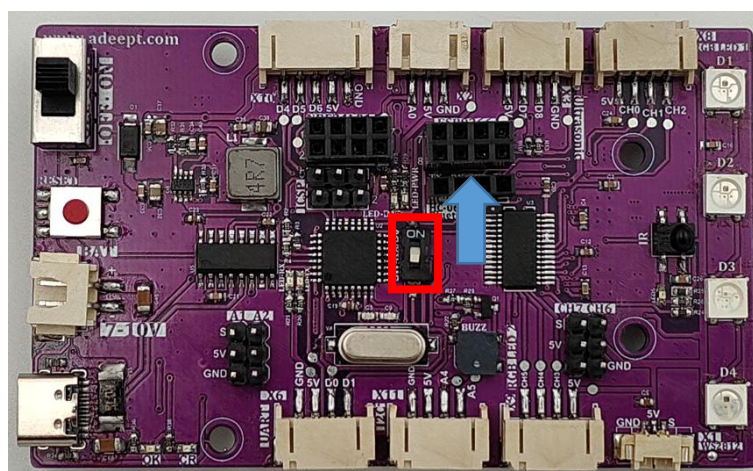




Note: When the ESP8266 module transmits data to the Arduino, it needs to occupy the RX interface of the Arduino, and when the Arduino uploads the program, it also needs to occupy the RX interface. The RX interface cannot satisfy both functions at the same time, so a switch is needed to distinguish them.

When the switch is flipped downward, the ESP8266 module is disconnected from the RX interface, and the program can be uploaded normally. When the switch is flipped upward, the RX interface is connected to the ESP8266, and the ESP8266 module will continue to occupy the RX interface. At this time, the program cannot be uploaded normally.

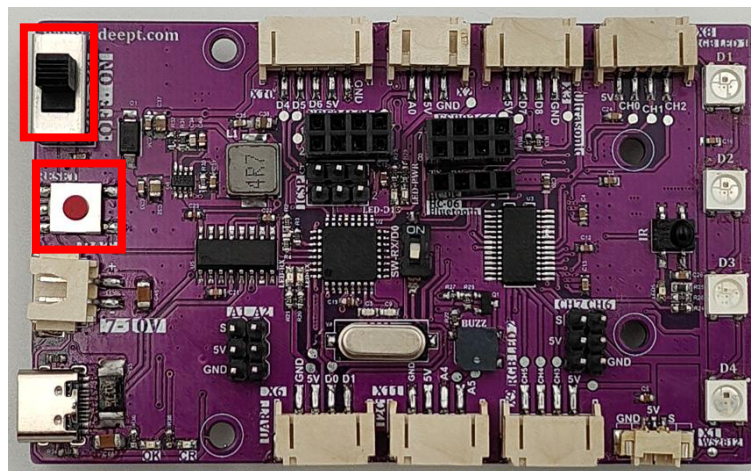
Now we need the communication function of ESP8266, so we need to toggle the switch up.



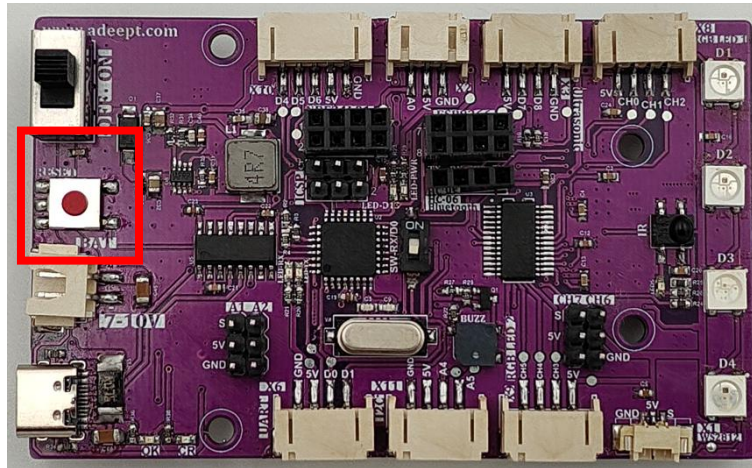
If you need to upload other programs later, and ESP8266 is installed on Adeept Robot Control Board, you need to flip the switch down to disconnect RX from ESP8266.

If the following error messages also appear, this may cause the program upload to fail. Please try to press the "RESET" button, or try to turn off the power switch and then turn on the power switch.

```
Output
Sketch uses 25084 bytes (77%) of program storage space. Maximum is 32256 bytes.
Global variables use 1200 bytes (56%) of dynamic memory, leaving 812 bytes for local variables. Maximum is 2048 bytes.
avrdude: loadaddr(): (b) protocol error, expect=0x14, resp=0xfc
```



6. After opening the Serial Monitor in the upper right corner, press the "RESET" button on the Adeept Robot Control Board, and you can see that the AP mode is being turned on. This interface can also see various commands sent from the mobile APP.



7. After uploading successfully, the mobile phone can detect a WiFi name named "Adeept_ESP8266", and the WiFi password is "12345678". The WiFi name and password can be modified through the procedure below. Modifications to the program are not recommended for initial use.

📶 Adeept_ESP8266

```
73 void setup()
74 {
75     Serial.begin(115200);    // set up a wifi serial communication baud
76
77     Serial.println("AT+CWMODE=3\r\n");//set to softAP+station mode
78     delay(3000);           //delay 4s
79     Serial.println("AT+CWSAP=\"Adeept_ESP8266\", \"12345678\", 8, 2\r\n");
80     delay(1000);           //delay 4s
81     Serial.println("AT+RST\r\n");    //reset wifi
```

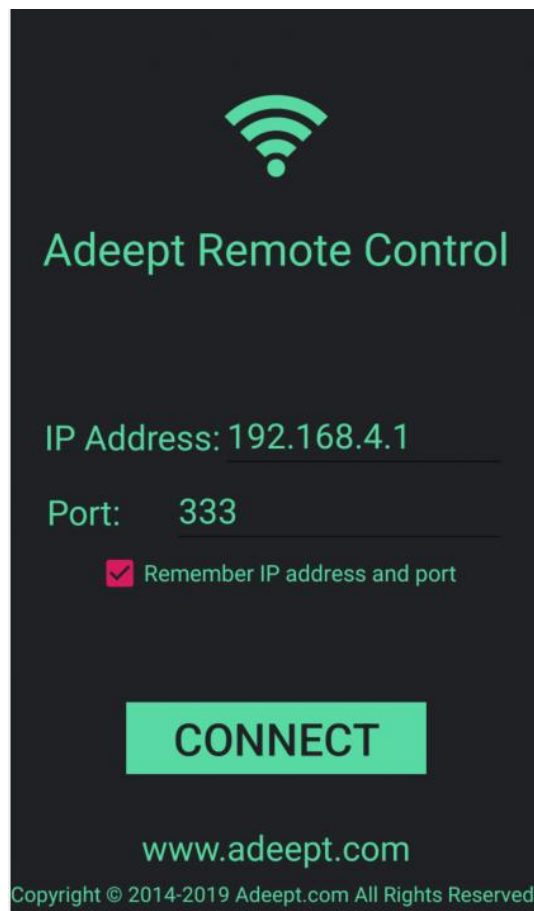
8. Use an Android phone to connect to "Adeept_ESP8266" WiFi. Since this WiFi can only be used for communication between the mobile phone and ESP8266, after the mobile phone is connected to WiFi, it cannot access the external network (you cannot use the mobile phone to access the Internet), and the mobile phone may prompt that this WiFi cannot connect to the network, which is a normal phenomenon.

Please make sure that the mobile phone can always connect to the WiFi "Adeept_ESP8266". A pop-up window may appear on the mobile phone to prompt you to change the WiFi. Please do not allow the mobile phone to connect to another WiFi. Otherwise, the mobile phone cannot communicate with the UnoCar-B car.

9. Open the installed "Adeept" APP, and enter the IP address and port number in AP mode. Then click "CONNECT". The IP address remains unchanged.

IP Address: 192.168.4.1

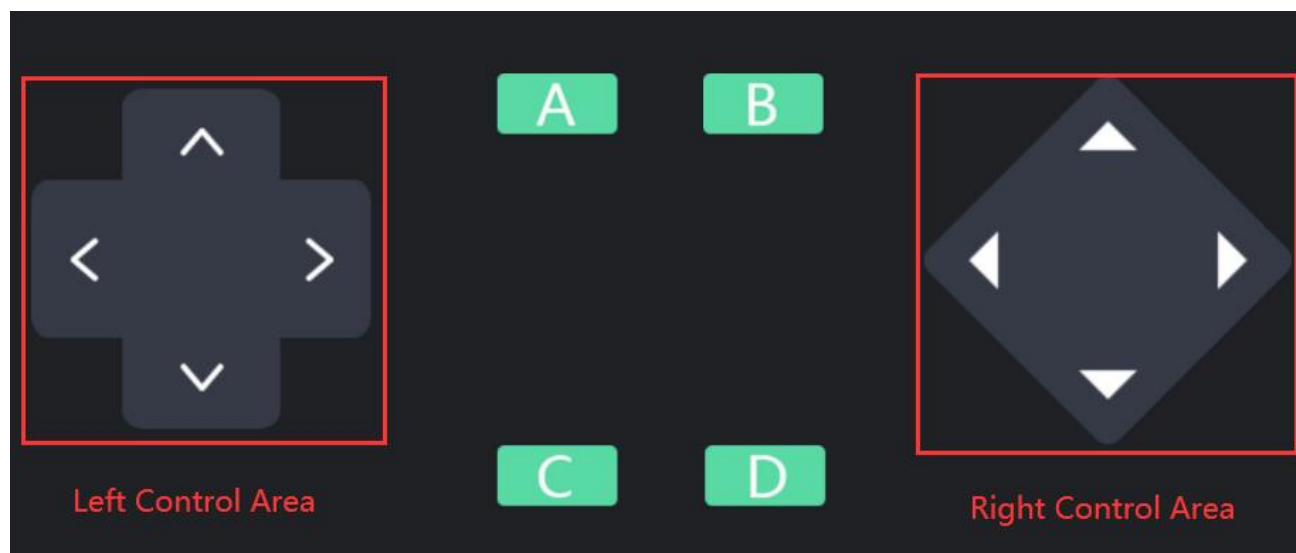
Port: 333











If the APP crashes at this time, it is because the mobile phone cannot communicate with the input IP. Please confirm whether the mobile phone is connected to the WiFi of

"Adeept_ESP8266". Please confirm whether the entered IP address and port number are correct.

10. After clicking "CONNECT", the APP enters the operation interface.



Left Control Area	Function	Right Control Area	Function
	Forward		/
	Backward		/
	Turn Left		Head Turn Left
	Turn Right		Head Turn Right
A	Ultrasonic Avoid Obstacles Function	C	Light Tracking Function
B	Stop Avoid Obstacles (click multiple times)	D	Stop Light Tracking (click multiple times)
Some functions may require multiple clicks to stop.			

16.5 Code

[Adeept_Car_For_Arduino.cpp](#) and [Adeept_Car_For_Arduino.h](#) are the library files of the car. These two files have modularized the code in the previous course, so that the program can call the code of each module.

The [16_APP_Control_car.ino](#) program is the main program, which realizes the functions required by the car.

Note: Since the Adeept Robot Control Board uses the same chip as the Arduino Uno, the [program storage space](#) of the chip is 32kb. When there are too many modified program codes, it may fail to upload to the Arduino board. Please adjust the code content appropriately.

The following is a display of the [16_APP_Control_car.ino](#) program content, which may be subject to change. Please refer to the actual code provided.

```
1. #include "Adeept_Car_For_Arduino.h"
2. #include <Servo.h>
3. // #include <Adafruit_NeoPixel.h>
4.
5. float distance;
6. float midDist;
7. float leftDist;
8. float rightDist;
9. #define motor_speed 60
10. #define avoid_Dist 35 // cm
11. #define minDist 15 // cm
12. int Track_value = -2;
13.
14. int value;
15. int threshold = 40;
16. int value_Init;
17.
18. #define Speed 50 // value:0-100
19. #define wheel_Steering 45
20. #define steering_Speed = 50 // value:0-100
21. int deviation = 0;
```

```
22. int IR_mark = 0;
23. int control_num = 0;
24. #define servo_Init 90
25. int servo_Angle2 = servo_Init;
26. int ws2812_flag= 0;
27. int Function_Stop_flag = 0;
28.
29.
30. String Move_UP1 = "forwardStart\n";
31. String Move_UP2 = "forwardStop\n";
32. String Move_Down1 = "backwardStart\n";
33. String Move_Down2 = "backwardStop\n";
34. String Move_Left1 = "leftStart\n";
35. String Move_Left2 = "leftStop\n";
36. String Move_Right1 = "rightStart\n";
37. String Move_Right2 = "rightStop\n";
38.
39. String Head_UP1 = "lookLeftStart\n";
40. String Head_UP2 = "lookLeftStop\n";
41. String Head_Down1 = "lookRightStart\n";
42. String Head_Down2 = "lookRightStop\n";
43. String Head_Left1 = "downStart\n";
44. String Head_Left2 = "downStop\n";
45. String Head_Right1 = "upStart\n";
46. String Head_Right2 = "upStop\n";
47.
48.
49. String Ultrasonic_ON = "aStart\n";
50. String Ultrasonic_ON2 = "aStop\n";
51. String Ultrasonic_OFF = "bStart\n";
52. // String Ultrasonic_OFF = "\r\n+IPD,4,7:bStart";
53. String Ultrasonic_OFF2 = "bStop\n";
54.
55. String Light_Tracking_ON = "cStart\n";
56. String Light_Tracking_ON2 = "cStop\n";
57. String Light_Tracking_OFF = "dStart\n";
58. // String Light_Tracking_OFF = "\r\n+IPD,4,7:dStart\n";
59. String Light_Tracking_OFF2 = "dStop\n";
60.
61. String comdata = "";
62. int judge;
63.
```

```
64. void setup()
65. {
66.   Serial.begin(115200);      // set up a wifi serial communication baud rate 115200
67.
68.   Serial.println("AT+CWMODE=3\r\n");//set to softAP+station mode
69.   delay(3000);              //delay 4s
70.   Serial.println("AT+CWSAP=\"Adeept_ESP8266\", \"12345678\", 8, 2\r\n"); //TCP Protocol, server IP addr, port
71.   delay(1000);              //delay 4s
72.   Serial.println("AT+RST\r\n"); //reset wifi
73.   delay(1000);              //delay 4s
74.   Serial.println("AT+CIPMUX=1\r\n");//set to multi-connection mode
75.   delay(1000);
76.   // Serial.println("AT+CIPSERVER=1,333\r\n");//set as server
77.   Serial.println("AT+CIPSERVER=1\r\n");//set as server
78.   delay(1000);
79.   Serial.println("AT+CIPSTO=7000\r\n");//keep the wifi connecting 7000 seconds
80.   delay(1000);
81.   RGB_Setup();              //RGB LED initialization
82.   RGB_brightness(2);        // value 0-10
83.   All_RGB(255,0,0);// Set RGB LED color value.
84.   Servo_Setup();            //Servo initialization
85.   PCA9685_Servo_Setup();    //PCA9685 Servo initialization
86.   Motor_Setup();            //Motor initialization
87.   AllMotorStop();
88.   Buzzer_Setup();           //Buzzer initialization
89.   WS2812_Setup();           //WS2812 LED initialization
90.   WS2812_Brightness(5);     // value 0-10
91.   Ultrasonic_Setup();       //Ultrasonic initialization
92.   Photosensitive_Setup();    //Light line initialization
93.   Tracking_Setup();         //Tracking Line initialization
94.   OLED_Setup();             //OLED initialization
95.   Matrix_Setup();
96.
97.   WS2812ColorAll(255, 255,0); // Green
98.   Servo_Angle(1, 90);
99.   Servo_Angle(2, 90);
100.   PCA9685_Servo_Angle(6, 0, 90);
101.   PCA9685_Servo_Angle(7, 0, 90);
102.   Buzzer_Silence();
103.   OLED_clear();
104.   delay(1000);
```

```
105. All_RGB(0,0,0); // Set RGB LED color value.
106. WS2812ColorAll(0,0,0);
107. }
108.
109. void loop()
110. {
111.     while(Serial.available()>0)
112.     {
113.         comdata += char(Serial.read());
114.         delay(1);
115.     }
116.     judgement();
117.     control(judge);
118.
119. }
120.
121. void judgement(){
122.     if (comdata.length() > 0){
123.         // if(comdata.endsWith(text4)||comdata.endsWith(phone4)){//left
124.         //     judge=4;
125.         // Serial.println(comdata);    //reset wifi1111111111
126.         // }
127.         if(comdata.endsWith(Move_UP1)){//forward
128.             judge=1;
129.             Serial.println(comdata); //print received data.
130.         }
131.         else if(comdata.endsWith(Move_Down1)){//backward
132.             judge=2;
133.             Serial.println(comdata);
134.         }
135.         else if(comdata.endsWith(Move_Left1)){//left
136.             judge=3;
137.             Serial.println(comdata);
138.         }
139.         else if(comdata.endsWith(Move_Right1)){//right.
140.             judge=4;
141.             Serial.println(comdata);
142.         }
143.         else if(comdata.endsWith(Move_UP2)||comdata.endsWith(Move_Down2)||comdata.endsWith(Move_Left2)||comdata.endsWith(Move_Right2)){//stop
144.             judge=5;
145.             Serial.println(comdata);
```

```
146.     }
147.     else if(comdata.endsWith(Head_Left1)){//trun left
148.         judge=6;
149.         Serial.println(comdata);
150.     }
151.     else if(comdata.endsWith(Head_Right1)){//trun right
152.         judge=7;
153.         Serial.println(comdata);
154.     }
155.     else if(comdata.endsWith(Head_Left2)||comdata.endsWith(Head_Right2)){//trun right
156.         judge=8;
157.         Serial.println(comdata);
158.     }
159.
160.     else if(comdata.endsWith(Ultrasonic_ON)){//avoid obstacles function.
161.         judge=9;
162.         Serial.println(comdata);
163.     }
164.     else if(comdata.endsWith(Light_Tracking_ON)){//light tracking function.
165.         judge=10;
166.         Serial.println(comdata);
167.     }
168.
169.     else if(comdata.endsWith(Ultrasonic_OFF)){//avoid obstacles function. bstart.
170.         judge=11;
171.         Serial.println(comdata);
172.     }
173.     else if(comdata.endsWith(Light_Tracking_OFF)){//light tracking function. dstart
174.         judge=12;
175.         Serial.println(comdata);
176.     }
177.     comdata = "";
178.     delay(10);
179. }
180.
181. // Serial.print("judge:");
182. // Serial.println(judge);
183. // return judge;
184. }
185.
186. void control(int value){
187.     switch (value) {
```



```
188.     case 1: // forward
189.         Servo_Angle(1, servo_Init + deviation);
190.         Motor(1, 1, motor_speed); //Motor1 forward
191.         Motor(2, 1, motor_speed); //Motor2 forward
192.         // control_num = 12;
193.         break;
194.
195.     case 2: // Down,
196.         Servo_Angle(1, servo_Init + deviation);
197.         Motor(1, -1, motor_speed); //Motor1 backward
198.         Motor(2, -1, motor_speed); //Motor2 backward
199.         // control_num = 13;
200.         break;
201.
202.     case 3: // left
203.         Servo_Angle(1, servo_Init + deviation + wheel_Steering); // left
204.         Motor(1, 1, motor_speed);
205.         Motor(2, 1, motor_speed);
206.         // control_num = 14;
207.         break;
208.
209.     case 4: // right
210.         Servo_Angle(1, servo_Init + deviation - wheel_Steering); // right
211.         Motor(1, 1, motor_speed);
212.         Motor(2, 1, motor_speed);
213.         // control_num = 15;
214.         break;
215.
216.     case 5: // stop
217.         Servo_Angle(1, servo_Init + deviation);
218.         Motor(1, 1, 0);
219.         Motor(2, 1, 0);
220.         // control_num = -1;
221.         break;
222.
223.     case 6: // trun left
224.         servo_Angle2 = servo_Angle2 + 1;
225.         if (servo_Angle2 > 180){
226.             servo_Angle2 = 180;
227.         }
228.         Servo_Angle(2, servo_Angle2);
229.         // control_num = 15;
```

```
230.     delay(10);
231.     break;
232.     case 7: // trun right
233.         servo_Angle2 = servo_Angle2 - 1;
234.         if (servo_Angle2 < 0){
235.             servo_Angle2 = 0;
236.         }
237.         Servo_Angle(2, servo_Angle2);
238.         // control_num = 15;
239.         delay(10);
240.         break;
241.
242.     case 8: // stop servo rotation.
243.         break;
244.
245.     case 9:
246.         Function_Stop_flag = 0;
247.         Avoid_Obstacles(); // Avoid Obstacles function
248.         break;
249.
250.     case 10:
251.         Function_Stop_flag = 0;
252.         Light_Tracking(); // Light Tracking function
253.         break;
254.
255.     case 11:
256.         Servo_Angle(1, servo_Init + deviation);
257.         Motor(1, 1, 0);
258.         Motor(2, 1, 0);
259.         break;
260.
261.     case 12:
262.         Servo_Angle(1, servo_Init + deviation);
263.         Motor(1, 1, 0);
264.         Motor(2, 1, 0);
265.         break;
266.
267.     default:
268.         break;
269. }
270. }
271.
```

```
272. int StopFunction(){
273.     while(Serial.available()>0){
274.         comdata += char(Serial.read());
275.         delay(1);
276.     }
277.     if (comdata.length() > 0){
278.         if(comdata.endsWith(Ultrasonic_OFF)){// Stop Avoid Obstacles function.
279.             Function_Stop_flag = 1;
280.             judge=11;
281.         }
282.         else if(comdata.endsWith(Light_Tracking_OFF)){ // Stop Light Tracking function.
283.             Function_Stop_flag = 2;
284.             judge=12;
285.         }
286.         comdata = "";
287.         delay(10);
288.     }
289. }
290.
291. void Avoid_Obstacles(){
292.     while (1){
293.         StopFunction();
294.         if (Function_Stop_flag == 1){ // Press OK, stop function.
295.             break;
296.         }
297.
298.         Servo_Angle(2, servo_Init + deviation);
299.         delay(80);
300.         int a = GetDistance();
301.         int b = GetDistance();
302.         int c = GetDistance();
303.         midDist = (a+b+c)/3;
304.         Serial.print("Mid:");
305.         Serial.println(midDist);
306.         // Servo_1_Angle(servo_Init); // front wheel
307.         Motor(1,1,0); //Stop the car
308.         Motor(2,1,0);
309.
310.         if (midDist > avoid_Dist){
311.             // Servo_1_Angle(servo_Init+ deviation); // front wheel
312.             Servo_Angle(1, servo_Init + deviation); // front wheel
313.             Motor(1,1,Speed); //forward
```

```
314.     Motor(2,1,Speed);
315. }
316. else if (midDist <= avoid_Dist){
317.     // Servo_1_Angle(servo_Init + deviation); // front wheel
318.     Servo_Angle(1, servo_Init + deviation); // front wheel
319.     Motor(1,1,0); //Stop the car
320.     Motor(2,1,0);
321.     // Servo_2_Angle(servo_Init - 60); // left distance.
322.     Servo_Angle(2, servo_Init + deviation - 60); // left distance.
323.     delay(400);
324.     int a = GetDistance();
325.     int b = GetDistance();
326.     int c = GetDistance();
327.     leftDist = (a+b+c)/3;
328.     Serial.print("Left:");
329.     Serial.println(leftDist);
330.     // Servo_2_Angle(servo_Init + 60); // right distance.
331.     Servo_Angle(2, servo_Init + deviation + 60); // right distance.
332.     delay(400);
333.     a = GetDistance();
334.     b = GetDistance();
335.     c = GetDistance();
336.     rightDist = (a+b+c)/3;
337.     Serial.print("Right:");
338.     Serial.println(rightDist);
339.     // Servo_2_Angle(servo_Init); // back to mid.
340.     Servo_Angle(2, servo_Init + deviation); // back to mid.
341.
342.     if ((leftDist < avoid_Dist)&&(rightDist < avoid_Dist)){ // Judgment left and right.
343.         if (leftDist >= rightDist){
344.             // There are obstacles on the right backward to the left.
345.             // Servo_1_Angle(servo_Init + wheel_Steering + deviation); //turn left backwa
rd
346.             Servo_Angle(1, servo_Init + deviation + wheel_Steering); // turn left back
ward
347.             Motor(1,-1,Speed); //backward
348.             Motor(2,-1,Speed);
349.             delay(500);
350.         }
351.         else{ //There are obstacles on the left.
352.             // Servo_1_Angle(servo_Init - wheel_Steering + deviation); //turn right backw
```

```

    and
353.         Servo_Angle(1, servo_Init + deviation - wheel_Steering);    // turn right bac
    kward
354.         Motor(1,-1,Speed); //backward
355.         Motor(2,-1,Speed);
356.         delay(500);
357.     }
358. }
359. else if ((leftDist > avoid_Dist)&&(rightDist <= avoid_Dist)){
360.     if (midDist < minDist){ // Obstacle ahead
361.         // Servo_1_Angle(servo_Init+ deviation); // backward
362.         Servo_Angle(1, servo_Init + deviation);    // backward
363.         Motor(1,-1,Speed);
364.         Motor(2,-1,Speed);
365.         delay(400);
366.     }
367.     // Servo_1_Angle(servo_Init + wheel_Steering + deviation); // turn left backwar
    d
368.     Servo_Angle(1, servo_Init + deviation + wheel_Steering);    // turn left backwa
    rd
369.     Motor(1,-1,Speed);
370.     Motor(2,-1,Speed);
371.     delay(500);
372. }
373. else if ((leftDist <= avoid_Dist) &&(rightDist > avoid_Dist)){ // There are obstacl
    es on the left.
374.     if (midDist < minDist){ // Obstacle ahead
375.         // Servo_1_Angle(servo_Init + deviation); // backward
376.         Servo_Angle(1, servo_Init + deviation);    // backward
377.         Motor(1,-1,Speed);
378.         Motor(2,-1,Speed);
379.         delay(500);
380.     }
381.     // Servo_1_Angle(servo_Init - wheel_Steering + deviation); //turn right backwar
    d
382.     Servo_Angle(1, servo_Init + deviation - wheel_Steering);    // turn right backw
    ard
383.     Motor(1,-1,Speed); //backward
384.     Motor(2,-1,Speed);
385.     delay(400);
386. }
387. else if ((leftDist >= avoid_Dist) &&( rightDist >= avoid_Dist)){

```



```
388.         if (leftDist > rightDist){ // The distance to the right is greater than the left
389.             if (midDist < minDist){
390.                 // Servo_1_Angle(servo_Init+ deviation); // backward
391.                 Servo_Angle(1, servo_Init + deviation); // backward
392.                 Motor(1,-1,Speed);
393.                 Motor(2,-1,Speed);
394.                 delay(500);
395.             }
396.             // Servo_1_Angle(servo_Init + wheel_Steering + deviation); // turn left backward
397.             Servo_Angle(1, servo_Init + deviation + wheel_Steering); // turn left backward
398.             Motor(1,-1,Speed);
399.             Motor(2,-1,Speed);
400.             delay(400);
401.         }
402.     else{
403.         if (midDist < minDist){
404.             // Servo_1_Angle(servo_Init+ deviation); // backward
405.             Servo_Angle(1, servo_Init + deviation); // backward
406.             Motor(1,-1,Speed);
407.             Motor(2,-1,Speed);
408.             delay(500);
409.         }
410.         // Servo_1_Angle(servo_Init + wheel_Steering + deviation); // turn left backward
411.         Servo_Angle(1, servo_Init + deviation + wheel_Steering); // turn left backward
412.         Motor(1,-1,Speed);
413.         Motor(2,-1,Speed);
414.         delay(400);
415.     }
416. }
417. }
418. // delay(100);
419. }
420. }
421.
422. void Light_Tracking(){
423.     value_Init = GetPhotosensitive();
424.     while (1){
425.         StopFunction();
```

```
426.     if (Function_Stop_flag == 2){ // Press OK, stop function.
427.         break;
428.     }
429.     value = GetPhotosensitive();
430.     if (value < (value_Init - threshold)){
431.         // Servo_1_Angle(servo_Init + wheel_Steering + deviation);
432.         Servo_Angle(1, servo_Init + deviation + wheel_Steering);
433.         Motor(1, 1, motor_speed);
434.         Motor(2, 1, motor_speed);
435.         Serial.print(value_Init);
436.         Serial.print(":");
437.         Serial.println(value);
438.     }
439.     else if (value > (value_Init + threshold)){
440.         // Servo_1_Angle(servo_Init - wheel_Steering + deviation);
441.         Servo_Angle(1, servo_Init + deviation - wheel_Steering);
442.         Motor(1, 1, motor_speed);
443.         Motor(2, 1, motor_speed);
444.         Serial.print(value_Init);
445.         Serial.print(":");
446.         Serial.println(value);
447.     }
448.     else{
449.         // Servo_1_Angle(servo_Init);
450.         Servo_Angle(1, servo_Init + deviation);
451.         Motor(1, 1, 0);
452.         Motor(2, 1, 0);
453.         Serial.print(value_Init);
454.         Serial.print(":");
455.         Serial.println(value);
456.     }
457. }
458. }
```