

Lesson 15 How to use APP to Control the Car

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

15.1 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.

15.2 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.

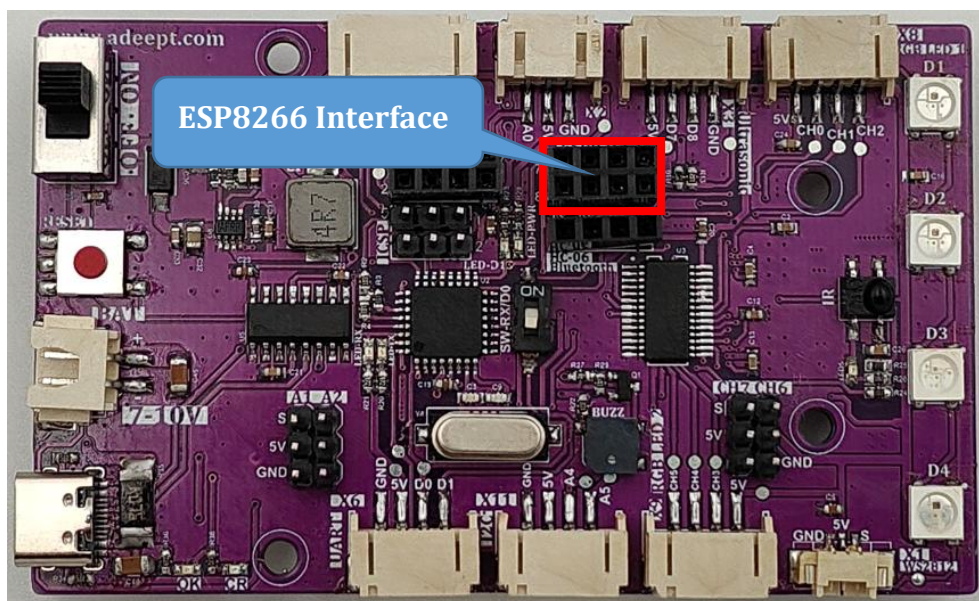
AP mode

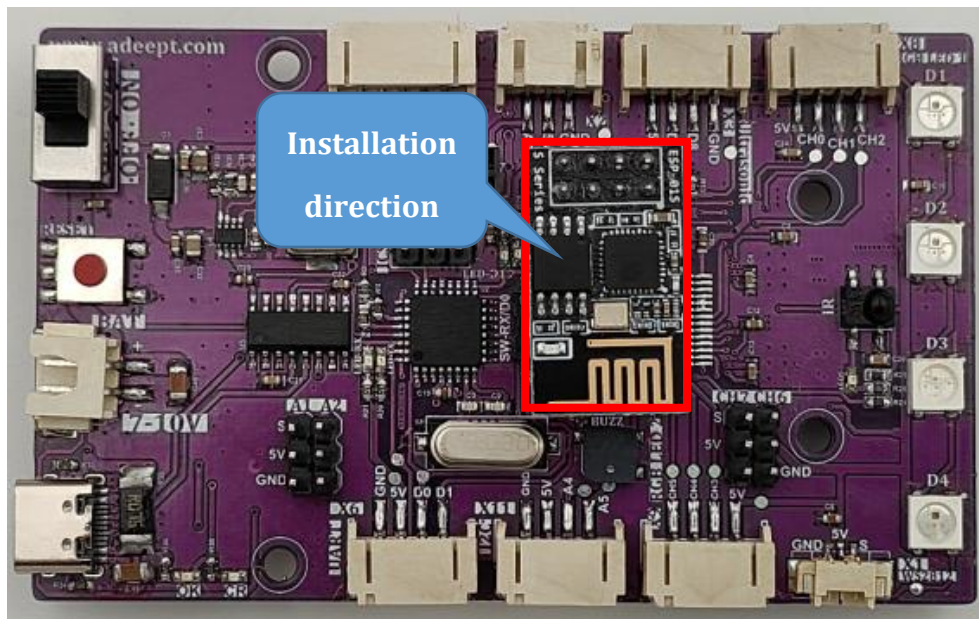
When ESP8266 selects AP mode, it creates a Hotspot network that is separate from the Internet and waits for other WiFi devices to connect. ESP8266 is used as a Hotspot. If a mobile phone or PC wants to communicate with ESP8266, it must be connected to the Hotspot of ESP8266. Only after a connection is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

15.3 Install of ESP8266 module





15.4 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.

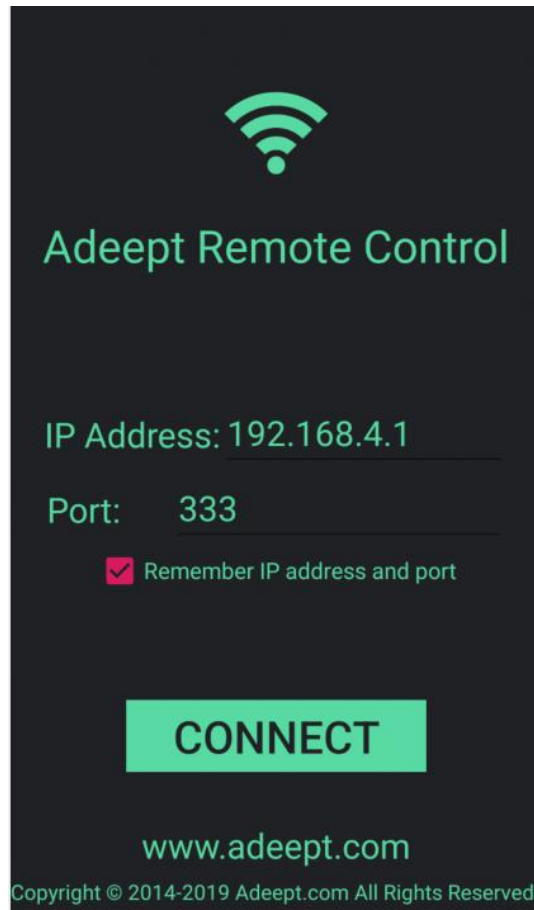
Link: <https://www.adeept.com/learn/detail-41.html>

Icon after successful installation:



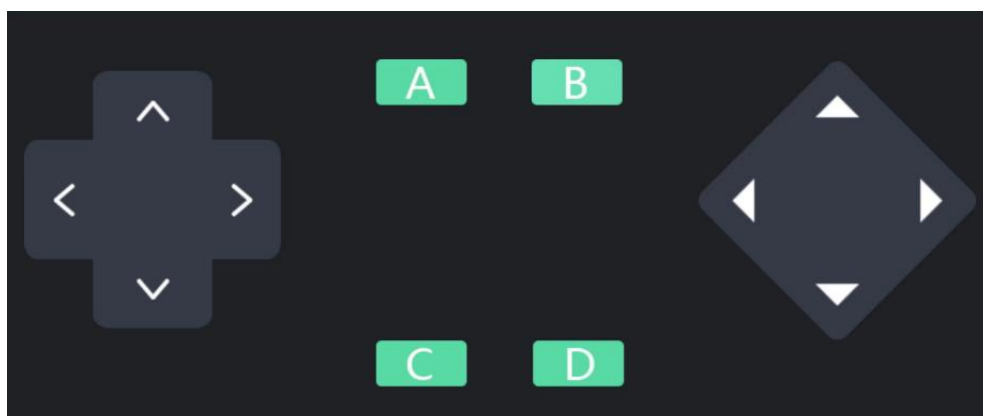
Connect to WiFi. Open the mobile app, enter the IP address of the ESP8266 in the IP address field of the mobile app, and enter port number. Click Connect.

Login interface



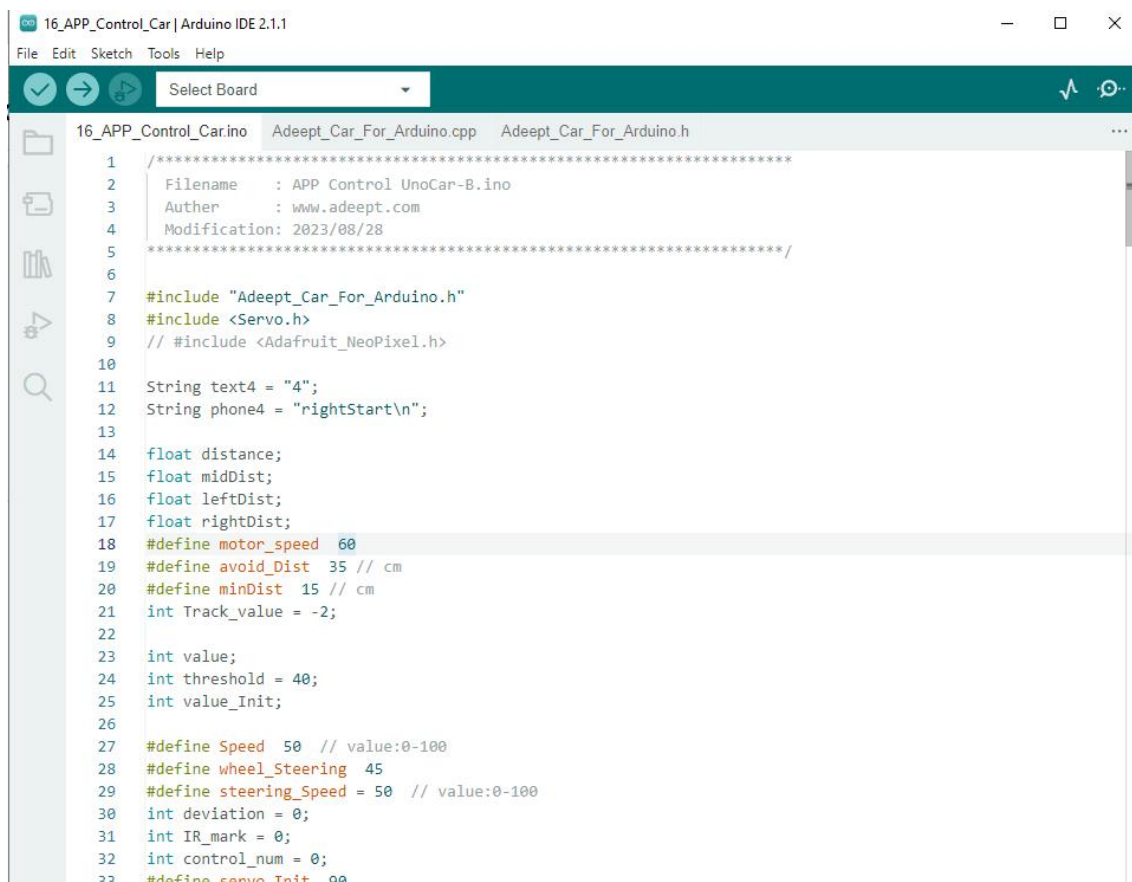
The login interface of the Adeept Remote Control app. It features a dark background with a green Wi-Fi icon at the top. Below the icon, the text "Adeept Remote Control" is displayed in green. There are two input fields: "IP Address: 192.168.4.1" and "Port: 333". A checkbox labeled "Remember IP address 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)



15.3 Control the car in AP mode

1. Connect your computer and Adeept Robot Control Board with a USB cable.
2. Open “15_APP_Control_car” folder in “/Code”, double-click “15_APP_Control_car.ino”.



```
16_APP_Control_Car | Arduino IDE 2.1.1
File Edit Sketch Tools Help
Select Board

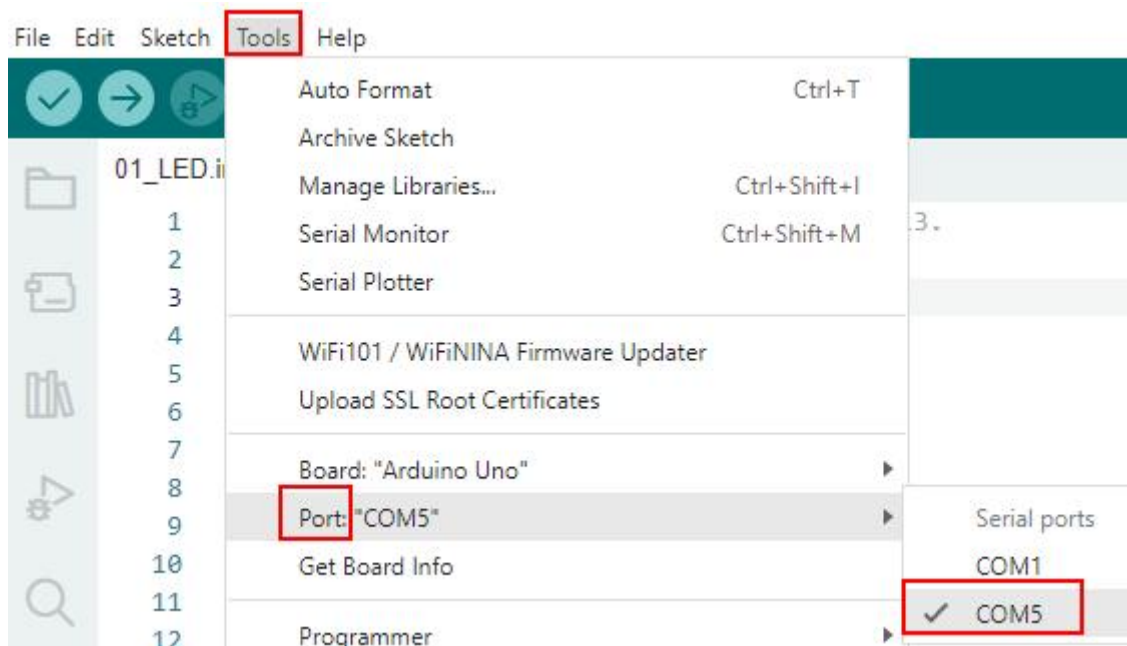
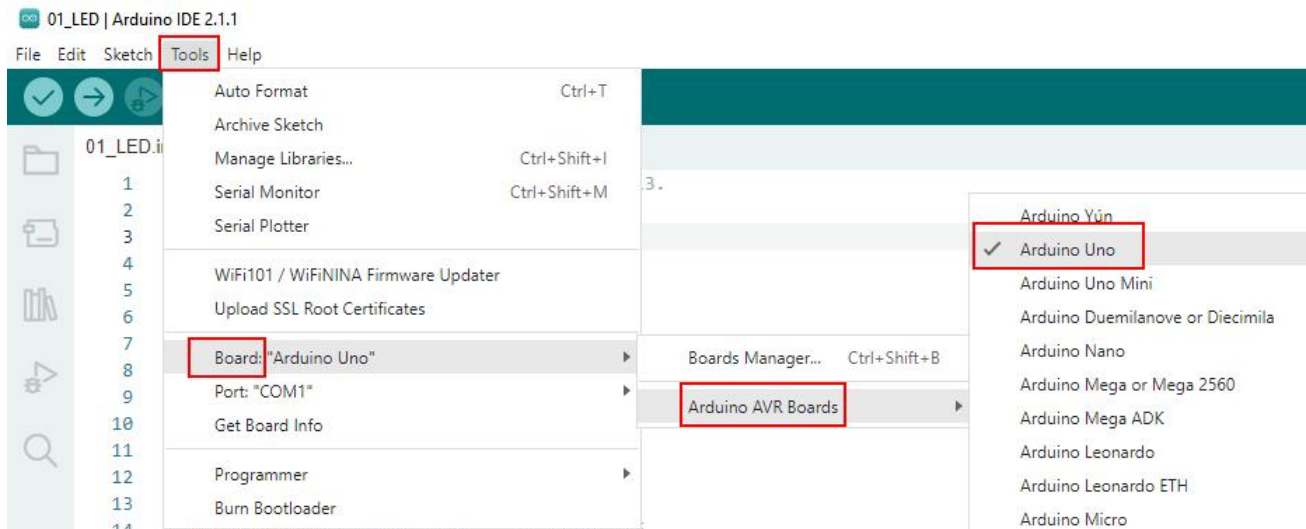
16_APP_Control_Car.ino Adeept_Car_For_Arduino.cpp Adeept_Car_For_Arduino.h
1  /*****
2  Filename   : APP_Control_UnoCar-B.ino
3  Author    : www.adeept.com
4  Modification: 2023/08/28
5  *****/
6
7  #include "Adeept_Car_For_Arduino.h"
8  #include <Servo.h>
9  // #include <Adafruit_NeoPixel.h>
10
11  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.

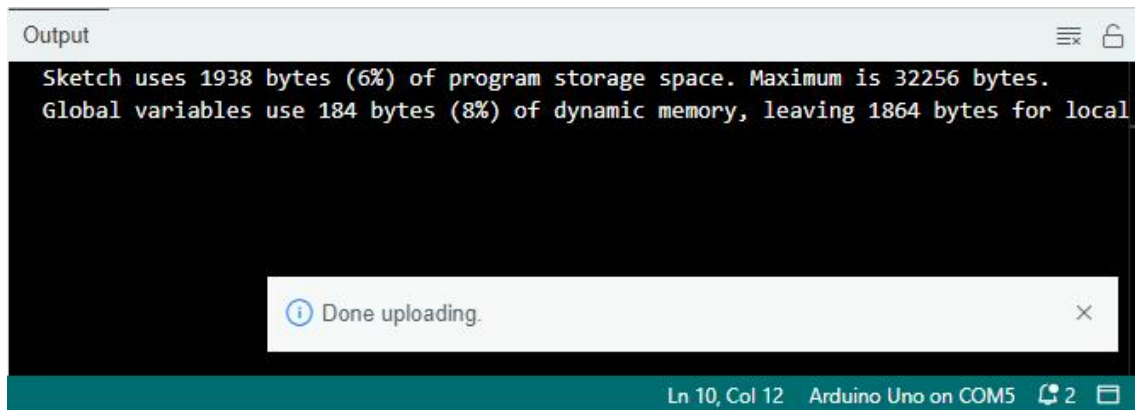
Board: **Tools**---->**Board**---->**Arduino AVR Boards**---->**Arduino Uno**

Port: **Tools** ---->**Port**---->**COMx**

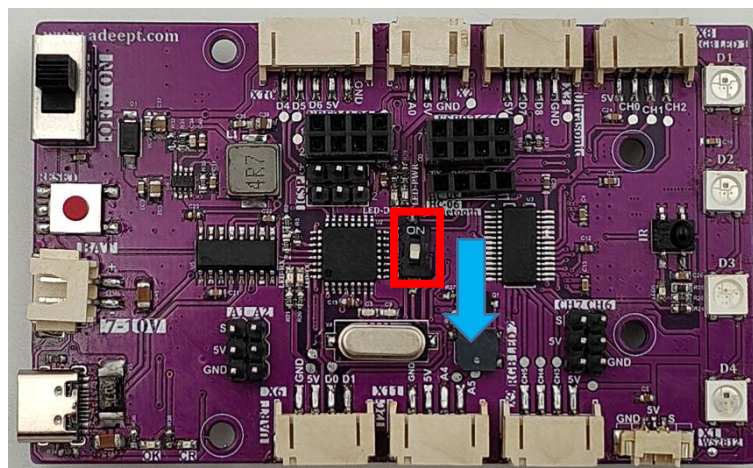
Note: The port number will be different in different computers.



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.



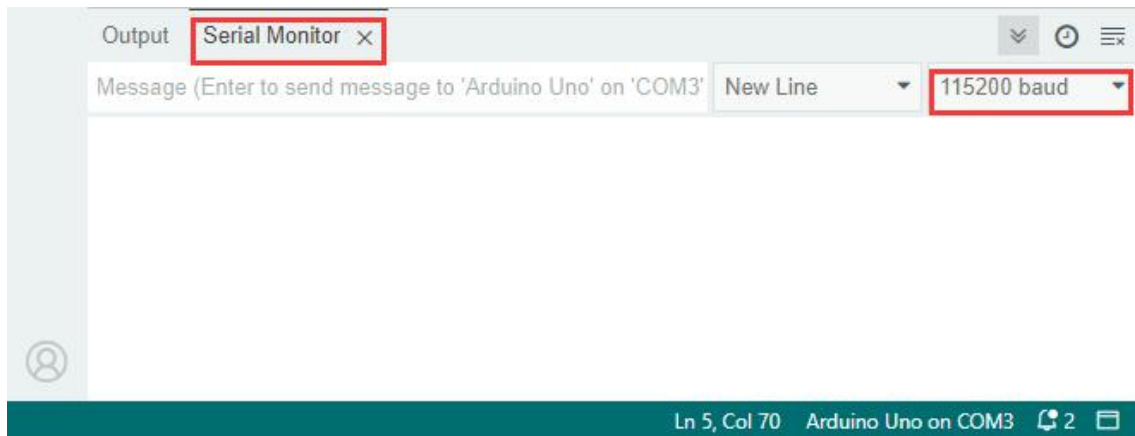
Note: If the upload program fails, a red error message appears. Please confirm whether the RX/D0 switch is in the correct position. Toggle down the paddle (white) of the switch in the picture below.



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.

5. Click Serial Monitor, Set the baud rate as 115200.

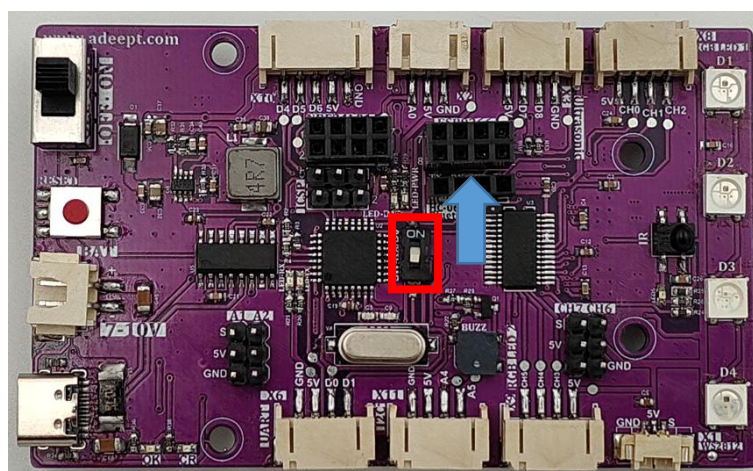




Note: When the ESP8266 module transmits data to the Arduino, it needs to occupy the RX interface of the Arduino, and when the Arduino uploads the program, it also needs to occupy the RX interface. The RX interface cannot satisfy both functions at the same time, so a switch is needed to distinguish them.

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.

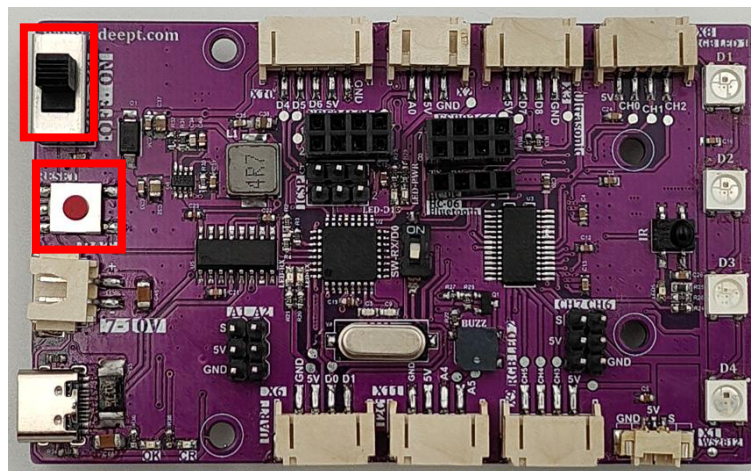
Now we need the communication function of ESP8266, so we need to toggle the switch up.



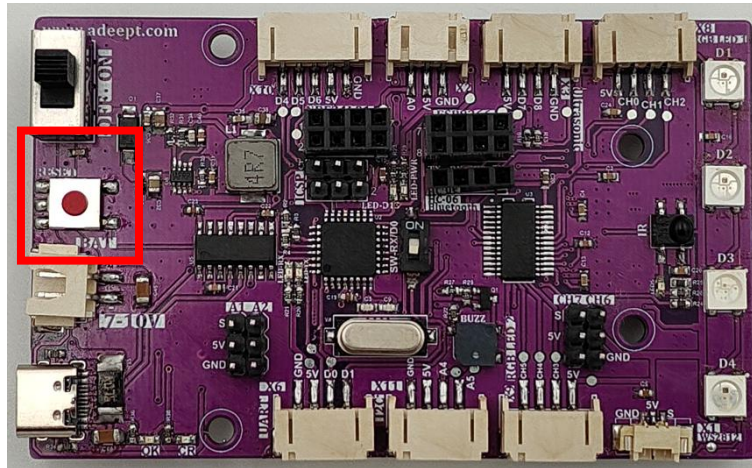
If you need to upload other programs later, and ESP8266 is installed on Adeept Robot Control Board, you need to flip the switch down to disconnect RX from ESP8266.

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
```

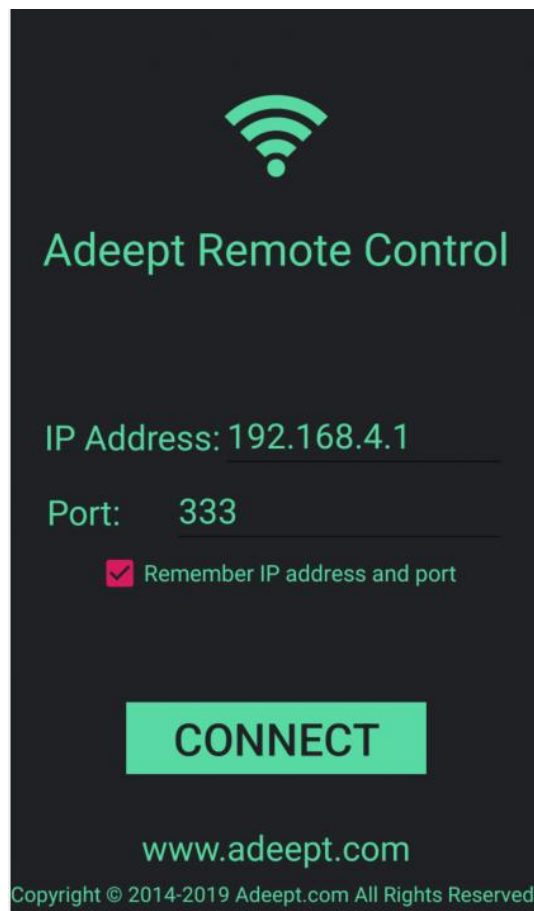
8. Use an Android phone to connect to "Adeept_ESP8266" WiFi. Since this WiFi can only be used for communication between the mobile phone and ESP8266, after the mobile phone is connected to WiFi, it cannot access the external network (you cannot use the mobile phone to access the Internet), and the mobile 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 Uno-Car 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: 192.158.4.1

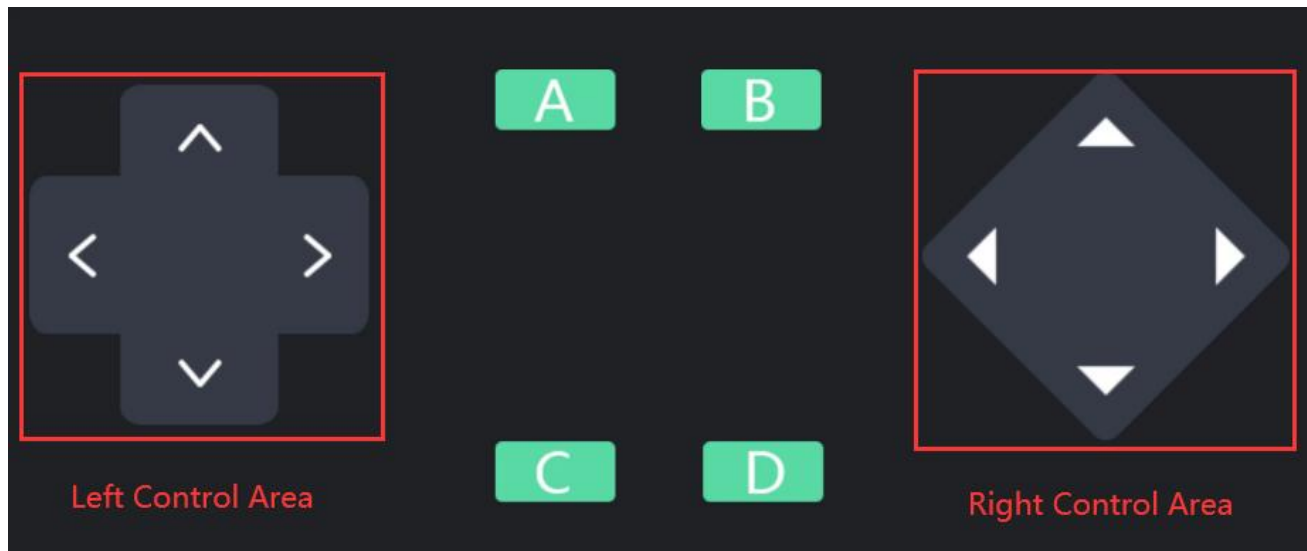
Port: 333











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.

15.5 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 [15_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 [15_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.  String text4 = "4";
6.  String phone4 = "rightStart\n";
7.
8.  float distance;
9.  float midDist;
10. float leftDist;
11. float rightDist;
12. #define motor_speed 60
13. #define avoid_Dist 35 // cm
14. #define minDist 15 // cm
15. int Track_value = -2;
16.
17. int value;
18. int threshold = 40;
19. int value_Init;
20.
21. #define Speed 50 // value:0-100
22. #define wheel_Steering 45
23. #define steering_Speed = 50 // value:0-100
24. int deviation = 0;
25. int IR_mark = 0;
```

```
26. int control_num = 0;
27. #define servo_Init 90
28. int servo_Angle2 = servo_Init;
29. int ws2812_flag= 0;
30. int Function_Stop_flag = 0;
31.
32.
33. String Move_UP1 = "forwardStart\n";
34. String Move_UP2 = "forwardStop\n";
35. String Move_Down1 = "backwardStart\n";
36. String Move_Down2 = "backwardStop\n";
37. String Move_Left1 = "leftStart\n";
38. String Move_Left2 = "leftStop\n";
39. String Move_Right1 = "rightStart\n";
40. String Move_Right2 = "rightStop\n";
41.
42. String Head_UP1 = "lookLeftStart\n";
43. String Head_UP2 = "lookLeftStop\n";
44. String Head_Down1 = "lookRightStart\n";
45. String Head_Down2 = "lookRightStop\n";
46. String Head_Left1 = "downStart\n";
47. String Head_Left2 = "downStop\n";
48. String Head_Right1 = "upStart\n";
49. String Head_Right2 = "upStop\n";
50.
51.
52. String Ultrasonic_ON = "aStart\n";
53. String Ultrasonic_ON2 = "aStop\n";
54. String Ultrasonic_OFF = "bStart\n";
55. // String Ultrasonic_OFF = "\r\n+IPD,4,7:bStart";
56. String Ultrasonic_OFF2 = "bStop\n";
57.
58. String Light_Tracking_ON = "cStart\n";
59. String Light_Tracking_ON2 = "cStop\n";
60. String Light_Tracking_OFF = "dStart\n";
61. // String Light_Tracking_OFF = "\r\n+IPD,4,7:dStart\n";
62. String Light_Tracking_OFF2 = "dStop\n";
63.
64. String comdata = "";
65. int judge;
66.
67. void setup()
```

```

68. {
69.   Serial.begin(115200);           // set up a wifi serial communication baud rate 1
    15200
70.
71.   Serial.println("AT+CWMODE=3\r\n");//set to softAP+station mode
72.   delay(3000);           //delay 4s
73.   Serial.println("AT+CWSAP=\"Adeept_ESP8266\", \"12345678\", 8, 2\r\n"); //TCP P
    rotocol, server IP addr, port
74.   delay(1000);           //delay 4s
75.   Serial.println("AT+RST\r\n"); //reset wifi
76.   delay(1000);           //delay 4s
77.   Serial.println("AT+CIPMUX=1\r\n");//set to multi-connection mode
78.   delay(1000);
79. // Serial.println("AT+CIPSERVER=1,333\r\n");//set as server
80.   Serial.println("AT+CIPSERVER=1\r\n");//set as server
81.   delay(1000);
82.   Serial.println("AT+CIPSTO=7000\r\n");//keep the wifi connecting 7000 seconds
83.   delay(1000);
84.
85.   Servo_Setup();           //Servo initialization
86.   PCA9685_Servo_Setup(); //PCA9685 Servo initialization
87.   Motor_Setup();           //Motor initialization
88.   AllMotorStop();
89.   Buzzer_Setup();           //Buzzer initialization
90.   WS2812_Setup();           //WS2812 LED initialization
91.   WS2812_Brightness(5); // value 0-10
92.   Ultrasonic_Setup();       //Ultrasonic initialization
93.   Photosensitive_Setup();    //Light Line initialization
94.   Tracking_Setup();          //Tracking Line initialization
95.   OLED_Setup();             //OLED initialization
96.   Matrix_Setup();
97.   Matrix_Clear();
98.   Buzzer_Alert(1,1);
99.
100.  WS2812ColorAll(255, 255,0); // Green
101.  Servo_Angle(1, 90);
102.  Servo_Angle(2, 90);
103.  PCA9685_Servo_Angle(6, 0, 90);
104.  PCA9685_Servo_Angle(7, 0, 90);
105.  Buzzer_Silence();
106.  OLED_clear();
107.  delay(1000);

```

```
108. WS2812ColorAll(0,0,0);
109. }
110.
111. void loop()
112. {
113.     while(Serial.available()>0)
114.     {
115.         comdata += char(Serial.read());
116.         delay(1);
117.     }
118.     judgement();
119.     control(judge);
120.
121. }
122.
123.
124. void judgement(){
125.     if (comdata.length() > 0){
126.         if(comdata.endsWith(Move_UP1)){//forward
127.             judge=1;
128.             Serial.println(comdata); //print received data.
129.         }
130.         else if(comdata.endsWith(Move_Down1)){//backward
131.             judge=2;
132.             Serial.println(comdata);
133.         }
134.         else if(comdata.endsWith(Move_Left1)){//left
135.             judge=3;
136.             Serial.println(comdata);
137.         }
138.         else if(comdata.endsWith(Move_Right1)){//right.
139.             judge=4;
140.             Serial.println(comdata);
141.         }
142.         else if(comdata.endsWith(Move_UP2)||comdata.endsWith(Move_Down2)||comdata.endsWith(Move_Left2)||comdata.endsWith(Move_Right2)){//stop
143.             judge=5;
144.             Serial.println(comdata);
145.         }
146.         else if(comdata.endsWith(Head_Left1)){//turn left
147.             judge=6;
148.             Serial.println(comdata);
```

```
149.     }
150.     else if(comdata.endsWith(Head_Right1)){//trun right
151.         judge=7;
152.         Serial.println(comdata);
153.     }
154.     else if(comdata.endsWith(Head_Left2)||comdata.endsWith(Head_Right2)){//
    trun right
155.         judge=8;
156.         Serial.println(comdata);
157.     }
158.
159.     else if(comdata.endsWith(Ultrasonic_ON)){//avoid obstacles function.
160.         judge=9;
161.         Serial.println(comdata);
162.     }
163.     else if(comdata.endsWith(Light_Tracking_ON)){//light tracking function.
164.         judge=10;
165.         Serial.println(comdata);
166.     }
167.
168.     else if(comdata.endsWith(Ultrasonic_OFF)){//avoid obstacles function. b
    start.
169.         judge=11;
170.         Serial.println(comdata);
171.     }
172.     else if(comdata.endsWith(Light_Tracking_OFF)){//light tracking function
    . dstart
173.         judge=12;
174.         Serial.println(comdata);
175.     }
176.     comdata = "";
177.     delay(10);
178. }
179.
180. }
181.
182.
183. void control(int value){
184.     switch (value) {
185.         case 1: // forward
186.             Motor(1, 1, motor_speed); //Motor1 forward
187.             Motor(2, 1, motor_speed); //Motor2 forward
```



```
188.     Motor(3, 1, motor_speed); //Motor3 forward
189.     Motor(4, 1, motor_speed); //Motor4 forward
190.     // control_num = 12;
191.     break;
192.
193.     case 2: // Down,
194.         Motor(1, -1, motor_speed); //Motor1 backward
195.         Motor(2, -1, motor_speed); //Motor2 backward
196.         Motor(3, -1, motor_speed); //Motor3 backward
197.         Motor(4, -1, motor_speed); //Motor4 backward
198.         // control_num = 13;
199.         break;
200.
201.     case 3: // left
202.         Motor(1, -1, motor_speed);
203.         Motor(2, -1, motor_speed);
204.         Motor(3, 1, motor_speed);
205.         Motor(4, 1, motor_speed);
206.         // control_num = 14;
207.         break;
208.
209.     case 4: // right
210.         Motor(1, 1, motor_speed);
211.         Motor(2, 1, motor_speed);
212.         Motor(3, -1, motor_speed);
213.         Motor(4, -1, motor_speed);
214.         // control_num = 15;
215.         break;
216.
217.     case 5: // stop
218.         Motor(1, 1, 0);
219.         Motor(2, 1, 0);
220.         Motor(3, 1, 0);
221.         Motor(4, 1, 0);
222.         // control_num = -1;
223.
224.         break;
225.
226.     case 6: // trun left
227.         servo_Angle2 = servo_Angle2 + 1;
228.         if (servo_Angle2 > 180){
229.             servo_Angle2 = 180;
```

```
230.     }
231.     Servo_Angle(2, servo_Angle2);
232.     // control_num = 15;
233.     delay(10);
234.     break;
235. case 7: // trun right
236.     servo_Angle2 = servo_Angle2 - 1;
237.     if (servo_Angle2 < 0){
238.         servo_Angle2 = 0;
239.     }
240.     Servo_Angle(2, servo_Angle2);
241.     // control_num = 15;
242.     delay(10);
243.     break;
244.
245. case 8: // stop servo rotation.
246.     break;
247.
248. case 9:
249.     Function_Stop_flag = 0;
250.     Avoid_Obstacles(); // Avoid Obstacles function
251.     break;
252.
253. case 10:
254.     Function_Stop_flag = 0;
255.     Light_Tracking(); // Light Tracking function
256.     break;
257.
258. case 11:
259.     Motor(1, 1, 0);
260.     Motor(2, 1, 0);
261.     Motor(3, 1, 0);
262.     Motor(4, 1, 0);
263.     break;
264.
265. case 12:
266.     Motor(1, 1, 0);
267.     Motor(2, 1, 0);
268.     Motor(3, 1, 0);
269.     Motor(4, 1, 0);
270.     break;
271.
```

```
272.     default:
273.         break;
274.     }
275. }
276.
277. int StopFunction(){
278.     while(Serial.available()>0){
279.         comdata += char(Serial.read());
280.         delay(1);
281.     }
282.     if (comdata.length() > 0){
283.         if(comdata.endsWith(Ultrasonic_OFF)){// Stop Avoid Obstacles function.
284.             Function_Stop_flag = 1;
285.             judge=11;
286.         }
287.         else if(comdata.endsWith(Light_Tracking_OFF)){ // Stop Light Tracking function.
288.             Function_Stop_flag = 2;
289.             judge=12;
290.         }
291.         comdata = "";
292.         delay(10);
293.     }
294. }
295.
296. void Avoid_Obstacles(){
297.     while (1){
298.         StopFunction();
299.         if (Function_Stop_flag == 1){ // Press OK, stop function.
300.             break;
301.         }
302.
303.         distance = GetDistance();
304.         Serial.print(distance);
305.         // Motor(1,1,0); //Stop the car
306.         // Motor(2,1,0);
307.         // Motor(3,1,0); //Stop the car
308.         // Motor(4,1,0);
309.
310.         if (distance > 30){
311.             Servo_Angle(2, 95);
312.             Motor(1,1,Speed); //forward
```

```
313.      Motor(2,1,Speed);
314.      Motor(3,1,Speed); //forward
315.      Motor(4,1,Speed);
316.  }
317.  else if (distance >= 10 and distance <=30){
318.      Motor(1,1,0); //Stop the car
319.      Motor(2,1,0);
320.      Motor(3,1,0); //Stop the car
321.      Motor(4,1,0);
322.      Servo_Angle(2, 50);
323.      if (distance > 20){
324.          Motor(1,-1,Speed); //Stop the car
325.          Motor(2,-1,Speed);
326.          Motor(3,1,Speed); //Stop the car
327.          Motor(4,1,Speed);
328.          // delay(2000);
329.      }
330.      else{
331.          Servo_Angle(2, 150);
332.          Motor(1,1,Speed); //Stop the car
333.          Motor(2,1,Speed);
334.          Motor(3,-1,Speed); //Stop the car
335.          Motor(4,-1,Speed);
336.          // delay(2000);
337.      }
338.  }
339. }
340. }
341.
342.
343. void Light_Tracking(){
344.     value_Init = GetPhotosensitive();
345.     while (1){
346.         StopFunction();
347.         if (Function_Stop_flag == 2){ // Press OK, stop function.
348.             break;
349.         }
350.         value = GetPhotosensitive();
351.         if (value < (value_Init - threshold)){
352.             Motor(1, -1, motor_speed);
353.             Motor(2, -1, motor_speed);
354.             Motor(3, 1, motor_speed);
```

```
355.     Motor(4, 1, motor_speed);
356.     Serial.print(value_Init);
357.     Serial.print(":");
358.     Serial.println(value);
359.
360. }
361. else if (value > (value_Init + threshold)){
362.     Motor(1, 1, motor_speed);
363.     Motor(2, 1, motor_speed);
364.     Motor(3, -1, motor_speed);
365.     Motor(4, -1, motor_speed);
366.     Serial.print(value_Init);
367.     Serial.print(":");
368.     Serial.println(value);
369. }
370. else{
371.     Motor(1, 1, motor_speed);
372.     Motor(2, 1, motor_speed);
373.     Motor(3, 1, motor_speed);
374.     Motor(4, 1, motor_speed);
375.     Serial.print(value_Init);
376.     Serial.print(":");
377.     Serial.println(value);
378. }
379. }
380.
381. }
```