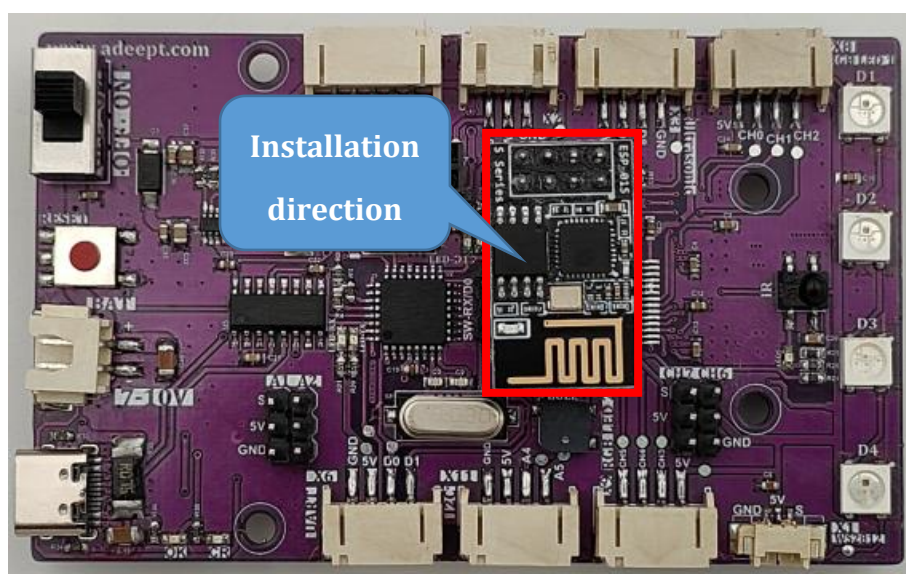
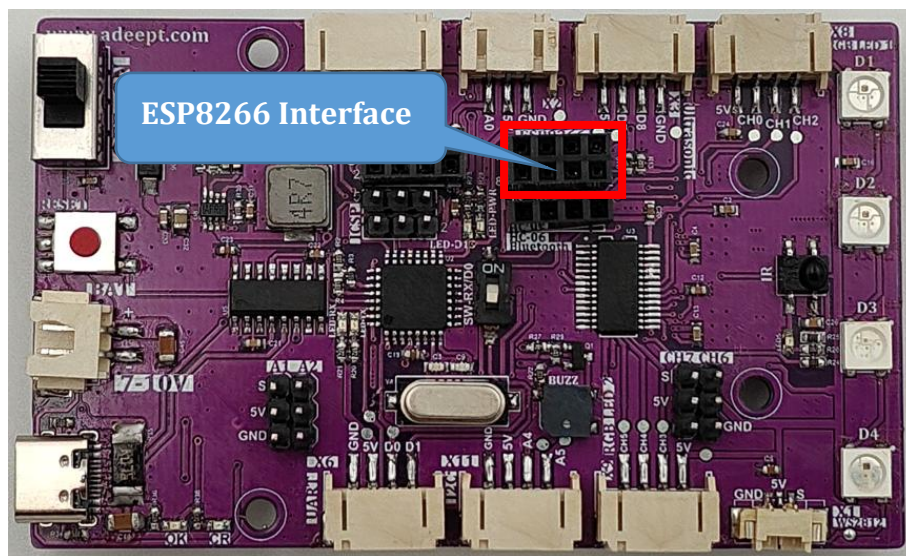


Lesson 17 How to use GUI to Control the Car

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

The principles and usage of computers and mobile phones are basically the same.

17.1 Install of ESP8266 module



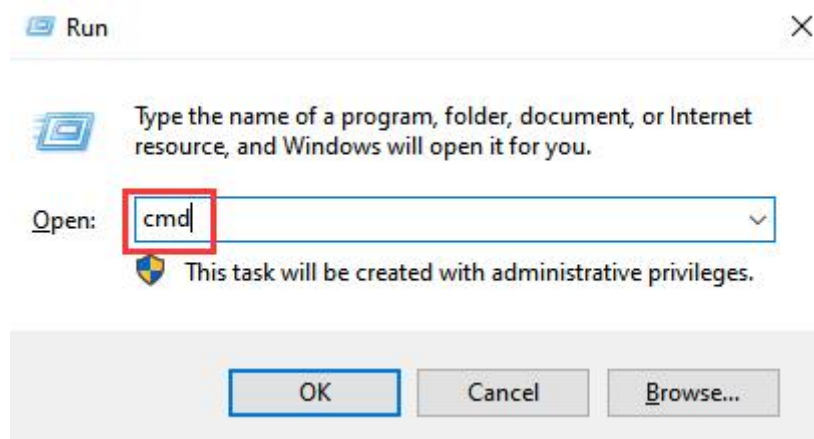
17.2 Configure the usage environment on the PC

The GUI program now is compatible with **Windows** and **MAC** operating system. Generally, it's named **GUI.py** and stored under the client directory of the robot software package.

- You need to install Python on your computer to run the program for PC. Since the code of this robot was developed and debugged by Python3, please download the Python3.7 version or higher in case of any error caused by incompatibility.
- Download Python3 at this link: <https://www.python.org/downloads/windows/>
- Double click the installation package to install Python.
- Pay attention to select **Add Python to PATH** when installing.

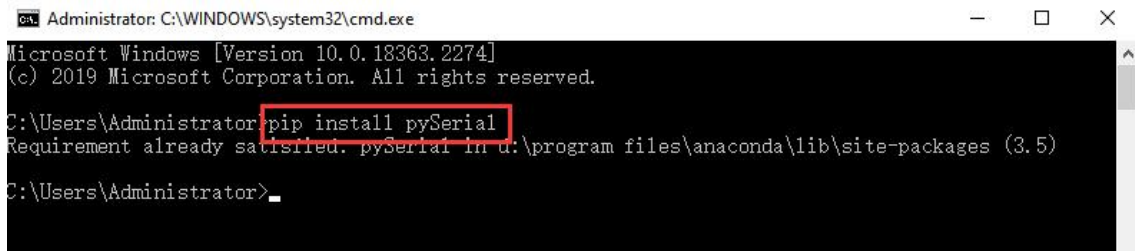
Installing pySerial

- pySerial encapsulates the serial communication module, supporting Linux, Windows, BSD (may support all operating systems that support POSIX), Jython (Java) and IronPython (.NET and Mono). The pySerial module encapsulates access to the serial port.
- Press Win + R key, type in "cmd", and click OK to start cmd. (MAC users do it in the terminal.)



- Type in the command below to install pySerial:

```
pip install pySerial
```



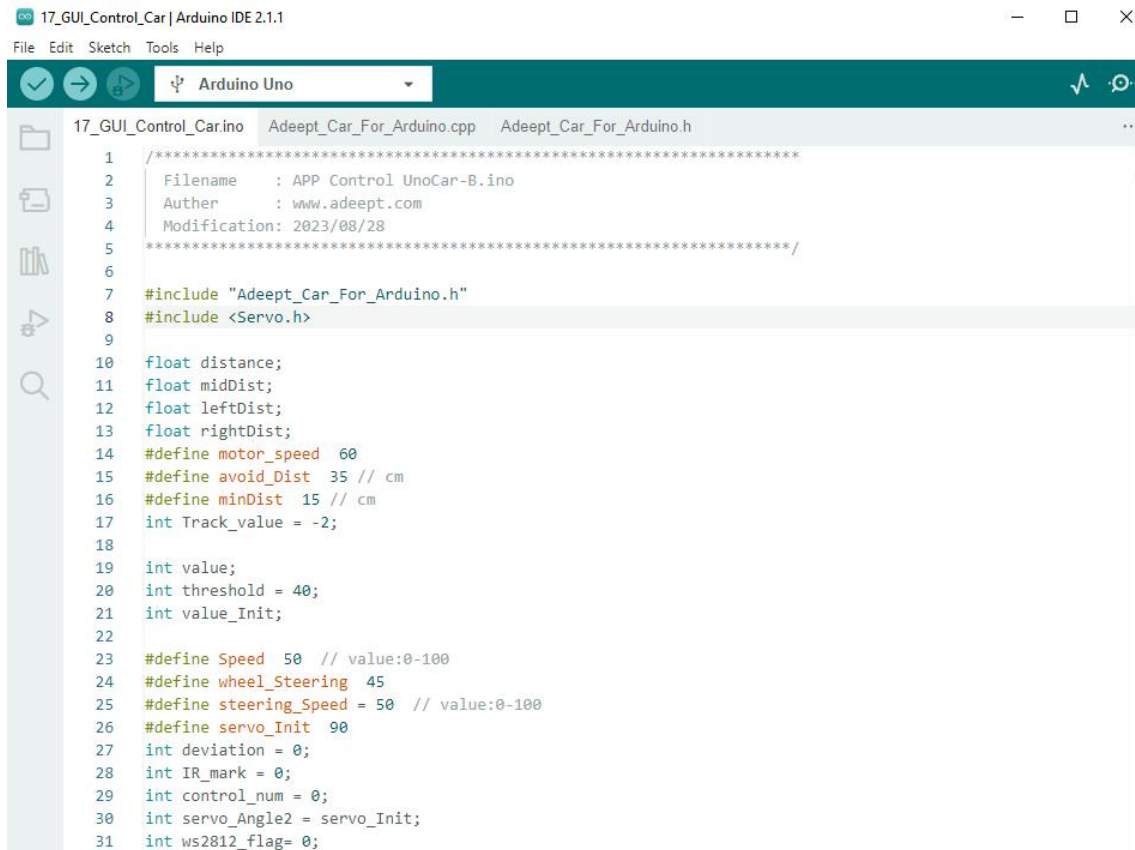
```
Administrator: C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.18363.2274]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>pip install pySerial
Requirement already satisfied: pySerial in d:\program files\anaconda\lib\site-packages (3.5)
C:\Users\Administrator>
```

- Press Enter to start downloading and installing pySerial .

17.3 Control the car in AP mode

1. Connect your computer and Adeept Robot Control Board with a USB cable.
2. Open “17_GUI_Control_Car” folder in “[Adeept_UnoCar-B/Code](#)”, double-click “[17_GUI_Control_Car.ino](#)”.



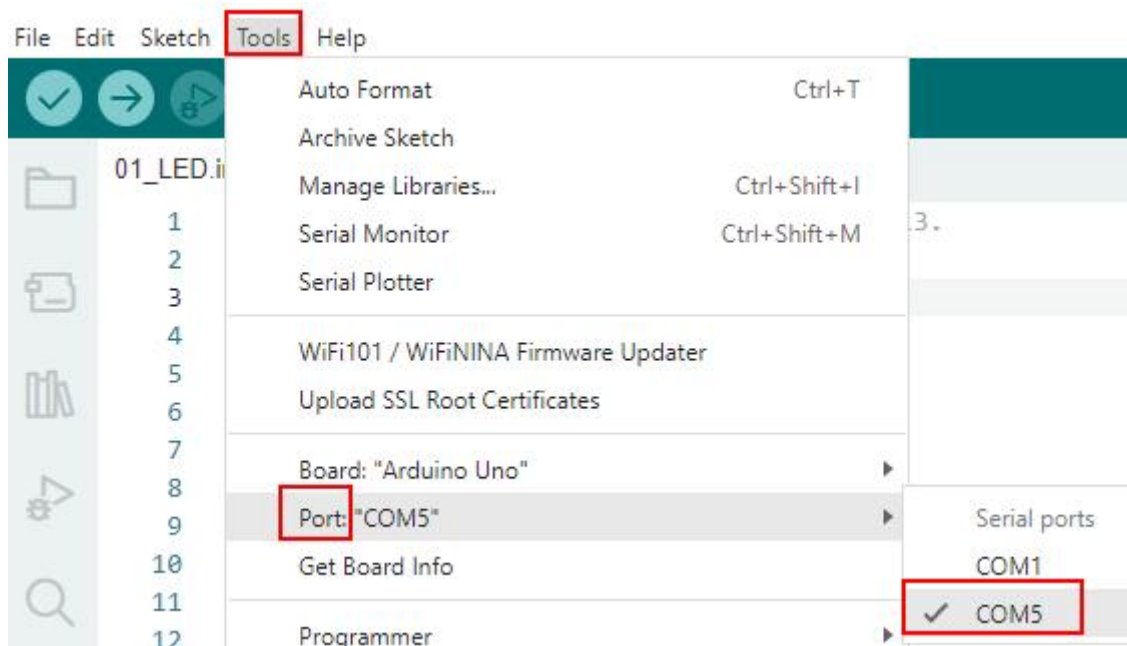
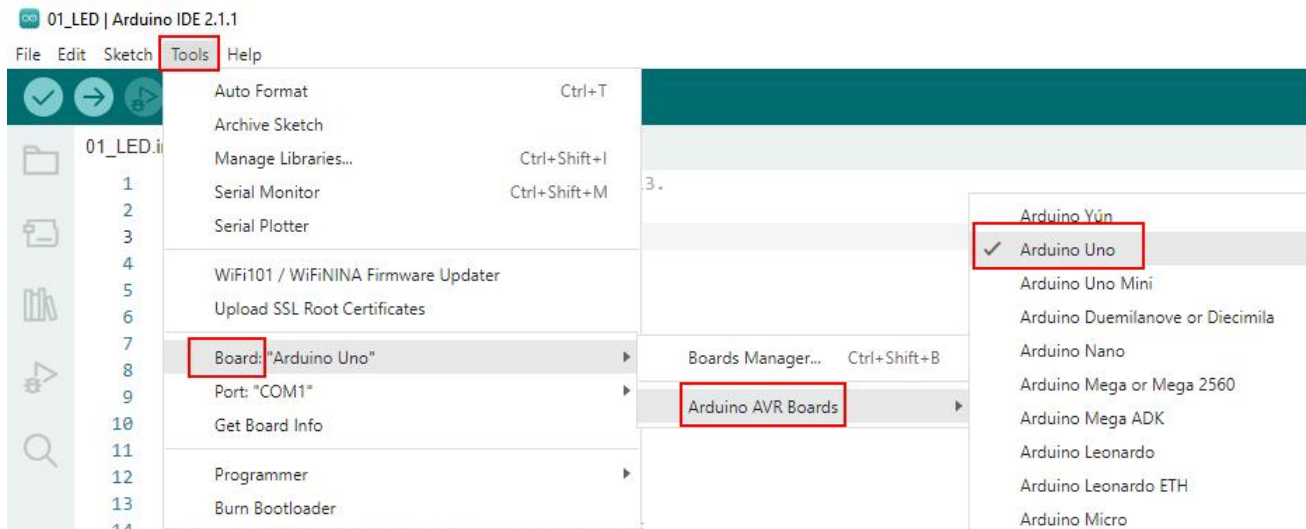
```
17_GUI_Control_Car.ino | Arduino IDE 2.1.1
File Edit Sketch Tools Help
Arduino Uno
17_GUI_Control_Car.ino Adeept_Car_For_Arduino.cpp Adeept_Car_For_Arduino.h
1  /*****
2  Filename   : APP_Control_UnoCar-B.ino
3  Auther    : www.adeept.com
4  Modification: 2023/08/28
5  *****/
6
7  #include "Adeept_Car_For_Arduino.h"
8  #include <Servo.h>
9
10 float distance;
11 float midDist;
12 float leftDist;
13 float rightDist;
14 #define motor_speed 60
15 #define avoid_Dist 35 // cm
16 #define minDist 15 // cm
17 int Track_value = -2;
18
19 int value;
20 int threshold = 40;
21 int value_Init;
22
23 #define Speed 50 // value:0-100
24 #define wheel_Steering 45
25 #define steering_Speed = 50 // value:0-100
26 #define servo_Init 90
27 int deviation = 0;
28 int IR_mark = 0;
29 int control_num = 0;
30 int servo_Angle2 = servo_Init;
31 int ws2812_flag= 0;
```


3. Select development board and serial port.

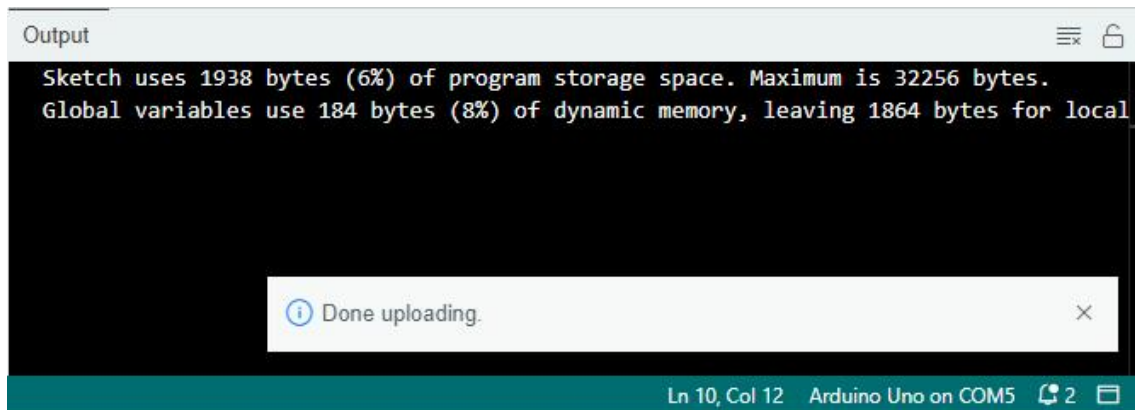
Board: **Tools**---->**Board**---->**Arduino AVR Boards**---->**Arduino Uno**

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

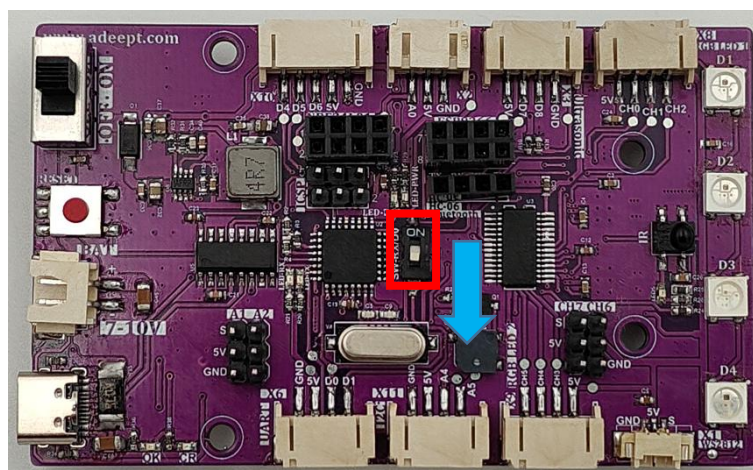
Note: The port number will be different in different computers.



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.



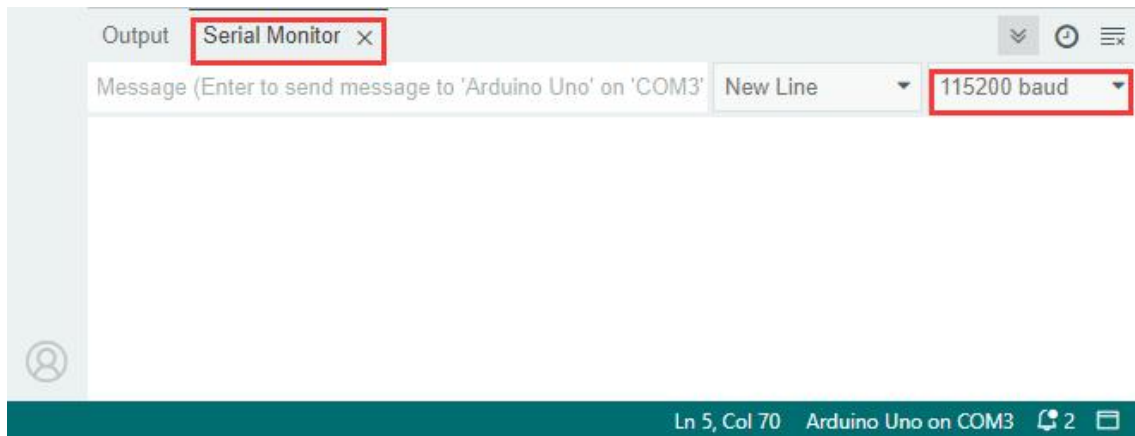
Note: If the upload program fails, a red error message appears. Please confirm whether the RX/D0 switch is in the correct position. Toggle down the paddle (white) of the switch in the picture below.



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.

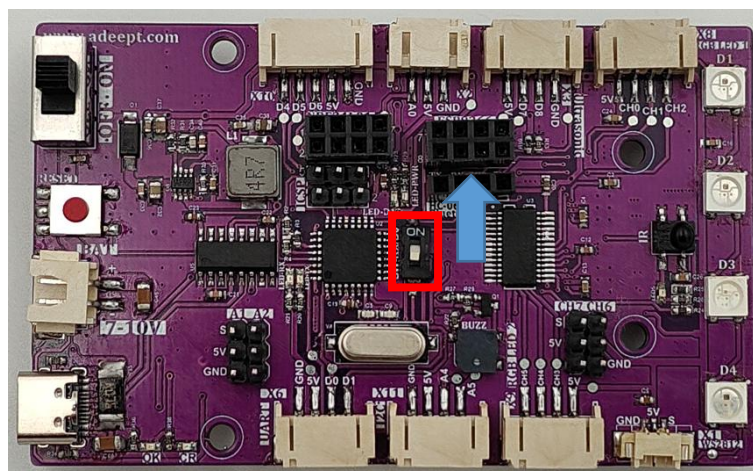




Note: When the ESP8266 module transmits data to the Arduino, it needs to occupy the RX interface of the Arduino, and when the Arduino uploads the program, it also needs to occupy the RX interface. The RX interface cannot satisfy both functions at the same time, so a switch is needed to distinguish them.

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.

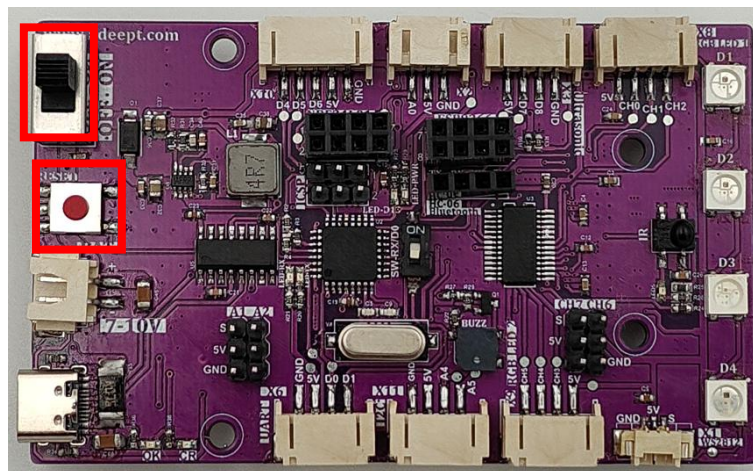
Now we need the communication function of ESP8266, so we need to toggle the switch up.



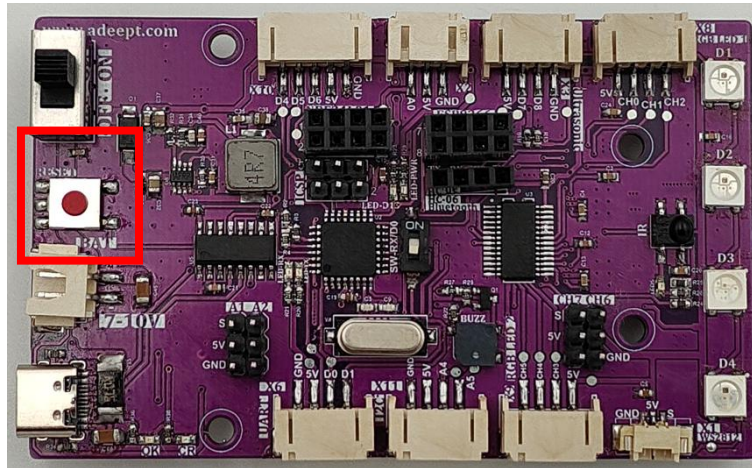
If you need to upload other programs later, and ESP8266 is installed on Adeept Robot Control Board, you need to flip the switch down to disconnect RX from ESP8266.

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 GUI.



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.

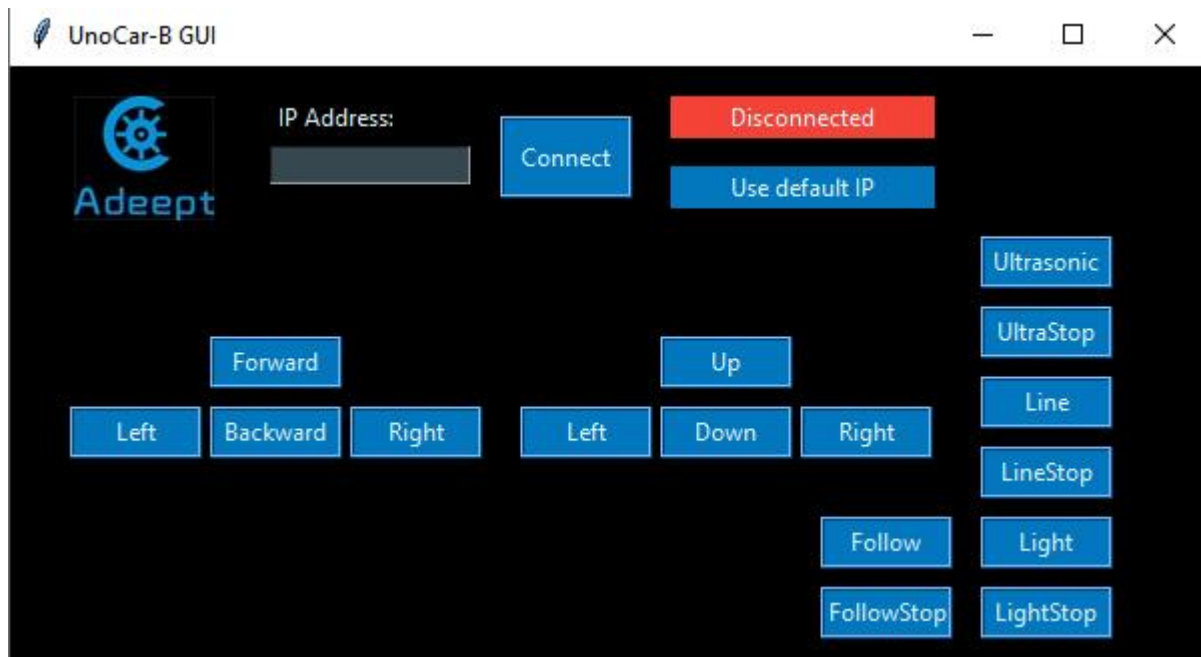
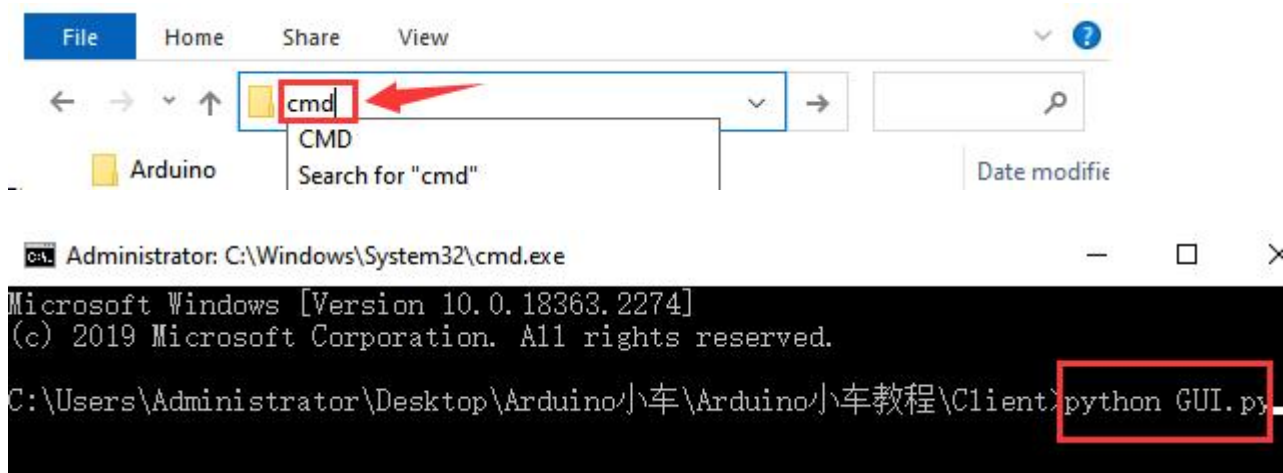
📶 Adeept_ESP8266

```
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 PC to connect to "Adeept_ESP8266" WiFi. Since this WiFi can only be used for communication between the PC and ESP8266, after the PC is connected to WiFi, it cannot access the external network (you cannot use the PC to access the Internet).

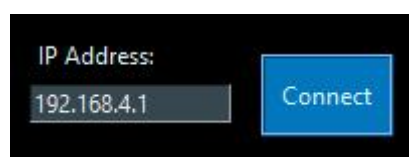
9. Enter the Client folder, double-click GUI.py to run the program.

10. Or enter the [Client](#) folder in cmd, then run the GUI.py program in CMD. (MAC users do it in the terminal.)











11. Open the GUI, and enter the IP address in AP mode. Then click "Connect". The IP address remains unchanged.

IP Address: 192.168.4.1



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



Left Control Area	keyboard keys	Function	Right Control Area	keyboard keys	Function
	W	Forward		I	/
	S	Backward		K	/
	A	Turn Left		J	Head Turn Left
	D	Turn Right		L	Head Turn Right



	Function		Function
Ultrasonic	Ultrasonic Avoid Obstacles Function	Light	Light Tracking Function
UltraStop	Stop Avoid Obstacles (click multiple times)	LightStop	Stop Light Tracking (click multiple times)
Line	Line Tracking Function	Follow	Ultrasonic Follow Function
LineStop	Stop Line Tracking (click multiple times)	FollowStop	Stop Ultrasonic Follow
Some functions may require multiple clicks to stop.			

17.5 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 [17_GUI_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 [17_GUI_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.
4. float distance;
5. float midDist;
6. float leftDist;
7. float rightDist;
8. #define motor_speed 60
9. #define avoid_Dist 35 // cm
10. #define minDist 15 // cm
11. int Track_value = -2;
12.
13. int value;
14. int threshold = 40;
15. int value_Init;
16.
17. #define Speed 50 // value:0-100
18. #define wheel_Steering 45
19. #define steering_Speed = 50 // value:0-100
20. #define servo_Init 90
21. int deviation = 0;
22. int IR_mark = 0;
23. int control_num = 0;
24. int servo_Angle2 = servo_Init;
25. int ws2812_flag= 0;
26. int Function_Stop_flag = 0;
27.
28.
29. const String Move_UP = "forward";
30. const String Move_Down = "backward";
31. const String Move_UD_Stop = "DS";
32. const String Move_Left = "turn_left";
33. const String Move_LR_Stop = "TS";
34. const String Move_Right = "turn_right";
35.
36. const String Head_UP = "up";
37. const String Head_Down = "down";
38. const String Head_Left = "lookleft";
39. const String Head_Right = "lookright";
40. const String Head_Stop = "stop";
41.
42. const String Ultrasonic_ON = "Ultra_Start";
```



```
43. const String Ultrasonic_OFF = "Ultra_Stop";
44.
45. const String Line_Tracking_ON = "Tracking_Start";
46. const String Line_Tracking_OFF = "Tracking_Stop";
47.
48. const String Light_Tracking_ON = "Light_Tracking";
49. const String Light_Tracking_OFF = "LightTrackingStop";
50.
51. const String Follow_ON = "UltraFollow";
52. const String Follow_OFF = "UltraFollowStop";
53.
54. String comdata = "";
55. // String text = "";
56. int judge;
57.
58. void setup()
59. {
60.   Serial.begin(115200);           // set up a wifi serial communication baud rate 115200
61.   RGB_Setup();                   //RGB LED initialization
62.   RGB_brightness(2);             // value 0-10
63.   All_RGB(250,0,0); // Set RGB LED color value.
64.   Servo_Setup();                 //Servo initialization
65.   PCA9685_Servo_Setup();         //PCA9685 Servo initialization
66.   Motor_Setup();                 //Motor initialization
67.   AllMotorStop();
68.   Buzzer_Setup();                //Buzzer initialization
69.   WS2812_Setup();                //WS2812 LED initialization
70.   WS2812_Brightness(5);          // value 0-10
71.   Ultrasonic_Setup();            //Ultrasonic initialization
72.   Photosensitive_Setup();         //Light line initialization
73.   Tracking_Setup();              //Tracking Line initialization
74.   OLED_Setup();                  //OLED initialization
75.   Matrix_Setup();
76.
77.   Serial.println("AT+CWMODE=3\r\n");//set to softAP+station mode
78.   delay(3000);                   //delay 4s
79.   Serial.println("AT+CWSAP=\"MY_ESP8266_5\", \"12345678\", 8, 2\r\n"); //TCP Protocol, serve
   r IP addr, port
80.   delay(1000);                   //delay 4s
81.   Serial.println("AT+RST\r\n");  //reset wifi
82.   delay(1000);                   //delay 4s
83.   Serial.println("AT+CIPMUX=1\r\n");//set to multi-connection mode
```

```
84. delay(1000);
85. // Serial.println("AT+CIPSERVER=1,333\r\n");//set as server
86. Serial.println("AT+CIPSERVER=1,4000\r\n");//set as server
87. delay(1000);
88. Serial.println("AT+CIPSTO=7000\r\n");//keep the wifi connecting 7000 seconds
89. delay(1000);
90.
91. WS2812ColorAll(255, 255,0); // Green
92. Servo_Angle(1, 90);
93. Servo_Angle(2, 90);
94. PCA9685_Servo_Angle(6, 0, 90);
95. PCA9685_Servo_Angle(7, 0, 90);
96. Buzzer_Silence();
97. OLED_clear();
98. delay(1000);
99. All_RGB(0,0,0);// Set RGB LED color value.
100. WS2812ColorAll(0,0,0);
101. }
102.
103.
104. void loop()
105. {
106.     while(Serial.available()>0)
107.     {
108.         comdata += char(Serial.read());
109.         delay(1);
110.     }
111.     judgement();
112.     control(judge);
113. }
114.
115. void judgement(){
116.     if (comdata.length() > 0){
117.         // if(comdata.endsWith(text4)||comdata.endsWith(phone4)){//left
118.         //     judge=4;
119.         // Serial.println(comdata);    //reset wifi1111111111
120.         // }
121.         if(comdata.endsWith(Move_UP)){//forward
122.             judge=1;
123.             Serial.println(comdata); //print received data.
124.         }
125.         else if(comdata.endsWith(Move_Down)){//backward
```

```
126.         judge=2;
127.         Serial.println(comdata);
128.     }
129.     else if(comdata.endsWith(Move_Left)){//left
130.         judge=3;
131.         Serial.println(comdata);
132.     }
133.     else if(comdata.endsWith(Move_Right)){//right.
134.         judge=4;
135.         Serial.println(comdata);
136.     }
137.     else if(comdata.endsWith(Move_UD_Stop)||comdata.endsWith(Move_LR_Stop)){//stop
138.         judge=5;
139.         Serial.println(comdata);
140.     }
141.     else if(comdata.endsWith(Head_Left)){//trun left
142.         judge=6;
143.         Serial.println(comdata);
144.     }
145.     else if(comdata.endsWith(Head_Right)){//trun right
146.         judge=7;
147.         Serial.println(comdata);
148.     }
149.     else if(comdata.endsWith(Head_Stop)){//stop servo rotation.
150.         judge=8;
151.         Serial.println(comdata);
152.     }
153.
154.     else if(comdata.endsWith(Ultrasonic_ON)){//avoid obstacles function.
155.         judge=9;
156.         Serial.println(comdata);
157.     }
158.     else if(comdata.endsWith(Ultrasonic_OFF)){//avoid obstacles function. bstart.
159.         judge=10;
160.         Serial.println(comdata);
161.     }
162.
163.     else if(comdata.endsWith(Line_Tracking_ON)){//line tracking function.
164.         judge=11;
165.         Serial.println(comdata);
166.     }
167.     else if(comdata.endsWith(Line_Tracking_OFF)){//light tracking function. dstart
```

```
168.         judge=12;
169.         Serial.println(comdata);
170.     }
171.     else if(comdata.endsWith(Light_Tracking_ON)){//light tracking function.
172.         judge=13;
173.         Serial.println(comdata);
174.     }
175.     else if(comdata.endsWith(Light_Tracking_OFF)){//light tracking function. dstart
176.         judge=14;
177.         Serial.println(comdata);
178.     }
179.     else if(comdata.endsWith(Follow_ON)){//follow function.
180.         judge=15;
181.         Serial.println(comdata);
182.     }
183.     else if(comdata.endsWith(Follow_OFF)){//light tracking function. dstart
184.         judge=16;
185.         Serial.println(comdata);
186.     }
187.
188.
189.     comdata = "";
190.     delay(10);
191. }
192.
193. // Serial.print("judge:");
194. // Serial.println(judge);
195. // return judge;
196. }
197.
198.
199. void control(int value){
200.     switch (value) {
201.         case 1: // forward
202.             Servo_Angle(1, servo_Init + deviation);
203.             Motor(1, 1, motor_speed); //Motor1 forward
204.             Motor(2, 1, motor_speed); //Motor2 forward
205.             // control_num = 12;
206.             break;
207.
208.         case 2: // Down,
209.             Servo_Angle(1, servo_Init + deviation);
```

```
210.    Motor(1, -1, motor_speed); //Motor1 backward
211.    Motor(2, -1, motor_speed); //Motor2 backward
212.    // control_num = 13;
213.    break;
214.
215.    case 3: // left
216.        Servo_Angle(1, servo_Init + deviation + wheel_Steering); // left
217.        Motor(1, 1, motor_speed);
218.        Motor(2, 1, motor_speed);
219.        // control_num = 14;
220.        break;
221.
222.    case 4: // right
223.        Servo_Angle(1, servo_Init + deviation - wheel_Steering); // right
224.        Motor(1, 1, motor_speed);
225.        Motor(2, 1, motor_speed);
226.        // control_num = 15;
227.        break;
228.
229.    case 5: // stop
230.        Servo_Angle(1, servo_Init + deviation);
231.        Motor(1, 1, 0);
232.        Motor(2, 1, 0);
233.        // control_num = -1;
234.        break;
235.
236.    case 6: // trun left
237.        servo_Angle2 = servo_Angle2 + 1;
238.        if (servo_Angle2 > 180){
239.            servo_Angle2 = 180;
240.        }
241.        Servo_Angle(2, servo_Angle2);
242.        // control_num = 15;
243.        delay(10);
244.        break;
245.    case 7: // trun right
246.        servo_Angle2 = servo_Angle2 - 1;
247.        if (servo_Angle2 < 0){
248.            servo_Angle2 = 0;
249.        }
250.        Servo_Angle(2, servo_Angle2);
251.        // control_num = 15;
```



```
252.     delay(10);
253.     break;
254.
255.     case 8: // stop servo rotation.
256.         break;
257.
258.     case 9:
259.         Function_Stop_flag = 0;
260.         Avoid_Obstacles(); // Avoid Obstacles function
261.         break;
262.
263.     case 11:
264.         Function_Stop_flag = 0;
265.         Line_Tracking(); // Line Tracking function
266.         break;
267.     case 13:
268.         Function_Stop_flag = 0;
269.         Light_Tracking(); // Light Tracking function
270.         break;
271.     case 15:
272.         Function_Stop_flag = 0;
273.         Keep_Distance(); // Follow function
274.         break;
275.
276.     case 10:
277.         Servo_Angle(1, servo_Init + deviation);
278.         Motor(1, 1, 0);
279.         Motor(2, 1, 0);
280.         break;
281.     case 12:
282.         Servo_Angle(1, servo_Init + deviation);
283.         Motor(1, 1, 0);
284.         Motor(2, 1, 0);
285.         break;
286.     case 14:
287.         Servo_Angle(1, servo_Init + deviation);
288.         Motor(1, 1, 0);
289.         Motor(2, 1, 0);
290.         break;
291.     case 16:
292.         Servo_Angle(1, servo_Init + deviation);
293.         Motor(1, 1, 0);
```

```
294.     Motor(2, 1, 0);
295.     break;
296.
297.
298.     default:
299.         break;
300. }
301. }
302.
303. int StopFunction(){
304.     while(Serial.available()>0){
305.         comdata += char(Serial.read());
306.         delay(1);
307.     }
308.     if (comdata.length() > 0){
309.         if(comdata.endsWith(Ultrasonic_OFF)){// Stop Avoid Obstacles function.
310.             Function_Stop_flag = 1;
311.             judge=10;
312.         }
313.         else if(comdata.endsWith(Line_Tracking_OFF)){ // Stop Line Tracking function.
314.             Function_Stop_flag = 2;
315.             judge=12;
316.         }
317.         else if(comdata.endsWith(Light_Tracking_OFF)){ // Stop Light Tracking function.
318.             Function_Stop_flag = 3;
319.             judge=14;
320.         }
321.         else if(comdata.endsWith(Follow_OFF)){ // Stop Follow function.
322.             Function_Stop_flag = 4;
323.             judge=16;
324.         }
325.         comdata = "";
326.         delay(10);
327.     }
328. }
329.
330. void Avoid_Obstacles(){
331.     while (1){
332.         StopFunction();
333.         if (Function_Stop_flag == 1){ // Press OK, stop function.
334.             break;
335.         }
```

```
336.
337.   Servo_Angle(2, servo_Init + deviation);
338.   delay(80);
339.   int a = GetDistance();
340.   int b = GetDistance();
341.   int c = GetDistance();
342.   midDist = (a+b+c)/3;
343.   Serial.print("Mid:");
344.   Serial.println(midDist);
345.   // Servo_1_Angle(servo_Init); // front wheel
346.   Motor(1,1,0); //Stop the car
347.   Motor(2,1,0);
348.
349.   if (midDist > avoid_Dist){
350.       // Servo_1_Angle(servo_Init+ deviation); // front wheel
351.       Servo_Angle(1, servo_Init + deviation); // front wheel
352.       Motor(1,1,Speed); //forward
353.       Motor(2,1,Speed);
354.   }
355.   else if (midDist <= avoid_Dist){
356.       // Servo_1_Angle(servo_Init + deviation); // front wheel
357.       Servo_Angle(1, servo_Init + deviation); // front wheel
358.       Motor(1,1,0); //Stop the car
359.       Motor(2,1,0);
360.       // Servo_2_Angle(servo_Init - 60); // left distance.
361.       Servo_Angle(2, servo_Init + deviation - 60); // left distance.
362.       delay(400);
363.       int a = GetDistance();
364.       int b = GetDistance();
365.       int c = GetDistance();
366.       leftDist = (a+b+c)/3;
367.       Serial.print("Left:");
368.       Serial.println(leftDist);
369.       // Servo_2_Angle(servo_Init + 60); // right distance.
370.       Servo_Angle(2, servo_Init + deviation + 60); // right distance.
371.       delay(400);
372.       a = GetDistance();
373.       b = GetDistance();
374.       c = GetDistance();
375.       rightDist = (a+b+c)/3;
376.       Serial.print("Right:");
377.       Serial.println(rightDist);
```

```

378.      // Servo_2_Angle(servo_Init); // back to mid.
379.      Servo_Angle(2, servo_Init + deviation);    // back to mid.
380.
381.      if ((leftDist < avoid_Dist)&&(rightDist < avoid_Dist)){ // Judgment left and right.
382.          if (leftDist >= rightDist){
383.              // There are obstacles on the right backward to the left.
384.              // Servo_1_Angle(servo_Init + wheel_Steering + deviation); //turn left backwa
rd
385.              Servo_Angle(1, servo_Init + deviation + wheel_Steering);    // turn left back
ward
386.              Motor(1,-1,Speed); //backward
387.              Motor(2,-1,Speed);
388.              delay(500);
389.          }
390.          else{ //There are obstacles on the left.
391.              // Servo_1_Angle(servo_Init - wheel_Steering + deviation); //turn right backw
ard
392.              Servo_Angle(1, servo_Init + deviation - wheel_Steering);    // turn right bac
kward
393.              Motor(1,-1,Speed); //backward
394.              Motor(2,-1,Speed);
395.              delay(500);
396.          }
397.      }
398.      else if ((leftDist > avoid_Dist)&&(rightDist <= avoid_Dist)){
399.          if (midDist < minDist){ // Obstacle ahead
400.              // Servo_1_Angle(servo_Init+ deviation); // backward
401.              Servo_Angle(1, servo_Init + deviation);    // backward
402.              Motor(1,-1,Speed);
403.              Motor(2,-1,Speed);
404.              delay(400);
405.          }
406.          // Servo_1_Angle(servo_Init + wheel_Steering + deviation); // turn left backwar
d
407.          Servo_Angle(1, servo_Init + deviation + wheel_Steering);    // turn left backwa
rd
408.          Motor(1,-1,Speed);
409.          Motor(2,-1,Speed);
410.          delay(500);
411.      }
412.      else if ((leftDist <= avoid_Dist) &&(rightDist > avoid_Dist)){ // There are obstacl

```

```

    es on the left.
413.         if (midDist < minDist){ // Obstacle ahead
414.             // Servo_1_Angle(servo_Init + deviation); // backward
415.             Servo_Angle(1, servo_Init + deviation);    // backward
416.             Motor(1,-1,Speed);
417.             Motor(2,-1,Speed);
418.             delay(500);
419.         }
420.         // Servo_1_Angle(servo_Init - wheel_Steering + deviation); //turn right backward
421.         Servo_Angle(1, servo_Init + deviation - wheel_Steering);    // turn right backward
422.         Motor(1,-1,Speed); //backward
423.         Motor(2,-1,Speed);
424.         delay(400);
425.     }
426.     else if ((leftDist >= avoid_Dist) &&( rightDist >= avoid_Dist)){
427.         if (leftDist > rightDist){ // The distance to the right is greater than the left
428.             if (midDist < minDist){
429.                 // Servo_1_Angle(servo_Init+ deviation); // backward
430.                 Servo_Angle(1, servo_Init + deviation);    // backward
431.                 Motor(1,-1,Speed);
432.                 Motor(2,-1,Speed);
433.                 delay(500);
434.             }
435.             // Servo_1_Angle(servo_Init + wheel_Steering + deviation); // turn left backward
436.             Servo_Angle(1, servo_Init + deviation + wheel_Steering);    // turn left backward
437.             Motor(1,-1,Speed);
438.             Motor(2,-1,Speed);
439.             delay(400);
440.         }
441.     else{
442.         if (midDist < minDist){
443.             // Servo_1_Angle(servo_Init+ deviation); // backward
444.             Servo_Angle(1, servo_Init + deviation);    // backward
445.             Motor(1,-1,Speed);
446.             Motor(2,-1,Speed);
447.             delay(500);
448.         }

```



```
449.          // Servo_1_Angle(servo_Init + wheel_Steering + deviation); // turn left backward
450.          Servo_Angle(1, servo_Init + deviation + wheel_Steering); // turn left backward
451.          Motor(1, -1, Speed);
452.          Motor(2, -1, Speed);
453.          delay(400);
454.      }
455.  }
456.  }
457.  // delay(100);
458.  }
459.  }
460.
461.
462. void Light_Tracking(){
463.     value_Init = GetPhotosensitive();
464.     while (1){
465.         StopFunction();
466.         if (Function_Stop_flag == 3){ // Press OK, stop function.
467.             break;
468.         }
469.         value = GetPhotosensitive();
470.         if (value < (value_Init - threshold)){
471.             // Servo_1_Angle(servo_Init + wheel_Steering + deviation);
472.             Servo_Angle(1, servo_Init + deviation + wheel_Steering);
473.             Motor(1, 1, motor_speed);
474.             Motor(2, 1, motor_speed);
475.             Serial.print(value_Init);
476.             Serial.print(":");
477.             Serial.println(value);
478.
479.         }
480.         else if (value > (value_Init + threshold)){
481.             // Servo_1_Angle(servo_Init - wheel_Steering + deviation);
482.             Servo_Angle(1, servo_Init + deviation - wheel_Steering);
483.             Motor(1, 1, motor_speed);
484.             Motor(2, 1, motor_speed);
485.             Serial.print(value_Init);
486.             Serial.print(":");
487.             Serial.println(value);
488.         }
```

```
489.     else{
490.         // Servo_1_Angle(servo_Init);
491.         Servo_Angle(1, servo_Init + deviation);
492.         Motor(1, 1, 0);
493.         Motor(2, 1, 0);
494.         Serial.print(value_Init);
495.         Serial.print(":");
496.         Serial.println(value);
497.     }
498. }
499. }
500.
501.
502. void Line_Tracking(){
503.     int value;
504.     while (1){
505.         StopFunction();
506.         if (Function_Stop_flag == 2){ // Press OK, stop function.
507.             Serial.println("2222222");
508.             break;
509.         }
510.         value = Track_Read(); //Read the value of the tracking module.
511.         // Serial.println(value);
512.         switch (value)
513.         {
514.             case 0: //000 stop
515.                 // Servo_1_Angle(servo_Init + deviation); // stop
516.                 Servo_Angle(1, servo_Init + deviation); // mid
517.                 Motor(1, 1, 0);
518.                 Motor(2, 1, 0);
519.                 if (Track_value != 0){
520.                     OLED_clear();
521.                     // OLED(2,0,0,"Control: Stop");
522.                     // OLED(2,0,30,"Value: 0 0 0");
523.                     // OLED(1,20,50,"www.adeept.com");
524.
525.                     // OLED(1,0,0,"Control: Stop");
526.                     // OLED(2,0,20,"Value:");
527.                     // OLED(2,30,40,"0 0 0");
528.
529.                     // OLED(1,0,0,"Control: Stop");
530.                     // OLED(2,0,20,"Value:");
```

```
531.         // OLED(2,30,40,"0 1 1");
532.         // OLED_dispaly();
533.     }
534.     Track_value = 0;
535.     break;
536.
537.     case 1:    //010 forward
538.         // Servo_1_Angle(servo_Init+ deviation);
539.         Servo_Angle(1, servo_Init + deviation);    // mid
540.         Motor(1, 1, motor_speed); //Motor1 forward
541.         Motor(2, 1, motor_speed); //Motor2 forward
542.         if (Track_value != 2){
543.             // OLED_clear();
544.             // OLED(2,0,0,"Control: Forward");
545.             // OLED(2,0,30,"Value: 0 1 0");
546.             // OLED(1,20,50,"www.adeept.com");
547.             // OLED(1,0,0,"Control: Forward");
548.             // OLED(2,0,20,"Value:");
549.             // OLED(2,30,40,"0 1 0");
550.         }
551.         Track_value = 2;
552.         break;
553.
554.     case 2:    //100 Left
555.         // Servo_1_Angle(servo_Init + wheel_Steering + deviation); // left
556.         Servo_Angle(1, servo_Init + deviation + wheel_Steering);    // left
557.         Motor(1, 1, motor_speed);
558.         Motor(2, 1, motor_speed);
559.         if (Track_value != 4){
560.             // OLED_clear();
561.             // OLED(2,0,0,"Control: Left");
562.             // OLED(2,0,30,"Value: 1 0 0");
563.             // OLED(1,20,50,"www.adeept.com");
564.             // OLED(1,0,0,"Control: Left");
565.             // OLED(2,0,20,"Value:");
566.             // OLED(2,30,40,"1 0 0");
567.         }
568.         Track_value = 4;
569.         break;
570.
571.     case 3:    //110 Left
572.         // Servo_1_Angle(servo_Init + wheel_Steering + deviation); // left
```

```
573.     Servo_Angle(1, servo_Init + deviation + wheel_Steering);    // left
574.     Motor(1, 1, motor_speed);
575.     Motor(2, 1, motor_speed);
576.     if (Track_value != 6){
577.         // OLED_clear();
578.         // OLED(2,0,0,"Control: Left");
579.         // OLED(2,0,30,"Value: 1 1 0");
580.         // OLED(1,20,50,"www.adeept.com");
581.         // OLED(1,0,0,"Control: Left");
582.         // OLED(2,0,20,"Value:");
583.         // OLED(2,30,40,"1 1 0");
584.     }
585.     Track_value = 6;
586.     break;
587.
588.     case 4:    //001 right
589.         // Servo_1_Angle(servo_Init - wheel_Steering + deviation);// right
590.         Servo_Angle(1, servo_Init + deviation - wheel_Steering);    // right
591.         Motor(1, 1, motor_speed);
592.         Motor(2, 1, motor_speed);
593.         if (Track_value != 1){
594.             // OLED_clear();
595.             // OLED(2,0,0,"Control: Right");
596.             // OLED(2,0,30,"Value: 0 0 1");
597.             // OLED(1,20,50,"www.adeept.com");
598.             // OLED(1,0,0,"Control: Right");
599.             // OLED(2,0,20,"Value:");
600.             // OLED(2,30,40,"0 0 1");
601.         }
602.         Track_value = 1;
603.         break;
604.
605.     case 5:    //011 right
606.         // Servo_1_Angle(servo_Init - wheel_Steering + deviation);// right
607.         Servo_Angle(1, servo_Init + deviation - wheel_Steering);    // right
608.         Motor(1, 1, motor_speed);
609.         Motor(2, 1, motor_speed);
610.         if (Track_value != 3){
611.             // OLED_clear();
612.             // OLED(2,0,0,"Control: Right");
613.             // OLED(2,0,30,"Value: 0 1 1");
614.             // OLED(1,20,50,"www.adeept.com");
```

```
615.         // OLED(1,0,0,"Control: Right");
616.         // OLED(2,0,20,"Value:");
617.         // OLED(2,30,40,"0 1 1");
618.     }
619.     Track_value = 3;
620.     break;
621.
622.     case 6:    //111 stop
623.         // Serial.println("1111111111111111");
624.         // Servo_1_Angle(servo_Init + deviation); // stop
625.         Servo_Angle(1, servo_Init + deviation);    // mid
626.         Motor(1, 1, 0);
627.         Motor(2, 1, 0);
628.
629.         // Serial.println("11111111333");
630.         if (Track_value != 7){
631.             //   OLED_clear();
632.             // OLED(1,0,0,"Control: Stop");
633.             // OLED(2,0,20,"Value:");
634.             // OLED(2,30,40,"0 1 1");
635.             // OLED_dispaly();
636.         }
637.         Track_value = 7;
638.         break;
639.     case 7:    //101 forward
640.         // Servo_1_Angle(servo_Init+ deviation);
641.         Servo_Angle(1, servo_Init + deviation);    // mid
642.         Motor(1, 1, motor_speed); //Motor1 forward
643.         Motor(2, 1, motor_speed); //Motor2 forward
644.         if (Track_value != 5){
645.             // OLED_clear();
646.             // OLED(2,0,0,"Control: Forward");
647.             // OLED(2,0,30,"Value: 1 0 1");
648.             // OLED(1,20,50,"www.adeept.com");
649.             // OLED(1,0,0,"Control: Forward");
650.             // OLED(2,0,20,"Value:");
651.             // OLED(2,30,40,"1 0 1");
652.         }
653.         Track_value = 5;
654.         break;
655.     default:
656.         break;
```



```
657.     }
658.     // delay(100);
659.     }
660. }
661.
662. void Keep_Distance(){
663.     Servo_Angle(2, servo_Angle2);
664.     while (1){
665.         StopFunction();
666.         if (Function_Stop_flag == 4){ // Press OK, stop function.
667.             break;
668.         }
669.         distance = GetDistance();
670.         if (distance < 30){
671.             Servo_Angle(1, servo_Init + deviation);    // front wheel
672.             Motor(1,-1,Speed); //forward
673.             Motor(2,-1,Speed);
674.         }
675.         else if (distance > 40){
676.             Servo_Angle(1, servo_Init + deviation);    // front wheel
677.             Motor(1,1,Speed); //forward
678.             Motor(2,1,Speed);
679.         }
680.         else {
681.             Motor(1,1,0); // stop
682.             Motor(2,1,0);
683.         }
684.         delay(100);
685.     }
686. }
1.
```