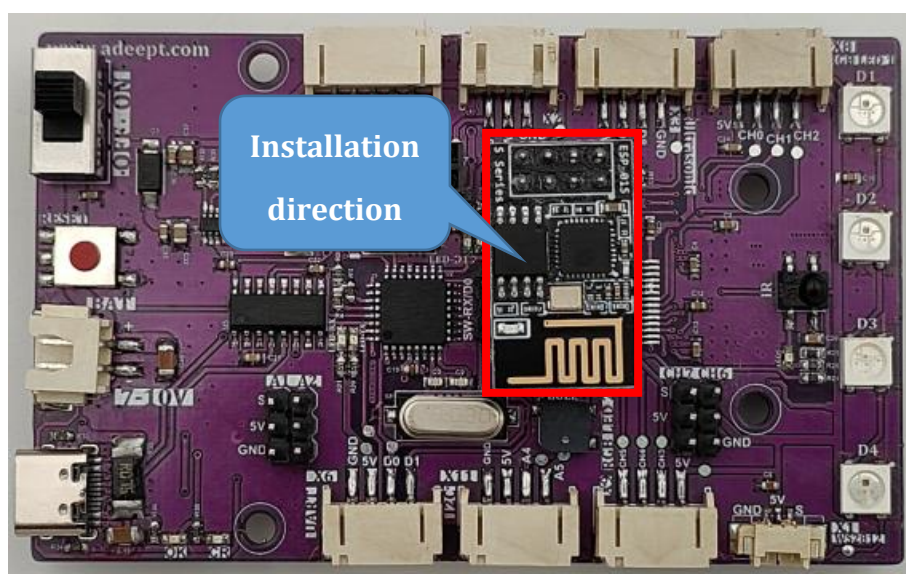
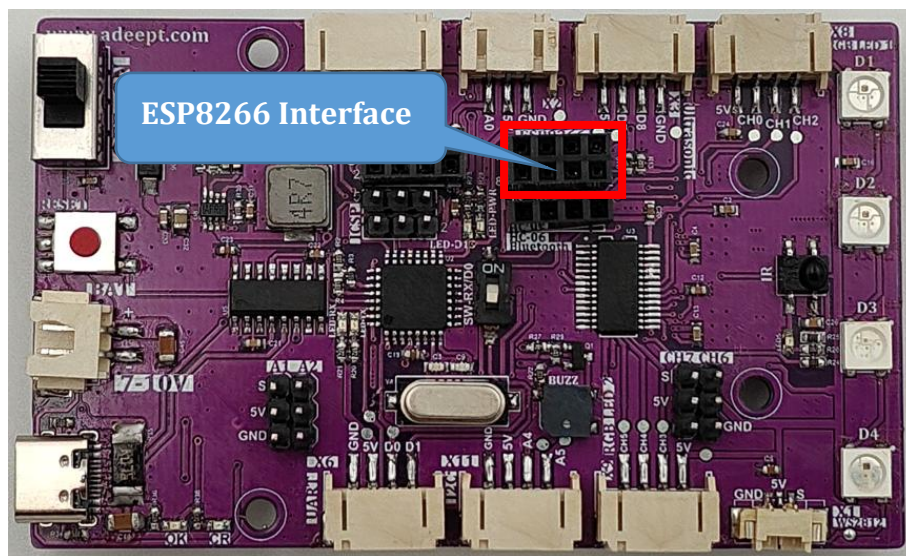


Lesson 16 How to use GUI to Control the Car

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.

16.1 Install of ESP8266 module



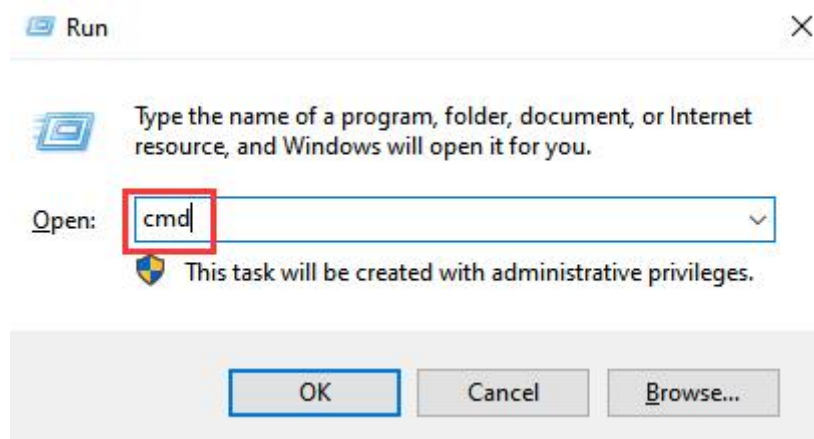
16.2 Configure the usage environment on the PC

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.

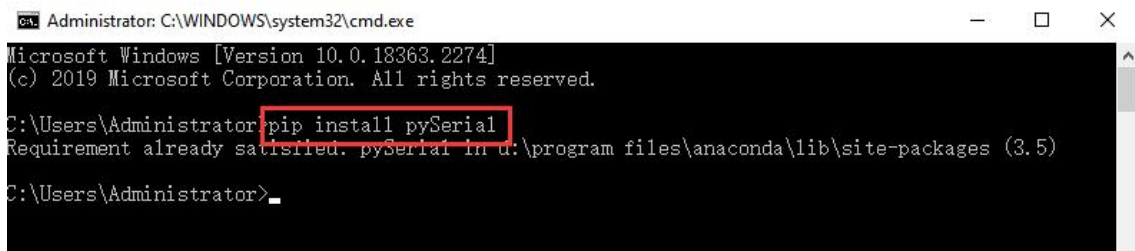
Installing pySerial

- pySerial encapsulates the serial communication module, supporting Linux, Windows, BSD (may support all operating systems that support POSIX), Python (Java) and IconPython (.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
```



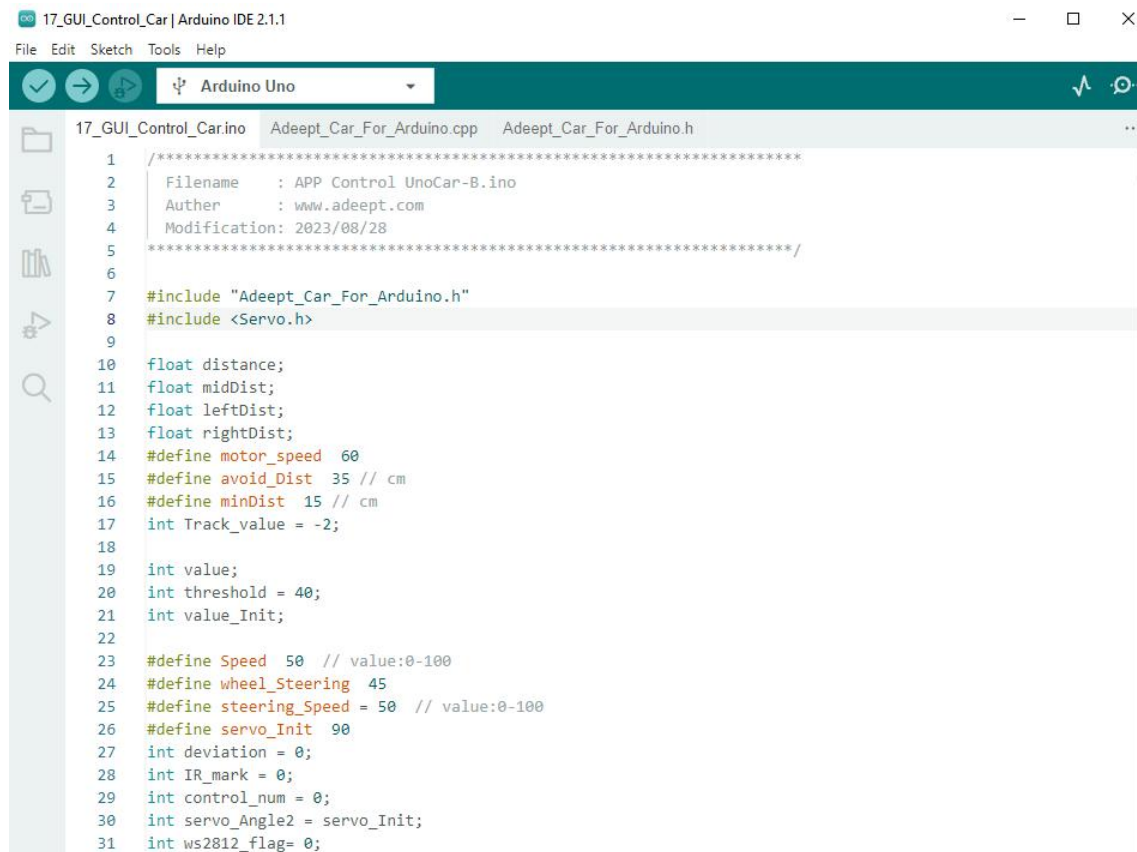
```
Administrator: C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.18363.2274]
(c) 2019 Microsoft Corporation. All rights reserved.

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.

16.3 Control the car in AP mode

1. Connect your computer and Adeept Robot Control Board with a USB cable.
2. Open “16_GUI_Control_Car” folder in “/Code”, double-click “16_GUI_Control_Car.ino”.



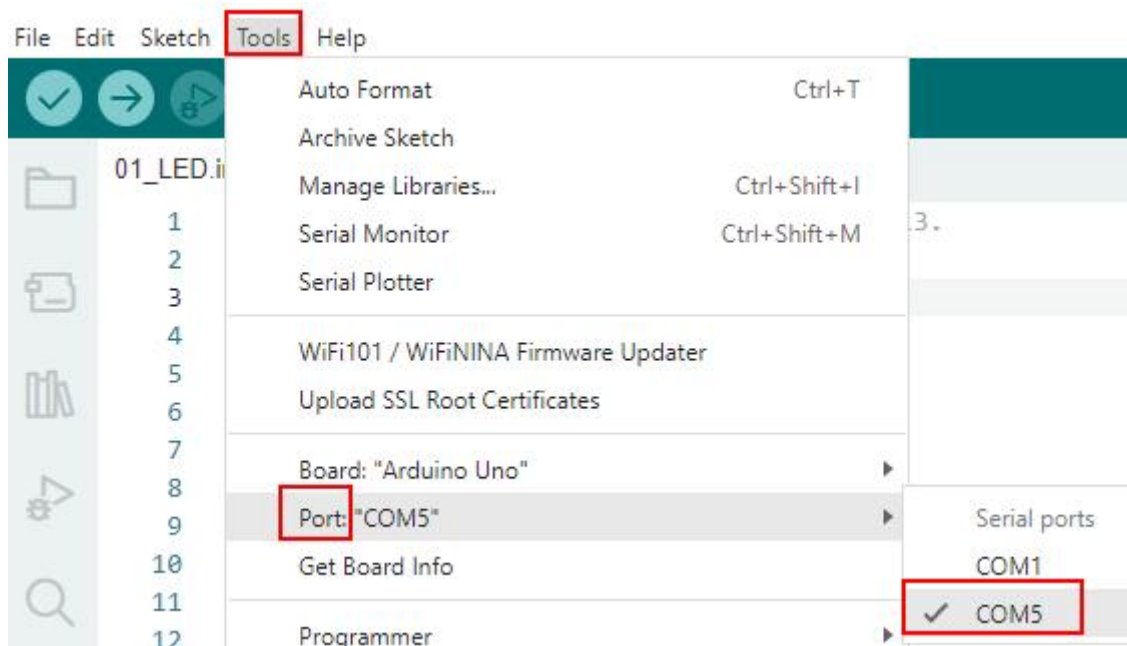
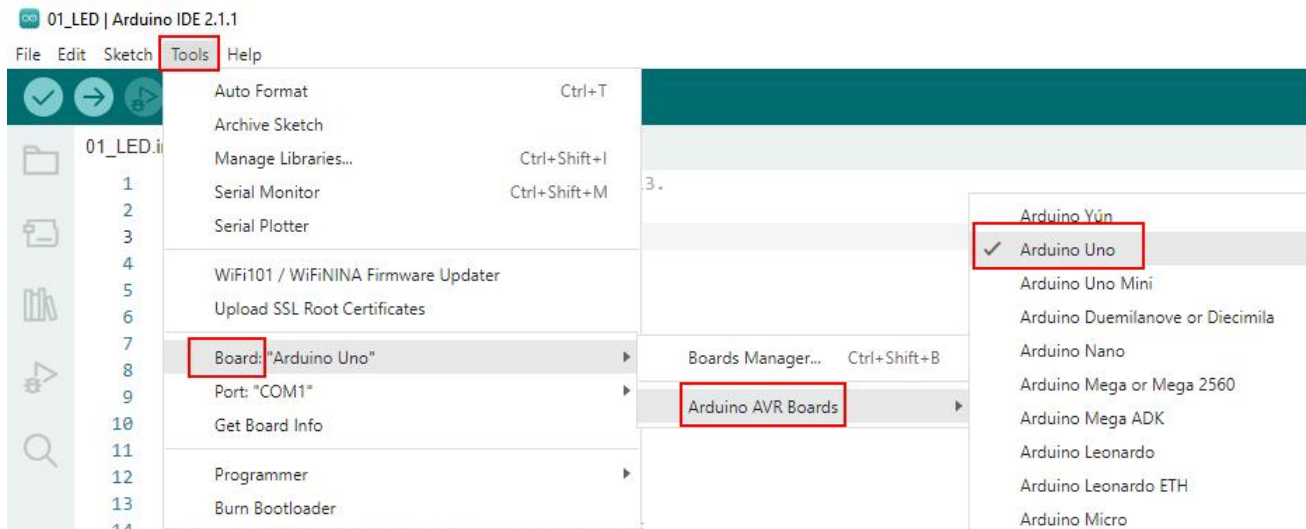
```
17_GUI_Control_Car.ino | Arduino IDE 2.1.1
File Edit Sketch Tools Help
Arduino Uno
17_GUI_Control_Car.ino Adeept_Car_For_Arduino.cpp Adeept_Car_For_Arduino.h
1  /*****
2  Filename   : APP_Control_UnoCar-B.ino
3  Auther    : www.adeept.com
4  Modification: 2023/08/28
5  *****/
6
7  #include "Adeept_Car_For_Arduino.h"
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.

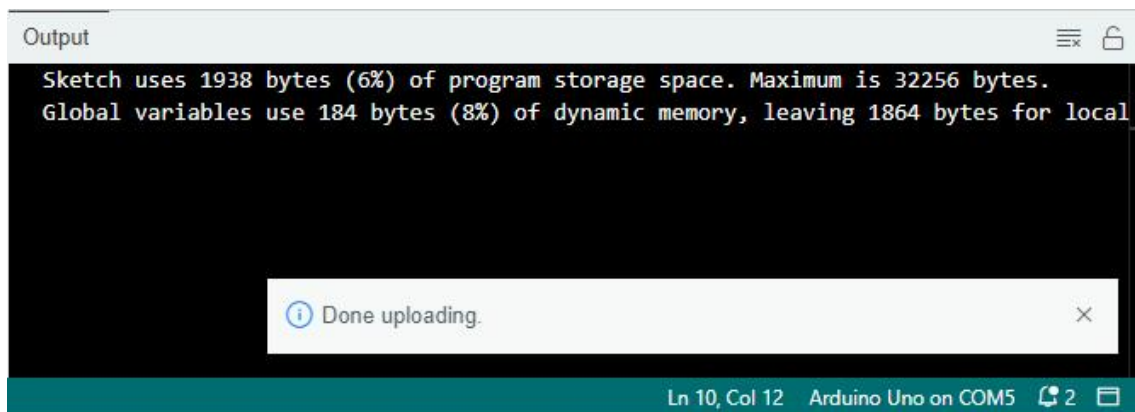
Board: **Tools**---->**Board**---->**Arduino AVR Boards**---->**Arduino Uno**

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

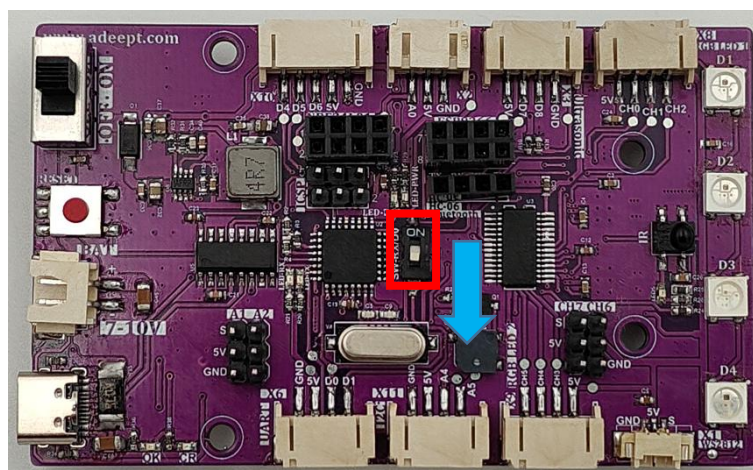
Note: The port number will be different in different computers.



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.



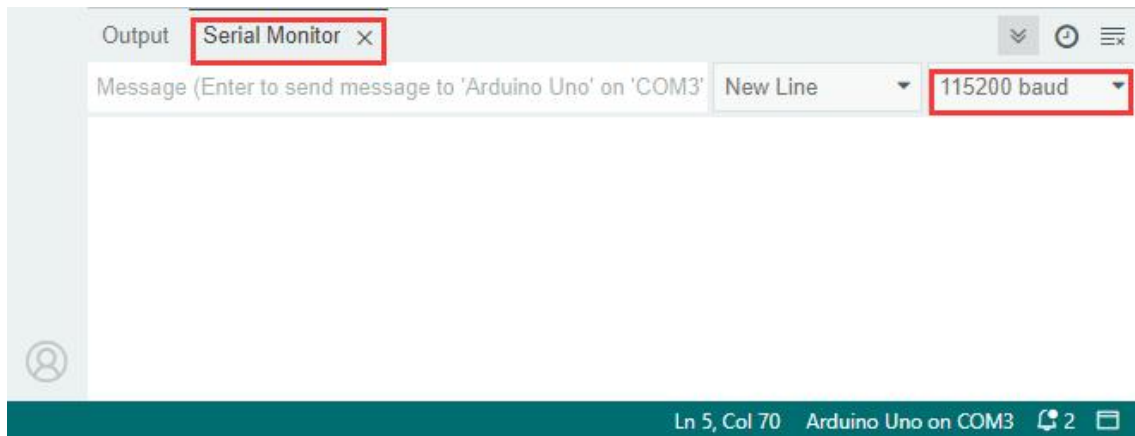
Note: If the upload program fails, a red error message appears. Please confirm whether the RX/D0 switch is in the correct position. Toggle down the paddle (white) of the switch in the picture below.



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.

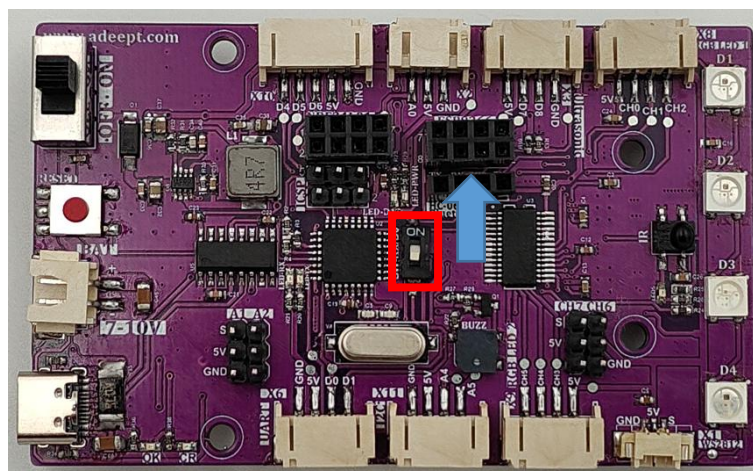




Note: When the ESP8266 module transmits data to the Arduino, it needs to occupy the RX interface of the Arduino, and when the Arduino uploads the program, it also needs to occupy the RX interface. The RX interface cannot satisfy both functions at the same time, so a switch is needed to distinguish them.

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.

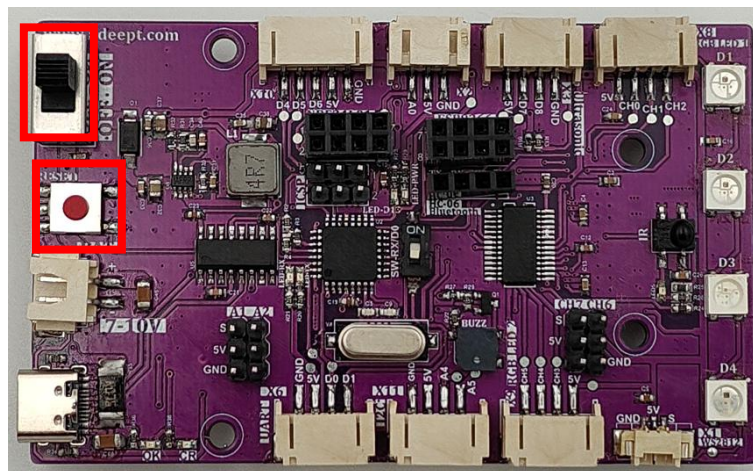
Now we need the communication function of ESP8266, so we need to toggle the switch up.



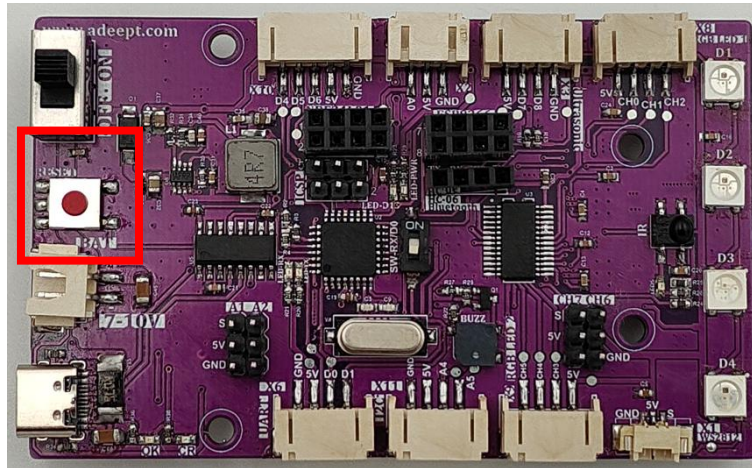
If you need to upload other programs later, and ESP8266 is installed on Adeept Robot Control Board, you need to flip the switch down to disconnect RX from ESP8266.

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.

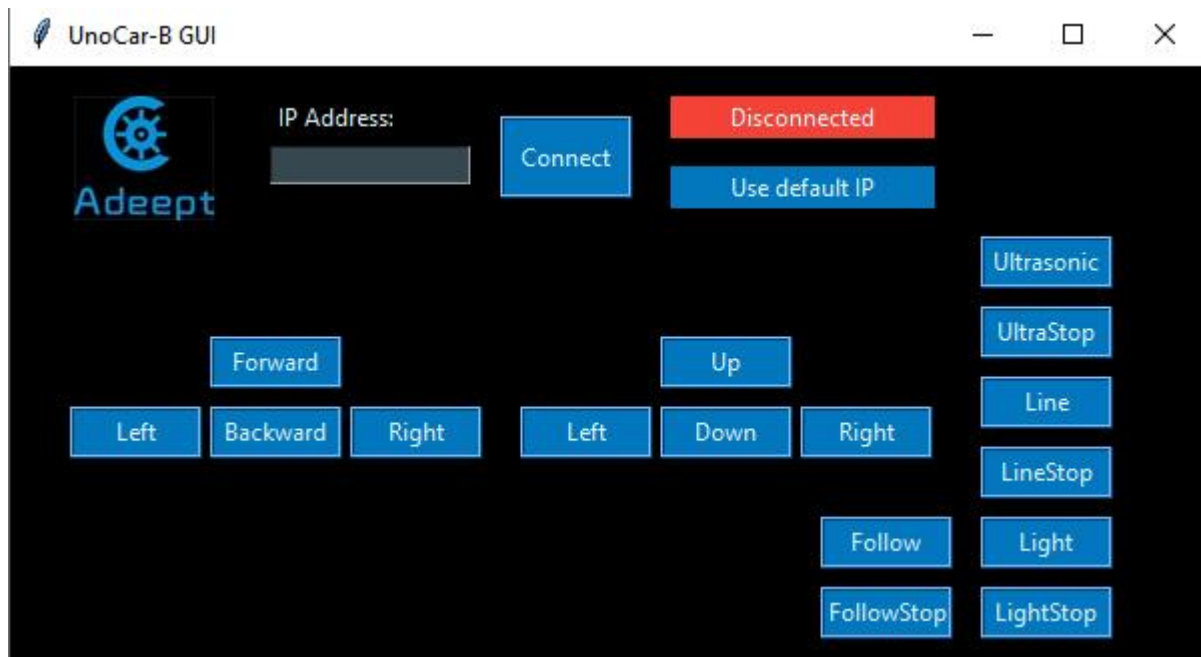
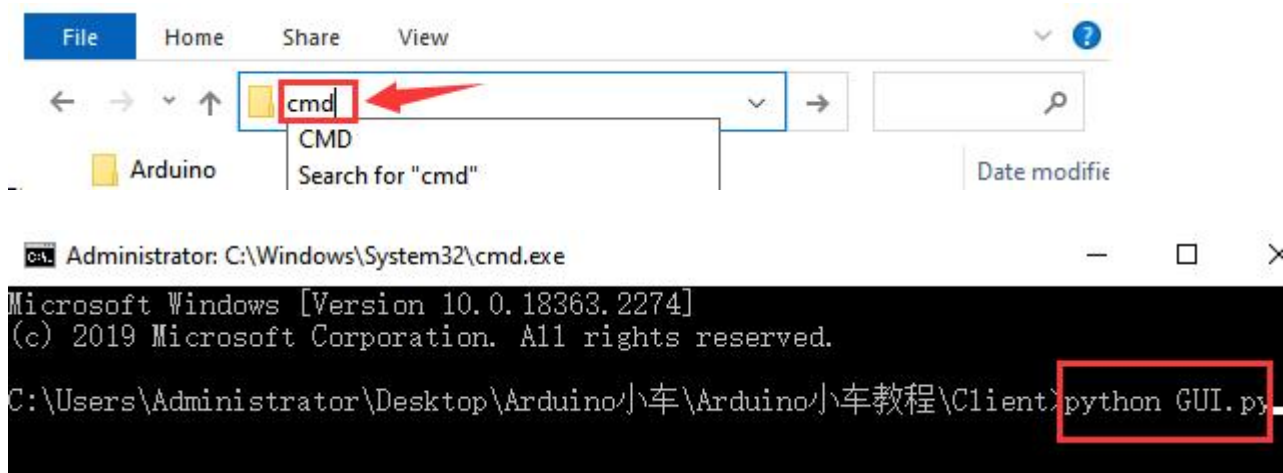
📶 Adeept_ESP8266

```
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).

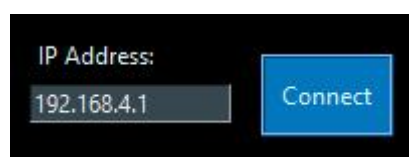
9. Enter the Client folder, double-click GUI.py to run the program.

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.			

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_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 [16_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.  #include <FlexiTimer2.h> //to set a timer to manage all server
4.  #include <Adafruit_SSD1306.h>
5.
6.  #define OLED_RESET      4
7.  Adafruit_SSD1306 display(128, 64, &Wire, OLED_RESET);
8.
9.  float distance;
10. float midDist;
11. float leftDist;
12. float rightDist;
13. #define motor_speed  60
14. #define avoid_Dist  35 // cm
15. #define minDist  15 // cm
16. int Track_value = -2;
17.
18. int value;
19. int threshold = 40;
20. int value_Init;
21.
22. #define Speed  50 // value:0-100
23. #define wheel_Steering  45
24. #define steering_Speed = 50 // value:0-100
25. #define servo_Init  90
26. int deviation = 0;
27. int IR_mark = 0;
28. int control_num = 0;
29. int servo_Angle2 = servo_Init;
30. int ws2812_flag= 0;
31. int Function_Stop_flag = 0;
32.
33.
34. const String Move_UP = "forward";
35. const String Move_Down = "backward";
36. const String Move_UD_Stop = "DS";
37. const String Move_Left = "turn_left";
38. const String Move_LR_Stop = "TS";
39. const String Move_Right = "turn_right";
40.
41. const String Head_UP = "up";
42. const String Head_Down = "down";
```



```
43.  const String Head_Left = "lookleft";
44.  const String Head_Right = "lookright";
45.  const String Head_Stop = "stop";
46.
47.  const String Ultrasonic_ON = "Ultra_Start";
48.  const String Ultrasonic_OFF = "Ultra_Stop";
49.
50.  const String Line_Tracking_ON = "Tracking_Start";
51.  const String Line_Tracking_OFF = "Tracking_Stop";
52.
53.  const String Light_Tracking_ON = "Light_Tracking";
54.  const String Light_Tracking_OFF = "LightTrackingStop";
55.
56.  const String Follow_ON = "UltraFollow";
57.  const String Follow_OFF = "UltraFollowStop";
58.
59.  String comdata = "";
60.  // String text = "";
61.  int judge;
62.
63.  void server() {
64.      while(Serial.available()>0) {
65.          comdata += char(Serial.read());
66.          delay(1);
67.      }
68.      judgement();
69.  }
70.
71.  void setup()
72.  {
73.      Serial.begin(115200);          // set up a wifi serial communication baud rate
                                     115200
74.      Servo_Setup();                //Servo initialization
75.      PCA9685_Servo_Setup();        //PCA9685 Servo initialization
76.      Motor_Setup();                //Motor initialization
77.      AllMotorStop();
78.      Buzzer_Setup();               //Buzzer initialization
79.      WS2812_Setup();               //WS2812 LED initialization
80.      WS2812_Brightness(5);         // value 0-10
81.      Ultrasonic_Setup();           //Ultrasonic initialization
82.      Photosensitive_Setup();        //Light Line initialization
83.      Tracking_Setup();              //Tracking Line initialization
```

```

84.    // OLED_Setup();                      //OLED initialization
85.    Matrix_Setup();
86.
87.    Serial.println("AT+CWMODE=3\r\n");//set to softAP+station mode
88.    delay(3000);        //delay 4s
89.    Serial.println("AT+CWSAP=\"MY_ESP8266_5\", \"12345678\",8,2\r\n"); //TCP Pr
    otocol, server IP addr, port
90.    delay(1000);        //delay 4s
91.    Serial.println("AT+RST\r\n");        //reset wifi
92.    delay(1000);        //delay 4s
93.    Serial.println("AT+CIPMUX=1\r\n");//set to multi-connection mode
94.    delay(1000);
95.    // Serial.println("AT+CIPSERVER=1,333\r\n");//set as server
96.    Serial.println("AT+CIPSERVER=1,4000\r\n");//set as server
97.    delay(1000);
98.    Serial.println("AT+CIPSTO=7000\r\n");//keep the wifi connecting 7000 seconds
99.    delay(1000);
100.
101.    WS2812ColorAll(255, 255,0); // Green
102.    Servo_Angle(1, 90);
103.    Servo_Angle(2, 90);
104.    PCA9685_Servo_Angle(6, 0, 90);
105.    PCA9685_Servo_Angle(7, 0, 90);
106.    Buzzer_Silence();
107.    // OLED_clear();
108.    delay(1000);
109.    WS2812ColorAll(0,0,0);
110.
111.    FlexiTimer2::set(50, server);
112.    FlexiTimer2::start();
113. }
114.
115.
116. void loop()
117. {
118.    // while(Serial.available()>0)
119.    // {
120.    //    comdata += char(Serial.read());
121.    //    delay(1);
122.    // }
123.    // judgement();
124.    control(judge);

```

```
125. }
126.
127. void judgement(){
128.     if (comdata.length() > 0){
129.         // if(comdata.endsWith(text4)||comdata.endsWith(phone4)){//Left
130.         //     judge=4;
131.         // Serial.println(comdata);    //reset wifi1111111111
132.         // }
133.         if(comdata.endsWith(Move_UP)){//forward
134.             judge=1;
135.             Serial.println(comdata); //print received data.
136.         }
137.         else if(comdata.endsWith(Move_Down)){//backward
138.             judge=2;
139.             Serial.println(comdata);
140.         }
141.         else if(comdata.endsWith(Move_Left)){//left
142.             judge=3;
143.             Serial.println(comdata);
144.         }
145.         else if(comdata.endsWith(Move_Right)){//right.
146.             judge=4;
147.             Serial.println(comdata);
148.         }
149.         else if(comdata.endsWith(Move_UD_Stop) || comdata.endsWith(Move_LR_Stop)
150.         ){//stop
151.             judge=5;
152.             Serial.println(comdata);
153.         }
154.         else if(comdata.endsWith(Head_Left)){//trun left
155.             judge=6;
156.             Serial.println(comdata);
157.         }
158.         else if(comdata.endsWith(Head_Right)){//trun right
159.             judge=7;
160.             Serial.println(comdata);
161.         }
162.         else if(comdata.endsWith(Head_Stop)){//stop servo rotation.
163.             judge=8;
164.             Serial.println(comdata);
165.         }
```

```
166.         else if(comdata.endsWith(Ultrasonic_ON)){//avoid obstacles function.
167.             judge=9;
168.             Serial.println(comdata);
169.         }
170.         else if(comdata.endsWith(Ultrasonic_OFF)){//avoid obstacles function.
171.             bstart.
172.             judge=10;
173.             Function_Stop_flag = 1;
174.             Serial.println(comdata);
175.         }
176.         else if(comdata.endsWith(Line_Tracking_ON)){//line tracking function.
177.             judge=11;
178.             Serial.println(comdata);
179.         }
180.         else if(comdata.endsWith(Line_Tracking_OFF)){//light tracking function
181.             . dstart
182.             judge=12;
183.             Function_Stop_flag = 2;
184.             Serial.println(comdata);
185.         }
186.         else if(comdata.endsWith(Light_Tracking_ON)){//light tracking function
187.             .
188.             judge=13;
189.             Serial.println(comdata);
190.         }
191.         else if(comdata.endsWith(Light_Tracking_OFF)){//light tracking functio
192.             n. dstart
193.             judge=14;
194.             Function_Stop_flag = 3;
195.             Serial.println(comdata);
196.         }
197.         else if(comdata.endsWith(Follow_ON)){//follow function.
198.             judge=15;
199.             Serial.println(comdata);
200.         }
201.         else if(comdata.endsWith(Follow_OFF)){//light tracking function. dstar
202.             t
203.             judge=16;
204.             Function_Stop_flag = 4;
205.             Serial.println(comdata);
206.         }
```

```
203.
204.
205.     comdata = "";
206.     delay(10);
207. }
208.
209. // Serial.print("judge:");
210. // Serial.println(judge);
211. // return judge;
212. }
213.
214.
215. void control(int value){
216.     switch (value) {
217.         case 1: // forward
218.             Motor(1, 1, motor_speed); //Motor1 forward
219.             Motor(2, 1, motor_speed); //Motor2 forward
220.             Motor(3, 1, motor_speed); //Motor3 forward
221.             Motor(4, 1, motor_speed); //Motor4 forward
222.             // control_num = 12;
223.             break;
224.
225.         case 2: // Down,
226.             Motor(1, -1, motor_speed); //Motor1 backward
227.             Motor(2, -1, motor_speed); //Motor2 backward
228.             Motor(3, -1, motor_speed); //Motor3 backward
229.             Motor(4, -1, motor_speed); //Motor4 backward
230.             // control_num = 13;
231.             break;
232.
233.         case 3: // left
234.             Motor(1, -1, motor_speed);
235.             Motor(2, -1, motor_speed);
236.             Motor(3, 1, motor_speed);
237.             Motor(4, 1, motor_speed);
238.             break;
239.
240.         case 4: // right
241.             Motor(1, 1, motor_speed);
242.             Motor(2, 1, motor_speed);
243.             Motor(3, -1, motor_speed);
244.             Motor(4, -1, motor_speed);
```



```
245.         break;
246.
247.     case 5: // stop
248.         Motor(1, 1, 0);
249.         Motor(2, 1, 0);
250.         Motor(3, 1, 0);
251.         Motor(4, 1, 0);
252.         // control_num = -1;
253.         break;
254.
255.     case 6: // trun left
256.         servo_Angle2 = servo_Angle2 + 1;
257.         if (servo_Angle2 > 180){
258.             servo_Angle2 = 180;
259.         }
260.         Servo_Angle(2, servo_Angle2);
261.         // control_num = 15;
262.         delay(10);
263.         break;
264.     case 7: // trun right
265.         servo_Angle2 = servo_Angle2 - 1;
266.         if (servo_Angle2 < 0){
267.             servo_Angle2 = 0;
268.         }
269.         Servo_Angle(2, servo_Angle2);
270.         // control_num = 15;
271.         delay(10);
272.         break;
273.
274.     case 8: // stop servo rotation.
275.         break;
276.
277.     case 9:
278.         Function_Stop_flag = 0;
279.         Avoid_Obstacles(); // Avoid Obstacles function
280.         break;
281.
282.     case 11:
283.         Function_Stop_flag = 0;
284.         Line_Tracking(); // Line Tracking function
285.         break;
286.     case 13:
```

```
287.     Function_Stop_flag = 0;
288.     Light_Tracking(); // Light Tracking function
289.     break;
290.     case 15:
291.         Function_Stop_flag = 0;
292.         Keep_Distance(); // Follow function
293.         break;
294.
295.     case 10:
296.         // Serial.println("judg = 10:");
297.         Function_Stop_flag = 1;
298.         Motor(1, 1, 0);
299.         Motor(2, 1, 0);
300.         Motor(3, 1, 0);
301.         Motor(4, 1, 0);
302.         break;
303.     case 12:
304.         Function_Stop_flag = 2;
305.         Motor(1, 1, 0);
306.         Motor(2, 1, 0);
307.         Motor(3, 1, 0);
308.         Motor(4, 1, 0);
309.         break;
310.     case 14:
311.         Function_Stop_flag = 3;
312.         Motor(1, 1, 0);
313.         Motor(2, 1, 0);
314.         Motor(3, 1, 0);
315.         Motor(4, 1, 0);
316.         break;
317.     case 16:
318.         Function_Stop_flag = 4;
319.         Motor(1, 1, 0);
320.         Motor(2, 1, 0);
321.         Motor(3, 1, 0);
322.         Motor(4, 1, 0);
323.         break;
324.
325.
326.     default:
327.         break;
328. }
```

```
329. }
330.
331. int StopFunction(){
332.     while(Serial.available()>0){
333.         comdata += char(Serial.read());
334.         delay(1);
335.     }
336.     if (comdata.length() > 0){
337.         if(comdata.endsWith(Ultrasonic_OFF)){// Stop Avoid Obstacles function.
338.             Function_Stop_flag = 1;
339.             judge=10;
340.         }
341.         else if(comdata.endsWith(Line_Tracking_OFF)){ // Stop Line Tracking function.
342.             Function_Stop_flag = 2;
343.             judge=12;
344.         }
345.         else if(comdata.endsWith(Light_Tracking_OFF)){ // Stop Light Tracking function.
346.             Function_Stop_flag = 3;
347.             judge=14;
348.         }
349.         else if(comdata.endsWith(Follow_OFF)){ // Stop Follow function.
350.             Function_Stop_flag = 4;
351.             judge=16;
352.         }
353.         comdata = "";
354.         delay(10);
355.     }
356. }
357.
358. void Avoid_Obstacles(){
359.     while (1){
360.         StopFunction();
361.
362.         Serial.print("11judge:");
363.         Serial.println(judge);
364.         Serial.print("11Function_Stop_flag:");
365.         Serial.println(Function_Stop_flag);
366.         if (Function_Stop_flag == 1){ // Press OK, stop function.
367.             break;
368.         }
```

```
369.
370.     distance = GetDistance();
371.     Serial.print(distance);
372.     // Motor(1,1,0); //Stop the car
373.     // Motor(2,1,0);
374.     // Motor(3,1,0); //Stop the car
375.     // Motor(4,1,0);
376.
377.     if (distance > 30){
378.         Servo_Angle(2, 95);
379.         Motor(1,1,Speed); //forward
380.         Motor(2,1,Speed);
381.         Motor(3,1,Speed); //forward
382.         Motor(4,1,Speed);
383.     }
384.     else if (distance >= 10 and distance <=30){
385.         Motor(1,1,0); //Stop the car
386.         Motor(2,1,0);
387.         Motor(3,1,0); //Stop the car
388.         Motor(4,1,0);
389.         Servo_Angle(2, 50);
390.         if (distance > 20){
391.             Motor(1,-1,Speed); //Stop the car
392.             Motor(2,-1,Speed);
393.             Motor(3,1,Speed); //Stop the car
394.             Motor(4,1,Speed);
395.             // delay(2000);
396.         }
397.         else{
398.             Servo_Angle(2, 150);
399.             Motor(1,1,Speed); //Stop the car
400.             Motor(2,1,Speed);
401.             Motor(3,-1,Speed); //Stop the car
402.             Motor(4,-1,Speed);
403.             // delay(2000);
404.         }
405.     }
406.     // else if (distance >= 10 and distance <=30){
407.     else {
408.         Motor(1,-1,Speed); //Stop the car
409.         Motor(2,-1,Speed);
410.         Motor(3,-1,Speed); //Stop the car
```

```
411.     Motor(4, -1, Speed);
412.     }
413. }
414. }
415.
416.
417. void Light_Tracking(){
418.     value_Init = GetPhotosensitive();
419.     while (1){
420.         StopFunction();
421.         if (Function_Stop_flag == 3){ // Press OK, stop function.
422.             break;
423.         }
424.         value = GetPhotosensitive();
425.         if (value < (value_Init - threshold)){ //left
426.             Motor(1, -1, motor_speed);
427.             Motor(2, -1, motor_speed);
428.             Motor(3, 1, motor_speed);
429.             Motor(4, 1, motor_speed);
430.             Serial.print(value_Init);
431.             Serial.print(":");
432.             Serial.println(value);
433.
434.         }
435.         else if (value > (value_Init + threshold)){ //right
436.             Motor(1, 1, motor_speed);
437.             Motor(2, 1, motor_speed);
438.             Motor(3, -1, motor_speed);
439.             Motor(4, -1, motor_speed);
440.
441.             Serial.print(value_Init);
442.             Serial.print(":");
443.             Serial.println(value);
444.         }
445.         else{
446.             Motor(1, 1, motor_speed);
447.             Motor(2, 1, motor_speed);
448.             Motor(3, 1, motor_speed);
449.             Motor(4, 1, motor_speed);
450.             Serial.print(value_Init);
451.             Serial.print(":");
452.             Serial.println(value);
```



```
453.     }
454.   }
455. }
456.
457.
458. void Line_Tracking(){
459.   int value;
460.   while (1){
461.     // StopFunction();
462.     if (Function_Stop_flag == 2){ // Press OK, stop function.
463.       break;
464.     }
465.     value = Track_Read(); //Read the value of the tracking module.
466.     // Serial.println(value);
467.     switch (value)
468.     {
469.       case 0: //000 Left
470.         Motor(1, -1, motor_speed);
471.         Motor(2, -1, motor_speed);
472.         Motor(3, 1, motor_speed);
473.         Motor(4, 1, motor_speed);
474.         if (Track_value != 0){
475.           display.clearDisplay();
476.           display.setTextSize(2);
477.           display.setCursor(0,0);
478.           display.print("Control: Stop");
479.           display.setCursor(0,30);
480.           display.print("Value: 0 0 0");
481.           display.setTextSize(1);
482.           display.setCursor(20,50);
483.           display.print("www.adeept.com");
484.           display.display();
485.         }
486.         Track_value = 0;
487.         break;
488.
489.       case 1: //010 forward
490.         Motor(1, 1, motor_speed); //Motor1 forward
491.         Motor(2, 1, motor_speed); //Motor2 forward
492.         Motor(3, 1, motor_speed); //Motor3 forward
493.         Motor(4, 1, motor_speed); //Motor4 forward
494.         if (Track_value != 2){
```

```
495.         display.clearDisplay();
496.         display.setTextSize(2);
497.         display.setCursor(0,0);
498.         display.print("Control: Forward");
499.         display.setCursor(0,30);
500.         display.print("Value: 0 1 0");
501.         display.setTextSize(1);
502.         display.setCursor(20,50);
503.         display.print("www.adeept.com");
504.         display.display();
505.     }
506.     Track_value = 2;
507.     break;
508.
509.     case 2:    //100 Left
510.         Motor(1, -1, motor_speed);
511.         Motor(2, -1, motor_speed);
512.         Motor(3, 1, motor_speed);
513.         Motor(4, 1, motor_speed);
514.         if (Track_value != 4){
515.             display.clearDisplay();
516.             display.setTextSize(2);
517.             display.setCursor(0,0);
518.             display.print("Control: Left");
519.             display.setCursor(0,30);
520.             display.print("Value: 1 0 0");
521.             display.setTextSize(1);
522.             display.setCursor(20,50);
523.             display.print("www.adeept.com");
524.             display.display();
525.         }
526.         Track_value = 4;
527.         break;
528.
529.     case 3:    //110 Left
530.         Motor(1, -1, motor_speed);
531.         Motor(2, -1, motor_speed);
532.         Motor(3, 1, motor_speed);
533.         Motor(4, 1, motor_speed);
534.         if (Track_value != 6){
535.             display.clearDisplay();
536.             display.setTextSize(2);
```

```
537.         display.setCursor(0,0);
538.         display.print("Control: Left");
539.         display.setCursor(0,30);
540.         display.print("Value: 1 1 0");
541.         display.setTextSize(1);
542.         display.setCursor(20,50);
543.         display.print("www.adeept.com");
544.         display.display();
545.     }
546.     Track_value = 6;
547.     break;
548.
549.     case 4:    //001 right
550.         Motor(1, 1, motor_speed);
551.         Motor(2, 1, motor_speed);
552.         Motor(3, -1, motor_speed);
553.         Motor(4, -1, motor_speed);
554.         if (Track_value != 1){
555.             display.clearDisplay();
556.             display.setTextSize(2);
557.             display.setCursor(0,0);
558.             display.print("Control: Right");
559.             display.setCursor(0,30);
560.             display.print("Value: 0 0 1");
561.             display.setTextSize(1);
562.             display.setCursor(20,50);
563.             display.print("www.adeept.com");
564.             display.display();
565.         }
566.         Track_value = 1;
567.         break;
568.
569.     case 5:    //011 right
570.         Motor(1, 1, motor_speed);
571.         Motor(2, 1, motor_speed);
572.         Motor(3, -1, motor_speed);
573.         Motor(4, -1, motor_speed);
574.         if (Track_value != 3){
575.             display.clearDisplay();
576.             display.setTextSize(2);
577.             display.setCursor(0,0);
578.             display.print("Control: Right");
```

```
579.         display.setCursor(0,30);
580.         display.print("Value: 0 1 1");
581.         display.setTextSize(1);
582.         display.setCursor(20,50);
583.         display.print("www.adeept.com");
584.         display.display();
585.     }
586.     Track_value = 3;
587.     break;
588.
589.     case 6:    //111 stop
590.         // Serial.println("1111111111111111");
591.         Motor(1, 1, 0);
592.         Motor(2, 1, 0);
593.         Motor(3, 1, 0);
594.         Motor(4, 1, 0);
595.
596.         if (Track_value != 7){
597.             display.clearDisplay();
598.             display.setTextSize(2);
599.             display.setCursor(0,0);
600.             display.print("Control: Stop");
601.             display.setCursor(0,30);
602.             display.print("Value: 1 1 1");
603.             display.setTextSize(1);
604.             display.setCursor(20,50);
605.             display.print("www.adeept.com");
606.             display.display();
607.         }
608.         Track_value = 7;
609.         break;
610.     case 7:    //101 forward
611.         Motor(1, 1, motor_speed); //Motor1 forward
612.         Motor(2, 1, motor_speed); //Motor2 forward
613.         Motor(3, 1, motor_speed); //Motor3 forward
614.         Motor(4, 1, motor_speed); //Motor4 forward
615.         if (Track_value != 5){
616.             display.clearDisplay();
617.             display.setTextSize(2);
618.             display.setCursor(0,0);
619.             display.print("Control: Forward");
620.             display.setCursor(0,30);
```

```
621.         display.print("Value: 1 0 1");
622.         display.setTextSize(1);
623.         display.setCursor(20,50);
624.         display.print("www.aadept.com");
625.         display.display();
626.     }
627.     Track_value = 5;
628.     break;
629.     default:
630.         break;
631. }
632. // delay(100);
633. }
634. }
635.
636. void Keep_Distance(){
637.     Servo_Angle(2, servo_Angle2);
638.     while (1){
639.         // StopFunction();
640.         if (Function_Stop_flag == 4){ // Press OK, stop function.
641.             break;
642.         }
643.         distance = GetDistance();
644.         if (distance < 30){
645.             Motor(1,-1,Speed); //backward
646.             Motor(2,-1,Speed);
647.             Motor(3,-1,Speed); //backward
648.             Motor(4,-1,Speed);
649.         }
650.         else if (distance > 40){
651.             Motor(1,1,Speed); //forward
652.             Motor(2,1,Speed);
653.             Motor(3,1,Speed); //forward
654.             Motor(4,1,Speed);
655.         }
656.         else {
657.             Motor(1,1,0); // stop
658.             Motor(2,1,0);
659.             Motor(3,1,0); // stop
660.             Motor(4,1,0);
661.         }
662.         delay(100);
```



```
663.    }
```

```
664.    }
```