

Lesson 14 How to use the IR module to Control the Car





In the last lesson we know how to use IR module to pass data with Adeept Robot Control Board. In this course we learn how to use infrared remote control to control the car.

14.1 Components used in this course

Please assemble Uno-Car according to the assembly tutorial and connect the circuit correctly.

14.2 Introduction of IR Control function

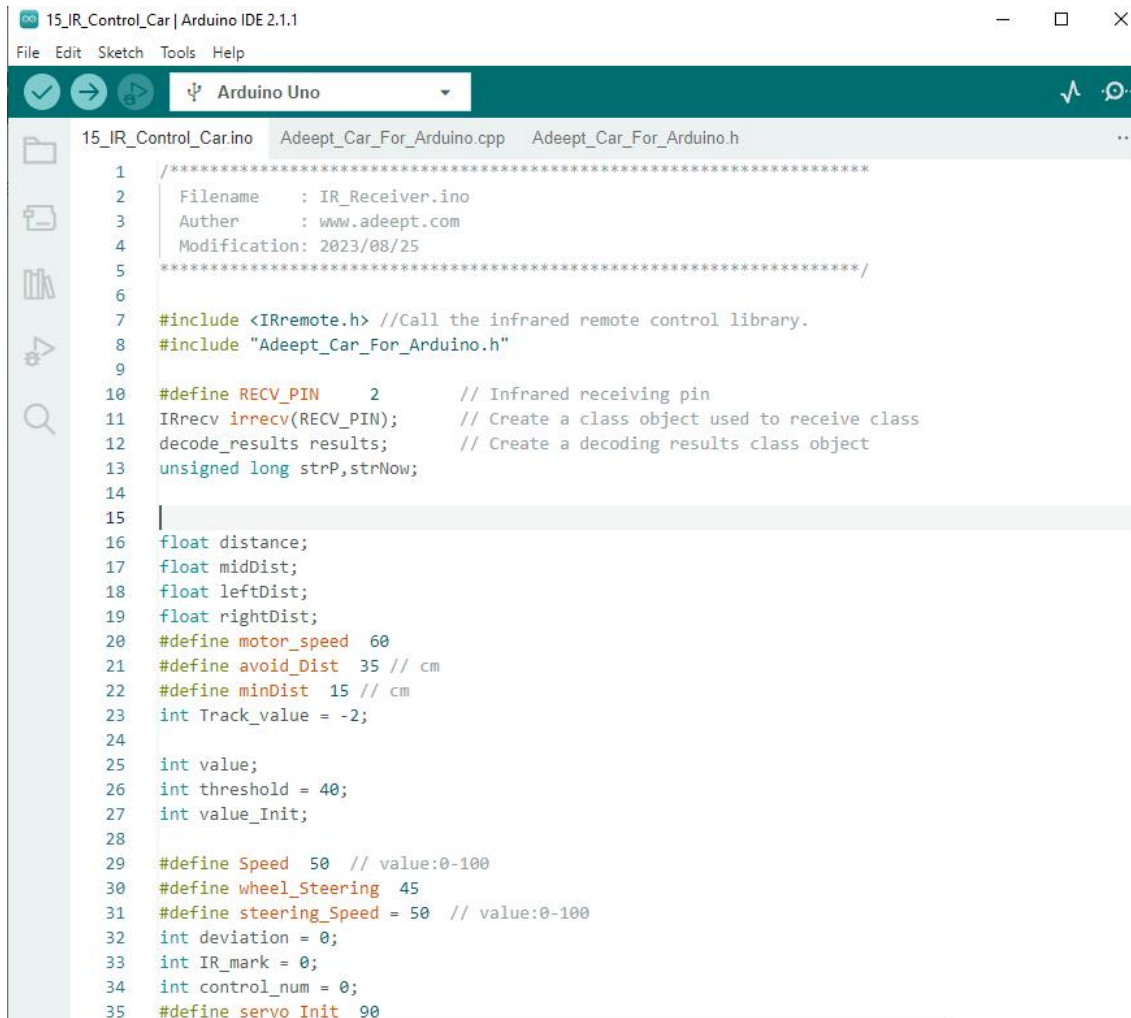
The corresponding relationship between the buttons of the infrared remote control and the functions of the car is as follows:

Button	Function	Button	Function
	Forward	5	Buzzer beep
	Backward	6	Turn off LED
	Left	7	
	Right	8	Matrix screen lights up
OK	Stop Function	9	
1	Servo 2 turn left	*	Avoid Obstacles Function
2	Servo 2 turn to 90	0	Matrix screen turns off
3	Servo 2 turn right	#	Light Tracking Function
4	Change the color of LED		



14.3 How to use the infrared control function

1. Connect your computer and Adeept Robot Control Board with a USB cable.
2. Open “14_IR_Control_car” folder in “/Code”, double-click “14_IR_Control_car.ino”.



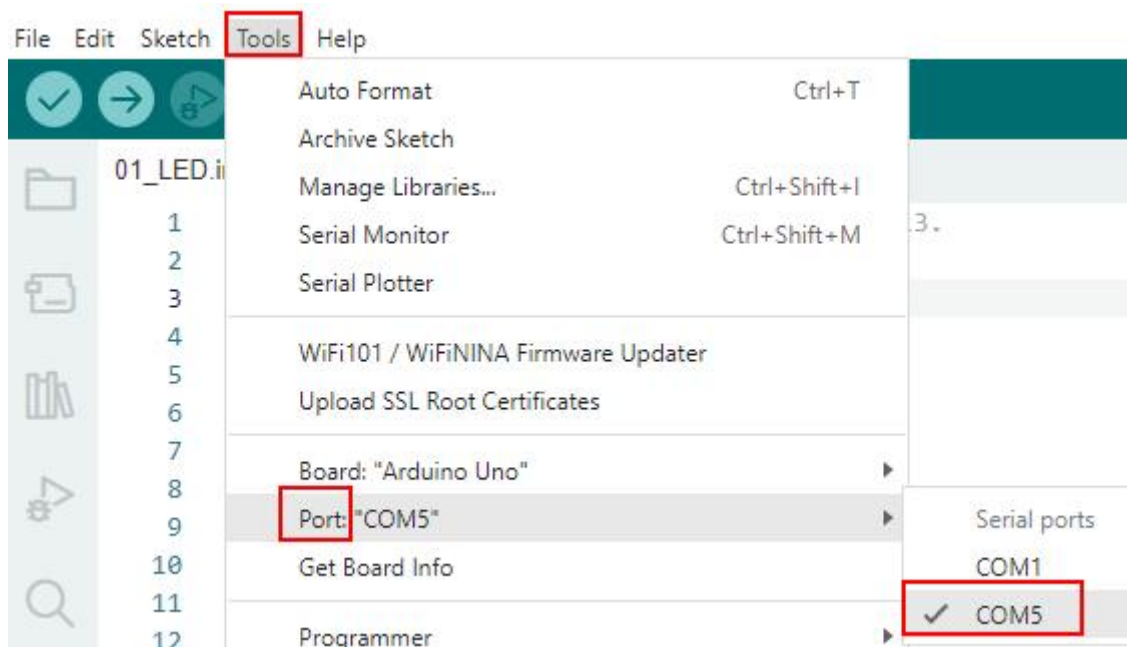
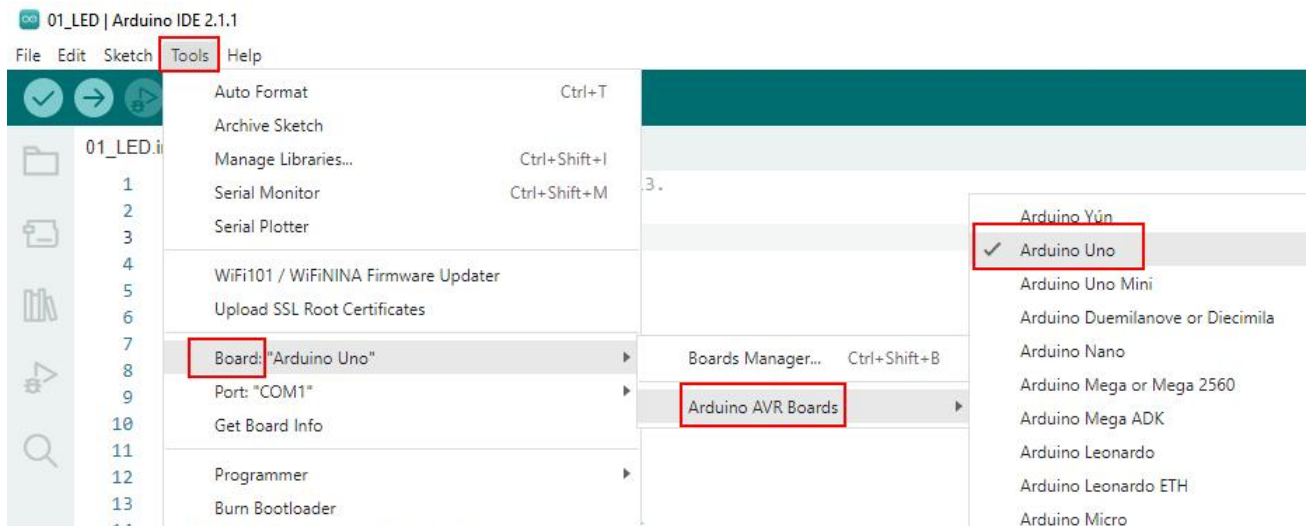
```
15_IR_Control_Car | Arduino IDE 2.1.1
File Edit Sketch Tools Help
Arduino Uno
15_IR_Control_Car.ino Adeept_Car_For_Arduino.cpp Adeept_Car_For_Arduino.h
1  /*****
2  Filename   : IR_Receiver.ino
3  Author    : www.adeept.com
4  Modification: 2023/08/25
5  *****/
6
7  #include <IRremote.h> //Call the infrared remote control library.
8  #include "Adeept_Car_For_Arduino.h"
9
10 #define RECV_PIN 2 // Infrared receiving pin
11 IRrecv irrecv(RECV_PIN); // Create a class object used to receive class
12 decode_results results; // Create a decoding results class object
13 unsigned long strP, strNow;
14
15
16 float distance;
17 float midDist;
18 float leftDist;
19 float rightDist;
20 #define motor_speed 60
21 #define avoid_Dist 35 // cm
22 #define minDist 15 // cm
23 int Track_value = -2;
24
25 int value;
26 int threshold = 40;
27 int value_Init;
28
29 #define Speed 50 // value:0-100
30 #define wheel_Steering 45
31 #define steering_Speed = 50 // value:0-100
32 int deviation = 0;
33 int IR_mark = 0;
34 int control_num = 0;
35 #define servo_Init 90
```


3. Select development board and serial port.

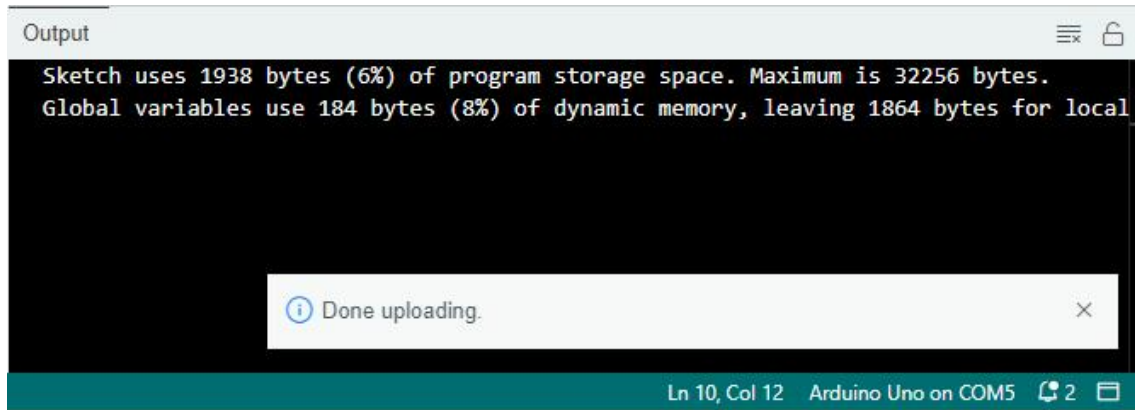
Board: **Tools**--->**Board**--->**Arduino AVR Boards**--->**Arduino Uno**

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

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

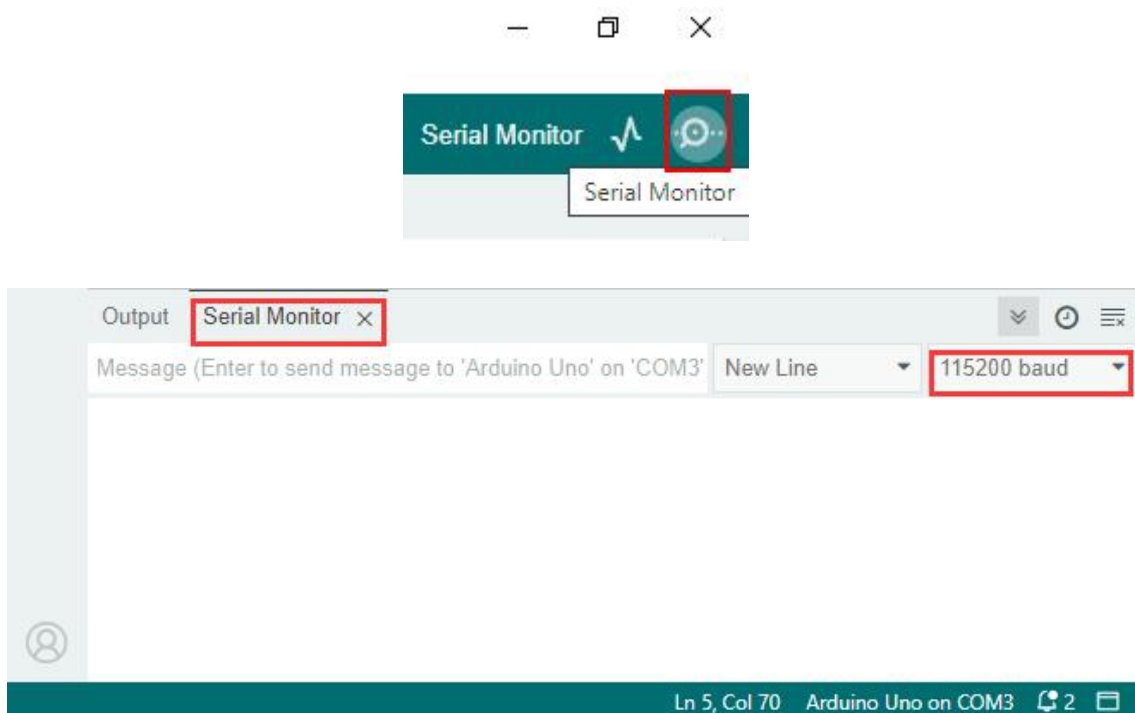


4. After opening, click  to upload the code program to the Arduino. If there is no error warning in the console below, it means that the Upload is successful.







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.



6. Use the IR remote to aim at the IR receiver on the expansion board. Press different buttons on the infrared remote control, you can see that the car has different behaviors.

Button	Function	Button	Function
	Forward	4	Change the color of LED
	Backward	5	Buzzer beep
	Left	6	Turn off LED
	Right	7	
OK	Stop Function (click multiple times)	8	Matrix screen lights up
		9	
1	Servo 2 turn left	*	Avoid Obstacles Function
2	Servo 2 turn to 90	0	Matrix screen turns off
3	Servo 2 turn right	#	Light Tracking Function
Some Functions may require multiple clicks to stop.			

Please familiarize yourself with the button functions of the infrared remote control first. Please see the list above for the button functions.

It is recommended to raise the vehicle body during testing so that the wheels are suspended in the air.

14.4 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 [14_IR_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 [14_IR_Control_car.ino](#) program content, which may be subject to change. Please refer to the actual code provided.

```
1.  #include <IRremote.h> //Call the infrared remote control library.
2.  #include "Adeept_Car_For_Arduino.h"
3.
4.  // #include <Adafruit_NeoPixel.h>
5.  #include <Adafruit_GFX.h>
6.  #include <Adafruit_SSD1306.h>
7.
8.  #define RECV_PIN    2          // Infrared receiving pin
9.  IRrecv irrecv(RECV_PIN);      // Create a class object used to receive class
10. decode_results results;        // Create a decoding results class object
11. unsigned long strP,strNow;
12.
13. #define OLED_RESET    4
14. Adafruit_SSD1306 display(128, 64, &Wire, OLED_RESET);
15.
16. float distance;
17. float midDist;
18. float leftDist;
19. float rightDist;
20. #define motor_speed  60
21. #define avoid_Dist  35 // cm
22. #define minDist  15 // cm
23. int Track_value = -2;
24.
25. int value;
26. int threshold = 40;
27. int value_Init;
28.
29. #define Speed  50 // value:0-100
30. #define wheel_Steering  45
```

```
31. #define steering_Speed = 50 // value:0-100
32. int deviation = 0;
33. int IR_mark = 0;
34. int control_num = 0;
35. #define servo_Init 90
36. int servo_Angle2 = servo_Init;
37. int ws2812_flag= 0;
38. int Function_flag = 0;
39.
40. // int time1;
41. // int time2;
42.
43. byte color_value[][3] = {{250,0,0},{250,160,0},{250,250,0},{0,250,0},{0,250,250},
    },{0,0,250},{130,0,250}};
44. void setup()
45. {
46.     Serial.begin(115200); // Initialize the serial port and set the baud rate to 115200
47.
48.
49.     Servo_Setup(); //Servo initialization
50.     PCA9685_Servo_Setup(); //PCA9685 Servo initialization
51.     Motor_Setup(); //Motor initialization
52.     AllMotorStop();
53.     Buzzer_Setup(); //Buzzer initialization
54.     WS2812_Setup(); //WS2812 LED initialization
55.     WS2812_Brightness(5); // value 0-10
56.     Ultrasonic_Setup(); //Ultrasonic initialization
57.     Photosensitive_Setup(); //Light line initialization
58.     Tracking_Setup(); //Tracking Line initialization
59.     // OLED_Setup(); //OLED initialization
60.     Matrix_Setup();
61.     Matrix_Clear();
62.
63.     irrecv.enableIRIn(); // Start the receiver
64.     Serial.println("UnoCar-B IR Control Start!");
65.     // Serial.println(RECV_PIN); //print the infrared receiving pin
66.
67.     // Buzzer_Alert(1,1);
68.     WS2812ColorAll(255, 255,0); // Green
69.     Servo_Angle(1, 90);
70.     Servo_Angle(2, 90);
```

```
71. PCA9685_Servo_Angle(6, 0, 90);
72. PCA9685_Servo_Angle(7, 0, 90);
73. Buzzer_Silence();
74. // OLED_clear();
75. delay(1000);
76.
77. WS2812ColorAll(0,0,0);
78.
79. display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
80. display.setTextColor(WHITE);//Sets the font display color
81. display.clearDisplay();//cls
82. }
83.
84. void loop()
85. {
86.   // if (irrecv.decode(&results)) {           // Waiting for decoding
87.   if (irrecv.decode(&results)){
88.     // if (irrecv.decode()){
89.     unsigned long value = results.value;
90.
91.     value = switch_irr(value);
92.
93.     Serial.println(value);
94.     // Serial.println(value, HEX);           // Print out the decoded results
95.
96.     irrecv.resume(); // Release the IRremote. Receive the
       next value
97.     // time1 = millis();
98.     control(value);
99.     // time2 = millis() - time1;
100.    // Serial.print("time:");
101.    // Serial.println(time2);
102.    // irrecv.resume();           // Release the IRremote. Receive t
       he next value
103.
104.    // delay(100);
105.   }
106.
107.   else{
108.     delay(60); //delayed judgment.
109.     if (irrecv.decode(&results)==false){
110.       // if (irrecv.decode()==false){
```



```
111.     Motor(1, 1, 0);
112.     Motor(2, 1, 0);
113.     Motor(3, 1, 0);
114.     Motor(4, 1, 0);
115. }
116. }
117. delay(100);
118. }
119.
120.
121. void control(unsigned long value){
122.     switch (value) {
123.         // Move Control
124.         case 12: // Signal when UP is pressed, action: forward
125.             Motor(1, 1, motor_speed); //Motor1 forward
126.             Motor(2, 1, motor_speed); //Motor2 forward
127.             Motor(3, 1, motor_speed); //Motor3 forward
128.             Motor(4, 1, motor_speed); //Motor4 forward
129.             control_num = 12;
130.
131.
132.             // display.clearDisplay();
133.             // display.setTextSize(2);
134.             // display.setCursor(0,0);
135.             // display.print("Control: Stop");
136.             // display.setCursor(0,30);
137.             // display.print("Value: 0 0 0");
138.             // display.setTextSize(1);
139.             // display.setCursor(20,50);
140.             // display.print("www.adeept.com");
141.             // display.display();
142.             break;
143.
144.         case 13: // Down,
145.             Motor(1, -1, motor_speed); //Motor1 backward
146.             Motor(2, -1, motor_speed); //Motor2 backward
147.             Motor(3, -1, motor_speed); //Motor3 backward
148.             Motor(4, -1, motor_speed); //Motor4 backward
149.             control_num = 13;
150.
151.             break;
152.
```

```
153.     case 14: // left,
154.         Motor(1, -1, motor_speed);
155.         Motor(2, -1, motor_speed);
156.         Motor(3, 1, motor_speed);
157.         Motor(4, 1, motor_speed);
158.         control_num = 14;
159.
160.         break;
161.
162.     case 15: // right
163.         Motor(1, 1, motor_speed);
164.         Motor(2, 1, motor_speed);
165.         Motor(3, -1, motor_speed);
166.         Motor(4, -1, motor_speed);
167.         control_num = 15;
168.
169.         break;
170.
171.     // case 4294967295:
172.     //     break;
173.
174.     // case 7:
175.     //     Servo_Angle(1, servo_Init + deviation + wheel_Steering); // left
176.     //     Motor(1, -1, motor_speed);
177.     //     Motor(2, -1, motor_speed);
178.     //     control_num = 7;
179.     //     break;
180.
181.     // case 9:
182.     //     Servo_Angle(1, servo_Init + deviation - wheel_Steering); // right
183.     //     Motor(1, -1, motor_speed);
184.     //     Motor(2, -1, motor_speed);
185.     //     control_num = 8;
186.     //     break;
187.
188.     // servo 2 control.
189.     case 1:
190.         servo_Angle2 = servo_Angle2 + 5;
191.         if (servo_Angle2 > 180){
192.             servo_Angle2 = 180;
193.         }
194.         Servo_Angle(2, servo_Angle2);
```

```
195.    // Serial.println(servo_Angle2);
196.    break;
197.
198.    case 2:
199.        Servo_Angle(2, servo_Init);
200.        servo_Angle2 = servo_Init;
201.        break;
202.
203.    case 3:
204.        servo_Angle2 = servo_Angle2 - 5;
205.        if (servo_Angle2 < 0){
206.            servo_Angle2 = 0;
207.        }
208.        Servo_Angle(2, servo_Angle2);
209.        // Serial.println(servo_Angle);
210.        break;
211.
212.    // LED
213.    case 4:
214.        WS2812ColorAll(color_value[ws2812_flag][0], color_value[ws2812_flag][1],
        color_value[ws2812_flag][2]);
215.        // All_RGB(color_value[ws2812_flag][0], color_value[ws2812_flag][1], colo
        r_value[ws2812_flag][2]);
216.        ws2812_flag++;
217.        // LED_status = 1;
218.        if(ws2812_flag > 6){
219.            ws2812_flag = 0;
220.        }
221.        break;
222.
223.    case 6:
224.        WS2812ColorAll(0,0,0);
225.        // All_RGB(0,0,0);
226.        // for (int i = 0; i < LEDS_COUNT; i++){
227.        //     strip.setLedColorData(i, 0, 0, 0);
228.        //     strip.show();
229.        // }
230.        // ws2812_flag = 0;
231.        // LED_status = 0;
232.        break;
233.
234.    case 5:
```

```
235.     Buzzer_Alert(3, 1);
236.     break;
237.
238.     case 8:
239.         Matrix_Smile();
240.         break;
241.
242.     case 0:
243.         Matrix_Clear();
244.         break;
245.
246.     // Function.
247.     case 10:
248.         Function_flag = 0;
249.         Avoid_Obstacles(); // Avoid Obstacles function
250.         break;
251.
252.     case 11:
253.         Function_flag = 0;
254.         Light_Tracking(); // Light Tracking function
255.         break;
256.
257.     // case 11:
258.         // Line_Tracking(); // Line Tracking functioin
259.         // Light_Tracking();
260.         // break;
261.
262.     case 16: // OK. Stop function.
263.         Servo_Angle(1, servo_Init + deviation);
264.         Motor(1, 1, 0);
265.         Motor(2, 1, 0);
266.         control_num = -1;
267.         break;
268.
269.     default:
270.         break;
271. }
272. }
273.
274. int StopFunction(){
275.     if (irrecv.decode(&results)) { // Waiting for decoding
276.         unsigned long value = results.value;
```

```
277.     value = switch_irr(value);
278.     // Serial.println(value);
279.     if (value == 16){ // OK, Stop function.
280.         Function_flag = 1;}
281.     else{
282.         Function_flag = 0;
283.     }
284.     irrecv.resume();                // Release the IRremote. Receive the
        next value
285. }
286. // delay(100);
287. }
288.
289.
290. // void Keep_Distance(){
291. //     Servo_Angle(2, servo_Angle2);
292. //     while (1){
293. //         StopFunction();
294. //         if (Function_flag == 1){ // Press OK, stop function.
295. //             break;
296. //         }
297. //         distance = GetDistance();
298. //         if (distance < 30){
299. //             Servo_Angle(1, servo_Init + deviation);    // front wheel
300. //             Motor(1,-1,Speed); //forward
301. //             Motor(2,-1,Speed);
302. //         }
303. //         else if (distance > 40){
304. //             Servo_Angle(1, servo_Init + deviation);    // front wheel
305. //             Motor(1,1,Speed); //forward
306. //             Motor(2,1,Speed);
307. //         }
308. //         else {
309. //             Motor(1,1,0); // stop
310. //             Motor(2,1,0);
311. //         }
312. //         delay(100);
313. //     }
314. // }
315.
316. void Avoid_Obstacles(){
317.     while (1){
```

```
318.    StopFunction();
319.    if (Function_flag == 1){ // Press OK, stop function.
320.        break;
321.    }
322.
323.    distance = GetDistance();
324.    Serial.print(distance);
325.    // Motor(1,1,0); //Stop the car
326.    // Motor(2,1,0);
327.    // Motor(3,1,0); //Stop the car
328.    // Motor(4,1,0);
329.
330.    if (distance > 30){
331.        Servo_Angle(2, 95);
332.        Motor(1,1,Speed); //forward
333.        Motor(2,1,Speed);
334.        Motor(3,1,Speed); //forward
335.        Motor(4,1,Speed);
336.    }
337.    else if (distance >= 10 and distance <=30){
338.        Motor(1,1,0); //Stop the car
339.        Motor(2,1,0);
340.        Motor(3,1,0); //Stop the car
341.        Motor(4,1,0);
342.        Servo_Angle(2, 50);
343.        if (distance > 20){
344.            Motor(1,-1,Speed); //Stop the car
345.            Motor(2,-1,Speed);
346.            Motor(3,1,Speed); //Stop the car
347.            Motor(4,1,Speed);
348.            // delay(2000);
349.        }
350.        else{
351.            Servo_Angle(2, 150);
352.            Motor(1,1,Speed); //Stop the car
353.            Motor(2,1,Speed);
354.            Motor(3,-1,Speed); //Stop the car
355.            Motor(4,-1,Speed);
356.            // delay(2000);
357.        }
358.    }
359.    // else if (distance >= 10 and distance <=30){
```

```
360.     else {
361.         Motor(1,-1,Speed); //Stop the car
362.         Motor(2,-1,Speed);
363.         Motor(3,-1,Speed); //Stop the car
364.         Motor(4,-1,Speed);
365.     }
366.     // delay(100);
367. }
368.
369. }
370.
371.
372. void Light_Tracking(){
373.     value_Init = GetPhotosensitive();
374.     while (1){
375.         StopFunction();
376.         if (Function_flag == 1){ // Press OK, stop function.
377.             break;
378.         }
379.         value = GetPhotosensitive();
380.         if (value < (value_Init - threshold)){
381.             Motor(1, -1, motor_speed);
382.             Motor(2, -1, motor_speed);
383.             Motor(3, 1, motor_speed);
384.             Motor(4, 1, motor_speed);
385.             Serial.print(value_Init);
386.             Serial.print(":");
387.             Serial.println(value);
388.
389.         }
390.         else if (value > (value_Init + threshold)){
391.             Motor(1, 1, motor_speed);
392.             Motor(2, 1, motor_speed);
393.             Motor(3, -1, motor_speed);
394.             Motor(4, -1, motor_speed);
395.             Serial.print(value_Init);
396.             Serial.print(":");
397.             Serial.println(value);
398.         }
399.         else{
400.             Motor(1, 1, motor_speed);
401.             Motor(2, 1, motor_speed);
```

```
402.     Motor(3, 1, motor_speed);
403.     Motor(4, 1, motor_speed);
404.     Serial.print(value_Init);
405.     Serial.print(":");
406.     Serial.println(value);
407. }
408. }
409.
410. }
411.
412.
413. int switch_irr(int irr_data)
414. {
415.     switch(irr_data)
416.     {
417.     case 16750695: return 0;
418.     case 16753245: return 1;
419.     case 16736925: return 2;
420.     case 16769565: return 3;
421.     case 16720605: return 4;
422.     case 16712445: return 5;
423.     case 16761405: return 6;
424.     case 16769055: return 7;
425.     case 16754775: return 8;
426.     case 16748655: return 9;
427.     case 16738455: return 10; // *
428.     case 16756815: return 11; // #
429.     case 16718055: return 12; // up
430.     case 16730805: return 13; // down
431.     case 16716015: return 14; // left
432.     case 16734885: return 15; // right
433.     case 16726215: return 16; // ok
434.     }
435. }
```