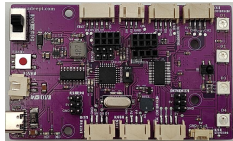




Lesson 13 How to use the Infrared module

In this lesson, we will learn how to use the Infrared module.

13.1 Components used in this course

Components	Quantity	Picture
Adeept Robot Control Board	1	
Type-C USB Cable	1	
IR remote control	1	

13.2 The introduction of the OLED screen

Infrared control is generally divided into two parts, namely the transmitting part and the receiving part, the IR receiver belongs to the receiving part, and the IR remote control belongs to the transmitting part.

There are two important components inside the IR receiver, namely IC and PD. IC is the processing element of the receiving head. It is mainly composed of silicon crystals and circuits.

It is a highly integrated device. Its main functions include filtering, shaping, decoding, and amplification. PD is a photodiode, and its main function is to receive optical signals.

The transmitting circuit of the IR remote control uses infrared light-emitting diodes to emit modulated infrared light waves. Press a certain key of the remote control, the remote control will send out a series of modulated signals. After the signal is received by the infrared integrated module, it outputs the demodulated digital pulse. Each key corresponds to a different pulse, so it can identify different pulses. pulses to identify different keys.

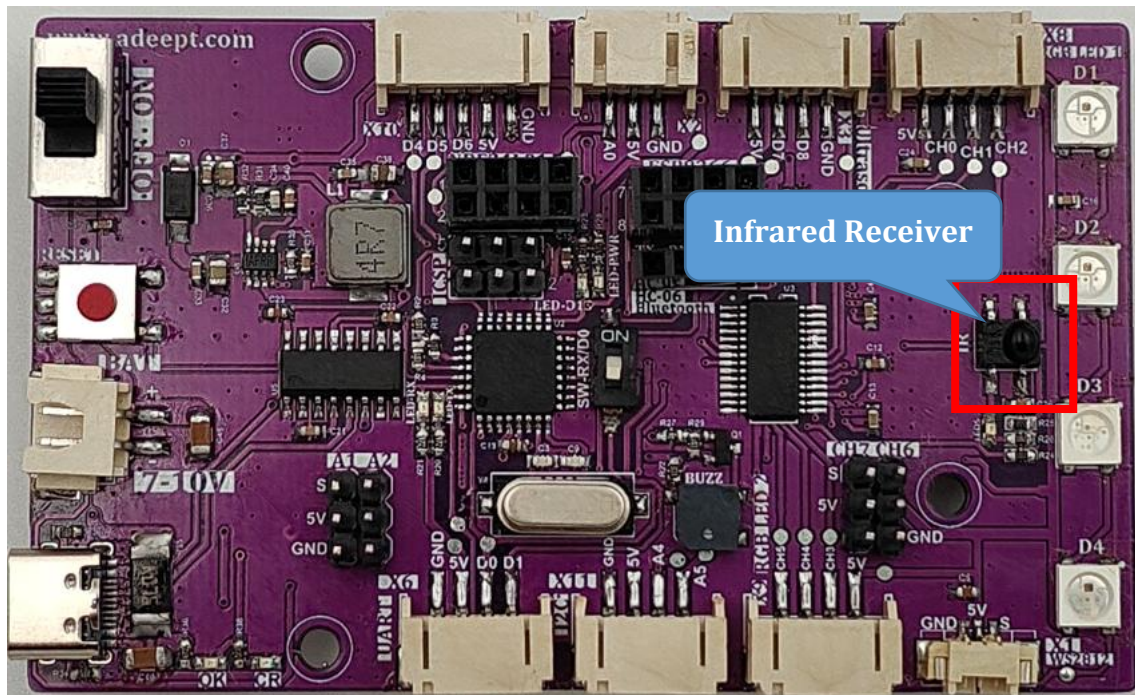
The following is the information in the corresponding program when each button of the infrared remote control is pressed.



Remote control button	Program value
0~9	0~9
*	10
#	11
UP	12
Down	13
Left	14
Right	15
OK	16

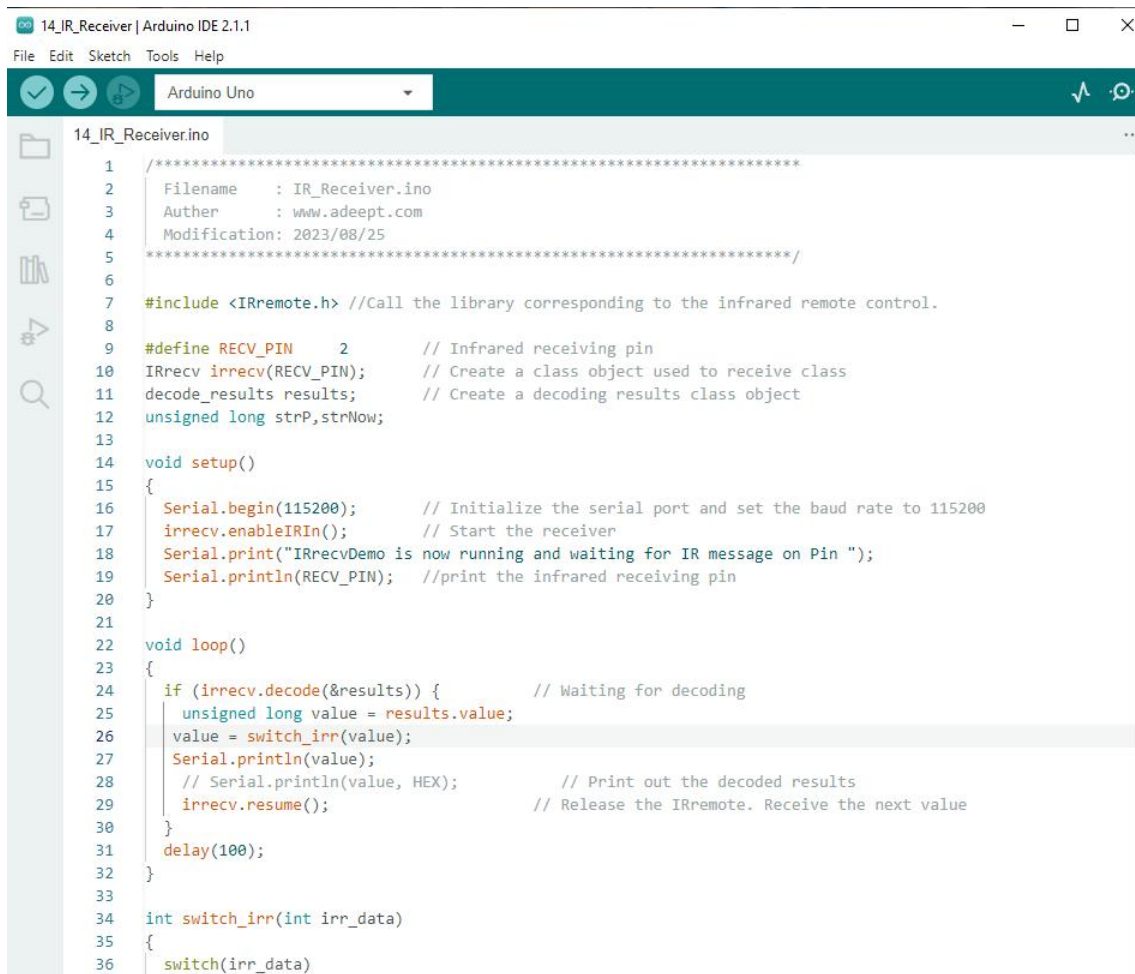
13.3 Wiring diagram

Figure as below:



13.4 How to use the Infrared module

1. Connect your computer and Adeept Robot Control Board with a USB cable.
2. Open "13_IR_receiver" folder in `/Code`, double-click "[13_IR_receiver.ino](#)".



```
14_IR_Receiver | Arduino IDE 2.1.1
File Edit Sketch Tools Help
Arduino Uno

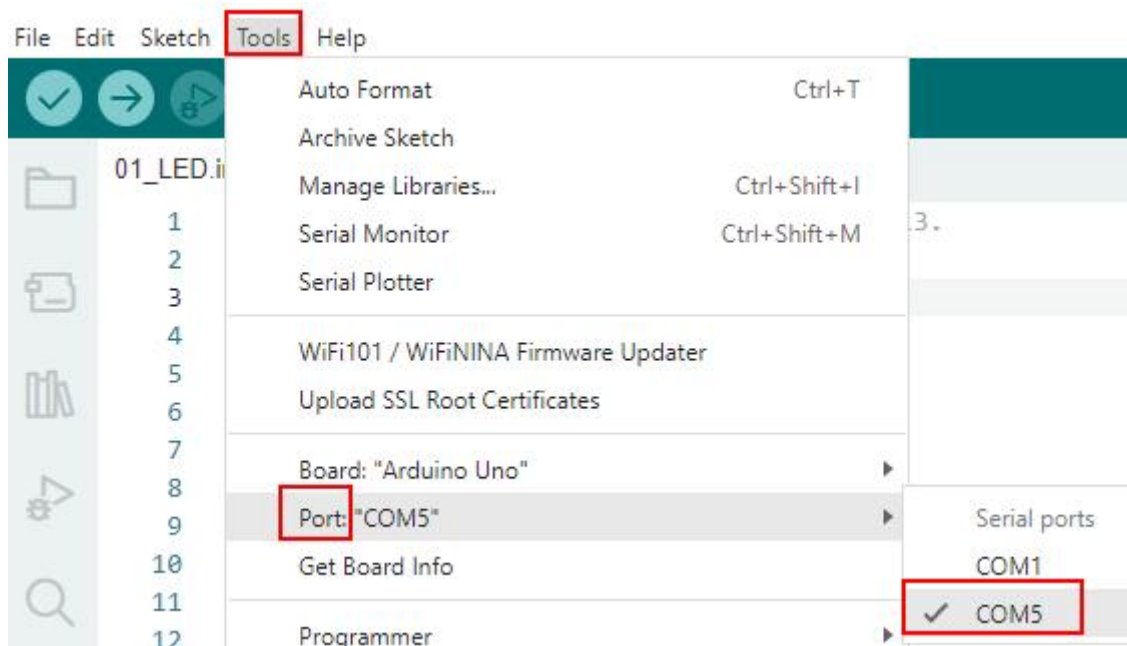
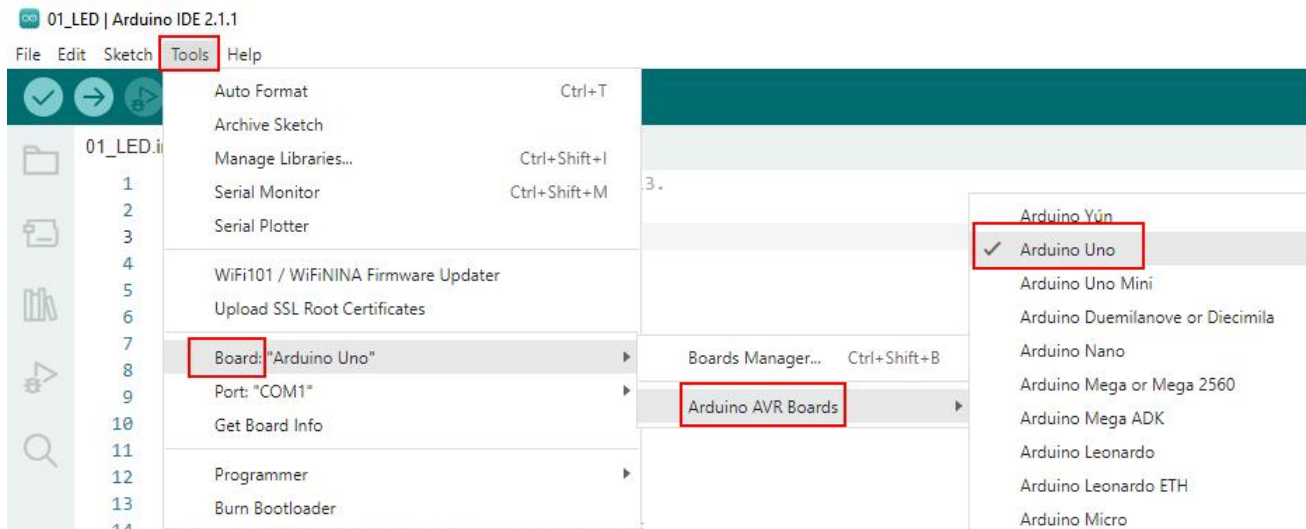
14_IR_Receiver.ino
1  /*****
2  Filename   : IR_Receiver.ino
3  Author    : www.adeept.com
4  Modification: 2023/08/25
5  *****/
6
7  #include <IRremote.h> //Call the library corresponding to the infrared remote control.
8
9  #define RECV_PIN 2 // Infrared receiving pin
10 IRrecv irrecv(RECV_PIN); // Create a class object used to receive class
11 decode_results results; // Create a decoding results class object
12 unsigned long strP, strNow;
13
14 void setup()
15 {
16   Serial.begin(115200); // Initialize the serial port and set the baud rate to 115200
17   irrecv.enableIRIn(); // Start the receiver
18   Serial.print("IRrecvDemo is now running and waiting for IR message on Pin ");
19   Serial.println(RECV_PIN); //print the infrared receiving pin
20 }
21
22 void loop()
23 {
24   if (irrecv.decode(&results)) { // Waiting for decoding
25     unsigned long value = results.value;
26     value = switch_irr(value);
27     Serial.println(value);
28     // Serial.println(value, HEX); // Print out the decoded results
29     irrecv.resume(); // Release the IRremote. Receive the next value
30   }
31   delay(100);
32 }
33
34 int switch_irr(int irr_data)
35 {
36   switch(irr_data)
```


3. Select development board and serial port.

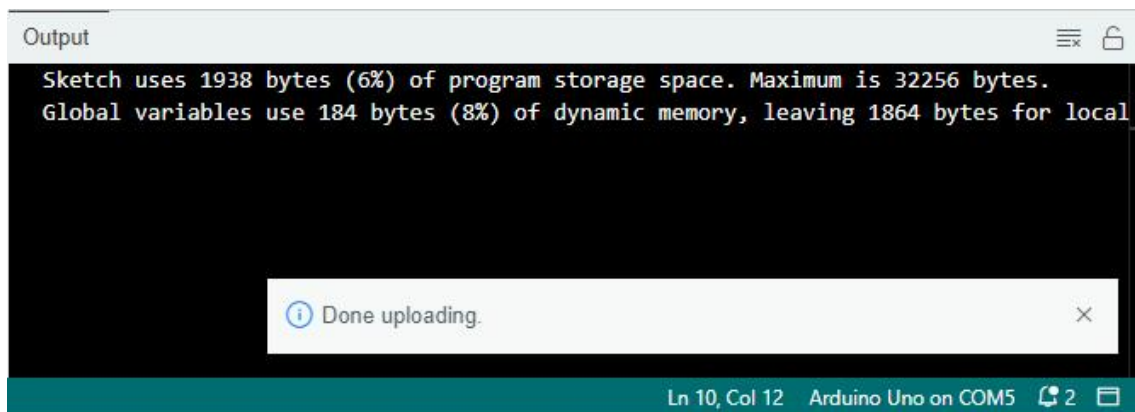
Board: [Tools](#)--->[Board](#)--->[Arduino AVR Boards](#)--->[Arduino Uno](#)

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

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

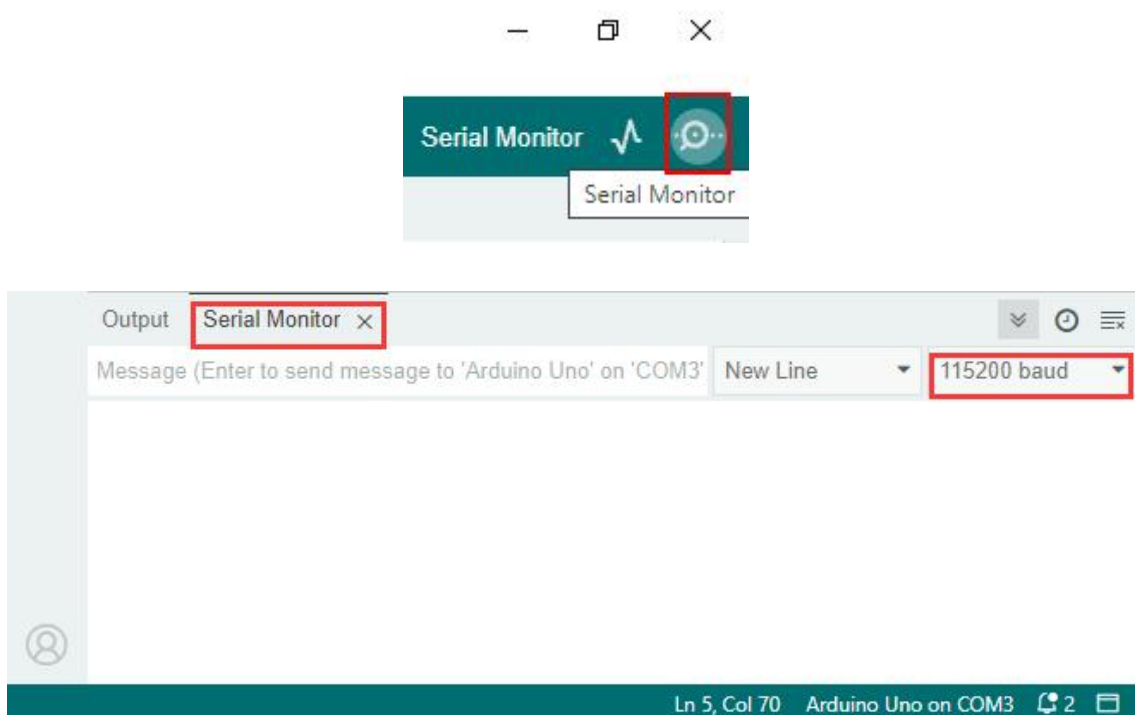


4. After opening, click  to upload the code program to the Arduino. If there is no error warning in the console below, it means that the Upload is successful.

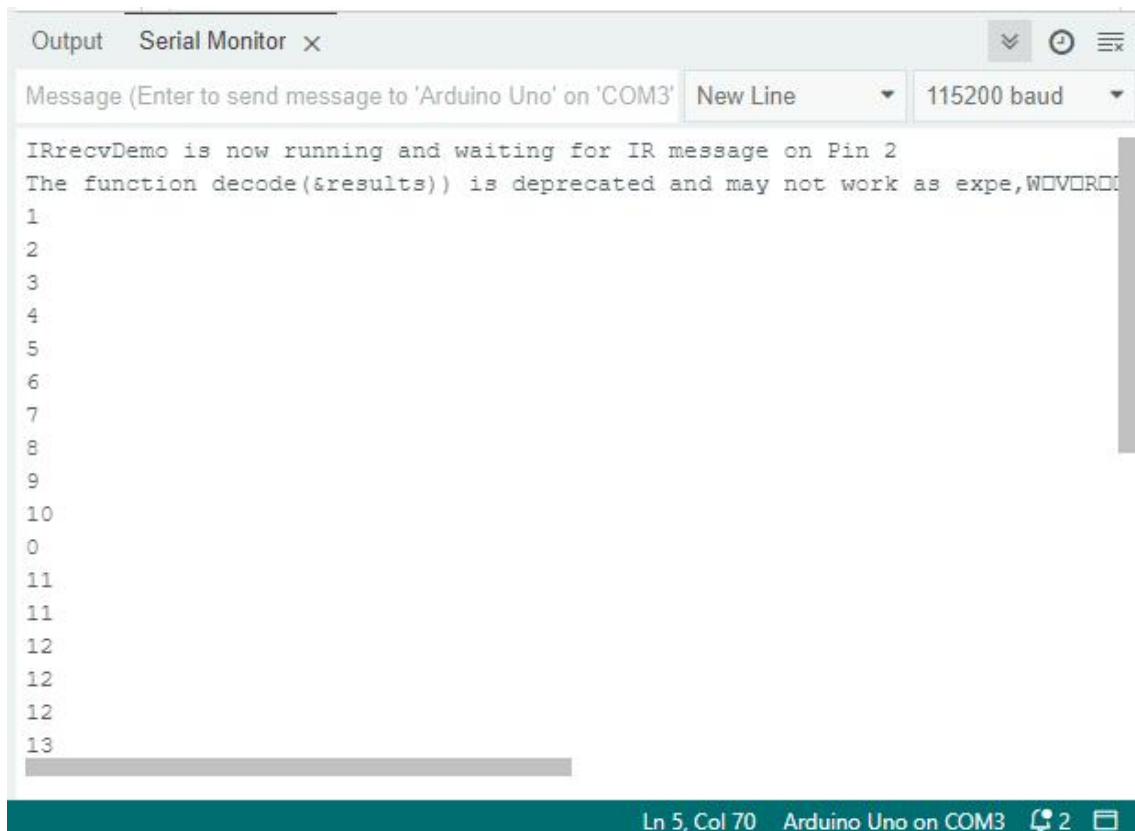


After assembling the car, please use the 18650 battery to provide power when uploading the program, otherwise the program may not be uploaded successfully due to excessive load.

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 a button, and the corresponding button information can be displayed on the screen.



13.5 Code

```
1. #include <IRremote.h> //Call the library corresponding to the infrared remote control.
2.
3. #define RECV_PIN    2      // Infrared receiving pin
4. IRrecv irrecv(RECV_PIN);  // Create a class object used to receive class
5. decode_results results;    // Create a decoding results class object
6. unsigned long strP,strNow;
7.
8. void setup()
9. {
10.   Serial.begin(115200);    // Initialize the serial port and set the baud rate to 115200
11.   irrecv.enableIRIn();     // Start the receiver
12.   Serial.print("IRrecvDemo is now running and waiting for IR message on Pin ");
13.   Serial.println(RECV_PIN); //print the infrared receiving pin
14. }
15.
```

```
16. void loop()
17. {
18.   if (irrecv.decode(&results)) {           // Waiting for decoding
19.     unsigned long value = results.value;
20.     value = switch_irr(value);
21.     Serial.println(value);
22.     // Serial.println(value, HEX);         // Print out the decoded results
23.     irrecv.resume();                       // Release the IRremote. Receive the next value
24.   }
25.   delay(100);
26. }
27.
28. int switch_irr(int irr_data)
29. {
30.   switch(irr_data)
31.   {
32.     case 16750695: return 0;
33.     case 16753245: return 1;
34.     case 16736925: return 2;
35.     case 16769565: return 3;
36.     case 16720605: return 4;
37.     case 16712445: return 5;
38.     case 16761405: return 6;
39.     case 16769055: return 7;
40.     case 16754775: return 8;
41.     case 16748655: return 9;
42.     case 16738455: return 10; // *
43.     case 16756815: return 11; // #
44.     case 16718055: return 12; // up
45.     case 16730805: return 13; // down
46.     case 16716015: return 14; // left
47.     case 16734885: return 15; // right
48.     case 16726215: return 16; // ok
49.   }
50. }
```