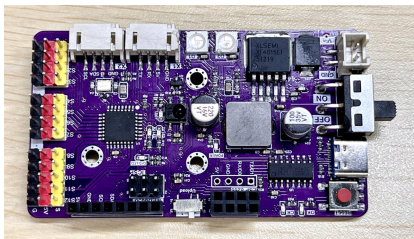


Lesson 8 How to Use the Infrared Module

8.1 Overview

This course mainly introduces the relevant knowledge of infrared modules, including their working principles, hardware connections, and programming applications on the Arduino platform. Be able to master the working mechanism of the transmitting and receiving parts in infrared remote control systems, learn to use infrared modules to achieve remote control functions of equipment, and lay the foundation for developing various infrared control projects.

8.2 Required Components

Components	Quantity	Picture
Adeept Pixie Board	1	
Type-C USB Cable	1	
IR remote control	1	
OLED Screen	1	

8.3 Principle Introduction

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.

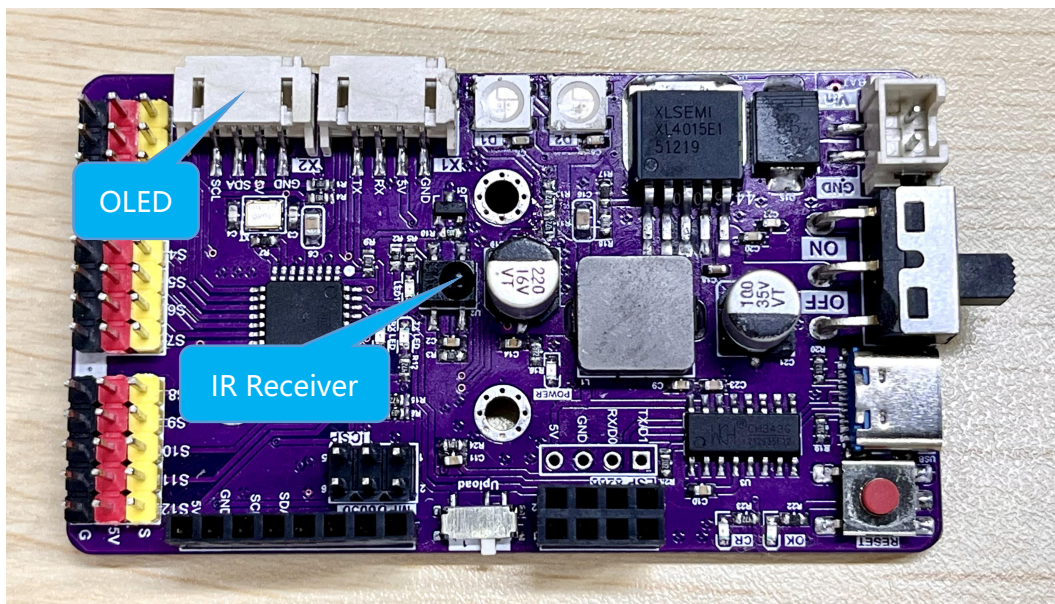
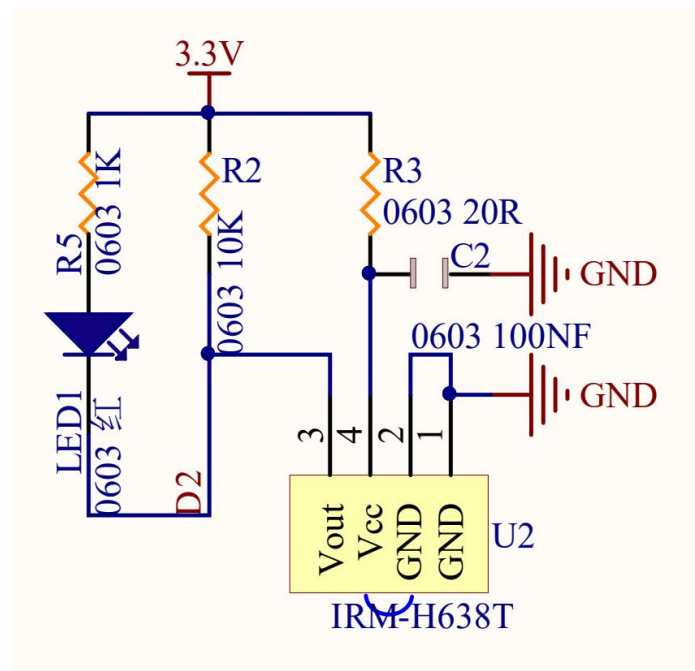
There are two important components inside the IR receiver, namely IC and PD. IC is the processing element of the receiving head. It is mainly composed of silicon crystals and circuits. It is a highly integrated device. Its main functions include filtering, shaping, decoding, and amplification. PD is 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.



8.4 Wiring Diagram



PINS of AdeepT Pixie Board	IR Receiver
D2	VOUT

3.3V	VCC
GND	GND
GND	GND

8.5 Demonstration

1. Connect your computer and Adept Pixie Board (Arduino Board) with a USB cable.
2. Open "**08_IR_Receiver**" folder in **"/Code"** , double-click "**08_IR_Receiver.ino**" .



```

08_IR_Receiver | Arduino IDE 2.3.6
File Edit Sketch Tools Help
Arduino Uno
08_IR_Receiver.ino
1  /**
2   * Filename : 07_IR_Receiver.ino
3   * Author : www.adeept.com
4   */
5  #include <Wire.h>
6  #include <SSD1306AsciiWire.h>
7  #include <IRremote.h> //Call the library corresponding to the infrared remote control.
8  SSD1306AsciiWire display;
9  #define RECV_PIN 2 // Infrared receiving pin
10
11 void setup()
12 {
13     // Initialize OLED
14     Wire.begin();
15     Wire.setClock(400000L);
16     display.begin(&Adafruit128x64, 0x3C);
17     display.setFont(Adafruit5x7);
18     display.clear();
19
20     Serial.begin(115200); // Initialize the serial port and set the baud rate to 115200
21     IrReceiver.begin(RECV_PIN, ENABLE_LED_FEEDBACK); // Start the receiver
22     Serial.print("IRrecvDemo is now running and waiting for IR message on Pin ");
23     Serial.println(RECV_PIN); //print the infrared receiving pin
24 }
25

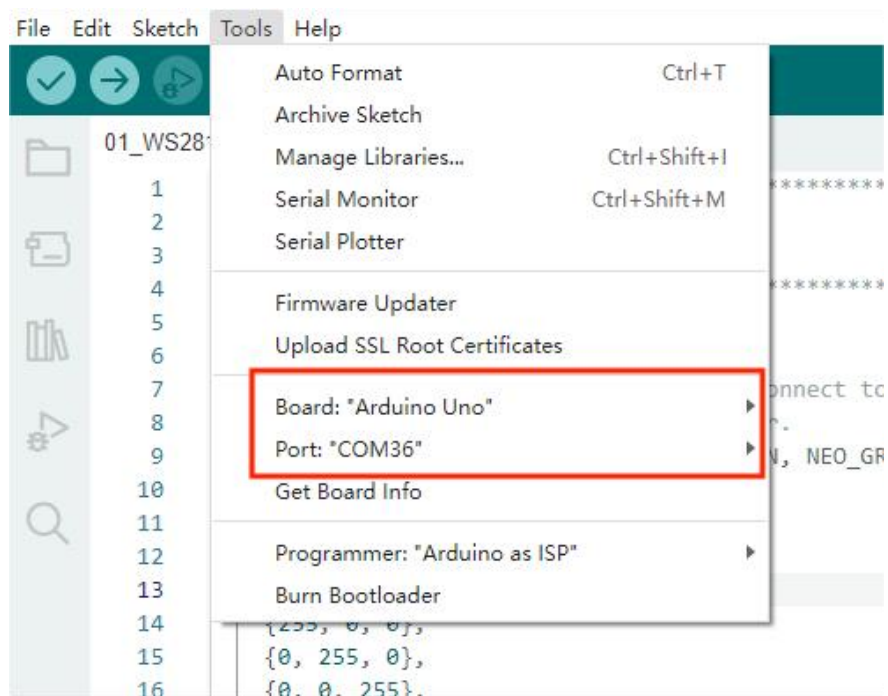
```

3. Select development board and serial port.

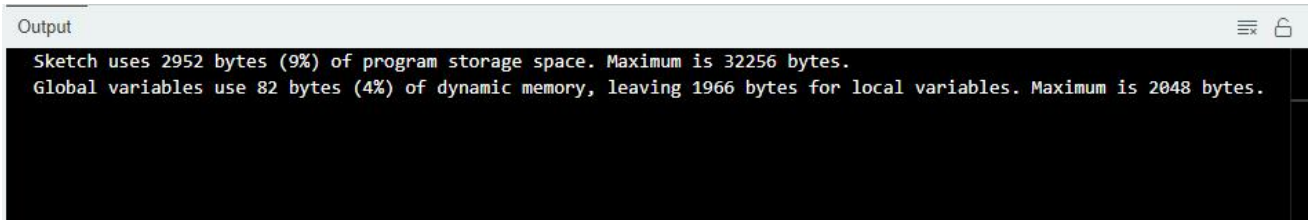
Board: **Tools---**>**Board---**>**Arduino AVR Boards---**>**Arduino Uno**

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.



5. After successfully uploading the code, run the program. 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.





8.6 Code

```

01  /*****
02  Filename   : 08_IR_Receiver.ino
03  Author    : www.adeept.com
04  *****/
05  #include <Wire.h>
06  #include <SSD1306AsciiWire.h>
07  #include <IRremote.hpp> //Call the library corresponding to the infrared remote control.
08  SSD1306AsciiWire display;
09  #define RECV_PIN 2 // Infrared receiving pin
10  String command_ir = "";
11
12  void setup()
13  {
14      // Initialize OLED
15      Wire.begin();
16      Wire.setClock(400000L);
17      display.begin(&Adafruit128x64, 0x3C);
18      display.setFont(Adafruit5x7);
19      display.clear();
20
21      Serial.begin(115200); // Initialize the serial port and set the baud rate to 115200
22      IrReceiver.begin(RECV_PIN, ENABLE_LED_FEEDBACK); // Start the receiver
23      Serial.print("IRrecvDemo is now running and waiting for IR message on Pin ");
24      Serial.println(RECV_PIN); //print the infrared receiving pin
25  }
26
27  void loop()
28  {
29      if (IrReceiver.decode()) { // Waiting for decoding
30          unsigned long value = IrReceiver.decodedIRData.command;
31          command_ir = switch_irr(value);
32          Serial.println(command_ir);
33          display.clear();
34          display.set2X();
35          display.setCursor(0, 0);
36          display.print("Receive:");
37          display.setCursor(0, 3);
38          display.print(command_ir);
39          display.print(" ");
40          IrReceiver.resume(); // Release the IRremote. Receive the next value
41      }
42      delay(100);
43  }
44

```

```
45 String switch_irr(unsigned long irr_data)
46 {
47     switch(irr_data)
48     {
49         case 69: return "A";
50         case 70: return "B";
51         case 71: return "C";
52         case 68: return "D";
53         case 64: return "UP";
54         case 67: return "E";
55         case 7: return "LEFT";
56         case 21: return "OK";
57         case 9: return "RIGHT";
58         case 22: return "0";
59         case 25: return "DOWN";
60         case 13: return "F";
61         case 12: return "1";
62         case 24: return "2";
63         case 94: return "3";
64         case 8: return "4";
65         case 28: return "5";
66         case 90: return "6";
67         case 66: return "7";
68         case 82: return "8";
69         case 74: return "9";
70         default: return "NONE";
71     }
72 }
```

Code explanation

Initialization Stage:

In the setup function, initialize I2C communication and initialize OLED in order to view the infrared decoding results.

Next, activate the infrared receiver to prepare it for receiving infrared signals.

Loop Control Process:

By waiting for the infrared signal to decode. When a valid infrared signal is received and decoded successfully, perform the following operations.

Stage 1: Retrieve key values from decoding results.

Stage 2: Call the switch_irr() function to convert key values and print the corresponding key values through the OLED.

Stage 3: Call `irrecv.resume()` to release the infrared receiver, allowing it to continue receiving the next infrared signal.

Stage 4: By `delaying(100)`; Delay the program by 100ms to avoid excessive data processing and ensure stable system operation.