

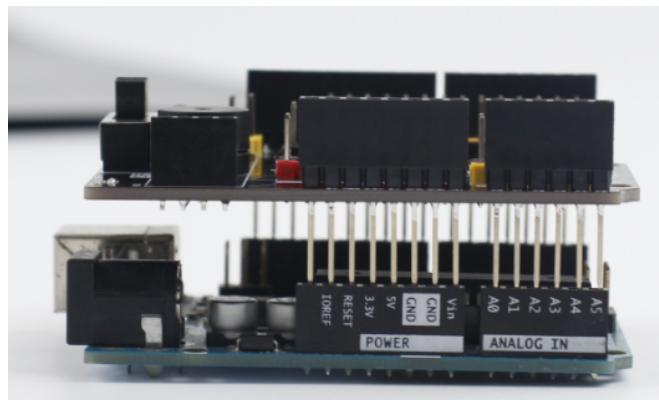
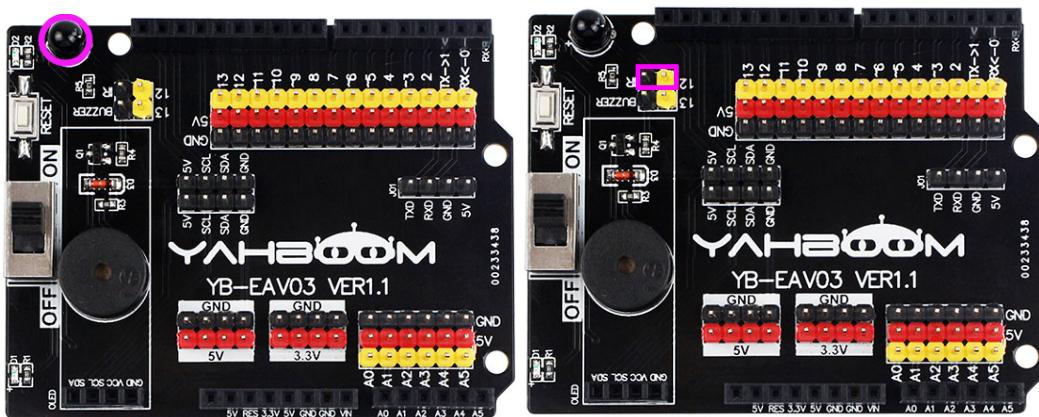
Uno IO expansion board ---- On-board IR tutorial

!Note:

- 1) In order to avoid the interference of sunlight on infrared sensor, we need to carry out this experiment indoors.
- 2) During remote control, the infrared remote control needs to face the infrared receiver on the expansion board.

1. About Hardware

In this experiment, we need to use the on board passive buzzer. **We need to turn the expansion board power switch to “ON” and connect the jumper cap to the 12 and IR pins. As shown below.**



2. About Infrared remote control

In this experiment, we used this Yahboom infrared remote controller (other infrared remote controller will not be suitable for our reference program because of the different encoding, but the infrared receiver is universal).

For different buttons, the key code values are as shown below.



User code: 0xFF

When the remote controller presses the light button, the infrared receiver will receive the data 00FF40BF.

All the received key code values are as follows:

00FF00FF		00FF30CF	
00FF708F		00FF40BF	
00FFA05F		00FF807F	
00FF20DF		00FF00FF	
00FF609F		00FF906F	
00FF10EF		00FF50AF	
00FFB04F		00FF08F7	
00FF8877		00FF48B7	
00FF28D7		00FFA857	
00FF6897		00FF18E7	
00FF9867		00FF58A7	

3. About code

For the code of this course, please refer to On_board_IR folder.

4. Compiling and downloading code

4.1 We need to open the **On_board_IR.ino** file by Arduino IDE software. Then

click “√”under the menu bar to compile the code, and wait for the word “**Done compiling** ” in the lower right corner, as shown below.

```

Buzzer_test | Arduino 1.8.5
File Edit Sketch Tools Help
Buzzer_test

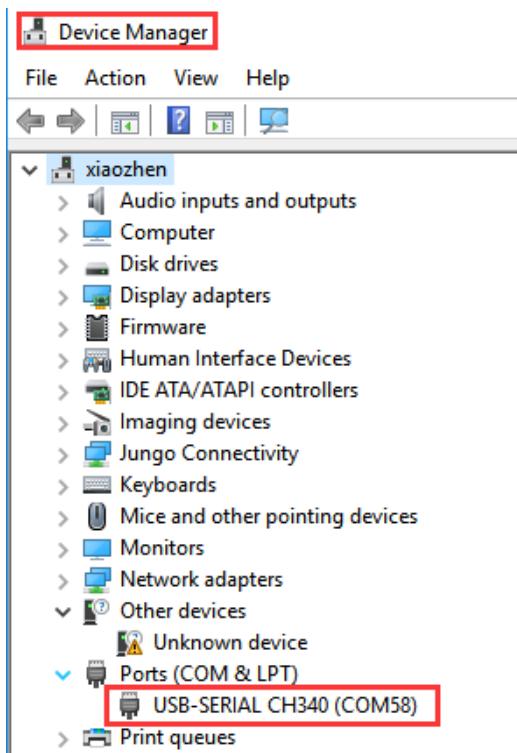
void setup()
{
    pinMode(buzzer, OUTPUT); //Set digital IO pin mode, OUTPUT is output
}

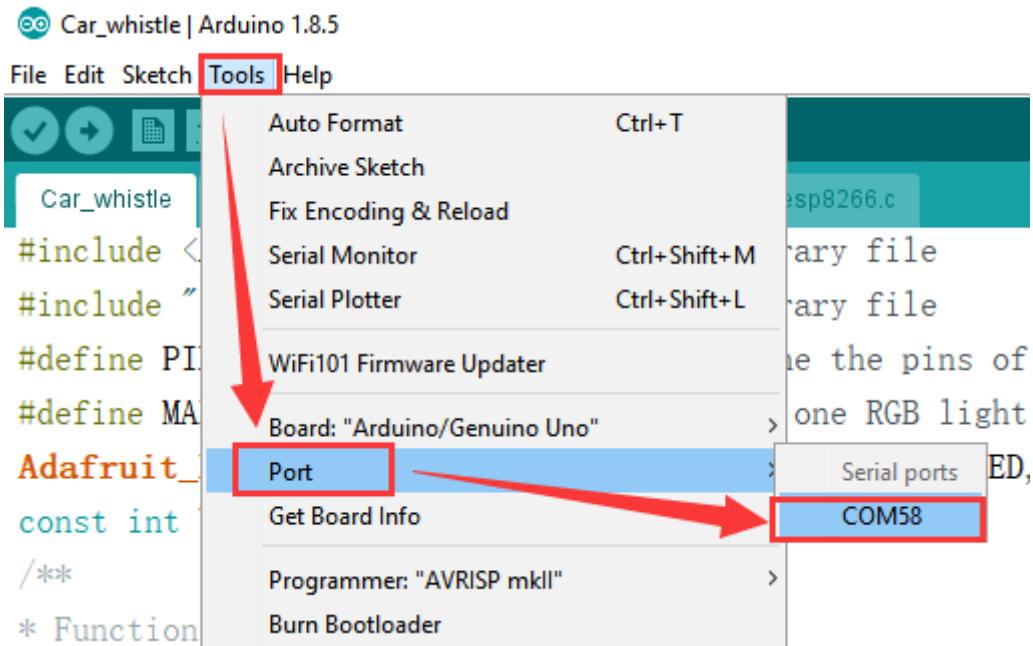
void loop()
{
    unsigned char i, j;
    for(i=0; i<80; i++) //Output a frequency of sound
    {
        digitalWrite(buzzer, HIGH);
        delay(1);
        digitalWrite(buzzer, LOW);
        delay(1);
    }
}

Sketch uses 1014 bytes (3%) of program storage space. Maximum is 32256
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes

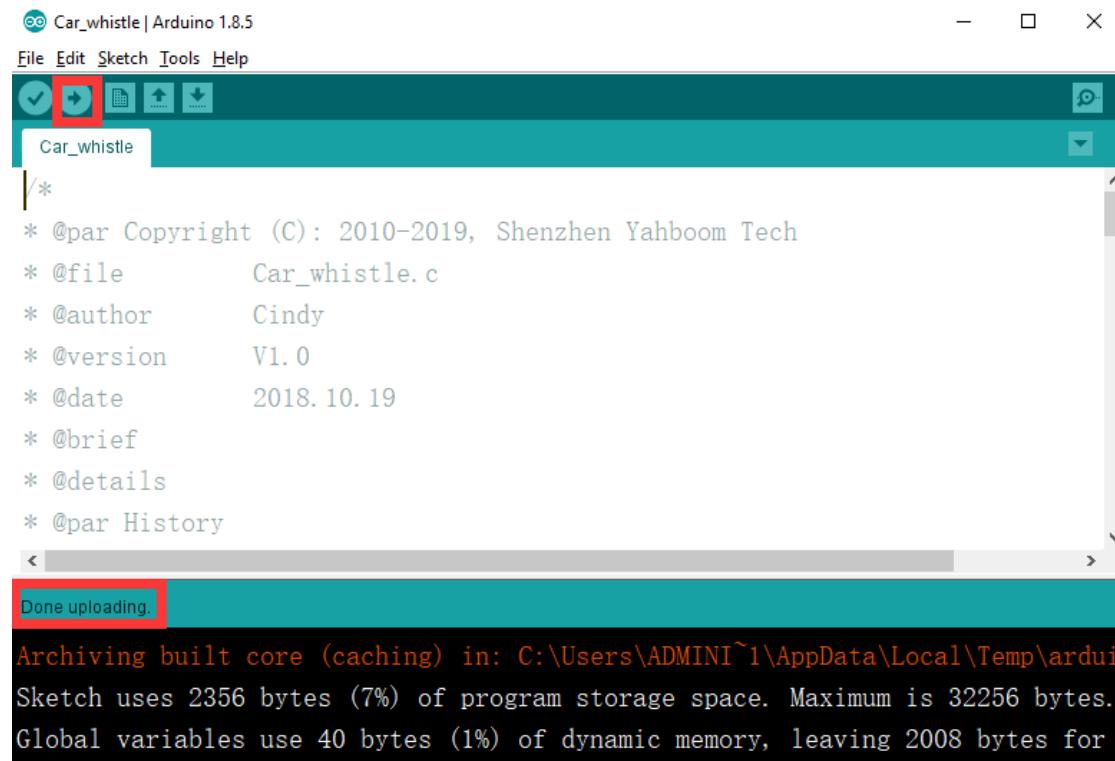
```

4.2 In the menu bar of Arduino IDE, we need to select 【Tools】---【Port】--- selecting the port that the serial number displayed by the device manager just now, as shown below.





4.3 After the selection is completed, you need to click "→" under the menu bar to upload the code to the UNO board. When the word "**Done uploading**" appears in the lower left corner, the code has been successfully uploaded to the UNO board, as shown below.



5.Experimental phenomena

After the program is downloaded.

We need to open the Serial Monitor in the upper right corner of the Arduino IDE interface. And we need to select the same baud rate as set in the program. As shown below.

```
/*
void setup()
{
    Serial.begin(9600); //Baud rate 9600
    irrecv.enableIRIn();
    pinMode(RECV_PIN, INPUT_PULLUP);
}
```

When we press the button of the red remote control, the corresponding string will be printed on the serial monitor. As shown below.

```
RUN
LEFT
RIGHT
BACK
BACK
TLEFT
TRIGHT
LIGHT
PLUS
LESS
0
1
2
3
4
5
```