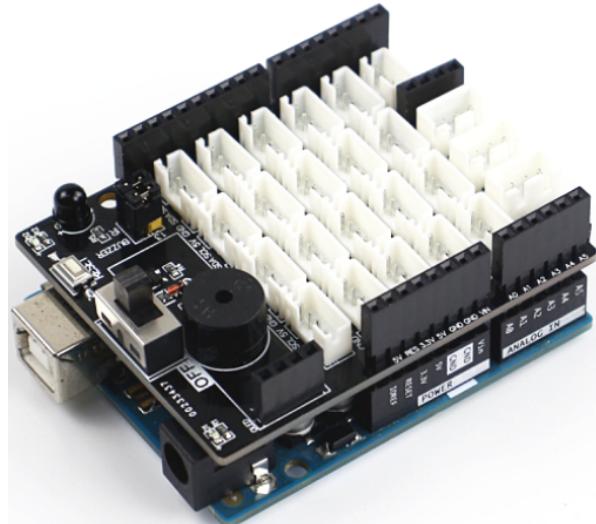
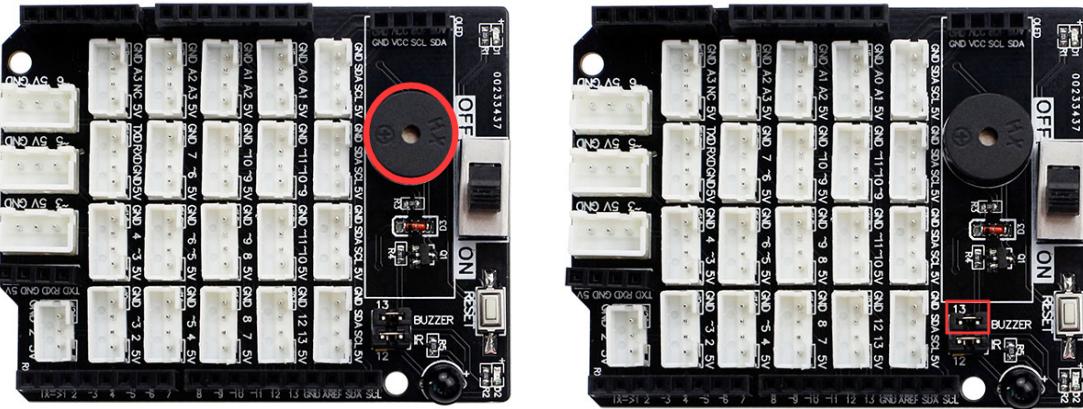


UNO sensor expansion board --- Buzzer

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 13 and BUZZER pins. As shown below.**



2. About Passive buzzer

Buzzers are divided into two types: "active buzzers" and "passive buzzers". Active means that they possess a multi-vibrator inside. It only needs to provide the working voltage externally, it can emit a fixed frequency sound.

Passive means that there is no internal oscillation source, and an external drive circuit is required to provide a certain frequency of the drive signal. In this experiment, we will use active buzzer.

3. About code

For the code of this course, please refer to **sing** folder and **test** folder.

3.1 test buzzer

```

int buzzer = 13; //buzzer connect to P13 pin of Arduino board

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);
    }
    for(i=0; i<100; i++)//output another frequency of sound
    {
        digitalWrite(buzzer,HIGH);
        delay(2);
        digitalWrite(buzzer,LOW);
        delay(2);
    }
}

```

As can be seen from the above code, 80, 100 in the for statement control the frequency, and delay controls the duration, which similar to the beat in music.

A song consists of several notes, each note uniquely corresponding to a frequency. If we know the frequency corresponding to the note, and then let the Arduino output to the buzzer or horn according to this frequency, the buzzer or horn will sound at the corresponding frequency.

3.2 buzzer play music

If we can control the frequency and beat, it is possible to play music. Therefore, we first need to understand the type of tone and the frequency of each tone, as shown in the following table:

L Tone	1 . .	2 . .	3 . .	4 . .	5 . .	6 . .	7 . .
A	221	248	278	294	330	371	416
B	248	278	294	330	371	416	467
C	131	147	165	175	196	221	248
D	147	165	175	196	221	248	278
E	165	175	196	221	248	278	312
F	175	196	221	234	262	294	330
G	196	221	234	262	294	330	371

M Tone	1	2	3	4	5	6	7
A	441	495	556	589	661	742	833
B	495	556	624	661	742	833	935
C	262	294	330	350	393	441	495
D	294	330	350	393	441	495	556
E	330	350	393	441	495	556	624
F	350	393	441	495	556	624	661
G	393	441	495	556	624	661	742

H Pitch	1	2	3	4	5	6	7
A	882	990	1112	1178	1322	1484	1665
B	990	1112	1178	1322	1484	1665	1869
C	525	589	661	700	786	882	990
D	589	661	700	786	882	990	1112
E	661	700	786	882	990	1112	1248
F	700	786	882	935	1049	1178	1322
G	786	882	990	1049	1178	1322	1484

We can determine the length of each tone adjustment according to the score of the song. When combined, we can play a piece of music.

Beat is 1

Beat is 2

ODE

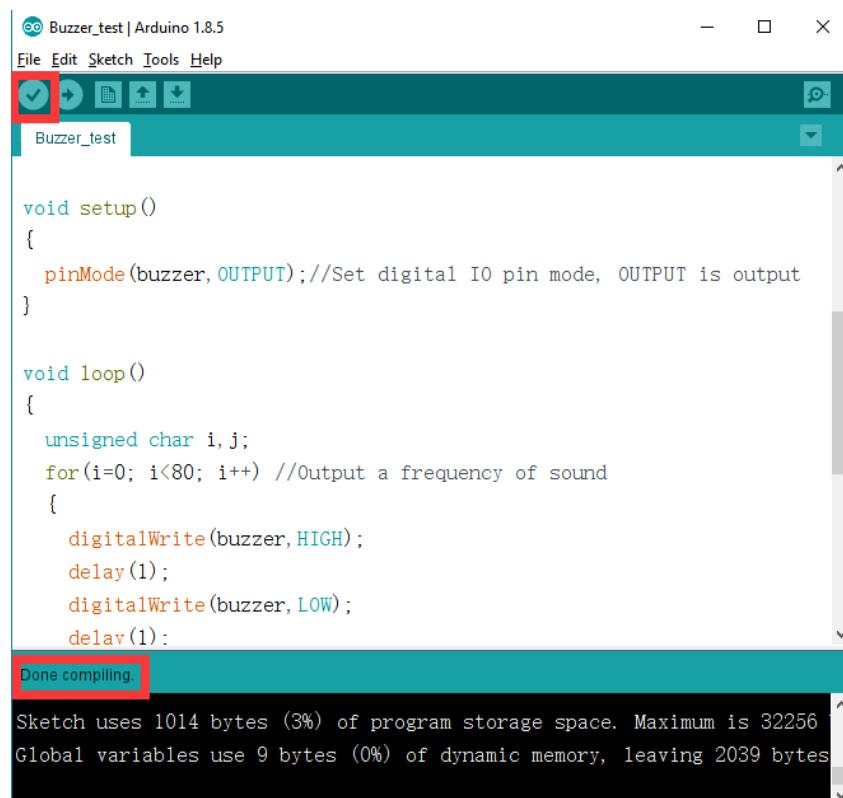
3 3 4 5 | 5 4 3 2 | 1 1 2 3 | 3. 2 2 - | 3 3 4 5 |

5 4 3 2 | 1 1 2 3 | 2. 1 1 - | 2 2 3 1 | 2 34 3 1 | 2 34 3 2 |

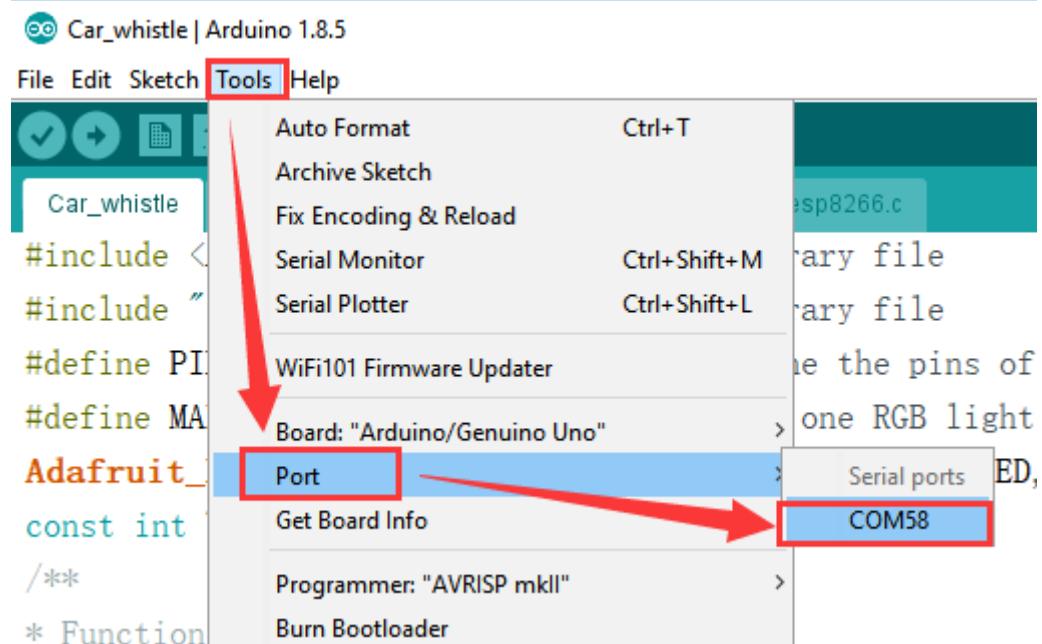
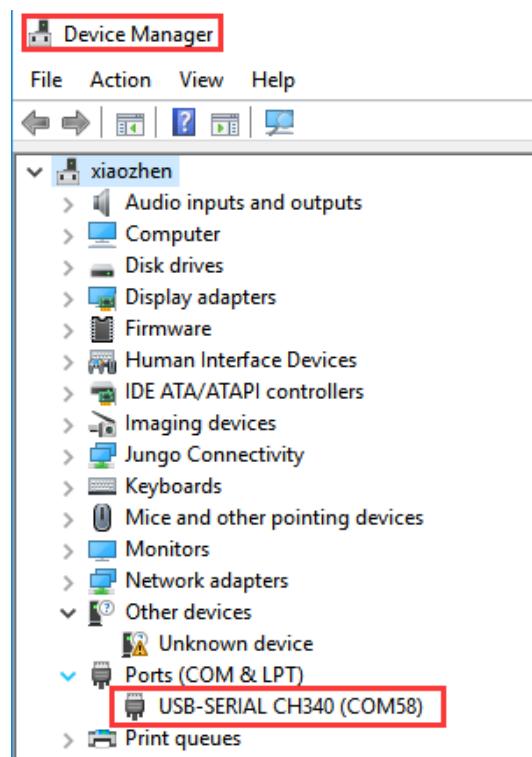
1 2 5 3 3 4 5 | 5 4 3 2 | 1 1 2 3 | 2. 1 1 - |

4. Compiling and downloading code

4.1 We need to open the **sing.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.



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.

```

  File Edit Sketch Tools Help
  Car_whistle
  /*
  * @par Copyright (C): 2010-2019, Shenzhen Yahboom Tech
  * @file      Car_whistle.c
  * @author    Cindy
  * @version   V1.0
  * @date      2018.10.19
  * @brief
  * @details
  * @par History
  */
  Done uploading.
  Archiving built core (caching) in: C:\Users\ADMINI~1\AppData\Local\Temp\arduino-build-15824\sketch\Car_whistle\cores\arduino
  Sketch uses 2356 bytes (7%) of program storage space. Maximum is 32256 bytes.
  Global variables use 40 bytes (1%) of dynamic memory, leaving 2008 bytes for

```

5. Experimental phenomena

After the program is downloaded, the buzzer will send two different frequencies of sound and keep looping in this state.

If you download **sing.ino** into Arduino Uno board, we can heard that the buzzer will play sing "ODE".

6. Program analysis

We can use **tone()** function to realize play music:

<https://www.arduino.cc/reference/en/language/functions/advanced-io/tone/>
tone(pin, frequency)
tone(pin, frequency, duration)

Parameters of the function:

pin: Connect to the speaker is an integer (int type)

frequency: Frequency, is an integer (int type)

duration: The duration of the note, is the millisecond value, unsigned long (unsigned long) no return value