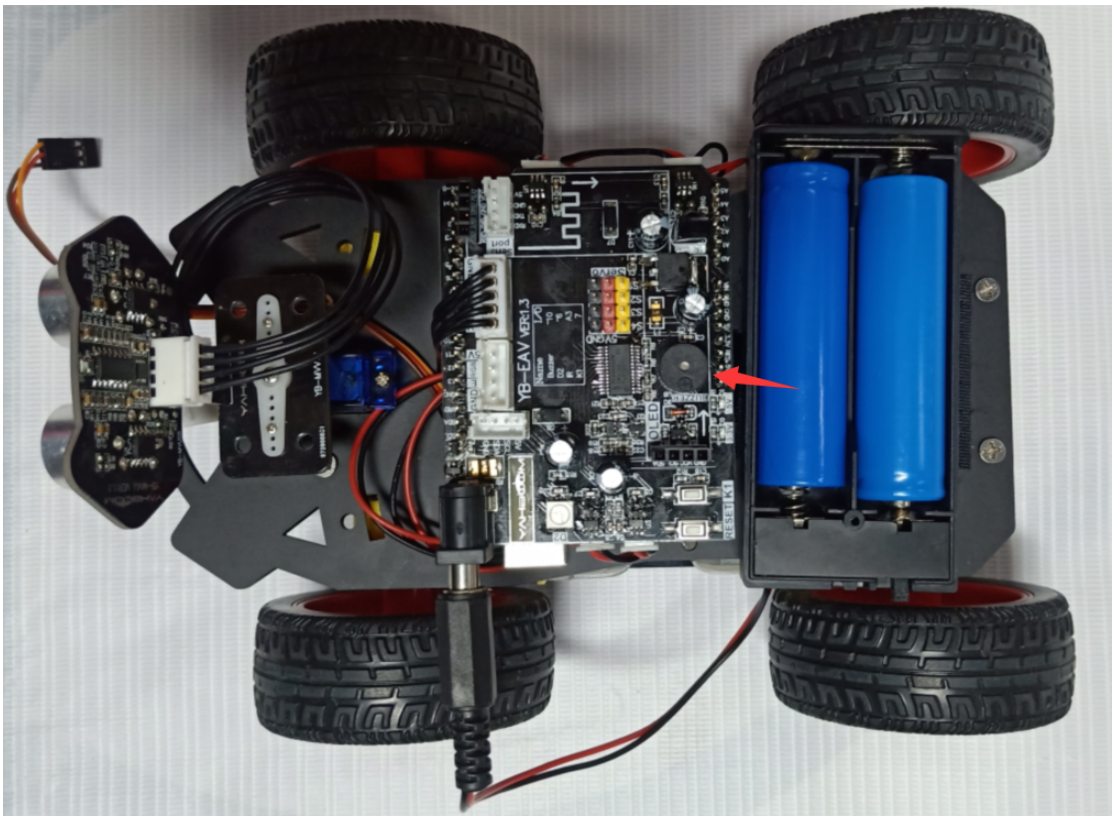
Basic course ---1.Whistle

1. Learning goal

In this lesson, we will learn how to control the buzzer on the expansion board.

2. Preparation

2.1 The position of the buzzer on the robot car. As shown below.



2.2 The pin of UNO board is connected the buzzer.

3. Principle of experimental

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.
From the hardware interface manual, we can know that buzzer are directly driven by P10 of UNO .

Classification	Function	The number of Drive chip PCA9685	Drive Method	Connection with CPU	Uno board
Left Motor	Left front motor forward	LINB(13)	PCA9685	I2C_SDA/I2C_SCL	A4/A5
	Left front motor reverse	LINA(12)			
	Left rear motor forward	RINB(15)			
	Left rear motor reverse	RINA(14)			
Right Motor	Right front motor forward	LED10			
	Right front motor reverse	LED11			
	Right rear motor forward	LED8			
	Right rear motor reverse	LED9			
Servo	Control S1	LED0			
	Control S2	LED1			
	Control S3	LED2			
	Control S4	S1(3)			
LOGO light	Control bluelight	LED7			
Tarcking sensor	Left tracking sensor	Uno board drive directly			A0
	Middle tracking sensor				A1
	Right tracking sensor				A2
Ultrasonic sensor	Ultrasonic Echo				12
	Ultrasonic RGB light				11
Key	K1				7
IR	IR control				A3
Bluetooth interface	RX				0
	TX				1
On board RGB Light	RGB Light on expansion board				6
Buzzer	Control buzzer				10

4. About code

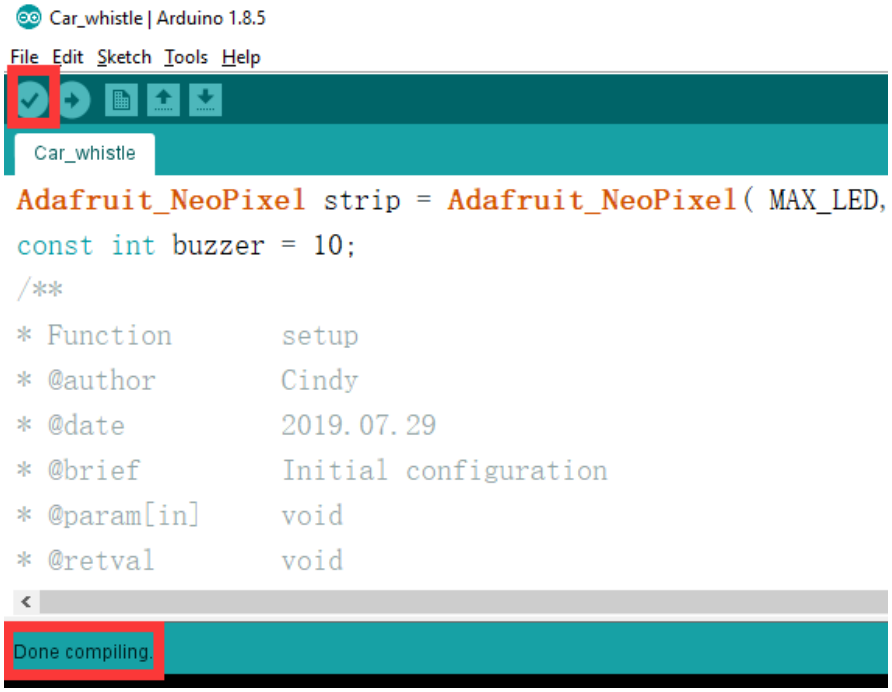
For the code of this course, please refer to: **Car_whistle.ino** in the **Car_whistle** folder.

```
#include <Arduino.h>                    //Library file
#include <Adafruit_NeoPixel.h>          //Library file
#define PIN 6                           //Define the pins of the RGB light
#define MAX_LED 1                       //Just one RGB light on the car
Adafruit_NeoPixel strip = Adafruit_NeoPixel( MAX_LED, PIN, NEO_RGB +
NEO_KHZ800 );
const int buzzer = 10;

void setup()
{
  // put your setup code here, it will run once:
  pinMode(buzzer,OUTPUT);
  strip.begin();
  strip.show();
}
```

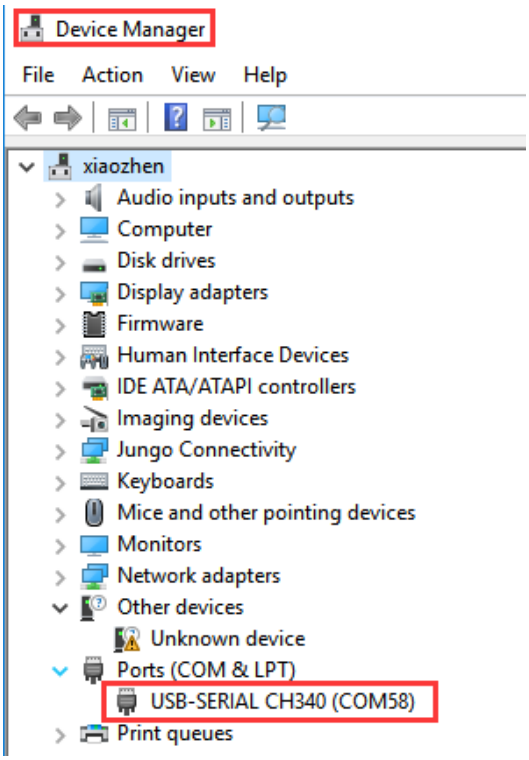
```
void whistle()
{
   for (int i = 0; i < 100; i++)
   {
      digitalWrite(buzzer, HIGH); //Sound
      delay(3);
      digitalWrite(buzzer, LOW);    //Nosound
      delay(1);
   }
}

void loop()
{
    whistle();
    delay(1000);
}
```

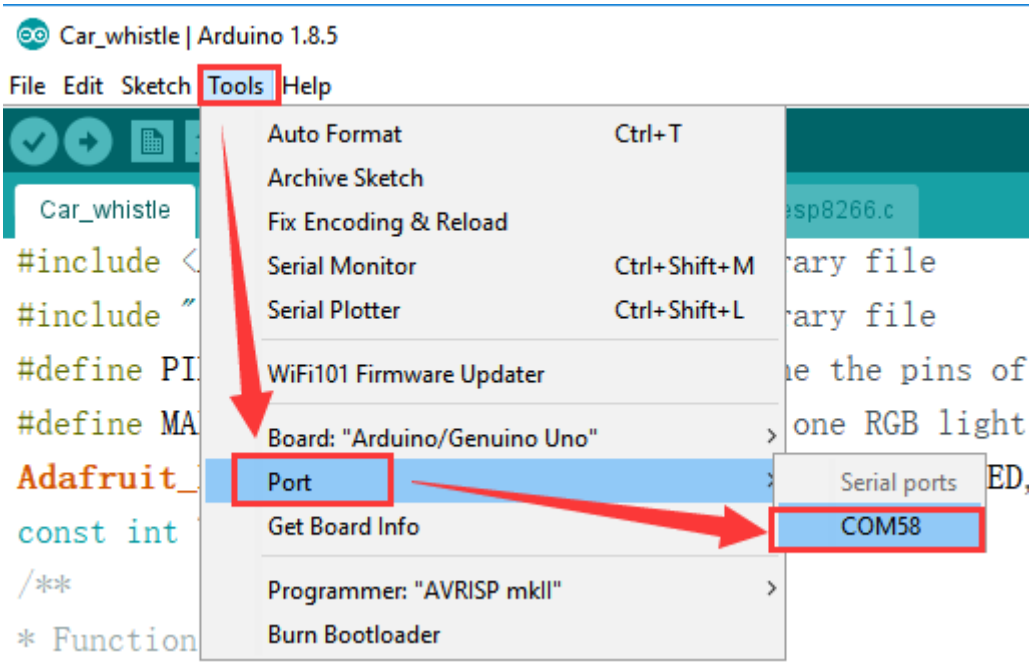
5. Compiling and downloading code

5.1 We need to open the **Car_whistle.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 in the figure below.



5.2 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 in the figure below.





5.3 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 in the figure below.

6. Experimental phenomena

After the program is downloaded, the buzzer will start whistle and keep looping in this state.