

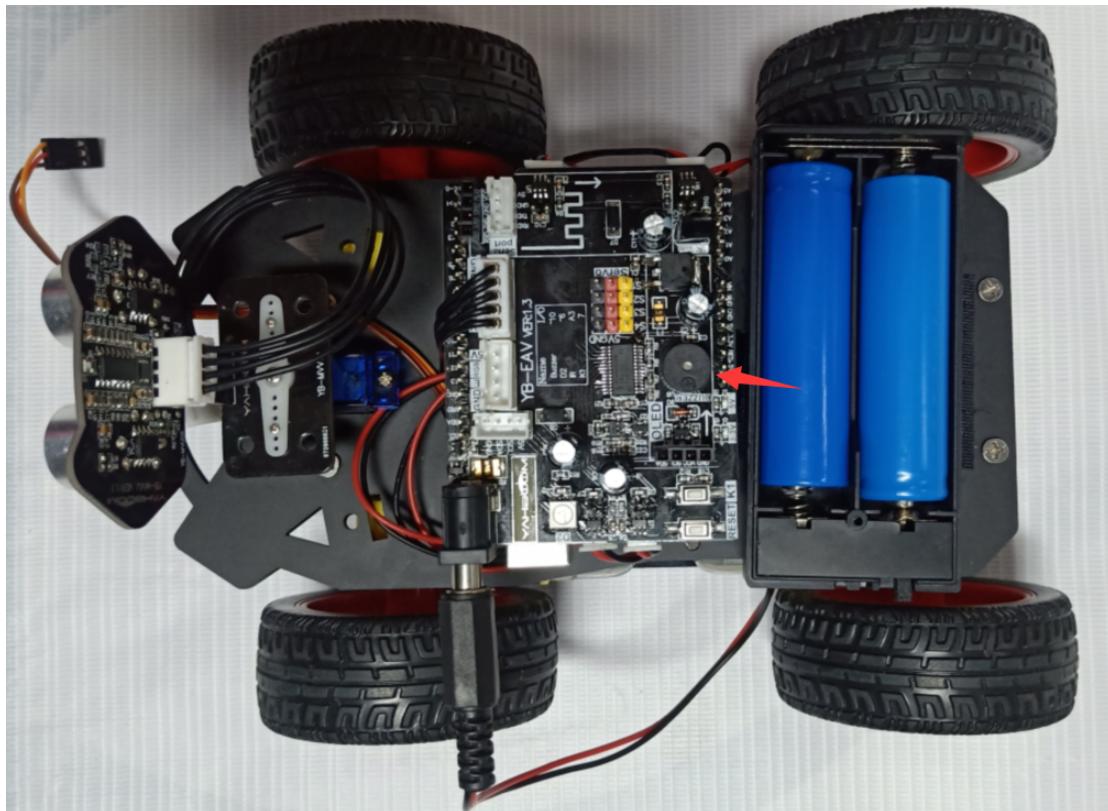
Expand course ---1.Singing

1. Learning goal

In this lesson, we will learn how to make the buzzer play a song.

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 pin	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	Arduino Uno drive directly		A0
	Control S2	LED1			
	Control S3	LED2			
	Control S4	S1 (3)			
LOGO light	Control bluetooth	LED7			
Tracking sensor	Left tracking sensor				A1
	Middle tracking sensor				A2
	Right tracking sensor				12
Ultrasonic sensor	Ultrasonic Echo				11
	Ultrasonic RGB light				7
Key IR	K1				A3
	IR control				0
Bluetooth interface	RX				1
	TX				6
On board RGB Light	RGB Light on expansion board				10
Buzzer	Control buzzer				

4. About code

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

```
#include <Arduino.h>      //Library file
const int buzzer = 10;    //Define the pins of buzzer

#define BL1 248
#define BL2 278
#define BL3 294
#define BL4 330
#define BL5 371
#define BL6 416
#define BL7 467

#define B1 495
#define B2 556
#define B3 624
#define B4 661
```

```
#define B5 742
#define B6 833
#define B7 935

#define BH1 990
#define BH2 1112
#define BH3 1178
#define BH4 1322
#define BH5 1484
#define BH6 1665
#define BH7 1869

#define NTC1 262
#define NTC2 294
#define NTC3 330
#define NTC4 350
#define NTC5 393
#define NTC6 441
#define NTC7 495

#define NTCL1 131
#define NTCL2 147
#define NTCL3 165
#define NTCL4 175
#define NTCL5 196
#define NTCL6 221
#define NTCL7 248

#define NTCH1 525
#define NTCH2 589
#define NTCH3 661
#define NTCH4 700
#define NTCH5 786
#define NTCH6 882
#define NTCH7 990

#define NTD0 -1
#define NTD1 294
#define NTD2 330
#define NTD3 350
#define NTD4 393
#define NTD5 441
#define NTD6 495
#define NTD7 556
```

```
#define NTDL1 147
#define NTDL2 165
#define NTDL3 175
#define NTDL4 196
#define NTDL5 221
#define NTDL6 248
#define NTDL7 278

#define NTDH1 589
#define NTDH2 661
#define NTDH3 700
#define NTDH4 786
#define NTDH5 882
#define NTDH6 990
#define NTDH7 1112

#define NTE1 330
#define NTE2 350
#define NTE3 393
#define NTE4 441
#define NTE5 495
#define NTE6 556
#define NTE7 624

#define NTEL1 165
#define NTEL2 175
#define NTEL3 196
#define NTEL4 221
#define NTEL5 248
#define NTEL6 278
#define NTEL7 312

#define NTEH1 661
#define NTEH2 700
#define NTEH3 786
#define NTEH4 882
#define NTEH5 990
#define NTEH6 1112
#define NTEH7 1248

//ODE
int tune[]=
{
```

```
NTC3, NTC3, NTC4, NTC5, NTC5, NTC4, NTC3, NTC2, NTC1, NTC1, NTC2,
NTC3, NTC3, NTC2, NTC2,
    NTC3, NTC3, NTC4, NTC5, NTC5, NTC4, NTC3, NTC2, NTC1, NTC1, NTC2,
NTC3, NTC2, NTC1, NTC1,
NTC2, NTC2, NTC3, NTC1, NTC2, NTC3, NTC4, NTC3, NTC1, NTC2, NTC3,
NTC4, NTC3, NTC2,
};

float durt[]=
{
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 0.5, 0.5, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 2, 0.5, 0.5, 1, 1, 1, 1, 1, 0.5, 0.5, 1, 1, 1, 0.5, 0.5, 1, 1,
0.5, 0.5, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0.5, 0.5, 1, 1, 1, 1, 2, 0.5, 0.5,
};

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

void Playmusic()
{
    int length = sizeof(tune)/sizeof(tune[0]);      //Calculated length
    for(int x=0; x < length;x++)
    {
        tone(buzzer,tune[x]);
        delay(500*durt[x]);
        //This is used to adjust the delay according to the beat.
        //The parameter 500 can be adjusted by yourself.
        //500 is suitable for this song.
        noTone(buzzer);
    }
}
void loop()
{
    Playmusic();
    delay(1000);
}
```

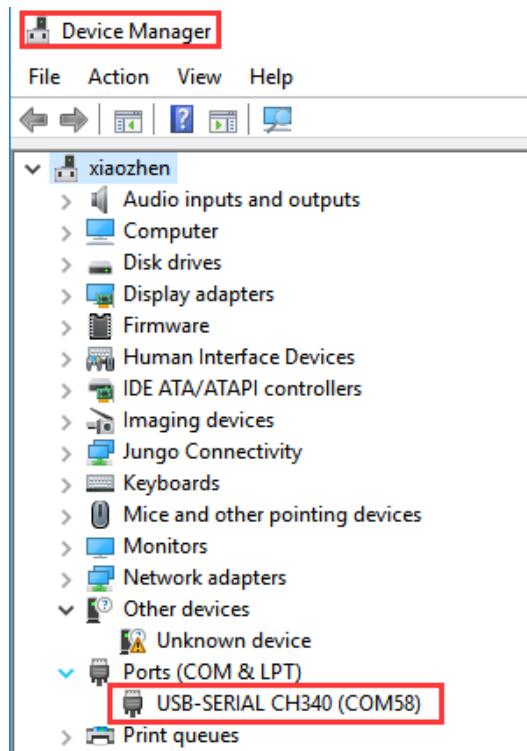
5. Compiling and downloading code

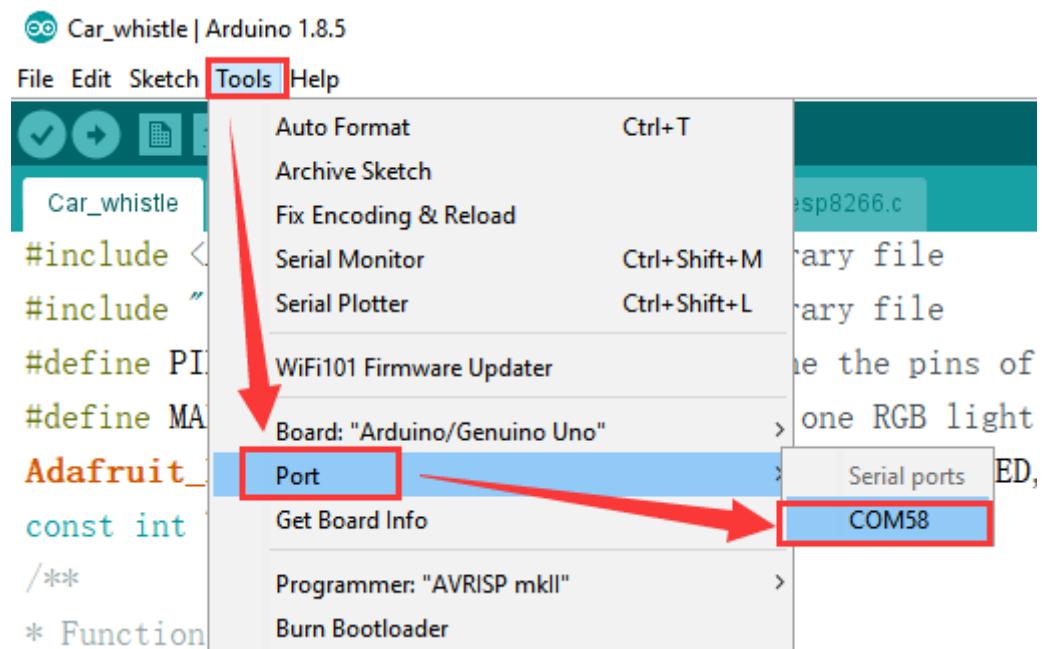
5.1 We need to open the **Car_sing.ino** file by Arduino IDE software. Then click “√” under the menu bar to compile the code, and wait for the word “**Done compiling**” in the lower left corner, as shown in the figure below.

```
#include <Arduino.h>      //Library file
const int buzzer = 10;    //Define the pins of buzzer

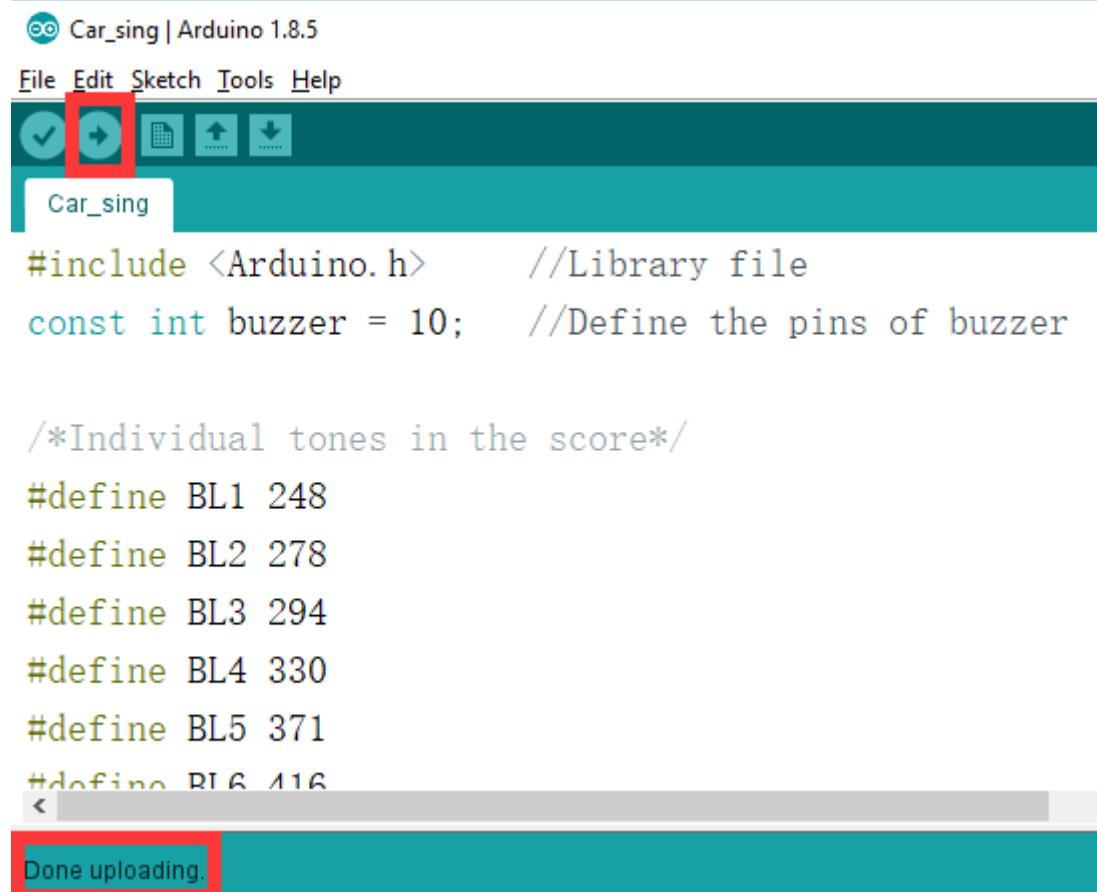
/*Individual tones in the score*/
#define BL1 248
#define BL2 278
#define BL3 294
#define BL4 330
#define BL5 371
#define RT6 416
```

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 play song 《Ode》 and keep looping in this state.