5.9 Voice Controlled Car

Note: If the recognition effect is not good, we can modify the detection reference value appropriately in the code.
In addition to making sounds, we can also trigger sound sensors by blowing air.

Note: Motor speed is affected by battery power.
For this course, when the battery power is high (the power value is above 26000), if the battery power is not enough, we need to charge battery in time or modify the motor speed in the code.

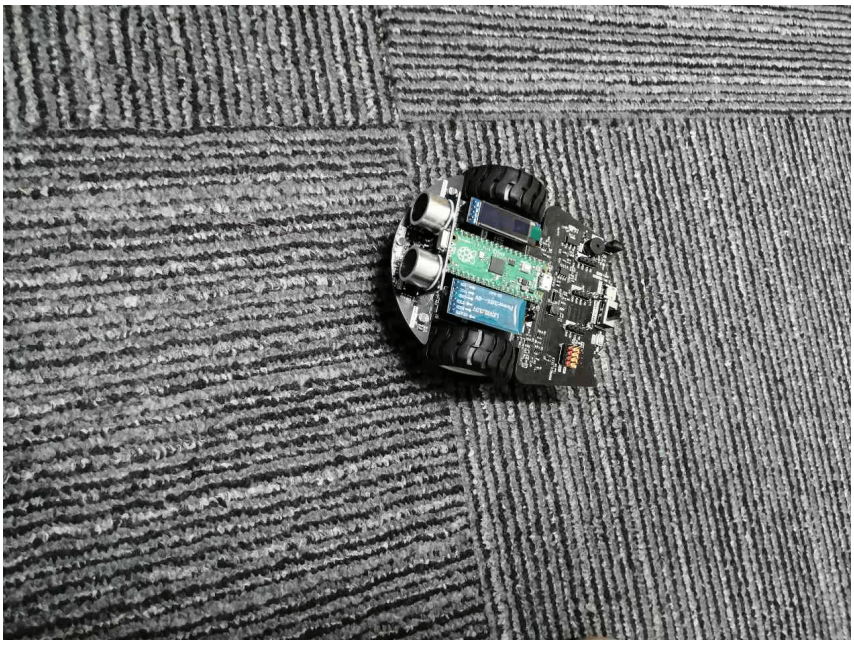
1. Learning Objectives

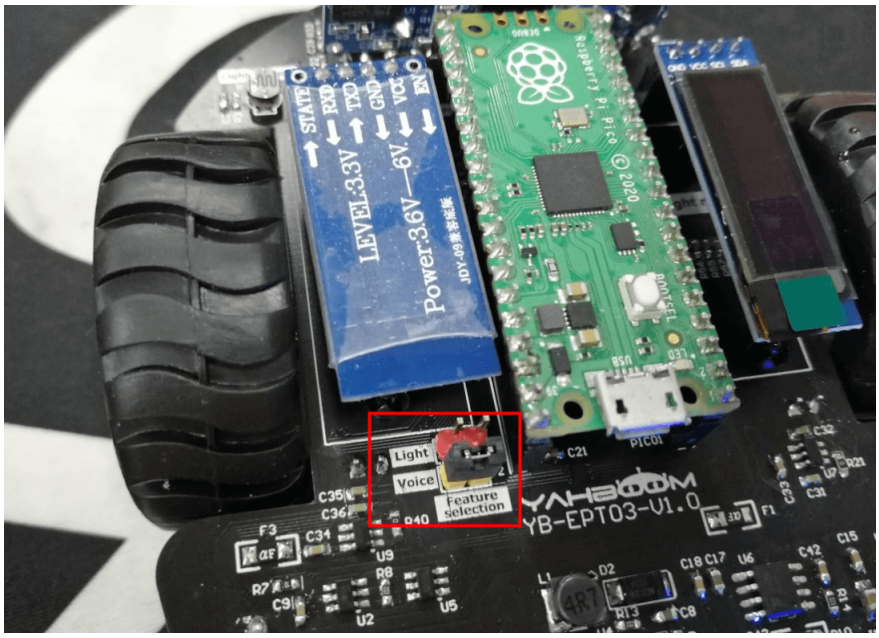
In this course, we will learn that how to realize control robot car by sound.

2. About Hardware

We need use motor, OLED, sound sensor, RGB light, and buzzer on Pico robot.

Please connect the jumper cap to the Voice pin before using. As shown below.





In the code, we continuously read the data detected by the sound sensor, and when this data exceeds the range we set, we control the car to run for a period of time.

3. About Code

Code path: Code -> 3.Robotics course -> 9.Voice control car.py

```
from pico_car import SSD1306_I2C, pico_car, ws2812b
from machine import Pin, I2C, ADC, PWM
import time

Motor = pico_car()
Motor.Car_Stop()
num_leds = 8  # Number of NeoPixels
# Pin where NeoPixels are connected
pixels = ws2812b(num_leds, 0)
pixels.fill(0,0,0)
pixels.show()
# set buzzer pin
BZ = PWM(Pin(22))
BZ.freq(1000)
CM = [0, 330, 350, 393, 441, 495, 556, 624]
#initialization oled
i2c=I2C(1, scl=Pin(15),sda=Pin(14), freq=100000)
oled = SSD1306_I2C(128, 32, i2c)
#initialization ADC
Sound = machine.ADC(27)

while True:
    #get value
    sounds = Sound.read_u16()
    print(sounds)
    oled.text('Sound:', 0, 0)
    oled.text(str(sounds), 50, 0)
    #Control action
    if sounds > 20000:
        while sounds > 10000:
            Motor.Car_Stop()
            sounds = Sound.read_u16()
            print(sounds)
            time.sleep(0.001)
        Motor.Car_Run(255,255)
        BZ.duty_u16(500)
        BZ.freq(CM[1])
        pixels.set_pixel(2,150,0,150)
        pixels.set_pixel(3,150,0,150)
        pixels.show()
        time.sleep(0.03)
        BZ.duty_u16(500)
        BZ.freq(CM[2])
        pixels.set_pixel(2,0,0,0)
        pixels.set_pixel(3,0,0,0)
        pixels.set_pixel(1,150,0,150)
        pixels.set_pixel(4,150,0,150)
        pixels.show()
        time.sleep(0.03)
        BZ.duty_u16(500)
        BZ.freq(CM[3])
```

```
            pixels.set_pixel(1,0,0,0)
            pixels.set_pixel(4,0,0,0)
            pixels.set_pixel(0,150,0,150)
            pixels.set_pixel(5,150,0,150)
            pixels.show()
            time.sleep(0.03)
            BZ.duty_u16(500)
            BZ.freq(CM[4])
            pixels.set_pixel(0,0,0,0)
            pixels.set_pixel(5,0,0,0)
            pixels.set_pixel(6,150,0,150)
            pixels.set_pixel(7,150,0,150)
            pixels.show()
            time.sleep(0.03)
            BZ.duty_u16(500)
            BZ.freq(CM[5])
            pixels.set_pixel(0,0,0,0)
            pixels.set_pixel(5,0,0,0)
            pixels.set_pixel(6,150,0,150)
            pixels.set_pixel(7,150,0,150)
            pixels.show()
            time.sleep(0.03)
            BZ.duty_u16(500)
            BZ.freq(CM[6])
            pixels.set_pixel(6,0,0,0)
            pixels.set_pixel(7,0,0,0)
            pixels.show()
            BZ.duty_u16(0)
            sounds = 0
            oled.show()
            oled.fill(0)
        else:
            Motor.Car_Stop()
            oled.show()
            oled.fill(0)
        time.sleep(0.01)
```

The performance of the sound sensor will be affected by the battery level.
We recommend adjusting comparison values in code after user testing.
When testing the sensor, it is recommended to turn off high-power loads such as the motor and RGB lights of the car.

from pico_car import SSD1306_I2C, pico_car, ws2812b

Using pico_car's SSD1306_I2C, pico_car, ws2812b, encapsulates the motor driver, RGB lights, and OLED libraries.

import time

The "time" library. This library handles everything time related, from measuring it to inserting delays into programs. The unit is seconds.

from machine import Pin, I2C, ADC, PWM

The machine library contains all the instructions that MicroPython needs to communicate with Pico and other MicroPython-compatible devices, extending the language of physical computing, using the Pin, PWM, ADC and I2C libraries here.

Motor = pico_car()

Initialize the motor drive.

pixels = ws2812b(num_leds, 0)

Initialize RGB lights, we have 8 RGB lights, here num_leds is set to 8.

pixels.fill(0,0,0)

Set all lights to 0,0,0, that is, turn off all lights, the parameters are (red, green, blue), and the color brightness is 0-255.

pixels.show()

Display the set lights.

pixels.set_pixel(i,255,0,0)

Use a for loop to set all car lights to red.

i2c=I2C(1, scl=Pin(15),sda=Pin(14), freq=100000)

Set the IIC 1 pin to SCL 15, SDA 14, and the frequency to 100000.

oled = SSD1306_I2C (128, 32, i2c)

Initialize the size of the OLED to 128*32, and pass in the IIC parameters set earlier.

oled.show ()

Display the set OLED content.

oled.fill (0)

Clear the settings and prepare for the next display.

Motor.Car_Run(150,150)

Control the car to move forward, the speed is set to 150, the parameters are (left motor speed, right motor speed), and the speed range is 0-255.

Motor.Car_Stop()

Control the car to stop.

BZ = PWM(Pin(22))

Set IO22 as a PWM output pin to control the buzzer.

BZ.freq(1000)

Set the PWM frequency to 1000.

BZ.duty_u16(0)

When the value is 0, the sound is turned off, and when the value is 500, the sound is turned on.

oled.text (str (sounds), 50, 0)

Convert the distance to a string and display it at the 50,0 position of the OLED.

Sound = machine.ADC(27)

Initialize ADC port 27.

sounds = Sound.read_u16()

The Sound.read_u16() function is used to detect the value of the sound sensor and assign it to the variable sounds.

4. Experimental Phenomenon

After the code is downloaded, we can see that the first line of the OLED displays the value of sound sensor.

When we blow on the sound sensor, the car honks and drives forward for some distance and the RGB lights create a running water light effect.

At the same time, the print shell will also print out the value of the sound sensor.