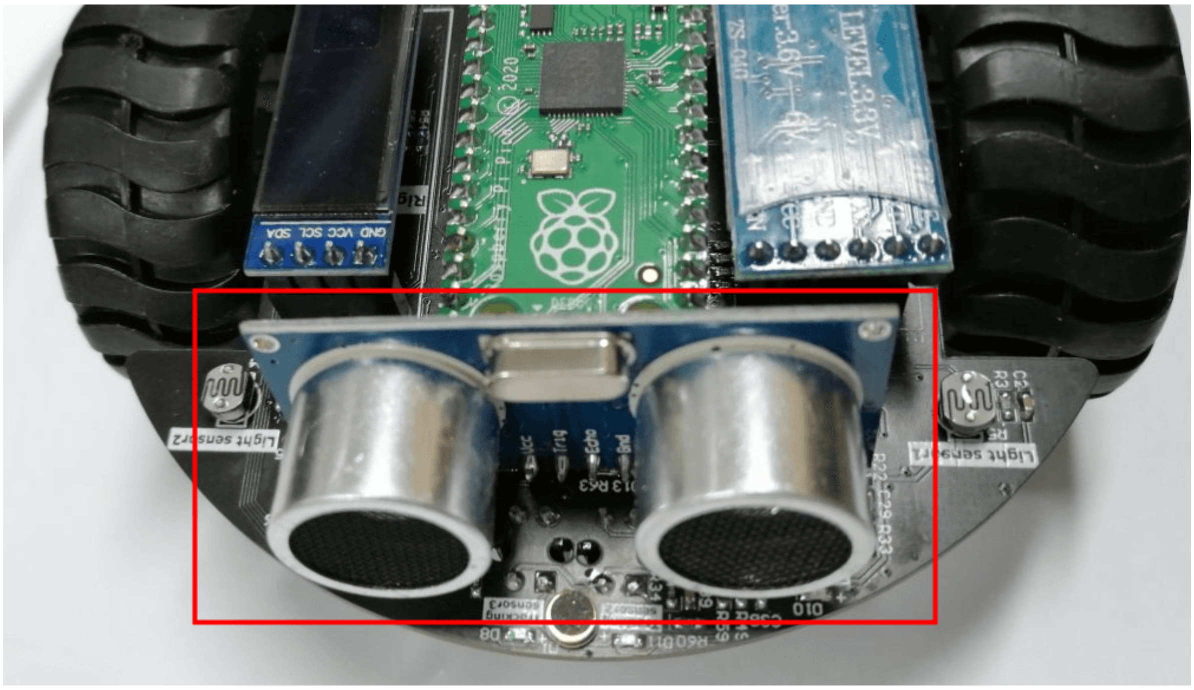
4.4 Ultrasonic sensor

I. Learning objectives

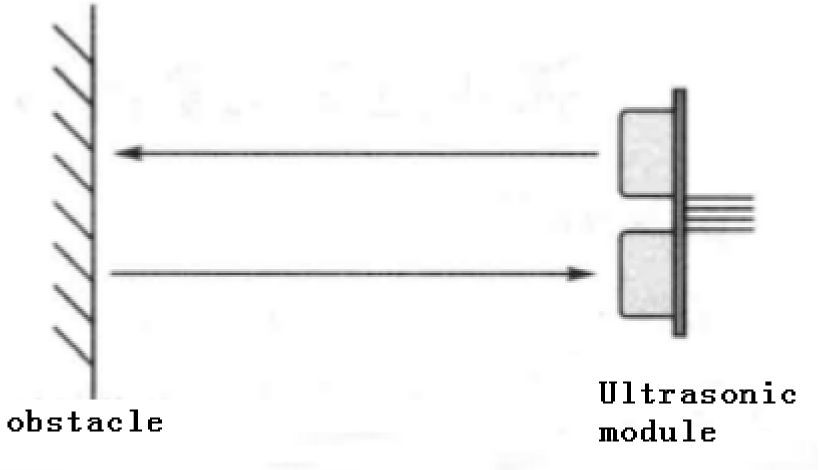
1. Learn to combine the ultrasonic sensor and OLED of the Raspberry Pi Pico 2/Pico mainboard and the car expansion board to conduct experiments.
2. Understand the use of ultrasonic sensors.

II. Hardware usage

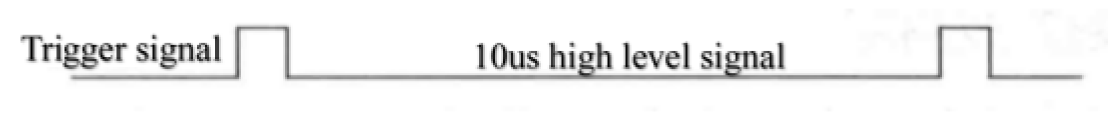
This course uses the ultrasonic sensor and OLED of the Pico 2/Pico mainboard and the car expansion board.



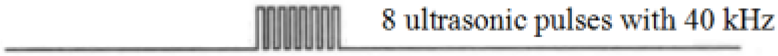
The ultrasonic module is a sensor that uses ultrasonic characteristics to detect the distance. It has two ultrasonic probes for transmitting and receiving ultrasonic waves. The range of measurement is 3-450 cm.



(1) You need to input a high level signal of at least 10us to the Trig pin to trigger the ranging function of the ultrasonic module.



(2) After the ranging function is triggered, the module will automatically send out 8 ultrasonic pulses with 40 kHz and automatically detect whether there is a signal return. This step is done internally by the module.



(3) When the module detects an echo signal, the ECHO pin will output a high level. The high level duration is the time from when the ultrasonic wave is sent to when it returns. You can calculate the distance by using the time function to calculate the high level duration.

3. Program Analysis

Code path: Code -> 2.Advanced course -> 4. Ultrasonic sensor.py

```
import time
from machine import Pin, I2C
from pico_car import SSD1306_I2C, ultrasonic
#initialization ultrasonic
ultrasonic = ultrasonic()
#initialization oled
i2c=I2C(1, scl=Pin(15),sda=Pin(14), freq=100000)
oled = SSD1306_I2C(128, 32, i2c)

while True:
    #get distance
    distance = ultrasonic.Distance_accurate()
    print("distance is %d cm"%(distance) )
    #display distance
    oled.text('distance:', 0, 0)
    oled.text(str(distance), 75, 0)
    oled.show()
    oled.fill(0)
    time.sleep(1)
```

from pico_car import SSD1306_I2C, ultrasonic

Use SSD1306_I2C and ultrasonic from pico_car, which are our packaged OLED and ultrasonic libraries.

import time

The "time" library. This library handles everything to do with time, from measuring it to inserting delays into your program. The units are in seconds.

from machine import Pin, I2C

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

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

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

oled = SSD1306_I2C(128, 32, i2c)

Initialize the size of OLED to 128*32, and pass the previously set IIC parameters into it.

ultrasonic = ultrasonic()

Initialize ultrasonic distance measurement.

distance = ultrasonic.Distance_accurate()

Assign the value returned by ultrasonic distance measurement to the variable distance.

oled.show()

Display the set OLED content.

oled.fill(0)

Clear the set content and prepare for the next display.

oled.text(str(distance), 75, 0)

Convert the distance to a string and display it at the 75,0 position of the OLED.

IV. Experimental Phenomenon

After the program is downloaded, we can see that the OLED displays 'distance: ' and the measured distance. The value will change according to the measurement result, and the Shell will also display the measured distance.

Note that the shortest ultrasonic measurement distance is 2-3cm.