

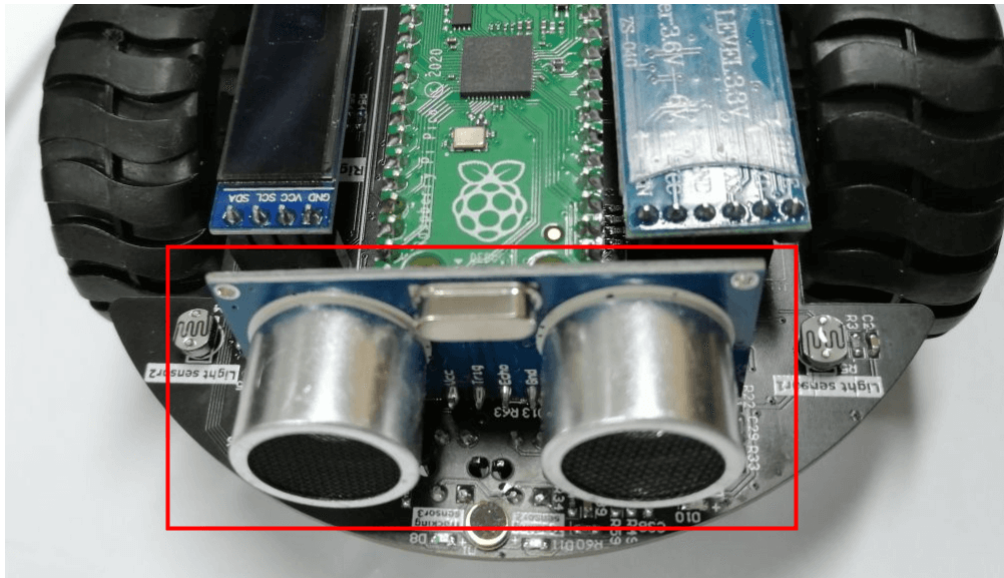
## 4.4 Ultrasonic sensor

### 1. Learning Objectives

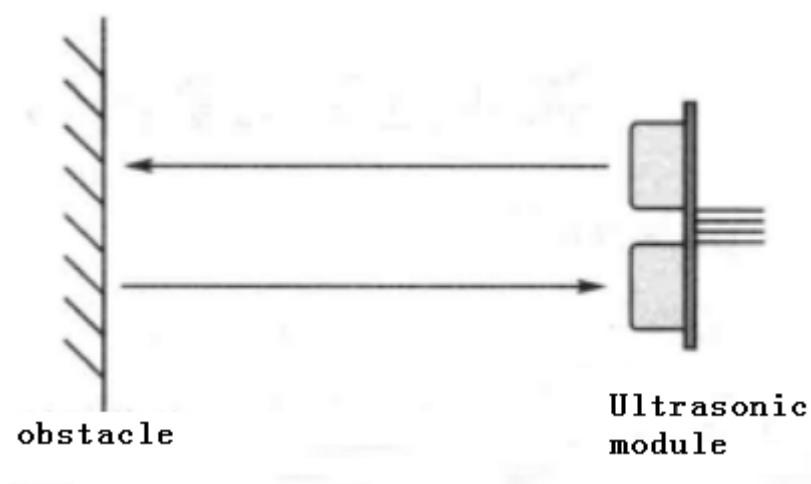
In this course, we will learn to experiment with the combination of ultrasonic sensors and OLEDs on the Raspberry Pi Pico robot.

### 2. About Hardware

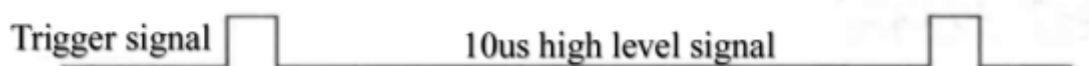
This course uses the ultrasonic sensor and OLED on Pico robot.



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 10 $\mu$ s 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.

**Formula: Distance = High level duration \* Speed of sound(340M/S)/2.**

### 3. About code

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 is our packaged OLED and ultrasonic library.

**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**

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 and I2C libraries here.

**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.

**ultrasonic = ultrasonic()**

Initialize ultrasonic ranging.

**distance = ultrasonic.Distance\_accurate()**

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

**oled.show()**

Display the set OLED content.

**oled.fill(0)**

Clear the settings and prepare for the next display.

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

Convert the distance to a string to display on the OLED at position 75,0.

#### **4. Experimental phenomenon**

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

At the same time, the print windows will also print the measured distance.



**Note: The shortest ultrasonic measurement distance is 2-3 cm.**