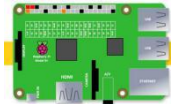# Lesson 8 How to Control the Ultrasonic Module

In this lesson, we will learn how to read the data of the RGB ultrasonic ranging module.

## 8.1 Components used in this course

| Components | Quantity | Picture |
|---|---|---|
| Raspberry Pi | 1 |  |
| Robot HAT | 1 |  |
| ultrasonic module | 1 |  |
| 4 pin wire | 1 |  |

## 8.2 Introduction of RGB ultrasonic ranging module

The ultrasonic ranging module used in our RaspTankPro product has four pins, namely VCC, GND, Echo and Trig. The HC-SR04 can provide a non-contact distance sensing function of 2cm-400cm, and the ranging accuracy can reach 3mm; The module includes an ultrasonic transmitter, receiver and control circuit. The basic working principle is as follows:

Use IO port TRIG to trigger distance measurement, and give a high level signal of at least 10us.

The module automatically sends eight 40khz square waves, and automatically detects whether there is a signal return.

There is a signal return, and a high level is output with the IO port ECHO. The duration of the high level is the time from emission to return of the ultrasonic wave.

The principle of distance detection by ultrasonic ranging sensor: the method of detecting distance by ultrasonic is called echo detection method, that is, the ultrasonic transmitter emits ultrasonic waves in a certain direction, and the timer starts timing at the same time as the launch time. The ultrasonic waves propagate in the air and encounter obstacles on the way. When the object surface (object) is blocked, it will be reflected back immediately, and the ultrasonic receiver will immediately stop timing when the reflected ultrasonic wave is received. The propagation speed of ultrasonic waves in the air is 340m/s. According to the time t recorded by the timer, the distance s from the launch point to the obstacle surface can be calculated, namely: s=340t/2. Using this principle of ultrasound, the ultrasonic ranging module is widely used in practical applications, such as car reversing radar, unmanned aerial vehicle, and smart car.
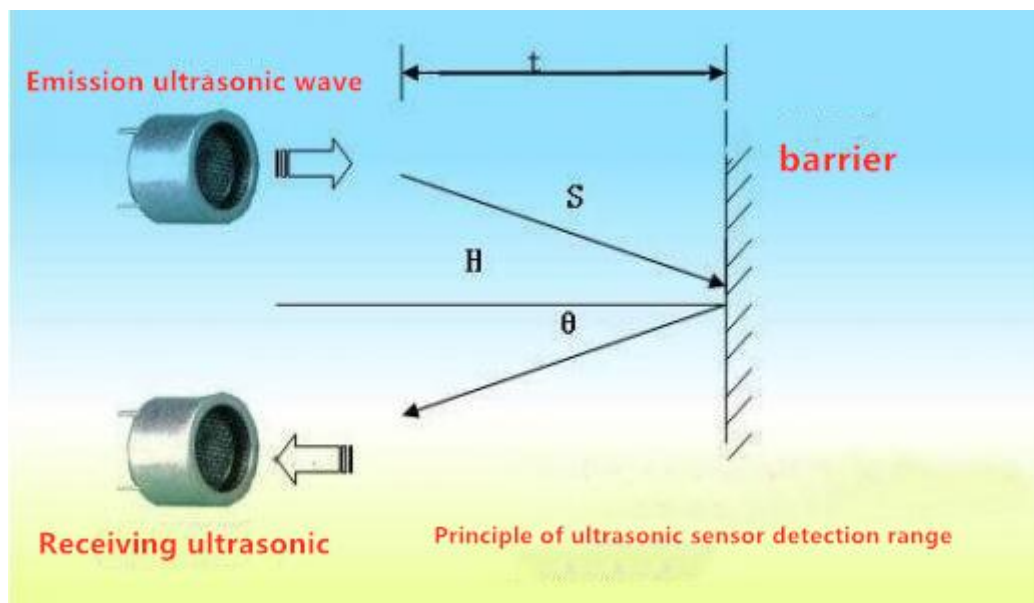


When using Robot HAT driver board, the ultrasonic sensor needs to be connected to the Ultrasonic interface on the driver board, and must not be connected to the IIC port to avoid burning

the ultrasonic module. (IIC is an interface used to connect I2C devices, and the pin positions of VCC and GND are different from Ultrasonic).



## 8.3Wiring diagram (Circuit diagram)

When using Robot HAT driver board, you need to connect the ultrasonic sensor to the Ultrasonic interface on the driver board. Do not connect it to the IIC port to avoid burning the ultrasonic module. (IIC is an interface used to connect I2C devices, and the pin positions of VCC and GND are different from Ultrasonic).
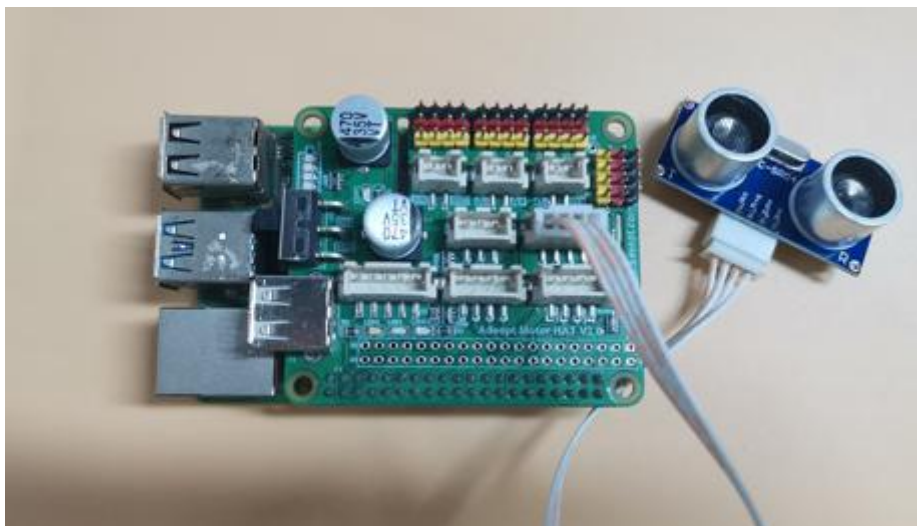


## 8.4 Obtain data from ultrasonic sensors

# Run the code

1. Remotely log in to the Raspberry Pi terminal.

```
Linux raspberrypi 4.19.118-v7l+ #1311 SMP Mon Apr 27 14:26:42 BST 2020 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Aug 29 08:17:49 2020 from 192.168.3.208

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
 a new password.

pi@raspberrypi:~ $
```

2. Enter the command and press Enter to enter the folder where the program is located:

**cd adeept_rasptank/server/**

```
pi@raspberrypi:~ $
pi@raspberrypi:~ $ cd adeept_rasptank/server/
pi@raspberrypi:~/adeept_rasptank/server $
```

3. View the contents of the current directory file:

**ls**

```
pi@raspberrypi:~/adeept_rasptank/server $
pi@raspberrypi:~/adeept_rasptank/server $ ls
app.py              FPV.py              move.py             server.py
appserverAP.py      FPVtest.py          OLED.py             serverTest.py
appserver.py        functions.py        PID.py              servo.py
base_camera.py      info.py             __pycache__         switch.py
camera_opencv.py    Instruction.txt     raspi-config.py     ultra.py
config.txt          Kalman_filter.py    requirements.txt    webServer.py
dist                LEDapp.py           robotLight.py
findline.py         LED.py              RPIservo.py
pi@raspberrypi:~/adeept_rasptank/server $
```

4. Enter the command and press Enter to run the program:

**sudo python3 ultra.py**

```
pi@raspberrypi:~/adeept_rasptank/server $
pi@raspberrypi:~/adeept_rasptank/server $ sudo python3 ultra.py
11.88 cm
11.24 cm
12.60 cm
12.77 cm
12.20 cm
```

5. After successfully running the program, the command window will display the distance data of the obstacle detected by the ultrasonic sensor. When you use an object to approach it directly in front of the ultrasonic sensor, the detected distance data will change.

6. When you want to terminate the running program, you can press the shortcut key "Ctrl + C" on the keyboard.

## 8.5 The main code program of this lesson

For the complete code, please refer to the file ultra.py.

```python
1.  import RPi.GPIO as GPIO
2.  import time
3.
4.  Tr = 11 # The pin number of the input end of the ultrasonic module
5.  Ec = 8 # Pin number of the output end of the ultrasonic module
6.
7.  GPIO.setmode(GPIO.BCM)
8.  GPIO.setup(Tr, GPIO.OUT,initial=GPIO.LOW)
9.  GPIO.setup(Ec, GPIO.IN)
```

Import the dependent library and initialize it.

```python
1.  def checkdist():
2.      GPIO.output(Tr, GPIO.HIGH) # Set the input terminal of the module to high level and send out an initial sound wave.
3.      time.sleep(0.000015)
4.      GPIO.output(Tr, GPIO.LOW)
5.
6.      while not GPIO.input(Ec): # When the module no longer receives the initial sound wave ①
7.          pass
8.      t1 = time.time() # Write down the time when the initial sound wave was emitted.
```

```
9.      while GPIO.input(Ec): # When the module receives the return sound wave.
10.       pass
11.    t2 = time.time() # Write down the time when the return sound wave was captured.
12.
13.    return round((t2-t1)*340/2,2) # Calculate the distance.
```

Define the ranging function. When the ultrasound module captures ultrasound, its output terminal will output high frequency. The GPIO.input method can determine whether the output terminal is outputting high-frequency signals.

```
1.  for i in range(10):
2.     print(checkdist())
3.     time.sleep(1)
```

The ultrasonic module measures the distance and displays the data on the console.