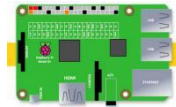
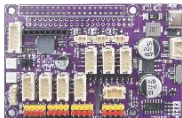




Lesson 12 Measure the Distance with an Ultrasonic Sensor

12.1 Overview

In this lesson, you'll learn to use an ultrasonic sensor with Raspberry Pi and Adeept Robot HAT V3.2 for distance measurement. It covers components, the principle, wiring, running the program step - by - step, and code implementation. By the end, you'll understand ultrasonic sensor operation and its integration into Raspberry Pi projects for distance - sensing.

12.2 Required Components

Components	Quantity	Picture
Raspberry Pi	1	
Adeept Robot HAT V3.2	1	
ultrasonic module	1	
4 pin cable	1	

12.3 Principle Introduction

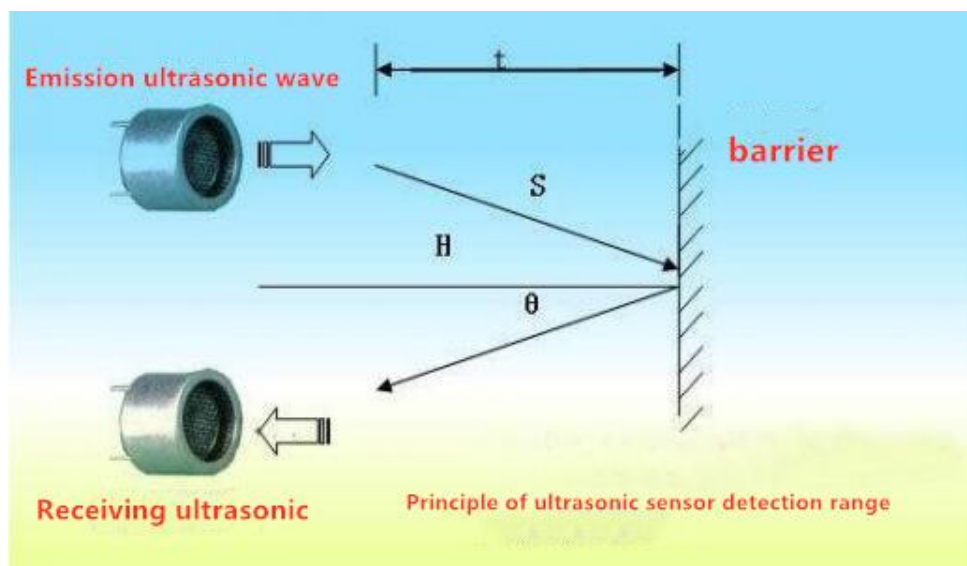
The ultrasonic ranging module used has four pins, namely VCC, GND, Echo and Trig. The HC-SR04 can provide a non-contact distance sensing function of 2cm-400cm, and the ranging accuracy can reach 3mm; The module includes an ultrasonic transmitter, receiver and control circuit. The basic working principle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

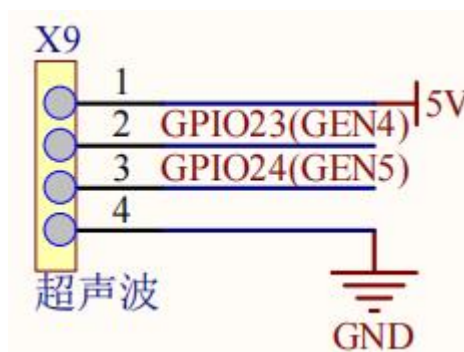
The principle of distance detection by ultrasonic ranging sensor: the method of detecting distance by ultrasonic is called echo detection method, that is, the ultrasonic transmitter emits ultrasonic waves in a certain direction, and the timer starts timing at the same time as the launch time. The ultrasonic 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.

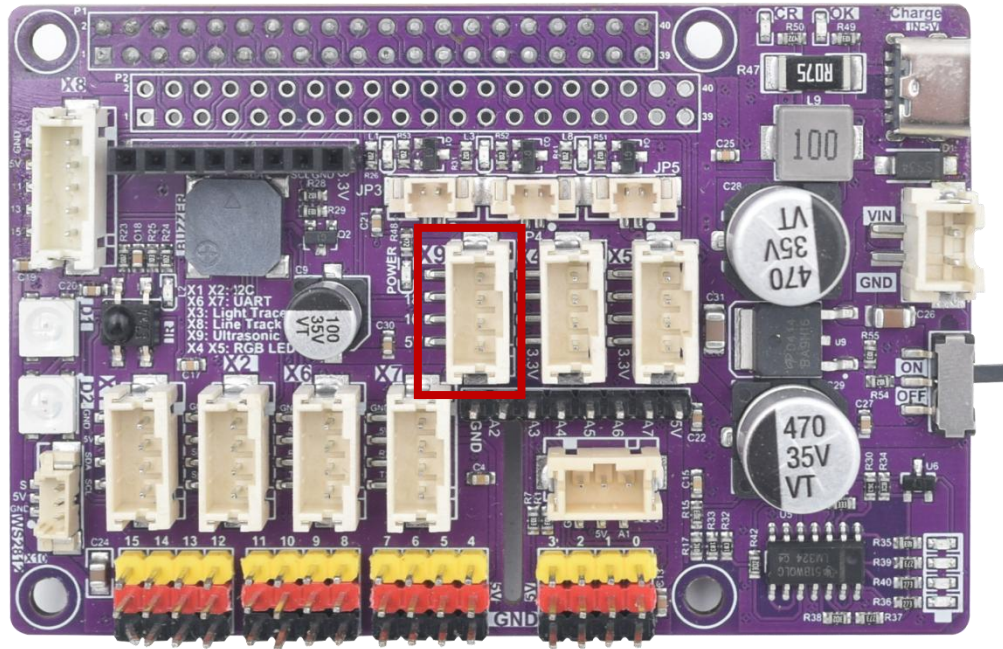


When using Adeept Robot HAT V3.2 expansion board, the ultrasonic sensor needs to be connected to the Ultrasonic interface on the driver board, and must not be connected to the IIC port to avoid burning the ultrasonic module. (IIC is an interface used to connect I2C devices, and the pin positions of VCC and GND are different from Ultrasonic).

PINS of Raspberry Pi	Ultrasonic
GPIO23	Trig
GPIO24	Echo
VCC	VCC
GND	GND

12.4 Wiring Diagram





12.5 Demonstration

1. **Remotely log:** Remotely log in to the Raspberry Pi terminal.
2. **Navigate to the Program Folder:** Enter the following command in the terminal and press Enter to access the folder where the program is located:

```
cd Adeept_4WD_Smart_Car_for_RPi/Examples/06_Ultrasonic/
```

```
pi@raspberrypi:~ $ cd Adeept_4WD_Smart_Car_for_RPi/Examples/06_Ultrasonic/
pi@raspberrypi:~/Adeept_4WD_Smart_Car_for_RPi/Examples/06_Ultrasonic $
```

3. **View Directory Contents:** Type "ls" in the terminal and press Enter. This will display all the files in the current directory, ensuring that the "**Ultrasonic.py**" file is present:

```
ls
```

```
pi@raspberrypi:~/Adeept_4WD_Smart_Car_for_RPi/Examples/06_Ultrasonic $ ls
Ultrasonic.py
```

4. Run the Program: Enter the command below and press Enter to start the Ultrasonic.py program:

```
sudo python3 Ultrasonic.py
```

```
pi@raspberrypi:~/Adeept_4WD_Smart_Car_for_RPi/Examples/06_Ultrasonic $ sudo python3 Ultrasonic.py
/usr/lib/python3/dist-packages/gpiozero/input_devices.py:852: PWMSoftwareFallback: For more accurate
readings, use the pigpio pin factory. See https://gpiozero.readthedocs.io/en/stable/api_input.html#dis
tancesensor-hc-sr04 for more info
warnings.warn(PWMSoftwareFallback(
200.00 cm
200.00 cm
200.00 cm
200.00 cm
200.00 cm
200.00 cm
200.00 cm
200.00 cm
200.00 cm
200.00 cm
19.15 cm
19.15 cm
19.15 cm
```

5. Observation and Termination: Continuously read sensor data through a loop, format it into centimeter units with two decimal places, and output it in real-time to the terminal. When you want to terminate the running program, you can press the shortcut key "**Ctrl + C**" on the keyboard.

12.6 Code

Complete code refer to [Ultrasonic.py](#)

```
01 #!/usr/bin/env/python
02 # File name   : Ultrasonic.py
03 # Website    : www.Adeept.com
04 # Author     : Adeept
05 # Date       : 2025/03/7
06 from gpiozero import DistanceSensor
07 from time import sleep
08
09 Tr = 23
10 Ec = 24
11 sensor = DistanceSensor(echo=Ec, trigger=Tr, max_distance=2) # Maximum detection distance 2m.
12
13 # Get the distance of ultrasonic detection.
14 def checkdist():
15     return (sensor.distance) *100 # Unit: cm
16
17 if __name__ == "__main__":
18     while True:
```

```
19 distance = checkdist()
20 print("%.2f cm" %distance)
21 sleep(0.5)
```

Code explanation

Initialization Stage:

Define a function named checkdist to obtain the distance detected by the ultrasonic sensor.

Loop Control Process:

After entering an infinite loop, execute the following steps in sequence:

Stage 1: Obtaining the Distance: distance = checkdist() calls the checkdist function to get the currently detected distance and stores the result in the variable distance.

Stage 2:Outputting the Distance: print("%.2f cm" % distance) uses a formatted string to output the distance to the console with two decimal places, and the unit is centimeters.

Stage 3: Delaying: sleep(0.5) calls the sleep function to pause the program for 0.5 seconds, avoiding overly frequent distance detections.

This code initializes the ultrasonic sensor, enters an infinite loop to continuously detect and output the distance, thus implementing a simple distance detection system.