

Grove - Ultrasonic Ranger



V1.0



V2.0

This Grove - Ultrasonic ranger is a non-contact distance measurement module which works at 40KHz. When we provide a pulse trigger signal with more than 10uS through signal pin, the Grove_Ultrasonic_Ranger will issue 8 cycles of 40kHz cycle level and detect the echo. The pulse width of the echo signal is proportional to the measured distance. Here is the formula: Distance = echo signal high time * Sound speed (340M/S)/2. Grove_Ultrasonic_Ranger's trig and echo signal share 1 SIG pin.

Warning

Do not hot plug Grove-Ultrasonic-Ranger, otherwise it will damage the sensor. The measured area must be no less than 0.5 square meters and smooth.

Version

Product Version	Changes	Released Date
Grove-Ultrasonic ranger V1.0	Initial	Mar 2012
Grove-Ultrasonic ranger V2.0	Improve the power stability with low-voltage main board with below changes: 1. Added an capacitance C14 2. Redesigned the layout to make it more tidy 3. Compatible with 3.3V voltage system	July 2017


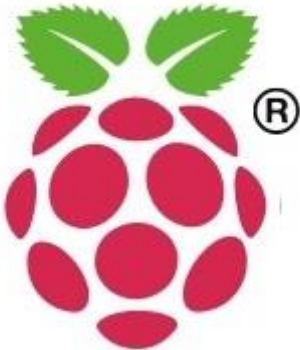

Specification

Parameter	Value/Range
Operating voltage	3.2~5.2V
Operating current	8mA
Ultrasonic frequency	40kHz
Measuring range	2-350cm
Resolution	1cm
Output	PWM
Size	50mm X 25mm X 16mm
Weight	13g
Measurement angle	15 degree
Working temperature	-10~60 degree C
Trigger signal	10uS TTL
Echo signal	TTL

Tip

More details about Grove modules please refer to [Grove System](#)

Platforms Supported

Arduino	Raspberry	ArduPy
		

Caution

The platforms mentioned above as supported is/are an indication of the module's software or theoretical compatibility. We only provide software library or code examples for Arduino platform in most cases. It is not possible to provide software library / demo code for all possible MCU platforms. Hence, users have to write their own software library.

Getting Started

Note

If this is the first time you work with Arduino, we firmly recommend you to see [Getting Started with Arduino](#) before the start.

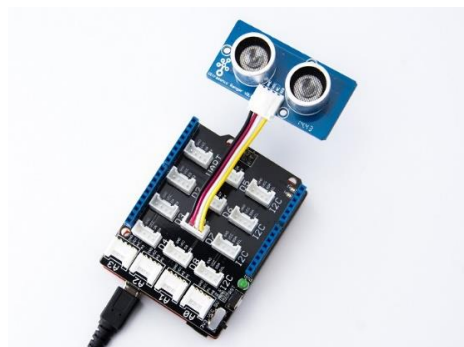
Play With Arduino

Hardware

- **Step 1.** Prepare the below stuffs:

Seeeduino V4.2	Base Shield	Grove - Ultrasonic Ranger
		

- **Step 2.** Connect Ultrasonic Ranger to port D7 of Grove-Base Shield.
- **Step 3.** Plug Grove - Base Shield into Seeeduino.
- **Step 4.** Connect Seeeduino to PC via a USB cable.



Note

If we don't have Grove Base Shield, We also can directly connect Grove_Ultrasonic_Ranger to Seeeduino as below.

Seeeduino	Grove-Ultrasonic Ranger
5V	Red
GND	Black
Not Conencted	White
D7	Yellow

Software

- **Step 1.** Download the [UltrasonicRanger Library](#) from Github.
- **Step 2.** Refer [How to install library](#) to install library for Arduino.
- **Step 3.** Copy the code into Arduino IDE and upload. If you do not know how to upload the code, please check [how to upload code](#).

```
1  #include "Ultrasonic.h"
2
3  Ultrasonic ultrasonic(7);
4  void setup()
5  {
6      Serial.begin(9600);
7  }
8  void loop()
9  {
10     long RangeInInches;
11     long RangeInCentimeters;
12
13     Serial.println("The distance to obstacles in front is: ");
14     RangeInInches = ultrasonic.MeasureInInches();
15     Serial.print(RangeInInches);//0~157 inches
16     Serial.println(" inch");
17     delay(250);
18
19     RangeInCentimeters = ultrasonic.MeasureInCentimeters(); // two measurements shou
20     Serial.print(RangeInCentimeters);//0~400cm
21     Serial.println(" cm");
22     delay(250);
23 }
```

- **Step 4.** We will see the distance display on terminal as below.

```
1 The distance to obstacles in front is:  
2 2 inches  
3 6 cm  
4 The distance to obstacles in front is:  
5 2 inches  
6 6 cm  
7 The distance to obstacles in front is:  
8 2 inches  
9 6 cm
```

Play with Codecraft

Hardware

Step 1. Connect a Grove - Ultrasonic Ranger to port D7 of a Base Shield.

Step 2. Plug the Base Shield to your Seeeduino/Arduino.

Step 3. Link Seeeduino/Arduino to your PC via an USB cable.

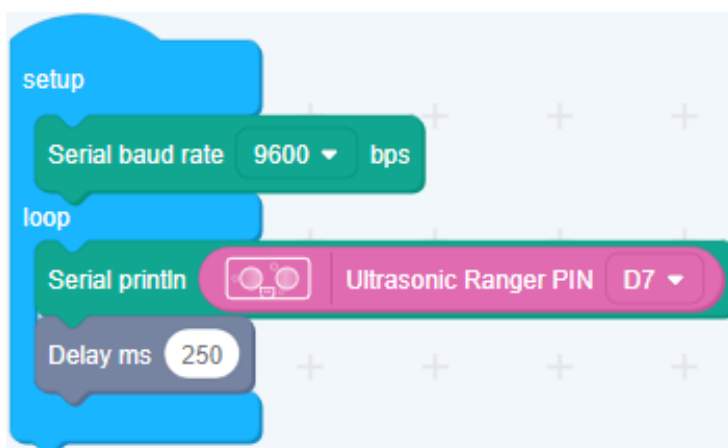
Software

Step 1. Open [Codecraft](#), add Arduino support, and drag a main procedure to working area.

Note

If this is your first time using Codecraft, see also [Guide for Codecraft using Arduino](#).

Step 2. Drag blocks as picture below or open the cdc file which can be downloaded at the end of this page.



Upload the program to your Arduino/Seeeduino.

Success

When the code finishes uploaded, you will see distance displayed in Serial Monitor.

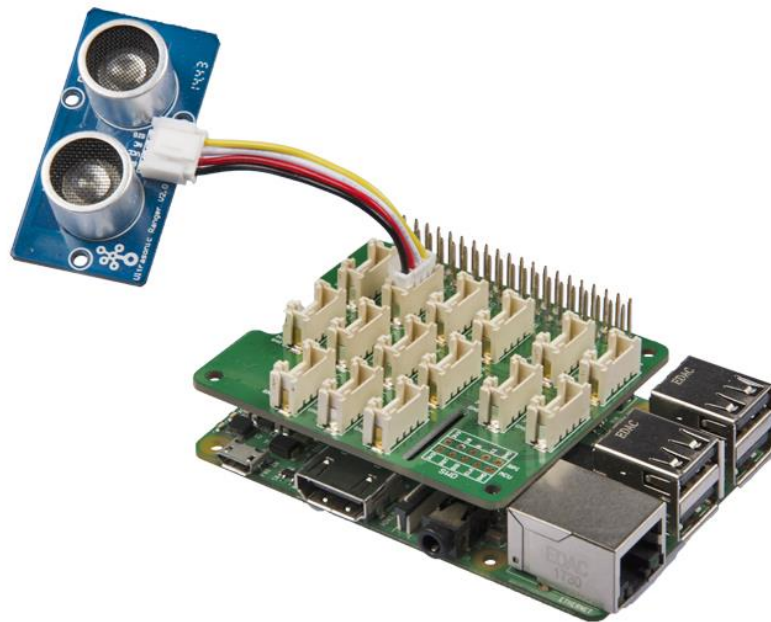
Play With Raspberry Pi (With Grove Base Hat for Raspberry Pi)

Hardware

- **Step 1.** Things used in this project:

Raspberry pi	Grove Base Hat for RasPi	Grove - Ultrasonic Ranger
		

- **Step 2.** Plug the Grove Base Hat into Raspberry.
- **Step 3.** Connect the Grove - Ultrasonic Ranger to port D5 of the Base Hat.
- **Step 4.** Connect the Raspberry Pi to PC through USB cable.





Note

For step 3 you are able to connect the ultrasonic ranger to **any GPIO Port** but make sure you change the command with the corresponding port number.

Software



Attention

If you are using **Raspberry Pi with Raspberrypi OS >= Bullseye**, you have to use this command line **only with Python3**.

- **Step 1.** Follow [Setting Software](#) to configure the development environment.
- **Step 2.** Download the source file by cloning the grove.py library.

```
1 cd ~
2 git clone https://github.com/Seeed-Studio/grove.py
```

- **Step 3.** Excute below commands to run the code.

```
1 cd grove.py/grove
2 python3 grove_ultrasonic_ranger.py 5 6
```

Following is the grove_ultrasonic_ranger.py code.

```
1 import sys
2 import time
3 from grove.gpio import GPIO
4
5 usleep = lambda x: time.sleep(x / 1000000.0)
6
7 _TIMEOUT1 = 1000
8 _TIMEOUT2 = 10000
9
10 class GroveUltrasonicRanger(object):
11     def __init__(self, pin):
12         self.dio =GPIO(pin)
13
14     def _get_distance(self):
15         self.dio.dir(GPIO.OUT)
16         self.dio.write(0)
```

```

17     usleep(2)
18     self.dio.write(1)
19     usleep(10)
20     self.dio.write(0)
21
22     self.dio.dir(GPIO.IN)
23
24     t0 = time.time()
25     count = 0
26     while count < _TIMEOUT1:
27         if self.dio.read():
28             break
29         count += 1
30     if count >= _TIMEOUT1:
31         return None
32
33     t1 = time.time()
34     count = 0
35     while count < _TIMEOUT2:
36         if not self.dio.read():
37             break
38         count += 1
39     if count >= _TIMEOUT2:
40         return None
41
42     t2 = time.time()
43
44     dt = int((t1 - t0) * 1000000)
45     if dt > 530:
46         return None
47
48     distance = ((t2 - t1) * 1000000 / 29 / 2)    # cm
49
50     return distance
51
52     def get_distance(self):
53         while True:
54             dist = self._get_distance()
55             if dist:
56                 return dist
57
58 Grove = GroveUltrasonicRanger
59
60 def main():
61     if len(sys.argv) < 2:
62         print('Usage: {} pin_number'.format(sys.argv[0]))
63         sys.exit(1)
64
65     sonar = GroveUltrasonicRanger(int(sys.argv[1]))
66
67     print('Detecting distance...')
68     while True:
69         print('{} cm'.format(sonar.get_distance()))
70         time.sleep(1)
71
72 if __name__ == '__main__':
73     main()

```




Success

If everything goes well, you will be able to see the following result




```
1 pi@raspberrypi:~/grove.py/grove $ python3 grove_ultrasonic_ranger.py 5 6
2 Detecting distance...
3 121.757901948 cm
4 246.894770655 cm
5 2.60205104433 cm
6 0.205533257846 cm
7 0.657706425108 cm
8 247.433267791 cm
9 122.485489681 cm
10 ^CTraceback (most recent call last):
11   File "grove_ultrasonic_ranger.py", line 110, in <module>
12     main()
13   File "grove_ultrasonic_ranger.py", line 107, in main
14     time.sleep(1)
15 KeyboardInterrupt
```

You can quit this program by simply press `Ctrl + C`.

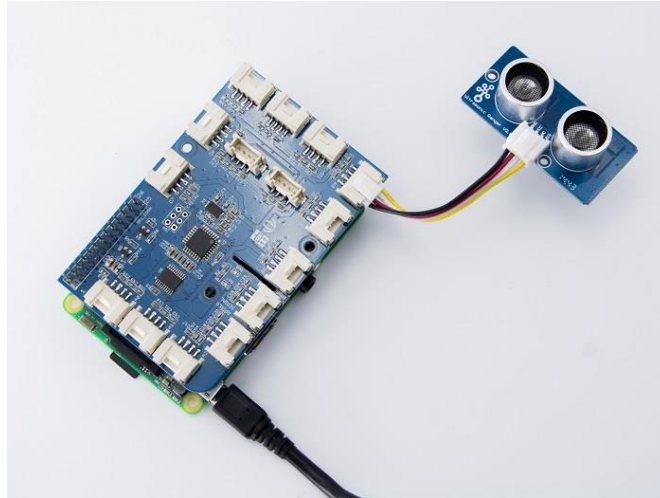
Play With Raspberry Pi (with GrovePi_Plus)

Hardware

- **Step 1.** Prepare the below stuffs:

Raspberry pi	GrovePi_Plus	Grove - Ultrasonic Ranger
		

- **Step 2.** Plug the GrovePi_Plus into Raspberry.
- **Step 3.** Connect Grove-Ultrasonic ranger to D4 port of GrovePi_Plus.
- **Step 4.** Connect the Raspberry to PC through USB cable.



Software



Attention

If you are using **Raspberry Pi with Raspberrypi OS >= Bullseye**, you have to use this command line **only with Python3**.

- **Step 1.** Follow [Setting Software](#) to configure the development environment.
- **Step 2.** Git clone the Github repository.

```
1 cd ~
2 git clone https://github.com/DexterInd/GrovePi.git
```

- **Step 3.** Excute below commands to use the ultrasonic_ranger to measure the distance.

```
1 cd ~/GrovePi/Software/Python
2 python3 grove_ultrasonic.py
```

Here is the grove_ultrasonic.py code.

```
1 # GrovePi + Grove Ultrasonic Ranger
2
3 from grovepi import *
4
5 # Connect the Grove Ultrasonic Ranger to digital port D4
6 # SIG,NC,VCC,GND
7
8 ultrasonic_ranger = 4
9
10 while True:
11     try:
12         # Read distance value from Ultrasonic
13         print ultrasonicRead(ultrasonic_ranger)
14
15     except TypeError:
16         print "Error"
17     except IOError:
18         print "Error"
```

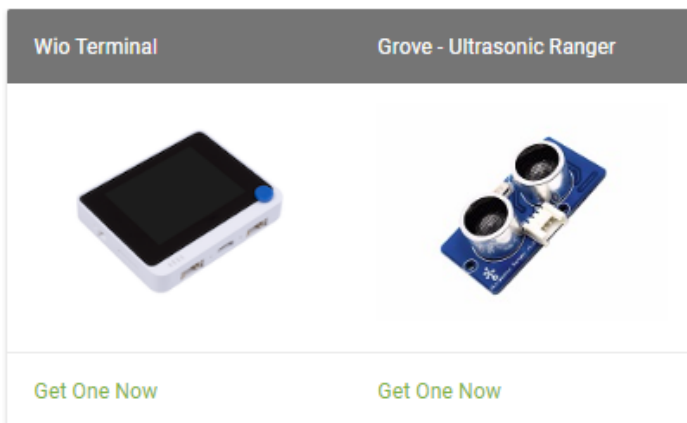
- **Step 4.** We will see the distance display on terminal as below.

```
1 pi@raspberrypi:~/GrovePi/Software/Python $ python3 grove_ultrasonic.py
2 9
3 9
4 9
5 9
6 9
7 9
8 9
9 9
10 9
11 9
12 9
```

Play With Wio Terminal (ArduPy)

Hardware

- **Step 1.** Prepare the below stuffs:



- **Step 2.** Connect Grove - Ultrasonic Ranger to **D0** port of Wio Terminal.
- **Step 3.** Connect the Wio Terminal to PC through USB Type-C cable.



Software

- **Step 1.** Follow [ArduPy Getting Started](#) to configure the ArduPy development environment on Wio Terminal.
- **Step 2.** Make sure that the ArduPy firmware contains the ultrasonic ranger ArduPy library using the following commands. For more information, please follow [here](#).

```
1 aip install Seeed-Studio/seeed-ardupy-ultrasonic-sensor
2 aip build
3 aip flash
```

- **Step 3.** Copy the following code and save it as `ArduPy-ultrasonic.py`:

```
1 from arduino import grove_ultra_ranger
2 from machine import LCD
3 from machine import Sprite
4 import time
5
6 Ultrasonic = grove_ultra_ranger(0)
7 lcd = LCD()
8 spr = Sprite(lcd) # Create a buff
9
10 def main():
11     spr.createSprite(320, 240)
12     while True:
13         spr.setTextSize(2)
14         spr.fillSprite(spr.color.BLACK)
15         spr.setTextColor(lcd.color.RED)
16         spr.drawString("Ultrasonic Sensor", 55, 10)
17         spr.drawFastHLine(40, 35, 240, lcd.color.DARKGREY)
18         spr.setTextColor(lcd.color.WHITE)
19         spr.drawString("- Centimeters: ", 20, 50)
20         spr.drawString("- Inches: ", 20, 80)
21         spr.drawNumber(Ultrasonic.cm, 200, 50)
22         time.sleep_ms(50) # Needed for data to be read again
23         spr.drawNumber(Ultrasonic.inch, 130, 80)
24         spr.pushSprite(0, 0)
25         time.sleep_ms(500)
26
27     print("The distance to obstacles in front is:", Ultrasonic.cm, 'centimeter')
28     print("The distance to obstacles in front is:", Ultrasonic.inch, 'inches')
29
30 if __name__ == "__main__":
31     main()
```

- **Step 4.** Save the `ArduPy-ultrasonic.py` in a location that you know. Run the following command and **replace** `<YourPythonFilePath>` with your `ArduPy-ultrasonic.py` location.

```
1 aip shell -n -c "runfile <YourPythonFilePath>"
2 # Example:
3 # aip shell -n -c "runfile /Users/ansonhe/Desktop/ArduPy-ultrasonic.py"
```

- **Step 5.** We will see the distance display on terminal as below, and displaying on the Wio Terminal LCD screen.

```
1 ansonhe@Ansons-Macbook-Pro ~: aip shell -n -c "runfile /Users/ansonhe/Desktop/ArduPy
2 Positional argument (/dev/cu.usbmodem1413101) takes precedence over --open.
3 Connected to ardupy
4 The distance to obstacles in front is: 3 centimeter
5 The distance to obstacles in front is: 1 inches
6 The distance to obstacles in front is: 7 centimeter
7 The distance to obstacles in front is: 2 inches
8 The distance to obstacles in front is: 6 centimeter
9 The distance to obstacles in front is: 2 inches
10 The distance to obstacles in front is: 6 centimeter
11 The distance to obstacles in front is: 2 inches
12 The distance to obstacles in front is: 4 centimeter
13 The distance to obstacles in front is: 1 inches
14 The distance to obstacles in front is: 7 centimeter
15 The distance to obstacles in front is: 2 inches
```



FAQs

Q1: How does the Grove-Ultrasonic sensor work?

- A1: When we provide a pulse trigger signal with more than 10uS through signal pin, the Grove_Ultrasonic_Ranger will issue 8 cycles of 40kHz cycle level and detect the echo. The pulse width of the echo signal is proportional to the measured distance. Here is the formula:
Distance = echo signal high time * Sound speed (340M/S)/2.

Q2: Why Grove-Ultrasonic sensor only has 1 signal pin, comparing with other ultrasonic sensor Trig and Echo pins?

- A2: Grove_Ultrasonic_Ranger's trig and echo signal share 1 SIG pin through MCU.

Q3: Can we connect multiple ultrasonic to one arduino?

- A4: Yes, Here is the example, one sensor is connected to D2 and other to D3.

```
1 #include "Ultrasonic.h"
2
3 Ultrasonic ultrasonic1(2);
4 Ultrasonic ultrasonic2(3);
5 void setup()
6 {
7     Serial.begin(9600);
8 }
9 void loop()
10 {
11     long RangeInCentimeters1;
12     long RangeInCentimeters2;
13
14     RangeInCentimeters1 = ultrasonic1.MeasureInCentimeters(); // two measurements sh
15     Serial.print(RangeInCentimeters1); //0~400cm
16     Serial.println(" cm");
17
18     RangeInCentimeters2 = ultrasonic2.MeasureInCentimeters(); // two measurements sh
19     Serial.print(RangeInCentimeters2); //0~400cm
20     Serial.println(" cm");
21
22     delay(250);
23 }
```

Resources

- [PDF] [Download Wiki PDF](#)
- [PDF] [Grove_Ultrasonic Ranger Schematic](#)
- [PDF] [Ceramic Ultrasonic Sensor NU40C16T/R-1](#)
- [Library] [Grove Ultrasonic Ranger library](#)
- [Codecraft] [CDC File](#)
- [Project] [The Color Helix](#)
- [Project] [Indoor Lightning Cloud](#)
- [Project] [Automatic Water Level Controller](#)
- [Example] [Example Measure distance and led display](#)
- [Example] [Example Measure and display the distance](#)

Project

Hacking the Stairs at Seeed's New Office: Turn the stairs at the office into an interactive installation, and even a cool way to convey the message "STAFF ONLY" to visitors.

Tech Support

Please submit any technical issue into our [forum](#).