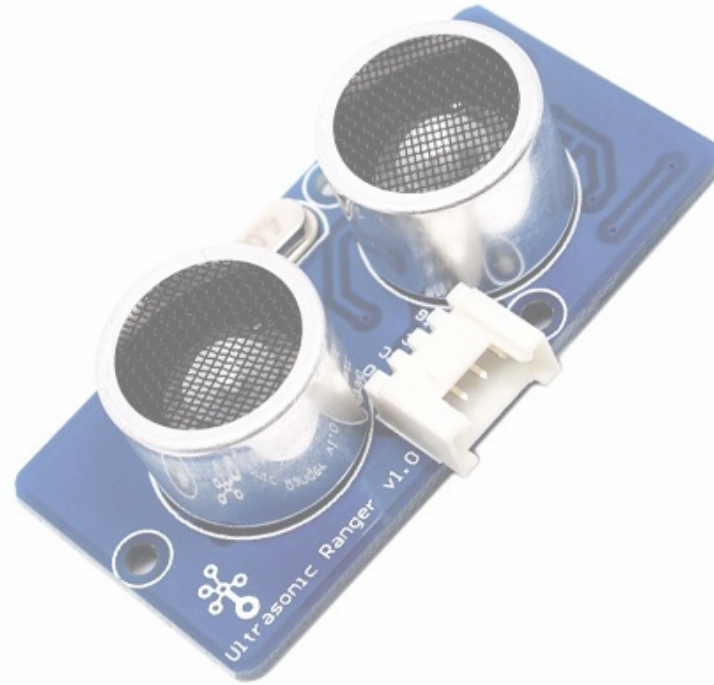


Grove - Ultrasonic Ranger



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

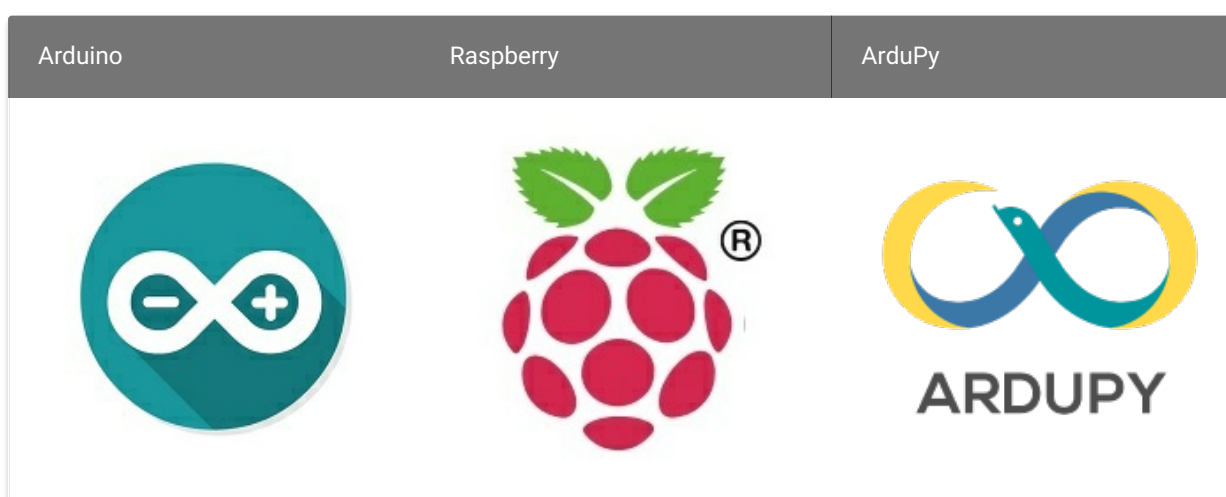
Product Version	Changes	Released Date
Grove-Ultrasonic ranger V1.0	Initial	Mar 2012
Grove-Ultrasonic ranger V2.0	Improve the power stability with low-voltage main board with below changes: 1. Added an capacitance C14 2. Redesigned the layout to make it more tidy 3. Compatible with 3.3V voltage system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



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.

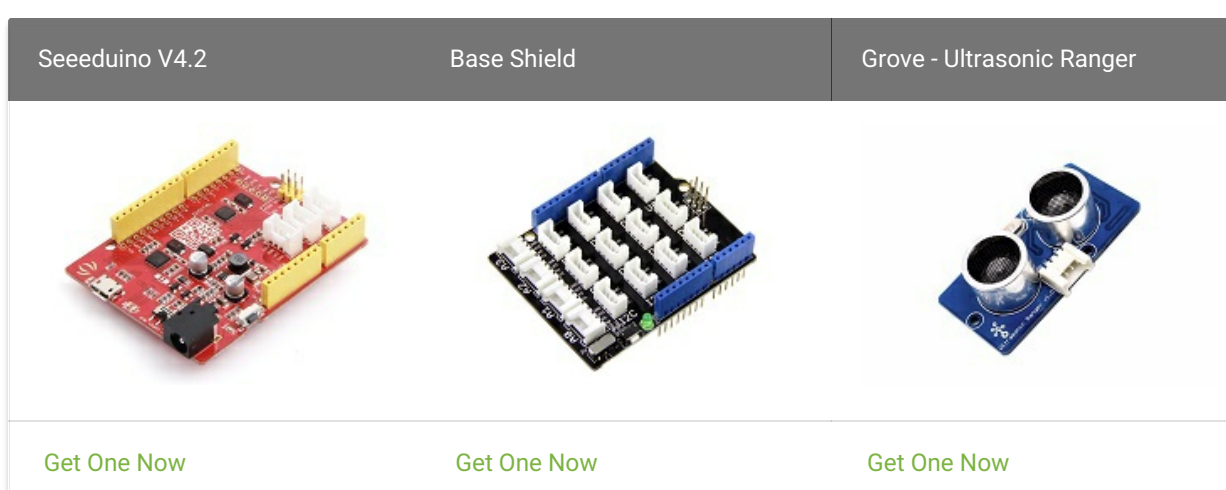
Getting Started

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

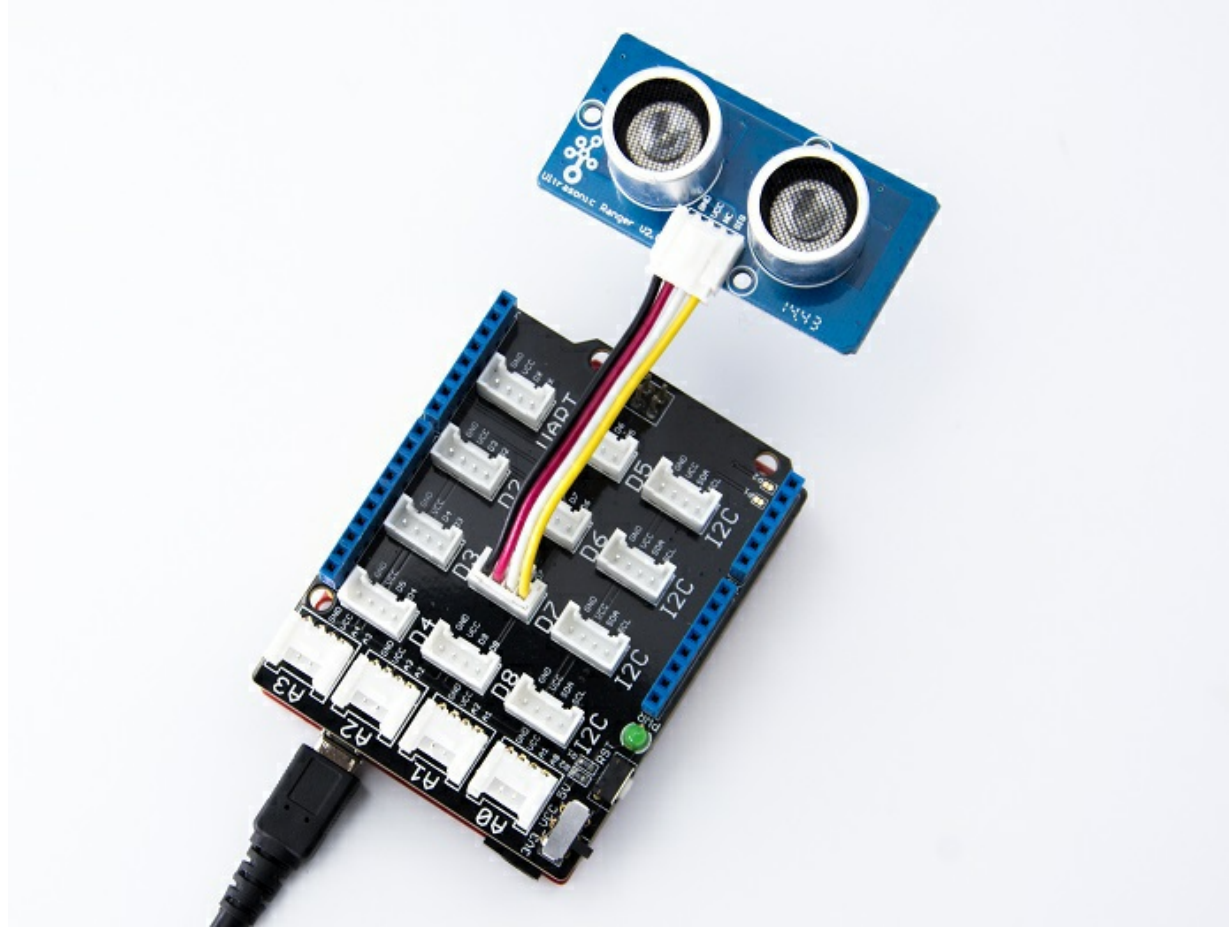
Play With Arduino

Hardware

- Step 1. Prepare the below stuffs:



- 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 Connected	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](#).

```
#include "Ultrasonic.h"

Ultrasonic ultrasonic(7);
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  long RangeInInches;
  long RangeInCentimeters;

  Serial.println("The distance to obstacles in front is: ");
  RangeInInches = ultrasonic.MeasureInInches();
  Serial.print(RangeInInches);//0~157 inches
  Serial.println(" inch");
  delay(250);

  RangeInCentimeters = ultrasonic.MeasureInCentimeters(); // two measurements should keep an interval
  Serial.print(RangeInCentimeters);//0~400cm
  Serial.println(" cm");
  delay(250);
}
```

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

```
The distance to obstacles in front is:
2 inches
6 cm
The distance to obstacles in front is:
2 inches
6 cm
The distance to obstacles in front is:
2 inches
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.

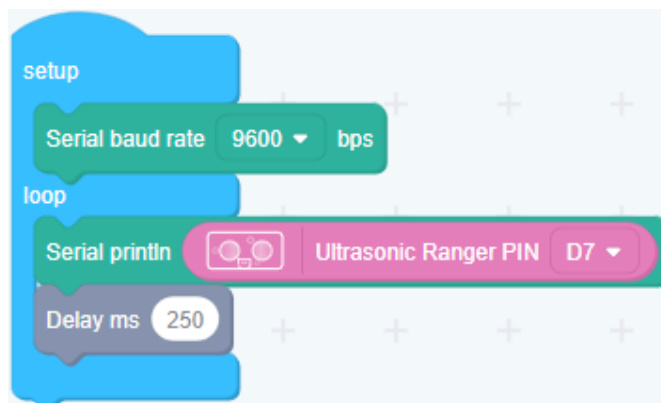
Software

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.

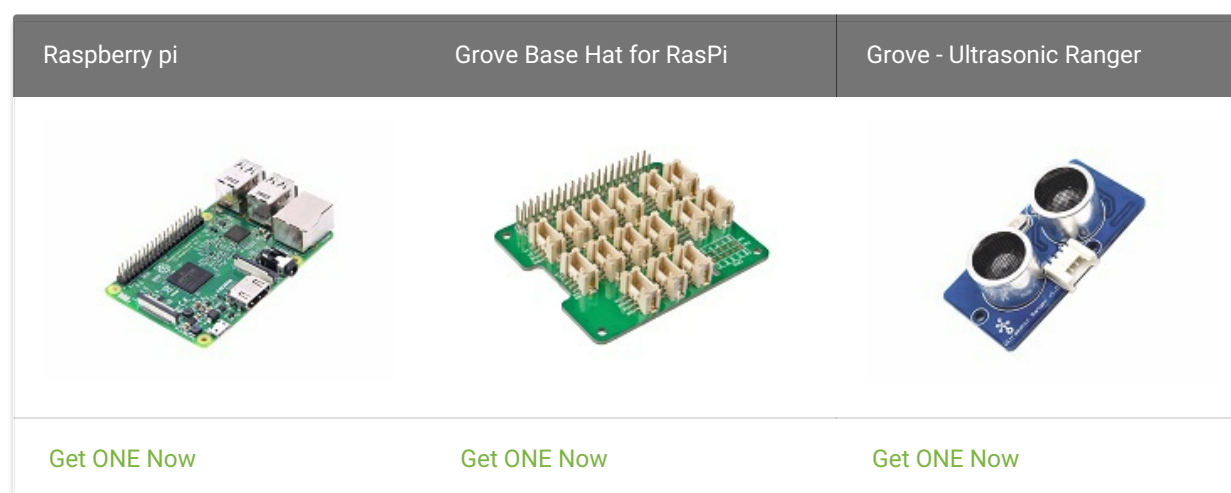
Success

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

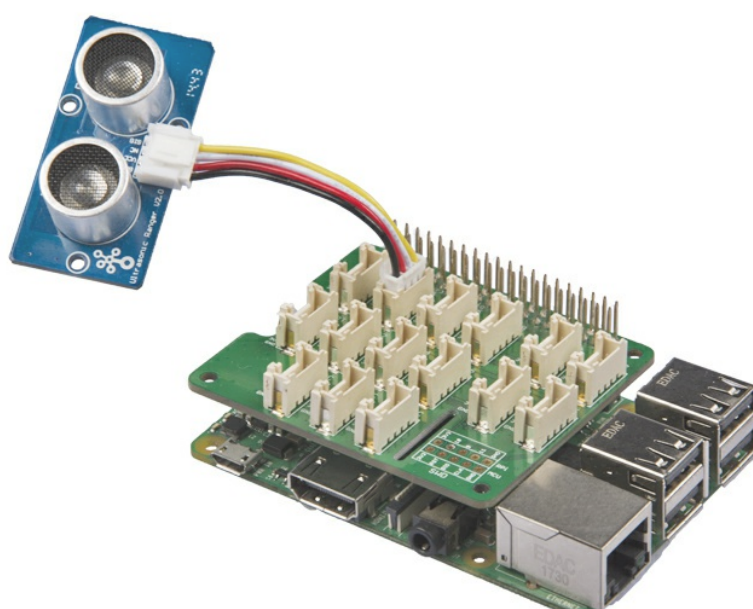
Play With Raspberry Pi (With Grove Base Hat for Raspberry Pi)

Hardware

- Step 1. Things used in this project:



- 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

- Step 1. Follow [Setting Software](#) to configure the development environment.

- Step 2. Download the source file by cloning the grove.py library.

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

- Step 3. Execute below commands to run the code.

```
cd grove.py/grove
python grove_ultrasonic_ranger.py 5 6
```

Following is the grove_ultrasonic_ranger.py code.

```
import sys
import time
from grove.gpio import GPIO

usleep = lambda x: time.sleep(x / 1000000.0)

_TIMEOUT1 = 1000
_TIMEOUT2 = 10000

class GroveUltrasonicRanger(object):
    def __init__(self, pin):
        self.dio = GPIO(pin)

    def _get_distance(self):
        self.dio.dir(GPIO.OUT)
        self.dio.write(0)
        usleep(2)
        self.dio.write(1)
        usleep(10)
        self.dio.write(0)

        self.dio.dir(GPIO.IN)

        t0 = time.time()
        count = 0
        while count < _TIMEOUT1:
            if self.dio.read():
                break
            count += 1
        if count >= _TIMEOUT1:
            return None

        t1 = time.time()
        count = 0
        while count < _TIMEOUT2:
            if not self.dio.read():
                break
            count += 1
        if count >= _TIMEOUT2:
            return None

        t2 = time.time()

        dt = int((t1 - t0) * 1000000)
        if dt > 530:
            return None

        distance = ((t2 - t1) * 1000000 / 29 / 2) # cm

        return distance

    def get_distance(self):
        while True:
            dist = self._get_distance()
            if dist:
                return dist

Grove = GroveUltrasonicRanger

def main():
    if len(sys.argv) < 2:
        print('Usage: {} pin_number'.format(sys.argv[0]))
        sys.exit(1)

    sonar = GroveUltrasonicRanger(int(sys.argv[1]))

    print('Detecting distance...')
    while True:
        print('{} cm'.format(sonar.get_distance()))
        time.sleep(1)

if __name__ == '__main__':
    main()
```

Success

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

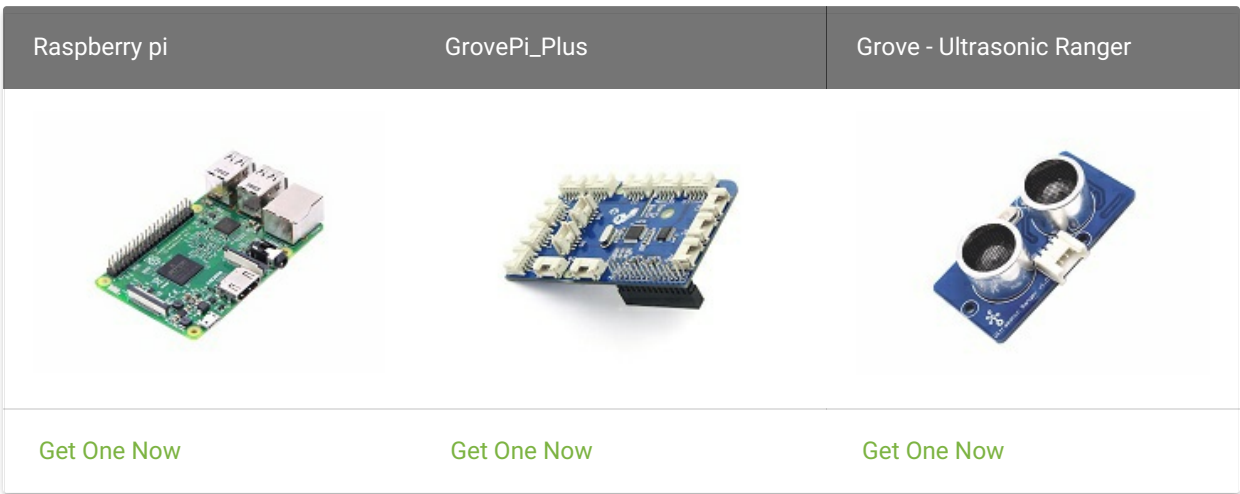
```
pi@raspberrypi:~/grove.py/grove $ python grove_ultrasonic_ranger.py 5 6
Detecting distance...
121.757901948 cm
246.894770655 cm
2.60205104433 cm
0.205533257846 cm
0.657706425108 cm
247.433267791 cm
122.485489681 cm
^C
Traceback (most recent call last):
  File "grove_ultrasonic_ranger.py", line 110, in <module>
    main()
  File "grove_ultrasonic_ranger.py", line 107, in main
    time.sleep(1)
KeyboardInterrupt
```

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

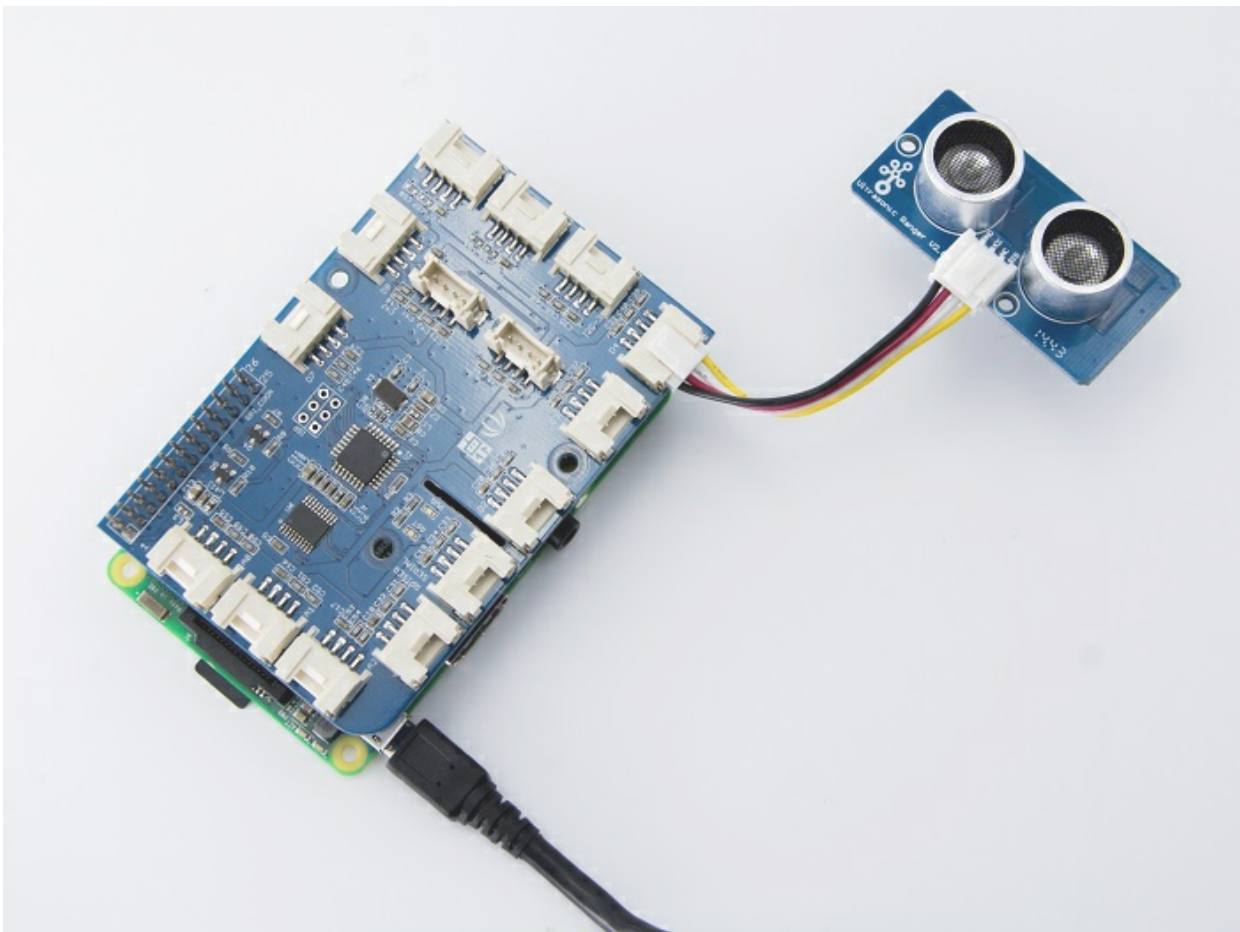
Play With Raspberry Pi (with GrovePi_Plus)

Hardware

- Step 1. Prepare the below stuffs:



- 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

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

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

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

```
cd ~/GrovePi/Software/Python
python grove_ultrasonic.py
```

Here is the grove_ultrasonic.py code.

```
# GrovePi + Grove Ultrasonic Ranger

from grovepi import *

# Connect the Grove Ultrasonic Ranger to digital port D4
# SIG,NC,VCC,GND

ultrasonic_ranger = 4

while True:
    try:
        # Read distance value from Ultrasonic
        print ultrasonicRead(ultrasonic_ranger)

    except TypeError:
        print "Error"
    except IOError:
        print "Error"
```

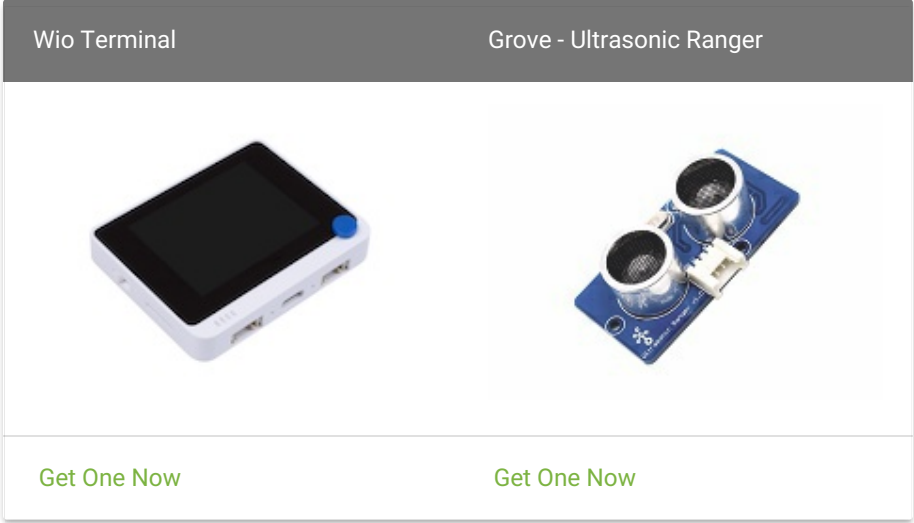
- Step 4. We will see the distance display on terminal as below.

```
pi@raspberrypi:~/GrovePi/Software/Python $ python grove_ultrasonic.py
9
9
9
9
9
9
9
9
9
9
9
9
9
```

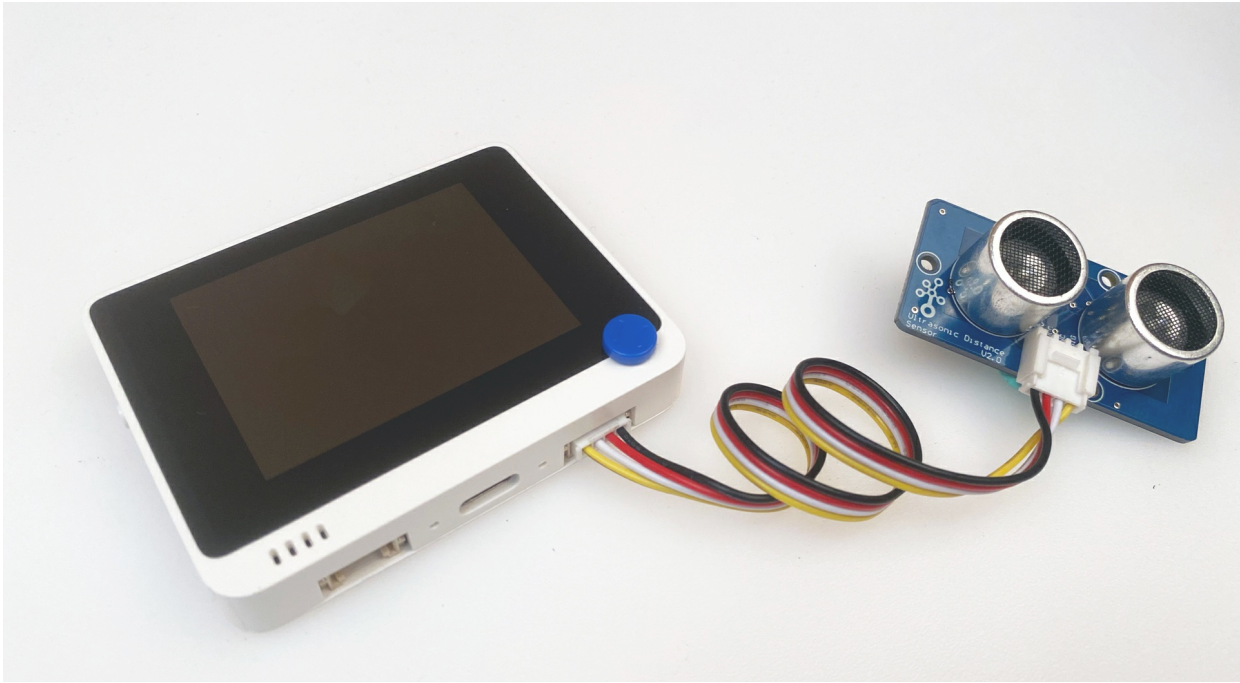
Play With Wio Terminal (ArduPy)

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](#).

```
pip install Seeed-Studio/seeed-ardupy-ultrasonic-sensor
pip build
pip flash
```

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

```

from arduino import grove_ultra_ranger
from machine import LCD
from machine import Sprite
import time

Ultrasonic = grove_ultra_ranger(0)
lcd = LCD()
spr = Sprite(lcd) # Create a buff

def main():
    spr.createSprite(320, 240)
    while True:
        spr.setTextSize(2)
        spr.fillSprite(spr.color.BLACK)
        spr.setTextColor(lcd.color.RED)
        spr.drawString("Ultrasonic Sensor", 55, 10)
        spr.drawFastHLine(40, 35, 240, lcd.color.DARKGREY)
        spr.setTextColor(lcd.color.WHITE)
        spr.drawString("- Centimeters: ", 20, 50)
        spr.drawString("- Inches: ", 20, 80)
        spr.drawNumber(Ultrasonic.cm, 200, 50)
        time.sleep_ms(50) # Needed for data to be read again
        spr.drawNumber(Ultrasonic.inch, 130, 80)
        spr.pushSprite(0, 0)
        time.sleep_ms(500)

        print("The distance to obstacles in front is:", Ultrasonic.cm, 'centimeter')
        print("The distance to obstacles in front is:", Ultrasonic.inch, 'inches')

if __name__ == "__main__":
    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.

```

aip shell -n -c "runfile <YourPythonFilePath>"
# Example:
# 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.

```

ansonhe@Ansons-Macbook-Pro ~: aip shell -n -c "runfile /Users/ansonhe/Desktop/ArduPy-ultrasonic.py"
Positional argument (/dev/cu.usbmodem1413101) takes precedence over --open.
Connected to ardupy
The distance to obstacles in front is: 3 centimeter
The distance to obstacles in front is: 1 inches
The distance to obstacles in front is: 7 centimeter
The distance to obstacles in front is: 2 inches
The distance to obstacles in front is: 6 centimeter
The distance to obstacles in front is: 2 inches
The distance to obstacles in front is: 6 centimeter
The distance to obstacles in front is: 2 inches
The distance to obstacles in front is: 4 centimeter
The distance to obstacles in front is: 1 inches
The distance to obstacles in front is: 7 centimeter
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: $\text{Distance} = \text{echo signal high time} * \text{Sound speed} (340\text{M/S})/2$.

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.

```
#include "Ultrasonic.h"

Ultrasonic ultrasonic1(2);
Ultrasonic ultrasonic2(3);
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  long RangeInCentimeters1;
  long RangeInCentimeters2;

  RangeInCentimeters1 = ultrasonic1.MeasureInCentimeters(); // two measurements
  should keep an interval
  Serial.print(RangeInCentimeters1);//0~400cm
  Serial.println(" cm");

  RangeInCentimeters2 = ultrasonic2.MeasureInCentimeters(); // two measurements
  should keep an interval
  Serial.print(RangeInCentimeters2);//0~400cm
  Serial.println(" cm");

  delay(250);
}
```

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](#)

Tech Support

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

