One Wire Temperature Sensor DS18B20

Get One Now

[https://www.seeedstudio.com/One-Wire-Temperature-Sensor-p-1235.html]

This 2m long One Wire Temperature Sensor has a waterproof probe and long wire shape, suitable for immersive temperature detection. The chip inside this sensor is **DS18B20** which is widely adopted.The original one includes three wires inside, you need to add an extra resistance to get it working. For this sensor, we adjusted it into a Grove port and had a resistance pre-assembled inside so that you can use it as a regular Grove sensor. This article is going to illustrate the usage of the sensor on Raspberry Pi and have a fun!

Features

- Requires only one wire for data interface
- Waterproof
- Grove compatible
- Accepts 3.0V to 5.5V power supply
- Wide temperature range: -55°C to +125°C
- High accuracy: ±0.5°C(-10°C to +85°C)

⚠ **Attention**
The cable part cannot be put under temperature higher than 70°C for a long time.

Specification

Operating Voltage	3.0-5.5V
Chip	DS18B20
Length	2m
Operating Temperature	-55°C to +125°C

Platform Supported

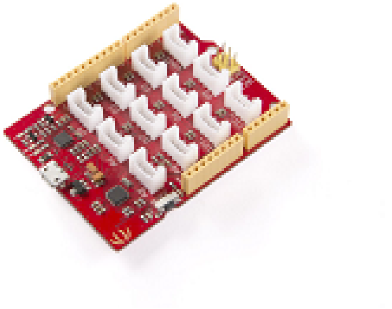
Arduino	Raspberry Pi	

Getting Started

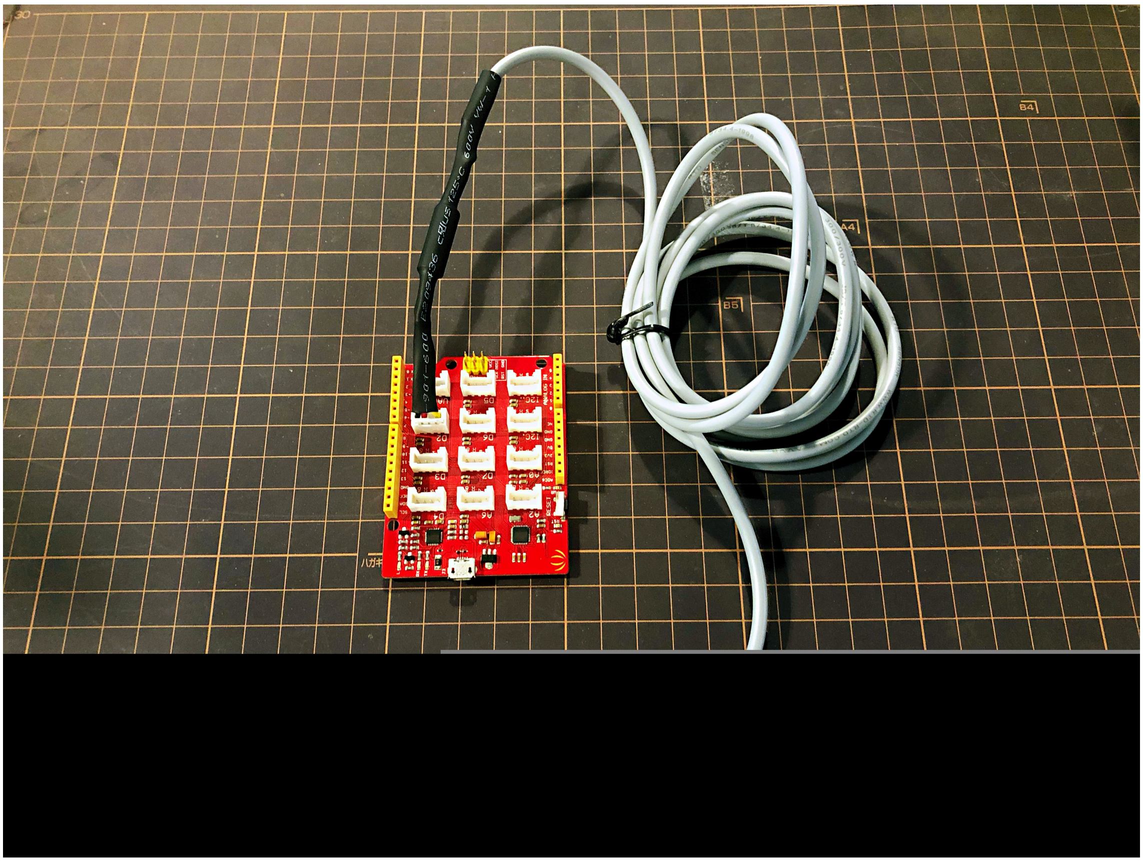
After this section, you can make One Wire Temperature Sensor run with only few steps.

Play with Arduino

Materials Required

Seeeduino Lotus V1.1	One Wire Temperature Sensor
	
Get ONE Now [https://www.seeedstudio.com/Seeeduino-Lotus-V1-1-ATMega328-Board-with-Grove-Interface.html]	Get ONE Now [https://www.seeedstudio.com/One-Wire-Temperature-Sensor-p-1235.html]

Hardware Connection



Tip

Please plug the USB cable， One Wire Temperature Sensor Interface into Seeeduino Lotus V1.1 Interface gently, otherwise you may damage the port.

- **Step 1.** Plug One Wire Temperature Sensor into **D2** interface of Seeeduino Lotus V1.1 with a Grove Cable.
- **Step 2.** Connect Seeeduino Lotus V1.1 to PC via a USB cable.
- **Step 3.** Download the code, please refer to the software part.
- **Step 4.** Run the code and the outcome will display on the screen of **Serial Monitor** in your Arduino IDE .

Software

Attention

If this is the first time you work with Arduino, we strongly recommend you to see Getting Started with Arduino [https://wiki.seeedstudio.com/Getting_Started_with_Arduino/] before the start.

- **Step 1.** Download the Library for Onewire [https://github.com/PaulStoffregen/OneWire/archive/master.zip] and Library for Arduino Temperature Control [https://github.com/milesburton/Arduino-Temperature-Control-Library/archive/master.zip]
- **Step 2.** Copy the whole **OneWire** and **Arduino-Temperature-Control-Library** files and paste them into your Arduino IDE library file.
- **Step 3.** Upload the demo code from **Software Code** below. If you do not know how to upload the code, please check How to upload code [https://wiki.seeedstudio.com/Upload_Code/].

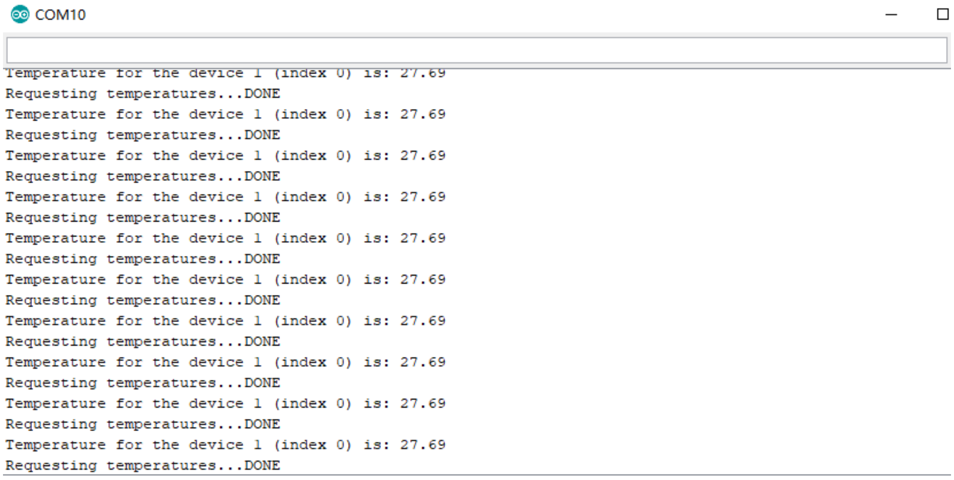
Software Code

```
// Include the libraries we need
#include <OneWire.h>
#include <DallasTemperature.h>

// Data wire is plugged into port 2 on the Arduino
#define ONE_WIRE_BUS 2


// Setup a oneWire instance to communicate with any OneW
OneWire oneWire(ONE_WIRE_BUS);

// Pass our oneWire reference to Dallas Temperature.
DallasTemperature sensors(&oneWire);

/*
```

```
 * The setup function. We only start the sensors here
 */
void setup(void)
{
  // start serial port
  Serial.begin(115200);
  Serial.println("Dallas Temperature IC Control Library

  // Start up the library
  sensors.begin();
}

/*
 * Main function, get and show the temperature
 */
void loop(void)
{
  // call sensors.requestTemperatures() to issue a globa
  // request to all devices on the bus
  Serial.print("Requesting temperatures...");
  sensors.requestTemperatures(); // Send the command to
  Serial.println("DONE");
  // After we got the temperatures, we can print them he
  // We use the function ByIndex, and as an example get
  float tempC = sensors.getTempCByIndex(0);

  // Check if reading was successful
  if(tempC != DEVICE_DISCONNECTED_C)
  {
    Serial.print("Temperature for the device 1 (index 0)
    Serial.println(tempC);
  }
  else
  {
    Serial.println("Error: Could not read temperature da
  }
}
```

Success

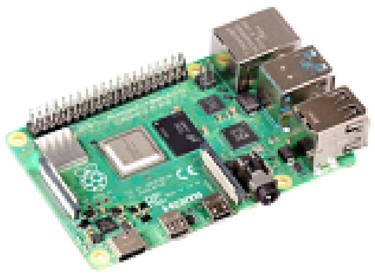
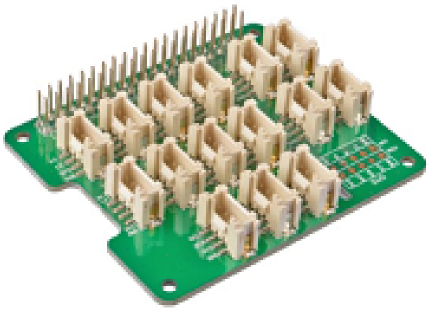
> If everything goes well, you can go to **Serial Monitor** to see an outcome as following:



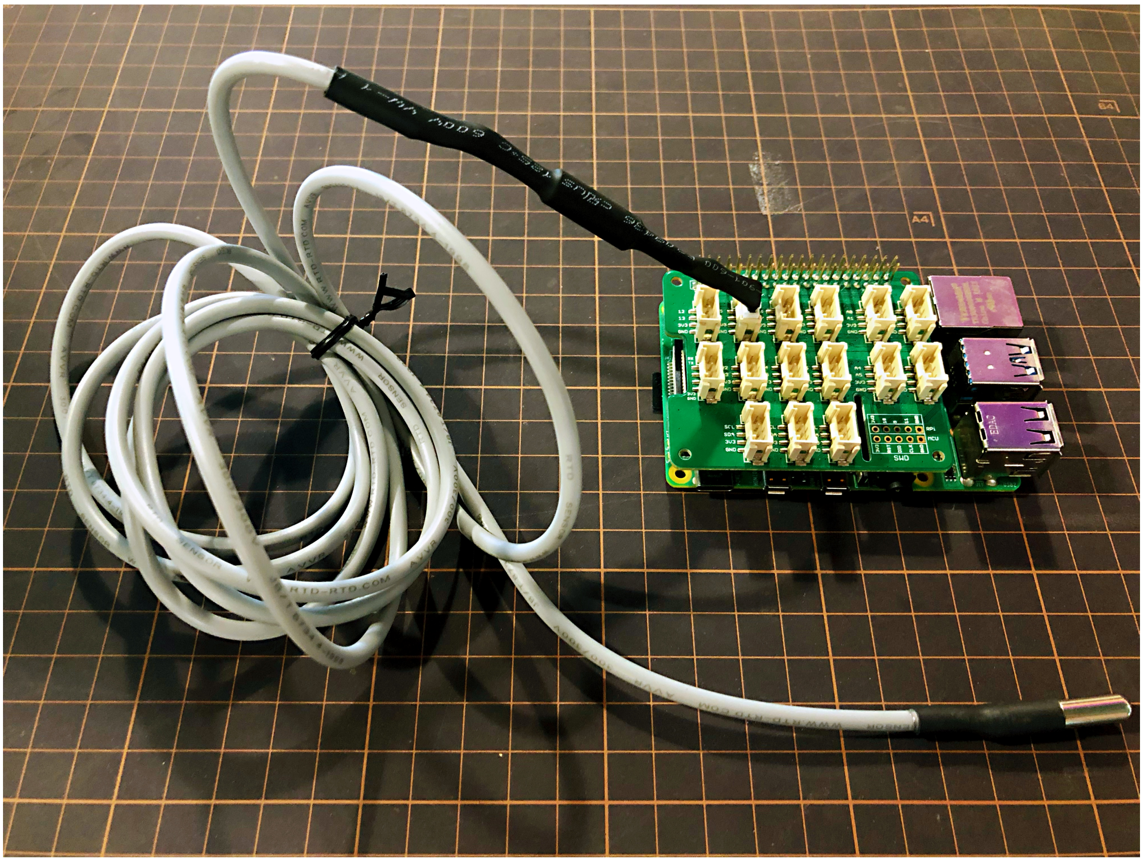
```
Temperature for the device 1 (index 0) is: 27.69
Requesting temperatures...DONE
Temperature for the device 1 (index 0) is: 27.69
Requesting temperatures...DONE
Temperature for the device 1 (index 0) is: 27.69
Requesting temperatures...DONE
Temperature for the device 1 (index 0) is: 27.69
Requesting temperatures...DONE
Temperature for the device 1 (index 0) is: 27.69
Requesting temperatures...DONE
Temperature for the device 1 (index 0) is: 27.69
Requesting temperatures...DONE
Temperature for the device 1 (index 0) is: 27.69
Requesting temperatures...DONE
Temperature for the device 1 (index 0) is: 27.69
Requesting temperatures...DONE
Temperature for the device 1 (index 0) is: 27.69
Requesting temperatures...DONE
Temperature for the device 1 (index 0) is: 27.69
Requesting temperatures...DONE
```

Play with Raspberry Pi

Materials required

Raspberry Pi 4 Model B	Grove - Base Hat for Raspberry Pi
	
Get One Now [https://www.seeedstudio.com/Raspberry-Pi-4-Computer-Model-B-8GB-p-4595.html]	Get One Now [https://www.seeedstudio.com/Grove Base-Hat-for-Raspberry-Pi.html]

Connecting hardware



Step 1. Connect One Wire Temperature Sensor to port **D5** of Grove - Base Hat for Raspberry Pi, Plugged into Raspberry Pi 4 Model B. And then connect the Raspberry Pi 4 Model B with a PC.

Step 2. After accessing the system of Raspberry Pi, git clone Seeed_Python_DS18B20 [https://github.com/Seeed-Studio/Seeed_Python_DS18B20.git] and install **grove.py** by inserting the following command:

```
pip3 install Seeed-grove.py
```

Or on supported GNU/Linux systems like the Raspberry Pi, you can install the driver locally from PyPI:

```
pip3 install seeed-python-Ds18b20
```

Step 3. To install system-wide (this may be required in some cases):

```
sudo pip3 install seeed-python-Ds18b20
```

And you can insert the following command to upgrade the driver locally from PyPI:

```
pip3 install --upgrade seeed-python-Ds18b20
```

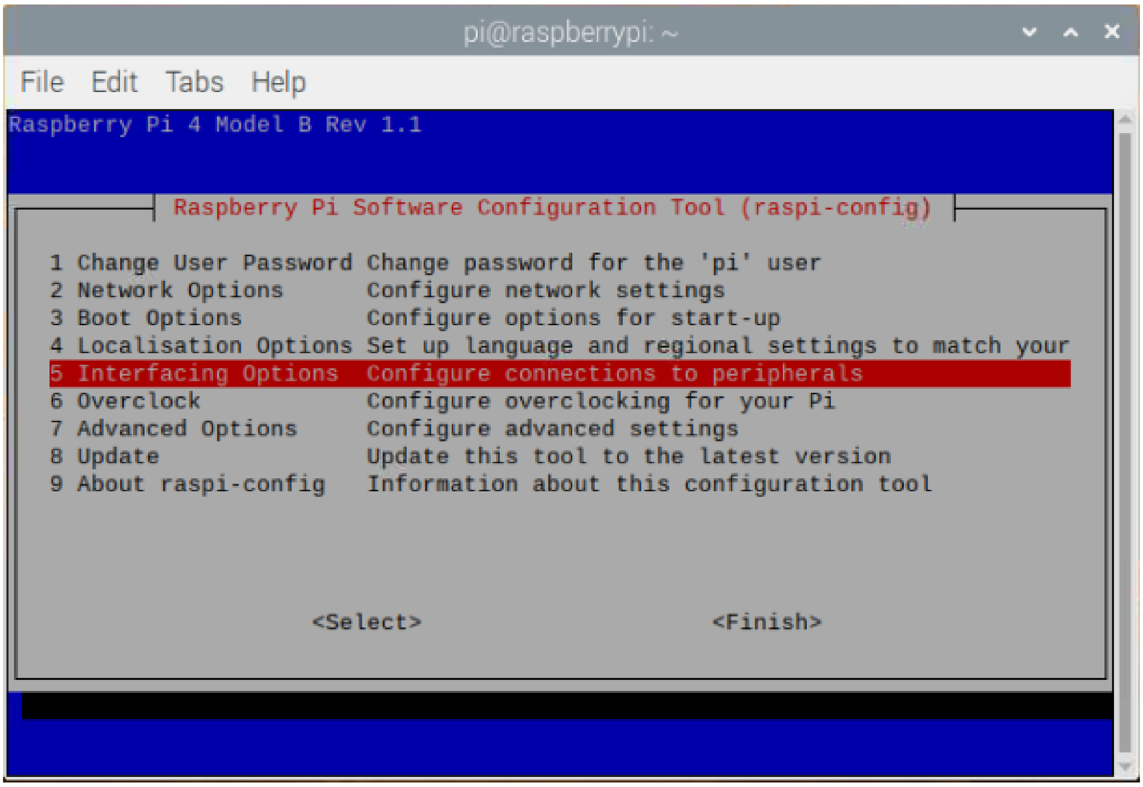
Software

```
import seeed_ds18b20
import time

def main():
    DS18B20 = seeed_ds18b20.grove_ds18b20()
    print("Please use Ctrl C to quit")
    while True:
        temp_c,temp_f = DS18B20.read_temp
        print('temp_c %.2f C    temp_f %.2f F' % (temp_c,
        print('\r', end='')
        time.sleep(0.5)

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

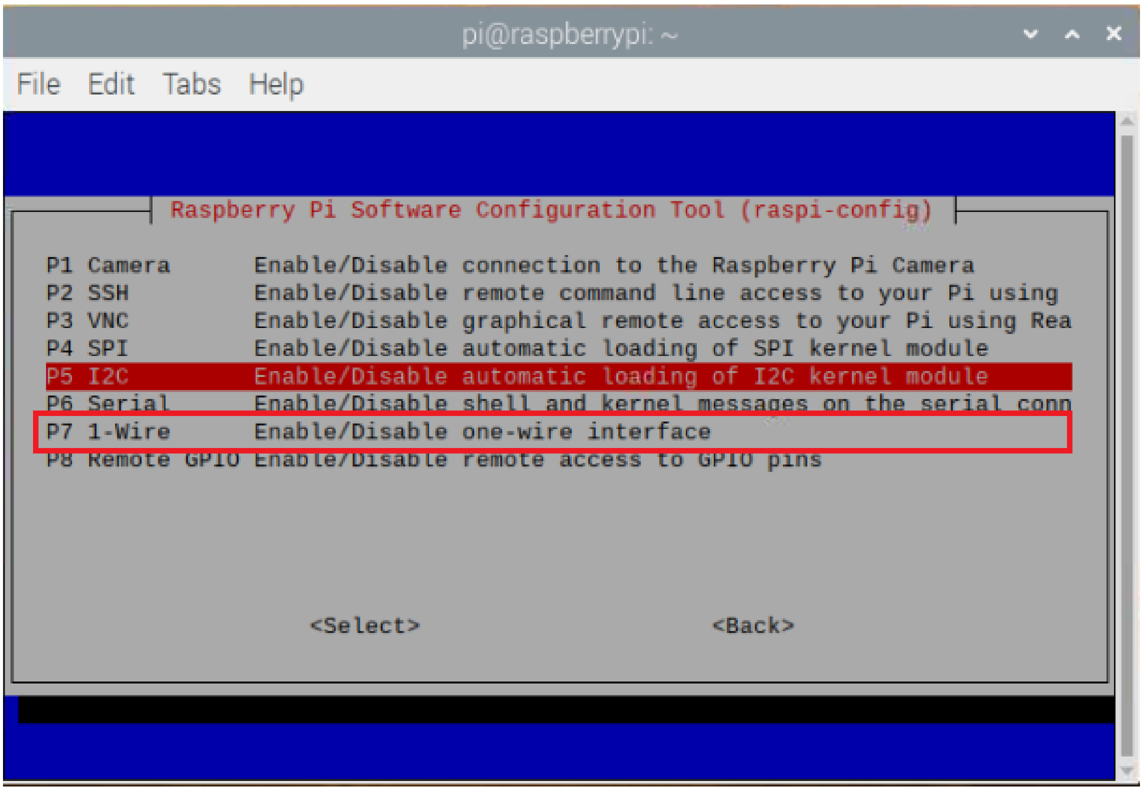
Attention

When running the demo code on Raspberry Pi, you **MIGHT** be reminded to enable the **1-Wire** interface. Then you should use the command **sudo**

raspi-config and reboot to enable the 1-Wire device:



```
Raspberry Pi 4 Model B Rev 1.1

         Raspberry Pi Software Configuration Tool (raspi-config)

 1 Change User Password Change password for the 'pi' user
 2 Network Options      Configure network settings
 3 Boot Options         Configure options for start-up
 4 Localisation Options Set up language and regional settings to match your
 5 Interfacing Options  Configure connections to peripherals
 6 Overclock            Configure overclocking for your Pi
 7 Advanced Options     Configure advanced settings
 8 Update               Update this tool to the latest version
 9 About raspi-config   Information about this configuration tool

                  <Select>                     <Finish>
```



```
         Raspberry Pi Software Configuration Tool (raspi-config)

 P1 Camera      Enable/Disable connection to the Raspberry Pi Camera
 P2 SSH         Enable/Disable remote command line access to your Pi using
 P3 VNC         Enable/Disable graphical remote access to your Pi using Rea
 P4 SPI         Enable/Disable automatic loading of SPI kernel module
 P5 I2C         Enable/Disable automatic loading of I2C kernel module
 P6 Serial      Enable/Disable shell and kernel messages on the serial conn
 P7 1-Wire      Enable/Disable one-wire interface
 P8 Remote GPIO Enable/Disable remote access to GPIO pins

                  <Select>                     <Back>
```

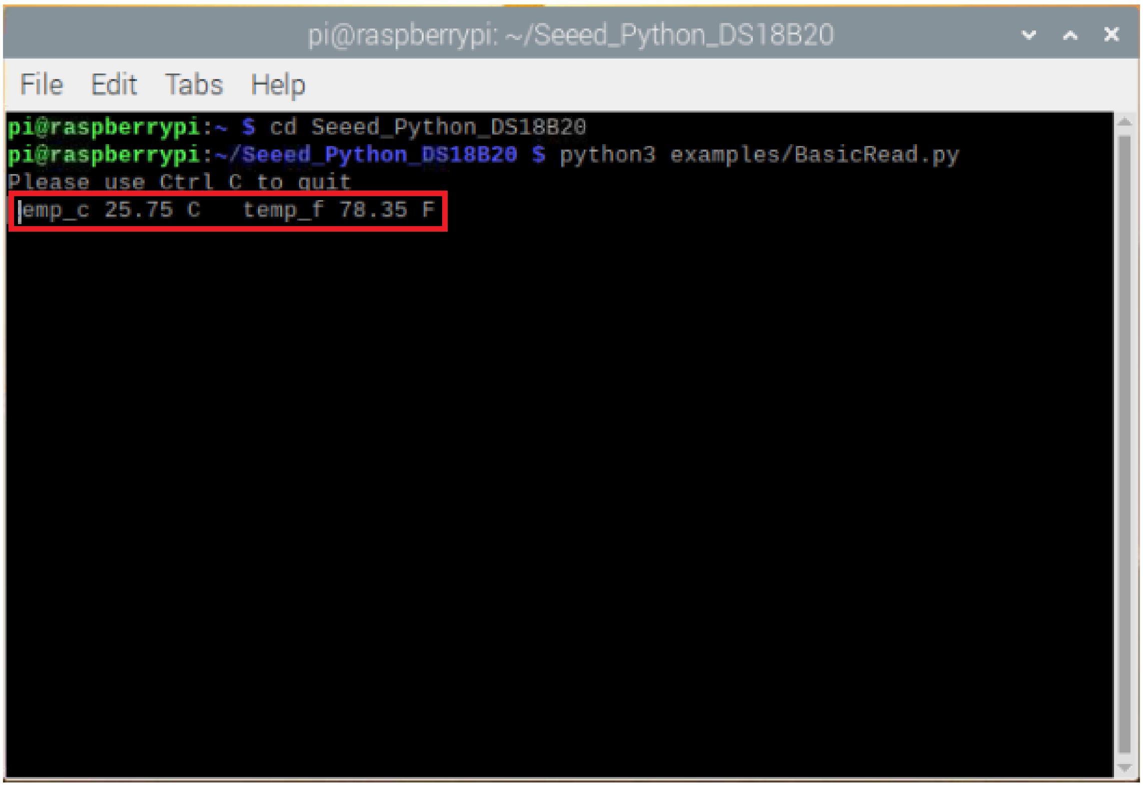
Step 4. Run the demo by the following command:

```
cd Seeed_Python_DS18B20
```

```
python3 examples/BasicRead.py
```

Success

The outcome will display as following if everything goes well:



```
pi@raspberrypi:~ $ cd Seeed_Python_DS18B20
pi@raspberrypi:~/Seeed_Python_DS18B20 $ python3 examples/BasicRead.py
Please use Ctrl C to quit
temp_c 25.75 C    temp_f 78.35 F
```

Resource

[PDF] DS18B20-Datasheet [https://files.seeedstudio.com/wiki/One-Wire-Temperature/res/DS18B20-Datasheet.pdf].

Tech Support

Please submit any technical issue into our forum [https://forum.seeedstudio.com/].

[https://www.seeedstudio.com/act-4.html?utm_source=wiki&utm_medium=wikibanner&utm_campaign=newproducts]