

MIT AI2 204

IoT with MIT App Inventor

Fundamental

X. Tang

IoT with MIT App Inventor

Start 7:05PM

**Please update your name with studentID
In Zoom meeting**

Sending Temperature and Humidity To your android device

Description: In this project, a temperature and humidity sensor module is connected to your Raspberry Pi. The Raspberry Pi sends the ambient temperature and humidity measurements every 10 seconds to the Android mobile phone where they are displayed on the screen.

Sending Temperature and Humidity To your android device

Hygrothermograph is an important tool in our lives to give us data on the temperature and humidity in our environment. In this project, we will use the RPi to read Temperature and Humidity data of the DHT11 Module.

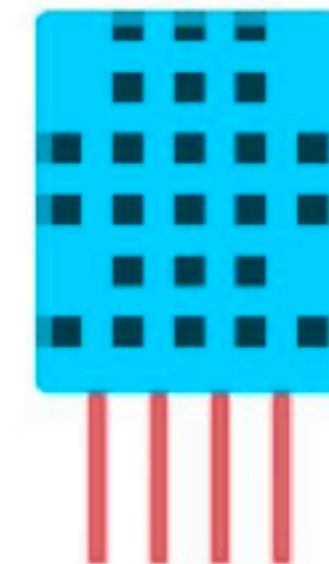
Component List

Raspberry Pi (with 40 GPIO) x1
GPIO Expansion Board & Ribbon Cable x1
Breadboard x1

Jumper Wire x4



DHT11 x1

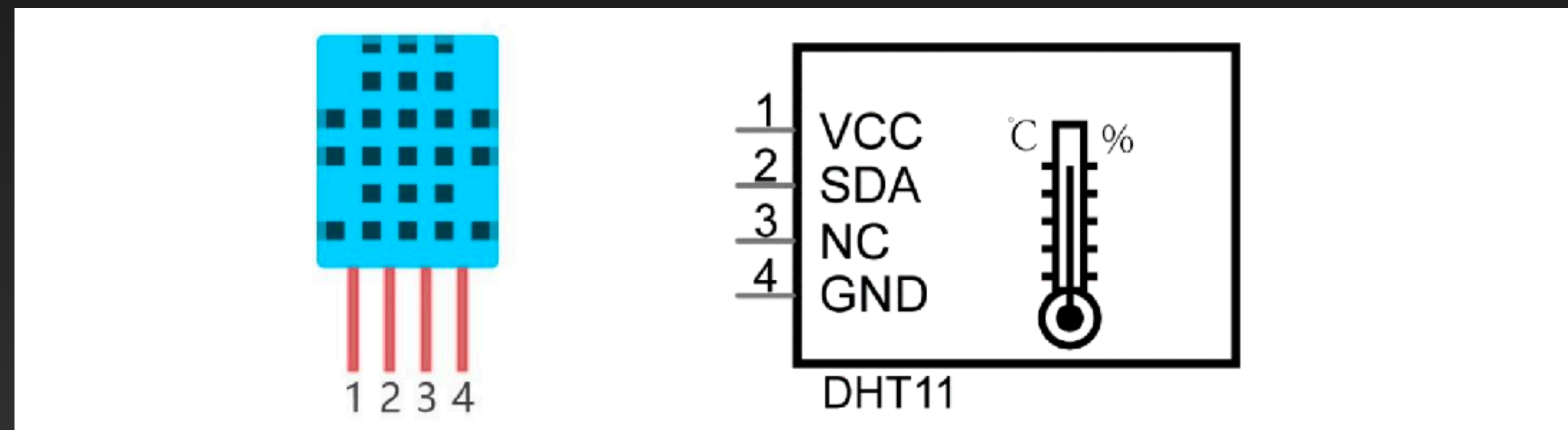


Resistor 10kΩ x1



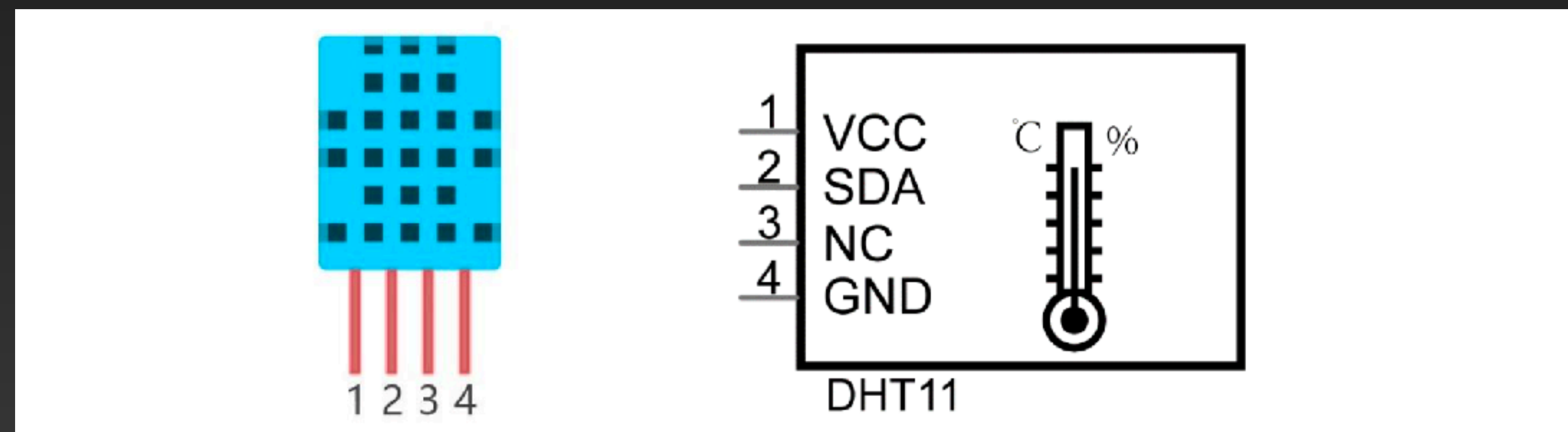
Sending Temperature and Humidity To your android device

The Temperature & Humidity Sensor DHT11 is a compound temperature & humidity sensor, and the output digital signal has been calibrated by its manufacturer.



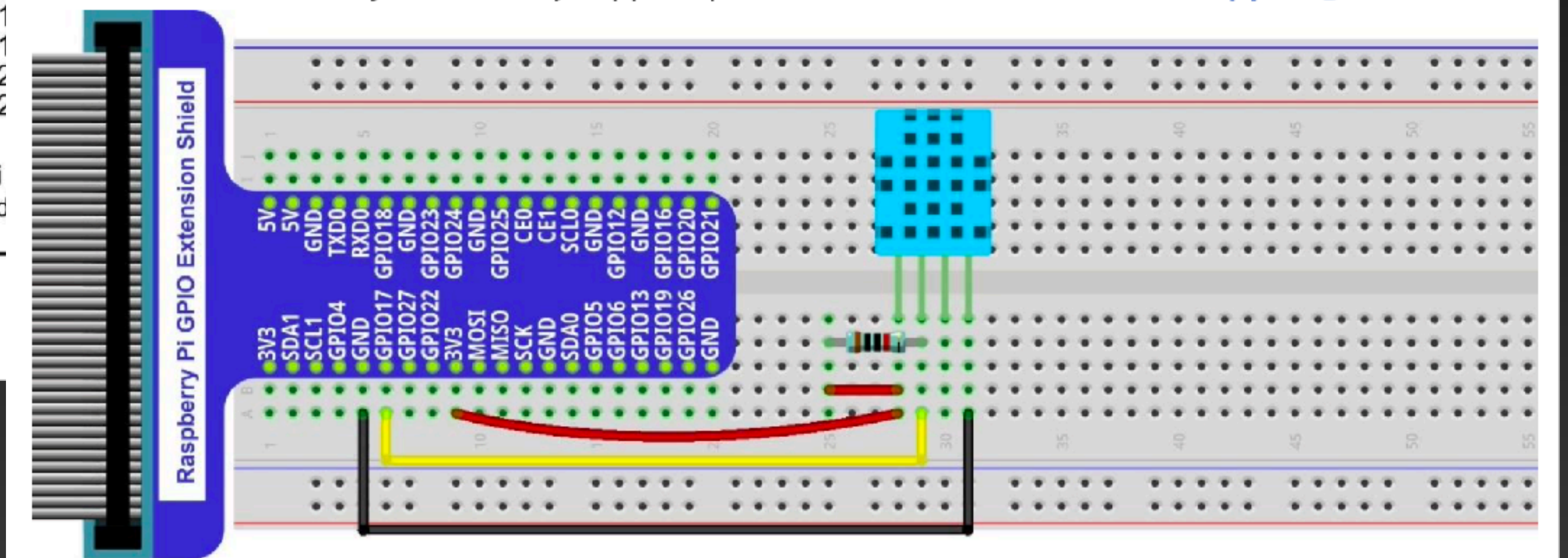
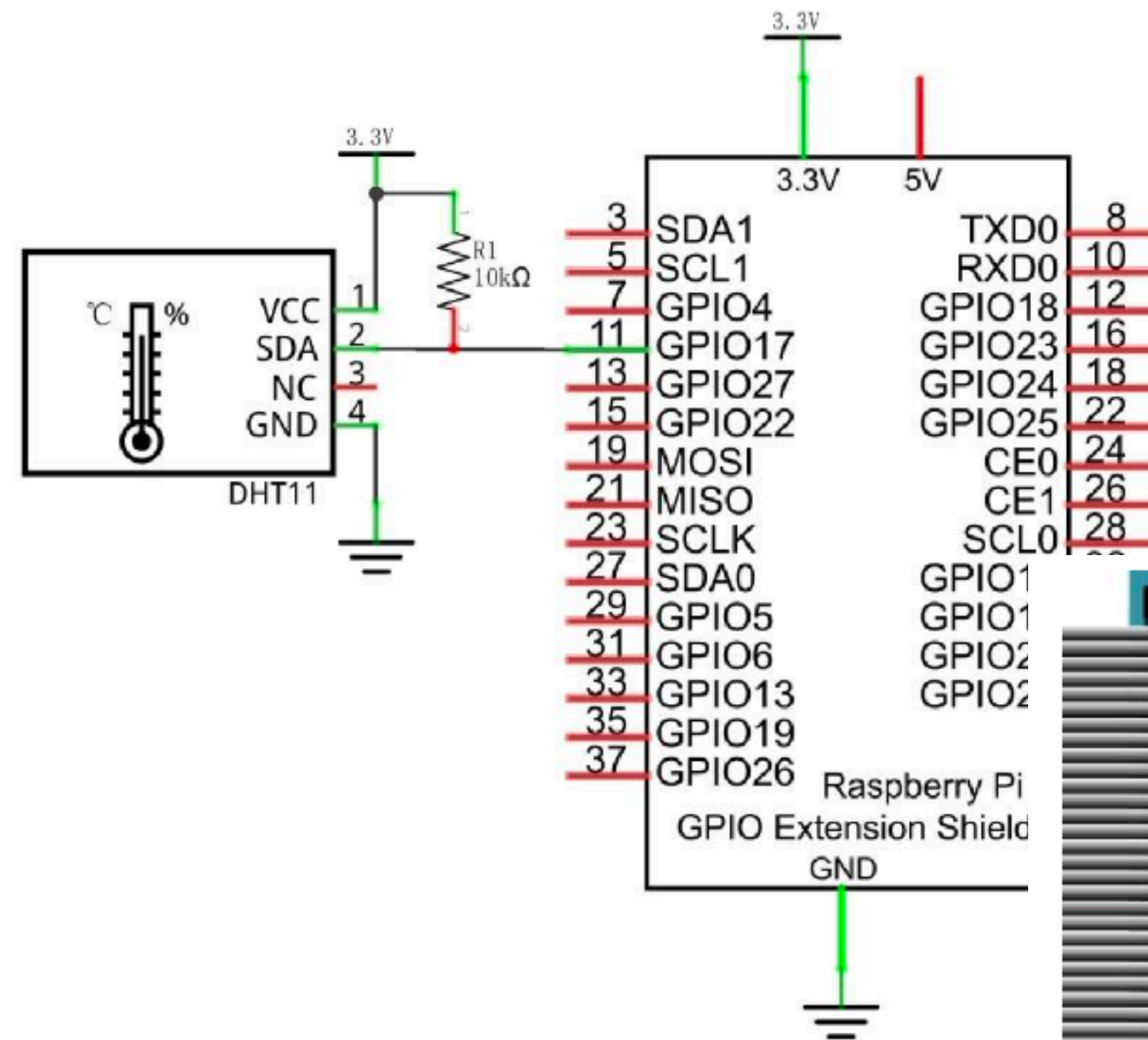
Sending Temperature and Humidity To your android device

After being powered up, it will initialize in 1 second. Its operating voltage is within the range of 3.3V-5.5V. The SDA pin is a data pin, which is used to communicate with other devices. The NC pin (Not Connected Pin) are a type of pin found on various integrated circuit packages. Those pins have no functional purpose to the outside circuit (but may have an unknown functionality during manufacture and test). Those pins **should not be connected** to any of the circuit connections.



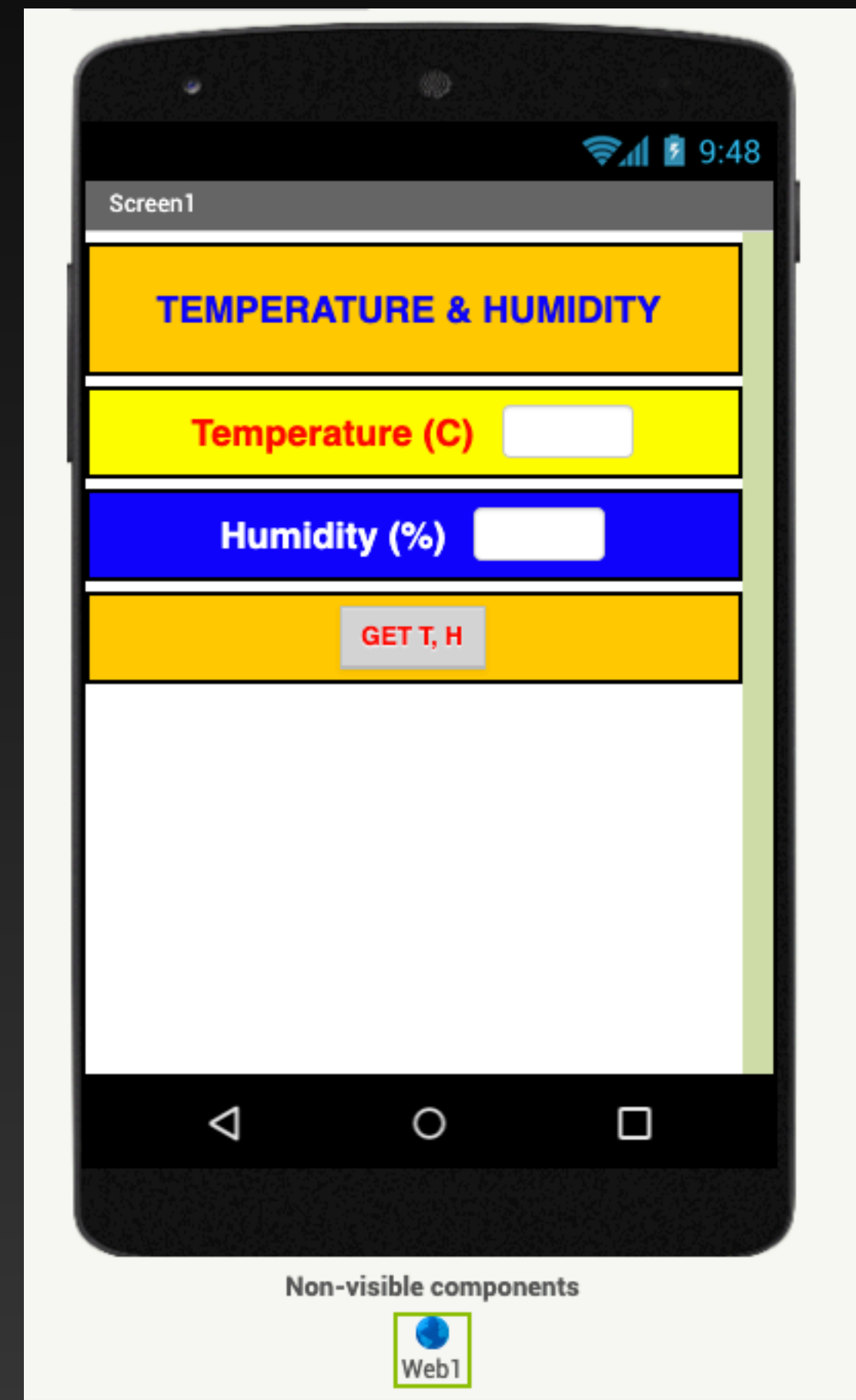
Sending Temperature and Humidity To your android device

Schematic diagram



Sending Temperature and Humidity To your android device

Build your MIT App



```
initialize global TH to ""
initialize global RaspberryPi to "http://192.168.1.xxx"

when Button1.Click
do
  set Web1.Url to get global RaspberryPi
  call Web1.Get

when Web1.GotText
  url responseCode responseType responseContent
do
  set global TH to get responseContent
  set TextBox1.Text to segment text get global TH
    start 1
    length 2
  set TextBox2.Text to segment text get global TH
    start 4
    length 2
```


Sending Temperature and Humidity To your android device

Prepare the environment:(pi4)

```
sudo pip3 install adafruit-circuitpython-dht
```

There is new updates to the library:

https://github.com/adafruit/Adafruit_CircuitPython_DHT

Sending Temperature and Humidity To your android device

Prepare the environment:(pi5)

```
mkdir dht_test
```

```
cd dht_test
```

```
python3 -m venv myenv
```

```
source myenv/bin/activate
```

```
python3 -m pip install adafruit-circuitpython-dht
```

```
python3 -m pip install flask
```

——-Find the python3 location and run with sudo command——-

```
sudo /home/pi/DHT11/myenv/bin/python3 DHT11Test.py
```

Sending Temperature and Humidity To your android device

Write the python's
control program

```
9 import time
10 from flask import Flask, render_template
11 import RPi.GPIO as GPIO
12 import adafruit_dht
13 from board import *
14
15 # GPIO17
16 SENSOR_PIN = D17
17
18 app = Flask(__name__)
19
20 @app.route('/', methods=['GET', 'POST'])
21 def get_data():
22
23     dht11 = adafruit_dht.DHT11(SENSOR_PIN, use_pulseio=False)
24     temp = dht11.temperature
25     hum = dht11.humidity
26     tempint = int(temp)
27     humint = int(hum)
28     if tempint < 10:
29         datat = "0" + str(tempint)
30     else:
31         datat = str(tempint)
32     datath = datat + "," + str(humint)
33     return(datath)
34
35 if __name__ == '__main__':
36     app.run(debug=True, port=80, host='0.0.0.0', use_reloader=False)
37
38
```

Prepare the Env and Test the Sensor

```
git clone --depth 1 https://github.com/freenove/  
Freenove\_Ultimate\_Starter\_Kit\_for\_Raspberry\_Pi
```

```
mv Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi ~/Freenove_Kit/
```

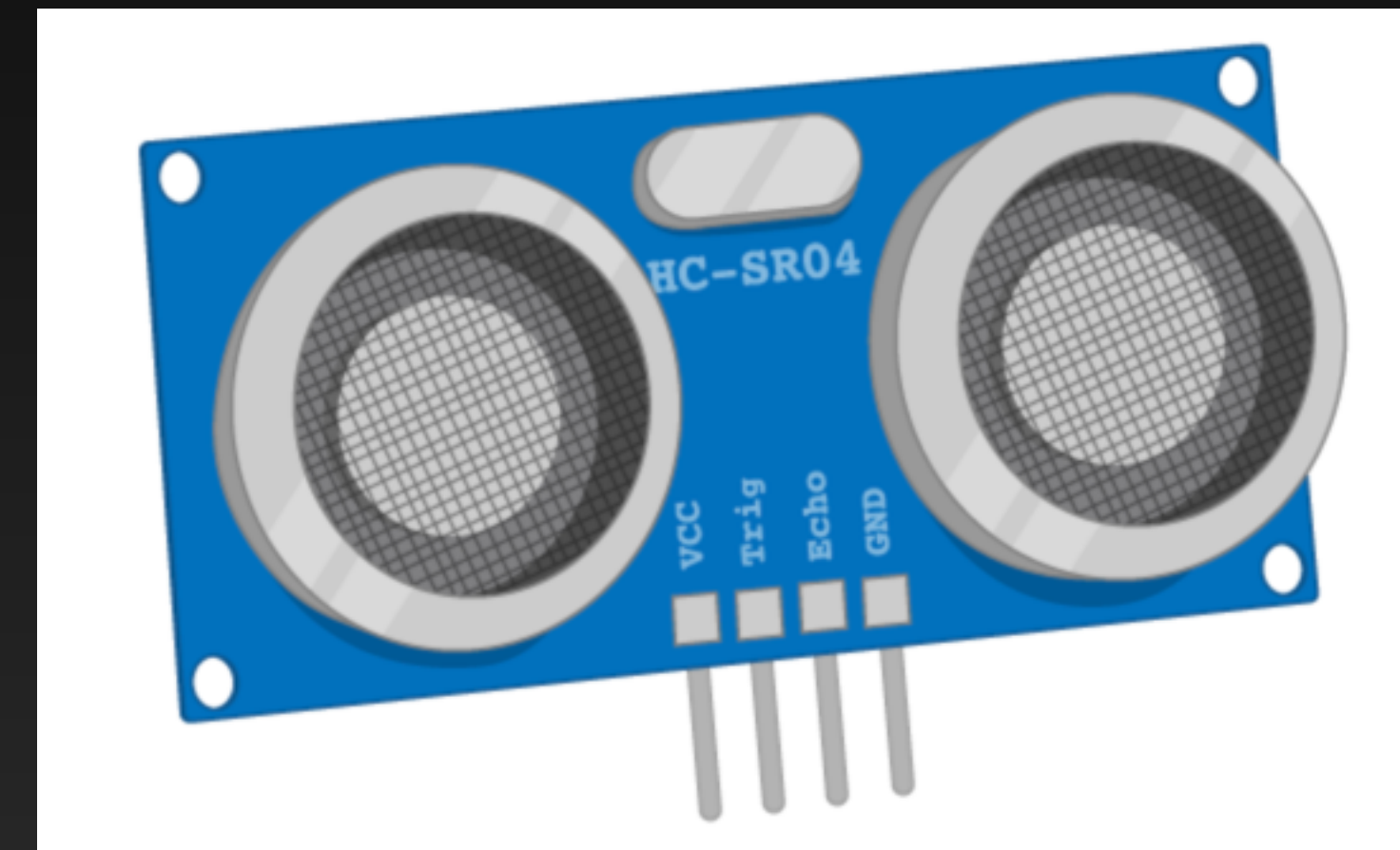
```
cd Freenove_Kit/Code/Python_GPIOZero_Code/21.1.1_DHT11/
```

```
python DHT11.py
```

Setup Distance Sensor

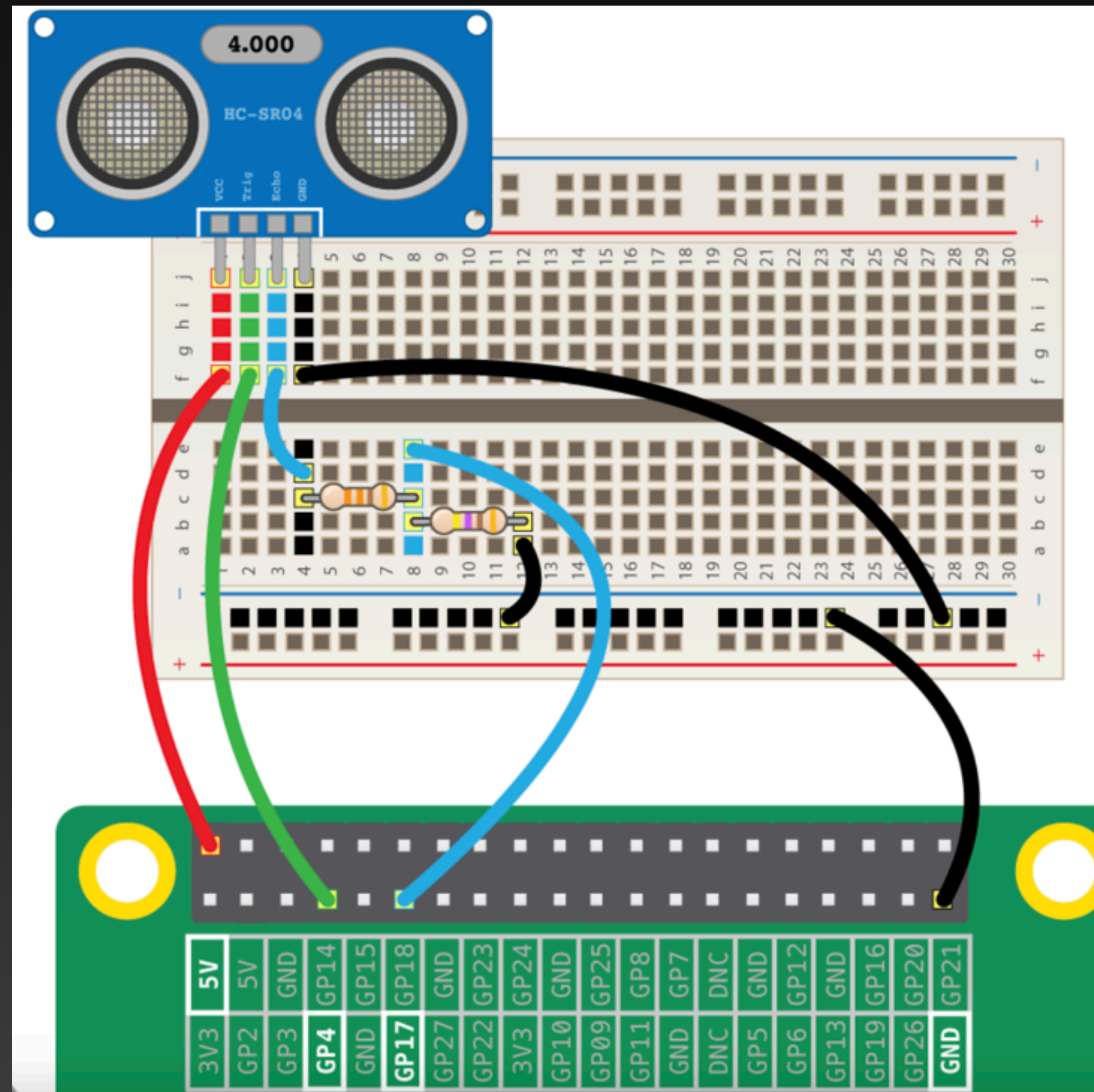
An ultrasonic distance sensor has four pins. They are called Ground (Gnd), Trigger (Trig), Echo (Echo), and Power (Vcc).

To use an ultrasonic distance sensor, you need to connect the Gnd pin to the ground pin on the Raspberry Pi, the Trig and Echo pins to GPIO pins on the Raspberry Pi, and the Vcc pin to the 3V3 pin on the Raspberry Pi.



Setup Distance Sensor

The circuit connects to two GPIO pins (one for echo, one for trigger), the ground pin, and a 5V pin. You'll need to use a pair of resistors (220Ω) as a potential divider:



Setup Distance Sensor

```
from gpiozero import DistanceSensor
ultrasonic = DistanceSensor(echo=17, trigger=4)
while True:
    print(ultrasonic.distance)
```