# Overview

A temperature sensor is a component that converts temperature readings into output signals. By material and component features, temperature sensors can be divided into two types: thermal resistor and thermocouple. For this lab, we will be working with the former. Essentially, the thermistor is a resistor whose resistance decreases with rising temperature. Using Python, we can periodically sample the resistance and convert it back into a temperature reading. Since their resistance changes acutely with temperature changes, thermistors are the most sensitive temperature sensors.

# Step 1: Connect It

Use the following wiring diagram to connect the sensor to your Raspberry Pi



**Important Notes:**

* The wire colors do not matter. Just use whatever is available at your station.
* **Make sure that you are plugging the wires into the correct location on both the Pi and the Sensor (especially the power and ground)**
* The wires move around a lot, so make sure they are firmly in place when you are testing

# Step 2: Code It

We have provided you with a Python code template that can run this sensor/device. The code is located at:

**/home/pi/Documents/rpi-iot-demos/temperature.py**

Use the Geany editor to open this file (NOTE: this will be slow, so be patient). Once it is open, take some time to read the code to see how it works.

Once you are ready, do the following:

1. Find the function called loop().
   1. This function starts when the script runs, and runs forever
2. Modify this function so that it gets the temperature (using the read() function), converts it to Fahrenheit, and prints it to the console.

def loop():

while True:

temperature\_in\_celsius = read()

temperature\_in\_fahrenheit = temperature\_in\_celsius \* (9/5) + 32

if temperature\_in\_celsius != None:

print ("Current temperature : %0.3f F" % temperature\_in\_fahrenheit)

# Step 3: Run / Test It

1. To run the script, open a Linux console and navigate to the folder with your code
   1. Helpful Linux Commands:  
      **ls** lists the contents of the directory  
      **cd <folder\_name>** opens a folder (don’t type the < > characters)  
      **cd ..** exits the folder you are currently in  
      **cd ~** takes you back to your home folder
2. Run the python program by typing the following:

**python3 temperature.py**

1. Try heating up the temperature sensor to see what happens!
   1. If it doesn’t work, make sure that your wiring is correct and that you see lights on the sensor / device
   2. Can you modify the program so that it only prints once the temperature goes above/below some value?
2. **To exit the program, press Ctrl-C in the terminal**

# Step 4: Make It Control Something

To make your program send MQTT messages, modify your main program to look like the following:

# --------------------------------------------

# Main Program Starts Here

# --------------------------------------------

if \_\_name\_\_ == '\_\_main\_\_':

comm.connect()

setup()

try:

setup()

print("Temperature sensor started.")

loop()

except KeyboardInterrupt:

destroy()

Now, go to the loop() function you modified earlier. You can use the comm.send function to send a message when the temperature is above/below a certain value, etc.

comm.send("CHANNEL NAME GOES HERE", "MESSAGE GOES HERE")