YouTube

Workshop Kit v2.0
Tutorial 7/9: Temperature Sensor

Contents

Things you will need

Raspberry Pi 3 Model B
Class 10 Micro SD Card
Keyboard + Mouse
Monitor + HDMI Cable
Power Supply (Recommended: 5V 2.5A)
Breadboard
1x Red LED
1x Blue LED
2x 330Ω Resistor
5x M/M Jumper Wire
7x M/F Jumper Wire
1x Button
1x Buzzer
1x DS18B20 Temperature Sensor
1x 4k7Ω Resistor

If you are connecting to your Raspberry Pi remotely using VNC or other means then Keyboard, mouse Monitor and HDMI cable are optional.

Prerequisites

You will need to install the latest version of Raspbian on to your Micro SD Card. Initial setup will require a keyboard, mouse, HDMI cable and Monitor/TV.

Wiring Diagram
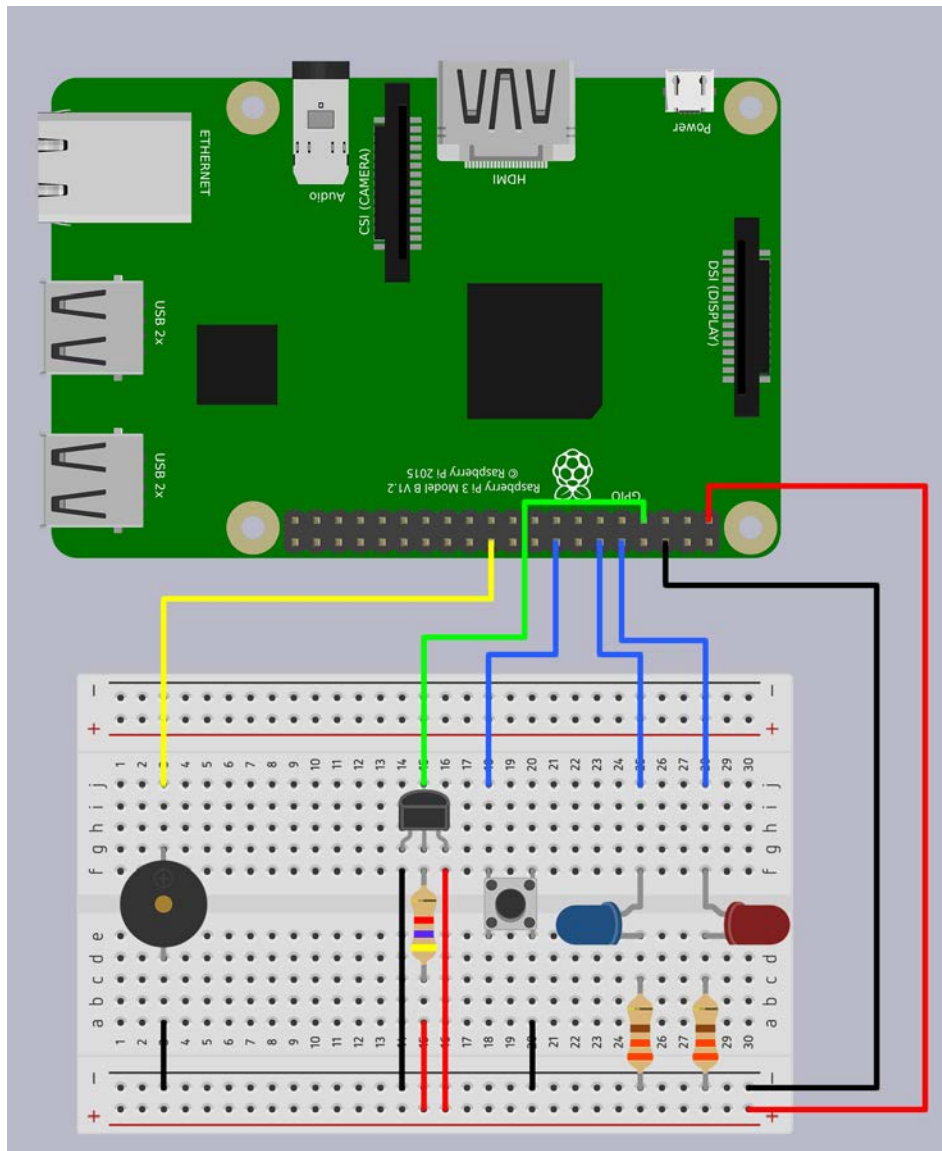
Introduction

In this tutorial we will be reading the temperature from the temperature sensor and displaying the temperature in Celsius and Fahrenheit the console. This tutorial is a little different from the others as we are not communicating with the GPIO directly. There will be additional configuration for the Raspberry Pi to be done will be explained in this tutorial.

These tutorials are written on a Raspberry Pi 3 Model B with a clean install of Rasbian 4.4.

Some parts of the tutorials may look familiar as the code examples are written in a way that there is very little work needed and minor modifications to the previous code to get you up and running faster.

Getting Started

Additional Setup

Raspi-config

Before we can get to the coding, there is 1 more step that is needed, we need to activate an interface and modify the boot configuration file.

Open a terminal by clicking this icon in the task bar and type sudo raspi-config and press enter. This will load the Raspberry Pi software Configuration Tool. Use the arrow down key and select Interfacing Options and press enter. Next select I2C and press enter twice.

To close press right arrow twice and press enter.
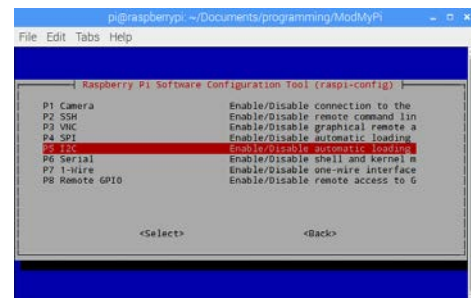
Boot Config

Last additional setup to do is to add an entry to the boot config.txt file. In terminal type sudo nano /boot/config.txt then press enter. Press the Page Down key till you reach the bottom of the file and enter dtoverlay=w1-gpio. Press Ctrl+O then enter to save, then press Ctrl+X to exit.
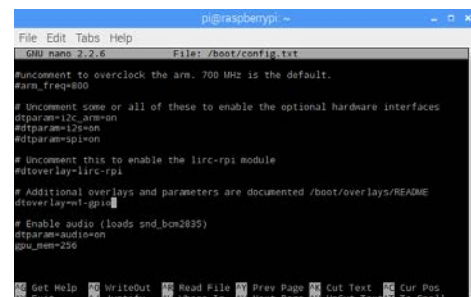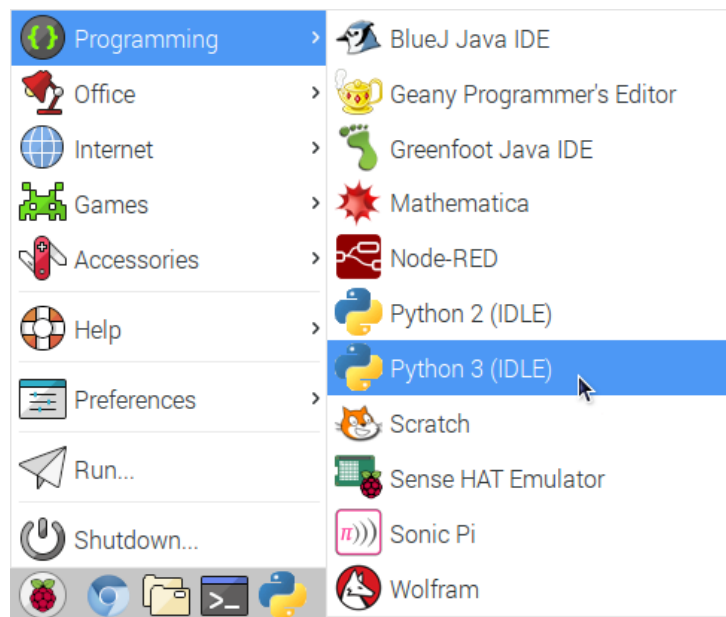

Reboot

You will need to reboot the Raspberry Pi for the changes to work. Since you have the terminal open, simply type sudo reboot and press enter

Load IDLE and Save

To get started, first you need to open Python 3 (IDLE). To do this click on the Raspberry Pi icon on the task bar, highlight "Programming" then click on "Python 3 (IDLE)"
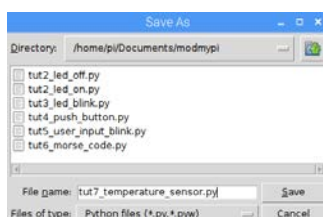

When IDLE has loaded, you will want to start working on a new file. You can do this by clicking on File and select "New File" or by pressing Ctrl+N
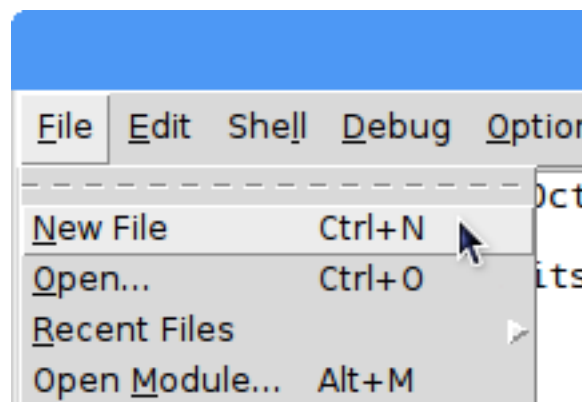
Save the Program

Now that we have IDLE running, first save a new file. First open up the File Manager by clicking on this icon on the taskbar and open the Documents folder.

Go back to the IDLE and click on File and select "Save As". Navigate to /home/pi/Documents/modmypi and enter tut7_temperature_sensor.py for the filename then click Save.

Writing the Code

The first thing you should type is a shebang line, docstring and import your modules

```python
#!/usr/bin/python3
# -*- coding: utf-8 -*-
'''
Display the temperature in Celsius and Fahrenheit
'''

# Builtin Python Libraries
import os
from glob import glob
```

The added 2nd line is a compatibility addition for if you run the code in python 2.7.9 as it does not support some character (in this case it is the ° symbol).

Display Variables

Now you need to create some variables, 2 string variables and 1 integer variables. The string variables will be used to display text in the terminal to let us know what the program is doing. The integer variables will be used for a sleep timer.

```python
# Display Variables
HEADER = 'Temperature Sensor Test (Ctrl+C to Quit)\n\n'
DISPLAY = '{h}Loop Count: {lc}/{tl}'

# Program Variables
BASE_LOC = '/sys/bus/w1/devices/'
DEVICE_LOC = '{}/w1_slave'.format(glob(BASE_LOC + '28*')[0])
```

Update Display Method

Next, it is time to create a Method that will update the text that will be displayed in the console. This will be called when you want to update the information in the console.

```python
def update_display(celsius, fahrenheit):
    ''' Updates the text displayed in the console '''

    # Clear the console
    os.system('clear')
    # Print the formatted text to the console
    print(DISPLAY.format(h=HEADER, c=celsius, f=fahrenheit))
```

Get Temperature Method

This method reads a file an extracts the temperature date from it and returns the value.

```python
def get_temp():
    '''reads a specific file from the system and extracts the temperature data'''
    read_file = open(DEVICE_LOC, 'r')
    tmp_txt = read_file.readlines()
    read_file.close()
    return round(float(tmp_txt[1].split(' ')[9][2:-1])/1000, 1)
```
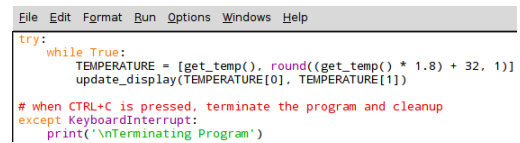
The Action Code – Genrate Temperature Variables

There is not much code here, all this will do is get the temperature data, converts Celsius to Fahrenheit and updates the text in the console.


```python
try:
    while True:
        TEMPERATURE = [get_temp(), round((get_temp() * 1.8) + 32, 1)]
        update_display(TEMPERATURE[0], TEMPERATURE[1])

# when CTRL+C is pressed, terminate the program and cleanup
except KeyboardInterrupt:
    print('\nTerminating Program')
```

Make sure to save your work by clicking File and select Save, or press Ctrl+S

Running the Program

Save your work and it is time to run it so that you can make sure that it works as it should. Go back to the File Manager and open the modmypi folder you created. Next click on tools and select "Open Current Folder in Terminal" or press F4.

In the terminal, type
python3 tut7_temperature_sensor.py
and press enter
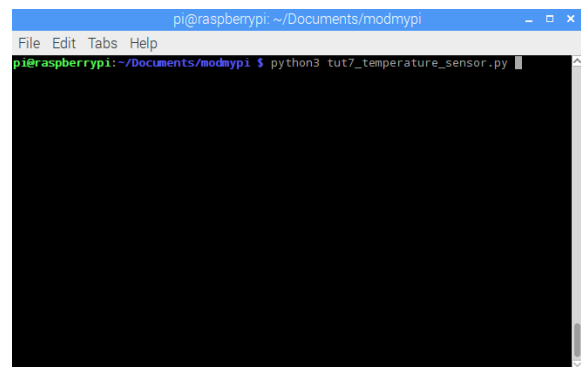
Results

If everything is working correctly you should see in the console showing you the current temperature in Celsius and Fahrenhiet. If you place your finger on the temperature sensor component you will start to see the temperature increase and removing your finger will let the temperature normalize to the room temperature.

In the console you should see something similar to this when running:
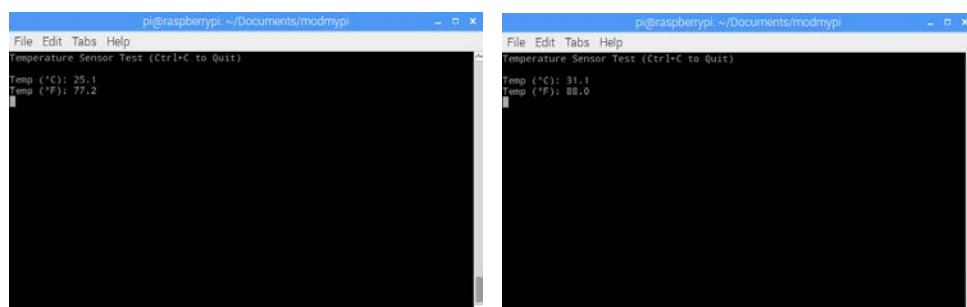
Code on GitHub

If you would like to download a copy of the code, you can download it from along with all the other tutorials, code and wiring diagrams from [GitHub here](#)

Thanks

Thank you for taking the time to follow this tutorial and hope that you have found this useful. Please feel free to follow the other tutorials that have been created for the ModMyPi YouTube Workshop Kit.