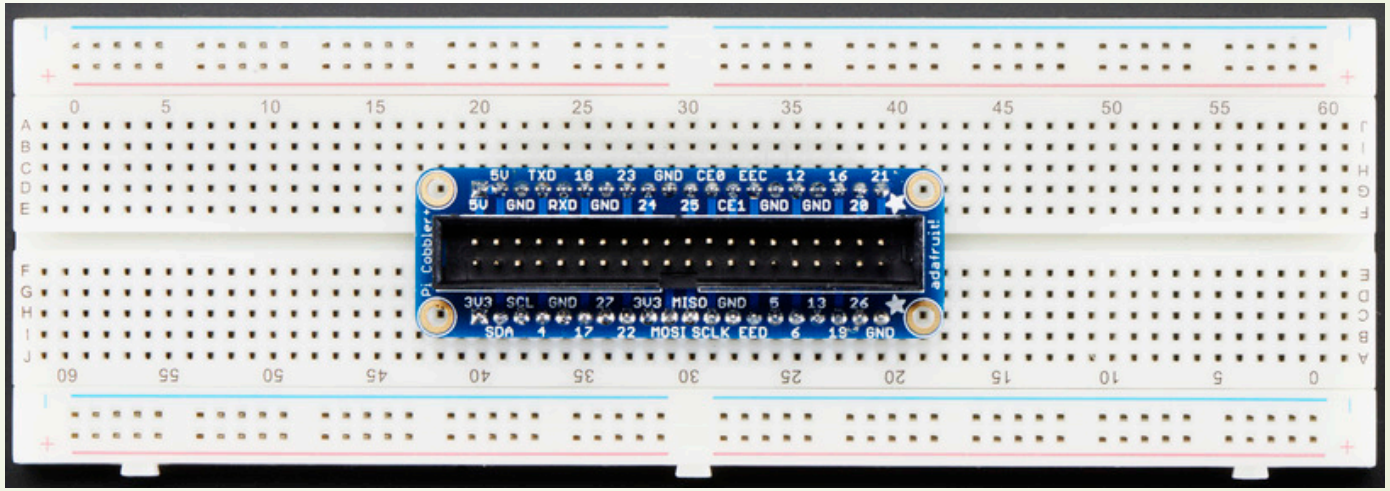
5. General Setup

Attaching the Adafruit Assembled Pi Cobbler Plus

- When doing any connections or alterations to the physical hardware it is **ALWAYS** necessary to shutdown the RPi and disconnect it from the power source. To do this simply enter the command

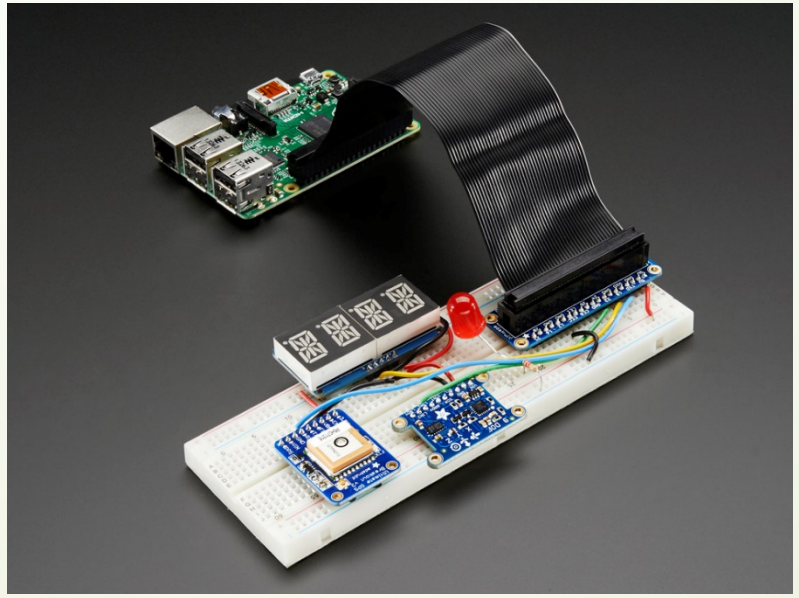
  sudo shutdown –h now

- Attach the *Adafruit Assembled Pi Cobbler Plus* to the breadboard in the orientation show below:



Attaching 40-pin GPIO Cable

- In this step it is very important to attach the *40-pin GPIO Cable* in the right orientation or risk shorting (breaking) the RPi
- While keeping the breadboard and pi cobbler connection in the same orientation as the previous step, put the RPi below it with the four USB connections facing to the right
- Take the *40-pin GPIO Cable* and with the white edge point to the left, attach it to the *Adafruit Assembled Pi Cobbler Plus* and the RPi



Programing On The Raspberry Pi

- To write a program on the RPi, type in the following code (Filename can be whatever you like):

  nano Filename.py

- To run a program on RPi simply type the following code (Filename is the same one as you chose above):

  sudo python Filename.py

- To remove an unwanted program on RPi type the following command:

  sudo rm Filename.py

Before Starting The Projects

- For the following projects we will be using Python packages that do not come pre-installed on the Raspberry Pi. Therefore we must download the necessary packages
- To do so, type the following commands in order:

```
sudo apt-get install python-pip
sudo pip install RPi.GPIO
sudo pip install requests

sudo apt-get update
sudo apt-get install git build-essential python-dev python-smbus
git clone https://github.com/adafruit/Adafruit_Python_BMP.git
cd Adafruit_Python_BMP
sudo python setup.py install
```

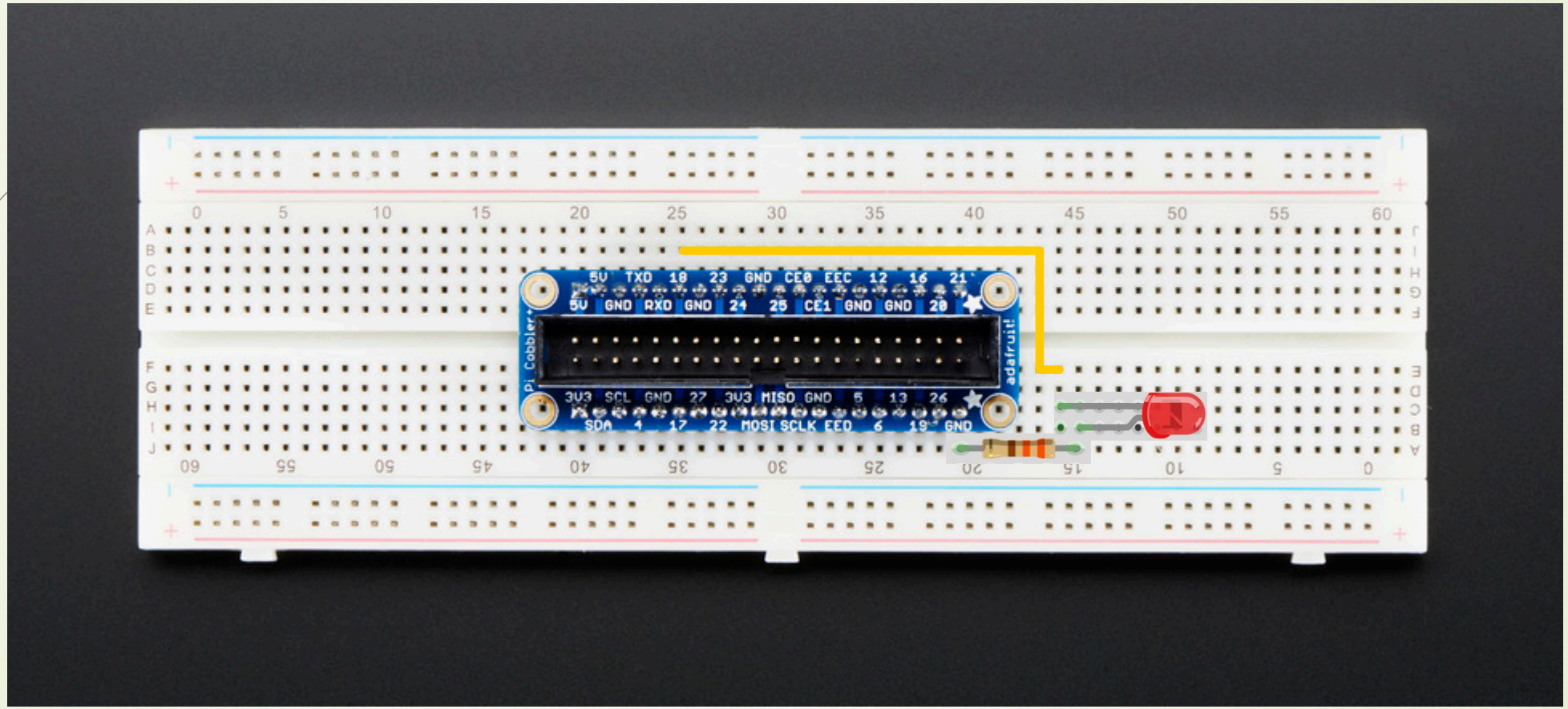
6. Blinking LED

Let's See If You Can Make Your Raspberry Pi Work!

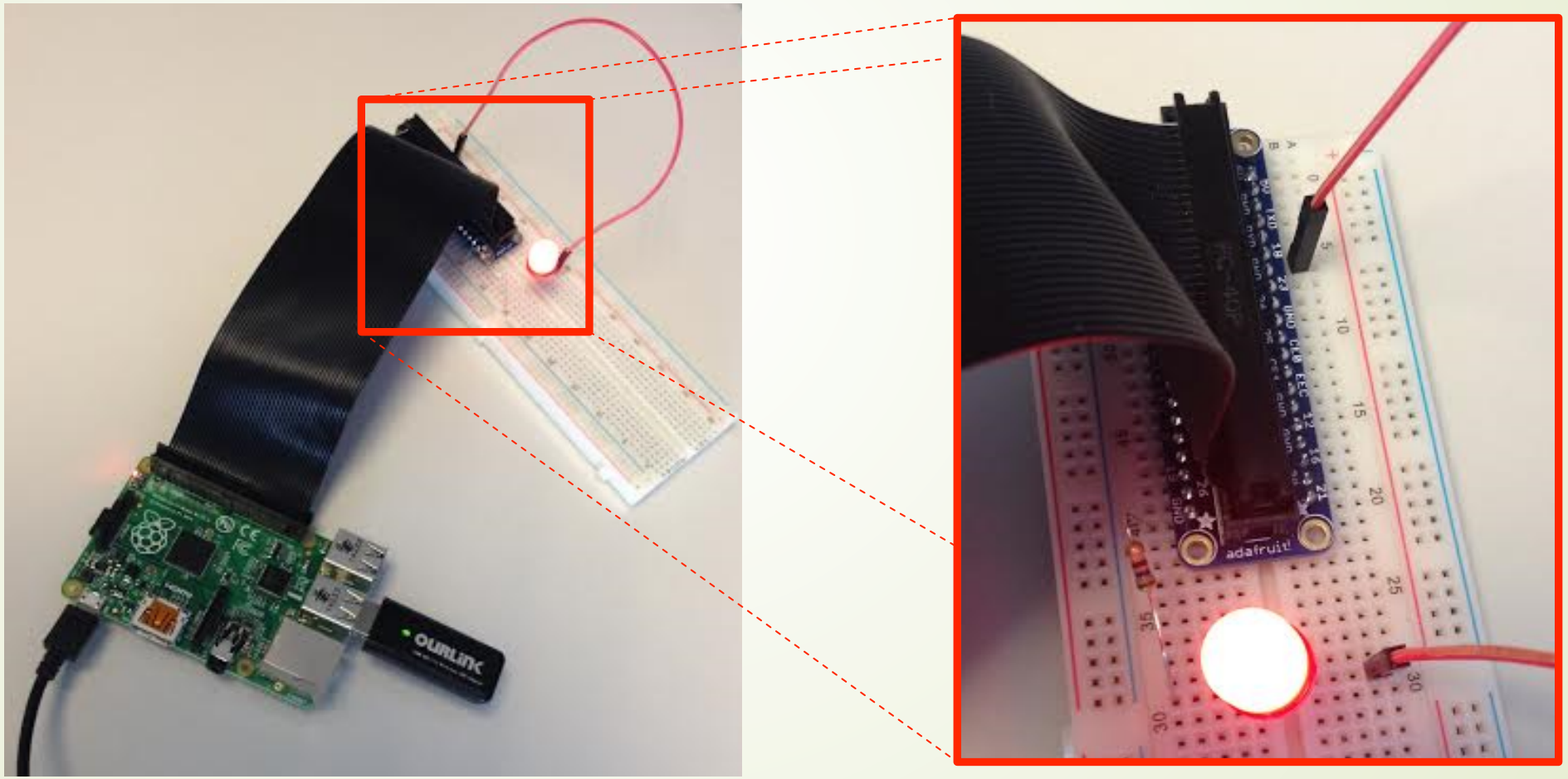
What You Will Need

- (1) male/male jumper wires
- (1) Diffused 10mm Red LED
- (1) 560 ohm 5% 1/4W Resistor

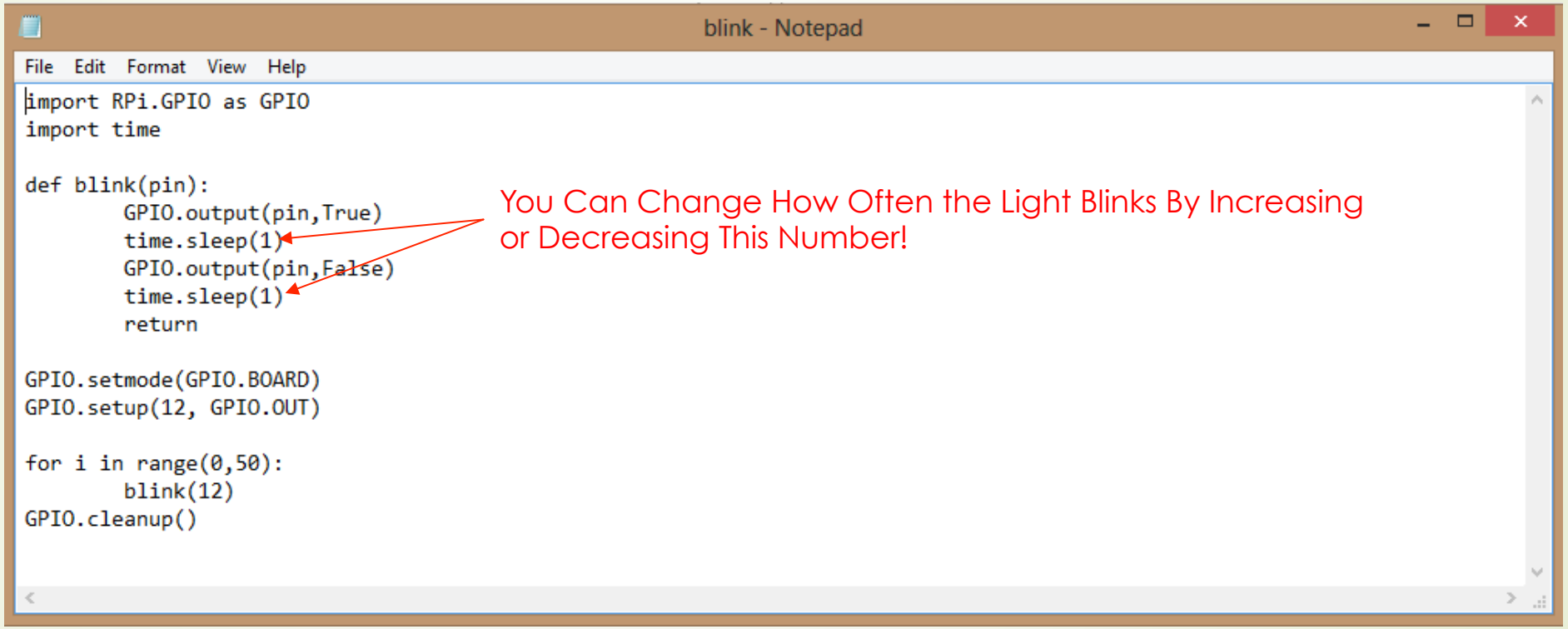
Hardware


5V TXD 18 23 GND CE0 EEC 12 16 21
5V GND RXD GND 24 25 CE1 GND GND 20
3V3 SCL GND 27 3V3 MISO GND 5 13 26
SDA 4 17 22 MOSI SCLK EED 6 19 GND
Pi Cobbler+
adafruit!

Physical Hardware



Software (The Code)



7. CPU Temperature

Now That We Know Our Raspberry Pi Works, Let's Begin Our Conversation About Weather!

What You Will Need

- As it turns out, you won't need anything!
- To begin though, it is important to remove the LED, resistor and any wires off of the bread board except the Pi Cobbler +, which should stay connected
- **REMEMBER** to shutdown the RPi before removing any parts!

Software

- This code also posts your information to a website. You can see it updating by going to the URL:

  http://piland.socialdevices.io/403/display

Untitled - Notepad

File Edit Format View Help

```
import time
import requests

room = 403      # Room number to use (1 through 999)

dataslot = 1    # Data slot number to use for CPU temp (1 through 30)
devicename = "Pi+CPU+Temp"      # Descriptive names for your device, put '+' for space char

baseur = "http://piland.socialdevices.io"       # Building url that data will be posted to
baseurl = baseur + "/" + str(room) + "/write/" + str(dataslot) + "?name=" + devicename + "&value="

while 1:

# Finding CPU temperature
        devicepath = open("/sys/class/thermal/thermal_zone0/temp")
        devicepath.seek(0,0);   # Read the temperature value
        rawtemp = devicepath.read().rstrip()
        tempC = float(rawtemp) / 1000.0 # Convert to the units we want
        url = baseurl + "%0.3f" % tempC + "+C"
        requests.get(url)    # Write data

        print(str(tempC)+' C') # Printing CPU Temperature

        time.sleep(5) # Assesing temperature every 5 seconds
```

Make Sure This number is Different From Everyone Else and They Match!

You Can Change How Often RPi Reads The Temperature By Increasing or Decreasing This Number!

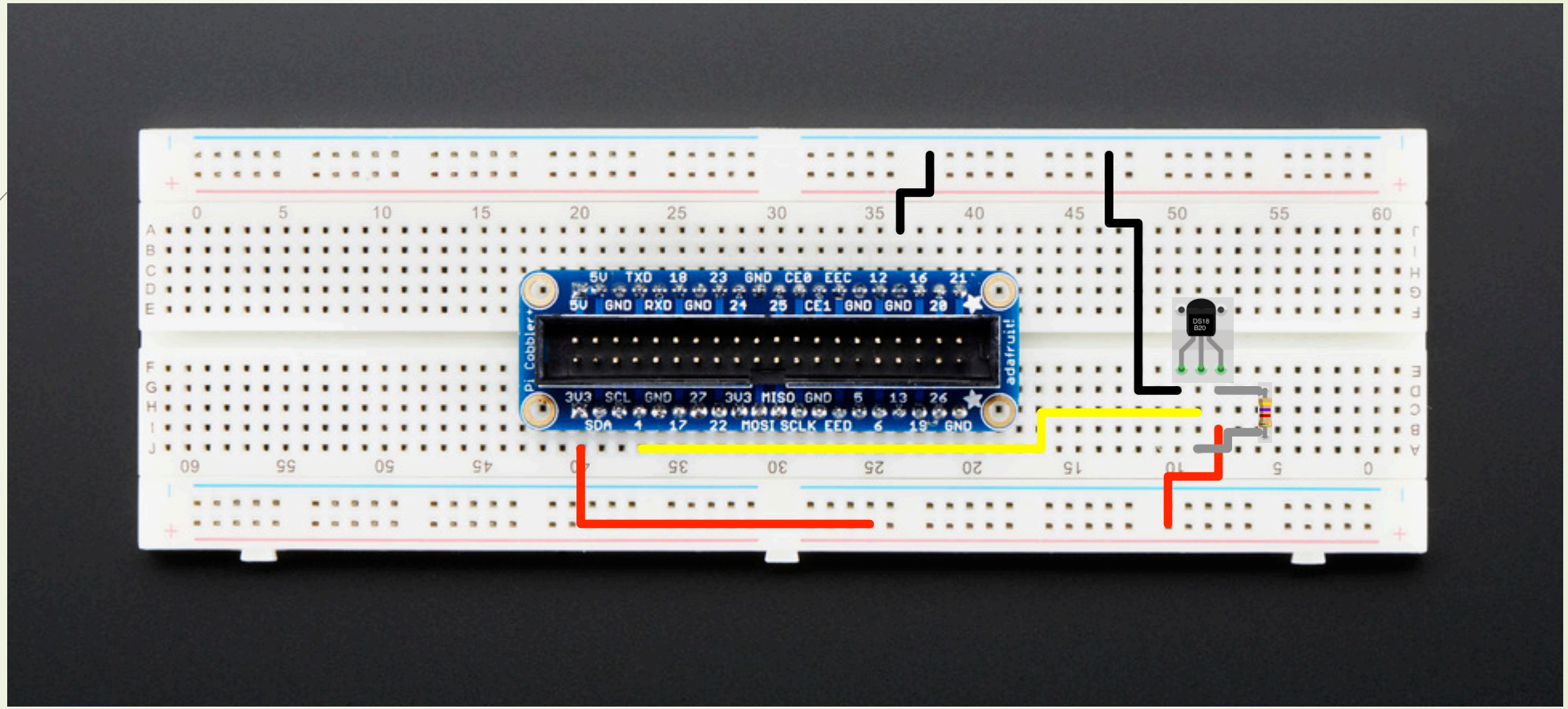
8. Inside Temperature Using DS18B20 Sensor

Now That We Have The CPU Temperature, Let's Get The Inside Temperature and Compare Them!

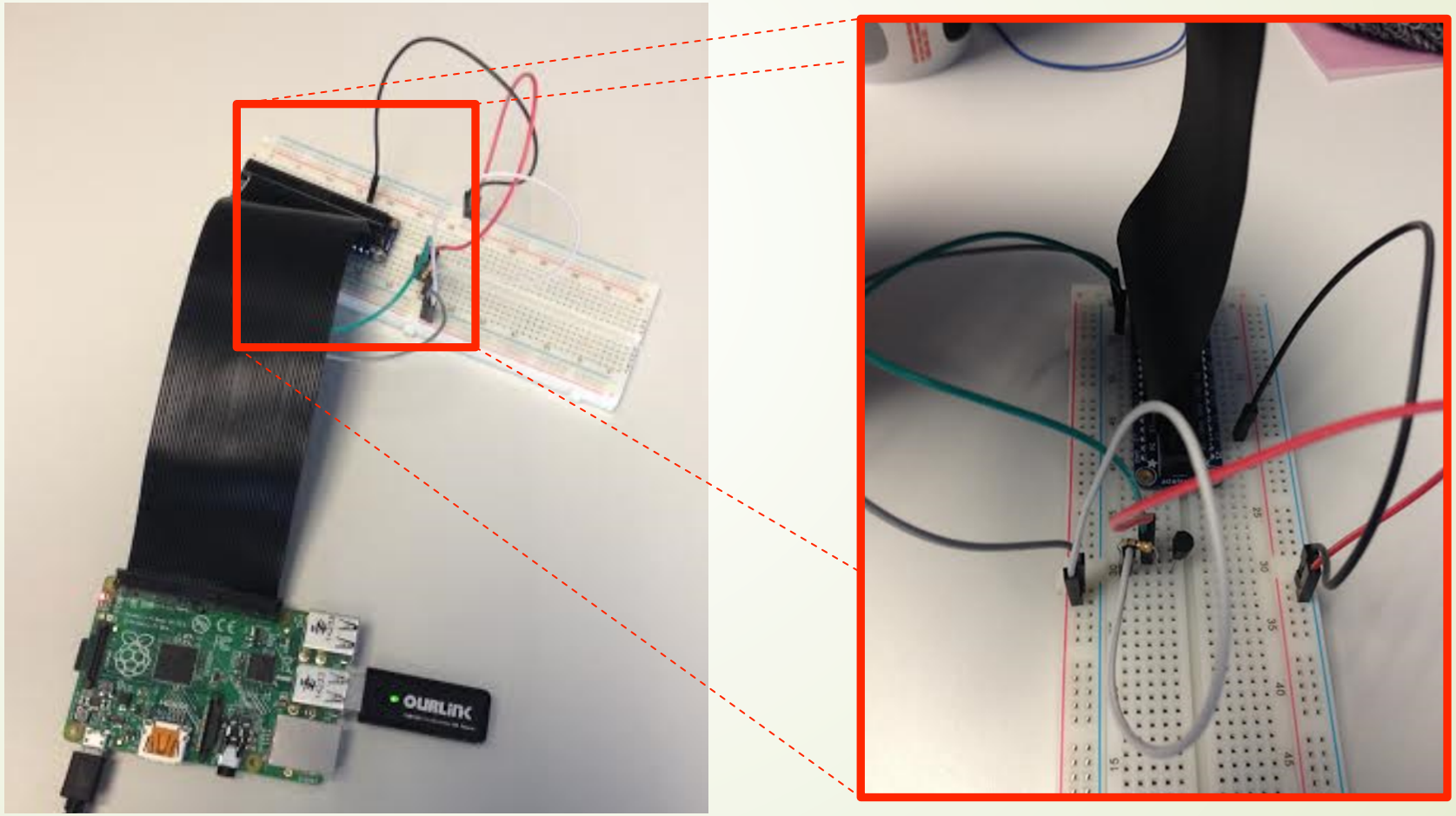
What You Will Need

- (1) DS18B20 Temperature Sensor
- (1) 10K 5% 1/4W Resistor (included)
- (5) male/male jumper wires

Hardware


5V TXD 18 23 GND CE0 EEC 12 16 21
5V GND RXD GND 24 25 CE1 GND GND 20
3V3 SCL GND 27 3V3 MISO GND 5 13 26
SDA 1 17 22 MOSI SCLK EED 6 19 GND
Pi Cobbler+
adafruit!
DS18 B20

Physical Hardware



Initializing the DS18B20 Sensor

- You will need to update /boot/config.txt. Use sudo nano /boot/config.txt to edit the file and add the following line to the end of the file:
  - dtoverlay=w1-gpio

- You will need to type the following commands to initialize the DS18B20 sensor for use:

- sudo modprobe w1-gpio
- sudo modprobe w1-therm

- Then you will need to get the device ID of the sensor:
- cd /sys/bus/w1/devices
- ls

- You should see a directory that looks like 28-000005aaee49. This is the device ID of the sensor. This device ID will be used in your python code to get the identified sensor.

Update the Module File to Run One Wire at boot time

- sudo nano /etc/modules
- Edit the file so it looks like below:

```
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.
# Parameters can be specified after the module name.
#
# sound devices
snd-bcm2835

# SPI devices
spi-dev

# I2C devices
i2c-dev
I2c-bcm2708

# 1-Wire devices
w1-gpio

# 1-Wire thermometer devices
w1-therm
```

Software

- Add the following (circled in red) to your previous code so that both temperatures will be posted to the website!

DS18B20_Temp - Notepad

File Edit Format View Help

```
import time
import requests

room = 403      # Room number to use (1 through 999)

dataslot = 1    # Data slot number to use for CPU temp (1 through 30)
dataslot2 = 2   # Data slot number to use for DS18B20 inside temp

devicename = "Pi+CPU+Temp"      # Descriptive names for your device, put '+' for space char
devicename2 = "Inside+Temp+DS18B20"

baseur = "http://piland.socialdevices.io"       # Building url that data will be posted to
baseurl = baseur + "/" + str(room) + "/write/" + str(dataslot) + "?name=" + devicename + "&value="
baseur2 = baseur + "/" + str(room) + "/write/" + str(dataslot2) + "?name=" + devicename2 + "&value="

while 1:

# Finding CPU temperature
        devicepath = open("/sys/class/thermal/thermal_zone0/temp")
        devicepath.seek(0,0);   # Read the temperature value
        rawtemp = devicepath.read().rstrip()
        tempC = float(rawtemp) / 1000.0 # Convert to the units we want
        url = baseurl + "%0.3f" % tempC + "+C"
        requests.get(url)    # Write data
```

Ln 21, Col 45

Software (cont.)

DS18B20_Temp - Notepad

File Edit Format View Help

```
# Finding CPU temperature
        devicepath = open("/sys/class/thermal/thermal_zone0/temp")
        devicepath.seek(0,0);   # Read the temperature value
        rawtemp = devicepath.read().rstrip()
        tempC = float(rawtemp) / 1000.0 # Convert to the units we want
        url = baseurl + "%0.3f" % tempC + "+C"
        requests.get(url)    # Write data

        print('CPU Temp is '+str(tempC)+' C') # Printing CPU Temperature

# Find temperature of DS18B20 - inside temperature

        temp_string = open("/sys/bus/w1/devices/28-000005aaee49/w1_slave")      # Accessing data from DS18B20
        thetext = temp_string.read()    # Reading temperature data
        temp_string.close()     # Closing temperature data
        tempdata = thetext.split("\n")[1].split(" ")[9] # Reformating temperature data
        tempC = float(tempdata[2:]) / 1000.0    # Putting temperature into the correct units
        ur2 = baseur2 + "%0.3f" % tempC + "+C"
        requests.get(ur2)    # write data

        print('DS18B20 Temp is '+str(tempC)+' C') # Printing DS18B20 Temperature

        time.sleep(5) # Assesing temperature every 5 seconds
```

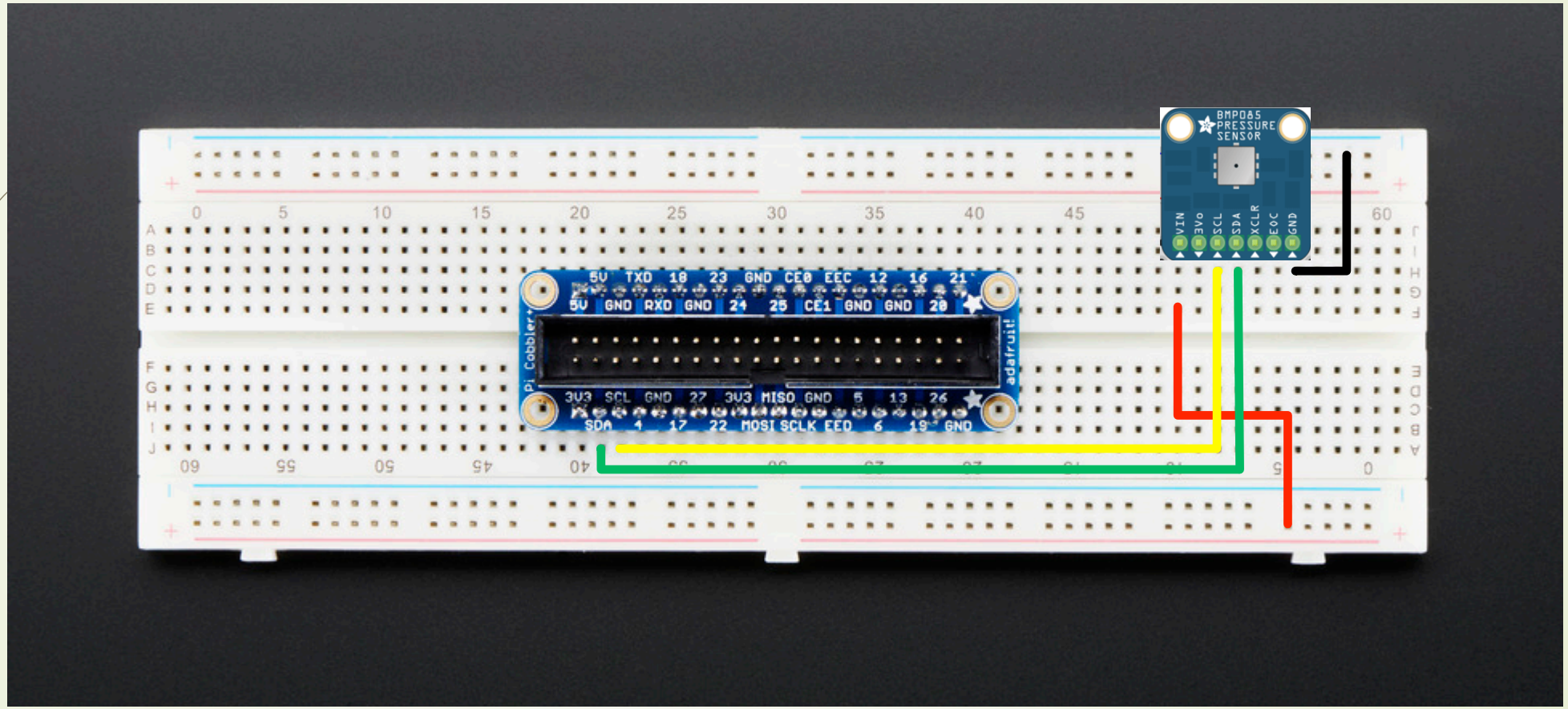
9. Inside Temperature and Pressure Using BMP180 Sensor

Now That We Have Inside Temperature, Let's Find Pressure! This Sensor Can Also Calculate Temperature So Let's Compare It To The DS18B20 Sensor!

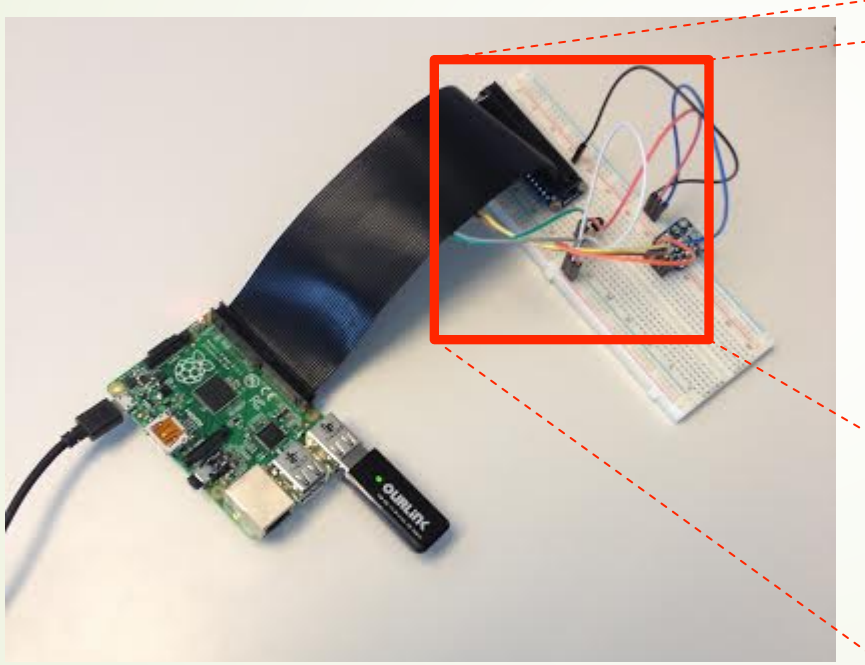
What You Will Need

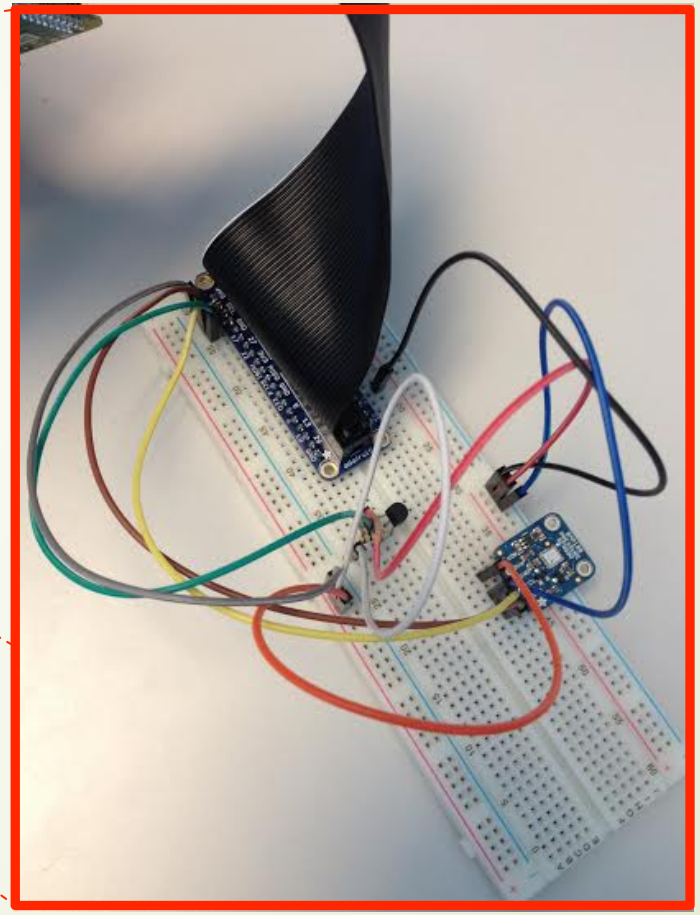
- (1) BMP180 Pressure Sensor
- (4) male/male jumper wires
- Do not disconnect the DS18B20 Sensor as we will be adding on to our pre-existing hardware in this project
- Please don't forget to SHUTDOWN the RPi before altering the hardware

Hardware


BMP085
PRESSURE
SENSOR
VIN
3Vo
SCL
SDA
XCLR
EOC
GND
Pi Cobbler+
adafruit!
5V TXD 18 23 GND CE0 EEC 12 16 21
5V GND RXD GND 24 25 CE1 GND GND 20
3V3 SCL GND 27 3V3 MISO GND 5 13 26
SDA 1 17 22 MOSI SCLK EED 6 19 GND

Physical Hardware





Software

- Add the following (circled in red) to your previous code so that temperature and pressure will be posted to the website!

BMP180_Temp - Notepad

File Edit Format View Help

```
import time
import requests
import Adafruit_BMP.BMP085 as BMP085

room = 403      # Room number to use (1 through 999)

dataslot = 1    # Data slot number to use for CPU temp (1 through 30)
dataslot2 = 2   # Data slot number to use for DS18B20 inside temp
dataslot3 = 3   # Data slot number to use for BMP180 inside temp
dataslot4 = 4   # Data slot number to use for BMP180 inside pressure

devicename = "Pi+CPU+Temp"      # Descriptive names for your device, put '+' for space char
devicename2 = "Inside+Temp+DS18B20"
devicename3 = "Inside+Temp+BMP180"
devicename4 = "Inside+Pressure+BMP180"


baseur = "http://piland.socialdevices.io"       # Building url that data will be posted to
baseurl = baseur + "/" + str(room) + "/write/" + str(dataslot) + "?name=" + devicename + "&value="
baseur2 = baseur + "/" + str(room) + "/write/" + str(dataslot2) + "?name=" + devicename2 + "&value="
baseur3 = baseur + "/" + str(room) + "/write/" + str(dataslot3) + "?name=" + devicename3 + "&value="
baseur4 = baseur + "/" + str(room) + "/write/" + str(dataslot4) + "?name=" + devicename4 + "&value="

while 1:

# Finding CPU temperature
        devicepath = open("/sys/class/thermal/thermal_zone0/temp")
        devicepath.seek(0,0);   # Read the temperature value
        rawtemp = devicepath.read().rstrip()
        tempC = float(rawtemp) / 1000.0 # Convert to the units we want
        url = baseurl + "%0.3f" % tempC + "+C"
        requests.get(url)    # Write data

        print('CPU Temp is '+str(tempC)+' C') # Printing CPU Temperature
```

Software (cont.)

BMP180_Temp - Notepad

File Edit Format View Help

```
        print('CPU Temp is '+str(tempC)+' C') # Printing CPU Temperature

# Find temperature of DS18B20 - inside temperature

        temp_string = open("/sys/bus/w1/devices/28-000005aaee49/w1_slave")      # Accessing data from DS1$
        thetext = temp_string.read()    # Reading temperature data
        temp_string.close()     # Closing temperature data
        tempdata = thetext.split("\n")[1].split(" ")[9] # Reformating temperature data
        tempC = float(tempdata[2:]) / 1000.0    # Putting temperature into the correct units
        ur2 = baseur2 + "%0.3f" % tempC + "+C"
        requests.get(ur2)    # write data

        print('DS18B20 Temp is '+str(tempC)+' C') # Printing DS18B20 Temperature

# Find temperature of BMP180 - inside temperature

        sensor = BMP085.BMP085()        # Gathering data from BMP180
        tempC = float(sensor.read_temperature())        # Reading temp data from BMP180
        ur3 = baseur3 + "%0.3f" % tempC + "+C"
        requests.get(ur3)    # write data

        print('BMP180 Temp is '+str(tempC)+' C') # Printing BMP180 Temperature

# Find pressure of BMP180 - inside pressure

        pressurePA = float(sensor.read_pressure())      # Reading pressure data from BMP180
        ur4 = baseur4 + "%0.3f" % pressurePA + "+Pa"
        requests.get(ur4)    # write data

        print('BMP180 Press is '+str(pressurePA)+' Pa') # Printing BMP180 Pressure

        time.sleep(5) # Assesing temperature every 5 seconds
```

10. Outside Temperature

Let's Take A Look Outside And Compare This Temperature To The Inside Temperature!

What We Will Need

- For this experiment won't actually need anything since we will be taking the data from the WeatherBug station on the roof of our school building
- Our local weather data can be found at the following website:

  http://isapidata.weatherbug.com/WxDataISAPI/WxDataISAPI.dll?GetData60&Magic=10991&RegNum=2553114&ZipCode=&StationID=AWSHQ&Units=0&Version=2.7&Fore=1&t=987689689

Software

- Add the following (circled in red) to your previous code so that the outside temperature will be posted to the website!

WB_Temp - Notepad

File Edit Format View Help

```
import time
import requests
import Adafruit_BMP.BMP085 as BMP085
import urllib2

room = 403      # Room number to use (1 through 999)

dataslot = 1    # Data slot number to use for CPU temp (1 through 30)
dataslot2 = 2   # Data slot number to use for DS18B20 inside temp
dataslot3 = 3   # Data slot number to use for BMP180 inside temp
dataslot4 = 4   # Data slot number to use for BMP180 inside pressure
dataslot5 = 5   # Data slot number to use for WeatherBug outside temp

devicename = "Pi+CPU+Temp"      # Descriptive names for your device, put '+' for space char
devicename2 = "Inside+Temp+DS18B20"
devicename3 = "Inside+Temp+BMP180"
devicename4 = "Inside+Pressure+BMP180"
devicename5 = "Outside+Temp+WB"

baseur = "http://piland.socialdevices.io"       # Building url that data will be posted to
baseurl = baseur + "/" + str(room) + "/write/" + str(dataslot) + "?name=" + devicename + "&value="
baseur2 = baseur + "/" + str(room) + "/write/" + str(dataslot2) + "?name=" + devicename2 + "&value="
baseur3 = baseur + "/" + str(room) + "/write/" + str(dataslot3) + "?name=" + devicename3 + "&value="
baseur4 = baseur + "/" + str(room) + "/write/" + str(dataslot4) + "?name=" + devicename4 + "&value="
baseur5 = baseur + "/" + str(room) + "/write/" + str(dataslot5) + "?name=" + devicename5 + "&value="

while 1:

# Finding CPU temperature
        devicepath = open("/sys/class/thermal/thermal_zone0/temp")
        devicepath.seek(0,0);   # Read the temperature value
        rawtemp = devicepath.read().rstrip()
        tempC = float(rawtemp) / 1000.0 # Convert to the units we want
        url = baseurl + "%0.3f" % tempC + "+C"
        requests.get(url)    # Write data
```

Software (cont.)

WB_Temp - Notepad

File Edit Format View Help

```
# Finding CPU temperature
        devicepath = open("/sys/class/thermal/thermal_zone0/temp")
        devicepath.seek(0,0);   # Read the temperature value
        rawtemp = devicepath.read().rstrip()
        tempC = float(rawtemp) / 1000.0 # Convert to the units we want
        url = baseurl + "%0.3f" % tempC + "+C"
        requests.get(url)    # Write data

        print('CPU Temp is '+str(tempC)+' C') # Printing CPU Temperature

# Find temperature of DS18B20 - inside temperature

        temp_string = open("/sys/bus/w1/devices/28-000005aaee49/w1_slave")      # Accessing data from DS1$
        thetext = temp_string.read()    # Reading temperature data
        temp_string.close()     # Closing temperature data
        tempdata = thetext.split("\n")[1].split(" ")[9] # Reformating temperature data
        tempC = float(tempdata[2:]) / 1000.0    # Putting temperature into the correct units
        ur2 = baseur2 + "%0.3f" % tempC + "+C"
        requests.get(ur2)    # write data

        print('DS18B20 Temp is '+str(tempC)+' C') # Printing DS18B20 Temperature

# Find temperature of BMP180 - inside temperature

        sensor = BMP085.BMP085()        # Gathering data from BMP180
        tempC = float(sensor.read_temperature())        # Reading temp data from BMP180
        ur3 = baseur3 + "%0.3f" % tempC + "+C"
        requests.get(ur3)    # write data

        print('BMP180 Temp is '+str(tempC)+' C') # Printing BMP180 Temperature

# Find pressure of BMP180 - inside pressure

        pressurePA = float(sensor.read_pressure())      # Reading pressure data from BMP180
        ur4 = baseur4 + "%0.3f" % pressurePA + "+Pa"
        requests.get(ur4)    # write data
```

Software (cont.)

WB_Temp - Notepad

File Edit Format View Help

```
        print('BMP180 Temp is '+str(tempC)+' C') # Printing BMP180 Temperature

# Find pressure of BMP180 - inside pressure

        pressurePA = float(sensor.read_pressure())      # Reading pressure data from BMP180
        ur4 = baseur4 + "%0.3f" % pressurePA + "+Pa"
        requests.get(ur4)    # write data

        print('BMP180 Press is '+str(pressurePA)+' Pa') # Printing BMP180 Pressure

# Find temperature of Weatherbug - outside temperature

        response = urllib2.urlopen('http://isapidata.weatherbug.com/WxDataISAPI/WxDataISAPI.dll?GetData60&Magic=10991&RegNum=2553114&ZipCode=&StationID=AWSHQ&Units' \
        '=0&Version=2.7&Fore=1&t=987689689')

        html = response.read()  # Reading URL

        j = 0   # Defining arbitrary value

        for in range(0,len(html)):
                if html[i] == "|"
                        j = j + 1
                if j == 3:
                        b = i+1
                        j = 4
                if j == 5:
                        e = i
                        j = 6

        tempC = (float(html[b:e])-32)/1.8
        ur5 = baseur5 + "%0.3f" % tempC + "+C"
        requests.get(ur5)    # write data

        print('Outside Temp is '+str(tempC)+' C') # Printing WeatherBug Temperature

        time.sleep(5) # Assesing temperature every 5 seconds
```

Bringing it all together

- Once you have all the code running and posting data to socialdevices.io, you will want to set your PI to run continuously and take data.
- To do this, we recommend you install a program called “tmux”. This application will let your program run in a separate process space even when you are disconnected from the unit. To install tmux type in “sudo apt-get install tmux”
- Now run “tmux” at the prompt. You will see a new prompt screen being generated.
- Now to run your program continuously type “sudo python WB_Temp.py &”

Running applications on boot

- To have a program run on startup, you can edit the rc.local file : sudo nano /etc/rc.local
- In the rc.local file, add a command such as the following which will run the application in the background:

sudo python /home/pi/WB_Temp.py &

References

http://workshop.raspberrypiaustralia.com/usb/ttl/connecting/2014/08/31/01-connecting-to-raspberry-pi-via-usb/

http://maxembedded.com/2014/07/using-raspberry-pi-gpio-using-python/#Circuit

https://github.com/IoTDevLabs/iot-educ/tree/master/rpi

https://learn.adafruit.com/adafruits-raspberry-pi-lesson-11-ds18b20-temperature-sensing/configure-and-test

https://learn.adafruit.com/using-the-bmp085-with-raspberry-pi/configuring-the-pi-for-i2c