In this lesson you will learn how to send temperature readings from a sensor on your Raspberry Pi to Adafruit IO. You must have an account for Adafruit IO, which is covered in another lesson.

Connect the temperature sensor to your Raspberry Pi

Wire the DS18B20 digital temperature sensor to the Raspberry Pi according to wiring diagram in Figure 1. You can use a breadboard. Alternatively, for a more permanent set up you could mount the sensor and resistor to a small square of circuit board and solder jumper cables to the sensor and resistor.

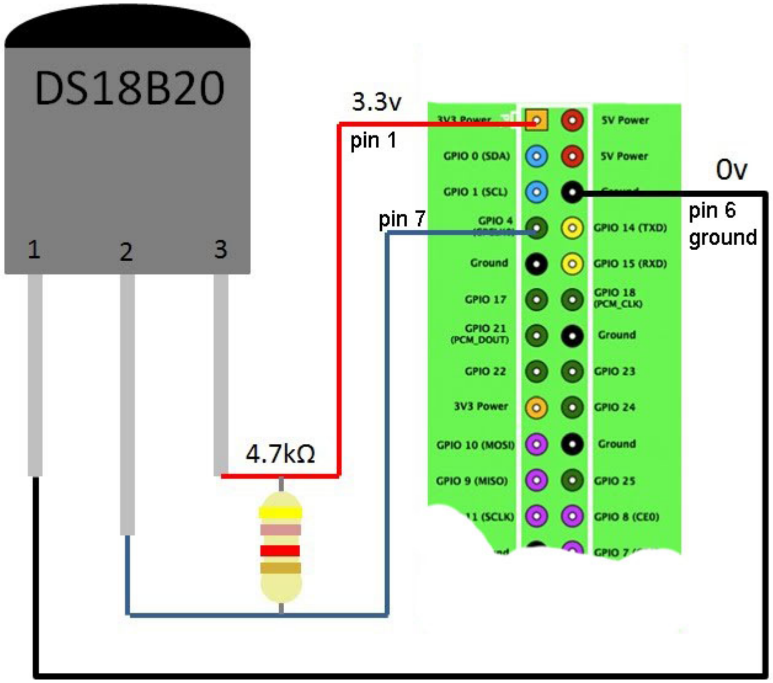

Figure 1 temperature sensor wiring diagram

Confirm that the boot configuration file for your Raspberry Pi contains a line that is required for use of the DS18B20 sensor.

1. Open the Terminal on your Raspberry Pi
2. Enter this command: **cd /boot**
3. Then enter this command: **ls**
4. A list of files will appear in the Terminal window. Included in the list should be a file named **config.txt**.
5. Enter the command: **cat config.txt**

6. The contents of config.txt will display in the Terminal window. There should be a line, probably near the bottom: **dtoverlay=w1-gpio,gpiopin=4**
7. If the line specified in step 6 is part of the config.txt file, then your Raspberry Pi is ready to make temperature measurements. If not, you will need to add the line to the file.
8. If you need to add the line, enter this command: **sudo nano config.txt**
9. Use the arrow keys to set the cursor at the bottom of the text and add the line required, exactly as given in step 6.
10. To save the file, hold down the control key (Ctrl) and press the X key. Then press the Y key to indicate you want to save the changes. Then press the Enter key.
11. To confirm your editing of the file enter this command: **cat config.txt** and check to see if the line is contained in the file
12. If you added the line to the file, reboot your RPi now.

Checking the function of your temperature sensor

It is now time to check your temperature sensor to confirm it is functioning. In addition, you need to determine the serial number of the sensor.

1. open the terminal and type in text of steps 2 through 5
2. sudo modprobe w1-gpio
3. sudo modprobe w1-therm
4. cd /sys/bus/w1/devices/
5. ls

You will see a listing of the current directory. There should be a directory listed that is the serial number of your temperature sensor. The serial number of one of GEAR's sensors is 28-0000093b782d, but every sensor will have a unique number. Write down your sensor number in your notebook as you will need it later in these instructions.

If you don't see a directory with lots of numbers and letters like the one above then:

- Check your circuit wiring.
- Make sure you have the correct resistor (this is very important - yellow, violet, red, gold).
- Feel the temperature gauge with your finger. If it feels hot then you have it wired back to front.

If you do see the directory then type cd followed by the directory name:

cd 28-0000093b782d [this is an example, use your sensor number instead]

cat w1_slave

The contents of the w1_slave file will now appear in the terminal window. The text includes the temperature data in Celsius (see example below). 25312 is 25.312 degrees Celsius.

Example text of the w1_slave file:

```
95 01 4b 46 7f ff 0b 10 0b : crc=0b YES
95 01 4b 46 7f ff 0b 10 0b   f=25312
```

When you entered the **cat w1_slave** command, if there is a readout like that above, with f=(then a number), your sensor is working!

Preparing Adafruit IO to receive the temperature data

You will need to make a **feed** and **dashboard** at the Adafruit IO web site. If you do not know how to do this, check the previous lesson covering Adafruit IO.

1. Create a feed for the temperature data (you could name it Temp1).
2. Create a dashboard to display the temperature data.
3. Open the dashboard and add a block (click the blue button with + symbol). Select the **Line Chart** as the block type.
4. Select your feed in the choose feeds step
5. In the Block settings step, add a title for your line chart and select a History setting (I suggest Live (no history) for this lesson). Then click the **Create block** button.

Now you are ready to write the Python program for sending temperature data to Adafruit IO. You must write the program using Python version 2. Below is the code. You do not need to add the comment lines (the ones that start with #)

```
# program to send temp data to adafruit io
# use a DS18B20 temperature sensor
# the config.txt file in the /boot directory must contain this line: dtoverlay=w1-gpio,gpiopin=4
# in above example, gpio pin specified is pin 4, so connect sensor to gpio pin 4
# if you want to use another gpio pin, then edit number in config.txt
# you will need to know the serial number of the temperature sensor
# only works with Python version 2
# code by J. La Favre


from Adafruit_IO import Client  # only works if you have installed Adafruit IO Python library on your RPi
import subprocess
import time

aio = Client('put your Adafruit IO key here') # put your key between the quotes



def get_temp(): # function that causes RPi to return a value from sensor

    subprocess.call(['modprobe', 'w1-gpio'])
    subprocess.call(['modprobe', 'w1-therm'])
    # two lines above result in a temperature measurement

    filename = '/sys/bus/w1/devices/28-0000093b782d/w1_slave'
    # my sensor serial number is 28-0000093b782d, you need to use your number

    tfile = open(filename)
    # open the file containing the temperature data

    text = tfile.read()
    # add the contents of the file to an object named text

    tfile.close()
    # close the file

    secondline = text.split('\n')[1]
    # the contents of the second line in object text are added to object named secondline
    temperaturedata = secondline.split(' ')[9]
    # split secondline at each space and add the contents of the tenth segment to object named temperaturedata

    temperature = float(temperaturedata[2:])
    # we don't want the first 2 characters in temperaturedata (f=), above line gets rid of them
    # finally, we have extracted just the temperature from the file and stored it in object named temperature

    temperature = temperature / 1000
    # we need to divide by 1000 to place the decimal point properly
    # the temperature in degrees C is now stored in object named temperature

    temp = float(temperature)
    # the contents of the object temperature are of the string  type
    # line above converts the string data to a floating point number and assigns it to object named temp

    tempDry = round(temp, 1)
    # round the value contained in object temp to one place past the decimal point
    #  assign the rounded value to an object named tempDry

    return tempDry
    # return the value contained in tempDry to the line in the program that called this function [get_temp()]

counter = 1
# counter for the while loop

print 'starting loop'
# in Python 2 we don't need to enclose text in parentheses
# in Python 3 correct syntax would be: print('starting loop')
```

```
while counter < 10:
#loop will run 9 times, it will make 9 temperature measurments

    temp = get_temp()
    # call the function named get_temp() and apply the returned value to object named temp

    aio.send('Temp1', temp)
    # send the value contained in temp via Adafruit IO feed named Temp1
    # change Temp1 to the name of your Adafruit IO feed

    print 'data value sent = ', temp
    # as program runs, you will see the temperature values being sent

    counter = counter + 1

    time.sleep(30)
    # pause for 30 seconds, then take next temp measurment

print 'finished running loop'
print 'program closing'
```