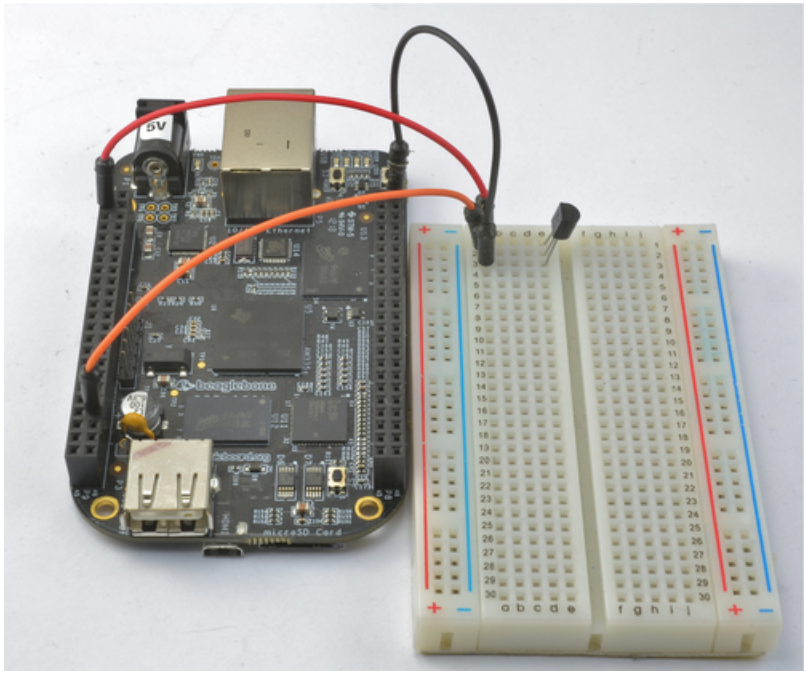
adafruit learning system

Measuring Temperature with a BeagleBone Black

Created by Simon Monk

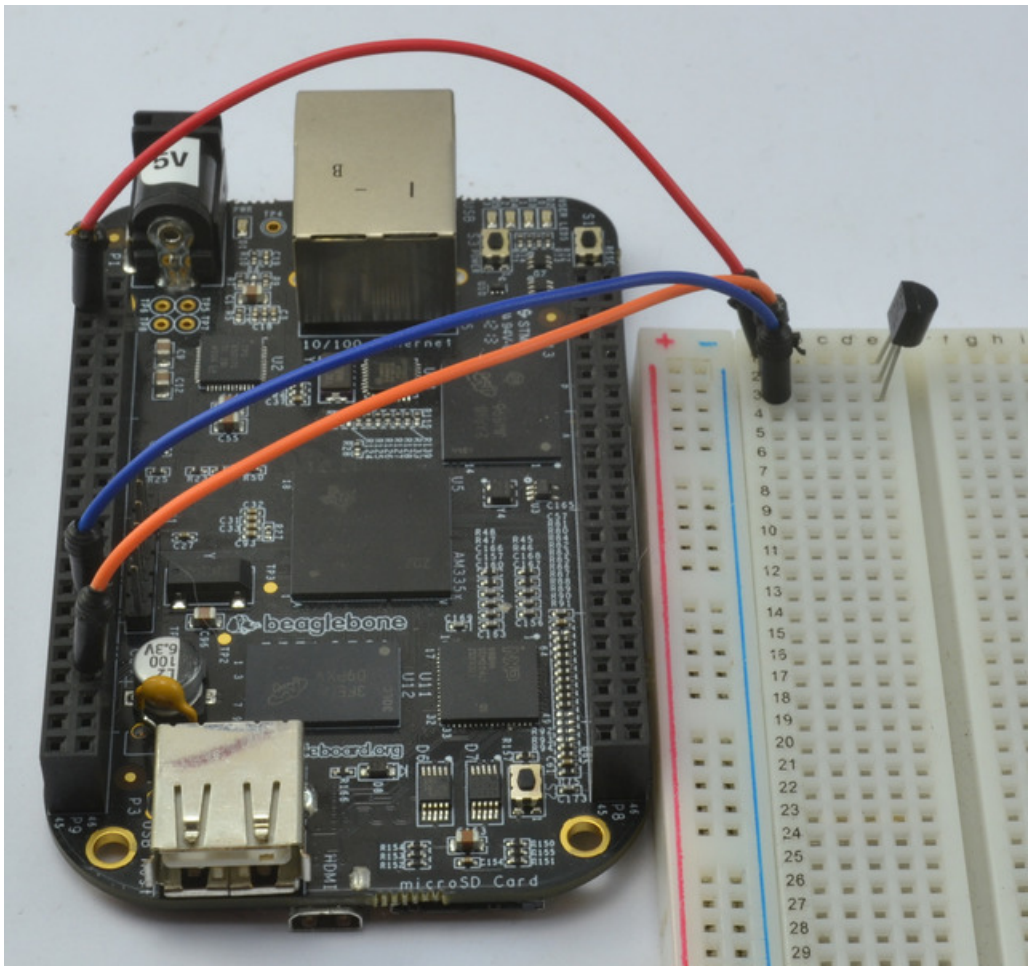


Last updated on 2018-08-22 03:36:13 PM UTC

Guide Contents

Overview

In this tutorial, you will learn how to connect temperature sensor to a BeagleBone Black.



Because the BBB runs Linux, there are many ways in which it can be programmed. In this tutorial we show how to read analog values form a TMP36 temperature sensor using Python.

The temperature sensor chip used is the TMP36. You can find out more about this chip here: http://learn.adafruit.com/tmp36-temperature-sensor/overview (https://adafru.it/cgZ)

You WIll Need

To try out this tutorial, you will need:

BeagleBone Black

TMP36

Half-sized Breadboard

Male to Male Jumpers

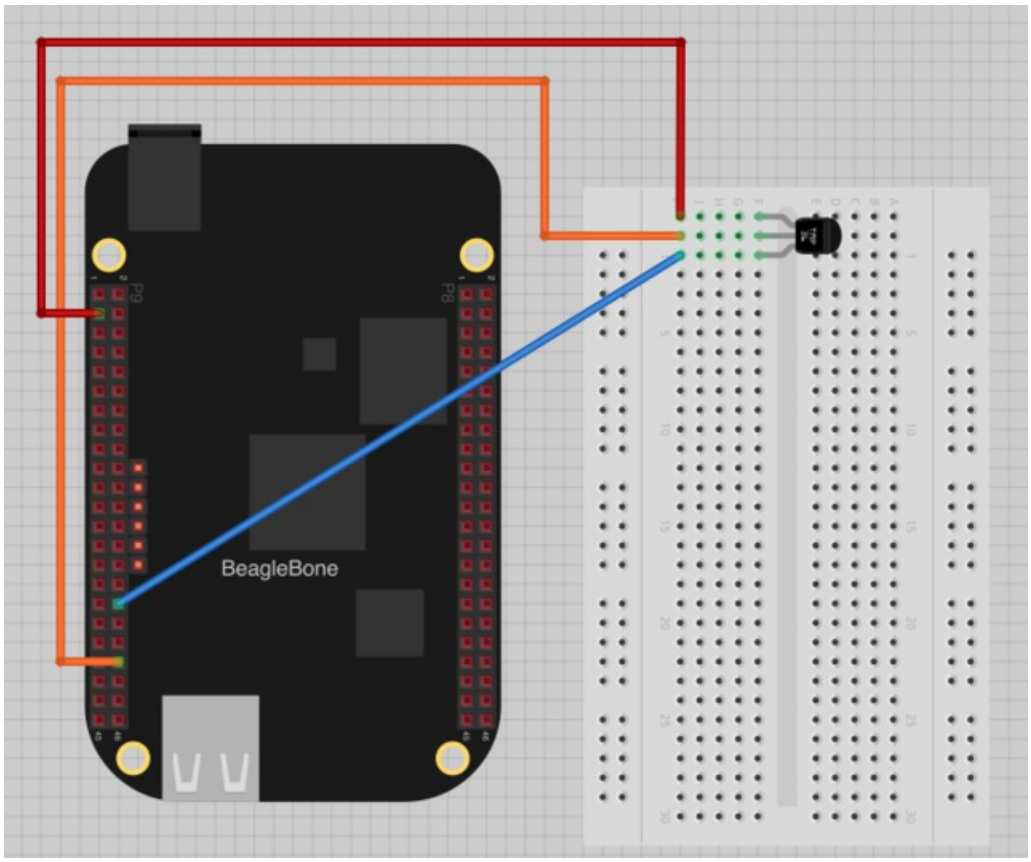
Installing the Python Library

This tutorial uses Ångström Linux, the operating system that comes pre-installed on the BBB.

Follow the instructions here, to install the Python IO BBIO library. http://learn.adafruit.com/setting-up-io-python-library-on-beaglebone-black (https://adafru.it/cgh)

Wiring

Wire up the solderless breadboard using the header leads as shown below.



Make sure that you get the TMP36 the right way around.

The blue lead is connected from the GNDA_ADC connection to the GND pin of the TMP36 temperature sensor. The red lead is connected from pin 3 of the other connector (3.3V) to the positive supply pin of the TMP36 and the orange lead to pin P9.40. Note that only certain pins can be used as analog inputs (see http://learn.adafruit.com/setting-up-io-python-library-on-beaglebone-black/adc (https://adafru.it/ch0))

The pins are numbered left to right, 1, 2 then on the next row down 3,4 etc. You can find out about all the pins available on the P8 and P9 connecters down each side of the BBB here: http://stuffwemade.net/hwio/beaglebone-pin-reference/ (https://adafru.it/cgi)

The Python Console

Before writing a Python program to measure and report the temperature, we can try some experiments in the Python Console.

To launch the Python Console type:

```
# python
Python 2.7.3 (default, Apr  3 2013, 21:37:23)
[GCC 4.7.3 20130205 (prerelease)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

First, we need to import the library, so enter the command:

```
>>> import Adafruit_BBIO.ADC as ADC
```

Now enter the command below into the Python Console to setup the ADC (Analog to Digital Convertor).

```
>>> ADC.setup()
```

You can now read the value at the analog input (between 0 and 1) using the command below. Try it a few times while putting your finger on the TMP36 to warm it up. The readings should gradually increase.

```
>>> ADC.read("P9_40")
0.38999998569488525
>>> ADC.read("P9_40")
0.38833332061767578
>>> ADC.read("P9_40")
0.39722222089767456
>>> ADC.read("P9_40")
0.42500001192092896
```

Note: Using the up arrow will repeat the last line that you entered in the Python Console.

These readings are not yet an actual temperature, so in the next section we will write a short Python program to display the temperature in both degrees C and F.

Writing a Program

Exit the Python Console by typing:

```
>>> exit()
```

This should take you back to the Linux prompt.

Enter the following command to create a new files called **tmp36.py**

```
nano tmp36.py
```

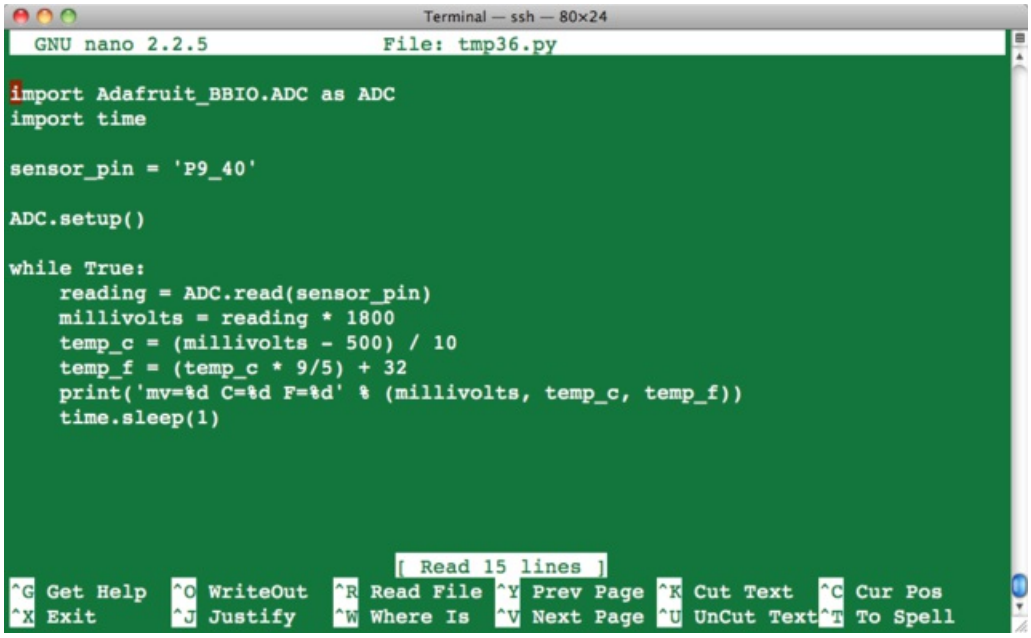
Now paste the code below into the editor window.

```
import Adafruit_BBIO.ADC as ADC
import time

sensor_pin = 'P9_40'

ADC.setup()

while True:
    reading = ADC.read(sensor_pin)
    millivolts = reading * 1800  # 1.8V reference = 1800 mV
    temp_c = (millivolts - 500) / 10
    temp_f = (temp_c * 9/5) + 32
    print('mv=%d C=%d F=%d' % (millivolts, temp_c, temp_f))
    time.sleep(1)
```



Save and exit the editor using CTRL-x and the Y to confirm.

To start the program, enter the command:

```
python tmp36.py
```

You will then see a series of readings.

```
mv=757 C=25 F=78
mv=760 C=26 F=78
mv=762 C=26 F=79
mv=765 C=26 F=79
mv=763 C=26 F=79
mv=763 C=26 F=79
mv=766 C=26 F=79
mv=768 C=26 F=80
```

When you want to stop the readings, use CTRL-c.

> Warning: The analog inputs of the BBB operate at 1.8V. Since the TMP36 has a theoretical maximum output of 3.3V, there is a potential for the BBB to be damaged if the voltage in millivolts exceeds 1.8V. This will only happen on a TMP36 if the temperature exceeds 130 degrees C (266 degrees F).

Next Steps

Now that you can measure the temperature, you could do other things with it like displaying a message when the temperature exceeds some limit.

You could also use the project as a temperature logger, using Python to write the readings to a file, each accompanied by a time stamp. If the temperatures are written one per line, with a comma between the time and the reading, then you will be able to import it directly into a spreadsheet and produce charts from the data.

About the Author.
As well as contributing lots of tutorials about Raspberry Pi, Arduino and now BeagleBone Black, Simon Monk writes books about open source hardware. You will find his books for sale here (https://adafru.it/caH) at Adafruit.