Blinking an LED with BeagleBone Black

Created by Simon Monk

https://learn.adafruit.com/blinking-an-led-with-beaglebone-black
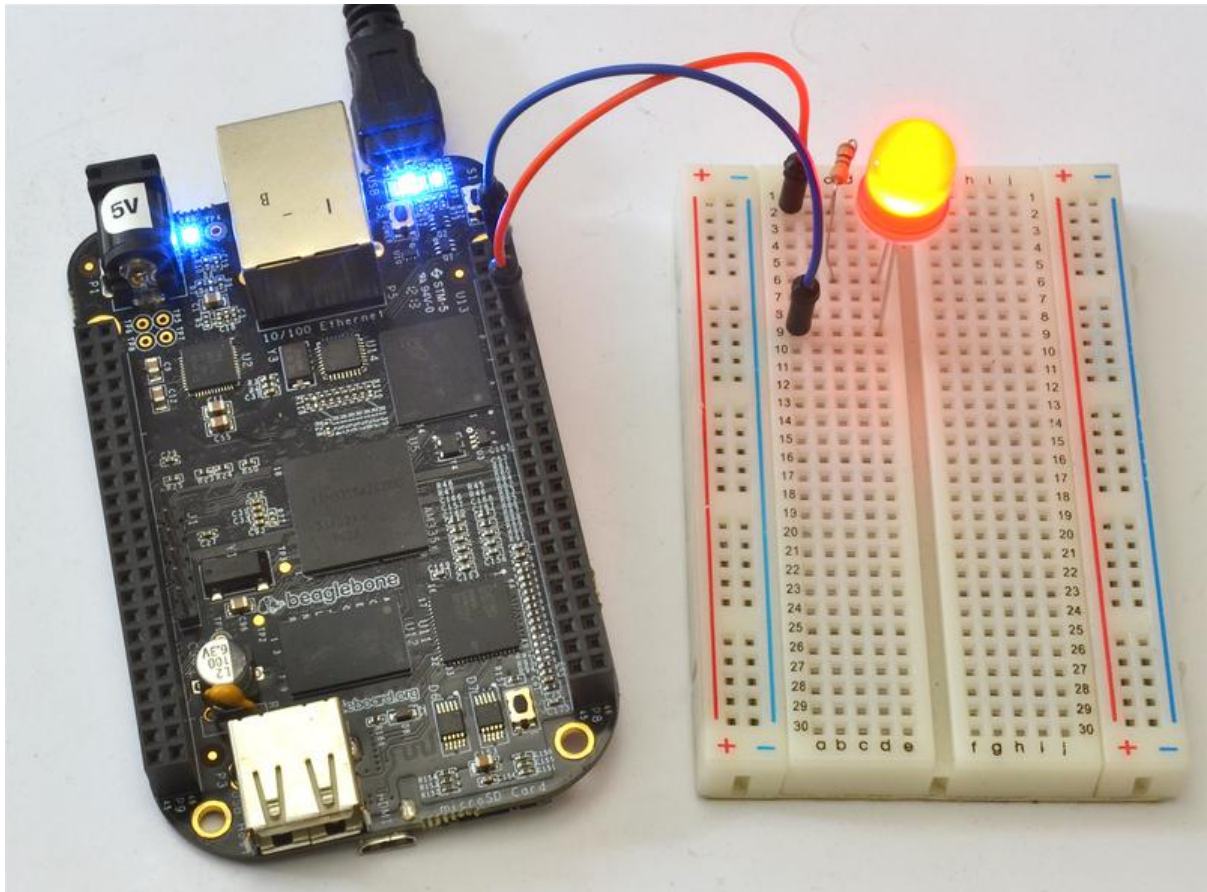
Last updated on 2023-08-29 02:21:01 PM EDT

Table of Contents

Overview

In this tutorial, you will learn how to control an external LED using a BeagleBone Black (BBB) from Python.
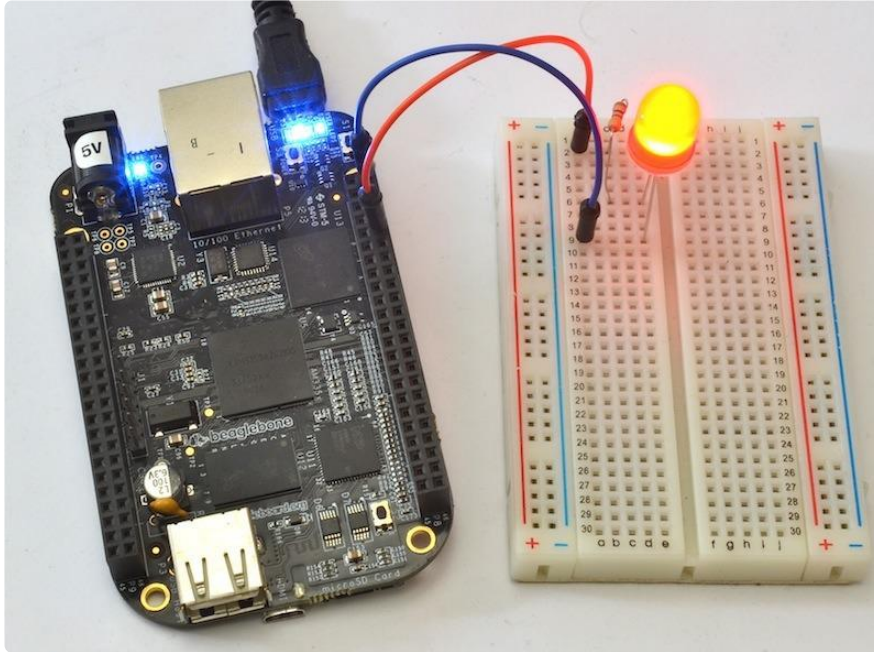
Because the BBB runs Linux, there are many ways in which it can be programmed. In this tutorial we show how to control an output pin and hence LED using Python.

You Will Need

BeagleBone Black

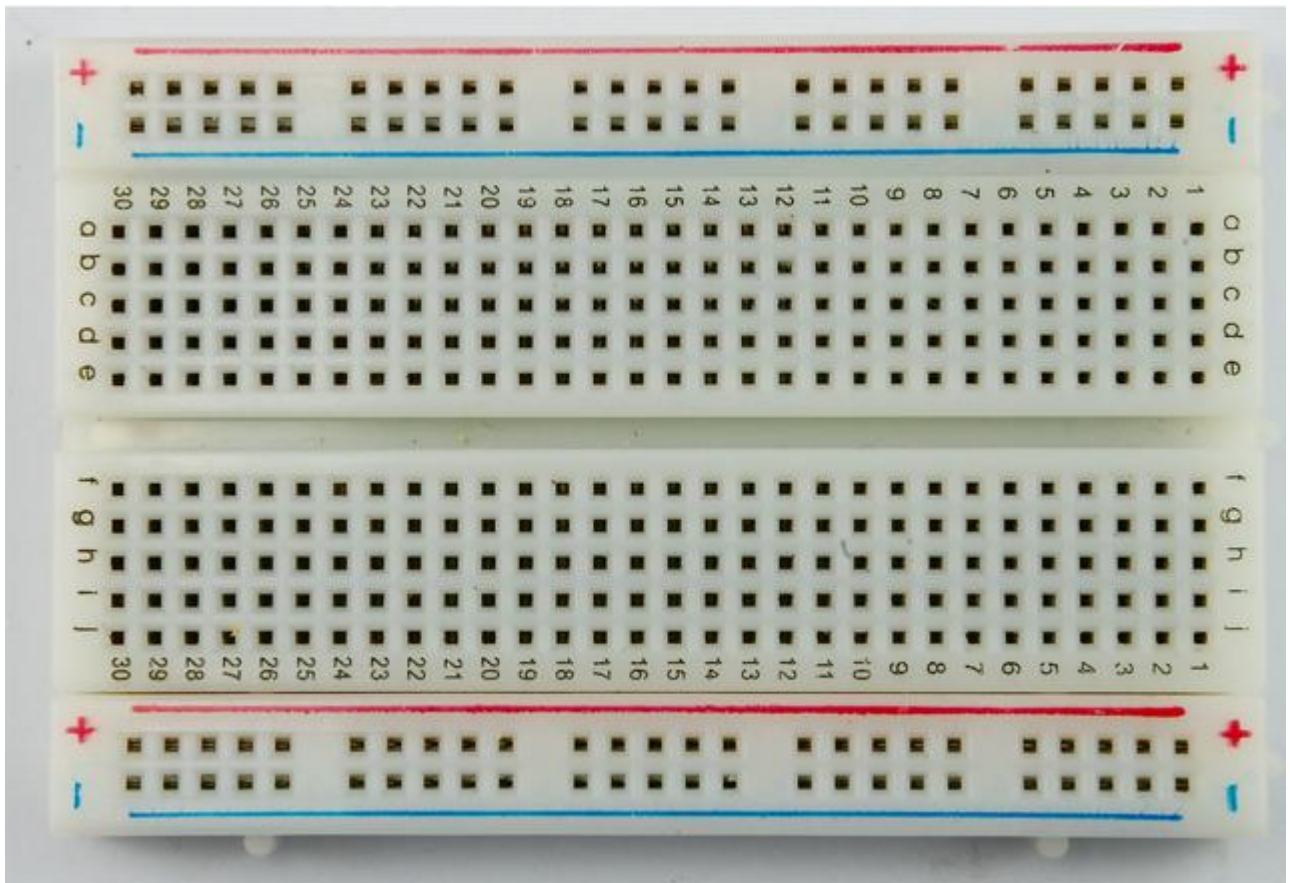LED your choice of color and size. I used 10mm Red

470Ω Resistor
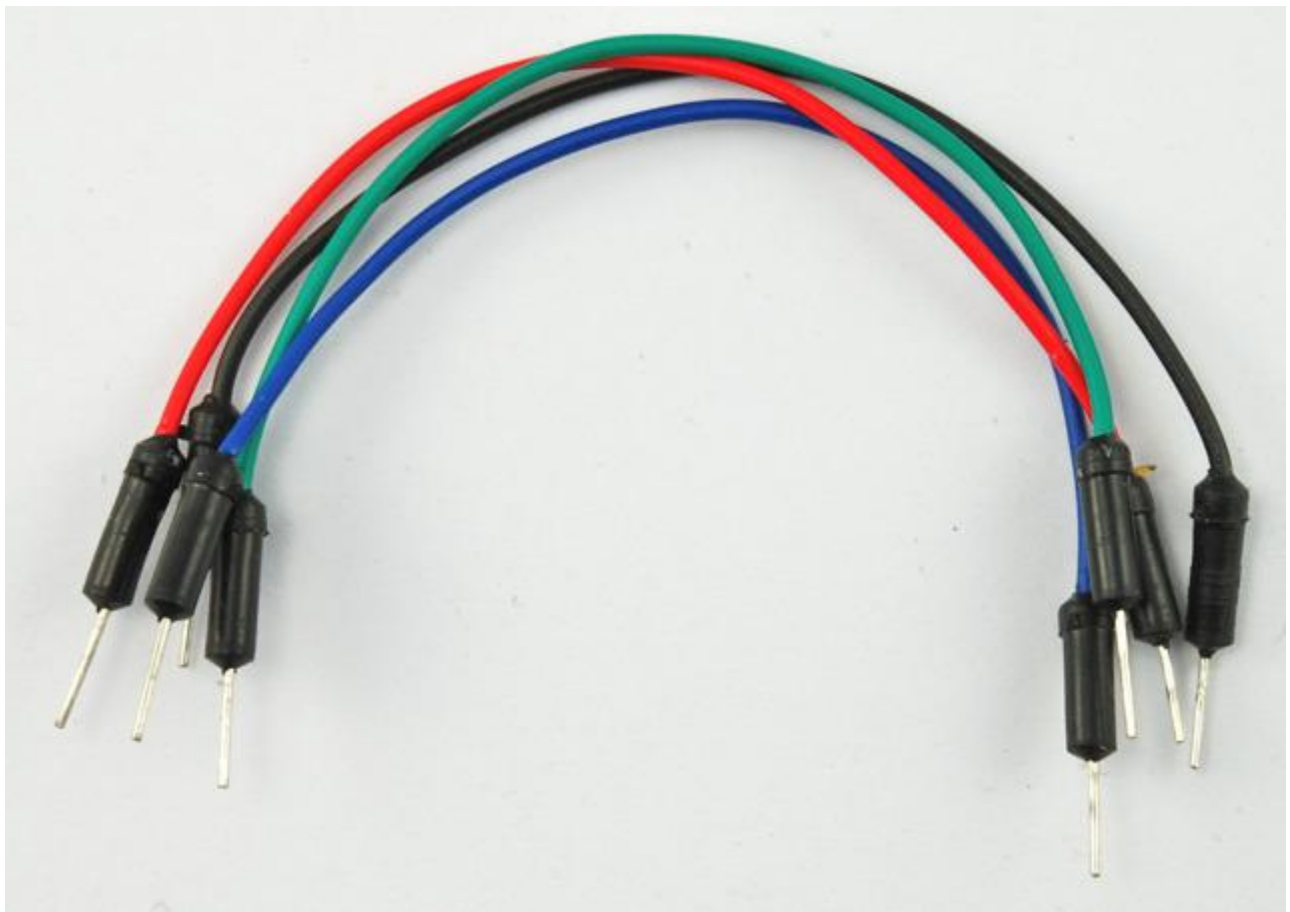
Half-size solderless breadboard

Male to Male jumper wires

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 library. http://learn.adafruit.com/setting-up-io-python-library-on-beaglebone-black ()

Wiring

Wire up the solderless breadboard using the header leads as shown below.
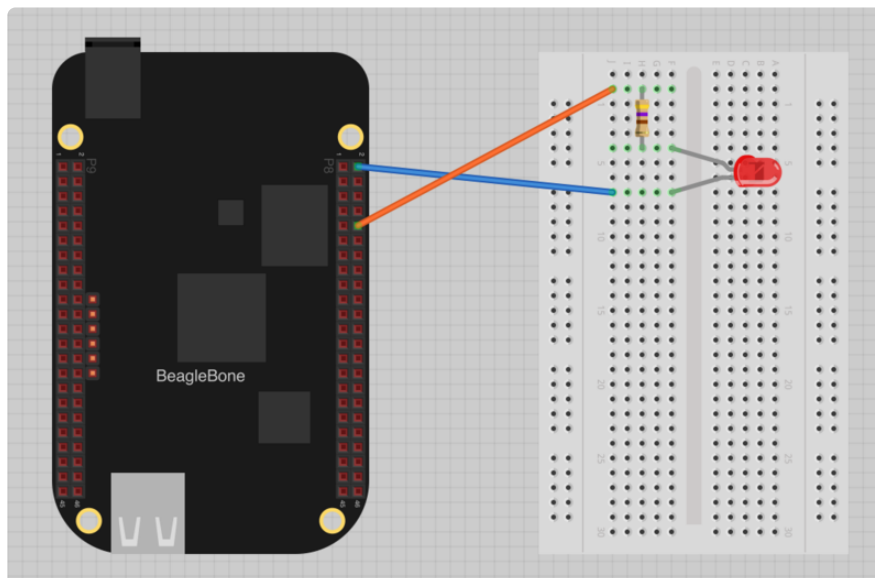
Push the LED leads into the breadboard, with the longer (positive) lead towards the top of the breadboard. It does not matter which way around the resistor goes.

The top two connections on the BBB expansion header we are using (P8) are both GND. The other lead is connected to pin 10, which is the right-hand connector on the fifth row down. 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/ ()

The Python Console

Before writing a Python program to control an LED, you can try some experiments from the Python console to control the LED.

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.GPIO as GPIO
```

By default, all the IO pins are set to be inputs. This is no good for our LED, we need the pin it is connected to to be an output, so type the command:

```
>>> GPIO.setup("P8_10", GPIO.OUT)
```

At this point, the LED should still be off. To turn it on, type:

```
>>> GPIO.output("P8_10", GPIO.HIGH)
```

Turn it off again by typing:

```
>>> GPIO.output("P8_10", GPIO.LOW)
```

Writing a Program

To make the LED flash, we are going to write a short Python program, so exit the Python Console by typing:

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

This should take you back to the Linux prompt.
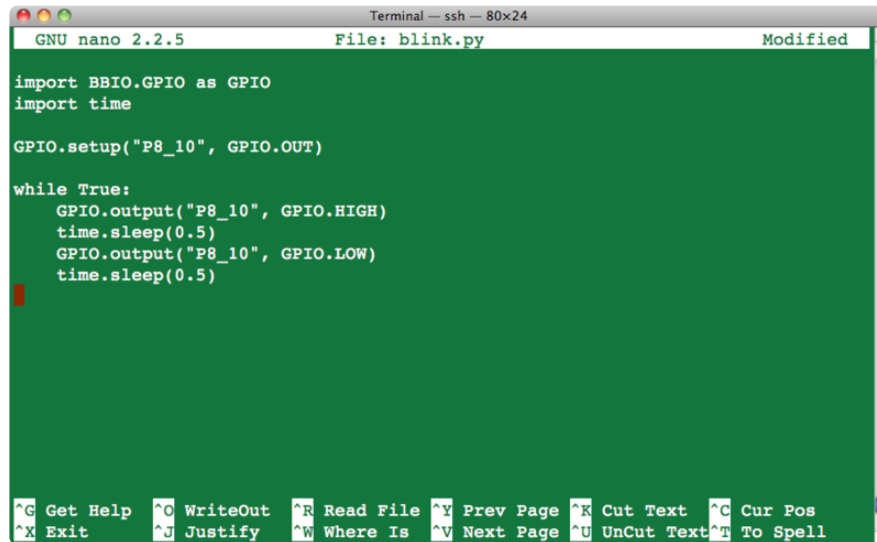Enter the following command to create a new files called blink.py

```
# nano blink.py
```

Now paste the code below into the editor window.

```
import Adafruit_BBIO.GPIO as GPIO
import time

GPIO.setup("P8_10", GPIO.OUT)

while True:
    GPIO.output("P8_10", GPIO.HIGH)
    time.sleep(0.5)
    GPIO.output("P8_10", GPIO.LOW)
    time.sleep(0.5)
```


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

To start the program, enter the command:

```
# python blink.py
```

When you want to stop the blinking, use CTRL-c to exit the program.

Next Steps

Try editing your program and changing the sleep period from half a second to 2 seconds and run the program again.

For extra credit, try controlling another LED on a different pin!

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 () at Adafruit.