

Search

 Translate

Content ▾ Participate ▾ Learn ▾ Discuss ▾ Build ▾ Find Products ▾ Store ▾

## BeagleBoard



# Controlling the BeagleBone Black's GPIO pins from the web using Drupal.

Posted by [CharlesGant](#) in [BeagleBoard](#) on Nov 26, 2014 10:47:39 PM



In my last tutorial I covered the [installation of Drupal onto the BeagleBone Black and how to setup an Apache-based web server](#). In this tutorial, I am going to continue building on that foundation and cover the basics of using that Drupal-based website to control the GPIO pins on the BeagleBone Black using PHP to run bash scripts. This will, in turn, allow you to use the internet to control anything that is connected to the BeagleBone Black's GPIO pins, and will further the foundation for some basic home automation tutorials I will write in the coming months.

### Prerequisites: What you need to know to be able to complete this project.

- Understand how Drupal-based websites work.
- Understand how PHP can be used to parse commands to the web server's terminal.
- Understand how LEDs work, and what the terms **Anode** and **Cathode** mean.
- Understand the basics of how Transistors work, and why they should be used for applications like this.
- Understand how the BeagleBone Black accesses its GPIO Pins.
- Basic knowledge of soldering.

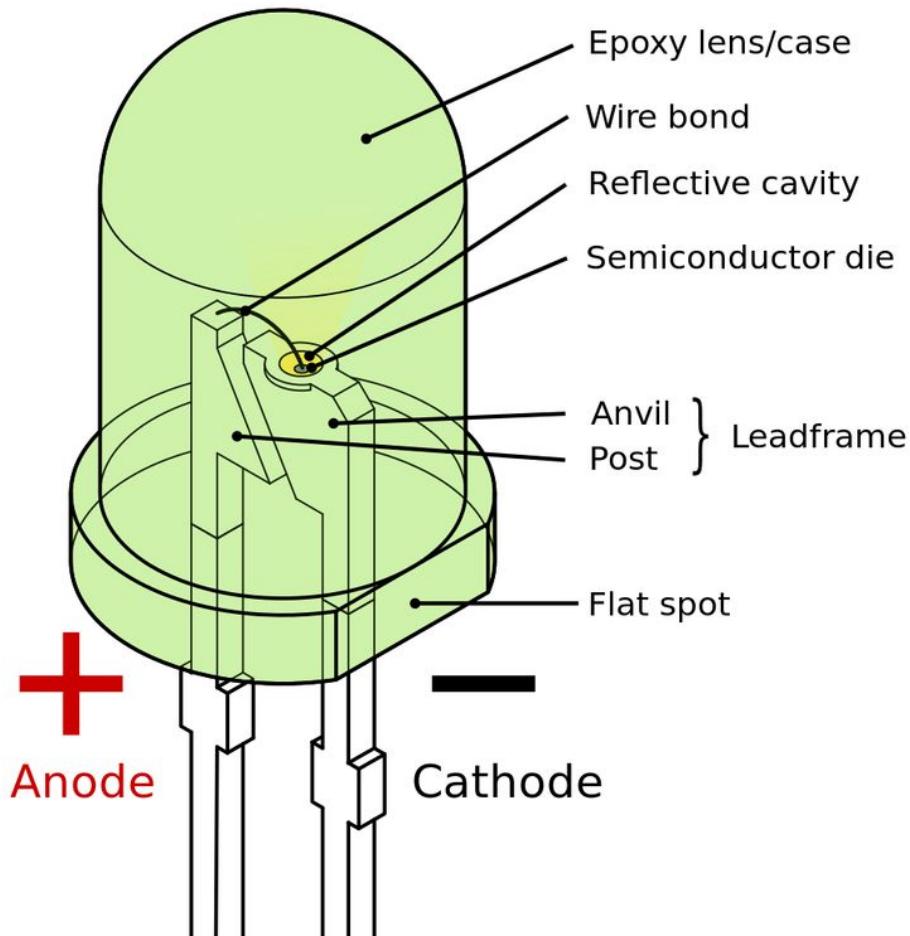
### Hardware Required: What hardware is needed to complete this project.

- [BeagleBone Black \(Rev C recommended\)](#), running the latest Debian image.
- 5mm Green LED
- 1x 460Ω 1/4w Resistor
- 1x 3.3kΩ 1/4w Resistors
- 1x 100Ω 1/4w Resistors
- 1x 2n3904 General Purpose NPN Transistors
- Female to Male Jumper Wires
- Male to Male Jumper Wires
- 5V Power Source for BeagleBone Black
- 5V Power Source for BeagleBone Black (Optional without BB View cape, Required if BB View Cape is used.)
- Solder

- Flux
- Soldering Iron

## Getting started: Soldering the resistor to the LED

For the purpose of this tutorial we are going to use an LED to indicate when a specific GPIO pin has been toggled on and off. As I am sure many of you reading this already know, Light Emitting Diodes (or LEDs) are one of the most revolutionary electronic components to have ever been created. In fact, this year's [Nobel Prize in Physics](#) was awarded to the team of scientists who created the efficient blue LED, which has revolutionized the way humans light the world. Since LEDs are so efficient, and sip very little power, the 3.3v coming out of the BeagleBone Black's GPIO pins would fry the LED if we tried to connect it without a current limiting resistor. While this is old-hat stuff for most of you, there will be some readers who will find this section interesting and helpful, and that is why I am including it. Below is a diagram that shows exactly what makes up your typical 5mm LED.

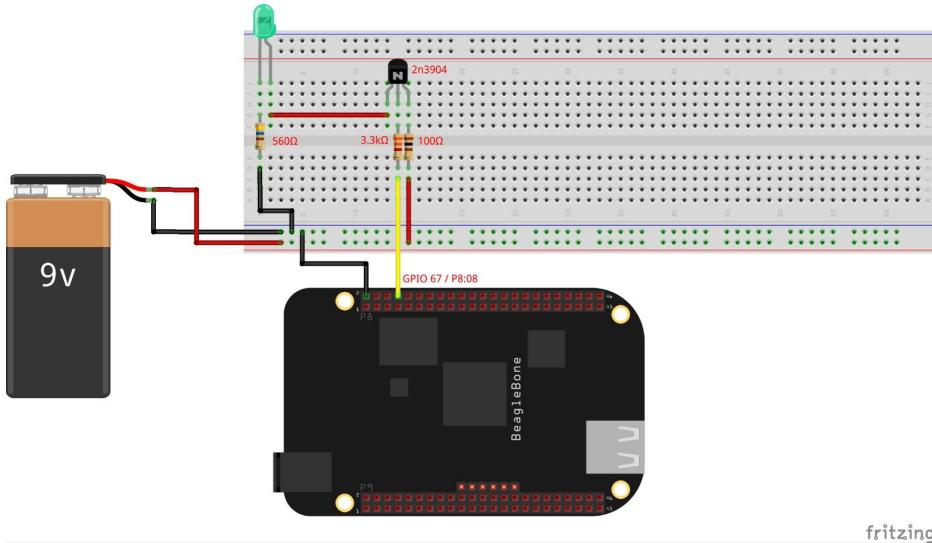


Since the LED is a diode, electricity will only flow one direction. This is why a LED is polarized, unlike a resistor that works in any orientation. In the diagram above, you will notice that the Anode (+ side) is the smaller element inside the LED, and that the Cathode is actually the leg that houses the crystal that emits the light. Another way to tell the Anode from the Cathode is that the Anode will usually be longer than the Cathode. Finally, the Cathode is usually closest to the flat spot on the LED's base ring.

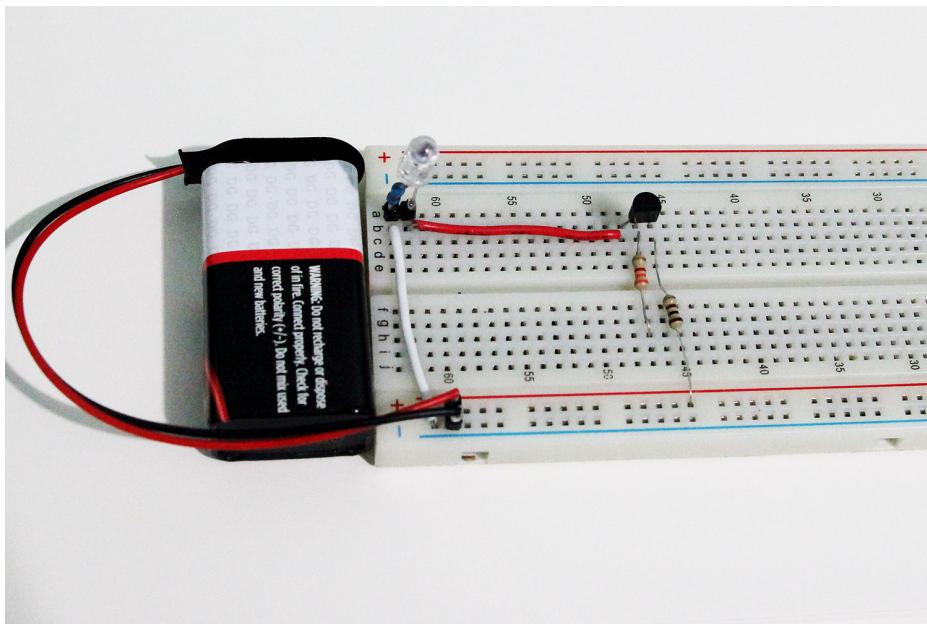
Placing the resistor on the Anode or Cathode will work, but I prefer to place it on the Cathode. This is just personal preference, but I find it to be considered best practice in several textbooks and guides. There is also no set standard for attaching a resistor directly to the leg of the LED, but I generally tend to go for the "wrap it around neatly" method. I also like to trim the Cathode down so that the resistor's lead matches up with the Anode leg length. For breadboard work I like to build easily reusable LED modules that include the resistor and are soldered to a 2-pin male header.



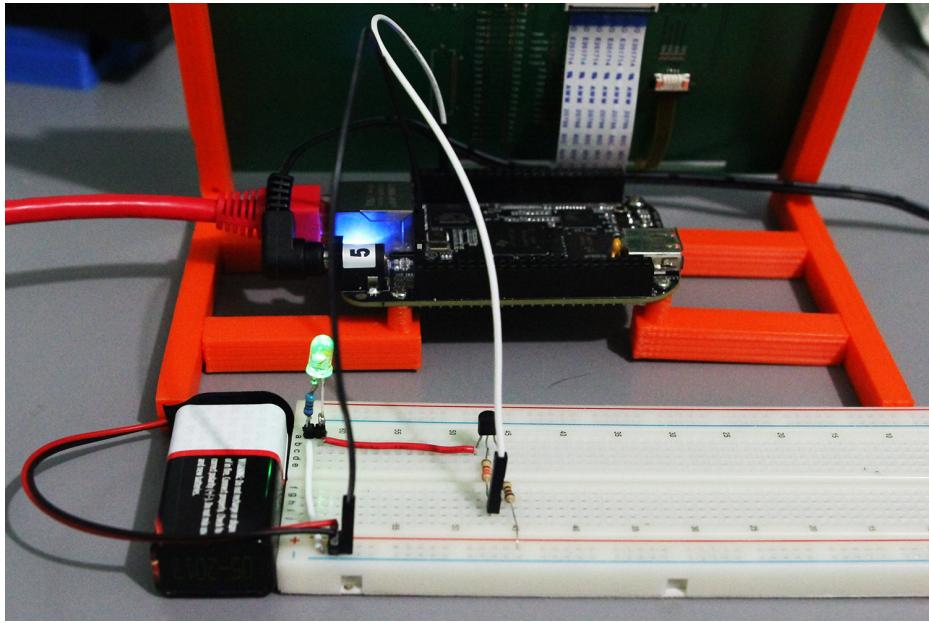
With that finished, we can now move on to setting up the transistor circuit on a breadboard. The transistor circuit is needed because the BeagleBone Black is only able to supply about 8mA to each of its GPIO pins and connecting something that draws more than 8mA can damage the board's ARM chip. We can turn on a general-use PNP transistor with just a few mA which will let us power the LED externally, and prevent any damage to the board. This is also handy for controlling relays and other components that can be triggered via the GPIO pins.



Here is a quick diagram of how I wired up the transistor circuit and attached the LED.



You will notice that I have a 9Volt battery powering the LED via the transistor. You do not have to use a 9V battery, and anything above 5V will work, but I would not use a power supply over 12v for this application.

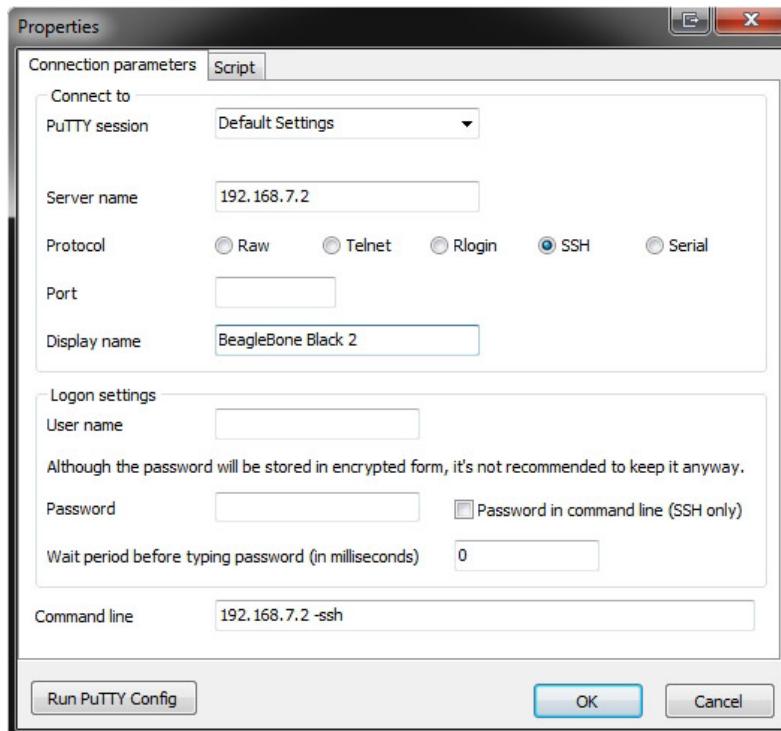


With the Transistor Circuit wired up, lets connect the BeagleBone Black. Basically we simply need to connect one of the GPIO Pins to the Transistor, and carry over the BeagleBone Black's ground circuit to our bread board. With all of this connected, we can now test the LED using the BeagleBone Black's terminal.

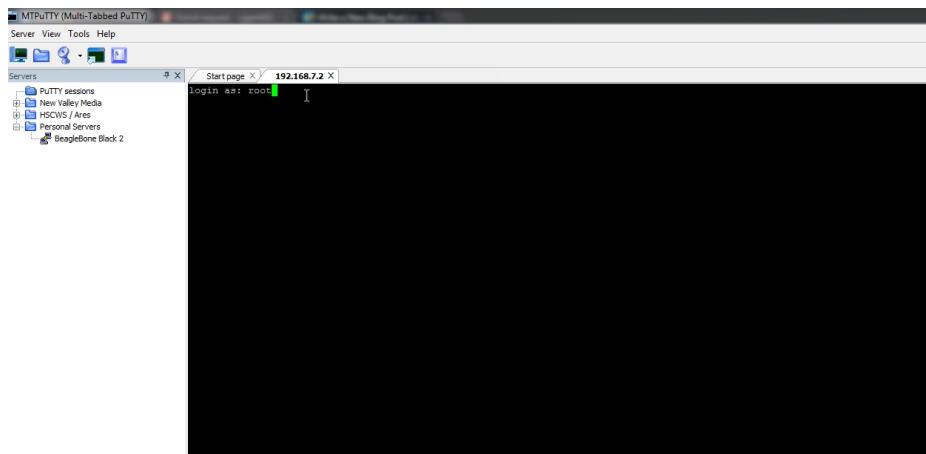
### Testing The Circuit: Using the terminal to trigger the GPIO Pin.

To test the circuit we need to access our BeagleBone Black via SSH. I covered how to do this in the [BeagleBone Black RSS News Reader tutorial](#), but will post it here as well.

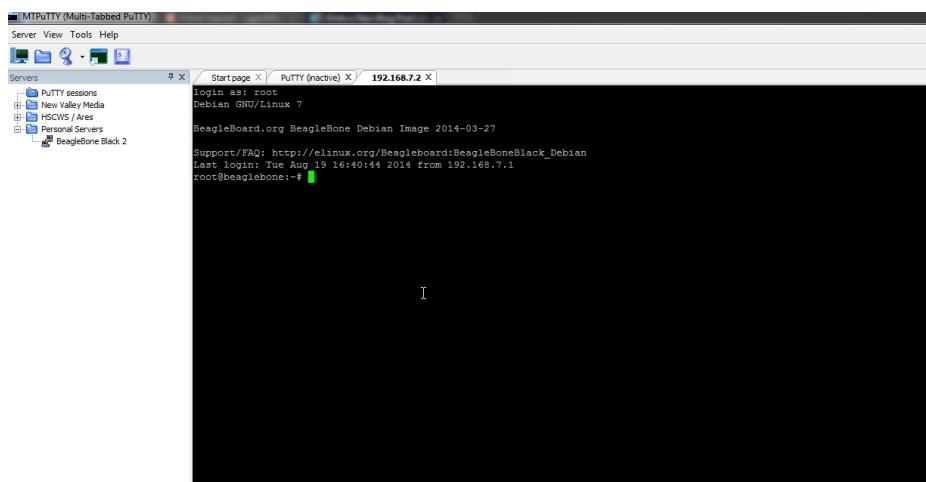
To access the BeagleBone Black via SSH I will be using a terminal for Windows called Putty, as well as an add-on called MTPutty, which allows me to run multiple putty instances from within a single tabbed interface. Both Putty and MTPutty are available for free, and take just a few moments to download and install. Chances are, if you are reading this, you already know how to use Putty. If you are using a Mac you can simply use the built-in Terminal to access your Black Via SSH. To connect to your Black, you need to enter the following credentials.



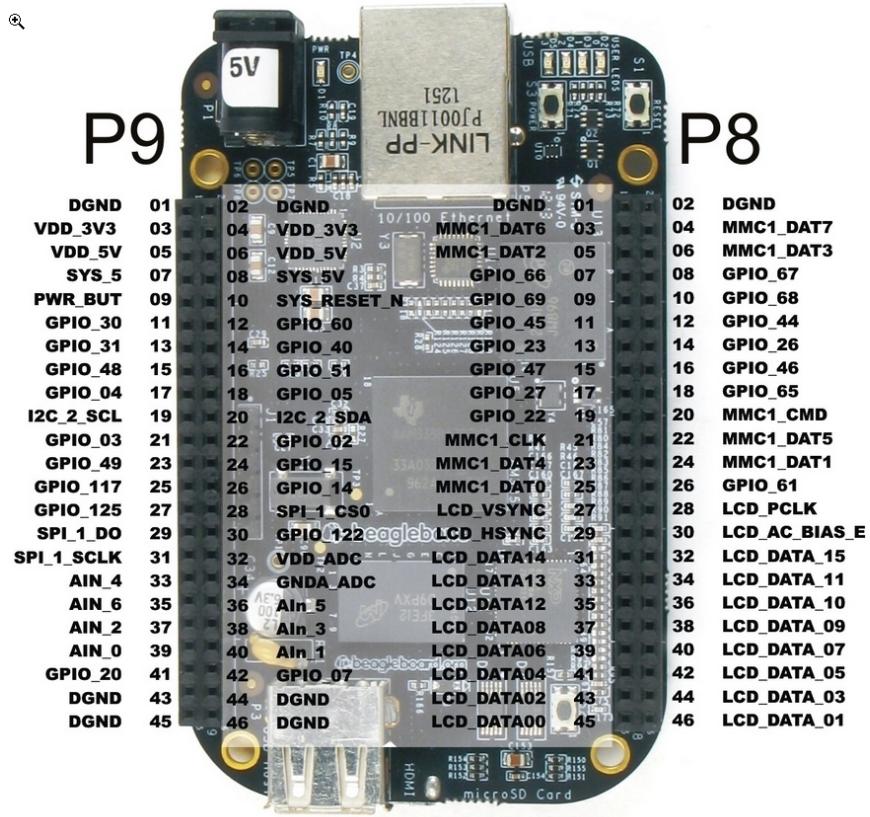
With these settings configured, click ok and then double click the BeagleBone Black name in the server list. This will open up a new terminal tab and you will be prompted to enter a username. Unless you have modified the root user, you should enter **root** as the user name.



BeagleBone has made it easy here as there is no password and after entering the username, the connection will finish, and you will be placed in your "home" directory.



Now that you are SSHed into the BeagleBone Black we will need to prepare GPIO 67 (P8:08). The pinout diagram can be seen below.



Before we get started with testing, lets take a moment and talk about how the BeagleBone Black's GPIO pins are addressed by the Linux-based Debian operating system. The BeagleBone Black addresses its GPIO Pins by reading the contents of files which are located in the `/sys/class/gpio` directory in the board's operating system. The most basic functionality is to write the pin high or low, and is what we will be doing in this tutorial.

To use a GPIO Pin you first need to tell Debian that you are going to use a particular pin in a specific GPIO header. The following code uses GPIO 8.8 as an example, but any GPIO Pin can be accessed by creating these files. Enter the following commands into the BeagleBone Black's terminal.

The first thing we need to do is create a new directory called `gpio67` in the `/sys/class/gpio`/`sys/class/gpio` directory.

```
+ expand source view plain
01. echo 67 > /sys/class/gpio/export
```

At this point you will need to let the BeagleBone Black know whether or not the specified GPIO pin is input or output.

```
+ expand source view plain
01. echo out > /sys/class/gpio/gpio67/direction
```

With this command, the BeagleBone Black no longer supplies 2.3V to the header and it is considered to be turned off. In order for us to turn on GPIO Pin 67 we need to set its value to 1. To turn it back off again, the value needs to be 0. We can do that with the following commands.

#### Set the GPIO Pin High

```
+ expand source view plain
01. echo 1 > /sys/class/gpio/gpio67/value
```

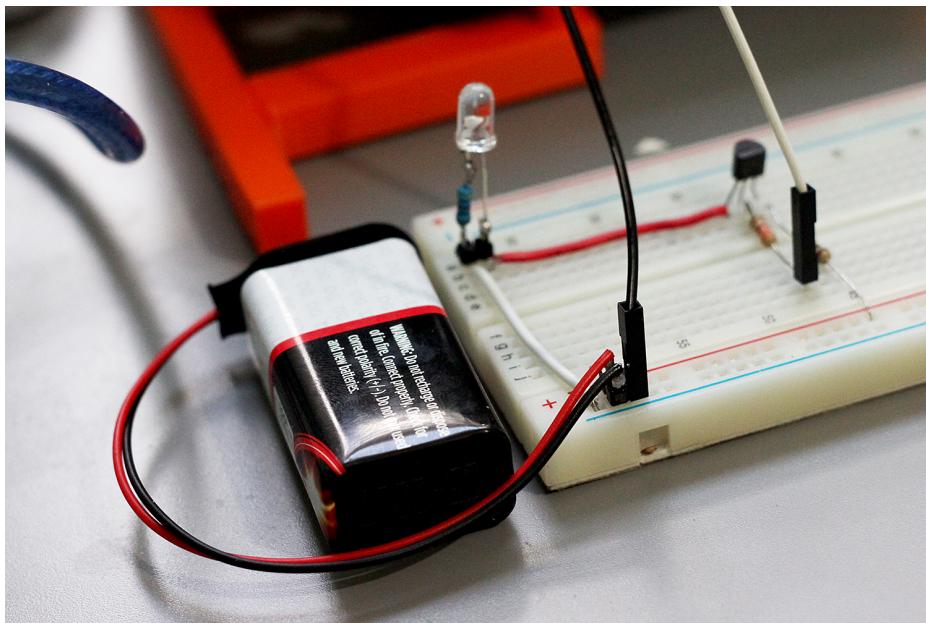
#### Set the GPIO Pin Low

```
+ expand source view plain
01. echo 0 > /sys/class/gpio/gpio67/value
```

#### Remove control of the GPIO Pin when finished.

```
+ expand source view plain
01. echo 67 > /sys/class/gpio/unexport
```

With those basics out of our way, let's move on to testing. Since we already have GPIO Pin 67 setup, and the rest of the circuit above is in order, we just need to give our breadboard power via a 9-volt battery. Connect the 9-volt to the breadboard as shown in the image below.

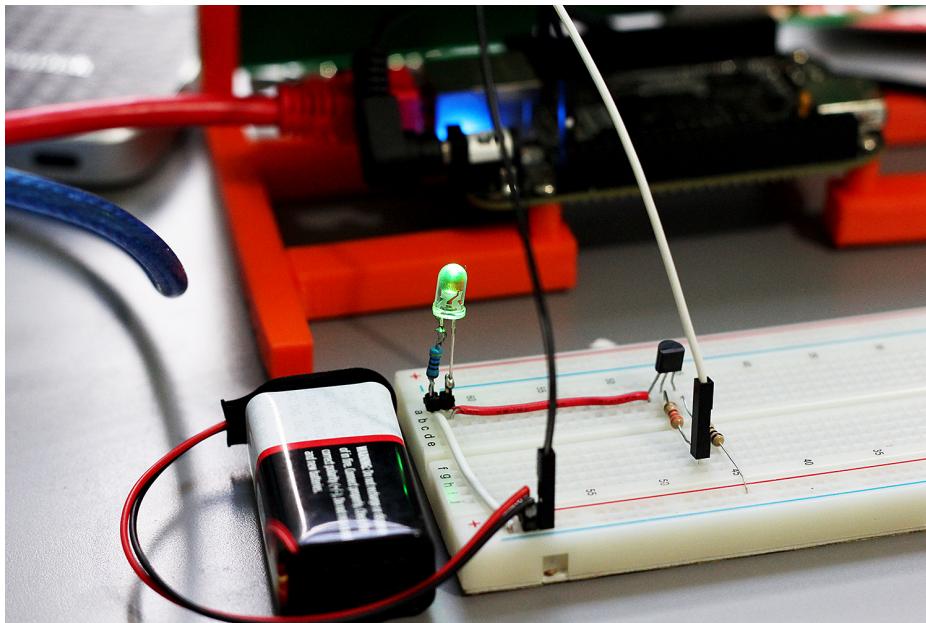


With everything connected, we should be able to use the command line to set GPIO Pin 67 high, which should in turn activate the transistor and allow the LED to light up.

### Set the GPIO Pin High

```
+ expand source view plain
01. echo 1 > /sys/class/gpio/gpio67/value
```

If everything was followed correctly, you will see the LED light up



Now set the GPIO Pin to low to turn off the circuit.

### Set the GPIO Pin Low

```
+ expand source view plain
```

```
01. echo 0 > /sys/class/gpio/gpio67/value
```

## Configuring The Website: Using the internet to control the GPIO Pin.

If you have been following along in my BeagleBone Black series you will recall that [my last tutorial demonstrated how to install an apache-based web server on the BeagleBone Black](#), and how to install and configure a Drupal-based website on top of that web server. For this next part, you will need to follow the steps in that tutorial and get a Drupal 7 website up and running on your BeagleBone Black before continuing with this tutorial.

Pull up the previously created website, and login as the administrator by visiting the [`\(my\_website\_url\)/user`](#).

There are a few different ways that we can build functionality into our website to control the GPIO pins on the BeagleBone Black. Note that the following methods only work when the website is hosted on the BeagleBone Black. We are going to begin with the two simplest methods. Method one will create two separate pages that run a php script when visited, while method two creates toggle buttons within a single page or block. While these methods work, they are highly vulnerable to security breaches, and should only be used on sites that are not directly open to the public.

### Method One: Controlling the GPIO via separate web pages.

Since the BeagleBone Black GPIO Pins are accessible using the command line, we can write a simple PHP script that turns any of the pins on and off using the commands listed above. First, we need to turn on the PHP filter in Drupal. (Note that turning this filter on in Drupal exposes your website to security vulnerabilities, and as such it should only be done on sites not publicly available) Navigate to [`\(my\_website\_url\)/admin/modules/`](#) and scroll down until you see the PHP Filter module under the Core heading. Enable this module by ticking the box to the left, then scroll to the bottom of the page and click save.

The screenshot shows the Drupal 7 modules configuration page. A red arrow points to the 'PHP Filter' module, which is highlighted with a red border. The module details are as follows:

- Name:** PHP Filter
- Version:** 7.32
- Description:** Allows embedded PHP code snippets to be evaluated.
- Requires:** Field (enabled), Field SQL storage (enabled)
- File:** modules/core/filter.module
- Path:** /modules/filter
- Module version:** 7.32
- Core:** Yes
- Enabled:** Yes
- Configuration:** /admin/config/filter/php-filter

### Creating the LED On Page

With the PHP filter enabled create a new basic page by navigating to [`\(my\_website\_url\)/node/add/page`](#). Fill out the page with the following information, or watch the video embedded below and follow along.

- **Title:** LED ON

- **Body:**

```
+ expand source view plain
01. <?php
02. exec("/bin/echo \"1\" > /sys/class/gpio/gpio67/value");
03. ?>
```

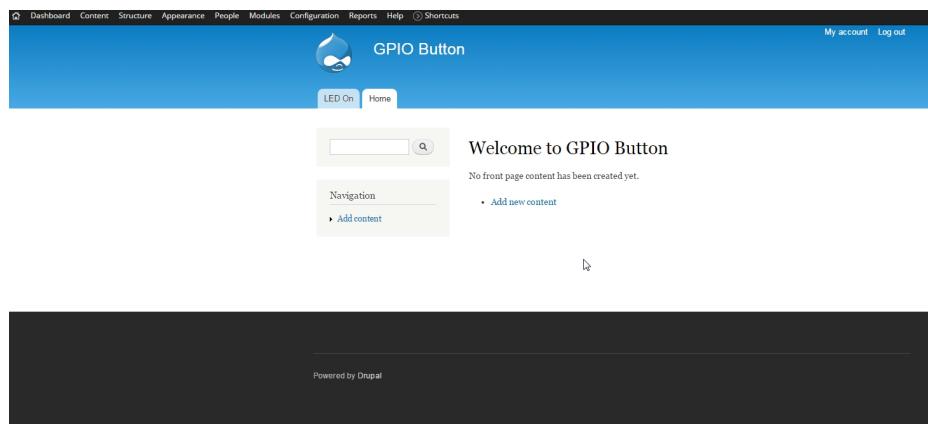
- **Text Format:** PHP Code

- **Menu Settings:** Tick the box for Provide Menu Link

- **Menu Link Title:** LED On
- **Parent Item:** <Main Menu>
- **Weight:** 1

- **Publishing Options:** Tick the box for "Published" and make sure the other two options are not checked.

Now click the "Save" button at the bottom of the page. If everything was entered correctly, you can return to the home page and there will be a new main menu link called "LED On."

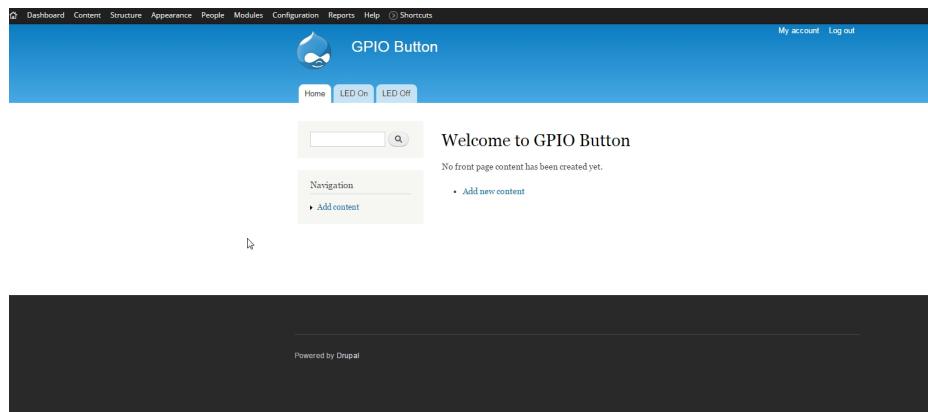


## Creating the LED Off Page

Now we need to create a web page that turns the LED off. Create a new basic page by navigating to `(my_website_url)/node/add/page`. Fill out the page with the following information, or watch the video embedded below and follow along.

- **Title:** LED Off
- **Body:**
  - ```
+ expand source  view plain
01. <?php
02. exec("/bin/echo \"0\" > /sys/class/gpio/gpio67/value");
03. ?>
```
- **Text Format:** PHP Code
- **Menu Settings:** Tick the box for Provide Menu Link
  - **Menu Link Title:** LED Off
  - **Parent Item:** <Main Menu>
  - **Weight:** 2
- **Publishing Options:** Tick the box for "Published" and make sure the other two options are not checked.

Now click the "Save" button at the bottom of the page. If everything was entered correctly, you can return to the home page and there will be a new main menu link called "LED Off."



## Setting Permissions

With both control pages created, we need to modify the permissions on the `gpio67/value` directory so that the `www-data` user (Apache) can write to it. Run the following command from within the terminal.

```
+ expand source  view plain
01. chmod 666 /sys/class/gpio/gpio67/value
```

With this done, we are able to test that everything works. With the transistor's base pin connected to GPIO67 and the 9-volt battery powering the breadboard, you should be able to click on the LED On menu link and the LED will turn on. Clicking the LED Off Menu Link will turn the LED Off. If your LED is not turning on and off, revisit the steps above. Alternatively, you can watch the video below and follow along.

### Building the LED On and Off pages for the BeagleBone



## Method TWO: Controlling the GPIO via toggle buttons on a single page or block.

While turning the LED on and off with separate web pages works, it is not the most efficient or best way to do things. This method could come in handy if you wanted to trigger a GPIO event whenever someone visits a webpage, but for general user triggered On and Off use, buttons would be a far better method. To trigger the GPIO pins via web buttons, we will utilize some existing code that I found during a quick search. This PHP code creates two basic buttons that fire off php scripts when pressed. You can view the original code at <http://www.berriman.co.uk/beaglebone-black-bbb-simple-web-io/>.

To make the code work we will need to modify two lines to point to the specific GPIO pin we are using to turn our LED circuit on and off. I have pasted the code you will need to use in the Page Building section below.

### Creating the LED On / Off Toggle Button Page

Now we need to create a web page that turns the LED off. Create a new basic page by navigating to *(my\_website\_url)/node/add/page*. Fill out the page with the following information.

- **Title:** LED On and Off Buttons

- **Body:**

```
1  <?php
2  /*
3  *
4  * Copyright 2013 Martin Berriman
5  *
6  * Redistribution and use with or without modification, are permitted
7  * provided that the following conditions are met:
8  *   1. Redistributions of source code must retain the above copyright
9  *      notice, this list of conditions and the following disclaimer.
10 *
11 *     * Edited by Charles Gantt for use on Element14
```

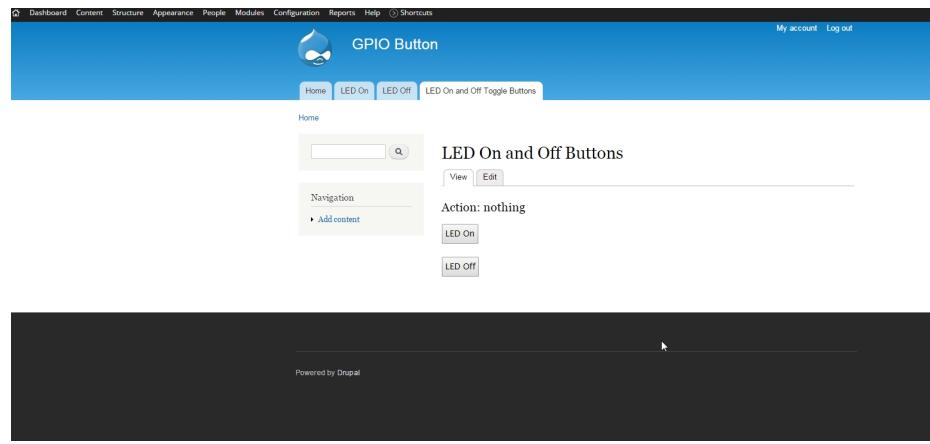
- **Text Format:** PHP Code

- **Menu Settings:** Tick the box for Provide Menu Link

- **Menu Link Title:** LED On and Off Toggle Buttons
  - **Parent Item:** <Main Menu>
  - **Weight:** 3

- **Publishing Options:** Tick the box for "Published" and make sure the other two options are not checked.

Now click the "Save" button at the bottom of the page. If everything was entered correctly, you can return to the home page and there will be a new main menu link called "LED On and Off Toggle Buttons." Click on that page and click the LED On button and then the LED Off Button. The LED should illuminate and then turn off. If this is correct, we can move on, otherwise revisit the steps above.



## Creating the LED On / Off Toggle Button Block

Create a new block by navigating to [\(my\\_website\\_url\)/admin/structure/block/add](http://mywebsiteurl/admin/structure/block/add). Fill out the block with the following information, or watch the video embedded below and follow along.

- **Title:** Access LED
- **Block Description:** GPIO67 LED Buttons
- **Body:**

```

1  <?php
2  /*
3  *
4  * Copyright 2013 Martin Berriman
5  *
6  * Redistribution and use with or without modification, are permitted
7  * provided that the following conditions are met:
8  * 1. Redistributions of source code must retain the above copyright
9  * notice, this list of conditions and the following disclaimer.
10 *
11 * Edited by Charles Gantt for use on Element14

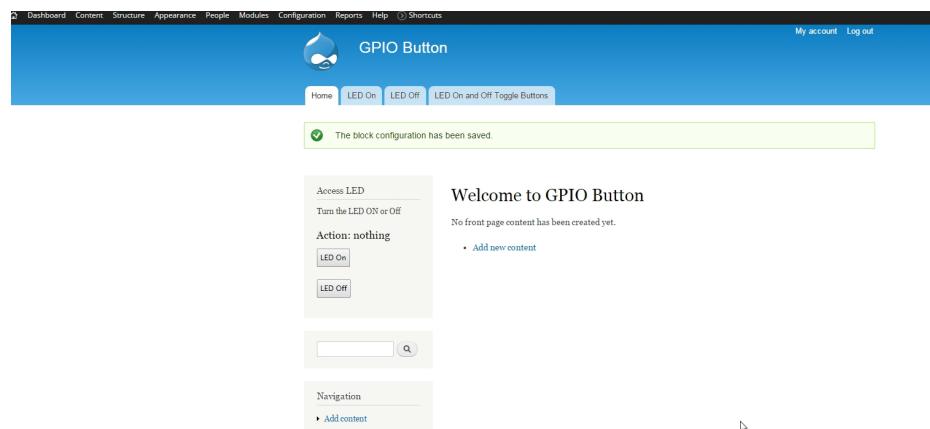
```

- **Text Format:** PHP Code

- **Region Settings:**

- **Bartic (Default Theme)**
  - **Sidebar First**

Now click save block. You can change the weight of this block by visiting the Block Administration page, and navigating to [\(my\\_website\\_url\)/admin/structure/block](http://mywebsiteurl/admin/structure/block). Now you will see the LED Toggle buttons when visiting any page on the website. The beauty of this block / button method is that you can now place your GPIO buttons anywhere on the website, and not have to display them as a page.



There are many ways to do this that are specific to Drupal, but the PHP code used in this last example will work on any web framework, and can be used independently as well. One of the safer ways to execute this code would be to set up a rule and use a flag to control when each script is run.

run. That is more advanced Drupal development work than I want to get into in this tutorial, but I would be glad to write a separate tutorial on it if there is enough interest. For the advanced Drupal developers out there, you can head over to my [Github and download a feature](#) module with this functionality built in. You will still have to create the node, and install some modules though. In the mean time, I urge everyone to play around with modifying this code, and making some awesome stuff.

## Building the LED Toggle Buttons for the BeagleBone B...



My next tutorial will involve modifying this code to trigger a set of relays, and notification LEDs that can be used to control anything from a basic lamp to a more full featured home automation system.

36006 Views

Tags: led, black, gpio, linux, beaglebone, debian, beaglebone\_black, bbb, beaglebone-black, drupal



### NEW! BeagleBone® AI

[Technical Specifications](#) | [Frequently Asked Questions](#) | [Comparison Chart](#) | [BeagleBone® Accessories](#) | [Getting Started](#) | [RoadTest](#)

Average User Rating

(0 ratings)

### 1 Comment

[Login or Register to comment](#)



gururajm Nov 7, 2016 2:30 PM

hi,

I purchased BeagleBone Black Industrial 4G is Red. I trying to power up using external power supply which is working perfectly with my old Beaglebone Black which has xAM3359AZCZ100 Black PCB board, where i can do ssh to the BBB but the same external power supply board if i connect to BeagleBone Black Industrial 4G is Red i cannot do ping/SSH no ethernet get up. Plz can any one suggest what wrong is there....

[Actions](#)

Like (0)

element14 is the first online community specifically for engineers. Connect with your peers and get expert answers to your questions.

[Content](#) | [Participate](#) | [Learn](#) | [Discuss](#) | [Build](#) | [Find Products](#)

Follow [element14](#)

| [Store](#)

[About Us](#) | [Feedback & Support](#) | [FAQs](#) | [Terms of Use](#) | [Privacy Policy](#)

| [Cookies](#) | [Sitemap](#)

Powered by

A Premier Farnell Company

© 2009-2019 Premier Farnell Ltd. All Rights Reserved. ICP 备案号 10220084.

Premier Farnell Ltd, registered in England and Wales (no 00876412), registered office: Farnell House, Forge Lane, Leeds LS12 2NE