Wiring an LED Guide for BeagleBone (Black/Green)

by Brian Fraser
Last update: March 9, 2023

Target Linux Kernel: 5.10+

This document guides the user through

1. Wiring an LED on P9.23 & controlling it via GPIO on BeagleBone Black/Green.
2. Controlling the LED via Linux kernel's LED support.

Table of Contents

Formatting

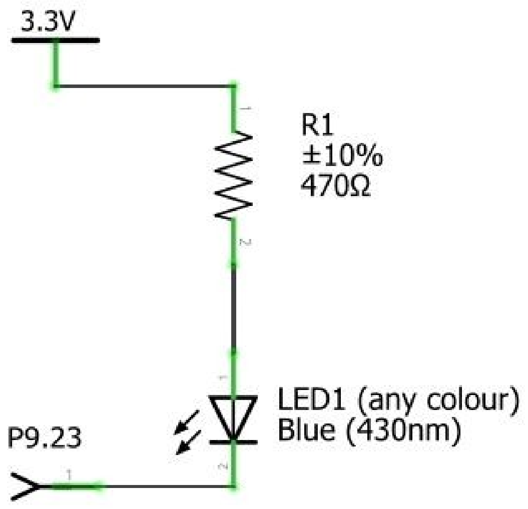
1. Host (desktop) commands starting with `(host)$` are Linux console commands:
   ```
   (host)$ echo "Hello world"
   ```
2. Target (board) commands start with (bbg)$:
   ```
   (bbg)$ echo "On embedded board"
   ```
3. Almost all commands are case sensitive.

Revision History

- Nov 2, 2019: Initial version for Kernel 4.9
- Mar 15, 2021: Updated prompt displays
- Mar 9, 2023: Updated .dts file to free up pins from universal cape on load (kernel 5.10).

1. Wiring an LED

1. **Critical Tips**
   - Do all your wiring and wiring changes with the power to the BeagleBone turned ***off*** (safest!)
   - Try to have some space between the components you place on the breadboard to prevent them from unexpectedly shorting (touching the wire leads). For example, on the breadboard, wire your resistor into column A, and your LED into column E so that they don't touch.
   - Note that it's OK for the plastic coated parts of the wires to touch; it's only the bare metal that can conduct electricity.
   - Avoid static! Before working on the components, try grounding yourself by touching some unpainted metal on your computer, such as the USB port.
2. Turn power off to BeagleBone.
3. If not done so already, place breadboard onto plastic mounting plate beside the BeagleBone.
   - Peal off the paper on the back of the breadboard to allow it to stick to the mounting plate.
4. Understand the LED circuit in Schematic 1:
   - P9.23 is the BealgeBone's GPIO port, configured as **output**.
   - When P9.23 is set to **high** (1) it's 3.3V (same voltage as voltage source on top) so no current flows through circuit. (Current needs a voltage differential to push it). Therefore the LED is off.
   - When P9.23 is set to **low** (0) it's 0V, so current flows from 3.3V to 0V, through the current limiting resistor. On the way, it flows through the LED and lights it up.
   - Therefore, this circuit is active-low.



Schematic 1: Active-low LED circuit.

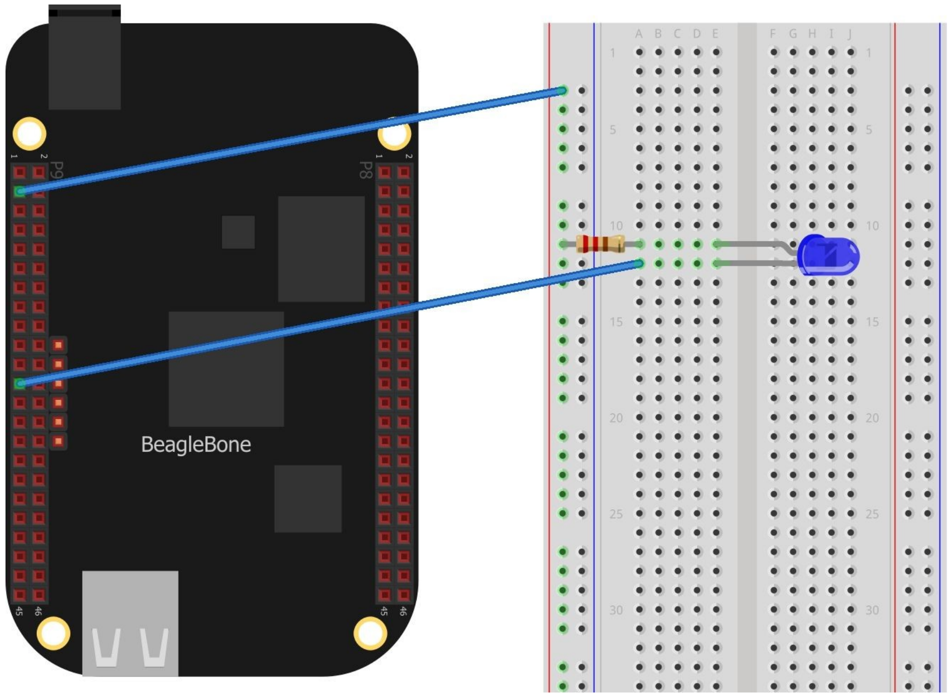


Diagram 1: BeagleBone and breadboard with LED wiring.

5. Wire up the circuit as shown in in Diagram 1 and photos at end of guide.
   In the diagram, the Ethernet connector on the BeagleBone is at the top; P9 is on the left.
   - Unplug power (micro-USB) from the BeagleBone.
   - Connect BBG's power pin (P9.3 is +3.3V) to + rail of your breadboard.
   - Insert your LED into two rows of the breadboard as shown.
     NOTE: Longer lead (end) of LED is where current must flow in; in the diagram it is on top, connecting to the resistor.
   - Connect 470 Ohm resistor (or similar resistance) between + rail of breadboard to longer lead of LED.
     It does not matter the orientation of the resistor.
   - Connect shorter lead of LED to P9.23 on BeagleBone via a jumper.
6. Troubleshooting:
   - Double check your circuit:
     - P9.3 is +3.3V, connected to + rail of your breadboard.
     - Resistor connects + rail of breadboard to longer lead of LED.
     - Jumper connects shorter lead of LED to P9.23.
   - Ensure you have your BeagleBone oriented the correct way (i.e., you're using P9 not P8).
   - Remember that for row 1 on the breadboard, holes a, b, c, d, and e are all shorted together.
     For example, you can connect the LED to a resistor by plugging the pin of the LED into row 17 pin e, and one end of the resistor into row 17 pin b.

2. Drive LED via GPIO

The LED is wired to P9.23, which is Linux GPIO number 49.

1. Boot your BeagleBone. Any kernel version should be OK.
   Note that when powering on, the LED may turn on (a little) dimly due to an internal pull-down resistor on the CPU when the pin is configured for input (default).
2. Enable the pin for GPIO if not already exported:
   ```
   (bbg)$ ls /sys/class/gpio/gpio49
   ```
   If found, continue on to next step; otherwise, export the pin:
   ```
   (bbg)$ echo 49 | sudo tee /sys/class/gpio/export
   ```
3. Change to folder
   ```
   (bbg)$ cd /sys/class/gpio/gpio49
   ```
4. Set direction
   ```
   (bbg)$ echo out > direction
   ```
5. Turn on/off: It is active low, so turn on with:
   ```
   (bbg)$ echo 0 > value
   ```
   And turn off with
   ```
   (bbg)$ echo 1 > value
   ```
6. Troubleshooting:
   - See the GPIO guide for more information on how to setup and troubleshoot GPIO pins.
   - If you turn on the LED and the LED does not light up, double check your circuit.

3. Device Tree: Enabling the LED

For Linux to treat the LED as an LED to be controlled via the kernel's LED support (vs a GPIO pin), we must load a device tree using the cape manage. **Do all the following on the <u>target</u>.**

1. Create a new folder to build the device tree overlay:
   ```
   (bbg)$ mkdir ~/ZenCapeLedDts
   (bbg)$ cd ~/ZenCapeLedDts
   ```
2. Still on the target, create the dts file:
   ```
   (bbg)$ nano LED_P9_23.dts
   ```
   - Copy the contents of the `LED_P9_23`.dts file (later section) into this file.
3. Compile the device tree source (.dts) file into a "blob overlay" file (.dtbo):
   ```
   (bbg)$ dtc -O dtb -o LED_P9_23-00A0.dtbo -b 0 -@ LED_P9_23.dts
   ```
4. Deploy the `.dtbo` file to the target's `/lib/firmware` folder:
   ```
   (bbg)$ sudo cp LED_P9_23-00A0.dtbo /lib/firmware
   ```
5. Load the device overlay in `/boot/uEnv.txt`
   - Copy the existing `uEnv.txt`:
     ```
     (bbg)$ sudo cp /boot/uEnv.txt /boot/uEnv-BeforeWireLed.txt
     ```
   - Edit `uEnv.txt`:
     ```
     (bbg)$ sudo nano /boot/uEnv.txt
     ```
   - Edit the Additional custom capes section.
     Below shows the setup you'll have if you are already loading the audio cape. If you are not loading these, you may comment out those lines

```
###Additional custom capes
uboot_overlay_addr4=/lib/firmware/BB-BONE-AUDI-02-00A0.dtbo
uboot_overlay_addr5=/lib/firmware/LED_P9_23-00A0.dtbo
```

- You can use `uboot_overlay_addr0` through `uboot_overlay_addr7` for any of the capes being loaded. If you use 0-3, it replaces the cape support for any automatically detected capes configured to load at that slot. If you have no physical capes connected (which are automatically detected), then you can use any of the 8 slots you like.
- You only need to load the capes you need.

6. Reboot the target
7. Troubleshooting
   - See the Audio guide's last section (on the course website) to recover from a corrupted `uEnv.txt`. You should be able to copy back the `/boot/uEnv-BeforeWireLed.txt` to restore your previous working state.
     (NOTE: Restore process not working on kernel 5.10 -- Mar 2023)

4. Driving the LEDs

You must have installed the LED driver (previous section) for this section to work.

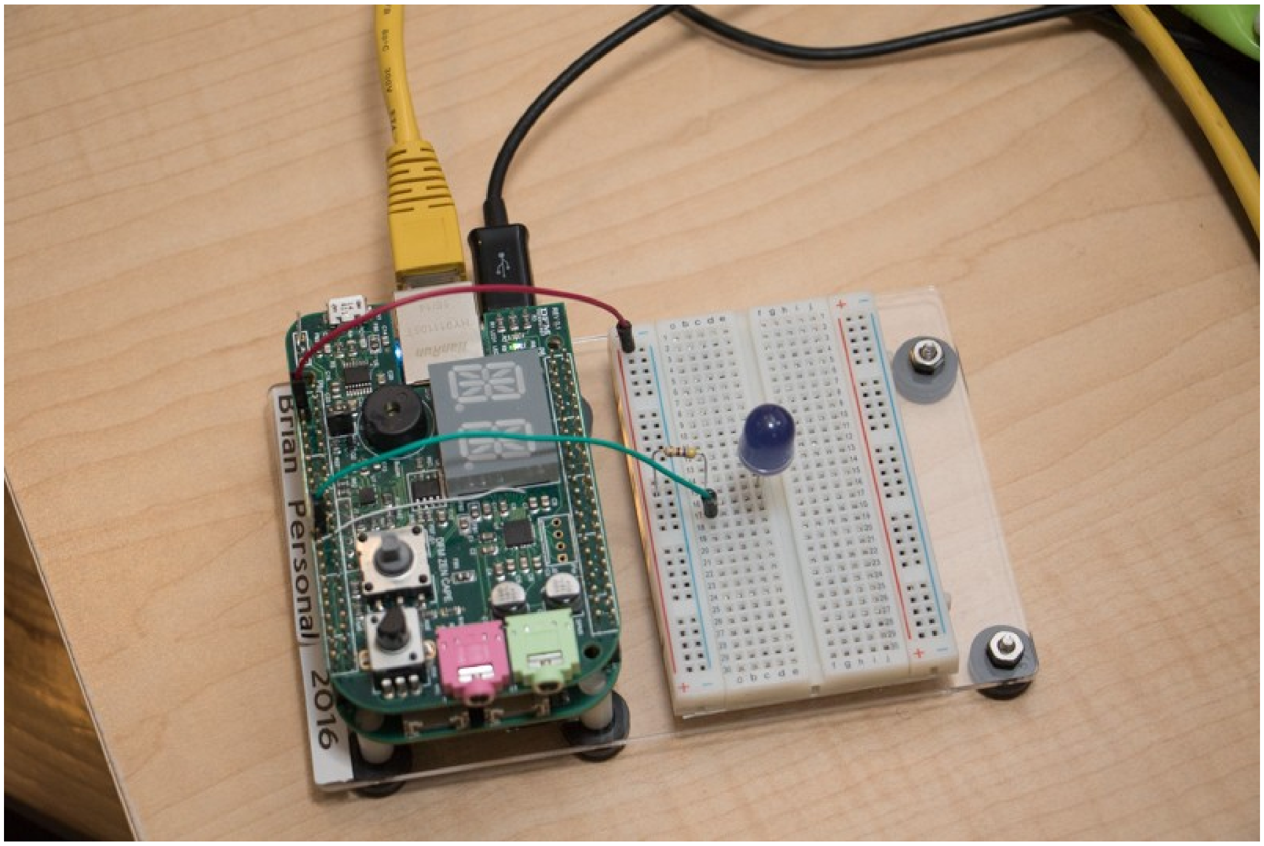
1. List all files in the `/sys/class/leds/` directory:
```
(bbg)$ ls /sys/class/leds/
beaglebone:green:usr0  beaglebone:green:usr1  beaglebone:green:usr2
beaglebone:green:usr3  leds:P9.23
```
2. Change to the new LED's folder:
```
(bbg)$ cd /sys/class/leds/leds\:P9.23
```
   - Note that you cannot just type ':' in a path, you must escape it.
3. Change the trigger to a heartbeat:
```
(bbg)$ echo heartbeat | sudo tee trigger
```
   - It should now be flashing. If not, double check your wiring.
4. Manually turn on/off the LED:
```
(bbg)$ echo none | sudo tee trigger
(bbg)$ echo 1 | sudo tee brightness
(bbg)$ echo 0 | sudo tee brightness
```
   - Note that the device tree was created for an active low LED circuit. Therefore, the logic for controlling this LED is not inverted (i.e., writing a 1 turns on, 0 turns off).

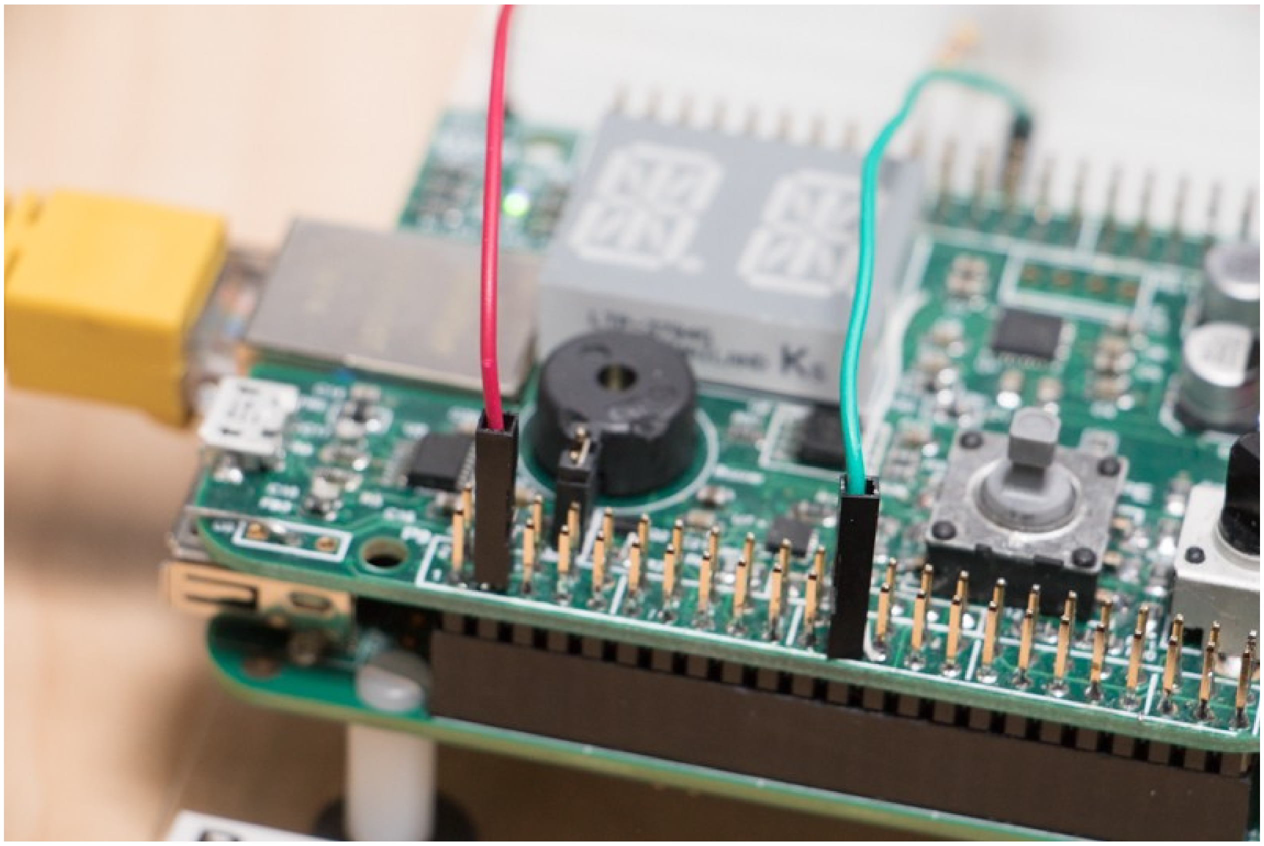
5. Device Tree Source (LED_P9_23.dts)

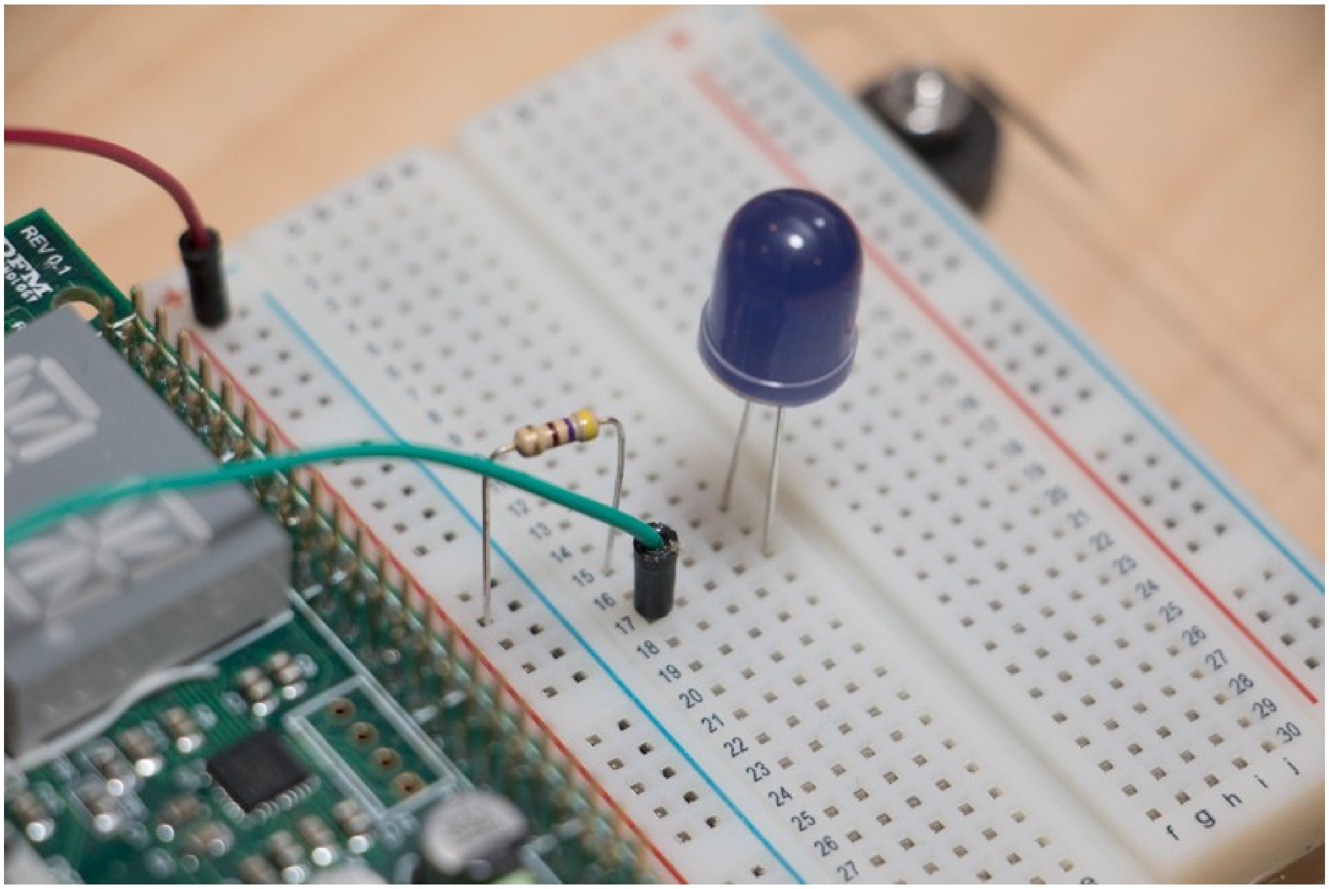
Here is the device tree source file used. You need not know its internal structure.

```
/*
* Configure P9_23 ("gpio1_17", Linux pin #47) to be an LED in Linux
* Based on: https://github.com/nomel/beaglebone/blob/master/led-header/generated/led-P9.23-00A0.dts
* Written by Brian Fraser; released under GPL and BSD.
* Updates:
*   Nov 2017 for Kernel 4.4+: Change "gpio2" to "gpio1" for gpios entry.
*   Nov 2019 for Kernel 4.9+: Change to uEnv.txt from cape manager
*   Feb 2023 for Kernel 5.10+ (Bullseye): Free up the pins used by the universal cape
*
* Compile with:
*    dtc -O dtb -o LED_P9_23-00A0.dtbo -b 0 -@ LED_P9_23.dts
* Copy the .dtbo to /lib/firmware
*    sudo cp LED_P9_23-00A0.dtbo /lib/firmware
* Load in /boot/uEnv.txt
*    uboot_overlay_addr4=/lib/firmware/LED_P9_23-00A0.dtbo
* Use like any Linux LEDs via /sys/class/leds/leds:P9.23/....
*/
/dts-v1/;
/plugin/;
/ {
     compatible = "ti,beaglebone", "ti,beaglebone-black";

     /* identification */
     part-number = "led-P9.23";
     /* version = "00A0"; */

     /* state the resources this cape uses */
     exclusive-use =
          "P9.23",    /* pin header uses this  */
          "gpio1_17"; /* hardware IP uses this */

     /* Free up the pins used by the cape from the pinmux helpers. */
     fragment@0 {
          target = <&ocp>;
          __overlay__ {
                P9_23_pinmux { status = "disabled"; };   /* P9_23: GPIO output for LED control */
          };
     };

     /* rxDisable_pullNone state */
     fragment_1 {
          target = <&am33xx_pinmux>;
          __overlay__ {
                gpio_P9_23_rxDisable_pullNone: pinmux_gpio_P9.23_rxDisable_pullNone {
                   pinctrl-single,pins = <
                        0x44 0xf
                   >;
                };
          };
     };

     fragment_2 {
          target = <&ocp>;
          __overlay__ {
                led_P9.23_helper {
                   compatible = "gpio-leds";
                   pinctrl-names = "default";
                   pinctrl-0 = <&gpio_P9_23_rxDisable_pullNone>;

                   leds-P9.23 {
                        label = "leds:P9.23";
                        gpios = <&gpio1 17 1>; /* flag 1 means active low */
                        linux,default-trigger = "heartbeat";
                        default-state = "off";
                   };
                };

          };
     };
};
```

6. LED Wiring Photos






REV 0.1
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
f
g
h
i
j