Wiring an LED Guide for BeagleBone (Black/Green)

by Brian Fraser
Last update: November 16, 2017

Target Linux Kernel: 4.4

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:

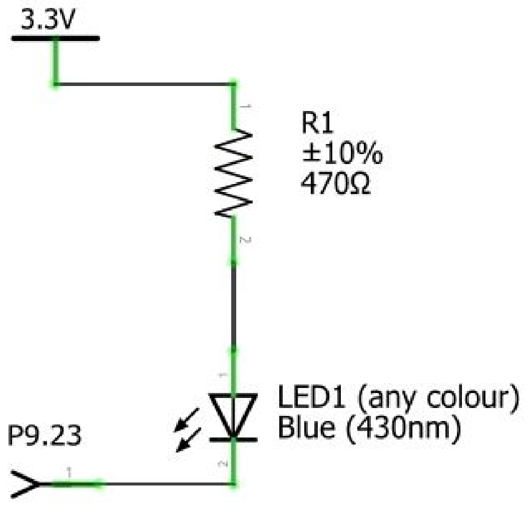
1. Host (desktop) commands starting with `$` are Linux console commands:
   ```
   $ echo "Hello world"
   ```
2. Target (board) commands start with #:
   ```
   # echo "On embedded board"
   ```
3. Almost all commands are case sensitive.

Revision History:

- Nov 16: Initial version for Kernel 4.4.

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.
   - 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.

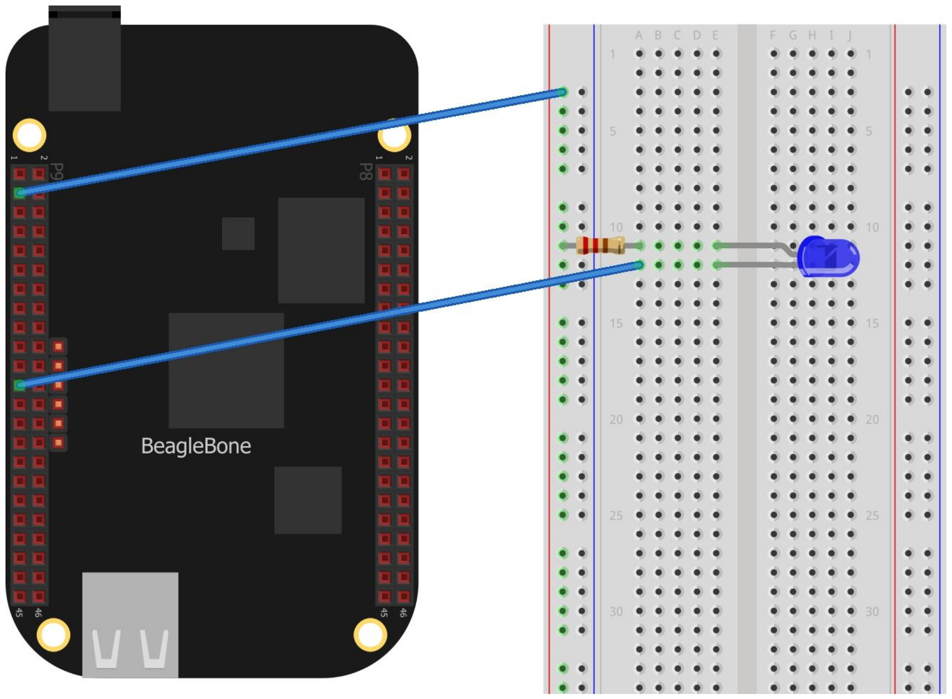


Diagram 1: BeagleBone and breadboard with LED wiring.

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 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 (need not be 4.4+).
   Note that when powering on, the LED may turn on 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:
   ```
   # echo 49 > /sys/class/gpio/export
   ```
3. Change to folder
   ```
   # cd /sys/class/gpio/gpio49
   ```
4. Set direction
   ```
   # echo out > direction
   ```
5. Turn on/off: It is active low, so turn on with:
   ```
   # echo 0 > value
   ```
   And turn off with
   ```
   # 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. Reboot your beaglebone and load a Linux 4.4+ kernel; if you previously had the pin mapped for GPIO it can interfere with the LED driver, even if you unexport the pin.
2. Download the file `LED_P9_23.dts` from the course website:
   ```
   # wget http://www.cs.sfu.ca/CC/433/bfraser/other/guide-code/LED_P9_23.dts
   ```
   - This requires an internet connection. Alternatively, you could `wget` the file on the host and then copy it to the target via NFS.
3. Compile the device tree source (.dts) file into a "blob object" file (.dtbo):
   ```
   # 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:
   ```
   # cp LED_P9_23-00A0.dtbo /lib/firmware
   ```
5. Using the cape manager, enable this cape:
   ```
   # export SLOTS=/sys/devices/platform/bone_capemgr/slots
   # echo LED_P9_23 > $SLOTS
   ```
   - You can see what happened by checking `dmesg`:

```
# dmesg
[..] bone_capemgr bone_capemgr: part_number 'LED_P9_23', version 'N/A'
[..] bone_capemgr bone_capemgr: slot #4: override
[..] bone_capemgr bone_capemgr: Using override eeprom data at slot 4
[..] bone_capemgr bone_capemgr: slot #4: 'Override Board Name,00A0,Override Manuf,LED_P9_23'
[..] bone_capemgr bone_capemgr: slot #4: dtbo 'LED_P9_23-00A0.dtbo' loaded; overlay id #0
```

6. Check listed capes (slot numbers may vary):
   ```
   # cat $SLOTS
    0: PF----  -1
    1: PF----  -1
    2: PF----  -1
    3: PF----  -1
    4: P-O-L-   0 Override Board Name,00A0,Override Manuf,LED_P9_23
   ```
7. Troubleshooting
   - If the cape overlay loads but you are unable to control the LED, try rebooting the board, and ensure you are loading a v4.4+ Linux kernel. This overlay may not work with older kernels (such as the one that may be pre-installed on your BeagleBone).
     - Check kernel version with:
       ```
       # uname -r
       4.4.95-bfraser-bone19
       ```
   - Check `dmesg` for any error messages when loading the cape overlay.

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:
   ```
   # 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:
   ```
   # 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:
   ```
   # echo heartbeat > trigger
   ```
   - It should now be flashing. If not, double check your wiring.
4. Manually turn on/off the LED:
   ```
   # echo none > trigger
   # echo 1 > brightness
   # echo 0 > 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.
*
* Updated Nov 2017 for Kernel 4.4: Change "gpio2" to "gpio1" for gpios entry.
*
* Compile with:
*    dtc -O dtb -o LED_P9_23-00A0.dtbo -b 0 -@ LED_P9_23.dts
* Copy the .dtbo to /lib/firmware
*    cp LED_P9_23-00A0.dtbo /lib/firmware
* Load with:
*    echo LED_P9_23 > /sys/devices/platform/bone_capemgr/slots
* 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 =
          /* the pin header uses */
          "P9.23",

          /* the hardware IP uses */
          "gpio1_17";


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

     fragment@1 {
          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 = "none";
                        default-state = "off";
                   };
                };

          };
     };

};
```

6. LED Wiring Photos