

Interaction with user LEDs and GPIOs on BBB

Name: Shreya Mamadapur
Instructor: Takis Zourntos

Student ID: C0774035

Introduction

This document shows how to interact(Turn ON/OFF) with user LEDs on the BBB and also shows how to configure GPIOs using sysfs.

User LEDs

As we know, in Linux things like hardware systems also show up as files. So we can manipulate the file contents to control our hardware like BBB.

There are 4 user LEDs on BBB. We can see the details of these by cd'ing to /sys/class/leds. We will need root privileges for this. So, log-in as root. The files that show up when you go to the specific usrled are:

brightness device max_brightness power subsystem trigger uevent

If we cat trigger, we can know by what the led is being triggered.

For example, usr1 is being triggered by [mmc0]

```
root@beaglebone:/sys/class/leds/beaglebone:green:usr1# cat trigger
none rc-feedback rfkill-any kbd-scrolllock kbd-numlock kbd-capslock kbd-kanalock kbd-shiftlock kbd-altgrlock kbd-ctrllock kbd-altlock kbd-shiftllock kbd-shiftrlock kbd-ctrlllock kbd-ctrlrlock usb-gadget usb-host [mmc0] mmc1 timer oneshot disk-activity ide-disk mtd nand-disk heartbeat backlight gpio cpu cpu0 activity default-on panic netdev
root@beaglebone:/sys/class/leds/beaglebone:green:usr1#
```

The default trigger to user LEDs:

usr0: heartbeat

usr1: mmc0

usr2: cpu0

usr3: mmc1

It must be noted that some of these are to imply to the user that the BBB is up and running and all the blink patterns have their significance.

If, however one wants to control these LEDs, we can echo "timer" to the trigger file. This adds delay_on and delay_off files to the beaglebone\green\:usrx directory. Also, when this echo command is written, the led starts blinking with a 50% duty-cycle.

Now we can set binary numbers in these to files to control the ON and OFF period of the desired LED.

For an example, let's turn ON the user-led 1 forever.

Interaction with user LEDs and GPIOs on BBB

```
shreya@ShreyasPC: ~
root@beaglebone:/sys/class/leds/beaglebone:green:usr1# ls -l
total 0
-rw-rw-r-- 1 root gpio 4096 Oct 1 13:55 brightness
lrwxrwxrwx 1 root gpio 0 Oct 1 13:55 device -> ../../../../leds
-r--r--r-- 1 root gpio 4096 Oct 1 13:55 max_brightness
drwxrwxr-x 2 root gpio 0 Oct 1 13:55 power
lrwxrwxrwx 1 root gpio 0 Oct 1 13:55 subsystem -> ../../../../class/leds
-rw-rw-r-- 1 root gpio 4096 Oct 1 13:55 trigger
-rw-rw-r-- 1 root gpio 4096 Oct 1 13:55 uevent
root@beaglebone:/sys/class/leds/beaglebone:green:usr1# echo timer > trigger
root@beaglebone:/sys/class/leds/beaglebone:green:usr1# ls -l
total 0
-rw-rw-r-- 1 root gpio 4096 Oct 1 13:55 brightness
-rw-r--r-- 1 root root 4096 Oct 1 14:53 delay_off
-rw-r--r-- 1 root root 4096 Oct 1 14:53 delay_on
lrwxrwxrwx 1 root gpio 0 Oct 1 13:55 device -> ../../../../leds
-r--r--r-- 1 root gpio 4096 Oct 1 13:55 max_brightness
drwxrwxr-x 2 root gpio 0 Oct 1 13:55 power
lrwxrwxrwx 1 root gpio 0 Oct 1 13:55 subsystem -> ../../../../class/leds
-rw-rw-r-- 1 root gpio 4096 Oct 1 14:53 trigger
-rw-rw-r-- 1 root gpio 4096 Oct 1 13:55 uevent
root@beaglebone:/sys/class/leds/beaglebone:green:usr1# echo 0 > delay_off
root@beaglebone:/sys/class/leds/beaglebone:green:usr1#
```

GPIOs:

We can access the gpios from P8 and P9 headers to interface any external circuitry. The details of GPIOs can be found in the directory /sys/class/gpio. We get a bunch of gpiobuttons and export/unexport files here. Export/unexport files are used basically to enable/disable the desired gpio. If our desired gpio is not listed, we can enable it by using `$echo gpiobutton > export`

Note that gpiobutton is not equal to pin number or the gpio_name. A gpiobutton can be calculated as:

$$\text{gpio1_20} = (1 \times 32) + 20 = 52$$

An example to change the direction of gpio1_20 from in to out (making it an output pin).

```
root@beaglebone:/sys/class/leds# cd ../gpio/
root@beaglebone:/sys/class/gpio# ls
export gpio114 gpio15 gpio27 gpio35 gpio45 gpio51 gpio67 gpio73 gpio8 gpio9
gpio10 gpio115 gpio19 gpio3 gpio36 gpio46 gpio60 gpio68 gpio74 gpio80 gpiochip0
gpio11 gpio116 gpio2 gpio30 gpio37 gpio47 gpio61 gpio69 gpio75 gpio81 gpiochip32
gpio110 gpio117 gpio20 gpio31 gpio38 gpio48 gpio62 gpio7 gpio76 gpio86 gpiochip64
gpio111 gpio12 gpio22 gpio32 gpio39 gpio49 gpio63 gpio70 gpio77 gpio87 gpiochip96
gpio112 gpio13 gpio23 gpio33 gpio4 gpio5 gpio65 gpio71 gpio78 gpio88 unexport
gpio113 gpio14 gpio26 gpio34 gpio44 gpio50 gpio66 gpio72 gpio79 gpio89
root@beaglebone:/sys/class/gpio# echo 52 > export
root@beaglebone:/sys/class/gpio# ls
export gpio114 gpio15 gpio27 gpio35 gpio45 gpio51 gpio66 gpio72 gpio79 gpio89
gpio10 gpio115 gpio19 gpio3 gpio36 gpio46 gpio52 gpio67 gpio73 gpio8 gpio9
gpio11 gpio116 gpio2 gpio30 gpio37 gpio47 gpio60 gpio68 gpio74 gpio80 gpiochip0
gpio110 gpio117 gpio20 gpio31 gpio38 gpio48 gpio61 gpio69 gpio75 gpio81 gpiochip32
gpio111 gpio12 gpio22 gpio32 gpio39 gpio49 gpio62 gpio7 gpio76 gpio86 gpiochip64
gpio112 gpio13 gpio23 gpio33 gpio4 gpio5 gpio63 gpio70 gpio77 gpio87 gpiochip96
gpio113 gpio14 gpio26 gpio34 gpio44 gpio50 gpio65 gpio71 gpio78 gpio88 unexport
root@beaglebone:/sys/class/gpio# cd gpio52
root@beaglebone:/sys/class/gpio/gpio52# ls
active_low device direction edge label power subsystem uevent value
root@beaglebone:/sys/class/gpio/gpio52# cat direction
in
root@beaglebone:/sys/class/gpio/gpio52# echo out > direction
root@beaglebone:/sys/class/gpio/gpio52# cat direction
out
root@beaglebone:/sys/class/gpio/gpio52#
```

Interaction with user LEDs and GPIOs on BBB

If we have an external circuit for an LED or relay/motor etc. We can turn it on by echoing "1" to the value file.

To set the value high,

```
root@beaglebone:/sys/class/gpio/gpio52# cat value
0
root@beaglebone:/sys/class/gpio/gpio52# echo 1 > value
root@beaglebone:/sys/class/gpio/gpio52# cat value
1
root@beaglebone:/sys/class/gpio/gpio52#
```

Conclusion

Linux is very useful for an embedded platform for its simplicity and ease of access to hardware as files.