

# Accessing the Hammer's I/O Ports and Peripherals via the Command Line

---

## BEEPER

To control the beeper, there is a file named "beep" located at: `/sys/class/input/input1/beep`. This file controls the state of the beeper. If you write an integer number to this file, the integer number is used as a divider in the CPU internal counter. The beeper starts to run with a frequency determined by the integer written to the "beep" file and the beep length is determined by the value stored in the "duration" file.

For example, type the following at the Hammer's command prompt:

```
echo 2048 > /sys/class/input/input1/beep    (turn on the beeper and set the divider)
echo 200 > /sys/class/input/input1/duration  (set the beep length to 2 seconds)
```

The file "duration" determines the amount of time the beeper runs. Each count is 10msec. A count of 200 = 2 seconds.

---

## LED's

```
echo 1 > /sys/class/leds/led0/brightness    (turns on Hammer LED0)
echo 0 > /sys/class/leds/led0/brightness    (turns off Hammer LED0)

echo 1 > /sys/class/leds/led1/brightness    (turns on Hammer Carrier LED1)
echo 0 > /sys/class/leds/led1/brightness    (turns off Hammer Carrier LED1)
```

You can also check the current status of the LED by using the "cat" command on the brightness file entry:

```
cat /sys/class/leds/led0/brightness
```

The above command will return the following based upon the current state of the LED0:

```
0 (if LED0 is off)
1 (if LED0 is on)
```

---

## USB – HOST

To use a USB Thumb Drive (USB Flash Drive), perform the following:

- 1) Plug in the USB Thumb Drive into the Hammer Carrier board's USB Host port.
- 2) The Hammer will delay for two seconds before scanning the USB Thumb Drive (to let the Thumb Drive settle before scanning).

- 3) Note that Linux will assign the USB Thumb Drive as: /dev/sda1
- 4) Type the following commands at the Hammer command prompt:

```
mkdir /mnt/thumb          (make a directory named "thumb" under /mnt )  
mount /dev/sda1 /mnt/thumb (mount the USB Thumb Drive at /mnt/thumb)  
cd /mnt/thumb           (your files are located here)  
ls                      ( type "ls" to display all of the files)
```

---

If you need more detailed control you can use the "**devmem**" program that comes preloaded onto the Hammer. The "devmem" program is located at: /bin. Devmem basically acts like the old basic programs: peek and poke. You can "peek or poke" values into memory locations such as the GPIO registers, CPU registers, control registers, etc.

devmem syntax:

**devmem address data\_type data**

address : memory address to act upon

data\_type : access operation type : [b]yte, [h]alfword, [w]ord

data : data to be written

examples:

```
devmem 0xffffdc008 w 0x0003 (write the word 0x0003 to the memory location 0xffffdc008)  
devmem 0xffffdc008         (reads the data located at memory location 0xffffdc008)
```

This next example will demonstrate how to control the LED located on the Hammer CPU module by using the "**devmem**" program. The LED on the Hammer is connected to the GPIO pin GPF0 (port F bit 0). In order to control the LED, you must:

- 1) set the GPIO pin GPF0 as an output
- 2) then write directly to the GPIO pin GPF0.

Set the GPIO pin GPF0 to an output by writing a 0x01 to the GPFCON control register located at address 0x56000050 (the Samsung S3C2410 User Manual lists the addresses and the details of the PORT F control and data registers):

```
devmem 0x56000050 w 0x0001 (set the GPIO pin GPF0 as an output)
```

Write to the GPFDAT data register for Port F located at address 0x56000054. Write a 1 to bit #0 (LSB) to turn off the LED and write a 0 to bit #0 (LSB) to turn on the LED:

```
devmem 0x56000054 w 0x00    (turn the LED on)  
devmem 0x56000054 w 0x01    (turn the LED off)
```

---