In Linux, everything is a file

Learning about Linux through SYSFS

Thanks to Bill Gatliff

The file interface abstraction

- What can we do with files?
  - open, read, write, close, delete
- What is an 'ioctl'?
  - Gets you to the hardware!
- What is a virtual file system?
  - Looks like a file, but executes code in the driver
  - Not really storing anything to media
  - A bit like a "ram disk"

What is SYSFS?

- Virtual file system that exposes drivers to userspace
- **`/sys/devices`** ← driver hierarchy
- **`/sys/bus`** ← links to bus owners
- **`/sys/class`** ← common interfaces
- **`/sys/block`** ← block interface

- Let's go thru some examples…

Hands On #3: Test USR0 and USR1 LED

```
select Applications
select Terminal
type testled
```

Cycles through a short LED test

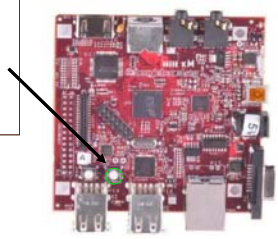
```
type cd /sys/class/leds; ls
type ls "beagleboard::usr0"
type cat "beagleboard::usr0/trigger"
```

USR0 LED will Turn on and off.

Reading the User Button

```
type cd /sys/class/gpio; ls
type echo "4" > export; ls
type echo "in" > gpio4/direction
type cat gpio4/value
type readgpio 4
press User Button

press <ctrl> C to stop
```



Reading Events

When the mouse is moved, events are triggered and sent via the USB port to the processor

```
type opkg install evtest

type <cd /dev/input; ls
type evtest event3
move the Mouse
press <ctrl> C to stop
```

Try other event numbers.
Which one in the keyboard?

Reading I2C Bus

Read the EEPROM inside the display that provides information about that display.

type *cd /sys/bus; ls*
type *cd i2c/devices; ls*
type *echo "eeprom 0x50" > i2c-3/new_device; ls*
type *i2cdump -y 0x3 0x50 b*

type *fbset*

Reading USB Ports

•Read what USB devices are connected to the processor

type *cd /sys/bus/usb/devices; ls*
type *cat usb1/speed*
type *cat usb?/manufacturer*
type *lsusb*
close Terminal Window