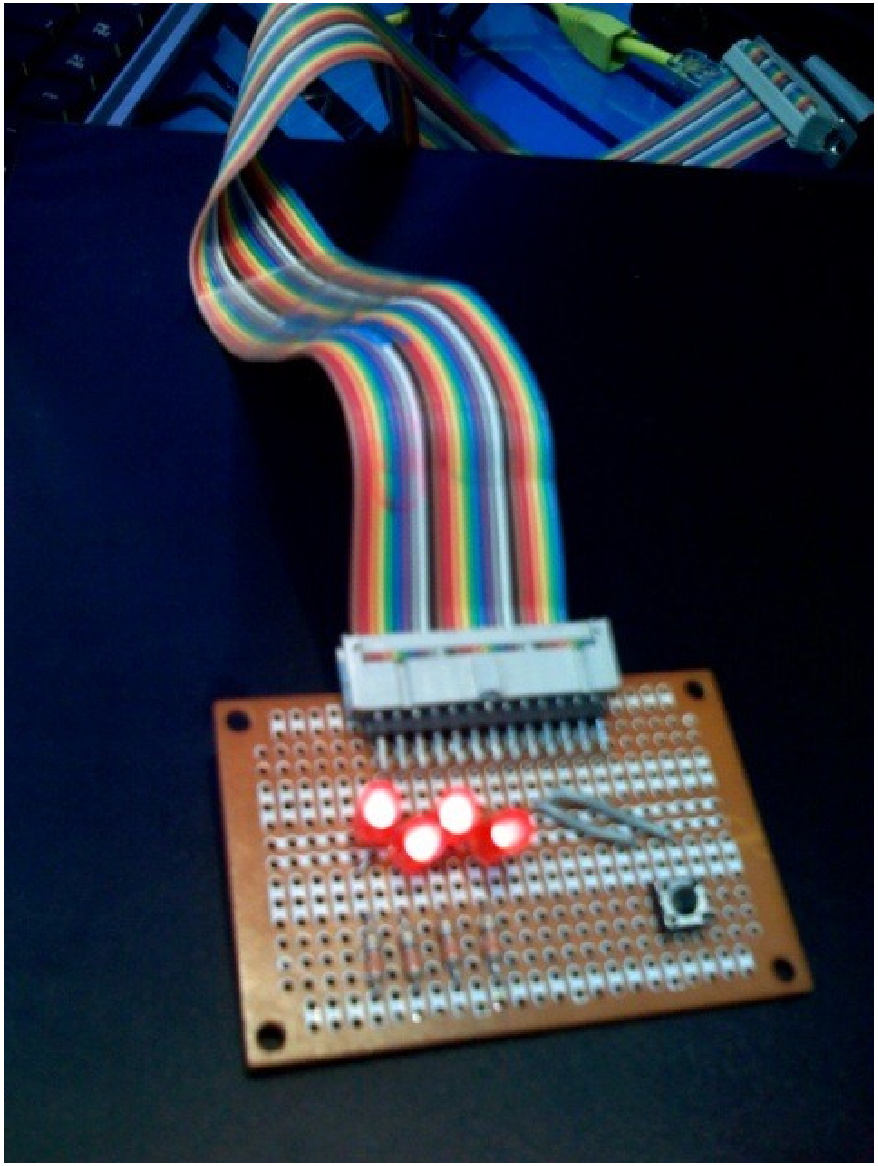
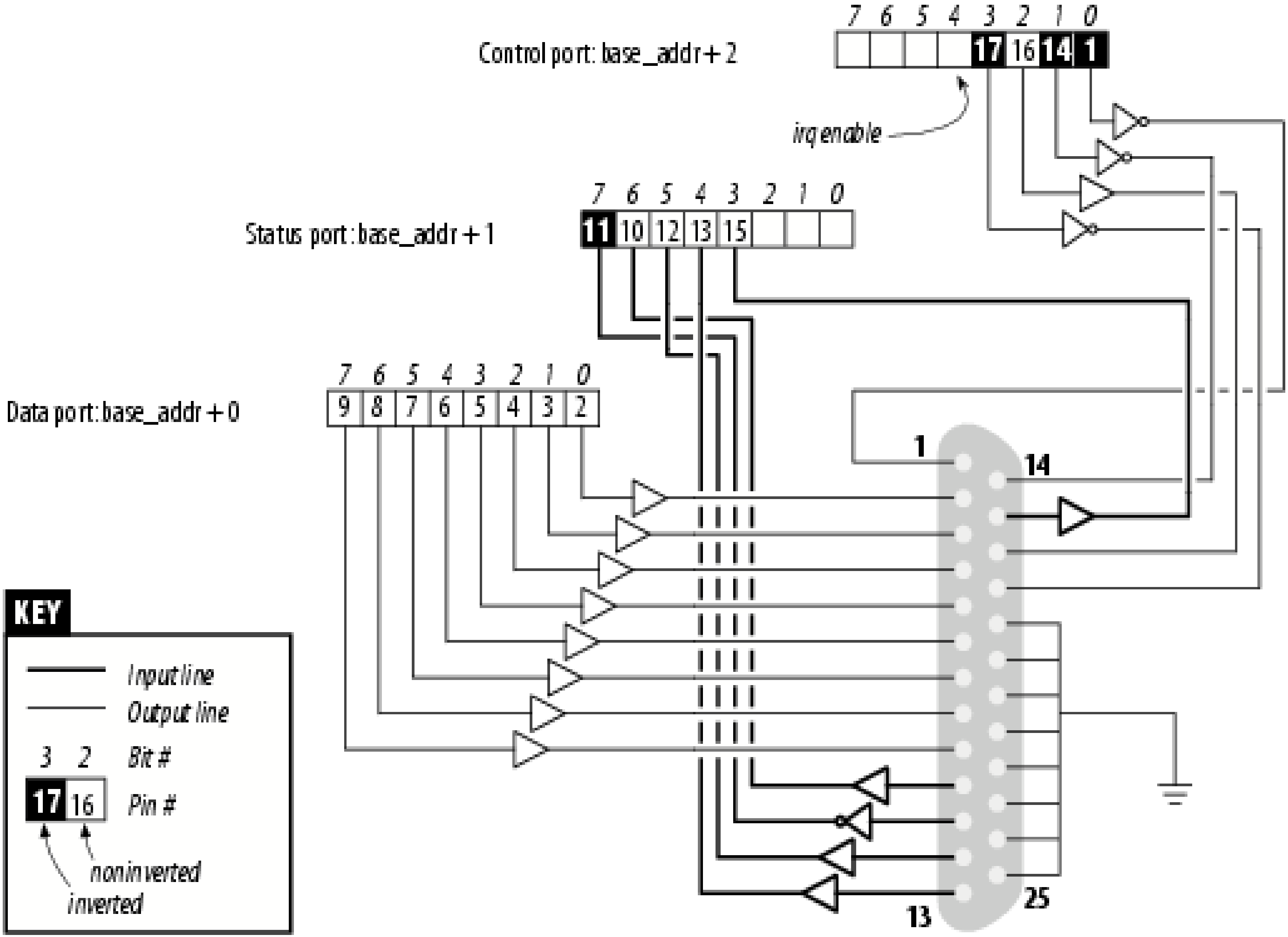
How to Write a Device Driver

Kristen Carlson Accardi

Open Source Technology Center

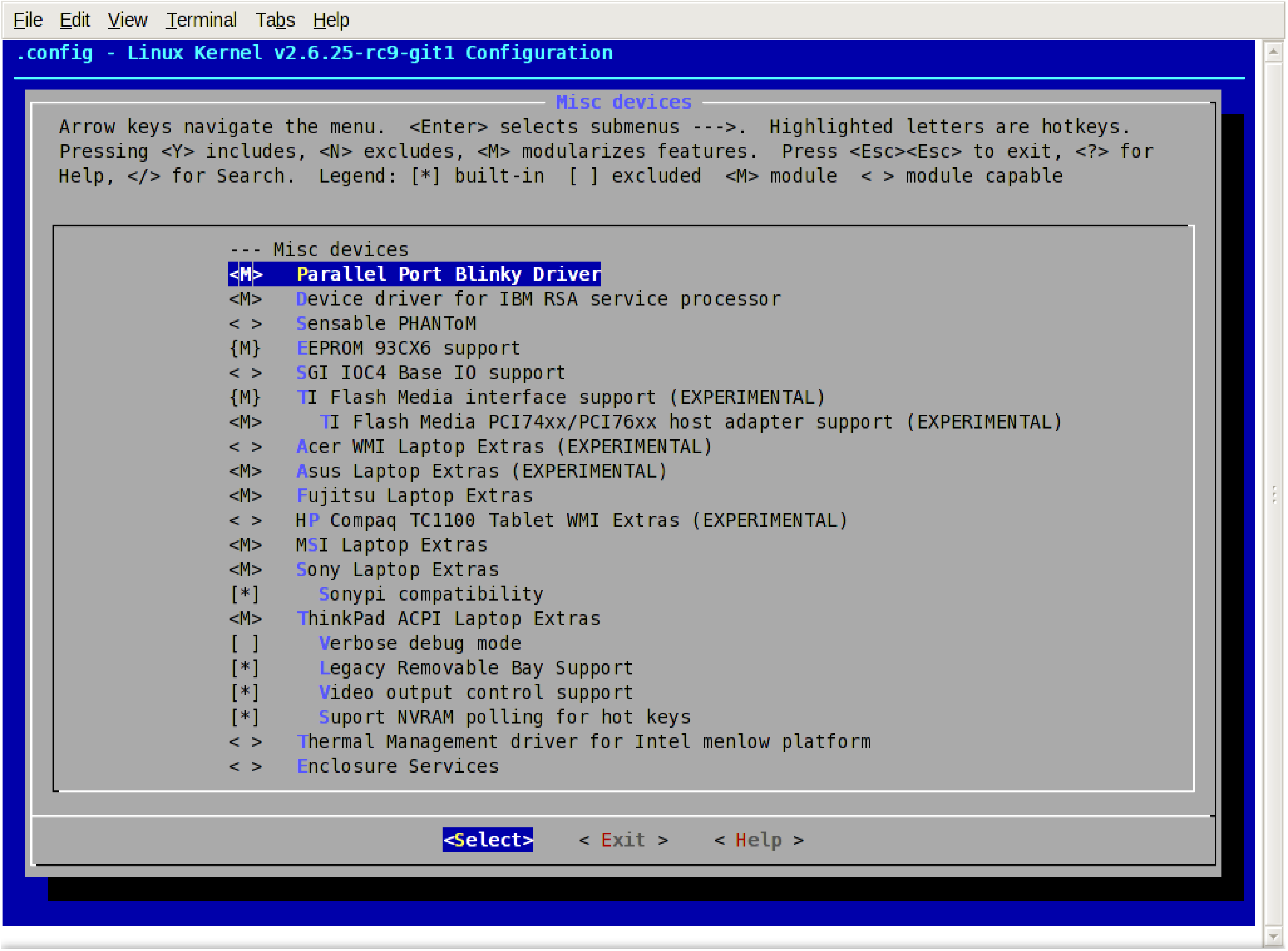




Control port: base_addr + 2
7 6 5 4 3 2 1 0
17 16 14 1
irq enable
Status port: base_addr + 1
7 6 5 4 3 2 1 0
11 10 12 13 15
Data port: base_addr + 0
7 6 5 4 3 2 1 0
9 8 7 6 5 4 3 2
1
14
13
25
KEY
Input line
Output line
3 2
Bit #
17 16
Pin #
noninverted
inverted

Build and Configure

```
Index: linux/drivers/misc/Kconfig
===================================================================
--- linux.orig/drivers/misc/Kconfig     2008-04-13 08:01:47.000000000 -0700
+++ linux/drivers/misc/Kconfig   2008-04-13 08:02:20.000000000 -0700
@@ -13,6 +13,11 @@
if MISC_DEVICES
+config PARPORT_BLINKY
+        tristate "Parallel Port Blinky Driver"
+        help
+          Yippee!!
+
 config ATMEL_PWM
         tristate "Atmel AT32/AT91 PWM support"
```



```
.config - Linux Kernel v2.6.25-rc9-git1 Configuration

                                    Misc devices
 Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are hotkeys.
 Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?> for
 Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable

                --- Misc devices
                <M>   Parallel Port Blinky Driver
                <M>   Device driver for IBM RSA service processor
                < >   Sensable PHANToM
                {M}   EEPROM 93CX6 support
                < >   SGI IOC4 Base IO support
                {M}   TI Flash Media interface support (EXPERIMENTAL)
                <M>     TI Flash Media PCI74xx/PCI76xx host adapter support (EXPERIMENTAL)
                < >   Acer WMI Laptop Extras (EXPERIMENTAL)
                <M>   Asus Laptop Extras (EXPERIMENTAL)
                <M>   Fujitsu Laptop Extras
                < >   HP Compaq TC1100 Tablet WMI Extras (EXPERIMENTAL)
                <M>   MSI Laptop Extras
                <M>   Sony Laptop Extras
                [*]     Sonypi compatibility
                <M>   ThinkPad ACPI Laptop Extras
                [ ]     Verbose debug mode
                [*]     Legacy Removable Bay Support
                [*]     Video output control support
                [*]     Suport NVRAM polling for hot keys
                < >   Thermal Management driver for Intel menlow platform
                < >   Enclosure Services

                          <Select>    < Exit >    < Help >
```

```
Index: linux/drivers/misc/Makefile
=========================================
==============
--- linux.orig/drivers/misc/Makefile     2008-04-13 08:01:47.000000000
-0700
+++ linux/drivers/misc/Makefile    2008-04-13 08:02:20.000000000
-0700
@@ -3,6 +3,7 @@
 #
 obj- := misc.o # Dummy rule to force built-in.o to be made

+obj-$(CONFIG_PARPORT_BLINKY)     += parport_blinky.o
 obj-$(CONFIG_IBM_ASM)          += ibmasm/
 obj-$(CONFIG_HDPU_FEATURES) += hdpuftrs/
 obj-$(CONFIG_MSI_LAPTOP)     += msi-laptop.o
```

```
$ make -j2; sudo make modules_install; sudo make install
```

Start with a skeleton

- Driver interfaces
  - Char
  - Block
  - Network
  - PCI
  - USB
  - ACPI
  - etc.

```
+MODULE_AUTHOR("Kristen Carlson Accardi
<kristen.c.accardi@intel.com");
+MODULE_DESCRIPTION("ACPI bay driver");
+MODULE_LICENSE("GPL");
+module_init(bay_init)
+module_exit(bay_exit)
```

```
+static int __init bay_init(void)
+{
+	/* request the port region */
+
+	/* map the ioport */
+
+	/* register yourself as a platform driver */
+	bay_platform_driver =
+		platform_device_register_simple("bay", 0, NULL, 0);
+	if (IS_ERR(bay_driver)) {
+		printk("Unable to register driver\n");
+		err = PTR_ERR(bay_platform_driver);
+		goto err_out_unmap;
+	}
+
+	/*
+	 * Assign an IRQ number (7) and request irq from OS
+	 *
+	 * XXX - should base irq on port address, but we'll hardcode
+	 * cause we are lazy!
+	 */
+
+	/* enable interrupts on the card */
+
+	/* Hurray! */
+	return 0;
```

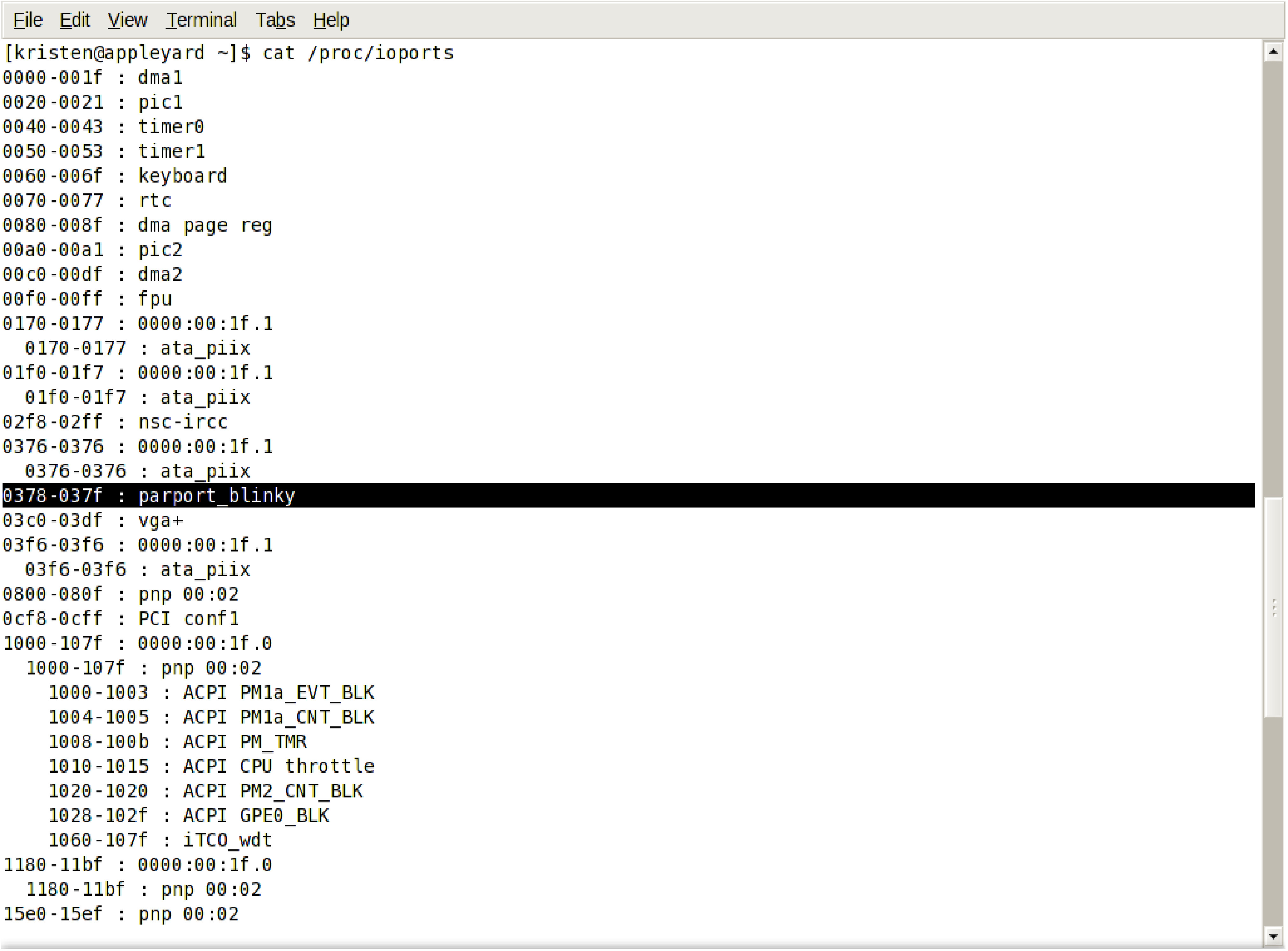
```
+	parport_blinky_platform_driver =
+		platform_device_register_simple("parport_blinky", 0,
NULL, 0);
+	if (IS_ERR(parport_blinky_platform_driver)) {
+		printk("Unable to register driver\n");
+		err = PTR_ERR(parport_blinky_platform_driver);
+		goto err_out_unmap;
+	}
```

```
+static void __exit parport_blinky_exit(void)
+{
+	free_irq(irq, NULL);
+	device_remove_file(&parport_blinky_platform_driver->dev,
+		&dev_attr_flashy);
+	platform_device_unregister(parport_blinky_platform_driver);
+	ioport_unmap(iobase);
+	release_region(base, 8);
+}
```

How to talk to hardware

- Memory Mapped I/O
  - Allocate I/O memory space
    - request_mem_region()
  - Map physical address to virtual
    - ioremap
  - Access with portable API
    - readl()/writel()
    - ioread32()/iowrite32()
- I/O Ports
  - Can make seem like MMIO

```
+   if (!request_region(base, 8, "parport_blinky"))
{
+       printk("unable to get i/o region\n");
+   }
+
+   iobase = ioport_map(base, 8);
+   if (!iobase) {
+       printk("Unable to map ioport\n");
+       err = -ENODEV;
+       goto err_out;
+   }
```

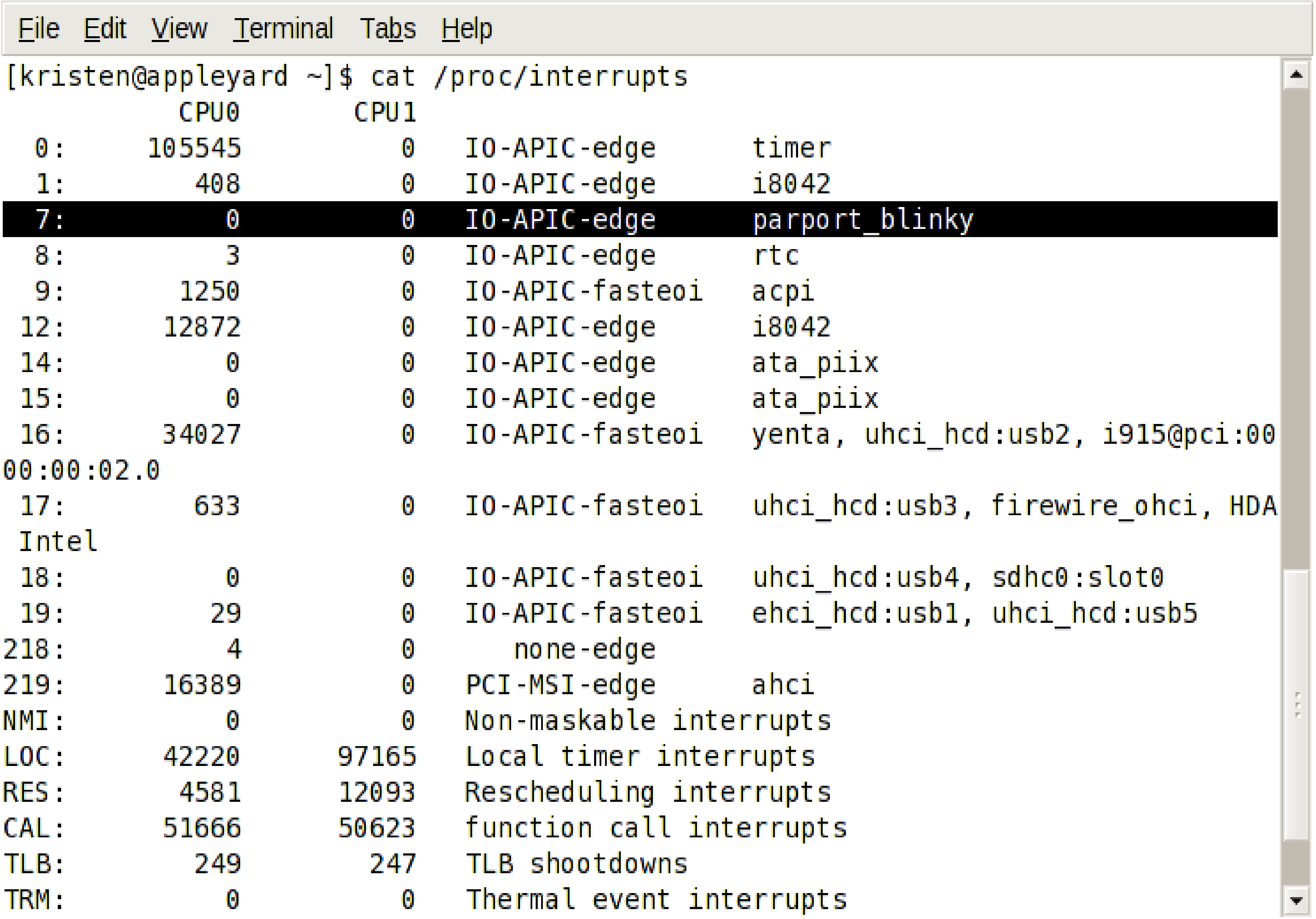


```
[kristen@appleyard ~]$ cat /proc/ioports
0000-001f : dma1
0020-0021 : pic1
0040-0043 : timer0
0050-0053 : timer1
0060-006f : keyboard
0070-0077 : rtc
0080-008f : dma page reg
00a0-00a1 : pic2
00c0-00df : dma2
00f0-00ff : fpu
0170-0177 : 0000:00:1f.1
  0170-0177 : ata_piix
01f0-01f7 : 0000:00:1f.1
  01f0-01f7 : ata_piix
02f8-02ff : nsc-ircc
0376-0376 : 0000:00:1f.1
  0376-0376 : ata_piix
0378-037f : parport_blinky
03c0-03df : vga+
03f6-03f6 : 0000:00:1f.1
  03f6-03f6 : ata_piix
0800-080f : pnp 00:02
0cf8-0cff : PCI conf1
1000-107f : 0000:00:1f.0
  1000-107f : pnp 00:02
    1000-1003 : ACPI PM1a_EVT_BLK
    1004-1005 : ACPI PM1a_CNT_BLK
    1008-100b : ACPI PM_TMR
    1010-1015 : ACPI CPU throttle
    1020-1020 : ACPI PM2_CNT_BLK
    1028-102f : ACPI GPE0_BLK
    1060-107f : iTCO_wdt
1180-11bf : 0000:00:1f.0
  1180-11bf : pnp 00:02
15e0-15ef : pnp 00:02
```

```
+static void parport_blinky_clear_interrupts(void)
+{
+   int val;
+
+   val = ioread8(iobase);
+   val &= 0x7f;
+   iowrite8(val, iobase);
+}
+
+static void parport_blinky_led_off(void)
+{
+   iowrite8(0, iobase);
+}
+
+static void parport_blinky_enable_interrupts(void)
+{
+   iowrite8(0x10, iobase + 2);
+}
+
```

Handling hardware events

```
+	/* request IRQ */
+	err = request_irq(irq, parport_blinky_irq, 0,
+			"parport_blinky", NULL);
+	if (err) {
+		printk("Unable to request irq\n");
+		goto err_remove_sysfs;
+	}
```



```
[kristen@appleyard ~]$ cat /proc/interrupts
           CPU0       CPU1
  0:     105545          0   IO-APIC-edge      timer
  1:        408          0   IO-APIC-edge      i8042
  7:          0          0   IO-APIC-edge      parport_blinky
  8:          3          0   IO-APIC-edge      rtc
  9:       1250          0   IO-APIC-fasteoi   acpi
 12:      12872          0   IO-APIC-edge      i8042
 14:          0          0   IO-APIC-edge      ata_piix
 15:          0          0   IO-APIC-edge      ata_piix
 16:      34027          0   IO-APIC-fasteoi   yenta, uhci_hcd:usb2, i915@pci:00
00:00:02.0
 17:        633          0   IO-APIC-fasteoi   uhci_hcd:usb3, firewire_ohci, HDA
 Intel
 18:          0          0   IO-APIC-fasteoi   uhci_hcd:usb4, sdhc0:slot0
 19:         29          0   IO-APIC-fasteoi   ehci_hcd:usb1, uhci_hcd:usb5
218:          4          0      none-edge
219:      16389          0   PCI-MSI-edge      ahci
NMI:          0          0   Non-maskable interrupts
LOC:      42220      97165   Local timer interrupts
RES:       4581      12093   Rescheduling interrupts
CAL:      51666      50623   function call interrupts
TLB:        249        247   TLB shootdowns
TRM:          0          0   Thermal event interrupts
```

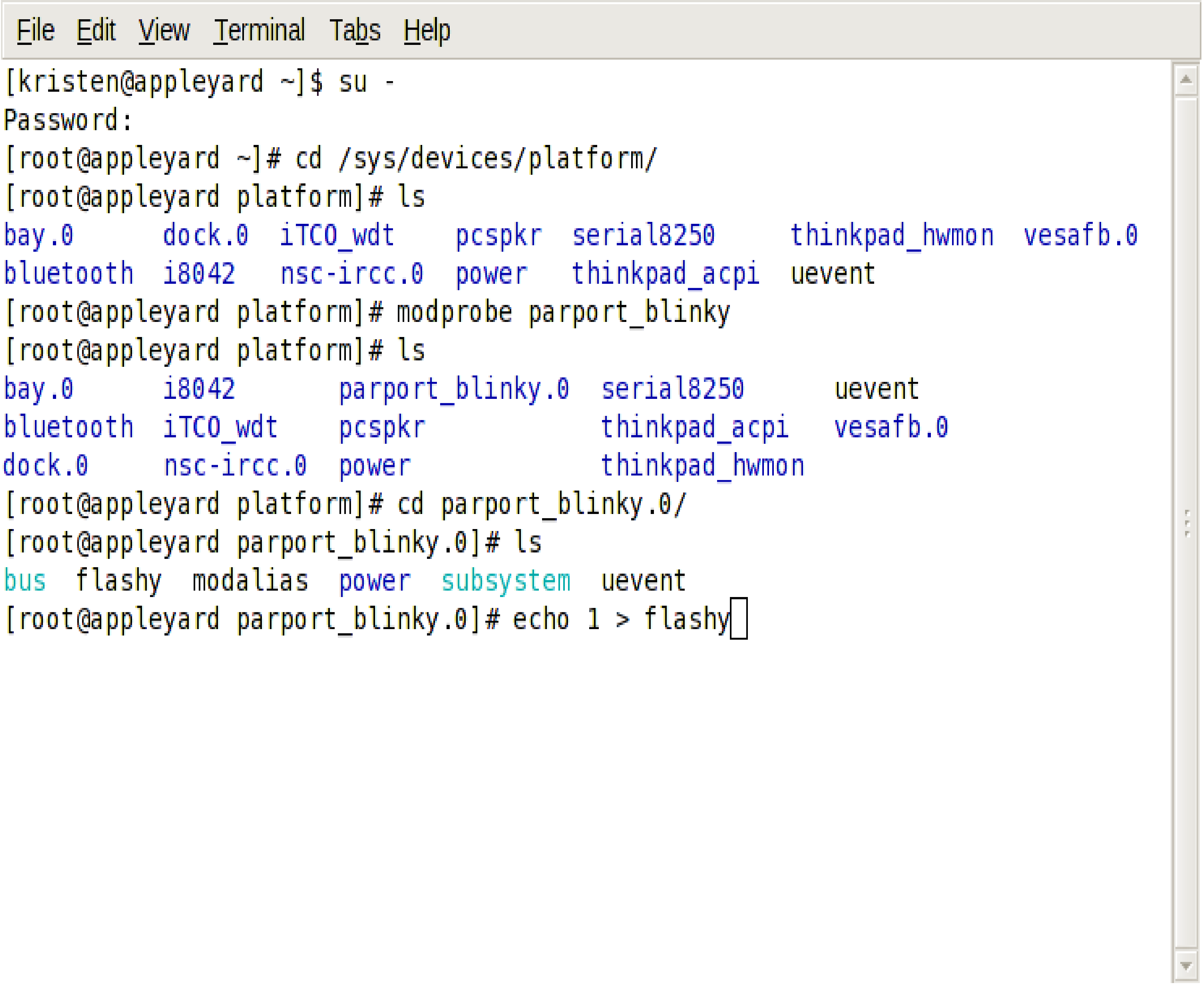
```
+irqreturn_t parport_blinky_irq(int irq, void *dev_id)
+{
+	parport_blinky_blinky();
+	parport_blinky_clear_interrupts();
+	return IRQ_HANDLED;
+}
```

```
+static void parport_blinky_blinky(void)
+{
+	int i;
+
+	if (flashy) {
+		for (i = 0; i < 7; i++) {
+			iowrite8((1 << i), iobase);
+			mdelay(50);
+		}
+	} else {
+		iowrite8(led_state, iobase);
+		led_state ? (led_state = 0) : (led_state = 0xff);
+	}
+}
+
```

Exposing a user interface

```
+   /* create sysfs files */
+   err = device_create_file(&parport_blinky_platform_driver->dev,
+               &dev_attr_flashy);
+   if (err) {
+       printk("Error %d registering sysfs file\n", err);
+       goto err_unregister;
+   }
+
+static int flashy = 0;

+static ssize_t write_flashy(struct device *dev, struct
device_attribute *attr,
+		const char *buf, size_t count)
+{
+	if (!count)
+		return -EINVAL;
+
+	sscanf(buf, "%d\n", &flashy);
+	return count;
+}
+static DEVICE_ATTR(flashy, S_IWUSR, NULL, write_flashy);
```



```
[kristen@appleyard ~]$ su -
Password:
[root@appleyard ~]# cd /sys/devices/platform/
[root@appleyard platform]# ls
bay.0      dock.0  iTCO_wdt    pcspkr  serial8250     thinkpad_hwmon  vesafb.0
bluetooth  i8042   nsc-ircc.0  power   thinkpad_acpi  uevent
[root@appleyard platform]# modprobe parport_blinky
[root@appleyard platform]# ls
bay.0      i8042      parport_blinky.0  serial8250     uevent
bluetooth  iTCO_wdt   pcspkr            thinkpad_acpi  vesafb.0
dock.0     nsc-ircc.0 power             thinkpad_hwmon
[root@appleyard platform]# cd parport_blinky.0/
[root@appleyard parport_blinky.0]# ls
bus  flashy  modalias  power  subsystem  uevent
[root@appleyard parport_blinky.0]# echo 1 > flashy
```

```
+static void parport_blinky_blinky(void)
+{
+	int i;
+
+	if (flashy) {
+		for (i = 0; i < 7; i++) {
+			iowrite8((1 << i), iobase);
+			mdelay(50);
+		}
+	} else {
+		iowrite8(led_state, iobase);
+		led_state ? (led_state = 0) : (led_state = 0xff);
+	}
+}
+
```