Linux *Kernel Hacking* Free Course 2003

UNIVERSITA' DEGLI STUDI DI ROMA TorVergata Università

D. P. Bovet, M. Cesati, V. Garofalo

The lp_led device

The objective

- Implement a simple circuit to be connected to the ISA/EISA parallel port
- Make a full fledged I/O driver for the new device
- Develop a suitable test program

The approach

- Introduce the parallel port
- Describe the circuit
- Describe the software

The parallel port (1)

- Uses a 25-pin connector
- Pins 18-25 must be connected to ground
- The first 17 pins carry data, status and control information
- Asynchronous port
- The 0V signal corresponds to bit 0 , while the 5V signal (on laptops, 3.5V) corresponds to bit 1

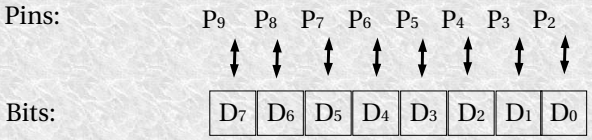
The parallel port (2)

- The first parallel ports were used mainly to connect a printer (hence the DOS name LPT1)
- Can be used both as an input and as an output port
- The pin description that follows refers to a generic parallel port, not to one used to connect a parallel printer

The parallel port interface

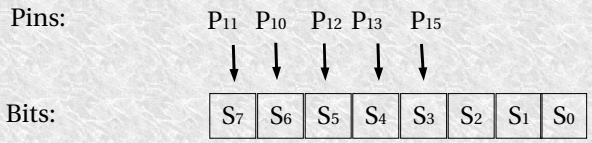
- Three 8-bit I/O ports available: Data, Status, and Control
- The I/O addresses of such ports are usually `0x378`, `0x379` and `0x37a` (check by issuing the `cat /proc/ioports` command while some device connected to the parallel port is working)

The Data port



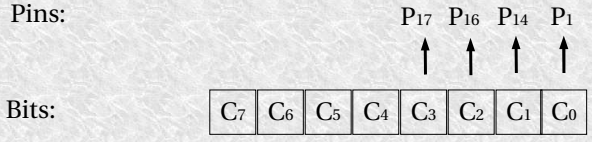
- The 8 lines of the data port are bidirectional

The Status port



- Bits S_0, S_1, S_2 have no corresponding pins
- The remaining bits are read-only bits

The Control port



- Bits C_4, C_5, C_6, C_7 have no corresponding pins
- Bit C4 is used by the interface as an Interrupt Enable Flag
- The remaining bits are write-only bits

More about the parallel port

- A good starting point is:
  http://lvr.com/parport.htm

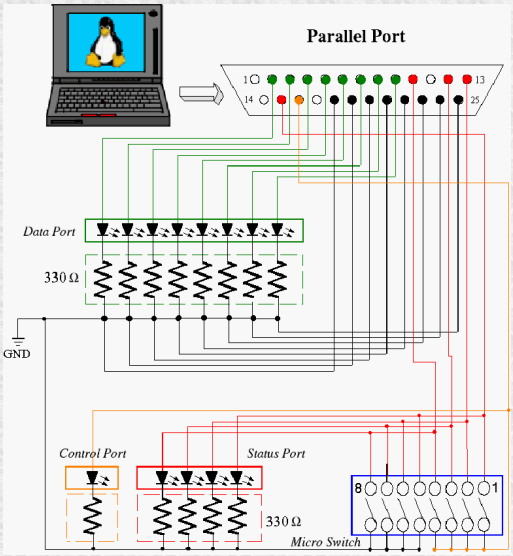
The circuit (1)

- We built a simple circuit connected to the parallel port by means of a parallel cable
- We'll call the corresponding device the *lp_led* device
- The circuit includes:
  - 8 green leds
  - 4 red leds and 4 microswitches
  - 1 orange led (power on)

The circuit (2)

- Writing a byte into the Data port causes the 8 green leds to be lighted in agreement with the byte value
- The value read from bits S_6-S_3 of the Status port depends from the the current setting of the 4 micro switches
- This setting also determines which red leds will be turned on

The circuit lay-out



An I/O driver for the lp_led device

Main I/O driver characteristics

- Closely inspired to the one included in the `linux/drivers/char/lp.c` file
- Supports the `open()`, `close()`, `read()` and `write()` file operations
- Can be configured as a static kernel component or as a module

Steps to be performed (1)

- Add a new kernel configuration option
- Select a free major number (minor=0)
- Create a new `/dev/lp_led` device file for the lp_led device
- Insert a new initialization function in `chr_dev_init()` (to be invoked only if the driver is compiled statically)

Steps to be performed (2)

- Add the functions that implement the file operations of the new device
- Write a suitable test program
- Compile the new kernel and boot it
- The details are shown in the `led_device.txt` file