1 Description

The pport driver allows easy access to the parallel port for programs in userspace. The implementation addresses two problems:

1. Reading the statuslines and writing the controllines with programs that cannot do ioctls or direct IO-operations like portwrites.
2. Do fast and timingcritical changes to the lines. In this mode the user creates a list of commands and timing information with a simple structure and tells the port to execute the commands. A running sequence can't be stopped.

2 Behavior of pport

Data is the datareg at base+0, **Status** is the statusreg at base+1 and **Ctrl** is the controlreg at base+2.
The pport module has an internal queue that may get filled at write operations and is emptied at reads.

- When a program wants to read n bytes from the pport device the driver will return the content of the queue if it is not empty. If the user wanted to read more bytes than the length of the queue, the driver will perform m reads of **data** to return n bytes.
  If the queue was empty and the user wanted to read only 1 byte, the driver will read the datareg and return the value.
- If the user wants to write a byte to the datareg he writes the two bytes 0x0 0xxx to the device. The first one specifies the action, the second is the data itself.
- To write to the ctrlreg, write 0x01 0xxx to the device.
- To read the statuslines, write 0x03 to the device. The driver reads the statusreg and puts the value into the queue. It can be read with the next read.
- A 0x04 byte starts a sequence. The next 4 bytes form an integer holding the number of commands in the sequence.
  After that every 4 bytes form one command. The fist byte is the action (read/write/ctrl/status), the second is the datavalue for write/ctrl actions and the last one is an optional wait in usec. If it is > 0 the driver will delay execution of the **next** command for the given number of μsec.
  The results of all readcommands are queued and can be read from the device after sequence execution.

3 Windows

At the moment there is no windows support. I've started working on the code but someone with a windows compiler would be needed to get it working. Send a mail if interested, my estimation would be 3h of work to get it going.

4 Linux

4.1 Installation

The user distribution contains a precompiled pport kernel module for a standard 2.4 kernel. If you have the right kernel running you don't have to compile the module as described below. If your kernel version differs just slightly, then the insmod command refuses to load the module, but it would still work. To force loading you could use ¡I¿insmod -f pport.o¡/i¿

- Edit the Makefile to reflect your kernel/system configuration.
- Run make and copy the new pport.o file to location /lib/modules/_your_kernel_version_/own/pport.o.
- Add following 2 lines to end of modules.dep:

```
/lib/modules/\_your\_kernel\_version\_/own/pport.o:
```

- Create a character device with major 62, minor 0 in the /dev directory (or elsewhere if you like):

```
mknod /dev/pport0 c 62 0
```

- Add entry in modules.conf to make kerneld load the module if /dev/pport is accessed:

```
alias char-major-62     pport
options pport io=0x378
```

- To test the driver you can use:

```
echo 00 xx | xxd -p -r > /dev/pport0     # to write xx to the output reg
echo 01 xx | xxd -p -r > /dev/pport0     # to write xx to the control reg
dd if=/dev/pport0 bs=1 count=1 | xxd     # to read one byte from input reg
```

4.2 Problems

- **Module loaded:** Use lsmod to make sure that pport.o is really loaded.
- **The data outputs are always 01010000:** Do you use the correct io-base in modules.conf? Is there a lpd running who also uses the parallel port?

5 History

- pport_v1.0 (29. Nov 2002): First working version
- pport_v1.1 (12. Jul 2004): Comments in source code added, changed control register bit interpretation: setting the bit to 1 sets the reg bit to 1 (which may set the parallel port line to 0 or 5V)
- pport_v1.2 (16. Jul 2005): Complete recoding of pport.c linux module (speedup, more robust). Renamed the java package tree including some small changes in the code.