USB to IEEE1284 converter

The objective of this project is to build a AN2131Q based USB to IEEE1284 converter with "true" parallel port register access emulation for peripherals with proprietary handshaking protocols. Saying this, I do not mean a military radar but rather home−brew MCU programmers and control systems. It can be used as a printer or scanner cable, however data transfer speeds do not come close to those of standard parallel port, doh. The design could be optimized to triple the speed of printing, but it was not my goal.

Originally the firmware was based on USS720 Instant USB to IEEE 1284 bridge by Lucent Technologies (data sheet is found on John Hyde's "USB Design by Example" CD−ROM, 1999) and the native Linux driver for this chip. Later I had to design my own interface to make it fit nicely into parport implementation.

Special thanks to:
Thomas Sailer <sailer@ife.ee.ethz.ch> for USS720.c
Tim Waugh <tim@cyberelk.demon.co.uk> for ieee1284_ops.h
Arnim Laeuger <arniml@users.sourceforge.net> for Ezhid project
Tony Cureington <tonycu@users.sourceforge.net> for Ezusb2131 project

USB1284 features

USB:
- −One interface, one alternate setting (0,0);
- −Control endpoint 0;
- −Bulk OUT endpoint 1 (64 bytes);
- −Bulk IN endpoint 2 (64 bytes);
- −Interrupt IN endpoint 4 (1 byte);

IEEE 1284:
- −transparent support for bidirectional communication;
- −parallel port register based operations;
- −hardware support for Compatibility, Nibble, EPP modes

There are two parts in the package:

Directory	***Files***	***Installation Instructions***
./driver	usb1284.c	1) cd ./driver 2) make install 3) make sure parport.o module is loaded: cat /proc/modules 4) if it is not loaded, load it: insmod parport
./firmware	pp.c parport.h usb.h delay.h ezusb_reg.h	1) cd ./firmware 2) make 3) assemble the board 4) plug in and load cat pp.ihx > /proc/ezusb/dev0 5) now you can access the board with user−level device drivers such as ppdev

Vendor Specific Requests

GET REQUEST

1) Status register:

bmRequest Type	bRequest	wValue		wIndex		wLength	Data
0xC0	0	−	−	−	−	1	Status register

2) Control register:

bmRequest Type	bRequest	wValue		wIndex		wLength	Data
0xC0	1	−	Control register	−	Mask	1	Control register

It is possible to set and get control register in one request (like parport_frob_control)

3) Done register

bmRequest Type	bRequest	wValue		wIndex		wLength	Data
0xC0	2	–	–	–	–	1	Done register

This requests returns the number of successfully transferred bytes in one of standard IEEE 1284 modes.

4) Data register

bmRequest Type	bRequest	wValue		wIndex		wLength	Data
0xC0	3	–	–	–	–	1	Data register

Standard data register.

SET REQUEST

1) Mode register

bmRequest Type	bRequest	wValue		wIndex		wLength	Data
0x40	2	–	Mode register	–	Byte count	–	–

To initiate any bulk transfer it is necessary to set mode register and indicate the number of bytes to transfer.

Currently supported modes are:

Bulk pipe OUT 1	
0x01	Compatibility
0x03	EPP data write
0x04	EPP address write

Bulk pipe IN 2	
0x81	Nibble
0x83	EPP data read
0x84	EPP address read

2) Data register

bmRequest Type	bRequest	wValue		wIndex		wLength	Data
0x40	3	–	Data register	–	Byte count	–	–

Standard data register

3) Interrupt mask register

bmRequest Type	bRequest	wValue		wIndex		wLength	Data
0x40	4	–	IRQ mask	–	–	–	–

All changes to status register pins are reported via interrupt pipe IN 4, if respective mask bits are set. No interrupt generated during standard IEEE1284 transfers.

AN2131Q Pin Breakdown

Control register				
Bit	***Pin D−sub***	***Signal***	***Inverted?***	***AN***
0	2	Data 0		PA0
1	3	Data 1		PA1
2	4	Data 2		PA2
3	5	Data 3		PA3
4	6	Data 4		PA4
5	7	Data 5		PA5
6	8	Data 6		PA6
7	9	Data 7		PA7
Status Register				
Bit	***Pin D−sub***	***Signal***	***Inverted?***	***AN***
3	15	nError		PB3
4	13	Select		PB4
5	12	PaperEnd		PB5
6	10	nAck		PB6
7	11	Busy	yes	PB7

Control register				
Bit	***Pin D−sub***	***Signal***	***Inverted?***	***AN***
0	1	nStrobe	yes	PC4
1	14	nAutoLF	yes	PC5
2	16	nInit		PC6
3	17	nSelection	yes	PC7
5		Direction		PB0

Michael Hetherington
chinook@pacific.net.sg