FIFO mode

- Read / Write FIFO with 1k TX and 1k RX buffer

Documentation

Linux ftdi_sio.h driver

- http://lxr.free-electrons.com/source/drivers/usb/serial/ftdi_sio.h

libftdi

- https://www.intra2net.com/en/developer/libftdi/index.php

libmpsse

- https://code.google.com/p/libmpsse/source/browse/trunk/src/mpsse.c

Interface	In Endpoint	Out Endpoint
A	0x02	0x81
B	0x04	0x83
C	0x06	0x85
D	0x08	0x87

bcdDevice	**iSerialNumber**	**Type**
0x400	0	TYPE_BM
0x200	0	TYPE_BM
0x200	*NA*	TYPE_AM
0x500	*NA*	TYPE_2232C
0x600	*NA*	TYPE_R
0x700	*NA*	TYPE_2232H
0x800	*NA*	TYPE_4232H
0x900	*NA*	TYPE_232H

FTDI Features

Up to 4 interfaces / channels?

Multi-Purpose UART / FIFO Controllers
Baud Rate Generators?
TX / RX LEDs

FT2232H just has two interfaces - Channel A and Channel B

FTDI Chip Operation Modes

The FTDI Chip has three primary modes of operation;

- UART mode - data endpoints read/write
- FIFO mode - data endpoints read/write to output pins
- MPSSE mode - Special protocol on data endpoints

FX2 Data Destinations

- Hardware accelerated FIFO -- Can be used when FX2 matches FTDI pins. Uses the FX2 engine to transmit data at ~48 megabytes/second.
- Software FIFO -- Used when FX2 doesn't match FTDI pins. Does all the work on the 8051 CPU. Very slow.
- Hardware UART -- Uses the hardware UART found in the larger FX2s
- Software UART on bit pin -- UART run on the 8051 CPU but using the bit set / bit clear operations.
- Software UART on byte port (Port E) -- UART run on the 8051 CPU but using the byte access operations.

FT245 Synchronous / Asynchronous FIFO Interface / CPU Style FIFO

The FTDI chip "FT245 XXX FIFO Interfaces" maps almost directly to the "Slave FIFO" interface of the FX2 when pins are connected correctly.

The CPU-style FIFO Interface is almost identical too.

FTDI Chip	**FX2**	**Usage**
ADBUF[7:0]	Port B	Bidirectional FIFO Data
RXF#	FLAG - Empty Flag (inverted?)	Low == Data available in FIFO **Can read**
TXE#	FLAG - Full Flag	High == FIFO is Full **Don't write**
RD#	SLWR?	Write FIFO byte to ADBUF[7:0]
WR#	SLRD?	Read FIFO byte from ADBUF[7:0]
SIWU#	PA3 - WU2	Remote wakeup?
CLKOUT	IFCLK	60MHz clock

OE#	SLOE	Output Enable
SIWU#		???
A0		Address Bit

MPSSE Mode

Multi-protocol synchronous serial engine (MPSSE) mode.

JTAG Mode

FTDI Chip	**Usage**	
TCK	Test Interface Clock	
TDI	Test Data Input	
TDO	Test Data Output	
TMS	Test Mode Select	
GPIO[0..7]	GPIO	

SPI Mode

FTDI Chip	**Usage**	
SK	SPI Master Serial Clock	
DO	SPI Master Data Out	
DI	SPI Master Data In	
CS	SPI Chip Select	
GPIO[0..7]		

Fast Serial Interface

FTDI Chip	**Usage**	
FSDI	Fast serial data input	
FSCLK	Fast serial clock input	
FSDO	Fast serial data output	
FSCTS	Fast serial "Clear to Send" signal output.	

Device Release Number

Type???	Device Release Number	Strings Start At
TYPE_AM	0x02	0x94
TYPE_BM	0x04	0x94
TYPE_2232C	0x05	0x96
TYPE_R	0x06	0x98
TYPE_2232H	0x07	0x9a
TYPE_4232H	0x08	0x9a
TYPE_232H	0x09	0xa0

Speeds

Instruction cycles are **four clock cycles**
Most operations take 1 or 2 **instructions cycles**.
Clock rate is 48MHz.
Instruction rate is 12MHz or 88.33ns.

MOV direct, direct (using AUTOPTR) == 3 instruction cycles == 4MHz == 1 byte moved == 4 megabytes/second
MOVX (DPTR -> Register), INC DPS, MOVX (Register->DPTR), INC DPS == 4 instruction cycles == 3MHz == 1 byte moved == 3 megabytes/second