Future Technology Devices International Ltd

FT2232D Dual USB to Serial UART/FIFO IC

The FT2232D is a dual USB to serial UART or FIFO interface with the following advanced features:

- Single chip USB to dual channel serial / parallel ports with a variety of configurations.
- Entire USB protocol handled on the chip. No USB specific firmware programming required.
- Transfer Data Rate 300 to 3 Mbaud.
- USB to parallel FIFO transfer data rate up to 1 megabyte / second.
- Multi-Protocol Synchronous Serial Engine (**MPSSE**) to simplify synchronous serial protocol (USB to JTAG, USB to I^2C, USB to SPI) design.
- CPU-style FIFO interface mode simplifies CPU interface design.
- MCU host bus emulation mode configuration option.
- Fast Opto-Isolated serial interface option.
- FTDI's royalty-free Virtual Com Port (VCP) and Direct (D2XX) drivers eliminate the requirement for USB driver development in most cases.
- Highly integrated design includes 3.3V LDO regulator for USB I/O, integrated POR function and on chip clock multiplier PLL (6MHz – 48MHz).
- Asynchronous serial UART interface option with full hardware handshaking and modem interface signals.
- Enhanced bit-bang Mode interface option with RD# and WR# strobes.
- Configurable I/O drive strength.
- Fully assisted hardware or X-On / X-Off software handshaking.
- UART Interface supports 7/8 bit data, 1/2 stop bits, and Odd/Even/Mark/Space/No Parity.
- Operational configuration mode and USB Description strings configurable in external EEPROM over the USB interface.
- Low operating and USB suspend current.
- Supports bus powered, self powered and high-power bus powered USB configurations.
- UHCI/OHCI/EHCI host controller compatible.
- USB 2.0 Full Speed (12Mbits/Second) compatible.
- Extended -40°C to 85°C industrial operating temperature range.
- Compact 48-LD Lead Free LQFP package
- +4.35V to +5.25V single supply operating voltage range.
- Dedicated Windows DLLs available for USB to JTAG, USB to SPI, and USB to I^2C applications.
- ESD protection for FT2232D IO's:
  Human Body Model (HBM) ±2kV,
  Machine Mode (MM) ±100V,
  Charge Device Model (CDM) ±500V,
  Latch-up free..

1 Typical Applications

- USB to Dual Port RS232 Converters
- USB to Dual Port RS422 / RS485 Converters
- Upgrading Legacy Peripheral Designs to USB
- USB Instrumentation
- USB JTAG Programming
- USB to SPI Bus Interfaces
- USB Industrial Control
- Field Upgradable USB Products
- Galvanically Isolated Products with USB Interface
- USB to synchronous serial interface
- Cellular and cordless phone USB data transfer cables and interfaces.
- USB Audio and Low Bandwidth Video data transfer
- PDA to USB data transfer
- USB Smart Card Readers
- USB Instrumentation
- USB Industrial Control
- USB MP3 Player Interface
- USB FLASH Card Reader / Writers
- Set Top Box PC - USB interface
- USB Digital Camera Interface
- USB Bar Code Readers
- Interfacing MCU / PLD / FPGA based designs to USB

1.0 Driver Support

The FT2232D requires USB drivers (listed below), available free from **http://www.ftdichip.com**, which are used to make the FT2232D appear as a virtual COM port (VCP). This then allows the user to communicate with the USB interface via a standard PC serial emulation port (for example TTY). Another FTDI USB driver, the D2XX driver, can also be used with application software to directly access the FT2232D though a DLL.

Royalty free VIRTUAL COM PORT (VCP) DRIVERS for...

- Windows 98, 98SE, ME, 2000, Server 2003, XP and Server 2008
- Windows XP and XP 64-bit
- Windows Vista and Vista 64-bit
- Windows XP Embedded
- Windows CE 4.2, 5.0 and 6.0
- Mac OS 8/9, OS-X
- Windows 7 32,64 bit
- Linux 2.4 and greater

Royalty free D2XX Direct Drivers (USB Drivers + DLL S/W Interface)

- Windows 98, 98SE, ME, 2000, Server 2003, XP and Server 2008
- Windows XP and XP 64-bit
- Windows Vista and Vista 64-bit
- Windows XP Embedded
- Windows CE 4.2, 5.0 and 6.0
- Windows 7 32,64 bit
- Linux 2.4 and greater

For driver installation, please refer to the application note AN232B-10.

1.1 Part Numbers

Part Number	Package
FT2232D	48 Pin LQFP

Table 1.1 Part Numbers

1.2 USB Compliant

The FT2232D is fully compliant with the USB 2.0 specification and has been given the USB-IF Test-ID (TID) 40680003.

2 FT2232D Block Diagram

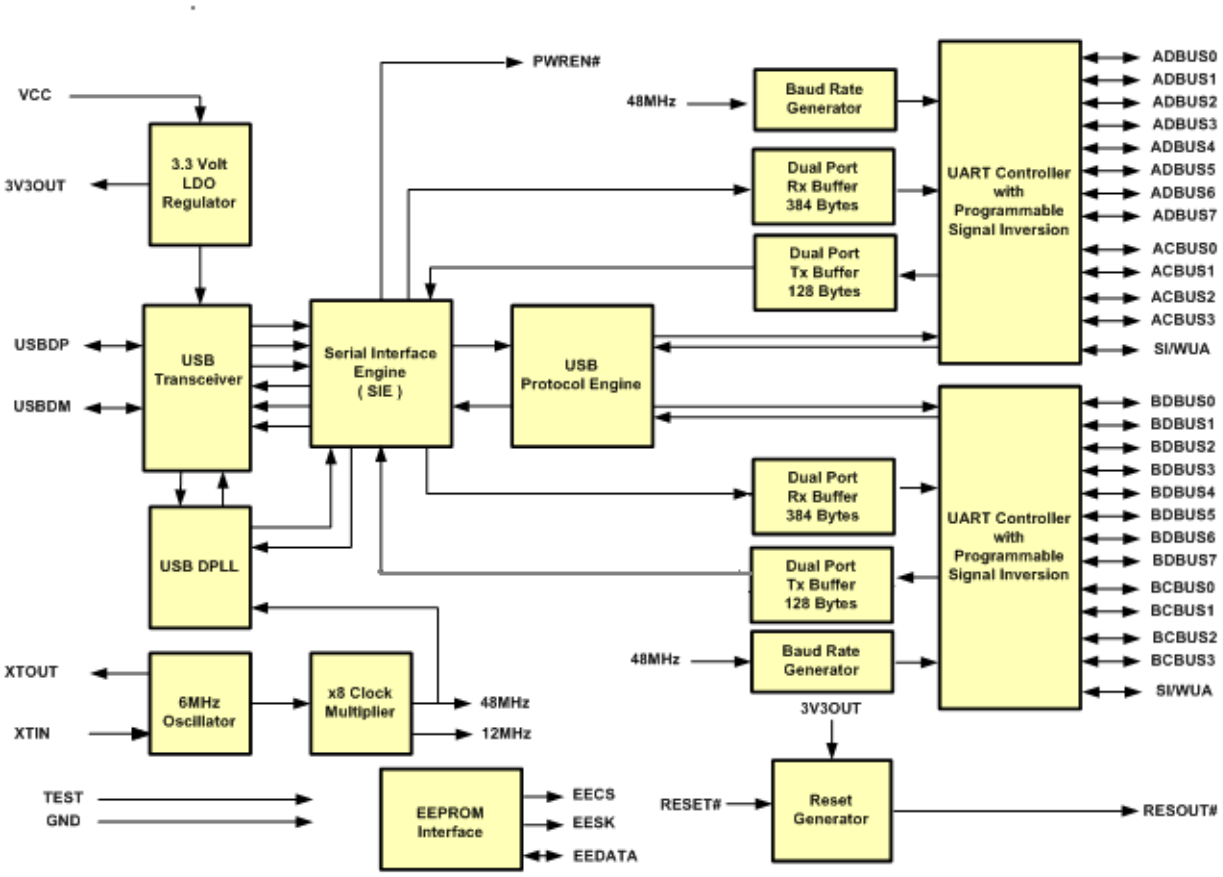

Figure 2.1 FT2232D Block Diagram

For a description of each function please refer to Section 4.1.

Table of Contents

3 Device Pin Out and Signal Description

3.0 48-Pin LQFP Package

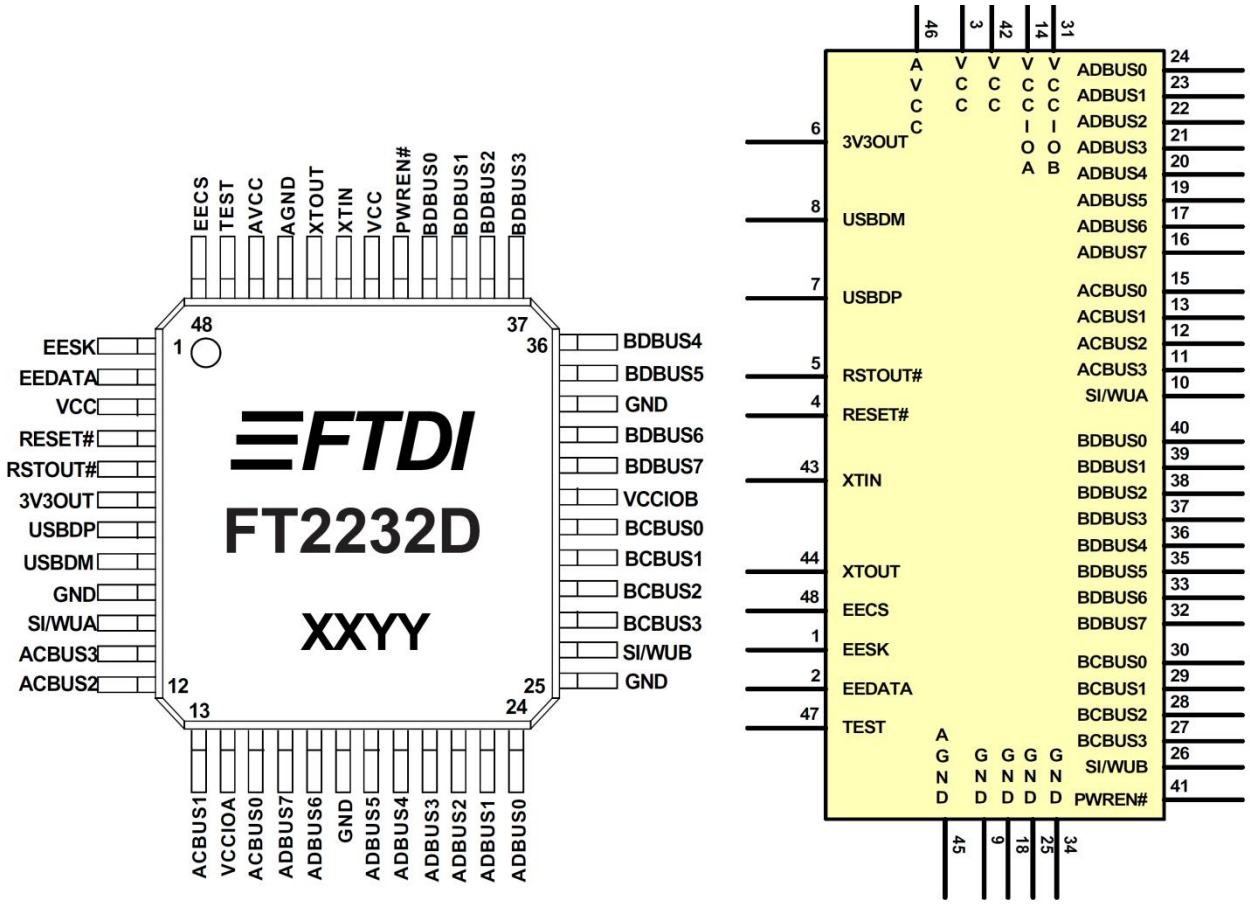

Figure 3.1 48 pin LQFP Package Pin Out and Schematic Symbol

3.1 Pin Out Description

This section describes the operation of the FT2232D pins. Common pins are defined in the first section, and then the I/O pins are defined by chip mode.

Note: The convention used throughout this document for active low signals is the signal name followed by#

3.2 Common Pins

The operation of the following FT2232D pins do not change regardless of the configured mode:-

Pin No.	Name	Type	Description
7	USBDP	I/O	USB Data Signal Plus (Requires 1.5K pull-up to 3V3OUT or RSTOUT#)
8	USBDM	I/O	USB Data Signal Minus

Table 3.1.1 USB Interface Group

Pin No.	Name	Type	Description
48	EECS	I/O	EEPROM – Chip Select. Tri-State during device reset. ****Note 1**
1	EESK	OUTPUT	Clock signal to EEPROM. Tri-State during device reset, else drives out. ****Note 1**
2	EEDATA	I/O	EEPROM – Data I/O Connect directly to Data-In of the EEPROM and to Data-Out of the EEPROM via a 2.2K resistor. Also, pull Data-Out of the EEPROM to VCC via a 10K resistor for correct operation. Tri-State during device reset. ****Note 1**

Table 3.2.2 EEPROM Interface Group

Pin No.	Name	Type	Description
4	RESET#	INPUT	Can be used by an external device to reset the FT2232D. If not required, tie to VCC. ****Note 1**
5	RSTOUT#	OUTPUT	Output of the internal Reset Generator. Drives low for 5.6 ms after VCC > 3.5V and the internal clock starts up, then clamps it's output to the 3.3V output of the internal regulator. Taking RESET# low will also force RSTOUT# to drive low. RSTOUT# is NOT affected by a USB Bus Reset.
47	TEST	INPUT	Puts device into I.C. test mode – must be tied to GND for normal operation.
41	PWREN#	OUTPUT	Goes Low after the device is configured via USB, then high during USB suspend. Can be used to control power to external logic using a P-Channel Logic Level MOSFET switch. Enable the Interface Pull-Down Option in EEPROM when using the PWREN# pin in this way.
43	XTIN	INPUT	Input to 6MHz Crystal Oscillator Cell. This pin can also be driven by an external 6MHz clock if required. Note: Switching threshold of this pin is VCC/2, so if driving from an external source, the source must be driving at 5V CMOS level or a.c. coupled to centre around VCC/2.
44	XTOUT	OUTPUT	Output from 6MHz Crystal Oscillator Cell. XTOUT stops oscillating during USB suspend, so take care if using this signal to clock external logic.

Table 3.3.3 Miscellaneous Signal Group

****Note 1 -** *During device reset, these pins are tri-state but pulled up to VCC via internal 200K resistors.*

Pin No.	Name	Type	Description
6	3V3OUT	OUTPUT	3.3 volt Output from the integrated L.D.O. regulator This pin should be decoupled to GND using a 33nF ceramic capacitor in close proximity to the device pin. It's prime purpose is to provide the internal 3.3V supply to the USB transceiver cell and the RSTOUT# pin. A small amount of current (<= 5mA) can be drawn from this pin to power external 3.3V logic if required.
3, 42	VCC	PWR	+4.35 volt to +5.25 volt VCC to the device core, LDO and non-UART / FIFO controller interface pins.
14	VCCIOA	PWR	+3.0 volt to +5.25 volt VCC to the UART / FIFO A Channel interface pins 10..13, 15..17 and 19..24. When interfacing with 3.3V external logic in a bus powered design connect VCCIO to a 3.3V supply generated from the USB bus. When interfacing with 3.3V external logic in a self powered design connect VCCIO to the 3.3V supply of the external logic. Otherwise connect to VCC to drive out at 5V CMOS level.
31	VCCIOB	PWR	+3.0 volt to +5.25 volt VCC to the UART / FIFO B Channel interface pins 26..30, 32..33 and 35..40. When interfacing with 3.3V external logic in a bus powered design connect VCCIO to a 3.3V supply generated from the USB bus. When interfacing with 3.3V external logic in a self powered design connect VCCIO to the 3.3V supply of the external logic. Otherwise connect to VCC to drive out at 5V CMOS level.
9,18, 25, 34	GND	PWR	Device - Ground Supply Pins
46	AVCC	PWR	Device - Analog Power Supply for the internal x8 clock multiplier. A low pass filter consisting of a 470 Ohm series resistor and a 100 nF to GND should be used on the supply to this pin.
45	AGND	PWR	Device - Analog Ground Supply for the internal x8 clock multiplier

Table 3.4.4 Power and Ground Group

3.3 IO Pin Definitions by Chip Mode

The FT2232D will default to dual serial mode (232 UART mode on both channel A and B, if no external EEPROM is used, or the external EEPROM is blank. The definition of the following pins varies according to the chip's mode:-

Channel A

Pin#	*Generic Pin name*	*Pin Definitions by Chip Mode **Note 2*						
		232 UART Mode	*245 FIFO*	*Enhanced Asynchronous and Synchronous Serial*	*MPSSE **Note 4*	*MCU Host Bus Emulation Mode **Note 5*	*Fast Opto-Isolated Serial Mode*	*CPU FIFO Interface Mode*
24	ADBUS0	TXD	D0	D0	TCK/SK AD0	**Note 3	D0	D0
23	ADBUS1	RXD	D1	D1	TDI/D0	AD1	D1	D1
22	ADBUS2	RTS#	D2	D2	TDO/DI	AD2	D2	D2
21	ADBUS3	CTS#	D3	D3	TMS/CS AD3	D3		D3
20	ADBUS4	DTR#	D4	D4	GPIOL0	AD4	D4	D4
19	ADBUS5	DSR#	D5	D5	GPIOL1	AD5	D5	D5
17	ADBUS6	DCD#	D6	D6	GPIOL2	AD6	D6	D6
16	ADBUS7	RI#	D7	D7	GPIOL3	AD7	D7	D7
15	ACBUS0	TXDEN	RXF#	WR# **Note 6	GPIOH0	I/O0	CS#	CS#
13	ACBUS1	SLEEP#	TXE#	RD# **Note 6	GPIOH1	I/O1	A0	A0
12	ACBUS2	RXLED#	RD#	WR# **Note 7	GPIOH2	IORDY	RD#	RD#
24	ADBUS0	TXD	D0	D0	TCK/SK AD0	**Note 3	D0	
11	ACBUS3	TXLED#	WR	RD# **Note 7	GPIOH3	OSC	WR#	WR#
10	SI/WUA	SI/WUA	SI/WUA	SI/WUA	**Note 8	**Note 8	**Note 8	**Note 8

Table 3.5.5 Pin Definition by Chip Mode (Channel A)

****Note 2:** 232 UART, 245 FIFO, CPU FIFO Interface, and Fast Opto-Isolated modes are enabled in the external EEPROM. Enhanced Asynchronous and Synchronous Bit-Bang modes, MPSSE, and MCU Host Bus Emulation modes are enabled using the driver command set bit mode. See Section 3.3 for details.

****Note 3:** Channel A can be configured in another IO mode if channel B is in Fast Opto-Isolated Serial Mode. If both Channel A and Channel B are in Fast Opto-Isolated Serial Mode all of the IO will be on Channel B.

****Note 4:** MPSSE is Channel A only.

****Note 5:** MCU Host Bus Emulation requires both Channels.

****Note 6:** The Bit-Bang Mode (synchronous and asynchronous) WR# and RD# strobes are on these pins when the main Channel mode is 245 FIFO, CPU FIFO interface, or Fast Opto-Isolated Serial Modes.

****Note 7:** The Bit-Bang Mode (synchronous and asynchronous) WR# and RD# strobes are on these pins when the main Channel mode is 232 UART Mode.

Note 8: SI/WU is not available in these modes.

Channel B

Pin#	Generic Pin name	Pin Definitions by Chip Mode **Note 2						
		232 UART Mode	245 FIFO	Enhanced Asynchronous and Synchronous Serial	MPSSE **Note 4	MCU Host Bus Emulation Mode **Note 5	Fast Opto-Isolated Serial Mode	CPU FIFO Interface Mode
40	BDBUS0	TXD	D0	D0	A8	FSDI	D0	D0
39	BDBUS1	RXD	D1	D1	A9	FSCLK	D1	D1
38	BDBUS2	RTS#	D2	D2	A10	FSDO	D2	D2
37	BDBUS3	CTS#	D3	D3	A11	FSCTS	D3	D3
36	BDBUS4	DTR#	D4	D4	A12	**Note 3	D4	D4
35	BDBUS5	DSR#	D5	D5	A13	D5		D5
33	BDBUS6	DCD#	D6	D6	A14	D6		D6
32	BDBUS7	RI#	D7	D7	A15	D7		D7
30	BCBUS0	TXDEN	RXF#	WR# **Note 9	CS#	CS#		CS#
29	BCBUS1	SLEEP#	TXE#	RD# **Note 9	ALE	A0		A0
28	BCBUS2	RXLED#	RD#	WR# **Note 7	RD#	RD#		RD#
27	BCBUS3	TXLED#	WR	RD# **Note 7	WR#	WR#		WR#
26	SI/WUB	SI/WUB	SI/WUB	SI/WUB	**Note 8	**Note 8	SI/WUB	**Note 8
40	BDBUS0	TXD	D0	D0	A8	FSDI	D0	

Table 3.6.6 Pin Definition by Chip Mode (Channel B)

Note 2: 232 UART, 245 FIFO, CPU FIFO Interface, and Fast Opto-Isolated modes are enabled in the external EEPROM. Enhanced Asynchronous and Synchronous Bit-Bang modes, MPSSE, and MCU Host Bus Emulation modes are enabled using the driver command set bit mode. See Section 3.3 for details.

Note 3: Channel A can be configured in another IO mode if channel B is in Fast Opto-Isolated Serial Mode. If both Channel A and Channel B are in Fast Opto-Isolated Serial Mode all of the IO will be on Channel B.

Note 4: MPSSE is Channel A only.

Note 5: MCU Host Bus Emulation requires both Channels.

Note 6: The Bit-Bang Mode (synchronous and asynchronous) WR# and RD# strobes are on these pins when the main Channel mode is 245 FIFO, CPU FIFO interface, or Fast Opto-Isolated Serial Modes.

Note 7: The Bit-Bang Mode (synchronous and asynchronous) WR# and RD# strobes are on these pins when the main Channel mode is 232 UART Mode.

Note 8: SI/WU is not available in these modes.

Note 9: The Bit-Bang Mode (synchronous and asynchronous) WR# and RD# strobes are on these pins when the main Channel mode is 245 FIFO, CPU FIFO interface. Bit-Bang mode is not available on Channel B when Fast Opto-Isolated Serial Mode is enabled.

3.4 IO Mode Command Hex Values

Enhanced Asynchronous and Synchronous Bit-Bang modes, MPSSE, and MCU Host Bus Emulation modes are enabled using the D2XX driver command FT_SetBitMode. The hex values used with this command to enable these modes are as follows-

Mode	Value (Hex)
Reset the IO bit Mode	0
Asynchronous Bit Bang Mode	1
MPSSE	2
Synchronous Bit bang Mode	4
MCU Host bus Emulation	8
Fast Opto-Isolated Serial Mode	10

Table 3.7.7 IO Mode Command Hex Values

See application note **AN2232-02, "Bit Mode Functions for the FT2232D"** for more details and examples.

Note that all other device modes can be enabled in the external EEPROM, and do not require these values to be configured.

In the case of Fast Opto-Isolated Serial mode sending a value of 10 will hold this device mode in reset, and sending a value of 0 will release this mode from reset.

4 Function Description

The FT2232D is a USB to serial UART interface device which incorporates the functionality of two of FTDI's second generation FT232BM and FT245BM chips into a single device. A single downstream USB port is converted to two IO channels which can each be individually configured as a FT232BM-style UART interface, or a FT245BM-style FIFO interface, without the need to add a USB hub. In addition a new high drive level option means that the device UART / FIFO IO pins will drive out at around three times the normal power level, meaning that the bus can be shared by several devices.

4.0 Key Features

Two Individually Configurable IO Channels: Each of the FT2232D's Channels (A and B) can be individually configured as a FT232BM-style UART interface, or as a FT245BM-style FIFO interface. In addition these channels can be configured in a number of special IO modes.
Integrated Power-On-Reset (POR) circuit: The device incorporates an internal POR function. A RESET# pin is available to allow external logic to reset the device where required, however for most applications this pin can simply be hardwired to Vcc. A RSTOUT# pin is provided in order to allow the new POR circuit to provide a stable reset to external MCU and other devices.
Integrated RCCLK circuit: Used to ensure that the oscillator and clock multiplier PLL frequency are stable prior to USB enumeration.
Integrated level converter on UART / FIFO interface and control signals: Each channel of the FT2232D has its own independent VCCIO pin that can be supplied by between 3V to 5V. This allows each channel's output voltage drive level to be individually configured. Thus allowing, for example 3.3V logic to be interfaced to the device without the need for external level converter I.C.'s.
Improved power management control for high-power USB Bus Powered devices: The PWREN# pin will become active when the device is enumerated by USB, and be deactivated when the device is in USB suspend. This can be used to directly drive a transistor or P-Channel MOSFET in applications where power switching of external circuitry is required. The BM pull down enable feature (configured in the external EEPROM) is also retained. This will make the device gently pull down on the FIFO / UART IO lines when the power is shut off (PWREN# is high). In this mode any residual voltage on external circuitry is bled to GND when power is removed, thus ensuring that external circuitry controlled by PWREN# resets reliably when power is restored.
Send Immediate / Wake Up Signal Pin on each channel: There is a Send Immediate / Wake Up (SI/WU) signal pins on each of the chips channels. These combine two functions on one pin. If USB is in suspend mode (and remote wakeup is enabled in the EEPROM), strobing this pin low will cause the device to request a resume from suspend (WakeUp) on the USB Bus. Normally, this can be used to wake up the Host PC. During normal operation, if this pin is strobed low any data in the device RX buffer will be sent out over USB on the next Bulk-IN request from the drivers regardless of the packet size. This can be used to optimise USB transfer speed for some applications.
Low suspend current: The suspend current of the FT2232D is typically under 100 µA (excluding the 1.5K pull up resistor on USBDP) in USB suspend mode. This allows greater margin for peripherals to meet the USB Suspend current limit of 500uA.
Programmable Receive Buffer Timeout: The TX buffer timeout is programmable over USB in 1ms increments from 1ms to 255ms, thus allowing the device to be better optimised for protocols requiring faster response times from short data packets.
Relaxed VCC Decoupling: The improved level of Vcc decoupling that was incorporated into BM devices has also been implemented in the FT2232D device.
Baud Rate Pre-Scaler Divisors: The FT2232D (UART mode) baud rate pre-scaler supports division by (n+0), (n+0.125), (n+0.25), (n+0.375), (n+0.5), (n+0.625), (n+0.75) and (n+0.875) where n is an integer between 2 and 16,384 (2^{14}).
Extended EEPROM Support: The FT2232D supports 93C46 (64 x 16 bit), 93C56 (128 x 16 bit), and 93C66 (256 x 16 bit) EEPROMs. The extra space is not used by the device. However it is available for use by other external MCU / logic whilst the FT2232D is being held in reset. There is now an additional 64 words of space available (128bytes total) in the user area when a 93C56 or 93C66 is used.
USB 2.0 (full speed option): An EEPROM based option allows the FT2232D to return a USB 2.0 device descriptor as opposed to USB 1.1. Note: The device would be a USB 2.0 Full Speed device (12Mb/s) as opposed to a USB 2.0 High Speed device (480Mb/s).

In addition to the BM chip features, the FT2232D incorporates the following new features and interface modes:-

Enhanced Asynchronous Bit-Bang Interface: The FT2232D supports FTDI's BM chip Bit Bang mode. In Bit Bang mode, the eight FIFO data lines can be switched between FIFO interface mode and an 8-bit Parallel IO port. Data packets can be sent to the device and they will be sequentially sent to the interface at a rate controlled by an internal time (equivalent to the baud rate prescaler). With the FT2232D device this mode has been enhanced so that the internal RD# and WR# strobes are now brought out of the device which can be used to allow external logic to be clocked by accesses to the Bit-Bang IO bus.

Synchronous Bit-Bang Interface: Synchronous Bit-Bang Mode differs from Asynchronous Bit-Bang mode in that the device is only read when it is written to. Thus making it easier for the controlling program to measure the response to an output stimulus as the data returned is synchronous to the output data.

High Output Drive Level Capability: The IO interface pins can be made to drive out at three times the standard drive level thus allowing multiple devices, or devices that require a greater drive strength to be interfaced to the FT2232D. This option is configured in the external EEPROM, ad can be set individually for each channel.

Multi-Protocol Synchronous Serial Engine Interface (M.P.S.S.E.): The Multi-Protocol Synchronous Serial Engine (MPSSE) interface is a new option designed to interface efficiently with synchronous serial protocols such as JTAG and SPI Bus. It is very flexible in that it can be configured for different industry standards, or proprietary bus protocols. For instance, it is possible to connect one of the FT2232D's channels to an SRAM configurable FPGA as supplied by vendors such as Altera and Xilinx. The FPGA device would normally be un-configured (i.e. have no defined function) at power-up. Application software on the PC could use the MPSSE to download configuration data to the FPGA over USB. This data would define the hardware's function on power up. The other FT2232 channel would be available for other devices. This approach would allow a customer to create a "generic" USB peripheral, whose hardware function can be defined under control of the application software. The FPGA based hardware could be easily upgraded or totally changed simply by changing the FPGA configuration data file. (See FTDI's MORPH-IC development module for a practical example, **www.morph-ic.com**)

MCU Host Bus Emulation: This new mode combines the 'A' and 'B' bus interface to make the FT2232D interface emulate a standard 8048 / 8051 style MCU bus. This allows peripheral devices for these MCU families to be directly attached to the FT2232D with IO being performed over USB with the help of MPSSE interface technology.

CPU-Style FIFO Interface: The CPU style FIFO interface is essentially the same function as the classic FT245 interface; however the bus signals have been redefined to make them easier to interface to a CPU bus.

Fast Opto-Isolated Serial Interface: A new proprietary FTDI protocol is designed to allow galvanically isolated devices to communicate synchronously with the FT2232D using just 4 signal wires (over two dual opto-isolators), and two power lines. The peripheral circuitry controls the data transfer rate in both directions, whilst maintaining full data integrity. Maximum USB full speed data rates can be achieved. Both 'A' and 'B' channels can communicate over the same 4 wire interface if desired.

4.1 Functional Block Descriptions

The following paragraphs detail each function within the FT2232D. Please refer to the block diagram shown in Figure 2.1.

3.3V LDO Regulator: The 3.3V LDO Regulator generates the 3.3 volt reference voltage for driving the USB transceiver cell output buffers. It requires an external decoupling capacitor to be attached to the 3V3OUT regulator output pin. It also provides 3.3V power to the RSTOUT# pin. The main function of this block is to power the USB Transceiver and the Reset Generator Cells rather than to power external logic. However, external circuitry requiring 3.3V nominal at a current of not greater than 5mA could also draw its power from the 3V3OUT pin if required

USB Transceiver: The USB Transceiver Cell provides the USB 1.1 or USB 2.0 full-speed physical interface to the USB cable. The output drivers provide 3.3 volt level slew rate control signalling, whilst a differential receiver and two single ended receivers provide USB data in, SEO and USB Reset condition detection.

USB DPLL: The USB DPLL cell locks on to the incoming NRZI USB data and provides separate recovered clock and data signals to the SIE block.

6MHz Oscillator: The 6MHz Oscillator cell generates a 6MHz reference clock input to the x8 Clock multiplier from an external 6MHz crystal or ceramic resonator.

x8 Clock Multiplier: The x8 Clock Multiplier takes the 6MHz input from the Oscillator cell and generates a 48MHz reference clock for the USB DPPL and the Baud Rate Generator blocks.

Serial Interface Engine (SIE): The Serial Interface Engine (SIE) block performs the Parallel to Serial and Serial to Parallel conversion of the USB data. In accordance to the USB 2.0 specification, it performs bit stuffing / un- stuffing and CRC5 / CRC16 generation / checking on the USB data stream.

USB Protocol Engine: The USB Protocol Engine manages the data stream from the device USB control endpoint. It handles the low level USB protocol (Chapter 9) requests generated by the USB host controller and the commands for controlling the functional parameters of the UART / FIFO controller blocks.

Dual Port TX Buffers (128 bytes): Data from the USB data out endpoint is stored in the Dual Port TX buffer and removed from the buffer to the transmit register under control of the UART FIFO controller.

Dual Port RX Buffers (384 bytes): Data from the UART / FIFO controller receive register is stored in the Dual Port RX buffer prior to being removed by the SIE on a USB request for data from the device data in endpoint.

Multi-Purpose UART / FIFO Controllers: The Multi-purpose UART / FIFO controllers handle the transfer of data between the Dual Port RX and TX buffers and the UART / FIFO transmit and receive registers. When configured as a UART it performs asynchronous 7 / 8 bit Parallel to Serial and Serial to Parallel conversion of the data on the RS232 (RS422 and RS485) interface. Control signals supported by UART mode include RTS, CTS, DSR, DTR, DCD and RI. There are also transmitter enable control signal pins (TXDEN) provided to assist with interfacing to RS485 transceivers. RTS/CTS, DSR/DTR and Xon/Xoff handshaking options are also supported. Handshaking, where required, is handled in hardware to ensure fast response times. The UART's also supports the RS232 BREAK setting and detection conditions.

Baud Rate Generator: The Baud Rate Generator provides a x16 clock input to the UART's from the 48MHz reference clock and consists of a 14 bit prescaler and 3 register bits which provide fine tuning of the baud rate (used to divide by a number plus a fraction). This determines the Baud Rate of the UART which is programmable from 183 baud to 3 million baud.

RESET Generator: The Reset Generator Cell provides a reliable power-on reset to the device internal circuitry on power up. An additional RESET# input and RSTOUT# output are provided to allow other devices to reset the FT2232D, or the FT2232D to reset other devices respectively. During reset, RSTOUT# is driven low, otherwise it drives out at the 3.3V provided by the onboard regulator. RSTOUT# can be used to control the 1.5K pull-up on USBDP directly where delayed USB enumeration is required. It can also be used to reset other devices. RSTOUT# will stay high- impedance for approximately 5ms after VCC has risen above 3.5V AND the device oscillator is running AND RESET# is high. RESET# should be tied to VCC unless it is a requirement to reset the device from external logic or an external reset generator I.C.

EEPROM Interface: When used without an external EEPROM the FT2232D be configured as a USB to dual serial port device. Adding an external 93C46 (93C56 or 93C66) EEPROM allows each of the chip's channels to be independently configured as a serial UART (232 mode), or a parallel FIFO (245 mode). The external EEPROM is used to enable the Fast Opto-Isolated Serial interface mode.

The external EEPROM can also be used to customise the USB VID, PID, Serial Number, Product Description Strings and Power Descriptor value of the FT2232D for OEM applications. Other parameters controlled by the EEPROM include Remote Wake Up, Soft Pull Down on Power-Off and USB 2.0 descriptor modes. The EEPROM should be a 16 bit wide

configuration such as a MicroChip 93LC46B or equivalent capable of a 1Mb/s clock rate at VCC = 4.35V to 5.25V. The EEPROM is programmable-on board over USB using a utility program available from FTDI's web site (**www.ftdichip. com**). This allows a blank part to be soldered onto the PCB and programmed as part of the manufacturing and test process. If no EEPROM is connected (or the EEPROM is blank), the FT2232D will default to dual serial ports. The device uses its built-in default VID, PID Product Description and Power Descriptor Value. In this case, the device will not have a serial number as part of the USB descriptor.

5 Devices Characteristics and Ratings

5.0 Absolute Maximum Ratings

These are absolute maximum ratings for the for the FT2232D device in accordance with the Absolute Maximum Rating System (IEC 60134). Exceeding these may cause permanent damage to the device.

Parameter	Value	Unit
Storage Temperature	-65°C to 150°C	Degrees C
Floor Life (Out of Bag) At Factory Ambient (30°C / 60% Relative Humidity)	192 (IPC/JEDEC J-STD-033A MSL Level 3 Compliant)**Note 10	Hours
Ambient Temperature (Power Applied)	-40°C to 85°C	Degrees C
VCC Supply Voltage	-0.5 to +6.00	V
DC Input Voltage – USBDP and USBDM	-0.5 to +3.8	V
DC Input Voltage – High Impedance Bidirectional	-0.5 to + (VCC +0.5)	V
DC Input Voltage – All Other Inputs	-0.5 to + (VCC +0.5)	V
DC Output Current – Outputs	24	mA
DC Output Current – Low Impedance Bidirectional	24	mA
Power Dissipation (VCC = 5.25V)	500	mW
Electrostatic Discharge Voltage (Human Body Model) (I < 1µA)	+/- 3000	V
Latch Up Current (Vi = +/- 10V maximum, for 10 ms	200	mA

Table 5.1 Absolute Maximum Ratings

****Note 10** if devices are stored out of the packaging beyond this time limit the devices should be baked before use. The devices should be ramped up to a temperature of 110°C and baked for 8 to 10 hours.

5.1 DC Characteristics

DC Characteristics (Ambient Temperature = -40°C to +85°C)

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
VCC1	VCC Operating Supply Voltage	4.35	5.0	5.25	V	
VCC2	VCCIO Operating Supply Voltage	3.0	-	5.25	V	
Icc1	Operating Supply Current	---	30	---	mA	Normal Operation
Icc2	Operating Supply Current	---	100	200	µA	USB Suspend ****Note 11**

Table 5.2 Operating Voltage and Current

****Note 11** - Supply current excludes the 200µA nominal drawn by the external pull-up resistor on USBDP.

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
Voh	Output Voltage High	3.2	4.1	4.9	V	I source = 2mA
Vol	Output Voltage Low	0.3	0.4	0.6	V	I sink = 2mA
Vin	Input Switching Threshold	1.3	1.2	1.5	V	
VHys	Input Switching Hysteresis	50	25	30	mV	

Table 5.3 IO Pin Characteristics (VCCIO = 5.0V, Standard Drive Level) **Note 12

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
Voh	Output Voltage High	2.2	2.7	3.2	V	I source = 1mA
Vol	Output Voltage Low	0.3	0.4	0.5	V	I sink = 2mA
Vin	Input Switching Threshold	1.0	1.2	1.5	V	
VHys	Input Switching Hysteresis	20	25	30	mV	

Table 5.4 IO Pin Characteristics (VCCIO = 3.0V – 3.6V, Standard Drive Level) **Note 12

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
Voh	Output Voltage High	3.2	4.1	4.9	V	I source
Vol	Output Voltage Low	0.3	0.4	0.6	V	I sink = 6 mA
Vin	Input Switching Threshold	1.3	1.6	1.9	V	
VHys	Input Switching Hysteresis	50	55	60	mV	

Table 5.5 IO Pin Characteristics (VCCIO = 5.0V, High Drive Level) **Note 12 and 13

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
Voh	Output Voltage High	2.2	2.8	3.2	V	I source = 3mA
Vol	Output Voltage Low	0.3	0.4	0.6	V	I sink = 8 mA
Vin	Input Switching Threshold	1.0	1.2	1.5	V	
VHys	Input Switching Hysteresis	20	25	30	mV	

Table 5.6 IO Pin Characteristics (VCCIO = 3.0V -3.6V, High Drive Level) **Note 12 and 13

****Note 12:** Inputs have an internal 200K pull-up resistor to VCCIO, which can alternativly be programmed to pull down using a configuration bit in the external EEPROM.

****Note 13:** The high output drive level is configured in the external EEPROM. Each channel can be configured individually.

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
Voh	Output Voltage High	4.0	-	5.0	V	Fosc = 6MHz
Vol	Output Voltage Low	0.1	-	1.0	V	Fosc = 6MHz
Vin	Input Switching Threshold	1.8	2.5	3.2	V	
Voh	Output Voltage High	4.0	-	5.0	V	Fosc = 6MHz

Table 5.7 XTIN / XTOUT Pin Characteristics

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
Voh	Output Voltage High	3.2	4.1	4.9	V	I source = 2mA
Vol	Output Voltage Low	0.3	0.4	0.6	V	I sink = 2 mA
Vin	Input Switching Threshold	1.3	1.6	1.9	V	
VHys	Input Switching Hysteresis	50	55	60	mV	

Table 5.8 RESET# and TEST EECS, EESK, EEDATA Pin Characteristics **Note 14

****Note 14** - EECS, EESK, EEDATA and RESET# pins have an internal 200K pull-up resistor to VCC

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
Voh	Output Voltage High	3.0	-	3.6	V	I source = 2mA
Vol	Output Voltage Low	0.3	-	0.6	V	I sink = 2mA

Table 5.9 RSTOUT# Pin Characteristics

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
UVoh	I/O Pins Static Output (High)	2.8	-	3.6	V	RI = 1.5kΩ to 3V3OUT (D+) RI = 15KΩ to GND (D-)
UVol	I/O Pins Static Output (Low)	0	-	0.3	V	RI = 1.5kΩ to 3V3OUT (D+) RI = 15kΩ to GND (D-)
UVse	Single Ended Rx Threshold	0.8	-	2.0	V	
UCom	Differential Common Mode	0.8	-	2.5	V	
UVDif	Differential Input Sensitivity	0.2	-	-	V	
UDrvZ	Driver Output Impedance	26	-	44	Ohms	

Table 5.10 USB I/O Pin (USBDP, USBDM) Characteristics **Note 15

****Note 15** - Driver Output Impedance includes the external 27R series resistors on USBDP and USBDM pins.

5.2 ESD Tolerance

ESD protection for FT2232D IO's

Parameter	Reference	Minimum	Typical	Maximum	Units
Human Body Model (HBM)	JEDEC EIA/JESD22-A114-B, Class 2		±2kV		kV
Machine Mode (MM)	JEDEC EIA/JESD22-A115-A, Class A		±100V		V
Charge Device Model (CDM)	JEDEC EIA/ JESD22-C101-D, Class-III		±500V		V
Latch-up	JESD78, Trigger Class-II		±200mA		mA

Table 5.11 ESD Tolerance

6 USB Power Configurations

The following sections illustrate possible USB power configurations for the FT2232D.

6.0 USB Bus Powered Configuration

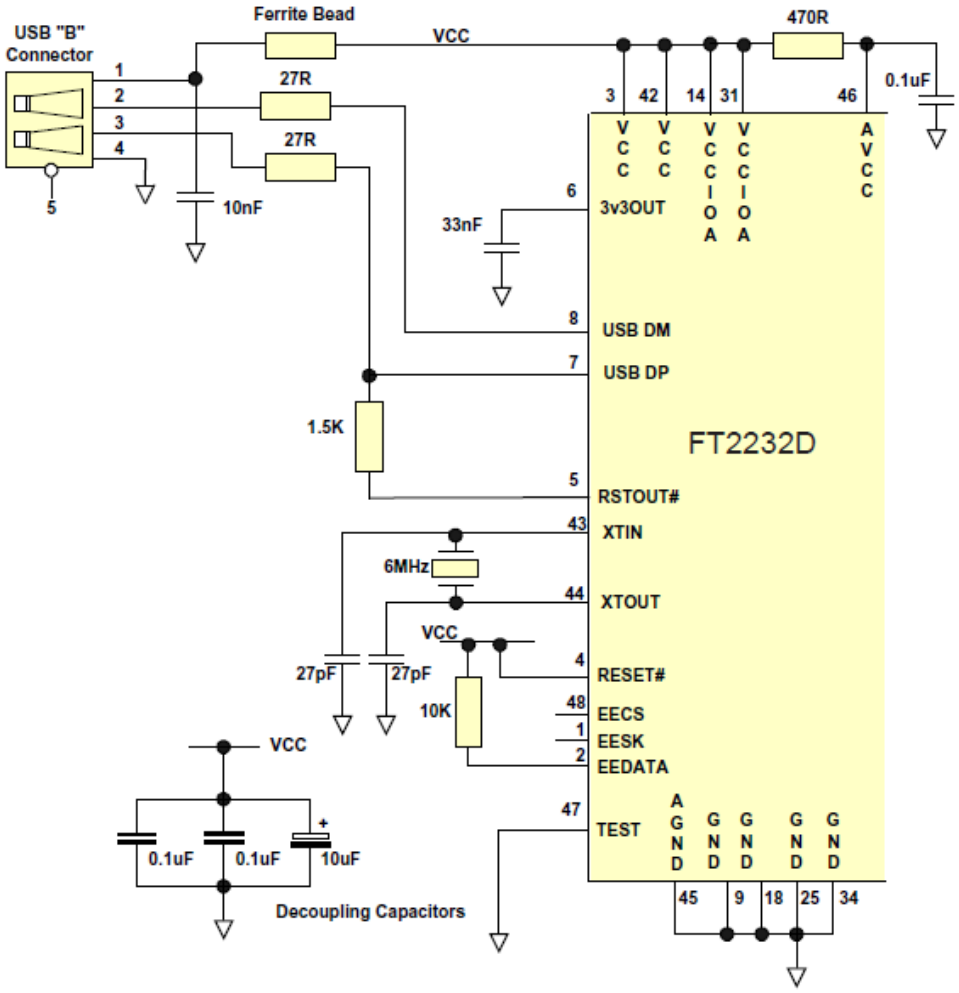

Figure 6.1 Bus Powered Configuration

Figure 6.1 illustrates the FT2232D in a typical USB bus powered configuration. A USB Bus Powered device gets its power from the USB bus. Basic rules for USB Bus power devices are as follows:-

a) On plug-in, the device must draw no more than 100mA

b) On USB Suspend the device must draw no more than 500µA.

c) A High Power USB Bus Powered Device (one that draws more than 100mA) should use the PWREN# pin to keep the current below 100mA on plug-in and 500µA on USB suspend.

d) A device that consumes more than 100mA cannot be plugged into a USB Bus Powered Hub

e) No device can draw more that 500mA from the USB Bus.

The power descriptor in the EEPROM should be programmed to match the current draw required by the device. A Ferrite Bead is connected in series with USB power to prevent noise from the device and associated circuitry (EMI) being radiated down the USB cable to the Host. The value of the Ferrite Bead depends on the total current required by the circuit - a suitable range of Ferrite Beads is available from Steward (**www.steward.com**) for example Steward Part # MI0805K400R-00 also available from DigiKey, Part # 240-1035-1.

6.1 USB Self Powered Configuration

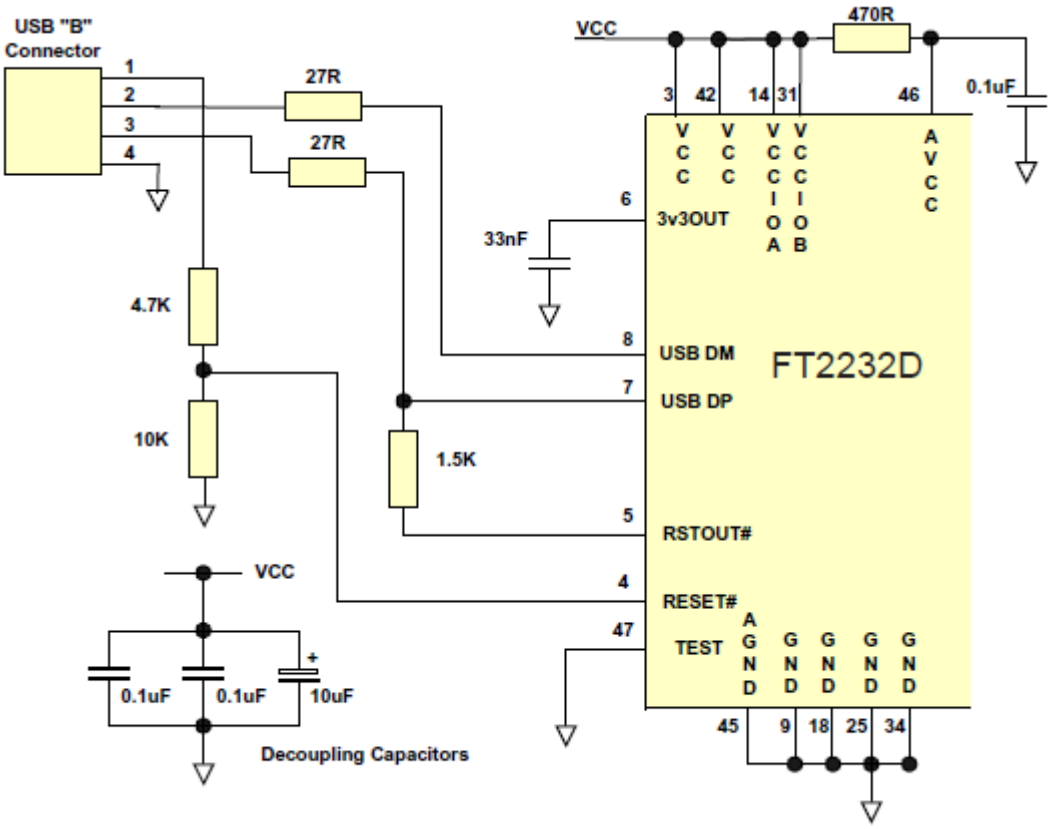

Figure 6.2 Self Powered Configuration

Figure 6.2 illustrates the FT2232D in a typical USB self powered configuration. A USB Self Powered device gets its power from its own POWER SUPPLY and does not draw current from the USB bus. The basic rules for USB Self power devices are as follows –

a) A Self-Powered device should not force current down the USB bus when the USB Host or Hub Controller is powered down.
b)A Self Powered Device can take as much current as it likes during normal operation and USB suspend as it has its own POWER SUPPLY.
c)A Self Powered Device can be used with any USB Host and both Bus and Self Powered USB Hubs.

The USB power descriptor option in the EEPROM should be programmed to a value of zero (self powered). To meet requirement a) the 1.5 K pull-up resistors on USBDP is connected to RSTOUT# as per the bus-power circuit. However, the USB Bus Power is used to control the RESET# Pin of the FT2232D device. When the USB Host or Hub is powered up RSTOUT# will pull the 1.5K resistor on USBDP to 3.3V, thus identifying the device as a full speed device to USB. When the USB Host or Hub power is off, RESET# will go low and the device will be held in reset. As RESET# is low, RSTOUT# will also be low, so no current will be forced down USBDP via the 1.5K pull-up resistor when the host or hub is powered down. Failure to do this may cause some USB host or hub controllers to power up erratically. Note: When the FT2232D is in reset, the I/O interface pins all go tri-state. These pins have internal 200K pull-up resistors to VCCIO, so they will gently pull high unless driven by some external logic.

6.2 Interfacing to 3.3V Logic

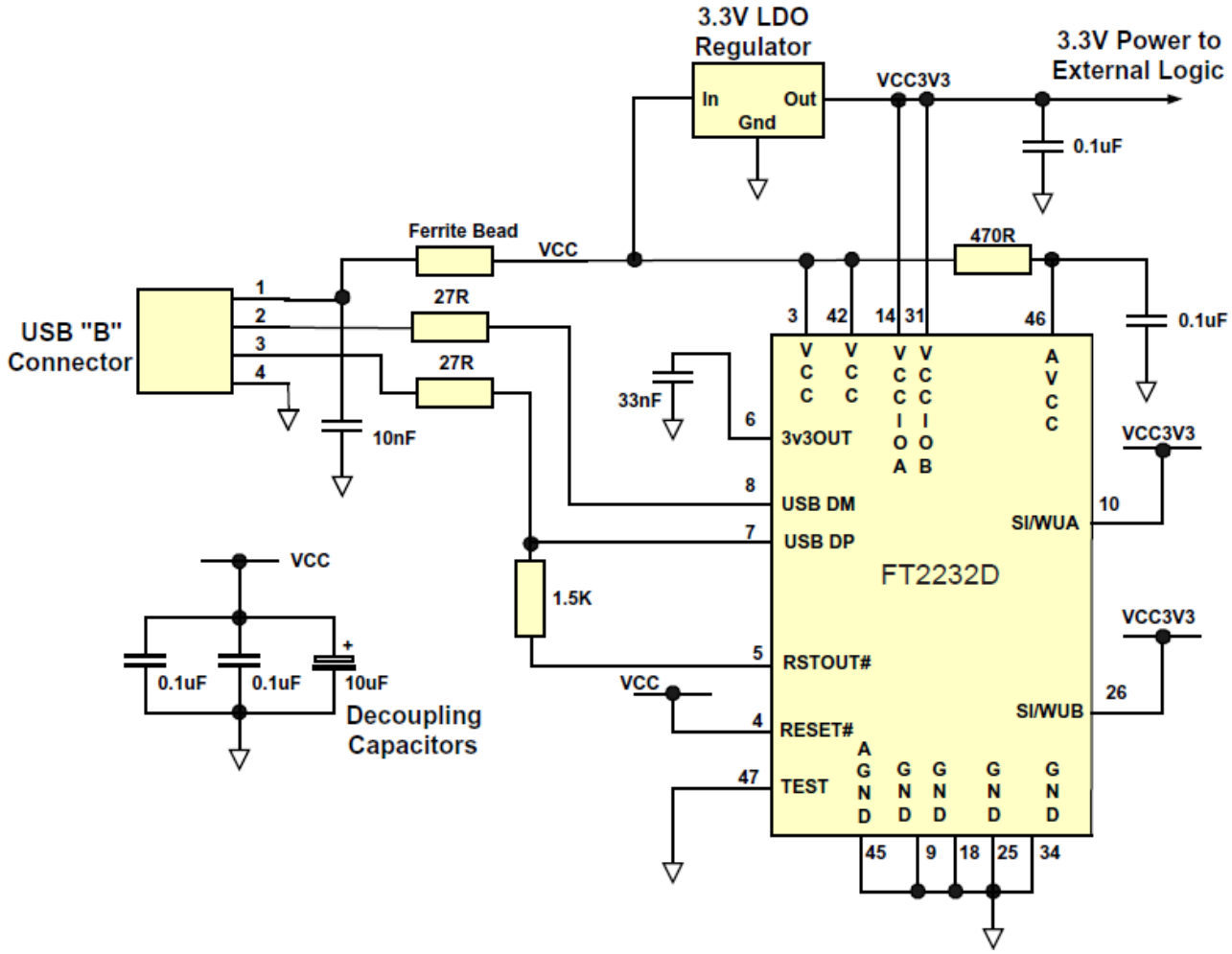

Figure 6.3 Bus Powered Circuit with 3.3V logic drive and IO supply voltage

Figure 6.3 shows how to configure the FT2232D to interface with 3.3V logic devices. In this example, a discrete 3.3V regulator is used to supply the 3.3V logic from the USB supply. VCCIOA and VCCIOB are connected to the output of the 3.3V regulator, which in turn will cause the device interface IO pins on both channels to drive out at 3.3V level. It is also possible to have one IO interface channel driving out at 5V level, and the other at 3.3V level. In this case one of the VCCIO pins would be connected to 5V, and the other connected to 3.3V.

For USB bus powered circuits some considerations have to be taken into account when selecting the regulator:-

a) The regulator must be capable of sustaining its output voltage with an input voltage of 4.35 volts. A Low Drop Out (LDO) regulator must be selected.

b) The quiescent current of the regulator must be low in order to meet the USB suspend total current requirement of <= 500µA during USB suspend.

An example of a regulator family that meets these requirements is the MicroChip (Telcom) TC55 Series. These devices can supply up to 250mA current and have a quiescent current of under 1µA. In some cases, where only a small amount of current is required (< 5mA) , it may be possible to use the in-built regulator of the FT2232D to supply the 3.3V without any other components being required. In this case, connect VCCIOA or VCCIOB to the 3V3OUT pin of the FT2232D.

Note: It should be emphasised that the 3.3V supply for VCCIO in a bus powered design with a 3.3V logic interface should come from an LDO which is supplied by the USB bus, or from the 3V3OUT pin of the FT232BM, and not from any other source. Please also note that if the SI/WU pins are not being used they should be pulled up to the same supply as their respective VCCIO pin.

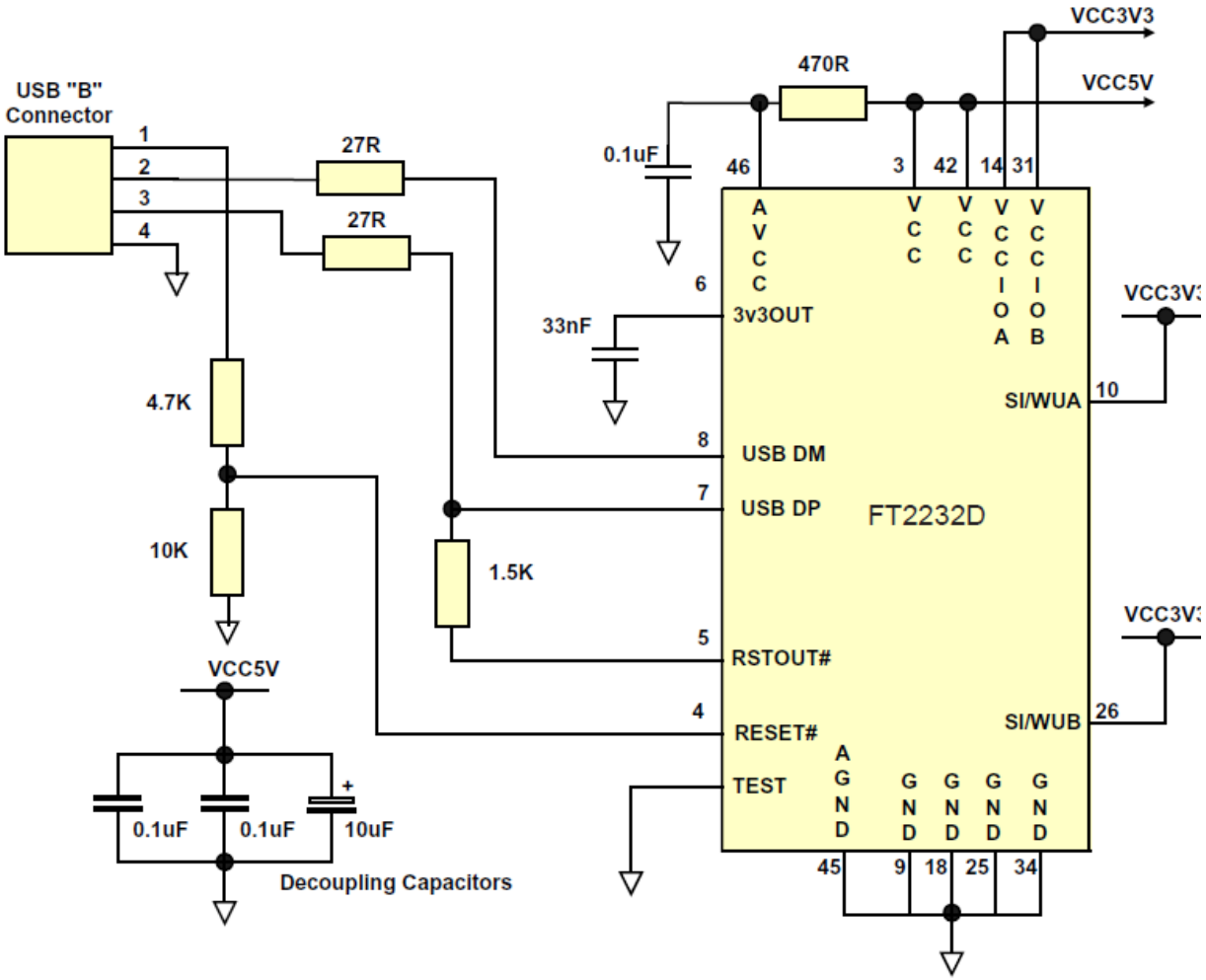

Figure 6.4 Self Powered Circuit with 3.3V logic drive and IO supply voltage

Figure 6.4 is an example of a FT2232D USB self powered design with 3.3V interface. In this case the VCCIOA and VCCIOB pins are supplied by an external 3.3V supply in order to make both of the device's IO channels drive out at 3.3V logic level, thus allowing them to be connected to a 3.3V MCU or other external logic. It is also possible to have one IO interface channel driving out at 5V level, and the other at 3.3V level. In this case one of the VCCIO pins would be connected to 5V, and the other connected to 3.3V. A USB self powered design uses its own power supplies, and does not draw any of its power from the USB bus. In such cases, no special care need be taken to meet the USB suspend current (0.5 mA) as the device does not get its power from the USB port. As with bus powered 3.3V interface designs, in some cases, where only a small amount of current is required (<5mA), it may be possible to use the in-built regulator of the FT2232D to supply the 3.3V without any other components being required. In this case, connect VCCIOA or VCCIOB to the 3V3OUT pin of the FT2232D. Note that if the SI/WU pins are not being used they should be pulled up to the same supply as their respective VCCIO pin.

6.3 Power Switching Configuration

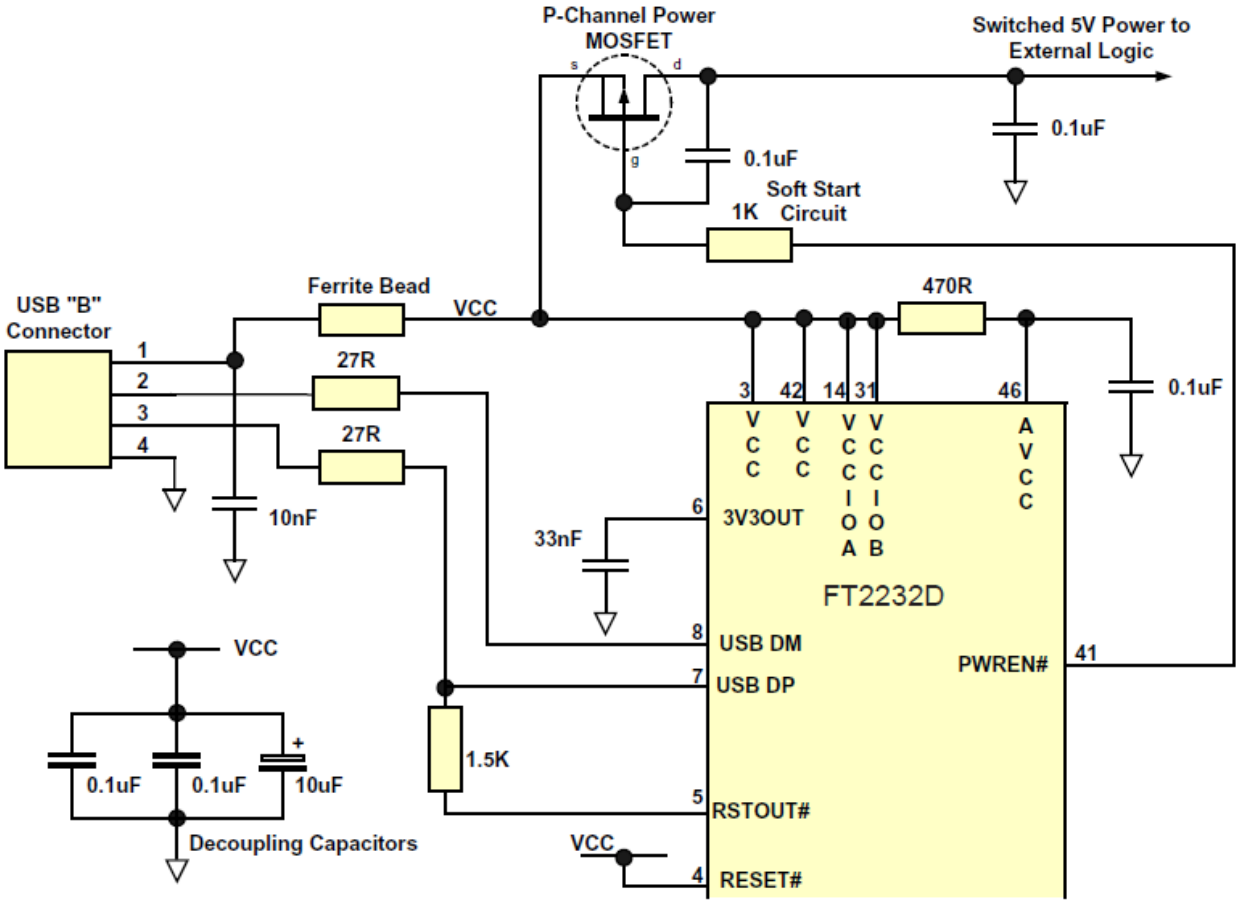

Figure 6.5 Bus Powered Circuit with Power Control

USB Bus powered circuits need to be able to power down in USB suspend mode in order to meet the <= 500µA total suspend current requirement (including external logic). Some external logic can power itself down into a low current state by monitoring the PWREN# pin. For external logic that cannot power itself down in that way, the FT2232D provides a simple but effective way of turning off power to external circuitry during USB suspend.

Figure 6.5 shows how to use a discrete P-Channel Logic Level MOSFET to control the power to external logic circuits. A suitable device would be an International Rectifier (**www.irf.com**) IRLML6402, or equivalent. It is recommended that a "soft start" circuit consisting of a 1K series resistor and a 0.1 µF capacitor are used to limit the current surge when the MOSFET turns on. Without the soft start circuit there is a danger that the transient power surge of the MOSFET turning on will reset the FT2232D, or the USB host / hub controller. The values used here allow attached circuitry to power up with a slew rate of ~12.5 V per millisecond, in other words the output voltage will transition from GND to 5V in approximately 400 microseconds.

Alternatively, a dedicated power switches I.C. with inbuilt "soft-start" can be used instead of a MOSFET. A suitable power switch I.C. for such an application would be a Micrel (**www.micrel.com**) MIC2025-2BM or equivalent. Please note the following points in connection with power controlled designs: –

a)The logic to be controlled must have its own reset circuitry so that it will automatically reset itself when power is re-applied on coming out of suspend.

b)Set the Pull-down on Suspend option in the FT2232D's EEPROM.

c)For USB high-power bus powered device (one that consumes greater than 100 mA, and up to 500 mA of current from the USB bus), the power consumption of the device should be set in the max power field in the EEPROM. A high-power bus powered device must use this descriptor in the EEPROM to inform the system of its power requirements.

d)For 3.3V power controlled circuits the VCCIO pins must not be powered down with the external circuitry. Either connect the power switch between the output of the 3.3V regulator and the external 3.3V logic, or if appropriate power the VCCIO pin from the 3V3OUT pin of the FT2232D.

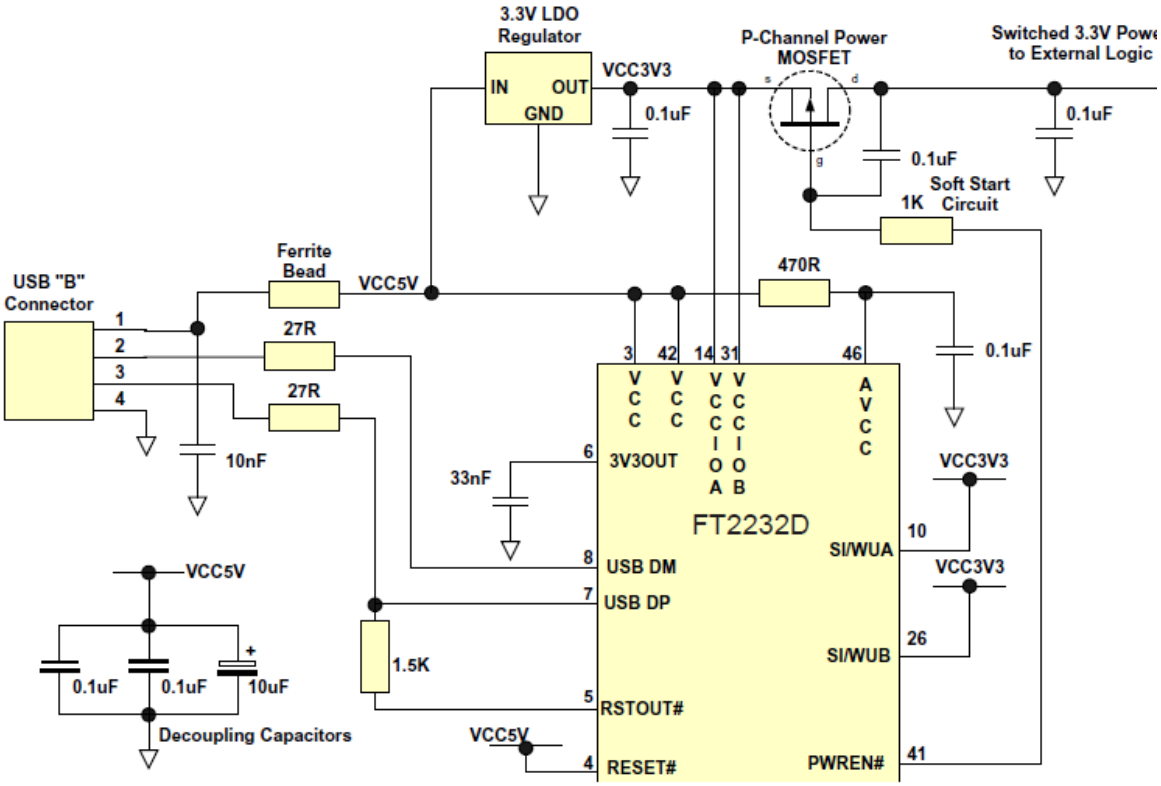

Figure 6.6 Bus Powered Circuit with Power Control and 3.3V Logic Drive/ IO Supply Voltage

Figure 6.6 is a FT2232D design example which effectively combines the circuits shown in Figures 6.4 and 6.5 to give a USB bus powered design with power switching and 3.3V logic drive level on both channels. Once again a P-Channel Power MOSFET and soft start circuit are used to control the power to external logic devices. A 3.3V LDO regulator which is supplied by the USB bus is used to provide the 3.3V supply for the VCCIO pins, as well as the external logic. If the SI/WU pins are not being used they should be pulled up to 3.3V.

7 Standard Device Configuration Examples

7.0 Oscillator Configurations

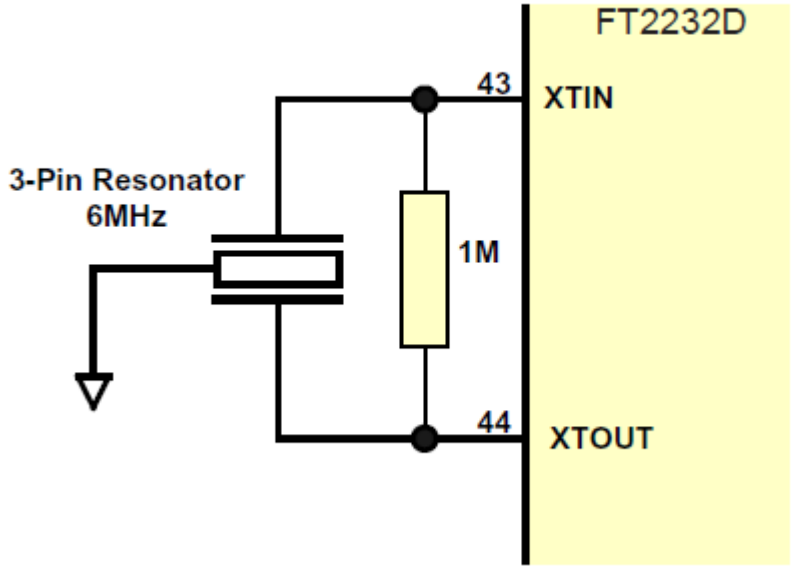

Figure 7.1 3-Pins Ceramic Resonator Configuration

Figure 7.1 illustrates how to use the FT2232D with a 3-Pin Ceramic Resonator. A suitable part would be a ceramic resonator from Murata's CERALOCK range. (Murata Part Number CSTCR6M00G15), or equivalent. 3-Pin ceramic resonators have the load capacitors built into the resonator so no external loading capacitors are required. This makes for an economical configuration. The accuracy of this Murata ceramic resonator is +/- 0.25% and it is specifically designed for USB full speed applications. A 1 MegaOhm loading resistor across XTIN and XTOUT is recommended in order to guarantee this level of accuracy. Other ceramic resonators with a lesser degree of accuracy (typically +/- 0.5%) are technically out-with the USB specification, but it has been calculated that using such a device will work satisfactorily in practice with a FT2232D design. An example of such a device is Murata's CSTLSM00G53.

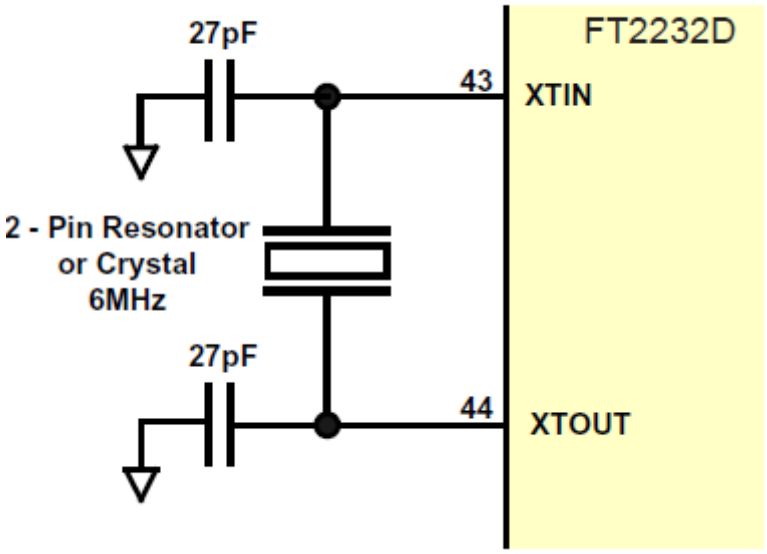

Figure 7.2 Crystal or 2-Pin Ceramic Resonator Configuration

Figure 7.2 illustrates how to use the FT2232D with a 6MHz Crystal or 2-Pin Ceramic Resonator. In this case, these devices do not have in-built loading capacitors so these have to be added between XTIN, XTOUT and GND as shown. A value of 27pF is shown as the capacitor in the example – this will be good for many crystals and some resonators but do select the value based on the manufacturers recommendations wherever possible. If using a crystal, use a parallel cut type. If using a resonator, see the previous note on frequency accuracy.
It is also possible to use a 6 MHz Oscillator with the FT2232D. In this case the output of the oscillator would be connected to XTIN, and XTOUT should be left unconnected. The oscillator must have a CMOS output.

7.1 EEPROM Configurations

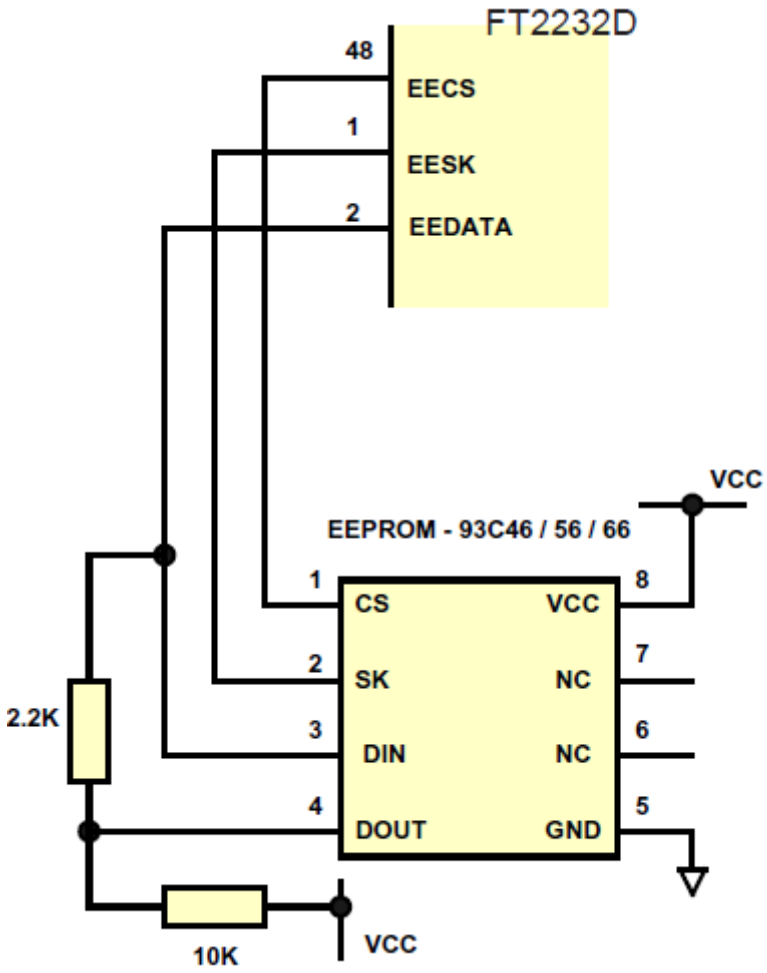

Figure 7.3 EEPROM Configuration

Figure 7.3 illustrates how to connect the FT2232D to the 93C46 (93C56 or 93C66) EEPROM. EECS (pin 48) is directly connected to the chip select (CS) pin of the EEPROM. EESK (pin 1) is directly connected to the clock (SK) pin of the EEPROM. EEDATA (pin 2) is directly connected to the Data In (Din) pin of the EEPROM. There is a potential condition whereby both the Data Output (Dout) of the EEPROM can drive out at the same time as the EEDATA pin of the FT2232D. To prevent potential data clash in this situation, the Dout of the EEPROM is connected to EEDATA of the FT2232D via a 2.2K resistor.

Following a power-on reset or a USB reset, the FT2232D will scan the EEPROM to find out (a) if an EEPROM is attached to the Device and (b) if the data in the device is valid. If both of these are the case, then the FT2232D will use the data in the EEPROM, otherwise it will use its built-in default values and configuration. The default port configuration of the FT2232D puts both Channel A and Channel B into serial UART mode.

When a valid command is issued to the EEPROM from the FT2232D, the EEPROM will acknowledge the command by pulling its Dout pin low. In order to check for this condition, it is necessary to pull Dout high using a 10K resistor. If the command acknowledge doesn't happen then EEDATA will be pulled high by the 10K resistor during this part of the cycle and the device will detect an invalid command or no EEPROM present.

There are two varieties of 93C46/56/66 EEPROM's on the market – one is configured as being 16 bits wide, the other is configured as being 8 bits wide. These are available from many sources such as Microchip, STMicro, ISSI etc. The FT2232D requires EEPROM's with a 16-bit wide configuration such as the Microchip 93LC46B device. The EEPROM must be capable of reading data at a 1Mb clock rate at a supply voltage of 4.35V to 5.25V. Most available parts are capable of this. Check the manufacturers data sheet to find out how to connect pins 6 and 7 of the EEPROM. Some devices specify these as no-connect, others use them for selecting 8 / 16 bit mode or for test functions. Some other parts have their pinout rotated by 90o so please select the required part and its options carefully.

It is possible to "share" the EEPROM between the FT2232D and another external device such as an MCU. However, this can only be done when the FT2232D is in its reset condition as it tri-states its EEPROM interface at that time. A typical configuration would use four bits of an MCU IO Port. One bit would be used to hold the FT2232D reset (using RESET#) on power-up, the other three would connect to the EECS, EESK and EEDATA pins of the FT2232D in order to read / write data to the EEPROM at this time. Once the MCU has read / written the EEPROM, it would take RESET# high to allow the FT2232D to configure itself and enumerate over USB.

The external EEPROM can be programmed over USB using utility software provided by FTDI. The external EEPROM is used to enable 245 FIFO, and Fast Opto-Isolated Serial interface modes on each channel.

8 Signal Descriptions by IO Mode and Interface Channel Configurations

8.0 232 UART Interface Mode Signal Descriptions and Interface Configurations

When either Channel A or Channel B is in 232 UART mode, the IO signal lines are configured as follows:-

Pin#		*Signal*	*Type*	*Description*
Channel A	*Channel B*			
24	40	TXD	OUTPUT	Transmit Asynchronous Data Output
23	39	RXD	INPUT	Receive Asynchronous Data Input **Note 16
22	38	RTS#	OUTPUT	Request To Send Control Output / Handshake signal
21	37	CTS#	INPUT	Clear To Send Control Input / Handshake signal **Note 16
20	36	DTR#	OUTPUT	Data Terminal Ready Control Output / Handshake signal
19	35	DSR#	INPUT	Data Set Ready Control Input / Handshake signal **Note 16
17	33	DCD#	INPUT	Data Carrier Detect Control Input **Note 16
16	32	RI#	INPUT	Ring Indicator Control Input. When the Remote Wake up option is enabled in the EEPROM, taking RI# low can be used to resume the PC USB Host controller from suspend. **Note 16
15	30	TXDEN	OUTPUT	Enable Transmit Data for RS485
13	29	SLEEP#	OUTPUT	Goes low during USB Suspend Mode. Typically used to power-down an external TTL to RS232 level converter I.C. in USB to RS232 converter designs.
12	28	RXLED#	O.C.	LED Drive - Pulses Low when Transmitting Data via USB.
11	27	TXLED#	O.C	LED Drive - Pulses Low when Receiving Data via USB.
10	26	SI/WU	INPUT	The Send Immediate / WakeUp signal combines two functions on a single pin. If USB is in suspend mode (PWREN# = 1) and remote wakeup is enabled in the EEPROM , strobing this pin low will cause the device to request a resume on the USB Bus. Normally, this can be used to wake up the Host PC. During normal operation (PWREN# = 0), if this pin is strobed low any data in the device TX buffer will be sent out over USB on the next Bulk-IN request from the drivers regardless of the pending packet size. This can be used to optimise USB transfer speed for some applications. Tie this pin to VCCIO if not used

Table 8.1 232 UART mode the IO signal lines Description

Note 16: These pins are pulled to up VCCIO via internal 200K resistors during Reset and USB Suspend mode. These can be programmed to gently pull low during USB suspend (PWREN# ="1") by setting this option in the EEPROM.

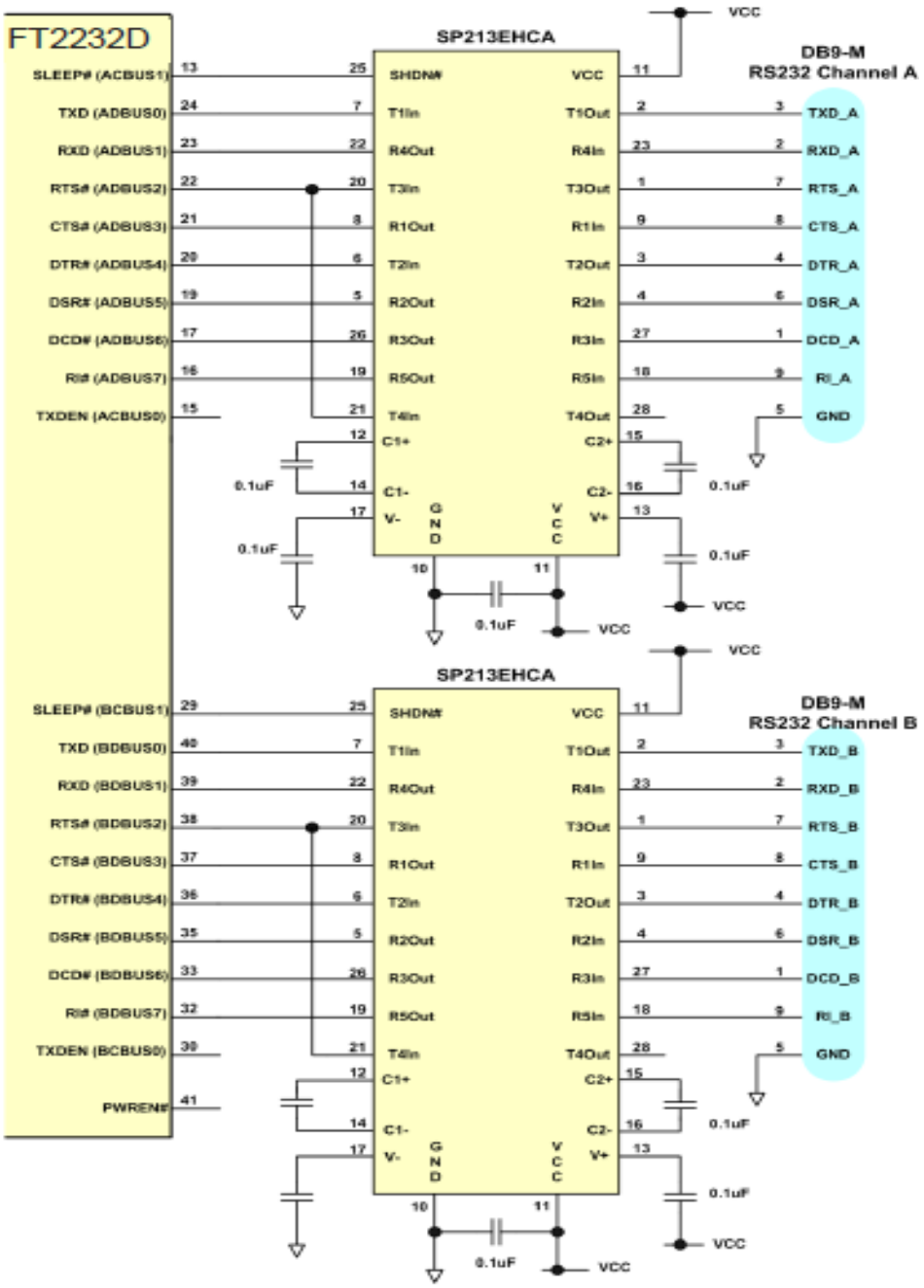

Figure 8.1 USB <=> Dual Port RS232 Converter Configuration

Figure 8.1 illustrates how to connect the FT2232D, when both channels A and B are configured as 232-style UART interfaces, to two TTL – RS232 Level Converter I.C.'s to make a USB <=> Dual Port RS232 converter using the popular "213" series of TTL to RS232 level converters. These devices have 4 transmitters and 5 receivers in a 28-LD SSOP package and feature an in-built voltage converter to convert the 5V (nominal) VCC to the +/- 9 volts required by RS232. An important feature of these devices is the SHDN# pin which can power down the device to a low quiescent current during USB suspend mode.
The device used in the example above is a Sipex SP213EHCA which is capable of RS232 communication at up to 500K baud. If a lower baud rate is acceptable, then several pin compatible alternatives are available such as the Sipex SP213ECA , the Maxim MAX213CAI and the Analog Devices ADM213E, which are all good for communication at up to 115,200 baud. If a higher baud rate is desired, use a Maxim MAX3245CAI part which is capable of RS232 communication at rates of up to 1M baud. The MAX3245 is not pin compatible with the 213 series devices, also its SHDN pin is active high, so connect it to PWREN# instead of SLEEP#. Dual RS232 level converters such as the Maxim MAX3187 may also be a suitable alternative.

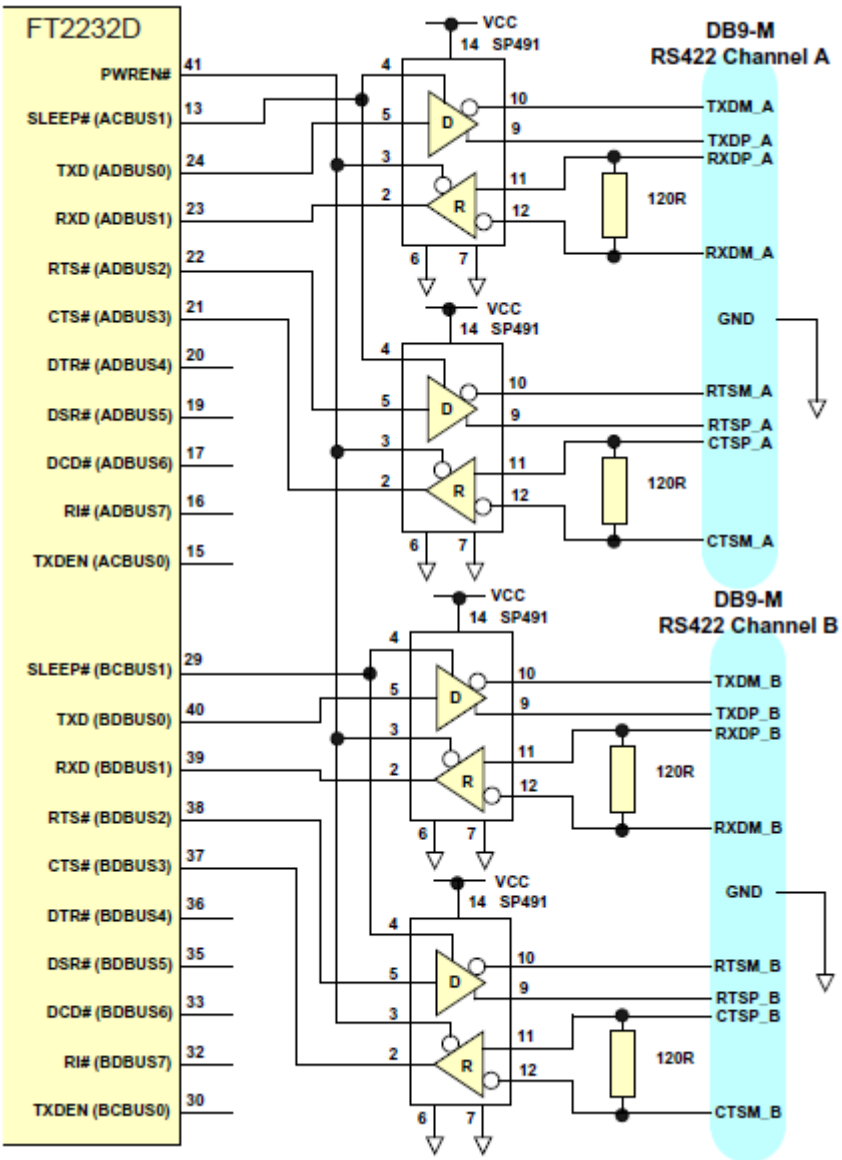

Figure 8.2 USB <=> Dual Port RS422 Converter Configuration

Figure 8.2 illustrates how to connect the UART interfaces of the FT2232D to two TTL - RS422 Level Converter I.C.'s to make a USB to dual port RS422 converter. There are many such level converter devices available - this example uses two Sipex SP491 devices which have enables on both their transmitters and receivers. Because the transmitter enables are active high, they are connected to the SLEEP# pins.

The receiver enables are active low and are both connected to the PWREN# pin. This ensures that both the transmitters and receivers are enabled when the device is active, and disabled when the device is in USB suspend mode. If the design is USB BUS powered, it may be necessary to use a P-Channel logic level MOSFET (controlled by PWREN#) in the VCC line of the SP491 devices to ensure that the USB standby current of 500µA is met. The SP491 is good for sending and receiving data at a rate of up to 5M Baud - in this case the maximum rate is limited to 3M Baud by the FT2232D.

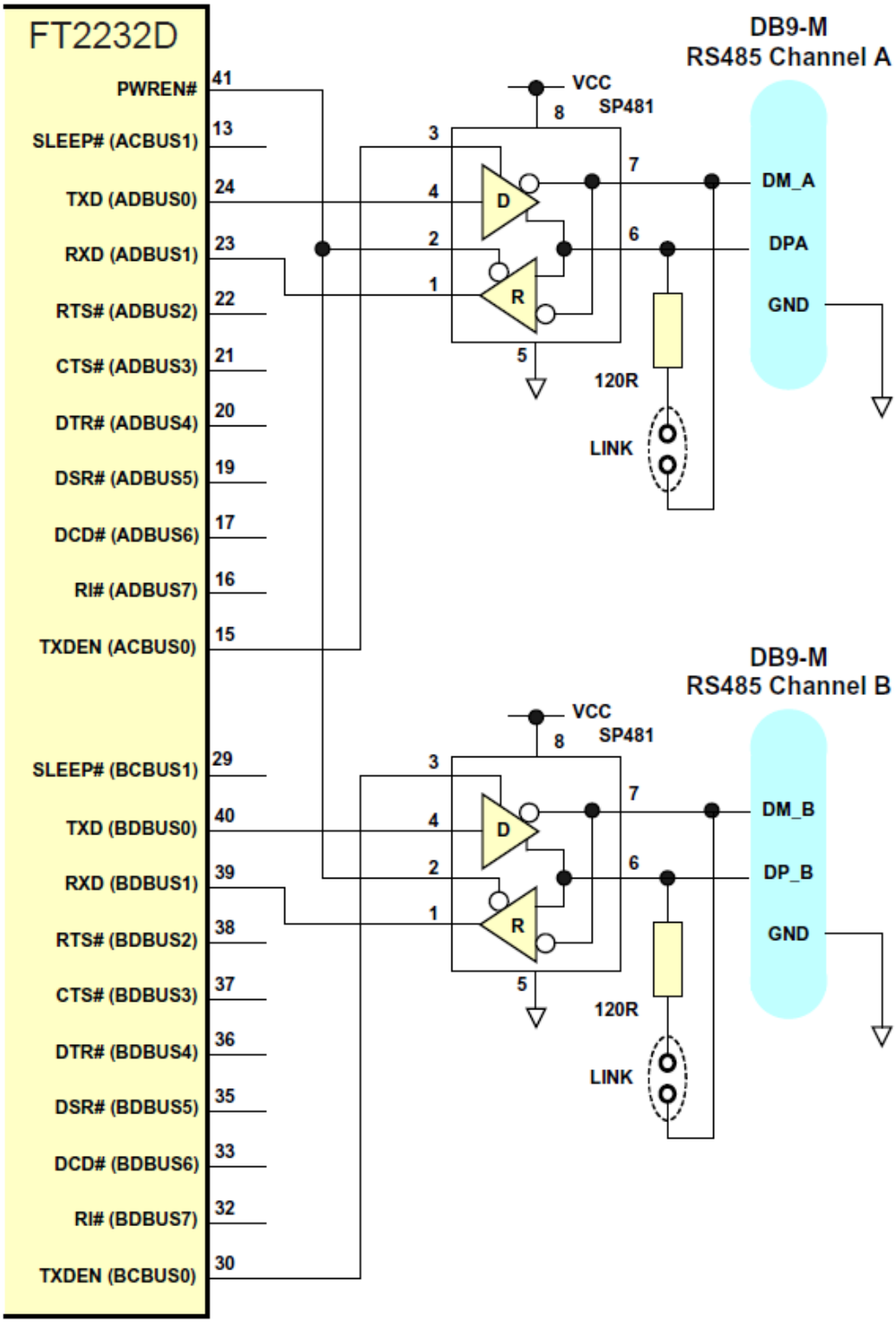

Figure 8.3 USB <=> Dual Port RS485 Converter Configuration

Figure 8.3 illustrates how to connect the UART interfaces of the FT2232D to two TTL – RS485 Level Converter I.C.'s to make a USB to dual port RS485 converter. This example uses two Sipex SP491 devices but there are similar parts available from Maxim and Analog Devices amongst others. The SP491 is a RS485 device in a compact 8 pin SOP package. It has separate enables on both the transmitter and receiver. With RS485, the transmitter is only enabled when a character is being transmitted from the UART. The TXDEN pins on the FT2232D are provided for exactly that purpose, and so the transmitter enables are wired to the TXDEN's. The receiver enable is active low, so it is wired to the PWREN# pin to disable the receiver when in USB suspend mode.

RS485 is a multi-drop network – i.e. many devices can communicate with each other over a single two wire cable connection. The RS485 cable requires to be terminated at each end of the cable. Links are

provided to allow the cable to be terminated if the device is physically positioned at either end of the cable.
In this example the data transmitted by the FT2232D is also received by the device that is transmitting. This is a common feature of RS485 and requires the application software to remove the transmitted data from the received data stream. With the FT2232D it is possible to do this entirely in hardware – simply modify the schematic so that RXD of the FT2232D is the logical OR of the SP481 receiver output with TXDEN using an HC32 or similar logic gate.

8.1 232 UART Mode LED Interface

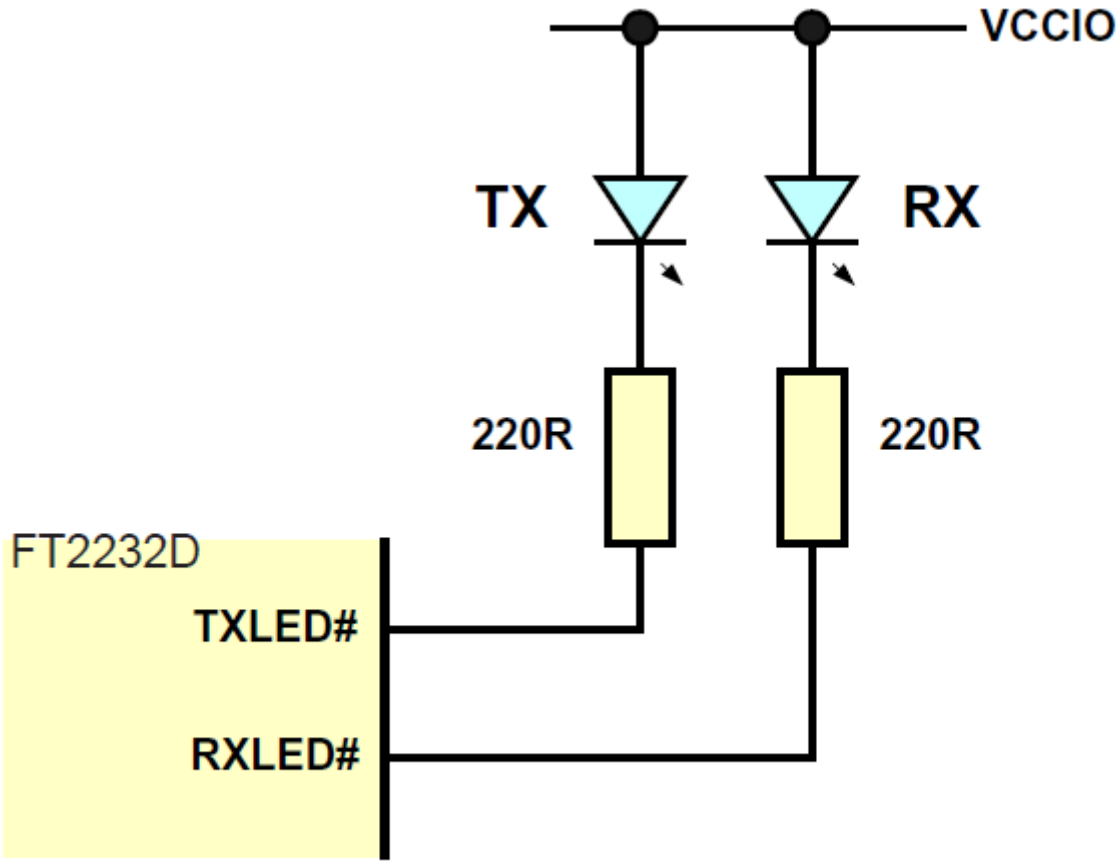

Figure 8.4 Dual LED Configuration

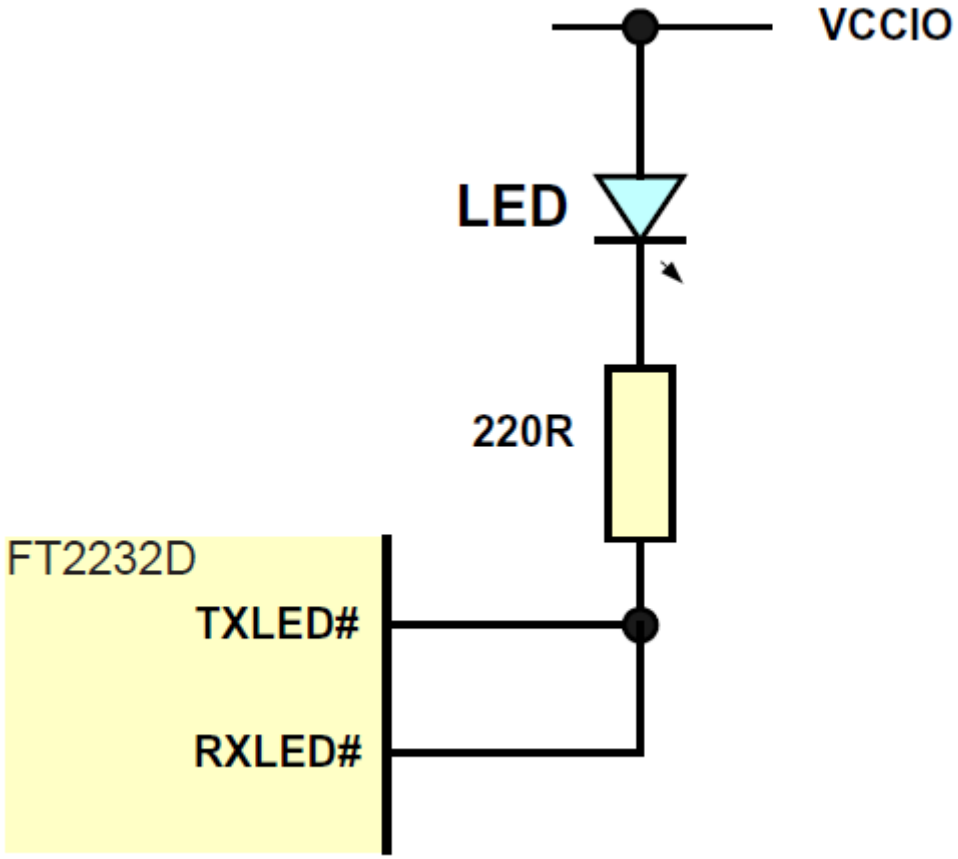

Figure 8.5 Single LED Configuration

When configured in UART mode the FT2232D has two IO pins on each channel dedicated to controlling LED status indicators, one for transmitted data the other for received data. When data is being transmitted / received the respective pins drive from tri-state to low in order to provide indication on the LED's of data transfer. A digital one-shot timer is used so that even a small percentage of data transfer is visible to the end user. Figure 8.4 shows a configuration using two individual LED's – one for transmitted data the other for received data. In Figure 8.5, the transmit and receive LED indicators are wire-OR'ed together to give a single LED indicator which indicates any transmit or receive data activity.
Another possibility (not shown here) is to use a 3 pin common anode tri-color LED based on the circuit in Figure 7.5 to have a single LED that can display activity in a variety of colors depending on the ratio of transmit activity compared to receive activity.
Note that the LED's are connected to the same supply as VCCIO.

8.2 245 FIFO Interface Mode Signal Descriptions and Interface Configurations

When either Channel A or Channel B is in 245 FIFO mode, the IO signal lines are configured as follows:-

Pin#		*Signal*	*Type*	*Description*
Channel A	*Channel B*			
24	40	D0	I/O	FIFO Data Bus Bit 0
23	39	D1	I/O	FIFO Data Bus Bit 1
22	38	D2	I/O	FIFO Data Bus Bit 2
21	37	D3	I/O	FIFO Data Bus Bit 3
20	36	D4	I/O	FIFO Data Bus Bit 4
19	35	D5	I/O	FIFO Data Bus Bit 5
17	33	D6	I/O	FIFO Data Bus Bit 6
16	32	D7	I/O	FIFO Data Bus Bit 7

Table 8.2 FIFO Data Bus Group **Note 17

Pin#		*Signal*	*Type*	*Description*
Channel A	*Channel B*			
15	30	RXF#	OUTPUT	When high, do not read data from the FIFO. When low, there is data available in the FIFO which can be read by strobing RD# low then high again **** Note 18**
13	29	TXE#	OUTPUT	When high, do not write data into the FIFO. When low, data can be written into the FIFO by strobing WR high then low. **** Note 18**
12	28	RD#	INPUT	Enables Current FIFO Data Byte on D0..D7 when low. Fetches the next FIFO Data Byte (if available) from the Receive FIFO Buffer when RD# goes from low to high. **** Note 17**
11	27	WR	INPUT	Writes the Data Byte on the D0..D7 into the Transmit FIFO Buffer when WR goes from high to low. **** Note 17**
10	26	SI/WU	INPUT	The Send Immediate / WakeUp signal combines two functions on a single pin. If USB is in suspend mode (PWREN# = 1) and remote wakeup is enabled in the EEPROM, strobing this pin low will cause the device to request a resume on the USB Bus. Normally, this can be used to wake up the Host PC. During normal operation (PWREN# = 0), if this pin is strobed low any data in the device TX buffer will be sent out over USB on the next Bulk-IN request from the drivers regardless of the pending packet size. This can be used to optimise USB transfer speed for some applications. Tie this pin to VCCIO if not used.

Table 8.3 FIFO Control Interface Group

****Note 17:** In Input Mode, these pins are pulled to VCCIO via internal 200K resistors. These can be programmed to gently pull low during USB suspend (PWREN# = "1") by setting this option in the EEPROM.

****Note 18:** During device reset, these pins are tri-state but pulled up to VCCIO via internal 200K resistors.

8.3 245 FIFO Mode Timing Diagram

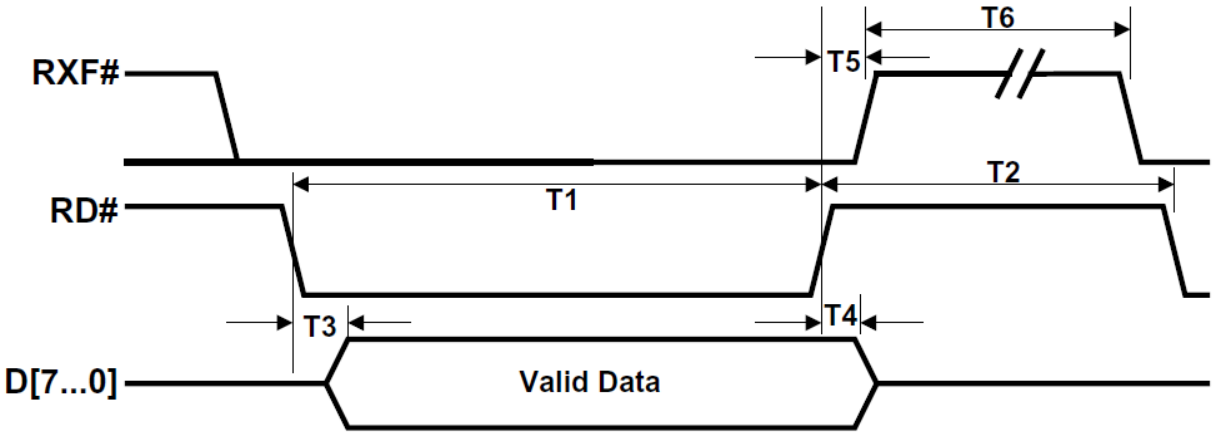

Figure 8.6 FIFO Read Cycle

Time	*Description*	*Min*	*Max*	*Unit*
T1	RD# Active Pulse Width	50	ns	RD# Active Pulse Width
T2	RD# to RD Pre-Charge Time	50 + T6	ns	RD# to RD Pre-Charge Time
T3	RD# Active to Valid Data ****Note 19**	20	50	ns
T4	Valid Data Hold Time from RD# Inactive ****Note 19**	0	ns	Valid Data Hold Time from RD# Inactive ****Note 19**
T5	RD# Inactive to RXF#	0	25	ns
T6	RXF# inactive after RD# cycle	80	ns	RXF# inactive after RD# cycle

Table 8.4 Read Cycle

**** Note 19:** Load 30 pF at standard drive level. These times will also vary if the high output drive level is enabled

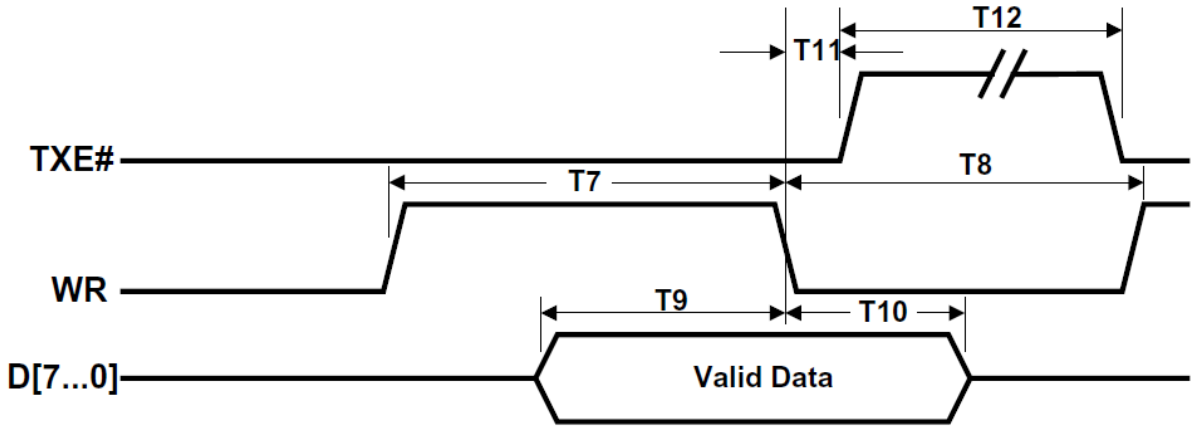

Figure 8.7 FIFO Write Cycle

Time	Description	Min	Max	Unit
T7	WR Active Pulse Width	50	ns	WR Active Pulse Width
T8	WR to WR Pre-Charge Time	50	ns	WR to WR Pre-Charge Time
T9	Data Setup Time before WR inactive	20	ns	Data Setup Time before WR inactive
T10	Data Hold Time from WR inactive	0	ns	Data Hold Time from WR inactive
T11	WR Inactive to TXE#	5	25	ns
T12	TXE inactive after WR cycle	80	ns	TXE inactive after WR cycle

Table 8.5 Write Cycle

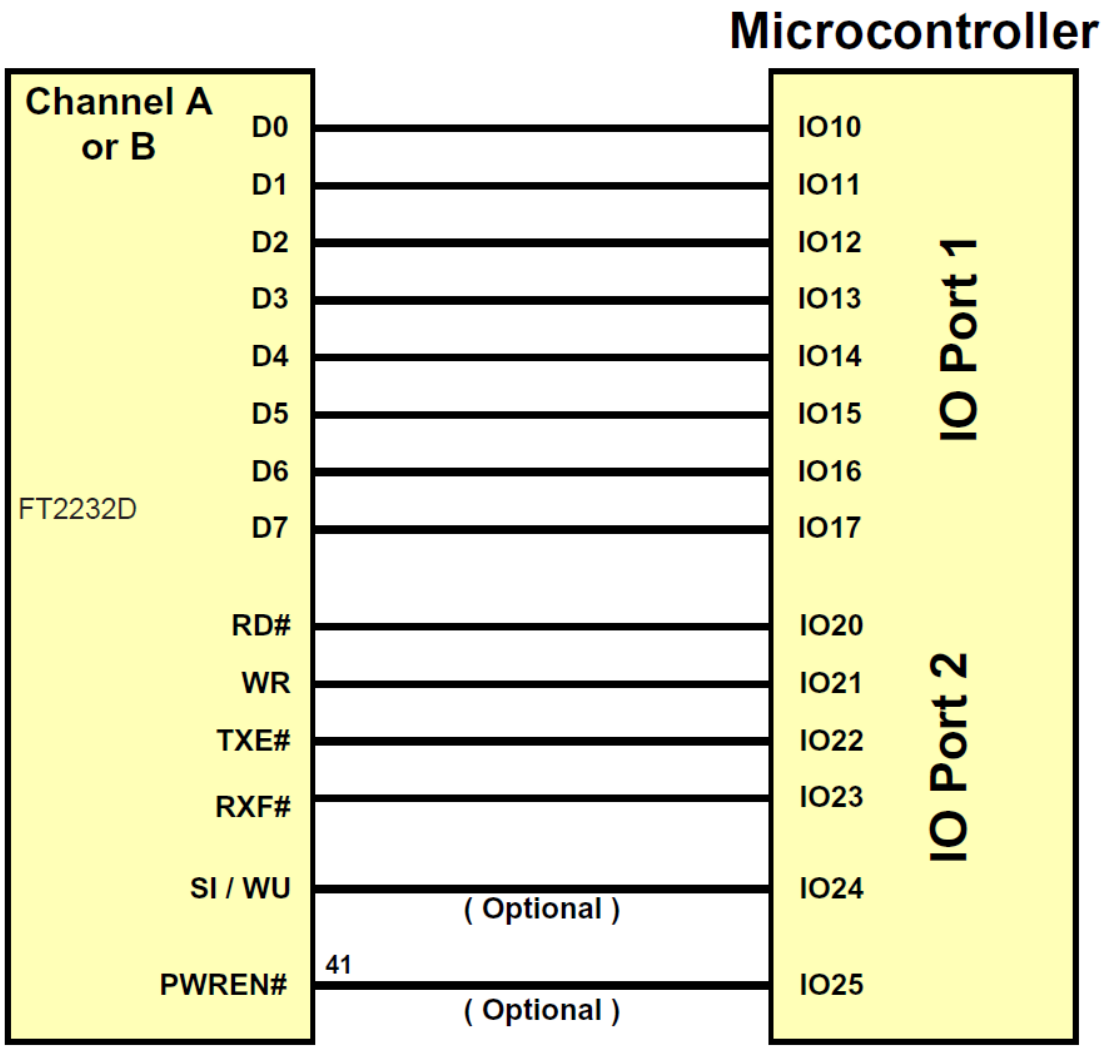

Figure 8.8 Microprocessor Interface Example

Figure 8.8 illustrates a typical interface between one of the channels of the FT2232D, configured in 245-style FIFO interface mode, and a Microcontroller (MCU). Either channel A or B, or both can be configured in this mode.
This examples uses two IO Ports of the MCU, one port (8 bits) to transfer data to one of the and the other port (4 / 5 bits) to monitor the TXE# and RXF# status bits and generate the RD# and WR strobes to the FT2232D as required. Optionally, SI / WU can be connected to another IO pin if either of the functions of this pin are required. If the SI / WU function is not required, tie this pin to VCCIO. If the MCU is handling power management functions, then PWREN# should also be connected to an IO pin of the MCU.
The 8 data bits on IO Port 1 can be shared with other peripherals when the MCU is not accessing the FT2232D

8.4 Enhanced Asynchronous and Synchronous Bit-Bang Modes - Signal Description and Interface Configuration

Bit-bang mode is a special FT2232D device mode that changes the 8 IO lines on either (or both) channels into an 8 bit bi-directional data bus. The are two types of Bit-bang mode for the FT2232D - Enhanced Asynchronous Bit-Bang Mode, which is virtually the same as FTDI BM chip-style Bit-Bang mode, with the addition of Read and Write strobes; and Synchronous Bit-Bang mode, where data is only read from the device when the device is written to. Bit-Bang mode is enabled by driver command. When either Channel A or Channel B (or both) have Enhanced Asynchronous Bit-Bang mode, or Synchronous Bit-Bang mode enabled the IO signal lines are configured as follows :-

Pin#		***Signal***	***Type***	***Description***
Channel A	***Channel B***			
24	40	D0	I/O	Bit-Bang Data Bus Bit 0
23	39	D1	I/O	Bit-Bang Data Bus Bit 1
22	38	D2	I/O	Bit-Bang Data Bus Bit 2
21	37	D3	I/O	Bit-Bang Data Bus Bit 3
20	36	D4	I/O	Bit-Bang Data Bus Bit 4
19	35	D5	I/O	Bit-Bang Data Bus Bit 5
17	33	D6	I/O	Bit-Bang Data Bus Bit 6
16	32	D7	I/O	Bit-Bang Data Bus Bit 7

Table 8.6 Bit-Bang Data Bus Group **Note 24

****Note 24:** In Input Mode, these pins are pulled to VCCIO via internal 200K resistors. These can be programmed to gently pull low during USB suspend (PWREN# = "1") by setting this option in the EEPROM.

Pin#		***Signal***	***Type***	***Description***
Channel A	***Channel B***			
15	30	WR#	OUTPUT	****Note 25**
13	29	RD#	OUTPUT	****Note 25**
12	28	WR#	INPUT	****Note 26**
11	27	RD#	INPUT	****Note 26**
10	26	SI/WU	INPUT	The Send Immediate / WakeUp signal combines two functions on a single pin. If USB is in suspend mode (PWREN# = 1) and remote wakeup is enabled in the EEPROM , strobing this pin low will cause the device to request a resume on the USB Bus. Normally, this can be used to wake up the Host PC. During normal operation (PWREN# = 0), if this pin is strobed low any data in the device TX buffer will be sent out over USB on the next Bulk-IN request from the drivers regardless of the pending packet size. This can be used to optimise USB transfer speed for some applications. Tie this pin to VCCIO if not used

Table 8.7 Bit-Bang Control Interface Group

****Note 25:** The Bit-Bang Mode (synchronous and asynchronous) WR# and RD# strobes are on these pins when the main Channel mode is 245 FIFO, or Fast Opto-Isolated Serial Mode. Bit-Bang mode is not available on Channel B when Fast Opto-Isolated Serial Mode is enabled.

****Note 26 :** The Bit-Bang Mode (synchronous and asynchronous) WR# and RD# strobes are on these pins when the main Channel mode is set to 232 UART Mode.

Enhanced Asynchronous Bit-Bang Mode

Enhanced Asynchronous Bit-Bang mode is the same as BM-style Bit-Bang mode, except that the internal RD# and WR# strobes are now brought out of the device to allow external logic to be clocked by accesses to the bit-bang IO bus.
On either or both channels any data written to the device in the normal manner will be self clocked onto the data pins (those which have been configured as outputs). Each pin can be independently set as an input or an output. The rate that the data is clocked out at is controlled by the baud rate generator.
For the data to change there has to be new data written, and the baud rate clock has to tick. If no new data is written to the channel, the pins will hold the last value written.
To allow time for the data to be setup and held around the WR# strobe, the baud rate should be less than 1 MegaBaud.
See the application note **AN232B-01, "FT232BM/FT245BM Bit Bang Mode"** for more details and a sample application.

Enabling

Asynchronous Bit-Bang mode is enabled using Set Bit Bang Mode driver command. A hex value of 1 will enable it, and a hex value of 0 will reset the device. See application note **AN2232-02, "Bit Mode Functions for the FT2232D"** for more details and examples of this.

Synchronous Bit-Bang Mode

With Synchronous Bit-Bang mode data will only be sent out by the FT2232D if there is space in the device for data to be read from the pins. This Synchronous Bit-Bang mode will read the data bus pins first, before it sends out the byte that has just been transmitted. It is therefore 1 byte behind the output, and so to read the inputs for the byte that you have just sent, another byte must be sent.
For example:-
(1) Pins start at 0xFF
Send 0x55, 0xAA
Pins go to 0x55 and then to 0xAA
Data read = 0xFF, 0x55
(2) Pins start at 0xFF
Send 0x55, 0xAA, 0xAA
(Repeat the last byte sent)
Pins go to 0x55 and then to 0xAA

Data read = 0xFF, 0x55, 0xAA

Enabling

Synchronous Bit-Bang mode is enabled using Set Bit Bang Mode driver command. A hex value of 4 will enable it, and a hex value of 0 will reset the device. See application note **AN2232-02, "Bit Mode Functions for the FT2232D"** for more details and examples.

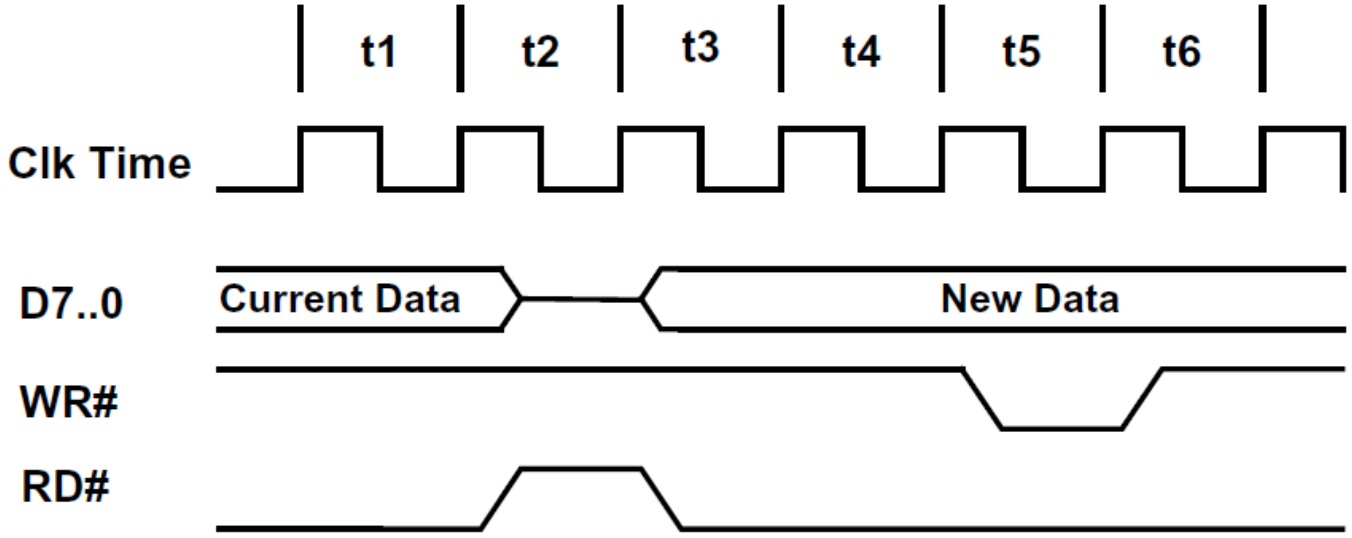

Figure 8.9 Synchronous Bit Bang Mode Signal Timing

Time	*Description*
t1	Current pin state is read
t2	RD# is set inactive
t3	RD# is set active again, and any pins that are output will change to the new data.
t4	Clock state for data setup
t5	WR# goes active
t6	WR# goes inactive

Table 8.8 Synchronous Bit Bang Mode Signal Timing

The internal RD# and WR# strobes are brought out of the device to allow external logic to be clocked by accesses to the bit-bang IO bus.

8.5 Multi-Protocol Synchronous Serial Engine (MPSSE) Mode Signal Descriptions and Interface Configurations

MPSSE Mode is designed to allow the FT2232D to interface efficiently with synchronous serial protocols such as JTAG and SPI Bus. It can also be used to program SRAM based FPGA's over USB. The MPSSE interface is designed to be flexible so that it can be configured to allow any synchronous serial protocol (industry standard or proprietary) to be interfaced to the FT2232D. MPSSE is available on channel A only. MPSSE is fully configurable, and is programmed by sending commands down the data pipe. These can be sent individually or more efficiently in packets. MPSSE is capable of a maximum sustained data rate of 5.6 Mega bits /s.

When Channel A is configured in MPSSE mode the IO signal lines are configured as follows:-

Pin# ***Channel A only***	***Signal***	***Type***	***Description***
24	TCK/SK	OUTPUT	Clock signal Output
23	TDI/D0	OUTPUT	Serial Data Out
22	TDO/DI	INPUT	Serial Data In ****Note 27**
21	TMS/CS	OUTPUT	Select Signal Out
20	GPIOL0	I/O	General Purpose input / Output ****Note 27**
19	GPIOL1	I/O	General Purpose input / Output ****Note 27**
17	GPIOL2	I/O	General Purpose input / Output ****Note 27**
16	GPIOL3	I/O	General Purpose input / Output ****Note 27**
15	GPIOH0	I/O	General Purpose input / Output ****Note 27**
13	GPIOH1	I/O	General Purpose input / Output ****Note 27**
12	GPIOH2	I/O	General Purpose input / Output ****Note 27**
11	GPIOH3	I/O	General Purpose input / Output ****Note 27**

Table 8.9 MPSSE Mode Configuration

****Note 27:** In Input Mode, these pins are pulled to VCCIO via internal 200K resistors. These can be programmed to gently pull low during USB suspend (PWREN# = "1") by setting this option in the EEPROM.

Enabling

MPSSE mode is enabled using Set Bit Bang Mode driver command. A hex value of 2 will enable it, and a hex value of 0 will reset the device. See application note **AN2232-02, "Bit Mode Functions for the FT2232D"** for more details and examples.

The MPSSE command set is fully described in application note **AN_108, "Command Processor for MPSSE and MCU Host Bus Emulation Modes".**

Example project code for the FT2232D that demonstrates how to use the devices Multi-Protocol Synchronous Serial Engine (MPSSE) to make a USB to SPI bus interface are available from FTDI's web site **http://www.ftdichip.com/Projects/MPSSE.htm#SPI**

Example project code for the FT2232D that demonstrates how to use the devices Multi-Protocol Synchronous Serial Engine (MPSSE) to make a USB to I2C bus interface are available from FTDI's web site **http://www.ftdichip.com/Projects/MPSSE.htm#I2C**

Example project code for the FT2232D that demonstrates how to use the devices Multi-Protocol Synchronous Serial Engine (MPSSE) to make a USB to JTAG bus interface are available from FTDI's web site **http://www.ftdichip.com/Projects/MPSSE.htm#JTAG**

8.6 MCU Host Bus Emulation Mode Signal Descriptions and Interface Configuration

MCU host bus emulation mode uses both of the FT2232D's A and B channel interfaces to make the chip emulate a standard 8048 / 8051 MCU host bus. This allows peripheral devices for these MCU families to be directly connected to USB via the FT2232D.
The lower 8 bits (AD7 to AD0) is a multiplexed Address / Data bus. A8 to A15 provide upper (extended) addresses.
There are 4 basic operations:-
1) Read (does not change A15 to A8)
2) Read Extended (changes A15 to A8)
3) Write (does not change A15 to A8)
4) Write Extended (changes A15 to A8)

Enabling

MCU Host Bus Emulation mode is enabled using Set Bit Bang Mode driver command. A hex value of 8 will enable it, and a hex value of 0 will reset the device. See application note **AN2232-02, "Bit Mode Functions for the FT2232D"** for more details and examples.

The MCU Host Bus Emulation Mode command set is fully described in application note **AN_108, "Command Processor For MPSSE and MCU Host Bus Emulation Modes"**.

When MCU Host Bus Emulation mode is enabled the IO signal lines on both channels work together and the pins are configured as follows :-

Pin#	*Signal*	*Type*	*Description*
24	AD0	I/O	Address / Data Bus Bit 0 ****Note 28**
23	AD1	I/O	Address / Data Bus Bit 1 ****Note 28**
22	AD2	I/O	Address / Data Bus Bit 2 ****Note 28**
21	AD3	I/O	Address / Data Bus Bit 3 ****Note 28**
20	AD4	I/O	Address / Data Bus Bit 4 ****Note 28**
19	AD5	I/O	Address / Data Bus Bit 5 ****Note 28**
17	AD6	I/O	Address / Data Bus Bit 6 ****Note 28**
16	AD7	I/O	Address / Data Bus Bit 7 ****Note 28**
15	I/O0	I/O	MPSSE mode instructions to set / clear or read the high byte of data can be used with this pin. ****Note 28, **Note 29**
13	I/O1	I/O	MPSSE mode instructions to set / clear or read the high byte of data can be used with this pin. In addition this pin has instructions which will make the controller wait until it is high, or wait until it is low. This can be used to connect to an IRQ pin of a peripheral chip. The FT2232D will wait for the interrupt, and then read the device, and pass the answer back to the host PC. I/O1 must be held in input mode if this option is used. ****Note 28, **Note 29**
12	IORDY	INPUT	Extends the time taken to perform a Read or Write operation if pulled low. Pull up to Vcc if not being used.
11	OSC	OUTPUT	Shows the clock signal that the circuit is using.
40	A8	OUTPUT	Extended Address Bus Bit 8
39	A9	OUTPUT	Extended Address Bus Bit 9
38	A10	OUTPUT	Extended Address Bus Bit 10
37	A11	OUTPUT	Extended Address Bus Bit 12
36	A12	OUTPUT	Extended Address Bus Bit 13
35	A13	OUTPUT	Extended Address Bus Bit 14
33	A14	OUTPUT	Extended Address Bus Bit 15
32	A15	OUTPUT	Extended Address Bus Bit 16
30	CS#	OUTPUT	Negative pulse to select device during Read or Write.
29	ALE	OUTPUT	Positive pulse to latch the address.
28	RD#	OUTPUT	Negative Read Output.
27	WR#	OUTPUT	Negative Write Output. (Data is setup before WR# goes low, and is held after WR# goes high)

Table 8.10 MCU Host Bus Emulation Mode IO Signal Lines Configuration

****Note 28:** In Input Mode, these pins are pulled to VCCIO via internal 200K resistors. These can be programmed to gently pull low during USB suspend (PWREN# = "1") by setting this option in the EEPROM.

****Note 29:** These instructions are fully described in the application note AN2232L-01 - **"Command Processor For MPSSE and MCU Host Bus Emulation Modes"**.

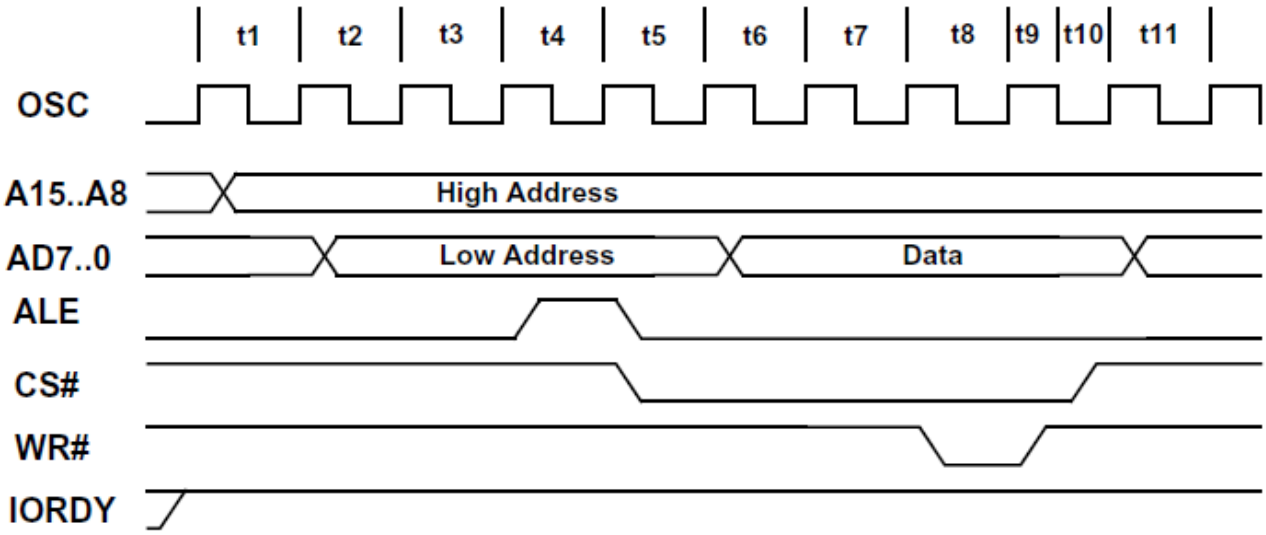

Figure 8.10 MCU Host Bus Emulation Mode Signal Timing - Write Cycle

Time	*Description*
t1	High address byte is placed on the bus if the extended write is used.
t2	Low address byte is put out.
t3	1 clock period for address is set up.
t4	ALE goes high to enable latch. This will extend to 2 clocks wide if IORDY is low.
t5	ALE goes low to latch address and CS# is set active low.
t6	Data driven onto the bus.
t7	1 clock period for data setup.
t8	WR# is driven active low. This will extend to 6 clocks wide if IORDY is low.
t9	WR# is driven inactive high.
t10	CS# is driven inactive, 1/2 a clock period after WR# goes inactive
t11	Data is held until this point, and may now change

Table 8.11 MCU Host Bus Emulation Mode Signal Timing - Write Cycle

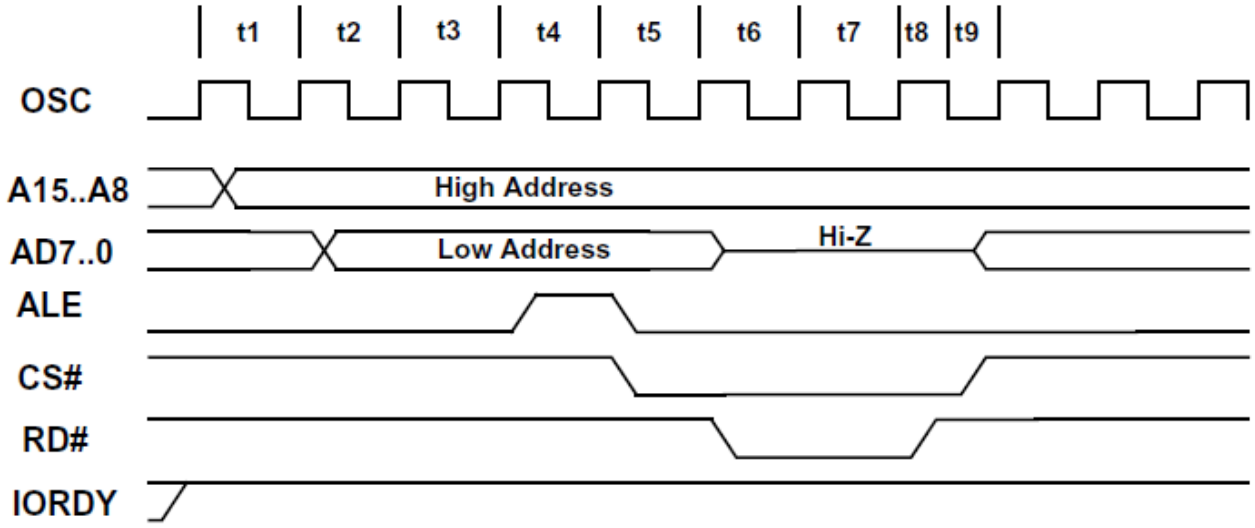

Figure 8.11 MCU Host Bus Emulation Mode Signal Timing - Read Cycle

Time	*Description*
t1	High address byte is placed on the bus if the extended read is used - otherwise t1 will not occur.
t2	Low address byte is put out.
t3	1 clock period for address set up.
t4	ALE goes high to enable address latch. This will extend to 2 clocks wide if IORDY is low.
t5	ALE goes low to latch address, and CS# is set active low. This will extend to 3 clocks if IORDY is sampled low. CS# will always drop 1 clock after ALE has gone high no matter the state of IORDY.
t6	Data is set as input (Hi-Z), and RD# is driven active low.
t7	1 clock period for data setup. This will extend to 5 clocks wide if IORDY# is sampled low.
t8	RD# is driven inactive high.
t9	CS# is driven inactive 1/2 a clock period after RD# goes inactive, and the data bus is set back to output.

Table 8.12 MCU Host Bus Emulation Mode Signal Timing - Read Cycle

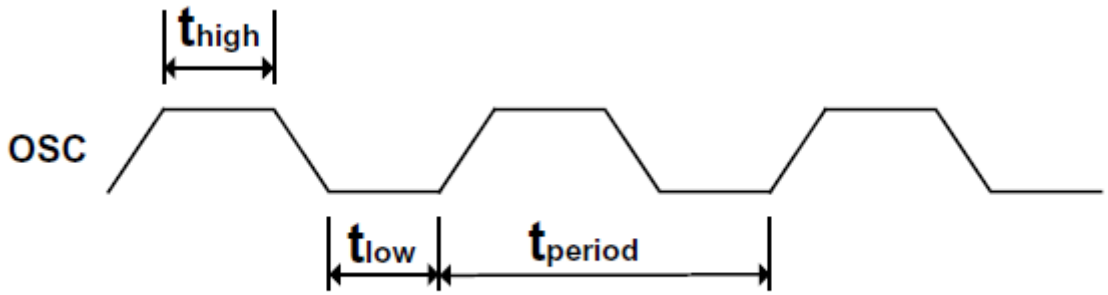

Figure 8.12 MCU Host Bus Emulation Mode Signal Timing - Clock (OSC) Signal

Time	Description	Min	Typical Value	Max	Unit
tperiod	Clock Period	41.6	83.3	125.0	ns
thigh	Clock signal high time	20.8	41.6	62.5	ns
tlow	Clock signal low time	20.8	41.6	62.5	ns

Table 8.13 MCU Host Bus Emulation Mode Signal Timing - Clock (OSC) Signal

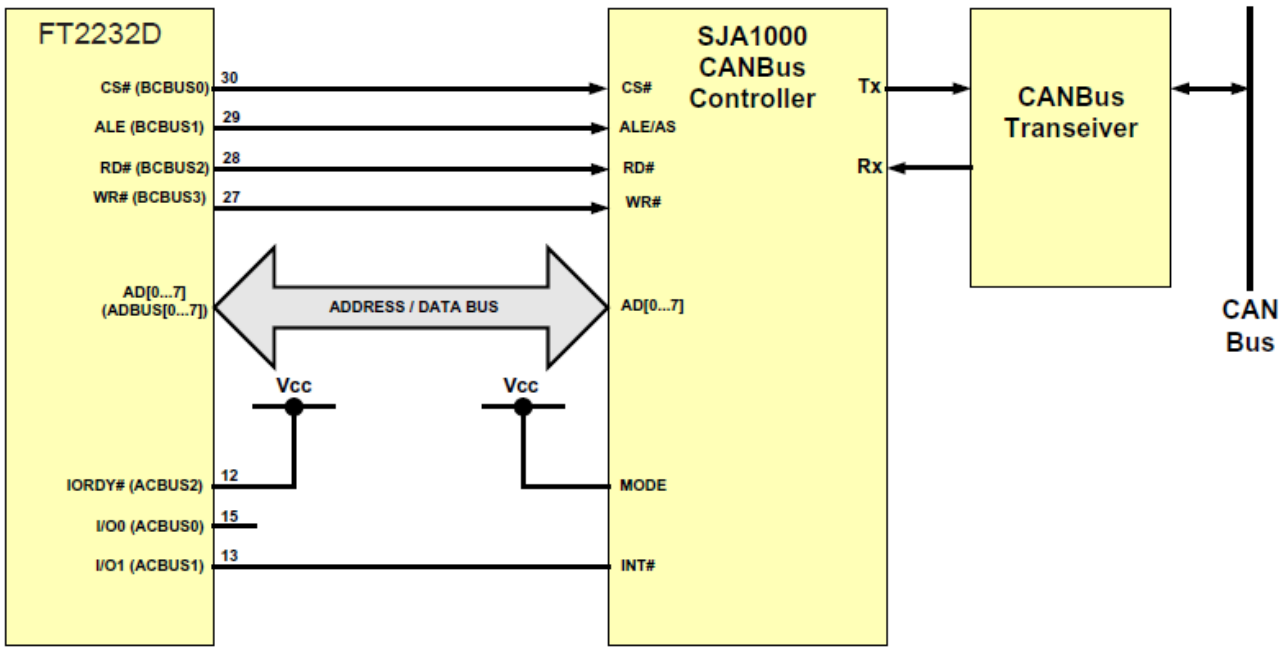

Figure 8.13 MCU Host Bus Emulation Example - USB <=> CAN Bus Interface

Figure 8.13 shows an example where the FT2232D is used to interface a Philips SJA1000 CAN Bus Controller to a PC over USB. In this example IORDY is not used and so is pulled up to Vcc. I/O1 is used to monitor the Interrupt output (INT) of the SJA1000. The MODE pin on the SJA1000 is pulled high to select Intel mode. See the semiconductors section of the Philips website (**www.philips.com**) for more details on the SJA1000 and suitable CAN Bus transceiver devices.

8.7 Fast Opto-Isolated Serial Interface Mode Signal Descriptions and Interface Configuration

Fast Opto-Isolated Serial Interface Mode provides a method of communicating with an external device over USB using 4 wires that can have opto-isolators in their path, thus providing galvanic isolation between systems. If either channel A or channel B are enabled in fast opto-isolated serial mode then the pins on channel B are switched to the fast serial interface configuration. The I/O interface for fast serial mode is always on channel B, even if both channels are being used in this mode. An address bit is used to determine the source or destination channel of the data. It therefore makes sense to always use at least channel B or both for fast serial mode, but not A own its own.

When either Channel B or Both Channel A and B are configured in Fast Opto-Isolated Serial Interface mode following IO signal lines are configured as follows :-

Pin#	*Signal*	*Type*	*Description*
40	FSDI	INPUT	Fast serial data input ****Note 30**
39	FSCLK	INPUT	Clock input to FT2232D chip to clock data in or out. The external device has to provide a clock signal or nothing will change on the interface pins. This gives the external device full control over the interface. It is designed to be half duplex so that data is only transferred in one direction at a time. ****Note 30**
38	FSDO	OUTPUT	Fast serial data output. Driven low to indicate that the chip is ready to send data.
37	FSCTS	OUTPUT	Clear To Send control signal output
26	SI/WU	INPUT	The Send Immediate / WakeUp signal combines two functions on a single pin. If USB is in suspend mode (PWREN# = 1) and remote wakeup is enabled in the EEPROM , strobing this pin low will cause the device to request a resume on the USB Bus. Normally, this can be used to wake up the Host PC. During normal operation (PWREN# = 0), if this pin is strobed low any data in the device TX buffer will be sent out over USB on the next Bulk-IN request from the drivers regardless of the pending packet size. This can be used to optimise USB transfer speed for some applications. Tie this pin to VCCIO if not used.

Table 8.14 Fast Opto-Isolated Serial Interface Mode IO Signal Lines Configuration

****Note 30:** Pulled up to VCCIO via internal 200K resistors. These pins can be programmed to gently pull low during USB suspend (PWREN# = "1") by setting this option in the EEPROM.

Fast Opto-Isolated serial interface mode is enabled in the external EEPROM.

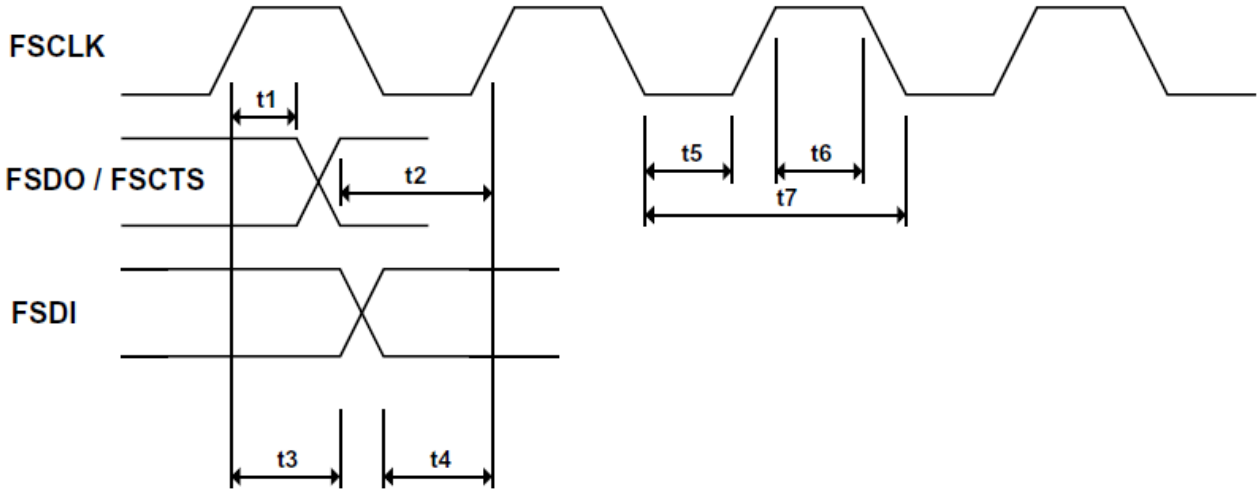

Figure 8.14 Fast Opto-Isolated Serial Signal Timing Diagram

Time	*Description*	*Min*	*Max*	*Unit*
t1	FSDO / FSCTS hold time	5	-	ns
t2	FSDO / FSCTS setup time	5	-	ns
t3	FSDI hold time	5	-	ns
t4	FSDI setup time	10	-	ns
t5	FSCLK low	10	-	ns
t6	FSCLK high	10	-	ns

Table 8.15 Fast Opto-Isolated Serial Signal Timing Diagram

Outgoing Fast Serial Data

To send fast serial data out of the chip, the external device must clock. If the chip has data ready to send, it will drive FSDO low to indicate the start bit. It will not do this if it is currently receiving data from the external device.

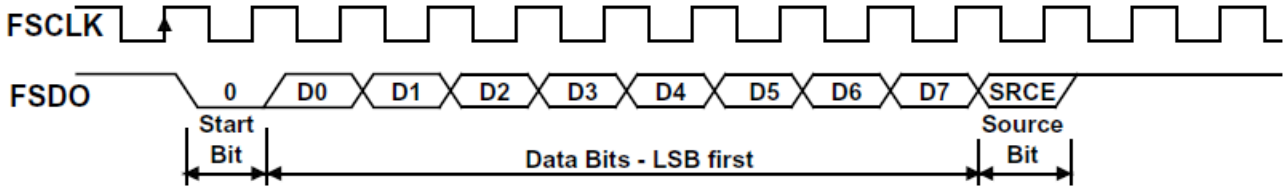

Figure 8.15 Fast Opto-Isolated Serial Data Format - Data output from the FT2232D

Notes:-

(i) Start Bit is always 0.
(ii) Data is sent LSB first.
(iii) The source bit (SRCE) indicates which channel the data has come from. A '0' means that it has come from Channel A, a '1' means that it has come from Channel B.

(iv) If the target device is unable to accept the data when it detects the start bit, it should stop the FSCLK until it can accept the data.

Incoming Fast Serial Data

The external device is allowed to send data into the chip if FSCTS is high. On receipt of a Zero start bit on FSDI, the chip will drop FSCTS on the next positive clock edge. The data from bits 0 to 7 is then clocked in (LSB first). The next bit determines where the data will be written to. It can go to either channel A or to channel B. A '0' will send it to channel A, providing channel A is enabled for fast serial mode, otherwise it will go to channel B. A '1' will send it to channel B, providing channel B is enabled for fast serial mode, otherwise it will go to channel A. (Either channel A, or channel B, or both must be enabled as fast serial mode or the circuit is disabled).

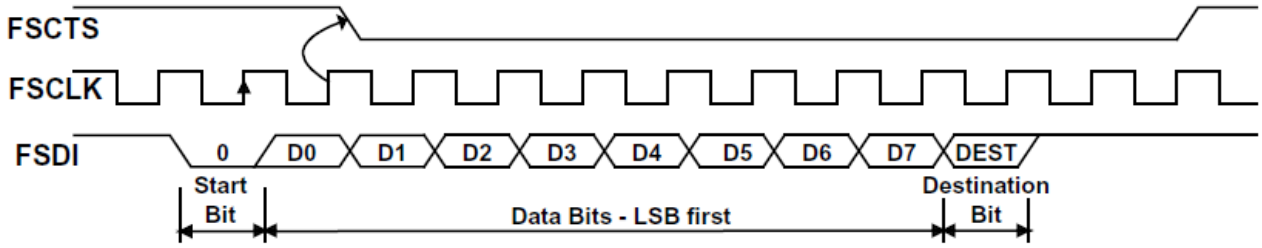

Figure 8.16 Fast Opto-Isolated Serial Data Format - Data input to the FT2232D

Notes:-

(i) Start Bit is always 0.
(ii) Data is sent LSB first.
(iii) The destination bit (DEST) indicates which channel the data should go to. A '0' means that it should go to channel A, a '1' means that it should go to channel B.
(iv) The target device should check CTS is high before it sends data. CTS goes low after data bit 0 (D0) and stays low until the chip can accept more data.

Contention

There is a possibility that contention may occur, where the interface goes from being completely idle to both sending and receiving at the same clock instance. In this case the chip backs off, and allows the data from the external device to be received.

Data Format

The data format for either direction is:-

1) Zero Start Bit
2) Data bit 0
3) Data bit 1
4) Data bit 2
5) Data bit 3
6) Data bit 4
7) Data bit 5
8) Data bit 6
9) Data bit 7

10) Source/Destination ('0' indicates channel A; '1' indicates channel B)

Reset / Enable

Fast serial mode is enabled by setting the appropriate bits in the external EEPROM. The fast serial mode can be held in reset by setting a bit value of 10 using the Set Bit Bang Mode command. While this bit is set the device is held reset - data can be sent to the device, but it will not be sent out by the device until the device is enabled again. This is done by sending a bit value of 0 using the set bit mode command. See application note **AN2232L-02, "Bit Mode Functions for the FT2232D"** for more details and examples.

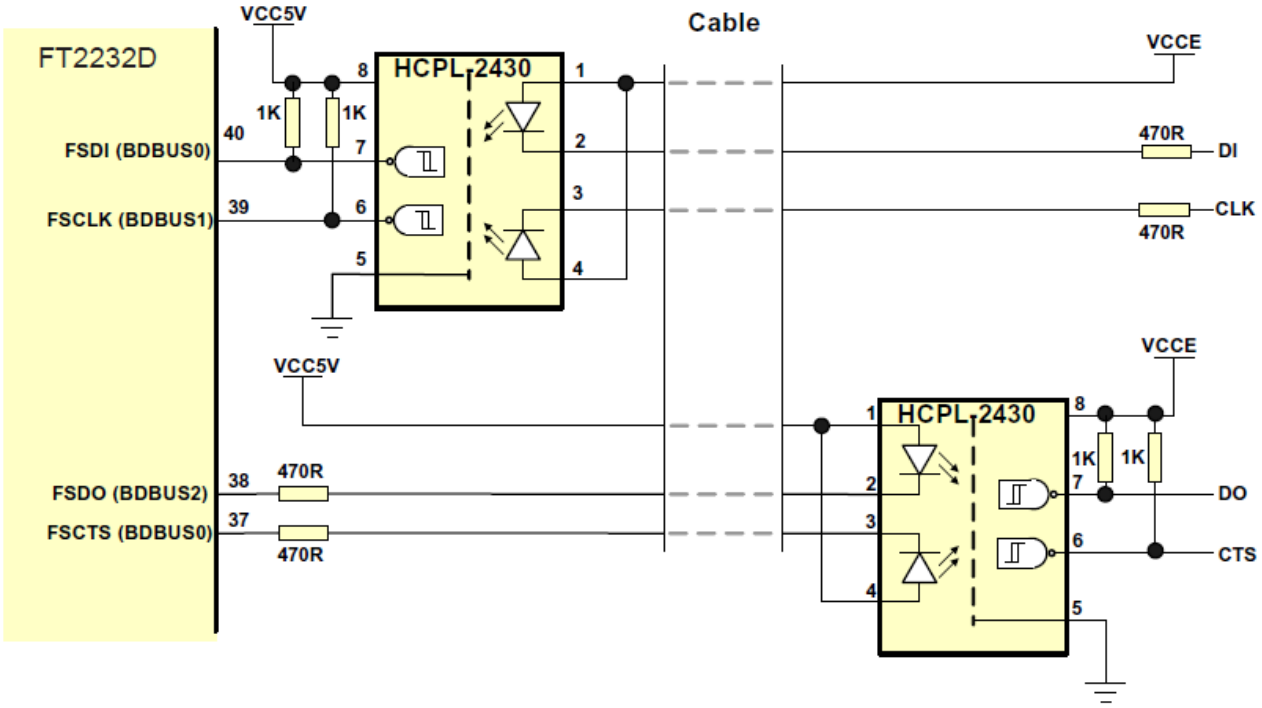

Figure 8.17 Fast Opto-Isolated Serial Interface Example

In the example shown in figure 8.18 two Agilent HCPL-2430 (see the semiconductor section at **www.agilent.com**) high speed opto-couplers are used to optically isolate an external device which interfaced to USB using the FT2232D. In this example VCC5V is the supply for the FT2232D (bus or self powered), and VCCE is the supply to the external device.
Care must be taken with the voltage used to power the photoLED's. It should be the same supply that the I/Os are driving to, or the LED's may be permanently on. Limiting resistors should be fitted in the lines that drive the diodes. The outputs of the opto-couplers are open-collector and so need a pullup resistor.

Testing

Fast serial mode has been tested using an Scenix (Ubicom), SX28 microcontroller (see **www.ubicom.com**) which was configured in loopback mode. This was done both with, and without HP HCPL-2430 opto-isolators in place. The isolators add a considerable delay to the turnaround time seen by the micro. This was close to 100 nS with the high speed HCPL-2430 device. This is the combined delay of the clock signal from the microcontroller going through an opto-coupler to the chip, and the data from the FT2232D chip going through the other opto-coupler back to the microcontroller.

8.8 CPU FIFO Interface Mode Signal Descriptions and Configuration Examples

CPU-style FIFO interface mode is designed to allow a CPU to interface to USB via the FT2232D. This mode is enabled in the external EEPROM. The interface is achieved using a chip select bit (CS#) and address bit (A0).

When either Channel A or Channel B are in CPU FIFO Interface mode the IO signal lines are configured as follows:-

Pin#		*Signal*	*Type*	*Description*
Channel A	*Channel B*			
24	40	D0	I/O	FIFO Data Bus Bit 0
23	39	D1	I/O	FIFO Data Bus Bit 1
22	38	D2	I/O	FIFO Data Bus Bit 2
21	37	D3	I/O	FIFO Data Bus Bit 3
20	36	D4	I/O	FIFO Data Bus Bit 4
19	35	D5	I/O	FIFO Data Bus Bit 5
17	33	D6	I/O	FIFO Data Bus Bit 6
16	32	D7	I/O	FIFO Data Bus Bit 7

Table 8.16 FIFO Data Bus Group **Note 20

Pin#		*Signal*	*Type*	*Description*
Channel A	*Channel B*			
15	30	CS#	INPUT	Chip Select Bit ** **Note 20**
13	29	A0	INPUT	Address Bit ** **Note 20**
12	28	RD#	INPUT	Negative read input ** **Note 20**
11	27	WR#	INPUT	Negative write input ** **Note 20**

Table 8.17 FIFO Control Interface Group

****Note 20:** In Input Mode, these pins are pulled to VCCIO via internal 200K resistors. These can be programmed to gently pull low during USB suspend (PWREN# = "1") by setting this option in the EEPROM

CS#	*A0*	*RD#*	*WR#*
1	X	X	X
0	0	Read Data Pipe	Write Data Pipe
0	1	Read Status	Send Immediate ****Note 21**

Table 8.18 Chip Select bit and Address bit truth table

Key: X = Not Used; 1 = Signal off; 0 = Signal off

****Note 21:** Has to be clocked by USB clock

Data Bit	*Data*	*Status*
bit 0	1	Data Available (=RXF)
bit 1	1	Space Available (=TXE)
bit 2	1	Suspend
bit 3	1	Configured
bit 4 ****Note 22**	X	X
bit 5 ****Note 22**	X	X
bit 6 ****Note 22**	X	X
bit 7 ****Note 22**	X	X

Table 8.19 Status Data bits

Key: X = Not Used; 1 = Signal off; 0 = Signal off

****Note 22:** *bits 4 to 7 will have arbitrary values when the status is read*.

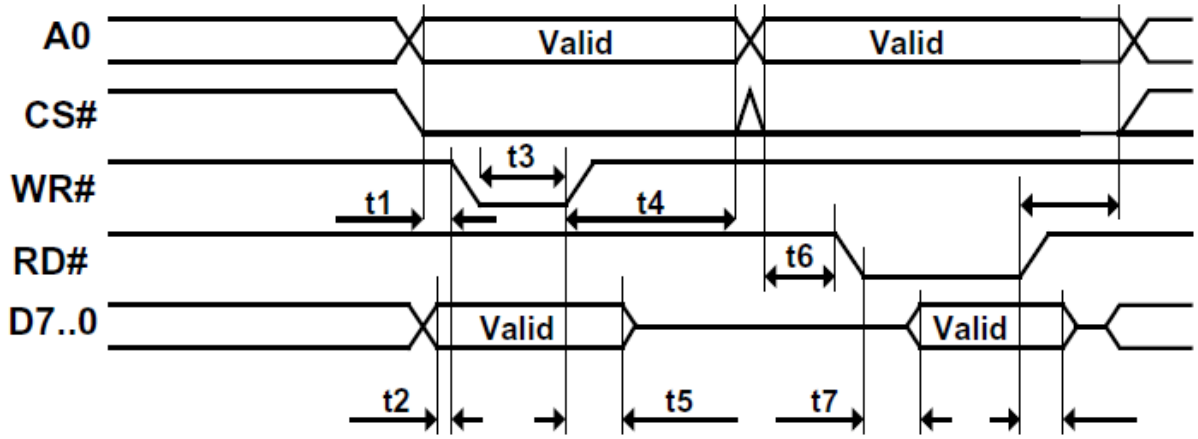

Figure 8.18 CPU FIFO Interface Mode - Signal Timing

Time	*Description*	*Min*	*Max*	*Unit*
t1	A0 / CS Setup to WR#	15	-	ns
t2	Data setup to WR#	15	-	ns
t3	WR# Pulse width	20	-	ns
t4	A0/CS Hold from WR#	5	-	ns
t5	Data hold from WR#	5	-	ns
t6	A0/CS Setup to RD#	15	-	ns
t7	Data delay from RD# ****Note 23**	15	50	ns
t8	A0/CS hold from RD#	5	-	ns
t9	Data hold time from RD# ****Note 23**	0	30	ns

Table 8.20 CPU FIFO Interface Mode - Signal Timing

****Note 23:** For standard output drive level Times may vary if high drive level is enabled

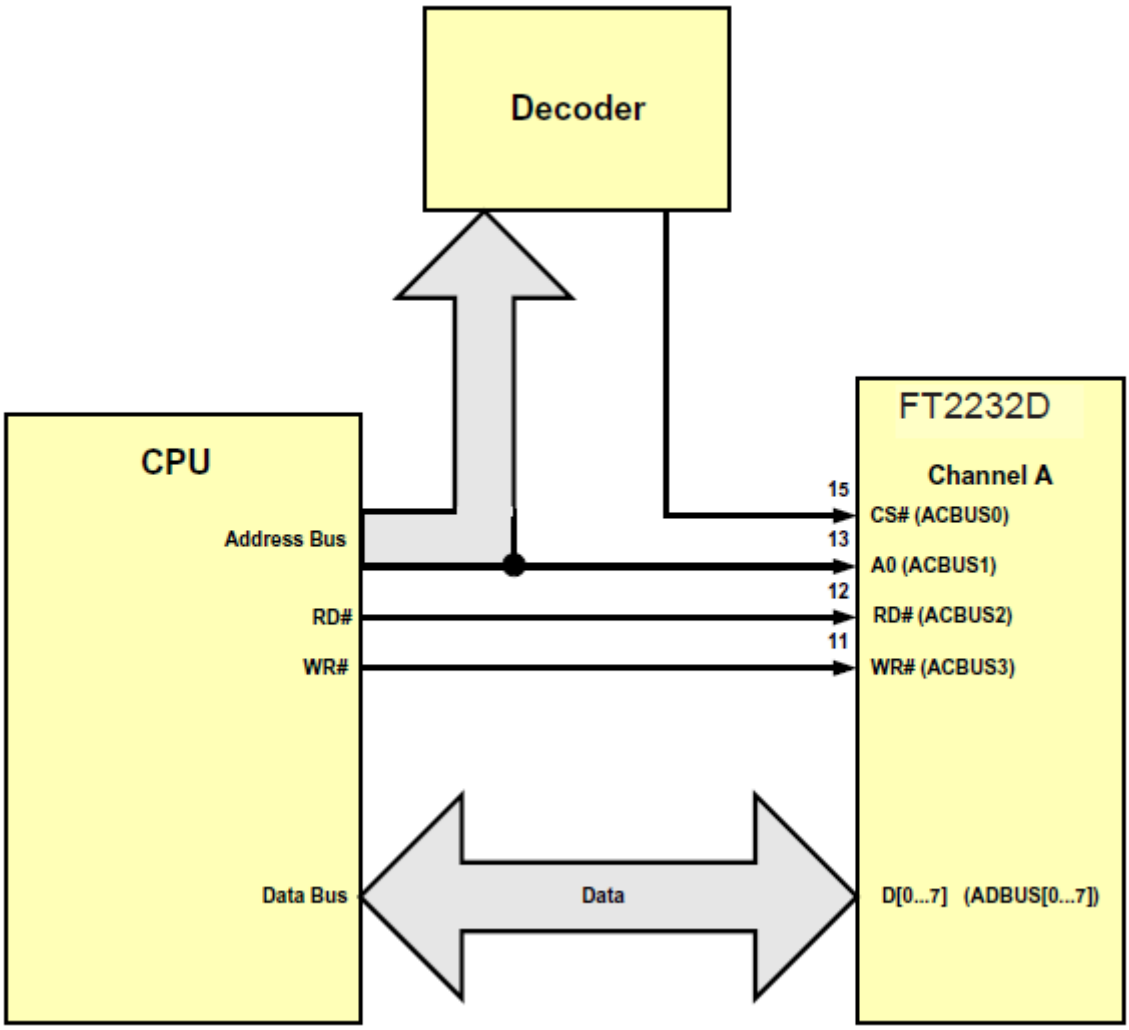

Figure 8.19 CPU FIFO Single Channel Interface Example 1

Figure 8.20 shows an example where channel A of the FT2232D is used in CPU FIFO mode to interface with a CPU. To read or write data to or from the CPU to the FT2232D, the FT2232D's Chip Select (CS#) would be set to 0. In order to read the status of the device the Address bit would then be set to 1, and RD# would be strobed causing the status data to be driven onto D0...D7. If data is available (D0 = 1) then it can be read by setting A0 to 0, and strobing RD#. If space is available (D1=1) then data can be written to the FT2232D by setting A0 to 0 and strobing WR#.
When CS# is set to 0 and A0 is set to 1, strobing WR# causes any data in the FT2232D's TX buffer to be sent out over USB on the next Bulk-In request, regardless of the pending packet size.

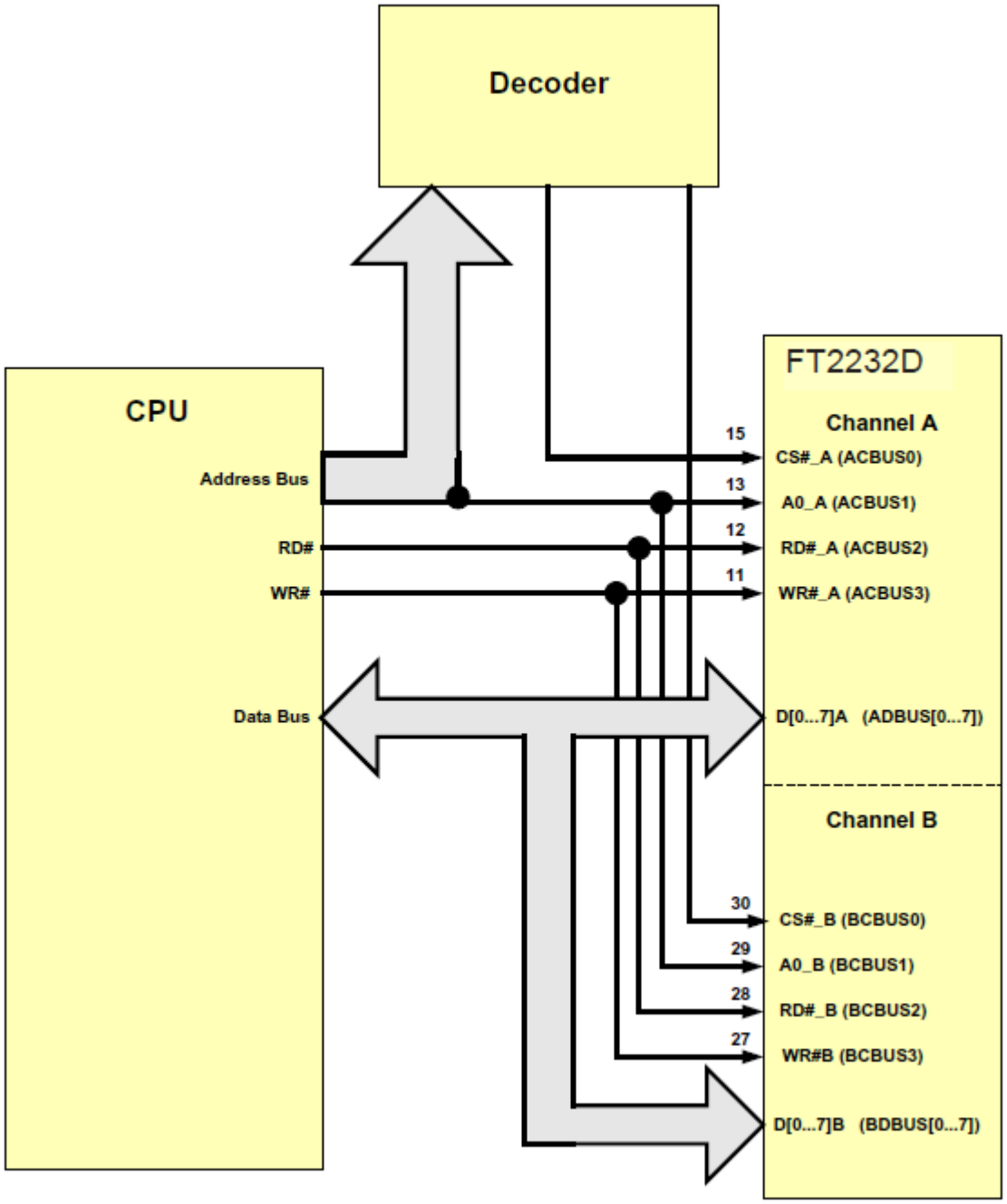

Figure 8.20 CPU FIFO Dual Channel Interface Example 2

Figure 8.21 shows an example where both channels A and B of the FT2232D are used in CPU FIFO mode to interface with a CPU. This configuration gives the CPU access to both of the FT2232D's data pipes.

9 Package Parameters

The FT2232D is supplied in a 48 pin lead (Pb) free LQFP package. This package has a 7mm x 7mm body (9mm x 9mm including leads) with leads on a 0.5mm pitch. The FT2232D is fully compliant with the European Union RoHS directive.
The below drawing shows the LQFP-48 package – all dimensions are in millimeters.

XXYY = Date Code (XX = 2 digit week number, YY = 1 or 2 digit year number.

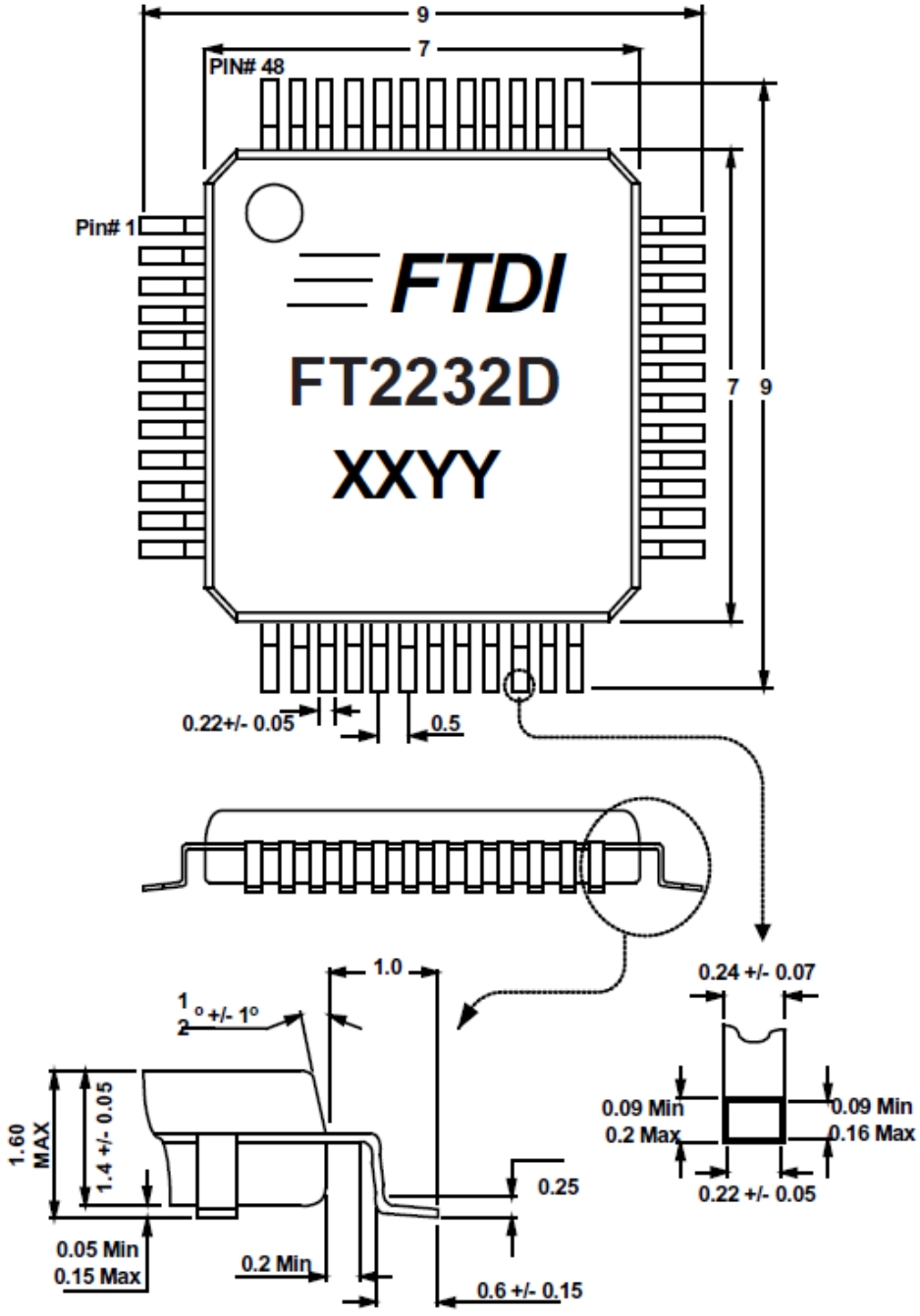

Figure 9.1 48LD Lead Free LQFP Package Dimensions

Tape and reel information is available in the following applications note:

http://www.ftdichip.com/Documents/AppNotes/AN_116_FTDI%20Devices%20Tape%20and%20Reel%20Dimensions.pdf

10 Contact Information

Head Office – Glasgow, UK

Future Technology Devices International Limited
Unit 1, 2 Seaward Place,
Centurion Business Park
Glasgow, G41 1HH
United Kingdom
Tel: +44 (0) 141 429 2777
Fax: +44 (0) 141 429 2758

E-mail (Sales) sales1@ftdichip.com
E-mail (Support) support1@ftdichip.com
E-mail (General Enquiries) admin1@ftdichip.com
Web Site URL http://www.ftdichip.com
Web Shop URL http://www.ftdichip.com

Branch Office – Taipei, Taiwan

Future Technology Devices International Limited (Taiwan)
2F, No. 516, Sec. 1, NeiHu Road
Taipei 114
Taiwan , R.O.C.
Tel: +886 (0) 2 8797 1330
Fax: +886 (0) 2 8751 9737

E-mail (Sales) tw.sales1@ftdichip.com
E-mail (Support) tw.support1@ftdichip.com
E-mail (General Enquiries) tw.admin1@ftdichip.com
Web Site URL http://www.ftdichip.com

Branch Office – Hillsboro, Oregon, USA

Future Technology Devices International Limited (USA)
7235 NW Evergreen Parkway, Suite 600
Hillsboro, OR 97123-5803
USA
Tel: +1 (503) 547 0988
Fax: +1 (503) 547 0987

E-Mail (Sales) us.sales@ftdichip.com
E-mail (Support) us.support@ftdichip.com
E-mail (General Enquiries) us.admin@ftdichip.com
Web Site URL http://www.ftdichip.com

Branch Office – ShangHai, China

Future Technology Devices International Limited (China)
Room 408, 317 Xianxia Road,
ChangNing District,
ShangHai, P.R. China

Tel: +86 (21) 62351596
Fax: +86 (21) 62351595

E-Mail (Sales) cn.sales@ftdichip.com
E-mail (Support) cn.support@ftdichip.com
E-Mail (General Enquiries) cn.admin@ftdichip.com
Web Site URL http://www.ftdichip.com

Distributor and Sales Representatives

Please visit the Sales Network page of the FTDI Web site for the contact details of our distributor(s) and sales representative(s) in your country.

Appendix A – References

AN2232-02, "Bit Mode Functions for the FT2232D"
http://www.ftdichip.com/Documents/AppNotes/AN2232C-02_FT2232CBitMode.pdf

AN232B-01, "FT232BM/FT245BM Bit Bang Mode"
http://www.ftdichip.com/Documents/AppNotes/AN232B-01_BitBang.pdf

AN2232L_108, "Command Processor for MPSSE and MCU Host Bus Emulation Modes".
http://www.ftdichip.com/Documents/AppNotes/AN_108_Command_Processor_for_MPSSE_and_MCU_Host_Bus_Emulation_Modes.pdf

Useful Application Notes and Projects

http://www.ftdichip.com/Documents/AppNotes/AN_107_AdvancedDriverOptions_AN_000073.pdf

http://www.ftdichip.com/Documents/AppNotes/AN_121_FTDI_Device_EEPROM_User_Area_Usage.pdf

http://www.ftdichip.com/Documents/AppNotes/AN_120_Aliasing_VCP_Baud_Rates.pdf

http://www.ftdichip.com/Resources/Utilities/AN_127_User_Guide_For_FT2232HD_Factory%20test%20utility.pdf

http://www.ftdichip.com/Documents/AppNotes/AN_131_FT2232D_H_Fast%20Opto-Isolated%20Serial%20Interface%20mode.pdf

http://www.ftdichip.com/Documents/AppNotes/AN2232C-02_FT2232CBitMode.pdf

http://www.ftdichip.com/Documents/AppNotes/AN_108_Command_Processor_for_MPSSE_and_MCU_Host_Bus_Emulation_Modes.pdf

http://www.ftdichip.com/Projects/MPSSE.htm#SPI

http://www.ftdichip.com/Projects/MPSSE.htm#I2C

http://www.ftdichip.com/Projects/MPSSE.htm#JTAG

Appendix B - List of Figures and Tables

List of Figures

List of Tables

Appendix C - Revision History

Version 0.91	Initial Datasheet Created	October 2006
Version 2.00	Contact details edited and Corrected FT2232D features (Removed reference to Isochronous support) Added links for sample project code that demonstrate how to use the devices Multi-Protocol Synchronous Serial Engine (MPSSE) Added Windows 7 support. Added TID number. Added apps notes references. Released on the Web	10/11/09
Version 2.01	Corrected table pin No. on Table 3.1.1 Corrected figure 6.6 label from Self Powered to Bus Powered	27/01/10
Version 2.02	Added ESD specifications	21/05/10
Version 2.03	Edited table 5.2	31/05/10
Version 2.04	Edited table 3.5.5 and 3.5.6 CPU FIFO mode signal names Added section 1.2 USB compliant Edited Figure 2.1	27/07/10
Version 2.05	Corrected table 5.5 and 5.6 to show "High Current" Corrected Section 8.7 title	18/04/11

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

FTDI:
FT2232D-REEL