Future Technology Devices International Ltd.

FT200XD

(USB I2C SLAVE IC)



The FT200XD is a USB to I^2C interface with the following advanced features:

- Single chip USB to I^2C interface.
- Up to 3.4MHz, high speed mode, I^2C supported
- Entire USB protocol handled on the chip. No USB specific firmware programming required.
- Fully integrated 2048 byte multi-time-programmable (MTP) memory storing device descriptors and CBUS I/O configuration.
- Fully integrated clock generation with no external crystal required plus optional clock output selection enabling a glue-less interface to external MCU or FPGA.
- 512 byte receive buffer and 512 byte transmit buffer utilising buffer smoothing technology to allow for high data throughput.
- FTDI's royalty-free Virtual Com Port (VCP) and Direct (D2XX) drivers eliminate the requirement for USB driver development in most cases.
- Unique USB FTDIChip-ID™ feature.
- Configurable CBUS I/O pin.
- Transmit and receive LED drive signals.
- USB Battery Charger Detection. Allows for USB peripheral devices to detect the presence of a higher power source to enable improved charging.
- Device supplied pre-programmed with unique USB serial number.
- USB Power Configurations; supports bus-powered, self-powered and bus-powered with power switching.
- Integrated +3.3V level converter for USB I/O.
- True 3.3VCMOS drive output and TTL input; Operates down to 1V8 with external pull-ups. Tolerant of 5V input.
- Configurable I/O pin output drive strength; 4 mA(min) and 16 mA(max).
- Integrated power-on-reset circuit.
- Fully integrated AVCC supply filtering - no external filtering required.
- + 5V Single Supply Operation.
- Internal 3V3/1V8 LDO regulators
- Low operating and USB suspend current; 8mA (active-typ) and 125uA (suspend-typ).
- UHCI/OHCI/EHCI host controller compatible.
- USB 2.0 Full Speed compatible.
- Extended operating temperature range; -40ºC to +85ºC.
- Available in compact Pb-free 10 Pin DFN package (3mmx3mm) (RoHS compliant).

Neither the whole nor any part of the information contained in, or the product described in this manual, may be adapted or reproduced in any material or electronic form without the prior written consent of the copyright holder. This product and its documentation are supplied on an as-is basis and no warranty as to their suitability for any particular purpose is either made or implied. Future Technology Devices International Ltd will not accept any claim for damages howsoever arising as a result of use or failure of this product. Your statutory rights are not affected. This product or any variant of it is not intended for use in any medical appliance, device or system in which the failure of the product might reasonably be expected to result in personal injury. This document provides preliminary information that may be subject to change without notice. No freedom to use patents or other intellectual property rights is implied by the publication of this document. Future Technology Devices International Ltd, Unit 1, 2 Seaward Place, Centurion Business Park, Glasgow G41 1HH United Kingdom. Scotland Registered Company Number: SC136640

1 Typical Applications

- Upgrading Legacy Peripherals to USB
- Utilising USB to add system modularity
- Incorporate USB interface to enable PC transfers for development system communication
- Motherboard and system monitoring through USB
- USB dongle implementations for Software/ Hardware Encryption and Wireless Modules
- Interfacing MCU/PLD/FPGA based designs to add USB connectivity
- USB Instrumentation
- USB Industrial Control
- USB Digital Camera Interface
- Capable of detecting dedicated USB charger ports for charging batteries with high current

1.1 Driver Support

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

- Windows 8 32-,64-bit
- Windows 7 32-,64-bit
- Windows Vista and Vista 64-bit
- Windows XP and XP 64-bit
- Windows Embedded Operating Systems
- Server 2003, XP and Server 2008
- Windows CE 4.2, 5.0 and 6.0
- Mac OS-X
- Linux 3.2 and greater
- Android OS

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

- Windows 8 32-,64-bit
- Windows 7 32-,64-bit
- Windows Vista and Vista 64-bit
- Windows XP and XP 64-bit
- Windows Embedded Operating Systems
- Server 2003, XP and Server 2008
- Windows CE 4.2, 5.0 and 6.0
- Mac OS-X
- Linux 2.6 and greater
- Android OS

The drivers listed above are all available to download for free from FTDI website (www.ftdichip.com). Various 3rd party drivers are also available for other operating systems - see FTDI website(www.ftdichip.com) for details.

For driver installation, please refer to http://www.ftdichip.com/Documents/InstallGuides.htm

1.2 Part Numbers

Part Number	Package
FT200XD-x	10 Pin DFN

Table 1.1 Ordering part numbers

Note: Packing codes for -x is:

- R: Taped and Reel, 5,000pcs per reel

- T: Tray packing, 490pcs per tray

For example: FT200XD-R is 5,000pcs taped and reel packing

Copyright © Future Technology Devices International Limited

1.3 USB Compliant

The FT200XD is fully compliant with the USB 2.0 specification and has been given the USB-IF Test-ID (TID) 40001459 (Rev D).



Copyright © Future Technology Devices International Limited

2 FT200XD Block Diagram

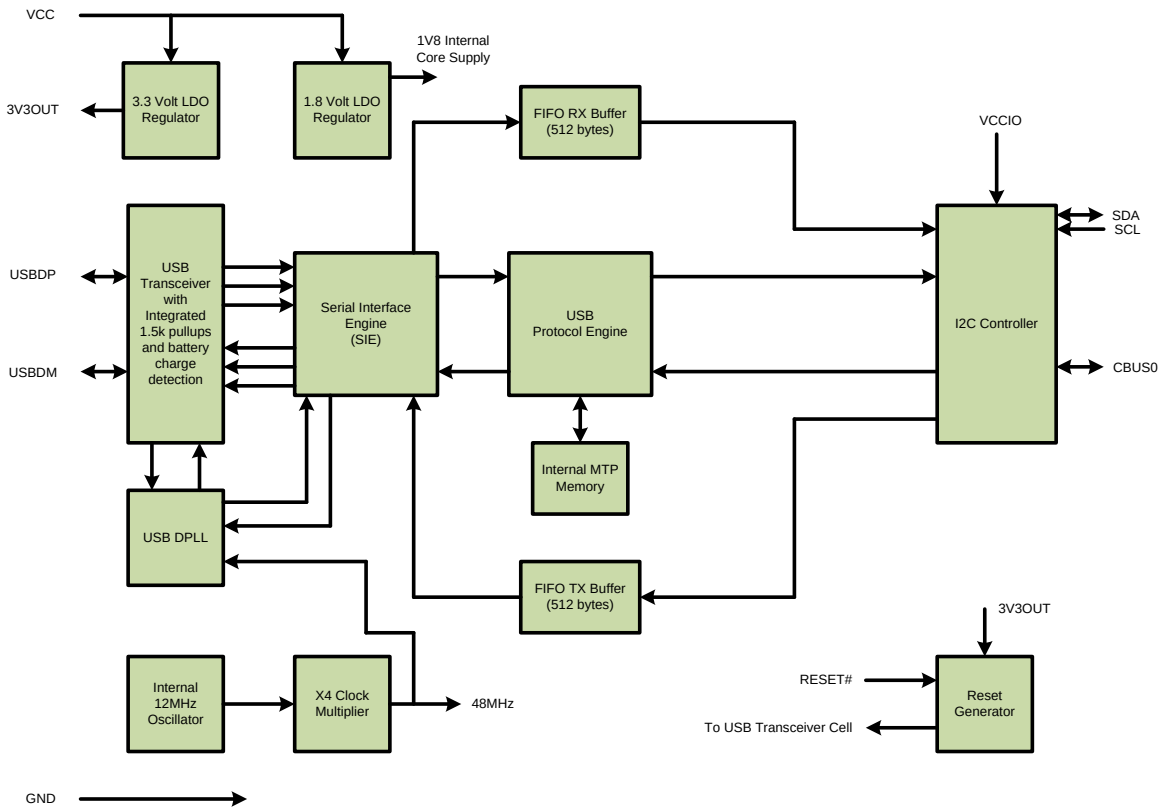


Figure 2.1 FT200XD Block Diagram

For a description of each function please refer to Section 4.

Table of Contents

1 Typical Applications 2
1.1 Driver Support 2
1.2 Part Numbers 2
1.3 USB Compliant 3
2 FT200XD Block Diagram 4
3 Device Pin Out and Signal Description 7
3.1 10-LD DFN Package 7
3.1.1 DFN Package Pin Out Description 7
3.2 CBUS Signal Options 9
4 Function Description 10
4.1 Key Features 10
4.2 Functional Block Descriptions 10
5 I2C Interface Description 12
6 Devices Characteristics and Ratings 13
6.1 Absolute Maximum Ratings 13
6.2 ESD and Latch-up Specifications 13
6.3 DC Characteristics 14
6.4 MTP Memory Reliability Characteristics 18
6.5 Internal Clock Characteristics 18
7 USB Power Configurations 19
7.1 USB Bus Powered Configuration 19
7.2 Self Powered Configuration 20
7.3 USB Bus Powered with Power Switching Configuration 21
8 Application Examples 22
8.1 USB to I2C Converter 22
8.2 USB Battery Charging Detection 23
9 USB and I2C Interfacing 25
9.1 Host Interface (USB) 25
9.1.1 VCP and D2xx Interfaces 25
9.1.2 Reading and Writing Data 26
9.2 I2C Interface 27
9.2.1 Addressing 27

Copyright © Future Technology Devices International Limited

9.2.2 Data Transfers 27
9.3 Other I2C Commands 29
10 Internal MTP Memory Configuration 31
10.1 Default Values 31
10.2 Methods of Programming the MTP Memory 32
10.2.1 Programming the MTP memory over USB 32
10.2.2 Programming the MTP memory over I2C 32
10.3 Memory Map 33
10.4 Hardware Requirements 34
10.5 Protocol 34
10.5.1 Address MTP memory (0x10) 34
10.5.2 Write MTP memory (0x12) 34
10.5.3 Read MTP memory (0x14) 34
10.5.4 Examples of Writing and Reading 35
11 Package Parameters 36
11.1 DFN-10 Package Mechanical Dimensions 36
11.2 Package Markings 37
11.3 Solder Reflow Profile 38
12 Contact Information 40
Appendix A – References 41
Document References 41
Acronyms and Abbreviations 41
Appendix B - List of Figures and Tables 42
List of Figures 42
List of Tables 42
Appendix C - Revision History 44

Copyright © Future Technology Devices International Limited

3 Device Pin Out and Signal Description

3.1 10-LD DFN Package

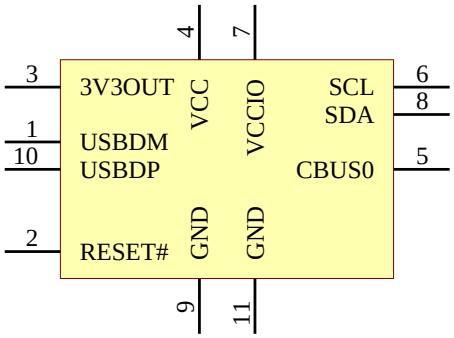


Figure 3.1 DFN Schematic Symbol

3.1.1 DFN Package Pin Out Description

Note: # denotes an active low signal.

Pin No.	Name	Type	Description
4	** VCC	POWER Input	5 V (or 3V3) supply to IC
7	VCCIO	POWER Input	1V8 - 3V3 supply for the IO cells
3	** 3V3OUT	POWER Output	3V3 output at 50mA. May be used to power VCCIO. When VCC is 3V3; pin 3 is an input pin and should be connected to pin 4.
9	GND	POWER Input	0V Ground input.

Table 3.1 Power and Ground

*Pin 11 is the centre pad on the chip package base. This should be connected to GND

** If VCC is 3V3 then 3V3OUT must also be driven with 3V3 input

Pin No.	Name	Type	Description
1	USBDM	INPUT	USB Data Signal Minus.
10	USBDP	INPUT	USB Data Signal Plus.
2	RESET#	INPUT	Reset input (active low).

Table 3.2 Common Function pins

Copyright © Future Technology Devices International Limited

Pin No.	Name	Type	Description
8	SDA	I/O	I^2C bi-directional data line. The signal is open drain.
6	SCL	Input	I^2C clock input
5	CBUS0	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device MTP memory. See CBUS Signal Options, Table 3.4.

Table 3.3 I^2C Interface and CBUS Group (see note 1)

Note:
1. When used in Input Mode, the input pins are pulled to VCCIO via internal 75kΩ (approx.) resistors. These pins can be programmed to gently pull low during USB suspend (PWREN# = "1") by setting an option in the MTP memory.
2. Clock stretching is not supported

Copyright © Future Technology Devices International Limited

3.2 CBUS Signal Options

The following options can be configured on the CBUS I/O pins. These options can be configured in the internal MTP memory using the software utility FT_PROG, which can be downloaded from the FTDI Utilities (www.ftdichip.com). The default configuration is noted below and described in Section 10.

CBUS Signal Option	Available On CBUS Pin	Description
TRI-STATE	CBUS0	IO Pad is tri-stated
DRIVE 1	CBUS0	Output a constant 1
DRIVE 0	CBUS0	Output a constant 0
PWREN#	CBUS0	Output is low after the device has been configured by USB, then high during USB suspend mode. This output can be used to control power to external logic P-Channel logic level MOSFET switch. Enable the interface pull-down option when using the PWREN# in this way.
SLEEP#	CBUS0	Goes low during USB suspend mode. Typically used to power down an external logic to reduce power consumption.
CLK24MHz	CBUS0	24 MHz Clock output.*
CLK12MHz	CBUS0	12 MHz Clock output.*
CLK6MHz	CBUS0	6 MHz Clock output.*
GPIO	CBUS0	CBUS bit bang mode option. . A separate application note, AN232R-01, available from FTDI website(www.ftdichip.com) describes in more detail how to use CBUS bit bang mode.
BCD Charger	CBUS0	Battery Charge Detect indicates when the device is connected to a dedicated battery charger host. Active high output.
BCD Charger#	CBUS0	Inverse of BCD Charger (open drain)
BitBang_WR#	CBUS0	Synchronous and asynchronous bit bang mode WR# strobe output.
BitBang_RD#	CBUS0	Synchronous and asynchronous bit bang mode RD# strobe output.
I2C_TXE#	CBUS0	Transmit buffer empty, used to indicate to I^2C master device status of the FT200XDtransmit buffer
I2C_RXF#	CBUS0	Receive buffer full, used to indicate to I^2C master device status of FT200XD receive buffer
VBUS Sense	CBUS0	Input to detect when VBUS is present.
Time Stamp	CBUS0	Toggle signal which changes state each time a USB SOF is received
Keep_Awake#	CBUS0	Prevents the device from entering suspend state when unplugged. May be used if programming the MTP memory over I^2C

Table 3.4 CBUS Configuration Control

*When in USB suspend mode the outputs clocks are also suspended.

Copyright © Future Technology Devices International Limited

4 Function Description

The FT200XD is a package and pin optimised USB to I^2C interface device which simplifies USB to I^2C designs and reduces external component count by fully integrating an external EEPROM and a clock circuit which requires no external crystal in a small form factor 3x3mm 10 pin package. It has been designed to operate efficiently with a USB host controller by using as little of the total USB bandwidth available as possible.

4.1 Key Features

Functional Integration. Fully integrated MTP memory, clock generation, AVCC filtering, Power-On_Reset (POR) and LDO regulators.

Configurable CBUS I/O Pin Options. The fully integrated MTP memory allows configuration of the Control Bus (CBUS) functionality and drive strength selection. There is 1 configurable CBUS I/O pin. These configurable options are defined in section 3.2.

The device is supplied with the most commonly used pin definitions pre-programmed - see Section 10 for details.

Asynchronous Bit Bang Mode with RD# and WR# Strobes. The FT200XD supports FTDI's previous chip generation bit-bang mode. In bit-bang mode, the 2 I^2C lines can be switched from the regular interface mode to a 2-bit general purpose I/O 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 timer (equivalent to the baud rate pre-scalar). In the FT200XD device this mode has been enhanced by outputting the internal RD# and WR# strobes signals which can be used to allow external logic to be clocked by accesses to the bit-bang I/O bus. This option will be described more fully in a separate application note available from FTDI website (www.ftdichip.com).

Synchronous Bit Bang Mode. The FT200XD supports synchronous bit bang mode. This mode differs from asynchronous bit bang mode in that the interface pins are only read when the device is written to. This makes it easier for the controlling program to measure the response to an output stimulus as the data returned is synchronous to the output data. An application note, AN232R-01, available from FTDI website(www.ftdichip.com) describes this feature.

Low Power Consumption, The FT200XD is capable of operating at a voltage supply between +3.3V and +5V with a nominal operational mode current of 8mA and a nominal USB suspend mode current of 125µA. This allows greater margin for peripheral designs to meet the USB suspend mode current limit of 2.5mA. An integrated level converter within the I^2C interface allows the FT200XD to interface to I^2C logic running at +1.8V to +3.3V (5V tolerant).

4.2 Functional Block Descriptions

The following paragraphs detail each function within the FT200XD. Please refer to the block diagram shown in Figure 2.1.

Internal MTP Memory. The internal MTP memory in the FT200XD is used to store USB Vendor ID (VID), Product ID (PID), device serial number, product description string and various other USB configuration descriptors. The internal MTP memory is also used to configure the CBUS pin functions. The FT200XD is supplied with the internal MTP memory pre-programmed as described in Section10. A user area of the internal MTP memory is available to system designers to allow storing of additional data from the user application over USB. The internal MTP memory descriptors can be programmed in circuit, over USB without any additional programming voltage requirements. The descriptors can be programmed using the FTDI utility software called FT_PROG, which can be downloaded from FTDI Utilitieson the FTDI website (www.ftdichip.com). Additionally the MTP memory can be configured over the I^2C interface.

+1.8V LDO Regulator. The +1.8V LDO regulator generates the +1.8V reference voltage for driving the internal core of the IC.

+3.3V LDO Regulator. The +3.3V LDO regulator generates the +3.3V reference voltage for driving the USB transceiver cell output buffers. It requires an external decoupling capacitor to be connected to the 3V3OUT regulator output pin. It also provides +3.3V power to the 1.5kΩ internal pull up resistor on USBDP. The main function of the LDO is to power the USB Transceiver and the Reset Generator Cells

Copyright © Future Technology Devices International Limited

rather than to power external logic. However, it can be used to supply external circuitry requiring a +3.3V nominal supply with a maximum current of 50mA.

USB Transceiver. The USB Transceiver Cell provides the USB 1.1 / USB 2.0 full-speed physical interface to the USB cable. The output drivers provide +3.3V level slew rate control signalling, whilst a differential input receiver and two single ended input receivers provide USB data in, Single-Ended-0 (SE0) and USB reset detection conditions respectfully. This function also incorporates a 1.5kΩ pull up resistor on USBDP. The block also detects when connected to a USB power supply which will not enumerate the device but still supply power and may be used for battery charging.

USB DPLL. The USB DPLL cell locks on to the incoming NRZI USB data and generates recovered clock and data signals for the Serial Interface Engine (SIE) block.

Internal 12MHz Oscillator - The Internal 12MHz Oscillator cell generates a 12MHz reference clock. This provides an input to the x4 Clock Multiplier function. The 12MHz Oscillator is also used as the reference clock for the SIE, USB Protocol Engine and UART FIFO controller blocks.

Clock Multiplier / Divider. The Clock Multiplier / Divider takes the 12MHz input from the Internal Oscillator function and generates the 48MHz, 24MHz, 12MHz and 6MHz reference clock signals. The 48MHz clock reference is used by the USB DPLL 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 with the USB 2.0 specification, it performs bit stuffing/un-stuffing and CRC5/CRC16 generation. It also checks the CRC 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 requests generated by the USB host controller and the commands for controlling the functional parameters of the I^2C in accordance with the USB 2.0 specification chapter 9.

FIFO RX Buffer (512 bytes). Data sent from the USB host controller to the I2C interface via the USB data OUT endpoint is stored in the FIFO RX (receive) buffer. Data is removed from the buffer to the I2C transmit register under control of the I2C Controller.(RX is relative to the USB interface).

FIFO TX Buffer (512 bytes). Data from the I^2C receive register is stored in the TX buffer. The USB host controller removes data from the FIFO TX Buffer by sending a USB request for data from the device data IN endpoint.(TX is relative to the USB interface).

I^2C Controller. Module to handle the latching in and out of serial data on the I^2C interface. Supports up to 3.4MHz, High Speed Serial Mode.

RESET Generator - The integrated Reset Generator Cell provides a reliable power-on reset to the device internal circuitry at power up. The RESET# input pin allows an external device to reset the FT200XD.

RESET# can be tied to 3V3OUT.

Copyright © Future Technology Devices International Limited

5 I²C Interface Description

I²C (Inter Integrated Circuit) is a multi-master serial bus invented by Philips. I²C uses two bi-directional open-drain wires called serial data (SDA) and serial clock (SCL). Common I²C bus speeds are the 100 kbit/s standard mode (SM), 400 kbit/s fast mode (FM), 1 Mbit/s Fast mode plus (FM+), and 3.4 Mbit/s High Speed mode (HS)

An I²C bus node can operate either as a master or a slave:

- Master node – issues the clock and addresses slaves
- Slave node – receives the clock line and address.

The FT200XD device only operates as a slave, and is capable of speeds up to 3.4MBit/s. There are four potential modes of operation for a given bus device, although most devices only use a single role (Master or Slave) and its two modes (Transmit and Receive):

- Master transmit – sending data to a slave
- Master receive – receiving data from a slave
- Slave transmit – sending data to a master
- Slave receive – receiving data from the master

The master is initially in master transmit mode by sending a start bit followed by the 7-bit address of the slave it wishes to communicate with, which is finally followed by a single bit representing whether it wishes to write(0) to or read(1) from the slave.

If the slave exists on the bus then it will respond with an ACK bit (active low for acknowledged) for that address. The master then continues in either transmit or receive mode (according to the read/write bit it sent), and the slave continues in its complementary mode (receive or transmit, respectively).

The address and the data bytes are sent most significant bit first. The start bit is indicated by a high-to-low transition of SDA with SCL high; the stop bit is indicated by a low-to-high transition of SDA with SCL high.

If the master wishes to write to the slave then it repeatedly sends a byte with the slave sending an ACK bit. (In this situation, the master is in master transmit mode and the slave is in slave receive mode.)

If the master wishes to read from the slave then it repeatedly receives a byte from the slave, the master sends an ACK bit after every byte but the last one. (In this situation, the master is in master receive mode and the slave is in slave transmit mode.)

The master then ends transmission with a stop bit, or it may send another START bit if it wishes to retain control of the bus for another transfer (a "combined message").

I²C defines three basic types of message, each of which begins with a START and ends with a STOP:

- Single message where a master writes data to a slave;
- Single message where a master reads data from a slave;
- Combined messages, where a master issues at least two reads and/or writes to one or more slaves

In a combined message, each read or write begins with a START and the slave address. After the first START, these are also called repeated START bits; repeated START bits are not preceded by STOP bits, which is how slaves know the next transfer is part of the same message.

Please refer to the I²C specification for more information on the protocol.

Copyright © Future Technology Devices International Limited

6 Devices Characteristics and Ratings

6.1 Absolute Maximum Ratings

The absolute maximum ratings for the FT200XD devices are as follows. These are in accordance with the Absolute Maximum Rating System (IEC 60134). Exceeding these may cause permanent damage to the device.

Parameter	Value	Unit	Conditions
Storage Temperature	-65°C to 150°C	Degrees C	
Floor Life (Out of Bag) At Factory Ambient (30°C / 60% Relative Humidity)	168 Hours (IPC/JEDEC J-STD-033A MSL Level 3 Compliant)*	Hours	
Ambient Operating Temperature (Power Applied)	-40°C to 85°C	Degrees C	
MTTF FT200XD	TBD	Hours	
MTTF FT200XD	TBD	Hours	
VCC Supply Voltage	-0.3 to +5.5	V	
VCCIO IO Voltage	-0.3 to +4.0	V	
DC Input Voltage – USBDP and USBDM	-0.5 to +3.63	V	
DC Input Voltage – High Impedance Bi-directional (powered from VCCIO)	-0.3 to +5.8	V	
DC Output Current – Outputs	22	mA	

Table 6.1 Absolute Maximum Ratings

* 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 +125°C and baked for up to 17 hours.

6.2 ESD and Latch-up Specifications

Description	Specification
Human Body Mode (HBM)	**> ± 2kV**
Machine mode (MM)	**> ± 200V**
Charged Device Mode (CDM)	**> ± 500V**
Latch-up	**> ± 200mA**

Table 6.2 ESD and Latch-Up Specifications

Copyright © Future Technology Devices International Limited

6.3 DC Characteristics

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

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
VCC	VCC Operating Supply Voltage	2.97	5	5.5	V	Normal Operation
VCC2	VCCIO Operating Supply Voltage	1.62	---	3.63	V	
Icc1	Operating Supply Current	6.8	8	9.1	mA	Normal Operation
Icc2	Operating Supply Current		125		µA	USB Suspend
3V3	3.3v regulator output	2.97	3.3	3.63	V	VCC must be greater than 3V3 otherwise 3V3OUT is an input which must be driven with 3.3V in this instance pins 3 and 4 of the package should be shorted together.

Table 6.3 Operating Voltage and Current

Copyright © Future Technology Devices International Limited

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
Voh	Output Voltage High	2.97	VCCIO	VCCIO	V	Ioh =-2mA I/O Drive strength* = 4mA
		2.97	VCCIO	VCCIO	V	I/O Drive strength* = 8mA
		2.97	VCCIO	VCCIO	V	I/O Drive strength* = 12mA
		2.97	VCCIO	VCCIO	V	I/O Drive strength* = 16mA
Vol	Output Voltage Low		0	0.4	V	Iol = +2mA I/O Drive strength* = 4mA
			0	0.4	V	I/O Drive strength* = 8mA
			0	0.4	V	I/O Drive strength* = 12mA
			0	0.4	V	I/O Drive strength* = 16mA
Vil	Input low Switching Threshold			0.8	V	LVTTL
Vih	Input High Switching Threshold	2.0			V	LVTTL
Vt	Switching Threshold		1.49		V	LVTTL
Vt-	Schmitt trigger negative going threshold voltage		1.15		V	
Vt+	Schmitt trigger positive going threshold voltage		1.64		V	
Rpu	Input pull-up resistance	40	75	190	KΩ	Vin = 0
Rpd	Input pull-down resistance	40	75	190	KΩ	Vin =VCCIO
Iin	Input Leakage Current	-10	+/-1	10	µA	Vin = 0
Ioz	Tri-state output leakage current	-10	+/-1	10	µA	Vin = 5.5V or 0

Table 6.4 I/O Pin Characteristics VCCIO = +3.3V (except USB PHY pins)

* The I/O drive strength and slow slew-rate are configurable in the MTP memory.

Copyright © Future Technology Devices International Limited

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
Voh	Output Voltage High	2.25	VCCIO	VCCIO	V	Ioh = +/-2mA I/O Drive strength* = 4mA
		2.25	VCCIO	VCCIO	V	I/O Drive strength* = 8mA
		2.25	VCCIO	VCCIO	V	I/O Drive strength* = 12mA
		2.25	VCCIO	VCCIO	V	I/O Drive strength* = 16mA
Vol	Output Voltage Low		0	0.4	V	Iol = +/-2mA I/O Drive strength* = 4mA
			0	0.4	V	I/O Drive strength* = 8mA
			0	0.4	V	I/O Drive strength* = 12mA
			0	0.4	V	I/O Drive strength* = 16mA
Vil	Input low Switching Threshold			0.8	V	LVTTL
Vih	Input High Switching Threshold	0.8			V	LVTTL
Vt	Switching Threshold		1.1		V	LVTTL
Vt-	Schmitt trigger negative going threshold voltage		0.8		V	
Vt+	Schmitt trigger positive going threshold voltage		1.2		V	
Rpu	Input pull-up resistance	40	75	190	KΩ	Vin = 0
Rpd	Input pull-down resistance	40	75	190	KΩ	Vin =VCCIO
Iin	Input Leakage Current	-10	+/-1	10	µA	Vin = 0
Ioz	Tri-state output leakage current	-10	+/-1	10	µA	Vin = 5.5V or 0

Table 6.5 I/O Pin Characteristics VCCIO = +2.5V (except USB PHY pins)

* The I/O drive strength and slow slew-rate are configurable in the MTP memory.

Copyright © Future Technology Devices International Limited

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
Voh	Output Voltage High	1.62	VCCIO	VCCIO	V	Ioh = +/-2mA I/O Drive strength* = 4mA
		1.62	VCCIO	VCCIO	V	I/O Drive strength* = 8mA
		1.62	VCCIO	VCCIO	V	I/O Drive strength* = 12mA
		1.62	VCCIO	VCCIO	V	I/O Drive strength* = 16mA
Vol	Output Voltage Low		0	0.4	V	Iol = +/-2mA I/O Drive strength* = 4mA
			0	0.4	V	I/O Drive strength* = 8mA
			0	0.4	V	I/O Drive strength* = 12mA
			0	0.4	V	I/O Drive strength* = 16mA
Vil	Input low Switching Threshold			0.77	V	LVTTL
Vih	Input High Switching Threshold	1.6			V	LVTTL
Vt	Switching Threshold		0.77		V	LVTTL
Vt-	Schmitt trigger negative going threshold voltage		0.557		V	
Vt+	Schmitt trigger positive going threshold voltage		0.893		V	
Rpu	Input pull-up resistance	40	75	190	KΩ	Vin = 0
Rpd	Input pull-down resistance	40	75	190	KΩ	Vin =VCCIO
Iin	Input Leakage Current	-10	+/-1	10	µA	Vin = 0
Ioz	Tri-state output leakage current	-10	+/-1	10	µA	Vin = 5.5V or 0

Table 6.6 I/O Pin Characteristics VCCIO = +1.8V (except USB PHY pins)

* The I/O drive strength and slow slew-rate are configurable in the MTP memory.

Copyright © Future Technology Devices International Limited

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
Voh	Output Voltage High	VCC-0.2			V	
Vol	Output Voltage Low			0.2	V	
Vil	Input low Switching Threshold		-	0.8	V	
Vih	Input High Switching Threshold	2.0	-		V	

Table 6.7 USB I/O Pin (USBDP, USBDM) Characteristics

6.4 MTP Memory Reliability Characteristics

The internal 2048 Byte MTP Memory has the following reliability characteristics:

Parameter	Value	Unit
Data Retention	10	Years
Write Cycle	2,000 minimum	Cycles
Read Cycle	Unlimited	Cycles

Table 6.8 MTP Memory Characteristics

6.5 Internal Clock Characteristics

The internal Clock Oscillator has the following characteristics:

Parameter	Value			Unit
	Minimum	Typical	Maximum	
Frequency of Operation (see Note 1)	11.98	12.00	12.02	MHz
Clock Period	83.19	83.33	83.47	ns
Duty Cycle	45	50	55	%

Table 6.9 Internal Clock Characteristics

Note 1: Equivalent to +/-1667ppm

Copyright © Future Technology Devices International Limited

7 USB Power Configurations

The following sections illustrate possible USB power configurations for the FT200XD.

7.1 USB Bus Powered Configuration

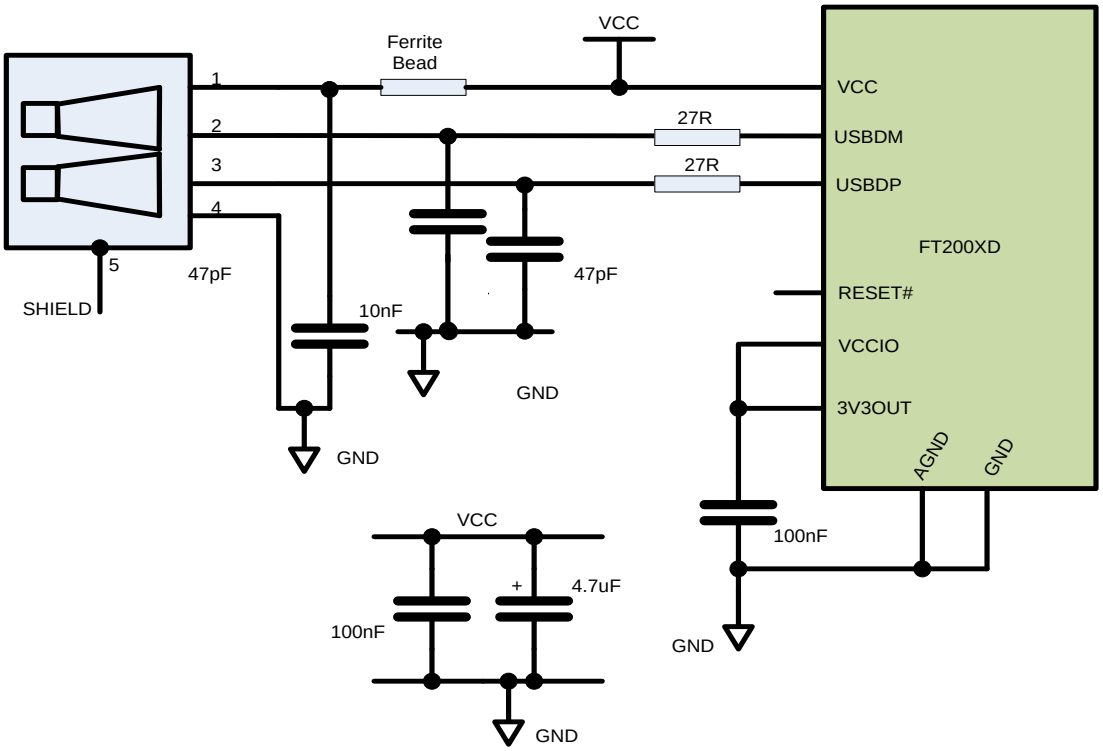


Figure 7.1 Bus Powered Configuration

Figure 7.1Illustrates the FT200XD in a typical USB bus powered design configuration. A USB bus powered device gets its power from the USB bus. Basic rules for USB bus power devices are as follows –

i) On plug-in to USB, the device must draw no more current than 100mA.
ii) In USB Suspend mode the device must draw no more than 2.5mA.
iii) A bus powered high power USB device (one that draws more than 100mA) should use the CBUS pins configured as PWREN# and use it to keep the current below 100mA on plug-in and 2.5mA on USB suspend. Using the CBUS pin for PWREN# prevents the FT200XD from indicating a USB Battery Charger is connected.
iv) A device that consumes more than 100mA cannot be plugged into a USB bus powered hub.
v) No device can draw more than 500mA from the USB bus.

The power descriptors in the internal MTP Memory of the FT200XDmustbe programmed to match the current drawn by the device.

A ferrite bead is connected in series with the USB power supply to reduce EMI noise from the FT200XD and associated circuitry being radiated down the USB cable to the USB host. The value of the Ferrite Bead depends on the total current drawn by the application. A suitable range of Ferrite Beads is available from Laird Technologies (http://www.lairdtech.com), for example Laird Technologies Part # MI0805K601R-10.

Note: If using PWREN# (available using the CBUS) the pin must be pulled to VCCIO using a 10kΩ resistor.

Copyright © Future Technology Devices International Limited

7.2 Self Powered Configuration

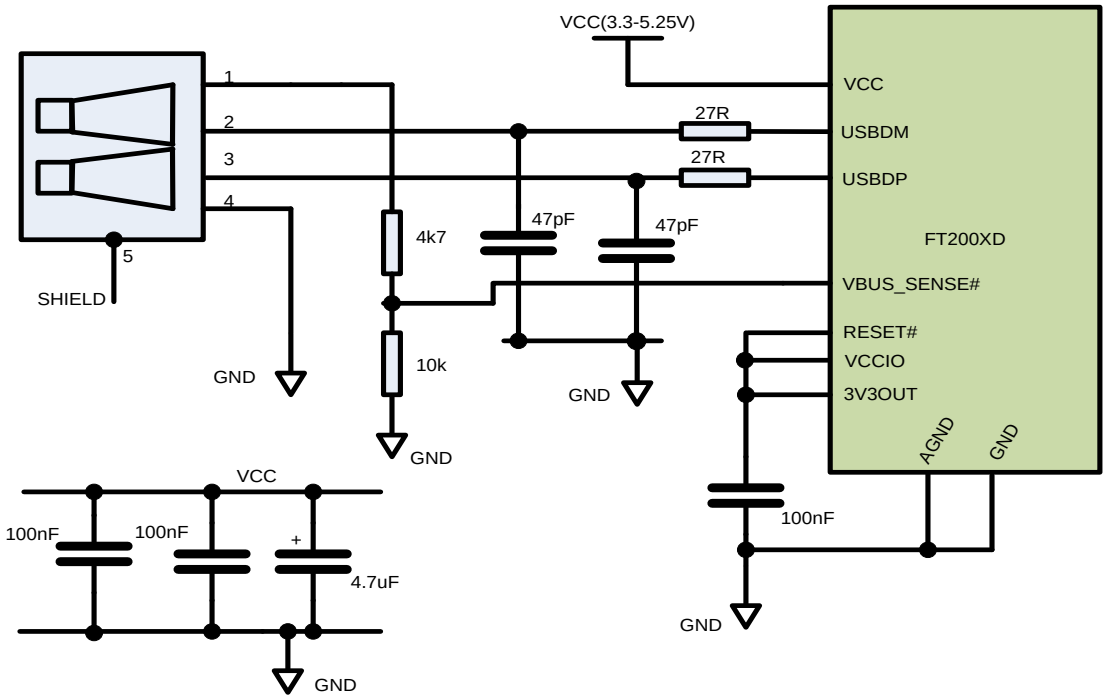


Figure 7.2 Self Powered Configuration

Figure 7.2 illustrates the FT200XD in a typical USB self-powered configuration. A USB self-powered device gets its power from its own power supply, VCC, and does not draw current from the USB bus. The basic rules for USB self-powered devices are as follows –

i) A self-powered device must not force current down the USB bus when the USB host or hub controller is powered down.
ii) A self-powered device can use as much current as it needs during normal operation and USB suspend as it has its own power supply.
iii) A self-powered device can be used with any USB host, a bus powered USB hub or a self-powered USB hub.

The power descriptor in the internal MTP memory of the FT200XD should be programmed to a value of zero (self-powered).

In order to comply with the first requirement above, the USB bus power (pin 1) is used to control the VBUS_Sense pin of the FT200XD device. When the USB host or hub is powered up an internal 1.5kΩ resistor on USBDP is pulled up to +3.3V, thus identifying the device as a full speed device to the USB host or hub. When the USB host or hub is powered off, VBUS_Sense pin will be low and the FT200XDis held in a suspend state. In this state the internal 1.5kΩ resistor is not pulled up to any power supply (hub or host is powered down), so no current flows down USBDP via the 1.5kΩ pull-up resistor. Failure to do this may cause some USB host or hub controllers to power up erratically.

Figure 7.2 illustrates a self-powered design which has a +3.3V to +5.25V supply.

Note:

1. When the FT200XD is in reset, the interface I/O pins are tri-stated. Input pins have internal 200kΩ pull-up resistors to VCCIO, so they will gently pull high unless driven by some external logic.

Copyright © Future Technology Devices International Limited

7.3 USB Bus Powered with Power Switching Configuration

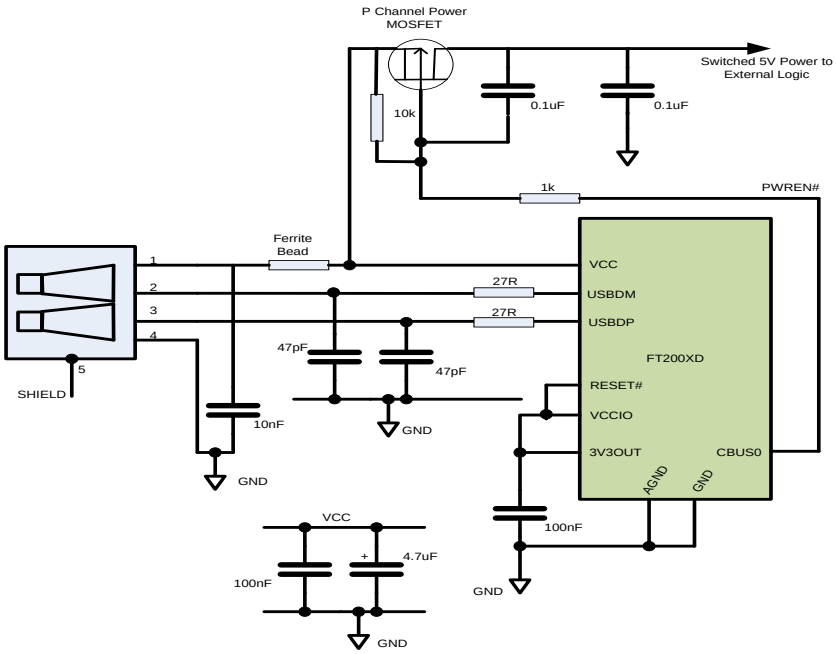


Figure 7.3 Bus Powered with Power Switching Configuration

A requirement of USB bus powered applications, is when in USB suspend mode, the application draws a total current of less than 2.5mA. This requirement includes external logic. Some external logic has the ability to power itself down into a low current state by monitoring the PWREN# signal. For external logic that cannot power itself down in this way, the FT200XD provides a simple but effective method of turning off power during the USB suspend mode.

Figure 7.3 shows an example of using a discrete P-Channel MOSFET to control the power to external logic. A suitable device to do this is an International Rectifier (www.irf.com) IRLML6402, or equivalent. It is recommended that a "soft start" circuit consisting of a 1kΩ series resistor and a 0.1µF capacitor is used to limit the current surge when the MOSFET turns on. Without the soft start circuit it is possible that the transient power surge, caused when the MOSFET switches on, will reset the FT200XD or the USB host/hub controller. The soft start circuit example shown in Figure 7.3 powers up with a slew rate of approximaely12.5V/ms. Thus supply voltage to external logic transitions from GND to +5V in approximately 400 microseconds.

As an alternative to the MOSFET, a dedicated power switch IC with inbuilt "soft-start" can be used. A suitable power switch IC for such an application is the Micrel (www.micrel.com) MIC2025-2BM or equivalent.

With power switching controlled designs the following should be noted:

i) The external logic to which the power is being switched should have its own reset circuitry to automatically reset the logic when power is re-applied when moving out of suspend mode.

ii) Set the Pull-down on Suspend option in the internal FT200XD MTP memory.

iii) The CBUS Pin should be configured as PWREN# in the internal FT200XD MTP memory, and used to switch the power supply to the external circuitry. Detection of a USB battery charger port is not possible.

iv) For USB high-power bus powered applications (one that consumes greater than 100mA, and up to 500mA of current from the USB bus), the power consumption of the application must be set in the Max Power field in the internal FT200XD MTP memory. A high-power bus powered application uses the descriptor in the internal FT200XD MTP memory to inform the system of its power requirements.

v) PWREN# gets its VCC from VCCIO. For designs using 3V3 logic, ensure VCCIO is not powered down using the external logic. In this case use the +3V3OUT.

Copyright © Future Technology Devices International Limited

8 Application Examples

The following sections illustrate possible applications of the FT200XD.

8.1 USB to I²C Converter

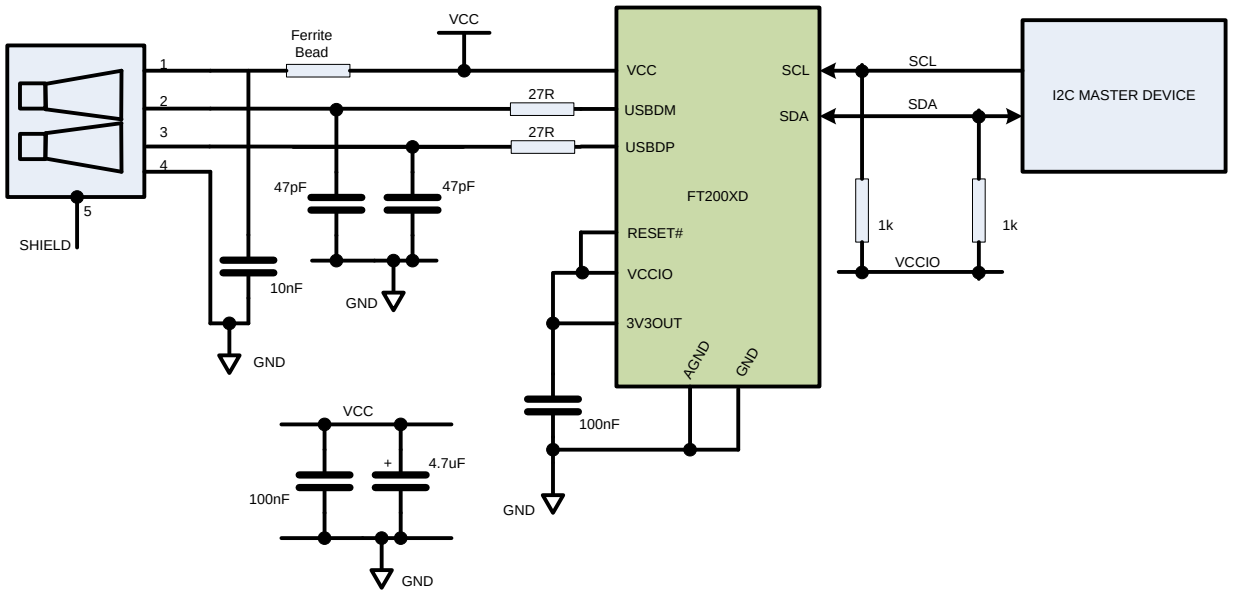


Figure 8.1 Application Example showing USB to I²C Converter

An example of using the FT200XD as an I²C peripheral is shown in Figure 8.1. The FT200XD is the slave on the I²C bus.

Therefore the clock supplied to the SCL pin must come from the I²C Master. The device will support standard I²C data rates such as 100 kbit/s standard mode (SM), 400 kbit/s fast mode (FM), 1 Mbit/s Fast mode plus (FM+), and 3.4 Mbit/s High Speed mode (HS).

The data line SDA is bi-directional.

The master is initially in master transmit mode by sending a start bit followed by the 7-bit address of the slave it wishes to communicate with, which is finally followed by a single bit representing whether it wishes to write(0) to or read(1) from the slave.

If the slave (FT200XD) exists on the bus then it will respond with an ACK bit (active low for acknowledged) for that address. The master then continues in either transmit or receive mode (according to the read/write bit it sent), and the slave continues in its complementary mode (receive or transmit, respectively).

The address and the data bytes are sent most significant bit first. The start bit is indicated by a high-to-low transition of SDA with SCL high; the stop bit is indicated by a low-to-high transition of SDA with SCL high.

If the master wishes to write to the slave then it repeatedly sends a byte with the slave sending an ACK bit. (In this situation, the master is in master transmit mode and the slave is in slave receive mode.)

If the master wishes to read from the slave then it repeatedly receives a byte from the slave, the master sending an ACK bit after every byte but the last one. (In this situation, the master is in master receive mode and the slave is in slave transmit mode.)

The master then ends transmission with a stop bit, or it may send another START bit if it wishes to retain control of the bus for another transfer (a "combined message").

I²C defines three basic types of message, each of which begins with a START and ends with a STOP:

Copyright © Future Technology Devices International Limited

- Single message where a master writes data to a slave;
- Single message where a master reads data from a slave;
- Combined messages, where a master issues at least two reads and/or writes to one or more slaves

In a combined message, each read or write begins with a START and the slave address. After the first START, these are also called repeated START bits; repeated START bits are not preceded by STOP bits, which is how slaves know the next transfer is part of the same message.

The I^2C address of the FT200XD is stored in the device internal MTP memory.

Please refer to the I^2C specification for more information on the protocol.

8.2 USB Battery Charging Detection

A recent addition to the USB specification (http://www.usb.org/developers/) is to allow for additional charging profiles to be used for charging batteries in portable devices. These charging profiles do not enumerate the USB port of the peripheral. The FT200XD device will detect that a USB compliant dedicated charging port (DCP) is connected. Once detected while in suspend mode a battery charge detection signal is provided to allow external logic to switch to charging mode as opposed to operation mode.

Copyright © Future Technology Devices International Limited

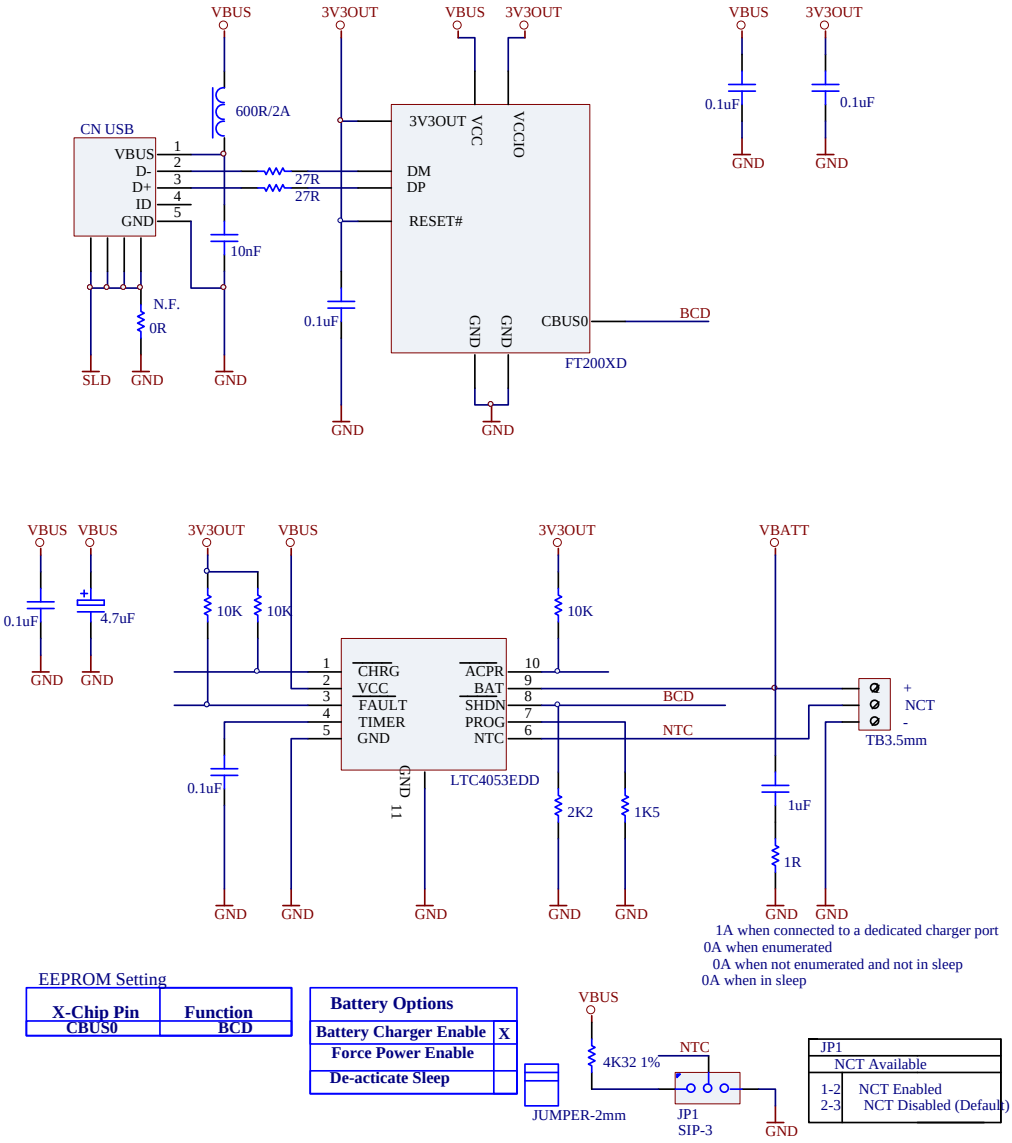


Figure 8.2 USB Battery Charging Detection

To use the FT200XD with battery charging detection the CBUS pins must be reprogrammed to allow for the BCD Charger output to switch the external charger circuitry on. The CBUS pins are configured in the internal MTP memory with the free utility FTPROG. If the charging circuitry requires an active low signal to enable it, the CBUS pin can be programmed to BCD Charger# as an alternative.

When connected to a USB compliant dedicated charging port (DCP, as opposed to a standard USB host) the device USB signals will be shorted together and the device suspended. The BCD charger signal will bring the LTC4053 out of suspend and allow battery charging to start. The charge current in the example above is 1A as defined by the resistance on the PROG pin.

To calculate the equivalent resistance on the PROG pin select a charge current, then Res = $1500V/I_{chg}$

For more configuration options of the LTC4053 refer to:

AN_175_Battery Charging Over USB

Note: If the FT200XD is connected to a standard host port such that the device is enumerated the battery charge detection signal is inactive as the device will not be in suspend.

Copyright © Future Technology Devices International Limited

9 USB and I²C Interfacing

This section covers the transfer of data from USB to I²C and vice versa.

Please note that the FT200XD and FT201X are I²C slave devices only and should be interfaced to an I²C host (often in a microcontroller or FPGA). If an I²C master is required, please see the FT232H, FT2232H, FT4232H and FT2232D devices.

This section covers transfer of data only. The USB and I²C interfaces on the FT200XD and FT201X can also be used for programming of the MTP memory, which is covered in a separate section.

Throughout this section, the reference FT-X applies to the FT200XD and FT201X devices, as the other members of the FT-X family do not have I²C interfaces.

9.1 Host Interface (USB)

From the host computer's point of view, the I²C data can be sent and received in the same way as when interfacing to one of the standard UART devices such as the FT232R. The FT-X handles the entire I²C protocol inside the chip and so reading and writing data does not require any special programming from the PC side. It can be treated as a simple data bridge.

9.1.1 VCP and D2xx Interfaces

Like the other FTDI devices, the host can use D2xx commands or can use a Virtual COM Port (VCP) to communicate with the device.

D2xx Interface

The D2xx method allows the application software to use the functions in the FTDI D2xx library to communicate directly with the device. This is a library provided free-of-charge by FTDI and is available within the driver download files at the link below.

http://www.ftdichip.com/Drivers/D2XX.htm

It includes functions to find the FTDI devices on the system, open a particular device, send data to the device and read data from the device. Any data written using FT_Write, for example, will be sent to the FT-X chip and will be available for the external I²C Master to read. As mentioned above, only the data itself needs to be sent as the FT-X handles all of the I²C specific protocol.

There are many other functions available and full details can be found in the D2xx Programmers Guide.

Virtual COM Port Interface

If using the Virtual Com Port (VCP), the device will appear as if it were a real COM port on the computer. This is useful where an application has already been written to use an RS232 port on the computer as it allows that application to treat the FT-X as if it were a real COM port. This port can be opened in a terminal program or a custom application in the same way as a serial port would be opened. Data can then be sent or received using standard serial / COM port functions.

No I²C decoding is required as the chip itself handles the I²C protocol. Because of this, even I²C versions of the FT-X family can be used with the VCP interface. Any data which the host PC sends to the Virtual COM Port (for example, typed into the terminal window in HyperTerminal) will be sent over USB to the FT-X and can then be read by the external I²C Master. Likewise, any data written to the FT-X over I²C will be sent to the PC where the terminal program will display it.

Since the FT-X is the I²C slave and does not generate a clock signal, the settings such as baud rate and handshaking do not have any effect.

Copyright © Future Technology Devices International Limited

D2xx and VCP Interface under Windows

In Windows systems, the VCP driver is actually an additional layer on top of the D2xx driver. The D2xx driver is always loaded and the VCP later may or may not be loaded on top of it, depending on the requirements of the application.

When VCP is disabled, the device will appear only under Universal Serial Bus Controllers in the Device Manager and no COM port is exposed. When the VCP is enabled, the device will appear under both the Universal Serial Bus Controllers and the Ports (COM & LPT) sections in Device Manager.

The VCP interface will be disabled by default in the FT200XD and FT201X but can be enabled in one of the following ways:

- An option bit in the MTP memory on the chip is checked each time that the device is enumerated, and can cause the host computer to load the VCP layer. This allows each individual hardware unit to enable or disable VCP as per its requirements. The MTP information in section 10 has further details. FT_Prog can also be used to modify the MTP settings.
- The user may open the Properties for the device under Universal Serial Bus Controllers in Device Manager in Windows and tick a box to load the VCP. Re-enumeration of the FT-X is necessary to enable the new setting.
- By editing the driver FtdiBus.inf file (please refer to AN_107). Note that the FT_INF utility available from the FTDI website can be used to help create the modified inf files.

In other Operating systems (e.g. Linux, Mac and Windows CE), there are separate drivers for D2xx and VCP, and only one of these drivers may be installed at any time. Therefore, the mode is selected by installing the associated driver instead and the selection defined in the bullet points above has no effect.

9.1.2 Reading and Writing Data

Data from FT-X to the Host

When data is to be sent from the host computer to the external I^2C master, the software application writes this data to the FTDI driver using VCP or D2xx and this data is sent by the driver over USB to the buffer in the device.

The external I^2C master device may then perform an I^2C read (Master receiver, Slave transmitter) operation to retrieve the data.

The external I^2C master should check if data is available to ensure that the data which has been read is valid. The methods for this are described in section 9.2.2 below.

Data from Host to the FT-X

When data is to be sent from the external I^2C master to the host computer, the external master performs an I^2C write (Master transmitter, Slave receiver) operation to the FT-X.

The FT-X stores this in its buffer and the data is sent back to the driver on the host computer over USB. As with other FTDI devices, this happens when either the buffer in the device fills or when the latency timer rolls over (so that partially filled buffers do not wait indefinitely), whichever happens first.

Note that when writing data to the FT-X over I^2C, the external master should check whether there is space available in the buffer inside the FT-X. It can do this using the methods shown in section 9.2.2 below.

The host computer application can then read the data from the driver buffer using the VCP interface or the D2xx commands. In the case of D2xx, the number of bytes available may be determined using the FT_GetQueueStatus command before doing the FT_Read.

Copyright © Future Technology Devices International Limited

9.2 I²C Interface

From an I²C point of view, the FT-X behaves as a standard I²C slave. Data transfers take the same form as a standard I²C communication. In addition to reading and writing, there are some other commands which can be used to determine the status of the device and these are covered in this section and in further detail in section 9.3.

Note: This section uses 7-bit addressing in the examples. It uses eight characters to describe the seven address bits and the single read/write bit. For example, 1110 011 1 means address 0x73 with the Read/Write bit set to 1, and therefore corresponds to a read of address 0x73. Data values are represented in bytes, for example, 1010 0101 for data value 0xA5.

9.2.1 Addressing

The FT-X can be given a custom slave address on the I²C bus. This is stored in MTP memory and can be re-programmed over USB or I²C. Please see the separate MTP Programming information in section 10. The slave address is used when reading and writing to the device. The FT-X supports both 7-bit and 10-bit addressing.

The general call address (0000 000 0) is for addressing every device on the bus. Some slaves do not support the General Call Address though the FT-X devices do support it. They will acknowledge this address, receive the second byte and interpret it. There are several commands available for this second byte. The FT-X will support the Software Reset command which is part of the I2C specification as well as other custom FTDI commands. These are covered in more detail later in the Other I2C Commands section.

9.2.2 Data Transfers

Reading and Writing

For all I²C data transfers to and from the FT-X, the slave address of the FT-X is used. The figure below shows a typical 7-bit address transfer for each direction.

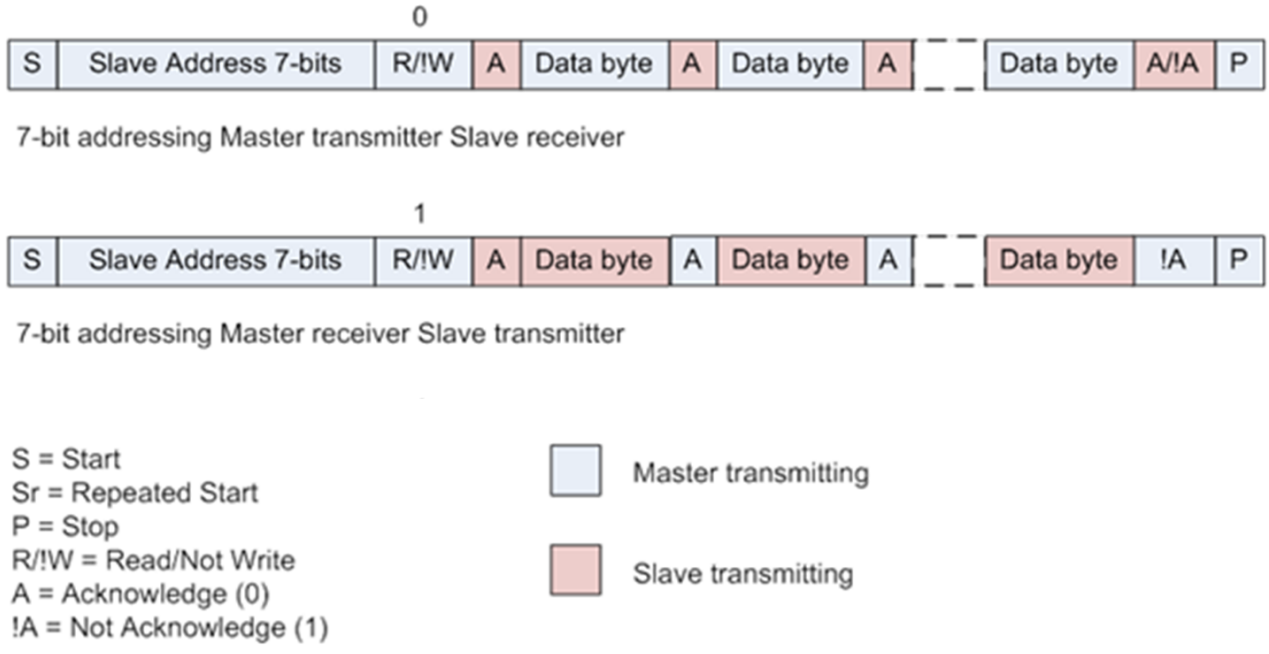


Figure 9.1 Data transfers using 7-bit addressing

The figure below summarises all of the transfers available (7-bit and 10-bit addressing).

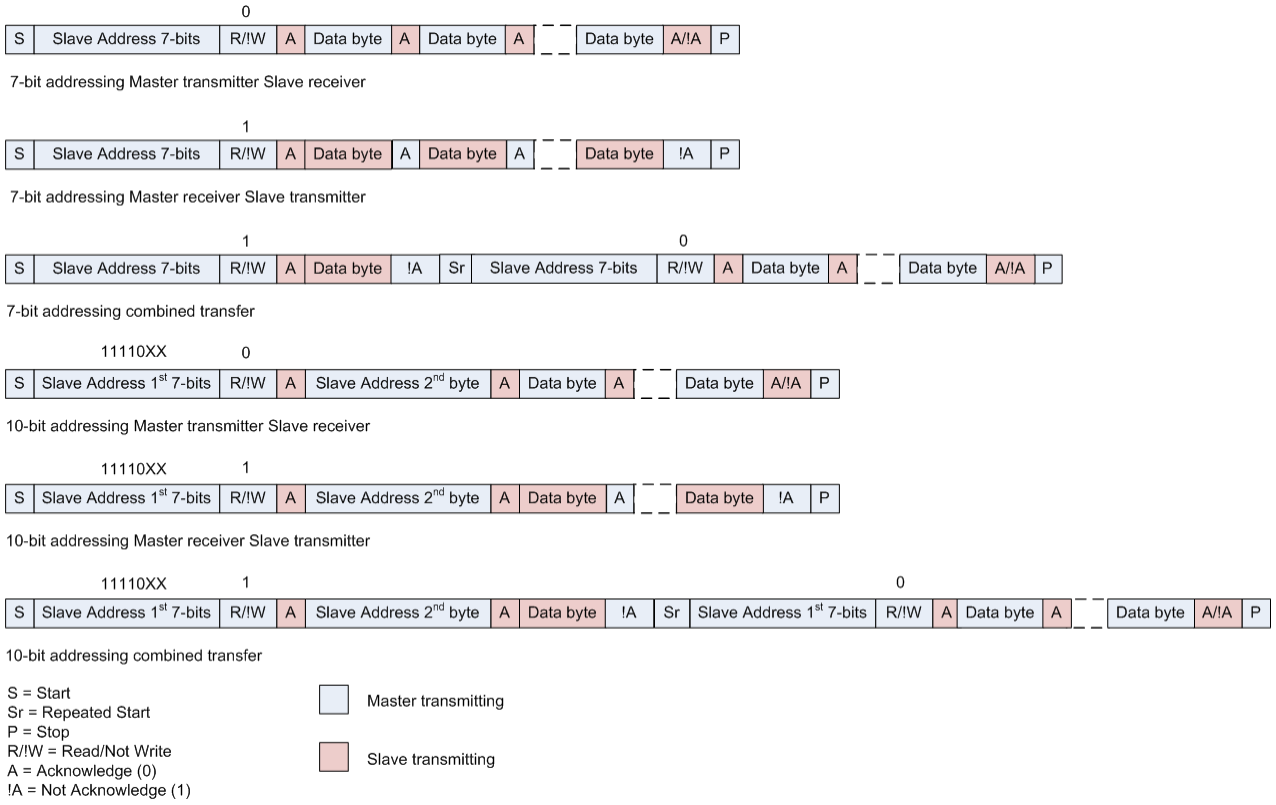


Figure 9.2 Data transfers with 7-bit and 10-bit addressing modes

Flow Control when Reading

When reading data, it is important to know if there is data available to read over I^2C and if the data read is valid. There are three methods by which to check whether the data being read is valid.

- Checking the I2C_TXE# line, which indicates whether the transmit buffer is empty. This signal can be mapped to the CBUS pins. This line does not indicate the number of bytes available. If the line is asserted, there are one or more bytes in the transmit buffer. Note that the Transmit buffer is the buffer which holds data which has come from the host computer and is going to be read by the external I^2C master.
- Do a read over I^2C (Master receiver, Slave transmitter) and check whether the FT-X acknowledges the address phase. The FT-X will NAK the address phase if there is no data to read.
- If bursting data, then there is not an address phase for each byte. In this case, it is not possible to tell when the buffer has emptied and therefore when you are no longer reading valid bytes. In this case, a Data Available check can be carried out first. If the byte after the general call address is 0x0C (0000 1100), the FT-X returns the Data Available Count to indicate the number of bytes available for burst read. Please also refer to section 9.3. Note that because this returns a single byte, the maximum value is 0xFF, and so a value of 0xFF indicates that there are 255 or more bytes available.



Figure 9.3 Checking the Data Available Count

Flow Control when Writing

When writing data (data going from I^2C toward the host computer), it is important to check if there is space available and if the FT-X accepted the byte written. There are two methods by which to check whether space is available.

- Check the I2C_RXF# line, which indicates whether the Receive buffer is full. This line indicates that there is at least one empty byte available in the buffer. However, there is no indication of how many bytes are available. This signal can be mapped to the CBUS pins. Note that the Receive buffer is the buffer which holds data which has come from the external I^2C master and is going to the host computer over USB.
- Do a write over I^2C (Master transmitter, Slave receiver) and check if the FT-X ACKs. The FT-X will NAK if there is no space in the buffer for the data to be written into. The FT-X will NACK in the address phase if this is the first byte being written and there is no space available. If the buffer becomes full during a burst write, the FT-X will NACK any bytes during the data phase which cannot be accommodated in the buffer.

Note that because the acknowledgement can be checked for each byte written (since the FT-X slave is generating the acknowledgement when writing to it); there is no command to ask the FT-X how much buffer space is available.

9.3 Other I^2C Commands

Soft reset

This uses the General Call Address. If the second byte is 0x06 (0000 0110) it will be interpreted by the FT-X as the Soft Reset command, which clears all buffers.

Flush command

This uses the General Call Address. If the second byte is 0x0E (0000 1110) it will be interpreted by the FT-X as the Flush command and the Transmit and Receive buffers will be flushed of all data.

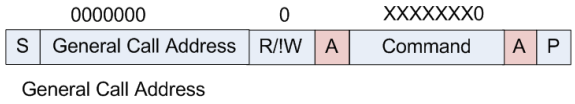


Figure 9.4 Flush data command

Read Data Available command

This uses the General Call Address. If the second byte is 0x0C (0000 1100) it will be interpreted by the block as the Read Data Available command and on the next I^2C read cycle following this command, the data will hold the number of bytes available for burst read.

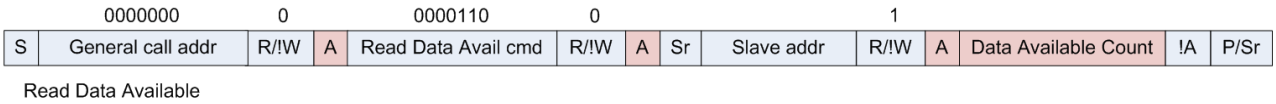


Figure 9.5 Reading the amount of data available

USB State command

This uses the General Call Address. If the second byte is 0x16 (0001 0110) it will be interpreted by the block as the USB State command and on the next read cycle following this command, the data will hold the coded USB State value. The USB state allows you to check the current USB enumeration state:

0x00 = Suspended.

0x01 = Default.

0x02 = Addressed.

0x03 = Configured.

Note: The Configured state is the normal operating state.

Further details of these states can be found in Chapter 9 of the USB 2.0 Specification from the USB Implementers Forum website (http://www.usb.org).

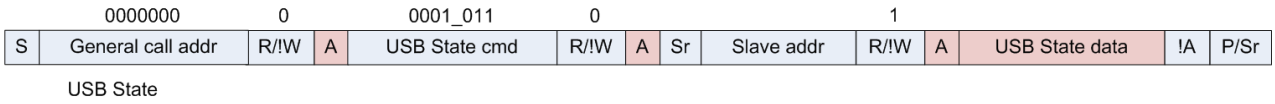


Figure 9.6 Checking the USB State

Device ID command

This command uses the Slave address of the FT-X. The master issues the 1111 100 0 command (0x7c write) followed by the slave address with the direction bit as either 0 or 1 (don't care). The master then sends a repeated start followed by the 1111 100 1 command (0x7c read). The FT-X responds with 3 bytes containing the manufacturer ID (12 bits), part ID (9 bits) and revision (3 bits). The master ends reading with a NACK.

Note: In the case where the master keeps ACKing the data, it has to wrap around and the slave (the FT-X) will keep sending the 3 bytes of its ID.

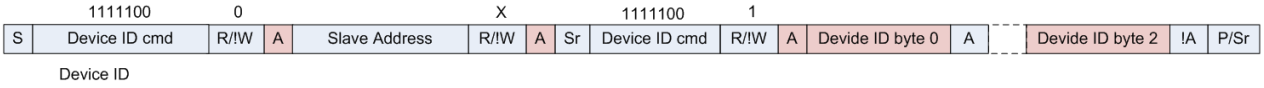


Figure 9.7 Reading the Device ID

10 Internal MTP Memory Configuration

The FT200XD includes an internal MTP memory which holds the USB configuration descriptors, other configuration data for the chip and also user data areas. Following a power-on reset or a USB reset the FT200XD will scan its internal MTP memory and read the USB configuration descriptors stored there

In many cases, the default values programmed into the MTP memory will be suitable and no re-programming will be necessary. The defaults can be found in Section 10.1.

The MTP memory in the FT200XD can be programmed over USB or over the I^2C bus if the values need to be changed for a particular application. Further details of this are provided from section 10.2 onwards.

Users who do not have their own USB Vendor ID but who would like to use a unique Product ID in their design can apply to FTDI for a free block of unique PIDs. See TN_100 – USB Vendor ID/Product ID Guidelines for more details.

10.1 Default Values

The default factory programmed values of the internal MTP memory are shown in the following table:

Parameter	Value	Notes
USB Vendor ID (VID)	0403h	FTDI default VID (hex)
USB Product UD (PID)	6015h	FTDI default PID (hex)
Serial Number Enabled	Yes	
Serial Number	See Note	A unique serial number is generated and programmed into the MTP memory during device final test.
Pull down I/O Pins in USB Suspend	Disabled	Enabling this option will make the device pull down on the I^2C and CBUS0 pins when in USB suspend mode (PWREN# is high).
Manufacturer Name	FTDI	
Product Description	FT200XD USB I2C	
Max Bus Power Current	90mA	
Power Source	Bus Powered	
Device Type	FT200XD	
USB Version	0200	Returns USB 2.0 device description to the host. Note: The device is a USB 2.0 Full Speed device (12Mb/s) as opposed to a USB 2.0 Hi-Speed device (480Mb/s).
Remote Wake Up	Disabled	
DBUS Drive Current Strength	4mA	Options are 4mA, 8mA, 12mA, 16mA
DBUS slew rate	Slow	Options are slow or fast

Copyright © Future Technology Devices International Limited

DBUS Schmitt Trigger Enable	Normal	Options are normal or Schmitt
CBUS Drive Current Strength	4mA	Options are 4mA, 8mA, 12mA, 16mA
CBUS slew rate	Slow	Options are slow or fast
CBUS Schmitt Trigger Enable	Normal	Options are normal or Schmitt
Load VCP Driver	Disabled	Enabling this will load the VCP driver interface for the device.
I^2C Address	22h	The I^2C device address
I^2C Device ID		The I^2C device ID
CBUS0	Keep_Awake#	Prevents the device from entering suspend state when unplugged. May be used if programming the MTP memory over I^2C

Table 10.1 Default Internal MTP memory Configuration

10.2 Methods of Programming the MTP Memory

10.2.1 Programming the MTP memory over USB

The MTP memory on all FT-X devices can be programmed over USB. This method is the same as for the EEPROM on other FTDI devices such as the FT232R. No additional hardware, connections or programming voltages are required. The device is simply connected to the host computer in the same way that it would be for normal applications, and the FT_Prog utility is used to set the required options and program the device.

The FT_Prog utility is provided free-of-charge from the FTDI website, and can be found at the link below. The user guide is also available at this link.

http://www.ftdichip.com/Support/Utilities.htm#FT_Prog

Additionally, D2XX commands can be used to program the MTP memory from within user applications. For more information on the commands available, please refer to D2XX Programmers Guide.

10.2.2 Programming the MTP memory over I^2C

In the FT200XD device, it is possible to program the MTP memory over its I^2C interface. This is the same interface which is used in the normal application of the FT200XD and would normally be connected to the I^2C master implemented in a microcontroller (MCU) or FPGA. However, special commands can also be used to access the MTP memory in the FT200XD over the same I^2C connection, allowing the MCU/FPGA to read and write locations in the MTP memory. No additional hardware, connections or programming voltages are required.

Warning: When programming locations in the MTP which are covered by the checksum, the checksum value must always be programmed to the correct value before any device reset takes place. If the device starts up or resets with an incorrect checksum, I^2C communication will not be possible. Recovery is only possible by re-programming the checksum to the correct value over USB.

Two examples where it may be desired to use the I^2C interface to write and read the MTP Memory are given below. In some cases, the application may use both of these possibilities.

Copyright © Future Technology Devices International Limited

1. To store and retrieve application specific data such as calibration constants in the user area (e.g. if the overall application was an analog measurement system). This can avoid the need for an extra EEPROM chip on the application board.
2. To read and write the configuration data (e.g. custom VID, PID, description strings or CBUS signal selection to enable signals for battery charging etc.) without a USB host. This could allow an MCU/FPGA to configure the FT200XD during production testing of the finished device or even when in use in the field.

The information in the rest of this chapter can be used to implement the storing and reading of application data in the user area as in example 1 above. Example 2 requires details of the configuration data stored in the MTP memory. Further details can be found in AN_201.

10.3 Memory Map

The FT-X family MTP memory has various areas which come under three main categories:

- User Memory Area
- Configuration Memory Area (writable)
- Configuration Memory Area (non-writable)

Memory Area Description	Word Address
User Memory Area 2 Accessible via USB, I²C and FT1248	0x3FF - 0x80
Configuration Memory Area Accessible via USB, I²C and FT1248	0x7E - 0x50
Configuration Memory Area Cannot be written	0x4E - 0x40
User Memory Area 1 Accessible via USB, I²C and FT1248	0x3E - 0x12
Configuration Memory Area Accessible via USB, I²C and FT1248	0x10 - 0x00

Figure 10.1 Simplified memory map for the FT-X

User Memory Area

The User Memory Areas are highlighted in Green on the memory map. They can be read and written via both USB and I²C. All locations within this range are freely programmable; no areas have special functions and there is no checksum for the user area.

Note that the application should take into account the specification for the number of write cycles in Section 6.4 if it will be writing to the MTP memory multiple times.

Copyright © Future Technology Devices International Limited

Configuration Memory Area (writable)

This area stores the configuration data for the device, including the data which is returned to the host in the configuration descriptors (e.g. the VID, PID and string descriptions) and also values which set the hardware configuration (the signal assigned to each CBUS pin for example).

These values can have a significant effect on the behaviour of the device. Steps must be taken to ensure that these locations are not written to un-intentionally by an application which is intended to access only the user area.

This area is included in a checksum which covers configuration areas of the memory, and so changing any value can also cause this checksum to fail. When changing any locations covered by the checksum, the checksum value must always be updated before a device reset or power cycle takes place (i.e. before the device uses the new MTP settings). Devices with incorrect checksum will require programming over USB to recover and correct the checksum.

Configuration Memory Area (non-writable)

This is a reserved area and the application should not write to this area of memory. Any attempt to write these locations will fail.

10.4 Hardware Requirements

The hardware is the same as for a typical USB-I^{2}C application and no additional hardware or programming voltages are required. The I^{2}C connections are the same as shown in Section 8.1. For the USB connections, either a bus-powered configuration (see Section 7.1 and 7.3) or a self-powered configuration (see Section 7.2) could be used.

Warning: When programming locations in the MTP which are covered by the checksum, the checksum value must always be programmed to the correct value before any device reset takes place. If the device starts up or resets with an incorrect checksum, I^{2}C communication will not be possible. Recovery is only possible by re-programming the checksum to the correct value over USB.

10.5 Protocol

The I^{2}C MTP memory protocol consists of 3 commands:

- Address MTP memory (0x10)
- Write MTP memory (0x12)
- Read MTP memory (0x14)

For further details on the I^{2}C protocol, refer to section 5.

10.5.1 Address MTP memory (0x10)

This consists of a general call with a command phase followed by 2 data bytes which represent the MTP memory address allowing users to address, potentially, up to 64K byte addresses.

10.5.2 Write MTP memory (0x12)

This consists of a general call with a command phase followed by 1 data byte which shall be programmed into the MTP memory at the address location set by the MTP memory address command.

10.5.3 Read MTP memory (0x14)

This consists of a general call with a command phase followed by 1 data byte which is the data read from the MTP memory at the address location set by the MTP memory address command.

Copyright © Future Technology Devices International Limited

10.5.4 Examples of Writing and Reading

When performing MTP memory write and read requests via the I^2C protocol, users must first issue the MTP address command along with 2 bytes representing the MTP memory address. The acknowledge phase of this command represents the current status of the MTP memory (whether it is busy or not). If the MTP memory is being accessed during an I^2C access then the respective command and data phases will NAK the master. The address will only be updated when the MTP memory is inactive.

Writing

The first part of the communication sets the address, and this is followed by the write command along with the data to be written. The MTP memory write itself will be initiated when the FT200XD receives an MTP memory write command followed by a single data byte. The ACK phase represents the current activity of the MTP memory (whether it is busy or not busy). A successful write will only occur when both status phases acknowledge the master indicating that the MTP memory can start the write. Users wishing to determine if the MTP memory write was successful should immediately try an MTP memory read to verify the new contents.

This process of writing to the MTP memory is shown below:

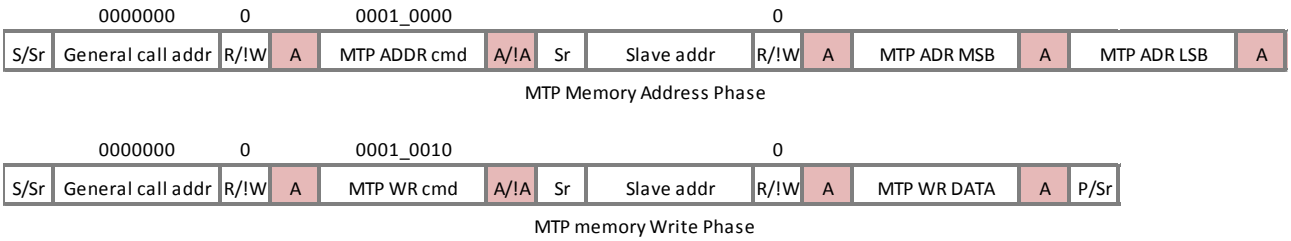


Figure 10.2 Write EEPROM command sequence

Reading

In a similar way, the read is carried out as shown below.The first part of the communication sets the address, and this is followed by the read command and a read cycle used to retrieve the data read back from that address.

An MTP memory read will be initiated when the I^2C slave receives an MTP read command. The acknowledgement during the command phase indicates whether or not the MTP memory will be able to service the read request. If it is possible to carry out the read, the I^2C moves into the data phase where it can either continue with the read transaction until the slave successful returns an ACK along with the data or if it receives NAK during the data phase it can abort the read transfer and try again.

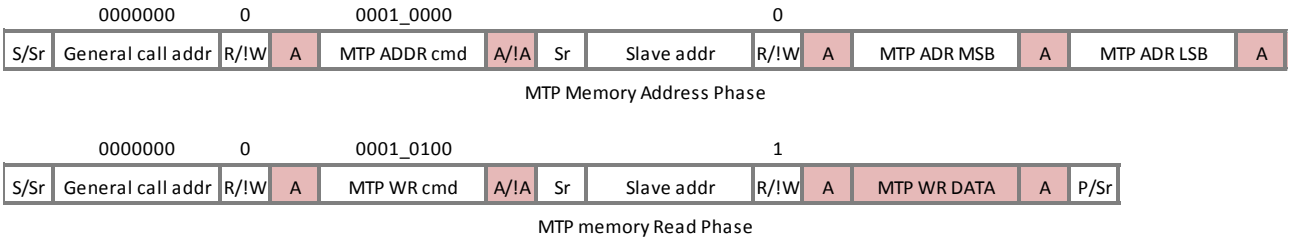


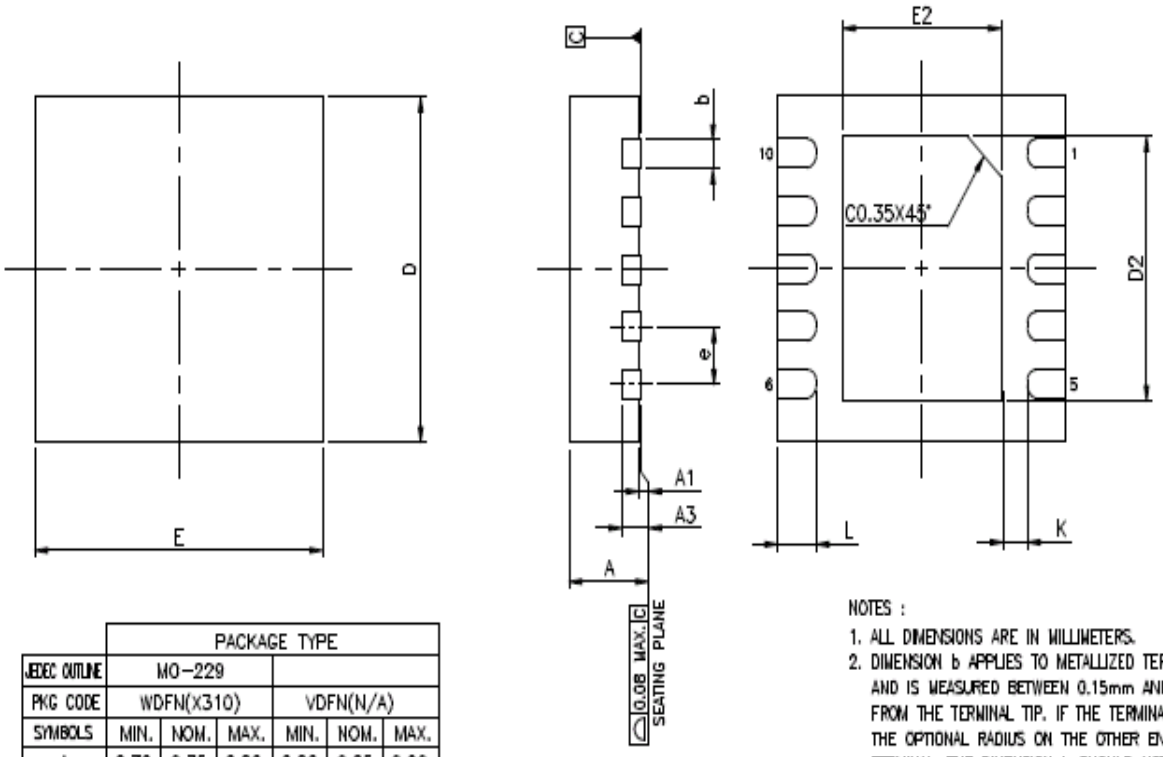
Figure 10.3 Read EEPROM command sequence

Copyright © Future Technology Devices International Limited

11 Package Parameters

The FT200XD is available in a 10 pin DFN package, the FT200XD.The solder reflow profile for the package is described in Section 0.

11.1 DFN-10 Package Mechanical Dimensions



	PACKAGE TYPE					
JEDEC OUTLINE	MO-229					
PKG CODE	WDFN(X310)			VDFN(N/A)		
SYMBOLS	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	0.70	0.75	0.80	0.80	0.85	0.90
A1	0.00	0.02	0.05	0.00	0.02	0.05
A3	0.20 REF.			0.20 REF.		
b	0.18	0.25	0.30	0.18	0.25	0.30
D	3.00 BSC			3.00 BSC		
E	3.00 BSC			3.00 BSC		
e	0.50 BSC			0.50 BSC		
L	0.30	0.40	0.50	0.30	0.40	0.50
K	0.20	—	—	0.20	—	—

PAD SIZE	E2			D2			LEAD FINISH		JEDEC CODE
E2XD2	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.	Pure Tin	PPF	
73X98 MIL	1.55	1.65	1.70	2.20	2.30	2.35	V	V	W30300-5

NOTES :
1. ALL DIMENSIONS ARE IN MILLIMETERS.
2. DIMENSION b APPLIES TO METALLIZED TERMINAL AND IS MEASURED BETWEEN 0.15mm AND 0.30mm FROM THE TERMINAL TIP. IF THE TERMINAL HAS THE OPTIONAL RADIUS ON THE OTHER END OF THE TERMINAL, THE DIMENSION b SHOULD NOT BE MEASURED IN THAT RADIUS AREA.
3. BILATERAL COPLANARITY ZONE APPLIES TO THE EXPOSED HEAT SINK SLUG AS WELL AS THE TERMINALS.

Figure 11.1 DFN-10 Package Dimensions

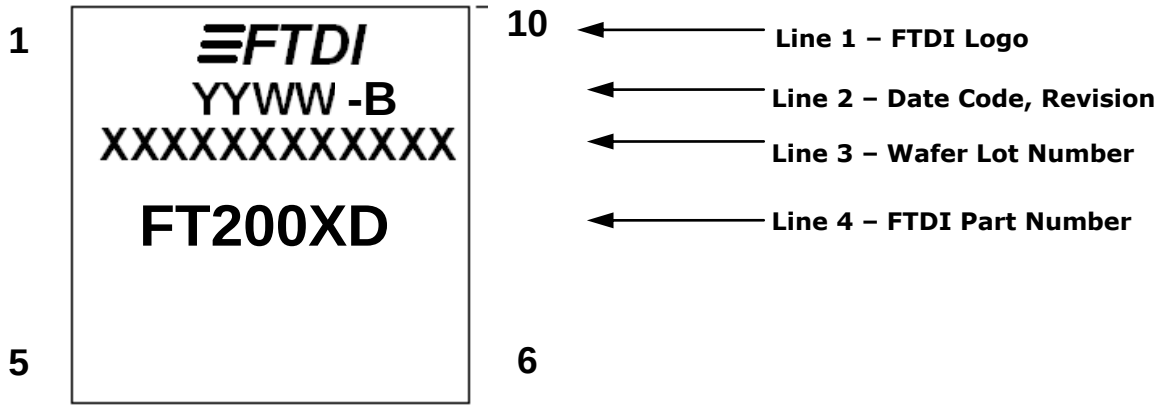
The FT200XD is supplied in a RoHS compliant leadless DFN-10package. The package is lead (Pb) free, and uses a 'green' compound. The package is fully compliant with European Union directive 2002/95/EC.

This package is nominally 3.00mm x 3.00mm. The solder pads are on a 0.50 mm pitch. The above mechanical drawing shows the DFN-10 package. All dimensions are in millimetres.

The centre pad on the base of the FT200XD is internally connected to GND. Do not place tracks on the top layer of the PCB in this area.

Copyright © Future Technology Devices International Limited

11.2 Package Markings



The date code format is **YYXWW** where XWW = 2 digit week number, YY = 2 digit year number.

The code **XXXXXXX** is the manufacturing LOT code.

Copyright © Future Technology Devices International Limited

11.3 Solder Reflow Profile

The FT200XD is supplied in Pb free DFN-10 packages. The recommended solder reflow profile for the package is shown in Figure 11.2.

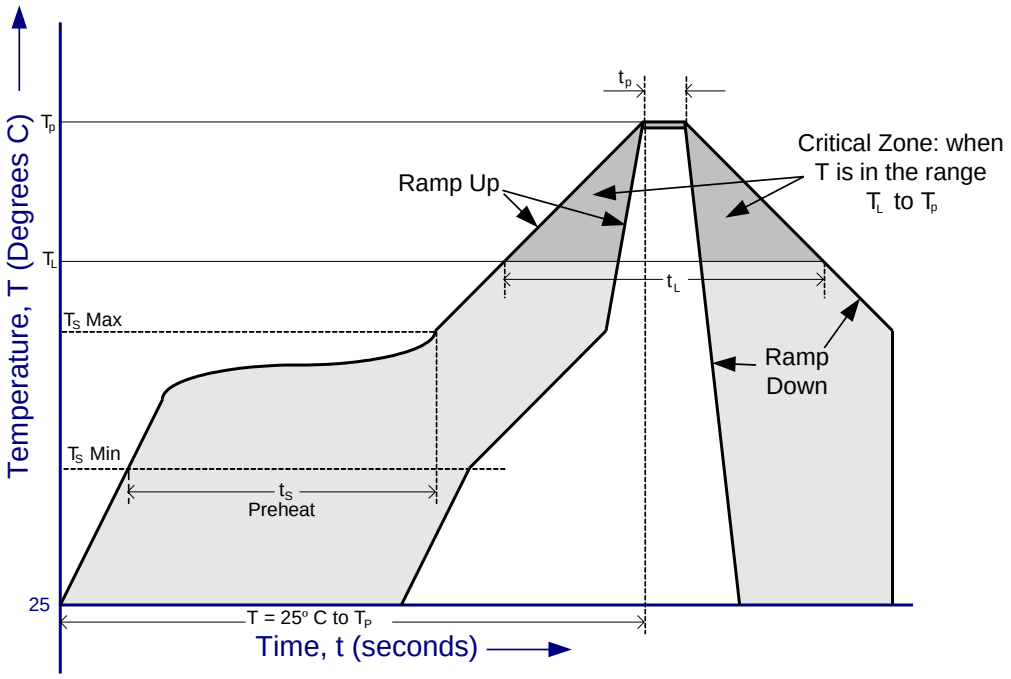


Figure 11.2 FT200XD Solder Reflow Profile

The recommended values for the solder reflow profile are detailed in Table 11.1. Values are shown for both a completely Pb free solder process (i.e. the FT200XD is used with Pb free solder), and for a non-Pb free solder process (i.e. the FT200XD is used with non-Pb free solder).

Profile Feature	Pb Free Solder Process	Non-Pb Free Solder Process
Average Ramp Up Rate (T_s to T_p)	3°C / second Max.	3°C / Second Max.
Preheat - Temperature Min (T_s Min.) - Temperature Max (T_s Max.) - Time (t_s Min to t_s Max)	 150°C 200°C 60 to 120 seconds	 100°C 150°C 60 to 120 seconds
Time Maintained Above Critical Temperature T_L: - Temperature (T_L) - Time (t_L)	 217°C 60 to 150 seconds	 183°C 60 to 150 seconds
Peak Temperature (T_p)	260°C	240°C
Time within 5°C of actual Peak Temperature (t_p)	20 to 40 seconds	20 to 40 seconds
Ramp Down Rate	6°C / second Max.	6°C / second Max.

Copyright © Future Technology Devices International Limited

Profile Feature	Pb Free Solder Process	Non-Pb Free Solder Process
Time for T= 25°C to Peak Temperature, T_p	8 minutes Max.	6 minutes Max.

Table 11.1 Reflow Profile Parameter Values

12 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

Branch Office – Tigard, Oregon, USA

Future Technology Devices International Limited (USA)
7130 SW Fir Loop
Tigard, OR 97223-8160
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

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 8791 3576

E-mail (Sales) tw.sales1@ftdichip.com
E-mail (Support) tw.support1@ftdichip.com
E-mail (General Enquiries) tw.admin1@ftdichip.com

Branch Office – Shanghai, China

Future Technology Devices International Limited (China)
Room 1103, No. 666 West Huaihai Road,
Shanghai, 200052
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

http://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.

System and equipment manufacturers and designers are responsible to ensure that their systems, and any Future Technology Devices International Ltd (FTDI) devices incorporated in their systems, meet all applicable safety, regulatory and system-level performance requirements. All application-related information in this document (including application descriptions, suggested FTDI devices and other materials) is provided for reference only. While FTDI has taken care to assure it is accurate, this information is subject to customer confirmation, and FTDI disclaims all liability for system designs and for any applications assistance provided by FTDI. Use of FTDI devices in life support and/or safety applications is entirely at the user's risk, and the user agrees to defend, indemnify and hold harmless FTDI from any and all damages, claims, suits or expense resulting from such use. This document is subject to change without notice. No freedom to use patents or other intellectual property rights is implied by the publication of this document. Neither the whole nor any part of the information contained in, or the product described in this document, may be adapted or reproduced in any material or electronic form without the prior written consent of the copyright holder. Future Technology Devices International Ltd, Unit 1, 2 Seaward Place, Centurion Business Park,

Appendix A – References

Document References

Useful Application Notes

AN232R-01 FT232R BitBangModes

AN_107 Advanced Driver Options

AN_121 Accessing the EEPROM User Area of FTDI Devices

AN_175 Battery Charger Detection over USB with FT-X Devices

TN_100 USB Vendor ID/ Product ID Guidelines

http://i2c2p.twibright.com/spec/i2c.pdf

http://www.usb.org/developers/

Acronyms and Abbreviations

Terms	Description
EHCI	Enhanced Host Controller Interface
FPGA	Field Programmable Gate Array
I2C	Inter-Integrated Circuit
LDO	Low Drop Out
LED	Light Emitting Diode
MCU	Micro Controller Unit
MTP	Multi-time Programmable
OHCI	Open Host Controller Interface
PLD	Programmable Logic Device
QFN	Quad Flat No-Lead
RoHS	Restriction of Hazardous Substances Directive
USB	Universal Serial Bus
UHCI	Universal Host Controller Interface
VCP	Virtual COM Port

Copyright © Future Technology Devices International Limited

Appendix B - List of Figures and Tables

List of Figures

Figure 2.1 FT200XD Block Diagram 4
Figure 3.1 DFN Schematic Symbol 7
Figure 7.1 Bus Powered Configuration 19
Figure 7.2 Self Powered Configuration 20
Figure 7.3 Bus Powered with Power Switching Configuration 21
Figure 8.1 Application Example showing USB to I^2C Converter 22
Figure 8.2 USB Battery Charging Detection 24
Figure 9.1 Data transfers using 7-bit addressing 27
Figure 9.2 Data transfers with 7-bit and 10-bit addressing modes 28
Figure 9.3 Checking the Data Available Count 28
Figure 9.4 Flush data command 29
Figure 9.5 Reading the amount of data available 29
Figure 9.6 Checking the USB State 30
Figure 9.7 Reading the Device ID 30
Figure 10.1 Simplified memory map for the FT-X 33
Figure 10.2 Write EEPROM command sequence 35
Figure 10.3 Read EEPROM command sequence 35
Figure 11.1 DFN-10 Package Dimensions 36
Figure 11.2 FT200XD Solder Reflow Profile 38

List of Tables

Table 1.1 Ordering part numbers 2
Table 3.1 Power and Ground 7
Table 3.2 Common Function pins 7
Table 3.3 I^2C Interface and CBUS Group (see note 1) 8
Table 3.4 CBUS Configuration Control 9
Table 6.1 Absolute Maximum Ratings 13
Table 6.2 ESD and Latch-Up Specifications 13
Table 6.3 Operating Voltage and Current 14
Table 6.4 I/O Pin Characteristics VCCIO = +3.3V (except USB PHY pins) 15
Table 6.5 I/O Pin Characteristics VCCIO = +2.5V (except USB PHY pins) 16
Table 6.6 I/O Pin Characteristics VCCIO = +1.8V (except USB PHY pins) 17
Table 6.7 USB I/O Pin (USBDP, USBDM) Characteristics 18
Table 6.8 MTP Memory Characteristics 18
Table 6.9 Internal Clock Characteristics 18
Table 10.1 Default Internal MTP memory Configuration 32

Copyright © Future Technology Devices International Limited

Table 11.1 Reflow Profile Parameter Values..... 39

Copyright © Future Technology Devices International Limited

Appendix C - Revision History

Document Title: USB I2C SLAVE IC FT200XD
Document Reference No.: FT_000628
Clearance No.: FTDI# 265
Product Page: http://www.ftdichip.com/Products/ICs/FTEX.htm
Document Feedback: Send Feedback

Revision	Changes	Date
1.0	Initial Release	2012-02-08
1.1	Added USB compliance in section 1.3; Clarified MTP reliability in table 6.8; Edited Table 9.1: Added I2C address and changed Load VCP Driver to Disabled	2012-04-17
1.2	Added section 9 with details of USB and I2C Interfacing; Added link to AN_201 (MTP Memory Configuration) in section 10; Removed references to LED signals on the CBUS pins as these are only available on the UART members of the FT-X family; Removed section 8.3 showing connection of the Tx/Rx LEDs.	2012-08-15
1.3	Added 5V tolerant statement on front page; Updated TID info; Updated USA address	2013-02-10
1.4	Updated section 9.3 to correct description of soft reset; Added note to MTP section 10 regarding checksum when programming over I2C	2017-05-12

Copyright © Future Technology Devices International Limited