USB Programming and Circuit Guide

January 2011 Application Note AN8082

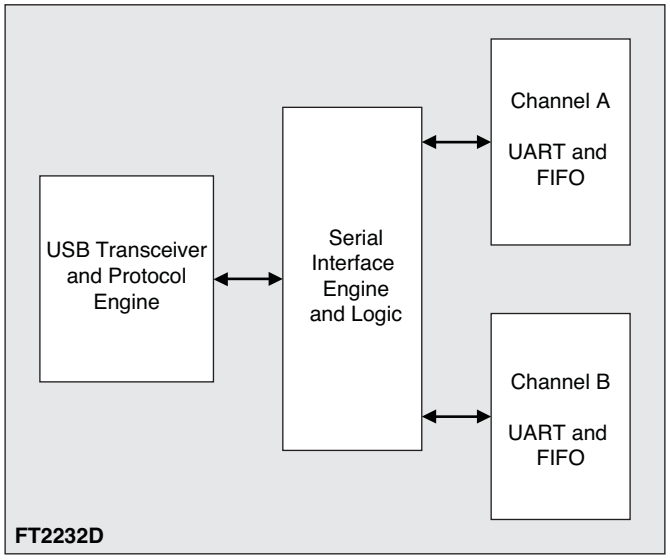
Introduction

For many years, the USB devices from Cypress Semiconductor have been the “go to” devices for well documented and easy-to-implement USB interfaces. In fact, the Lattice USB download cable and several evaluation boards incorporate the CY7C68013A as the USB-to-JTAG interface. This device provides the JTAG interface from custom software that is executed by the embedded 8051 microprocessor core. The custom code is downloaded at enumeration (plugging the USB cable into a computer) which provides a very flexible solution. Recently introduced devices from Future Technology Devices International Ltd. (FTDI) provide a viable alternate solution to the USB interface on the basis of size, cost, features, and ease of use. This application note describes the features of the FT2232 with regard to programming and interfacing with Lattice Semiconductor’s programmable devices.

The FTDI FT2232D Device

Figure 1 illustrates a top-level block diagram of the FT2232D device highlighting the USB interface, the serial control and logic, and the two UART channels. The USB transceiver simply connects to the USB cable and transfers the data to and from the serial interface engine. The serial interface engine manages the data flow to and from the two UART channels A and B. The two interface channels are not exactly the same. Channel A can be configured as a Multi-Protocol Synchronous Serial Engine (MPSSE) which supports JTAG, I^2C, and several other protocols, as well as a standard UART. Channel B does not support the MPSSE but does support a second independent standard UART interface. Thus, Channel A should be wired to the JTAG and/or I^2C signals to take advantage of the built-in support and Channel B should be wired to the UART interface, if used. In the example schematic provided in Appendix B, an external MUX is used to switch the signals from Channel A to either the JTAG circuits or the I^2C bus.

Figure 1. FTDI FT2232D Block Diagram



Installing the FTDI Drivers

When a board with the FT2232D device is first connected to a computer running the Windows operating system, The Found New Hardware Wizard will start and prompt the user for the drivers. It is best to have the drivers already downloaded and saved in a local folder for easy access. The drivers for the FT2232D are compressed in a file named “CDM 2.04.16 WHQL Certified.zip” and can be downloaded from www.ftdichip.com.

© 2011 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal. All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

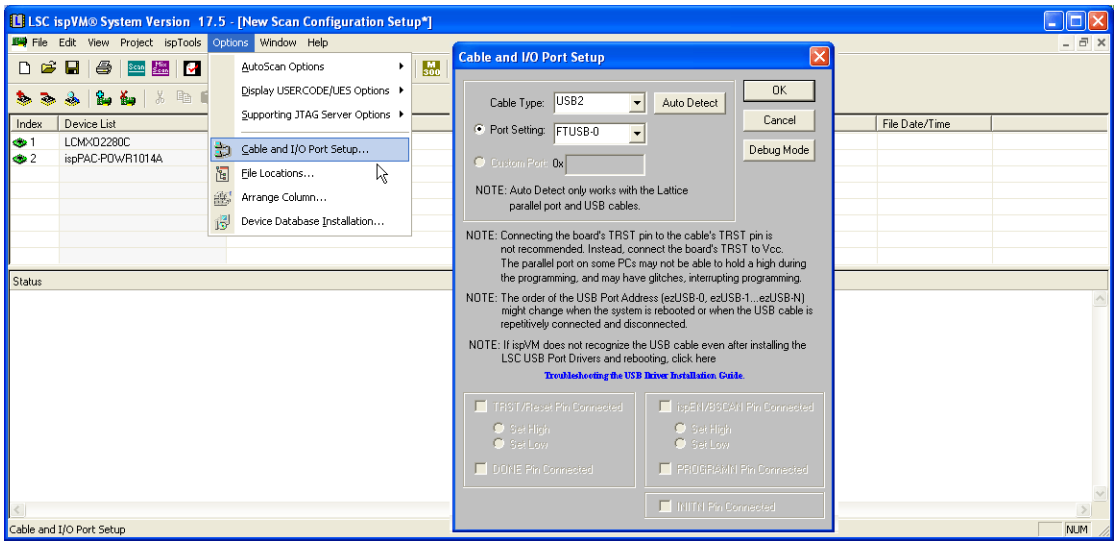
Configuring the FT2232D Device with FT_Prog

When the FT2232D is powered up, it reads configuration information from a serial memory. In the schematic shown in the appendix this serial memory is U2. This memory holds information regarding the USB parameters such as self powered or bus powered. If bus powered, how much power is needed. The memory also holds ID codes and serial numbers that may be used by the operating system at enumeration and by the software that communicates with the FTDI device. FTDI provides an easy-to-use software utility to preload the serial memory called FT_Prog which can be downloaded from www.ftdichip.com/support/utilities.htm. For additional information regarding the contents of the memory, please refer to the FTDI documentation.

Programming JTAG Devices Using ispVM™

Lattice Semiconductor's universal JTAG programming software ispVM supports the FT2232D device to program individual devices or a chain of devices from either JED or SVF source files. Figure 2 shows the menu option and cable settings for ispVM so that the devices connected to the FT2232D can be scanned and programmed.

Figure 2. Cable and I/O Port Settings in ispVM to Support FT2232D



Using the I²C and UART features of the FT2232D

The FT2232D device is not only a convenient solution to programming JTAG devices over USB but, it also provides development and debugging options as well. The schematic in Appendix B, shows an analog MUX (U3) which is used to divert the signals from channel A to either a JTAG chain or an I²C bus. Pin 16 of U1 can then be controlled by the host software to switch between the JTAG chain and the I²C bus. FTDI provides the drivers and example files of how to configure the FT2232D to interface with the I²C bus.

The ispMACH® 4000ZE Pico Evaluation Board is wired with a MUX attached to Channel A for both JTAG and I²C support. The Pico View utility from Lattice is built using the FTDI examples and drives pin 16 of U1 to switch the MUX from JTAG to I²C. The I²C bus on the Pico Evaluation Board is connected to both the ispMACH 4000ZE and the ispPAC®-POWR6AT6 devices. The Pico View utility provides an intuitive interface to read and write to the registers of the devices.

When developing embedded microprocessor code, it is useful to have a serial port to debug and control the design. Channel B provides that path when connected to a UART port of the microprocessor. At the host computer side, FTDI provides the drivers to emulate a virtual COM port over USB so that applications like HyperTerminal can communicate with the microprocessor.

The MachXO™ Control Evaluation Board is wired in this manner to provide programming and debugging support over one simple USB cable. Channel A provides the JTAG path to reprogram the MachXO while developing

LatticeMico8™ embedded microcontroller applications (see RD1026, LatticeMico8 Microcontroller User's Guide). Channel B is wired to MachXO pins used by the embedded UART (see RD1042, WISHBONE UART Reference Design) that is connected to the LatticeMico8 on the internal WISHBONE bus (see RD1043, LatticeMico8 to WISHBONE Interface Adapter Reference Design). Thus, HyperTerminal can be used as a user interface or debugging tool to display temporary or internal values.

Parts Placement, Routing and Miscellaneous

The signals USB+ and USB- are a high-speed differential pair and need to be routed as such between J1 and U1. Using equidistant paths and minimal vias will tend to preserve signal integrity. The distance from J1 and U1 should be kept relatively short (25mm to 50 mm). The crystal, X1, should be very close to U1 and should have only a ground plan below it – no high-speed data, clock or address lines. Furthermore, the crystal should have a guard ring around it that is tied to ground.

Mixed VCCJ JTAG chains should be designed with care as the I/O of the FT2232D is fixed at 3.3V. So the last device in the chain should have a VCCJ of 3.3V to meet the input levels of the FT2232D chip.

Summary

On-board USB circuitry simplifies programming support and enables sophisticated debugging through either the virtual RS-232 com port or the I^2C DLL. The circuit provided in Appendix B can be cut and pasted as a proven working design to reduce design time and risk. Contact Lattice Technical Support for a copy of the ORCAD design files. Programming support is provided by ispVM from Lattice and drivers and examples are provided by FTDI.

Related Lattice Semiconductor Literature

- DS1002, MachXO Family Data Sheet
- DS1022, ispMACH 4000ZE Family Data Sheet
- AN8080, Using a Discrete Crystal as a PLD Clock Source
- RD1026, LatticeMico8 Microcontroller User's Guide
- RD1042, WISHBONE UART Reference Design
- RD1043, LatticeMico8 to WISHBONE Interface Adapter Reference Design

References

- Future Technology Devices International, FT2232D Data Sheet, www.ftdichip.com
- Future Technology Devices International, application note AN_108, Command Processor for MPSSE and MCU Host Bus Emulation Modes
- Future Technology Devices International, application note AN_109, Programming Guide for High Speed FTCI2C DLL
- Future Technology Devices International, Software Application Development D2XX Programmer's Guide
- Citizen Crystal, HCM49 Data Sheet (used on the MachXO Mini Evaluation Board), www.citizencrystal.com/images/pdf/m-hcm49.pdf
- Citizen Crystal, Precautions in Oscillation Circuit Design technical note, www.citizencrystal.com/images/pdf/tech-09.pdf
- Citizen Crystal, Reference for Selecting Constants of Oscillation Circuit technical note, www.citizencrystal.com/images/pdf/tech-10.pdf

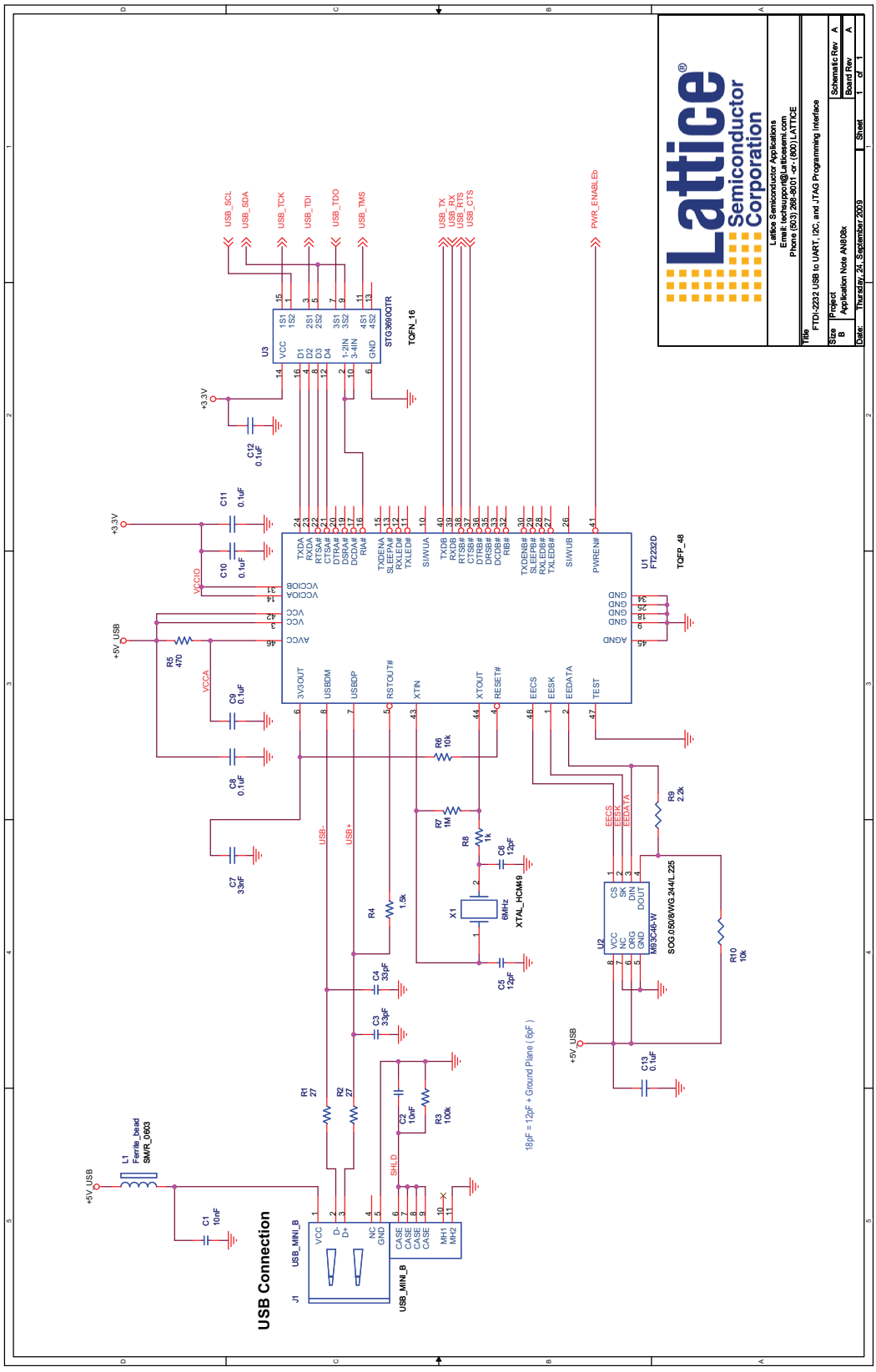
Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
October 2009	01.0	Initial release.
January 2011	01.1	Removed references to outdated programming utility.

Appendix A. USB Circuit Bill of Materials

Quantity	Reference Designator	Description	Manufacturer	Part Number
1	J1	USB Mini Connector	Hirose	UX60-MB-5ST
1	U1	USB Interface Chip	FTDI	FT2232D
1	U2	Serial Memory	STMicroelectronics	M93C46_WMN6TP
1	U3	Analog MUX	STMicroelectronics	STG3690QTR
1	X1	6 MHz Crystal	Citizen America	HCM49 6.000MABJ-UT
2	R1,R2	Resistor 27 ohm 5%	Any	Any
1	R3	Resistor 100k ohm 5%	Any	Any
1	R4	Resistor 1.5k ohm 5%	Any	Any
1	R5	Resistor 470 ohm 5%	Any	Any
2	R6,R10	Resistor 10k 5%	Any	Any
1	R7	Resistor 1M ohm 5%	Any	Any
1	R8	Resistor 1k ohm 5%	Any	Any
1	R9	Resistor 2.2k ohm 5%	Any	Any
2	C1,C2	Capacitor 10nF	Any	Any
2	C3,C4	Capacitor 33pF	Any	Any
2	C5,C6	Capacitor 12pF	Any	Any
1	C7	Capacitor 33nF	Any	Any
6	C8-C13	Capacitor 0.1uF	Any	Any
1	L1	Ferrite Bead	Steward	MI0603J600R-00

Appendix B. USB Circuit Schematic