Future Technology Devices International Ltd.

Application Note

AN_165 Establishing Synchronous 245 FIFO Communications using a Morph-IC-II

Document Reference No.: FT_000387
Version 1.1

Issue Date: 2012-06-26

The Morph-IC-II module is an FPGA-USB development platform that supports a number of serial communications interfaces including Synchronous 245 FIFO. A number of supporting source code samples have been provided by FTDI to assist in the development of Morph-IC-II applications. One of these examples is a Synchronous 245 FIFO application. This HDL application illustrates how to establish communications between the on board FT2232H of the Morph-IC-II and another synchronous 245 FIFO slave device.

The FTDI UM232H is a module that can support synchronous 245 FIFO.

This application note illustrates how to establish Synchronous 245 communications between a UM232H and a Morph-IC-II module.

Future Technology Devices International Limited (FTDI)

Unit 1, 2 Seaward Place, Glasgow G41 1HH, United Kingdom
Tel.: +44 (0) 141 429 2777 Fax: + 44 (0) 141 429 2758
E-Mail (Support): **support1@ftdichip.com** Web: http://www.ftdichip.com

Table of Contents

1 Introduction

The aim of this application note is to illustrate the synchronous 245 FIFO capabilities of the Morph-IC-II and UM232H, by providing a synchronous 245 FIFO application, including source code, and giving step by step instructions on how to verify the application.

The following equipment used in this application :

- 1 x FTDI Morph-IC-II – A USB-FPGA development module
- 1 x FTDI UM232H – A Hi-Speed USB to Serial/FIFO Module
- A proto-typing setup and USB cables
- Synchronous 245 application project files – The HDL collateral supplied for this application

  can be downloaded from here:

  http://www.ftdichip.com/Products/Files/Synchronous 245 Morph-IC-II Application.zip
- FT_Prog – A Programming Utility for FTDI devices
- MorphLd-II – Programming utility for the Morph-IC-II
- Quartus-II – A HDL Tool-Chain for Altera FPGAs
- Terminal.x – An FTDI utility for transferring data over different interfaces

Note: All sample code and utilities provided in this note are for illustration purposes and are not guaranteed or supported.

On completion of reading this app note the reader should be able to:

- Configure a FT2232H device for synchronous 245 FIFO mode
- Configure a FT232H device for synchronous 245 FIFO mode
- Program an Altera based FPGA to host the synchronous 245 FIFO devices

1.1 What is a UM232H?

The UM232H is an evaluation module containing the FT232H chip. This module provides access to the serial/FIFO data channel. This module may be used to convert one USB port to either: UART, Synchronous 245 FIFO, Asynchronous 245 FIFO, FT1248 or MPSSE.

For more information on the modules please see:

UM232H Datasheet

http://www.ftdichip.com/Support/Documents/DataSheets/Modules/DS_UM232H.pdf

FT232H Datasheet

http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232H.pdf

1.2 What is Morph-IC-II?

Morph-IC-II is an FTDI low cost USB-FPGA development platform. The major components of this module are the FTDI FT2232H and an Altera Cyclone II FPGA.

The FT2232H is a dual channel USB communications device which converts USB data into a range of different interfaces including UART, Synchronous 245 FIFO, Asynchronous 245 FIFO and more. The FT2232H provides one programing channel for the FPGA (passive serial) and one application data channel to access data after configuration of the FPGA. Passive serial is an interface widely used by Altera FPGAs for programming and configuration. This interface is supported by the FT2232H's MPSSE (Multi-Protocol Synchronous Serial Engine).

For additional information please refer to the following documentation:

Morph-IC-II Datasheet

http://www.ftdichip.com/Support/Documents/DataSheets/Modules/DS_Morph-IC-II.pdf

MorphLd and MorphIO-II Utilities for Morph-IC-II

http://www.ftdichip.com/Support/Documents/AppNotes/AN_141_MorphIO-II%20and%20MorphLd%20Utilities%20for%20Morph-IC-II.pdf

1.3 What is Synchronous 245 FIFO?

Synchronous 245 FIFO is a half-duplex point-to-point communications interface. This interface is synchronised to transmit data at a fixed clock rate of 60MHz, and can support data flow rates up to 35MByte per second. Synchronous 245 FIFO contains all the signals used by Asynchronous 245 FIFO plus an additional 2 lines: clock out which is a 60MHz clock signal and output enable used to enable the outputs of a slave device.

Synchronous 245 FIFO can transfer data at much higher data rates than Asynchronous 245 FIFO. Synchronous 245 FIFO requires the master and the slave devices to be synchronised to the same 60MHz clock. Using this application note and the supporting hardware and application files, establishing a successful Synchronous 245 communication link can be made easy.

For additional information please see:

AN_130 FT2232H Used In An FT245 Style Synchronous FIFO Mode

http://www.ftdichip.com/Support/Documents/AppNotes/AN_130_FT2232H_Used_In_FT245%20Synchronous%20FIFO%20Mode.pdf

DS_FT2232H

http://www.ftdichip.com/Support/Documents/DataSheets/Modules/DS_FT2232H_Mini_Module.pdf

Located at www.ftdichip.com

2 Using and Understanding Synchronous 245 FIFO

2.1 Syncronous 245 FIFO Signal Flow

Synchronous 245 FIFO mode can be used to transfer data from two points: master and slave. The slave synchronous 245 system generates the 60MHz clock signal synchronised to a USB interface. This clock signal is used to synchronise the data sample rate and phase of the data being sent.

The slave device indicates when it is prepared for beginning a read or a write process by setting the levels of the status lines RXF# and TXE#. A logic low RXF# indicates that the slave has data that can be transferred, the master can respond to this by setting OE# (output enable) to logic low which changes the direction of the slave's data port to be in output mode. The slave's data port is in input mode most of the time, but when the master requires data, the port of slave can be set to output mode. Once this data port is in output mode it only then becomes reasonable for the master to demand data from the slave. A logic low TXE# indicates that the slave device will allow for data to be written to its registers. The master can respond to this indication by initiating the transfer of data from the master to the slave device.

The master controls the two strobe commands RD# and WR#. A logic low RD# will command that on the next rising clock edge; the master samples the transferring data and the slave to begin transferring the next byte of data. A logic low WR# commands that on the next rising clock edge; the slave samples the transferring data and the master to begin to transfer the next byte of data. The port direction of both master and slave synchronous 245 interfaces are shown in Figure 1. A signal plot of a read operation is given in Figure 2 and a write operation is given in Figure 3.

Synchronous 245 FIFO contains a SI/WU (Send Immediate/WakeUp) signal which combines two functions. If the FT2232H USB is in suspend mode and remote wakeup is enabled in the EEPROM, strobing this line low will cause the device to request a resume on the USB BUS. Normally, this can be used to wake up the Host PC. When the FT2232H device is not in suspend mode, if the SI/WU line is strobed low any data in the device TX buffer will be sent out over USB on the next Bulk-IN request from the USB host regardless of the pending packet size. This can be used to optimise USB transfer speed for some applications.

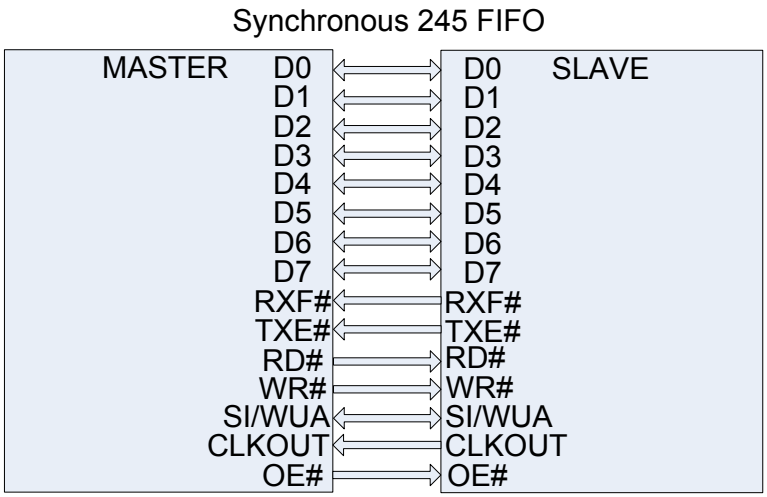
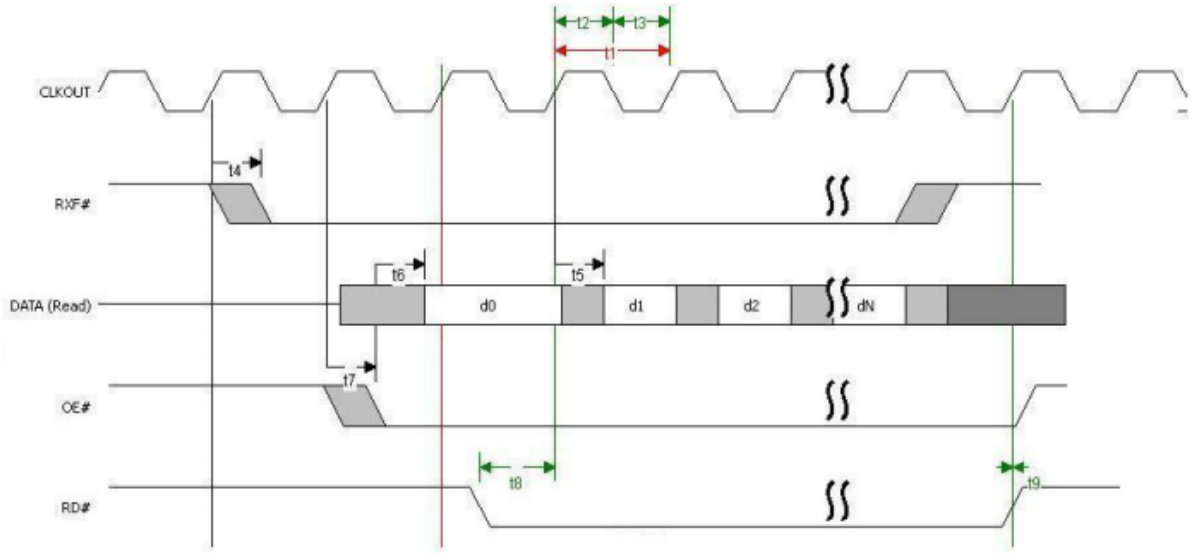
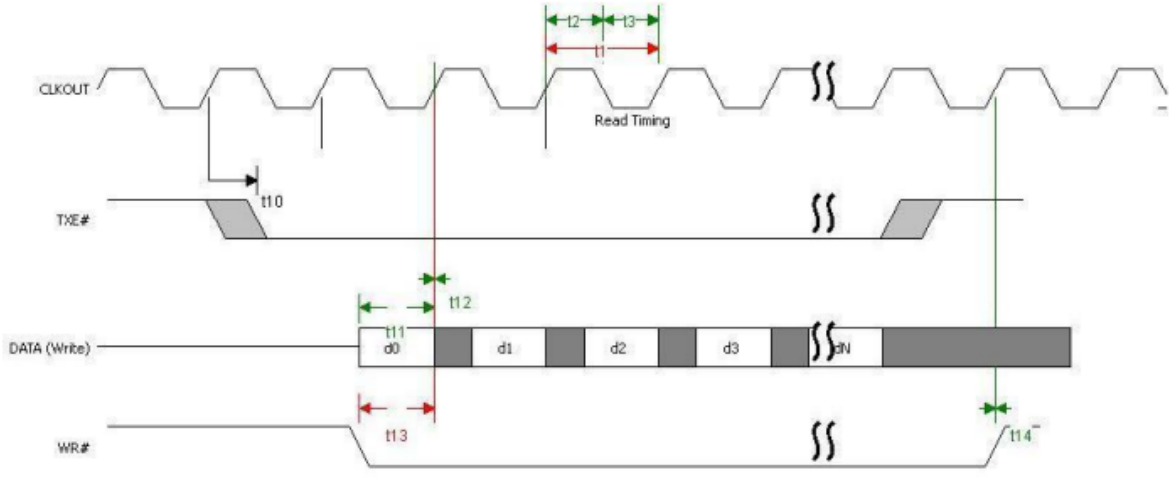


Figure 1 – Synchronous 245 FIFO Data Flow Directions



Read Timing (External system read data from FTDI chip)

Figure 2 – Read Timing



Write Timing (External system write data into FTDI chip)

Figure 3 – Write Timing

Time	Description	Min	Max	Units
T1	RD# inactive to RXF#	1	14	ns
T2	RXF# inactive after RD# cycle	49		ns
T3	RD# to DATA	1	14	ns
T4	RD# active pulse width	30		ns
T5	RD# active after RXF#	0		ns
T6	WR# active to TXE# inactive	1	14	ns
T7	TXE# active to TXE# after WR# cycle	49		ns
T8	DATA to WR# active setup time	5		ns
T9	DATA hold time after WR# inactive	5		ns
T10	WR# active pulse width	30		ns
T11	WR# active after TXE#	0		ns

Table 2.1 Asynchronous FIFO Timings (based on standard drive level outputs)

2.2 Synchronous 245 between Two Slave Devices via a Master

Direct communications between two slave Synchronous 245 devices is not possible without the introduction of an intermediary master device to control both systems. The FPGA of the Morph-IC-II provides this master function as noted in Figure 4.

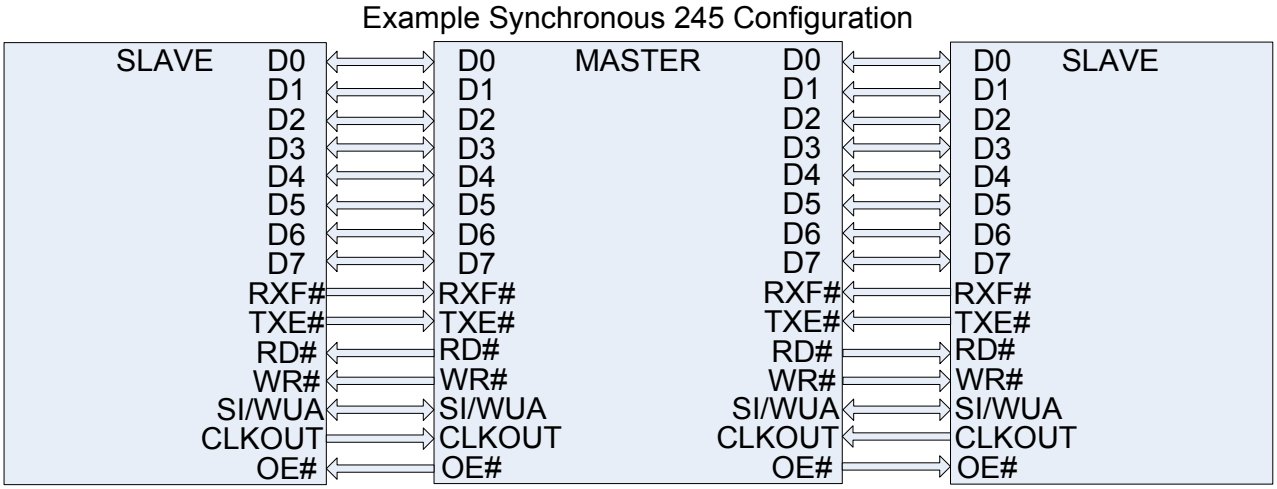


Figure 4 – Synchronous 245 Interface Block Diagram

3 Example Synchronous 245 FIFO Application

Included in the Morph-IC-II download is a Quartus-II Archive File labelled "MorphIC_HS_245_Sync_fifo.qar". Contained in this file are a collection of RTL files used to synthesize a master synchronous 245 device that controls communication between two synchronous 245 slave devices. Also contained in this archive folder is a Quartus-II project folder that is ready to compile. This project file configures all the device settings, the pin-map and calls for a RBF file containing the entire project to be outputted after the project compiles. This RBF file can be loaded to the Morph-IC-II to synthesize the Synchronous 245 FIFO application hardware in the FPGA.

Links for all the necessary utilities and applications are given in Appendix B.

3.1 An Outline of the Synchronous 245 Application

An outline of the Synchronous 245 application is illustrated in Figure 5. This diagram illustrates the components used and their functions.

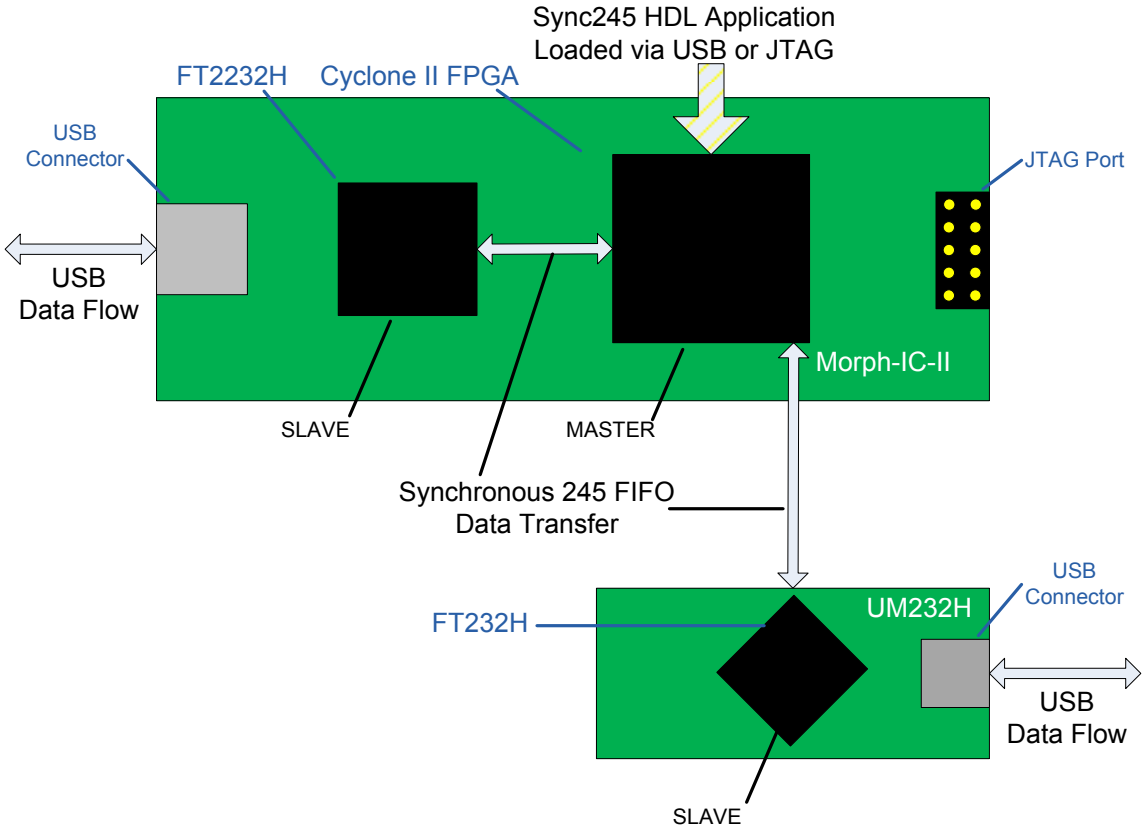


Figure 5 – A Block Diagram of the Synchronous 245 FIFO Application

In this example, a synchronous 245 application RBF file is first loaded to the FPGA via USB or JTAG, this application HDL creates 2 master Synchronous 245 devices. When the master device is synthesised, synchronous 245 data can be transferred from one slave device to another slave device via the master.

In this application USB data can be transmitted to either of the FT2232H or FT232H chips. The USB data is then translated to the synchronous 245 interface. In order to transfer the synchronous 245 data, communications between the synchronous 245 master device and the transmitting slave device the master device synchronises to the 60MHz clkout signal output by the transmitting slave device. When the master has synchronised to clkout, it can then read the status lines of the slave and respond by setting the control lines to allow for the transfer of data from the slave to the master.

In a similar manner the master device can transfer synchronous 245 data to the other slave device. Here the master is synchronised to clkout of the receiving slave device, it sets the control lines and reads status lines then begins the transfer of data.

3.2 RTL Code

The following code sample lists the ports of the Synchronous 245 Interface application which is available in the "Morph-IC-II Application and Utilities" download. These ports include a reset line, the synchronous 245 FIFO data interface of the on board FT2232H of the Morph-IC-II and the synchronous 245 FIFO data interface of UM232H.

```
entity morphic_hs_245_sync_fifo is
   generic ( loopback_to_hsext : integer := 0 );
   port (
-- Inputs
      rst       : in  std_logic;

-- Morphic on board FT2232H signals
      mdata     : inout std_logic_vector(7 downto 0);   -- Port A Data Bus
      mclk60    : in std_logic;
      mrxfn     : in std_logic;
      mtxen     : in std_logic;
      mrdn      : out std_logic;
      mwrn      : out std_logic;
      moen      : out std_logic;
      msndimm   : out std_logic;           -- unused

-- High speed Synchronous 245 signals
      hsndimm   : out std_logic;           -- unused
      hclk60    : in  std_logic;           -- 60MHz clock input
      hdata     : inout std_logic_vector(7 downto 0);
      hrxfn     : in std_logic;            -- RX Full #
      htxen     : in std_logic;            -- TX Full #
      hoen      : out std_logic;           -- OE# HBDBUS6
      hrdn      : out std_logic;           -- RD#
      hwrn      : out std_logic            -- WR#
);
   end morphic_hs_245_sync_fifo;
```

The following VHDL files are comprised to the application RBF file when the Quartus-II project for this application is compiled:

add8.vhd
– An eight bit full adder
dncntlg.vhd
– A generic n-bit down counter
dpram4.vhd
– A four byte dual port RAM
fadd1.vhd
– A one bit full adder
hs245_sif.vhd
– 245 Fast Serial Interface for testing
MorphIC-HS-245_Sync_fifo.vhd
– Top level entity used to communicate between two Synchronous 245 devices
seq_trig.vhd
– Monitors Bus and triggers when presend value does not match the previous value + 1
sync_fifo.vhd
– FIFO to Buffer with two different clock domains.
Syncflop.vhd
– Synchronises signals with different clock domains
upcntg.vhd
– A generic n-bit up counter

The name of the Quartus-II project is MorphIC_HS_245_Sync_fifo.qpf

3.3 Reset Polarity

When programming the Morph-IC-II with the default USB-to-FPGA utilities an active high reset needs to be used. When programming the Morph-IC-II over JTAG an active low reset should be used.

For programming over USB, set the HDL as follows:
MorphIC_HS_245_Sync_fifo.vhd-
193 –reset_n <= rst; -- polarity to programme over JTAG
194 reset_n <= not rst; -- polarity to programme over USB
Note this is the default setting.

For programming over JTAG, set the HDL as follows:
MorphIC_HS_245_Sync_fifo.vhd-
193 reset_n <= rst; -- polarity to programme over JTAG
194 –reset_n <= not rst; -- polarity to programme over USB

4 Example Application Procedure

In this section a step by step guide is given for establishing Synchronous 245 FIFO communications between a Morph-IC-II and a UM232H. This guide covers the following processes:

- Configuring the EEPROM of the UM232H
- Connecting the UM232H to the Morph-IC-II
- Compiling the Quartus-II project and editing the Pin-Map of the application
- Loading the application to the FPGA
- Verification of Synchronous 245 FIFO communications

4.1 Configuring the EEPROM of the UM232H

The EEPROM of the UM232H is set to UART mode by default. To establish Synchronous 245 FIFO communications the Serial/FIFO is required to be set into in 245 FIFO mode by setting the parameters as shown in **bold** in the table below The default EEPROM configuration of the Morph-IC-II is suitable for Synchronous 245 FIFO applications.

Parameter	Value	Notes
USB Vendor ID (VID)	0403h	FTDI default VID (hex)
USB Product UD (PID)	6014h	FTDI default PID (hex)
bcd Device	009h	
Pull down I/O Pins in USB Suspend	Disabled	Enabling this option will make the device pull down on the UART interface lines when in USB suspend mode (PWREN# is high).
Manufacturer Name	FTDI	
Max Bus Power Current	150mA	
Power Source	Bus Powered	
Device Type	FT232H	
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	Enabled	Taking RI# low will wake up the USB host controller from suspend in approximately 20 ms.
High Current I/Os	Disabled	Enables the high drive level on the UART and ACBUS I/O pins.
Load VCP Driver	Enabled	Makes the device load the VCP driver interface for the device.
Hardware Specific Port A	**245 FIFO**	Select 245 FIFO mode to communicate in 245 FIFO signals through Port A.
Driver	**D2XX Direct**	Suppresses loading of VCP driver.

Table 4.1 Recommended EEPROM Configuration

For a detailed guide in how to programme and EEPROM:

FT_PROG User Guide:
http://www.ftdichip.com/Support/Documents/AppNotes/AN_124_User_Guide_For_FT_PROG.pdf

4.2 Connecting the Morph-IC-II and UM232H

In order to establish communications between a UM232H and a Morph-IC-II, both devices need to be correctly connected to each other as shown in Figure 6 where the pin labels along with the pin designators of each connected pin are given. The pins of the Morph-IC-II can be relocated since they are all general purpose I/O pins, the Morph-IC-II pin locations illustrated here are chosen to be in a simple location and they all correlate with the pin-map in the Quartus-II project supplied for this application.

It should be noted that in the UM232H VBUS is connected to VCC to power the chip and V3V3 is connected to VIO to allow the IOs of the FT232H to be powered; these connections are illustrated in Figure 6.

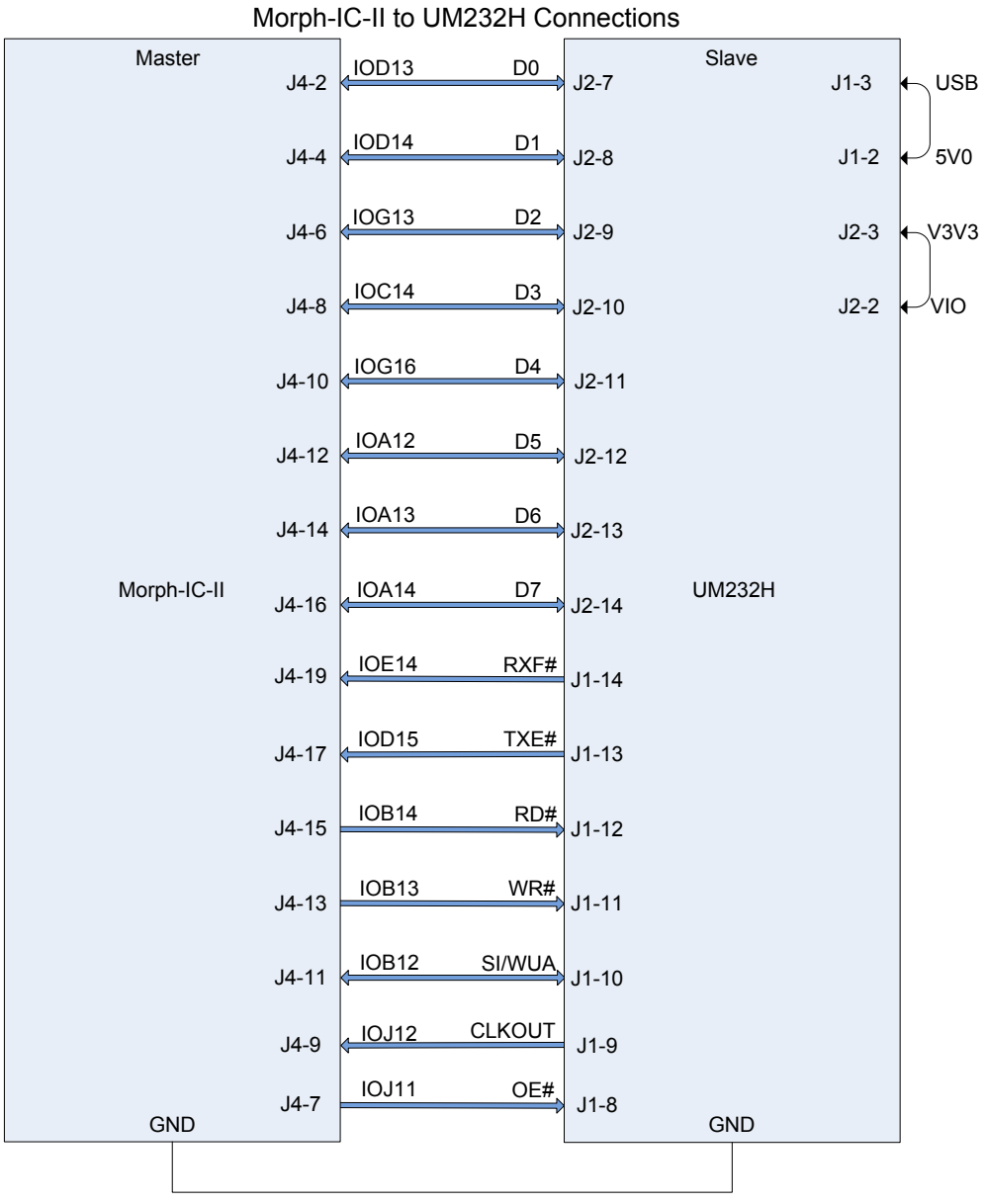


Figure 6 – Morph-IC-II and FT232H wire scheme

The basic wire configuration of the Morph-IC-II is illustrated in Figure 7. In this diagram it can be seen that there is another synchronous 245 FIFO network to that of the UM232H. This Synchronous 245 interface is found between the FT2232H and the FPGA and is internally wired on the Morph-IC-II.

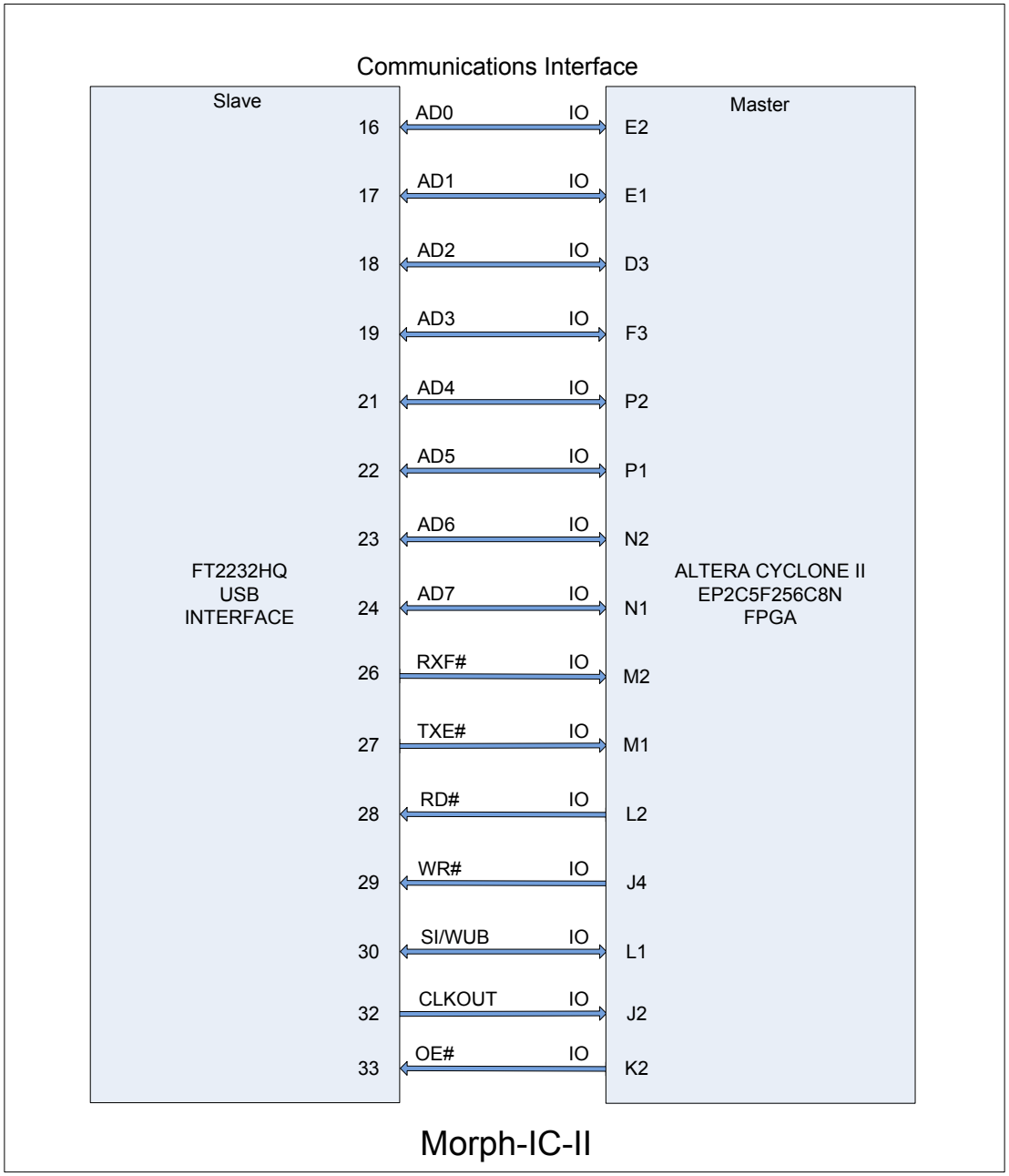


Figure 7 – Communications between the FT2232H of the Morph-IC-II and the FPGA of the Morph-IC-II

4.3 Editing the HDL Project

The Synchronous 245 application contains a Quartus II archive file containing all the source code and compiling parameters. A pre-compiled RBF is also included thus the Quartus II software package is not required for this project, however if any editing is required, the archived project file can be opened and edited using Quartus II.

For Quartus II download and support, please refer to www.altera.com

4.4 Load RBF

This section will describe how an application is programmed onto the Morph-IC-II. The Morph-IC-II uses a *.RBF (Raw Binary File) as a standard format. Included in the Synchronous 245 Application download file is a utility called MorphLd-II.exe. This utility can be used to load RBF files to the Morph-IC-II. The icon is illustrated in Figure 8.

To load the featured Synchronous 245 application to the Morph-IC-II; open the MorphLd-II.exe utility and select Morph-IC-II B as the subject device port. The next step it to click on the Browse button inside the MorphLd-II.exe panel, and then open the file "MorphIC_HS_245_Sync_fifo.rbf" as illustrated in Figure 9.

With the suitable RFB file and subject device port selected the Morph-IC-II can be programmed by clicking "Program" as shown Figure 10.

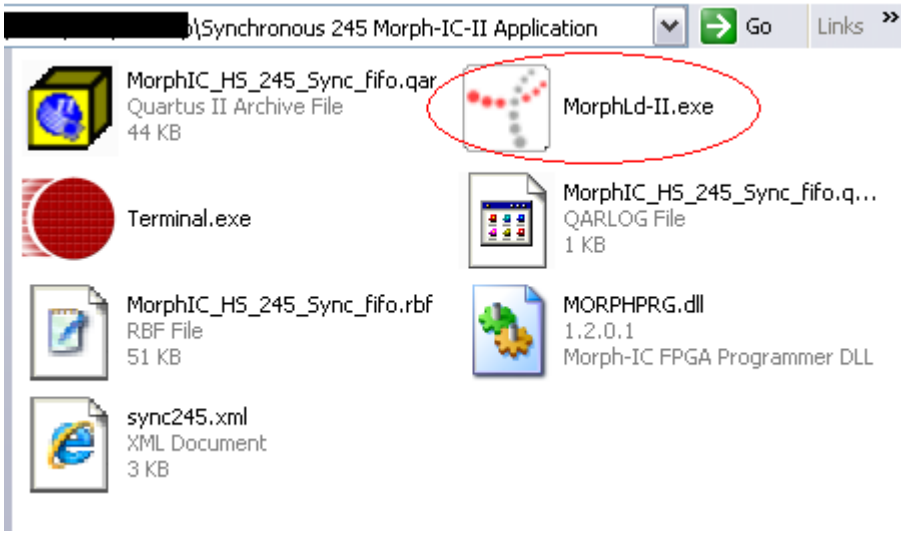


Figure 8 – Opening the MorphLd-II Utility

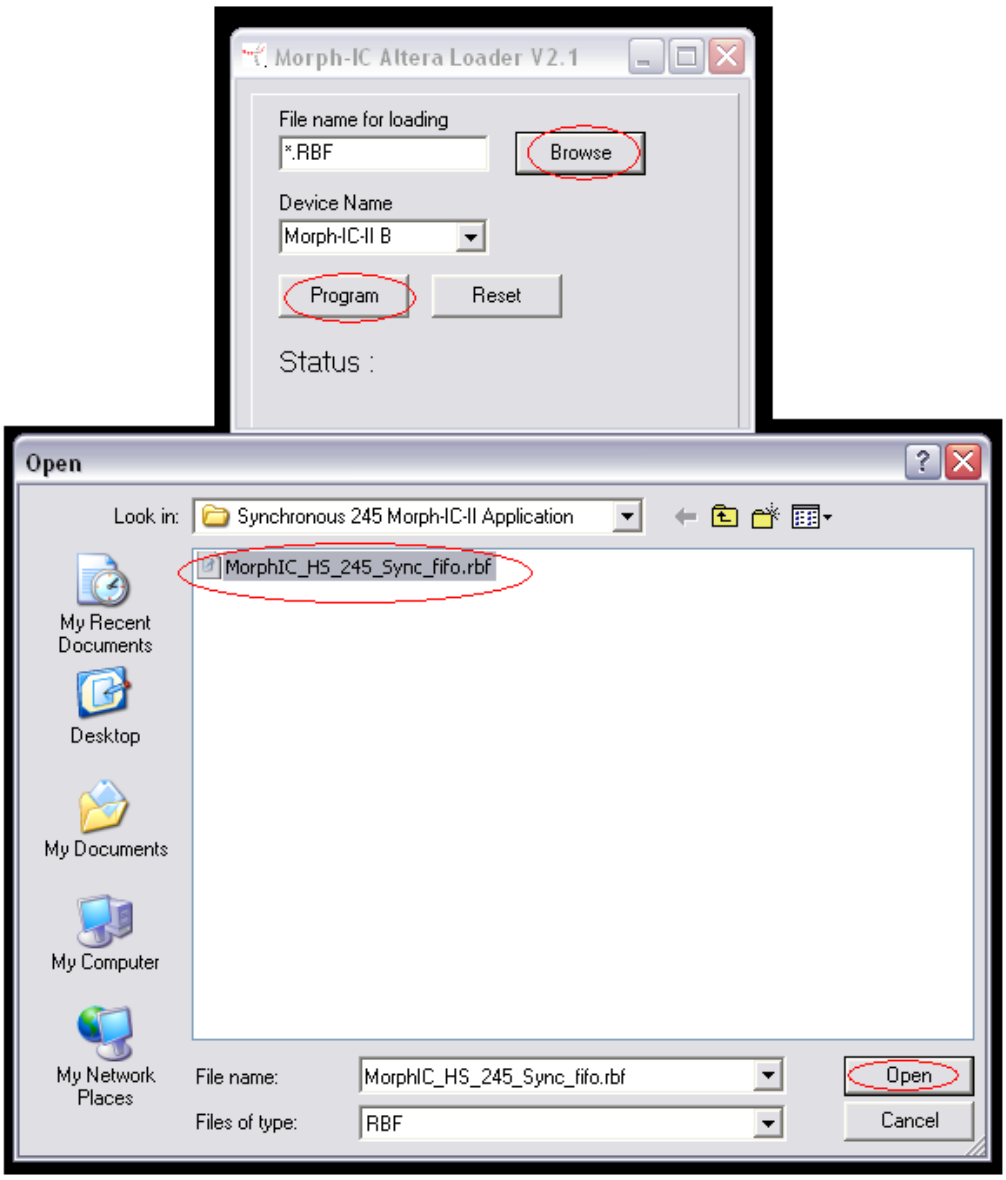


Figure 9 – Select the RBF File to be Loaded

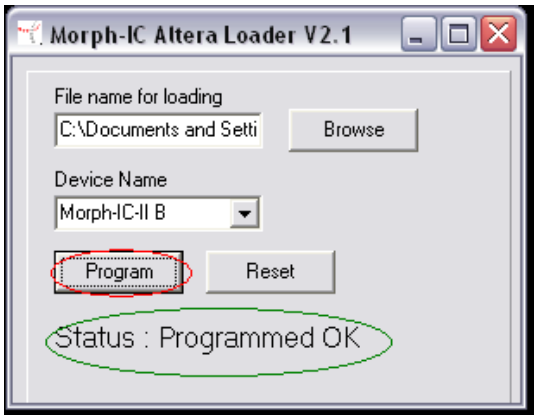


Figure 10 – Program the Morph-IC-II

4.5 Test for communications

In this section a description of how to set up two terminal programs which will be used to enter text to be sent from one device to the other and to display the received text from the receiving device. The two terminals used to do this are shown in Figure 11. To set these terminals the following steps should be completed:

Select the devices required for communication. It can be seen in Figure 11 that each terminal communicates with one specific device, the top terminal communicates with Channel A of the Morph-IC-II (a channel dedicated for communications) and the bottom terminal communicates with the UM232H. In this experiment both terminals should be set up to communicate with Channel A of the Morph-IC-II and the UM232H respectively.

Define the mode of communications. The mode of communications can be specified by selecting "Special Modes" and checking the "Enable Synchronous 245" checkbox as illustrated in Figure 11. Once this is completed open the ports of both devices by clicking open.

Send text from one device and read that text with the other device. To carry out a basic test of communications from one device to another type anything in to the ASCII format box on one terminal then click return, it can be seen in Figure 12 the string "hi" was typed in one terminal and retrieved by the other and similarly for the string "hello" but in the opposite direction.

Set a location and a name for a received file. In order to send a large amount of data using this terminal utility the location and a name must be defined for the file that will be received. It is essential that the file name has the same extension as the transmitted file. To set the location and name of a file open the "File Xfer" tab, open the "RCV file" panel, in this panel select a directory and a name for the received file. An example this is shown in Figure 13.

Sending a file. To send a file to the location set by other terminal's RCV file function, open the "File Xfer" tab, select "Send File", select a file with the same extension as the file set by the RCV file and click on "Open". An example this is shown in Figure 14.

Close File and Verify. Once the transfer has completed the received file can be closed thus completing the transfer and receive process. At this stage it is possible to verify all the data has been transferred correctly without corruption. An example of this step is shown in Figure 15.

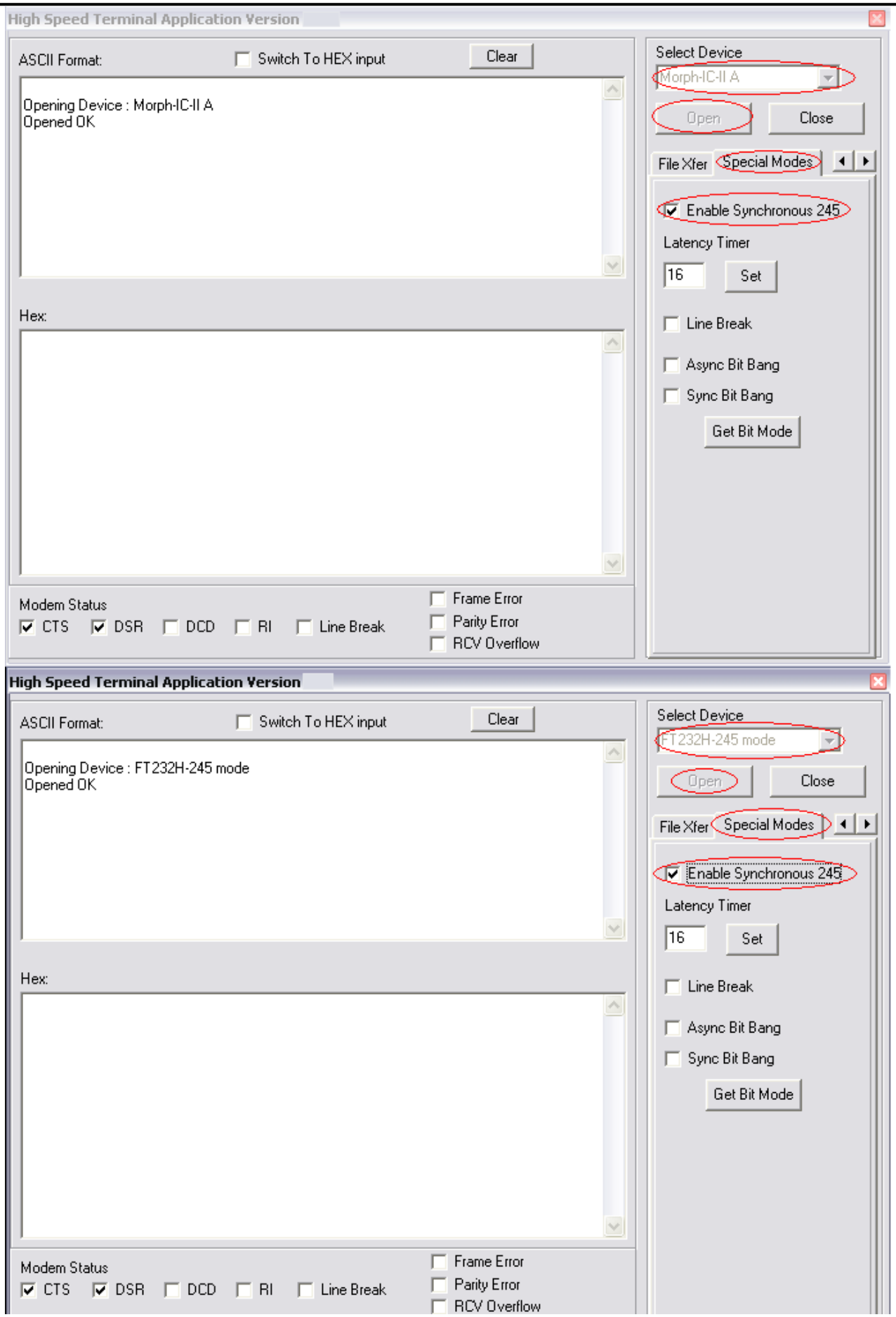


Figure 11 – Configuring the Terminals for Synchronous 245 Communications

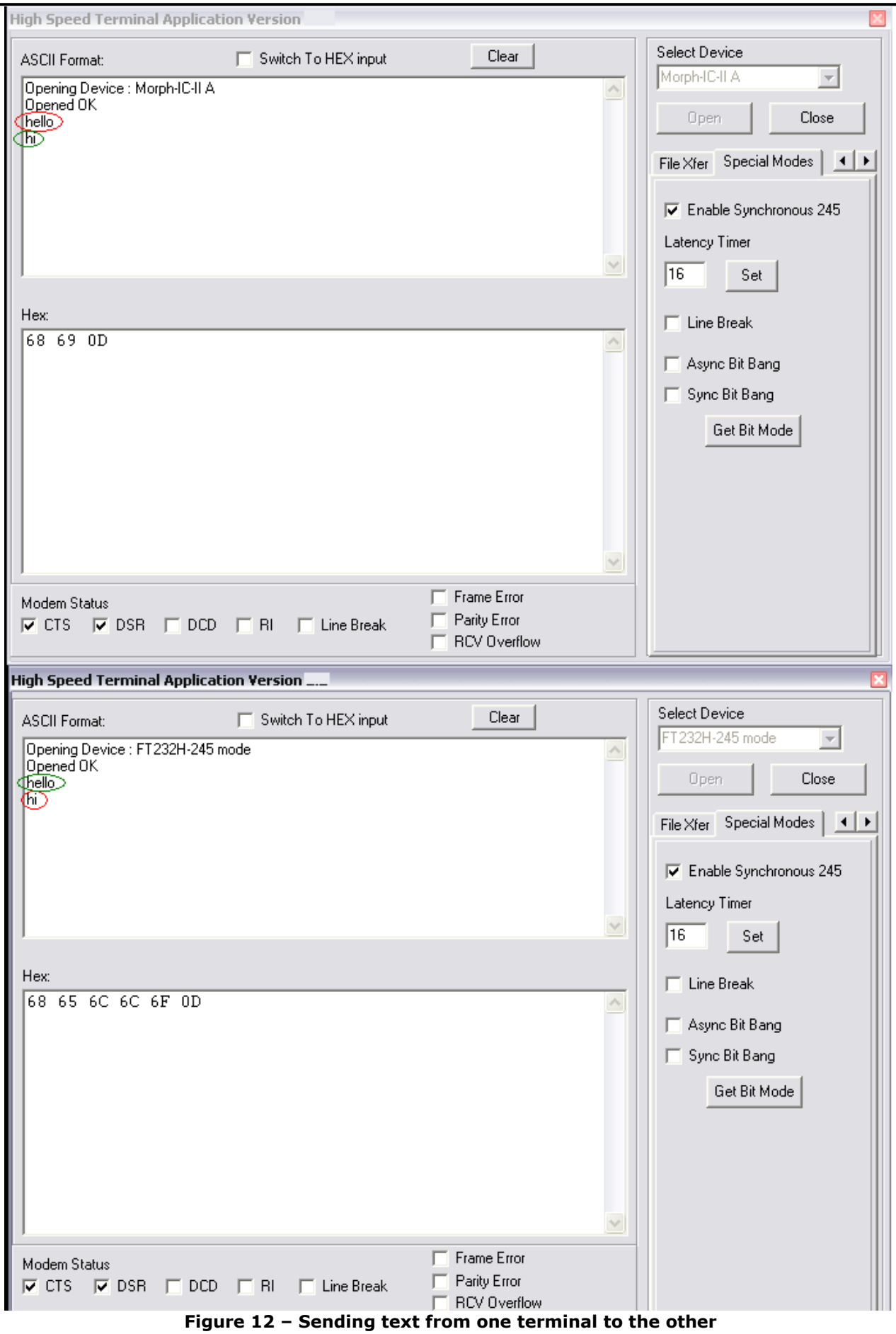


Figure 12 – Sending text from one terminal to the other

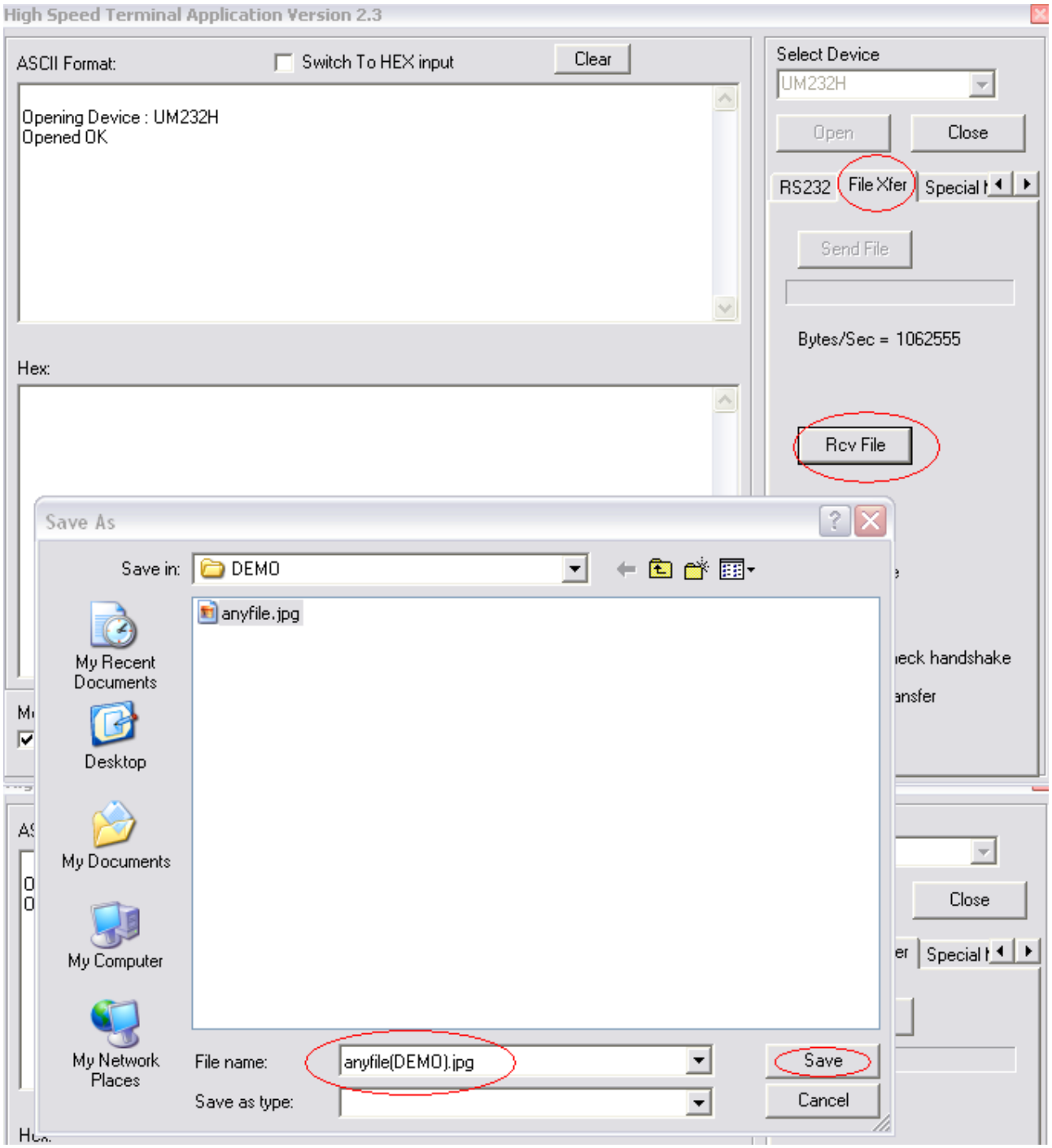


Figure 13 – Setting the Location for a Received File

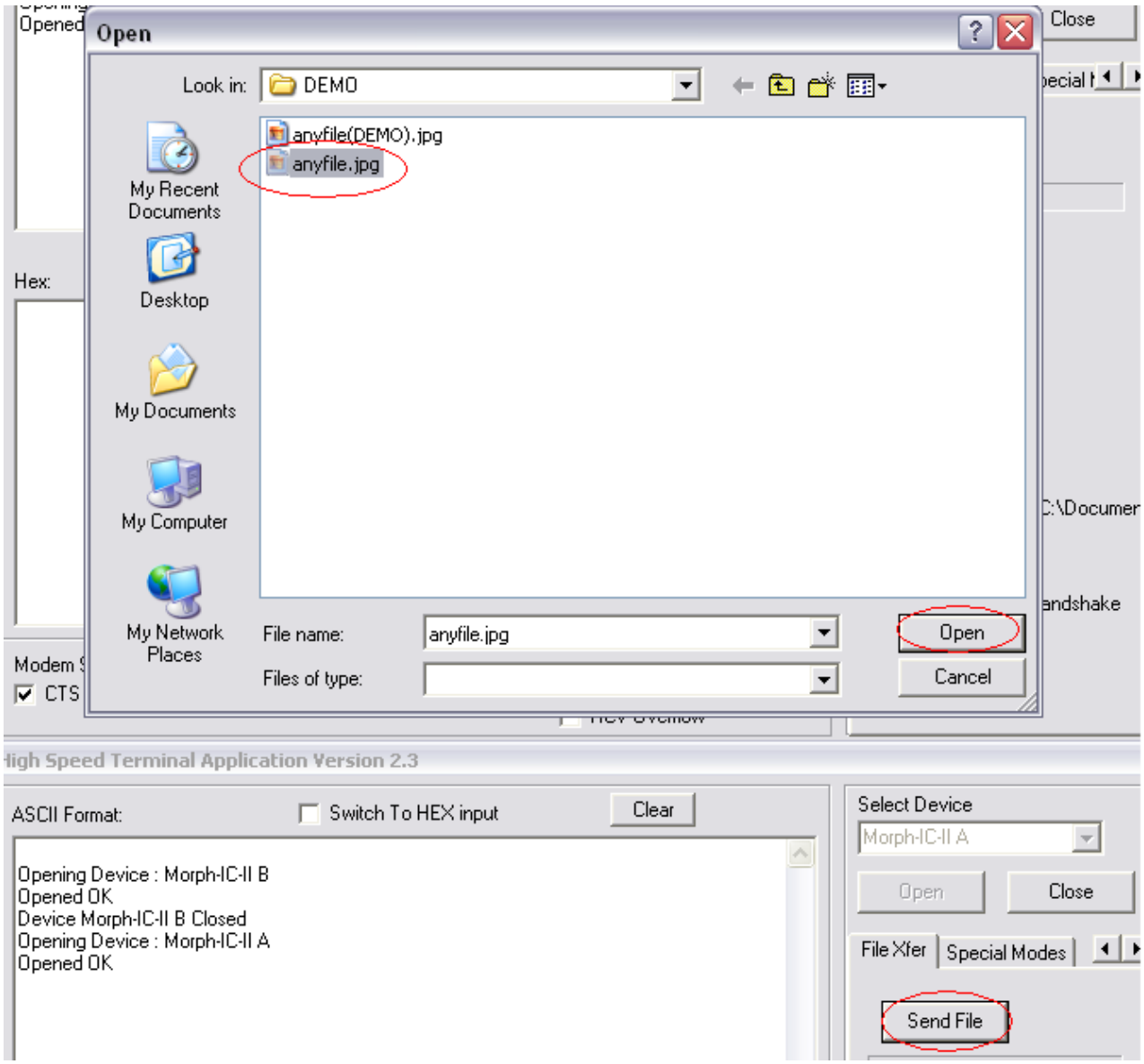


Figure 14 – Sending a File

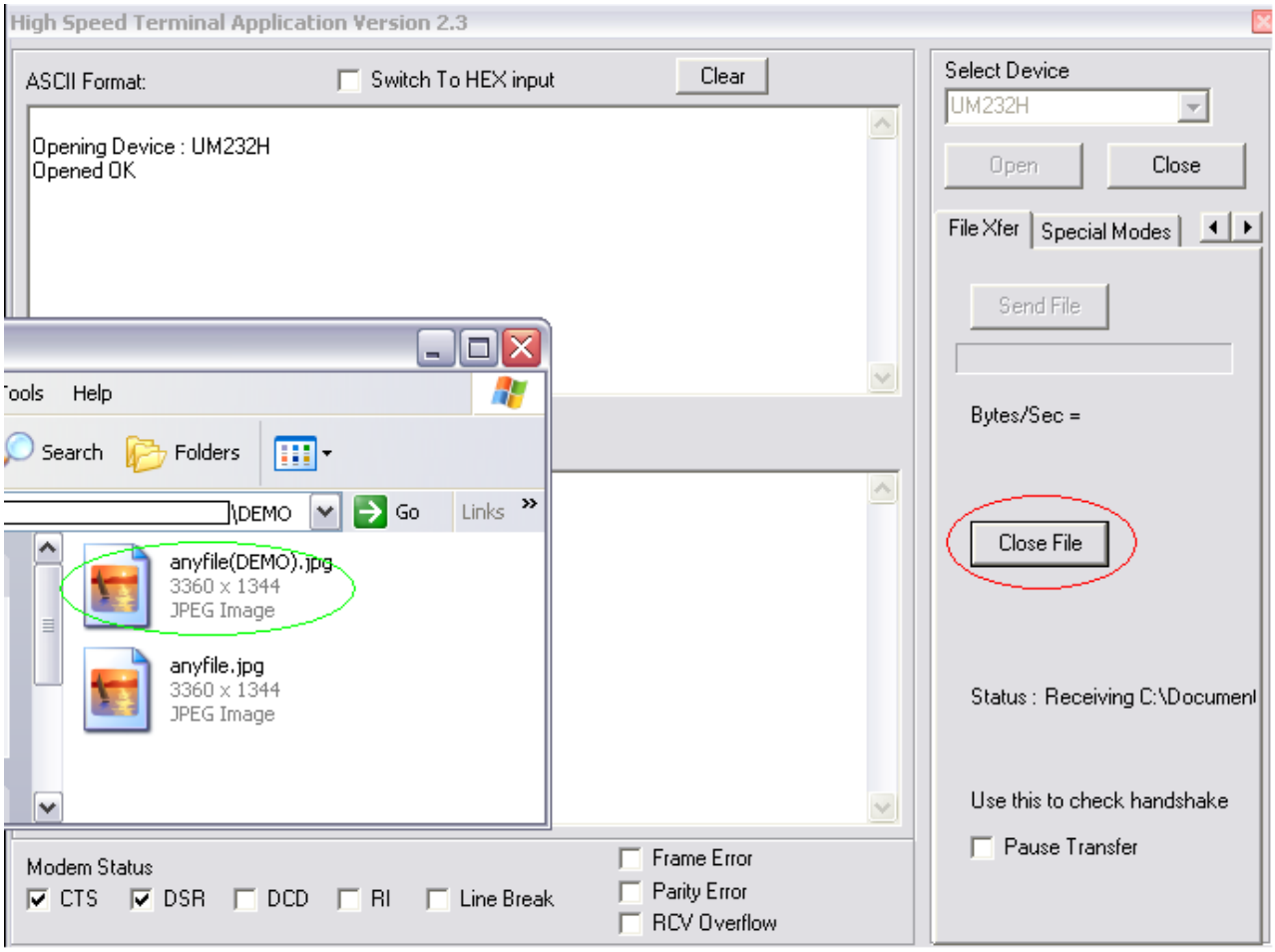


Figure 15 – Close File and Verify

5 Summary

This application note provides a background explanation of synchronous 245 FIFO mode and illustrates an example of establishing synchronous 245 FIFO communications between two slave devices via a master using an FTDI Morph-IC-II.

6 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 – 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

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 8791 3570
Fax: +886 (0) 2 8791 3576

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

Appendix A – Abbreviations

Terms	Description
EEPROM	Electrically Erasable Programmable Read Only Memory
FIFO	First In First Out
FPGA	Field Programmable Gate Array
FTDI	Future Technology Devices International Ltd.
MPSSE	Multi-Protocol Synchronous Serial Engine
RBF	Raw Binary Format
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus

Appendix B – References

Synchronous 245 Morph-IC-II Application

http://www.ftdichip.com/Products/Files/Synchronous 245 Morph-IC-II Application.zip

"Morph-IC-II Applications and Utilities" download

http://www.ftdichip.com/Support/Utilities/MorphIC-II%20Package.zip

Hi-Speed Mini Modules

http://www.ftdichip.com/Products/EvaluationKits/HiSpeedModules.htm

FT_Prog

http://www.ftdichip.com/Resources/Utilities/FT_PROG.zip

D2xx Programmers Guide
http://www.ftdichip.com/Documents/ProgramGuides/D2XX_Programmer's_Guide(FT_000071).pdf

Interfacing FT2232H device to SPI
http://www.ftdichip.com/Projects/MPSSE/AN_114_FTDI_Hi_Speed_USB_To_SPI_Example.pdf

Recovery utility

http://www.ftdichip.com/Resources/Utilities/SPITest.zip

Quartus-II

http://www.altera.com/products/software/quartus-ii/web-edition/qts-we-index.html

Appendix C – Revision History

Revision	Changes	Date
1.0	First Issue	2011-09-12
1.1	Corrected Figure 6 – Morph-IC-II and FT232H wire scheme	2012-06-26
	Updated contact information	