

# DSO 100M FPGA DEVELOPMENT SYSTEM User Manual

The DSO 100M is an Open Source Digital Storage Oscilloscope Development System created by Earth People Technology. It consists of a board, FPGA design source files and a C# project with all source code needed to build a complete DSO. The board is a four channel oscilloscope that accepts analog signals +/- 40Volts up to 5MHz. Each channel has its own low pass filter, amplification and sampling.

The C# project is used to build the UnoProLyzer application and runs on a Windows 10/8/7 PC. It displays each analog signal data from the DSO 100M in a window with color coded signals. This application includes a trigger mechanism, zoom in/out, scroll, amplitude measurement, vertical and timebase selection.

Circuit designs, software and documentation are copyright © 2019, Earth People Technology, Inc

Microsoft and Windows are both registered trademarks of Microsoft Corporation. Altera is a trademark of the Altera Corporation. All other trademarks referenced herein are the property of their respective owners and no trademark rights to the same are claimed.

http://www.earthpeopletechnology.com/



# **Table of Contents**

1	Introduction and General Description
2	DSO 100M Getting Started
3	DSO 100M Driver
4	UnoProLyzer Application
5	DSO 100M Description
	5.1.1 Power
	5.1.2 Analog Inputs
	5.1.3 Bias Amplifiers
	5.1.4 Programmable Gain Amplifiers
	5.1.5 Analog To Digital Converters
	5.1.6 FPGA Configuration
	5.1.7 FT2232H Dual Channel USB to Serial Chip
6	FPGA Code Description
7	EPT Drivers
1	7.1 USB Driver
1	7.2 JTAG DLL Insert to Quartus Prime
	7.2.1 Installing Quartus
	7.2.2 Downloading Quartus
	7.2.3 Quartus Installer
	7.2.4 Adding the EPT_Blaster to Quartus Prime
1	7.3 Active Host Application DLL
8	FPGA Active Transfer Library
3	B.1 EPT Active Transfer System Overview 73
	8.2 Active Transfer Library
	8.2.1 Active Trigger EndTerm
	8.2.2 Active Transfer EndTerm
	8.2.3 Active Block EndTerm
	B.3  Timing Diagram for Active Transfer EndTerms
	8.3.1 Active Trigger EndTerm Timing
	8.3.2 Active Transfer EndTerm Timing
	8.3.3 Active Block EndTerm Timing
9	PC Active Host Description
10	Active Host Application
2	10.1 Trigger EndTerm
2	10.2 Transfer(Byte) EndTerm
	10.3 Block EndTerm
	10.4 Active Host DLL
	<u>10.4.1</u> Active Host Open Device
	10.4.2 Active Host Read Callback Function
	10.4.3 Active Host Triggers
	10.4.4 Active Host Byte Transfers



10.4.5 Active Host Block Transfers	121
11 Assembling, Building, and Executing a .NET Project on the PC	124
11.1 Creating a Project	125
11.1.1 Setting up the C# Express Environment x64 bit	126
11.2 Assembling Files into the Project	133
11.2.1 Add Files to Project	135
11.2.2 Adding Controls to the Project	137
11.2.3 Adding the DLL's to the Project	141
11.2.4 Building the Project	142
12 FPGA DSO 100M Verilog Description	<u>143</u>
12.1 UC Controller Description Error! Bookmark no	ot defined.
12.2 ADC Storage Description	153
13 Compiling, Synthesizing, and Programming FPGA	159
13.1 Setting up the Project and Compiling	159
13.1.1 Selecting Pins and Synthesizing	165
13.1.2 Configuring the FPGA	173
14 PC UnoProLyzer Code Base Description	
A	
2-DSO 100M Getting Started	<del>5</del>
3 DSO 100M Driver	<del>7</del>
4 UnoProLyzer Application	<del>7</del>
5 DSO 100M Description	
5.1.1 Power	
5.1.2 Analog Inputs	<del>30</del>
5.1.3 Bias Amplitiers	
5.1.4 Programmable Gain Amplifiers	
5.1.5 Analog To Digital Converters	
5.1.6 FPGA Contiguration	<del>30</del>
5.1.7 — FT2232H Dual Channel USB to Serial Chip	
6 FPGA Code Description	
6.1 Active Host End Lerms UnoProLyzer Project	
6.2 EDGA II + L + C	
<del>6.3</del> – FPGA Host Interface	<del></del>
6.4 UC Controller Description	<del></del>
<del>0.3</del> —ADC Storage Description	
- EPI DIVERS	
7.1 USB DIVE	
7.2 JIAG DEL Insert to Quartus Prime	
7.2.1 — Installing Quartus	<del>49</del>
7.2.2 Downloading Quartus	<del></del>
7.2.3 Quartus Installer	
+.2.4 Auding the EF1_Blaster to Quartus Prime	<del></del>

Formatted	<b></b>
Formatted	
Formatted	
Formatted	
Formatted	
Formatted	<u> </u>
Formatted	<u> </u>
Formatted	
Formatted	(
Formatted	
Formatted	 []
Formatted	
Formatted	
Formatted	 
Formatted	
Formatted	
Formatted	
I UTIMALLEU	L



8 FPGA Active Transfer Library	62
8.1 EPT Active Transfer System Overview	
8.2 Active Transfer Library	63
8.2.1—Active Trigger EndTerm	65
8.2.2—Active Transfer EndTerm	69
8.2.3—Active Block EndTerm	71
8.3—Timing Diagram for Active Transfer EndTerms	<del>74</del>
8.3.1 Active Trigger EndTerm Timing	<del>74</del>
8.3.2—Active Transfer EndTerm Timing	<del>74</del>
8.3.3 Active Block EndTerm Timing	<del>75</del>
9 Compiling, Synthesizing, and Programming FPGA	<del>75</del>
9.1 Setting up the Project and Compiling	<del>76</del>
9.1.1—Selecting Pins and Synthesizing	<del>82</del>
9.1.2—Configuring the FPGA	<del>90</del>
10—Active Host Application	<del>97</del>
10.1 Trigger EndTerm	<del>97</del>
10.2 Transfer(Byte) EndTerm	<del>98</del>
10.3 Block EndTerm	<del>98</del>
10.4 Active Host DLL	<del>98</del>
10.4.1 Active Host Open Device	100
10.4.2 Active Host Read Callback Function	102
10.4.3 Active Host Triggers	103
10.4.4 Active Host Byte Transfers	105
10.4.5—Active Host Block Transfers	107
H—Assembling, Building, and Executing a .NET Project on the PC	110
H.I. Creating a Project.	++++
H.I.I. Setting up the C# Express Environment x64 bit	<u>H2</u>
H.2 Assembling Files into the Project	119
H.2.1 — Changing Project Name	121
11.2.2 Adding Controls to the Designt	122
11.2.5 - Adding the DLL's to the Project	123
11.2.4 Auding the DLL S to the Project	127 128
11.2.5 — Dunumg the Project	120 120
11.2. UnoProLyzer Code Rose Description	127
TI.5 — Chor rollyzer Code Dase Description	152

-	Formatted	
-	Formatted	(
\	Formatted	(
$\langle$	Formatted	
'	Formatted	(
/	Formatted	(
ľ	Formatted	(
//	Formatted	(
//	Formatted	(
$\ $	Formatted	(
$\ $	Formatted	(
	Formatted	<b></b>
	Formatted	(
	Formatted	
	Formatted	(
	Formatted	



# **1** Introduction and General Description

The Earth People Technology DSO 100M development system hardware consists of a High Speed USB to parallel (8 bit) bus chip, FPGA, analog signal conditioning and Analog to Digital Converters. The USB interface provides both Configuration of the FPGA and a High Speed transfer path. The Verilog firmware includes the Active Transfer Library which is used in the FPGA to provide functions for control and data transfer to/from the PC. The UnoProLyzer software consists of the Active Host SDK for the PC. This SDK provides virtual "pipes" over the USB to provide address selectable paths to send/receive data from the FPGA.





These virtual pipes are called "EndTerms". There are three types of EndTerms.

- Trigger EndTerm
- Transfer EndTerm (Single Byte)
- Block EndTerm (256 Byte Block)

The EndTerms are designed for ease of use and are commensurate between the C# SDK and the Verilog Transfer Library. For example, send a byte from the C# project to the FPGA using the Transfer EndTerm function. The byte be available in the FPGA via the USB to Serial chip within 1 millisecond. The same goes for sending data from FPGA to the PC.

All of the drivers, libraries, and project source code are available at www.earthpeopletechnology.com.

# 2 DSO 100M Getting Started



Formatted: Font: (Default) Times New Roman



The DSO 100M board comes pre-loaded with the DSO\_100M HDL project in the FPGA. This project allows the user to test out the functions of the Active Host API and the board hardware.

To test drive the application, connect the DSO 100M to the Windows PC using a Micro B USB cable. Load the driver for the board. See the section "EPT Drivers" for instructions on loading the DPL driver. If the USB driver fails to load, the Windows OS will indicate that no driver was loaded for the device. In the case of the failed USB driver, try rebooting the PC and following the steps in the EPT Drivers section of this User Manual.



Below is a list of the steps to take to get started using the UnoProLyzer.

- 1. Install the DSO 100M Driver
- 2. Install the UnoProLyzer Application
- 3. Connect the DSO 100M board to the USB Port on the PC
- 4. Click on the UnoProLyzer Icon under "All Programs"
- 5. When the application opens up, click on the drop down box in the upper right corner.
- 6. Select the "EPT USB<->Serial&JTAG Cable B".
- 7. Click on the "Open" button
- 8. Click on one of the "Select Channels" buttons.
- 9. Then click on the "Start" button.
- 10. All selected analog channels should appear on the display.





# 3 DSO 100M Driver

Connect the DSO 100M board to a USB port on the PC. Load the driver located on the UNOPROLYZER\_OSCILLOSCOPE\_1.1\_DVD at /DSO\_100M/Drivers/EPT\_2.08.24



Double click on the CDM v2.08.30 Setup.exe to install the USB drivers. The DSO 100M uses the ftdibus.sys driver. This driver is loaded upon connection of the USB to Serial Cable to the PC by the ftdibus.inf file. To install these two files onto your PC, follow the instructions from the "Update Driver Software" utility. This utility will automatically load when the board is connected to the PC.

# 4 UnoProLyzer Application

When the application loads, click on the drop down box at the upper right of the window. Select the EPT USB<->Serial&JTAG Cable B in the list. Next, Click "Open", select number of channels and click "Start". The analog data will display on the graph.





Go to the upper right of the window and click on the drop down box.



Select the "EPT USB<-> Serial&JTAG Cable B. Then click on the Open button







Next, select the number of channels to display. The channels have to be selected in sequential order, you cannot pick out single channel (except for channel 1). So for two channels, click on the "1 to 2" button. This will display the data from both channel 1 and 2.

Page 10



EPT U	ISB <-> Serial&JTAG Cable 👻
Horizontal (Scale) Horizontal Position Ch 1	Open Close Scan Control Start Stop Single CONVST Scan Channel Select 1 Only age Scale 1 to 2 1 to 3 1 to 4
Channel Select	Registers Conversion Setup Average Reset

Next, click on the Start button.



,			
/	EPT	USB <-> Serial&	JTAG Cable 🔻
,		Open S	Close can Control
	Horizontal (Scale)	Start	Stop
,			Single CONVST
Ch 1	Horizontal Position		Scan Channel Select
,	< >		1 Only
	Vertical Position Vertical Position	oltage Scale	1 to 2
,			1 to 3 1 to 4
	Channel	Reg	isters
	Select		Setup Average Reset

The data from the two channels will appear at the same latitude on the graph.





Next, locate the Channel Select drop down box and click on channel 1.





Locate the Vertical Position slider and pull it down. The channel 1 data will change position in the graph depending on where you move the slider. The voltage magnitude data also adjusts to indicate the magnitude of the data relative to the position of channel 1 data.





The selected channel will show up as a large icon. Its position indicates the zero position of the data. The magnitude information along the y-axis is only for the selected channel.



Then connect a signal to the channel 1 input on the UnoProLogic2.



If you don't have a 0-5 Volt signal to connect to the UnoProLogic2, you can use your finger and touch it to the bottom of the Analog Input Connector. The ambient electricity from your body has just enough current to give the Analog inputs a deflection from zero.





Now the UnoProLogic2 and UnoProLyzer are ready to measure an 0-5VDC signals.







To set up triggering, locate the "Trigger Menu". In this menu, locate the "Trigger Channel" drop down box.





Click on the drop down box and select the channel in which to scan for the trigger value.





Next, adjust the "Trigger Voltage" level to the appropriate trigger threshold point.





Click on the "Trigg On" button to turn on triggering.





When the UnoProLyzer connects with the EPT-5M57-AP-U2 and all four channels are setup and start button has been pressed, the display will show all four analog inputs on the screen.





All of the channels will be grouped at the same spot on the graph by default. To separate these channels, click on the drop down box underneath the "Channel Select" lable.





Select a channel and us the "Vertical Position" slider to adjust the position of the channel graph in the display. Notice that the channel indicator highlights in bold when the channel is selected.





Grab the "Vertical Scale" slider and push up on it until the scale is 1/3. This divides each data element in channel 1 by 1/3.





Repeat this process for all six signals. Leaving one division between each channel.





Each channel can also be scaled in the Horizontal Scale. Provide Channel 1 with a signal. A simple way to do this is to touch your finger to the analog input. Then click the "Stop" button. The "Stop" button will cause the ProLyzer to stop filling the circular buffers with new data. All of the previously stored data is preserved in the circular buffers. This allows to you to scroll through the data to view events that occurred in the past.





Turn off each channel (except for Channel 1) by first selecting the channel, then clicking on the "On/Off" button directly below the "Channel Select" drop down box.





Select Channel 1 and grab the "Time (Scale) " slider and slide it to the left. Notice the signal is zooming out.

# 5 DSO 100M Description

The DSO 100M board is equipped with an Altera EP4CE6E22C8 FPGA; which is programmed using the Altera Quartus Prime software. The FPGA has 6672 Logic Elements and 276480 Total RAM Bits. An on board 66 MHz oscillator is used by the EPT Active Transfer Library to provide data transfer rates of up to 8 Mega Bytes per second. The board contains four separate analog signal paths. Each analog channel has a dedicated 100MHz ADC. Each channel has a high accuracy bias and signal amplifiers



capable of providing a 10MHz analog signal for conversion at each ADC. The hardware features are as follows.

- Altera EP4CE6 FPGA with 6272 Logic Cells
- Dual Channel High Speed USB FT2232H
- 66 MHz oscillator for driving USB data transfers and users code
- 100MHz oscillator for scaling up/down for users needs
- Four 100MSPS ADCs
- High accuracy Op-Amps capable of providing 10MHz analog signals at the ADCs







#### 5.1.1 Power

The DSO 100M runs from USB power only. The USB port provides +5VDC. This power is split into three power rails; +3.3V, +2.5V, +1.2V and ±5V. The DSO 100M consumes approximately 300mA from the USB port. This is a total 1.5W from the +5VDC of the USB port. The standard USB port can safely supply a maximum of 500mA from each port. So, the DSO 100M consumes a safe margin with respect to the maximum.





The +3.3V, +2.5V and +1.2V supplies the power for FPGA. The +3.3V is used to power the FT2232H chip, Configuration Flash, Oscillators, ADC's, Digital Potentiometers, EEPROM, and bus Transceivers. The ±5V supply is used by the analog Op-Amps.

### 5.1.2 Analog Inputs

There are four analog inputs and each input uses a BNC connector on the PCB. Each analog input path has been tuned to 50 Ohm impedance on the top layer of the PCB. This matches the impedance of the BNC connector and minimizes reflections and noise.

The analog input circuit's responsibility is to protect the input amplifiers and divide the input analog signal for compatibility with the ADC. The input protection is provided by a diode clamp to the +5VDC power supply and a diode clamp to the -5VDC power supply.



Formatted: Centered

Formatted: Centered

### **5.1.3** Bias Amplifiers

Each analog channel after the conditioning is applied to a bias amplifier. The bias amplifier is used to bias the input analog signal at zero voltage. The conditioned analog signal is allowed to fluctuate between ±5VDC. The DSO 100M provides an adjustable gain on the bias amp. This gain is used to provide an offset to correct for any distortion to the signal because of the signal conditioning. This gain is adjusted using a digital potentiometer (Analog Devices AD5263). This Digi-pot is controlled via the FPGA.





The Digi-Pot is a 256 position 50K Ohm selectable resistance. The Digi-Pot is controlled via I2C from the FPGA. The Host PC selects and initiates the I2C bus to command the Digi-Pot to a selected position, and thusly a resistance. Each position is a 195 Ohm step (50,000 Ohm/256 Steps) from 0 Ohm to 50,000 Ohm. The Digi-Pot uses a ±5VDC to allow the resistance to select the bias from the full analog input voltage swing.

Digi-pots and the Op-Amps allow an undistorted analog signal to be applied to the ADC's. The gain amplifier can add a gain of up to 50 with no offset. Below is a video showing a 2MHz signal on channel 1. It is perfectly centered around the zero point. The function generator output is set to +/- 600mV. The video starts at 2MHz and then the function generator is dialed down to 1MHz and finally 100KHz. The reason for the incorrect calibration on the software display, is the gain and bias tables need to be completed to scale the input value to the full range of the ADC.

Formatted: Font: (Default) Times New Roman, Font color: Auto

Formatted: Centered





#### 5.1.3–Programmable Gain Amplifiers

#### **<u>5.1.4</u>** Programmable Gain Amplifiers

After the analog input signal is applied to the bias amplifier, it passes through a gain amplifier. This gain amplifier is used to amplify the analog input signal to match the signal dynamic range of the ADC input. This amplifier circuit uses a Digi-Pot (Analog Devices AD5263) with 50K Ohm of total resistance. So, the amplifier can provide a gain of up to 50. This gain is used to boost the incoming conditioned analog signal to meet the  $\pm 1V$  input of the ADC. This gain is used in conjunction with the Vertical setting from the UnoProLyzer software.





The Vertical selection has discrete positions such as -20V, -10V, -5V, -1V, 0V etc... that correspond to specific gain selections of the gain amplifier. The gain is selected by selecting resistance values from the Digi-Pot. The Digi-Pot is a 256 position 50K Ohm selectable resistance. The Digi-Pot is controlled via I2C from the FPGA. The Host PC selects and initiates the I2C bus to command the Digi-Pot to a selected position, and thusly a resistance. Each position is a 195 Ohm step (50,000 Ohm/256 Steps) from 0 Ohm to 50,000 Ohm.

#### **OPA2209 Selected Parameters**

Parameter	Min	Typ	Max	Unit	
Vos Input offset		<u>±35</u>	<u>±150</u>	μV	
voltage					
dVos/dT Input offset		1	3	<u>μV/°C</u>	
voltage drift					
PSRR vs power supply		0.05	0.5	$\mu V/V$	4
Input bias current		<u>±1</u>	<u>±4.5</u>	nA	
Jos Input offset current		<u>±0.7</u>	<u>±4.5</u>	<u>nA</u>	
en Input voltage noise		0.13		μVPP	
Noise density @ f =		<u>3.3</u>		<u>nV/√Hz</u>	
<u>10 Hz</u>					
Noise density @ f =		<u>2.25</u>		<u>nV/√Hz</u>	
<u>100 Hz</u>					
Noise density @ f =		2.2		<u>nV/√Hz</u>	
<u>1kHz</u>					
In Input current noise		500		<u>fA/√Hz</u>	
<u>density</u>					
V <sub>CM</sub> Common-mode	(V-) + 1.5		(V+) - 1.5	<u>V</u>	
voltage range					
CMRR Common-	120	<u>130</u>		<u>dB</u>	
mode rejection ratio					
GBW Gain bandwidth		18		MHz	
product					
SR Slew rate		<u>6.4</u>		<u>V/µs</u>	
		1		1	

**5.1.4** The Gain and Bias Amplifiers are powered from a +/- 5V supply fed from a DC-\* DC brick to voltage errors to a minimum.

Page 35 Formatted: Centered

ĺ	Formatted:	Font: Bold
/(	Formatted:	Font: 8 pt
/l	Formatted:	Font: 8 pt, Subscript
Å	Formatted:	Font: 8 pt
Å	Formatted:	Font: 8 pt
Å	Formatted:	Font: 8 pt, Subscript
λ	Formatted:	Font: 8 pt
Å	Formatted:	Font: 8 pt
λ	Formatted:	Centered
λ	Formatted:	Font: 8 pt
1	Formatted:	Font: 8 pt, Subscript
-{	Formatted:	Font: 8 pt
-{	Formatted:	Font: 8 pt
-{	Formatted:	Font: 8 pt, Subscript
1	Formatted:	Font: 8 pt
1	Formatted:	Font: 8 pt
ľ	Formatted:	Font: 8 pt, Subscript
ľ	Formatted:	Font: 8 pt
ľ	Formatted:	Font: 8 pt
ľ	Formatted:	Font: 8 pt
ľ	Formatted:	Font: 8 pt
J	Formatted:	Font: 8 pt
ľ,	Formatted:	Font: 8 pt
ľ,	Formatted:	Font: 8 pt, Subscript
ľ,	Formatted:	Font: 8 pt
ľ	Formatted:	Font: 8 pt
ľ,	Formatted:	Font: 8 pt
ľ,	Formatted:	Font: 8 pt
ľ	Formatted:	Font: 8 pt
Y	Formatted:	Normal



### **<u>5.1.5</u>** Analog To Digital Converters

The DSO 100M contains four Analog Devices AD9283BRSZ-80 chips. The AD9283 is an 8-bit monolithic sampling analog-to-digital converter with an on-chip track-and-hold circuit. The chip operates at a 100 MSPS conversion rate, with outstanding dynamic performance over its full operating range.



The analog signal is applied differentially to the inputs of the AD9283. The signal is buffered and fed forward to an on-chip sample-and-hold circuit. The ADC core architecture is a bit-per-stage pipeline type converter utilizing switch capacitor techniques. The bit-per-stage blocks determine the 5 MSBs and drive a FLASH converter to encode the 3 LSBs. Each of the 5 MSB stages provides sufficient overlap and error correction to allow optimization of performance with respect to comparator accuracy. The output staging block aligns the data, carries out the error correction and feeds the data to the eight output buffers. The AD9283 includes an on-chip reference (nominally 1.25 V) and generates all clocking signals from one externally applied encode command. This makes the ADC easy to interface with and requires very few external components for operation.

#### AD9283 Selected Parameters

Parameter	Min	Тур	Max	Unit	•
Resolution		8		Bits	
Differential Nonlinearity		<u>±0.5</u>	+1.25	LSB	
Integral Nonlinearity	-1.25	<u>±0.75</u>	+1.25	LSB	
Gain Error	<u>-6</u>	<u>±2.5</u>	<u>+6</u>	<u>% FS</u>	
Input Voltage Range		<u>±512</u>		<u>mV p-p</u>	
Common-Mode Voltage		<u>±200</u>		mV	
Input Offset Voltage	<u>-35</u>	<u>±10</u>	<u>+35</u>	<u>mV</u>	
Reference Voltage	1.2	1.25	1.3	<u>V</u>	
Input Resistance	2	<u>10</u>	<u>13</u>	<u>kΩ</u>	
Input Capacitance		2		pF	

λ	Formatted: Font: Bold
1	Formatted Table
1	Formatted: Font: 8 pt
-	Formatted: Font: 8 pt
-(	Formatted: Font: 8 pt
-	Formatted: Font: 8 pt
-	Formatted: Font: 8 pt
1	Formatted: Font: 8 pt
1	Formatted: Font: 8 pt


Analog Bandwidth, Full	1	475		MHz
Power				
Maximum Conversion Rate		<u>80</u>		MSPS
Minimum Conversion Rate			1	MSPS
Encode Pulsewidth High	<u>5.0</u>		<u>1000</u>	<u>ns</u>
(ter)				
Encode Pulsewidth Low	<u>5.0</u>		<u>1000</u>	ns
(t <sub>er</sub> )				
Aperture Delay (t.)		<u>0</u>		ns
Aperture Uncertainty		<u>5</u>		<u>ps rms</u>
(Jitter)				
Output Valid Time (tw)	<u>2.0</u>	<u>3.0</u>		<u>ns</u>
Output Propagation Delay		<u>4.5</u>	7.0	ns
(ten)				
Effective Number of Bits				
$f_{\rm IN} = 10.3 \text{ MHz}$		<u>7.5</u>		Bits
$f_{IN} = 27 \text{ MHz}$		<u>7.5</u>		Bits
$f_{\rm IN} = 41 \text{ MHz}$		7.5		Bits

#### <u>Output Coding ( $V_{REF} = 1.25V$ )</u>

Step	AIN-/AIN	Digital Output
255	0.512	<u>1111 1111</u>
•	4	<u>.</u>
•	2	
128	0.002	<u>1000 0000</u>
127	<u>-0.002</u>	0111 1111
•	<b>.</b>	<u>.</u>
<u>.</u>	<u>.</u>	<u>.</u>
D	-0.512	0000 0000

#### AD9283 ADC Encoding



The ENCODE input is fully TTL/CMOS compatible with a nominal threshold of 1.5 V. Care was taken on the chip to match clock line delays and maintain sharp clock logic transitions. Any high speed A/D converter is extremely sensitive to the quality of the sampling clock provided by the user. This ADC uses an on-chip sample-and-hold

Formatted	(
Formatted	
Formatted	(
Formatted	
Formatted	(
Formatted	〔
Formatted	
Formatted	
Formatted	
Formatted	
Formatted	(
Formatted	
Formatted	〔
Formatted	(
Formatted	
Formatted	(
Formatted	
Formatted	(
Formatted	
Formatted	(
Formatted	



circuit which is essentially a mixer. Any timing jitter on the ENCODE will be combined with the desired signal and degrade the high frequency performance of the ADC. The user is advised to give commensurate thought to the clock source.

#### AD9283 Analog Input

The analog input to the ADC is fully differential and both inputs are internally biased. This allows the most flexible use of ac or dc and differential or single-ended input modes. For peak performance the inputs are biased at  $0.3 \times VD$ . See the specification table for allowable common-mode range when dc coupling the input. The inputs are also buffered to reduce the load the user needs to drive. For best dynamic performance, the impedances at AIN and AIN should be matched. The importance of this increases with sampling rate and analog input frequency. The nominal input range is 1.024 V p-p.

#### AD9283 Voltage Reference

A stable and accurate 1.25 V voltage reference is built into the AD9283 (VREF OUT). In normal operation, the internal reference is used by strapping Pins 2 and 3 of the AD9283 together. The input range can be adjusted by varying the reference voltage applied to the AD9283. No degradation in performance occurs when the reference is adjusted  $\pm$ 5%. The full-scale range of the ADC tracks reference voltage changes linearly. Whether used or not, the internal reference (Pin 2) should be bypassed with a 0.1  $\mu$ F capacitor to ground.

#### 5.1.6 FPGA Configuration

The EP4CE6 FPGA is configured for operation when the power is applied to the board. A dedicated Configuration Flash chip is included on the DSO 100M for the purpose of configuring the FPGA as power up. The <u>DSO</u> uses the second channel of the FT2232H chip as a dedicated Flash programming port. The Configuration Flash can be programmed directly from Quartus Prime by using the EPT-Blaster driver. Follow the instructions in the "EPT Drivers" section of this manual.



#### 5.1.7 FT2232H Dual Channel USB to Serial Chip

The DSO 100M contains an FTDI 2232H dual channel high speed (480 Mb/s) USB to FIFO (first in-first out) integrated circuit to interface between the Host PC and the



Formatted: Font: Bold

Formatted: Font: Bold



FPGA. The FT2232H provides a means of data conversion from USB to serial/ parallel data and serial/parallel to USB for data being sent from the FPGA to the PC. Channel A is configured as a Flash Configuration bus and Channel B is configured as an 8 bit parallel bus. FPGA Programming commands are transmitted via the channel A interface. Channel B has one dual port 4Kbyte FIFO for transmission from Host PC to the FPGA, it also has one dual port 4Kbyte FIFO for receiving data from the FPGA to the Host PC. The module uses the +5Vbus from the Host USB for self power.

# 6 FPGA Code Description

The DSO 100M comes complete with step by step instructions on building an entire communications system from FPGA to Windows Host. <u>The development of this communications system is made considerably easier because of the use of a custom software item called "EndTerms".</u>

provides an easy to use programming interface These will be explained in later sections. First, the software tools and how to install them will be reviewed. The tools required are the free Visual Studio IDE and the free Quartus Prime.



## 7.0–EPT Drivers

The communication path between DSO 100M and the PC is made possible by the use of EndTerms. These EndTerms provide a virtual "pipe" of data capable of sending and receiving bytes between the Windows PC and the FPGA. In order to fully explain the FPGA Code used in the DSO 100M, a quick explanation is needed of the transfer mechanism in the UnoProLyzer C# project.





complementary HDL Endterms in the Active Transfer Library. Users have seamless bidirectional communications at their disposal in the form of:

- Trigger Endterm
- Transfer Endterm
- Block Endterm

User code writes to the Endterms as function calls. Just include the address of the individual module (there are eight individually addressable modules of each Endterm). Immediately after writing to the selected Endterm, the value is received at the HDL Endterm in the FPGA.



## 7.0-EPT Drivers

The above is a code sample from a C# Windows Form. You can see functions that write a byte to the FPGA (EPT\_AH\_SendByte(0x01, (char)LEDStatus)) and write a trigger bit to the FPGA (EPT\_AH\_SendTrigger((byte)0x02).

The above code is the interface Verilog which resides in the FPGA. When the C# Windows form sends a byte or trigger, the signals in the FPGA code react and allow the user code to receive the byte and trigger and perform some function with the information. In the case of this example, the LEDs will change state.

Receiving data from the FPGA is made simple by Active Host. Active Host transfers data from the FPGA as soon as it is available. It stores the transferred data into circular buffer. When the transfer is complete, Active Host invokes a callback function which is registered in the users application. This callback function provides a mechanism to



transparently received atafiom the FPGA. The user application does not meet to schedule area diffiom the USB or call any blocking threads

#### 23.0 Active Transfer EndTerms FPGA Code

The Active Transfer Library is a portfolio of HDL modules that provides an easy to use yet powerful USB transfer mechanism. The user HDL code communicates with EndTerms in the form of modules. These EndTerm modules are commensurate with the Active Host EndTerms. There are three types of EndTerms in the Active Transfer Library:

- Trigger Endterm
- Transfer Endterm
- Block Endterm

They each have a simple interface that the user HDL code can use to send or receive data across the USB. Writing to an EndTerm will cause the data to immediately arrive



at the commensurate EndTerm in the Active Host/user application. The transfer through the USB is transparent. User HDL code doesn't need to set up Endpoints or respond to Host initiated data requests. The whole process is easy yet powerful.

#### 32.0 FPGA Host Interface

The FPGA code base for the DSO 100M is quite large and comprises several sections. The following sections of this document will attempt to explain the details of the code. However, a thorough examination of the code will be necessary to come to a complete understanding of the code. Also, there are several testbenches with fully simulated code to help the user. The following components make up the DSO 100M Code Base:

- Host Interface Component
- ADC Sample Storage Component
- Data Transmit Component



# 7.0-EPT Drivers

From the above block diagram, all components require the Active Transfer Library to perform their respective functions. All components communicate with the PC transparently.

The above block diagram shows the Host Interface Component only.



**7.0-EPT Drivers** The above diagram shows the Host Interface Component State Machine

**51.0 UC Controller Description** 



# 7.0-EPT Drivers



# 7.0-EPT Drivers



## 7.0-EPT Drivers

## 687 EPT Drivers

The DSO 100M requires drivers for any interaction between PC. Once the driver is installed, this will allow Windows to recognize the USB Chip and setup a pathway for Windows to communicate with the USB hardware. The UnoProLyzer application can control and receive data from the DSO 100M and the user can program the Config flash.

## 68.1<u>7.1</u> USB Driver

To install the driver onto your PC, use the EPT\_2.08.24 Folder. The installation of the EPT\_2.08.24 driver is easily accomplished using the "Update Driver Software" utility in Device Manager.

Locate the EPT\_2.08.24 folder in the Drivers folder of the DSO 100M DVD using Windows Explorer.

anize 👻 Include in library 👻 Share wit	h 🔻 Burn New folder				8∷ -	=
Products ^	Name	Date modified	Туре	Size		
Earth People Technology	Arduino-1.0.1	1/30/2013 8:29 PM	File folder			
EPT USB-CPLD Development Syste	EPT_2.08.24	2/9/2013 3:38 PM	File folder			
Arduno_IDE     Documentation	EPT_Blaster	2/9/2013 3:38 PM	File folder			
🔒 Drivers						
📕 Arduino-1.0.1						
3 EPT_2.08.24						
EPT_Blaster						
Projects_ActiveHost_64Bit						
🎉 Projects_Arduino						
Projects_HDL						

Plug in the DSO 100M device into an available USB port.





### DSO Development System User Manual

Windows will attempt to locate a driver for the USB device. When it does not find one, it will report a error, "Device driver software was not successfully installed". Ignore this error.







Locate Device Manager and click on it.

Locate the entry under "Other devices". Right click "EPT USB <->Serial&JTAG Cable" and select "Update Driver Software...".



File Action View Help	
🕨 🧄 🗊 🔛 🔛 🖬 🐼 😰 🕵 🔇	ž
HP-DeskTop-3005     Grmputer     Display adapters     DVD/CD-ROM drives     DVD/CD-ROM drives     Monitors     Monitors     NOTGEAR WG311v3 54Mbps Wirel     Revise ID/Cr GER Servise Controller	ess PCI Adapter
Other devices	r.
FPT USB <-> Senial@JTAG Cabi- FPT USB <-> Senial@JTAG Cabi- FPT USB <-> Senial@JTAG Cabi- Protescors Sont (COM & LPT) Sont (	Update Driver Software Disable Uninstall Scan for hardware changes Properties st Controller troller
USB Composite Device	troller

At the Update Driver Software Window, select "Browse my computer for driver software".



Click the Browse button and browse to the \Drivers\EPT\_2.08.24 folder of the EPT FPGA Development System DVD. Click the Ok button.



Browse for driver software on your computer	Browse For Folder
bronde for arren borthare on your compater	Select the folder that contains drivers for your hardware
Search for driver software in this location:	
C:\Users  Browse	Documentation
include subfolders	▲ ▲ Drivers ▲ Arduino-1.0.1 ▲ ▲ EPT_2.09.24 ▲ ■ md64 ▲ 1386
Let me pick from a list of device drivers on my computer This list will show installed driver software compatible with the device, and all driver software in the same category as the device.	EPT 2.08.24

Click the Next button

DIOWS	e for driver softwa	re on your comp	outer		
Search f	or driver software in this	location:			
logy\E	T USB-CPLD Developm	ent System CD\Drive	s\EPT_2.08.24 -	Browse	
🔽 Inclu	le subfolders				
→ Le	t me pick from a l	ist of device driv	ers on my com	puter ce. and all driver	
sc	ftware in the same cate	gory as the device.			

On Windows 10 PCs, the following message will appear.





The next window is the Windows Security notice. The EPT driver is not signed by Windows. How to Disable Driver Signature Verification on 64-Bit Windows 10 Windows 10 implements extra protection against malicious driver files that intend to do harm to the users PC. This implementation unfortunately locks out any third party driver file if it is unsigned. Currently all EPT boards have unsigned drivers. Follow the instructions below to allow Windows 10 to allow third party unsigned drivers to be installed on your PC.

- Under Windows 10, You must restart the computer in the "options menu" mode. This will allow you to go through the process of allowing all unsigned drivers to self install on your Windows 10.
- First, the computer must be restarted into the so-called "options menu". The easiest way to get there is via the "Run"-dialog, which is opened by means of the key combination Win+R. The command you have to enter, to boot into the Options menu is as follows:





- shutdown.exe /r /o /f /t 00 Caution: This command starts the reboot process immediately!
- You'll find the explanation of the individual parameters below:
  - shutdown.exe It's a Command-Line application which is inclusive with Windows. It does various kinds of restarts and shutdowns.
  - /r means "restart"
  - o /o means "the PC should start in the Option menu"
  - $\circ$  /f means "restart directly and close all opened programs immediately"
  - /t 00 shows the time until the restart happens (in seconds). In this case 0 seconds, which equals an immediate restart Then click on the Recovery option on the left hand side.
- After you have successfully rebooted into the "Options menu", click "Troubleshoot" and then "Advanced options".
- Now click on the "startup settings"-button and the press "reboot".
- After another reboot, you're at the startup settings page. Here you can choose between various options, which can be entered by pressing the respective number key. For our needs, you have to click option 7 "Disable driver signature enforcement". This deactivates the driver check and enables you to install unsigned drivers in Windows 10.
- In the last step and after another restart, you are able to install unsigned drivers by using Windows "Device Manager". Don't get confused. There still will be a question in the beginning, as you can see in the screenshot above, but nevertheless the installation of the driver will be possible without problems.

After these tasks are successfully completed, you can proceed with the following sections.

Windows will add the EPT\_2.08.24 driver to the System Registry.

📋 Updat	Driver Software - EPT USB <-> Serial&UTAG Cabl	e
Installin	g driver software	

When Windows has completed the update driver the following screen will be displayed.





ndows has finished installing the driver software for this device:
USB Serial Converter A

Channel A of the DSO 100M is ready for use.

Next, repeat the process for Channel B.



The driver files will automatically install in the System Registry.





When this is complete, the drivers are installed and the DSO 100M can be used with for programming and USB data transfers.

## 68.27.2 JTAG DLL Insert to Quartus Prime

The JTAG DLL Insert to Quartus Prime allows the Programmer Tool under Quartus to recognize the DSO 100M. The DSO 100M can then be selected and perform programming of the FPGA. The file, jtag\_hw\_mbftdi\_blaster.dll must be placed into the folder that hosts the jtag\_server for Quartus. This dll is available for Windows 7, 8, and 10 64-bit.

#### 68.2.17.2.1 Installing Quartus

Locate the Quartus\_Prime folder on the EPT FPGA Development System DVD. <u>The</u> Quartus software is updated twice per year. EPT will periodically upgrade the Quartus version on the DVD. So, the version of Quartus may differ between the user manual and the DVD. But, EPT tests the compatibility of DSO 100M with each version of Quartus.



Organize 🔻 🛛 🔂 Open	Share with 🔻 🛛 Burn	New fold	er				
🎉 MEGA_	USB_CPLD_PROJECT_CD	*	Name	^	Date modified	Туре	Size
UNO_A	NALOG_MONITOR_PROJECT_CD		😋 12.1sp1_24	_quartus_free_windows.exe	3/3/2013 1:12 AM	Application	3,633,187
Januar Ardui	ino_IDE						
🌗 Docu	mentation						
🍶 Drive	rs	=					
Droje	cts_ActiveHost_32Bit						
Proje	cts_ActiveHost_64Bit						
Proje	cts_Arduno						
Quart	tus_12.1sp1						
🔒 User Man	iual						
White Par	per	-					

If you don't have the EPT FPGA Development System DVD, you can download the Quartus Prime by following the directions in the Section Downloading Quartus.

If you don't need to download Quartus, double click on the Prime\_xxx\_quartus\_free\_widows.exe (the xxx is the build number of the file, it is subject to change). The Quartus Prime Lite Edition will start the installation process.

	caracting 12.1sp1_243_quartus_tree_windows/devices/web/amail.qda
D	estination folder
Q	2:\Users\NELSON~1\AppData\Local\Temp v Browse
le le	stallation progress

When the install shield window pops up click "Yes" or if needed, enter the administrator password for the users PC. Click "Ok"

Next, skip down to the Quartus Installer section to complete the Quartus installation.

### 68.2.2<u>7.2.2</u> Downloading Quartus

The first thing to do in order build a project in Quartus is to download and install the application. You can find the latest version of Quartus at:

https://www.altera.com/download/dnl_	
index.jsphttps://fpgasoftware.intel.com/?edition=pro	

Page 56 Field Code Changed



Click on the Download Windows Version.



The next page will require you to sign into your "<u>Intel FPGA</u>" account. If you do not have one, follow the directions under the box, "Don't have an account?"



MEASURABLE ADVANTAGE *	rvices Send Markets Stechnology Straining St	Search ← Q winload Center P Documentation ≜ myAltera Account Support ≥ About ≥ Buy
wryAlfena View myAlfans Home Manage Postie Change Posterord Lagout Heijp myAlfena Account Help Terms and Conditions	myAltera Account Sign In Nore > myAltera Account Sign In User Name Passwerd Emember me Sign in	Forget Your User Name or Password?
	Den't have a Create Your myAltera Account Your myAltera account allows you fine service rec Enter your email address. (If your email address already exists in our system w Creater Account)	an account? puest, register for a class, download suftware, and more, e will rebrieve the associated information.)
oducts Solutions Support	intel	USA (English) 🌐 Sign In
Intel® FPGA Program		
Sign In		Register
Sign In Intel Employee? Sign in here		Register If you don't already have access to the Intel® FPGA Program, select the option that best meets your needs.
Sign In Intel Employee? Sign in here Intel Customer or Partner?		Register If you don't already have access to the Intel® FPGA Program, select the option that best meets your needs. Register now for an individual account
Sign In Intel Employee? Sign in here Intel Customer or Partner? Username		Register If you don't already have access to the Intel® FPGA Program, select the option that best meets your needs. Register now for an individual account If you do business with Intel® as an individual, you can request for basic access to Intel® FPGA Program tools ar resources.
Sign In Intel Employee? Sign in here Intel Customer or Partner? Username Password Passed		Register If you don't already have access to the Intel® FPGA Program, select the option that best meets your needs. Register now for an individual account If you do business with Intel as an individual, you can request for basic access to Intel® FPGA Program tools an resources. Register now for a premier account
Sign In Intel Employee? Sign in here Intel Customer or Partner? Username Password By aging in you agree to our Terms of Service		Register         If you don't already have access to the intel® FPGA Program, select the option that best meets your needs.         Register now for an individual account         If you do business with Intel as an individual, you can request for basic access to Intel® FPGA Program tools an resources.         Register now for a premier account         If you do business with Intel on behaft of a company, you

Once you have created your Intel FPGA account, enter the User Name and Password. The next window will ask you to allow pop ups so that the file download can proceed.



	40 Search	- 9
EASURABLE ADVANTAGE"	Download Center Documentation	ra / Logout
Devices Section Tools & Section 2015	ervices & End Markets & Technology & Training & Support & About & Buy	
esign Software Upartua II Subscription Edition IngaCute P Ubary IngaCute P	Thank you for downloading         Mar Super's Downlad's * Their per downlad!         And Control Co	stallation
able & Adapter Drivers pard Layout and Test SDL Models PICE Models	While file from the intermet can be used, that like type can provide the may our compared. Hyou do not but the source, do not run or serve this software. <u>What is the mix?</u>	
BIS Models ichematic Review Worksheets ayout Review Worksheets iadence PCB Libraries fentor Graphics PCB Libraries Jerber Files	Uowinosal irouaiesmooting steps 1. Duale yrop-up blocker • If a yellow por up blocker bar appears just below your freternet Explorer menu bar (see Figure 2 the yellow bar and select "Immorrhy Allow Pop-ups." Any third-party pop-up blockers must als disable to dowinolad a File. Finance 2. Worksome Finance Rhou-Lin Machane.	), click on o be
egacy Software IAX+PLUS II	Pop-up blocked. To see this pop-up or additional options click here	

Click on the "Allow Once" button. The next window will appear. It is the Download Manager.

AND THE RA.	
About the Download Manager	List resour
The Download Manager provides a more effective, more efficie computer of and on again. After you initiate a download by clic install and begin to download the requested file.	nt way to download thes than you normally see with a browser, especially for large files or file sets. It can pause and restart downloads even if you tarn your sing the link below, a security warning appears. From the security warning window, click instal to continue downloading your file. The Download Manager with
Click to Download Your File Now	
If the Download Manager fails to start, or if you do not accept th	e Security certificate, you can download the file without using the Download Manager
ſ	Internet Explorer Security
ALTERA, ARRIA, CYC	A website wants to open web content using this     program on your computer
	The program will appen autides of Protected mode, Internet Explane's <u> hypototect mode</u> relate protect your company. If you do not hout the weblie, do not spon this program.
	Iame: Deveload Hanager     Noterier: Alassa Technologies Inc.
	Do not show me the warring for this program again
	( How ) Doo't allow

Click on the "Allow" button. This will bring up the Save As dialog box. Save the altera\_installer.external.exe to a download file.

Organize      New folder				)EE •	
Recent Places   Libraries   Libraries   Construction   Music   Placements   Music  Places    Longroup   Romgroup   Romgroup    Romgroup     Compare		Hame 11.102_239,quarks, free, sindows.ee 102.1020_232,quarks, free, sindows.ee 102.1237,quarks, free, sindows.ee 102.1230_232,quarks, free, sindows.ee 102.1230_232,quark	Date modified 6/15/2012 318 PM 8/11/2012 11-20 AM 11/25/2012 10-54 3/3/2013 11-24 AM 5/20/2012 8:52 PM	Type Application Application Application Application	5izv 2,5i 4,0: 3,6:
File name:	- rmal ace	(e.)			1
Save as type: ".exe					•





\_\_\_\_\_v2.2.6

🜗 E<u>x</u>it

Click the Save button. This will start the Download Manager.



💐 Launch

Click "Ok" and "Yes" to the following screen.

## 68.2.37.2.3 Quartus Installer

Click "Next" on the Introduction Window.



Click the checkbox to agree to the license terms. Then click "Next".





To continue installing the sol	ftware, you must agree to the terms of the software license agreement below.	
ADERA.	This document contrast these separatil locarity appendix multi- separative provides the separative provides and the separative pro- lation of the separative provides and the separative pro- pagements. Accessing of the terms and contrastive of the document of they were individually and associately accessed. The terms and concent as device the set of the set of the set of the set of the set of the document of the set of the set of the set of the set of the document of device the set of the set of the set of the set of the document of device the set of the set of the set of the set of the document of device the set of the set of the set of the set of the document of device the set of the set of the set of the set of the document of device the document.	E
<ul> <li>Introduction</li> <li>License</li> <li>Installer Setup</li> <li>Select Destination</li> <li>Select Products</li> <li>Summary</li> </ul>	The following providen toplots to all three learning appreciated below: All and my low you data and direction in a own nervers or on anyones housed by their parts. If Alters a pointing any shown is to you for use the three, we can be obtained in you do be information, data or software housed by a the parts. All bounds every resonable where the amade to keep add and information ones. All and a strain the amade to keep add and information ones. Addoeded by the thild party of for any system downthme related to the third part years. POR QUARTUS IL VERSION 12.1, ALL DISTERUITIONS' (VEB DOWNLOAD, CDS, DVDS).	
> Installation	ALTERA PROGRAM LICENSE SUBSCRIPTION AGREEMENT	
	I agree to the terms of the icense agreement.	

Click "Next" and accept the defaults.

	Specify the Installation Files Directory	
	Select this option if you want the Altera Installer to disk or local network.	retrieve installation files from your hard
	Installer Source Directory:	
		Browse
	Download Installation Files from the Internet	
	Select this option if you want the Altera Installer to Internet.	retrieve installation files from the
	Courte have an an annual of the the television	
	specity now you are connected to the internet:	
> License	<ul> <li>No proxy (recommended)</li> </ul>	
> License > Installer Setup	No proxy (recommended)     Manual proxy configuration	
> license > Installer Setup > Select Destination	Specify now you are connected to the internet:     No proxy (recommended)     Manual proxy configuration     Web Proxy:	Portz
> Installer Setup > Select Destination > Select Products	Setury now you are connected to the internet: Nov (recommended) Nanual proxy configuration Web Proxy: Usemame:	Portz
> Incense > Installer Setup > Select Destination > Select Products	Security may face are connected to the antennect Ne ho proxy (recommended) Manual proxy configuration Web Proxy: Usemanic Password;	Portz
> Installer Setup > Select Destination > Select Products > Summary	Security more plot are connected to the antennect Ne No proxy (recommended) Meanual proxy configuration Web Proxy: Usemannec Paseword:	Port:

Click "Next" to accept the defaults



Sector Sector     Sector Sector Sector     Sector Sector Sector     Sector Sector Sector     Sector Sector Sector     Sector	Altera Installer (12.1 Build 177)	and the second se	
Petitization Directory:     Control     Control     Control     Control     Control     Control     Petitization Directory:     Control     Control     Control     Control     Control     Control     Control     Petitization Directory:     Control     Control <th>Select Destination Specify the software destination</th> <th>ition directory.</th> <th></th>	Select Destination Specify the software destination	ition directory.	
citateralization     Biologicality       citateralization     Biologicality       Available Space:     Biologicality       available Space:     Biologicality       > Introduction     Biologicality       > Introduction     Biologicality       > Constaller Setup     Biologicality       > Select Destination     Accessions       > Summary     Accessions       > Summary     Biologicality       > Installation     Biologicality		Destination Directory:	
A valable Space: 000 G  Tencoary Description for statisticon files:  A valable Space: 000 G  Tencoary Description for statisticon files:  A valable Space: 000 G  Tencoary Description for statisticon files:  Description for the statisticon files:  Description for the statisticon files:  A valable Space: 000 G  Tencoary Description for the statisticon files:  A valable Space: 000 G  Tencoary Description for the statisticon files:  A valable Space: 000 G  Tencoary Description for the statisticon files:  A valable Space: 000 G  Tencoary Description for the statisticon files:  A valable Space: 000 G  Tencoary Description for the statisticon files:  A valable Space: 000 G  Tencoary Description for the statisticon files:  A valable Space: 000 G  Tencoary Description for the statisticon files:  A valable Space: 000 G  Tencoary Description for the statisticon files:  A valable Space: 000 G  Tencoary Description for the statisticon files:  A valable Space: 000 G  Tencoary Description for the statisticon files:  A valable Space: 000 G  Tencoary Description for the statisticon files:  A valable Space: 000 G  Tencoary Description for the statisticon files:  A valable Space: 000 G  Tencoary Description for the statisticon files:  A valable Space: 000 G  Tencoary Description for the statisticon for the valable of the va		c:\altera\12.1	Browse
Terporary Directory for instalation fliss:       CUlterry(MS: SOLO-1)-Option Local Temp       Bit Instalation       > Introduction       > Introduction       > Compare failed       > Compare failed       > Select Destination       > Select Destination       > Select Products       > Sommary       > Installation       > Select Destination		Available Space:	800 G
ClustryHESCH-14pobralLocalTemp  CutaryHESCH-14pobralLocalTemp  Available Space: Available Space: CutaryHESCH-14pobralLocalTemp  Available Space: CutaryHESCH-14pobralLocalTemp  Available Space: CutaryHESCH-14pobralLocalTemp  CutaryHESCH-14pobralLoc		Temporary Directory for installation files:	
Arabbé Sarce:     000 G       > Introduction     Bornisation files after completion.       > Introduction     Bornisation files after completion.       > Uconse     Specify the Plogram Fider:       > Installer Setup     Extern 121. Biol 177 (Cory 2)       > Select Destination     Accessorie Alars 121. Biol 270       > Select Destination     Accessorie Alars 121. Biol 270       > Summary     Alars 121. Biol 270       > Installation     Barne & Hoole Barne & Hoole Barne & Hoole	/////	C:\Users\NELSON~1\AppData\Local\Temp	Browse
		Available Space:	800 G
I throduction  Download only and install lites  Dogram Folder  Software Setup  Download only and install lites  Dogram Folder  Advana 12.1 Build 177 (Goy 2)  Download only and install lites  Advana 12.1 Build 177 (Goy 2)  Download only and install lites  Advana 12.1 Build 177 (Goy 2)  Advana 12.1 Build 177  Advana 12.		Remove saved installation files after completion.	
> Intraduction       Program Folder:         > LiCense       Specify the Program Folder:         > Installer Setup       Extring Folder:         > Select Destination       Accessorias         > Select Products       April 121. Bild 177 (Copy 2)         > Select Products       April 122. Bild 177         > Summary       April 121. Bild 243         > Installation       Berne & Holde         > Installation       Berne & Holde		Download only and install later.	
> License     Speed (He Pagam Folder:       > Installer Setup     Data 12.1 Biol 177 (Soy 2)       > Dataller Setup     Data 177 (Soy 2)       > Select Destination     Accessories       > Select Products     Alara       > Select Products     Alara       > Summary     Alarbid 423       > Installation     Barner & Holde       > Installation     Barner & Holde		Brogram Folder	
Installer Setup     Acter 12.1 Bid 317 (Goy 2)       Extra proder:     Extra proder:       > Select Destination     Accession       > Select Destination     Accession       > Select Products     Ahra       Alara     Ahra       > Select Products     Ahra       > Summary     Aperior       > Installation     Barres & hole       Barres & hole     Installation		Specify the Program Folder:	
Singular Setup     Easting Folders:     Accessions     Access		Altera 12.1 Build 177 (Copy 2)	
> Select Destination         Accessors         *           > Select Products         Alara         #           > Select Products         Alara         #           > Select Products         Alara         #           > Summary         Alara (Select Products)         #           > Summary         Alar (Select Products)         #           > Installation         Barnes & Hobe         #		Existing Folders:	
Select Products     Administrate Tools     Area     Akera	> Select Destination	Accessories	*
Select Products     Advard212.1864 177     Advard212.1249E Biold 723     Summary     AND VISION Engine Control Center     AugueCol     Simstallation     Bother     Bother     Chandels.http://chandels.exee.		Administrative Tools	=
Summary     Alter 12.101 bild 33     Summary     Alter 12.101 bild 34     ViSIOI Egne Control Center     AUPEOI     Subscription     Brocker     Brocker     Control Center     Automatication		Altera 12.1 Build 177	
> Summary A&O VEDR Topic Control Center A		Altera 12.1sp1 Build 243	
> Installation Barner & Hoble Bother End works the discussion		AMD VISION Engine Control Center	
Bother +		Barnes & Noble	
		Brother Ochschel Made Comment	-
- Dark Hanks Conned		- Outh	Next a

At the Select Products Window, de-select the Quartus Prime Supbscription Edition by clicking on its check box so that the box is not checked. Then click on the check box by the Quartus Prime Web Edition (Free).

4.9 G         1.4 G           986 M         150 M           8.1 G         3.8 G           986 M         150 M           2.5 G         2.4 G           3.1 G         491 M
986 M 150 M 8.1 G 3.8 ( 986 M 150 M 2.5 G 2.4 ( 3.1 G 491 M 3.1 G 491 M
8.1 G         3.8 0           986 M         150 M           2.5 G         2.4 0           3.1 G         491 M           3.1 G         491 M
986 M 150 M 2.5 G 2.4 G 3.1 G 491 M 3.1 G 491 M
2.5 G 2.4 G 3.1 G 491 M 3.1 G 491 M
3.1 G 491 M 3.1 G 491 M
3.1 G 491 M
267 M 48 I
b Edition software is final software that the back the ba
b Edition software is tion software that ree Nios II
b Edition software is tion software that free Nios II

Click "Next" to accept the defaults



Installation Summa	iry
Review the install	lation summary to verify your software installation options, and then click Next to begin the installation.
	C:\altera\12.1
	Cache Location: C:\Users\NELSON~1\AppData\Local\Temp\atera_12.1_177
7-01010	Program Folder: Abox 13 1 Build 177 (Conv. 2)
	Selected Products:
-	Quartus II Web Edition (Free) (includes Nios II EDS)
> Introduction	Quartus II vole Editori (rite) (includes into II Edit)     Quartus II software (64-bit)     Aria II GX
> License	Cyclone II     Cyclone III/III LS
> Installer Setup	Cyclone IV E     Cyclone IV GX
. Calant Dantiantian	Cyclone V     Legacy Families
> Select Destination	• MAX II • MAX V
> Select Products	ModelSim-Altera Starter Edition (Free)
> Summary	
> Installation	
accept the d	cencel     efaults
accept the d	Akera software is instaled.      Ownhoading Quartus II Web Edition (Free) (includes Nos II EDS)
Altera Installer (12.1 Build 1 Altera Installer (12.1 Build 1 Installation Please wait while the	
Attera Installer (12.1 Build 1 Materia Installer (12.1 Build 1 Distallation Please wait while the	Alter software is installed.      Products      Produ
Atters Installer (12.1 Build 3 Please wait while the	Reak Hest> Concel  efaults  IT77  Altera software is instaled.  Products  Quartus II Web Editon (Free) (includes Nos II EDS)  Quartus II Web Editon (Free) (includes Nos II EDS)  Quartus II Web Editon (Free) (includes Nos II EDS)
Atters Installer (12.1 Build 3 Please wait while the	Concel      C
ALDERVAL	(Next) Test
Attera Installer (121 build J Merzin Attera Installer (121 build J Please wat while the	
Attera Installer (12.1 build st Attera Installer (12.1 build st Dease wat while the Please wat	Reack Hest Concel  efaults  IV77  Altera software is instaled.   Products  Products Products  Products  Products Products
Accept the de Attra Installer (12.1 Build of Installion Please wat white the Accept the def Please wat white the Accept the def Please wat white the Please wat	
Accept the d	
Accept the de Attera finitalier (12.1 build 13 Dease wait while the Please wait while the Installer Setup > Select Destination	Redx     Redx
Accept the d Attra Installer (12.1 build 3 Dease wait while the Please wait while the Accept the dealer of the dealer Installer Setup Select Destination Select Product	
Atters Installer (12.1 build st Please wait while the Please wait	
Accept the d Attra Installer (121 bild J Please wat while the Please wat while the Control of the second Installer Setup > Select Destination > Select Products > Summary	
Accept the de Attraction (121 bird of Prese wat while the Prese wat while the Prese wat while the Control of the traction > License > Installer Setup > Select Products > Summary > Installation	

Wait for the download to complete. The file is 3.5 GB, so this could take a couple of hours depending on your internet connection. When installation is complete, the following window appears.



🎨 Altera Installer (12.1 Build 177)			_ = ×
Installation Please wait while the Altera :	software is installed.		
l	Installation successful.		100%
ADERA.	Products Quartus II Web Edition (Free) (includes Nios II EDS) Quartus II Web Edition (Free) (includes Nios II EDS) Quartus II software (64-bit) Arria II GX	Downloaded	Installed
> Introduction	tera Installer The Altera Installer finished installing the "Quartus II Web Edition (Free) (ncludes Nios II EDS)	T.	****
> License > Installer Setup	* ModelSim-Altera Starter Edition (Free) software version 12.1.		
> Select Destination	Start Quartus II 12.1		
> Select Products	Create Desktop Shortcuts		
> Installation	Quartus II Web Edition (Free) and Nios II SBT / Nios II ModelSim-Altera Starter Edition (Free)	GCC4 Toolchai	n
	Rate your installation experience		

Click "Ok", then click "Finish". The Quartus Prime is now installed and ready to be used.

#### 68.2.47.2.4 Adding the EPT\_Blaster to Quartus Prime

Close out the Quartus Prime application. Locate the Drivers EPT\_Blaster folder on the DSO 100M DVD.



Follow these directions:

- 1. Open the C: \EPT FPGA Development System DVD \Drivers \EPT\_Blaster x64 folder.
- 2. Select the file "jtag\_hw\_mbftdi\_blaster.dll" and copy it.
- 3. Browse over to  $\underline{C:\intelFPGA\_lite}xx.x\quartus\bin64.$





- 4. Right click in the folder and select Paste
- 5. Click Ok.
- 6. Open the Quartus Prime application.

ize - B Open with Burn Ne	w folder				
🔒 ip	(*)	Name	Date modified	Type	Size
🎍 nios2eds		jam2.jam	11/7/2012 8:39 PM	JAM File	265 83
📕 quertus		jam2 ic.jam	11/7/2012 8:39 PM	JAM File	264 KB
🍌 bin	1.00	itag_atlantic.dll	11/7/2012 8:39 PM	Application extens	20 KB
🗼 bin64	1	itag_client.dll	11/7/2012 8:38 PM	Application extens	205 KE
🍶 common	1.0	(%) jtag_hw_mbftdi_blaster.dll	1/16/2013 10:37 PM	Application extens	67 KE
🌲 cusp		itag_hw_pli-blaster.dll	11/7/2012 8:39 PM	Application extens	33 KB
🎳 drivers		jtag_hw_usb-blaster.dll	11/7/2012 8:39 PM	Application extens	17 KE
dsp_builder		itag_hw_virtual_chain.dll	11/7/2012 8:38 PM	Application extens	34 KB
🗼 eda		jtag_pli-blaster_vpi.dll	11/7/2012 8:39 PM	Application extens	41 KE
🇼 extlibs32		itagconfig.exe	11/7/2012 8:39 PM	Application	35 KB
🗼 libraries		itagserver.exe	11/7/2012 8:39 PM	Application	269 KE
🗼 lenf		ibizma.dll	11/7/2012 6:57 PM	Application extens	128 KE
🌲 qdesigns		Imgrd.exe	12/8/2005 4:30 PM	Application	1,147 KE
🍶 sopc_builder		Imtools.exe	12/8/2005 5:33 PM	Application	1,060 KE
Hig altera inspector log zin		I Inutil eve	12/8/2005 4-30 PM	Application	1 112 82

The DLL is installed and the JTAG server should recognize it. Go to the section "Programming the FPGA" of this manual for testing of the programming. If the driver is not found in the Programmer Tool->Hardware Setup box, see the JTAG DLL Insert to Quartus Prime Troubleshooting Guide.

## 68.37.3 Active Host Application DLL

Download the latest version of Microsoft Visual C# Express environment from Microsoft. It's a free download.

https://visualstudio.microsoft.com/vs/express/

Go to the website and click on the "+" icon next to the Visual C# Express. Please note that Microsoft updates the Visual Studio each year. So, the version number in the user manual will not reflect the same version number as the latest Visual Studio version number. Also, the Visual C# Express and Visual Studio Express are used interchangeably. The Visual Studio Express includes the Visual C# Express. The user can download either Visual C# Express or Visual Studio Express. Each version of Visual Studio Express is tested shortly after its release with the DSO 100M.









Click "Next", accept the license agreement. Click "Next".



Visual C# 2010 Express will install. This may take up to twenty minutes depending on your internet connection.

Setup complete	Visual C# 2010 Express
Microsoft Visual C# 2010 Express has been	installed successfully.
<ol> <li>Visit <u>Microsoft Update</u> to download the latest service</li> </ol>	packs and security updates.

The installed successfully window will be displayed when Visual C# Express is ready to use.

## **FPGA Active Transfer Library**

To use the Active Host Application Software, the Active Host DLL and the ftd2xx DLL must be included in the Microsoft Visual project. The Active Host Application Software will allow the user to create a custom applications on the PC using the EndTerms to perform Triggers and Data Transfer to/from the DSO 100M. The methods and parameters of the Active Host DLL are explained in the Active Host Application section. Locate the \Projects\_ActiveHost\_64Bit and \Projects\_ActiveHost\_32Bit folders on the EPT FPGA Development System DVD.



# **FPGA Active Transfer Library**

Locate the Projects\_ActiveHost\_64Bit folders in the EPT FPGA Development System using Windows Explorer.

Locate the Projects\_ActiveHost\_64Bit \ActiveHost\_1.0.0.11\Bin folder and copy the ActiveHost64.dll and the ftd2xx64.dll.

Save the DLL's in the bin\x64\Release folder of the user project under the Microsoft C# Express project. See the Active Host Application section of the EPT FPGA Development System User Manuals for instructions on how to add the dll to the Microsoft C# Express project.



## **FPGA Active Transfer Library**

At this point, all the software and drivers should be loaded on the users PC. Before, we get into the description of how the software and FPGA code works,



A description of the how the EndTerms functions between PC and FPGA is given. The Active Transfer Library is an HDL library designed to transfer data to and from the DSO 100M via High Speed (480 MB/s) USB. It is a set of pre-compiled HDL files that the user will add to their project before building it. The description of what the library does and how to use its components are described in this manual.







from PC application code through the USB driver to the user FPGA code. The user code connects to the Endterms in the Active Host dll. These Host Endterms have complementary HDL Endterms in the Active Transfer Library. Users have seamless bidirectional communications at their disposal in the form of:

- Trigger Endterm
- Transfer Endterm
- Block Endterm

User code writes to the Endterms as function calls. Just include the address of the individual module (there are eight individually addressable modules of each Endterm). Immediately after writing to the selected Endterm, the value is received at the HDL Endterm in the FPGA.

The EndTerm channels allow unique data sharing to be dedicated on a single channel.





By allowing communications on multiple channels that can be a single eight bit word (Transfer) up to 256 eight bit words (Block), the system can be flexible enough to handle any number of different size packets. And the parser becomes a simple switch statement. This flexible system also allows a simplified User API. To initiate a Trigger, the user calls EPT AH SendTrigger(). Then, two bytes are sent:

- Command Byte
- Payload Byte



71



The Trigger EndTerm is meant to provide 255 single point events. The Transfer and Block EndTerms are meant to transfer data.

## 79.1<u>8.1</u> EPT Active Transfer System Overview

The Active Transfer System components consist of the following:

- active\_transfer\_library.v
- ft\_245\_state\_machine.v
- endpoint\_registers.vqm
- active\_trigger.vqm
- active\_transfer.vqm
- active\_block.vqm

The Active\_Transfer\_Library provides the communication to the USB hardware. While separate Input and Output buses provide bi-directional communications with the plug in modules. See Figure 6 for an overview of the EPT Active\_Transfer system.

Figure 6 EPT Active Transfer Library Overview



Figure 6 shows how the modules of the EPT Active Transfer Library attach to the overall user project. The EPT Active\_Transfer\_Library.vqm, Active\_Trigger.vqm, Active\_Transfer.vqm and Active\_Block.vqm modules are instantiated in the top level of the user project. The User\_Code.v module is also instantiated in the top level. The Active\_Transfer modules communicate with the User\_Code through module parameters. Each module is a bi-directional component that facilitates data transfer from PC to FPGA. The user code can send a transfer to the Host, and the Host can send a transfer to the user code. This provides significant control for both data transfers and signaling from the user code to PC. The Triggers are used to send momentary signals that can turn on (or off) functions in user code or PC. The Active Transfer is used to send a single byte. And the Active Block is used to send a block of data. The


Active\_Transfer and Active\_Block modules have addressing built into them. This means the user can declare up to 8 individual instantiations of Active\_Transfer or Active\_Block, and send/receive data to each module separately.

### 79.2<u>8.2</u> Active Transfer Library

The Active Transfer Library contains the command, control, and data transfer mechanism that allows users to quickly build powerful communication schemes in the FPGA. Coupled with the Active Host application on the PC, this tools allows users to focus on creating programmable logic applications and not have to become distracted by USB Host drivers and timing issues. The Active Transfer Library is pre-compiled file that the user will include in the project files.





_//T		*******		**********************************				
//#								
//#	Copyright	Earth Pe	eople Techno	logy Inc. 2015				
//#								
//#								
//#	File Name:	EPT_4CE6	5_AF_D1_Top.	V				
//#	Author:	Earth Pe	eople Techno	logy				
//#	Date:	October	6, 2015					
//#	Revision:	А						
//#								
//#	Development	: EPT Dat	ta Collectio	n Project				
//#	Application	: Altera	Cyclone IV	FPGA				
//#	Description	: This fi	ile contains	verilog code which will allow access				
//#		to Acti	ive Transfer	Library.				
//#								
//#								
//#								
//#	********	*******	*********	***************************************				
//#								
//#	Revision His	story:						
//#	DAT	E	VERSION	DETAILS				
//#	10/	6/15	1	Created				
//#								
//#								
//#								
-//#	**********	*******	**********	******				



module EPT\_4CE6\_AF\_D1\_Top (

input wire input wire output wire inout wire	[1:0] [1:0] [2:0] [7:0]	aa, bc_in, bc_out, bd_inout,	
	17.01	VTO 1	
output wire	[/:0]	X10_1,	//X10 D2-D9
output wire	[2:0]	XIO_2,	//XIO D10-D12
input wire	[4:0]	XIO_2_IN,	//XIO D14-D18
output wire	[7:0]	XIO_3,	//XIO D22-D29
output wire	[7:0]	XIO_4,	//XIO D30-D37
output wire	[7:0]	XIO_5,	//XIO D38-D45
output wire	[7:0]	XIO_6,	//XIO D46-D53
input wire	[5:0]	XIO_7,	//XIO D69,D70,D71,D74,D75,D76

The interface from the library to the user code is two uni directional buses, UC\_IN[22:0] and UC\_OUT[20:0]. The UC\_IN[22:0] bus is an output bus (from the library, input bus to the Active Modules) that is used channel data, address, length and control information to the Active Modules. The UC\_OUT[21:0] bus is an input bus (to the library, output bus from the Active Modules) that is used to communicate data, address, length, and control information to the Active Modules.

The control buses, aa[1:0], bc\_in[1:0], bc\_out[2:0], and bd\_inout[7:0] are used to channel data, and control signals to the USB interface chip. These signals are connected directly to input and output pins of the FPGA.

#### 79.2.18.2.1 Active Trigger EndTerm

The Active Trigger has eight individual self resetting, active high, signals. These signals are used to send a momentary turn on/off command to Host/User code. The Active Trigger is not addressable so the module will be instantiated only once in the top level.



```
743
       wire [22*3-1:0] uc_out_m;
744
       eptWireOR # (.N(3)) wireOR (UC OUT, uc out m);
745
                                        ACTIVE TRIGGER INST
           active_trigger
746
            (
747
                                        (CLK_66),
             .uc_clk
748
             .uc_reset
                                        (RST),
749
             .uc_in
                                        (UC_IN),
750
             .uc_out
                                        (uc_out_m[ 0*22 +: 22 ]),
751
752
             .trigger_to_host
                                        (trigger_to_host),
753
             .trigger_to_device
                                        (trigger_in_byte)
754
755
           );
756
```

To send a trigger, decide which bit (or multiple bits) of the eight bits you want to send the trigger on. Then, set that bit (or bits) high. The Active Transfer Library will send a high on that trigger bit for one clock cycle (66 MHz), then reset itself to zero. The bit can stay high on the user code and does not need to be reset to zero. However, if the user sends another trigger using the trigger byte, then any bit that is set high will cause a trigger to occur on the Host side.





So, care should be used if the user code uses byte masks to send triggers. It is best to set only the trigger bits needed for a given time when sending triggers.

The user code must be setup to receive triggers from the Host. This can be done by using an asynchronous always block. Whenever a change occurs on a particular trigger bit (or bits), a conditional branch can detect if the trigger bit is for that block of code. Then, execute some code based on that trigger.



	H	//
309		// Detect Trigger In
310	L	//
311		<pre>always @(trigger_in_byte or trigger_in_reset or reset)</pre>
312	Ę	begin
313		if(!reset)
314	白	begin
315		<pre>trigger_in_detect = 1'b0;</pre>
316	-	end
317		<pre>else if (trigger_in_reset)</pre>
318	白	begin
319		<pre>trigger_in_detect = 1'b0;</pre>
320	F	end
321		<pre>else if (trigger_in_byte &gt; 8'h0)</pre>
322	白	begin
323		<pre>trigger_in_detect = 1'b1;</pre>
324	-	end
325	L	end
326		
327	曱	//
328		<pre>// Store the value of Trigger In</pre>
329	L	//
330		always @(posedge CLK_66 or negedge reset)
331	F	begin
332		if(!reset)
222		begin
333		bogin
334		<pre>trigger_in_store &lt;= 8'h0f;</pre>
334 335		<pre>trigger_in_store &lt;= 8'h0f; trigger_in_reg &lt;= 1'b0;</pre>
334 335 336		<pre>trigger_in_store &lt;= 8'h0f; trigger_in_reg &lt;= 1'b0; trigger_in_reset &lt;= 1'b0;</pre>
334 335 336 337	-	<pre>trigger_in_store &lt;= 8'h0f; trigger_in_reg &lt;= 1'b0; trigger_in_reset &lt;= 1'b0; end</pre>
334 335 336 337 338		<pre>bogin trigger_in_store &lt;= 8'h0f; trigger_in_reg &lt;= 1'b0; trigger_in_reset &lt;= 1'b0; end else if (trigger_in_detect &amp; !trigger_in_reg) </pre>
334 335 336 337 338 339		<pre>bogin trigger_in_store &lt;= 8'h0f; trigger_in_reg &lt;= 1'b0; trigger_in_reset &lt;= 1'b0; end else if (trigger_in_detect &amp; !trigger_in_reg) begin if (trigger_in_bute_!= 0)</pre>
333 335 336 337 338 339 340 341		<pre>bogin trigger_in_store &lt;= 8'h0f; trigger_in_reg &lt;= 1'b0; trigger_in_reset &lt;= 1'b0; end else if (trigger_in_detect &amp; !trigger_in_reg) begin if (trigger_in_byte != 0) trigger_in_store[7:0] &lt;= trigger_in_bite[7:0];</pre>
333 334 335 336 337 338 339 340 341 242		<pre>bogin trigger_in_store &lt;= 8'h0f; trigger_in_reg &lt;= 1'b0; trigger_in_reset &lt;= 1'b0; end else if (trigger_in_detect &amp; !trigger_in_reg) begin if (trigger_in_byte != 0) trigger_in_store[7:0] &lt;= trigger_in_byte[7:0]; trigger_in_reg &lt;= 1!b1;</pre>
333 334 335 336 337 338 339 340 341 342 343		<pre>bogin trigger_in_store &lt;= 8'h0f; trigger_in_reg &lt;= 1'b0; trigger_in_reset &lt;= 1'b0; end else if (trigger_in_detect &amp; !trigger_in_reg) begin if (trigger_in_byte != 0) trigger_in_store[7:0] &lt;= trigger_in_byte[7:0]; trigger_in_reg &lt;= 1'b1; end</pre>
333 335 336 337 338 339 340 341 342 343 344		<pre>bogin trigger_in_store &lt;= 8'h0f; trigger_in_reg &lt;= 1'b0; trigger_in_reset &lt;= 1'b0; end else if (trigger_in_detect &amp; !trigger_in_reg) begin if (trigger_in_byte != 0) trigger_in_store[7:0] &lt;= trigger_in_byte[7:0]; trigger_in_reg &lt;= 1'b1; end else if (trigger_in_reg)</pre>
333 335 336 337 338 339 340 341 342 343 344 345		<pre>bogin trigger_in_store &lt;= 8'h0f; trigger_in_reg &lt;= 1'b0; trigger_in_reset &lt;= 1'b0; end else if (trigger_in_detect &amp; !trigger_in_reg) begin if (trigger_in_byte != 0) trigger_in_store[7:0] &lt;= trigger_in_byte[7:0]; trigger_in_reg &lt;= 1'b1; end else if (trigger_in_reg) begin</pre>
333 335 336 337 338 339 340 341 342 343 344 345 346		<pre>bogin trigger_in_store &lt;= 8'h0f; trigger_in_reg &lt;= 1'b0; trigger_in_reset &lt;= 1'b0; end else if (trigger_in_detect &amp; !trigger_in_reg) begin if(trigger_in_byte != 0) trigger_in_store[7:0] &lt;= trigger_in_byte[7:0]; trigger_in_reg &lt;= 1'b1; end else if (trigger_in_reg) begin trigger_in_reg &lt;= 1'b0;</pre>
333 334 335 336 337 338 339 340 341 342 343 344 345 346 347		<pre>bogin trigger_in_store &lt;= 8'h0f; trigger_in_reg &lt;= 1'b0; trigger_in_reset &lt;= 1'b0; end else if (trigger_in_detect &amp; !trigger_in_reg) begin if(trigger_in_byte != 0) trigger_in_store[7:0] &lt;= trigger_in_byte[7:0]; trigger_in_reg &lt;= 1'b1; end else if (trigger_in_reg) begin trigger_in_reg &lt;= 1'b0; trigger_in_reset &lt;= 1'b1;</pre>
333 334 335 336 337 338 339 340 341 342 343 344 345 344 345 346 347 348		<pre>bogin trigger_in_store &lt;= 8'h0f; trigger_in_reg &lt;= 1'b0; trigger_in_reset &lt;= 1'b0; end else if (trigger_in_detect &amp; !trigger_in_reg) begin if(trigger_in_byte != 0) trigger_in_store[7:0] &lt;= trigger_in_byte[7:0]; trigger_in_reg &lt;= 1'b1; end else if (trigger_in_reg) begin trigger_in_reg &lt;= 1'b0; trigger_in_reset &lt;= 1'b1; end</pre>
333 334 335 336 337 338 339 340 341 342 343 344 345 344 345 346 347 348 349		<pre>bogin trigger_in_store &lt;= 8'h0f; trigger_in_reg &lt;= 1'b0; trigger_in_reset &lt;= 1'b0; end else if (trigger_in_detect &amp; !trigger_in_reg) begin if(trigger_in_byte != 0) trigger_in_store[7:0] &lt;= trigger_in_byte[7:0]; trigger_in_reg &lt;= 1'b1; end else if (trigger_in_reg) begin trigger_in_reset &lt;= 1'b0; trigger_in_reset &lt;= 1'b1; end else if (!trigger in detect)</pre>
333 334 335 336 337 338 339 340 341 342 343 344 345 344 345 346 347 348 349 350		<pre>bogin trigger_in_store &lt;= 8'h0f; trigger_in_reg &lt;= 1'b0; trigger_in_reset &lt;= 1'b0; end else if (trigger_in_detect &amp; !trigger_in_reg) begin if(trigger_in_byte != 0) trigger_in_store[7:0] &lt;= trigger_in_byte[7:0]; trigger_in_reg &lt;= 1'b1; end else if (trigger_in_reg) begin trigger_in_reset &lt;= 1'b0; trigger_in_reset &lt;= 1'b1; end else if (!trigger_in_detect) begin</pre>
333 334 335 336 337 338 339 340 341 342 343 344 345 344 345 346 347 348 349 350 351		<pre>bogin trigger_in_store &lt;= 8'h0f; trigger_in_reg &lt;= 1'b0; trigger_in_reset &lt;= 1'b0; end else if (trigger_in_detect &amp; !trigger_in_reg) begin if(trigger_in_byte != 0) trigger_in_store[7:0] &lt;= trigger_in_byte[7:0]; trigger_in_reg &lt;= 1'b1; end else if (trigger_in_reg) begin trigger_in_reset &lt;= 1'b0; trigger_in_reset &lt;= 1'b1; end else if (!trigger_in_detect) begin trigger in reg &lt;= 1'b0; trigger in reg &lt;= 1'b0;</pre>
333 334 335 336 337 338 339 340 341 342 343 344 345 344 345 346 347 348 349 350 351 352		<pre>bogin trigger_in_store &lt;= 8'h0f; trigger_in_reg &lt;= 1'b0; trigger_in_reset &lt;= 1'b0; end else if (trigger_in_detect &amp; !trigger_in_reg) begin if(trigger_in_byte != 0) trigger_in_store[7:0] &lt;= trigger_in_byte[7:0]; trigger_in_reg &lt;= 1'b1; end else if (trigger_in_reg) begin trigger_in_reset &lt;= 1'b0; trigger_in_reset &lt;= 1'b1; end else if (!trigger_in_detect) begin trigger_in_reg &lt;= 1'b0; trigger_in_reset &lt;= 1'b0; trigger_in_reset &lt;= 1'b0; trigger_in_reset &lt;= 1'b0; trigger_in_reset &lt;= 1'b0;</pre>
333         334         335         336         337         338         339         340         341         342         343         344         345         346         347         348         349         350         351         352         353		<pre>bogin trigger_in_store &lt;= 8'h0f; trigger_in_reg &lt;= 1'b0; trigger_in_reset &lt;= 1'b0; end else if (trigger_in_detect &amp; !trigger_in_reg) begin if (trigger_in_byte != 0) trigger_in_store[7:0] &lt;= trigger_in_byte[7:0]; trigger_in_reg &lt;= 1'b1; end else if (trigger_in_reg) begin trigger_in_reset &lt;= 1'b0; trigger_in_reset &lt;= 1'b1; end else if (!trigger_in_detect) begin trigger_in_reset &lt;= 1'b0; trigger_in_reset &lt;= 1'b0; trigger_in_reset &lt;= 1'b0; trigger_in_reset &lt;= 1'b0; end</pre>



### 79.2.2<u>8.2.2</u> Active Transfer EndTerm

The Active Transfer module is used to send or receive a byte to/from the Host. This is useful when the user's microcontroller needs to send a byte from a measurement to the Host for display or processing. The Active Transfer module is addressable, so up to eight individual modules can be instantiated and separately addressed.

757	active_transfer	ACTIVE_TRANSFER_INST
758 🚍	(	
759	.uc_clk	(CLK_66),
760	.uc_reset	(reset),
761	.uc_in	(UC_IN),
762	.uc_out	(uc_out_m[ 1*22 +: 22 ]),
763		
764	.start_transfer	<pre>(transfer_out_reg) ,</pre>
765	.transfer_received	(transfer_in_received),
766		
767	.uc_addr	(3'h2),
768		
769	.transfer_to_host	(transfer_out_byte),
770	.transfer_to_device	(transfer_in_byte)
771	);	
772		

To send a byte to the Host, select the appropriate address that corresponds to an address on Host side. Place the byte in the "transfer\_to\_host" parameter, then strobe the "start\_transfer" bit. Setting the "start\_transfer" bit to high will send one byte from the "transfer\_to\_host" byte to the Host on the next clock high signal (66 MHz). The "start\_transfer" bit can stay high for the duration of the operation of the device, the Active Transfer module will not send another byte. In order to send another byte, the user must cycle the "start\_transfer" bit to low for a minimum of one clock cycle (66 MHz). After the "start\_transfer" bit has been cycled low, the rising edge of the bit will cause the byte on the "transfer to host" parameter to transfer to the host.



181	Ę	//
182		// Transfer byte to Device
183	L	//
184		always @(TRANSFER_OUT_EN or reset)
185	Ę	begin
186		if(!reset)
187	白	begin
188		<pre>transfer_out_detect = 1'b0;</pre>
189	F	end
190		else
191	白	begin
192		<pre>if(transfer_to_device_reset)</pre>
193		<pre>transfer_out_detect = 1'b0;</pre>
194		<pre>else if(TRANSFER_OUT_EN)</pre>
195	白	begin
196		<pre>transfer_out_byte = TRANSFER_OUT_BYTE;</pre>
197		<pre>transfer_out_detect = 1'b1;</pre>
198	F	end
199	F	end
200	L	end
201		
202	F	//
203		<pre>// Reset transfer_to_device_reset</pre>
204	L	//
205		always @(posedge CLK_66 or negedge reset)
206	曱	begin
207		if (!reset)
208	P	begin
209		<pre>transfer_to_device_reset &lt;= 1'b0;</pre>
210	-	end
211		else
212	H	begin
213		1f(transfer_out_detect)
214		<pre>transfer_to_device_reset &lt;= 1'b1;</pre>
215		else
216		<pre>transfer_to_device_reset &lt;= 1'b0;</pre>
217		ena
218		ena

To receive a byte, the Active Host will send a byte using it's dll. The user code must monitor the transfer\_received port. The transfer\_received port will assert high for one clock cycle (66 MHz) when a byte is ready for reading on the transfer\_to\_device port. User code should use an asynchronous always block to detect when the



transfer\_received port is asserted. Upon assertion, the user code should read the byte from the transfer\_to\_device port into a local register.

220	Ξ	//		
221	T	// Transfer to Host		
222	L	//		
223		always @(posedge CLK 66 or negedge reset)		
224		begin		
225	Ť	if (!reset)		
226	F	begin		
227	T	<pre>transfer out &lt;= 1'b0;</pre>		
228		transfer out reg <= 1'b0:		
229		transfer out byte <= 8'h0;		
230	-	end		
231		else		
232	Ē	begin		
233	T	if(start transfer byte & !transfer out)		
234		begin		
235	T	transfer out byte <= TRANSFER HOST BYTE;		
236		transfer out reg <= 1'b1;		
237		<pre>transfer out &lt;= 1'b1;</pre>		
238	-	end		
239		<pre>else if(start transfer byte &amp; transfer out)</pre>		
240		begin		
241	T	<pre>transfer out reg &lt;= 1'b0;</pre>		
242		<pre>transfer out &lt;= 1'b1;</pre>		
243	-	end		
244		<pre>else if(!start transfer byte &amp; transfer out)</pre>		
245	Ē	begin		
246		<pre>transfer out reg &lt;= 1'b0;</pre>		
247		<pre>transfer out &lt;= 1'b0;</pre>		
248	-	end		
249	-	end		
250	L	end		

### 79.2.38.2.3 Active Block EndTerm

The Active Block module is designed to transfer blocks of data between Host and User Code and vice versa. This allows buffers of data to be transferred with a minimal amount of code. The Active Block module is addressable, so up to eight individual modules can be instantiated and separately addressed. The length of the block to be transferred must also be specified in the uc\_length port.



811	active_block	BLOCK_TRANSFER_INST
812 📮	(	
813	.uc_clk	(CLK_66),
814	.uc_reset	(RST),
815	.uc_in	(UC_IN),
816	.uc_out	(uc_out_m[ 2*22 +: 22 ]),
817		
818	.start_transfer	(block_out_reg),
819	.transfer_received	(block_in_rcv),
820		
821	.transfer_ready	(block_byte_ready),
822		
823	.uc_addr	(3'h4),
824	.uc_length	(BLOCK_COUNT_8),
825		
826	.transfer_to_host	(block_out_byte),
827	.transfer_to_device	(block_in_data),
828		
829	.STATE_OUT	(block_state_out),
830	.TEST_BUS	(block_out_test_bus)
831		
832 L	);	
833		

To send a block, it's best to have buffer filled in a previous transaction, Then assert the start\_transfer bit. This method is opposed to collecting and processing data bytes after the start\_transfer bit has been asserted and data is being sent to the Host.

Once the buffer to send is filled with the requisite amount of data, the address and buffer length should be written to the uc\_addr and uc\_length ports. Set the start\_transfer bit high, the user code should monitor the transfer\_ready port. At the rising edge of the transfer\_ready port, the byte at transfer\_to\_host port is transferred to the USB chip. Once this occurs, the user code should copy the next byte in the buffer to transfer\_to\_host port. On the next rising edge of transfer-ready, the byte at transfer\_to\_host will be transferred to the USB chip. This process continues until the number of bytes desicribed by the uc\_length have been transferred into the USB chip.



542	Ē	//
543		<pre>// Registers to start Block Transfer Out</pre>
544	L	//
545		always @(posedge CLK_66 or negedge RST)
546	Ę	begin
547		if(!RST)
548	白	begin
549		<pre>block_out_reg &lt;= 1'b0;</pre>
550		<pre>start_block_transfer_reg &lt;= 1'b0;</pre>
551	-	end
552		else
553	白	begin
554		<pre>if(start_block_transfer &amp; !start_block_transfer_reg)</pre>
555		<pre>start_block_transfer_reg &lt;= 1'b1;</pre>
556		<pre>else if(start_block_transfer_reg &amp; !block_out_reg)</pre>
557	皁	begin
558		<pre>block_out_reg &lt;= 1'b1;</pre>
559	-	end
560		<pre>else if(block_out_counter &gt;= BLOCK_COUNT_8)</pre>
561	P	begin
562		<pre>block_out_reg &lt;= 1'b0;</pre>
563		<pre>start_block_transfer_reg &lt;= 1'b0;</pre>
564	-	end
565		end
566	-	end
567		
568	F	//
569	L	// Data for Block fransfer Out
570		//
572		always (poseuge CLA_00 of negeuge KSI)
573	T	if (IDST)
574		begin
575	Ť	block out counter <= 0:
576	F	end
577		else
578	L L	begin
579	T	if(block byte ready)
580	Ē	begin
581	Τ	<pre>block out counter &lt;= block out counter + 1'd1;</pre>
582	-	end
583		<pre>else if(block_out_counter &gt;= BLOCK_COUNT_8 )</pre>
584	þ	begin
5.9.5		<pre>block_out_counter &lt;= 0;</pre>
202		
586	-	end
586 587	-	end



To receive a buffer from the Host, the user code should monitor the transfer\_received port for assertion. When the bit is asserted, the next rising edge of transfer\_ready will indicate that the byte at transfer\_to\_device is ready for the user code to read.

[Add code snippet showing Active Block Module bytes received by the user code]

### 79.38.3 Timing Diagram for Active Transfer EndTerms

The Active Transfer Library uses the 66 MHz clock to organize the transfers to Host and transfer to Device. The timing of the transfers depends on this clock and the specifications of the USB chip. Users should use the timing diagrams to ensure proper operation of user code in data transfer.



#### 79.3.18.3.1 Active Trigger EndTerm Timing



#### DSO Development System User Manual 15.15 ns CLOCK (66 MHz) 30.1 ns 30.1 ns 5.15 TRANSFER\_RECEIVED 5.4 ns 5.4 ns TRANSFER\_TO\_DEVICE Figure xx Active Transfer To Device Timing 79.3.38.3.3 Active Block EndTerm Timing 15.15 ns 303.1 n 15.15 ns START\_TRANSFE 135 ns TRANSFER\_READY 30.3 ns TRANSFER\_TO\_HOST BYTE 3 BYTE 5 Figure xx Active Block To Host Timing 15.15 ns +++ 15.15 ns 1080 n TRANSFER\_RECEIVED

 TRANSFER\_RECEIVED
 15.15 ns

 15.15 ns
 15.15 ns

 15.15 ns

### 9 PC Active Host Description

The communication path between DSO 100M and the PC is made possible by the use of EndTerms. These EndTerms provide a virtual "pipe" of data capable of sending and receiving bytes between the Windows PC and the FPGA. In order to fully explain the FPGA Code used in the DSO 100M, a quick explanation is needed of the transfer mechanism in the UnoProLyzer C# project.

The Active Host SDK is provided as a dll which easily interfaces to application software written in C#, C++ or C. It runs on the PC and provides transparent connection





The above is a code sample from a C# Windows Form. You can see functions that write a byte to the FPGA

• (EPT\_AH\_SendByte(0x01, (char)LEDStatus))

and write a trigger bit to the FPGA

• (EPT\_AH\_SendTrigger((byte)0x02).



953		
954	L */	
955	active_trigger	ACTIVE_TRIGGER_INST
956	₽ <b>(</b>	
957	.uc_clk	(CLK_66),
958	.uc_reset	(RST),
959	.uc_in	(UC_IN),
960	.uc_out	(uc_out_m[ 0*22 +: 22 ]),
961		
962	.trigger_to_host	(trigger_out),
963	.trigger_to_device	(trigger_in_byte)
964		
965	L );	
966		
967	active_transfer	ACTIVE_TRANSFER_INST_1
968 -	- (	
969	.uc_clk	(CLK_66),
970	.uc_reset	(RST),
971	.uc_in	(UC_IN),
972	.uc_out	(uc_out_m[ 1*22 +: 22 ]),
973		
974	.start_transfer	<pre>(led_start_transfer) ,</pre>
975	.transfer_received	<pre>(led_transfer_in_received),</pre>
976		
977	.transfer_busy	() r
978		
979	.uc_addr	(3'h1),
980		
981	.transfer_to_host	(led_host_transfer_byte),
982	.transfer_to_device	(led_device_transfer_byte)
983	);	

The above code is the interface Verilog which resides in the FPGA. When the C# Windows form sends a byte or trigger, the signals in the FPGA code react and allow the user code to receive the byte and trigger and perform some function with the information. In the case of this example, the LEDs will change state.

Receiving data from the FPGA is made simple by Active Host. Active Host transfers data from the FPGA as soon as it is available. It stores the transferred data into circular buffer. When the transfer is complete, Active Host invokes a callback function which is registered in the users application. This callback function provides a mechanism to transparently receive data from the FPGA. The user application does not need to schedule a read from the USB or call any blocking threads.

#### Active Host Application Figure xx Active Block To Device Timing



### 82-Active Host Application



The FPGA on the DSO 100M can be programmed with the Active Transfer Library and custom HDL code created by the user. Programming the FPGA requires the use of the Quartus Prime software and a standard USB cable. There are no extra parts to buy, just plug in the USB cable. Once the user HDL code is written according to the syntax rules of the language (Verilog and VHDL) it can be compiled and synthesized using the Quartus Prime software. This manual will not focus on HDL coding or proper coding techniques, instead it will use the example code to compile, synthesize and program the FPGA.

### 85.0 Setting up the Project and Compiling

Once the HDL code (Verilog or VHDL) is written and verified using a simulator, a project can be created using Quartus Prime. Writing the HDL code and simulating it will be covered in later sections. Bring up Quartus Prime, then use Windows Explorer to browse to c:/altera/xxx/quartus/qdesigns create a new directory called: "EPT\_Transfer\_Demo".

ize 🔻 🙀 Open 🛛 Include in library 👻 Share with 🕈	Burn New folder			(日•	0	
👃 bin 🔹	Name	Date modified	Туре	Size		
bin64	EPT_Transfer_Test	3/13/2013 8:52 PM	File folder			
a common	🕌 fir_filter	3/3/2013 1:38 AM	File folder			
🚛 cusp	incr_comp_makefile	3/3/2013 1:38 AM	File folder			
a drivers E	whol_verilog_tutorial	3/3/2013 1:38 AM	File folder			
dsp_builder						
603 CON						
Bhaning						
inf						
adesigns						
EPT Transfer Test						
🗿 fir filter						
incr_comp_makefile						
🔰 vhdl_venlog_tutorial						
🗼 sopc_builder						
🚭 altera_inspector.log.zip						
F 01 *						





# **Active Host Application**

-Under Quartus, Select File >New Project Wizard. The Wizard will walk you through setting up files and directories for your project.



### **Active Host Application**

Select Next. At the Add Files window: Browse to the \Projects\_HDL\EPT\_Transfer\_Demo \src folder of the EPT FPGA Development System DVD. Copy the files from the \src directory.

- <u>Active\_block.vqm</u>
- Active\_transfer.vqm
- Active\_trigger.vqm
- <u>Active\_transfer\_library.v</u>
- <u>ft\_245\_state\_machine.v</u>
- endpoint\_registers.vqm
- eptWireOr.v
- mem\_array.v
- read\_control\_logic.v
- write\_control\_logic.v
- EPT\_4CE6\_AF\_D1\_Top.v



# **Active Host Application**

Select Next, at the Device Family group, select Cyclone IV for Family. In the Available Devices group, browse down to EP4CE6E22C8 for Name.

Select Next, leave defaults for the EDA Tool Settings.



## **Active Host Application**

Select Next, then select Finish. You are done with the project level selections.

Next, we will select the pins and synthesize the project.

#### **127.0.0** Selecting Pins and Synthesizing

With the project created, we need to assign pins to the project. The signals defined in the top level file (in this case: EPT\_4CE6\_AF\_D1\_Top.v) will connect directly to pins on the FPGA. The Pin Planner Tool from Quartus Prime will add the pins and check to verify that our pin selections do not violate any restrictions of the device. In the case of this example we will import pin assignments that created at an earlier time. Under Assignments, Select Import Assignments.



## **Active Host Application**

At the Import Assignment dialog box, Browse to the <u>\Projects\_HDL\EPT\_Transfer\_Demo \ DSO 100M\_TOP folder of the EPT FPGA</u> Development System DVD. Select the "DSO 100M\_Top.qsf" file.

Click Ok. Under Assignments, Select Pin Planner. Verify the pins have been imported correctly.



# **Active Host Application**

The pin locations should not need to be changed for EPT USB FPGA Development System. However, if you need to change any pin location, just click on the "location" column for the particular node you wish to change. Then, select the new pin location from the drop down box.



### **Active Host Application**

Exit the Pin Planner. Next, we need to add the Synopsys Design Constraint file. This file contains timing constraints which forces the built in tool called TimeQuest Timing Analyzer to analyze the path of the synthesized HDL code with setup and hold times of the internal registers. It takes note of any path that may be too long to appropriately meet the timing qualifications. For more information on TimeQuest Timing Analyzer, see

http://www.altera.com/literature/hb/qts/qts\_qii53018.pdf?GSA\_pos=1&WT.oss\_r=1& WT.oss=TimeQuest Timing Analyzer

Browse to the \Projects\_HDL\EPT\_Transfer\_Demo \ DSO 100M\_TOP folder of the EPT FPGA Development System DVD. Select the "DSO 100M\_Top.sdc" file.

Copy the file and browse to c:\altera\xxx\quartus\qdesigns\EPT\_Transfer\_Demo directory. Paste the file.



# **Active Host Application**

Select the Start Compilation button.

If you forget to include a file or some other error you should expect to see a screen similar to this:



# **Active Host Application**

Click Ok, the select the "Error" tab to see the error.

The error in this case is the missing file "sync\_fifo". Click on the Assignment menu, then select Settings, then select Files. Add the "sync\_fifo.v" file from the database.



## **Active Host Application**

Click Ok then re-run the Compile process. After successful completion, the screen should look like the following:

At this point the project has been successfully compiled, synthesized and a programming file has been produce. See the next section on how to program the FPGA.

### 188.0.0 Configuring the FPGA

Configuring the FPGA is quick and easy. All that is required is a standard USB Micro B cable and the EPT\_Blaster Driver DLL. Connect the DSO 100M to the PC, open up Quartus Prime, open the programmer tool, and click the Start button. To program the DPL Configuration Flash, follow the steps to install the USB Driver and the JTAG Driver Insert for Quartus Prime.



## **Active Host Application**

If the project created in the previous sections is not open, open it. Click on the Programmer button.

The Programmer Window will open up with the programming file selected. Click on the Hardware Setup button in the upper left corner.



# **Active Host Application**

If you successfully double clicked, the "Currently selected hardware:" dropdown box will show the "EPT Blaster v1.5b".



# **Active Host Application**

Click on the "Add File" button



# **Active Host Application**

At the Browse window, double click on the output files folder.

Double click on the "EPT\_4CE6\_D1\_Top.pof" file. Click the Open button in the lower right corner.

Select the EPCS1 under "Device".



# **Active Host Application**

Next, selet the checkbox under the "Program/Configure" of the Programmer Tool.

Click on the Start button to to start programming the FPGA. The Progress bar will indicate the progress of programming.



### **Active Host Application**

When the programming is complete, the Progress bar will indicate success.

At this point, the DSO 100M is programmed and ready for use. To test that the FPGA is properly programmed, bring up the Active Transfer Demo Tool. Click on one of the LED's and verify that the LED selected lights up. Press one of the switches on the board and ensure that the switch is captured on the Active Host Test Tool. Now you are ready to connect to the Arduino Due and write some code to transfer data between microcontroller and PC.

### 24310 Active Host Application

The Active Host SDK is provided as a dll which easily interfaces to application software written in C#, C++ or C. It runs on the PC and provides transparent connection from PC application code through the USB driver to the user FPGA code. The user code connects to "Endterms" in the Active Host dll. These host "Endterms" have complementary HDL "Endterms" in the Active Transfer Library. Users have seamless bi-directional communications at their disposal in the form of:

- Trigger Endterm
- Transfer Endterm
- Block Endterm

User code writes to the Endterms as function calls. Just include the address of the individual module (there are eight individually addressable modules of each Endterm).



Immediately after writing to the selected Endterm, the value is received at the HDL Endterm in the FPGA. The Trigger Endterms are used as "switches". The user code can set a Trigger bit in the FPGA and cause an event to occur. The Transfer Endterm sends one byte to the FPGA. The Block Endterm sends a block of bytes. By using one of the Active Host Endterms, the user can create a dynamic, bi-directional, and configurable data transfer design.



### 243.1<u>10.1</u> Trigger EndTerm

The Trigger EndTerm is a software component that provides a direct path from the users application to the commensurate Trigger EndTerm in the FPGA. The Trigger has eight bits and is intended to be used to provide a switch at the opposite EndTerm. They are fast acting and are not stored or buffered by memory. When the user code sets a Trigger, it is immediately passed through to the opposite EndTerm via the USB driver. When receiving Trigger, the user application is required to respond to a callback from the Active Host dll.

### 243.2<u>10.2</u> Transfer(Byte) EndTerm

The Transfer EndTerm is a software component that provides a direct path from the users application to the commensurate Transfer EndTerm in the FPGA. It is used to transfer a byte to and from the FPGA. Eight separate Transfer EndTerm modules can be instantiated in the FPGA. Each module is addressed by the user application. Sending a byte is easy, just use the function call with the address and byte value. The byte is immediately sent to the corresponding EndTerm in the FPGA. Receiving a byte is just as easy, a callback function is registered at initialization. When the FPGA transmits a byte using its EndTerm, the callback function is called in the user application. The user code must store this byte in order to use it. The incoming Transfers are stored in a



circular buffer in memory. This allows the user code to fetch the transfers with out losing bytes.

### 243.310.3 Block EndTerm

The Block EndTerm is a software component that provides a direct path from the users application to the commensurate Block EndTerm in the FPGA. The Block EndTerm is used to transfer a complete block to the FPGA. Block size is limited to 1 to 256 bytes. Eight separate Block EndTerm modules can be instantiated in the FPGA. Each module is addressed by the user application. Sending a block is easy, just use the function call with the address, block length, byte array. The block is buffered into a circular buffer in memory then transmitted via the USB bus to the Block EndTerm in the FPGA. Receiving a block is just as easy, a callback function is registered at initialization. When the FPGA transmits a block using its EndTerm, the callback function is called in the user application. The incoming Transfers are stored in a circular buffer in memory. This allows the user code to fetch the transfers with out losing bytes.

### 243.4<u>10.4</u> Active Host DLL

The Active\_Host DLL is designed to transfer data from the FPGA when it becomes available. The data will be stored into local memory of the PC, and an event will be triggered to inform the user code that data is available from the addressed module of the FPGA. This method of automatically moving data from the user code Endterm in the FPGA makes the data transfer transparent.

The data seamlessly appears in Host PC memory from the Arduino. The user code will direct the data to a control such as a textbox on a Windows Form. The transparent receive transfer path is made possible by a Callback mechanism in the Active Host dll. The dll calls a registered callback function in the user code. The user code callback can be designed to generate any number of events to handle the received data.

The user application will access the FPGA by use of functions contained in the Active Host dll. The functions to access the FPGA are:

• EPT\_AH\_GetName()

- EPT\_AH\_GetVersionString()
- EPT\_AH\_GetVersionControl()
- EPT\_AH\_GetInterfaceVersion()
- EPT\_AH\_CheckCompatibility()
- EPT\_AH\_Open()
- EPT\_AH\_Close()
- EPT\_AH\_Initialize()



- EPT\_AH\_Release()
- EPT\_AH\_QueryDevices()
- EPT\_AH\_SelectActiveDeviceByName()
- EPT\_AH\_SelectActiveDeviceByIndex()
- EPT\_AH\_GetDeviceName()
- EPT\_AH\_GetDeviceSerial()
- EPT\_AH\_OpenDeviceByIndex()
- EPT\_AH\_CloseDeviceByIndex()
- EPT\_AH\_CloseDeviceByName()
- EPT\_AH\_SendTrigger ()
- EPT\_AH\_SendByte ()
- EPT\_AH\_SendBlock ()
- EPT\_AH\_SendTransferControlByte()
- EPT\_AH\_RegisterReadCallback ()
- EPT AH GetLastError()
- EPT\_AH\_PerformSelfTest()
- EPT\_AH\_LEDBlinky()
- EPT\_AH\_SetDebugMode()
- EPT\_AH\_RegisterReadCallbackForChannel()
- EPT\_AH\_FlushDeviceChannelBuffer()
- EPT\_AH\_GetDeviceChannelFreeBufferBytes()
- EPT\_AH\_GetDeviceChannelPendingBufferBytes()
- EPT\_AH\_SetChannelConnectionFlag()
- EPT\_AH\_GetChannelConnectionFlag()

### 243.4.1<u>10.4.1</u> Active Host Open Device

To use the library functions for data transfer and triggering, an Earth People Technology device must be opened. The first function called when the Windows Form loads up is the <project\_name>\_Load(). This function is called automatically upon the completion of the Windows Form, so there is no need to do anything to call it. Once this function is called, it in turn calls the ListDevices(). Use the function List Devices() to detect all EPT devices connected to the PC.



```
// Main object loader
private void EPT_Transfer_Demo_Load(object sender, System.EventArgs e)
{
    // Call the List Devices function
    ListDevices();
    //Active Host Debug
    //EPT_AH_SetDebugMode(1);
}
```

The ListDevices() function calls the EPT\_AH\_Open() function to load up the ActiveHost Dll. Next, it calls EPT\_AH\_QueryDevices() which searches through the registry files to determine the number of EPT devices attached to the PC. Next, EPT\_AH\_GetDeviceName() is called inside a for loop to return the ASCII name of each device attached to the PC. It will automatically populate the combo box, cmbDevList with all the EPT devices it finds.

```
// List Devices function
private unsafe Int32 ListDevices ()
   {
   Int32 result;
   Int32 num devices;
   Int32 iCurrentIndex;
   // Open the DLL
  result = EPT_AH_Open(null, null, null);
if (result != 0)
      MessageBox.Show("Could not attach to the ActiveHost library");
      return 0;
      3
   // Query connected devices
   num_devices = EPT_AH_QueryDevices();
   //Prepare the Combo box for population
   iCurrentIndex = cmbDevList.SelectedIndex;
   cmbDevList.Items.Clear();
   // Go through all available devices
   for (device_index = 0; device_index < num_devices; device_index++)</pre>
      {
          String str;
          str = Marshal.PtrToStringAnsi((IntPtr)EPT_AH_GetDeviceName(device_index));
          cmbDevList.Items.Add(str);
      3
  return 0:
}
```




The user will select the device from the drop down combo box. This can be seen when the Windows Form is opened and the cmbDevList combo box is populated with all the devices. The selected device will be stored as an index number in the variable device\_index.

	Transfort	
USB High Speed		
Open	Close	

In order to select the device, the user will click on the "Open" button which calls the Open\_Device() function. The device\_index is passed into the

EPT\_AH\_OpenDeviceByIndex() function. If the function is successful, the device name is displayed in the label, labelDeviceCnt. Next, the device is made the active device and the callback function is registered. Finally, the Open button is grayed out and the Close button is made active.



```
// Open the device
public unsafe Int32 OpenDevice()
     device index = (int)cmbDevList.SelectedIndex;
     if (EPT_AH_OpenDeviceByIndex(device_index) == 0)
     {
         String message = "Could not open device " +
              Marshal.PtrToStringAnsi((IntPtr)EPT_AH_GetDeviceName(device_index)) + ", " +
Marshal.PtrToStringAnsi((IntPtr)EPT_AH_GetDeviceSerial(device_index));
         MessageBox.Show(message);
         return 0;
    }
     else
     {
         labelDeviceCnt.Text = "Connected to device " +
              Marshal.PtrToStringAnsi((IntPtr)EPT_AH_GetDeviceName(device_index)) + ", " +
Marshal.PtrToStringAnsi((IntPtr)EPT_AH_GetDeviceSerial(device_index));
    }
     // Make the opened device the active device
     if (EPT_AH_SelectActiveDeviceByIndex(device_index) == 0)
     {
         String message = "Error selecting device: %s " +
             Marshal.PtrToStringAnsi((IntPtr)EPT_AH_GetLastError());
         MessageBox.Show(message);
         return 0;
     }
     // Register the read callback function
     RegisterCallBack();
     btnOpenDevice.Enabled = false;
     btnCloseDevice.Enabled = true;
     return 0;
h
```

#### 243.4.2<u>10.4.2</u> Active Host Read Callback Function

The local callback function is populated. It resides in the active\_transfer.cs file. This function will be called from the Active Host dll. When the EPT Device has transferred data to the PC, the callback function will do something with the data and command.

```
// Actual callback function which will read messages coming from the EPT device
unsafe void EPTReadFunction (Int32 device_id, Int32 device_channel, byte command, byte payload,
    {
        char* message = (char *)data;
        // Select current device
        EPT_AH_SelectActiveDeviceByIndex(device_id);
        // Make sure we have data in the system
        if (message != null)
        {
        MessageBox.Show("Message " + Marshal.PtrToStringAnsi((IntPtr) message));
        }
    }
}
```





Because the callback function communicates directly with the dll and must pass pointers from the dll to the C# Windows Form, the Marshaling scheme must be used. Marshaling allows pointer variables created in the dll to be passed into the C#. It is an advanced topic and will not be covered in this manual.

#### 243.4.310.4.3 Active Host Triggers

The user application can send a trigger to the FPGA by using the EPT\_AH\_SendTrigger() function. First, open the EPT device to be used with EPT\_AH\_OpenDeviceByIndex(). Call the function with the bit or bits to assert high on the trigger byte as the parameter. Then execute the function, the trigger bit or bits will momentarily assert high in the user code on the FPGA.

```
private void btnTrigger1_Click(object sender, EventArgs e)
{
    EPT_AH_SendTrigger((char) 1);
}
```

To detect a trigger from the FPGA, the user application must subscribe to the event created when the incoming trigger has arrived at the Read Callback function. The Read Callback must store the incoming trigger in a local variable. A switch statement is used to decode which event should be called to handle the incoming received data.

- TRIGGER\_IN
- TRANSFER\_IN
- BLOCK\_IN



```
unsafe void EPTReadFunction (Int32 device_id, Int32 device_channel, byte command, byte payload,
{
  byte* message = data;
   // Select current device
  EPT_AH_SelectActiveDeviceByIndex(device_id);
   //Add command and device_channel to the receive object
   EPTReceiveData.Command = ((command & 0x38) >> 3);
  EPTReceiveData.Address = device_channel;
   //Check if the command is Block Receive. If so,
   //use Marshalling to copy the buffer into the receive
    //object
   if (EPTReceiveData.Command == BLOCK OUT COMMAND)
   {
       EPTReceiveData.Length = data_size;
      EPTReceiveData.cBlockBuf = new Byte[data size];
      Marshal.Copy(new IntPtr(message), EPTReceiveData.cBlockBuf, 0, data_size);
       //UpdateTextBlock();
  }
  else
   {
       EPTReceiveData.Payload = payload;
   }
   this.Invoke(new EventHandler(EPTParseReceive));
}
private void EPTParseReceive(object sender, System.EventArgs e)
{
    switch (EPTReceiveData.Command)
    {
        case TRIGGER_OUT_COMMAND:
            TriggerOutReceive();
            break;
        case TRANSFER OUT COMMAND:
             TransferOutReceive();
            break:
        case BLOCK OUT COMMAND:
            BLockOutReceive();
            break;
        default:
            break;
    }
}
```

The event handler function for the TRIGGER\_IN's uses a switch statement to determine which trigger was asserted and what to do with it.



```
public void Receive_Trigger_In(object sender, EventArgs e)
ł
    switch (ept_data.Payload)
    {
        case 0x01:
            lLableSwitch1.Text = "Switch 1\n Pressed";
            break:
        case 0x02:
            lLableSwitch2.Text = "Switch 2\n Pressed";
            break;
        case 0x04:
            lLableSwitch1.Text = "";
            lLableSwitch2.Text = "";
            break;
    }
}
```

The receive callback method is complex, however, Earth People Technology has created several projects which implement callbacks. Any part of these sample projects can copied and pasted into a user's project.

## 243.4.4<u>10.4.4</u> Active Host Byte Transfers

The Active Host Byte Transfer EndTerm is designed to send/receive one byte to/from the EPT Device. To send a byte to the Device, the appropriate address must be selected for the Transfer module in the FPGA. Up to eight modules can be instantiated in the user code on the FPGA. Each module has its own address.

```
private void btnWriteByte_Click(object sender, EventArgs e)
{
    int ibyte, address_to_device;
    ibyte = Convert.ToInt32(tbNumBytes.Text);
    address_to_device = Convert.ToInt32(tbAddress.Text);
    EPT_AH_SendByte(address_to_device, (char)ibyte);
}
```

Use the function EPT\_AH\_SendByte() to send a byte the selected module. First, open the EPT device to be used with EPT\_AH\_OpenDeviceByIndex(). Then add the address of the transfer module as the first parameter of the EPT\_AH\_SendByte() function. Enter the byte to be transferred in the second parameter. Then execute the function, the byte will appear in the ports of the Active Transfer module in the user code on the FPGA.

To transfer data from the FPGA Device, a polling technique is used. This polling technique is because the Bulk Transfer USB is a Host initiated bus. The Device will not transfer any bytes until the Host commands it to. If the Device has data to send to the Host in an asynchronous manner (meaning the Host did not command the Device to send data), the Host must periodically check the Device for data in it's transmit FIFO. If



data exists, the Host will command the Device to send it's data. The received data is then stored into local memory and register bits are set that will indicate data has been received from a particular address.

To receive a byte transfer from the Active host dll, user code must subscribe to the event created when the incoming byte transfer has arrived at the Read Callback function. The Read Callback must store the incoming transfer payload and module address in a local memory block. A switch statement is used to decode which event should be called to handle the incoming received data. The event handler function will check for any bytes read for that address.

unsafe void EPTReadFunction (Int32 device\_id, Int32 device\_channel, byte command, byte payload,

```
{
  byte* message = data;
   // Select current device
  EPT_AH_SelectActiveDeviceByIndex(device_id);
   //Add command and device channel to the receive object
   EPTReceiveData.Command = ((command & 0x38) >> 3);
   EPTReceiveData.Address = device_channel;
   //Check if the command is Block Receive. If so,
   //use Marshalling to copy the buffer into the receive
    //object
   if (EPTReceiveData.Command == BLOCK OUT COMMAND)
   {
       EPTReceiveData.Length = data_size;
      EPTReceiveData.cBlockBuf = new Byte[data size];
       Marshal.Copy(new IntPtr(message), EPTReceiveData.cBlockBuf, 0, data_size);
       //UpdateTextBlock();
   }
   else
   {
       EPTReceiveData.Payload = payload;
   this.Invoke(new EventHandler(EPTParseReceive));
}
```



{

}

DSO Development System User Manual

```
private void EPTParseReceive(object sender, System.EventArgs e)
            switch (EPTReceiveData.Command)
            {
                case TRIGGER_OUT_COMMAND:
                     TriggerOutReceive();
                     break;
                case TRANSFER_OUT COMMAND:
                     TransferOutReceive();
                     break;
                case BLOCK OUT COMMAND:
                     BLockOutReceive();
                     break:
                default:
                     break;
            }
       }
      The EventHandler function EPTParseReceive() is called by the Read Callback function.
      The EPTParseReceive() function will examine the command of the incoming byte
      transfer and determine which receive function to call.
public void TransferOutReceive()
    string WriteRcvChar = "";
    WriteRcvChar = String.Format("{0}", (int)EPTReceiveData.Payload);
    tbDataBytes.AppendText(WriteRcvChar + ' ');
tbAddress.Text = String.Format("{0:x2}", (uint)System.Convert.ToUInt32(EPTReceiveData.Address.ToString()
```

For our example project, the TransferOutReceive() function writes the Transfer byte received to a text block. The receive callback method is complex, however, Earth People Technology has created several projects which implement callbacks. Any part of these sample projects can copied and pasted into a user's project.

#### 243.4.510.4.5 Active Host Block Transfers

The Active Host Block Transfer is designed to transfer blocks of data between Host and FPGA and vice versa through the Block EndTerm. This allows buffers of data to be transferred with a minimal amount of code. The Active Host Block module (in the User Code) is addressable, so up to eight individual modules can be instantiated and separately addressed. The length of the block to be transferred must also be specified. The Block EndTerm is limited to 1 to 256 bytes.

To send a block, first, open the EPT device to be used with EPT\_AH\_OpenDeviceByIndex(). Next, use the EPT\_AH\_SendBlock() function to send the block. Add the address of the transfer module as the first parameter. Next, place the pointer to the buffer in the second parameter of EPT\_AH\_SendBlock(). Add the length





}

DSO Development System User Manual

of the buffer as the third parameter. Then execute the function, the entire buffer will be transferred to the USB chip. The data is available at the port of the Active Block module in the user code on the FPGA.

```
public unsafe void BlockCompare(object data)
{
    int BlockAddress = (int)data;
    byte[] cBuf = new Byte[device[BlockAddress].Length];
    if ((device[BlockAddress].Repititions > 0) &
        !device[BlockAddress].TransferPending & !BlockTransferStop)
    {
        device[BlockAddress].TransferPending = true;
        Buffer.BlockCopy(block_8_in_payload, 0, cBuf, 0,
            device[BlockAddress].Length);
        fixed (byte* pBuf = cBuf)
        {
          EPT AH SendBlock(device[BlockAddress].Address,
                           (void*)pBuf, (uint)device[BlockAddress].Length);
        }
        Thread.Sleep(1);
        EPT_AH_SendTransferControlByte((char)2, (char)2);
        Thread.Sleep(1);
        EPT_AH_SendTrigger((char)128);
        Thread.Sleep(1);
        EPT_AH_SendTransferControlByte((char)2, (char)0);
        if (BlockTransferInfinite)
            device[BlockAddress].Repititions = 1;
        else
            device[BlockAddress].Repititions--;
    }
```

To receive a block transfer from the FPGA Device, a polling technique is used by the Active Host dll. This is because the Bulk Transfer USB is a Host initiated bus. The Device will not transfer any bytes until the Host commands it to. If the Device has data to send to the Host in an asynchronous manner (meaning the Host did not command the Device to send data), the Host must periodically check the Device for data in its transmit FIFO. If data exists, the Host will command the Device to send its data. The received data is then stored into local memory and register bits are set that will indicate data has been received from a particular address. The receive callback function is then called from the Active Host dll. This function start a thread to do something with the block data.

To receive a byte transfer from the callback function, user code must subscribe to the event created when the incoming byte transfer has arrived at the Read Callback function. The Read Callback must store the incoming transfer payload and module



address in a local memory block. A switch statement is used to decode which event should be called to handle the incoming received data. The event handler function will check for any bytes read for that address.

```
unsafe void EPTReadFunction (Int32 device_id, Int32 device_channel, byte command, byte payload,
{
    byte* message = data;
```

```
// Select current device
  EPT_AH_SelectActiveDeviceByIndex(device_id);
   //Add command and device_channel to the receive object
   EPTReceiveData.Command = ((command & 0x38) >> 3);
  EPTReceiveData.Address = device_channel;
   //Check if the command is Block Receive. If so,
    //use Marshalling to copy the buffer into the receive
   //object
   if (EPTReceiveData.Command == BLOCK_OUT_COMMAND)
   {
       EPTReceiveData.Length = data_size;
      EPTReceiveData.cBlockBuf = new Byte[data_size];
      Marshal.Copy(new IntPtr(message), EPTReceiveData.cBlockBuf, 0, data_size);
       //UpdateTextBlock();
  }
  else
   {
      EPTReceiveData.Payload = payload;
   }
   this.Invoke(new EventHandler(EPTParseReceive));
}
private void EPTParseReceive(object sender, System.EventArgs e)
{
    switch (EPTReceiveData.Command)
    {
        case TRIGGER_OUT_COMMAND:
            TriggerOutReceive();
            break;
        case TRANSFER_OUT_COMMAND:
            TransferOutReceive();
            break;
        case BLOCK OUT COMMAND:
            BLockOutReceive();
            break;
        default:
            break;
    }
}
```



The EventHandler function EPTParseReceive() is called by the Read Callback function. The EPTParseReceive() function will examine the command of the incoming byte transfer and determine which receive function to call.

```
public void Receive_Block_In(object sender, EventArgs e)
    device[ept data.Address].TransferPending = false;
    Thread.Sleep(5);
    if (device[ept_data.Address].ContinuosCountTest == false)
    {
        Thread t = new Thread(new ParameterizedThreadStart(BlockCompare));
        t.Start(ept_data.Address);
    }
    if (device[ept_data.Address].Repititions == 0)
    {
        Thread u = new Thread(new ParameterizedThreadStart(Display_Block_In));
        u.Start(BlockCount);
    }
    else if (BlockTransferInfinite | device[ept_data.Address].ContinuosCountTest)
    {
        if ((BlockCount % 100) == 0)
        {
            Thread u = new Thread(new ParameterizedThreadStart(Display_Block_In));
            u.Start(BlockCount);
        }
    }
}
```

For our example project, the Receive\_Block\_In() function writes the Transfer block received to a text block. The receive callback method is complex, however, Earth People Technology has created several projects which implement callbacks. Any part of these sample projects can copied and pasted into a user's project.

# 244<u>11</u> Assembling, Building, and Executing a .NET Project on the PC

The Active Host Application DLL is used to build a custom standalone executable on the PC that can perform Triggers and Transfer data to/from the DSO 100M. A standalone project can be range from a simple program to display and send data from the user to/from the Arduino Due. Or it can more complex to include receiving data, processing it, and start or end a process on the Arduino. This section will outline the procedures to take an example project and Assemble it, Build it, and Execute it. This guide will focus on writing a Windows Forms application using the C# language for the Microsoft Visual Studio with .NET Framework. This is due to the idea that beginners can write effective Windows applications with the C# .NET Framework. They can focus on a subset of the language which is very similar to the C language.



Anything that deviates from the subset of the C language, presented as in the Arduino implication (such as events and controls), will be explained as the explanation progresses. Any language can be used with the Active Host Application DLL.

# 244.1<u>11.1</u> Creating a Project

Once the application is installed, open it up. Click on File->New Project.

e) N	licrosoft Visual C# 2010 Express				2
File	Edit View Debug Tools	Window Help			
Ċ,	New Project	Ctrl+Shift+N	🛛 🖄 tooltip	-   🔩 🕾 🏷 🖬 🚽 -	
đ	Open Project	Ctrl+Shift+O			
2	Open File	Ctrl+O			
	Close				
T)	Close Solution				
И	Save Selected Items	Ctrl+S			
	Save Selected Items As				
2	Save All	Ctrl+Shift+S			
	Export Template				
1	Page Setup				
3	Print	Ctrl+P			
	Recent Files	,			
	Recent Projects and Solutions	,	•		
	Exit	Alt+F4		•	ņ
	8 a s a		-		
	Error List Errol Results				•
	- A Find Kesuis				
ead	ly .				

At the New Project window, select the Windows Forms Application. Then, at the Name: box, type in EPT\_Transfer\_Demo

Recent Templates	Set by:	Default - US TH		Search Installed Templates	Q
installed Templates	_			Tange Visual Cit	
Visual C#		Windows Forms Application	Visual C#	A project for creating an applicati	on with a
Online Templates		WPF Application Windo	ws Forms Application Joual C#	Windows Forms user interface	
	-	Console Application	Visual C#		
	CH	Class Library	Visual C#		
	đ	WPF Browser Application	Visual C#		
	c#	Empty Project	Visual C#		
	_				
					_
reame: EPT_Tri	nster_rest				

The project creation is complete.





Save the project, go to File->Save as, browse to a folder to create EPT\_Transfer\_Demo folder. The default location is c:\Users\<Users Name>\documents\visual studio 2010\Projects.

C Project Location					×
😋 🔵 🗢 🕌 « Earth People Technology 🕨 EPT_T	ransf	er_Test + EPT_Transfer_Test	✓ 47 Search EPT_Tra	nsfer_Test	٩
Organize 🔻 New folder				855 👻	0
ActiveHost_1.0.0.6	^	Name	Date modified	Туре	
ActiveHost_1.0.0.7					
Boards			No items match your search.		
EPT_570_AP_Active_Transfer					
EPT_570_AP_Data_Collector					
EPT_FT201X_Interface					
EPT_FT2232_Interface					
EPT_FT2232H_JTAG	-				
EPT_FT2232H_SVF_Programmer					
EPT_FT2232H_USBTransfer					
▲ → EPT_Transfer_Test					
EPT_Transfer_Test					
Image: Barbary Barb					
Image: Book State Sta	-	1			
					_
Folder: EPT_Transfer_Test					
			Select Folder	Cance	8

# 244.1.1<u>11.1.1</u> Setting up the C# Express Environment x64 bit

The project environment must be set up correctly in order to produce an application that runs correctly on the target platform. If your system supports 64 bit operation, perform the following steps





lick on System.		Formatted: Left
	•	
heck under System\System type:		Formattada Jat
	•	





Go to Tools->Options, locate the "Show all settings" check box. Check the box.

General Fonts and Colors Keyboard	items shown in Window menu           10         items shown in recently used lists
> Debugging	Visual experience
	Automatically adjust visual experience based on client performance
	Enable rich client visual experience
	Use hardware graphics acceleration if available
	Visual Studio is currently using hardware-accelerated rendering. The visual experience settings automatically change based on system capabilities.
	✓ Show status bar
	Close button affects active tool window only
	Auto Hide button affects active tool window only
	Restore File Associations
Show all settings	OK Cance

In the window on the left, go to "Projects and Solutions". Locate the "Show advanced build configurations" check box. Check the box.





#### Go to Build->Configuration Manager.

₫	EPT_Trar	nsfer_Te:	st - Micro	soft V	isual C# 20	010 Exp	ress						-		
File	e Edit	View	Project	Build	Debug	Data	Tool	s Windo	w Help						
11	- 📴 ר	🞽 🖬	a   🗸	*	Build Solu	ition			-6 g	•	x86			- 12	) EP
14	4   I2	후 릐	0]† 0]}		Rebuild S	olution			Ĕ‡	응; 6;		12, 23	- 22 -		
×	Solutio	n Explor	er	2	Publish E	PT_Tran	sfer_T	est	×						
To		🄉 🛃			Configura	ition Ma	nager								
polbox 📅 Data Sources	S S	olution ' EPT_T Pr Re E Fo	EPT_Tran ransfer_1 operties ferences orm1.cs ogram.cs	sfer_T Fest	est' (1 proj	ect)		🖳 Form					8	)	
														)	

In the Configuration Manager window, locate the "Active solution platform:" label, select "New" from the drop down box.



Configuration Manager			? ×				
Active solution configuration:		Active solution platform:					
Debug   Project contexts (check the project configurations to build or dep		x86 • • • • • • • • • • • • • • • • • • •					
EPT_Transfer_Test	Debug	▼ x86	▼ ▼				
			Close				

In the New Solution Platform window, click on the drop down box under "Type or select the new platform:". Select "x64".

Configuration Manager			8 23
Active solution configura Debug Project contexts (check th	tion: • ie project configurations to build or de	Active solution platform: x86 eploy):	•
Project EPT_Transfer_Test	New Solution Platform Type or select the new platfor Any CPU Any CPU Ranium 64 Create new project platfo	orms OK Cancel	Euid
			Close

Click the Ok button. Verify that the "Active Solution Platform" and the "Platform" tab are both showing "x64".



onfiguration Manager		$\frown$	?
Active solution configuration:	Acti	ive solution platform:	
Release	x64		
Project contexts (check the proje	ct configurations to build or deploy)	:	
Project	Configuration	Platform	Build
EPT_Transfer_Test	Release	▼ x64	<b>~</b>
		$\smile$	
			Close

Also, select "Release" under "Active solution configuration". Click Close. Then, using the Solution Explorer, you can right click on the project, select Properties and click on the Build tab on the right of the properties window.

Leg E	PI_Iransfer_Test - Microsof	t Visi	ial C# 2010 Expr	ess				
File	Edit View Project B	uild	Debug Data	Tools Window H	lelp			
1: 6	🗅 🐱 🕶 🚅 🛃 🗸 🗉	) 🗎	19-04-	🖉 - 🖳 🕨 Relea	ase	<b>∗</b> x64		-   0
ŧ 🕸	中國 속 의 교 아 교	15	朝麗葉回	200 및: 및: 및:   음	붉 람	e: 🕂 🕂	] % % 물; -	
<b>x</b>	Solution Explorer		<b>*</b> ₽ ×	Form1.cs [Design]	×			
3				·				
olbox	Solution 'EPT_Transfe	r_Tes	ť (1 project)	🖳 Form1				
2	Properties		Build			1		
Data	References		Rebuild					
Sou	Form1.cs		Publish					
ß	Program.cs		Add		•			
			Add Reference					
			Add Service Re	ference				2
			Set as StartUp F	Project				
			Debug		•			
		Ж	Cut	Ctrl+	Х			
		12	Paste	Ctrl+	v			
		$\boldsymbol{\times}$	Remove	Del				
			Rename			0		5
			Properties	Alt+I	Enter			
		_						
				-				

Verify that the "Platform:" label has "Active (x64)" selected from the drop down box.



Solution Explorer + I ×	EPT_Transfer_Test* >	Form1.cs (Design)
Solution 'EPT_Transfer_Test' (1 project)	Application	Configuration: Active (Release)
<ul> <li>Properties</li> </ul>	Build	
i ≣ Menece i ∏ Fond.a G Progen.cs	Build Events Debug Resources Settings Reference Paths Signing Security Publish	General Candidional compliation symbols Candidional compliation Candi

Next, unsafe code needs to be allowed so that C# can be passed pointer values from the Active Host. Click on the Build tab and locate the "Allow unsafe code" check box. Check the box

Data_Collector × ac	tive_transfer.cs Form1.Designer.cs	Form1.cs Fo
Application	General	
Build	Conditional compilation symbols:	
Build Events	Define DEBUG constant	
Debug	Define TRACE constant	
Resources	Allow unsafe code	
Settings	Optimize code	
Reference Paths	Errors and warnings	
Signing	Warning level:	4
Security	Suppress warnings:	
Publish	Treat warnings as errors	
	None	
	© All	

Click on the Save All button on the tool bar. The project environment is now setup and ready for the project files. Close the Project.

# 244.2<u>11.2</u> Assembling Files into the Project

The
following is an example for demonstration purposes only. Create the fictional
EPT_Transfer_Demo project
Раде



Irganize 👻 Include in library 👻 Share with 👻	Burn	New folder			
BPT USB-CPLD Development System CD     BANG	E	Nume bin bin construction construction construction bin construction construct	Dete modified 3/2/2013 11:29 PM 3/2/2013 11:29 PM 3/2/2013 11:29 PM 3/2/2013 11:28 PM 3/2/2013 11:28 PM 3/2/2013 11:28 PM 2/7/2013 11:38 PM 2/2/2013 11:36 PM 2/2/2013 11:36 PM	Type File folder File folder Visual C# Poiet f Visual Studio Project f Visual Studio Proj Visual C# Source f JNET Managed Re Visual C# Source f	Size 7 KB 5 KB 1 KB 19 KB 6 KB 1 KB

## 244.2.1—Add Files to Project

#### \*\*\*NOTE\*\*\*

If you named your project something other than EPT\_Transfer\_Demo, you will have to make changes to the \*.cs files above. This is because Visual C# Express links the project files and program files together. These chages can be made by modifying the following:

- 244.Change namespace of Form1.cs to new project name.244.Change class of Form1.cs to new project name.
- 244. Change constructor of Form1.cs to new project name.





### 244.2.1—Add Files to Project

244.Change namespace of Form1.Designer.cs to new project name.244.Change clase of Form1.Designer.cs to new project name.

244. Change the this.Name and this.Text in Form1Designer.cs to new project name.

244. Change this.Load in Form1Designer.cs to include new project name.

244. Change namespace in Program.cs to new project name

244. Change Application.Run() in Program .cs to new projectname.



#### 244.2.1—Add Files to Project

Open the EPT\_Transfer\_Demo project. Right click on the project in the Solutions Explorer. Select Add->Existing Item.

<u></u>	EPT_Transfer_Test - Micro	soft	/isual C# 2010 Expre	55					
File	Edit View Project	Buil	d Debug Data	Tools Win	dow Help				
1 6	🗊 🔁 • 💕 🛃 🕼   🐰	Ep.	13 9 - (2 - 6	2 - B,	Release		x64	- 🖄	EPT_570
1	후   을 속 킄   현 애	<u>111</u>	日日 昭 平   日	]a 3()¢ 0()a	다. 	응; 응	:     % %	- Eŝ -	
*	Solution Explorer		<b>-</b> ₽ ×	Form1.cs [	Design] $\times$				
10	🔚 👔 🛃							~	
box	Solution 'EPT_Tran	sfer_	Test' (1 project)	🖳 Fo	rm1			83	
-	EPT_Transfer_		Build			1			
Data	References		Rebuild						
Sou	Form1.cs	2	Publish						
ŝ	Program.c		Add		•	8	New Item	Ctrl+Shift+A	I
			Add Reference				Existing Item	Shift+Alt+A	
			Add Service Referen	ice		<b>E</b>	New Folder		
			Set as StartUp Proje	ct		1	Windows Form		
			Debug		•	1	User Control		
		Ж	Cut		Ctrl+X	93	Class	Shift+Alt+C	
		1	Paste		Ctrl+V	-			
		×	Remove		Del				
			Rename			-	0	6	
			Properties		Alt+Enter				
		_				-			

Browse to the EPT\_Transfer\_Demo project folder and select the active\_transfer\_xx.cs file (choose either the 32 bit or 64 bit version, depending on whether your OS is 32 or 64 bit). Click Add.

e Edit View Project Debug Data Tools Window	Help		
1	tooltip - 💀 🕾 🛃 🚽		
다 Ha Na Mail 프 프 프			
Solution Explorer * 🔍 🗙 Form1.cs 🗙			<ul> <li>Properties</li> </ul>
Solution 'FPT Transfer Test' (1 project)	Add Existing Item - EPT_Transfer_Test		nsf
	🕖 🚽 « EPT_Transfer_Test 🕨 EPT_Transfer_Test		ransfer_Test
References	Organize 👻 New folder		88 • 🗂 🕢 GI
Form1.cs     Form1.Designer.cs	Microsoft Visual C# 2010 Express	Name	Date m
Torm1.resx	Projects	🍶 bin	8/24/20
Program.cs		i obj	8/24/20
	🚖 Favorites	Properties	8/24/20
	E Desktop	active_transfer.cs	9/16/20
	Downloads	Form1.cs	9/17/20
	💔 Dropbox	S Form1.Designer.cs	9/17/20
	1 Recent Places	Form1.resx	9/17/20
		Program.cs	8/22/20
× >	Calibraries		
Notation Explorer 🔄 Class View 100 %			
Error List	+ Homegroup		
34 Errors 0 Warnings (i) 0 Messages			
Description	Computer *	•	
0.1 The name 'm bBert' doer not wirt in the current	Element of a book of	Visual C+ Film	(1 and must suffice)
The name 'Continue' does not exist in the current	active_transfer.cs	- Visual C# Files	( .c., .rest .sett) +
2 The name 'FT Clear' days get mid in the current		Add	Cancel
4 The name 'm bRest' does not exist in the current			
The name in provides not exist in the current dama	formation of the second s	E 17 EDT Trand	Task The operation
The name in the ort uses not exist in the current conto	en romitics :	D I/ EPI_ITANSI	file contain

In the C# Express Solution Explorer, you should be able to browse the files by clicking on them. There should be no errors noted in the Error List box.







## 244.2.2211.2.2 Adding Controls to the Project

Although, the C# language is very similar to C Code, there are a few major differences. The first is C# .NET environment is event based. A second is C# utilizes classes. This guide will keep the details of these items hidden to keep things simple. However, a brief introduction to events and classes will allow the beginner to create effective programs.

Event based programming means the software responds to events created by the user, a timer event, external events such as serial communication into PC, internal events such as the OS, or other events. The events we are concerned with for our example program are user events and the timer event. The user events occur when the user clicks on a button on the Windows Form or selects a radio button. We will add a button to our example program to show how the button adds an event to the Windows Form and a function that gets executed when the event occurs.

The easiest way to add a button to a form is to double click the Form1.cs in the Solution Explorer. Click on the  $\overrightarrow{}$  button to launch the Toolbox.





<b>e</b>	PT_Transfer_Test - Microsoft Visual C# 2010 E	xpress				
File	Edit View Project Debug Data Too	ols Window Help				
1 6	1 🖼 • 💕 🔙 🖉   🕉 🛍 🛍 👘 • 🕫	- 🕨 🌌 tooltip	- 🗠 🕾 🔆 🖬 🛃 🖕			
: \$	1  ほそ引  ゆや亜  雪別路道	티아꺄먂먂루 을 찾 띾 띾 판	흰   백 백   몰: -			
-	Solution Explorer 🚽 🕂 👻	Form1.cs Form1.Designer.cs Form	m1.cs [Design] × active transfer.cs	-	Toolbox	- # X
2					> All Window	vs Forms 🔺
tab	Solution 'EPT Transfer Test' (1 project)	Carlos and			⊿ Common 0	Controls
ase [	Image: A manual state of the state of t	Re Active Host			R Poin	ter 😑
- a	Properties			E	ab Butte	on
prer	References	-			Chec	:kBox
	active_transfer.cs	Close			Chec	:kedListBox
	Form1.cs				E Com	boBox
	Form1.Designer.cs				📷 Date	TimePicker
	Program.cs	Transfer Single Byte	Transfer Multiple Bytes		A Labe	d i i i i i i i i i i i i i i i i i i i
	_ ,	Write LoopBack	Multi Byte		<u>A</u> Linkl	Label
					ES ListB	ox
		Send Byte 255 Address 2			bas ListV	iew
		Receive Byte			<ul> <li>Mask</li> </ul>	kedTextBox
				-	Mon 🔛	thCalendar
		1	1		- Noti	iylcon
		🖏 timerUSB			10 Num	rericUpDown
		0 0000			Pictu	reBox
	Solution Evolorer Re Class View				💷 Prog	ressBar
					C Padi	Rutton
	Error List					• # X
	3 0 Errors 2 Warnings 0 0 Messa	iges				
	Description		File	Line	Column	Project
	📑 Error List 🚽 Find Results					
Rea	ły					

Locate the button on the Toolbox, grab and drag the button onto the Form1.cs [Design] and drop it near the top.



File Edit View Project Debug Data Format Tools Window Help         File Edit View Project Debug Data Format Tools Window Help         Solution Explore         Solution Explore         Solution Explore         Solution SPT_Transfer_Test         Solution Explore         Solution Explore         Solution SPT_Transfer_Test         Solution Explore         Solution Explore         Solution SPT_Transfer_Test         Solution Explore         Solution Explo	EPT Transfer Test - Microsoft Visual C# 2010 E	xpress				_ D _X
Solution Explorer <ul> <li>FormLes'</li> <li>FormLes'</li></ul>	File Edit View Project Debug Data For	mat Tools Window Help -   ▶   透 tooltip E   age 20: 00: 02: 1 章 笠 鈴 谷   田 臣	-   Q 雷 火 副 盛 。   1 ] g 明 二 -			
Control Contro Control Control Control Control Control Control Control Control Co	Solution Explorer  Solution Explorer Solution EPT_Transfer_Text () project)  Generation Generation FOT_Loss FormL.cs For	FormLos' FormLosigner.cs' Form	mLcs (Design)" × active_transfer.cs	•	Image: Second Secon	P      P
Error List Column Project	Solution Explorer     Class View				Pictu Prog	ressBar
Description File Line Column Project	Error List	iges		_		<del>-</del> 4 ×
	Description		File	Line	Column	Project

Go to the Properties box and locate the (Name) cell. Change the name to "btnOpenDevice". Locate the Text cell, and change the name to Open.



😫 EPT_Transfer_Test - Microsoft Visual C# 2010 Express				
File Edit View Project Debug Data Format Tools Window Help				
i 🛅 🖽 🕶 🚅 🛃 🕼 🛝 🗠 🖄 🔊 🔹 🔍 🕨 🔯 tooltip 💿 🔹 🖓 🛠 🎦 🚉 🖕				
[尊] [[우리] 교사교 [국민 國帝] ~~ 않 않 않 않 하 [만 문] 집 입 말.				
Solution Explorer PT Transfer Test () project Solution EPT Transfer Test () project Promances Solution EPT Transfer Test () project Promances Solution EPT Transfer Test () project Solution EPT Test () project	ansfer.cs		Toolbox  > All Windows Forms A Common Controls  Common Controls  Pointer  Dutton CneckBox ComboBox CombBox CombBox CombBox CombBox CombBox CombBox CombBox CombBox Co	- # × *
Image: Solution Explorer     Image: Solution Explorer       Image: Solution Explorer </td <td></td> <td></td> <td>Notifylcon NumericUpDov PictureBox ProgressBar RodioRutton</td> <td>m • # ×</td>			Notifylcon NumericUpDov PictureBox ProgressBar RodioRutton	m • # ×
Description F	File	Line	Column Project	
Ready	215,71		<u>⊒</u> ! <u></u> 2 × 4	

Double click on the Open button. The C# Explorer will automatically switch to the Form1.cs code view. The callback function will be inserted with the name of the button along with "\_click" appended to it. The parameter list includes (object sender, System.EventArgs e). These two additions are required for the callback function to initiate when the "click" event occurs.

Private void btnOpenDevice\_click(object sender, System.EventArgs e)

There is one more addition to the project files. Double click on the Form1.Designer.cs file in the Solution Explorer. Locate the following section of code.

```
//
// btnOpenDevice
//
this.btnOpenDevice.Location = new System.Drawing.Point(240, 13);
this.btnOpenDevice.Name = "btnOpenDevice";
this.btnOpenDevice.Size = new System.Drawing.Size(50, 23);
this.btnOpenDevice.TabIndex = 2;
this.btnOpenDevice.Text = "Open";
this.btnOpenDevice.UseVisualStyleBackColor = true;
this.btnOpenDevice.Click += new System.EventHandler(this.btnOpenDevice_Click);
This code sets up the button, size, placement, and text. It also declares the
```

"System.EventHandler()". This statement sets the click method (which is a member of



the button class) of the btnOpenDevice button to call the EventHandler – btnOpenDevice\_Click. This is where the magic of the button click event happens.

```
private void btnOpenDevice_Click(object sender, EventArgs e)
ł
    //Open the Device
    OpenDevice();
}
private void btnCloseDevice_Click(object sender, EventArgs e)
if (EPT_AH_CloseDeviceByIndex(device_index) != 0)
   btnBlkCompare8.Enabled = false;
   btnBlkCompare16.Enabled = false;
   btnTrigger1.Enabled = false;
   btnTrigger2.Enabled = false;
   btnTrigger3.Enabled = false;
   btnTrigger4.Enabled = false;
   btnLEDReset.Enabled = false;
btnOpenDevice.Enabled = true;
btnCloseDevice.Enabled = false;
}
```

When btnOpenDevice\_Click is called, it calls the function "OpenDevice()". This function is defined in the dll and will connect to the device selected in the combo box. This is a quick view of how to create, add files, and add controls to a C# project. The user is encouraged to spend some time reviewing the online tutorial at <a href="http://www.homeandlearn.co.uk/csharp/csharp.html">http://www.homeandlearn.co.uk/csharp/csharp.html</a> to become intimately familiar with Visual C# .NET programming. In the meantime, follow the examples from the Earth People Technology to perform some simple reads and writes to the EPT USB-FPGA Development System.

#### 244.2.2311.2.3 Adding the DLL's to the Project

Locate the EPT FPGA Development System DVD installed on your PC. Browse to the Projects\_ActiveHost folder (choose either the 32 bit or 64 bit version, depending on whether your OS is 32 or 64 bit). Open the Bin folder, copy the following files:

- ActiveHostXX.dll
- ftd2xxXX.dll

and install them in the  $bin\x64\x64$  folder of your EPT\_Transfer\_Demo project.





Organize 💌 🔳 Open with Share with 💌 Bi	um Ne	ew folder			
🅌 Arduino_IDE	^	Name	Date modified	Туре	Size
Documentation		ActiveHost64.dll	3/2/2013 9:44 PM	Application extens	27 K
Drivers		EPT_Transfer_Test.exe	3/2/2013 11:28 PM	Application	28 K
Projects_ActiveHost_32Bit		EPT_Transfer_Test.pdb	3/2/2013 11:28 PM	Program Debug D	56 K
Projects_ActiveHost_64Bit		EPT_Transfer_Test.vshost.exe	3/2/2013 11:21 PM	Application	12 K
ActiveHost_1.0.0.8		EPT_Transfer_Test.vshost.exe.manifest	8/31/2009 12:40 AM	MANIFEST File	1 K
FPT_Data_Collector		S ftd2xx64.dll	1/18/2013 3:54 PM	Application extens	252 K
EPI_Iranster_lest	-				
bin bin	-				
Dahua					
Relaye					
Behun					
🔒 Debug 🔒 Release					

Save the project.

#### 244.2.2411.2.4 Building the Project

Building the EPT\_Transfer\_Demo project will compile the code in the project and produce an executable file. To build the project, go to Debug->Build Solution.



The C# Express compiler will start the building process. If there are no errors with code syntax, function usage, or linking, then the environment responds with "Build Succeeded".





Find Results	
<	
🙀 Error List 🖳 Find Results	
Build succeeded	Ln 1

# **12 FPGA DSO 100M Verilog Description**

- At this point, the following has been accomplished:
  - Description of the DSO 100M hardware
  - Instructions on loading the drivers for the DSO 100M
  - Instructions for loading the Quartus Prime and JTAG \*.dll
  - Instructions on how to create a Quartus project
  - Instructions for loading the Visual Studio Express
  - Description of the FPGA EndTerm interface
  - Description of the PC Active Host interface
  - Instructions on how to create a C# Windows Form

So, now a description of the FPGA code for the DSO 100M can be given. This code description is meant to be thorough, however, it cannot cover every detail of the code. If any questions arise that cannot be determined from code and comments. Please send an email for assistance on the EPT support email:

support@earthpeopletechnology.com

The FPGA code base for the DSO 100M is quite large and comprises several sections.





The following sections of this document will attempt to explain the details of the code. However, a thorough examination of the code will be necessary to come to a complete understanding of the code. Also, there are several testbenches with fully simulated code to help the user. The following components make up the DSO 100M Code Base:

- Host Interface Component
- ADC Sample Storage Component
- Data Transmit Component

These components depend on the Active Transfer Library (and conversely on the Active Host dll) to communicate, initiate functions, start ADC sampling and transmit data.

All source files, compiled projects, testbenches, test models are included on the EPT Project DVD. Locate the Projects\_HDL folder on the DVD for the project.

Page 137 Formatted: Centered



> Drivers	^ Name ^	Date modified	Туре	Size
> Projects_ActiveHost	Altera	6/24/2019 7:38 PM	File folder	
Projects_HDL	EPT_4CE6_AF_D1_Top	6/24/2019 7:38 PM	File folder	
EPT_4CE6_AF_DSO_100M	ModelSim	6/24/2019 7:38 PM	File folder	
Altera	sim	6/24/2019 7:38 PM	File folder	
> EPT 4CE6 AF D1 Top	src	6/24/2019 7:38 PM	File folder	
> ModelSim	test	6/24/2019 7:38 PM	File folder	
sim	Testbench	6/24/2019 7:38 PM	File folder	
src				
test				

## **12.1 Verilog Top Level Description**

The top level of the FPGA code is made up of several modules. Each of these modules are connected to the top level through Leif instantiations. This user manual will use block diagrams, source code, pseudo code and timing diagrams to explain the operation of the code. Below is a simplified abstraction of the top level. The Verilog code is based on a command/response scheme. The PC application sends commands to the FPGA and the FGPA responds with status or data. Both the PC and FPGA code use the EndTerms to send and receive commands and data. The use of EndTerms allows a simple communication system but has a lot of flexibility to add new functions quickly.



#### VERILOG MODULES FOR THE HOST INTERFACE OF THE DSO 100M BOARD



## **12.2 Host Interface Description**

The PC communicates with the FPGA code using one of the following three methods:

- Host Memory Read/Writes
- Host Control Register Read/Writes
- Block data Read (ADC data)



Formatted: Normal

The above block diagram shows the Host Interface Component and the ADC storage registers only. The PC will initiate the Host Read/Write and the Host Control Register Read/Write communications. This means the FPGA code does not send data to the PC unless the PC commands it to. The ADC data is transmitted only when commanded to by the PC.



The above diagram shows the Host Interface Component State Machine







# 12.2.2 Host Interface Control Register







# CONTROL REGISTER STATE MACHINE If(TRANSFER\_3\_RECEIVED) If(ITRANSFER\_4\_RECEIVED) IDLE CR NUMBER RECIEVED CR COMPLETE LATCH CR LATCH CR CONTENTS NUMBER WAIT FOR CR If(TRANSFER\_4\_RECEIVED CONTENTS CONTROL REGISTER WRITE CYCLE CONTROL REGISTER READ CYCLE \_ CONTROL REGISTER NUMBER ACTIVE TRANSFER 3 VALID VALID TRANSFER\_3\_RECEIVED ſ TRANSFER\_3\_START CONTROL REGISTER CONTENTS ACTIVE TRANSFER 4 VALID VALID TRANSFER\_4\_RECEIVED TRANSFER\_4\_START









# 12.3 I2C Communications Interface

#### **I2C COMMUNICATIONS PATH**








	I2C READ CYCLE I2C WRITE CYCLE	
12C		UC CONTROLLER
COMMUNICATIONS	ADDRESS VALID VALID	
	DATA VALID VALID	
	WRITE LATCH	
	READ LATCH	
	OUTPUT ENABLE	
	I2C START	EPT-4CE6-TOP

# 12.4 SPI Communications







#### **12.5 ADC Storage Description** ADC STORAGE AND BLOCK READOUT HIERARCHICAL ABSTRACTION mem\_array\_0 8 x 255 BLOCK CK mem\_array\_1 8 x 255 BLOCK CK ACTIVE TRANSFER LIBRARY mem\_array\_2 8 x 255 BLOCK 10 mem\_array\_3 8 x 255 BLOCK 11 BLOCK ENDTERM СК ADC STORAGE MULTIPLEXOR mem\_array\_4 8 x 255 BLOCK 12 mem\_array\_5 8 x 255 BLOCK 13 mem\_array\_6 8 x 255 BLOCK 14 СК mem\_array\_3 8 x 255 BLOC7 \_15 CK adc\_convst\_block\_count ADC\_BUS\_CH1 ADC\_BUS\_CH2 ADC\_BUS\_CH3 ADC\_BUS\_CH4

Formatted: Centered







Formatted: Normal







# 12.6 ADC Conversion Encode Clock

The ADC\_ENCODE signal is produced by a counter that counts from 0 to adc\_convst\_delay\_value.



```
always @(posedge CLK_260 or negedge RST)
begin
  if(!RST)
  begin
     ADC ENCODE <= 1'b0;
     adc_convst_cmd_count <= 0;</pre>
  end
  else
  begin
     if(adc_convst_cmd_count < adc_convst_delay_value)</pre>
     begin
        adc_convst_cmd_count <= adc_convst_cmd_count + 1'dl;</pre>
     end
     else
     begin
        ADC ENCODE <= !ADC ENCODE;
        adc_convst_cmd_count <= 0;</pre>
      end
  end
end
```

### 12.7 ADC Conversion Start

The ADC conversion command will start the conversion of each of the four ADC's. The command is first set up by the first bit of the adc\_control\_reg, then the fourth bit of the trigger in byte is used to set the adc\_convst\_cmd. To reset the adc\_convst\_cmd, the fourth bit of adc\_control\_reg is set high. The host will write to the control register and set the first bit high. Then the trigger in byte fourth bit is set from the host.





Formatted: Normal

Page 150

adc convst cmd reg <= 1'b0;

end end end



## 12.8 DSO Main Clock

//	
// Instantiate the DSO Main	Clock
DSO Main Clk	DSO MAIN CLOCK INST
(	
.areset	(1'b0),
.inclk0	(aa[1]),
.c0	(CLK_400),
.cl	(CLK_260),
.c2	(CLK_130),
.c3	(CLK 66),
.locked	0
);	

### 12.9 Serial Clock

```
//-----
// Instantiate the Serial Clock
//-----
serial_clock SERIAL_CLOCK_INST
(
.CLK (CLK_66),
.RST_N (RST),
.SERIAL_CLK (i2c_serial_clk)
);
```



# 12.10 Active Transfer Library

active\_transfer\_library ACTIVE\_TRANSFER\_LIBRARY\_INST ( .aa (aa),

.bc\_in .bc\_out .bd\_inout

.UC IN

.UC\_OUT

(aa), (bc\_in), (bc\_out), (bd\_inout),

(UC\_IN), (UC\_OUT)//,

(ft245 test state out),

(active\_transfer\_test\_out)

//.STATE\_OUT //.TEST\_OUT

);

# 13 Compiling, Synthesizing, and Programming FPGA



The FPGA on the DSO 100M can be programmed with the Active Transfer Library and custom HDL code created by the user. Programming the FPGA requires the use of the Quartus Prime software and a standard USB cable. There are no extra parts to buy, just plug in the USB cable. Once the user HDL code is written according to the syntax rules of the language (Verilog and VHDL) it can be compiled and synthesized using the Quartus Prime software. This manual will not focus on HDL coding or proper coding techniques, instead it will use the example code to compile, synthesize and program the FPGA.

# 13.1 Setting up the Project and Compiling

Once the HDL code (Verilog or VHDL) is written and verified using a simulator, a project can be created using Quartus Prime. Writing the HDL code and simulating it

Page 152 Formatted: Left



will be covered in later sections. Bring up Quartus Prime, then use Windows Explorer to browse to c:/altera/xxx/quartus/qdesigns create a new directory called: "EPT\_Transfer\_Demo".

	Irganize * 🔄 Open Include in library *	Share with +	Burn New folder			(22 · ·	
0	and the second s					10 mm	
	in bin	· ·	Name	Date modified	Туре	Size	
	Common		EPT_Transfer_Test	3/13/2013 8-52 PM	File folder		
	cutp		ir_filter	3/3/2013 1:38 AM	File folder		
	drivers		incr_comp_makefile	3/3/2013 1:38 AM	File folder		
	🍐 dsp_builder		vhdl_verilog_tutorial	3/3/2013 1:38 AM	File folder		
	🗼 eda						
	🗼 extibs32						
	🎍 libraries						
	Ja Imf						
	qdesigns     GOT Transfer Text						
	fir filter						
	incr_comp_makefile						
	🕌 vhdl_verilog_tutorial						
	📕 sopc_builder						
	altera_inspector.log.zip						
	I 01	* 0 C C C C C C C C C C C C C C C C C C					
	File folder	413 902 PM					
							_
	C Quartus II 32-bit File Edit View Project Assignments P D D D D D D D D D D D D D D D D D D D	Tocessing Tools	Window Help 🖓	77 <b>600</b> 0	Search	altera.com	
	Quartus II 32-bit File Edit Vew Project Assignments P  G 20 (日本) 10 (日本) 10 (日本) 10 (日本)  File 日本) 10 (日本) 10 (日本)  File 日本) 10 (日本)  Completion Herarchy	rocessing Tools ただがかい ゆきx	Window Help (C) (C) (C) (C) (C) (C) (C) (C)	Take an nline Training (lass for Free	Search	altera.com	4
	C Quartus II 32-bit File: Sat: Wew Project: Assignments P 口 20 日 20 日 20 日 注 10 日 10 10 日 20 日 20 日 20 日 20 日 20 10 日 20 日 20 日 20 日 20 10 日 20 10	rocessing Tools	Vindow Heb 💎	Take an nline Training lass for Free ADDER	Search V V V V Conserver V Software V126 V Software V	altera.com	
	Quartus II 32-bit      File Edit View Project Assignments P     B    B    B    B    B    B    B	rocessing Tools	Window Hep 💎	Take an nline Training lass for Free ADD TERM	Search	atera con	
	C Quartus II 32-bit       File     Satisfy Wave Project: Assignments       D     D D       D     D       D	rocessing Tools → Constant of the first of	Vindow Heb Vindow Heb Vindo	Take an nine training lass for free ADDIER	Concernant Concernant	atera.com tt tt ♥	
	C Quartus II 32-bit  File Edit Vew Project Assignments P  D D D D D D D D D D D D D D D D D D	rocessing Tools	Window Hep (*) Window Hep (*) Window Available Quartus*II Software v120 SP2 Denoted	ADIEN	Courter Cou	altera.com R K ♥	
	Ce Quartus II 32-bit       File     Edit Verre       Pile     Edit Verre       Pile     Pile	rocessing Tools	Vindov Heb (*) * * * * * * * * * * * * * * * * * * *	Take an nine training lass for free ADDIER ADDIER	Search	altera.com	
	Ce Quartus II 32-bit       File     Edit       Dial     Dial       Dial     Dial <tr< td=""><td>Tocessing Tools</td><td>Vindor Heb C Vindor Heb C Vindor Villable Coarting II Software V12.0 SP2 Denoted</td><td>Take an nine training lass for Free ADDER ADDER</td><td></td><td>atera.com</td><td>]</td></tr<>	Tocessing Tools	Vindor Heb C Vindor Heb C Vindor Villable Coarting II Software V12.0 SP2 Denoted	Take an nine training lass for Free ADDER ADDER		atera.com	]
	Quertus II 32-bit      File Eait View Project Assignments IP     Dis IIII IIIIIIIIIIIIIIIIIIIIIIIIIII	rocessing Tools	Vindov Help (*) * * * * * * * * * * * * * * * * * * *	Take an nine training lass for free ALD TER ALD TER ALD TER	Castel	altera.com	]
	Ce Quartus II 32-bit       File     Sati Vew Project Assignments       D     D       D <td< td=""><td>rocesarg Tools ************************************</td><td>Vindov Heb (*) Vite Availabil Coarter V12/0 SP2 Oraclear</td><td>Take an Inine Training Iass for Free ADDIER ADDIER ADDIER</td><td>Carden</td><td>altera.com tt t *</td><td>]</td></td<>	rocesarg Tools ************************************	Vindov Heb (*) Vite Availabil Coarter V12/0 SP2 Oraclear	Take an Inine Training Iass for Free ADDIER ADDIER ADDIER	Carden	altera.com tt t *	]
	Quartus II 32-bit      File Eait View Project Assignments P      D B	rocessing Tools * the fight of	Vindov Help (*) (*) (*) (*) (*) (*) (*) (*)	ADEX ART	Candh Candh Candhar	altera.com tel tel @ left tel @ /ersion 1 uy Software	]
	C: Quartus II 32-bit       File:     Sati Vew Project: Assignments: P       D: D	rocessing Tools * d tot tot tot tot tot tot tot tot tot t	Vindov Heb C	Take an nine training lass for Free ALDIER ALDIER ALDIER	Cardh Cardh Cardha	altera.com et ta de et ta de fersion 1 uy Software wentoed New ervice Page	]
	Quartus II 32-bit      File Est Vew Project Assignments P     D Si II (IIII)     D Si IIII (IIIII)     D Si IIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	rocesang Tools	Vindov Help (*) (*) Cuarture II Softwarrev V12.0 SP2 (*) Cuarture V12.0 SP2 (*) Cuarture V12.0 SP2 (*) Cuarture II (*) (*) (*) (*) (*) (*) (*) (*)	ADER ART	Earch Course C	altera.com eltera	]
	C Quartus II 32-bit File Sat: Vew Project Assignments P D D D D D D D D D D D D D D D D D D D	rocessing Tools * def tool tools * def tool tools * def tools * de	Vindov Heb C	Take an nine training lass for free ALDIER ALDIER ART	Courted Cou	altera.com	]
	Quertur II 32-bit      File Edit View Project Ausgimments IP      Dis Edit () () () () () () () () () () () () ()	rocessing Tools	Vindov Help (*) (*) (*) (*) (*) (*) (*) (*)	ALTER	Cardh Cardhar Cardh	altera.com cil et al. cil et al. (ersion 1) ay Software combace Next commentation	<b>x</b>
	C Quartus II 32-bit File Sat: Vew Project Assignments P P D D D D D D D D D D D D D D D D D D D	rocesarg Tools * * * * * * * * * *	Vindov Heb (*) Vindov Heb (*) Vindov Heb (*) Norr Available Quartus*11 Software viz2.0 SP2 Oranization Oranizi Oranization Oranization Oranization Oranization Ora	Take an nine training lass for free ADDIER ADDIER ADDIER	Control Contro	atera con el el el el el el el el dersion 1 ay Software ay software envice Pack scumentation	]
	Counties II 32-bit      File     Safe View Project Ausgoments     Point	Toolsary Tools	Vindov Hep () () () () () () () () () ()	Take an nine Training lass for free ALGER ART	Cardh Cardon	altera con altera	<b>x</b>
	C Quertus II 32-bit File Eait View Project Assignments P D D D D D D D D D D D D D D D D D D D	rocessing Tools * * * * * * * * * *	Vindov Help (*) * * * * * * * * * * * Norr Available Quartus* II Software v72.0 SP2 * * * * * * * * * Output: Out	ADER	Cardh Cardh Card	ettera con ettera	<b>x</b> ]
	C Quartus II 32-bit File Est: Vew Project Assignments P D D D D D D D D D D D D D D D D D D D	roceaning Tools		алена д ну л	Carden	المعنى المعنى المعنى المعنى المعن المعنى المعنى المعن المعنى المعنى المعن المعنى المعنى المعن المعنى المعنى المعن المعنى المعنى	]

<u>Under Quartus, Select File->New Project Wizard. The Wizard will walk you through</u> setting up files and directories for your project.





At the Top-Level Entity page, browse to the c:/altera/xxx/quartus/qdesigns directory to store your project. Type in a name for your project "EPT 4CE6 AF D1 Top".







Select Next. At the Add Files window: Browse to the
\Projects_HDL\EPT_Transfer_Demo \src folder of the EPT FPGA Development
System DVD. Copy the files from the \src directory.

- Active block.vqm
- Active\_transfer.vqm
- Active\_trigger.vqm
- Active\_transfer\_library.v
- ft\_245\_state\_machine.v
- endpoint\_registers.vqm
- eptWireOr.v
- mem\_array.v
- read\_control\_logic.v
- write control logic.v
- EPT 4CE6 AF D1 Top.v

acgory.			Device		
General	Files				
Files	Select the design files you want to include in the project. Click Add All to add all design files in the project directory	to the project.			
<ul> <li>Operating Settings and Conditions Voltage</li> </ul>	File name:		Add		
Temperature		Type			
Early Timing Estimate	/EPT US8-CPLD Development System CD/Projects_HDL/EPT_Transfer_Test/src/write_control_logic.v	Verilog HDL File	AUU AI		
Incremental Compliation	EPT US8-CPLD Development System CD/Projects_HDL/EPT_Transfer_Test/src/read_control_logic.v	Verilog HDL File	Remove		
Physical Synthesis Optimizations	EPT USB-CPLD Development System CD/Projects_HDL/EPT_Transfer_Test/src/mem_array.v Verlog HDL File				
Design Entry/Synthesis	EPT US8-CPLD Development System CD/Projects HDL/EPT Transfer Test/src/EPT 570 AP U2 Top.v	Verilog HDL File Verilog HDL File			
Simulation	EPT US8-CPLD Development System CD/Projects_HDL/EPT_Transfer_Test/src/active_trigger.vgm	Verilog Quartus Ma	Down		
Formal Verification	VEPT US8-CPLD Development System CD/Projects_HDU/EPT_Transfer_Test/src/active_transfer_library.vgm	Verilog Quartus Ma	Propertie		
Analysis & Synthesis Settings	IEPT USB-CPLD Development System CD/Projects_PDC/EPT_Transfer_Test/src/active_transfer.vqm	Verilog Quartus Ma			
VHDL Input					
Verilog HDL Input					
Fitter Settings					
TimeQuest Timing Analyzer					
Assembler					
SignalTap II Logic Analyzer					
Logic Analyzer Interface					
PowerPlay Power Analyzer Settings					
SSN Analyzer					

Select Next, at the Device Family group, select Cyclone IV for Family. In the Available Devices group, browse down to EP4CE6E22C8 for Name.



Device family					Show in 'Availa	able devices' list		
Family: Cyclo	Family: Cyclone IV E					Any		-
Devices: All					Dia anumb	A		_
Devices: All 👻					Pin count:	Any		•
Target device					Speed grade:	Any		•
Auto devic	e selected by the Fi	tter			Name filter:			
	e selected by the fi							
Other: n/a	a 5:							
Other: n/a vailable devices	s: Core Voltage	LEs	User I/Os	Memory	Bits Em	bedded multiplier 9-bit elements	PLL	GI
Other: n/a wailable devices Name EP4CE6E22A7	s: Core Voltage 1.2V	LEs 6272	User I/Os 92	<b>Memory</b> 276480	Bits Em	bedded multiplier 9-bit elements	<b>PLL</b> 2	<b>GI</b> 10
Other: n/i wailable devices Name EP4CE6E22A7 EP4CE6E22C6	s: Core Voltage 1.2V 1.2V	LEs 6272 6272	<b>User I/Os</b> 92 92	Memory 276480 276480	Bits Em 30 30	bedded multiplier 9-bit elements	<b>PLL</b> 2 2	GI 10 10
Other: n/i wailable devices Name EP4CE6E22A7 EP4CE6E22C6 EP4CE6E22C7	s: Core Voltage 1.2V 1.2V 1.2V 1.2V	LEs 6272 6272 6272	<b>User I/Os</b> 92 92 92	Memory 276480 276480 276480	Bits Em 30 30 30 30	bedded multiplier 9-bit elements	PLL 2 2 2	GI 10 10 10
Other: n/s wailable devices Name EP4CE6E22A7 EP4CE6E22C6 EP4CE6E22C7 EP4CE6E22C8	Core Voltage 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V	LEs 6272 6272 6272 6272	User I/Os 92 92 92 92 92	Memory 276480 276480 276480 276480 276480	Bits Em 30 30 30 30 30 30	bedded multiplier 9-bit elements	PLL 2 2 2 2 2	GI 10 10 10 10
Other: n/i vailable devices Name EP4CE6E22A7 EP4CE6E22C6 EP4CE6E22C7 EP4CE6E22C8 EP4CE6E22C8	Core Voltage 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V 1.0V	LEs 6272 6272 6272 6272 6272 6272	User I/Os 92 92 92 92 92 92 92	Memory 276480 276480 276480 276480 276480 276480	Bits Em 30 30 30 30 30 30 30	bedded multiplier 9-bit elements	PLL 2 2 2 2 2 2 2 2	GI 10 10 10 10 10
Other: n/: vailable devices Name EP4CE6E22A7 EP4CE6E22C6 EP4CE6E22C7 EP4CE6E22C8 EP4CE6E22C8L EP4CE6E22C9L	Core Voltage 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V 1.0V 1.0V	LEs 6272 6272 6272 6272 6272 6272 6272	User I/Os 92 92 92 92 92 92 92 92 92 92	Memory 276480 276480 276480 276480 276480 276480 276480	Bits Em 30 30 30 30 30 30 30 30 30	bedded multiplier 9-bit elements	PLL 2 2 2 2 2 2 2 2 2 2 2 2 2	GI 10 10 10 10 10 10
Other: n/: vailable device: Name EP4CE6E22A7 EP4CE6E22C6 EP4CE6E22C7 EP4CE6E22C8 EP4CE6E22C8 EP4CE6E22C9L EP4CE6E22C9L EP4CE6E2217	Core Voltage 1.2V 1.2V 1.2V 1.2V 1.2V 1.0V 1.0V 1.0V 1.2V	LEs 6272 6272 6272 6272 6272 6272 6272 627	User I/Os 92 92 92 92 92 92 92 92 92 92 92	Memory           276480           276480           276480           276480           276480           276480           276480           276480	Bits Em 30 30 30 30 30 30 30 30 30 30	bedded multiplier 9-bit elements	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	GI 10 10 10 10 10 10 10
Other: n/i wailable device: Name EP4CE6E22A7 EP4CE6E22C6 EP4CE6E22C7 EP4CE6E22C8 EP4CE6E22C8L EP4CE6E22C8L EP4CE6E22C8L EP4CE6E2217 EP4CE6E2218L	S: Core Voltage 1.2V	LEs 6272 6272 6272 6272 6272 6272 6272 627	User I/Os 92 92 92 92 92 92 92 92 92 92	Memory           276480           276480           276480           276480           276480           276480           276480           276480           276480           276480           276480           276480           276480	Bits Em 30 30 30 30 30 30 30 30 30 30	bedded multiplier 9-bit elements	PLL 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	GI 10 10 10 10 10 10 10 10
Other: n/i vailable device: Name P4CE622A7 P4CE622C7 P4CE622C7 P4CE622C7 P4CE622C7 P4CE622C7 P4CE622C7 P4CE62227 P4CE62217 P4CE62217 P4CE62217	s: 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V 1.0V 1.0V 1.0V 1.2V 1.	LEs 6272 6272 6272 6272 6272 6272 6272 627	User I/Os 92 92 92 92 92 92 92 92 92 92 92 92 92	Memory 276480 276480 276480 276480 276480 276480 276480 276480 276480	Bits Em 30 30 30 30 30 30 30 30 30 30 30 30 30	bedded multiplier 9-bit elements	PLL 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	GI 10 10 10 10 10 10 10 10 10

Select Next, leave defaults for the EDA Tool Settings.



Tool Name <none> <none> Timing Symbol Signal Integrity Paundary Case</none></none>	Format(s)	* *	Run Tool Automatically Run this tool automatically to synthesize the Run gate-level simulation automatically afte
<none> </none>		* * *	Run this tool automatically to synthesize the Run gate-level simulation automatically afte
<none> <none> Timing Symbol Signal Integrity Devedoes Core</none></none>	<	* *	Run gate-level simulation automatically afte
<none> Timing Symbol Signal Integrity</none>	<pre> </pre> </td <td>•</td> <td></td>	•	
Timing Symbol Signal Integrity	<none></none>	•	
Symbol Signal Integrity	<none></none>	*	
Signal Integrity			
David and Care	<ivone></ivone>	-	
boundary Scan	<none></none>	+	

Select Next, then select Finish. You are done with the project level selections.

Summary [page 5 of 5]	
When you click Finish, the project will be created	d with the following settings:
Project directory:	C:/altera/12.0sp1/quartus/qdesigns
Project name:	Active_Transfer_Example
Top-level design entity:	Active_Transfer_Example
Number of files added:	7
Number of user libraries added:	0
Device assignments:	
Family name:	MAX II
Device:	EPM570T100C5
EDA tools:	
Design entry/synthesis:	<none> (<none>)</none></none>
Simulation:	<none> (<none>)</none></none>
Timing analysis:	0
Operating conditions:	
VCCINT voltage:	3.3V
Junction temperature range:	0-85 °C
	A Rade Next S Einich Cancel Hale





Next, we will select the pins and synthesize the project.

## 13.1.1 Selecting Pins and Synthesizing

With the project created, we need to assign pins to the project. The signals defined in the top level file (in this case: EPT 4CE6 AF D1 Top.v) will connect directly to pins on the FPGA. The Pin Planner Tool from Quartus Prime will add the pins and check to verify that our pin selections do not violate any restrictions of the device. In the case of this example we will import pin assignments that created at an earlier time. Under Assignments, Select Import Assignments.



<u>At the Import Assignment dialog box, Browse to the</u> <u>\Projects HDL\EPT Transfer Demo \ DSO 100M TOP folder of the EPT FPGA</u> <u>Development System DVD. Select the "DSO 100M\_Top.qsf" file.</u>





	Import Assignments	×
Specify the	e source and categories of assignments to import.	
File name:	: C:/Jolly/Code_FPGA/EPT_4CE6_AF_Transfer_Demo/EPT_4CE6_AF_Transfer_Demo/EPT_4CE6_AF_D1_Top/EPT_4CE6_AF_D1_Top.qsf	Categories
🖌 Copy e	existing assignments into EPT_4CE6_AF_D1_Top.qsf.bak before importing	Advanced
	OK Cancel	Help

Click Ok. Under Assignments, Select Pin Planner. Verify the pins have been imported correctly.





The pin locations should not need to be changed for EPT USB FPGA Development System. However, if you need to change any pin location, just click on the "location" column for the particular node you wish to change. Then, select the new pin location from the drop down box.





Exit the Pin Planner. Next, we need to add the Synopsys Design Constraint file. This file contains timing constraints which forces the built in tool called TimeQuest Timing Analyzer to analyze the path of the synthesized HDL code with setup and hold times of the internal registers. It takes note of any path that may be too long to appropriately meet the timing qualifications. For more information on TimeQuest Timing Analyzer, see

http://www.altera.com/literature/hb/qts/qts\_qii53018.pdf?GSA\_pos=1&WT.oss\_r=1& WT.oss=TimeQuest Timing Analyzer



Browse to the \Projects_HDL\EPT_T	ransfer_De	mo \ DSO 10	0M_TOP t	folder o	of the
EPT FPGA Development System DV	D. Select t	he "DSO 100	M Top.sd	lc" file	
IIII = IIIIIIIIIIIIIIIIIIIIIIIIIII	gy\DUEPROLOGIC_USB	_FPGA_PROJECT_1.1_DVD\F	Projects_HDL\EPT		- 🗆 ×
File Home Share View					^ <b>(</b> )
Copy Paste  Paste shortcut  ↓  ↓  ↓  ↓  ↓  ↓  ↓  ↓  ↓  ↓  ↓  ↓  ↓	s • Properties • History	Select all Select none			
Clipboard Organize New	Open	Select			
	EPT_4CE6_AF_Transfer_Demo	→ EPT_4CE6_AF_D1_Top →	v C	Search EPT_4CE	6_AF_D1_Top 🔎
Drivers	^ Name	*	Date modified	Туре	Size
Projects_ActiveHost	👪 db		10/8/2015 3:02 PM	File folder	
Projects_Arduno	incremental_db		10/8/2015 3:02 PM	File folder	
FPT 4CF6 AF Transfer Demo	output_files		10/8/2015 3:02 PM	File folder	
FPT 4CF6 AF D1 Top	EPT_4CE6_AF_D1	_Top.jdi	2/14/2014 12:17 AM	JDI File	1 KB
a db	EPT_4CE6_AF_D1	_Top.qpf	11/4/2013 7:28 PM	QPF File	2 KB
incremental db	EPT_4CE6_AF_D1	_Top.qsf	10/8/2015 2:46 PM	QSF File	9 KB
output_files	EPT_4CE6_AF_D1	_Top.sdc	10/12/2015 4:00 PM	SDC File	4 KB
ModelSim	EP1_4CE6_AF_D1	_l op_assignment_defaults.qdf	5/19/2015 4:55 PM	QDF File	47 KB
\mu sim					
🕌 src					
) test					
🕌 Testbench	*				_
8 items   1 item selected 3.12 KB   State: 🐉 Shared					§== 🖬

Copy the file and browse to c:\altera\xxx\quartus\qdesigns\EPT Transfer Demo directory. Paste the file.

nize 🔻 🗋 Open Burn New folde					
EPT_Transfer_Test fr_filter incr_comp_makefile vhdl_veniog_tutorial incr_comp_huider abera_inspector.log.sip 91 book	*	Name db in crematal_db output_files isimulation EPT_370_AP_U2_Top.phi EPT_370_AP_U2_Top.qpf EPT_370_AP_U2_Top.qpf	Date modified 3/13/2013 9:42 PM 3/13/2013 9:38 PM 3/13/2013 9:42 PM 3/13/2013 9:42 PM 3/13/2013 9:42 PM 3/13/2013 9:42 PM 3/13/2013 9:38 PM	Type File folder File folder File folder JDT File QPF File QSF File	5ize 1 K5 2 K3 7 K3
CONTEMP Cocuments and Settings July Business Opportunities Capitol_College Code_FPGA Documents Documents		EPT_570_AP_U2_Top.sdc	3/13/2013 9-41 PM 1/29/2013 10-30 PM	BAK File SDC File	7 x3. 4 x3

Select the Start Compilation button.



S Quartus II 64-Bit - C:/Jolly/Code_FPGA/EPT_4	4CE6_AF_Tra	ansfer_Demo/EPT_4CE6_AF_Trans	sfer_Demo/EPT_4CE6_AF_D1_Top/EPT_4CE6_	AF_D1_Top - EP 🗆 🛛
File Edit View Project Assignments Processing T	ools Window	Help		Search altera.com
🗋 🥁 🔄 🍠 👗 🗈 🛍 🖃 (* 🛛	_D1_Top	- 😿 😢 🦯 🏈 🧇 🔅	🗩 🖉 छ छ 👯 🚓 🗣 👗 🕘	7
Project Navigator 🗘 🗗 🗙	1	Home		
Entity				^
Cydone IV E: EP4CE6E22C8				
Page EPT_4CE6_AF_D1_Top *(A)	Star	t Designing		
			Wah us Subscription	
		* -	Web vs Subscription	
<			Edition	
🛆 Hierarchy 📄 Files 🔗 Design Units 🏾 🌂 🕨			Buy Software	
Tasks 🛛 🖓 🛪		New Project Wizard		
Flow: Compilation			Documentation	
Task				
?  Comple Design			Training	
Analysis & Synthesis     Eltter (Place & Route)				
?   Assembler (Generate programming files)			Support	v
< >>	<			>
🗙 🗚 🙆 🛕 🔏 💎 < <search>&gt;</search>		~		
I Type ID Message				
22				
20 <				>
System (Processing)				
Starts a new compilation				0% 00:00:00 .:

If you forget to include a file or some other error you should expect to see a screen similar to this:

Quartus II 64-Bit - C:/Jolly/Code_FPGA/EPT_4	4CE6_AF_Transfer_Demo/EPT_4CE6_	AF_Transfer_Demo/EPT_4CE	6_AF_D1_Top/EPT_4CE6_AF_D1_	_Top - EP 🗧	- ×
File Edit View Project Assignments Processing Tr	ools Window Help 🐬			Search altera.com	٠
🗋 😂 🗔 🥔 🐰 🗈 🛍 🤟 🖓 🗧	_D1_Top 🔹 🐹 🐓 🏒 🧳 🤅	🖉 🤣 💷 🕨 🧭 🕲 🛈 🕅	2 😤 🥥 👋 👗 💿 🖘		
Project Navigator 🛛 🖗 🛪 🗙	1 Home	Compilation Report - EPT	4CE6_AF_D1_Top 🔀		
Entity Cyclone IV E: EP4CE622C8  Cyclone IV E: EP4CE622C8  Herarchy Entity Fasts Flow: Compliation Task Customize Task Compleation Comp	Table of Contents     4 Ø       Image: Tow Summary     Image: Tow Summary       Image: Tow Edgesd Time     Image: Tow Edgesd Time       Image: Tow Edgesd Time     Tow Edgesd Time       Image: Tow Edgesd Time     Image: Tow Edgesd Time	Tow Stammary Flow Status Quartus II 64 Ht Version Revision Name Top-level Entity Name Family Device Total logic elements Total onchraitional functions Dedicated logic registers Total incident Total money total Entedded Multipler 9-bit elements Total PLLS	Flow Failed - Mon Oct 12 16:05:54 2015           13.1.0 Eudi 162 10/23/2013 31 Web Edit           ETT_CES_AP_D_1_Top           ETT_CES_AP_D_1_Top           ETT_CES_AP_D_1_Top           ETT_CES_AP_D_1_Top           ETT_CES_AP_D_1_Top           ETT_CES_AP_D_1_Top           ETT_CES_AP_D_1_Top           ETT_CES_AP_D_1_Top           ETA_DES_CES_CES           Find           VA.antif Aration Merge           VA.antif Aration Merge	ton	
	rarchy "lfsr:LFSR_INST" sages file C:/Jolly/Code_FFGA. is 6 Synthesis was unsuccessful. 4 error:	/EPT_4CE6_AF_Transfer_D al. 2 errors, 11 warnin s, 11 warnings	emo/EFT_4CE6_AF_Transfer_D gs	emo/EPT_4CE6_X	۲۲_D1_
				11%	00:00:02



Click Ok, the select the "Error" tab to see the error.

SQuartus II 64-Bit - C:/Jolly/Code_FPGA/EPT_	ACE6_AF_Transfer_Demo/EPT_4CE6_AF_Tra	nsfer_Demo/EPT_4CE6_AF_D1_Top/EPT_4C	E6_AF_D1_Top - EP 🗕 🗖 🗙
File Edit View Project Assignments Processing T	ools Window Help 🐬		Search altera.com
🗋 💕 🚽 🥔 🐰 🖙 🖎 🤟 (*) 👘 EPT_4CE6_AR	_D1_Top 🔹 🐹 🐓 🍠 🍣 🧇	💿 🕨 🧭 🖄 🕲 😫 🤹 🖉	9 7
Project Navigator 부 중 ×	home 🗙 🕯	Compilation Report - EPT_4CE6_AF_D1_Top 🛛	
Entity	Table of Contents	ummary	
Cyclone IV E: EP4CEE2223  Herarchy File Files F	Too Summary Too Settings Too Non-Defaul Global Settings Too Non-Defaul Global Settings Too Non-Defaul Global Settings Too Log Too Log Too Log Poor Log Poor Log Poor Massages Poor Massages Poor Massages Total In Total IN	state         Flow Falled - Mon Oct 12 If UII 64-81 Version           III 64-81 Version         13.10 dublics 12 0/23/2001           Name         EPTCEG_AF_D_I_Top al Entity Name           EPT_ACEG_AF_D_I_Top al Entity Name         EPT_ACEG_AF_D_I_Top Cryclone IV E           EPT_ACEG_AF_D_I_Top al combinational functions         N/A until Partition Merge states in N/A unil Partition Merge states N/A unil Partition Merge tail pins           N/A unil Partition Merge tail pins         N/A unil Partition Merge N/A unil Partition Merge Lid Multiplier 9-bit elements           N/A unil Partition Merge Lid         N/A unil Partition Merge	5:05:54:2015
< , , , , , , , , , , , , , , , , , , ,			
X     X <td>v ntiation <u>sprat at 155</u>, v(31): camm rarchy <u>fistifues</u> (1957) is 6 Synthesis was unsuccessful. 2 ion was unsuccessful. 4 errors. 11</td> <td>t elaborate array of instances beca CE6_AF_Transfer_Demo/EPT_4CE6_AF_Tr errors, 11 warnings warnings</td> <td>use the declaration for the ansfer_Demo/EFT_4CE6_AF_D1_ &gt;</td>	v ntiation <u>sprat at 155</u> , v(31): camm rarchy <u>fistifues</u> (1957) is 6 Synthesis was unsuccessful. 2 ion was unsuccessful. 4 errors. 11	t elaborate array of instances beca CE6_AF_Transfer_Demo/EPT_4CE6_AF_Tr errors, 11 warnings warnings	use the declaration for the ansfer_Demo/EFT_4CE6_AF_D1_ >

The error in this case is the missing file "sync\_fifo". Click on the Assignment menu, then select Settings, then select Files. Add the "sync\_fifo.v" file from the database.



2	Settings - EPT_4C	E6_AF_D1_Top			
Category:					Device
General Files Ubraries Operating Settings and Conditions Voltage Temperature Compliation Process Settings Early Timing Estimate Incremental Compliation Physical Synthesis Optimizations EDA Tool Settings	Files Select the design files you want to indu project. File name:/sc/fisr.v File Name /src/mux.v BPT_4CE6_AF_D1_Top.sdc /src/chue_thigger.vgm	de in the project. Click Add All to a Type Verlog HDL File Synopsys Design Constraints File	dd all des Library	ign files in the project dir Design Entry/Synthes <none> <none> <none></none></none></none>	Add All Remove
Design Entry/Synthesis Simulation Formal Verification Board-Level Analysis A Synthesis Settings VHDL Input Veriol of NDL Input Default Parameters Fitter Settings TimeQuest Timing Analyzer Assembler Design Assistant SignaTap II Logic Analyzer Logic Analyzer Interface PowerRay Power Analyzer Settings SSN Analyzer	/src/flpflop.v /src/flpflop.v /src/endpoint_registers.vgm /src/active_transfer_lbrary.v /src/active_transfer.vgm /src/active_block.vgm /src/active_block.vgm /src/prite_control_logic.v /src/prite_control_logic.v /src/prite_control_logic.v /src/b_define.v /src/b_define.v /src/b_define.v /src/b_define.v /src/b_define.v /src/b_define.v /src/b_define.v /src/b_define.v /src/b_define.v /src/b_define.v /src/b_define.v /src/b_define.v	Verlog HDL File Verlog HDL File Verlog Quartus Mapping File Verlog Quartus Mapping File Verlog Quartus Mapping File Verlog Quartus Mapping File Verlog HDL File Chain Description File		</td <td>Up Down Properties</td>	Up Down Properties
	<	Buy Software OK	Ca	ncel Apply	Help

<u>Click Ok then re-run the Compile process. After successful completion, the screen should look like the following:</u>



🔩 Quartus II 64-Bit - C:/Jolly/Code_FPGA/EPT	4CE6_AF_Transfer_Demo/EPT_4CE6_	AF_Transfer_Demo/EPT_4CE	6_AF_D1_Top/EPT_4CE6_AF_D1_T	Гор - ЕР –	
File Edit View Project Assignments Processing T	ools Window Help 🗬			Search altera.co	om 🚯
🗋 💕 🚽 🥔 👗 🖻 🖏 🤟 (*) 😢 EPT_4CE6_AF	_D1_Top 🔹 🐹 🔮 🥒 🧳 🎚	🖗 🧶 💿 🕨 🤣 🕲 🗘	2 😪 🥥 🧶 🕹 🔍 🔻		
Project Navigator 🛛 🖗 🛎 🗙	1 Home	Compilation Report - EPT,	_4CE6_AF_D1_Top 🔀		
Entity	Table of Contents 🕴 🖉	Flow Summary			
Cyclone IV E: EP4CE6E22C8	E Flow Summary	Flow Status	Successful - Mon Oct 12 16:10:04 2015		
EPT_4CE6_AF_D1_Top 4	Flow Settings	Quartus II 64-Bit Version	13.1.0 Build 162 10/23/2013 SJ Web Editio	'n	
	Flow Non-Default Global Settings	Top-level Entity Name	EPT_4CE6_AF_D1_Top		
	Flow Elapsed Time	Family	Cyclone IV E		
	Flow OS Summary	Device	EP4CE6E22C8		
	Flow Log	Timing Models	Final		
	Analysis & Synthesis	Total logic elements	632 / 6,272 ( 10 % )		
August Bart dans um Mich	🕨 🧰 Fitter	Pedicated logic registers	594 / 6,272 ( 9 % ) 411 / 6 272 ( 7 % )		
A Herarchy E Hies of Design Units	Flow Messages	Total registers	411		
Tasks 0.6 ×	Flow Suppressed Messages	Total pins	86 / 92 ( 93 % )		
Flow: Compilation	Assembler	Total virtual pins	0		
	🕨 🚞 TimeQuest Timing Analyzer	Total memory bits	2,048 / 276,480 ( < 1 % )		
Task ^		Embedded Multiplier 9-bit elements	0/30(0%)		
🖌 🔺 🕨 Comple Design		TOTALPELS	0/2(0%)		
V 🕨 Analysis & Synthesis					
V Fitter (Place & Route)					
Assembler (Generate programming files) v					
< >>	< >				
× ▲ ② ▲ ▲ <	~				
<sup>‡</sup> Type ID Message					^
332102 Design is not fully cons	trained for setup requirement	3			
332102 Design is not fully cons	trained for hold requirements				
► ↓ Overtue II of bit limegr	eso Timing Incluzer was succe	ssful. O errors, 10 war	nings		
293000 Quartus II Full Compilat	ion was successful. 0 rrors,	59 warnings			
őess <					>
System Processing (185)					
				100%	00:00:31

At this point the project has been successfully compiled, synthesized and a programming file has been produce. See the next section on how to program the FPGA.

### 13.1.2 Configuring the FPGA

<u>Configuring the FPGA is quick and easy. All that is required is a standard USB Micro B</u> <u>cable and the EPT\_Blaster Driver DLL. Connect the DSO 100M to the PC, open up</u> <u>Quartus Prime, open the programmer tool, and click the Start button. To program the</u> <u>DPL Configuration Flash, follow the steps to install the USB Driver and the JTAG</u> Driver Insert for Quartus Prime.







If the project created in the previous sections is not open, open it. Click on the Programmer button.

🚳 Quartus II 64-Bit - C:/Jolly/Code_FPGA/EPT_4	ACE6_AF_Transfer_Demo/EPT_4CE6_	AF_Transfer_Demo/EPT_4CE	6_AF_D1_Top/EPT_4CE6_AF_D1_T	Top - EP	
File Edit View Project Assignments Processing Te	ools Window Help 💎		_	Search altera.co	m 🚯
🗋 💕 😹 🍠 🐰 🖓 🖏 🖏 🖉 (*) 😢 📴 EPT_4CE6_AF	_D1_Top 🔹 💢 🐓 🏒 🏈 🖗	😻 🤣 💿 🕨 🤯 🖄 🛈 🖡	****		
Project Navigator 🕴 🗗 🗙	h Home	Compliation Report - EPT	4CE6_AF_D1_Top 🔀		
Entity	Table of Contents	Flow Summary			
Cyclone IV E: EP4CE622C8     Merenchy      Files J <sup>P</sup> Design Units	Flow Summary     Flow Summary     Flow Settings     Flow Incode Eduk Global Settings     Flow Log Eduk Global Settings     Flow Log     Flow Log     Flow Log     Flow Log     Flow Log     Flow Summary     Flow Log     Flow Supposed Messages     Flow Supposed Messages     Flow Supposed Messages     Flow Supposed Messages     Flow Supposed Messages	Row Status Quartus II 64-81 Version Revision Name Top-level Entity Name Family Device Timing Models Total logic elements Total combinational functions Deckate dogic registers Total progressers Total PLLs	$\begin{split} & \text{Successful} - \text{Non Cct 12 is 10:04 2015} \\ & \text{S1.0 Buil ds 2012320133 Veb Eddes} \\ & \text{EPT_accEa, AF, D_1 Top} \\ & \text{EPT_accEa, AF, D_1 Top} \\ & \text{Cyclore IV E} \\ & \text{EP4ccEs2CS} \\ & \text{Final} \\ & \text{EP4ccEs2C3} \\ & \text{Final} \\ & \text{S1} (1, 272 (10 \%) \\ & \text{S9} (4, 272 (10 \%) \\ & \text{S1} (1, 272 (1, 273$	'n	
< >	< >				
<ul> <li>X M O A A A A A A A A A A A A A A A A A A</li></ul>	trained for setup requirement trained for hold requirements est Timing Analyzer was succe ion was successful. 0 errors,	s ssful. 0 errors, 10 war 59 warnings	nings		^
System         Processing (185)           Opens a Programmer window				100%	> 00:00:31

The Programmer Window will open up with the programming file selected. Click on the Hardware Setup button in the upper left corner.

vare Setup No I	lardware					Mode:	JTAG		<ul> <li>Pro</li> </ul>	ogress:		
real-time ISP to allo	w background program	nming (for MAX II and	MAX V devices)									
Start	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine	Security Bit	Erase	CLAMP	
Stop												
Detect												
elete												
file												
ge File												
ve File												
Device												
Up												
lown												





The Hardware Setup Window will open. In the "Available hardware items", double click on "EPT-Blaster v1.5b".

hardware setup app	lies only to the cur	rent programmer	window.	mis programming
Currently selected h	ardware: No Ha	ardware		
Available hardwar	e items			
Hardware		Server	Port	Add Hardware
				Remove Hardware

If you successfully double clicked, the "Currently selected hardware:" dropdown box will show the "EPT-Blaster v1.5b".

Currently : Available	selected hardware	EPT-Blast	er v1.5b (64)	[MBUSB-0]	> ·
Hardwa	are		Server	Port	Add Hardware
EPT-Bla	ster v1.5b (64)		Local	MBUSB-0	Remove Hardware





Click on the "Mode:" drop down box. Select the "Active Serial Programming" option.



### Click on the "Add File" button

EPT-Blaster v1.5b (64) ISP to allow background progr	[MBUSB-0] amming (for MAX II and	MAX V devices)			Mode: Active Senal Programming						
Fle	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine	Security Bit	Erase	ISP CLAMP	





۳	Select Pi	ogramming File				×
Look in:	\Jolly\Code_FPGA\EPT_4CE6_Ansf	er_Demo\EPT_4CE6_AF_D1_Top <	G	00	1	
My Computer	Name	Size Typ	pe	Date Modifie	ed	
lanelso_000	🌗 db	File	lder	10/12/2015	4:10:10 PM	1
	output_files	File	lder	10/8/2013 9	4:10:10 PM	1
	<					>
File name:					Open	
Files of type: POF Fil	es (*.pof)			•	Cancel	

# At the Browse window, double click on the output files folder.

\$	Select Program	ming File	×
Look in:	\Jolly\Code_FPGA\EPT_4CE6_AFPT_4CE6_AF	_D1_Top\output_files 🔻 🔇 🔅	) 🗿 📙 🗉 🔳
My Computer	Name	Size Type Dat	te Modified
k nelso_000	EPT_4CE6_AF_D1_Top.pof	128 KB pof File 10/	12/2015 4:10:04 PM
	<		>
File name: EPT_4	CE6_AF_D1_Top.pof		Open
Files of type: POF Fi	les (*.pof)		✓ Cancel





Double click on the "EPT\_4CE6\_D1\_Top.pof" file. Click the Open button in the lower right corner.

#### Select the EPCS1 under "Device".



Next, selet the checkbox under the "Program/Configure" of the Programmer Tool.





<u>Click on the Start button to to start programming the FPGA. The Progress bar will indicate the progress of programming.</u>



When the programming is complete, the Progress bar will indicate success.



able real-time I	SP to allow background program	nming (for MAX II an	d MAX V devices)			Mode:	neure sene		- PR	J. 630.	1007	e (ouccessio
No Start	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine	Security Bit	Erase	ISP CLAMP	
ili Stop	output_files/EPT_4CE6_A	EPCS1	00755C55	00000000	-							
Auto Detect												
Palata												
Delete												
Add File												
hange File												
Save File												
dd Device												
1 <sup>10</sup> Up												
Ne Down	ASDI V											
	EPCS1											
	↓ 0010											

At this point, the DSO 100M is programmed and ready for use. To test that the FPGA is properly programmed, bring up the Active Transfer Demo Tool. Click on one of the LED's and verify that the LED selected lights up. Press one of the switches on the board and ensure that the switch is captured on the Active Host Test Tool. Now you are ready to connect to the Arduino Due and write some code to transfer data between microcontroller and PC.

#### 244.2.25—

Once the project has been successfully built, it produces an \*.exe file. The file will be saved in the Release or Debug folders.

ganize 👻 📰 Open Share with 👻 E	urn New fol	der			
Arduino_IDE	*	Name	Date modified	Туре	Size
Documentation		ActiveHost64.dll	3/2/2013 9:44 PM	Application extens	27 KB
Drivers		EPT_Transfer_Test.exe	3/2/2013 11:28 PM	Application	28 KB
Projects_ActiveHost_32Bit		EPT_Transfer_Test.pdb	3/2/2013 11:28 PM	Program Debug D	56 KB
Projects_ActiveHost_64Bit		EPT_Transfer_Test.vshost.exe	3/2/2013 11:21 PM	Application	12 KB
ActiveHost_1.0.08		EPT_Transfer_Test.vshost.exe.manifest	8/31/2009 12:40 AM	MANIFEST File	1 KB
EPT_Data_Collector		ftd2xx64.dll	1/18/2013 3:54 PM	Application extens	252 KB
EPI_Iranster_lest	-				
Line CP 1_ITANSIEF_LESC					
Bebug					
Belease					
¥64					
Debug					
Release					
i obj					
Propertier.	*				





EPT USB <-> Serial&JTAG	i Cab 🗸 🛛 🗤		Close	1			
	Ope	n	Close				
Transfer Controls							
Send Byte: 255	Address 2 Red	ceive Byte		^			
Byte	LoopBack						
Multiple Byte: 255							
Multi Data		L		$\sim$			
Multi Byte			Rst				
	LED Controls						
Switches	LED's		1	2 3	4	5	6
Switch Controls	Random		7	8 9	10	11	12
	Shift Left		13	14 15	16	17	18
	Chift Dialat		19	20 21		23	24
Rst	Shift Right			20 21		2.5	24
	Shift Up		25	26 27	28	29	30
	Shift Dn		31	32 33	34	35	36
	Static			Load	EPT Ima	ige	
	Enter Timer Value	e 65535	Rst	EPT	Face	e	
Block Controls							
Block Send E	Block LoopBack		Block Re	ceive			
^	BLOCK 8	Repititions		^			
	BLOCK 16	Infinite					
	BLOCK 128	Stop					
	USR BLCK	Jtop					
~				~			
Rst			Rst				
Address 4 Length	8 Er	rors	Bytes	Transferred			
				~ ~	പം	10:33	PM

With the application loaded, select the USB FPGA board from the dropdown combo box and click on the "Open" button.





Transfer Controls	Open	Close		
Send Byte: 255 Byte Multiple Byte: 255	Address 2 Recei	ive Byte	^	
Multi Byte		Rst	<u> </u>	
Switches Switch Controls Rst Block Controls	LED Controls LED's Random Shift Left Shift Right Shift Up Shift Dn Static Enter Timer Value	1 7 13 19 25 31 65535 Rat	2 3 8 9 14 15 20 21 26 27 32 33 Load EP EPT	4         5         6           10         11         12           16         17         18           22         23         24           28         29         30           34         35         36           T Image Face         Face         10
Block Send Blo	ck LoopBack BLOCK 8	Block F Repititions	Receive	

Click on one of the LED buttons in the middle of the window. The corresponding LED on the DSO 100M board should light up.

To exercise the Single Byte Transfer EndTerm, click the "LoopBack" button in the Transfer Controls group. Type in several numbers separated by a space and less 256 into the Multiple Byte textbox. Then hit the Multi Byte button. The numbers appear in the Receive Byte textbox.



# **<u>14</u> <u>PC</u>** UnoProLyzer Code Base Description

At this point, the following has been accomplished:

- Description of the DSO 100M hardware
- Instructions on loading the drivers for the DSO 100M
- Instructions for loading the Quartus Prime and JTAG \*.dll
- Instructions on how to create a Quartus project
- Instructions for loading the Visual Studio Express
- Description of the FPGA EndTerm interface
- Description of the PC Active Host interface
- Instructions on how to create a C# Windows Form
- FPGA code Description DSO 100M

So, now a description of the PC application, UnoProLyzer, for the DSO 100M can be given. This code description is meant to be thorough, however, it cannot cover every detail of the code. If any questions arise that cannot be determined from code and comments. Please send an email for assistance on the EPT support email:

support@earthpeopletechnology.com

## 14.1 UnoProLyzer Application Overview





# 14.2 Software Flow Diagram

