Pico Computing
E-16 / E-17 / M-501 / M-503 Getting Started Guide
December 2011

Contents

1. Overview

Thank you for choosing Pico Computing FPGA products. This manual will help you setup and interface with the Pico FPGA cards on Windows systems. We'll describe the process in several steps:

1. System Requirements
2. How to install the software
3. Load bit files
4. How to run through an example program that uses the card
5. Modify and build the example
6. Building from ISE GUI
7. Hardware trouble shooting
8. picocommand command line utility

2. System Requirements

In order to operate a Pico Computing FPGA product, your system must meet the following minimum system requirements:

1. Workstation
    a. System must be able to handle full height, full length PCI card
    b. System must have at least one available x16 PCIe slot
    c. System's power supply must have available PCIe power (GPU style connector)
    d. Power supply that can deliver a maximum of 240W (6 Virtex-6 LX240Ts)
    e. System must have cooling fans to keep the FPGA(s) under 85C
    f. Windows 7 64-bit
    g. Recommended motherboard chipset in the Intel Tylersburg Series
2. Laptop
    a. System must have a ExpressCard 34 or 54 slot available
    b. Windows 7 64-bit

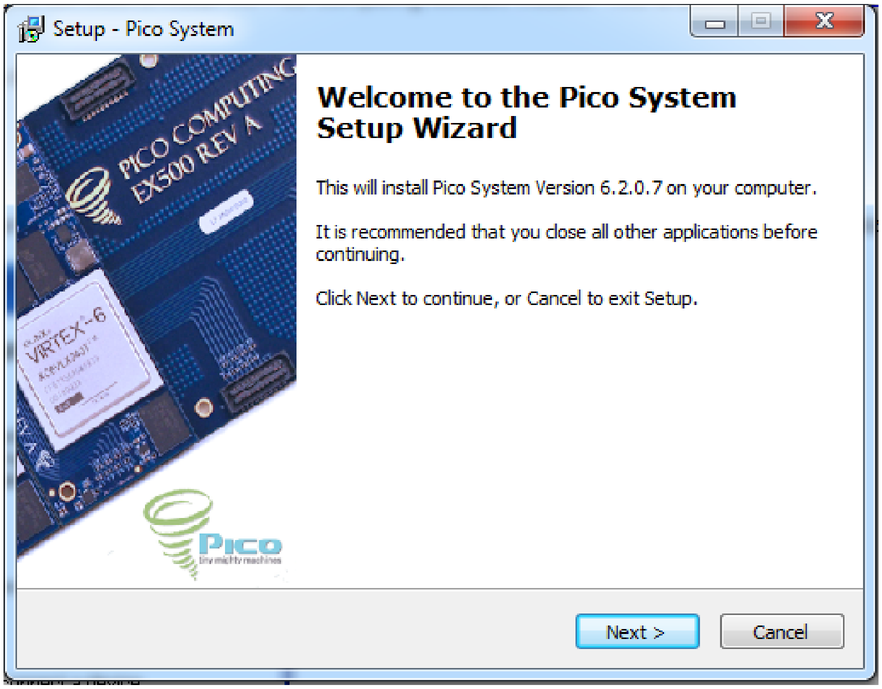
3. Installing the M-501 / M-503 / E-17 Pico Software

See the following section for Installing the E-16

This section describes the procedure for installing the M-501, M-503 and E-17 and associating the drivers with the device. After the software and hardware installation, you should take time to go through the PicoBus_Counter example to verify correct installation of the software and hardware, and to learn how to use the card.

Before installing the Card into a slot in your desktop or laptop, you will need to install the Pico Computing software installer. This is available from http://www.picocomputing.com/downloads/software.php, or on the software CD provided with the card. Version 6.2.0.0 and later are covered by this guide.

1. Run PicoInstaller_6.2.0.7.exe.



Welcome to the Pico System Setup Wizard

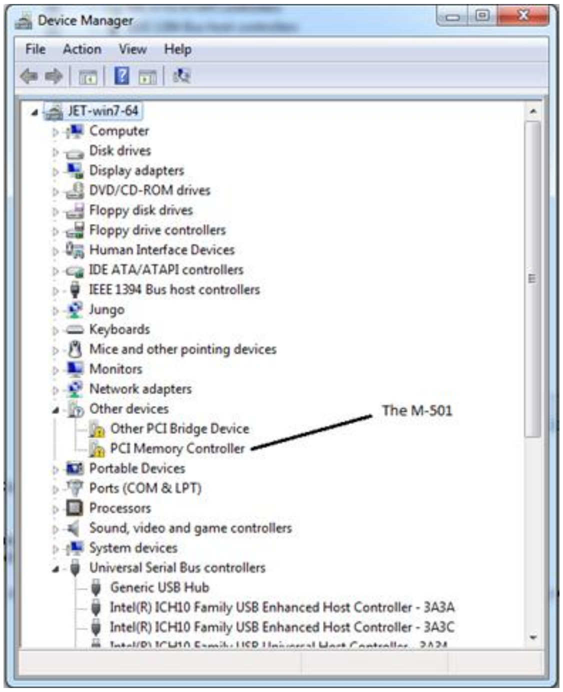
This will install Pico System Version 6.2.0.7 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

2. Click Next to get started
3. Set the directory to install the Pico system to. Click next.
4. Run through the rest of the installer dialog.
5. As automatic hardware installation has been disabled, the user must install the hardware. To do this, Window's driver installation wizard must be run. Desktops and servers must physically connect the card to the backplane or ExpressCard slot while system power is off, and start the system. Laptops may simply insert the card into an ExpressCard slot.
6. **PLEASE NOTE**: You must have administrator rights to perform this hardware installation.
7. Run Device Manager (in a command window, type **devmgmt.msc**).
8. Depending on if a previous version of the Pico software has been installed on the machine, Device Manager will display one of the following:

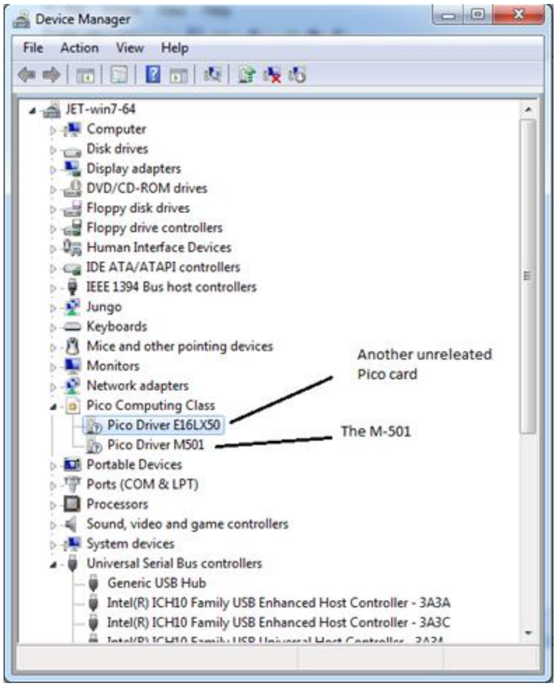
No software previously installed



JET-win7-64
- Computer
- Disk drives
- Display adapters
- DVD/CD-ROM drives
- Floppy disk drives
- Floppy drive controllers
- Human Interface Devices
- IDE ATA/ATAPI controllers
- IEEE 1394 Bus host controllers
- Jungo
- Keyboards
- Mice and other pointing devices
- Monitors
- Network adapters
- Other devices
  - Other PCI Bridge Device
  - PCI Memory Controller
- Portable Devices
- Ports (COM & LPT)
- Processors
- Sound, video and game controllers
- System devices
- Universal Serial Bus controllers
  - Generic USB Hub
  - Intel(R) ICH10 Family USB Enhanced Host Controller - 3A3A
  - Intel(R) ICH10 Family USB Enhanced Host Controller - 3A3C
  - Intel(R) ICH10 Family USB Universal Host Controller - 3A34

The M-501

Software previously installed



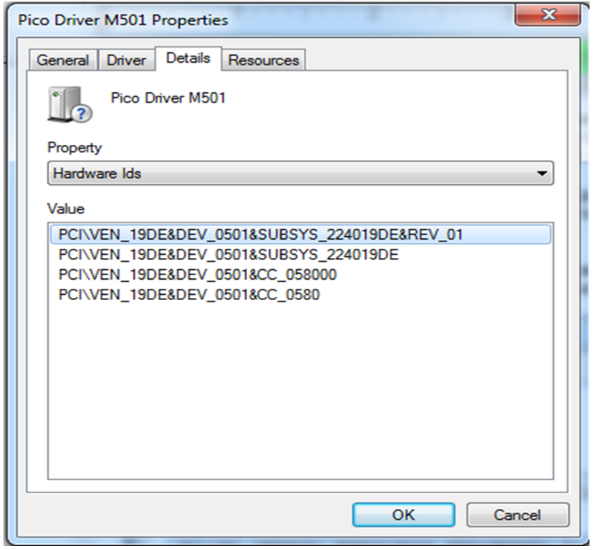
JET-win7-64
- Computer
- Disk drives
- Display adapters
- DVD/CD-ROM drives
- Floppy disk drives
- Floppy drive controllers
- Human Interface Devices
- IDE ATA/ATAPI controllers
- IEEE 1394 Bus host controllers
- Jungo
- Keyboards
- Mice and other pointing devices
- Monitors
- Network adapters
- Pico Computing Class
  - Pico Driver E16LX50
  - Pico Driver M501
- Portable Devices
- Ports (COM & LPT)
- Processors
- Sound, video and game controllers
- System devices
- Universal Serial Bus controllers
  - Generic USB Hub
  - Intel(R) ICH10 Family USB Enhanced Host Controller - 3A3A
  - Intel(R) ICH10 Family USB Enhanced Host Controller - 3A3C
  - Intel(R) ICH10 Family USB Universal Host Controller - 3A34

Another unreleated Pico card

The M-501

9. If you have previously installed software, you must uninstall all devices in the Pico Computing Class listing to remove the old version of the driver. To uninstall a device:
   a. Right click on the device. Select **Uninstall**
   b. Click the check box marked **"Delete the driver software for this device".** Click OK.
   c. The devices will disappear from the Device Manager's list. Right click any icon in Device Manager and Select **Scan for Hardware Changes.**
   d. The devices should reappear, but no driver task will be started for them (Device Manager will display them as though no Pico software was ever installed)
10. It is possible that the M501, M503, or E17 is not distinguishable from other devices that Device Manager labels as a "PCI Memory Controller”. If you are unsure which device is your Pico card you may do the following:
    a. Right click the device in question. Select **Properties**
    b. Select the **Details** tab
    c. In the drop down menu labeled *Property* select **Hardware IDs.**
    d. The cards will display a set of properties matching one of the :

M501 Properties



Pico Driver M501

Property: Hardware Ids

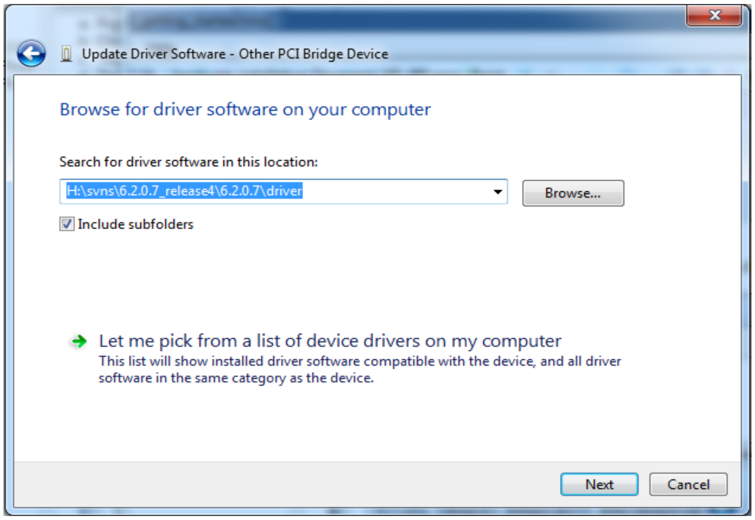
Value

```
PCI\VEN_19DE&DEV_0501&SUBSYS_224019DE&REV_01
PCI\VEN_19DE&DEV_0501&SUBSYS_224019DE
PCI\VEN_19DE&DEV_0501&CC_058000
PCI\VEN_19DE&DEV_0501&CC_0580
```

Device Listing for other Pico Cards

E16LX50:	PCI\VEN_10b5&DEV_9056
E17FX70T:	PCI\VEN_19DE&DEV_0E17&SUBSYS_00004658&REV_01
E17SX50T:	PCI\VEN_19DE&DEV_0E17&SUBSYS_00005358&REV_01
M501:	PCI\VEN_19DE&DEV_0501&SUBSYS_224010DE&REV_01
M503:	PCI\VEN_19DE&DEV_0503&SUBSYS_224019DE&REV_01

11. Right click on the Pico card in the list. Select Update Driver Software, then select Browse My Computer for Software. The wizard should display the following prompt



Browse for driver software on your computer

Search for driver software in this location:

H:\svns\6.2.0.7_release4\6.2.0.7\driver

☑ Include subfolders

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.

12. In the search box, specify the driver directory. This is the directory the Pico software was installed under.

13. Windows Security may issue a "are you sure?" prompt. Click **Install**

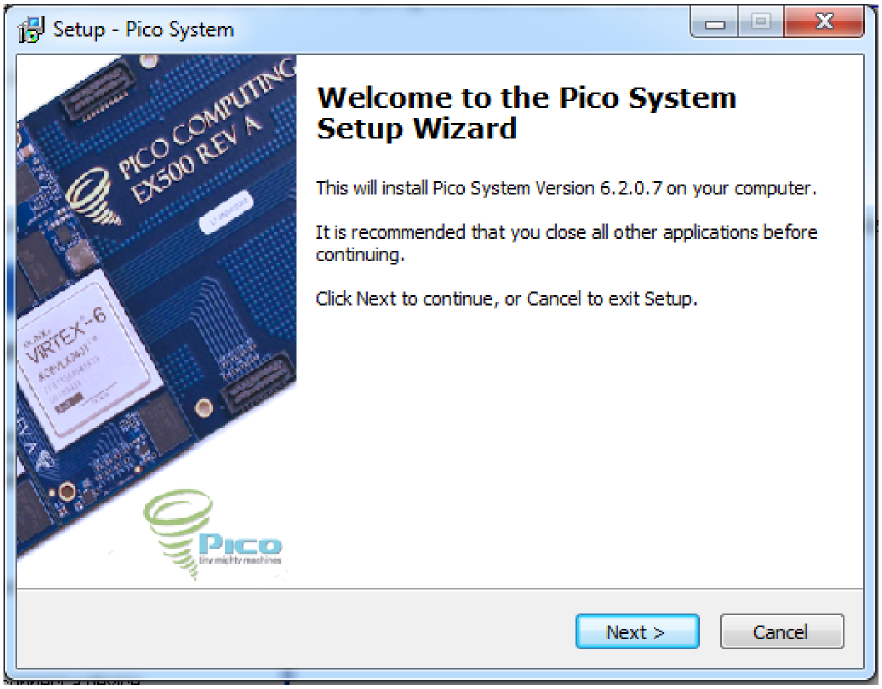
Congratulations, you have installed your Pico card, reboot your system and it is ready to use.

4. Installing the E-16 Pico Software

This section describes the procedure for installing the E-16 card and associating the drivers with the device. After the software and hardware installation, you should take time to go through the PicoBus_Counter example to verify correct installation of the software and hardware, and to learn how to use the card.

Before installing the Card into a slot in your laptop, you will need to install the Pico Computing software installer. This is available from http://www.picocomputing.com/downloads/software.php, or on the software CD provided with the card. Version 6.2.0.0 and later are covered by this guide.

1. Run PicoInstaller_6.2.0.7.exe. The screen should look like this:



Setup - Pico System

Welcome to the Pico System Setup Wizard

This will install Pico System Version 6.2.0.7 on your computer.

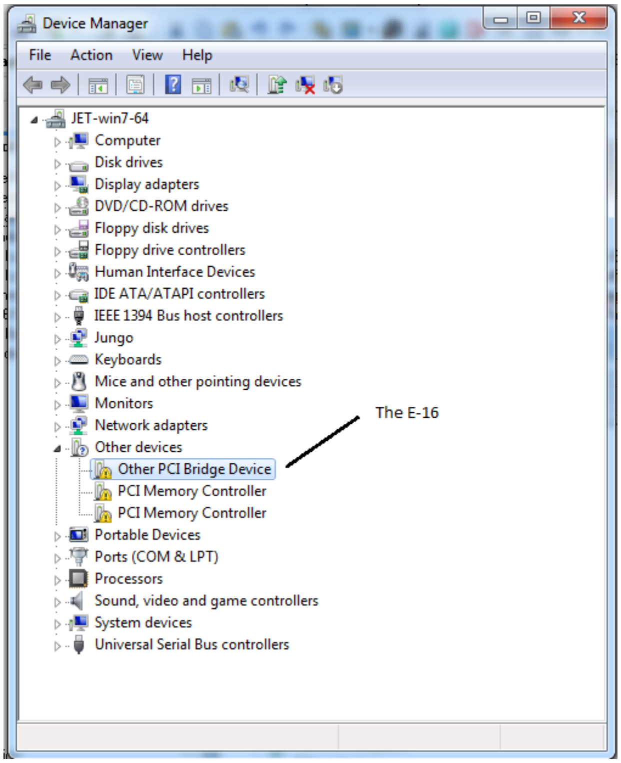
It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

2. Click Next to continue
3. Set the directory to install the Pico system to. Click next.
4. Run through the rest of the installer dialog.
5. Automatic hardware installation is disabled, the user must install the hardware. To do this, Window's driver installation wizard must be run. Desktops and servers must physically connect the card to the backplane or ExpressCard slot while system power is off, and start the system. Laptops may simply insert the card into an ExpressCard slot.
6. **PLEASE NOTE**: You must have administrator rights to perform this hardware installation.
7. Run Device Manager (in a command window, type **devmgmt.msc**).

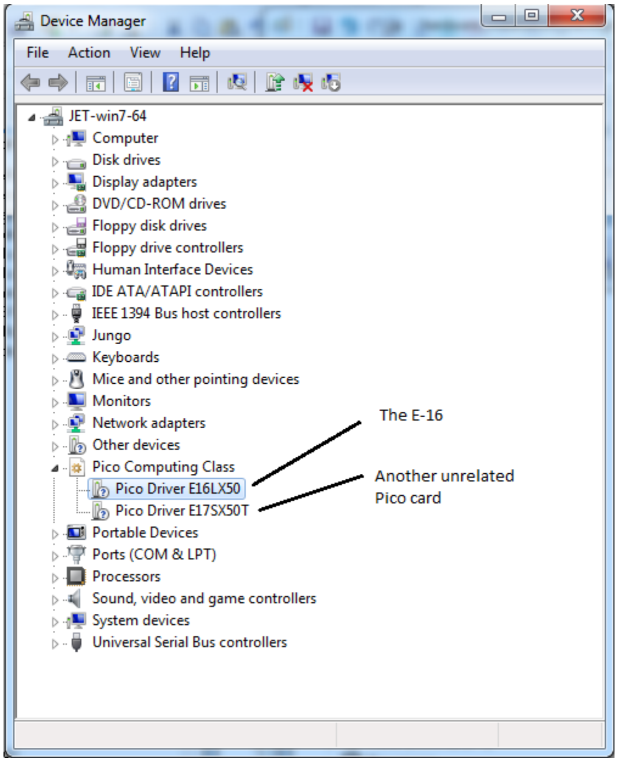
8. Depending on if a previous version of the Pico software has been installed on the machine, Device Manager will display one of the following depending on if this is a fresh installation or if you have had a previous installation:

No software previously installed



The E-16

Software previously installed



The E-16

Another unrelated Pico card

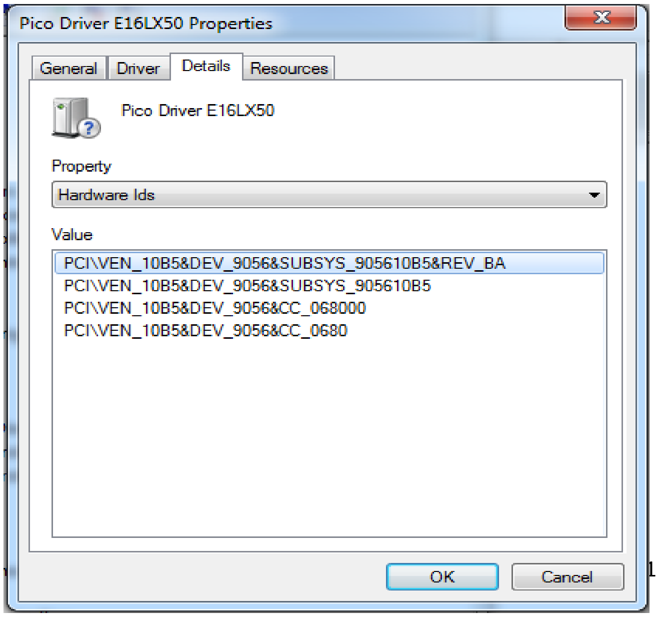
9. It is possible that the Pico card is not distinguishable from other devices that Device Manager labels the E-16 as "Other PCI Bridge Device" If you are unsure which device is your Pico card you may do the following:

a. Right click the device in question. Select **Properties**

b. Select the **Details** tab

c. In the drop down menu labeled *Property* select **Hardware IDs.**

d. E-16s will display the following properties:

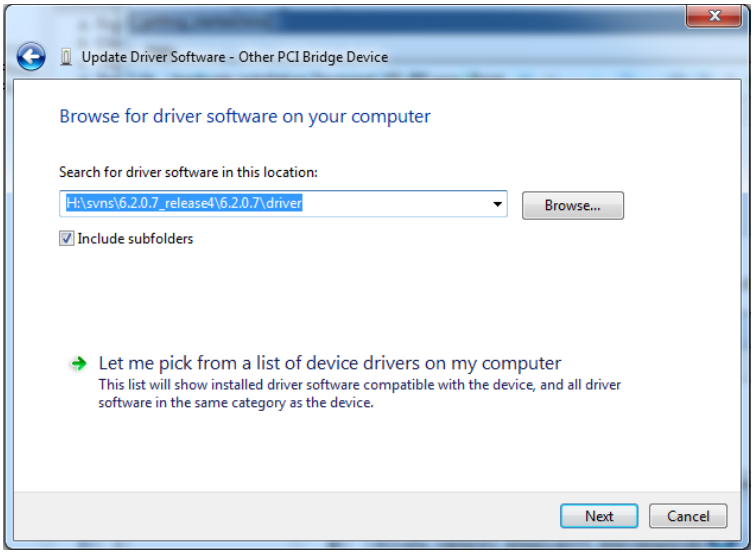


Property: Hardware Ids

Value:

```
PCI\VEN_10B5&DEV_9056&SUBSYS_905610B5&REV_BA
PCI\VEN_10B5&DEV_9056&SUBSYS_905610B5
PCI\VEN_10B5&DEV_9056&CC_068000
PCI\VEN_10B5&DEV_9056&CC_0680
```

9. Right click on the Pico card on the list. Select Update Driver Software, then select Browse My Computer for Software. The wizard should display the following prompt



Update Driver Software - Other PCI Bridge Device

Browse for driver software on your computer

Search for driver software in this location:

H:\svns\6.2.0.7_release4\6.2.0.7\driver

☑ Include subfolders

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.

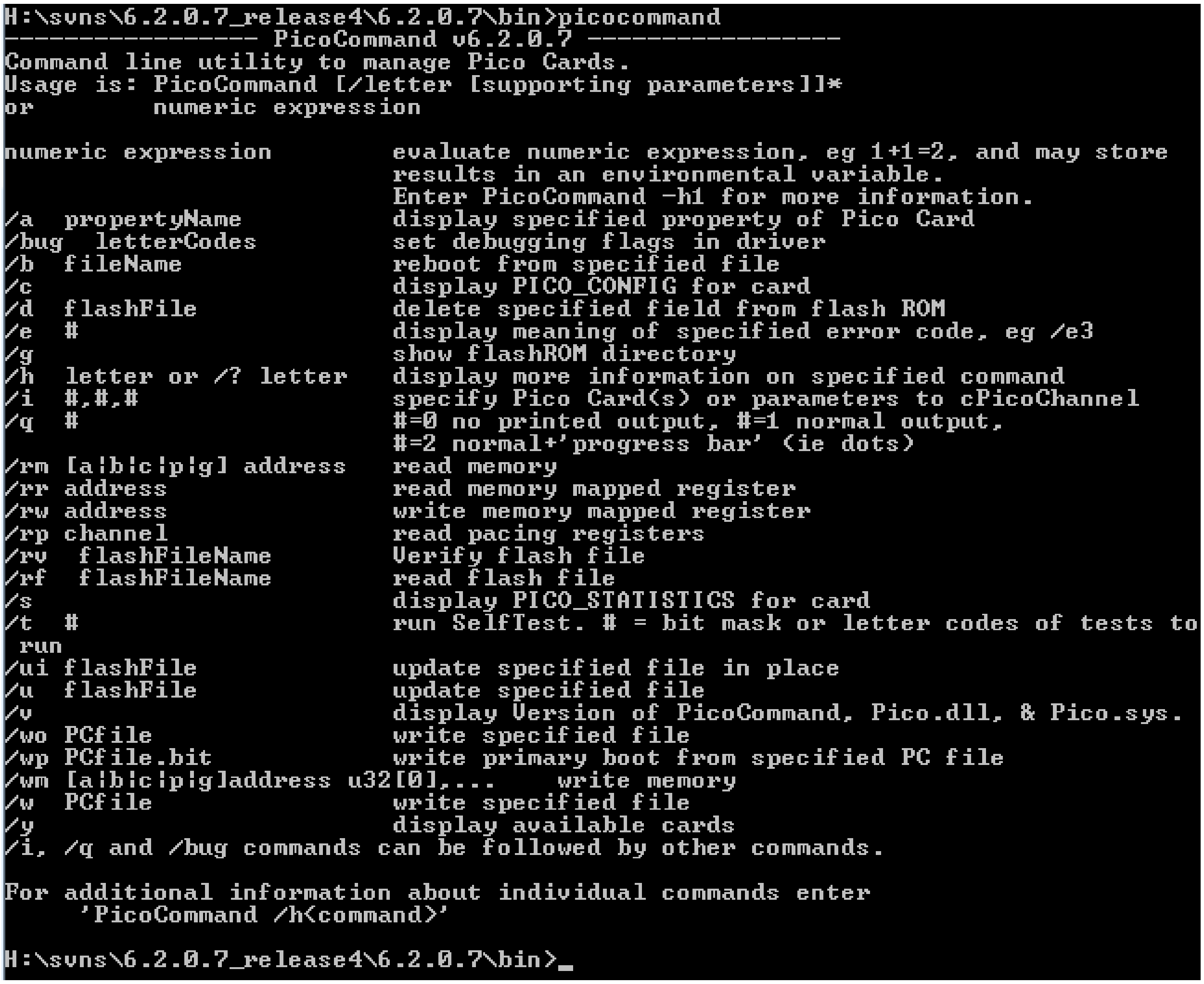
10. In the search box, specify the driver directory. This is the directory the Pico software was installed under.
11. Windows Security may issue a "are you sure?" prompt. Click **Install**

Congratulations, you have installed your Pico card, reboot your system and it is ready to use.

5. Loading bit files

You can boot images from the PCs hard drive using the command line utility picocommand.exe**,** or you may write you own program using the PicoChannel API (see PicoChannel doc).

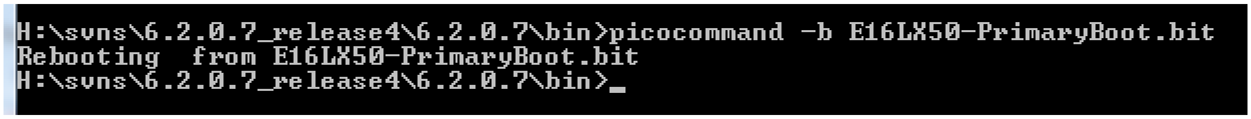
Open a command prompt and navigate to the bin directory under the %picobase% directory (default: c:\Pico\{version #}\bin)
Type “picocommand” and the screen should look like this:



```
H:\svns\6.2.0.7_release4\6.2.0.7\bin>picocommand
----------------- PicoCommand v6.2.0.7 -----------------
Command line utility to manage Pico Cards.
Usage is: PicoCommand [/letter [supporting parameters]]*
or        numeric expression

numeric expression        evaluate numeric expression, eg 1+1=2, and may store
                          results in an environmental variable.
                          Enter PicoCommand -h1 for more information.
/a  propertyName          display specified property of Pico Card
/bug  letterCodes         set debugging flags in driver
/b  fileName              reboot from specified file
/c                        display PICO_CONFIG for card
/d  flashFile             delete specified field from flash ROM
/e  #                     display meaning of specified error code, eg /e3
/g                        show flashROM directory
/h  letter or /? letter   display more information on specified command
/i  #,#,#                 specify Pico Card(s) or parameters to cPicoChannel
/q  #                     #=0 no printed output, #=1 normal output,
                          #=2 normal+'progress bar' (ie dots)
/rm [a|b|c|p|g] address   read memory
/rr address               read memory mapped register
/rw address               write memory mapped register
/rp channel               read pacing registers
/rv  flashFileName        Verify flash file
/rf  flashFileName        read flash file
/s                        display PICO_STATISTICS for card
/t  #                     run SelfTest. # = bit mask or letter codes of tests to
 run
/ui flashFile             update specified file in place
/u  flashFile             update specified file
/v                        display Version of PicoCommand, Pico.dll, & Pico.sys.
/wo PCfile                write specified file
/wp PCfile.bit            write primary boot from specified PC file
/wm [a|b|c|p|g]address u32[0],....    write memory
/w  PCfile                write specified file
/y                        display available cards
/i, /q and /bug commands can be followed by other commands.

For additional information about individual commands enter
     'PicoCommand /h<command>'

H:\svns\6.2.0.7_release4\6.2.0.7\bin>_
```

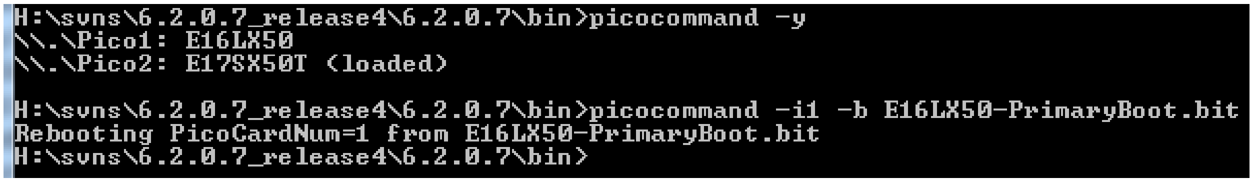
This is the help menu for picocommand

To load the primary boot image, type "**picocommand -b <name of primary boot image>**



```
H:\svns\6.2.0.7_release4\6.2.0.7\bin>picocommand -b E16LX50-PrimaryBoot.bit
Rebooting  from E16LX50-PrimaryBoot.bit
H:\svns\6.2.0.7_release4\6.2.0.7\bin>_
```

It's a good habit to refer to a specific card number when loading a boot image. The command to load an image onto a specific card is "**picocommand –i<number card> -b <name of primary boot image>** You can find out the card numbers by typing "**picocommand –y".**



```
H:\svns\6.2.0.7_release4\6.2.0.7\bin>picocommand -y
\\.\Pico1: E16LX50
\\.\Pico2: E17SX50T (loaded)

H:\svns\6.2.0.7_release4\6.2.0.7\bin>picocommand -i1 -b E16LX50-PrimaryBoot.bit
Rebooting PicoCardNum=1 from E16LX50-PrimaryBoot.bit
H:\svns\6.2.0.7_release4\6.2.0.7\bin>
```

In the screen shot above, this system has an E-16LX50 and E-17SX50T; and the Pico driver has arbitrarily assigned the E-16LX50 to card #1 and the E-17SX50T #2 for this system boot.
You can verify that the image is properly loaded by running the selftest, or just inspecting the pacing registers. Enter:

picocommand /rr 0x10000010

This will read the channel 1 pacing register on card #1 and return the value 0x980FFFFF.

6. Running the PicoBus Counter example

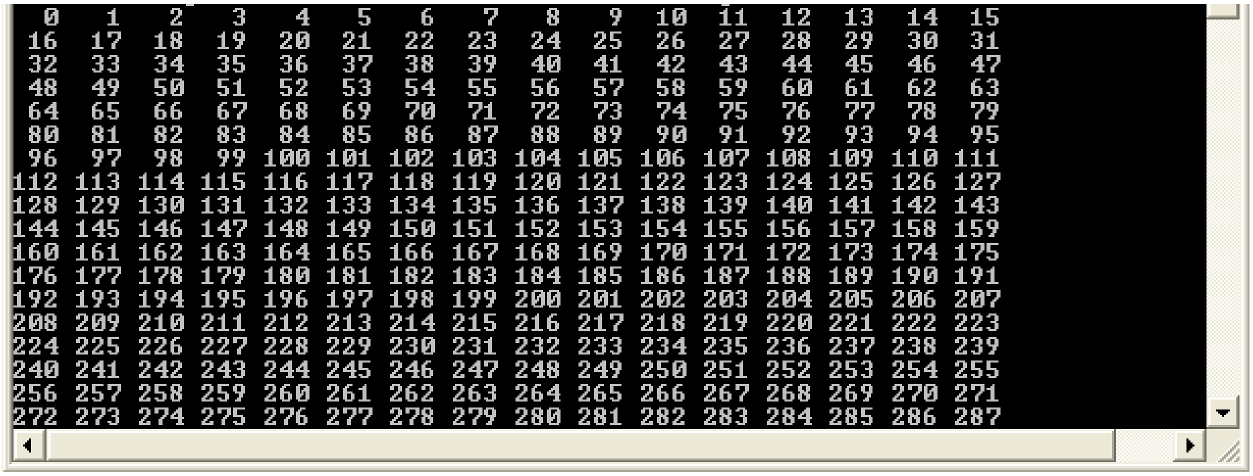
The PicoBus Counter example provides a quick and easy way to verify correct operation of your Pico card, and also demonstrates the basic methods of communication between the card and a host computer. This example is provided as a pre-built executable and corresponding FPGA image. You can also use Microsoft Visual Studio to modify and build your own application, based on the PicoBus sample.

NOTE: Visual Studio 2008 or better, or Visual Studio Express 2008 or better, is required to build all Pico software samples.

To run the PicoBus Counter example:

Open a Windows Command prompt.
Navigate to c:\Pico\ {version #}\Samples\PicoBus_counter\software
Type "PicoBus_counter.exe" and press enter. This will run the PicoBus_counter program.

When it runs, the PicoBus_counter program will first load the FPGA image onto the Pico card via the ExpressCard (PCI Express) interface, and then run through the demo. After the demo has successfully completed, the console window display should look like this:



```
  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
 16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31
 32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47
 48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63
 64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79
 80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95
 96  97  98  99 100 101 102 103 104 105 106 107 108 109 110 111
112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127
128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143
144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159
160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175
176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191
192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207
208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223
224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239
240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255
256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271
272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287
```

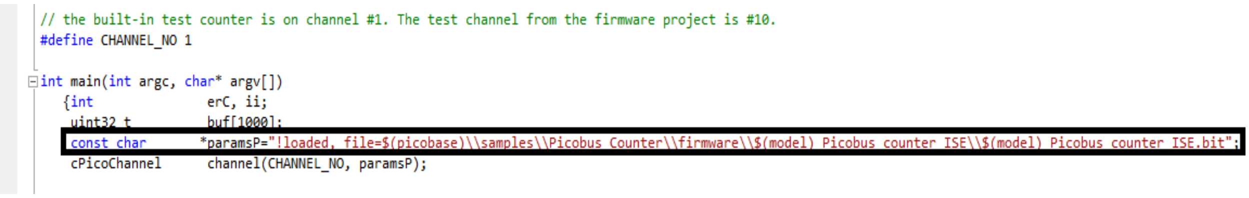
What you are seeing in this simple example are incrementing integers being produced by the FPGA firmware, and transmitted across the ExpressCard connection via the PicoBus hardware/software interface.

7. Modifying and Building the PicoBus Counter Example

The source for the PicoBus counter software is in the following directory

c:\pico\{version #}\Samples\PicoBus_counter\software

In this folder you will also find a Visual Studio project. If you examine this project, you will find that the file PicoBus_Counter.cpp makes reference to a file called <model number>_PicoBus_counter_ISE.bit, located in c\pico\{version #}\Samples\PicoBus_counter\firmware. This is the bit file that is loaded into the FPGA when this example is executed. To change the software application to use another bit image, change line 20:



```
// the built-in test counter is on channel #1. The test channel from the firmware project is #10.
#define CHANNEL_NO 1

int main(int argc, char* argv[])
  {int                erC, ii;
   uint32_t           buf[1000];
   const char        *paramsP="!loaded, file=$(picobase)\\samples\\Picobus_Counter\\firmware\\$(model)_Picobus_counter_ISE\\$(model)_Picobus_counter_ISE.bit";
   cPicoChannel       channel(CHANNEL_NO, paramsP);
```

8. Building from ISE GUI

For Pico products, firmware bit images are typically created using the Xilinx ISE toolchain. This page describes rebuilding the Picobus_counter firmware.

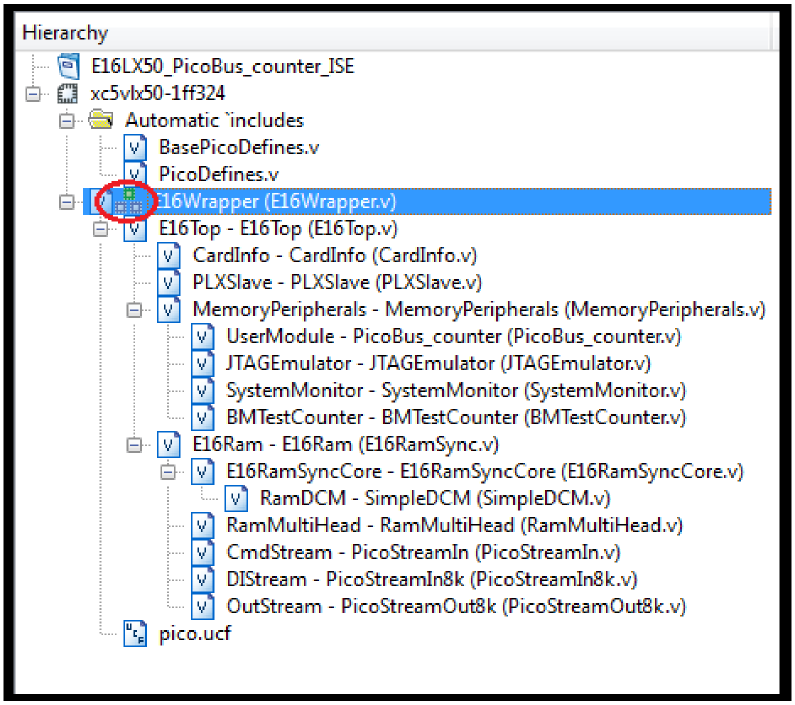
NOTE: A version of **ISE Design Suite 13** is required to build bit images. Xilinx's free development platform, ISE Webpack, is sufficient for this purpose.

Building from ISE GUI

For example purposes we will build on the E-16LX50 card.

1. Open a command prompt and navigate to c:\pico\samples\picobus_counter\firmware\e\E16LX50_Picobus_counter_ISE. The contents of this directory are:
   - E16LX50_Picobus_counter_ISE.bit:
     - This is the bit image generated from a previous run of the ISE tool chain.
   - E-16LX50_Picobus_counter_ISE.tcl:
     - A. script file, automatically generated from within the ISE GUI, used in the batch build process.
   - E16LX50_Picobus_counter_ISE.xise:
     - The ISE project file for Picobus_counter. Very similar to a Visual Studio project.
   - PicoDefines.v:
     - All Pico E-16LX50 firmware is built from common source, located in e:\pico\firmware\picoe16. PicoDefines.v contains preprocessor definitions that differentiate the generated bit file from other possible bit files, i.e. cause the toolchain to generate a bit file containing a Picobus_counter module, targeting the <%WHICH_CARD%>'s FPGA.

2. If you have installed the Xilinx Design Suite, 'running' the .xise project will start the ISE GUI with this project. After loading finishes, the GUI display will include a sub-window with 4 tabs (**Start**, **Design**, **Files**, **Libraries**). Select **Design.** Two windows will appear within the tab; *Hierarchy* and *Processes***.** The *Hierarchy* window will show the following:



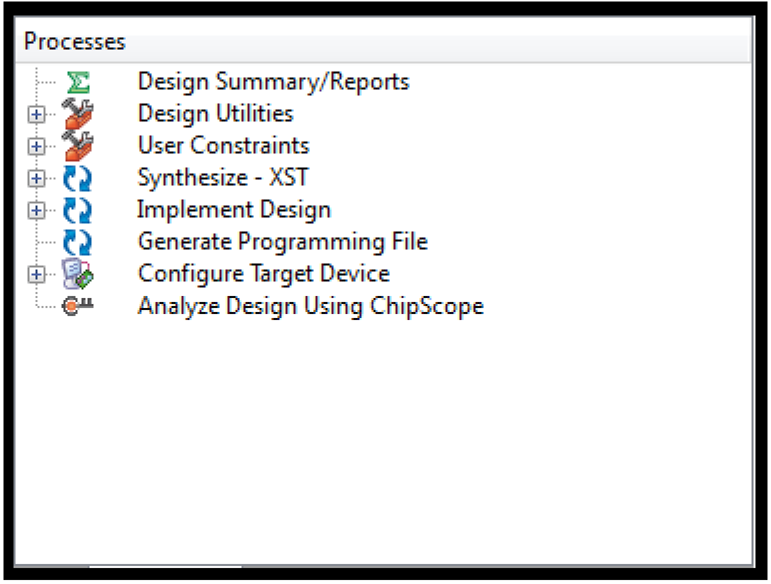
Hierarchy

- E16LX50_PicoBus_counter_ISE
- xc5vlx50-1ff324
  - Automatic `includes
    - BasePicoDefines.v
    - PicoDefines.v
  - E16Wrapper (E16Wrapper.v)
    - E16Top - E16Top (E16Top.v)
      - CardInfo - CardInfo (CardInfo.v)
      - PLXSlave - PLXSlave (PLXSlave.v)
      - MemoryPeripherals - MemoryPeripherals (MemoryPeripherals.v)
        - UserModule - PicoBus_counter (PicoBus_counter.v)
        - JTAGEmulator - JTAGEmulator (JTAGEmulator.v)
        - SystemMonitor - SystemMonitor (SystemMonitor.v)
        - BMTestCounter - BMTestCounter (BMTestCounter.v)
      - E16Ram - E16Ram (E16RamSync.v)
        - E16RamSyncCore - E16RamSyncCore (E16RamSyncCore.v)
          - RamDCM - SimpleDCM (SimpleDCM.v)
        - RamMultiHead - RamMultiHead (RamMultiHead.v)
        - CmdStream - PicoStreamIn (PicoStreamIn.v)
        - DIStream - PicoStreamIn8k (PicoStreamIn8k.v)
        - OutStream - PicoStreamOut8k (PicoStreamOut8k.v)
    - pico.ucf

3. The *Hierarchy* window describes a (gasp!) module hierarchy. In the pictured case, the E16Wrapper module is *Top Level Module* (the module indicated by the circled icon), which provides a direct interface to the hardwired ports on the FPGA (Pico.ucf maps port names from the Top Level Module to actual ball specifications on the FPGA pad). The Top Level Module also serves as a wrapper for modules implementing the on-chip memory interface, the DMA interface to the dedicated PCIe chip, and the end-user sandbox logic. Users wishing to modify this sample to use their own code instead of the Picobus_counter module should replace the sandbox module PicoBus_counter with their own, and modify PicoDefines.v to reflect the new module name.

4. Other Projects have a similar hierarchy; typically the top level model has Top in the name.

5. Select the **Top Level Module file** in the hierarchy**.** The *Processes* Window should display the following:



Processes

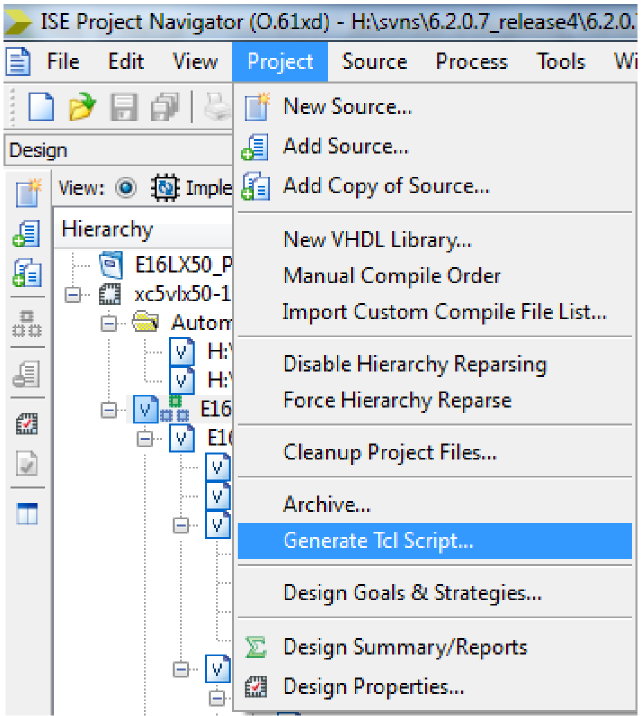
- Design Summary/Reports
- Design Utilities
- User Constraints
- Synthesize - XST
- Implement Design
- Generate Programming File
- Configure Target Device
- Analyze Design Using ChipScope

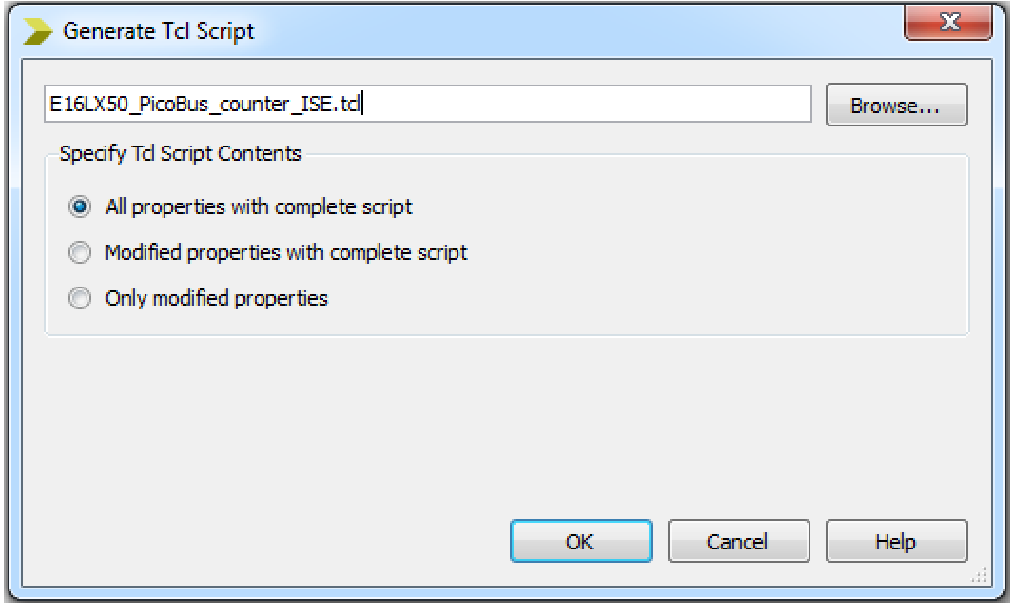
6. Double Clicking on **Generate Programming File** will initiate the toolchain and generate a bitfile with name matching the Top Level Module, *E16Wrapper.bit*

Building from Batch Files

As of Version 6.2.0.7, building from batch files is possible but not useful during firmware development. Given an .xise project, ISE can generate a .tcl script. which can be used by the xtclsh utility (xilinx's tcl interpretor) to orchestrate a firmware build.
NOTE: This method was developed for use in our installer builder, and is known to be very finicky. Pico Computing ***does not offer support of this build method.***

1. In ISE, click **Project**, then **Generate Tcl Script.** A dialog will appear. Leave all settings at defaults and click **OK.** This will re-generate the file "E16LX50_PicoBus_counter_ISE.tcl".




Generate Tcl Script
Browse...
Specify Tcl Script Contents
○ Modified properties with complete script
○ Only modified properties
OK
Cancel
Help

E16LX50_PicoBus_counter_ISE.tcl

● All properties with complete script

2. Save your work and close ISE.
3. From Command Line, navigate to %picobase%\firmware and type **build_model ..\samples\picobus_counter\firmware\E16LX50_Picobus_Counter_ISE**. This will kick off a firmware build of the E16 project.
4. The Build_model batch file renames the output bit file to *E16LX50_Picobus_Counter_ISE.bit.*

9. PICO COMMAND

Pico Command

Command Line utility to manage Pico card

Usage is: picocommand [/letter [supporting parameters]]* or numeric expression

Enter "picocommand -h1" for more information

Numeric	Expression	Evaluate numeric expression, eg 1+1=2, and may store results in anenvironmental variable.
/ a	property name	Display specified property of the Pico card
/ bug	letter code	Set debugging flags in the driver
/ b	file name	Reboot from specified file
/ c		Display Pico-CONFIG for card
/ d	flash file	Delete specified field from FLASH ROM
/ e	#	Display meaning of specified error code
/ g		Show FLASH ROM Directory
/ h	letter or /? Letter	Display more information on specified command
/ i	#,#,#	Specify Pico Card(s) or parameters to cPicoChannel1
/ q	#	#=0 no printed output, #=1 normal output, #=2 normal & progress bar
/ rm	[a|b|c|p|g] address	Read memory
/ rr	address	Read memory maped register
/ rw	address	Write memory mapped register
/ rp	channel	Read pacing registers

/ fv	flash file name	Verify FLASH file
/ rf	flash file name	Read FLASH file
/ s		Display PICO_STATISTICS for card
/ t	#	Run SelfTest. #= bit mask or letter codes of tests to run
/ ui	flash file	Update specified file in place
/ u	flash file	Update specified file
/ v		Display Version of PicoCommand, Pico.dll, & Pico.sys
/ wo	PC file	Write specified file
/ wp	PC file.bit	Write primary boot frm specified PC file
/ wm	[a|b|c|p|g]address u32[0]…	Write memory mapped register
/ w	PC file	Write specified file
/ y		Display available cards
/l, /q and /but commands can be followed by other commands		
For additional information about individual commancds enter: "picocommand /h<command>"		