

Pico Computing Inc.



PicoUtil and PicoCommand User's Manual

Version 5.0.0.3. Apr 16th, 2010.

Pico Computing, Inc.
150 Nickerson, Suite 311
Seattle, WA, 98109-1634
(206) 283-2178
www.picocomputing.com

1 Overview

Thank you for choosing Pico Computing FPGA products. This manual will help you getting started with the Pico E-17. The E-17 is modular ExpressCard format platform that can be used into two fundamental ways, either plug-in or stand-alone. This document describes how to use E-17 as a plug-in card. Standalone modes of use, using the Xilinx ISE Design Suite software for development, are described in the E-17 Base System Builder documentation.

Plug-In Mode

In plug-in mode, the E-17 is physically attached to a host computer using the PCI Express interface for host system communication. FPGA bitmaps may be downloaded via either the PCI Express interface, or via a JTAG programming cable. The card may be physically inserted into an ExpressCard slot (for example in a notebook PC), or it may be installed into a desktop workstation or server using a PCI Express to ExpressCard adapter.

In some cases, for example when operating temperature is a factor, the E-17 may be attached to a notebook PC using an ExpressCard extender. In cluster situations, multiple E-17 cards may be installed into a backplane, which is then plugged into an 8X or 16X PCI Express socket.

Other manuals in this help library can be located at [GuideToDocumentation.pdf](#)

NOTE: This link will access the pdf from the PicoComputing.com Website .

2 Installing E-16

This section describes the procedure for installing the E-16 card and enabling the drivers. After the E-16 has been installed, 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.

Installing the Pico Computing Software

Before installing the Card into a slot in your laptop or server, you will need to install the Pico Computing software installer. This installer will ask you for the type of Card being installed, and will copy relevant files to your host system. The installer will also copy certain driver-related files to your system.

Note: you should not plug the E-17 card into the slot of the host computer before running the installer.

Installing and Enabling the E-16 Card

1. During installation of software, you will be asked to put your E-16 card into the slot of your host computer. When you do this, Windows will recognize that a new device has been attached and will display the Found New Hardware Wizard dialog:



2. Select "No, not at this time" and click Next.
3. On the next Wizard screen, select "Install from a list or specific location", and Click Next.
4. Select "Search for the best driver in these locations", and select "Include the location in the search:"
Navigate to C:\Pico\Driver, or to the directory in which you have installed the Pico software. (For brevity, this document assumes you have installed into C:\Pico.)

[SCREEN IMAGE]

5. Click Next.

When your E-16 hardware is installed, the Wizard screen should look like this:



Congratulations, you have installed your E-16 card, and it is ready to use.

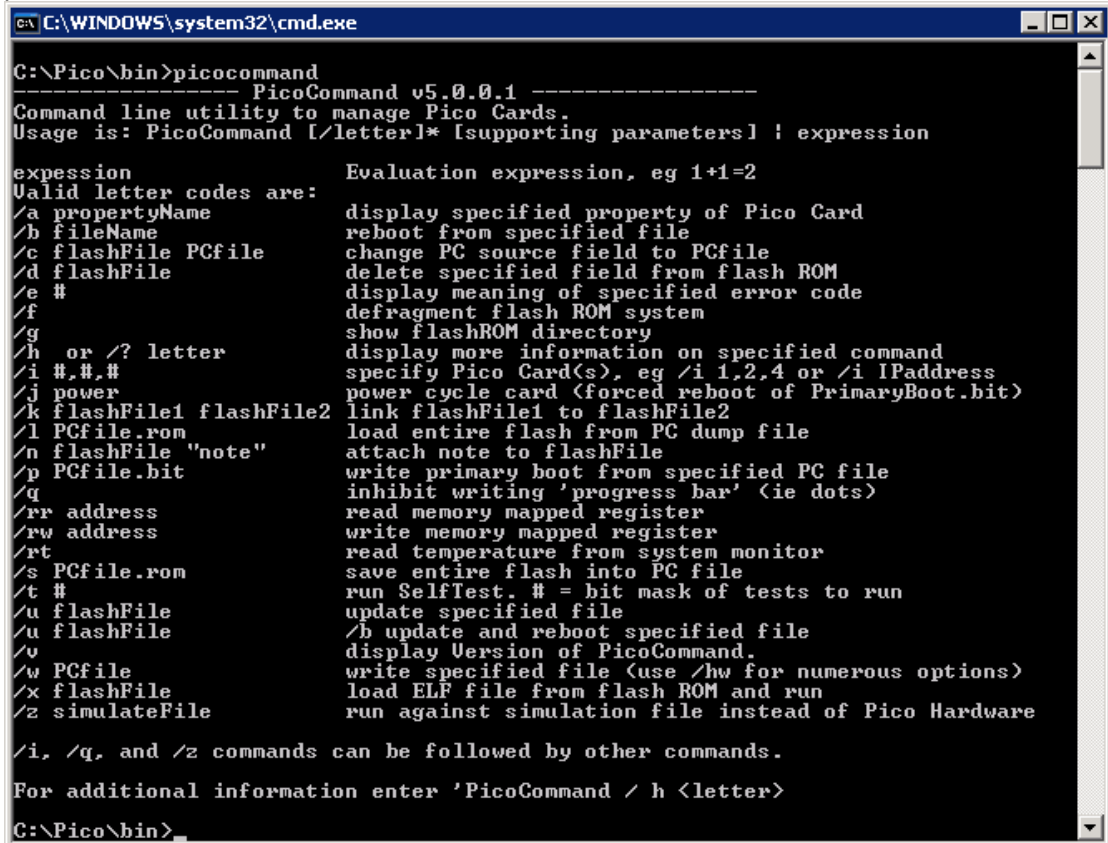
If you were unable to install the E-16 card using the above procedure, please refer to the Hardware Troubleshooting section of this document.

3 Loading Firmware onto the E17

Once power is supplied to the E-17, it will automatically load the PrimaryBoot.bit from flash. You can download and boot other images from the flash using the command line utility [picocommand.exe](#), the GUI [PicoUtil.exe](#), or you may write you own program using the PicoChannel.

3.1 Loading Firmware using Picocommand

1. Open a command prompt and navigate to C:\Pico\bin
2. Type picocommand and the screen should look like this:



```

C:\WINDOWS\system32\cmd.exe
C:\Pico\bin>picocommand
----- PicoCommand v5.0.0.1 -----
Command line utility to manage Pico Cards.
Usage is: PicoCommand [/letter]* [supporting parameters] : expression

expression          Evaluation expression, eg 1+1=2
Valid letter codes are:
/a propertyName     display specified property of Pico Card
/b fileName         reboot from specified file
/c flashFile PCfile change PC source field to PCfile
/d flashFile        delete specified field from flash ROM
/e #                display meaning of specified error code
/f                 defragment flash ROM system
/g                 show flashROM directory
/h or /? letter    display more information on specified command
/i #,#,#           specify Pico Card(s), eg /i 1,2,4 or /i IPaddress
/j power           power cycle card (forced reboot of PrimaryBoot.bit)
/k flashFile1 flashFile2 link flashFile1 to flashFile2
/l PCfile.rom      load entire flash from PC dump file
/n flashFile "note" attach note to flashFile
/p PCfile.bit      write primary boot from specified PC file
/q                 inhibit writing 'progress bar' (ie dots)
/rr address        read memory mapped register
/rw address        write memory mapped register
/rt               read temperature from system monitor
/s PCfile.rom     save entire flash into PC file
/t #              run SelfTest. # = bit mask of tests to run
/u flashFile      update specified file
/v flashFile      /b update and reboot specified file
/w PCfile         display Version of PicoCommand.
/x flashFile      write specified file (use /hw for numerous options)
/z simulateFile   load ELF file from flash ROM and run
                  run against simulation file instead of Pico Hardware

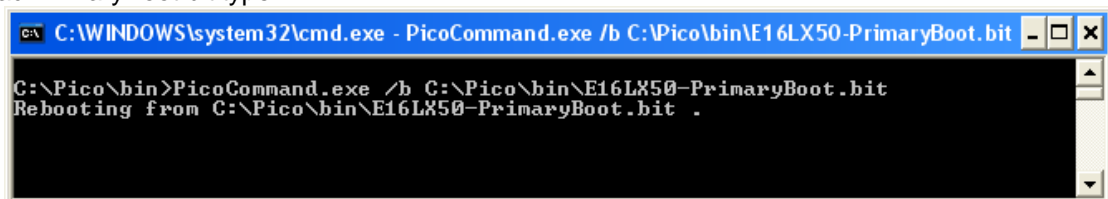
/i, /q, and /z commands can be followed by other commands.

For additional information enter 'PicoCommand / h <letter>'

C:\Pico\bin>

```

- a. This is the help menu for picocommand
3. To load PrimaryBoot.bit type:



```

C:\WINDOWS\system32\cmd.exe - PicoCommand.exe /b C:\Pico\bin\E16LX50-PrimaryBoot.bit
C:\Pico\bin>PicoCommand.exe /b C:\Pico\bin\E16LX50-PrimaryBoot.bit
Rebooting from C:\Pico\bin\E16LX50-PrimaryBoot.bit .

```

4. You can verify that the file is properly loaded by running the selftest of just inspecting the pacing registers. Enter:

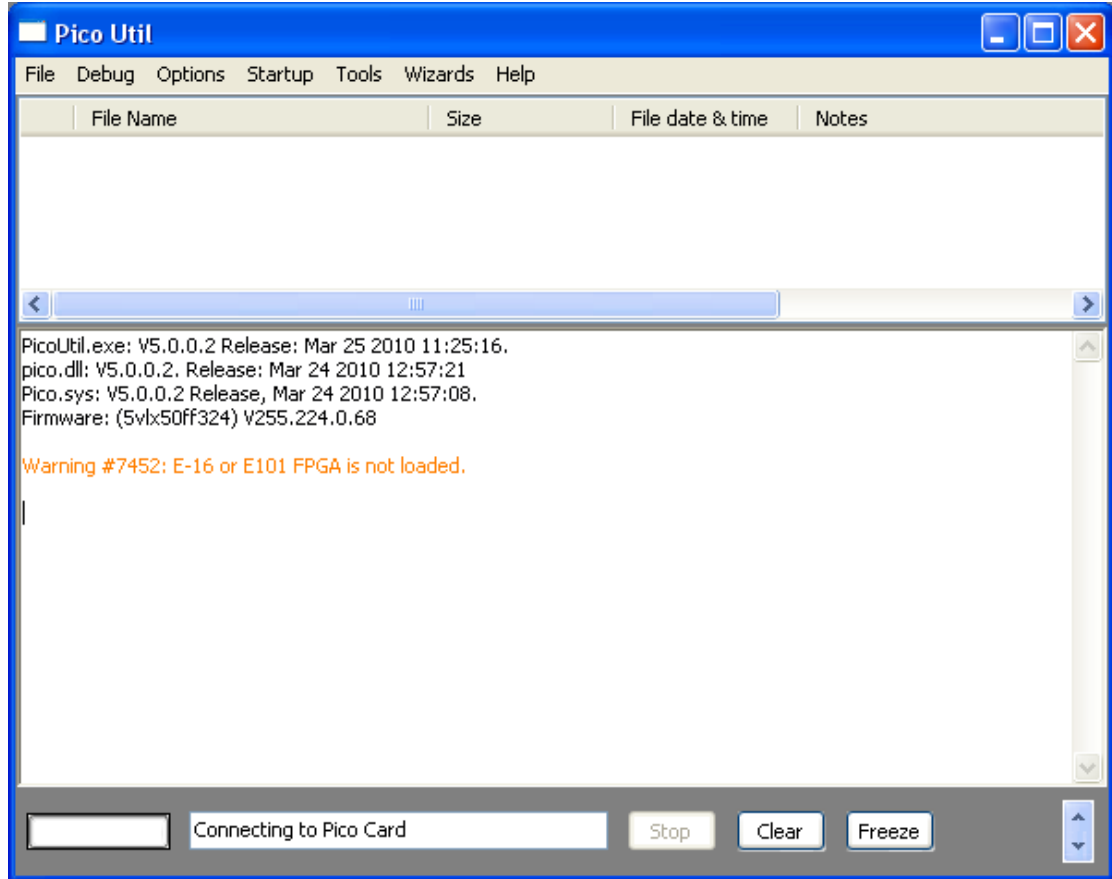
You can verify that the file is properly loaded by running the selftest of just inspecting the pacing registers. Enter:

picocommand /rr 0x10000010

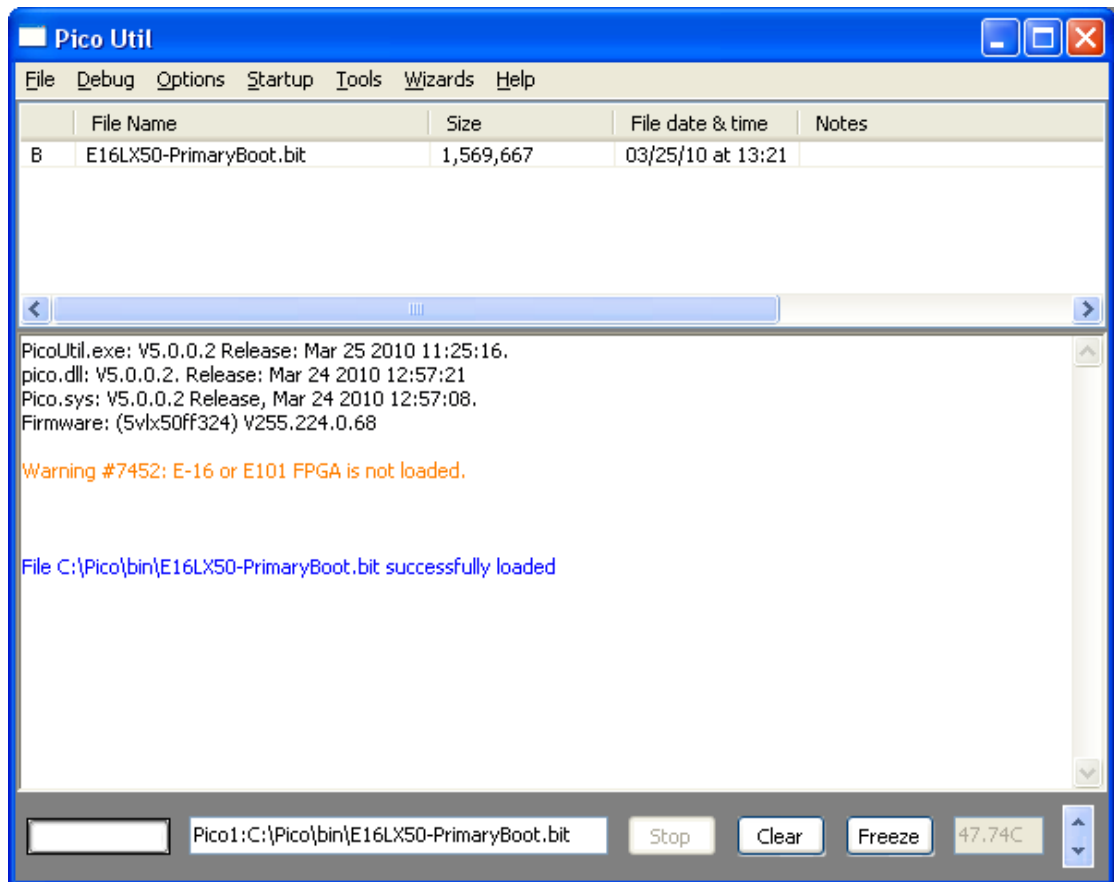
This will read the pacing register and return the value 0x980FFFFFF.

3.2 Loading Firmware using PicoUtil

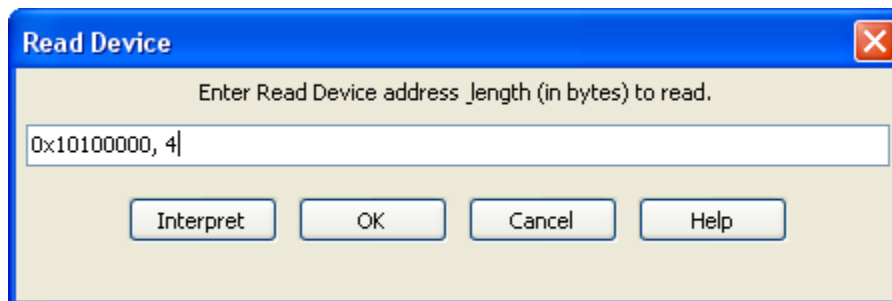
- Enter **PicoUtil** from the Windows cmd.exe line or press the Picoutil icon on the desktop. Either will load the following program:



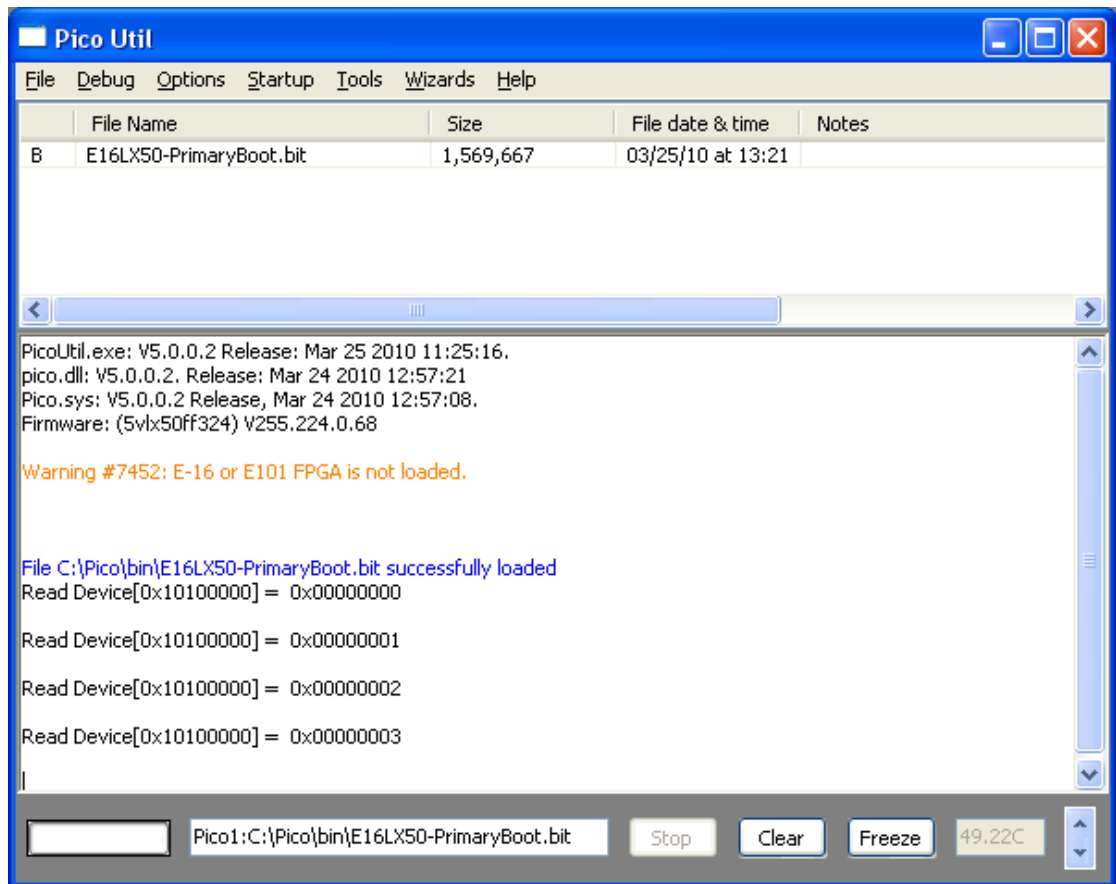
- To load a file go to the "File" menu select "Load FPGA file", navigate to C:\Pico\bin. Select E16LX50-PrimaryBoot.bit and hit open.
- When the file is completely loaded the dashboard will display the name of the file loaded. You can verify that the file is properly loaded by running the selftest or inspecting the pacing register



- Go to the Debug - Device - Read Device Register and enter what is below:



- Pico Util will go to that device register and read 4 bytes, if I do the same thing four times the screen will look like this:



- After performing four single word reads we see the value of the counter incrementing up.

4 Running PicoBus Counter

The PicoBus Counter Example

The PicoBus Counter example provides a quick and easy way to verify correct operation of your E-17 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 bitmap. You can also use Microsoft Visual Studio to modify and build your own application, based on the PicoBus sample.

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

To run the PicoBus Counter example:

1. Open a Windows Command prompt.
2. Navigate to C:\pico\Samples\PicoBus_counter\software_using_channel
3. 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 bitmap onto the E-17 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:

```

C:\WINDOWS\system32\cmd.exe
C:\Pico\Samples\PicoBus_counter\software_using_channel>PicoBus_counter.exe
 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.

Modifying and Building the PicoBus Counter Example

The source for the PicoBus counter software is in the following directory (this is *not* the firmware: see [E-17 PicoBus Firmware](#)):

```
C:\pico\Samples\PicoBus_counter\software_using_channel
```

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 `E16LX50_PicoBus_counter.bit`, located in `C:\pico\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:

```

14: // the built-in test counter is on channel #1. The test channel from the firmware project is #10.
15: #define CHANNEL_NO 10
16:
17: int main(int argc, char* argv[])
18: {
19:     int          erC, ii;
20:     uint32_t     buf[1000];
21:     const char   *bitFileNameP="$picobase\samples\Picobus_Counter\firmware\model_Picobus_counter.bit";
22:     cPicoChannel_Exx channel(CHANNEL_NO, NULL, bitFileNameP); //create channel

```

For more information, refer to the PicoBus documentation in `C:\Pico\Samples\PicoBus_counter\doc`.

5 E-16 PicoBus Firmware

NOTE: Xilinx ISE Design Suite 11 is required to build bit images, and webpack will work!

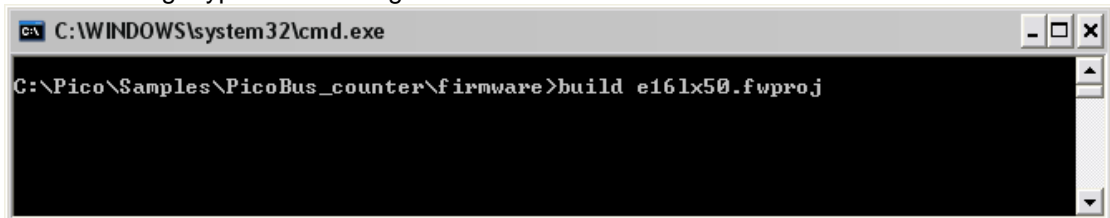
To build E-16 bit image, there are batch files supplied. These batch files call the different Xilinx tools to

build the bit images.

5.1 Batch File

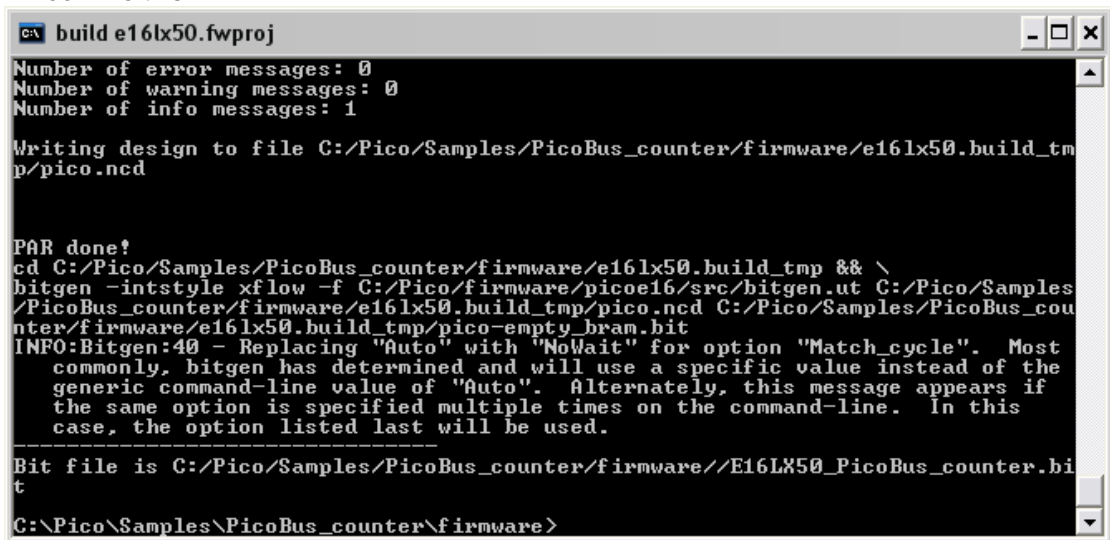
To build the firmware follow these steps:

1. Open a command prompt and navigate to C:\Pico\Samples\PicoBus_counter\firmware
2. To build the bit image type the following and hit enter:



```
C:\WINDOWS\system32\cmd.exe
C:\Pico\Samples\PicoBus_counter\firmware>build e16lx50.fwproj
```

3. This will call the Xilinx tools, and when the bit image is created a log file will open and the screen shot will look like this:



```
build e16lx50.fwproj
Number of error messages: 0
Number of warning messages: 0
Number of info messages: 1

Writing design to file C:/Pico/Samples/PicoBus_counter/firmware/e16lx50.build_tmp/pico.ncd

PAR done!
cd C:/Pico/Samples/PicoBus_counter/firmware/e16lx50.build_tmp && \
bitgen -intstyle xflow -f C:/Pico/firmware/picoe16/src/bitgen.ut C:/Pico/Samples
/PicoBus_counter/firmware/e16lx50.build_tmp/pico.ncd C:/Pico/Samples/PicoBus_cou
nter/firmware/e16lx50.build_tmp/pico-empty_bram.bit
INFO:Bitgen:40 - Replacing "Auto" with "NoWait" for option "Match_cycle". Most
commonly, bitgen has determined and will use a specific value instead of the
generic command-line value of "Auto". Alternately, this message appears if
the same option is specified multiple times on the command-line. In this
case, the option listed last will be used.
-----
Bit file is C:/Pico/Samples/PicoBus_counter/firmware//E16LX50_PicoBus_counter.bi
t
C:\Pico\Samples\PicoBus_counter\firmware>
```

- a. Note: the cursor will return to the command prompt when the log file is closed.
4. Bit image will be placed in
C:\pico\Samples\PicoBus_counter\firmware\E16LX50_PicoBus_counter.bit

Modifying the firmware

The PicoBus_counter.v is a good place to start modifying the firmware, and this file can be found in C:\Pico\Samples\PicoBus_counter\firmware\PicoBus_counter.v. The file can be opened with any text editor.

For more information, refer to the PicoBus documentation in C:\Pico\Samples\PicoBus_counter\doc.

6 Hardware Troubleshooting

Installation Problems

The E-16 is not recognized by Windows when inserted into the ExpressCard slot.

Some laptops require a re-boot before the Windows will see the E-16.