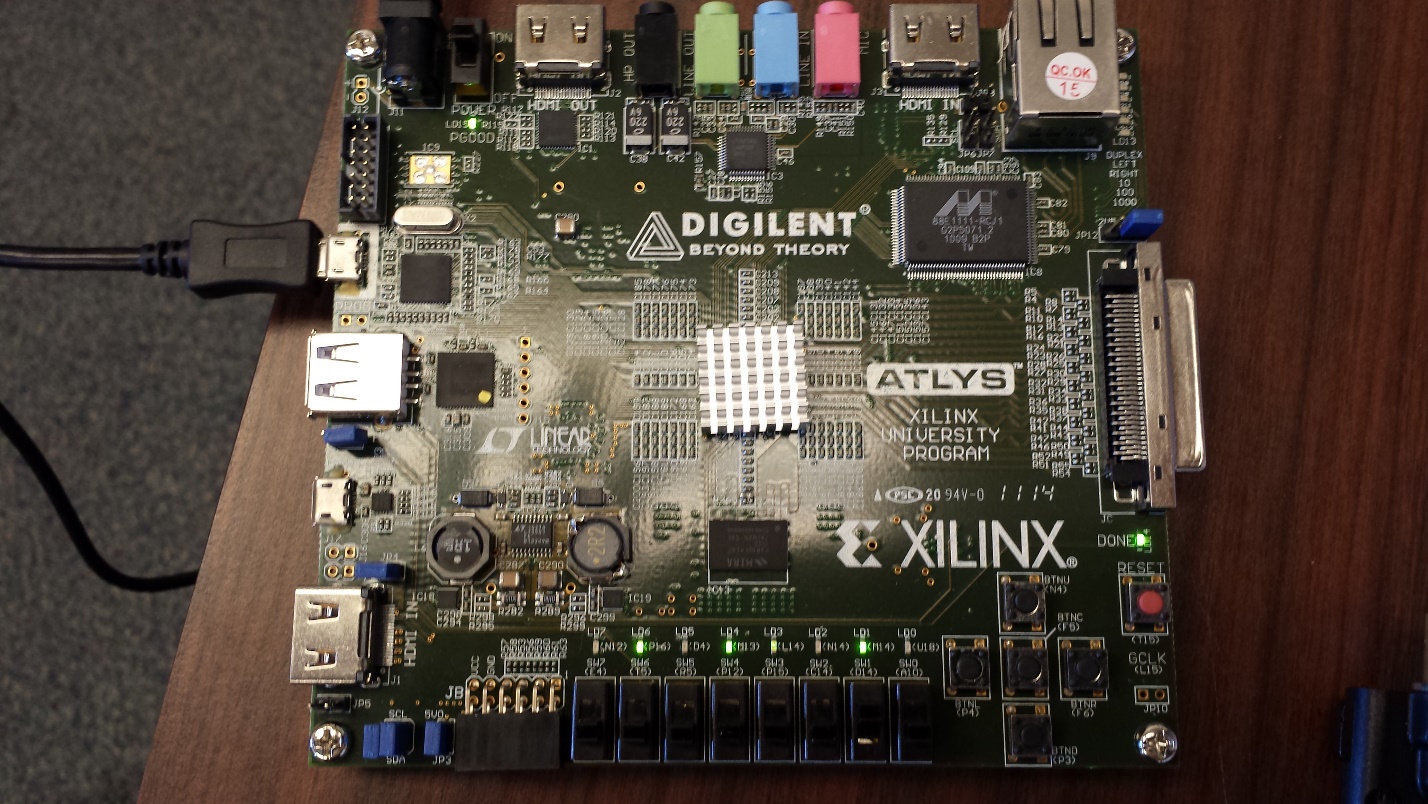
**Communicating with the Digilent Atlys board over USB**

Directory Structure:

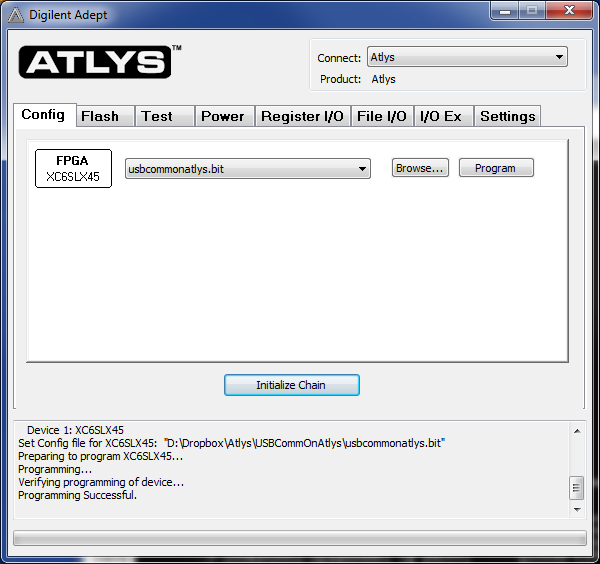
* DPCDemo – VC++ project used to communicate with board
* USBCommOnAtlys – Xilinx project
* digilent.adept.sdk\_v2.3.1 – latest version of Adept2 sdk
* Adept SDKv1-3 – version 1 Adept SDK

**Instructions:**

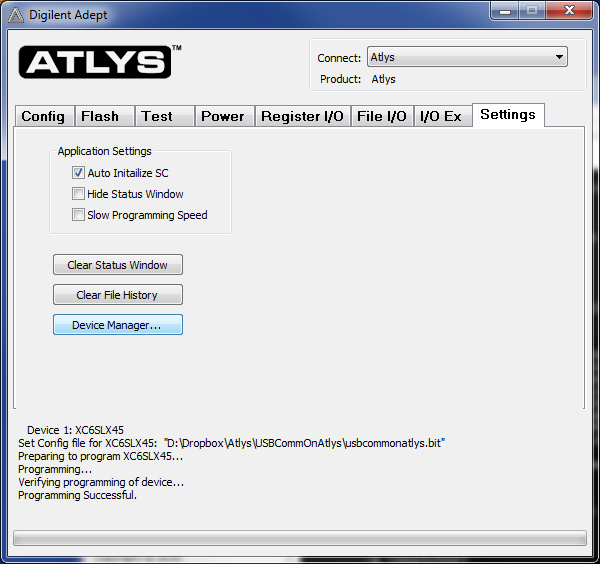
* Build Xilinx project [USBCommOnAtlys.xise](file:///D:\Dropbox\Atlys\USBCommOnAtlys\USBCommOnAtlys.xise):
  + [Block Diagram](USBCommOnAtlysBlockDiagram.pdf)
  + USBCommOnAtlys.v – top level Verilog component
  + 512x8 independently clocked FIFO using Coregen
  + Modified version of [dpimref.vhd](http://www.digilentinc.com/Products/Detail.cfm?Prod=ADEPT2) to connect to FIFOs
* Install Digilent [Adept 2](http://www.digilentinc.com/Products/Detail.cfm?Prod=ADEPT2)



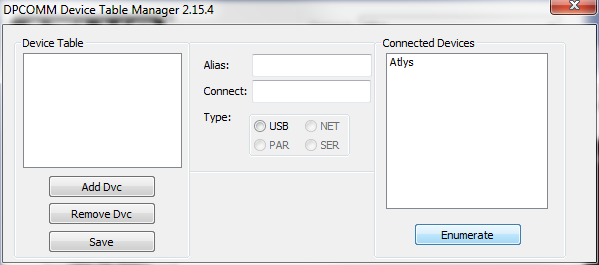
* Connect power and USB to Atlys board



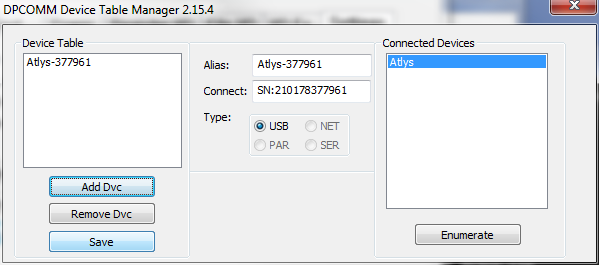
* Laucnh Adept2 and Program Atlys board with USBCommOnAtlys.bit
  + SW2:SW0 control LD2:0
  + LD7:3 are connected to Counter[26:22] clocked at 100MHz so you should see a frequency of 100 MHz/2^27 = 0.75 Hz on LD7



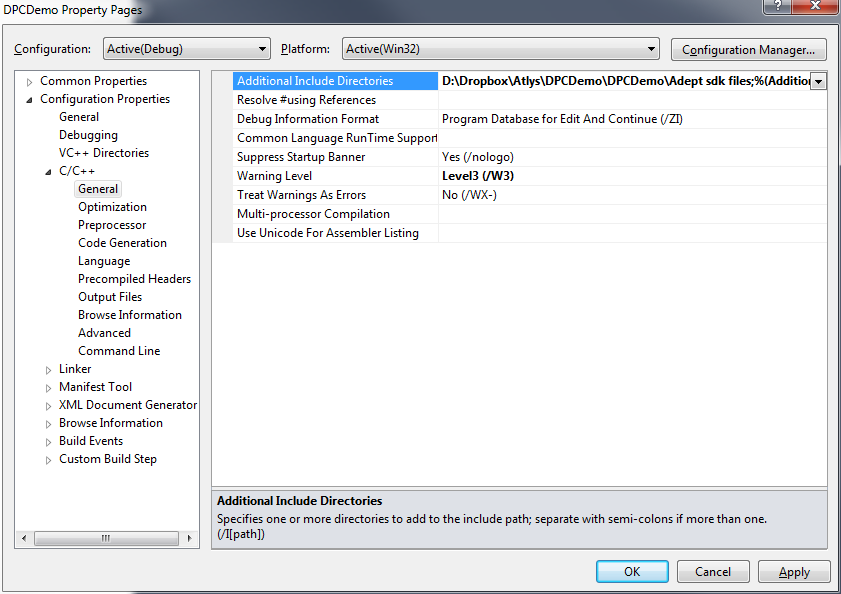
* Click on Settings -> Device Manager …



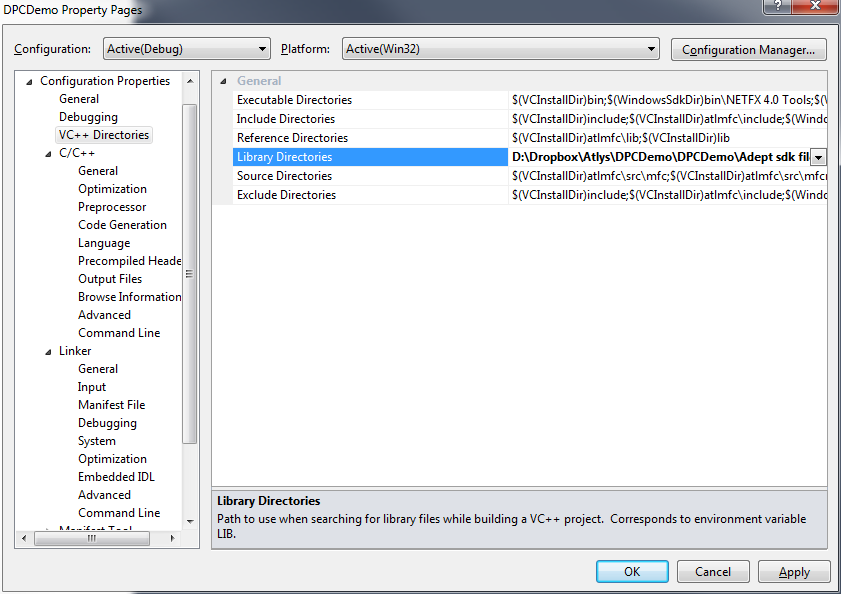
* Click on Enumerate



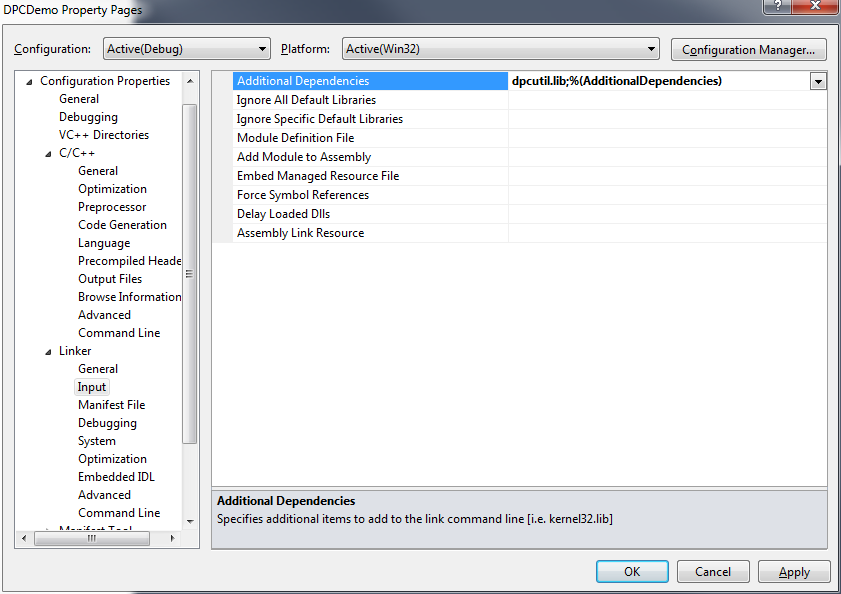
* Click on Atlys, give it an Alias, click on Add Dvc, click Save and close window. This sets up a default device needed for communication
* Build DPCDemo.exe
* Download latest version of [Adept2 SDK](http://www.digilentinc.com/Products/Detail.cfm?Prod=ADEPT2)
* Download Adept SDKv1-3. This is on the same page but listed un support documents
* Open [digilent.adept.sdk\_v2.3.1\samples\Building an Adept SDK Project in Visual Studio .pdf](file:///D:\Dropbox\Atlys\digilent.adept.sdk_v2.3.1\samples\Building%20an%20Adept%20SDK%20Project%20in%20Visual%20Studio%20.pdf) and follow instructions to create a project based on Adept SDKv1-3\Adept SDK\dpcdemo source



* + The instructions are for an earlier version of Visual Studio. Right click on the project to access the page and add the additional include folder that includes dpcutil.h



* + Add the library directory from the SDK that contains dpcutil.lib



* + Add dpcutil.lib
  + Build Project
* Open command prompt and execute DPCDemo.exe to transfer files in both directions

*D:\Dropbox\Atlys\DPCDemo\Debug>****DPCDemo.exe***

*Digilent DPCUTIL demo: version 1.00*

*DPCDemo.exe [options] <parameter 1> <parameter 2> <parameter 3>*

*Options:*

*-x Launch device dialog box*

*-g <register> Get register byte*

*-p <register> <data byte> Put register byte*

*-l <register> <filename> <# bytes> Stream file into register*

*-s <register> <filename> <# bytes> Stream register to file*

*D:\Dropbox\Atlys\DPCDemo\Debug>****type WriteToFIFO.txt***

*01234567890123456789*

*D:\Dropbox\Atlys\DPCDemo\Debug>****DPCDemo.exe -l 0 WriteToFIFO.txt 11***

*Stream to register complete!*

*D:\Dropbox\Atlys\DPCDemo\Debug>DPCDemo.exe -s 1 ReadFromFIFO.txt 10*

*Stream from register complete!*

*D:\Dropbox\Atlys\DPCDemo\Debug>****type ReadFromFIFO.txt***

*0123456789*

* I’m not sure why I have to write an extra character to read out what I wrote