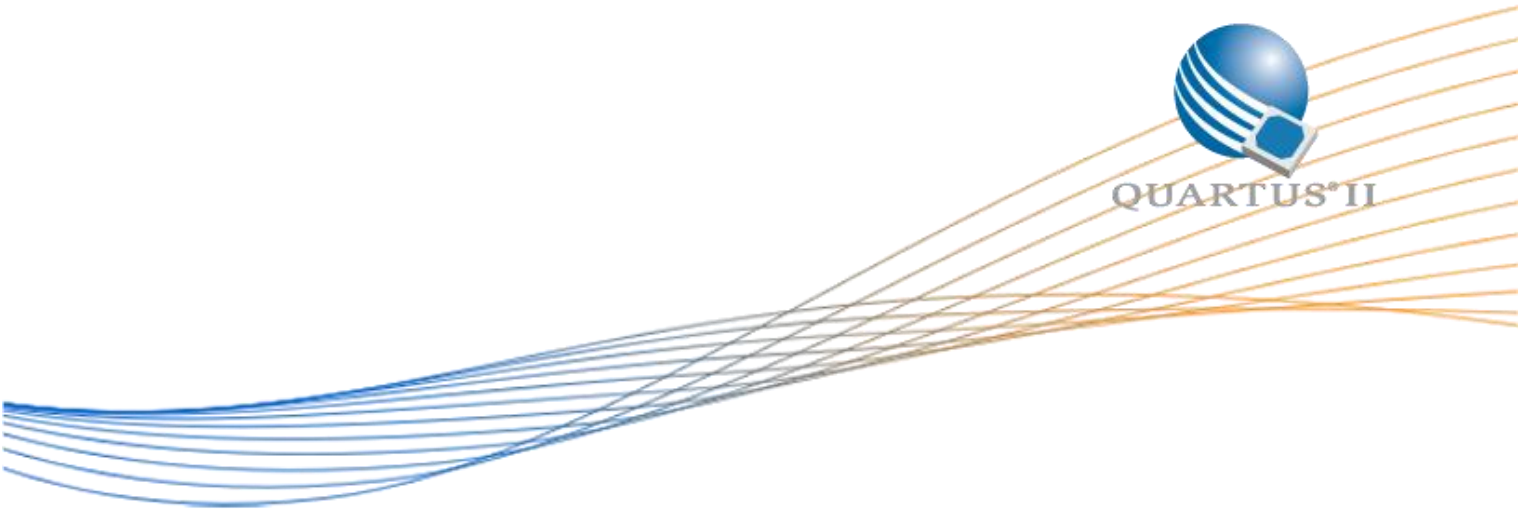


ADC and LCD Controller with a NIOS II processor for the MAX10 FPGA Development Kit using a Digilent PmodCLP LCD

Table of Contents

Overview

The following external parts are needed to demonstrate the design example;

- MAX10 10M50 FPGA Development kit
- Mini-USB cable for programming the device
- A Digilent PmodCLP LCD screen – can be purchased here
- Adafruit electret microphone available from adafruit.com
    - More information about soldering and connecting the microphone can be found in the Appendix.
- **IMPORTANT**: only use the 12V, 2A AC adapter that came with this kit. Do not use other power supplies from other Altera kits, these have higher voltage and may blow out the kit's power circuits

Theory of Operation

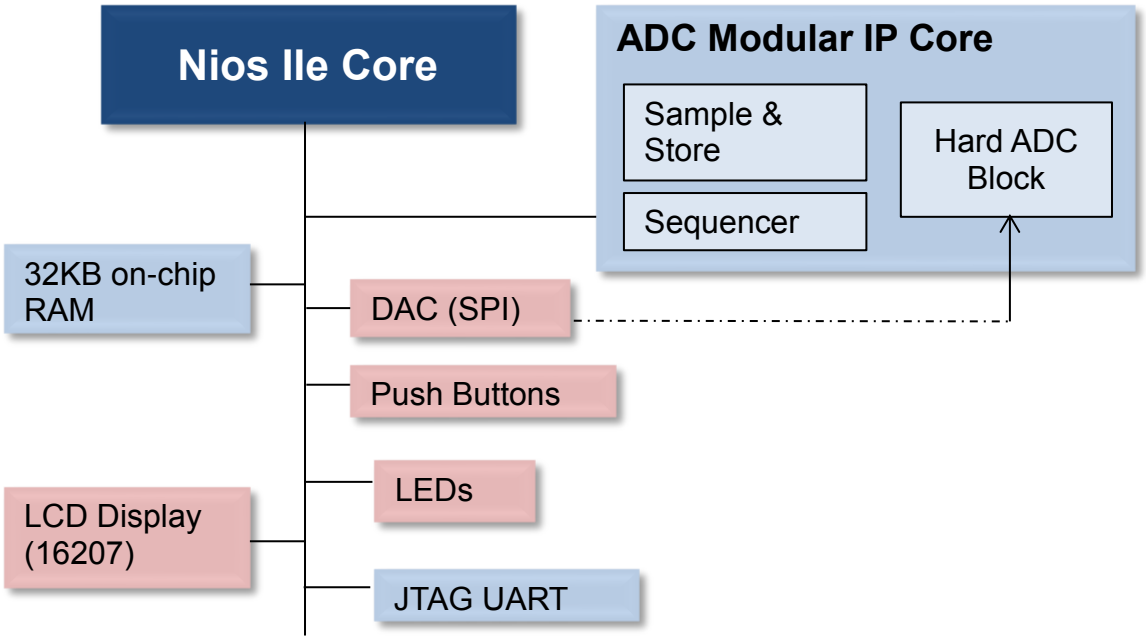
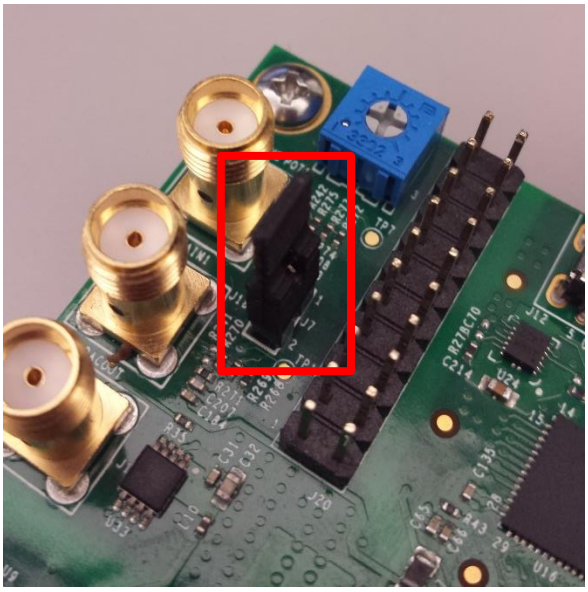


Figure 1: ADC/LCD Design Block Diagram

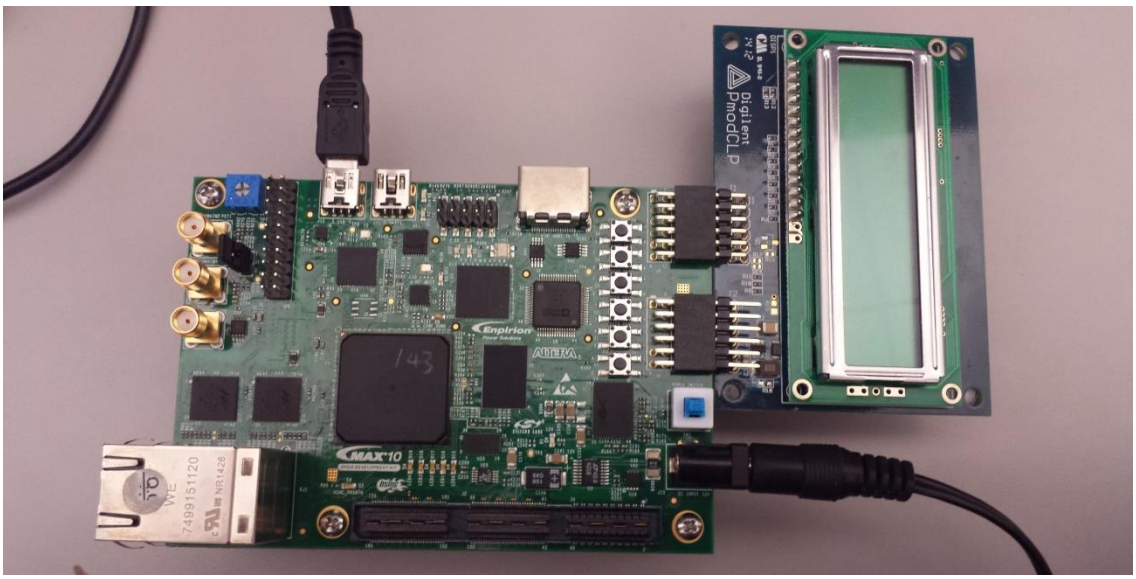
The ADC accepts an analog voltage reading from a potentiometer or microphone on the MAX10 FPGA development kit through channel 7. The NIOS II processor is then used to program an LCD to display the equivalent digital voltage reading.

ADC Simple Demo Setup

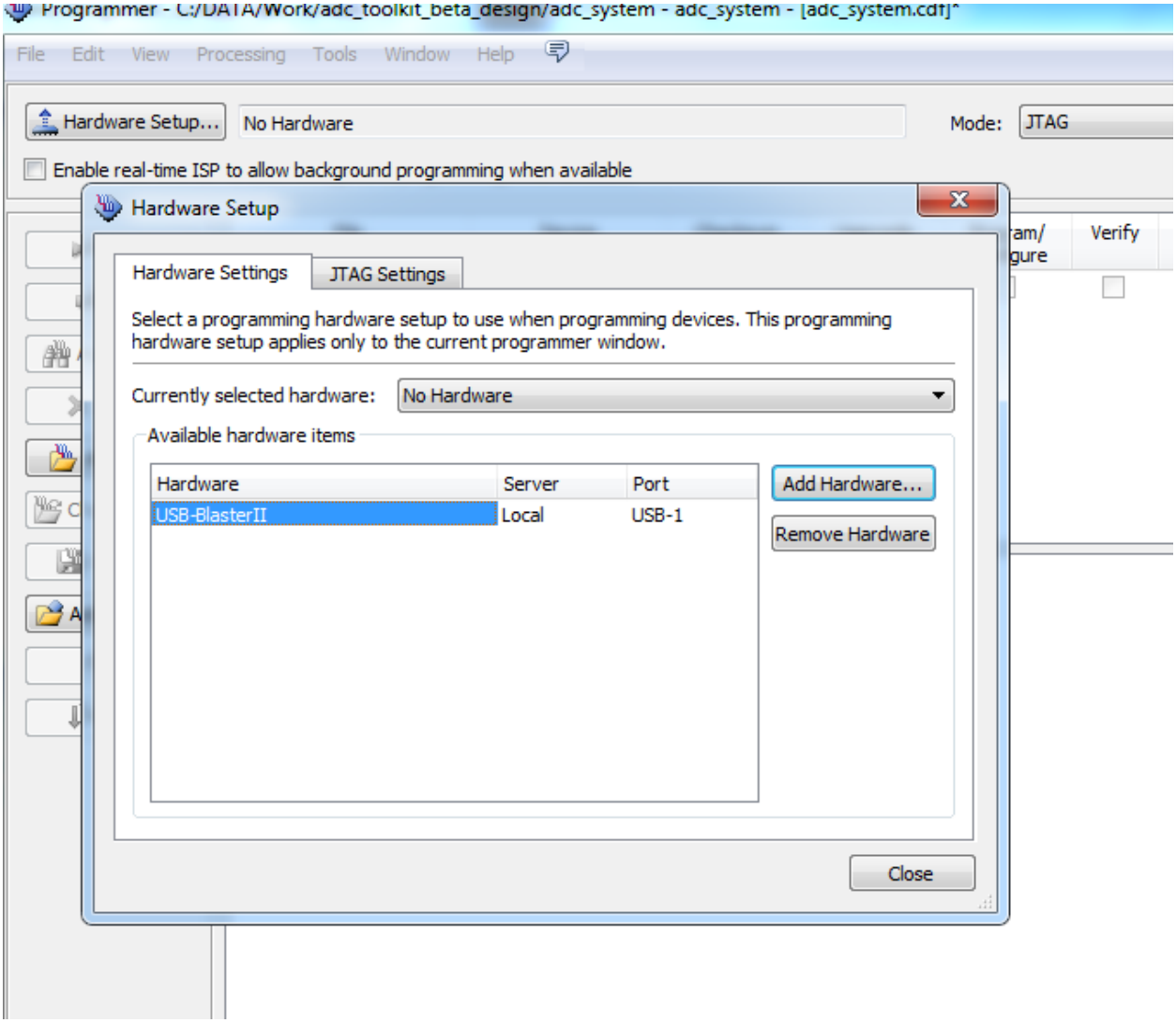
1. Connect the LCD card to the 2 PMOD connectors on the 10M50 Dev Kit. The PMOD Card's J1 connector mates with the Dev Kit's J5 and the PMOD Card's J2 connector mates with the Dev Kit's J4.
2. Add or make sure the jumper from your kit is placed onto the J7 jumper pins on the board near the SMAs. This connects the POT to an **ADC channel 6** in the design.



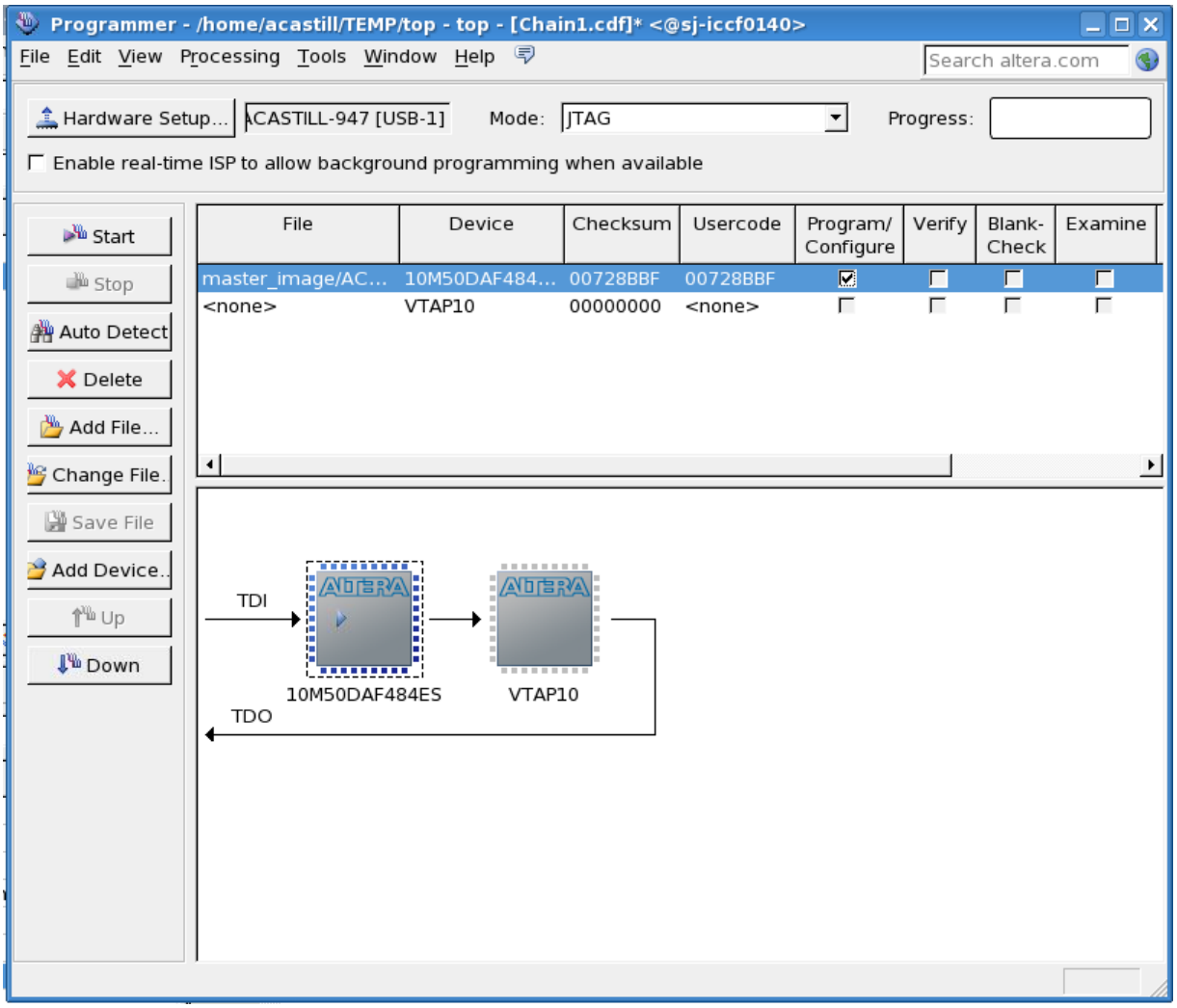
3. Connect the power cord to the power plug, and connect a mini USB from your PC/laptop to the J12 USB connector (labeled as USB 1 on the silkscreen) on the top left of the kit. You're set up should look like below:



4. The design package includes a pre-compiled version of the design that downloads to the MAX 10 embedded flash. This version of the configuration includes the combined FPGA configuration and NIOS executable. Follow these instructions to load the design onto your kit:
    a. Make sure the board is powered on (push the blue power button) and that the USB is connected on J12/
    b. Startup Quartus. Then start the programmer from Tools → Programmer.
    c. Click Hardware Setup → USBBlaster II and select Add Hardware

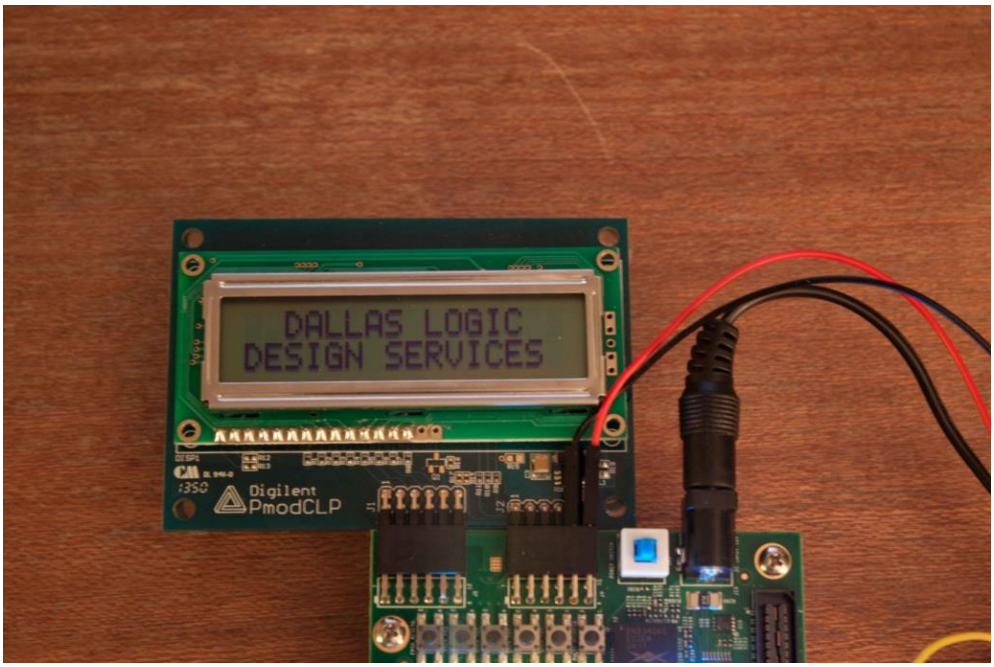


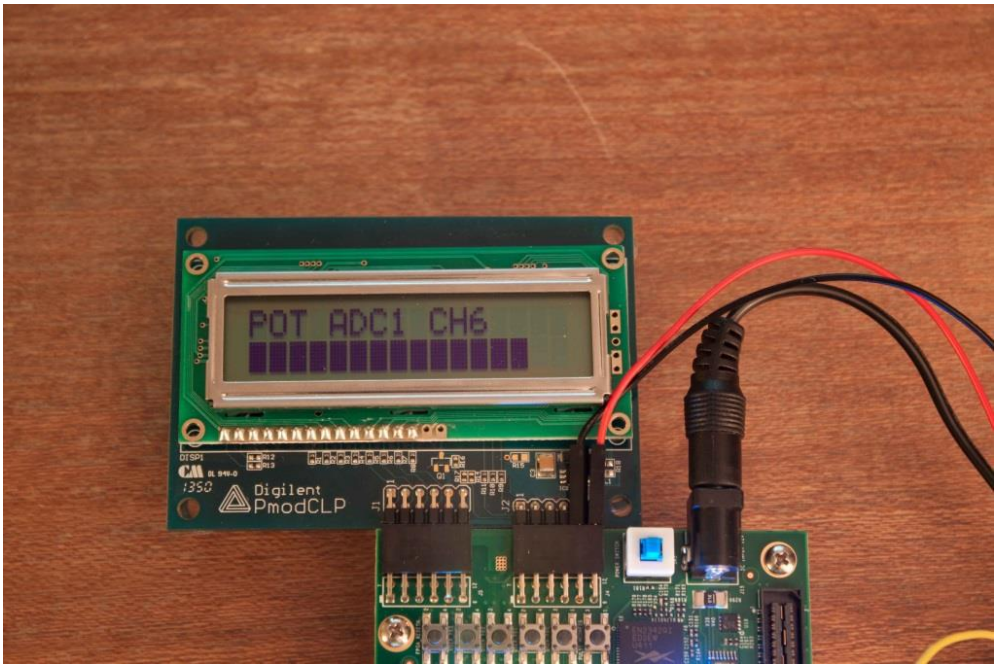
    d. Click “Auto Detect” and double click where it says <none> on the first row to add your .sof/.pof file from the master_image folder. Note: A virtual TAP controller (VTAP) will always show in the JTAG chain – this is correct behavior.



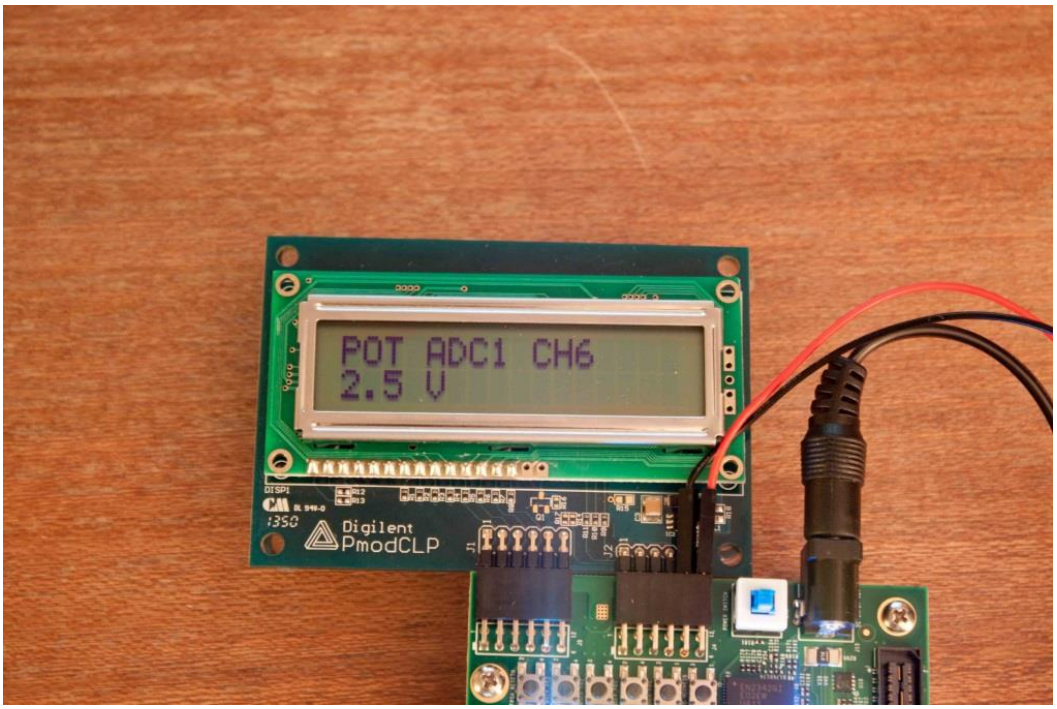
e. Fill in the checkbox for “Program” and select “Start” to begin downloading the design to the board – your dev kit will start running the design when the process is successful.

5. Push the USER_PB3 button (2nd push button from the top of the board) to scroll through the startup screen and title. Stop when you reach the POT ADC1 CH6 screen title.





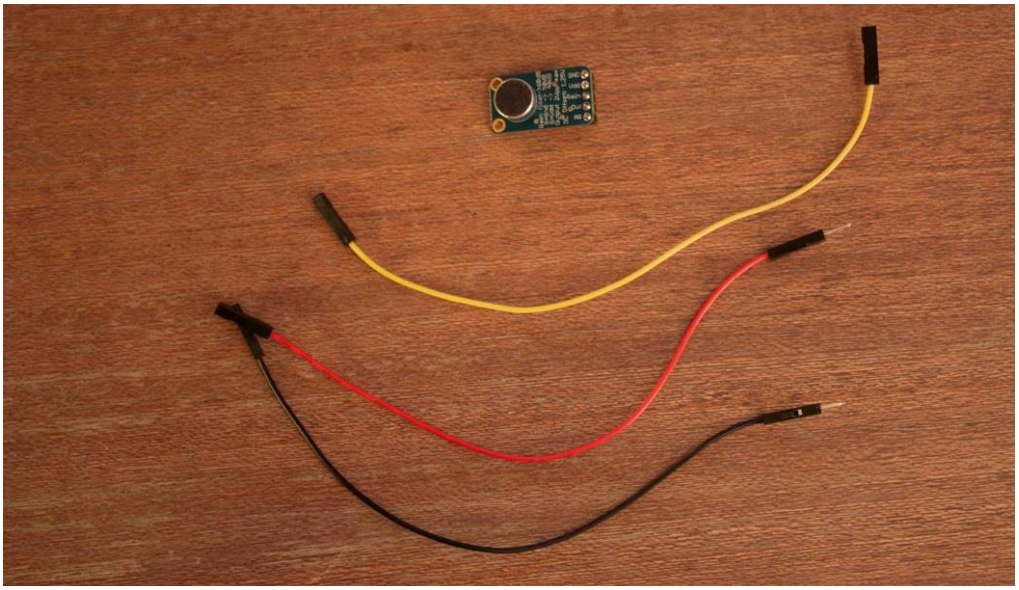
6. Use a small screwdriver or your fingers (if knob is attached to your POT) to turn the potentiometer and see the voltage meter change up or down. MAX 10 FPGA's ADC block input channel is measuring the analog voltage and using a Nios II system to display to the LCD.
7. If you push the upper right push button USER_PB1, the bar graph will change to a numeric voltage reading. You can go back to meter mode by pushing the button again.



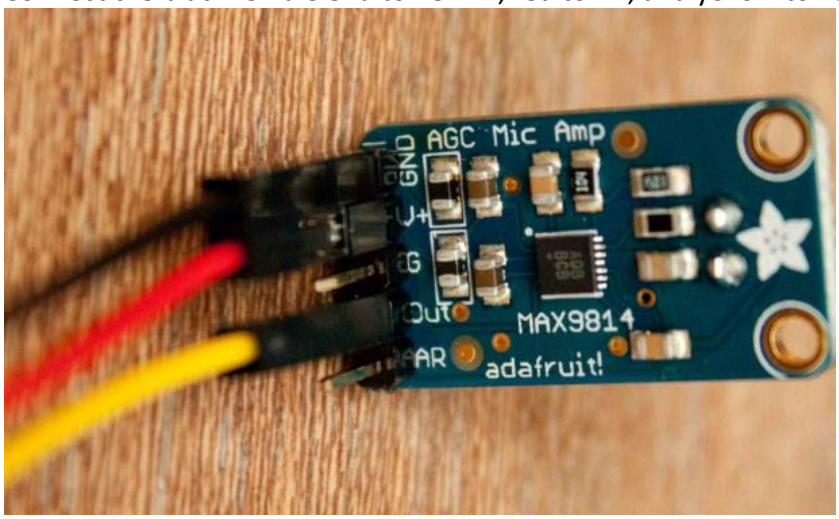
Optional Microphone Demo Setup

The same design file/POF in the device can be used for a microphone to ADC demonstration. Here are the instructions.

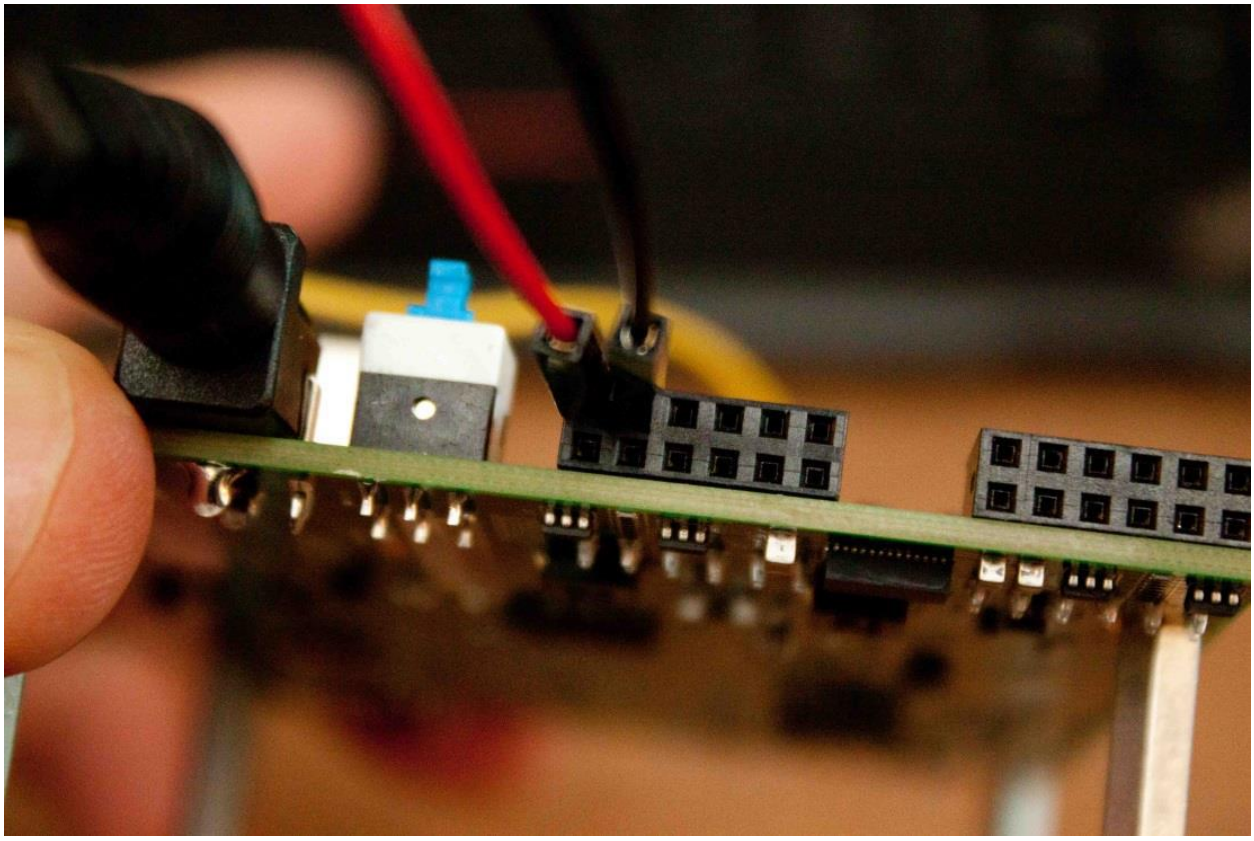
1. Take two male-female wires and one female-female wire. I chose red, black for the male-female and yellow for the female-female. Your colored wires may vary depending on what you received.



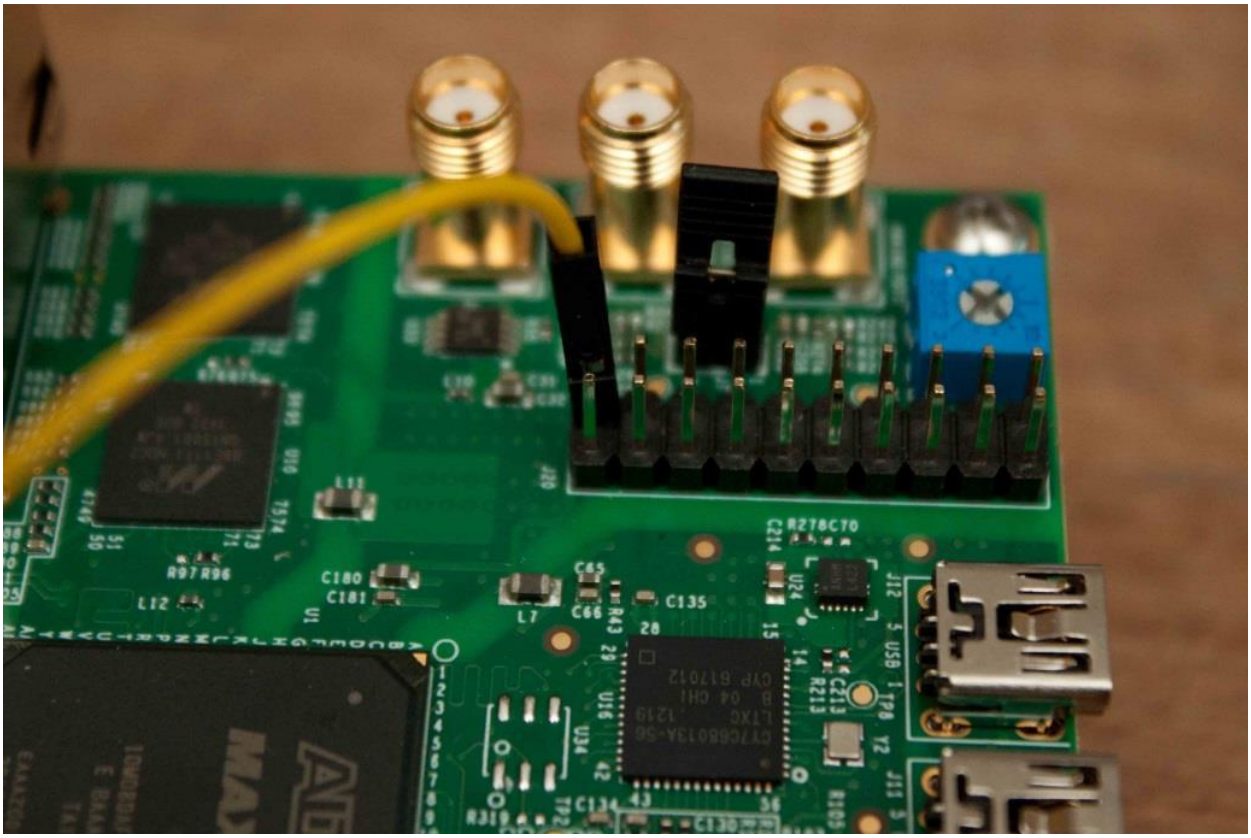
2. Connect the black female end to “GND”, red to V+, and yellow to “Out” on the microphone



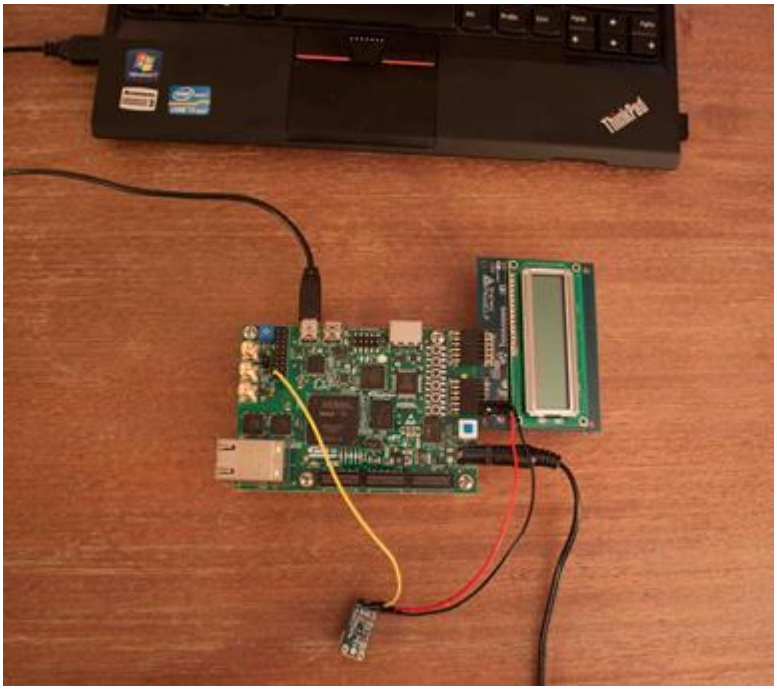
3. Temporarily remove the LCD to make it easier to connect the microphone
    a. Connect the male end of the GND/Black wire to pin 5 of the J4 connector.
    b. Connect the male end of the V+/Red wire to pin 6 of the J4 connector. Looking at the edge connector, this is the PMOD closest to the power button, in the two top left most signals. This powers the microphone from the PMOD J4 connector’s VCC and GND.



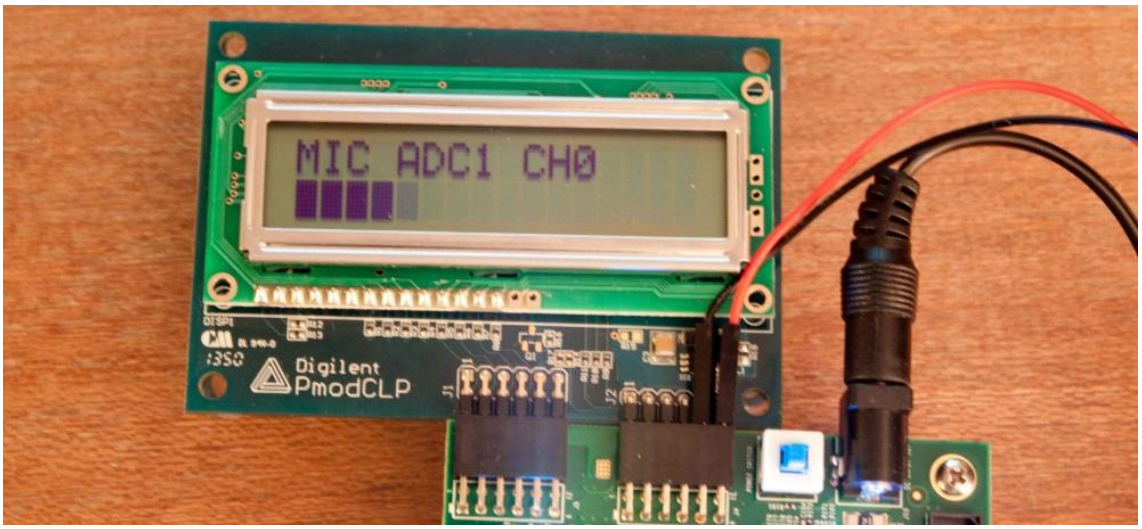
4. Connect the remaining female end of the yellow/”Out” wire to pin 1 of the 2x10J20 connector. **This routes to ADC channel 0**. See image below.



5. Your final set up should look like this after your replace the LCD:



6. Re-program the .sof/.pof file in the master_image if necessary, and turn on the dev kit. Like before, press USER_PB3 to go through the start-up screens until you see the following:



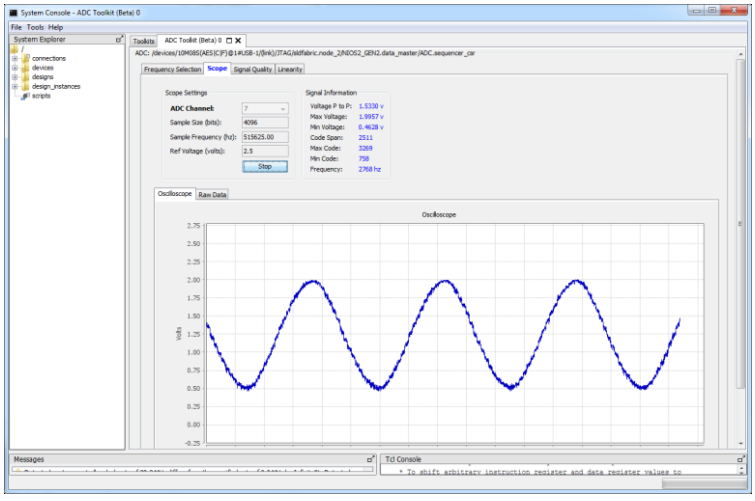
7. For this demo, you can blow into the microphone, speak, or play a song from your smart phone and put the microphone up to the speaker. The MIC is connected to ADC1 CH0 – it is not on a dedicated channel.

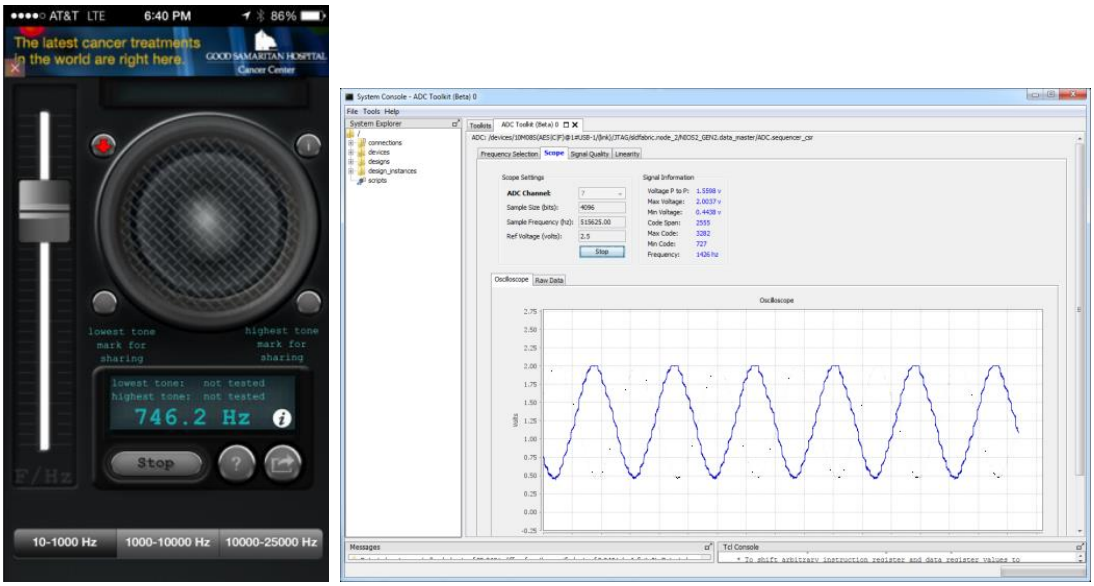
   You should see the bar graph change as the volume varies. You can also press USER_PB1 to switch between a bar graph or a numerical reading.

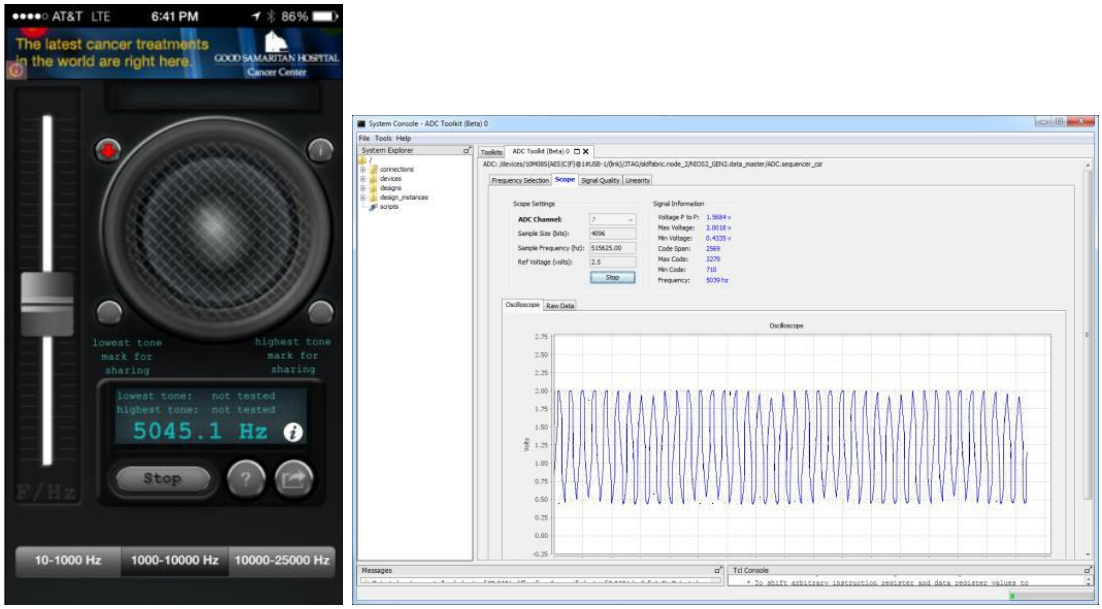
Optional Cool Microphone Soundwaves Demo

You are now ready for introduction into the ADC Toolkit (in Beta starting in 14.1). In this part of the demo, we illustrate Tools capabilities and features. This should not be used at this early stage to measure ADC performance. In this demo we can show how everything demonstrated on the LCD can be quickly acquired and displayed by the toolkit.

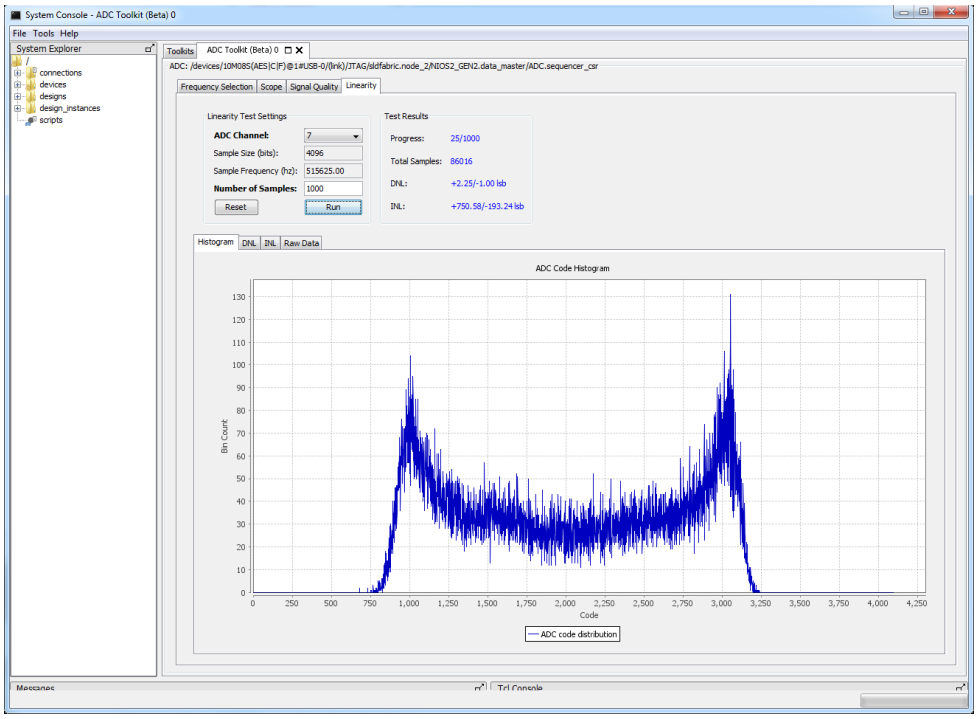
1. Download one of the following sound wave apps for your smart phone:
   - Android Sound Wave Generator (free) – this one is untested
   - iPhone Tone Generator (free)
2. Launch the app on your phone and turn up sound on phone volume
3. Open the ADC Toolkit from System Console. Click on the Tools menu -> System Debugging Tools -> System Console
4. Click on Load Design… and click on the "master_image/niosMAX10DevKit_adc_lcd.sof" file that came with this design. System Console does not configure the device using the SOF, but instead extracts debug file information from the SOF format.
5. After loading the SOF file into the system console, the JTAG connection to the toolkit becomes available, click on Launch shown in red below to open it.
6. Select ADC Channel 7 from the scope view and click on "Run" to see the output of the microphone. It should contain a noisy output.
7. Open your iPhone or Android tone generator App and hold the speaker up to the microphone. It's best to let the phone lie on the desktop next to the phone speaker source. Turn up the volume and run the tones from the app.
8. Sweep from low to high frequency on the application, you will see analog capture in the form of sine waves that increase with frequency as you sweep the Apps tones to higher frequency. See examples below…







9. Click Run on Signal integrity to see FFT results and center frequency.
10. Hit Stop and hit run on the Linearity tab to see a histogram of measured values.
11. If you run the histogram in the Linearity panel, you get the expected bath tub result after accumulating samples from a sine wave.



12. As you go higher in frequency you will find that you stop hearing the sound but the ADC Toolkit still shows waveforms, this is because human audible range will cut off at high frequency but the microphone still captures it. The older the person listening the lower their high frequency cut off will be. The youngest people in the room will always hear higher frequencies than the older folks.

How to compile the hardware

Follow the steps on the Design Store web page to extract and install the devkit_adc_lcd_controller platform file. The following steps describe how to setup a project in Quartus II software in order to program the MAX10 FPGA device with the ADC/LCD demo design.

Note – extract platform

i) Launch Quartus II software and open the project top.qpf using File->Open Project.

ii) Compile the Project by clicking the  button.

iii) Launch the Quartus II programmer from the Tools menu or alternatively by clicking the  button.

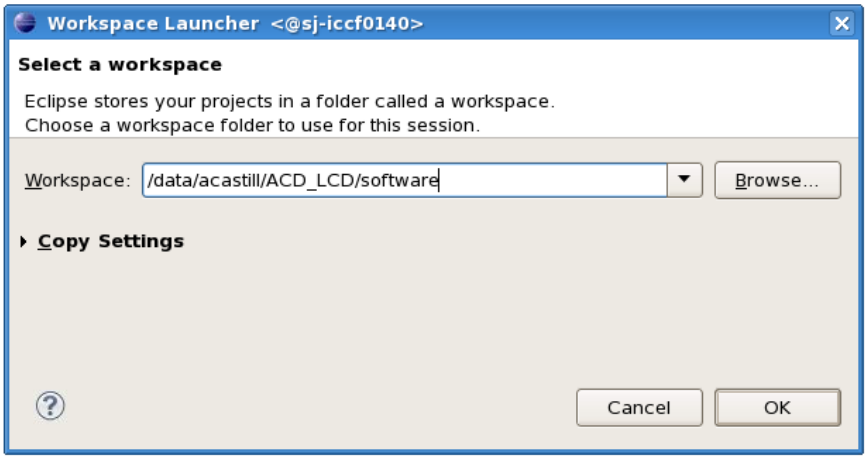
iv) Download the .sof file output_files/top.sof and program the device using the programmer as previously described.

How to compile the software

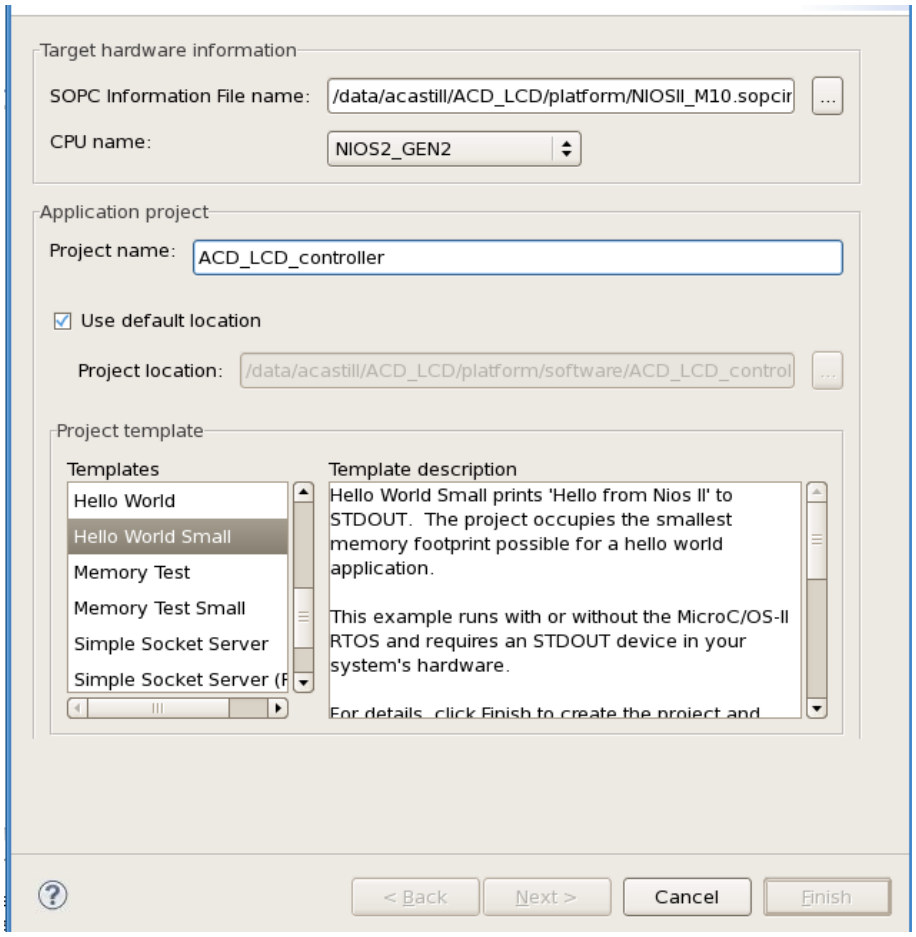
The following steps describe how to use Nios II Software Build Tools for Eclipse software to perform the following tasks;

i) Create a software project (BSP and application) from a .sopcinfo file
ii) Add source code (that is used to the program the LCD) to the project
iii) Download and run source code on the target processor (NIOS II)

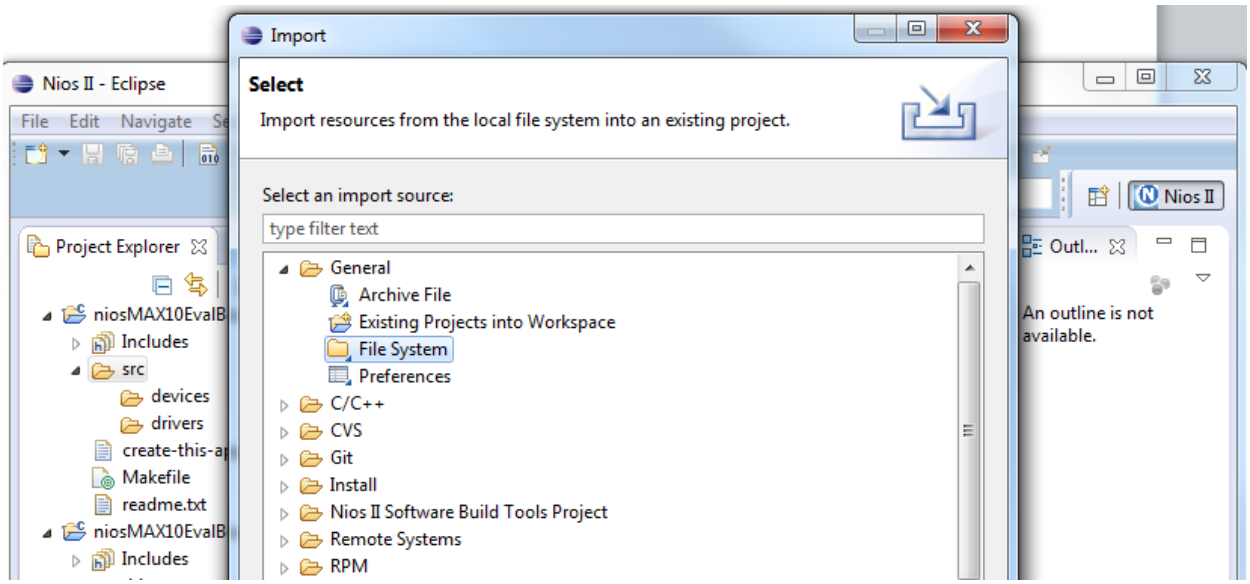
1. Open Quartus II on a windows platform
2. Launch Nios II Software Build Tools for Eclipse from the Tools menu
3. Specify the workspace directory for the project and click ok.



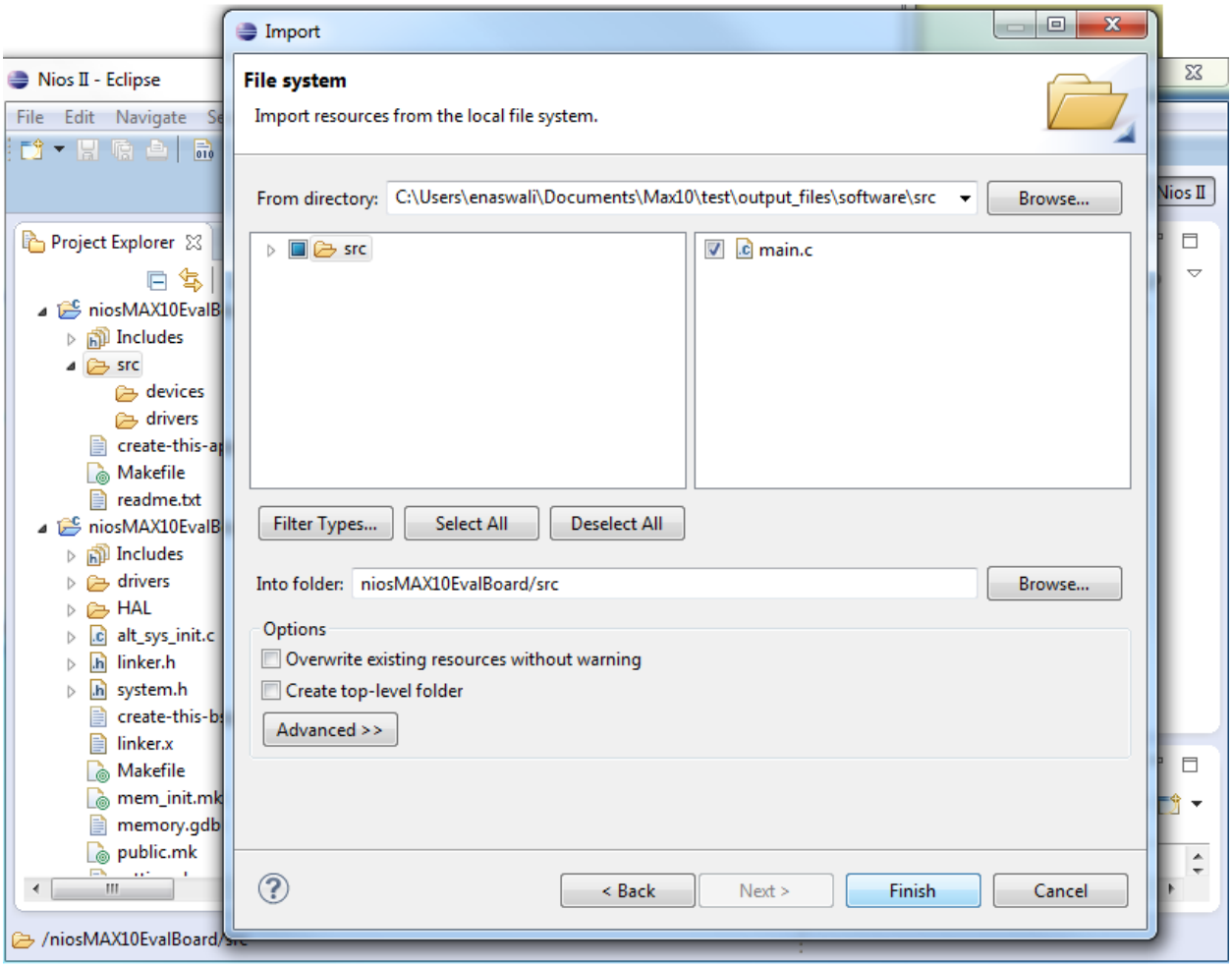
4. A blank workspace window pops up, right click anywhere in the Project Explorer and select the following to create a new software project; New->Nios II Application and BSP (board support package) from Template.
5. In the window that appears, browse to the location of the .sopcinfo file in the project folder and add it to the project. Provide a name for the software project i.e. ADC_LCD_controller and select “Hello World Small” for your template. Click Finish.



6. A BSP is created and located at ADC_LCD_controller and an application is created and located at ADC_LCD_controller. The application contains source code from the template design “Hello World Small” that needs to be replaced with the source code located in the software/src folder.
7. Select and right click hello_world_small.c under the main project explorer tab and click delete.
8. Right click the project name and left click the import command.
9. Navigate to General → File System
    a. Browse to the location of src, click on the box next to it, add it as shown and click Finish.

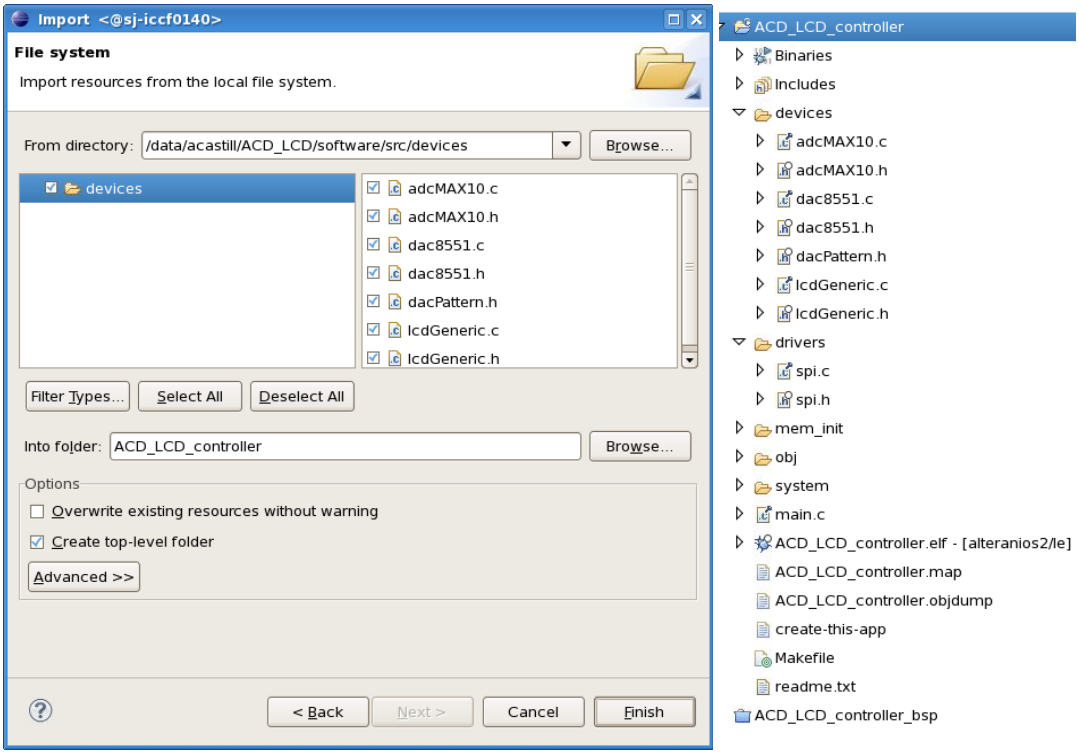


b. You should see main.c and 2 folders: device and drivers added to your project.



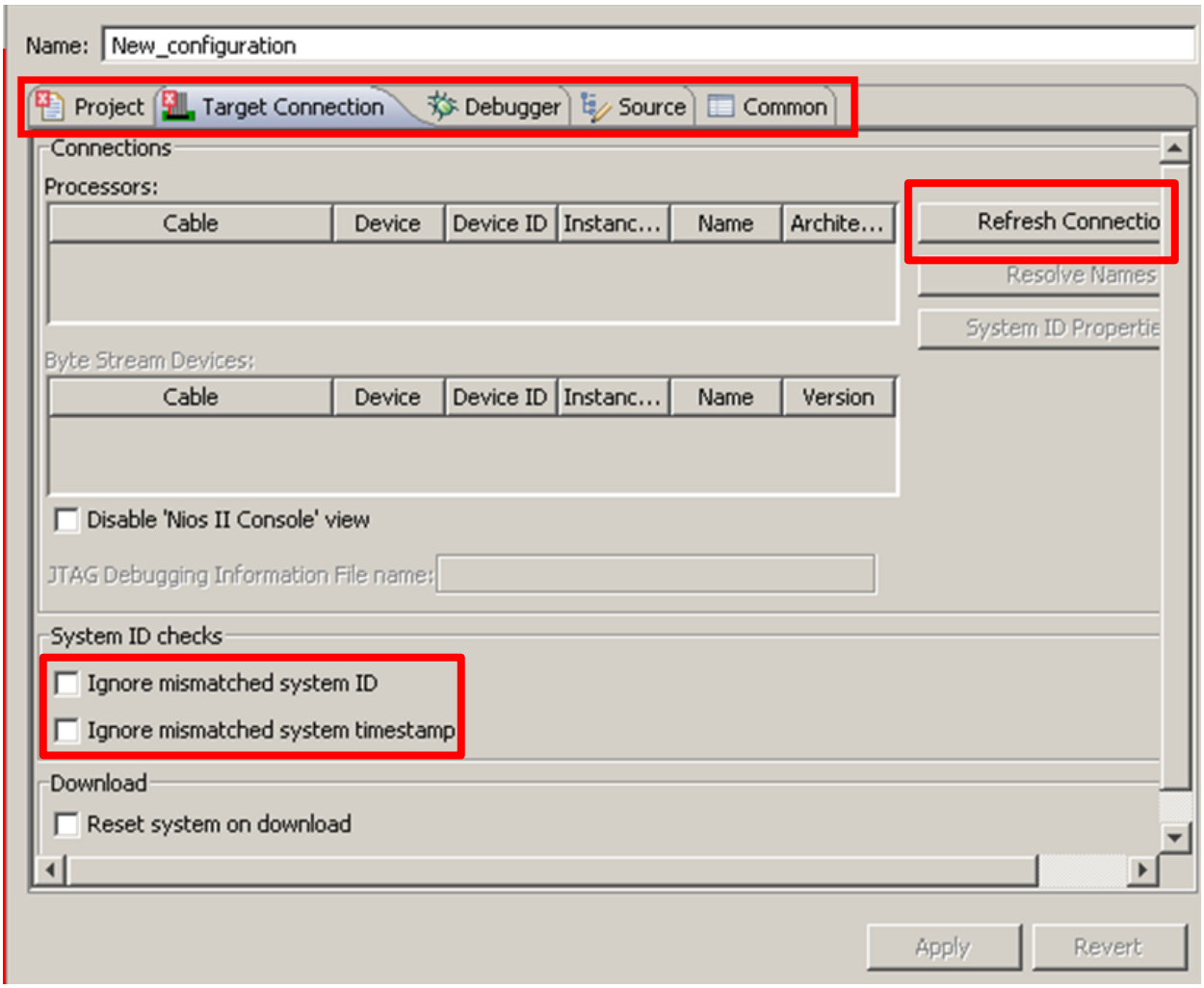
10. Import the contents of the devices and drivers folders under software → src

Check the folder to select all the items at once, and check “Create top-level folder”



Your project directory on the left should look similar to this (mem_init will be created later)

11. Compile the project by selecting Build All from the Projects Menu or alternatively Ctrl+B.
12. Program the FPGA device by launching Quartus II programmer from the menu and loading the file output_files/top.sof .
13. . Load the executable to the processor by right clicking on the Application project and selecting Run-As Nios II Hardware.
14. In the Run Configurations page under the Target connections tab. Check the options; Ignore mismatched system ID and Ignore mismatched system timestamp. If you don’t see the connection specified, click Refresh Connections. Click Apply followed by Run.



15. Now your hardware image is downloaded to the FPGA through the Quartus programmer and the NIOS code has also been downloaded through Eclipse. You can proceed to the demonstration as discussed in the earlier steps. The method described works well if you are modifying your code. You can maintain the same FPGA download and iterate through code revisions.

Merging the NIOS executable into the FPGA configuration file

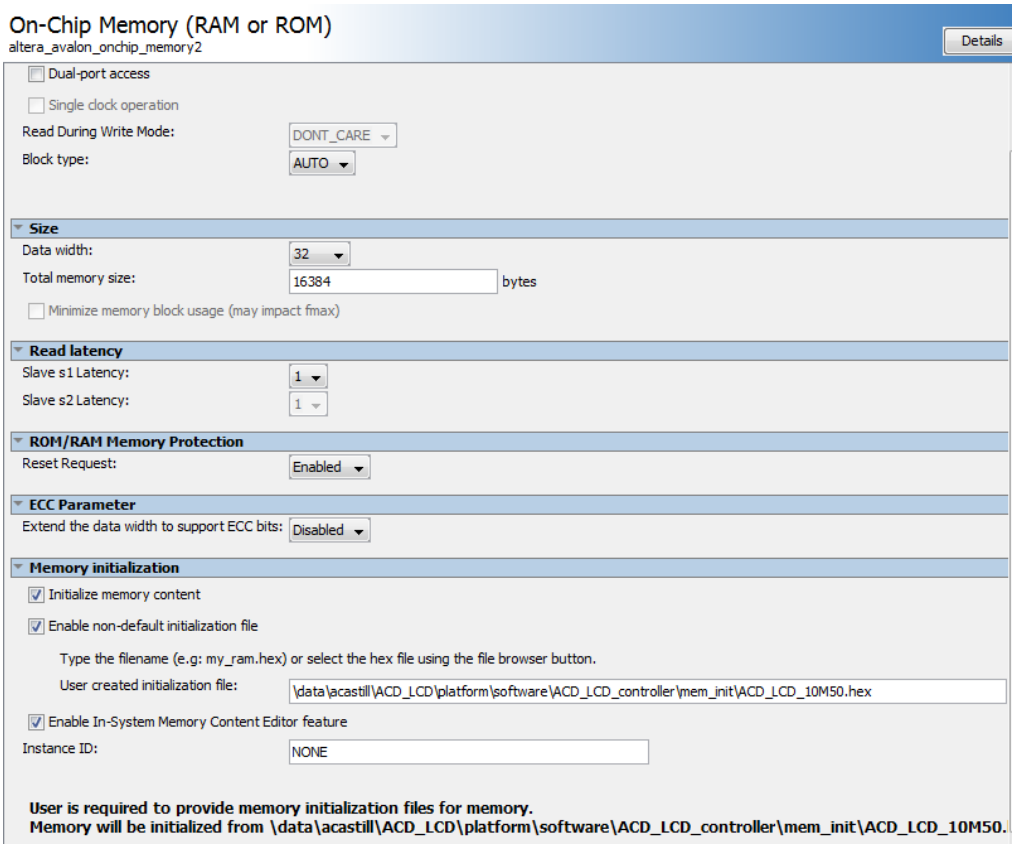
The master_image directory shipped with the design example .sof and .pof files have the NIOS executable incorporated in them. The prior steps detailing how to build your hardware and 21 software show you how to compile the hardware without the image loaded in the .sof or .pof file. In those steps, you load the NIOS executable from Eclipse through the Run → NIOSII Hardware step.

This section will allow you to merge the .elf executable into the .sof once your software is stable. In Eclipse, right click the project and select Make Targets. Click on mem_init_generate. You will generate a .hex file that is in the location where your software build is located (e.g):

<project directory>/mem_init/<name>.hex

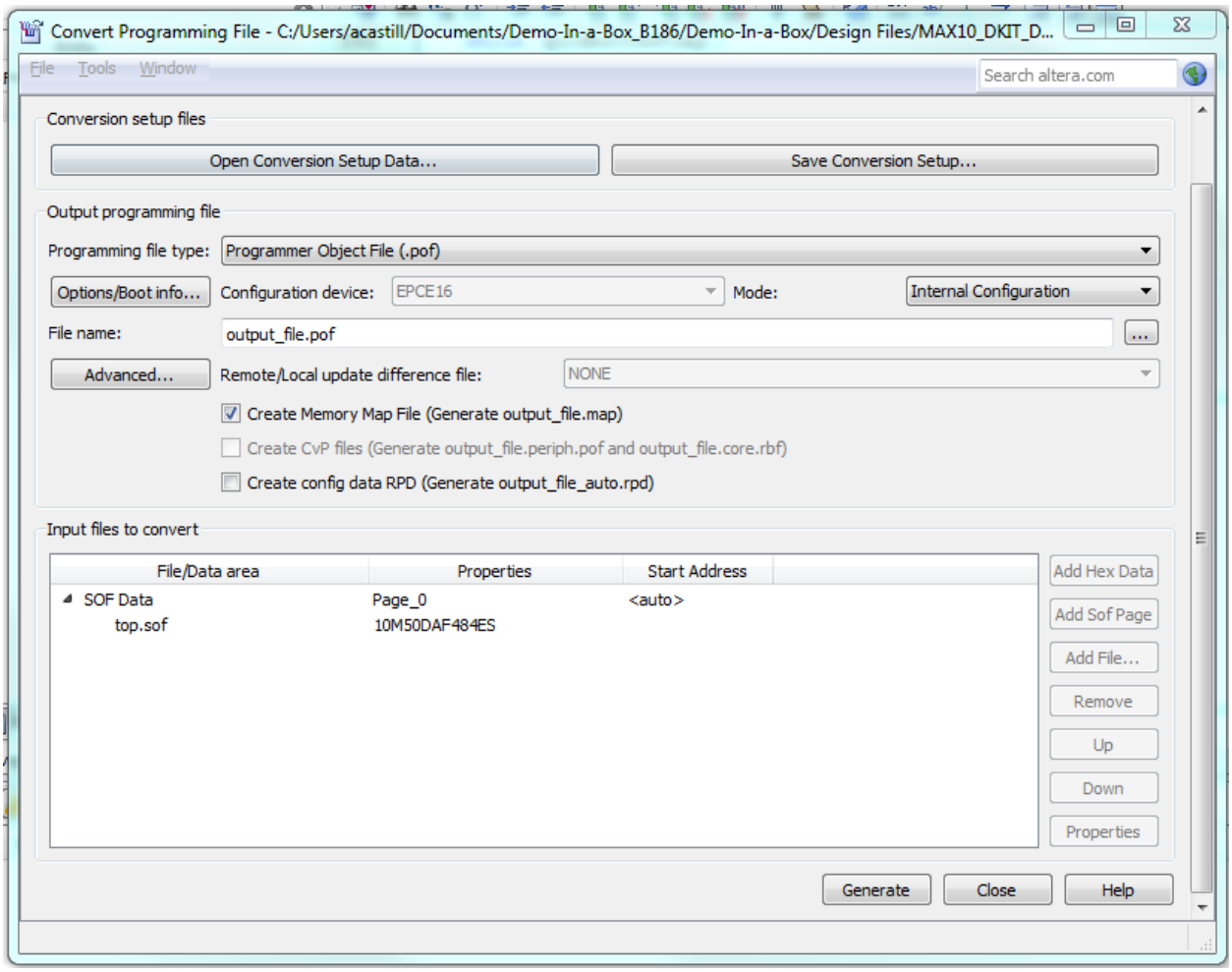
(The name of the .hex file is determined by the settings in your mem_init file under the BSP directory.)

Next, you need to return to Qsys and change the ONCHIP_RAM component by double clicking it. The initialize memory content and enable non-default initialization file needs to on. Enter the location where the .hex file is located.



Next, click Generate HDL → Generate. Return to Quartus and click compile and you will generate a new .sof file with the NIOS executable included. Run the programmer and download the new .sof. Run the demonstration as described in the previous steps.

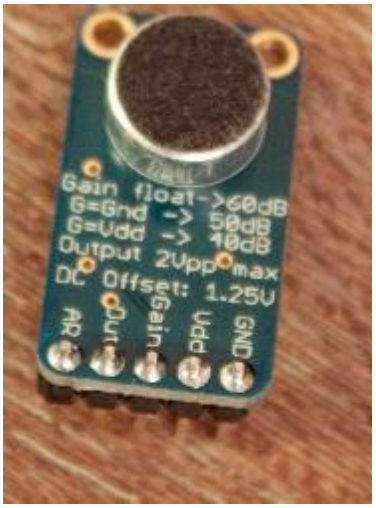
To generate a .pof file for non-volatile demonstrations, in Quartus run File → Convert Programming Files. Change mode to Internal Configuration. Highlight SOF data, click add file and select output_files/top.sof.

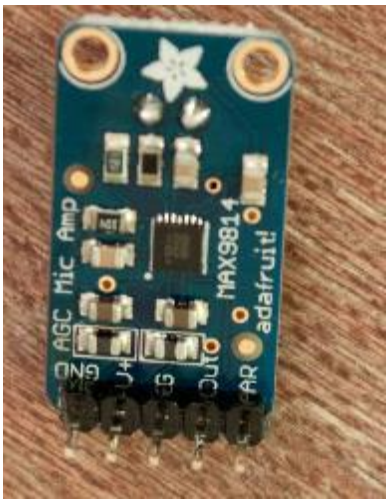


Click generate, and you will see the output_files/top.pof file to download with the programmer. The demonstration will continue to run even after power cycling the development kit.

Appendix A – Microphone soldering instructions

Solder the 5-pin header that comes with the board such that the long end is at the back side of the microphone PCB. The short end of the connectors should be soldered at the front of the PCB. If the pin header has more than 5 pins, cut off the extra pins using wire cutters.





Acknowledgements: This design example was created by Dallas Logic http://dallaslogic.com/