

ADC and LCD Controller with a NIOS II processor for the MAX 10 FPGA Evaluation Kit using the SainSmart LCD Arduino shield

©2014 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

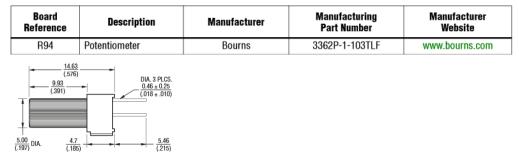
Table of Contents

Overview	2
Theory of Operation	3
ADC Simple Demo Setup	3
Optional Microphone Demo Setup	9
Optional Cool Microphone Soundwaves Demo	11
How to compile the hardware	17
How to compile the software	17
Merging the NIOS executable into the FPGA configuration file	20
Appendix A - Microphone soldering instructions	22

Overview

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

- MAX10 FPGA evaluation kit.
 - A potentiometer must be purchased and soldered onto the kit at R94. More information about purchasing and soldering the potentiometer can be found here <u>MAX10 FGPA Evaluation Kit User Guide</u>. Potentiometer must be purchased and soldered onto the kit. The Evaluation kit user guide shows model to order with finger adjust knob. Part number 3362P-103TLF must be ordered and soldered to the Evaluation kit R94 location.



- Mini-USB cable for providing power to the kit.
- USB-Blaster II cable for programming the device (note this kit gets its program download through the JTAG port. The Blaster cable bridges USB to JTAG).
- Sainsmart LCD keypad Arduino shield. This can be purchased here <u>Sainsmart</u>
- Adafruit electret microphone available from adafruit.com. More information about soldering and connecting the microphone can be found in the Appendix.

Theory of Operation

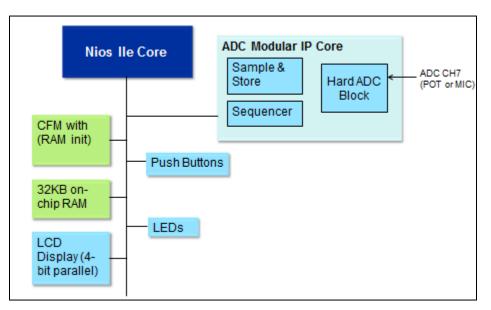


Figure 1: ADC/LCD Design Block Diagram

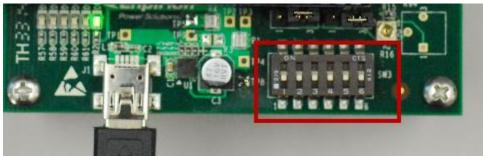
The ADC accepts an analog voltage reading from a potentiometer or microphone on the MAX10 FPGA evaluation kit through channel 7. The NIOS II processor is then used to program an LCD to display the equivalent digital voltage reading.

ADC Simple Demo Setup

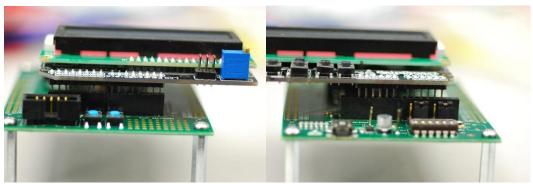
1. Place your MAX 10 10M08 Eval kit and sainsmart LCD keypad shield on the bench



2. All switches on the DIPSWITCH on located near bottom edge of the board must be in the DOWN position or the demo will not run. Please make sure all dipswitches are in the down/off position.

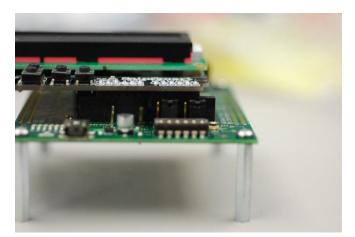


3. Pair the Sainsmart LCD's male pins with the Arduino female connectors on the Eval kit. There is a gap difference between headers on top and bottom such that you cannot insert the Arduino shield in the wrong direction. To insert the Sainsmart LCD shield correctly the Altera logo and the SainSmart logo should be in same direction and orientation. Make sure the LCD shield is pushed down snug with the Arduino female connectors.



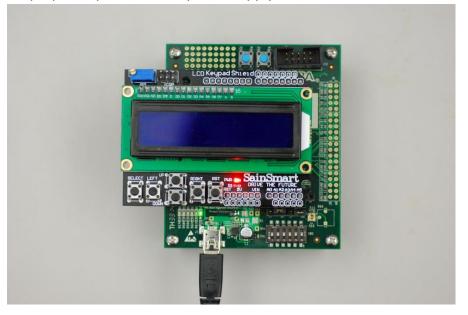
Top Arduino connector

Bottom Arduino connector



Make sure connection is snug.

4. Connect mini-USB cable to the board to power up, the other end to a USB connection on a laptop, computer, or USB power supply.



5. If not previously pre-programmed with the demo design POF, follow these instructions:

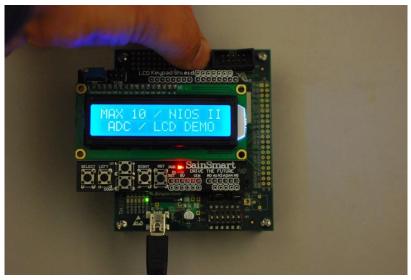
The design package includes a pre-compiled version of the design that downloads to the MAX 10 embedded flash. This version of the configuration includes the combined FPGA configuration and NIOS executable. Startup Quartus. Then start the programmer: Tools \rightarrow Programmer. Once the programmer starts, make sure the link is established to the kit:

Programmer - C:/Users/I	landis/Documents/ev	/alkit_lcd_controller/	'top - top - [Chain	2.cdf]						- 0	X
e Edit View Processi	ng Tools Window	Help 🐬						Sea	arch altera	.com	
Hardwise Setup US Enable real-time ISP to all		nming when available		> м	ode: JTAG	•	Progress:				
Start	File	Device	Checksum	Usercode	Program/ V Configure	erify Blank- Check	Examine	Security Bit	Erase	ISP CLAMP	
🕮 Stop											
Auto Detect											
💢 Delete											
Ъ Add File											
Change File											
Save File											
Add Device											
1 ¹⁰ Up											
Ju Down											
1 m Down											

Load the pre-canned file which includes both the FPGA configuration: add file → master_image/niosMAX10EvalKit_adc_lcd.pof. Then click start and you can run the demonstration. Note that by loading the .pof file in that directory, you have loaded a non-volatile image into the MAX 10 FPGA. You can also load .sof file in the same master_image directory for a volatile image of the same design (will be lost upon power down).

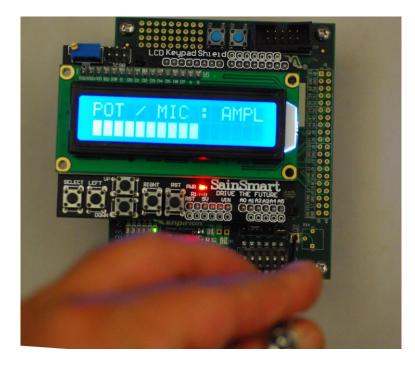
6. Click the top right blue push button (SW1) next to the JTAG header to begin operation of the design. You will not see anything on the LCD until you hit SW1 at least once, then you will see a Dallas Logic DSN splash screen, followed by the ADC/LCD demo title after another click. One more click and the design is in ADC measure mode.



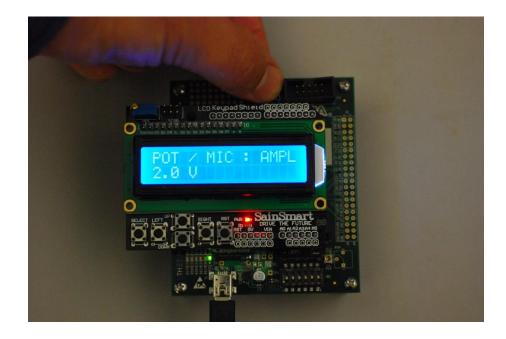




7. Use a small screwdriver or your fingers (if knob is attached to your POT) to turn the potentiometer and see the voltage meter change up or down. MAX 10 FPGA's ADC block input channel is measuring the analog voltage and using a Nios II system to display to the LCD. (Note: picture below is using a variable resistor R16, not the potentiometer recommended on R94.



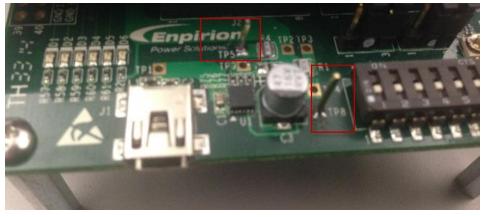
8. If you push the upper right push button SW1 one more time you can change the voltage meter scale to a numeric voltmeter. You can go back to meter mode by pushing the button again.



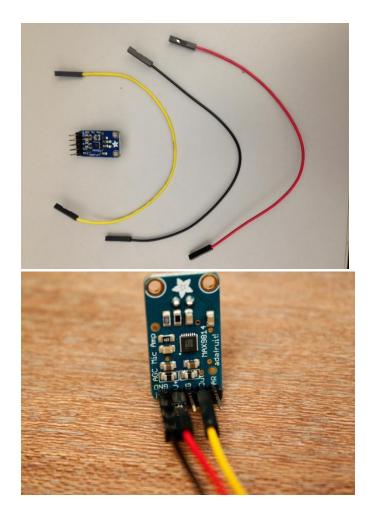
Optional Microphone Demo Setup

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

1. 2 male headers must be soldered to the TP5 and TP8 test point through-holes. These will provide power and GND to the microphone.

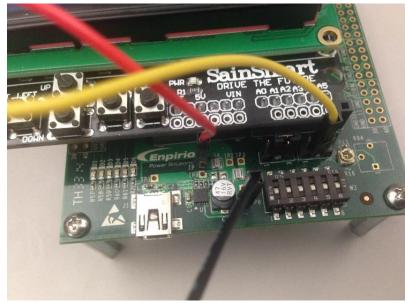


2. With an assembled Adafruit microphone (see Appendix), connect a cable to V+, GND, and Out on the adafruit PCB.



3. Connect the cable for V+ (red in the above image) to the TP5 test point, the GND cable to the TP8 test point, and the Out cable to the center of the J7 jumper header. You must remove the J7 jumper in order to connect the cable to the J7 center header. This is the same channel input as the POT into the MAX 10 ADC. Note that you must return the jumper

if you want the potentiometer to control the input voltage.



4. Power the board and run the ADC/LCD demo design in meter mode. For this demo you can blow into the microphone or it's good to play a song from your smart phone and put the microphone up to the speaker on your phone. A song with distinct beats will show well on the meter.

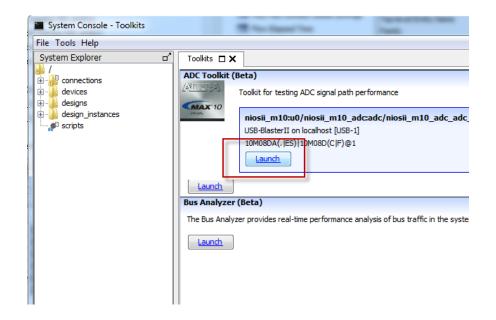
Optional Cool Microphone Soundwaves Demo

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.

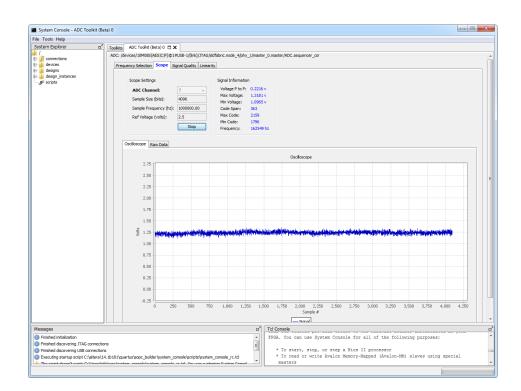
- 1. Download one of the following sound wave apps for your smart phone:
 - Android Sound Wave Generator (free) this one is untested
 - <u>iPhone Tone Generator (free)</u>
- 2. Launch the app on your phone and turn up sound on phone volume
- 3. Open the ADC Toolkit from System Console. Click on the Tools menu -> System Debugging Tools -> System Console

File Edit View Project Assignments Processing	Tool	s Window Help 💎												
C C C C C C C C C C C C C C C C C C C	<u>ک</u> او	Run Simulation Tool Launch Simulation Library Compiler Launch Design Space Explorer		> <			•	V	<u>e</u>	ð	Ŭ	N	2	9
Entity	Ō	TimeQuest Timing Analyzer												
A MAX 10: 10M08DAF484C8GES		Advisors •	-											
▷ 📴 adc_system 🛅	🧼 👘	Chip Planner Design Partition Planner Physical Design Planner Netlist Viewers												
		SignalTap II Logic Analyzer In-System Memory Content Editor Logic Analyzer Interface Editor In-System Sources and Probes Editor SignalProbe Pins Programmer JTAG Chain Debugger Fault Injection Debugger												
< III		System Debugging Tools		DI.	Syste	m Co	onso	le						
Hierarchy Files of Design Units T IF Status Module % Progress ()	a 4	IP Catalog Nios II Software Build Tools for Eclipse Qsys	3	3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Trans Bus A Extern	naly	zer ((Beta)		ce To	olkit		1	<u>A</u>
Full Compilation 100% 00:02:5	ø	Td Scripts												
Analysis & Synthesis 100% 00:01:2 Fitter 100% 00:01:0 Assembler 100% 00:00:0		Customize Options												
TimeQuest Timing Analyzer 100% 00:00:1		License Setup												

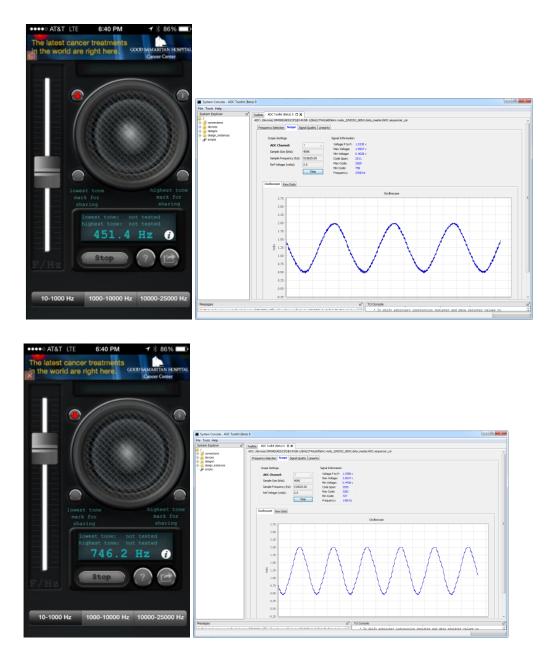
- 4. Click on Load Design... and click on the "master_image/niosMAX10EvalKit_adc_lcd.sof" file that came with this design. System Console does not configure the device using the SOF, but instead extracts debug file information from the SOF format.
- 5. After loading the SOF file into the system console, the JTAG connection to the toolkit becomes available, click on Launch shown in **red** below to open it.

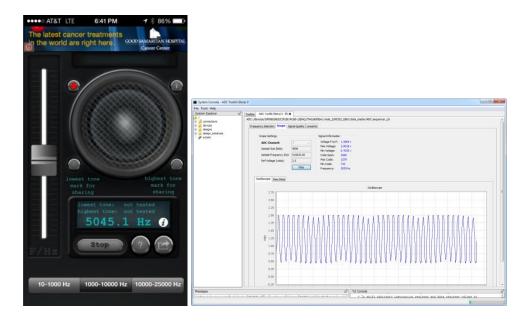


6. Select ADC Channel 7 from the scope view and click on "**Run**" to see the output of the microphone. It should contain a noisy output.

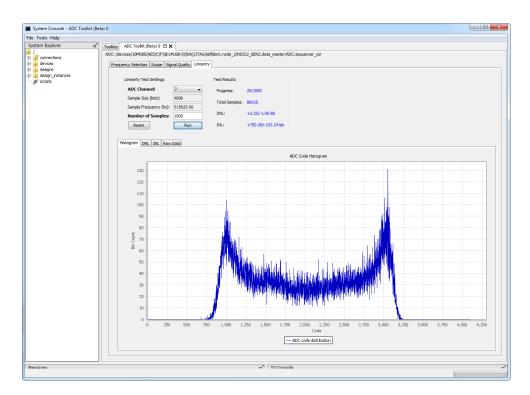


 Open your iPhone or Android tone generator App and hold the speaker up to the microphone. It's best to let the phone lie on the desktop next to the phone speaker source. Turn up the volume and run the tones from the app. 8. Sweep from low to high frequency on the application, you will see analog capture 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 evalkit_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. Alternatively, create a output_files/.pof file by selecting File → Convert Programming Files. Set the Mode from the default 1-bit Passive Serial to Internal Configuration. Highlight SOF data in the input files to convert field and you will see Add File become available to click. Click on add file and add the .sof file located under output_files/top.sof and click generate.
- v) You can download the volatile .sof or non-volatile .pof file.

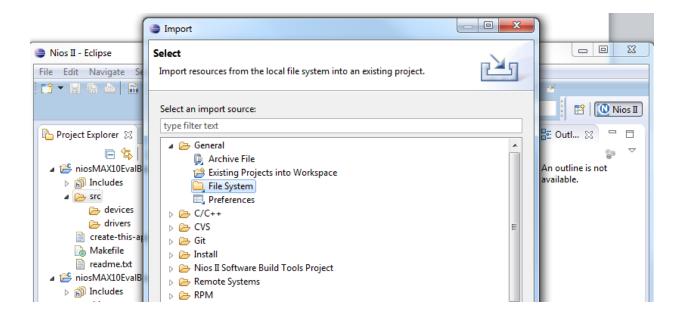
How to compile the software

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

Workspace	Launcher		•	22
Select a wor	kspace			
	s your projects in a folder called a workspace. orkspace folder to use for this session.			
Workspace:	C:\Users\enaswali\Documents\Max10\MAX10_EVAL_DEMO		•	Browse
🔲 Use this a	the default and do not ask again			
		C	ж	Cancel

- 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 platform/<project>.sopcinfo file and add it to the project. Provide a name for the software project i.e. niosMAX10EvalBoard and click Finish.
- 6. A BSP is created and located at niosMAX10EvalBoard_bsp and an application is created and located at niosMAX10EvalBoard. The application contains source code from the template design "Hello World Small" that needs to be replaced with source code from the MAX10 EVAL DEMO design, the source code is located at the software/src.
- 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 \rightarrow File System



Browse to the location of src, click on the box next to it, add it as shown and click Finish.

 Nios II - Eclipse File system Import resources from the local file system. 	23
	ios II
Includes Includes Incl	
▷ Includes ▷ drivers ▷ HAL ▷ alt_sys_init.c ▷ Inker.h ▷ system.h □ create-this-b: □ linker.x	
Makefile mem_init.mk public.mk / III / Cancel	□ * * * *

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

- 10. Compile the project by selecting Build All from the Projects Menu or alternatively Ctrl+B.
- 11. Program the FPGA device by launching Quartus II programmer from the menu and loading the file output_files/top.sof .
- 12. Load the executable to the processor by right clicking on the Application project and selecting Run-As Nios II Hardware.
- 13. 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.

Name: New_configuration						
🐏 Project 🎘 Target Conne	ection 3	🌾 Debugge	r 🛯 🧤 Sourc	e) 🗖 Co	mmon	
Connections						<u> </u>
Processors:						
Cable	Device	Device ID	Instanc	Name	Archite	Refresh Connectio
						Resolve Names
						System ID Propertie
Byte Stream Devices:						
Cable	Device	Device ID	Instanc	Name	Version	
Disable 'Nios II Console'						
☐ Ignore mismatched syste	m ID					
Ignore mismatched syste		ηp				
Download Reset system on download	ad					
						Apply Revert

14. 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

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 \rightarrow 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): ./platform/software/evalkit_lcd_adc_v14/mem_init/NIOSII_M10_ONCHIP_RAM_16K.hex

Next, you need to return to Qsys and change the ONCHIP_RAM_16K component by doubleclicking it. The initialize memory content and enable non-default initialization file needs to on. Enter the location where the .hex file is located.

	landis/platform_test/14.1/evalkit_lcd_controller/platform/NIOSII_M10.qsys) <@sj-slscf2680-02>
🙀 Parameters 🛛	· ····· · ···· · ····· · ····· · ······
System: NIOSII_M10 Path: ONCHIP_RAM_1	
On-Chip Memory (RAM or RO altera_avalon_onchip_memory2	DM)
▼ Memory type	
Type:	RAM (Writable)
Dual-port access	
Single clock operation	
Read During Write Mode:	DONT_CARE -
Block type:	AUTO V
▼ Size	
Data width:	32 💌
Total memory size:	16384 bytes
Minimize memory block usage (may ir	
* Read latency	
Slave s1 Latency.	1 💌
Slave s2 Latency:	
* ROM/RAM Memory Protection	
Reset Request:	Enabled 💌
▼ ECC Parameter	
Extend the data width to support ECC bit	5: Disabled 👻
Memory initialization	
Initialize memory content	
🗹 Enable non-default initialization file	
Type the filename (e.g: my_ram.he	x) or select the hex file using the file browser button.
User created initialization file:	/home/llandis/platform_test/14.1/evalkit_lcd_controller/platform/software/evalkit_lcd_adc_v14/mem_init/NIOSII_M10_ONCHIP_RAM_16K
🔄 Enable In-System Memory Content Ed	tor feature
Instance ID:	NONE
User is required to provide memory in	itialization files for memory.
<u></u>	

Next, click Generate HDL \rightarrow 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.

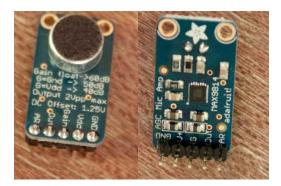
To generate a .pof file for non-volatile demonstrations, in Quartus run File \rightarrow Convert Programming Files. Change mode to Internal Configuration. Highlight SOF data, click add file and select output_files/top.sof.

Convert Programming <u>F</u> ile <u>T</u> ools <u>W</u> indow	g File - /home/llandis/platfo	orm_test/14.1/evalkit_lcd_	controller/top - top <@sj-slscf268	0-02> X Search altera.com
	convert and the type of progr file information from other fi		n setup information created here for	
	Open Conversion Setup Dat	a	Save C	onversion Setup
Output programming file	e			
Programming file type:	Programmer Object File (.p	oof)		_
Options/Boot info	Config <u>u</u> ration device:	PCE16	<u>M</u> ode:	Internal Configuration
File <u>n</u> ame:	output_files/output_file.pof			
Advanced	Remote/ <u>L</u> ocal update differ	ence file:	NE	
	Create Memory Map File	(Generate output_file.map))	
		te output_file.periph.pof ar	d output_file.core.rbf)	
	「 Create config data RPD (Generate output_file_auto.	rpd)	
Input files to convert—				
File/Data area	Properties	Start Address		Add He <u>x</u> Data
⊡- SOF Data	Page_0 10M08SAE144ES	<auto></auto>		Add <u>S</u> of Page
top.sor	1000034214423			Add File
				 Remove
				Up
				Down
				Properties
				<u>G</u> enerate Close Help

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 <u>http://dallaslogic.com/</u>.