

RA8875 Driver Board for Touch Display

Interfacing Adafruit's RA8875 TFT LCD driver board with an Altera DE2

Krista Oribello and Michael Federau

Preface

The purpose of this app note is to provide a tutorial on how to use the Adafruit RA8875 Driver Board and 800x480 TFT LCD with the DE2. These hardware components were used in the Tabletop Notifier Capstone project.

The DE2 uses SPI to connect and send commands to the RA8875 component. Adafruit provided a graphics library for the driver board to handle basic interfacing, drawing, writing text, and reading functionalities. These files, written for use with Arduino microcontrollers, have been ported for use with Nios II UCOS on the DE2.

The tutorial explains how to setup and use a volatile version of our Qsys and Quartus files which have been included into our LCD App Note package. The Eclipse software files that we have also provided are used to show the different capabilities of the LCD library.

Folder Organization

Create a new folder at the location you wish to work from. Label this folder “lcdtest”. Inside the “lcdtest” folder, create another folder called “software” (this is where your eclipse will work from later). Also inside “lcdtest”, copy the provided “LCD_App.vhd” and “nios2_ttn_vol.qsys” files.

Hardware

The RA8875 Adafruit LCD Controller requires 7 GPIO pins to operate correctly.

- 3.3V power (pin 28 on GPIO 1, with provided pin assignments file)
- GRN (pin 29 on GPIO 1, with provided pin assignments file)
- SPI Clock (pin 30 on GPIO 1, with provided pin assignments file)
- SPI SS (pin 31 on GPIO 1, with provided pin assignments file)
- SPI MOSI (pin 32 on GPIO 1, with provided pin assignments file)
- SPI MISO (pin 34 on GPIO 1, with provided pin assignments file)
- Reset (pin 36 on GPIO 1, with provided pin assignments file)

If you are using the Tabletop Notifier custom project board. Simply use a 40-pin ribbon cable to connect the DE2's GPIO header 1 to the header labeled “1” on the custom project board.

Startup

Quartus and Qsys

Begin by opening up Quartus 12.1sp1. Select create a new project wizard. For the project workspace, choose the previously created “lcdtest” folder. Name the project “LCD_App”. Include the “LCD_App.vhd” and “nios2_ttn_vol.qsys” files. Choose the Cyclone II device EP2C35F672C6 and click finish.

Select Assignments from the toolbar and import assignments. Import the LCD_App.qsf file provided. (NOTE: Quartus automatically creates a LCD_App.qsf file when you first build this project. This is NOT the same file. If you've imported the correct file it should tell you that ~400 pins were rewritten)

Next open the nios2_ttn_vol.qsys file. Here you can check the IP cores and configuration used. Switch to the generation tab and click Generate. Six or so warnings may appear, but this is okay.

After generation, go back to the Quartus window and double-click Compile Design in the Project Navigator. Once it has finished compiling, open Program Device. Add LCD_App.sof to the programmer if the file is not already there. Connect the DE2 and LCD driver board as outlined in the Quick Start hardware section. Connect the DE2 USB and turn on the power. If the LCD board has been properly connected to the DE2, the LCD should flicker for a brief moment. Start the programmer.

Eclipse

Next, go back to Quartus and under Tools, select Nios II Software Build Tools for Eclipse. Select the empty “software” folder you made earlier to use as your Eclipse workspace. Select New and create a Nios II Application and BSP from Template. In SOPC Information File name, open the nios2_ttn_vol.sopcinfo file that was generated by quartus. Name the project “lcd” and Finish.

Using Windows Explorer, open the software folder that is being used in Eclipse. Eclipse automatically provides a hello_ucosii.c user main file. This can be deleted. Copy all C and header software files from the LCD Appnote Files provided. These are: “Adafruit_LCD.c”, “Adafruit_LCD.h”, “glcdfont.c”, and “lcd_test.c”.

In Eclipse, refresh the project folder so the four files are shown. Build the project and run as Nios II Hardware configuration.

The program should be running through the different functions of the Adafruit_LCD library. Look at the comments within the lcd_test.c file to see what commands are required for initializing the LCD, drawing shapes, writing text, and playing with the touchscreen capabilities.

References

https://github.com/adafruit/Adafruit_RA8875

<https://github.com/adafruit/Adafruit-GFX-Library>

If changes need to be made in either Adafruit_LCD.c or Adafruit_LCD.h files, use the above links for reference.