ECE315 – Lab #3
Introduction to Analog-Digital Converters, Digital-Analog Converters and Photocell Light Sensors

Lab Dates, Report Dates and Demo Due Dates:

Please login to Moodle at eclass.srv.ualberta.ca to access all relevant lab dates. The lab schedule section contains the important dates for this semester.

Objectives:

- To introduce the students to the analog-digital converters on the MCF54415-based NetBurner boards.
- To introduce the students to the digital-analog converters on the MCF54415 –based NetBurner boards.
- To introduce the students to a simple sensor that detects light.
- To demonstrate context switching using specially crafted tasks.

Equipment, Parts, and Provided Software:

The following new equipment and software will be used in lab #3. This is in addition to the provided equipment, hardware, and software in labs #1 and #2.

<u>New Provided Equipment</u>

- 10-pin ribbon cable attached between the J2 header and the students' breadboard

<u>New Provided Software</u>

- AD class for controlling the on-board Analog-Digital Converters
- DA class for controlling the on-board Digital-Analog Converters.

<u>New Portion of Lab Kit for Lab #3</u>

- 16-33K Ohm Photocell

- 10K Ohm +/- 5% 0.25 Watt Resistor
- 10-pin header

Documentation:

Please login to the Moodle lab pages at eclass.srv.ualberta.ca to see a list of necessary documents for the ColdFire board used in ECE315. The documents are listed under the Lab Reference Materials heading.

ColdFire MCF 54415 Reference Manual chapters 29 and 30 contain information on the analog-digital converter module and the digital-analog converter module, respectively, of the ColdFire MCF 54415 microprocessor.

CDS Photoconductive Photocell Datasheet contains information on the light detecting photocell that we will be using to build our small light meter.

Introduction:

In lab #3, we will be working with the on-board analog-digital converters, the on-board digital-analog converters and one external component that detects light. The PDV-P8103 is a very simple component that senses light in the 400-700 nm range. The human visible range is 390 – 700 nm [1], so this sensor detects the same range we do.

Light Meter Application

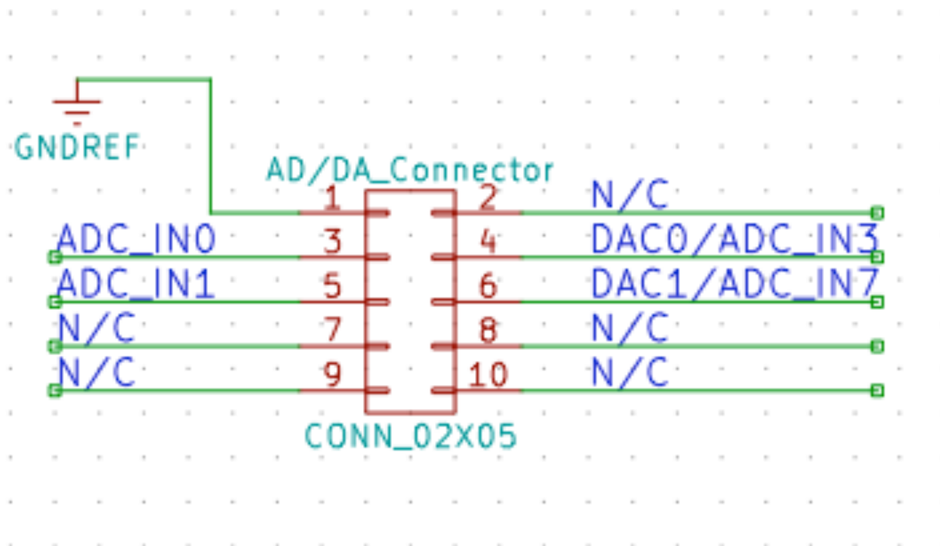
We'll use this sensor to create a simple light meter. Cupping the hand over the sensor or using a smartphone flash close to the sensor will vary the light that will be detected by the light sensor. Output from the light sensor will be connected to one of the analog-digital converter lines. The range of values will be displayed on the LCD from lab #1 in the form of a bar graph to create our light meter application. Given that our LCD can display twelve sprites per line we can divide up the total range of the light sensor into twelve ranges.

The bar graph on any of the 6 lines of the LCD should display a full line when the flash of your smartphone is used close to the photocell. The bar graph should display very little of the bar graph when your hand is cupped over the photocell.

The photocell can be viewed as a variable resistor. The value of the resistance varies based of the light detected and varies inversely to exposure to light.

Photocell Circuit Design

In order to design your circuit you'll need the photocell, the 10K Ohm resistor, and the ten-pin header in your kit. The 10-pin ribbon cable on the buffer board is your source for the AD and DA pins on the MOD54415 NetBurner board. The schematic for the pins is below. Please note that N/C means "No Connect" indicating that nothing is attached to that pin.



Consider your photocell as a variable resistor and create a voltage divider. Two analog-digital converters are brought out to the 10-pin ribbon cable. The two analog-digital converters are ADC_IN0 and ADC_IN1. Please use ADC_IN0. ADC_IN0 was verified on all stations prior to the start of the lab.

Using the Analog-Digital Converters

The code package for lab #3 contains a class called AD that interfaces the analog-digital converters (ADC) built into the board. The class initialization method is complete. The other methods in the class are to be completed by the student. Consult the code package for comments on which registers to modify. You'll be using the sim2 structures to modify the contents of the registers. This vendor-provided C structure allows us to change the contents of registers by treating them as a variable. To see the complete structure consult:
C: \nburn\MOD5441X\include\sim5441x.h

For example, to modify the contents of the status register of the ADC module we would use the following instruction:

```
sim2.adc.sr = 0xffff;
```

The status register of the ADC now contains the hex value 0xffff. The rest of the registers are modified in the same way.

Using the Digital-Analog Converters

The code package for lab #3 contains a class called DA that interfaces the analog-digital converters (DAC) built into the board. The class initialization method is complete. The other methods in the class are to be completed by the student. Once you have completed the DAC methods, you will be able to use DAC0 to output signals that can be viewed using our oscilloscopes. The goal is to view how long the while loop of each task takes. This portion of the lab is more experimental that developmental. Two dummy tasks that can be configured to take more processing time have been included in the code package to generate data for measurement on the oscilloscope.

Prelab:

1. Research and design a circuit that uses the photocell in your kit. You'll need the 10K Ohm resistor, the photocell and the 10-pin header in your kit to create a voltage divider circuit.
2. You are to arrive at the lab with the circuit pre-wired according to your design.
3. Skim Chapters 29 and 30 of the ColdFire Reference Manual for general information on the ADC and DAC modules of the ColdFire.

Exercise 1:

Goal: To build a new project for lab 3.

Steps:

1. Connect the 50-pin cable from the ColdFire microprocessor to your breadboard.
2. Connect the 10-pin cable from the ColdFire microprocessor to your breadboard.
3. Get the lab3.zip file from the eClass course website.
4. Create a new project called lab3 in your ECE 315 workspace. Follow the same procedure as you did in the tutorial.
5. Unzip and import all of the files into your project and delete the main.cpp file before building your project.
6. Run the project and check the heartbeat and context switching pins (J2[3]andJ2 [4]). Verify that the default webpage is being served.

7. We will be using the LCD for output in this lab. Please remember that the photocell, encoder, multiplexor, and the LCD use 3.3V. Using the wrong voltage may damage the board and components.

Exercise 2:

Goal: Complete the AD class at the recommended locations to read the data from the photocell.

Steps:

1. Before connecting up your photocell to the NetBurner board you should test your photocell circuit by hooking up the side of the photocell that has the varying voltage to the oscilloscope. You should see a voltage that varies between 0 and 3.3 volts maximum depending on light conditions. Place your hand over the photocell. The voltage should vary.
2. Once you have verified that your photocell works as expected, connect the varying voltage side of the photocell to ADC_IN0. This pin is buffered and is one of only two input pins on the ten-pin header that brings out the AD and DA pins from the MCF54415.
3. Complete the following AD methods:
   a. AD::StartAD
   b. AD::StopAD
   c. AD::ADDone
   d. AD::GetADResult
4. Complete the UserMain while loop to implement the following pseudocode

   StartAD
   BusyWait until done
   Read result
   StopAD
5. Display the values read in from AD to the serial console. The A/D is a 12-bit unsigned number. This is an intermediate step to verify the AD operation. Modifying the light levels over the photocell should modify the results read from the serial console.

Exercise 3:

Goal: Create the LCD code that displays the input from the photocell in bar-graph format to the LCD.

Steps:

1. Modify the LCD class to complete the LCD::DrawBarGraph method to display the value read in from the AD to the LCD.
2. The parameters are BYTE line (0-5) and BYTE length (0-11).
3. We can display 12 sprites on each line so divide the range of the AD by 12 to make maximum use of the LCD.
4. Once you have completed your modifications to the LCD::DrawBarGraph method you'll need to incorporate them into the UserMain while loop after the AD method calls.
5. Cupping you hand over the photocell should display 0 segments of the 12 possible segments. Using the flash of your smartphone should display 12 segments on the bar graph.

Exercise 4:

Goal: Create the DA code that outputs three different voltage levels corresponding to the three separate tasks in the code package. Take measurements of the time taken for each task.

Steps:

1. In order for the code in the two dummy tasks to be compiled in, we need to turn the optimization level in our project settings to none. With the current and default level of optimization our dummy code does not execute because it is optimized out.
    a. Right-click on your project folder and select Properties.
    b. Click on Settings and select Optimizations under the GNU C++ Compiler section of the Tool Settings tab.
    c. Change the optimization level to None.
2. Complete the DA::DACOutput method to send data out via the DAC0 pin. Currently, the suggested parameter is: float volts. You'll need to convert that to the 12-bit integer value that the DA expects.
3. Each task outputs a unique value at the beginning of its run loop. The UserMain task outputs 3.0 volts, Task1 outputs 2.0 volts and Task 1 outputs 1.0 volts. You may not see the full three-volt value when UserMain runs. That is acceptable as long as three distinct voltages are seen on the oscilloscope.
4. Measure the time taken for each of the three tasks and record the order of the tasks as they execute.
5. Modify the for loop constants in task 1 and task 2 by doubling each of them. Again, measure the time taken by each of the three tasks and record the order of the tasks as they execute.
6. Record these values in a table and include them in your report. Discuss the results versus expectations in your report.

Report Requirements:

- Your typed report is to briefly describe your design solution for the three lab exercises.
- Include your hardware design for the photocell
- Include the table entries for exercise four as well as the discussion on whether the results were as expected.
- Be sure to provide neatly formatted and well-commented code for all exercises.
- Briefly describe your test cases for verifying that each of your solutions worked properly. Describe which case was tested, as well as the anticipated and actual results for each case. Place this data in a table. **Do not write out your test cases. Please tabulate them instead.** Non-tabulated test cases will not receive full credit on the reports.

Consult the report writing guidelines if you have any questions regarding the report format. The report writing guidelines are available on the Moodle eClass page under Lab Reference Materials.

Marking Scheme:

Lab #3 is worth 20% of the final lab mark. Please view the lab 3 marking sheet to ensure that you have completed all of the requirements of the lab. The marking sheet also contains a limited test suite in the demo section. Please make use of it.

Report Submission:

Reports and code should be submitted via eClass by section. Please note that late reports and demos will not be accepted.

References:

[1] https://en.wikipedia.org/wiki/Visible_spectrum

[2] NetBurner Examples formed the basis for this lab.
C:\nburn\examples\MOD54415X\MOD5441x-ADC
C:\nburn\examples\MOD54415X\MOD5441x-DAC