EECE 344 Digital Systems Design

Lab 6: Basic Thermometer

1 Pre-Lab

1. Review the lecture notes about the concepts of ADC.
2. Read the textbook Sections 10.4 through 10.5 about the ADC module on Cortex-M4.
3. Find the device driver code you wrote from Lab 3 for the LCD.

2 Objective

The objective of this lab is to practice data acquisition by ADC on TM4C:

1. Structure your program to utilize your previous device driver code for the LCD.
2. Initialize a GPIO Port and use one of its pins as the analog input channel.
3. Construct an interfacing circuit to connect the LCD to the microcontroller.
4. Initialize and configure the ADC module on TM4C to measure the room temperature.

3 Requirements

Required parts: the Tiva LaunchPad for TM4C123, several jumpers, ADALM 2000 (as the +5V power source and oscilloscope), and a breadboard.

Work in pairs for this lab assignment.

4 System Requirements

One of the popular uses of the microcontroller is to build a data acquisition system. As discussed in the lecture, the microcontroller is able to acquire data by interfacing sensors which measure several ambient physical quantities. The sensors turn the physical quantities into the form of analog signal, or continuous voltage. Then after appropriate signal conditioning, the ADC should encode the analog signal into the digital signals for the microcontroller to process.

Recall that an ADC has the following characteristics:

1. Resolution
2. Reference voltage
3. Step size
4. Conversion time
5. Digital output
6. Parallel or serial output
7. Analog input channels
8. Start conversion and end of conversion signals

The following figure shows an example of a simple data acquisition system built with a microcontroller with sensors and a display. It is a humidity and thermometer combo where the temperature is shown in the unit of Celsius degrees.

In this lab, you will build a basic data acquisition system as a thermometer to monitor the ambient temperature continuously. Your thermometer is supposed to work as follows.

1. The microcontroller uses the integrated temperature sensor to measure the ambient temperature.
2. The measured temperature is encoded as digital outputs by the ADC on the microcontroller.
3. The microcontroller processes the digital output from the ADC to translate it into Celsius degrees.
4. The LCD that is interfaced to the microcontroller displays the Celsius degrees.
5. Wait for about 3 seconds and repeat the Steps 1 through 4 as long as the MCU is powered.

4.1 Design and Implement the Embedded Software

Remember to use **friendly code** in your configuration. We will write two groups of functions for our system. One for the ADC/GPIO and the other for the LCD. In fact, we can reuse all of the functions we had written for the LCD in one of our past labs, with one new addition. The new addition would be a function that sends a string of characters to the LCD. We can name this function `LCD_OutString( )`. You might also find the header file `stdio.h` and the function `sprintf` helpful.

For the GPIO and the ADC, follow the following steps:

1. Initialize GPIO Port E to use PE3 as AIN0:
    a. Activate the clock for PE;
    b. Correctly configure DEN, AMSEL, AFSEL.
2. Use the ADC0 module. Configure or use the following registers:
    a. RCGCADC
    b. ACTSS: use SS3 sequencer
    c. EMUX: use software trigger
    d. SSMUX3: get AIN0 as the input
    e. SSCTL3: measure the temperature
    f. PSSI
    g. RIS
    h. SSFIFO3
    i. ISC

In your `main` function, after all initialization is done, use a `while` loop to start the temperature measurement and continuously monitor the temperature. The measurement takes place approximately every 3 seconds. Each measured result is to be translated to some degrees in Celsius and displayed in the LCD. You can use a pseudo delay to get the 3 seconds.

4.2 Interfacing LCD

You can use the LCD in its 4-bit mode as you had done in the past labs. The LCD is to show the temperature to users. Power the LCD with your 5 Volts source.

5 Grading

The grading of your lab comprises the following components. Refer to the rubrics about the ranking of each component.

5.1 Performance of the Lab

Show the performance of your built system. Explain the interfacing circuit to your instructor.

5.2 Demonstration

When demonstrating the program, you are expected to explain each line of code if asked. Each student will be asked a different question to demonstrate his/her understanding of the project at hand.

Explain the interfacing circuit to your instructor. To show the functionality of the thermometer, when a user warms the Lauchpad, the LCD should show an increasing temperature reading.

5.3 Deliverables

Please indicate who your lab partner is in your deliverables. Submit your deliverables on Canvas to “Lab 6: Basic Thermometer” under “Assignments” by the due time and date. Refer to the schedule shown on Canvas. For this lab, your deliverables are:

1. Provide the flowchart you designed for your code. Use software to create the flowchart, as any hand-drawn flowchart will be classified as at best second ranking or lower according to the rubrics.
2. Your interfacing circuit drawn by a professional software, such as PCB Artist (available at https://www.4pcb.com/free-pcb-design-software.html). Any hand-drawn chart will be classified as at best second ranking or lower according to the rubrics.

5.4 Code

Submit your code on Canvas to “Lab 6: Basic Thermometer” under “Assignments” by the due time and date. Your project folder should contain all C code, compressed into a zip file. Include comments and indentation in your code. Points will be deducted for sloppy code lacking documentation.

5.5 Lab Report

You (as an individual student, not a group) may choose to write a lab report for this lab if you have not fulfilled the requirement of writing three lab reports. If you choose to write a report for Lab 6, submit your report on Canvas “Assignments” by the due time and date. Follow the instructions in the section on "Lab Report Writing Guidelines" in “EECE 344 Digital Systems Design Lab Policy.” In the body of your report, make sure you include the following:

1. The flowchart of the program you designed. You may use the flowchart you prepared for Deliverables (5.3 above). Use software to create the flowcharts, as any hand-drawn graph will be classified as at best second ranking or lower according to the rubrics.
2. Your interfacing circuit drawn by professional software. You may use the circuit you prepared for Deliverables (5.3 above). Any hand-drawn circuit will be classified as at best second ranking or lower according to the rubrics.
3. A photo of the circuit on the breadboard for the system.

Provide technical writing explaining your design of your code. Describe any challenges in your experiment and how you solved the problems. Draw a conclusion about designing and implementing this lab on the LaunchPad.