ECGR 4101/5101, Fall 2015: Lab 2

Using the Analog to Digital Converter

Version 1.0 – 9/8/2015

Learning Objectives:

This lab will show students how to utilize the analog to digital converter (ADC). In this lab, we will be connecting a potentiometer to an ADC input pin, reading the analog value, then expressing that voltage on an LED array bar.

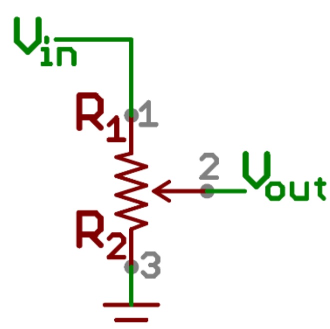
The Supply:

For this lab exercise, you will need a few additional components:

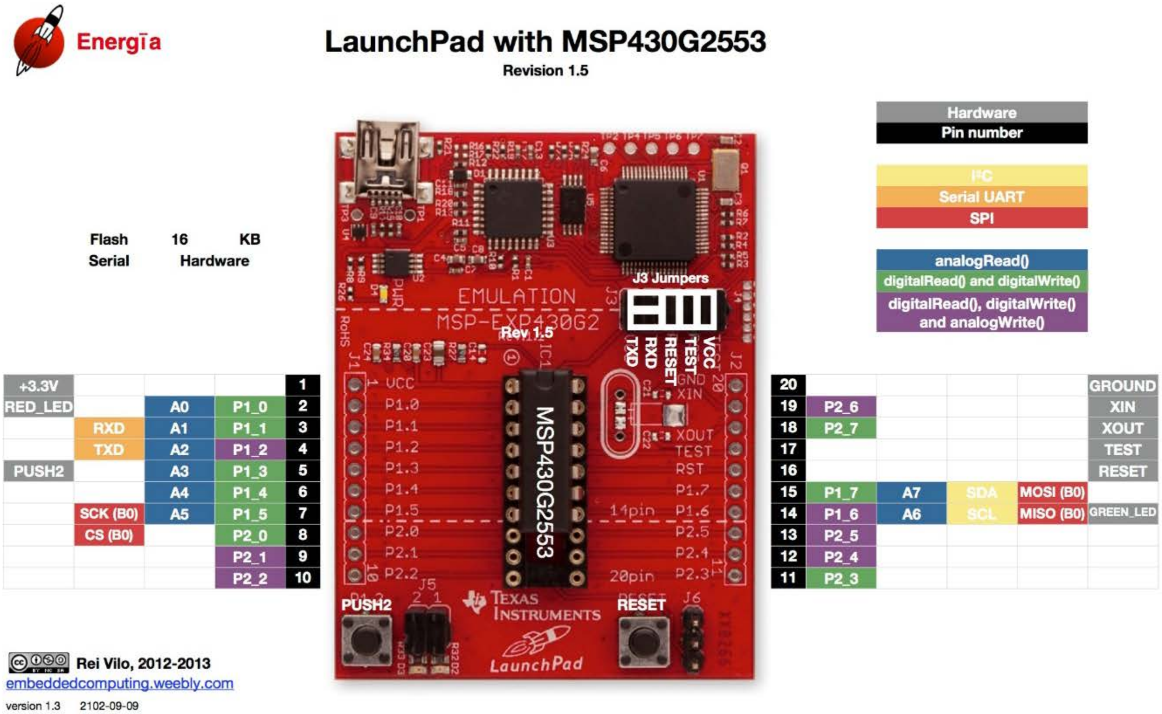
- Breadboard – You can pick this up at your local Radio Shack for about $5.00. I believe the campus bookstore also stocks these. Any size should do.
- LED Bar – You will get this from the T.A. during class.
- Resistors – You will need small, current-limiting resistors in series with the LED bar to prevent burning out the LEDs.
- Potentiometer – The parts drawers in EPIC 2148 should contain these. If you can't find any, see Eddie Hill (his office is directly across from Dr. Conrad's) to get some.
- Female Headers/Jumper Wires – You will need some method of connecting your board to a breadboard. You can pick up some female header pins from Eddie Hill as well to solder jumper wire too. **DO NOT SOLDER DIRECTLY TO THE LAUNCHPAD.**

Getting Started: Completing the Circuit

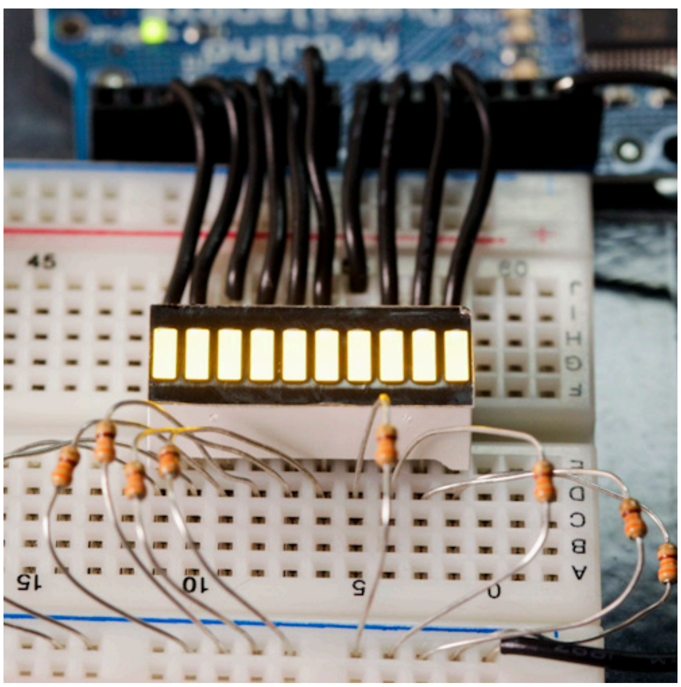
Before we get started coding in this lab, we are going to need to connect the additional hardware. We shall begin by connecting the potentiometer to one of the analog inputs on the board. Pin 2 on the board is designated as "A0", which means that it is an analog input pin attached to channel 0. I am going to reference this one in the rest of the lab, but feel free to choose whichever analog pin you want. The diagram below indicates w hich pins are analog inputs by the blue labels off to the side.



Potentiometers have 3 terminals: an input, divided voltage output, and … another output. Potentiometers also have a knob, that when turned, changes the amount of resistance between Vin and Vout. When we configure the potentiometer as shown to the right, it functions as a voltage divider. We will connect 3.3 Volts to Vin, connect Vout to our analog input pin (P2), and ground the last terminal. Now, as the potentiometer is turned, it will either approach 3.3 volts or 0 volts, depending on the direction and how you have oriented it in your circuit.



The LED bar is the next component to connect. It is simply an array of LEDs in a bar, so each one will need their own current limiting resistor. Each LED in the bar should be connected to one of the GPIO on the launchpad. You can choose whichever pins you would like. When you have it connected, your circuit should look something like the image below.



Again, for this lab we will be using the MSP430G2553. In Code Composer Studio, create a new project for this lab just like we did in the previous lab.

Analog Digital Converters: Living in an Analog World

The ADC converts analog signals into digital values. The digital value is determined by the reference voltage and the resolution of the ADC. The digital value is a scaled value from 0 to the reference voltage. By default, the reference voltage is usually the operating voltage of the microcontroller (3.3 Volts in our case). The resolution of the ADC is the number of bits we have available to represent this number, which for the G2553, is 10 bits. The more bits, the more accurately we can represent the analog value. The following formula where N is the number of bits, n is the converted digital value, Vin is the analog value being converted, and V+ref is the reference voltage.

$$n = \left[\frac{(V_{\text{in}})(2^N - 1)}{V_{+\text{ref}}} + \frac{1}{2}\right]_{\text{int}} \quad (\text{if } V_{-\text{ref}} = 0)$$

The ½ and []int indicates how the microcontroller does the rounding, as there can't be any decimal number in the solution. The int simply indicates the dropping of the decimal from the solution and + ½ accounts for the rounding.

Here are a few helpful links for using the ADC:
http://coder-tronics.com/msp430-adc-tutorial/
http://www.ti.com/lit/ug/slau144j/slau144j.pdf - See Pages 536-558

Controlling the LED Bar

The LED Bar functions just like a normal LED. If you have it connected properly, you will only need to write a 1 to each pin's output register to turn the LED on. There are 10 LEDs available on the LED Bar, so we will need to scale the ADC reading to evenly display across the 10 LEDs. For example, when the voltage read is 0 volts, all of the LEDs are off; when the voltage is 3.3 volts, all of the LEDs are on; and when the voltage is 1.65 volts, half of the LEDs are on. That way, as you turn the knob on the potentiometer, the LED bar goes up and down accordingly.

It would be a good idea to create a function to drive the led bar, in which a value, 0 – 10, can be passed to the bar to turn on the LEDs up to the specified number.

Requirements:

- The code must be written in C using Code Composer Studio
- The MSP430 reads analog voltages from a potentiometer
- The LED Bar Properly displays the scaled voltage representation

To Demo and Submit:

To submit, have the demonstration sheet below printed off. Demonstrate your working (or partially working) code to the TA. After grading, the TA will take your demonstration sheet and save for grading.

Embedded Systems Lab Demonstration Validation Sheet

This sheet should be modified by the student to reflect the current lab assignment being demonstrated

Lab Number:	Lab 2 – ADC
Team Members	Team Member 1: Team Member 2:
Date:	

Lab Requirements

REQ Number	Objective	Self-Review	TA Review
1	ADC Voltages are Properly Read		
2	The LED Bar lights up and can be controlled properly		
3	The ADC Value is scaled properly and displayed on the LED Bar		