

ECGR 4101/5101, Fall 2018: Lab 2

Using the Analog to Digital Converter

Version 1.0 – 9/6/2018

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:

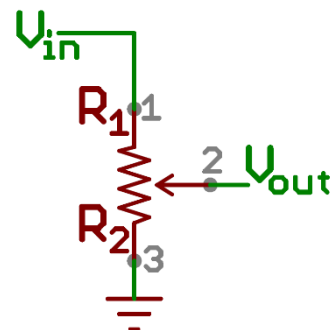
For this lab exercise, you will need a few additional components:

- 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. You will use this for PROTOTYPING your circuit before soldering.
- LED Bar (and 20-pin DIP socket)– You will get this from the instructor during class.
- Resistors – You will need small, current-limiting resistors in series with the LED bar to prevent burning out the LEDs. You will get this from the instructor during class.
- Potentiometer – You will get this from the instructor during class.
- Photo Resistor – You will get this from the instructor during class.
- Perfboard – You will get this from the instructor during class.
- Female Headers, 10-pin – You will get this from the instructor during class.
- 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. The diagram below indicates which 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 V_{in} and V_{out} . When we configure the potentiometer as shown to the right, it functions as a voltage divider. We will connect 3.3 Volts to V_{in} , connect V_{out} 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 photo resistor should be connected to A1.



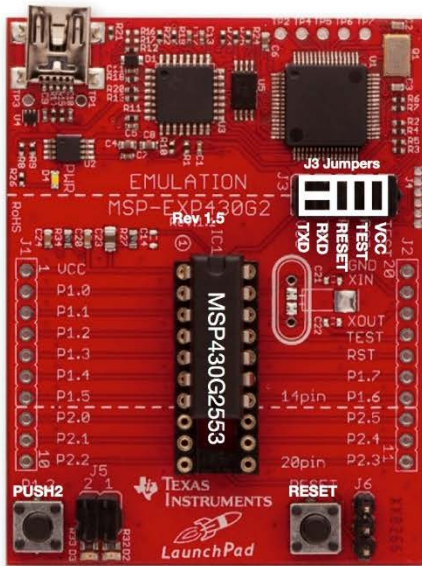
LaunchPad with MSP430G2553

Revision 1.5

Flash 16 KB
Serial Hardware

+3.3V				1
RED_LED		A0	P1_0	2
	RXD	A1	P1_1	3
	TXD	A2	P1_2	4
PUSH2		A3	P1_3	5
		A4	P1_4	6
	SCK (B0)	A5	P1_5	7
	CS (B0)		P2_0	8
			P2_1	9
			P2_2	10

Rei Vilo, 2012-2013
embeddedcomputing.weebly.com
 version 1.3 2102-09-09



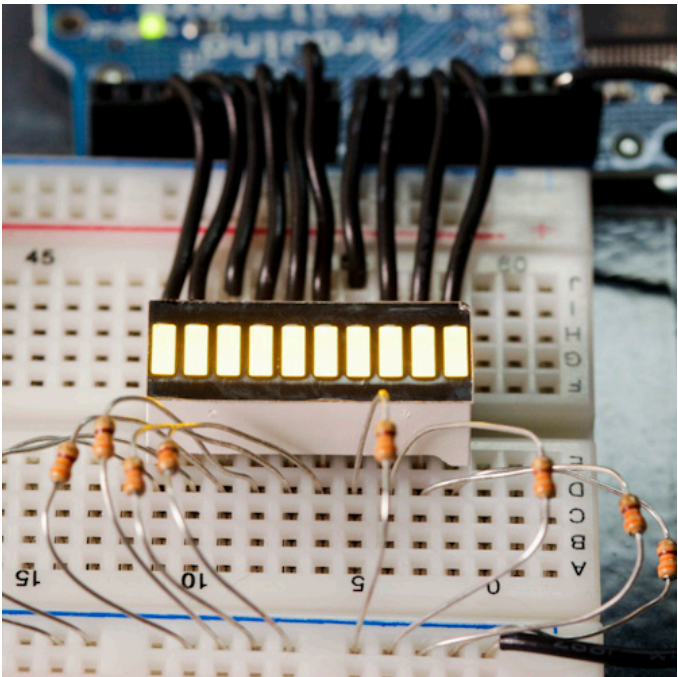
Hardware
Pin number
IFC
Serial UART
SPI
analogRead()
digitalRead() and digitalWrite()
digitalRead(), digitalWrite() and analogWrite()

20				GROUND
19	P2_6			XIN
18	P2_7			XOUT
17				TEST
16				RESET
15	P1_7	A7	SDA	MOSI (B0)
14	P1_6	A6	SCL	MISO (B0)
13	P2_5			GREEN_LED
12	P2_4			
11	P2_3			

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 should use the following pins, from Most significant bit to least significant bit:

P1.5 P1.4 P2.7 P2.6 P2.5 P2.4 P2.3 P2.2 P2.1 P2.0

When you have it connected on the breadboard for prototyping, , 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, V_{in} is the analog value being converted, and V_{+ref} is the reference voltage.

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

The $\frac{1}{2}$ 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 $+\frac{1}{2}$ 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, only the right-most LED is on; when the voltage is 3.3 volts, only the left-most LED is on; and when the voltage is 1.65 volts, only one of the center LEDs is on. That way, as you turn the knob on the potentiometer, a single LED on the bar goes up and down accordingly. Avoid oscillations (flashing between two LEDs).

The same type of functionality should be done for the photo resistor: full light will be one extreme of LEDs lit versus full dark.

It would be a good idea to create a function to drive the led bar, in which a value, 0 – 9, can be passed to the bar to turn on the LEDs up to the specified number.

Perfboard:

Part of this exercise will be to solder a board with the components. Make sure you do NOT solder the LED bar directly to the board – solder the socket to the board.

Requirements:

- The code must be written in C using Code Composer Studio
- The MSP430 reads analog voltages from a potentiometer
- The MSP430 reads analog voltages from a photo resistor
- The input displaced is selected based on toggling the SW2.
- The LED Bar Properly displays the scaled voltage representation (one LED at a time)
- The soldering work should be neat and clean.

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 soldered board 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 from POT		
2	ADC voltages are properly read from photo resistor		
3	Input voltage to display is toggled by pressing the SW2		
4	The LED Bar lights up and can be controlled properly		
5	The ADC Value is scaled properly and displayed on the LED Bar (one LED at a time, with no oscillations)		
6	Grade will also be determined by the quality of the board soldering work (turn in the board in the bag, stapled to the check-off sheet).		