California State University Chico
December 17, 2018

Position Measurement System

Moses Peace McCabe
Group Member: Hugo Romero

I – Introduction

The objective for this lab is to develop a means for a digital computer to sense its analog world. This is accomplished by presenting a technique for the system to measure analog inputs using an analog to digital converter (ADC). The system uses the ADC to collect information (data) from the external world. The input (data) signal is an analog voltage, and the system output is a binary number. To collect data from the external world into the computer a sensor must be used to collect the data and it must be converted from analog into digital form. This lab used a potentiometer sensor to read the potential difference between two distances.

II – Body

This lab begins be following the published procedure from the Lab Report 6 instruction manual which was divided into three sessions. For the first sessions of this lab, the potentiometer was set up. This was accomplished by soldering three wires onto the potentiometer terminals which makes it suitable for connecting to a breadboard. A copy of a ruler was then added to the side of the potentiometer to help determine its distance. A wire wrapped around the potentiometer slider to be used as a pointer which pointed to the ruler. The first session was completed by connecting the Potentiometer to the microcontroller Ground pin, Voltage pin, and PE2.

The second sessions were focused on writing two software to configure the ADC. For the first software, the following steps were performed to initialize the ADC. The first five step configured PortE pin 2 (PE2) as an input pin. These are accomplished by activating the clock on Port E, making PE an input by writing zero to the DIR register, enabling alternate function on PE2 by writing one to AFSEL register, enabling analog function on PE2 by writing one to AMSEL register, and disabling digital I/O on PE2 by writing zero to the DEN register. The next sequence of steps is specific to ADC. This steps include turning on the ACD clock by setting bit 16 to the SYSTCTL_RCGCO_R register, configuring the maximum sampling rate of the ADC by setting bit 8 or 9 of the SYSTCRT_RCGCO register, setting the priority of sequencer three, disable sequencer three by writing a zero to bit 3 of the ADC_ACTSS register (this prevents erroneous execution if a trigger event were to occur during configuration), configure the sampling sequencer (sequencer 3) to software trigger, configure the corresponding input sequence in the ADCSSMUXn register, configure the sample control bits in the corresponding nibble in the ADC0SSCTLn register, and enable sequencer three by writing one to bit 3 of the ADC_ACTSS register.
The second software in this sequences performs the ADC conversion. This was accomplished in four steps; first, the ADC is started using a software trigger. Second, a function waits for the ADE to complete by polling the RIS register bit 3. Third, the ADC read the scanner (sensor) which is the 12-bit sample read out of sequencer three. The final step, the RIS bit is clear by writing to the ISC register.

Once ADC is initialized and set up to scan the sensor (read input its analog world), the final step is to create a control system that converted the analog input into a digital signal. This control system is the main of the program. Its goals are to initialize SysTick and set-up SysTick handler to read ADC input data at a specific rate, and to convert the analog signal into digital signal. SysTick was configured to interrupt the system at a frequency of 40 Hz. This logic is due to the facts that the output of the conversion was a one-digit number at one point in time, and there is a finite time in between conversion. Using SysTick period interrupt, then this time become the

time between SysTick interrupt. In other words, this forces the sampling system to run at 40 Hz. The SysTick Handle manage two tasks. The first task is to read the ADC analog signal into a global variable call ‘Data’. And the second is to set another global variable call ‘Flag’ to one, which indicates an interrupt has occurred and analog data has been scanned. Once the global variable ‘Flag’ is set to one, the conversion function is called. This function turned the analog signal into 8-bits digit fixed point number by using the equation shown in figure 1. The Conversion Function result is an integer number. Which is later converted and displayed as a decimal number using the Nokia LCD display.

- ## Fixed Point Conversion Equation

  Fixed Point equation used in program to converted analog to digital signal

$$\text{FIXED POINT POSITION} = \big((0.4196 \times 2^8)(\text{ADC}) + (20.515 \times 2^8) + 128\big)/(256)$$

Figure 1

- ## Circuit Image

  Below is the image of the circuit build during this lab to test the software. Which consists of a microcontroller, a Nokia LCD Displace, and a potentiometer

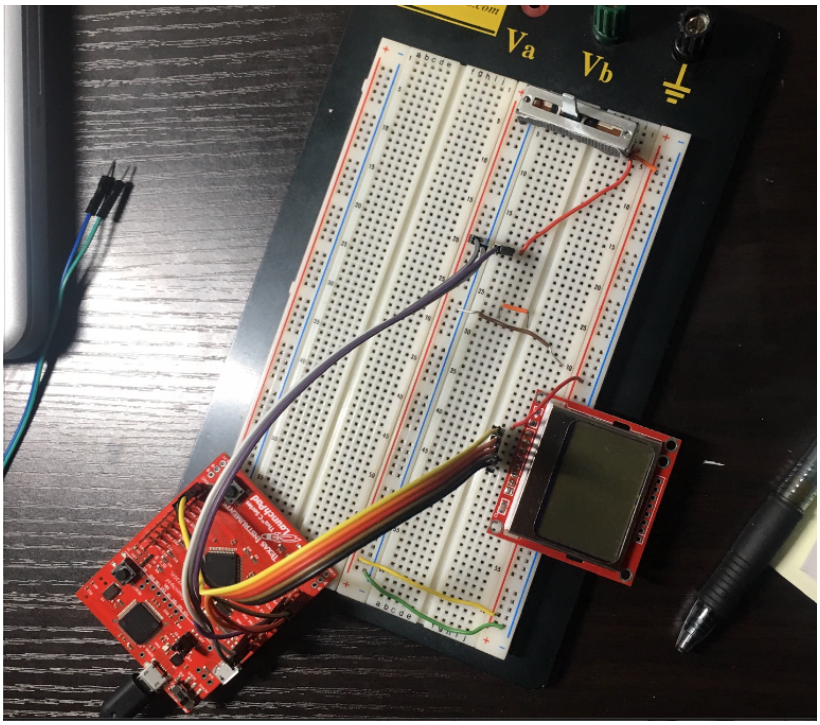
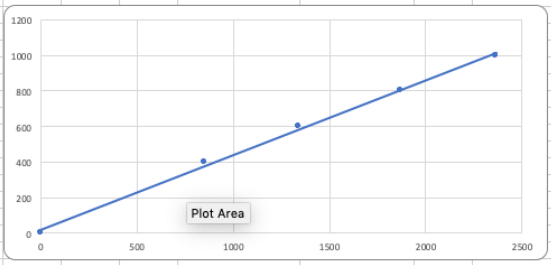


Figure 2

- ## Calibration Table and Graph

The figure below shows the table and graph used to calibrate the potentiometer and develop the ADC equation in figure 1.

Ax	Value			Fixed Distance	Value (ADC)	Measured Fixed Point			X (2^8)	Y (2^8)	Distance comp
0	0			0	0	20.515			107.418	5251.84	21.015
0.4	852			400	852	378.0142			107.418	5251.84	378.5142
0.6	1339			600	1339	582.3594			107.418	5251.84	582.8594
0.8	1872			800	1872	806.0062			107.418	5251.84	806.5062
1	2366			1000	2366	1013.2886			107.418	5251.84	1013.7886
1.2	2802			1200	2802	1196.2342			107.418	5251.84	1196.7342
1.4	3584			1400	3584	1524.3614			107.418	5251.84	1524.8614
2	4095			2000	4095	1738.777			107.418	5251.84	1739.277



0.4196x + 20.515 = graph equation

X = 0.4196(2^8)

Y = 20.515(2^8)

21.015

Figure 3

- ## Logic Analyzer

The figure below shows a screenshot of the logic analyzer, which shows how the output of the ADC samples were changing with different input for the potentiometer.

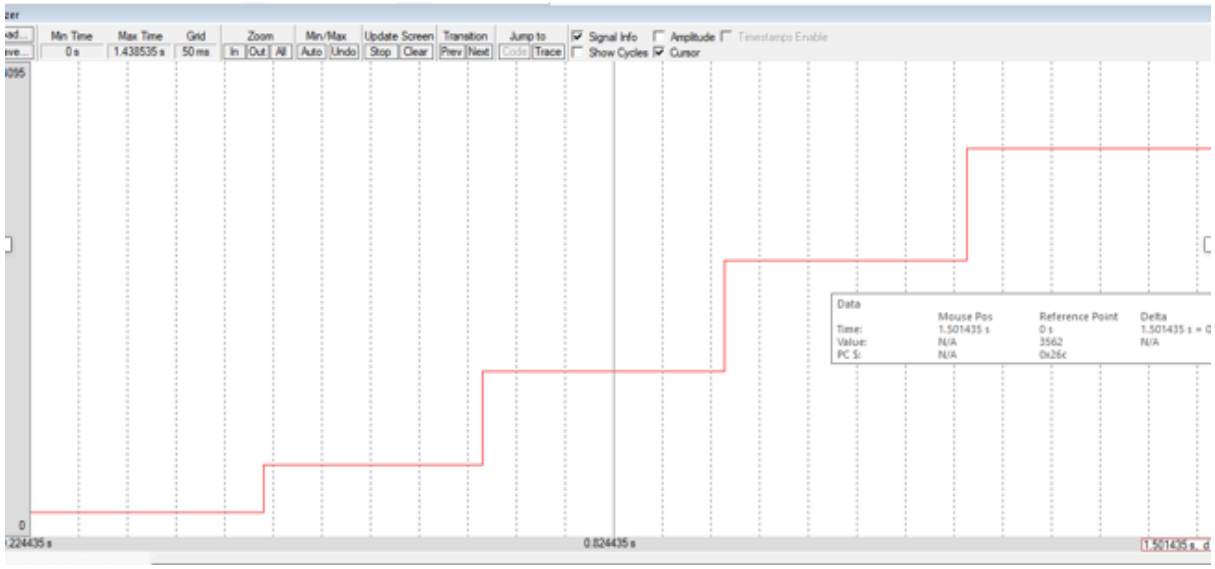


Figure 4

III – Discussion/Conclusion

The objective of this lab was accomplished is two form. The first objective is to learn how to perform fixed-point format calculations, and the second objective is to combine previously acquired knowledge such as SysTick, Interrupts, etc…, and newly acquired knowledge such as ADC to develop a means for a digital computer to read analog signal for its outside world.

The difficult part of this experiment was to carefully calibrate the potentiometer. The ruler on the potentiometer was sometimes hard to read due to its size, which led to an error in the collected data. Difficulty in differentiating the exact location of the Potentiometer pointer on the ruler led to an incorrect location reading for some input voltages.

The second problematic potion of the experiment is due to reading the data from the outside world. How the data is stored in the software account for its accuracy during conversion. For example, storing the data in an unsigned short (2 byte) variable, lead to inaccurate converted output which is due to the size of data being converted.

Overall, this was a great and challenging experiment with a very detail procedure. The experiment emphasizes the important of ADC and the role embedded system plays in our world.