Lab 5:
Microcontroller Interfacing to Analog Sensor

Riley Myers, Caitlin Armstrong

September 10, 2019

1 Introduction

There are many different kinds of sensors that microcontrollers can interface to. Some sensors may provide only analog signals while the microcontroller is inherently a digital device and requires a special method to process these signals. Analog-to-digital converters are the means by which a signal is converted from a analog signal to a digital signal, and are often included inside a microcontroller. The process for converting a signal includes sampling, conversion and encoding of the result into a digital value. In this lab you will use the ADC module of the ATmega1284 to convert an analog signal to digital and transmit the result to the computer using serial communication. The particular sensor used in this lab is a 3-axis accelerometer.

2 Lab

This lab involves the following parts:

1. Creating code that reads values from the analog-to-digital converter on the microcontroller using **timers and interrupts**
2. Connecting the microcontroller to an external analog accelerometer
3. Using the microcontroller to translate the results from all 3 axes to the computer using serial communication
4. Recording the data and using MATLAB to convert the raw measurements to specific force values of units m/s^2

2.1 Analog to Digital Converter

The ATmega1284 has an internal 10-bit successive-approximation register analog-to-digital converter(ADC). Successive approximation register ADCs contain a sampling mechanism, a reference voltage source and an analog comparator. They function by sampling the input voltage and setting the reference source to half of its maximum voltage. Both the reference and the sampled voltage are fed into the comparator and the result is measured. If the sampled voltage is higher than the reference, then the most significant bit of the output is set. Then, the reference is set to half the previous value, offset by the halfway voltage if the previous bit is set. Then the result of the comparison is used to determine the next bit in the output. This process continues until all the bits in the output have been determined.

The ADC can be configured to acquire a single sample on demand or to be automatically triggered using an a signal source, e.g., a timer. Using a timer, the data acquisition can be obtained at a fixed interval, i.e. a fixed sampling rate. The ADC can also be configured to trigger an interrupt when a conversion is finished. Other settings that will need to be configured are the input channels, the analog reference voltage, and the ADC clock speed.

2.2 Analog Accelerometer

The analog accelerometer used in this lab is the Analog Devices ADXL335. This is a 3-axis accelerometer with ± 3g sensitivity. An accelerometer measures the non-gravitational force per unit mass. When stationary, the specific force is referred to as acceleration due to gravity.

3 Lab Assignment

1. Demonstrate that you are reading values from the accelerometer and transmitting them over UART. You need to use timers and interrupts, and clearly indicate the sampling rate of your system. (**All code must be either written in C and/or Assembly and no Arduino Sketch will be allowed**).
2. Confirm that the values read from the accelerometer seem reasonable.
3. Plot the accelerometer values using MATLAB, **in real-time**.

4 Questions to Consider

1. Given the ATmega128 microcontroller, what is the maximum sampling rate for a single channel?
2. Since three channels are used to process the accelerometer data, what is the maximum sampling rate you can use?
3. Based on your answers above, what is the maximum frequency that you can correctly manipulate using your current microcontroller?
4. What system limitation do you encounter that prevent you from increasing your sampling rate? What could you change to improve this?