EE 308 – LAB 9

The 68HCS12 Analog to Digital Converter

The analog to digital converter is described in the *ATD Block Users Guide*. The HCS12 has two 8-channel 10-bit A/D converts, ATD0 and ATD1. For simplicity this lab will work with ATD1 only. (Ports AD0 and AD1, part of ATD0, are used by DBug-12 at startup to determine whether to execute DBug12, or to run code from EEPROM or the bootlaoder. Thus, by using ATD1, we will not accidently have the HCS12 run from EEPROM or the bootloader.) ATD1 can have up to eight inputs on pins PAD8 through PAD15.

The A/D converter also uses two dedicated pins V_{RH} and V_{RL} for high and low voltage references respectively. On your HCS12 board, V_{RH} is connected to V_{CC}, and V_{RL} is connected to GND. When the HCS12 is set up to do 10-bit conversions, an input voltage of V_{RL} gives an output of `0x000`, and an input of V_{RH} gives an output of `0x3FF`. (This assumes that the DJM bit of ATD1CTL5 register is set, so that the data is right-justified in the results registers.) If we measure a voltage between V_{RL} and V_{RH}, we can compute the value by simple ratios

$$voltage = \frac{measurement * (V_{RH} - V_{RL})}{1024} + V_{RL}$$

For example, if $V_{RH} = 5$ volts, and $V_{RL} = 0$ volts, and the measurement is 0x2B0 (688_{10}), then the measured voltage is

$$\frac{688 * (5 - 0)}{1024} + 0 = 3.359 \text{ volts.}$$

To make sure that you do not damage the A/D converter on your HCS12 you should be sure to do the following:

- You should have a 1 to 10 KΩ resistor in series with the A/D input pin to prevent damage if your input voltage rises above V_{RH}.
- You should never allow the voltage at the A/D pin to go below V_{RL}.

1. Write a program which uses the RTI interrupt. The RTI should generate an interrupt every 64 ms. The RTI interrupt service routine should set a flag which tells the main program the interrupt has occurred. In your RTI interrupt service routine write the 10-bit result of the conversion on Port PAD8 to your LEDs (connected to Ports A and B). In the main program, write the full 10-bit value to the terminal using the `printf()` function.
2. Connect a voltage from a pot to PAD8, so you can vary PAD8 from 0 to 5 Volts. Compare the value displayed on the LEDs with multimeter measurements for several different input voltages.
3. The A/D conversion measurements can be improved by averaging the values in the registers ATD1DR0 through ATD1DR7. In your RTI routine, average the 8 values. Display the

averaged 10-bit value on the LEDs. In the main program, write the averaged 10-bit value to the terminal. Is the value more stable than it was when you displayed the unaveraged value?

Pre-Lab

1. Write the program for Part 1 of Lab 9.
2. Calculate the expected 10-bit result when the voltage input to PAD8 is:
   (a) 0.0 V
   (b) 1.0 V
   (c) 2.0 V
   (d) 2.5 V
   (e) 3.0 V
   (f) 4.0 V
   (g) 5.0 V