EE 308 – LAB 9

The 68HC12 Analog to Digital Converter

The analog to digital converter is described in Section 15 of the HC12 reference manual. The analog to digital converter can have up to eight inputs on pins Port AD0 through Port AD7. (Ports AD0 and AD1 are used by DBug-12 at startup to determine whether to execute DBug12 or run code from EEPROM or the bootlaoder. You should not use Ports AD0 or AD1 as A/D inputs unless all other six inputs are being used.) The A/D converter also uses two dedicated pins V_{RH} and V_{RL} for high and low voltage references respectively. On your HC12 board, V_{RH} is connected to V_{CC}, and V_{RL} is connected to GND. When the HC12 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`. (Remember that in 10-bit mode, the 10-bit result is left-justified in the 16-bit result registers. The value of `0x3FF` assumes we have shifted the result register 6 bits to the right to right-justify the result.) 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, 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 HC12 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 8 MSB of the conversion on Port PAD3 to your LEDs (connected to Port A or B). In the main program, write the full 10-bit value to the terminal using the D-Bug12 `printf()` function.
2. Connect a voltage from a pot to PAD3, so you can vary PAD3 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 ADR0 through ADR7. In your RTI routine, average the 8 values. Display the 8 most significant bits of the average 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?