ECE263

Lab 8

Spring 2013

EXTRA CREDIT

Sawtooth Waveform Generator
A/D, D/A & SPI

Revised: 22 Apr 2013 DAL

(intentionally left blank)

Object:

In this lab you will generate an analog sawtooth waveform by using three I/O devices internal to the MC9S12DP256 microcontroller and one external I/O device. You will use the Agilent Logic Analyzer and the Rigol Oscilloscope to verify the correct operation of your programs.

Material:

Your choice.

Background

One unfortunate fact for digital designers is that the world is analog. Nothing is clear cut, true-false, black-white, up-down, instead everything varies over a range of values. In order for engineers to measure or control things in the real world, they have to use sensors and controllers that can convert from analog to digital or digital to analog. The type of sensors used most commonly output voltage, current or frequency.

The MC9S12DP256 Microcontroller contains two eight channel analog to digital converters, referred to as either ADCs or ATDs, that can be used to convert analog voltages to get either 8-bit or 10-bit binary numbers as a result. The detailed operation of the ADC function is described in the *S12ATD10B8CV2* manual. The ADC has two reference voltages, V_L and V_H that set the allowed voltage range. The voltages to be measured are compared to this range and a binary weighted result is obtained.

In order to demonstrate the capabilities of the analog to digital converter, the Dragon 12-Plus board has a single turn, 50K potentiometer attached to channel PAD07 of ATD0 of the MC9S12 as shown in Figure 1. One end of the potentiometer is attached to V_H and the other end is attached to GND (which is also V_L). The wiper of the pot is attached PAD07 and can generate any voltage over the range of the ADC.

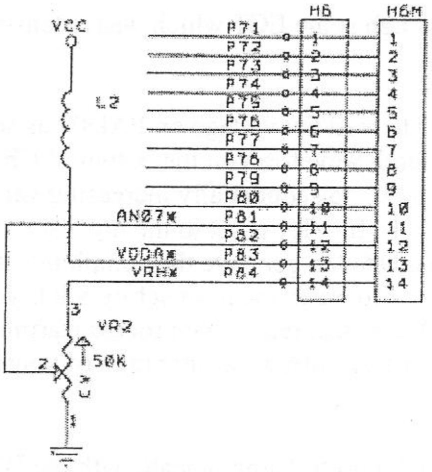

Figure 1: Trimmer Pot VR1

The corollary to an analog to digital converter is a digital to analog converter (DAC). A DAC accepts a binary weighted number as an input and then generates an output voltage that is proportional to high and low reference points similar to the way an ADC measures a voltage. The MC9S12 does not have an internal DAC so the Dragon 12Plus board has added an external, two channel, 10-bit DAC: the Linear Technology LTC1661. Like the ADC, the output range of the DAC has been set to be 0V to V_H (equal to V_{CC}) The detailed operation of the DAC is described in the *LTC1661 Specification.*

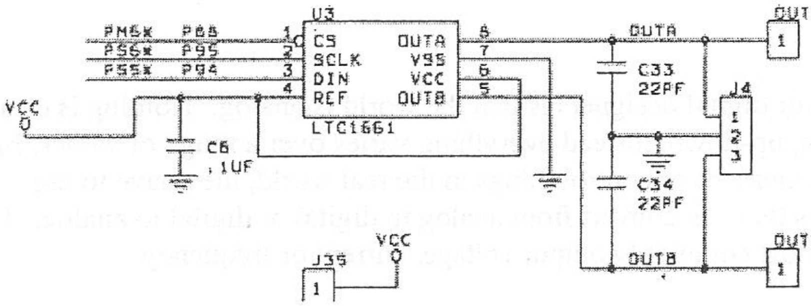

Figure 2: LTC1661 Digital to Analog Converter

The digital interface to the LTC1661 is via a serial peripheral interface (SPI) bus. The SPI is a two way serial communication port that can both send and receive data synchronously using a common clock generated by a bus master. The SPI has two data lines, master out slave in (MOSI) and master in slave out (MISO), a serial clock (SCLK), and 1 to n chip select lines. The detailed operation of the SPI function is described in the *S12SPIV2* manual. There are three identical SPI ports in the MC9S12DP256 and the DAC is attached to SPI0 which shares four the pins with Port S. The chip select for the DAC is connected to an I/O port pit, specifically Port M bit 6.

As shown in Figure 2 above, the LTC1661 DAC is an "output only" device therefore the MISO is not used and the MC9S12 must ignore any data transferred in on MISO during a I/O transfer.

The final peripheral to be used for this Lab is the ECT which was extensively covered in Lab 6.

Design Description

For this lab you will first use the ADC to read the voltage on PAD07 as set by the trimmer pot (VR1). You will then generate a sawtooth waveform on the output of Channel A (J4 pin 1) of the DAC by setting the voltage out of the DAC to sequentially increasing values starting from $000 and ending at the voltage read from VR1. If VR1 is set to the highest voltage, i.e. the full range, it would take 1024 ($3FF) writes to the DAC to generate the complete output waveform. Once you have done enough DAC conversions to reach the limit set by VR1, you will take a new measurement of the voltage from VR1 and then repeat the process starting at 0V. Note that the upper end point will be dynamically shifting during the operation of your program in response to the user changing the position of the pot.

You will need to do transfers on the SPI port to communicate with the DAC, do "bit banging" of I/O Port M bit 6 to select the DAC, and you will use the ECT to set the sample rate at which to update the DAC output. You will set ECT to produce a sample rate of 2 KHz, which translates to 500 usec between each SPI transfer to the DAC.

After each new ADC conversion, you will shift the 10-bit ADC reading two places to the right (divide by 4) to get an 8-bit number which you will then place in memory variable NUM_0. After each output to the DAC you will divide that 10-bit number by 4 also and place in memory variable NUM_1. Using sections of the programs from prior labs, the most current reading of the voltage on VR1 and the most current voltage output by the DAC will be displayed on LED display as selected by the MODE key.

Use the UP and DOWN keys to dynamically increase or decrease the output rate for the DAC.

Prelab:

1. Draw a diagram of the expected sawtooth output of the DAC and show how it relates to the reading of the voltage on VR1, outputs via the SPI, and writes to Port M bit 6.

2. Determine how long it will take to do one complete waveform assuming VR1 is set to the highest voltage.

3. Draw a flow chart of the program.

4. Generate the assembly code for the program.

Do the steps above **BEFORE** coming to lab. I guarantee that you will not have time to design the program, write the code, compile it and debug it all in the time allocated for the lab.

Procedure: Working with your lab partner(s), complete the following steps:

Using the procedures you learned in the previous labs,

1. Create a new project and enter your program as *main.asm*. Generate the executable object file using the *Make* facility and then download the program on the Dragon 12 Plus.
2. Set up the logic analyzer to monitor MOSI, SCLK, amd DAC ship select.
3. Attach an oscilloscope probe to J4 pin 1 to monitor the output voltage from Channel A of the DAC
4. Attach a second scope probe to PAD07 to monitor the voltage of VR1.
5. Use a screwdriver and turn the wiper on the trimmer pot VR1 and observe the voltage change on the oscilloscope.
6. Do an A/D conversion and display the number using the True Time Debugger and manually calculate the voltage. Does the number agree with reading from the scope?
7. Now finish debugging your program.
8. Print out the following waveforms:
   - a typical SPI transfer sequence showing the clock pulses and MOSI data
   - a complete set of SPI transfers to the DAC for one cycle of the sawtooth
   - some sample showing the timing of sections of the program, e.g. the display update clock, the sample rate clock
   - a printout of the total sawtooth waveform from the DAC on the oscilloscope
   - a close up of the analog output showing the discrete steps in the waveform
9. Demonstrate the operation of the program to the instructor or the TA and get the Lab Verification sheet signed.

Debug Hints:

1. Use bit banging on unused I/O port bits to show when your program is in the various ISRs or subroutines
2. Test one section at a time, such as the A/D conversion. an SPI transfer, or an output to the DAC to set a voltage, etc.
3. When using the ECT to generate the sample timing, set the output pin to toggle mode so you can check the timing of the sample rate