

# EE 335 Advanced Microcontroller Engineering

## Oregon Tech Portland, Winter 2014

### Lab Assignment #4 — Serial Peripheral Interface Due February 6

#### Objective:

The student will control peripheral devices using the Serial Peripheral Interface. The student will use a DAC to synthesize a sine wave.

#### Equipment and Software needed:

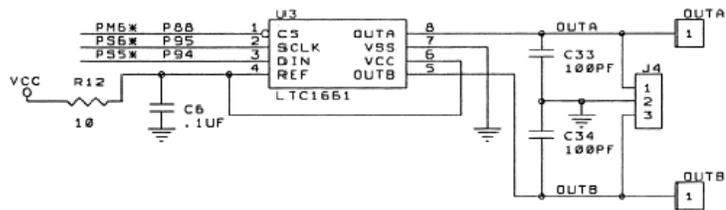
- Dragon12-Plus and other things you know by now.
- Oscilloscope

#### General Instructions:

The 68HCS12 has three SPI interfaces. The Dragon12 board has a dual 10 bit DAC, the LTC1661, connected to the MOSI and SCK pins of SPI0.

The data sheet for the LTC1661 is provided on the course file

distribution. The select input of the LTC1661 is connected to pin 6 of port M. The \*SS and MISO connections of SPI0 are not used.



The configuration of the SPI will require you to do some investigation of the LTC1661 data sheet. The CPOL, CPHA, and LSBFE bits will need to be set to match the characteristics of the LTC1661. Set the SPI baud rate to the maximum that will both meet the LTC1661 specifications and not exceed the limits of the 68HCS12 (minimum divisor of 4).

The code necessary to write to the DAC will be similar to the example in the text "Extending the Interface". The CS pin is controlled in software and must be low while two bytes are transmitted, just like \*SS is low for two bytes in the text example.

You might want to try a small test program just to make sure you are operating the DAC correctly.

#### Part 1: Generating the Sine Wave (80% of grade)

In this lab you are to use one of the DACs to generate a 100Hz sine wave. To do this, first create a timer interrupt routine that gets invoked every 100 microseconds. In the interrupt routine you will need to send a value to the DAC. Each value sent will be from consecutive entries in a sine table that represents a full cycle in 100 values.

This means that at index  $i$  in the table is the value  $\sin\left(\frac{2\pi i}{100}\right)$ . Of course we need

values suitably scaled and offset for the DAC. We want zero to be DAC value 511 and the peak values to be 0 and 1022. In addition, looking at the data sheet for the DAC, the values need to be multiplied by 4 to put them in the right bit positions for the DAC commands. We can do all of this manipulation when we built the table.

The easiest way to make the table is to use a spreadsheet program where the cell values are  $\text{round}(\sin(\text{index} \times 2 \times 3.14159 / 100) \times 511 + 511, 0) \times 4$ . This data is then copied into the program as a table. I'll save you this work by providing the table:

```
sine:  dw    2044,2172,2300,2428,2552,2676,2796,2916,3028,3140
      dw    3244,3348,3444,3536,3620,3696,3768,3836,3892,3944
      dw    3988,4024,4052,4072,4084,4088,4084,4072,4052,4024
      dw    3988,3944,3892,3836,3768,3696,3620,3536,3444,3348
      dw    3244,3140,3028,2916,2796,2676,2552,2428,2300,2172
      dw    2044,1916,1788,1660,1536,1412,1292,1172,1060,948
      dw    844,740,644,552,468,392,320,252,196,144
      dw    100,64,36,16,4,0,4,16,36,64
      dw    100,144,196,252,320,392,468,552,644,740
      dw    844,948,1060,1172,1292,1412,1536,1660,1788,1916
```

You will also need to set up the heartbeat monitor so you know if the program is running or crashed.

Do not use the SPI interrupt for this assignment. We want to send values every 100 microseconds, not at the maximum possible rate. So just poll the SPI flags in a timer interrupt that executes every 100 microseconds.

## Part 2: Modification (20% of grade)

Add a new variable to the program *period*, a 16 bit unsigned value. The period is to represent the desired period of the sine wave in microseconds. You will need to calculate the time of the next interrupt based on the period setting rather than just a constant value. Since the period can only be changed before the program runs, you can make this calculation part of the initialization code.

Run the program with two or three different values for period to check operation. Don't make the period too small or it won't work! **Discover what the minimum period is.**

**Don't forget the heartbeat monitor interrupt.**

**To turn in:**

- Commented program listings
- Sketch or screenshot of the generated waveforms showing the axis scales.
- Discussion of the results, including the minimum period.