Lab 5g Digital to Analog Conversion

This laboratory assignment accompanies the book, Embedded Microcomputer Systems: Real Time Interfacing, Second edition, by Jonathan W. Valvano, published by Thomson, copyright © 2006.

Goals	• DAC conversion, • SPI interface, • Sine-wave generation.
Review	• Data sheets on the Maxim MAX5154 dual 12-bit DAC, • Valvano Chapter 7 on SPI interfacing, • Valvano Chapter 8 on NPN transistors, • Valvano Chapter 11 on DAC converters.
Starter files	• `OC` project

Background

Most digital music devices rely on high-speed digital to analog converters (DAC) to create the analog waveforms required to produce high-quality sound. In Labs 5g and 6g together, you will create a very simple music player that illustrates a typical application of the DAC. It will be appropriate to read both labs before started to design this lab. Your goal of these two labs together is to play your favorite song. However, in this lab, the specific objectives are to interface a 12-bit DAC, and then create a sine-wave output. For the first step, you will interface a Maxim MAX5154 12-bit DAC to the SPI port, as shown in Figure 5.1. Please refer to the MAX5154 data sheets for the synchronous serial protocol. In this lab, the output of the DAC will be connected to a voltmeter, an oscilloscope or a spectrum analyzer. You are allowed to use any DAC chip you want, as long as it runs on a single +5V supply and has an SPI interface. Many DACs, such as the MAX5154, require a reference voltage. A stable 2.50V reference can be created using a reference chip such as the REF03 or MAX6225. The REF03 reference can be powered from the +5V supply, but the MAX6225 requires a supply voltage (called Vin) above 8 V. To use the MAX6225 you will need to connect MAX6225 Vin (pin 2) directly to the wall-wart (pin 40, labeled Vin on 9S12C32 docking module.) Read the MAX5154 data sheet to see how to connect OSA, CL and PDL pins.

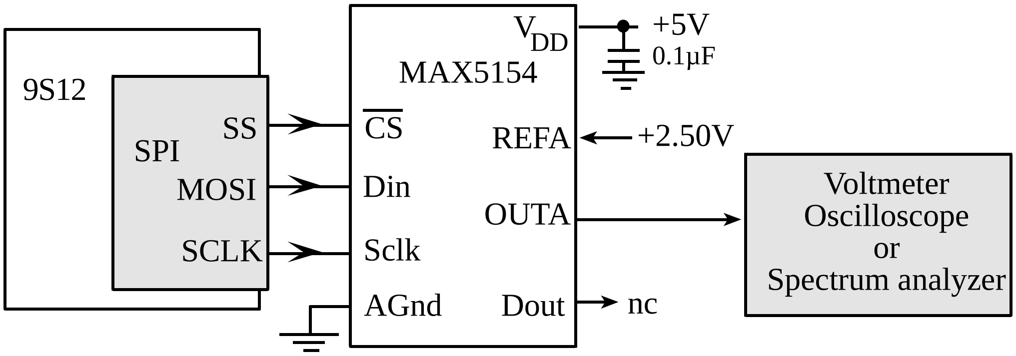

Figure 5.1. Block diagram of the DAC interface.

The second step is to design a low-level device driver for the DAC. For the MAX5154, two 8-bit SPI frames are required to set the DAC output. Next, you will design a data structure to store the sine-wave. The main program will input from switches and allow the operator to select from three pre-defined frequencies. There is an option in Lab 6 to use both channels, so you may wish to write drivers for OUTA and OUTB.

If you output a sequence of numbers to the DAC that form a sine-wave, then you should be able to see the wave on an oscilloscope, as shown in Figure 5.2. This measured data was collected using a 12-bit DAC (MAX5154) having a range of 0 to +5 V. The plot on the left was measured with a digital scope (without a speaker attached). The plot on the right shows the frequency response of this data, plotting amplitude (in dB) versus frequency (in kHz). This measured waveform is approximately 2.5+2.5*sin(2π262πt) volts. The two peaks in the spectrum are at DC and 262 Hz (e.g., 20*log(2.5)= 8.0 dB).

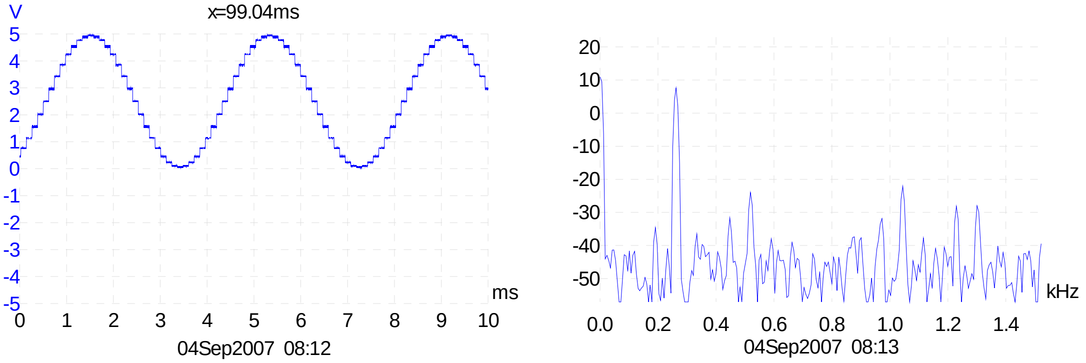

Figure 5.2. Experimental data of a 262 kHz sine-wave. The plot on the right is the Fourier Transform (frequency spectrum dB versus kHz) of the data plotted on the left.

Preparation (do this before your lab period)

1. Draw the circuit required to interface the DAC to the 9S12 SPI port. Include signal names and pin numbers. The bypass capacitor on the +5 V supply of the DAC can be any value from 0.01 µF to 0.22 µF. Draw the circuit required to interface three push button switches.

2. Write a low-level device driver for the SPI interface. Include two functions that implement the SPI/DAC interface. The function **`DAC_Init()`** initializes the SPI protocol, and the function **`DAC_Out()`** sends a new data value to the DAC. Create separate **`DAC.h`** and **`DAC.c`** files. Write a second low-level device driver for the three switches, creating separate **`Switch.h`** and **`Switch.c`** files.

3. Write a couple of simple main programs that test the SPI/DAC interface. You will use this software to test the SPI interface, and the DAC hardware. This main program could be used for static testing.

```
void main(void){unsigned short number;
  SCI_Init(19200);   // initialize SCI interface
  DAC_Init();        // initialize SPI/DAC interface
  while(1){
    number = SCI_InUHex();   // read from PC keyboard
    number = number&0x0FFF;  // 12-bit only
    DAC_OutA(number);        // output to SPI
  }
}
```

This main program could be used for dynamic testing. It creates a triangle waveform.

```
void main(void){unsigned short n;
  DAC_Init();    // initialize SPI/DAC interface
  while(1){
    for(n=0; n<4096 ; n++){ // up 0 to 4095
      DAC_OutA(n);          // output to SPI
    }
    for(n=4094; n>0 ; n--){ // down 4094 to 1
      DAC_Out(n);           // output to SPI
    }
  }
}
```

4. Design and write the sine-wave generator software. Use output compare interrupts to create the frequency. The frequency will be selected amongst three possibilities, and the user will be able to select the frequency using touch switches. Add minimally intrusive debugging instruments to allow you to visualize when interrupts are being processed.

A "syntax-error-free" hardcopy listing for the software is required as preparation. The TA will check off your listing at the beginning of the lab period. You are required to do your editing before lab. The debugging will be done during lab. Document clearly the operation of the routines. Figure 5.3 shows the data flow graph of the sine-wave generator.

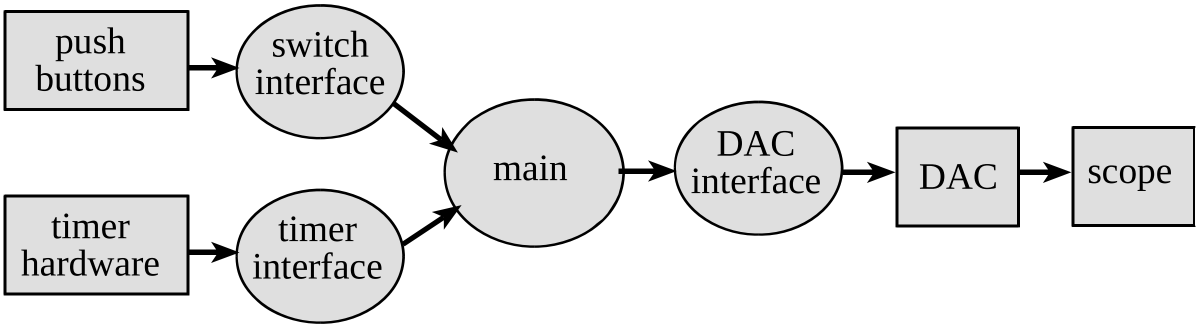

Figure 5.3. Data flows from the memory and the switches to the scope.

Figure 5.4 shows a possible call graph of the system. Dividing the system into modules allows for concurrent development and eases the reuse of code.

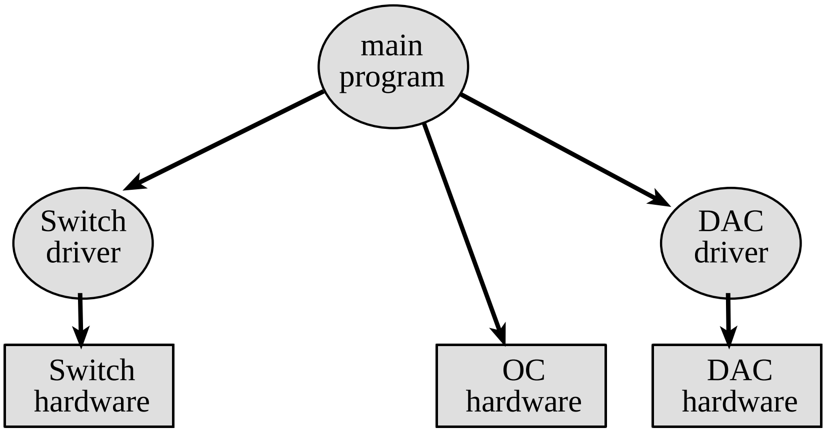

Figure 5.4. A call graph showing the modules used by the sine-wave generator.

Procedure (do this during your lab period)

1. Use the simple main programs to debug the SPI/DAC interface. Experimentally measure the DAC output versus digital input for 8 different digital inputs. Compare the measured data with the expected values. Calculate resolution, range, precision and accuracy.
2. Debug the sine-wave generator system.
3. Using an oscilloscope and spectrum analyzer, measure the time-domain and frequency-domain outputs from your system at one frequency, like Figure 5.2.

Deliverables (exact components of the lab report)

A) Objectives (1/2 page maximum)
B) Hardware Design
  Detailed circuit diagram of all hardware attached to the 9S12 (preparation 1)
C) Software Design (a hardcopy software printout is due at the time of demonstration)
  If you organized the system different than Figure 5.3 and 5.4, then draw its data flow and call graphs
D) Measurement Data
  Show the data and calculated resolution, range, precision and accuracy (procedure 1)
  Show the experimental response of DAC (procedure 3)
E) Analysis and Discussion (1/2 page maximum)

Checkout (show this to the TA)

You should be able to demonstrate the sine-wave generator software with the DAC output connected to an oscilloscope. You should be prepared to discuss alternative approaches and be able to justify your solution.

A hardcopy printout of your software will be given to your TA, and graded for style at a later time.