Benjamin Babtsov
03/18/2015

Function Generator Serial D2A Micro-Controller Applications –
Report - Micro-Controller DAC Control

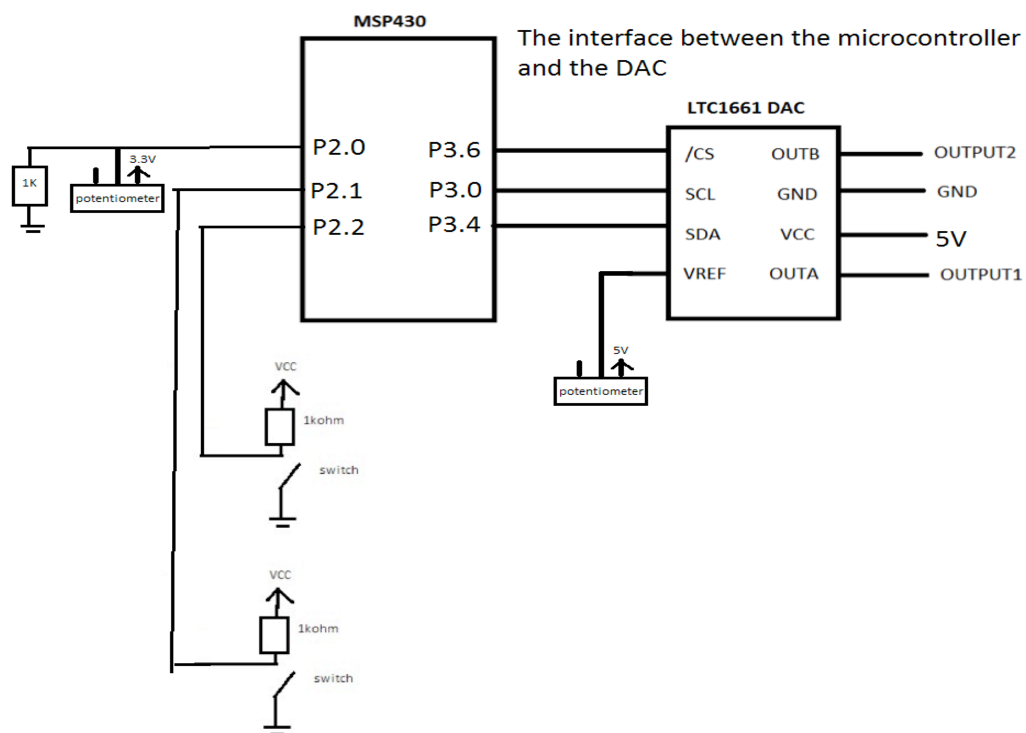
Introduction

The purpose of this module was to introduce the students to topic that he or she might not have seen or practiced in the past microprocessors course they took. These topics inside the SPI interface, which allows for serial communication between one device and the other. The "master device" (the device that has the control) also sends a clock signal to the "slave" device (the device that is being controlled). The clock signal dictates the timing of the "slave".

The other topic introduced in this module was the digital to analog converter, a device that allowed us to convert digital signals from the microcontroller to analog signals that allowed us to imitate various waveforms.

Design

The following circuitry was used to interface between the MSP and the DAC, as well as two switches and two potentiometers.

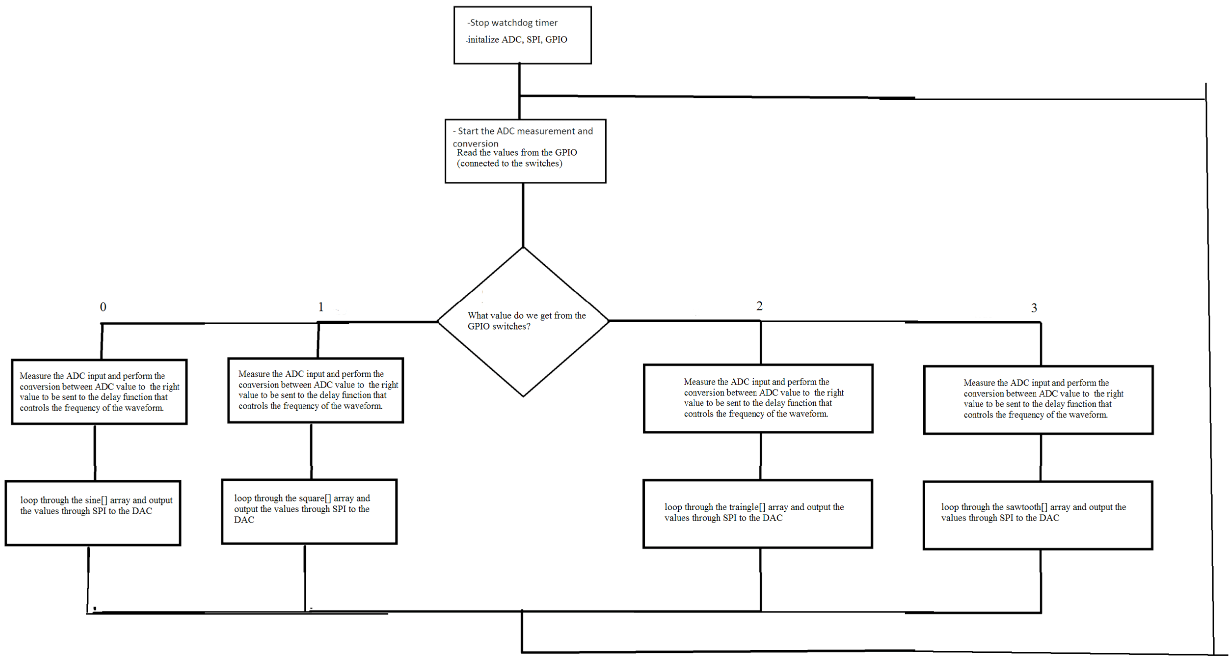


Notice that the pin used for ADC was P2.0. Connecting the ADC to a potentiometer, the user is able to control the value read by the ADC, which in turn would control the frequency of the waveform. The amplitude of the waveform, on the other hand, is controlled by another potentiometer connected to Vref in the ADC. Pins P2.1 and P2.2 were used as GPIO inputs and were connected to external switches to select a waveform type.

In order to generate the various waveforms a LUT for every waveform was stored in the memory. Each LUT contained 50 samples of one period of the various waveforms. The values of the LUT would go from 0 all the way to 1023. This is because a 10-bit ADC was used and it could accept this range of inputs (for optimal performance. The following MATLAB script was used to generate the LUT values:

```
t = [0: 2*pi/49 : 2*pi];
sin1 = 1023/2*(1+sin(t));
square1 =1023/2*(1+square(t));
sawtooth1 = 1023/2*(1+sawtooth(t));
traingle1 = 1023*abs(sawtooth(t));
sin1 = floor(sin1);
square1 = floor(square1);
traingle1 = floor(traingle1);
sawtooth1 = floor(sawtooth1);
sprintf('Sine:')
sprintf('%d,',sin1)
sprintf('Square')
sprintf('%d,',square1)
sprintf('Traingle')
sprintf('%d,',traingle1)
sprintf('Sawtooth')
sprintf('%d,',sawtooth1)
```

The following is the software outline:



Here is the bill of materials:

Part	Cost
TI MSP430F2274 microcontroller	$5.22
LTC1661	$2.25
2 10K resistors	$0.06
2 Potentiometer	$0.60
DIP switch bank	$1.11
SIP resistor	$0.33
TOTAL	$9.57

Conclusion

As a conclusion, this module allowed me to practice some new modules of microcontrollers that I didn't have experience with in pervious classes like SPI and DAC. Some difficulties encountered in the design were the incorrect synchronization of the DAC CS and the SPI interface (it was solved by adding a slight delay). The ADC sampling also created a challenge, as the sampling was relatively slow to the rate at which the microcontroller has to output data to the DAC. Preforming the ADC sampling less frequently and changing the ADC conversion formula from floats to integers helped overcoming this challenge.