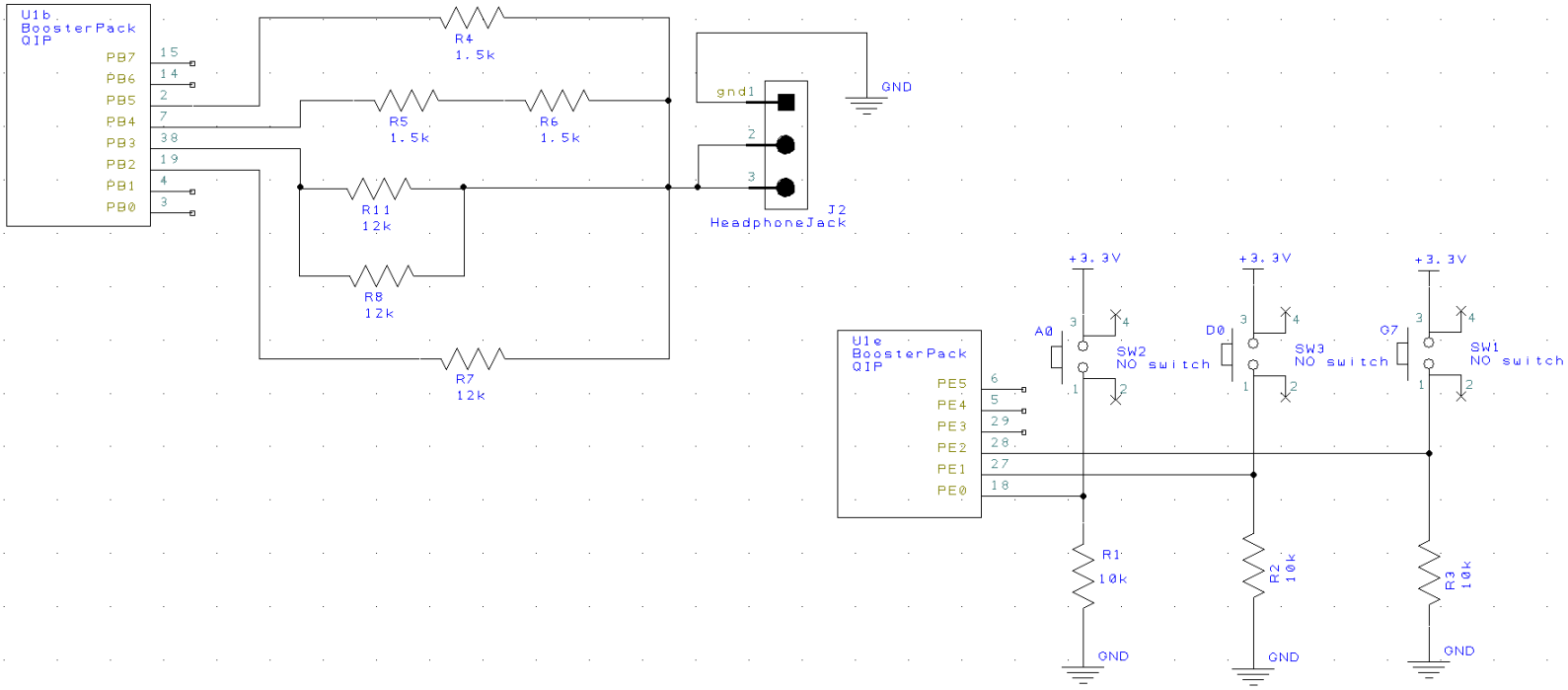
Name: Noah Najvar, Adrian Gallegos **EID:** ndn387, ag76424 **Professor:** VT

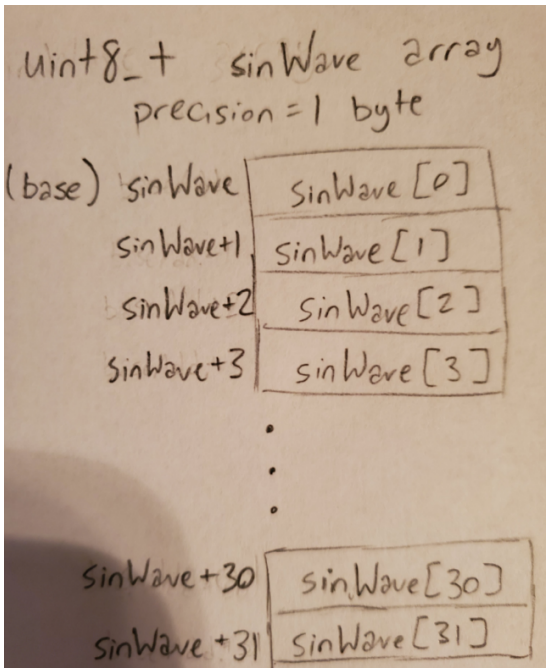
Lab 6: Piano DAC

2. Circuit Diagram



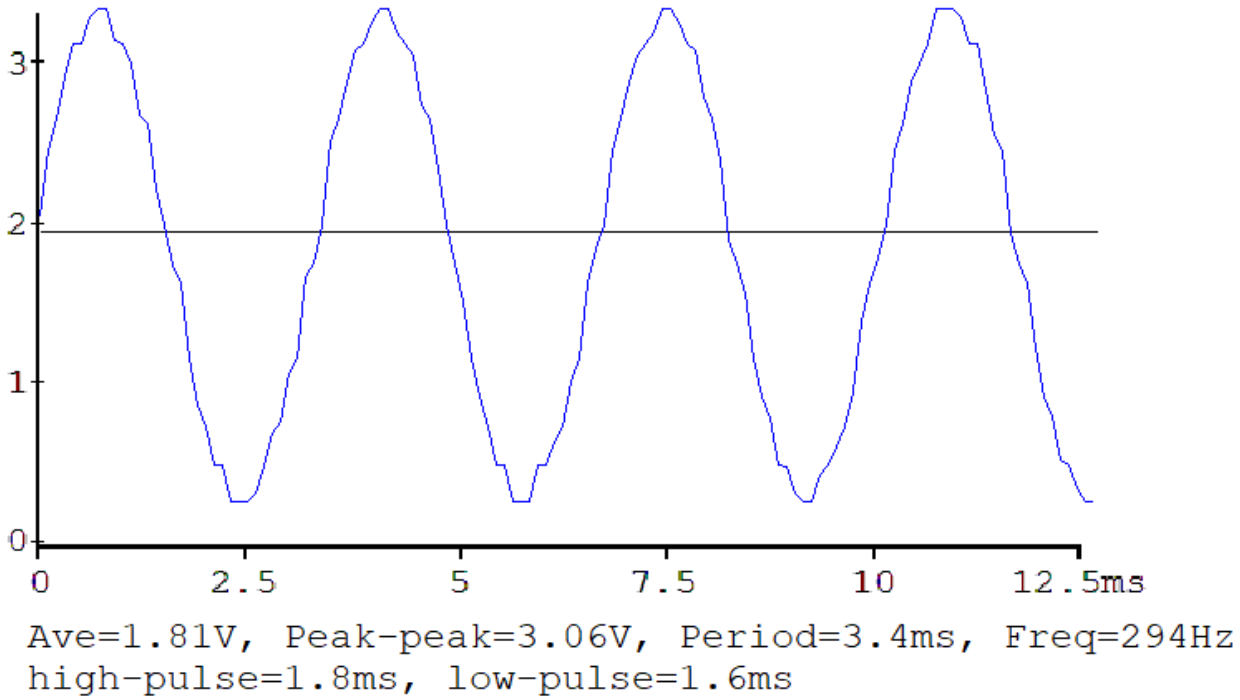
3. Software Design

a.



b. We used the same system organization as the lab doc.

4. TExaS Scope



Sine wave for D0 (293.7 Hz)

5. Measurement Data

Bit3 bit2 bit1 bit0	Theoretical DAC voltage (V)	Measured DAC voltage (V)
0	0.00	0.000
1	0.22	0.233
2	0.44	0.461
3	0.66	0.694
4	0.88	0.893
5	1.10	1.126
6	1.32	1.354
7	1.54	1.587
8	1.76	1.720
9	1.98	1.952
10	2.20	2.179
11	2.42	2.410
12	2.64	2.607
13	2.86	2.838
14	3.08	3.065
15	3.30	3.296

Table 6.3. Static performance evaluation of the DAC (if you implement a 6-bit DAC, then make a table with 16 measurements: 0,1,7,8,15,16,17,18,31,32,33,47,48,49,62,63).

Range: 0.000 V to 3.296 V
Resolution: 3.296 V/15 = 0.2197 V
Precision: 4 bits or 16 values
Accuracy: 94.1%-99.88%

6. Questions

a. When does the interrupt trigger occur?
   The interrupt trigger occurs when the SysTick current register reaches 0, while a button is pressed.
b. In which file is the interrupt vector?
   The interrupt vector is located in the startup.s.

c. List the steps that occur after the trigger occurs and before the processor executes the handler.

   After the trigger occurs and before the processor executes the handler we:

   - PUSH registers (R0-R3, R12)
   - Set new PC based on interrupt and PUSH old PC
   - Set IPSR based on interrupt and PUSH old PSR
   - Set LR based on and PUSH old LR

d. It looks like BX LR instruction simply moves LR into PC, how does this return from interrupt?

   Before the ISR starts, the PC register gets pushed onto the stack; when the ISR finishes, it gets popped off the stack, allowing the program to continue from where it left off.