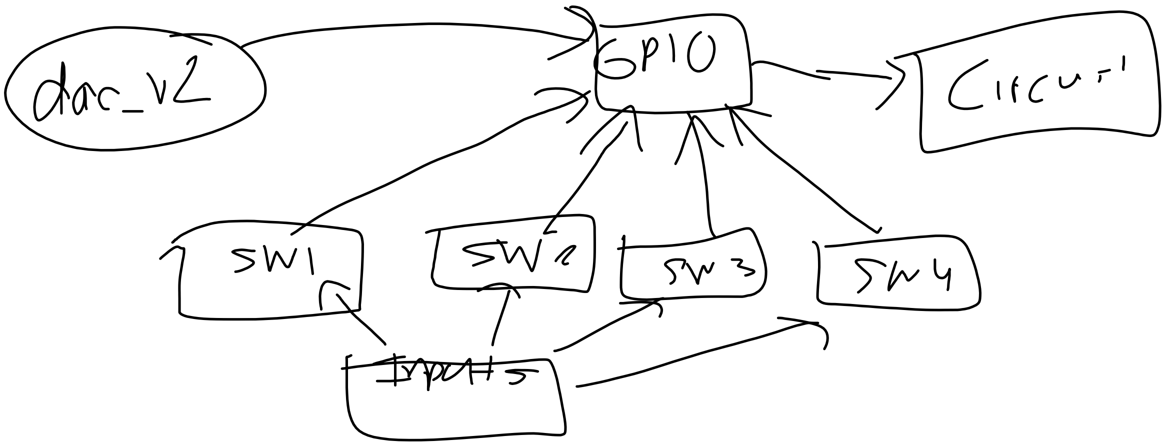
Purpose

The purpose of this lab is two combines two separate processes and implement it together using a binary-weighted DAC. Besides building the circuit itself, we also have to implement the code using our recent knowledge of python

Pre-Lab

a) Variables can be used to hold a value. In statically typed languages, variables have predetermined types, and a variable can only be used to hold values of that type. In Python, we may reuse the same variable to store values of any type.
b) The type of statements that should be used in dacOut are if and elif statements followed by the presence or absence of a not.
c) You can visually see the waveform on the screen as well make it capture a wave at one certain point in time. By zooming in on the x axis and y axis as well as increasing the x value and y value, you can capture one period of a wave form altogether.



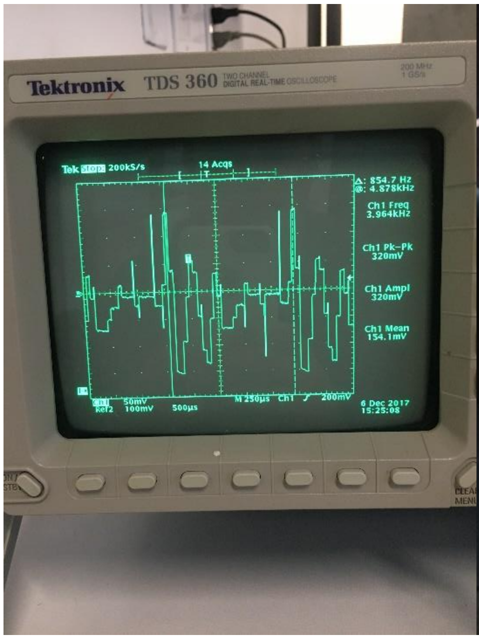
Procedure

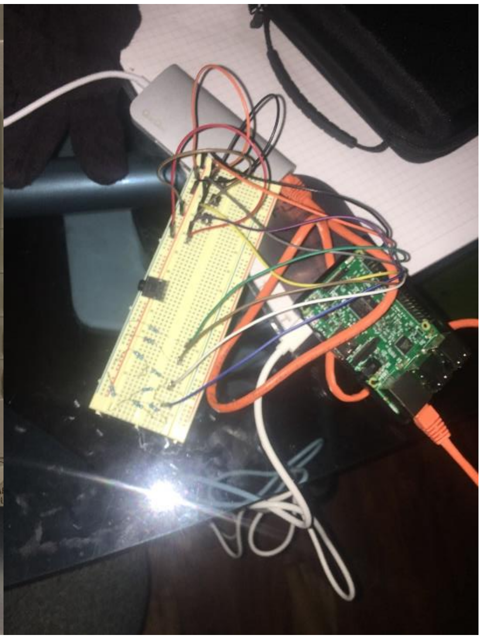
1. Complete the python code by first completing part A
2. Once completed, created a circuit to test your DAC using switches and resistors along with you raspberry pi.
3. Measure your voltage across the resistor by using a multimeter and test every combination of switches as possible.
4. Once verifying it works, complete part B of the code.
5. After uncommenting and writing sound functions, execute the code first to see if there are any errors
6. If no errors, connect to the oscilloscope and capture each individual waveform.
7. Using the headphone jack, listen to the waveform and make several observations

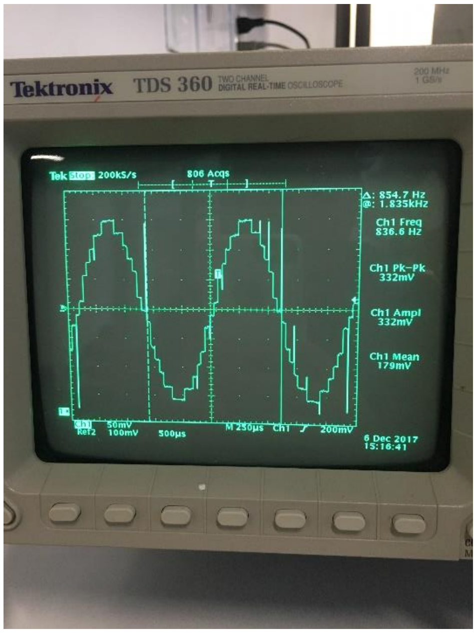
Observation

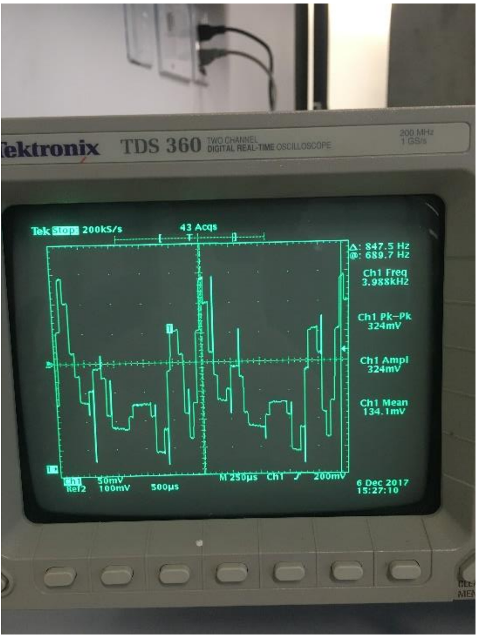
As the note changes pitch, so does the frequency. The sound goes higher since I chose high notes in my code. Having seven years of musical experience both in classical and mariachi bands, I can easily differentiate the pitch and the difference of each note. Depending on the frequency, the note will sound different.

Pictures









DAC Output

Input 3	Input 2	Input 1	Input 0	Voltage
0	0	0	0	0
0	0	0	1	0.476
0	0	1	0	0.946
0	0	1	1	0.670
0	1	0	0	1.893
0	1	0	1	1.116
0	1	1	0	1.338
0	1	1	1	1.562
1	0	0	0	1.753
1	0	0	1	1.977
1	0	1	0	2.199
1	0	1	1	2.423
1	1	0	0	2.645
1	1	0	1	2.87
1	1	1	0	3.091
1	1	1	1	3.315

Conclusion

Lab 9.2 has by far been the most difficult as it implemented us to use new found python knowledge, hardware knowledge, logical analysis, as well implementing new machinery into our computation. This displays how a project should be approached by parts and with time. Dedicating several days to do part A, then part B, then recording your results in a machine teaches us engineers patience and efficiency. There is no room for error, which should not happen at all due to our practice in debugging. Overall, by implementing a DAC into our lab, we can explore sound in a whole other dimension by not only hearing the pitch, but by also seeing the waveforms. By providing a physical image of what was going on through our circuit helped us learn more about our field as well as the wonders of electricity and the circuitry behind it. By combining software and hardware, Electrical and Computer Engineers are able to adapt and learn from the experience, preparing them for future tasks.