**Microprocessor write up HW 4**

In this homework assignment I created a music player that played “Joy to the World” and the theme song from family guy. The music player also had four buttons that either started/paused the song, reset the song, selected which song to play, or sped up/slowed down the song. I also added a few resistors in series to the speaker and ground to lower the volume.

The first thing I did to start this assignment was to look at the example tone 0 and tone 4 code provided to us. I looked at how the sound was being produced, and modified the tone 4 code to make the music player. First to determine the frequency of notes, I used C5 as the half period of 955. Then I calculated the half steps down by multiplying by 2^(1/12) and the half steps up by dividing by 2^(1/12). Then to play the song, I made an array and put the corresponding note frequencies in the right locations. To determine how long each note should play for, I used a relative base length of 500 for 1 beat of C5. Then to determine the base lengths for other notes, I did 500\*(955/(half period of specified note)).

The Start/pause button is able to start and pause the music playing from the speakers. To do this I had a condition check to see which state the button is in. If the button is playing, then it will stop the music from playing and store the current array note and array length values into a global variable. If the button is paused, then the music will start playing again, but at the saved locations for the note and lengths array.

The Reset Button causes the current song to start playing from the beginning. To do this, I had a condition see if the reset button was pressed or not. If it was pressed then the counters for the arrays were all set to zero. This will cause the song to start from the beginning.

The Select button chose between which two songs to play. To do this I had two states. One was to change to song 2 and the other state was to switch back to the original song. A condition is checked to see which state the button is currently in. If it is the first press then it changes the song select state to 2, which plays the second song. If it is the second press then it reverts back to the original song.

The speed button changed the tempo the song played to either two times the current speed or slowed it down to half the speed. To implement this I had a condition to check which state the button is in right now. If it’s the first press, it sets the variable “scale” to 0.5. So in the sound handler interrupt, when comparing the length counter to the length in the array, it will take half the time. If it’s the second press, then the scale variable is set to 2. This makes the beat twice as long.

Some issues I encountered during this assignment was using too much memory. To solve this issue, I declared my arrays as const int. This makes the array go into Read Only Memory, which gives me a lot more memory to work. Also I had an issue with the WDT detecting multiple button presses for a single press. This causes the buttons to change into incorrect states sometimes. I solved this by creating a counter called debouncer. It will iterate through the WDT interrupt 5000 times before the input of the button press is acted upon.

Some virtues in my program is that the arrays can play any song as long as it is hard coded into it. So if you want, you can change up the type of songs that are played by this program. My program right now has the limitation of only playing two songs. Also it cannot play an indefinitely long song because it only has 16k memory in the Read Only Memory. ­­­