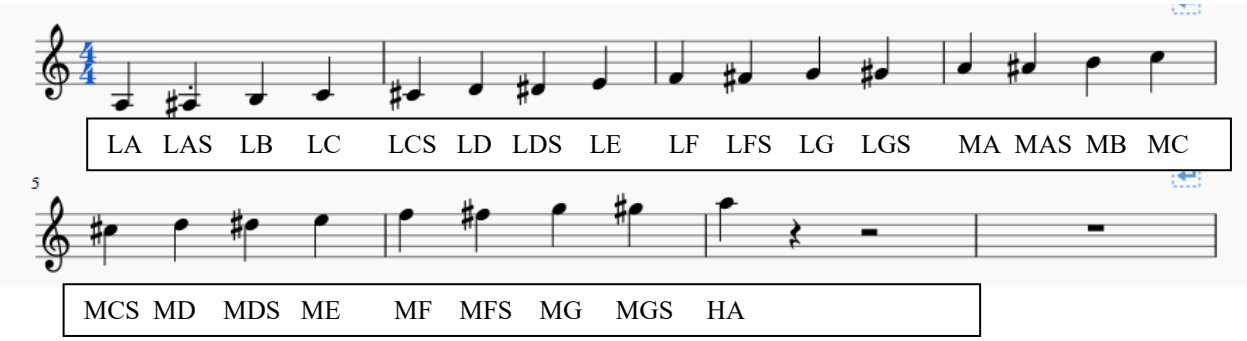
Homework #4 – Due 2/12/20
Please submit to the Dropbox in MyCourses

Datasheet Reading: Sections 19-22 (2016 version)

1. Given 25 notes of the musical scale below:



They are each produced by a square wave with frequency as defined by the following formula:

$$f_{note} = 440 * 2^{P/12}$$ **:where P ranges from -12 (low A - LA) to 12 (high A - HA)**

The period of the square wave is illustrated in figure 1 below.

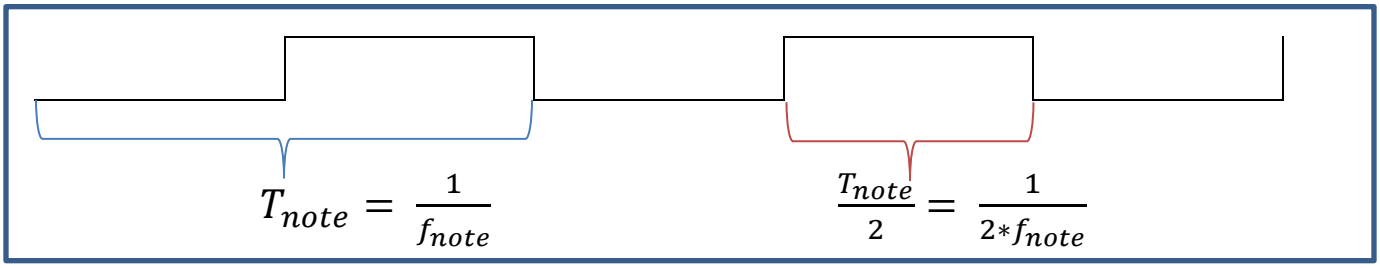


Figure 1: Equations for Period and ½ Period of a note square wave

$$\text{Clock cycle in } \tfrac{1}{2}\, T_{note} = \frac{1}{2*f_{note}} \div \frac{1}{16\ MHz} = \frac{16MHz}{2*f_{note}} = (OCR1A + 1)$$

- For the 25 notes (f_{note} = 220 through f_{note} = 880) determine the OCR1A values needed to output the square wave of each note using Timer1 in CTC mode. Hint: write a program to do the math for you and print the results to the serial monitor.

- Open another program and use #define to create 25 constants for the results of part 1. Also create time delay constants (explained in the reading music section below)
- In the initialization section setup Timer1 to create a square wave.
- Create a function called playNote()
    a. It takes in an integer representing the max count for a given note and assigns that value to the OCR1A register.
    b. It also takes in a delay constant to hold the note a given amount of time
    c. It must then turn off CTC mode and delay a short amount of time (I used 50) to put a break between notes.
    d. You can use the Arduino delay() instruction in this function
- In the main loop call the function with different notes and delays in between. You can look up simple songs on the internet and experiment with the delays to actually play a song. See my example on page 3.
- Use the speaker from your lab kit and drive one pin with the waveform output and the other to ground.
- YOU MAY NOT USE MARY HAD A LITTLE LAMB FOR YOUR ASSIGNMENT. I only give that as an example.

Reading Music

Rhythms
A basic set of note values include the whole note, dotted half note, half note and quarter note. In a single piece of music a composer will use a variety of rhythmic values.


The **WHOLE NOTE** receives 4 BEATS (Count 1-2-3-4)


The **DOTTED HALF NOTE** receives 3 BEATS (Count 1-2-3)


The **HALF NOTE** receives 2 BEATS (Count 1-2)


The **QUARTER NOTE** receives 1 BEAT (Count 1)

Rests
Music is both a combination of sound and silence. The length of the silence in music is determined by a rest.


The **WHOLE REST** is a unique rest. It often means to rest for four beats, though simply means rest for the "whole" measure.

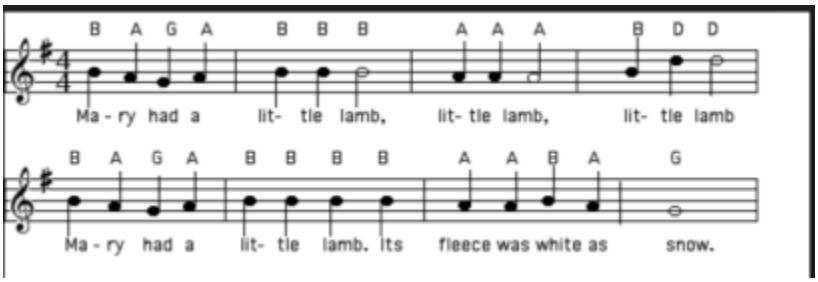

The **HALF REST** receives 2 BEATS (Count 1-2)


The **QUARTER REST** receives 1 BEAT (Count 1)

I made the whole note 1 second and divided down from there.

Example is on the next page

Given the snippet below of *Mary Had a Little Lamb*



My code was:

```
playnote(MB,QUARTER);
playnote(MA,QUARTER);
playnote(LG,QUARTER);
playnote(MA,QUARTER);
playnote(MB,QUARTER);
playnote(MB,QUARTER);
playnote(MB,HALF);
playnote(MA,QUARTER);
playnote(MA,QUARTER);
playnote(MA,HALF);
playnote(MB,QUARTER);
playnote(MD,QUARTER);
playnote(MD,HALF);
playnote(MB,QUARTER);
playnote(MA,QUARTER);
playnote(LG,QUARTER);
playnote(MA,QUARTER);
playnote(MB,QUARTER);
playnote(MB,QUARTER);
playnote(MB,QUARTER);
playnote(MB,QUARTER);
playnote(MA,QUARTER);
playnote(MA,QUARTER);
playnote(MB,QUARTER);
playnote(MA,QUARTER);
playnote(LG,WHOLE);
```

2. The problem with using a timer in the CTC mode is that you can only create an output wave with a 50% duty cycle. Sometimes a wave with a different duty cycle is required. Use **Timer1 in normal mode** to create an output wave that is high for 250 ms and low for 750 ms. Assume this output is driving an LED causing it to blink. You can use a two-state state machine. Instead of maintaining a state counter and delaying 1 ms on each loop as was done in lab, you can simply check for the TOV1 bit to be set and use that as the exit condition. Note, you will need to use a pre-scaler. While PWM can also be used to solve this problem, it should be done using normal mode.

3. In homework 1, every student came up with an invention that is controlled by a microcontroller. While meeting with your homework group, each group member should explain their invention. Then, as a group, discuss the feasibility of the invention. For each invention, answer the questions “is it feasible? If not, why not?” List the input and output devices that the invention would require.