Name: Rylan Bumbasi

Red ID#: 822563190

Section: 02

Date: 10/22/2021

Lab 05 Report

1) **What does your program do?**
   My program will take user input from a user via Multiplexing a keypad and generate a waveform with the period correlating to the buttons pressed. Once the period for the wave form is selected, it will generate the waveform in pin PD3. An AUX port is connected to pin PD3 meaning that when the waveform is generated, the AUX port will output a square wave with the period corresponding to the button pressed. The values of periods that will be sent through the keypad will be the periods for notes A4 – C6.

2) **What variables/registers does your program use?**
   My program makes use of the following registers, DDRD, DDRB, PORTD, PORTB, TCCR2A, TCCR2B, OCR2A and OCRA2B. DDRD and DDRB registers are used to set the data direction of the rows and columns of the keypad (rows as outputs and columns as inputs). The PORTD register is used to turn the rows on the keypad to 1. The PORTB register is used to enable the pull up resistors for the keypad buttons. TCCR2A register is used to set the PWM generation mode, timer compare mode of the timer and enable non-inverting mode. TCR2B register is used to set the pre-scale of the timer and also to set WGM22 value for enabling fast PWM mode.

3) **What does the main algorithm/logic of your program do?**

   My program will first initialize a unsigned char array that contains the values of periods to be sent based off of the button pressed. For this lab I hand calculated all 16 values of period and hard coded them into my array. To determine which pre-scale should be used, I first calculated the period when x = 0 since that would give me the greatest period value for all index values of the keypad. After this calculation, I determined that Pre-Scale 256 is large enough to contain all values of periods for my keypad, so I scaled every period calculated in the matrix to a pre-scale value of 256. Next, the program will perform and initialization stage where the data direction of the rows and columns of the keypad were properly set and pull up resistors were enabled. In this initialization stage, Timer 2 is also set to fast PWM mode (wraps around OCR2A value) and non-inverting mode. After the initialization stage, the program will enter an infinite while loop that calls the function scanKeypad(). The scanKeypad() function is used to detect located which button is pressed. After scanKeypad() determines which button is pressed, it will call a function transmitNote(i , j). which will set OCRA to the value of period correlating to the button pressed. Since the duty cycle for every button in this lab was 50%, then OCRB = OCRA / 2. After OCRA and OCRB are initialized to the correct values, the timer will start at a prescale of 256. When the timer starts, the waveform generated will be sent to pin PD3 and the sound can be observed from the aux port. Once the button is depressed, the timer will stop and continue searching for the event that a button is pressed.

4) What external hardware connections did you need to make?

For this lab 2 hardware connections were needed to be made, connecting the keypad, and connecting AUX port. The keypad was connected to Pins D4-7 / B0-3. The Aux Port GND was connected to GND and both output pins were connected to PD3. All connections for this lab were made via the breadboard.

Appendix Code

```
/*
 * BumbasiRylan_Lab03.c
 *
 * Created: 9/17/2021 1:28:31 PM
 * Author : Rylan Bumbasi
 * RedID: 822563190
 */



void transmitNote(int row, int column);
void scanKeypad();


#include <avr/io.h>


unsigned char keypadNotes[4][4] = {
      {141, 133, 126, 117},
      {112, 105, 99, 94},
      {88, 83, 79, 75},
      {70, 66, 63, 59}
};


int main(void)
{
      // Initializing Columns and Rows
      DDRD |= (1<<4|1<<5|1<<6|1<<7); // Set rows as output (PORT
D 4 - 7)
      DDRB  &= ~(1<<0|1<<1|1<<2|1<<3); // Set columns as Inputs
      PORTD |= (1<<4|1<<5|1<<6|1<<7); // Set rows as to high
      PORTB |= (1<<0|1<<1|1<<2|1<<3); // Set Columns to high (to
enable pull up resistors)

      DDRD |= (1<<3); // Set PD3 to output (For PWM generation
from timer)

      TCCR2A |= (1 << COM2B1); // Set OC2B for Non-Inverting Mode

      TCCR2A |= (1 << WGM21) | (1 << WGM20); // Set timer 2 to
Fast PWM Mode (top = OCR2A)
      TCCR2B |= (1 << WGM22);
```

```
     /* Replace with your application code */
    while (1)
    {
           scanKeypad();
    }
}



void scanKeypad()     {
     for(int i = 0; i <= 3; i++) { // index through matrix rows
           PORTD &= ~(1 << (i + 4)); // set the current row value
to 0
           for(int j = 0; j <= 3; j++) { // index through matrix
columns
                if( !(PINB & (1<<j))) // check to see if current
column and row are both 0
                {
                      transmitNote(i, j);
                      while(!(PINB & (1<<j)));
                      TCCR2B &= ~(1 << CS22);
                      TCCR2B &= ~(1 << CS21); // Turn Timer off
                }

           }
           PORTD |= (1 << (i + 4)); // set the current row value
back to 1

     }
}

void transmitNote(int row, int column) {
     OCR2A = keypadNotes[row][column];
     OCR2B = OCR2A / 2;
     TCCR2B |= (1 << CS22) | (1 << CS21); // Turn timer on (Pre-
Scale 256);

}
```