CPE301 – SPRING 2019

Design Assignment 2C

Student Name: Chris Barr

Student #: 2000682859

Student Email: barrc1@unlv.nevada.edu

Primary Github address: https://github.com/BarrChris

Directory: https://github.com/BarrChris/submission\_da.git

Submit the following for all Labs:

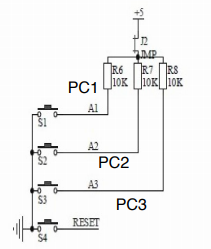
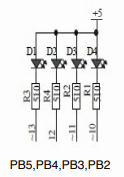
1. In the document, for each task submit the modified or included code (only) with highlights and justifications of the modifications. Also, include the comments.
2. Use the previously create a Github repository with a random name (no CPE/301, Lastname, Firstname). Place all labs under the root folder ESD301/DA, sub-folder named LABXX, with one document and one video link file for each lab, place modified asm/c files named as LabXX-TYY.asm/c.
3. If multiple asm/c files or other libraries are used, create a folder LabXX-TYY and place these files inside the folder.
4. The folder should have a) Word document (see template), b) source code file(s) and other include files, c) text file with youtube video links (see template).

1. **COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS**

List of Components used

* Atmega328P
* Multi-Function Shield
* LED (PB2)
* Pushbutton (PC2)

Block diagram with pins used in the Atmega328P

These are block diagrams for the 2 main components I used, PC2 (pushbutton), PB2 (LED)

1. **INITIAL/MODIFIED/DEVELOPED CODE OF DA 2A**

//Task 1

#include <avr/io.h>

#include <stdio.h>

int main**(**void**)**

**{**

DDRB **=** **(**1**<<**2**);** //Sets PB2 as an output

TCCR1B **=** 5**;** //Set prescaler to 1024

**while(**1**)** //infinite loop to keep the LED's to blink

**{**

TCNT1 **=** 0**;** //Reset counter

**while** **(**TCNT1 **!=** 4530**){}** //Delay to keep LED off

PORTB **^=** **(**1**<<**2**);** //LED turns on

TCNT1 **=** 0**;** //Reset counter

**while** **(**TCNT1 **!=** 6796**){}** //Delay to keep LED on

PORTB **^=** **(**1**<<**2**);** //LED turns off

**}**

**return** 0**;**

**}**

//Task 2

#include <avr/io.h>

#include <stdio.h>

int main**(**void**)**

**{**

DDRB **=** **(**1**<<**2**);** //PB2 as an output

PORTB **=** **(**1**<<**2**);** //Sets PB2

DDRC **=** **(**0**<<**2**);** //PC2 as an input

PORTC **=** **(**0**<<**2**);** //Sets PC2

TCCR1B **=** 5**;** //Sets prescaler to 1024

**while** **(**1**)** //loop forever to keep program running

**{**

**if** **(!(**PINC **&** **(**1 **<<** PINC2**)))** //If the pushbutton is pressed, go through this if statement

**{**

PORTB **&=** **~(**1**<<**2**);** //Keeps LED on until otherwise

**while** **(**TCNT1 **!=** 19530**){}** //Delay to keep LED off

TCNT1 **=** 0**;** //Reset counter

**}**

**else** //else, if the pushbutton is not pressed, go through this else statement

**{**

PORTB **|=** **(**1**<<**2**);** //Keeps LED off until otherwise

TCNT1 **=** 0**;** //Reset counter

**}**

**}**

**return** 0**;**

**}**

1. **DEVELOPED MODIFIED CODE FOR DA 2C from DA 2A**

/\*

\* task1a

\*

\* Created: 3/22/2019 10:13:32 PM

\* Author : Chris Barr

\*/

#include <avr/io.h>

#include <stdio.h>

int main**(**void**)**

**{**

DDRB **|=** **(**1**<<**5**)** **|** **(**1**<<**2**);** //Sets PB5/PB2 as outputs

PORTB **|=** **(**1**<<**5**)** **|** **(**1**<<**2**);** //Clears PB5/PB2

TCCR0B **=** 5**;** //Set prescaler to 1024

int x**=**0**;**

**while(**1**)** //infinite loop to keep the LED's to blink

**{**

TCNT0 **=** 0**;** //Reset counter

x **=** 0**;**

**while** **(**x **<=** 32**)** //TCNT to 4530

**{**

**while(**TCNT0 **!=** 137**){}**

TCNT0 **=** 0**;**

x**++;**

**}** //Delay to keep LED off

PORTB **^=** **(**1**<<**2**);** //LED turns on

TCNT0 **=** 0**;** //Reset counter

x **=** 0**;**

**while** **(**x **<=** 32**)** //TCNT to 6796

**{**

**while(**TCNT0 **!=** 206**){}**

TCNT0 **=** 0**;**

x**++;**

**}** //Delay to keep LED on

PORTB **^=** **(**1**<<**2**);** //LED turns off

**}**

**return** 0**;**

**}**

/\*

\* task1b.c

\*

\* Created: 3/23/2019 12:17:12 AM

\* Author : Chris Barr

\*/

#include <avr/io.h>

#include <stdio.h>

int main**(**void**)**

**{**

DDRB **|=** **(**1**<<**5**)** **|** **(**1**<<**2**);** //Sets PB5/PB2 as outputs

PORTB **|=** **(**1**<<**5**)** **|** **(**1**<<**2**);** //Clears PB5/PB2

DDRC **=** **(**0**<<**2**);** //PC2 as an input

PORTC **=** **(**0**<<**2**);** //Sets PC2

TCCR0B **=** 5**;** //Sets prescaler to 1024

int x **=** 0**;**

**while** **(**1**)** //loop forever to keep program running

**{**

**if** **(!(**PINC **&** **(**1 **<<** PINC2**)))** //If the pushbutton is pressed, go through this if statement

**{**

PORTB **&=** **~(**1**<<**2**);** //Keeps LED on until otherwise

**while** **(**x **<=** 76/\*TCNT0 != 19530\*/**)** //Delay to keep LED on

**{**

**while** **(**TCNT0 **!=** 253**);**

TCNT0 **=** 0**;**

x**++;**

**}**

**while(**TCNT0 **!=** 51**);**

TCNT0 **=** 0**;** //Reset counter

x **=** 0**;**

**}**

**else** //else, if the pushbutton is not pressed, go through this else statement

**{**

PORTB **|=** **(**1**<<**2**);** //Keeps LED off until otherwise

TCNT0 **=** 0**;** //Reset counter

**}**

**}**

**return** 0**;**

**}**

/\*

\* task2a.c

\*

\* Created: 3/23/2019 1:13:59 AM

\* Author : Chris Barr

\*/

#include <avr/interrupt.h>

#include <avr/io.h>

#include <stdio.h>

int main**(**void**)**

**{**

DDRB **|=** **(**1**<<**5**)** **|** **(**1**<<**2**);** //Sets PB5/PB2 as outputs

PORTB **|=** **(**1**<<**5**)** **|** **(**1**<<**2**);** //Clears PB5/PB2

TIMSK0 **|=** **(**1 **<<** TOIE0**);**

sei**();**

//enables interrupt

TCCR0B **=** 5**;** //Set prescaler to 1024

**while(**1**)** //infinite loop to keep the LED's to blink

**{**

**}**

**return** 0**;**

**}**

ISR **(**TIMER0\_OVF\_vect**)**

**{**

int x **=** 0**;**

TCNT0 **=** 0**;** //Reset counter

x **=** 0**;**

**while** **(**x **<=** 32**)** //TCNT to 4530 with overflow 32 times

**{**

**while(**TCNT0 **!=** 137**){}**

TCNT0 **=** 0**;** //Reset counter

x**++;** // Increment Over Flow Counter

**}** //Delay to keep LED off

PORTB **^=** **(**1**<<**2**);** //LED turns on

TCNT0 **=** 0**;** //Reset counter

x **=** 0**;**

**while** **(**x **<=** 32**)** //TCNT to 6796 with overflow 32 times

**{**

**while(**TCNT0 **!=** 206**){}**

TCNT0 **=** 0**;** //Reset counter

x**++;** // Increment Over Flow Counter

**}** //Delay to keep LED on

PORTB **^=** **(**1**<<**2**);** //LED turns off

**}**

/\*

\* task2b.c

\*

\* Created: 3/23/2019 2:02:29 AM

\* Author : Chris Barr

\*/

#include <avr/interrupt.h>

#include <avr/io.h>

#include <stdio.h>

int main**(**void**)**

**{**

DDRB **|=** **(**1**<<**5**)** **|** **(**1**<<**2**);** //Sets PB5/PB2 as outputs

PORTB **|=** **(**1**<<**5**)** **|** **(**1**<<**2**);** //Clears PB5/PB2

DDRC **=** **(**0**<<**2**);** //PC2 as an input

PORTC **=** **(**0**<<**2**);** //Sets PC2

TCCR0B **=** 5**;** //Sets prescaler to 1024

TIMSK0 **|=** **(**1 **<<** TOIE0**);**

sei**();** // enables interrupt

**while** **(**1**)** //loop forever to keep program running

**{**

**}**

**return** 0**;**

**}**

ISR **(**TIMER0\_OVF\_vect**)**

**{**

int x **=** 0**;**

**if** **(!(**PINC **&** **(**1 **<<** PINC2**)))** //If the pushbutton is pressed, go through this if statement

**{**

PORTB **&=** **~(**1**<<**2**);** //Keeps LED on until otherwise

**while** **(**x **<=** 76/\*TCNT0 != 19530\*/**)** //Delay to keep LED off

**{**

**while** **(**TCNT0 **!=** 253**);** // Inner delay loop for 8 bit timer

TCNT0 **=** 0**;** // Reset counter

x**++;** // Over Flow Counter

**}**

**while(**TCNT0 **!=** 51**);** // Additional delay

TCNT0 **=** 0**;** // Reset counter

x **=** 0**;** // Reset Over Flow Counter

**}**

**else** //else, if the pushbutton is not pressed, go through this else statement

**{**

PORTB **|=** **(**1**<<**2**);** //Keeps LED off until otherwise

TCNT0 **=** 0**;** //Reset counter

**}**

**}**

/\*

\* task3a.c

\*

\* Created: 3/23/2019 2:12:00 AM

\* Author : Chris Barr

\*/

#include <avr/interrupt.h>

#include <avr/io.h>

#include <stdio.h>

int main**(**void**)**

**{**

DDRB **|=** **(**1**<<**5**)** **|** **(**1**<<**2**);** //Sets PB5/PB2 as outputs

PORTB **|=** **(**1**<<**5**)** **|** **(**1**<<**2**);** //Clears PB5/PB2

TCCR0B **=** 5**;** //Set prescaler to 1024

TCCR0A **=** **(**1 **<<** WGM01**);** //Set CTC Mode

TIMSK0 **=** **(**1 **<<** OCIE0A**);** //Set interrupt on compare match

sei**();**

OCR0A **=** 0xFF**;**

**while(**1**)** //infinite loop to keep the LED's to blink

**{**

//main loop

**}**

**return** 0**;**

**}**

ISR **(**TIMER0\_COMPA\_vect**)**

**{**

int x **=** 0**;**

TCNT0 **=** 0**;** //Reset counter

x **=** 0**;**

**while** **(**x **<=** 32**)** //TCNT to 4530

**{**

**while(**TCNT0 **!=** 137**){}** // Delay for 0.29 seconds

TCNT0 **=** 0**;** // Reset Counter

x**++;** // Over flow counter

**}** //Delay to keep LED off

PORTB **^=** **(**1**<<**2**);** //LED turns on

TCNT0 **=** 0**;** //Reset counter

x **=** 0**;** // Over flow counter reset

**while** **(**x **<=** 32**)** //TCNT to 6796

**{**

**while(**TCNT0 **!=** 206**){}** // Delay for 0.435 seconds

TCNT0 **=** 0**;** // Reset counter

x**++;** // Over flow counter

**}** //Delay to keep LED on

PORTB **^=** **(**1**<<**2**);** //LED turns off

**}**

/\*

\* task3b.c

\*

\* Created: 3/23/2019 7:09:29 PM

\* Author : Chris Barr

\*/

#include <avr/interrupt.h>

#include <avr/io.h>

#include <stdio.h>

int main**(**void**)**

**{**

DDRB **|=** **(**1**<<**5**)** **|** **(**1**<<**2**);** //Sets PB5/PB2 as outputs

PORTB **|=** **(**1**<<**5**)** **|** **(**1**<<**2**);** //Clears PB5/PB2

TCCR0B **=** 5**;** //Set prescaler to 1024

TCCR0A **=** **(**1 **<<** WGM01**);** //Set CTC Mode

TIMSK0 **=** **(**1 **<<** OCIE0A**);** //Set interrupt on compare match

sei**();**

OCR0A **=** 0xFF**;**

**while** **(**1**)** //loop forever to keep program running

**{**

**}**

**return** 0**;**

**}**

ISR **(**TIMER0\_COMPA\_vect**)**

**{**

int x **=** 0**;** // Used to counter overflow through the 8 bit timer0

**if** **(!(**PINC **&** **(**1 **<<** PINC2**)))** //If the pushbutton is pressed, go through this if statement

**{**

PORTB **&=** **~(**1**<<**2**);** //Keeps LED on until otherwise

**while** **(**x **<=** 76/\*TCNT0 != 19530\*/**)** //Delay to keep LED off

**{**

**while** **(**TCNT0 **!=** 253**);** // Delay for 1.25 seconds

TCNT0 **=** 0**;** // Reset counter

x**++;** // Over flow counter

**}**

**while(**TCNT0 **!=** 51**);** // Additional delay

TCNT0 **=** 0**;** //Reset counter

x **=** 0**;** // Over flow counter reset

**}**

**else** //else, if the pushbutton is not pressed, go through this else statement

**{**

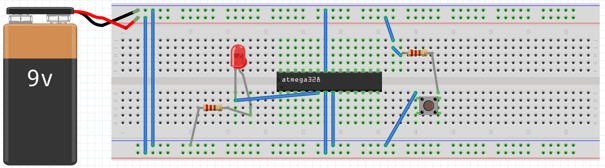
PORTB **|=** **(**1**<<**2**);** //Keeps LED off until otherwise

TCNT0 **=** 0**;** //Reset counter

**}**

**}**

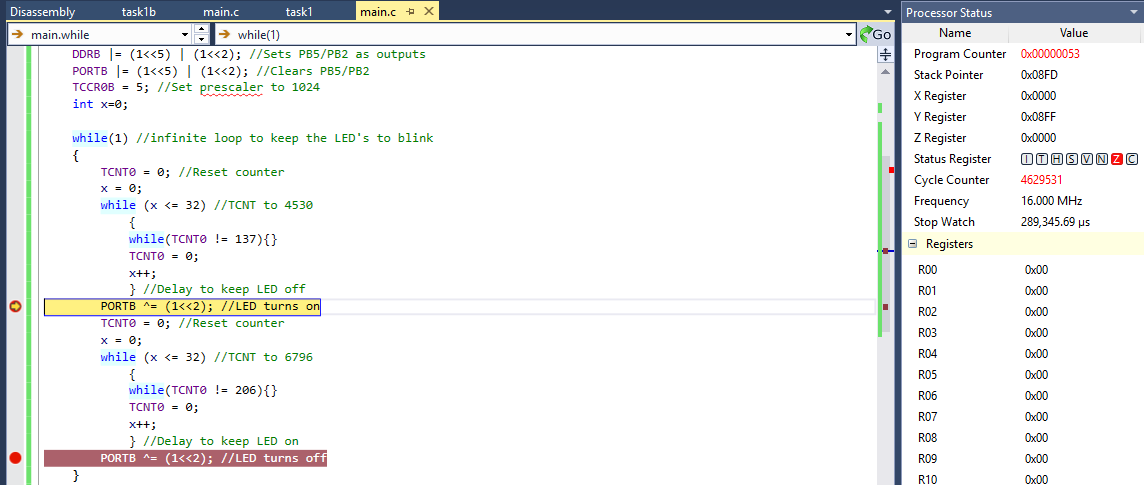
1. **SCHEMATICS**



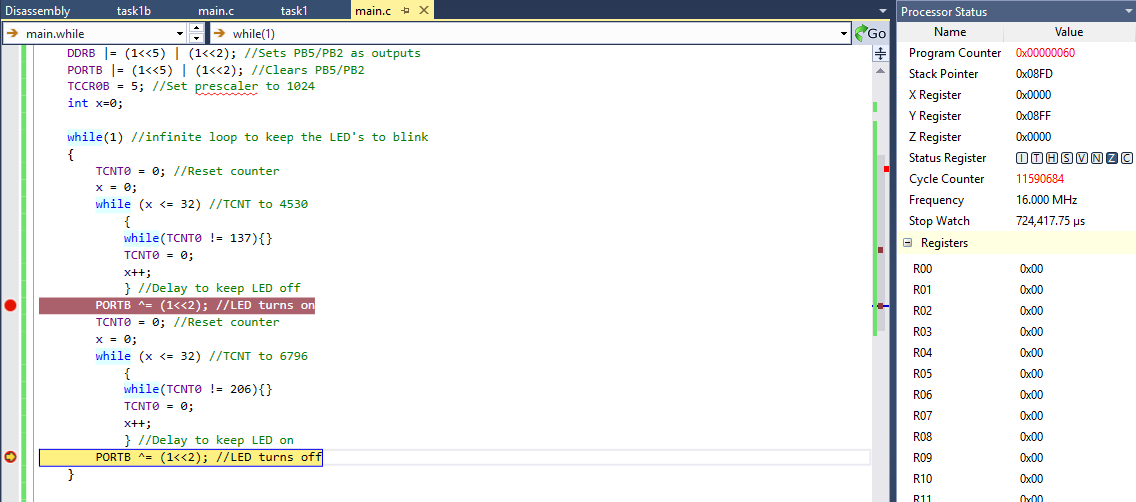
Note: It’s 5V not 9V, also assume the resistors/pushbutton/LED is the correct version on the board.

1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**

Task 1-3 part a:

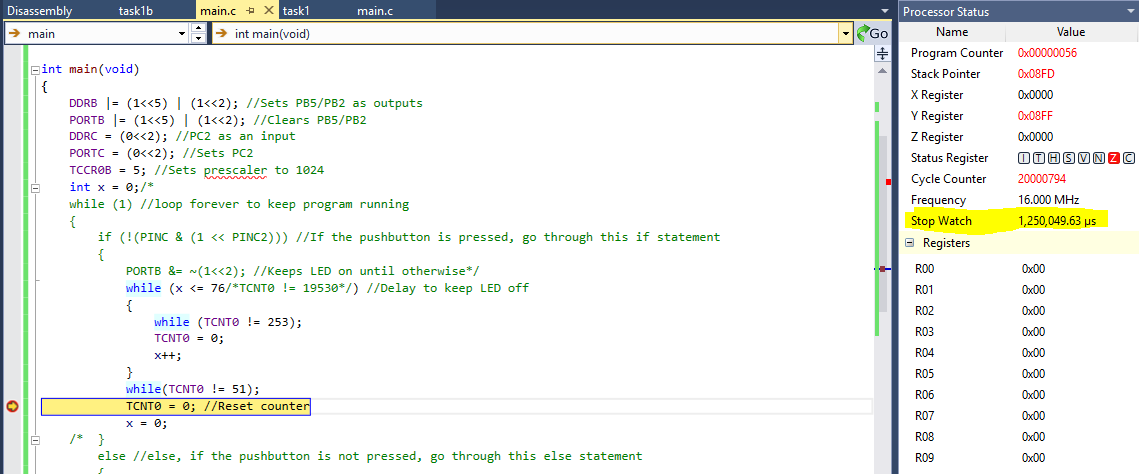


This shows the LED staying on for 290000 us (0.29 seconds) after going through the first delay loop.



This shows the LED staying off for 435000 us (0.435 seconds) after going through the second delay loop, but the stop watch shows the accumulation of 725000 us (0.725 seconds) which proves the LED’s stay on and off at the correct time.

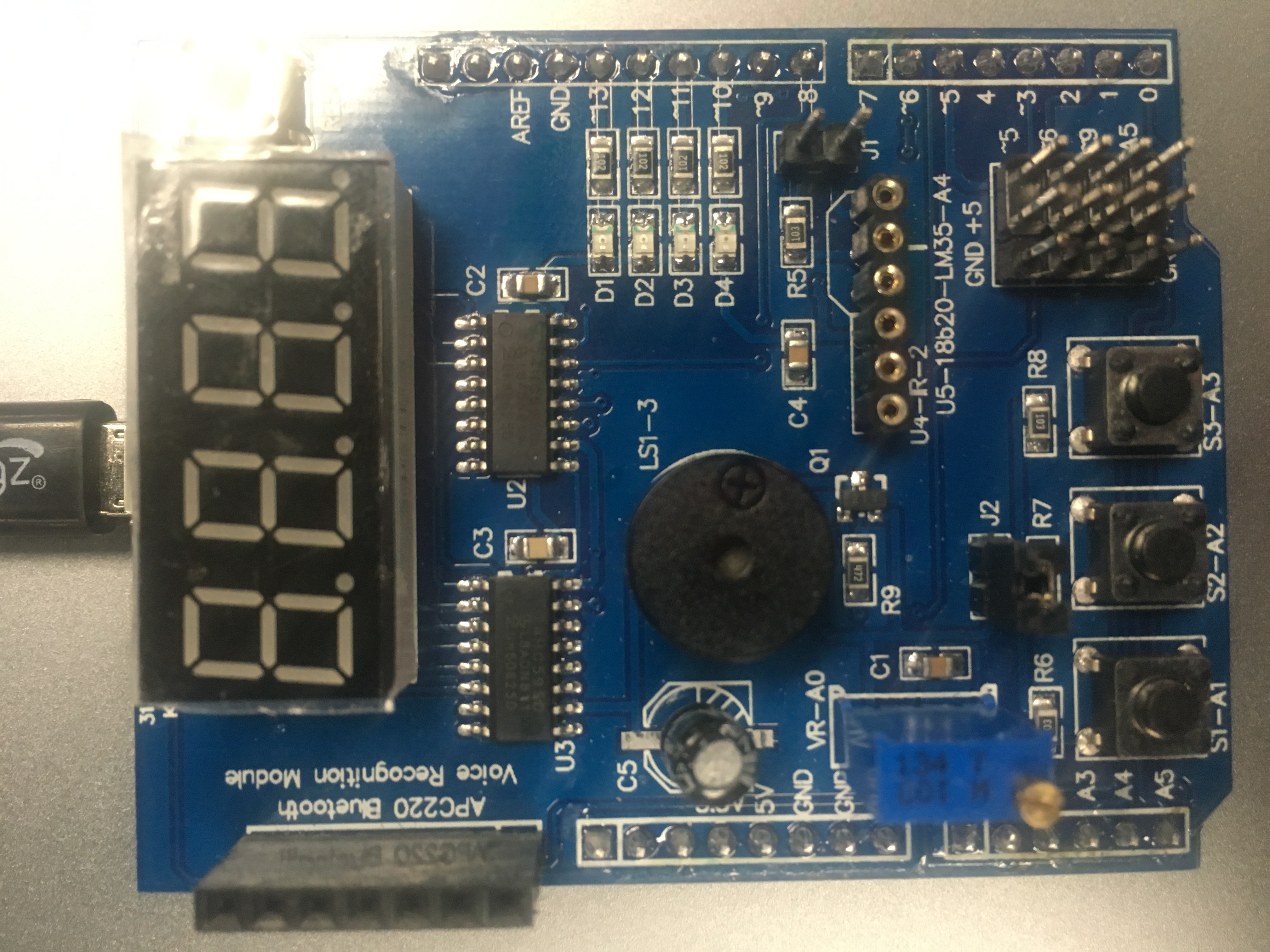
Task 1-3 part b:



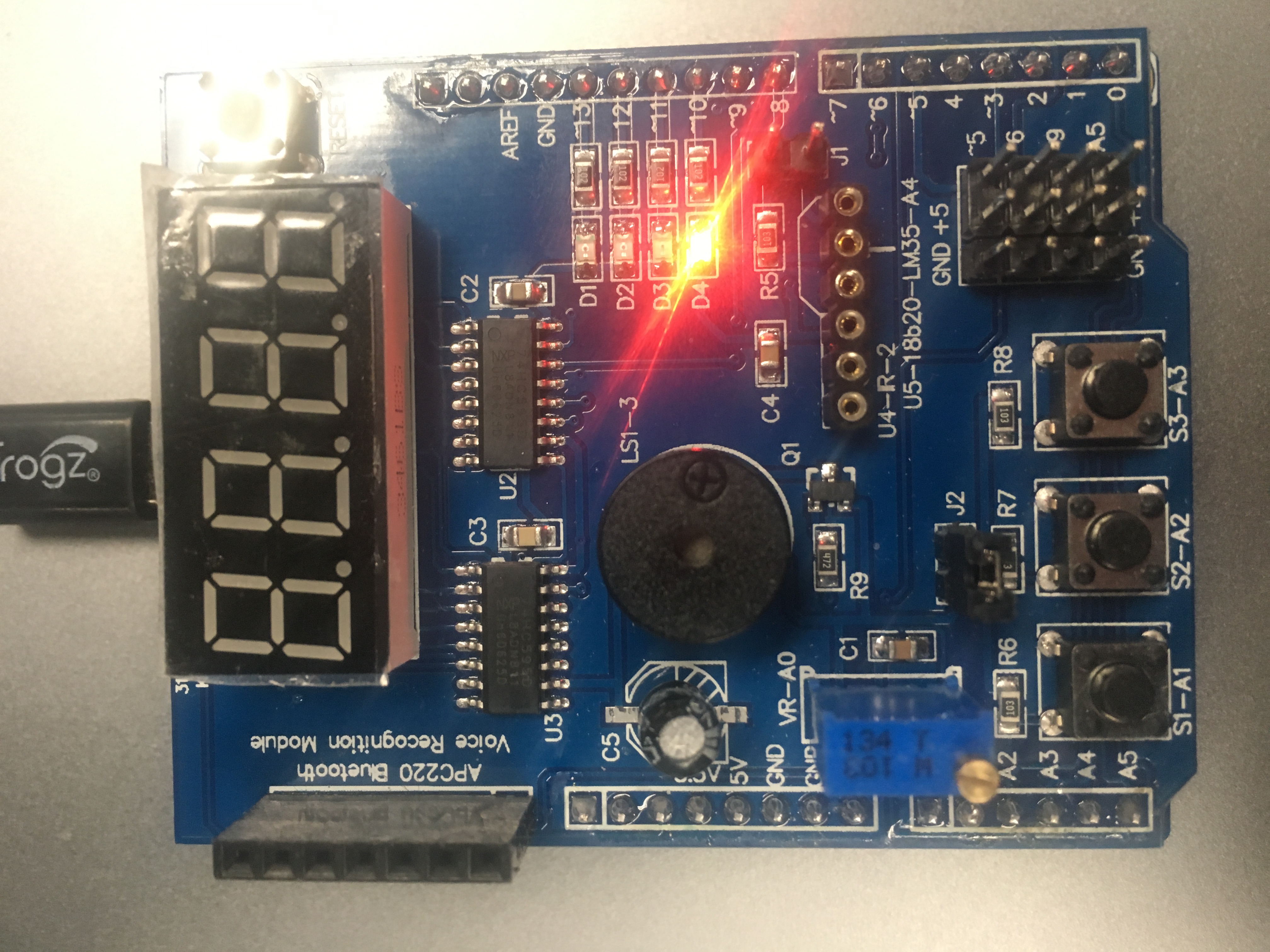
I implemented the code to just go straight through the delay loop to test the timing. The stop watch proves that 1250000 us (1.25 seconds) will pass, so it will go through the correct amount of delay.

**Note: ALL tasks roughly have the same code, which means the same delay in the simulation. Simulation is only testing the while loop (the delay) in order to be sure it’s the correct timing.**

1. **SCREENSHOT OF EACH DEMO (BOARD SETUP)**

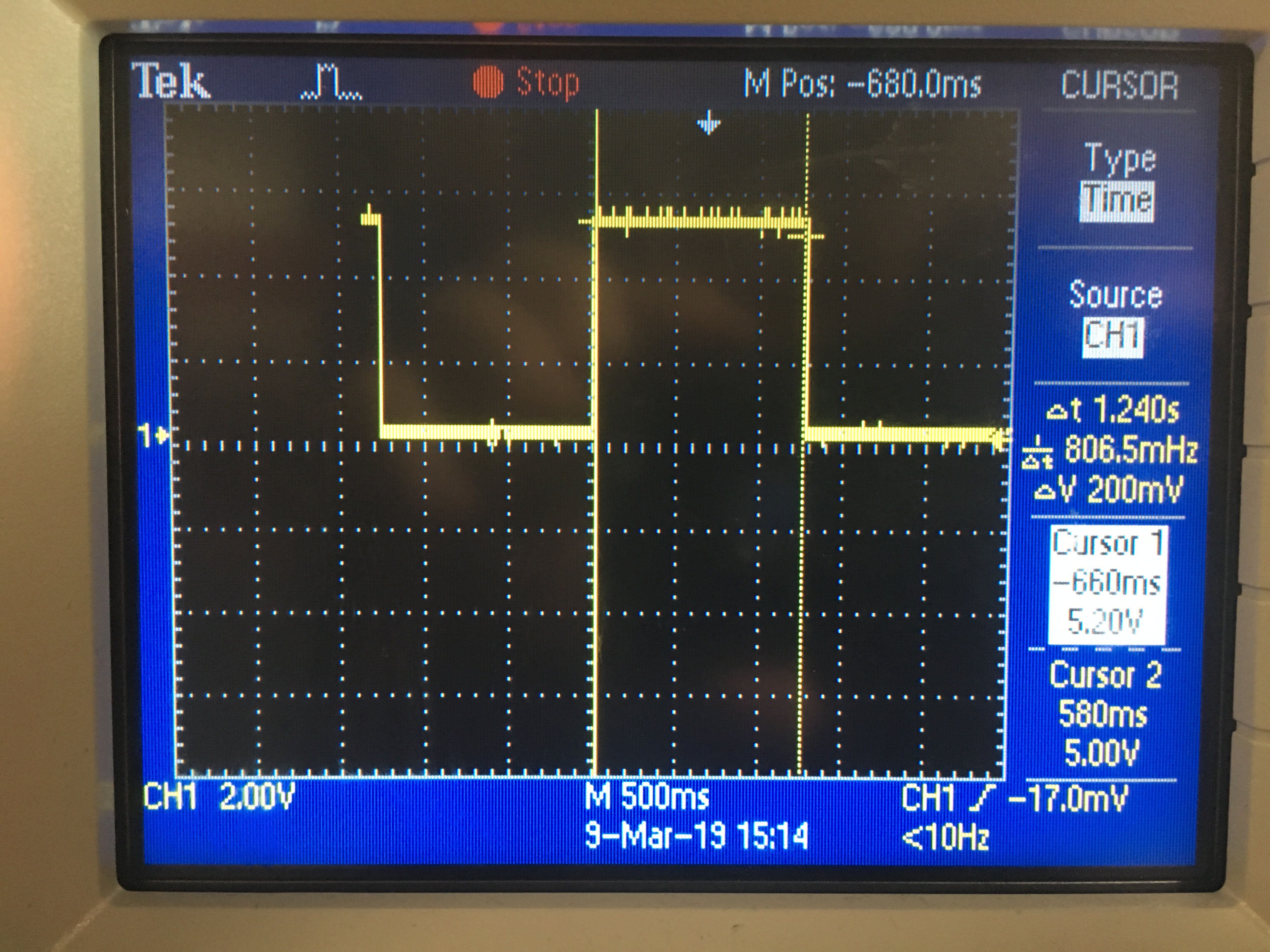


For all tasks, the LED will be turned off and then turned on at the specific D4 LED



Here’s where the LED will be turned on from either the blinking during task 1 or after the pushbutton is pressed during task 2

NOTE: Same results as DA2A, just different code setup.



Waveform for (Emulation) for 1.25 seconds on tasks 1-3 part b.

1. **VIDEO LINKS OF EACH DEMO**

<https://www.youtube.com/watch?v=YajvKW9my2U>

1. **GITHUB LINK OF THIS DA**

<https://github.com/BarrChris/submission_da/tree/master/DesignAssignments/DA2C>

**Student Academic Misconduct Policy**

<http://studentconduct.unlv.edu/misconduct/policy.html>

“This assignment submission is my own, original work”.

Chris Barr