CPE301 – SPRING 2019

Design Assignment 2B

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:

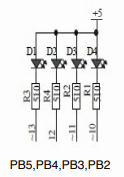
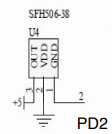
1. In the document, for each task submit the modified or included code (only) with highlights and justifications of the modifications. Also, include the comments.
2. Use the previously create a Github repository with a random name (no CPE/301, Lastname, Firstname). Place all labs under the root folder 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)
* M-M Wire

Block diagram with pins used in the Atmega328P



1. **INITIAL/MODIFIED/DEVELOPED CODE OF TASK 2/B**

AVR

;

; DA2A(part2\_avr).asm

;

; Created: 3/4/2019 6:42:16 PM

; Author : barrc1

;

; Replace with your application code

.CSEG

.org 0x0000

LDI R16**,** 0xFF ;Sets PB5 as output

**OUT** DDRB**,** R16

CBI DDRC**,** 2 ;Sets PC2 as input

LDI R20**,**5 ;set clock prescaler to 1024

STS TCCR1B**,** R20 ;send prescaler

**OUT** PORTB**,** R16

**LOOP:**

;Loops back here when nothing is pressed

SBI PORTB**,** 2 ;Keeps LED turned off

SBIC PINC**,** 2 ;Skips next instruction if pushbutton is pressed

RJMP **LOOP** ;If pushbutton isn't pressed, continues through this loop

;Goes here once button is pressed

LDI R18**,** 0b1110\_00 ;Turns on LED

**OUT** PORTB**,** R18

LDI R20**,** 0 ;Clears counter

STS TCNT1H**,** R20

STS TCNT1L**,** R20

RCALL DELAY ;Delay for 1.25 seconds

RJMP **LOOP** ;Loops back until pushbutton is pressed again

delay**:**

**LDS** R29**,** TCNT1H ;Loads upper bits into register R29

**LDS** R28**,** TCNT1L ;Loads lower bits into register R28

CPI R28**,**0x4A ;comparing lower 8 bits of timer to 0x8C

BRSH body ;if lower bits are the same or higher to 0x8C, jump to body and start counting there

RJMP delay ;else, keep checking lower bits

body**:**

CPI R29**,**0x4C ;comparing upper 8 bits of timer to 0x1A

BRLT delay ;If lower, then go back and check the lower bits

**RET** ;When the timer is over, return to turn off the lead

C Code

/\*

\* DA2A(part2\_ccode).c

\*

\* Created: 3/4/2019 7:04:53 PM

\* Author : barrc1

\*/

#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 OF TASK 2/B from TASK 2/B**

AVR

;

; DA2B.asm

;

; Created: 3/9/2019 12:14:34 PM

; Author : Chris

;

;.include<m328pdef.inc>

.ORG 0;location for reset

**JMP** MAIN

.ORG 0x02;location for external interrupt 0

**JMP** EX0\_ISR

MAIN**:**

LDI R20**,**HIGH**(**RAMEND**)**

**OUT** SPH**,**R20

LDI R20**,**LOW**(**RAMEND**)**

**OUT** SPL**,**R20 ; initialize stack

LDI R16**,** 0xFF

LDI R20**,**0x2;make INT0 falling edge triggered

STS EICRA**,**R20

SBI DDRB**,**2;PORTB.5 = output

SBI PORTD**,**2;pull-up activated

LDI R20**,**1**<<**INT0;enable INT0

**OUT** EIMSK**,**R20

SEI;enableinterrupts

HERE**:**

**OUT** PORTB**,** R16

RJMP HERE

EX0\_ISR**:**

**IN** R21**,**PORTB

LDI R22**,(**1**<<**2**)** ; 00100000 for toggling PB5

EOR R21**,**R22

**OUT** PORTB**,**R21

RCALL DELAY

RETI

DELAY**:**

LDI R20**,** 125

D0**:**

LDI R21**,** 125

D1**:**

LDI R22**,** 255

D2**:**

**NOP**

**NOP**

**DEC** R22

BRNE D2

**DEC** R21

BRNE D1

**DEC** R20

BRNE D0

**RET**

C Code

/\*

\* DA2B\_CCODE.c

\*

\* Created: 3/9/2019 3:08:37 PM

\* Author : Chris

\*/

#define F\_CPU 16000000UL

#include <avr/io.h>

#include <avr/interrupt.h>

#include <util/delay.h>

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

**{**

//Port Direction/Set-up

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

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

PORTD **=** 1**<<**2**;** //Enables PD2

//Interrupts

EICRA **=** 0x02**;**

EIMSK **=** 1**<<**INT0**;** //Enable PC set 0 interrupt

sei**();** //Enables global interrupt

**while** **(**1**);**

**}**

ISR**(**INT0\_vect**)**

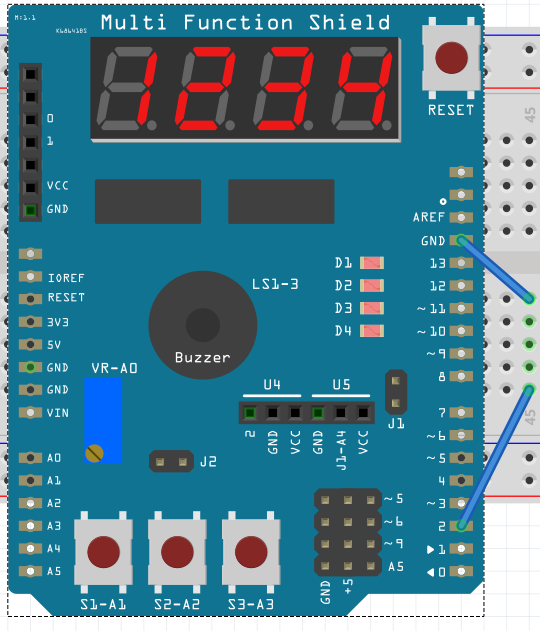
**{**

PORTB **^=** **(**1**<<**2**);** //Toggle PB2 when interrupt

\_delay\_ms**(**1250**);** //Delays for 1.25 seconds

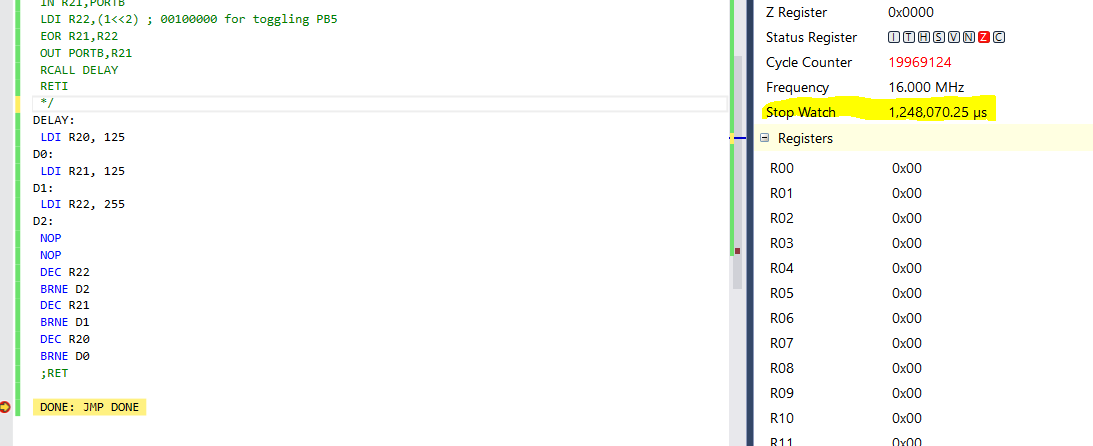
**}**

1. **SCHEMATICS**



PD2 touching ground will turn on LED D4 (PB2)

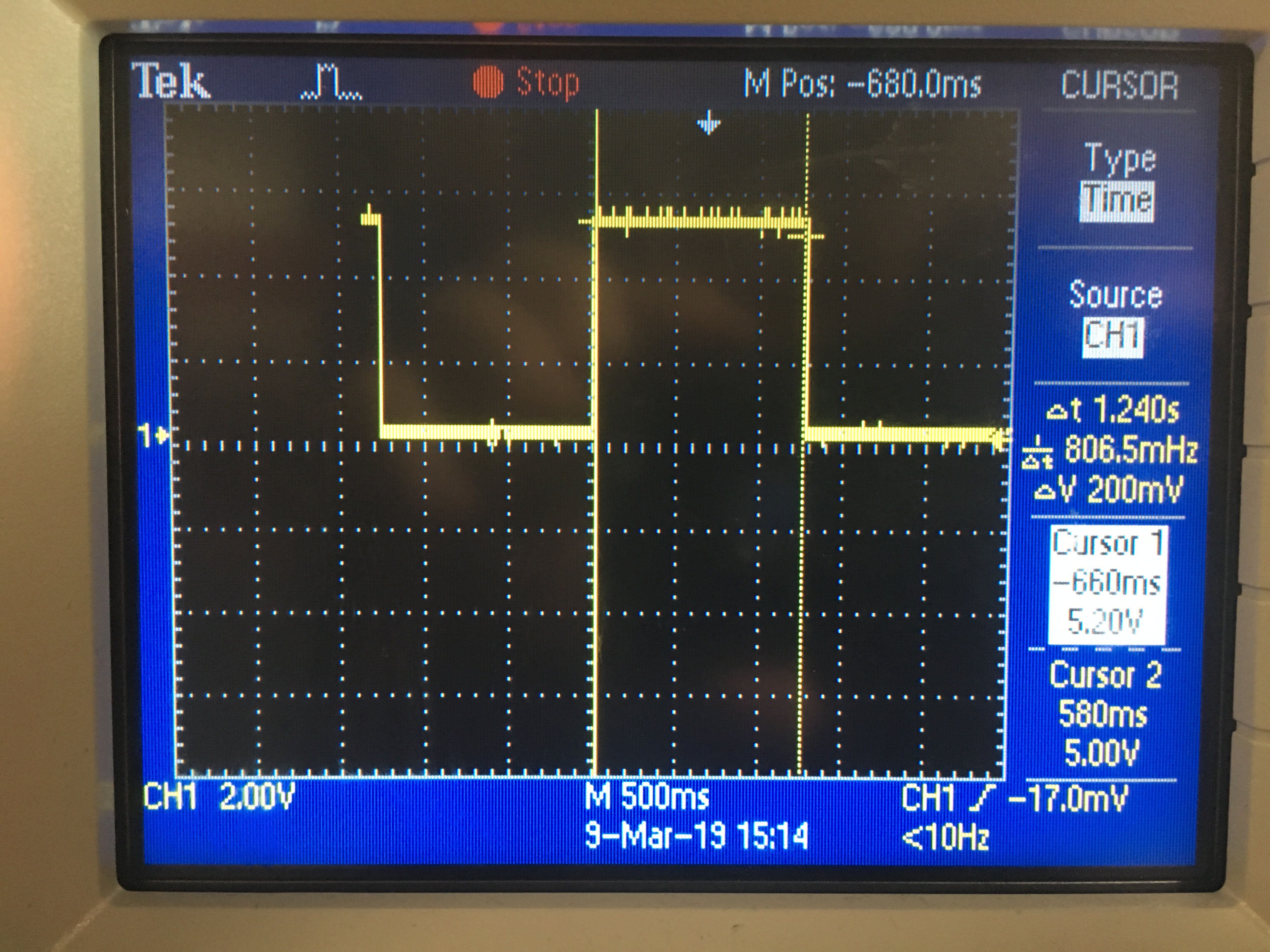
1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**



AVR Simulation for the delay to check 1.25 seconds

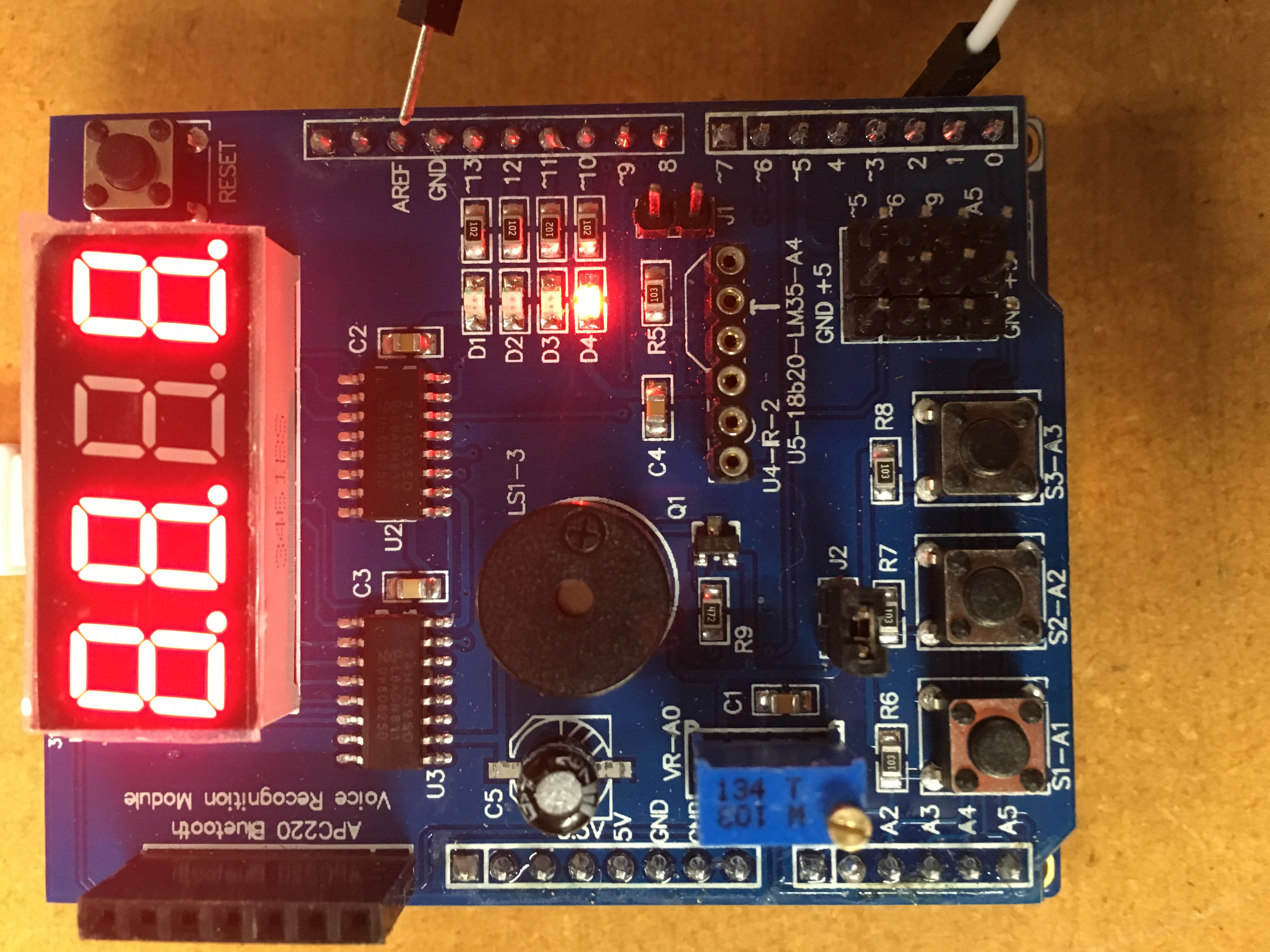


C Code Simulation for the delay to check 1.25 seconds



Emulation to check if 1.25 seconds (note: it’s 1.25 seconds, the knob wouldn’t move 0.01 second more)

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



Output is essentially the same for either C Code or AVR. PD2 touches ground causes an interrupt to turn on PB2 (LED) as shown above. Proof shown in video.

1. **VIDEO LINKS OF EACH DEMO**

<https://www.youtube.com/watch?v=E3yYEjTtbPM>

1. **GITHUB LINK OF THIS DA**

<https://github.com/BarrChris/submission_da/tree/master/DesignAssignments/DA2B/DA2B>

**Student Academic Misconduct Policy**

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

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

Chris Barr