CPE301 – SPRING 2019

Design Assignment X

Student Name: Jett Guerrero

Student #: 5002377713

Student Email: guerrj1@unlv.nevada.edu

Primary Github Address: https://github.com/guerrj1/Submission\_DA.git

Directory: https://github.com/guerrj1/Submission\_DA/tree/master/DA2A

Submit the following for all Labs:

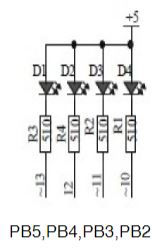
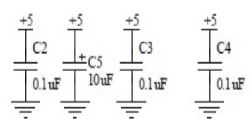
1. In the document, for each task submit the modified or included code (only) with highlights and justifications of the modifications. Also, include the comments.
2. Use the previously create a Github repository with a random name (no CPE/301, Lastname, Firstname). Place all labs under the root folder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

-ATmega328P

-Arduino Multi-Function Shield

-Micro-USB cable

PB2 and PC2 PINS were used on the Atmega328P and Arduino Shield

1. **DEVELOPED CODE OF TASK 1**

Task 1 Assembly Code

;DA2AT1A

.INCLUDE "M328pDEF.INC"

.CSEG

.org 0x0000

SBI DDRB, 5 ;makes PB5 an output

SBI PORTB,5 ;turns off LED for PB5

SBI DDRB, 2 ;makes PB2 an output

LDI R20,5 ;set clock prescaler to 1024

STS TCCR1B,R20

begin:

LDI R20,0x00 ;resetting the counter to 0

;ON

SBI PORTB, 2 ;sets LED PB2 on

STS TCNT1H,R20

STS TCNT1L,R20

RCALL delay\_on ;calling timer to wait for 0.435 sec

;OFF

CBI PORTB, 2 ;clears LED PB2 off

STS TCNT1H,R20

STS TCNT1L,R20

RCALL delay\_off ;calling timer to wait for 0.29 sec

RJMP begin ;repeat main loop

;Delay for LED on-----------

delay\_on:

LDS R29, TCNT1H ;loading upper bit of counter to R29

LDS R28, TCNT1L ;loading lower bit of counter to R28

CPI R28,0x8C ;comparing if lower 8 bits of timer is 0x8c - lower bits of 1A8C

BRSH body\_on ;if lower bits of timer have reached desired amount, check the upper bits

RJMP delay\_on ;otherwise, keep checking lower bits

body\_on:

CPI R29,0x1A ;check to see if upper timer bits have reached the desired value

BRLT delay\_on ;if not, recheck the lower bits

RET ;once the timer reached the desired value, toggle the LED

;Delay for LED off------------

delay\_off:

LDS R29, TCNT1H ;loading upper bit of counter to R29

LDS R28, TCNT1L ;loading lower bit of counter to R28

CPI R28,0xB2 ;comparing if lower 8 bits of timer is 0xB2 - lower bits of 11B2

BRSH body\_off ;if lower bits of timer have reached desired amount, check the upper bits

RJMP delay\_off ;otherwise, keep checking lower bits

body\_off:

CPI R29,0x11 ;check to see if upper timer bits have reached the desired value

BRLT delay\_off ;if not, recheck the lower bits

RET ;once the timer reached the desired value, toggle the LED

Task 1 C Code

//DA2AT1C.c

#include <avr/io.h>

#include <stdio.h>

int main(void)

{

DDRB = (1<<2); //sets pb2 as an input

DDRB = (0<<5); //sets pb5 as an input

TCCR1B = 5; //sets the prescaler of 1024

while(1)

{

TCNT1 = 0; //t count

while (TCNT1 != 4530) //while tcount is not equal to the calculated tcnt1

{} //creates the delay

PORTB ^= (1<<2); //sets pb2 led to on

TCNT1 = 0; //resets count to 0

while (TCNT1 != 6796) //while tcount is not euqal to the calculated tcnt1 value

{} //set delay

PORTB ^= (1<<2); //exclusive OR the previous value to toggle

}

return 0;

}

1. **DEVELOPED FOR TASK 2**

Task 2 Assembly Code

; DA2AT2A.asm

.CSEG

.org 0x0000

SBI DDRB, 5 ;makes PB5 an output

SBI PORTB,5 ;turns of LED for PB5

CBI DDRC, 2 ;makes PC2 an input

LDI R17,0 ;used to set or reset PB5

SBI DDRB, 2 ;makes PB2 an output

LDI R20,5 ;set clock prescaler to 1024

STS TCCR1B, R20

check:

SBIC PINC, 2 ;skip next instruction if PC2 is clear

RJMP waiting ;jumps to waiting label for waiting for a pressed input

CBI PORTB, 2 ;clears PB2

STS TCNT1H, R20

STS TCNT1L, R20

RCALL delay\_on ;jumping for LED on delay

RJMP check ;jump back to check

;waiting for input from pushbutton

waiting:

SBI PORTB, 2 ;sets PB2 on

RJMP check ;relative jump back to check label

;delay for LED on

delay\_on:

LDS R29, TCNT1H ;loading upper bit of counter to R29

LDS R28, TCNT1L ;loading lower bit of counter to R28

CPI R28,0x4A ;comparing if lower 8 bits of timer is 0x08

BRSH body\_on ;if lower bits of timer have reached desired amount, check the upper bits

RJMP delay\_on ;otherwise, keep checking lower bits

body\_on:

CPI R29,0x4C ;check to see if upper timer bits have reached the desired value

BRLT delay\_on ;if not, recheck the lower bits

RET ;once the timer reached the desired value, toggle the LED

Task 1 C Code

//DA2AT2C.c

#define *F\_CPU* 16000000UL //sets CPU clock speed

#include <avr/io.h>

#include <util/delay.h>

int main(void)

{

DDRB |= (1<<2); //sets pb2 as input

PORTB |= (1<<2); //sets pb2 to high for LED

DDRC &= (0<<2); //sets PC2 to input

PORTC |= (0<<2); //sets PC2 to low 0

while (1) //while loop continuously checking

{

if (!(PINC & (1 << PINC2))) //if pinc2 was pressed then released

{

PORTB &= ~(1<<2); //then portb2 led is on

*\_delay\_ms*(1250); //delays for 1.250 secs while on

}

else

{

PORTB |= (1<<2); //toggles pb2 led

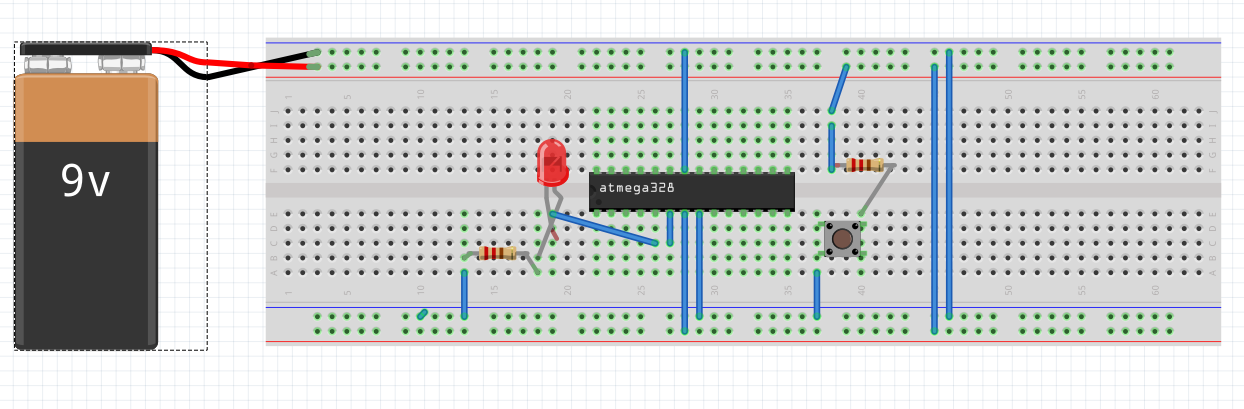
}

}

return 0;

}

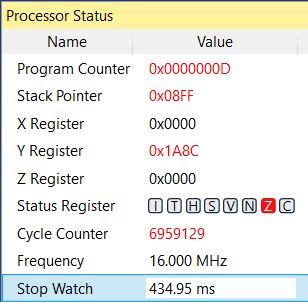
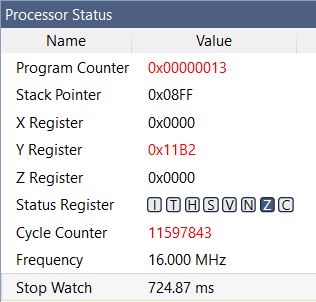
1. **SCHEMATICS**



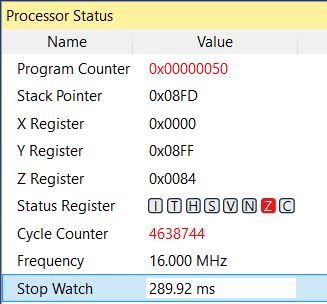
Breadboard Schematic using Fritzing

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

Task 1 Duty Cycle and Period Verification

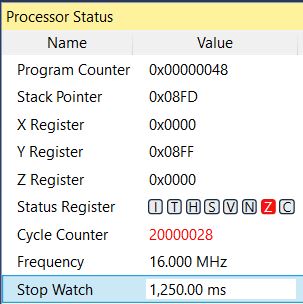
** **

60% Duty Cycle 434.95ms ≈ 0.435s Total Period 724.87ms ≈  0.725s

****

40% Duty Cycle 289.92ms ≈ 0.29s

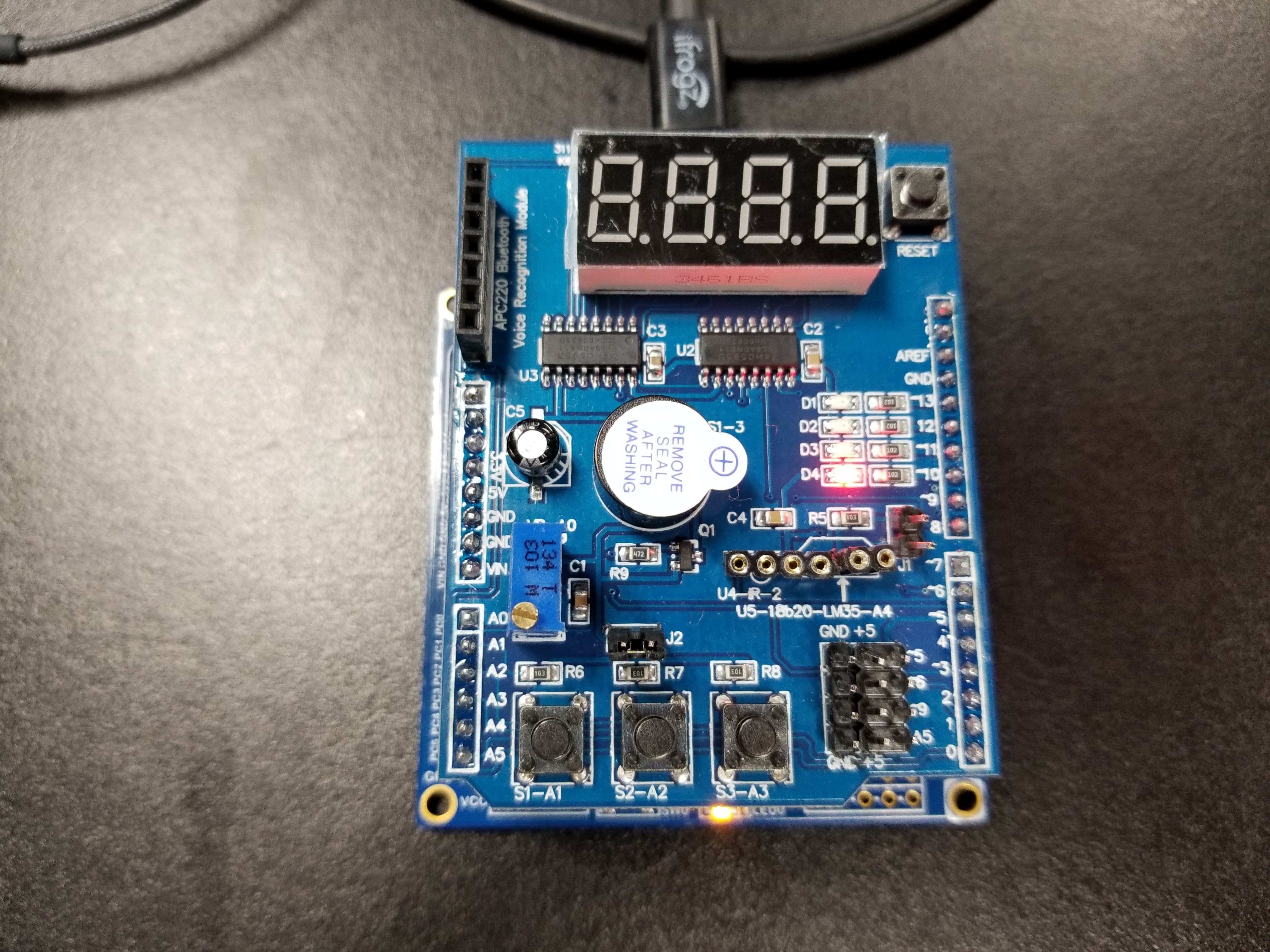
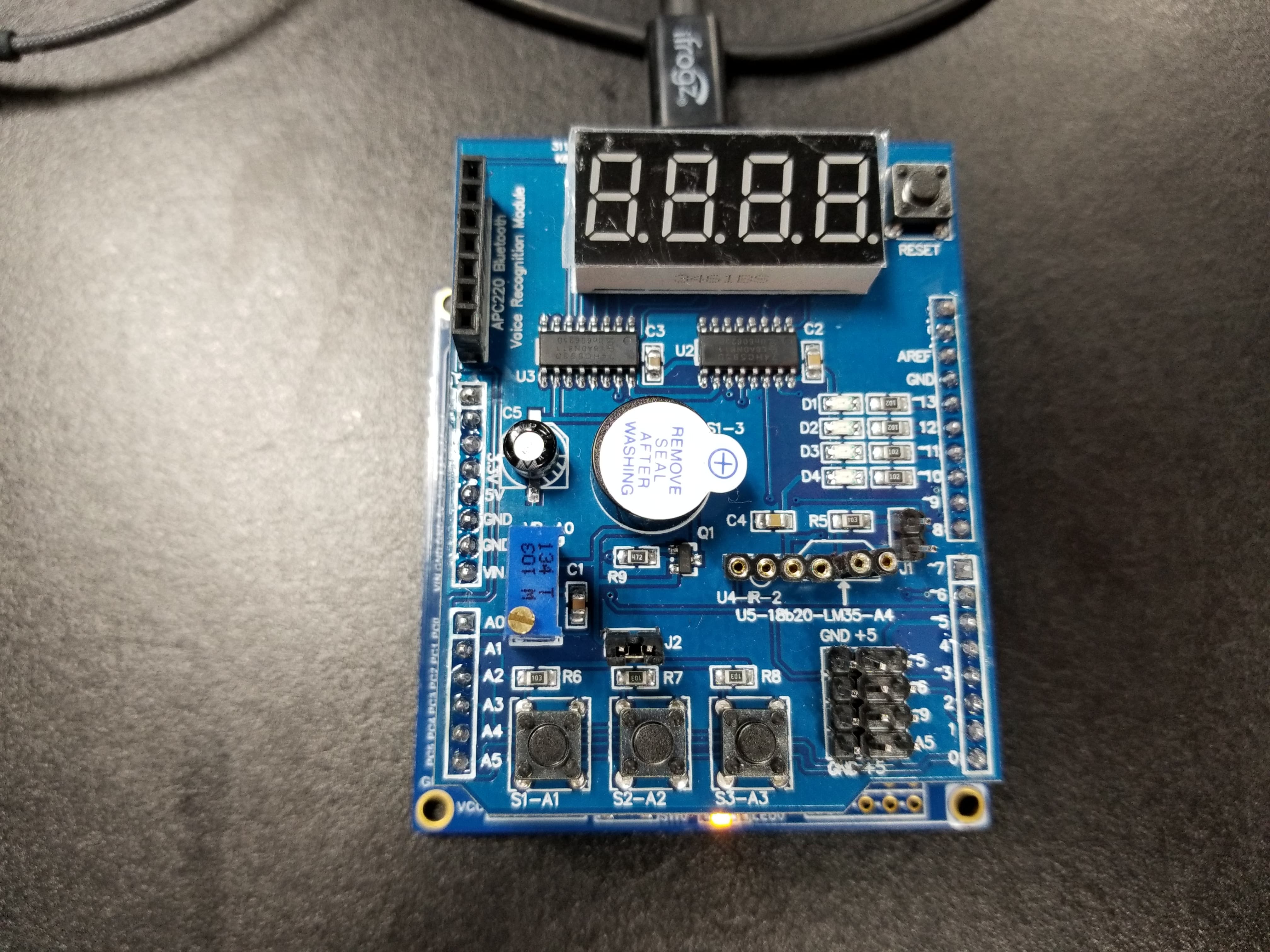
Task 2 Duty Cycle and Period Verification

****

LED ON Delay for 1,250ms ≈ 1.250s

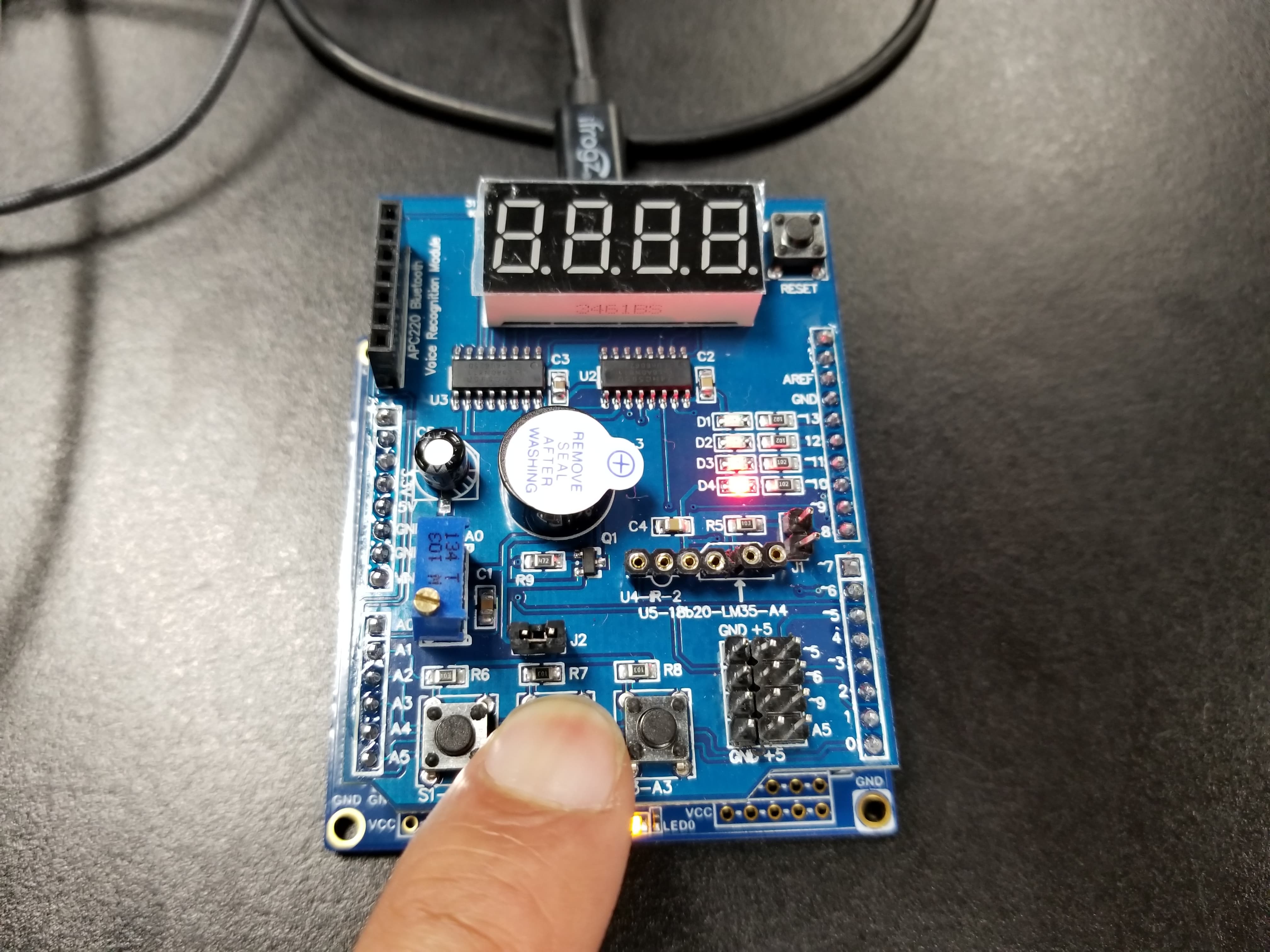
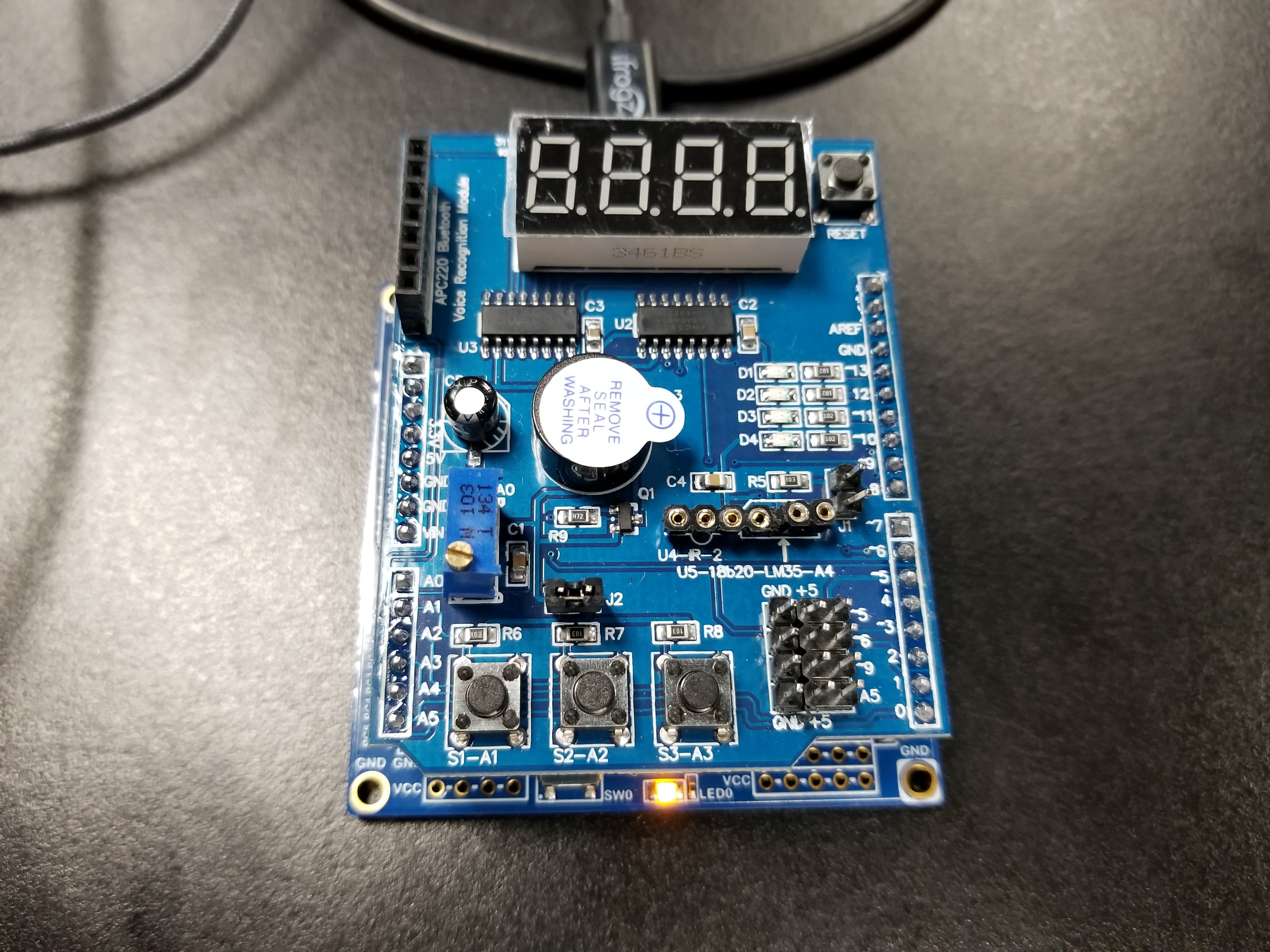
1. **SCREENSHOT OF EACH DEMO (BOARD SETUP)**

Task 1

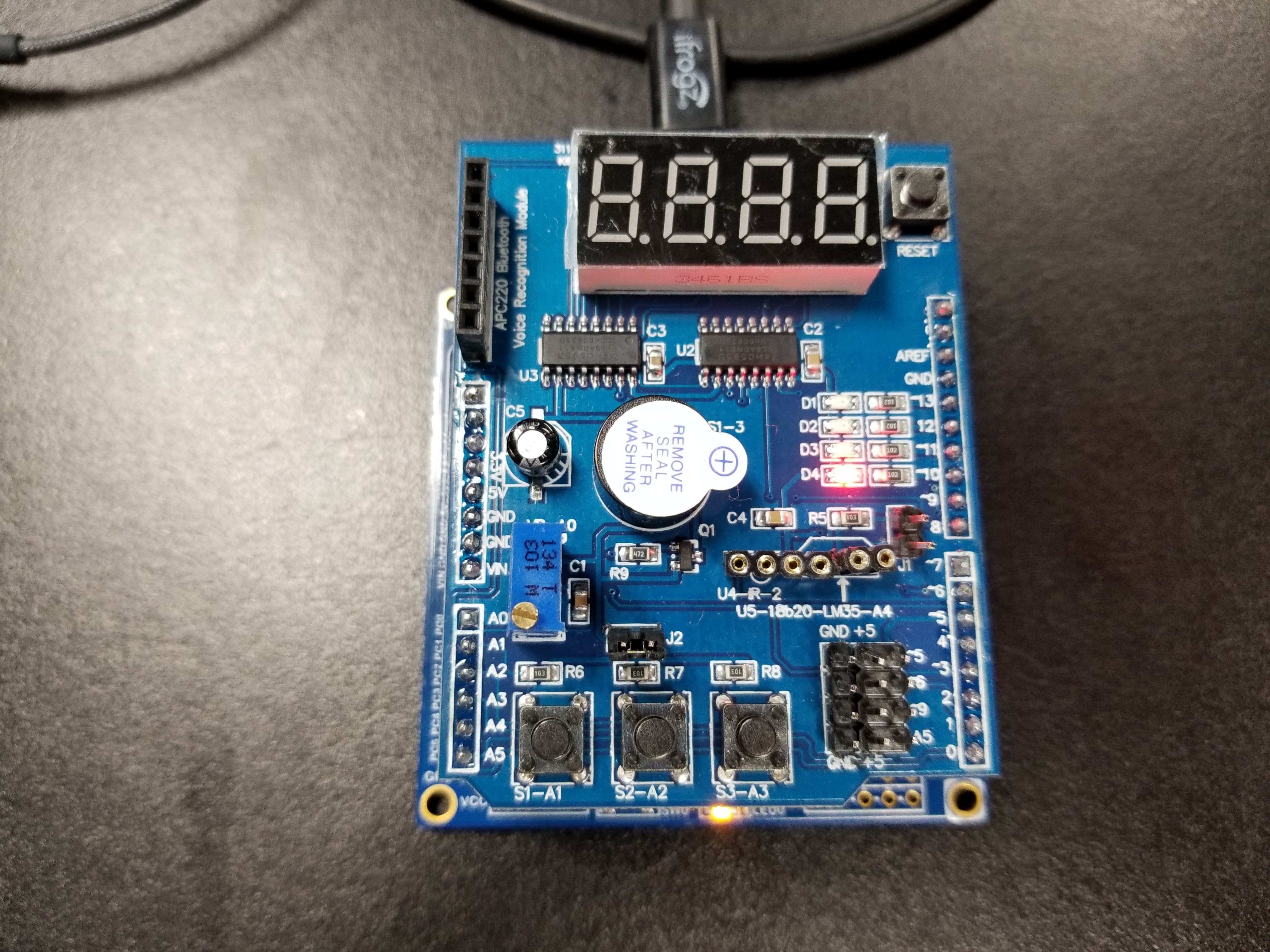


LED OFF LED ON

Task 2



LED OFF LED Turns OFF After Push Button is Pressed



LED Stays ON for the 1.250s

1. **VIDEO LINKS OF EACH DEMO**

Task 1 Assembly Demo

<https://youtu.be/Jb8sLYPLfoo>

Task 1 C Demo

<https://youtu.be/pDwe_vSzwnU>

Task 2 Assembly Demo

<https://youtu.be/B7cBseegcOA>

Task 2 C Demo

<https://youtu.be/yB9F5vjMUPY>

1. **GITHUB LINK OF THIS DA**

https://github.com/guerrj1/Submission\_DA/tree/master/DA2A

**Student Academic Misconduct Policy**

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

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

Jett Guerrero