CPE301 – SPRING 2023

Design Assignment 2

Student Name: David Lenzin
Student #: 2001654470
Student Email: lenzin@unlv.nevada.edu
Primary Github address: https://github.com/dlenzin15/submissions
Directory: submissions/DA2

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

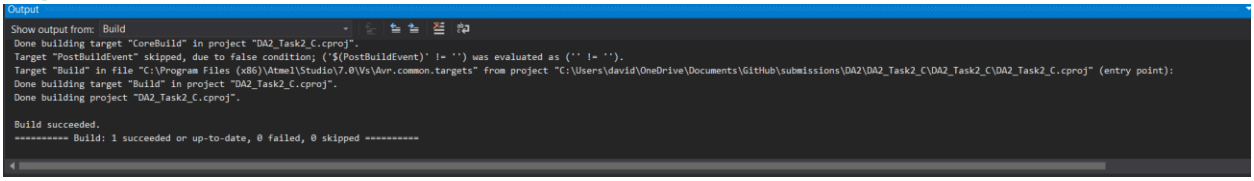
Atmega328PB board was used. No other hardware was introduced. See schematic below for pin layout.

2. INITIAL/MODIFIED/DEVELOPED CODE OF TASK 2

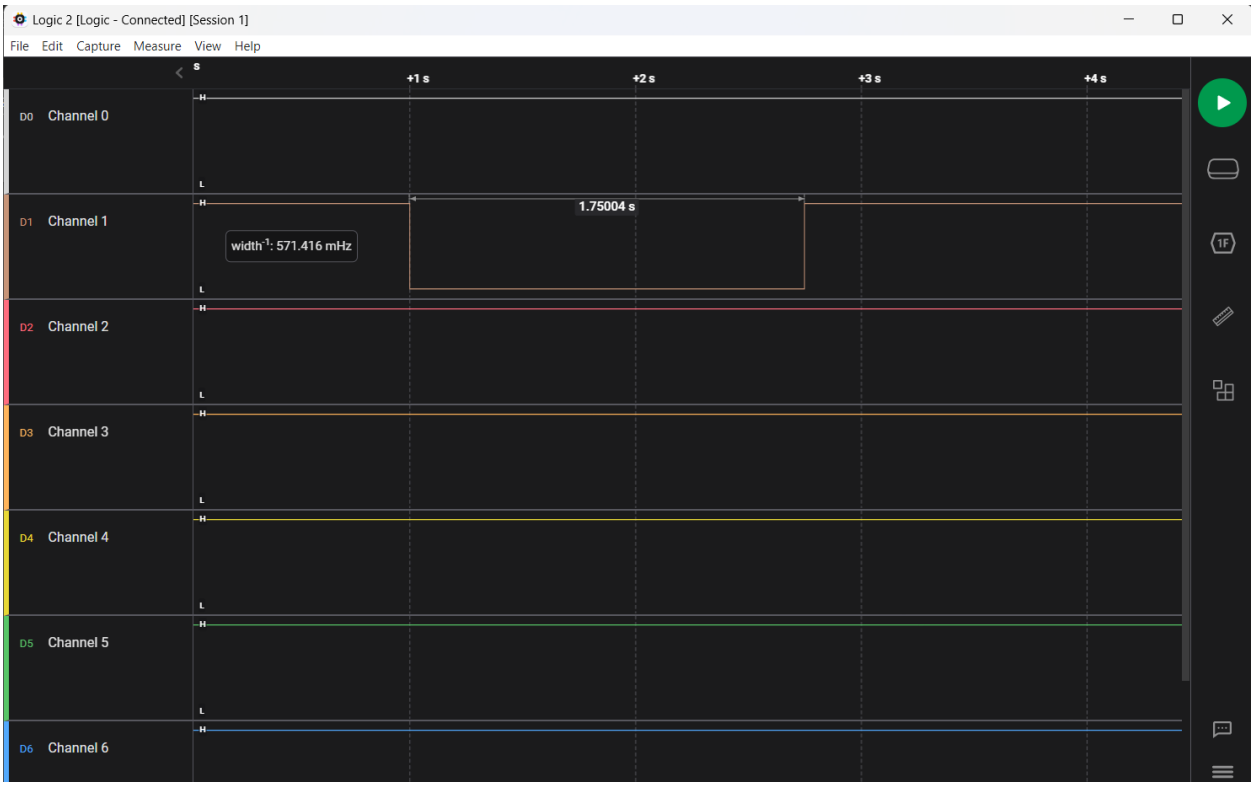
Insert initial code here:
C Code:

```c
/*
 * DA2_Task2_C.c
 *
 * Created: 3/6/2023 7:29:13 PM
 * Author : david
 */

#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>


int main(void)
{
    DDRC &= ~(1<<1); //Set PC5 to an input, which is connected to switch 1
       PORTC |= (1<<1);     //Pull-up resistor

       DDRB |= (1<<5); //Set PB5 as an output, which is connected to LED
       PORTB |= (1<<5); //Initially turn LED off
    while (1)
    {
              if (!(PINC & (1 << 1))) {   //If switch is pressed
                     PORTB &= ~(1 << 5);  //Turn on LED
                     _delay_ms(1750);     //Delay for 1.75 seconds
                     PORTB |= (1 << 5);   //Turn off the LED
              }
              PORTB |= (1 << 5);   //Keep the LED to off if button not pressed
    }
}
```

Compilation:



Waveform:

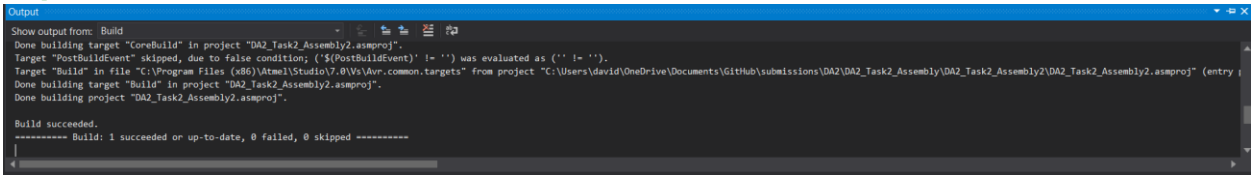


Assembly Code:

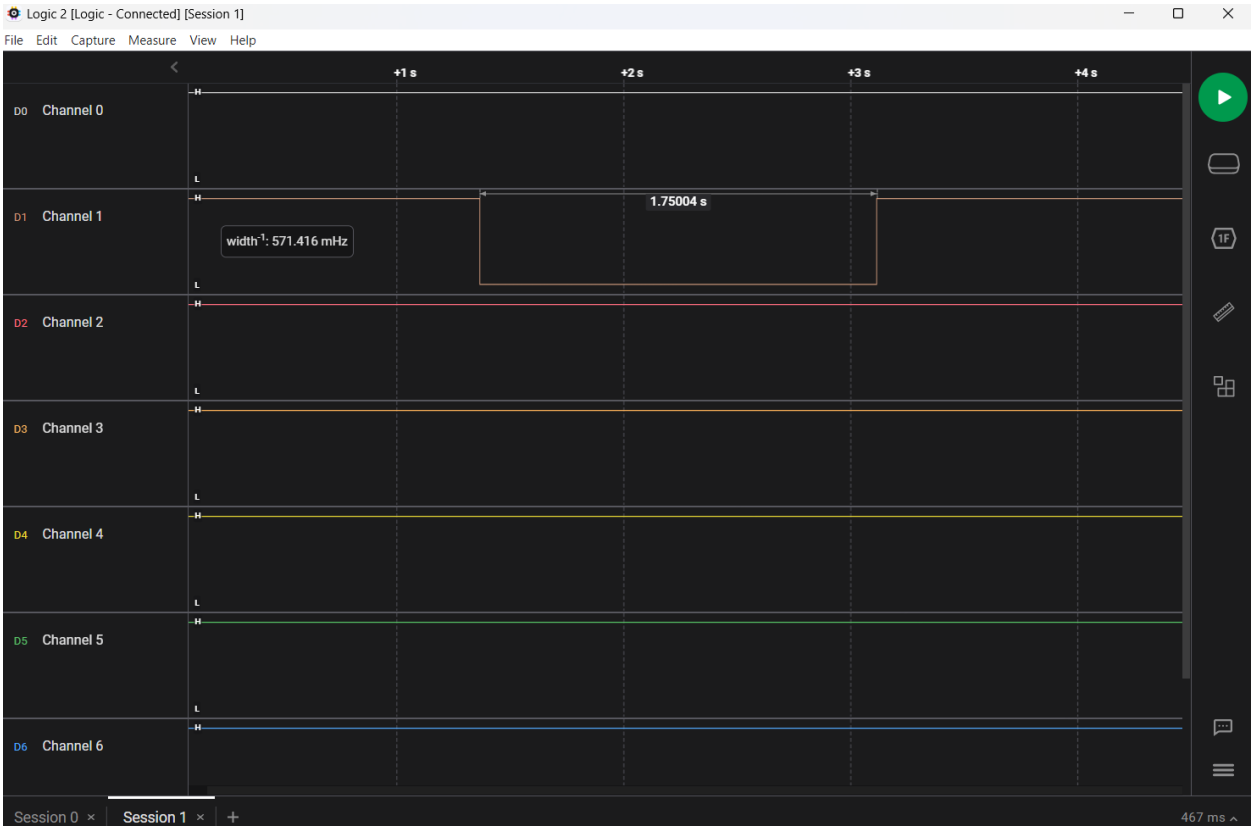
```
;
; DA2_Task2_Assembly.asm
;
; Created: 3/6/2023 9:39:36 PM
; Author : david
;

.include "m328pbdef.inc"    ; Include the header file for the Atmega328PB board

ldi r16, (1<<5)
out DDRB, r16                       ; Set PB5 as an output
out PORTB, r16                              ; Initially turn off the LED

loop:
       sbis PINC, 1                 ; check if switch at PC1 is pressed
    rjmp led_on                             ; jump to button_pressed if it is
       rjmp loop                            ; Loop forever

led_on:
       ldi r16, ~(1<<5)             ; Turn on the LED
       out PORTB, r16
       rjmp delay                           ; Jump to delay function

delay:                                      ; Delay function to generate a delay of 1.75
seconds
    ldi  r21, 143
    ldi  r22, 12
    ldi  r23, 66
L1: dec  r23
```

```
    brne L1
    dec  r22
    brne L1
    dec  r21
    brne L1

       ldi r16, (1<<5)                  ; Turn off the LED
       out PORTB, r16
       rjmp loop                        ; Return to loop
```

Compilation:



Waveform:



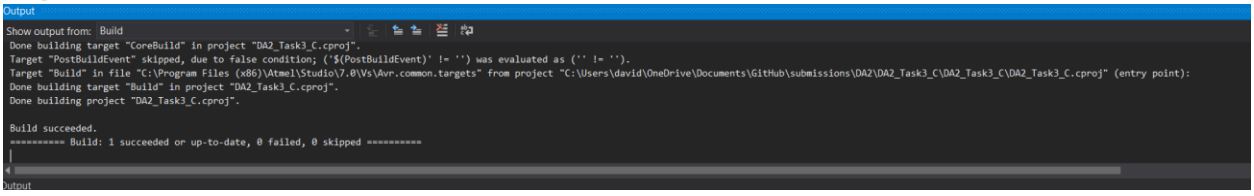
3. DEVELOPED MODIFIED CODE OF TASK 3

Insert only the modified sections here

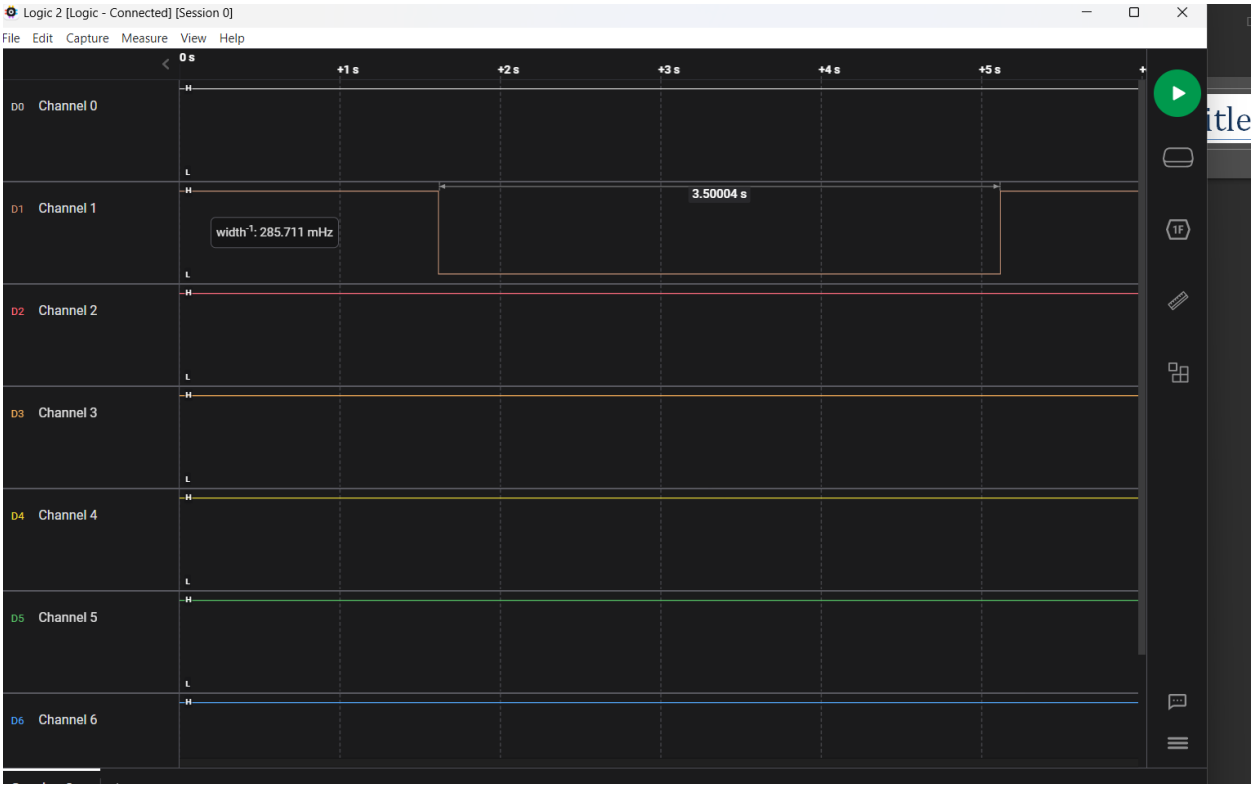
C Code:

```c
/*
 * DA2_Task3_C.c
 *
 * Created: 3/6/2023 8:36:47 PM
 * Author : david
 */

#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>


ISR(INT0_vect) {
	PORTB &= ~(1<<5);	//Turn LED on
	_delay_ms(3500);	//Delay for 3.5 seconds
	PORTB |= (1<<5);	//Turn LED off
}

int main(void)
{
	PORTD |= (1<<2);	//Activate Pull-up resistor for INT0 pin

	DDRB |= (1<<5); //Set PB5 as an output, which is connected to LED
	PORTB |= (1<<5); //Initially turn LED off

	EIMSK = (1 << INT0); //Enable interrupts on external pin INT0
	EICRA = 0x03; //The rising edge of INT0 generate an interrupt request
	sei(); //Enable interrupts

	while (1);
}
```

Compilation:



Waveform:

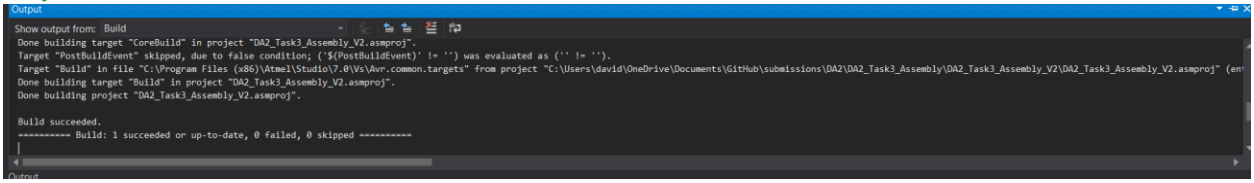


Assembly:

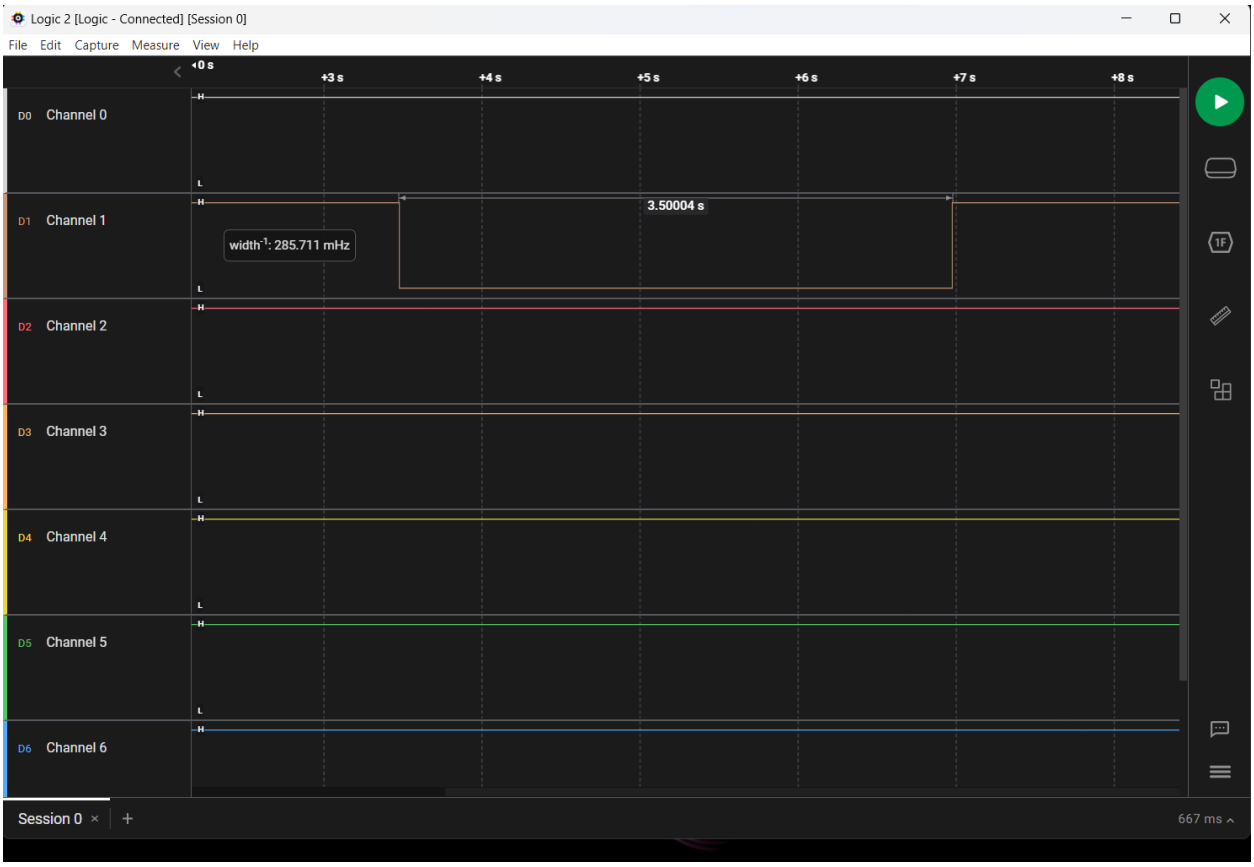
```
;
; DA2_Task3_Assembly.asm
;
; Created: 3/7/2023 9:04:30 PM
; Author : david
;

.equ F_CPU = 16000000
.include "m328pbdef.inc"          ;Include the header file

.org 0x0000
jmp main

.org 0x0002                                ; External interrupt request 0 vector
jmp INT0_ISR

main:
       ldi r16, (1<<5)
       out DDRB, r16                    ; Set PB5 as an output
       out PORTB, r16                          ; Initially turn off the LED

       ldi r17, (1<<2)
       OUT PORTD, r17                          ; Pull up for INTO pin

       ldi r18, (1<<INT0)               ; load the bit mask for INT0 into r16
       out EIMSK, r18                          ; enable interrupts on external pin
INT0

       ldi r18, 0x03                    ; load the value 0x03 into r16
```

```
	sts EICRA, r18 ; set the rising edge of INT0 to
generate an interrupt request

	sei ; enable interrupts globally

	loop:
		rjmp loop ; Infinite Loop

INT0_ISR:
	ldi r16, ~(1<<5) ; Turn on the LED
	out PORTB, r16

	; Delay loops to generate a delay of 3.5 seconds
    ldi  r21, 2
    ldi  r22, 29
    ldi  r23, 23
    ldi  r24, 133
	   L1: dec  r24
		brne L1
		dec  r23
		brne L1
		dec  r22
		brne L1
		dec  r21
		brne L1

		ldi r16, (1<<5) ; Turn off the LED
		out PORTB, r16
		reti ; Return from Interrupt
```

Compilation:

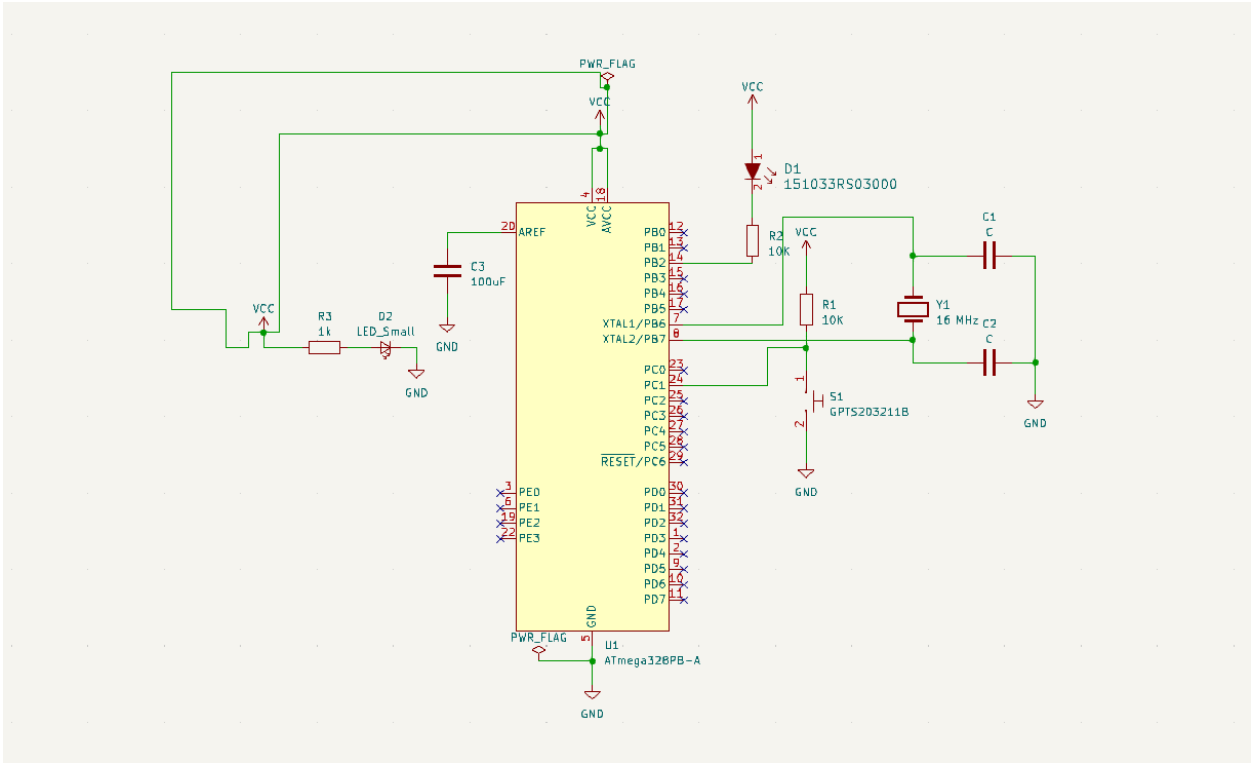


Waveform:

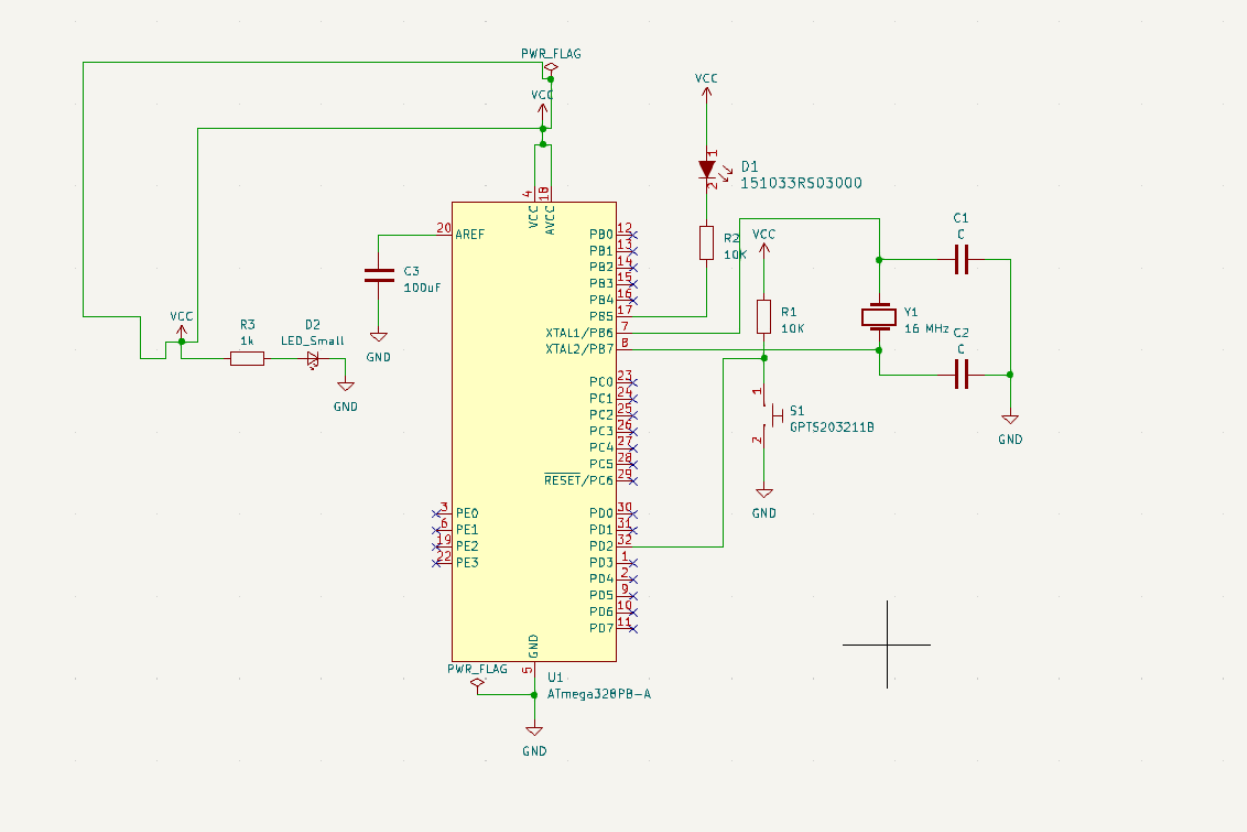


4. SCHEMATICS

Task 2:

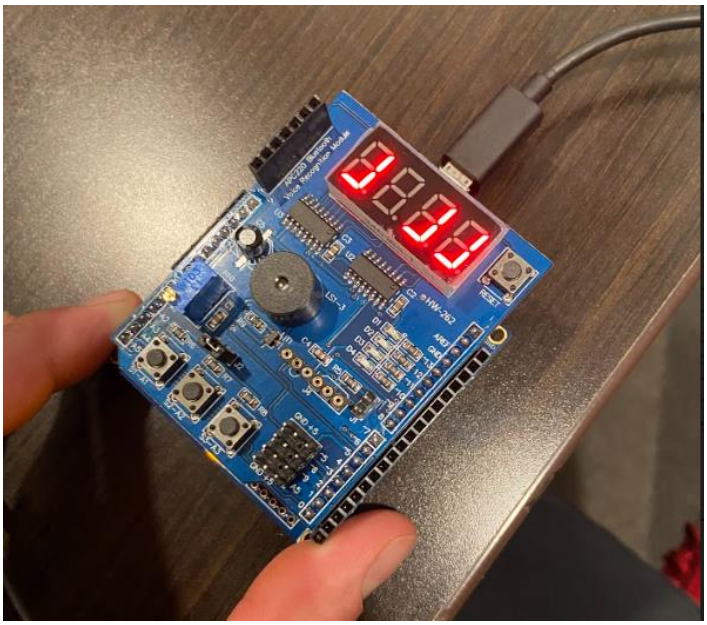


Task 3:

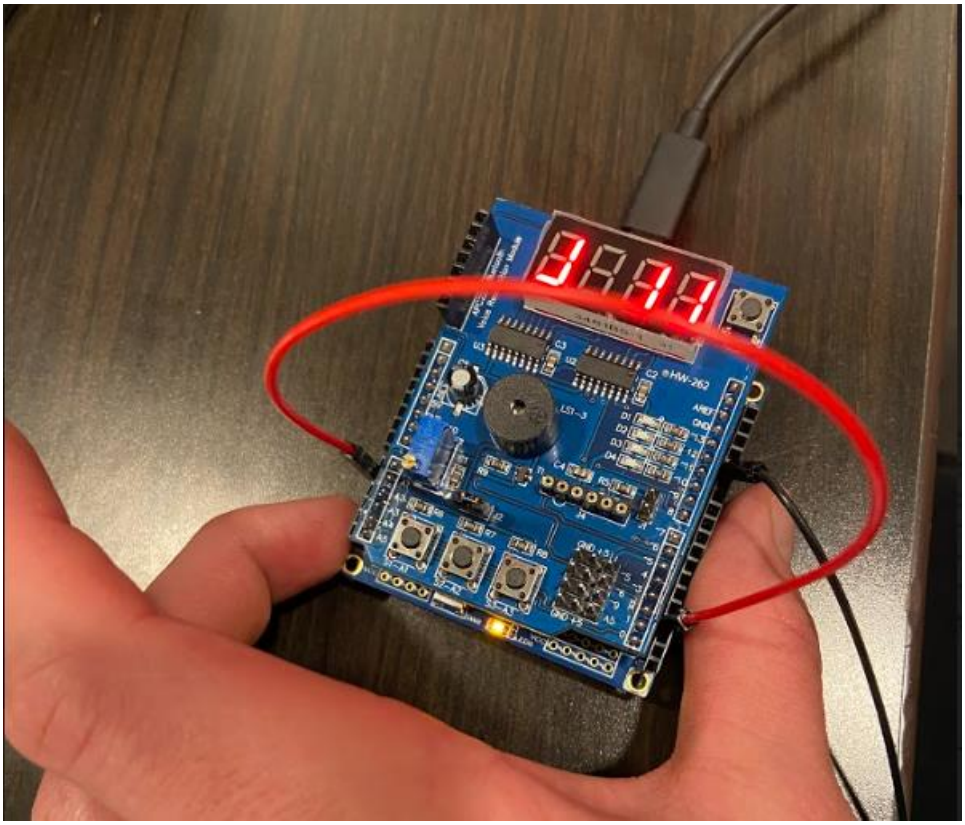


5. SCREENSHOT OF EACH DEMO (BOARD SETUP)

Task 2:



Task 3:



6. VIDEO LINKS OF EACH DEMO

Link to Playlist: https://www.youtube.com/playlist?list=PLlHKEZIJ23uAW9Wtslh2ZcF6vdceCNBiI

Link to Task 2 C: https://youtu.be/OvwKFsnboKM

Link to Task 2 Assembly: https://youtu.be/ozF4z_ViWo8

Link to Task 3 C: https://youtu.be/cus2H9IzRUw

Link to Task 3 Assembly: https://youtu.be/A67nv6LcItk

7. GITHUB LINK OF THIS DA

https://github.com/dlenzin15/submissions/tree/main/DA2

Student Academic Misconduct Policy
http://studentconduct.unlv.edu/misconduct/policy.html

"*This assignment submission is my own, original work*".
David Lenzin