CPE301

Design Assignment 2C

Student Name: Kyungseo Yun

Student #: 2001091216

Student Email: yunk93@unlv.nevada.edu

Primary Github address: <https://github.com/biscuit0x/submission_yun.git>

Directory: submission\_yun/DesignAssignments/DA2C/

**1. COMPONENTS**

ATMEGA328, multi-function shield, breadboard, wires

**2. C CODE**

**TASK 1A**

#include "avr/io.h"

int main()

{

DDRB |= 0X08; //set PB3 as output

PORTB |= 0X08; //turn off PB3

TCCR0A = 0;

TCCR0B = 0X05; //prescale = 1024

int counter = 0;

while(1)

{

counter = 0; //reset counter

TCNT0 = 0; //reset timer

while(counter != 15) //DC40, 0.25 sec

{

while(TCNT0 != 255)

{

}

TCNT0 = 0;

counter += 1;

}

PORTB ^= (1<<3); //toggle PB3

counter = 0; //reset counter

TCNT0 = 0; //reset timer

while(counter != 23) //DC60, 0.375 sec

{

while(TCNT0 != 255)

{

}

TCNT0 = 0;

counter += 1;

}

TCNT0 = 0; //reset timer

}

}

**TASK 1B**

#include "avr/io.h"

int main()

{

DDRB = 0X04; //PB2 as output

PORTB = 0X04;

TCCR0A = 0; //set timer0 normal mode

TCCR0B = 0X05; //prescaler = 1024

int counter = 0; //initialize counter

while(1)

{

while(PINC & (1 << PINC3)) //wait for PINC3 input

{

}

PORTB ^= (1<<2); //toggle LED

counter = 0; //reset counter

TCNT0 = 0; //reset timer

while(counter != 82) //count 1.333 sec

{

while(TCNT0 != 255)

{

}

counter++;

TCNT0 = 0;

}

PORTB ^= (1<<2); //toggle LED

}

}

**TASK 2A**

#include "avr/io.h"

#include "avr/interrupt.h"

int main()

{

DDRB |= 0X08; //set PB3 as output

PORTB |= 0X08; //turn off PB3

TCCR0A = 0;

TCCR0B = 0X05; //prescale = 1024

TIMSK0 = (1<<TOIE0);//enable TIMER0 overflow interrupt

sei(); //enable global interrupt

while(1)

{

}

}

ISR (TIMER0\_OVF\_vect)

{

int counter = 0; //reset counter

TCNT0 = 0; //reset timer

while(counter != 15) //DC40, 0.25 sec

{

while(TCNT0 != 255)

{

}

TCNT0 = 0;

counter += 1;

}

PORTB ^= (1<<3); //toggle PB3

counter = 0; //reset counter

TCNT0 = 0; //reset timer

while(counter != 23) //DC60, 0.375 sec

{

while(TCNT0 != 255)

{

}

TCNT0 = 0;

counter += 1;

}

}

**TASK 2B**

#include "avr/io.h"

#include "avr/interrupt.h"

int main()

{

DDRB |= 0X04; //set PB2 as output

PORTB |= 0X04;

TCCR0A = 0; //timer0 normal mode

TCCR0B = 0X05; //prescaler = 1024

TIMSK0 = (1<<TOIE0); //enable overflow interrupt

sei(); //enable global interrupt

while(1) //wait for interrupt

{

}

}

ISR (TIMER0\_OVF\_vect)

{

while(PINC & (1 << PINC3)) //wait for PINC3 input

{

}

PORTB ^= (1<<2); //toggle LED

int counter = 0; //reset counter&timer

TCNT0 = 0;

while(counter != 82) //count 1.333 sec

{

while(TCNT0 != 255)

{

}

TCNT0 = 0;

counter += 1;

}

PORTB ^= (1<<2); //reset timer

}

**TASK 3A**

#include "avr/io.h"

#include "avr/interrupt.h"

int main()

{

PORTB |= 0X08; //toggle PB3

DDRB |= 0X08; //set PB3 as output

TCCR0B = 0X05; //set timer0 to CTC mode

TCCR0A |= (1<<WGM01); //prescale = 1024

OCR0A = 0XFF; //compare value = 0XFF

TIMSK0 = (1<<OCIE0A); //enable timer0 compare interrupt

sei(); //enable global interrupt

while(1)

{

}

}

ISR (TIMER0\_COMPA\_vect)

{

int counter = 0; //reset counter

TCNT0 = 0; //reset timer

while(counter != 15) //DC40, 0.25 sec

{

while((TIFR0 &(1<<OCF0A))==0)

{

}

TCNT0 = 0;

TIFR0 |= (1<<OCF0A); //reset interrupt flag register

counter += 1;

}

PORTB ^= (1<<3); //toggle PB3

counter = 0; //reset counter

TCNT0 = 0; //reset timer

//TIFR0 |= (1<<OCF0A);

while(counter != 23) //DC60, 0.375 sec

{

while((TIFR0&(1<<OCF0A))==0)

{

}

TCNT0 = 0;

TIFR0 |= (1<<OCF0A); //reset interrupt flag register

counter += 1;

}

}

**TASK 3B**

#include "avr/io.h"

#include "avr/interrupt.h"

int main()

{

PORTB |= 0X04; //turn off PB2

DDRB |= 0X04; //set PB2 as output

TCCR0B = 0X05; //set timer0 to CTC mode

TCCR0A |= (1<<WGM01); //prescale = 1024

OCR0A = 0XFF; //compare value = 0XFF

TIMSK0 = (1<<OCIE0A); //enable timer0 compare interrupt

sei(); //enable global interrupt

while(1)

{

}

}

ISR (TIMER0\_COMPA\_vect)

{

while(PINC & (1 << PINC3)) //wait for PINC3 input

{

}

PORTB ^= (1<<2); //toggle LED

int counter = 0; //reset counter

TCNT0 = 0; //reset timer

while(counter != 82) //count 1.333 sec

{

while((TIFR0 &(1<<OCF0A))==0)

{

}

TCNT0 = 0;

TIFR0 |= (1<<OCF0A); //reset interrupt flag register

counter += 1;

}

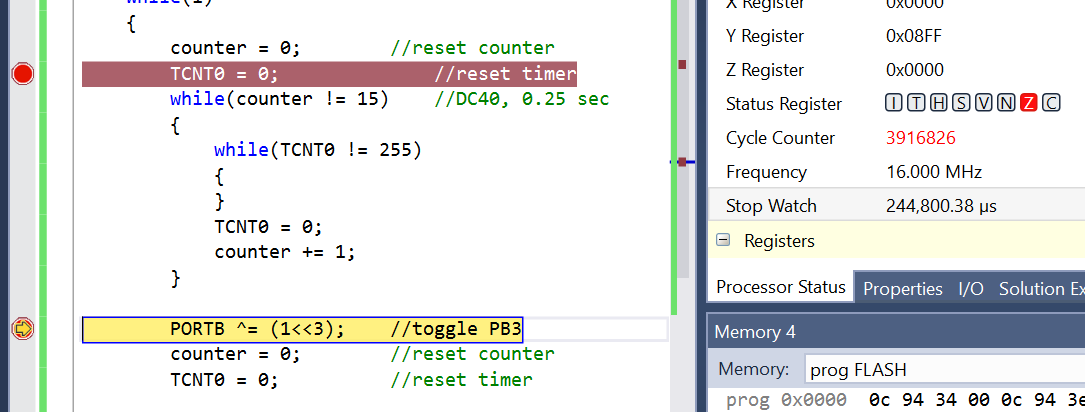
PORTB ^= (1<<2); //toggle LED

}

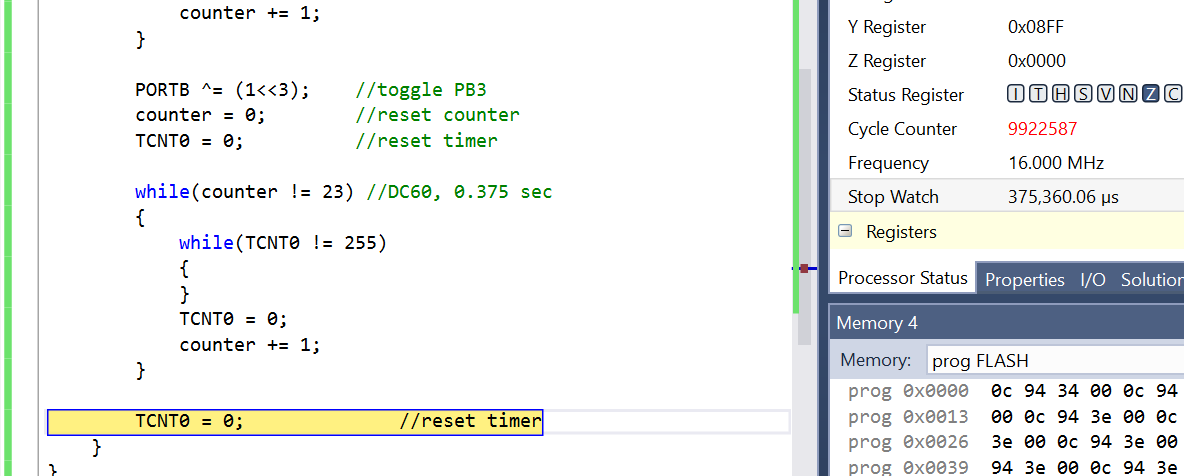
**3. SCREENSHOTS**

**TASK 1A**

0.25 sec(40% DC)

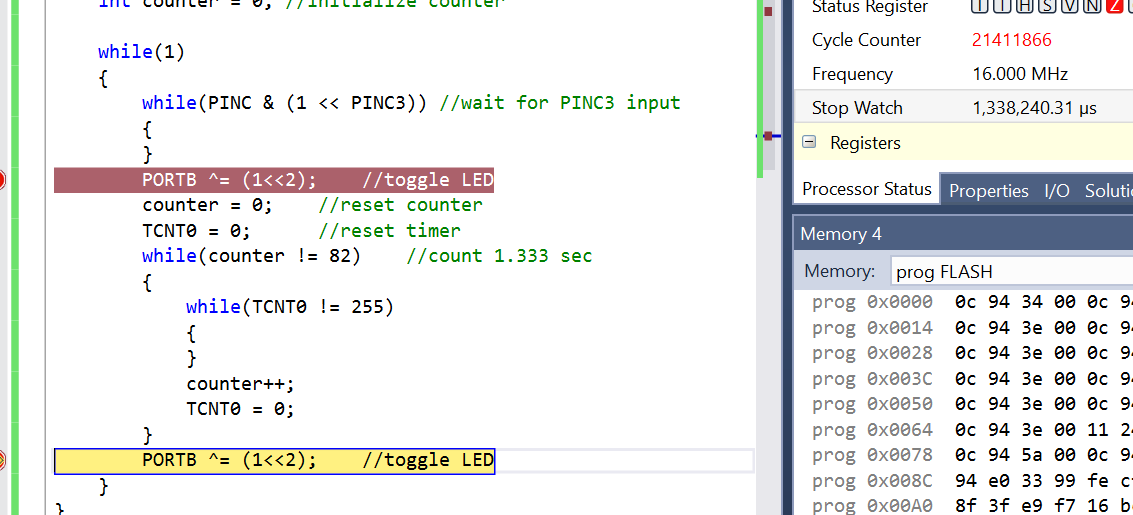


0.375 sec(60% DC)



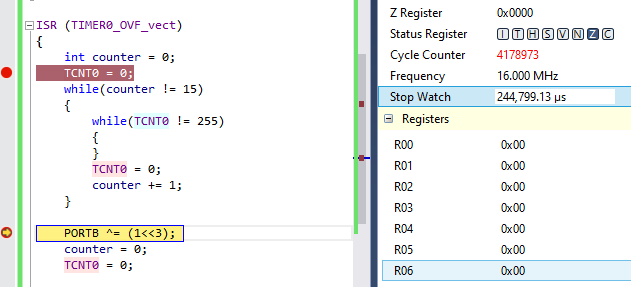
**TASK1B**

PB2 on for 1.333 sec

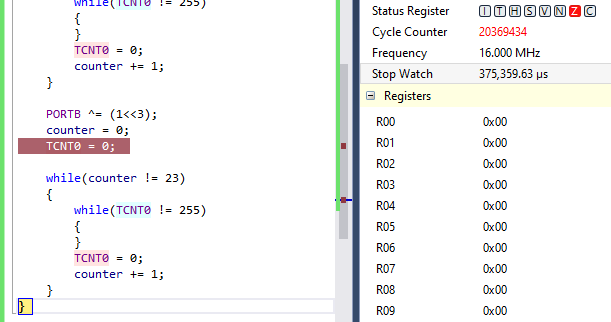
****

**TASK 2A**

0.25 sec (40% DC)

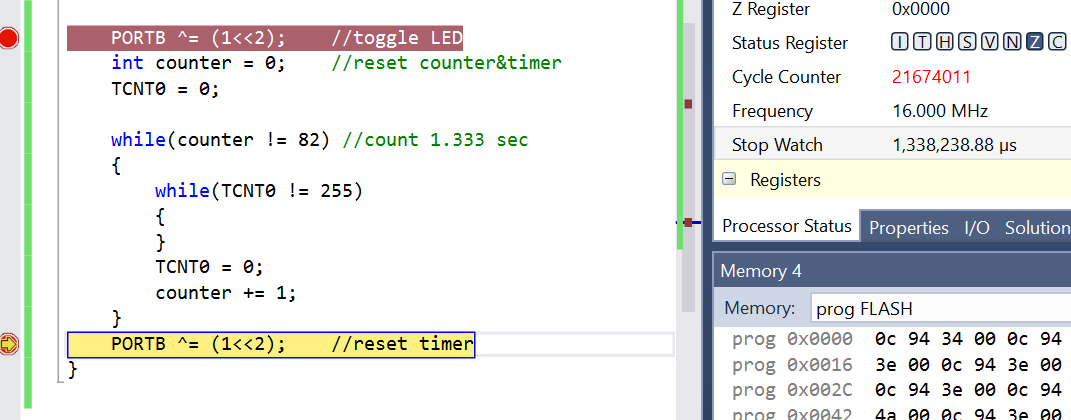


0.375 sec (60% DC)

****

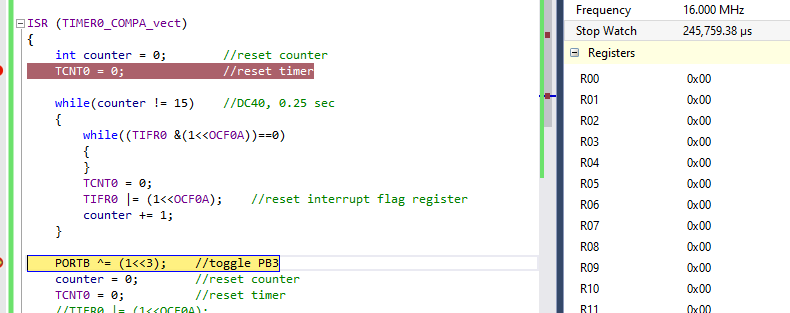
**TASK2B**

PB2 on for 1.333 sec

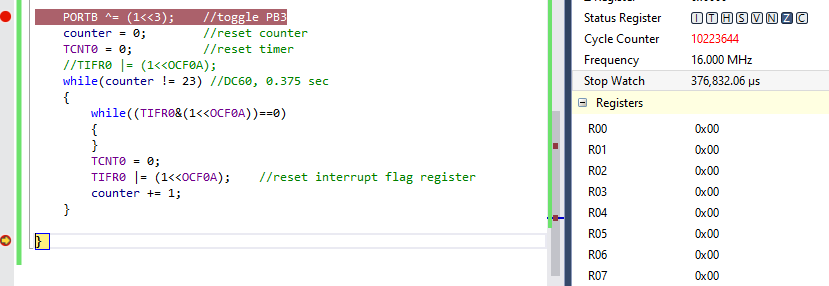
****

**TASK 3A**

0.25 sec (40% DC)

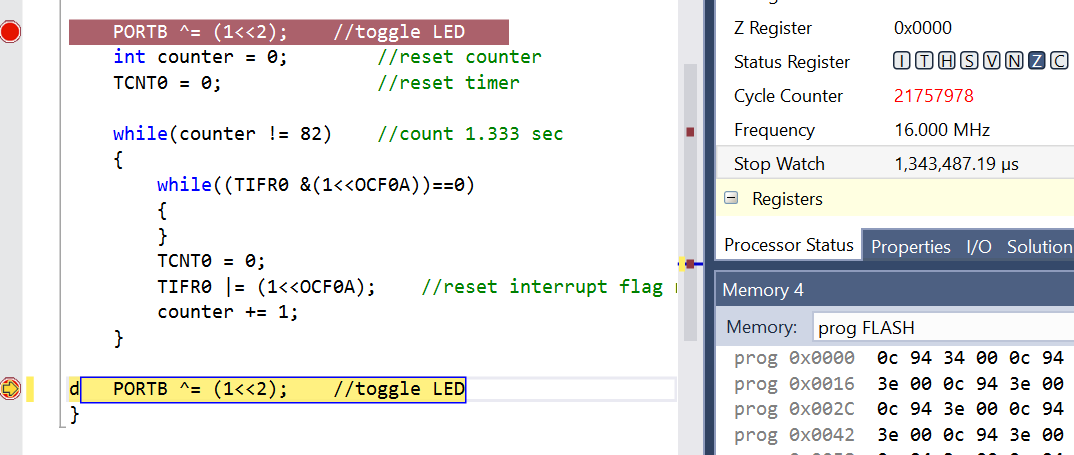
****

0.375 sec (60% DC)

****

**TASK3B**

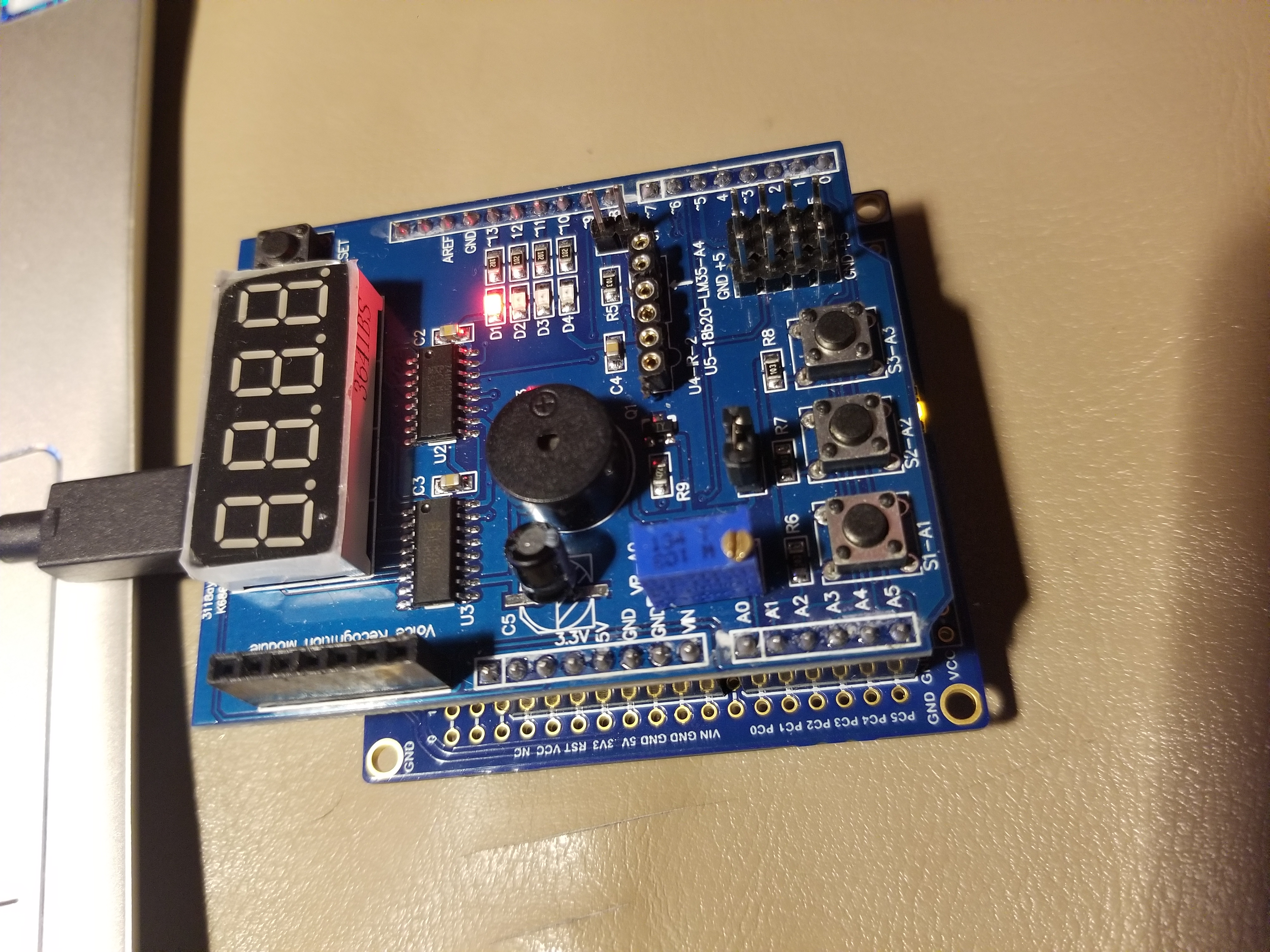
PB2 on for 1.333 sec

****

**3. SCHEMATIC**



**4. PHOTO / VIDEO**

****

1A : <https://youtu.be/orFIuuKqWOc>

1B : <https://youtu.be/_xH8FsgY4z0>

2A : <https://youtu.be/2-Z6GXzRoBM>

2B : <https://youtu.be/JHIFcd4IOpg>

3A : https://youtu.be/j-wZzvH0tGc

3B : https://youtu.be/TWXzXIFuZGU

**6. GITHUB ADDRESS**

[https://github.com/biscuit0x/submission\_yun/tree/master/DesignAssignments/DA2C](https://github.com/biscuit0x/submission_yun/tree/master/DesignAssignments/DA2CB)