CPE301 – FALL 2019

Design Assignment 3A

Student Name: Andrew Buchanan

Student #:5003154346

Student Email: buchaa2@unlv.nevada.edu

Primary Github address: <https://github.com/buchaa2/103EPC>

Directory: https://github.com/buchaa2/103EPC/tree/master/DA3B

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

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <stdlib.h>

#include <math.h>

#include <avr/interrupt.h>

#include <stdio.h>

#include <util/delay.h>

#include <math.h>

#define BAUDRATE 9600

#define BAUD\_PRESCALLER (((*F\_CPU* / (BAUDRATE \* 16UL))) - 1)

void USART\_init(void); //initialize USART

void USART\_transmit(unsigned char data); //send through USART

unsigned char USART\_receive(void); //receive through USART

void USART\_putstring(char\* StringPtr); //scan through string

void ADC\_init (void); //function to initialize ADC

volatile unsigned int ADC\_temp;

char outs[20];

*uint8\_t* OVFLOWCOUNT = 0; //initialize the overflow count for interrupt

*uint8\_t* OVFLOWLIMIT = 250; //set the over flow limit for 1 sec delay

int main(void)

{

USART\_init(); //initialize USART

ADC\_init (); //initialize ADC

USART\_putstring("Connected\r\n");

TCCR0A = 0x00; //set normal operation

TCCR0B |= (1 << CS02); //set prescalar to 256

TCNT0 = 6;

TIMSK0 |= (1 << TOIE0);

sei(); //set interrupt

while (1)

{ //sit here waiting for interrupt

}

}

void USART\_init(void)

{

UBRR0H = (*uint8\_t*)(BAUD\_PRESCALLER >> 8);

UBRR0L = (*uint8\_t*)(BAUD\_PRESCALLER);

UCSR0B = (1 << RXEN0) | (1 << TXEN0);

UCSR0C = (3 << UCSZ00);

}

unsigned char USART\_receive(void)

{

while(!(UCSR0A & (1 << RXC0)));

return UDR0;

}

void USART\_transmit(unsigned char data)

{

while(!(UCSR0A & (1 << UDRE0)));

UDR0 = data;

}

void USART\_putstring(char\* StringPtr)

{

while(\*StringPtr != 0x00)

{

USART\_transmit(\*StringPtr);

StringPtr++;

}

}

void ADC\_init (void)

{

ADMUX = (0<<REFS1)|

(1<<REFS0)|

(0<<ADLAR)|

(1<<MUX2)|

(0<<MUX1)| // ADC5 PC5

(0<<MUX0);

ADCSRA = (1<<ADEN)|

(0<<ADSC)|

(0<<ADATE)|

(0<<ADIF)|

(0<<ADIE)|

(0<<ADPS2)| // ADC Prescaler

(1<<ADPS1)|

(1<<ADPS0);

}

ISR (TIMER0\_OVF\_vect)

{

OVFLOWCOUNT++;

if (OVFLOWCOUNT == OVFLOWLIMIT)

{

ADCSRA|=(1<<ADSC);

while((ADCSRA&(1<<ADIF))==0);

ADCSRA |= (1<<ADIF);

//take value from ADC and transmit it

int var = ADCL;

var = var | (ADCH<<8);

var = (var/1024.0) \* 5000/10;

var = (var\*2)+32;

var = var % 100;

USART\_transmit((var/10)+'0');

var = var % 10;

USART\_transmit((var)+'0');

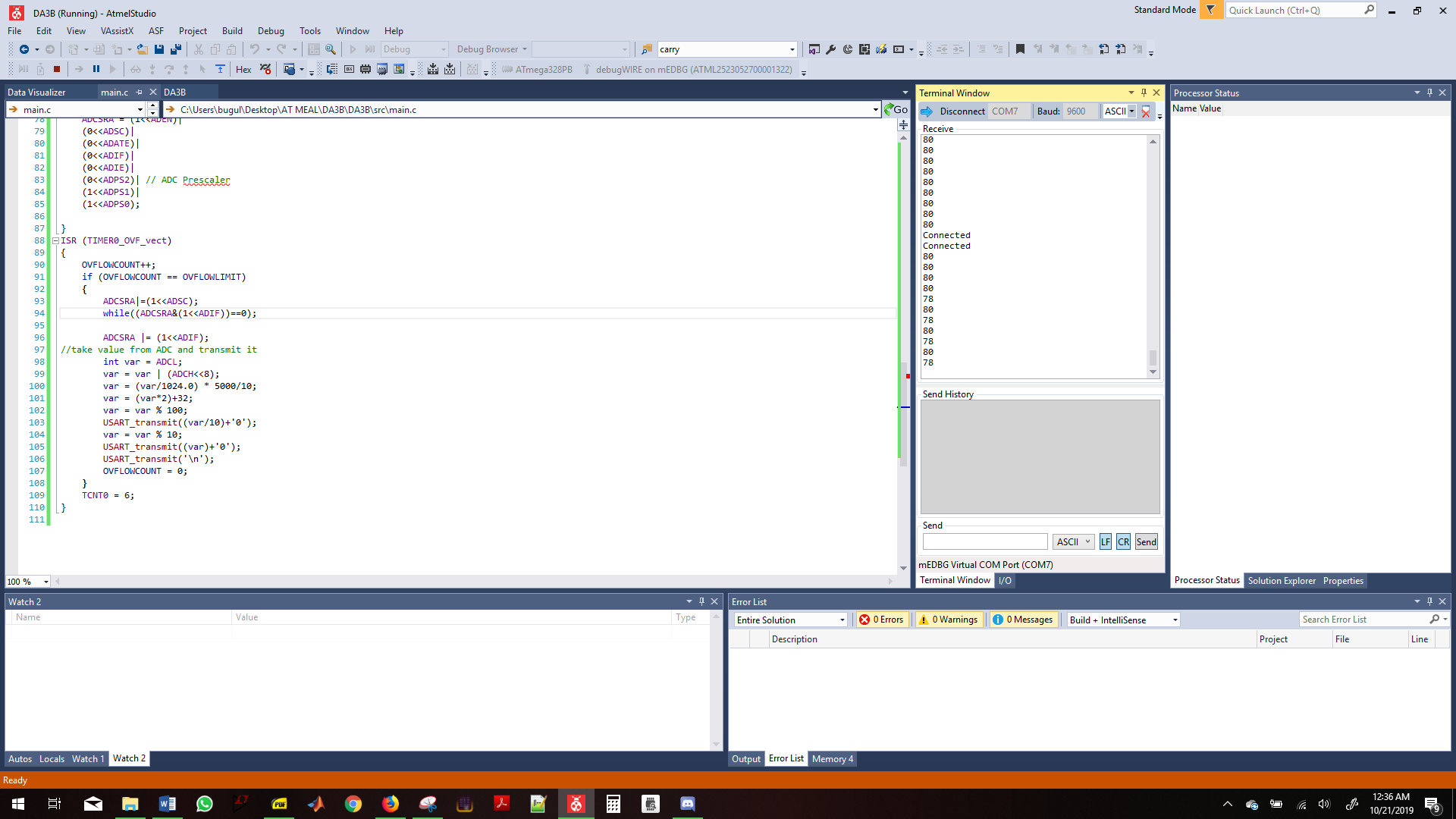
USART\_transmit('\n');

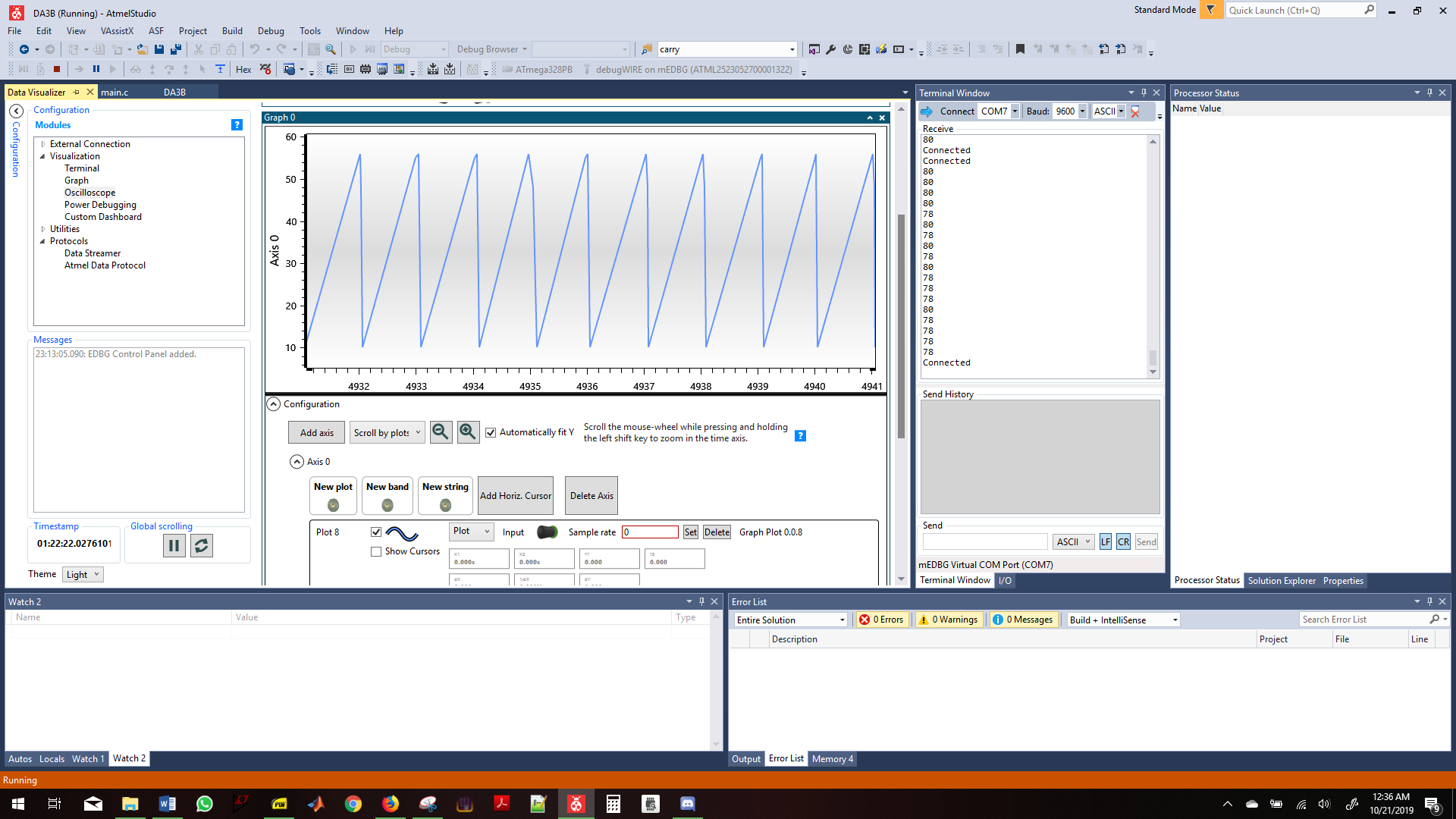
OVFLOWCOUNT = 0;

}

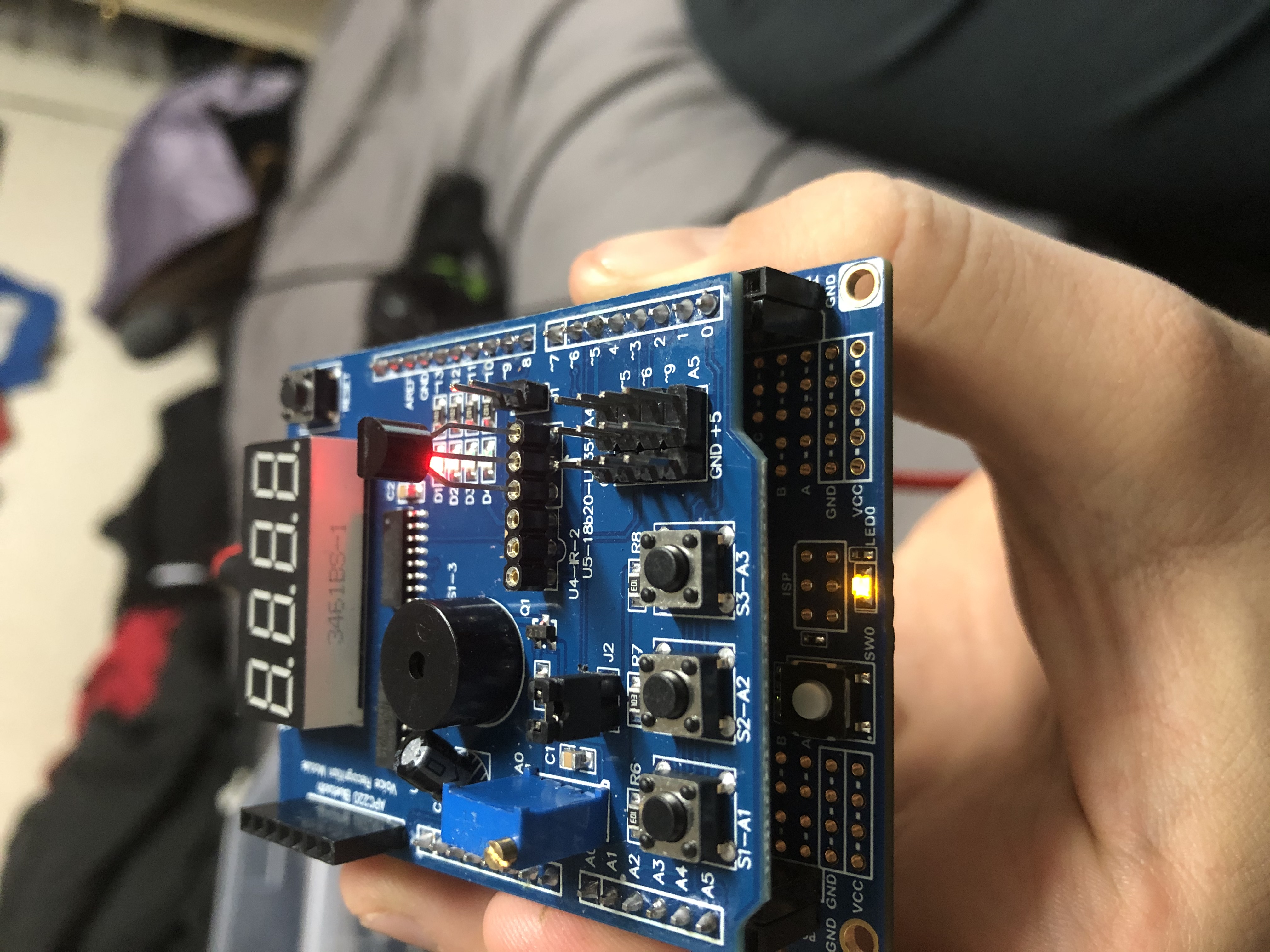
TCNT0 = 6;

1. }**DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A**
2. **SCHEMATICS**
3. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**





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

****

1. **VIDEO LINKS OF EACH DEMO**

https://youtu.be/lF5moTANEWs

1. **GITHUB LINK OF THIS DA**

https://github.com/buchaa2/103EPC/tree/master/DA3B

**Student Academic Misconduct Policy**

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

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

Andrew Buchanan