CPE301 – Fall 2019

Design Assignment 2B

Student Name: Jacob Patrick Reed

Student #: 1008448895

Student Email: reedj35@unlv.nevada.edu

Primary Github address: <https://github.com/reedjacobp>

Directory: <https://github.com/reedjacobp/submission_da>

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

Block diagram with pins used in the Atmega328P

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

/\*

\* DA3B.c

\*

\* Created: 10/20/2019 8:48:28 PM

\* Author : jreed

\*/

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <avr/interrupt.h>

#define BAUDRATE 9600

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

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

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

void USART\_send(unsigned char data); //function to send through USART

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

float temp;

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

*uint8\_t* OVF\_LIMIT = 250; //set the limit the count can reach to set 1 sec delay

int main(void)

{

USART\_init(); //initialize USART

ADCSRA = 0x84;

ADMUX = 0x44;

TCCR0A = 0x00; //normal operation

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

TCNT0 = 16; //TOP = 256-250 = 16

TIMSK0 |= (1 << TOIE0);

sei(); //enable interrupt

while (1)

{

}

}

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\_send(unsigned char data)

{

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

UDR0 = data;

}

void USART\_putstring(char\* StringPtr)

{

while(\*StringPtr != 0x00)

{

USART\_send(\*StringPtr);

StringPtr++;

}

}

ISR (TIMER0\_OVF\_vect)

{

OVF\_COUNT++; //increment the overflow counter

if (OVF\_COUNT == OVF\_LIMIT) //check to see if the limit was reached

{

ADCSRA |= (1 << ADSC);

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

{

ADCSRA |= (1 << ADIF);

int temp = ADCL;

temp = temp | (ADCH<<8);

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

temp = (temp\*2)+32; //equation to convert celsius to farenheit. can't use used 2 instead of 9/5

//USART\_send((a/100)+'0');

temp = temp % 100;

USART\_send((temp/10)+'0');

temp = temp % 10;

USART\_send((temp)+'0');

//USART\_send('\r');

USART\_send('\n');

}

OVF\_COUNT = 0; //reset overflow counter

}

TCNT0 = 16; //reset TOP

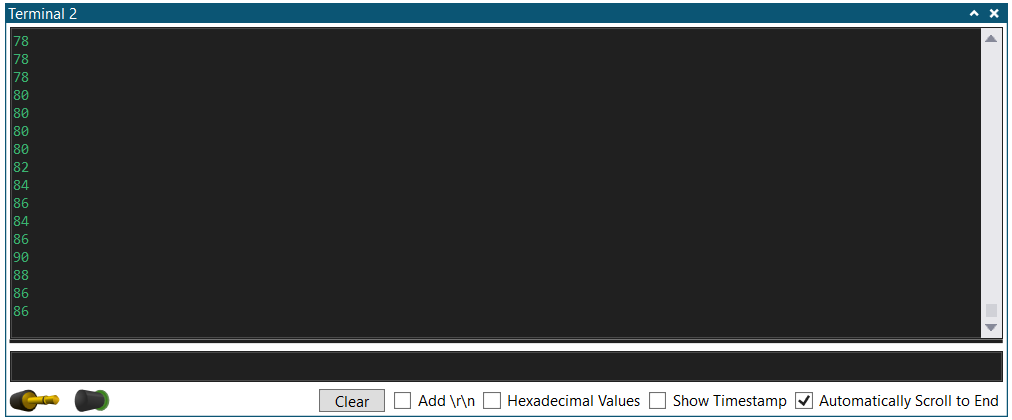
1. }**DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A**

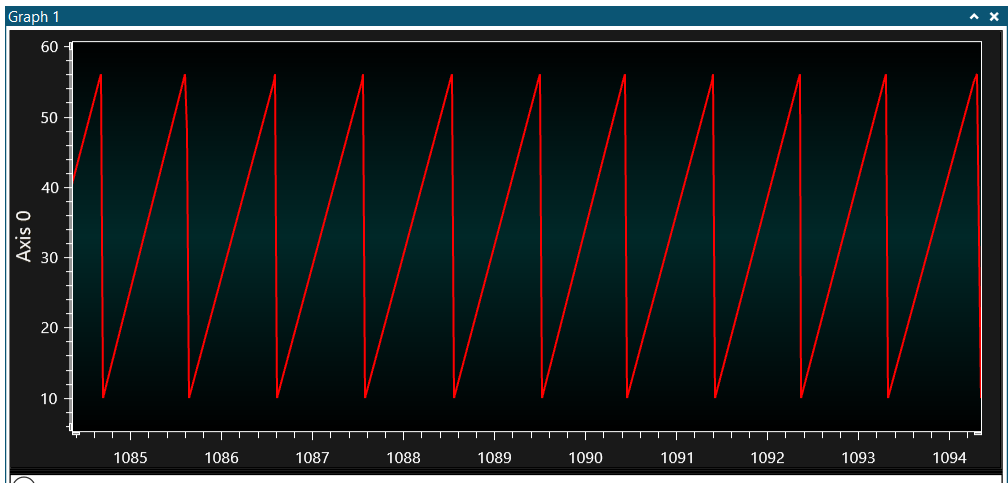
N/A

1. **SCHEMATICS**

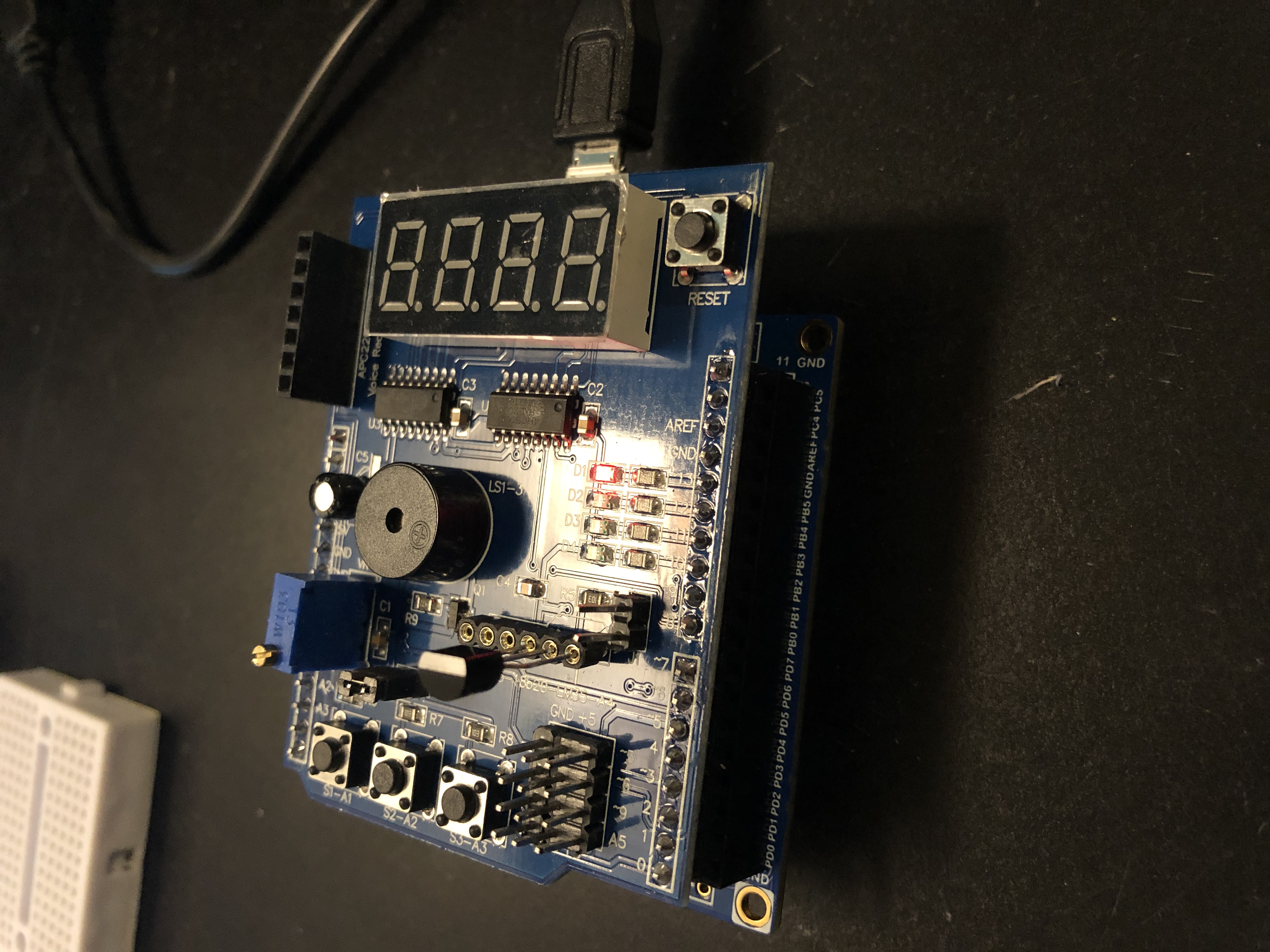
Use fritzing.org

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





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



1. **VIDEO LINKS OF EACH DEMO**

https://youtu.be/SWATbKt9W2s

1. **GITHUB LINK OF THIS DA**

**Student Academic Misconduct Policy**

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

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

NAME OF THE STUDENT