**CPE301 – SPRING 2019**

Design Assignment 3A

Student Name: John Galanza

Student #: 50029181771

Student Email: galanj1@unlv.nevada.edu

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

Directory: <https://github.com/JohnGalanza/supersmashjoe>

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
* USB cable

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

#define F\_CPU 16000000UL

#define BAUD 9600

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

#define UBRR\_9600 103

#include <util/delay.h>

#include <avr/interrupt.h>

#include <stdio.h>

#include <avr/io.h>

#include <util/setbaud.h>

#include <stdlib.h>

//FUNCTION DECLARATIONS

void USART\_init(void);

void USART\_tx\_string(char \*data);

volatile uint8\_t OVFCount; //Global variable

volatile int yurr; //to make delay happen

char buffer[20];

char num[20];

char dec[20];

int main(void)

{

TIMSK0 |= (1<<TOIE0);

TCCR0A = 0x00; //NORMAL mode

TCCR0B |= (1<<CS02)|(1<<CS00); //Prescaler set to 1024

TCNT0 = 0x1F; //This and ovf of 70 produce a 1s delay

USART\_init(); //initialize USART

\_delay\_ms(500);

USART\_tx\_string("\r\nConnected!\r\n");

sei();

while (1)

{

}

}

//Initializes

void USART\_init(void)

{

UBRR0H = UBRRH\_VALUE;

UBRR0L = UBRRL\_VALUE;

UCSR0C = \_BV(UCSZ01)|\_BV(UCSZ00); // 8-Bit data

UCSR0B = \_BV(RXEN0) |\_BV(TXEN0); //Enable RX and TX

}

//Sends data to serial port

void USART\_tx\_string(char \*data)

{

while((\*data!= '\0'))

{

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

UDR0 = \*data;

data++;

}

}

unsigned char USART\_Recieve(void)

{

//wait for data to be received

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

//gets and returns data received

return UDR0;

}

ISR(TIMER0\_OVF\_vect)

{

if(OVFCount<70)

{

TCNT0 = 0x1F;

OVFCount++;

if(OVFCount == 70)

{

TCNT0 = 0x1F;

OVFCount = 0;

int r = rand(); //generates a random number

int x = rand();

int y = rand();

itoa(r, buffer, 10); //converts int to char

itoa(x, num, 10); //similar to above step

itoa(y, dec, 10); //similar to above step

USART\_tx\_string("STRING: ");

USART\_tx\_string("YEE"); //a string

USART\_tx\_string("\n"); //for formatting

USART\_tx\_string("INTEGER: ");

USART\_tx\_string(buffer); //displays converted int onto terminal

USART\_tx\_string("\n"); //for formatting

USART\_tx\_string("FLOAT: ");

USART\_tx\_string(num); //displays a "float" number

USART\_tx\_string(".");

USART\_tx\_string(dec);

USART\_tx\_string("\n"); //for formatting

}

}

}

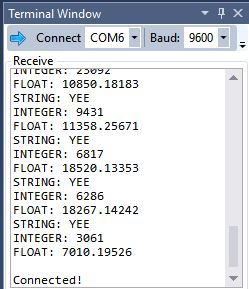
1. **DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A**

There was only 1 task

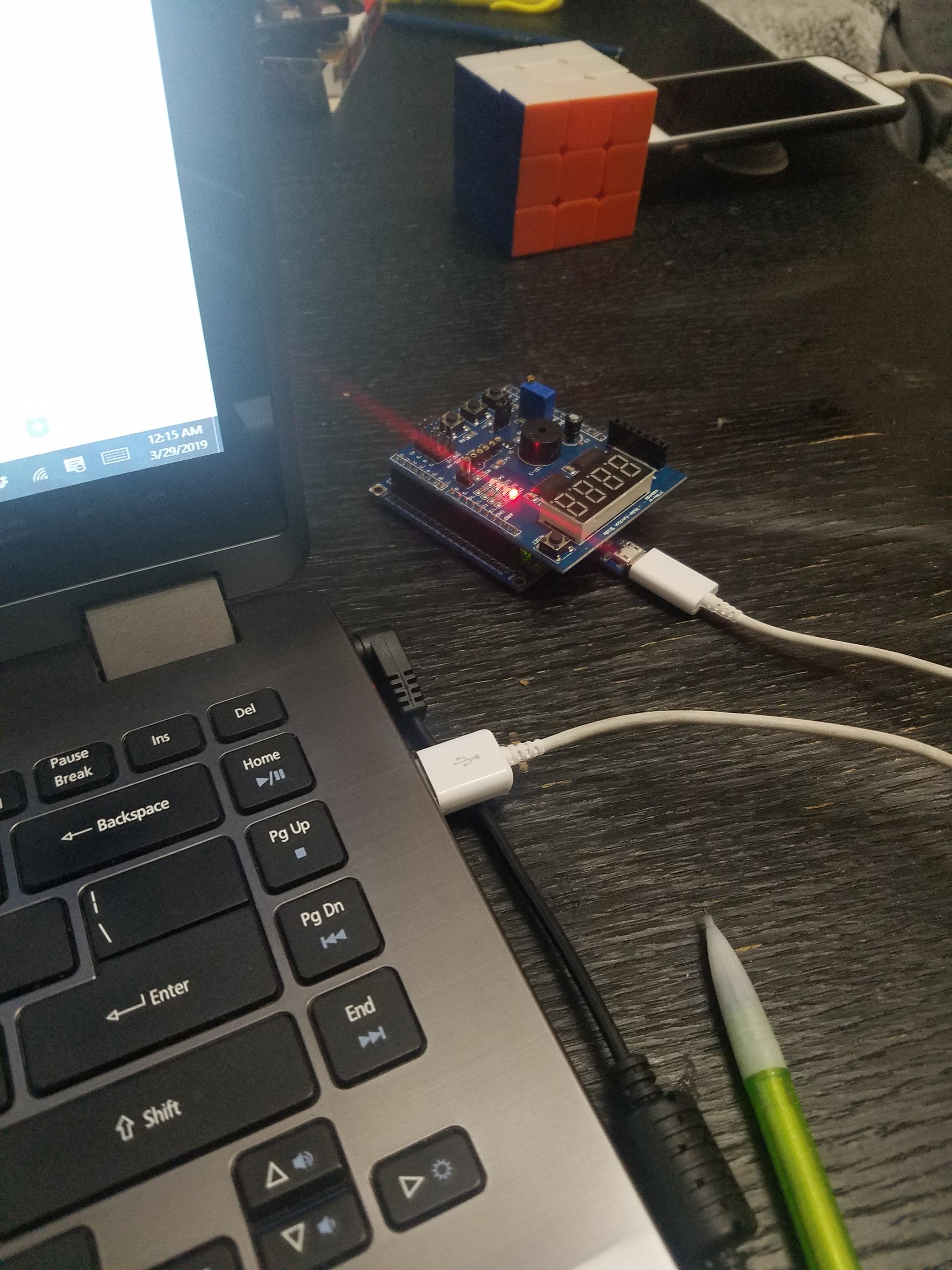
1. **SCHEMATICS**

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/0pWnAcYQCFY**

1. **GITHUB LINK OF THIS DA**

https://github.com/JohnGalanza/supersmashjoe/tree/master/DA3A

**Student Academic Misconduct Policy**

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

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

John Galanza