**CPE301 – SPRING 2019**

Design Assignment 5

Student Name: John Galanza

Student #: 5002981771

Student Email: galanj1@unlv.nevada.edu

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

Directory: https://github.com/JohnGalanza/supersmashjoe/tree/master/DA5

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 cord
* LM35
* NRF24L01+ RF Module

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

//DA5 use earlier ADC code to transmit ADC value of internal temp sensor

//between two RF modules

//Atmega will also alternate between TX and RX modes every 0.5s

//Additional commented sections were for testing

#ifndef F\_CPU // Sets clock frequency.

#define F\_CPU 16000000UL

#endif

#include <avr/io.h> // Includes needed libraries.

#include <util/delay.h>

#include <avr/interrupt.h>

#include <stdbool.h>

#include <stdlib.h>

#include <stdio.h>

#include <string.h>

#include "inc\nrf24l01.c" // Include nRF24L01+ library.

#include "inc\nrf24l01-mnemonics.h"

#include "inc\spi.c"

#ifndef BAUD // Sets up UART for printf();

#define BAUD 9600

#endif

#include "inc\STDIO\_UART.c"

//also additional code

//#define BAUD\_RATE 9600

void print\_config(void); // Functions

void adc\_init(void);

//additional testing code below

void USART\_init();

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

void USART\_send(unsigned char ch);

void USART\_print(char\* str);

volatile unsigned int adc\_temp;

//ends here

volatile bool message\_received = false; // Used in IRQ ISR.

volatile bool status = false; // Used in IRQ ISR.

int tempf = 0; // Used for temperature data.

int main(void)

{

adc\_init(); // Initializes the ADC.

//USART\_init();

char tx\_message[32]; // Defines string array.

strcpy(tx\_message,"Wiggidy Wack"); // Copies string into array.

uart\_init(); // Initializes UART.

nrf24\_init(); // Initializes nRF24L01+ and print configuration info.

print\_config(); // Configures prints.

nrf24\_start\_listening(); // Start listening to incoming messages.

nrf24\_send\_message(tx\_message); // Sends message.

//USART\_print("ON");

while (1)

{

ADCSRA |= (1<<ADSC); // Starts conversion.

while((ADCSRA&(1<<ADIF))==0); // Waits for conversion to finish.

ADCSRA |= (1<<ADIF); // Resets flag for conversion.

tempf = ADCL; // Records temp sensor data.

tempf = tempf | (ADCH<<8);

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

char temp[5]; // Variable used to store tempf string.

itoa(tempf, temp, 10); // Converts tempf integer to string.

//additional code here

//USART\_print(temp);

//Above should cause terminal window to pop up

//and also show a value maybe ???

\_delay\_ms(2000);

message\_received = true;

if (message\_received)

{

message\_received = false; // Message received, print it.

printf("Message received: %s\n",nrf24\_read\_message());

\_delay\_ms(500);

status = nrf24\_send\_message(temp); // Send message as response.

if (status == true) printf("Successfully sent message\n");

}

}

}

ISR(INT0\_vect) // Interrupt on IRQ pin.

{

message\_received = true;

}

void read\_adc(void)

{

unsigned char i = 4;

adc\_temp = 0;

while(i--)

{

ADCSRA|=(1<<ADSC);

while(ADCSRA & (1<<ADSC));

adc\_temp += ADC;

\_delay\_ms(50);

}

adc\_temp = adc\_temp/4; //averages a few samples

}

void USART\_init(void)

{

UBRR0L = 8;

UCSR0C = (1<<UCSZ01)|(1<<UCSZ00); //asynchronous 8 N 1

UCSR0B = (1<<TXEN0)|(1<<RXEN0); //enable receiver, transmitter & RX interrupt

}

void USART\_send(unsigned char ch)

{

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

UDR0 = ch;

}

void USART\_print(char\* str)

{

int i = 0;

while (str[i] != 0)

{

USART\_send(str[i]); //increments i to go through the whole string

i++;

}

}

//Sends data to serial port

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

{

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

{

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

UDR0 = \*data;

data++;

}

}

void print\_config(void)

{

uint8\_t data;

printf("Startup successful\n\n nRF24L01+ configured as:\n");

printf("-------------------------------------------\n");

nrf24\_read(CONFIG,&data,1);

printf("CONFIG 0x%x\n",data);

nrf24\_read(EN\_AA,&data,1);

printf("EN\_AA 0x%x\n",data);

nrf24\_read(EN\_RXADDR,&data,1);

printf("EN\_RXADDR 0x%x\n",data);

nrf24\_read(SETUP\_RETR,&data,1);

printf("SETUP\_RETR 0x%x\n",data);

nrf24\_read(RF\_CH,&data,1);

printf("RF\_CH 0x%x\n",data);

nrf24\_read(RF\_SETUP,&data,1);

printf("RF\_SETUP 0x%x\n",data);

nrf24\_read(STATUS,&data,1);

printf("STATUS 0x%x\n",data);

nrf24\_read(FEATURE,&data,1);

printf("FEATURE 0x%x\n",data);

printf("-------------------------------------------\n\n");

}

void adc\_init (void) // Sets up and enables ADC.

{

ADMUX = (0<<REFS1)| // Reference Selection Bits.

(1<<REFS0)| // AVcc - external cap at AREF.

(0<<ADLAR)| // ADC Left Adjust Result.

(0<<MUX2)| // Analog Channel Selection Bits.

(0<<MUX1)| // ADC0 (PC0).

(0<<MUX0);

ADCSRA = (1<<ADEN)| // ADC Enable.

(0<<ADSC)| // ADC Start Conversion.

(0<<ADATE)| // ADC Auto Trigger Enable.

(0<<ADIF)| // ADC Interrupt Flag.

(0<<ADIE)| // ADC Interrupt Enable.

(1<<ADPS2)| // ADC Pre-scaler Select Bits.

(0<<ADPS1)|

(1<<ADPS0);

}

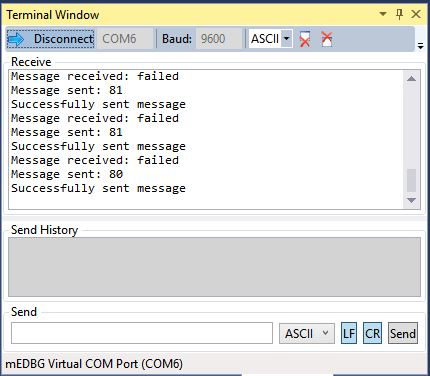
1. **DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A**

Insert only the modified sections here

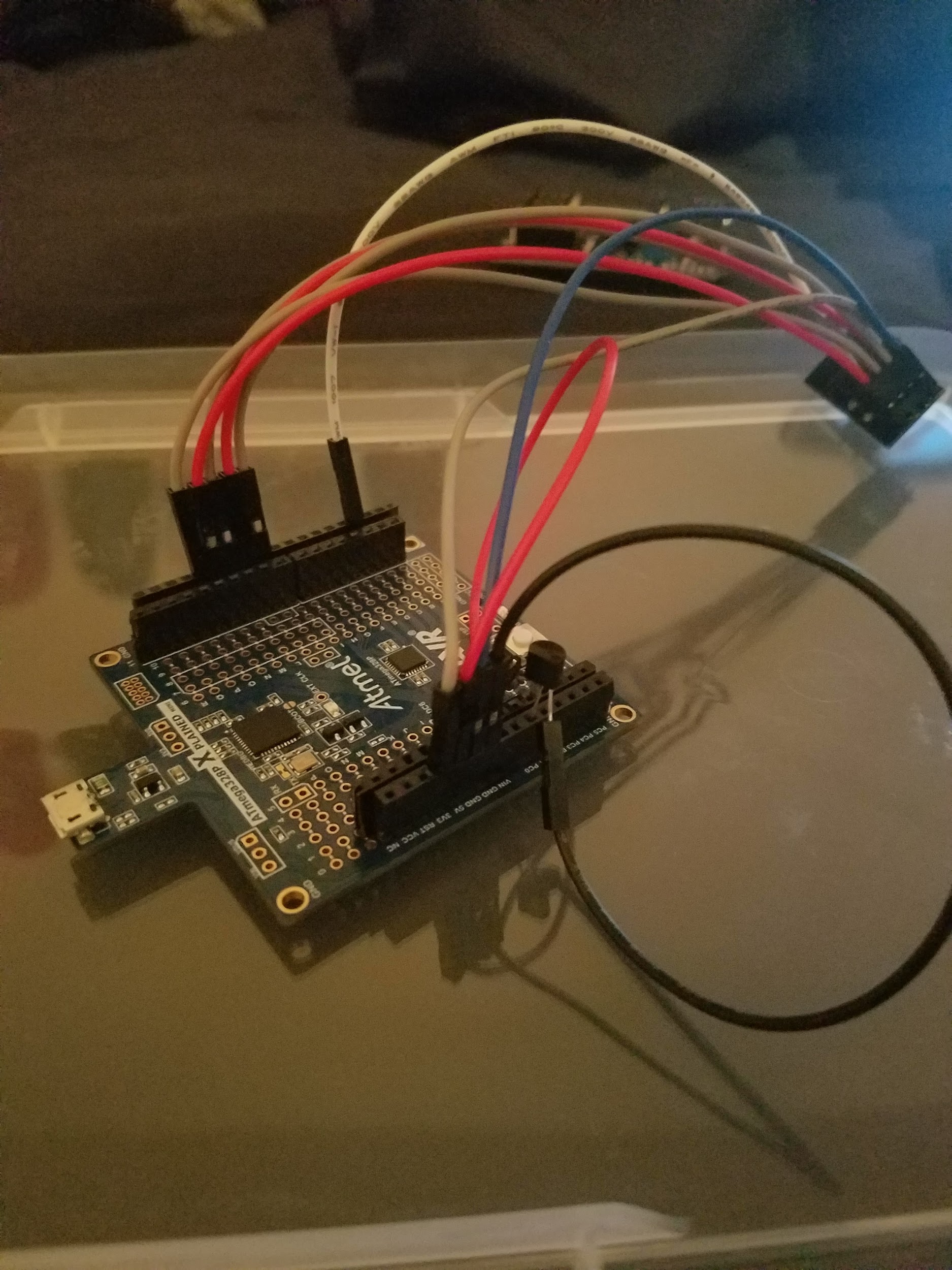
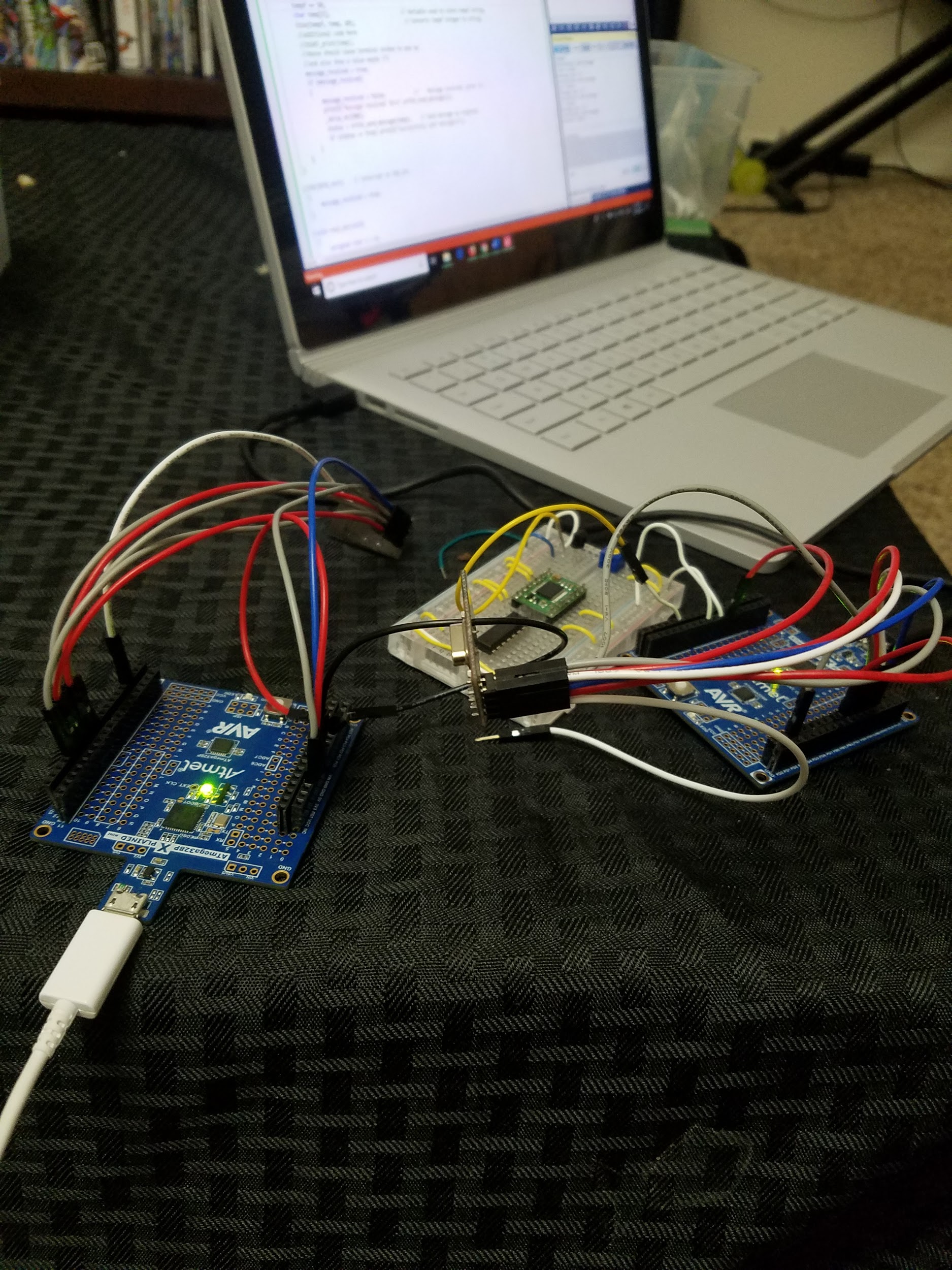
1. **SCHEMATICS**

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**
2. **GITHUB LINK OF THIS DA**

https://github.com/JohnGalanza/supersmashjoe/tree/master/DA5

**Student Academic Misconduct Policy**

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

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

John Galanza