**CPE301 – SPRING 2019**

MIDTERM 1

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>

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/Midterm, 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**

* Atmega 328p
* LM 35 temperature sensor
* FTDI chip
* female to male jumper wires
* breadboard
* laptop

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

#define F\_CPU 16000000UL

#define BAUD\_RATE 9600

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

#include <stdio.h>

#include <stdlib.h> //all the headerss

void read\_adc(void); //Function declarations

void adc\_init(void);

void USART\_init();

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

void USART\_send(unsigned char ch);

void USART\_print(char\* str);

volatile uint8\_t OVFCount;

volatile unsigned int adc\_temp;

char outs[20];

int main(void)

{

adc\_init(); //Initializes ADC

USART\_init(); //Initialize USART

while (1)

{

ADCSRA|=(1<<ADSC);

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

ADCSRA |= (1<<ADIF);

int a = ADCL;

a = a | (ADCH<<8);

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

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

a = a % 100;

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

a = a % 10;

USART\_send((a)+'0'); //converts the number

USART\_send('\r');

\_delay\_ms(1000);

char setMUX[] = "AT+CIPMUX=0\r\n"; //Repeating the steps ESPlorer used

USART\_print(setMUX); //Sends string to USART using USART\_send to send each character of string

\_delay\_ms(1000);

char thingSpeakINIT[] = "AT+CIPSTART=\"TCP\",\"api.thingspeak.com\",80\r\n" ;

USART\_print(thingSpeakINIT);

\_delay\_ms(1000); //add in delay to allow for proper interactions

char SENDthingSpeak[] = "AT+CIPSEND=80\r\n"; //saying we'll send more data than we actually will

USART\_print(SENDthingSpeak);

\_delay\_ms(1000);

char GETthingSpeak; //sends data to thingSpeak

snprintf(GETthingSpeak, sizeof(GETthingSpeak), "GET https://api.thingspeak.com/update?api\_key=R9HDZJDAPCP0G215&field1=0", a);

USART\_print(GETthingSpeak);

\_delay\_ms(1000);

char thingSpeakClose[] = "AT+CIPCLOSE\r\n";

USART\_print(thingSpeakClose);

\_delay\_ms(1000);

}

return 0;

}

void adc\_init(void)

{

//set up and enable 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 bit

(0<<MUX1)| //chooses pc5 pin28

(0<<MUX0);

ADCSRA =(1<<ADEN)|

(0<<ADSC)|

(0<<ADATE)|

(0<<ADIF)|

(0<<ADIE)|

(1<<ADPS2)| //prescaler

(0<<ADPS1)|

(1<<ADPS0);

}

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(stri[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++;

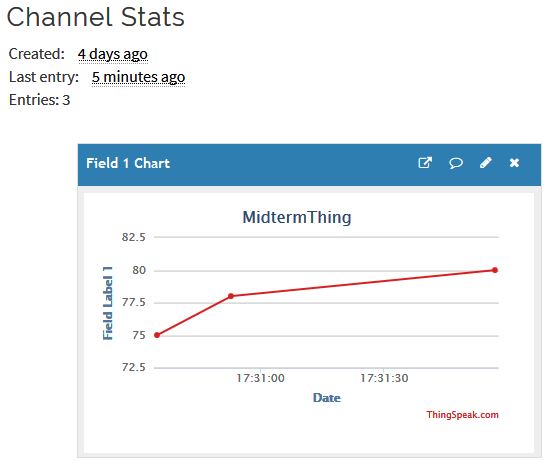
}

}

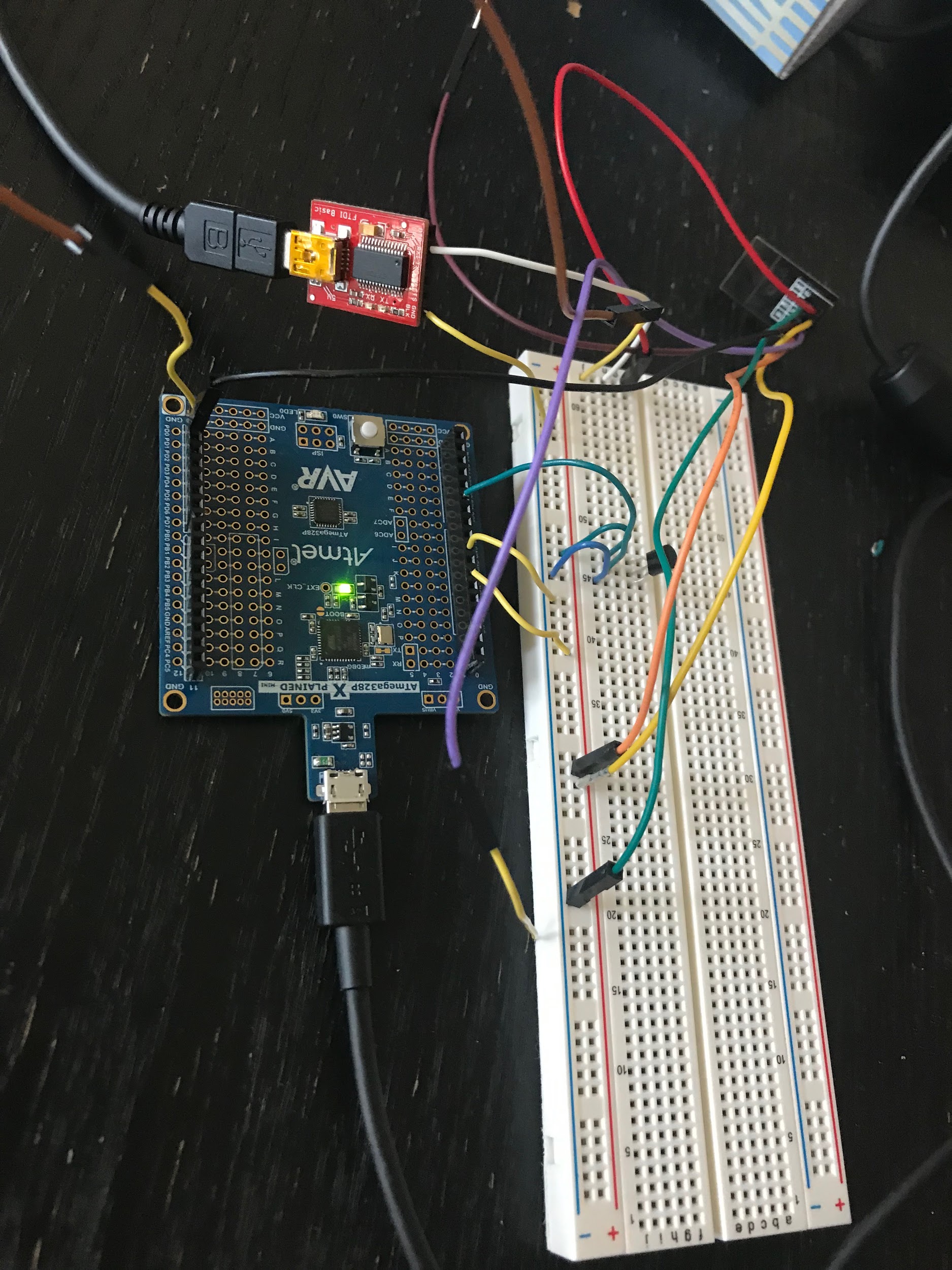
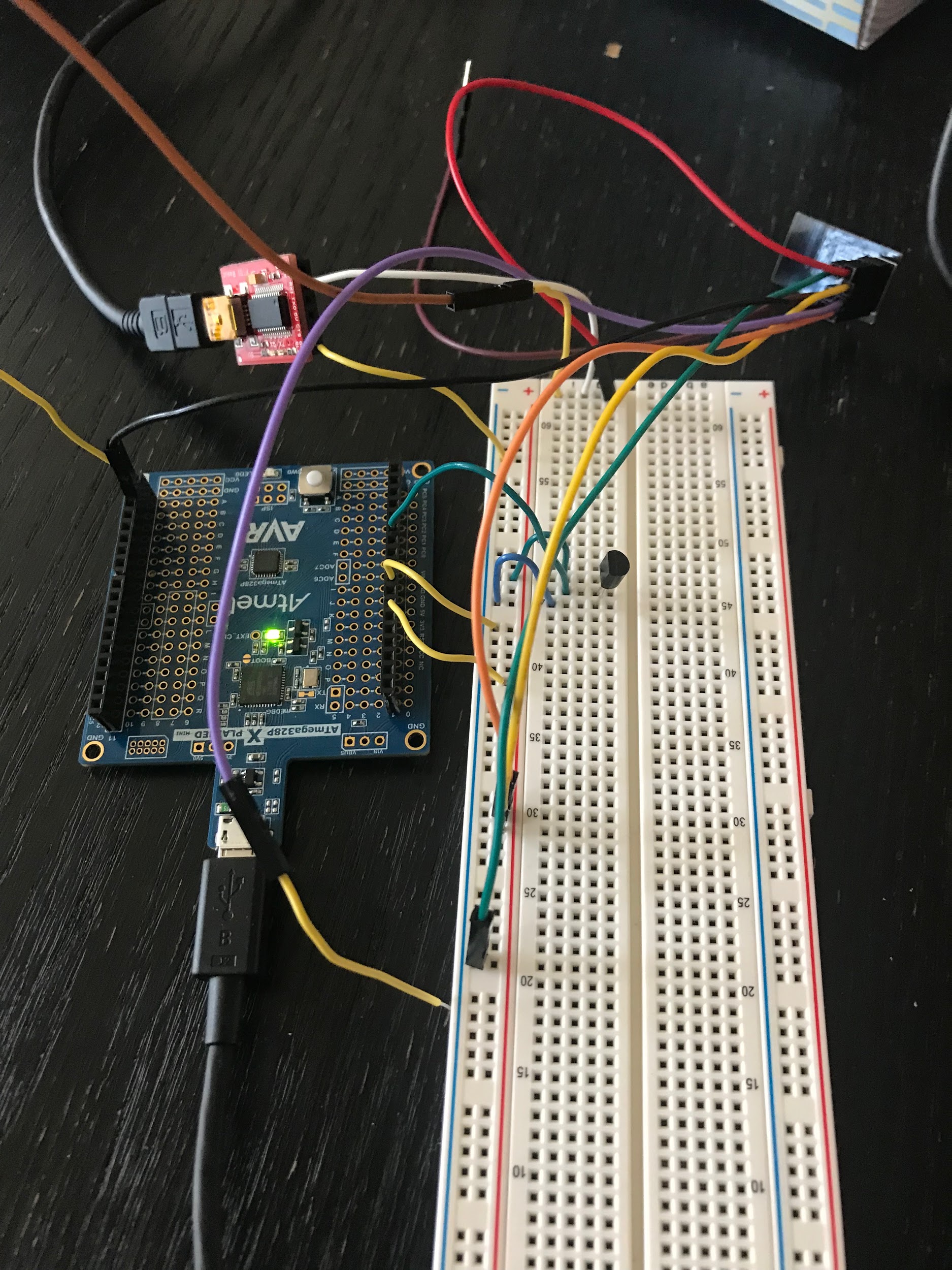
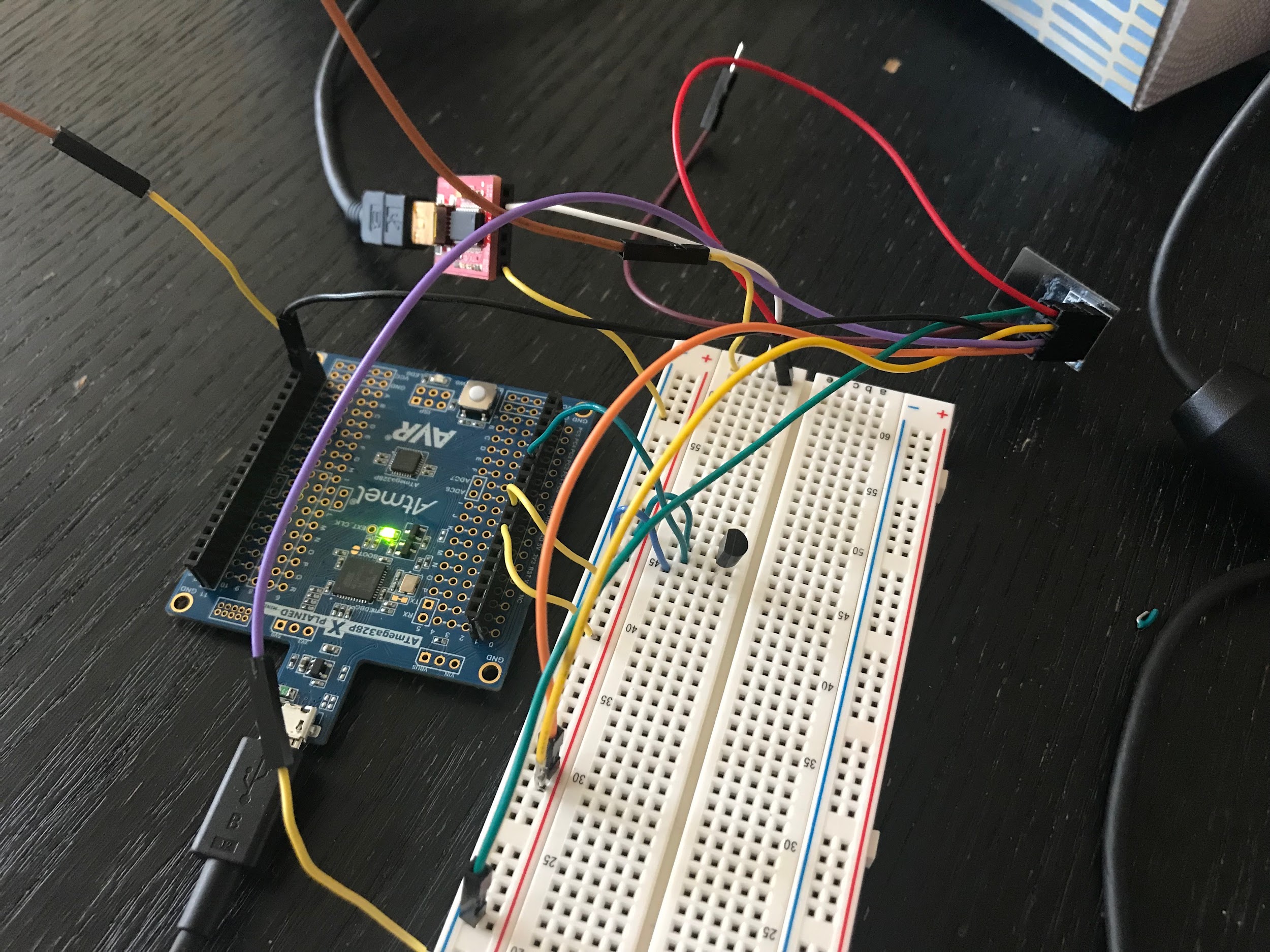
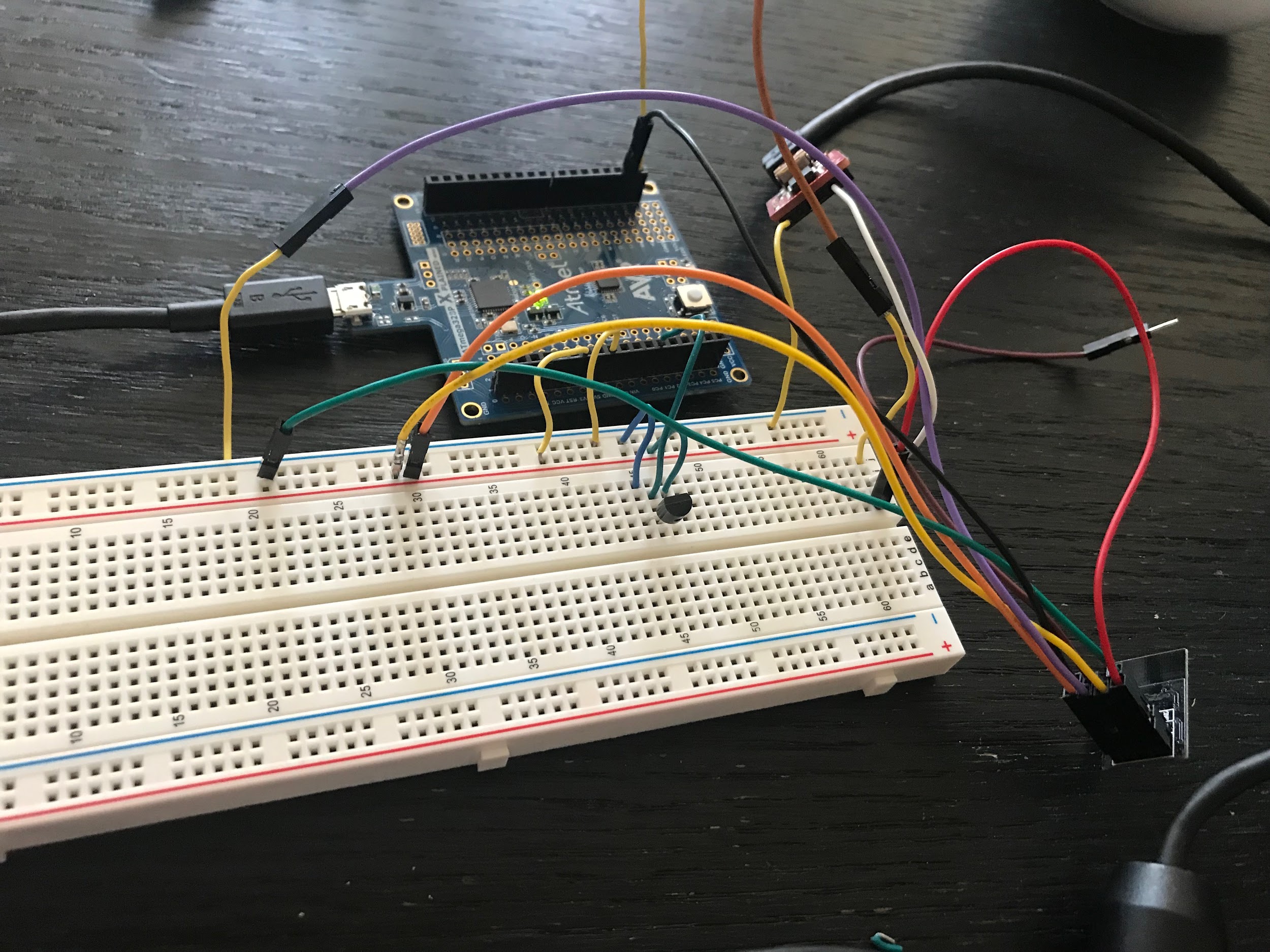
1. **DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A**
2. **SCHEMATICS**

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/CVL937mhxgE>

1. **GITHUB LINK OF THIS DA**

<https://github.com/JohnGalanza/supersmashjoe/tree/master/Midterm%201>

**Student Academic Misconduct Policy**

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

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

John Galanza