CPE301 – SPRING 2019

Design Assignment 4B

Student Name: Brian West

Student #: 5003032874

Student Email: westb2@unlv.nevada.edu

Primary Github address: <https://github.com/westbrian2/Spring2019>

Directory: Spring2019/DesignAssignments

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**

Mini Xplained board, ULN2003, stepper motor, servo motor

1. **Task1**

#define F\_CPU 16000000UL

#include <avr/io.h>

#include <util/delay.h>

void read\_adc(void);

volatile unsigned int adc\_temp;

int main(void){

DDRB=0x0F; //getting lower nibble to output

int step=1; //is used to keep track of the step

//ADC set up

ADMUX=((0<<REFS1)|(1<<REFS0)|(1<<ADLAR)|(0<<MUX3)|(0<<MUX2)|(0<<MUX1)|(0<<MUX0)); //reference Vcc, left shift(8bits), and PC0 is input

ADCSRA=((1<<ADEN)|(0<<ADSC)|(0<<ADATE)|(0<<ADIF)|(0<<ADIE)|(1<<ADPS2)|(0<<ADPS1)|(1<<ADPS0)); //prescaler=32, enable with no interrupts

// setup timer withCTC

OCR0A=78;

TCCR0A=(1<<WGM01);

TCCR0B=(1<<WGM02)|(1<<CS02)|(1<<CS00);

while(1){

read\_adc();

OCR0A=((adc\_temp/255.0)\*171.0)+79;

switch(step){

case 1: PORTB=0x01;

step++;

break;

case 2: PORTB=0x02;

step++;

break;

case 3: PORTB=0x04;

step++;

break;

case 4: PORTB=0x08;

step=1;

break;

}

TIFR0=0x02;

while((TIFR0&(1<<OCF0A))==0){

}

}

}

void read\_adc(void) {

ADCSRA |= (1<<ADSC); //start conversion

while(ADCSRA & (1<<ADSC)); //waiting for coversion to finish

adc\_temp= ADCH;

}

1. **Task 2**

#define F\_CPU 16000000UL

#include <avr/io.h>

#include <util/delay.h>

void read\_adc(void);

volatile unsigned int adc\_temp;

int main(void)

{

//set up ADC to read pot

ADMUX=((0<<REFS1)|(1<<REFS0)|(1<<ADLAR)|(0<<MUX3)|(0<<MUX2)|(0<<MUX1)|(0<<MUX0)); //reference Vcc, left shift(8bits), and PC0 is input

ADCSRA=((1<<ADEN)|(0<<ADSC)|(0<<ADATE)|(0<<ADIF)|(0<<ADIE)|(1<<ADPS2)|(0<<ADPS1)|(1<<ADPS0)); //prescaler=32, enable with no interrupts

//setting up pwm

TCCR1A|=(1<<COM1A1)|(1<<COM1B1)|(1<<WGM11);

TCCR1B|=(1<<WGM13)|(1<<WGM12)|(1<<CS11)|(1<<CS10);

ICR1=4999;

DDRB|=(1<<PB1);

while (1){

read\_adc();

OCR1A=((adc\_temp/255.0)\*438)+97;

}

}

void read\_adc(void) {

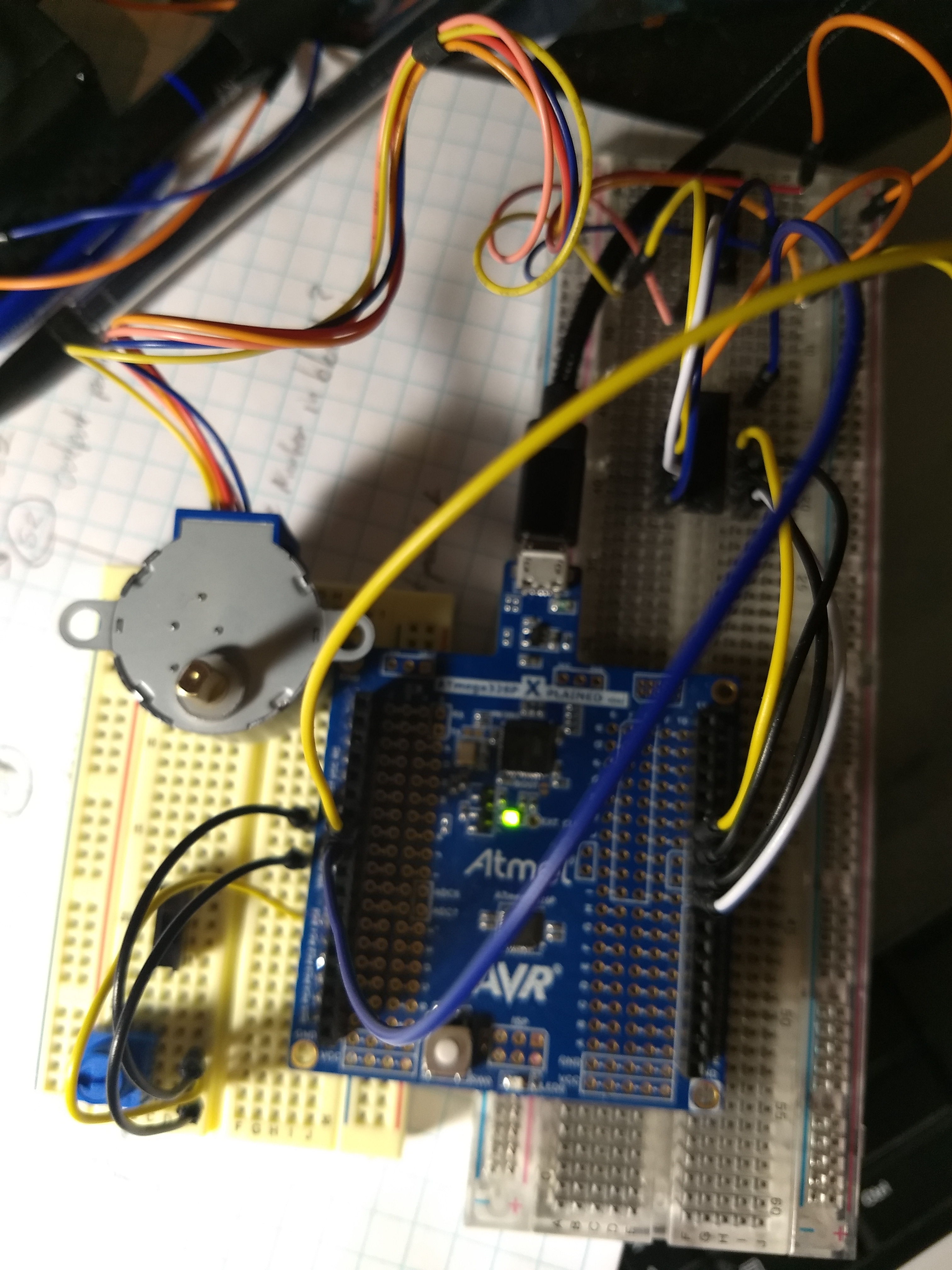
ADCSRA |= (1<<ADSC); //start conversion

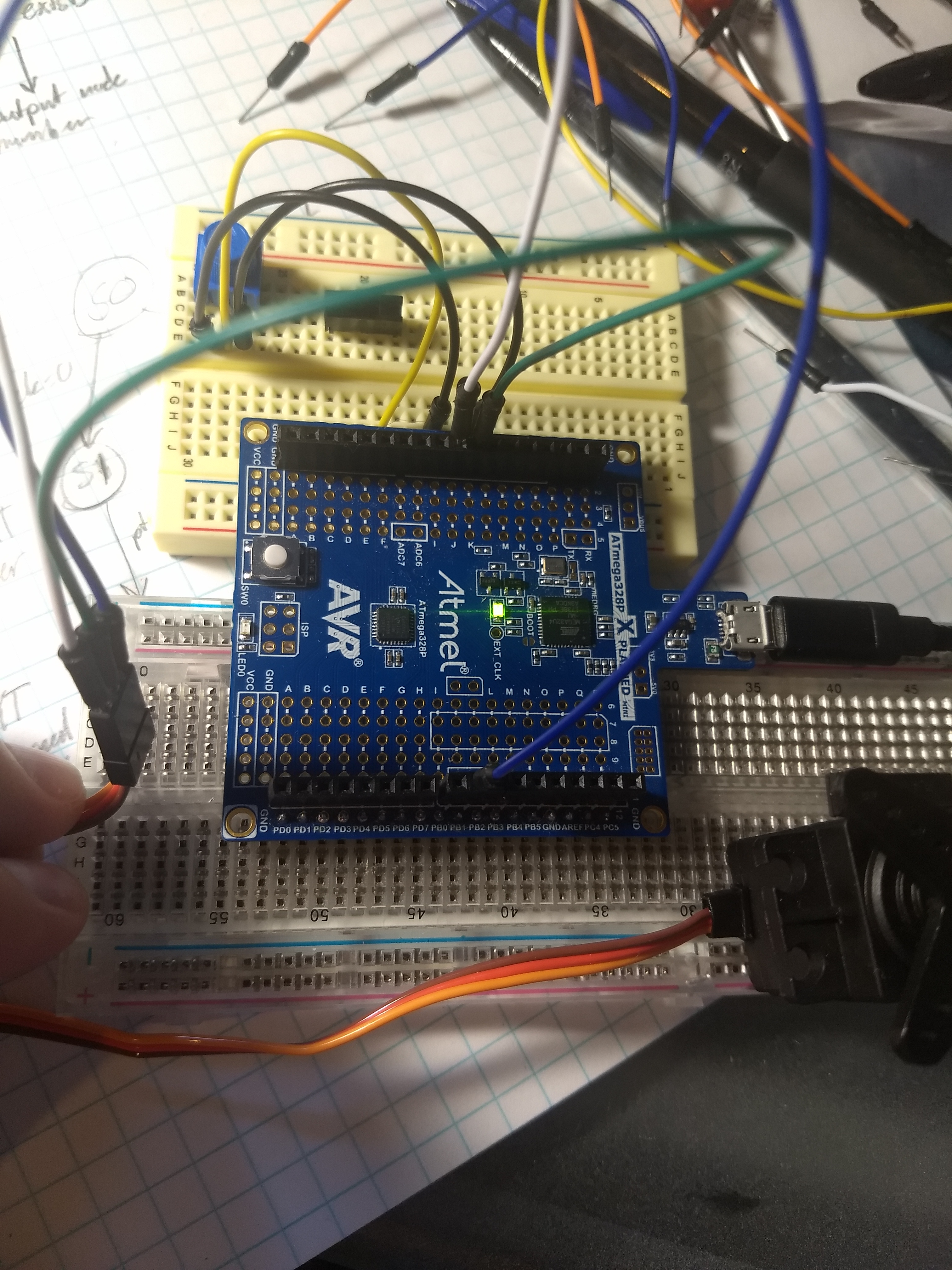
while(ADCSRA & (1<<ADSC)); //waiting for coversion to finish

adc\_temp= ADCH;

}

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





1. **VIDEO LINKS OF EACH DEMO**

TASK1: <https://youtu.be/aI9G1igFBRo>

TASK 2: <https://youtu.be/iVA7DjhyElI>

1. **GITHUB LINK OF THIS DA**

<https://github.com/westbrian2/Spring2019>/DA4B\_submission

**Student Academic Misconduct Policy**

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

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

NAME OF THE STUDENT