CPE301 – SPRING 2019

Design Assignment 4B

Student Name: Allis Hierholzer

Student #: 2000160356

Student Email: hierholz@unlv.nevada.edu

Primary Github address: https://github.com/acexhp/submission\_da.git

Directory: Repository/cpe301/DesignAssignment/DA4B

Task:

The goal of the assignment is to modify the above codes to do the following:

1. Write an AVR C program to control the speed of the Stepper Motor using a potentiometer connected to PC0. Use a timer in CTC mode to control the delay. 2. Write an AVR C program to control the position of the Servo Motor using a potentiometer connected to PC0. When pot value is 0 the servo is at position 0 deg. and when pot value is max (approx. 5V) the servo is at position 180 deg.

Submission:

The following are required for successful completion of the design assignment:

a. AVR C code that has been compiled and working.

b. The C code should be well documented with explanation of every instruction.

c. A word document that contains the flow chart of the assembly code along with the snapshots of the schematics, components connected on the breadboard and screenshots.

1. **COMPONENTS LIST**

* Breadboard
* Stepper Motor (Bi-polar)
* Servo Motor
* ATMEGA328P XPLAINED MINI
* ATMEL STUDIO 7.0
* Multifunction Shield

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

/\*

\* DA4B1.c

\*/

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <util/delay.h>

// Function Prototypes

void set\_register(void);

void set\_adc(void);

int main(void)

{

int period; //variable used to set the period

while (1)

{

set\_register(); //calls function that sets Register 0

set\_adc(); //calls function that sets up adc

DDRB=0x0F; //enable output

OCR0A = ADC;

// Rotate stepper motor counterclockwise

if ((OCR0A>=0)&&(OCR0A<=10))

{

period =2 ; //lowest possible period, highest frequency

PORTB = 0x09;

*\_delay\_ms*(period);

PORTB = 0x03;

*\_delay\_ms*(period);

PORTB = 0x06;

*\_delay\_ms*(period);

PORTB = 0x0c;

*\_delay\_ms*(period);

}

//period = 10

else if((OCR0A>10)&&(OCR0A<=20))

{

period=10;

PORTB = 0x09;

*\_delay\_ms*(period);

PORTB = 0x03;

*\_delay\_ms*(period);

PORTB = 0x06;

*\_delay\_ms*(period);

PORTB = 0x0c;

*\_delay\_ms*(period);

}

//period = 20

else if((OCR0A>20)&&(OCR0A<=30))

{

period=20;

PORTB = 0x09;

*\_delay\_ms*(period);

PORTB = 0x03;

*\_delay\_ms*(period);

PORTB = 0x06;

*\_delay\_ms*(period);

PORTB = 0x0c;

*\_delay\_ms*(period);

}

//period = 30

else if((OCR0A>30)&&(OCR0A<=40))

{

period=30;

PORTB = 0x09;

*\_delay\_ms*(period);

PORTB = 0x03;

*\_delay\_ms*(period);

PORTB = 0x06;

*\_delay\_ms*(period);

PORTB = 0x0c;

*\_delay\_ms*(period);

}

//period = 40

else if((OCR0A>40)&&(OCR0A<=50))

{

period=40;

PORTB = 0x09;

*\_delay\_ms*(period);

PORTB = 0x03;

*\_delay\_ms*(period);

PORTB = 0x06;

*\_delay\_ms*(period);

PORTB = 0x0c;

*\_delay\_ms*(period);

}

//period = 50, high period, low frequency

else if((OCR0A>50)&&(OCR0A<=60))

{

period=50;

PORTB = 0x09;

*\_delay\_ms*(period);

PORTB = 0x03;

*\_delay\_ms*(period);

PORTB = 0x06;

*\_delay\_ms*(period);

PORTB = 0x0c;

*\_delay\_ms*(period);

}

else if(OCR0A>60) //stepper motor turns off

{

PORTB=0;

}

}

}

void set\_register(void)

{

TCCR0A |= (1<<COM0A1)|(1<<WGM01); //Sets to CTC mode

TCCR0B |= (1<<CS02)|(1<<CS00); //Pre-scaler is set to 1024

}

//ADC set up

void set\_adc(void)

{

ADCSRA |= ((1<<ADEN)|(1<<ADSC)|(1<<ADPS1)|(1<<ADPS0)); //adc enable, start conversion, scaler

ADMUX |= (1<<REFS0); //Reads external capacitor

}

/\*

\* DA4B2.c

\*/

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <util/delay.h>

//Function Prototypes

void set\_register(void);

void set\_adc(void);

int main(void)

{

while (1)

{

set\_register(); //calls function that sets Timer/Counter control Register 0

set\_adc(); //calls function that sets up adc

ICR1=4999; //sets top register

DDRB|=(1<<PB1); //output

OCR1A=ADC; //controls servo, reads from PB1

*\_delay\_ms*(100); //controls degree

}

}

void set\_register(void)

{

TCCR1A |= (1<<COM1A1)|(1<<COM1B1)|(1<<WGM11); //sets Mode = 14, Fast PWM

TCCR1B |= (1<<WGM13)|(1<<WGM12)|(1<<CS11)|(1<<CS10); //Prescaler = 64

}

//ADC set up

void set\_adc(void)

{

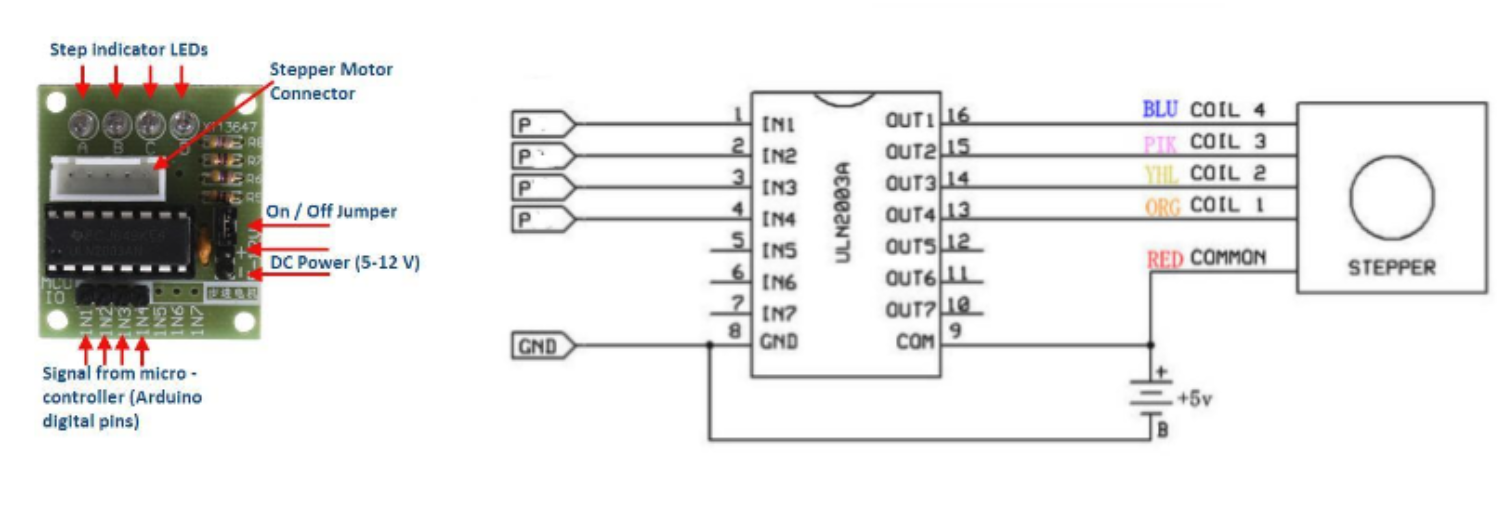
ADCSRA |= ((1<<ADEN)|(1<<ADSC)|(1<<ADPS1)|(1<<ADPS0)); //adc enable, start conversion, scaler

ADMUX |= (1<<REFS0); //Reads external capacitor

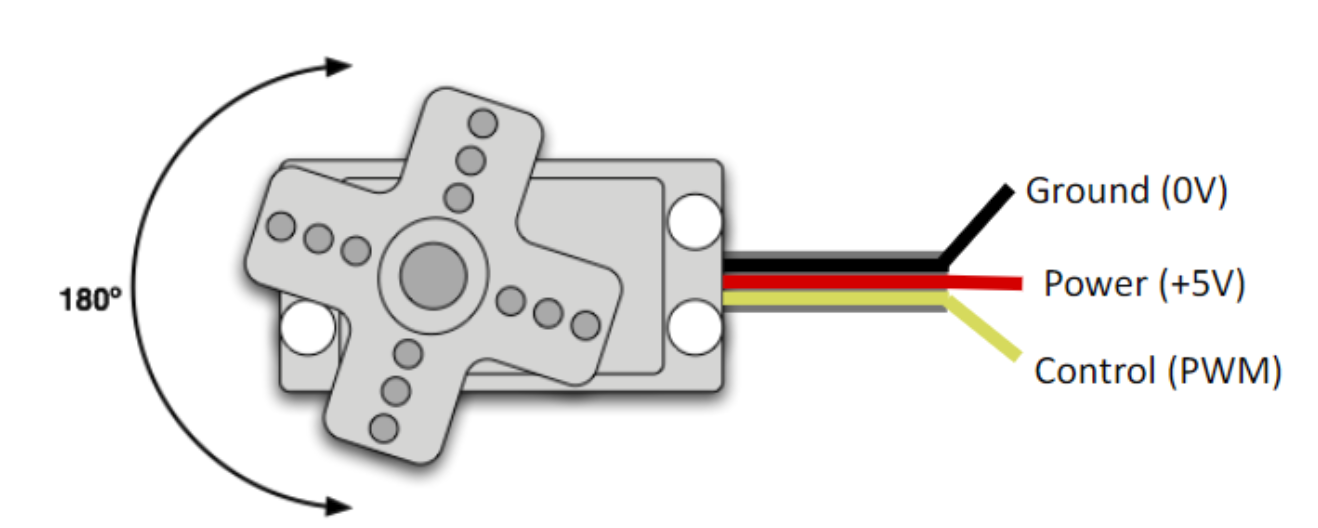
}

1. **SCHEMATICS**

**Task 1**

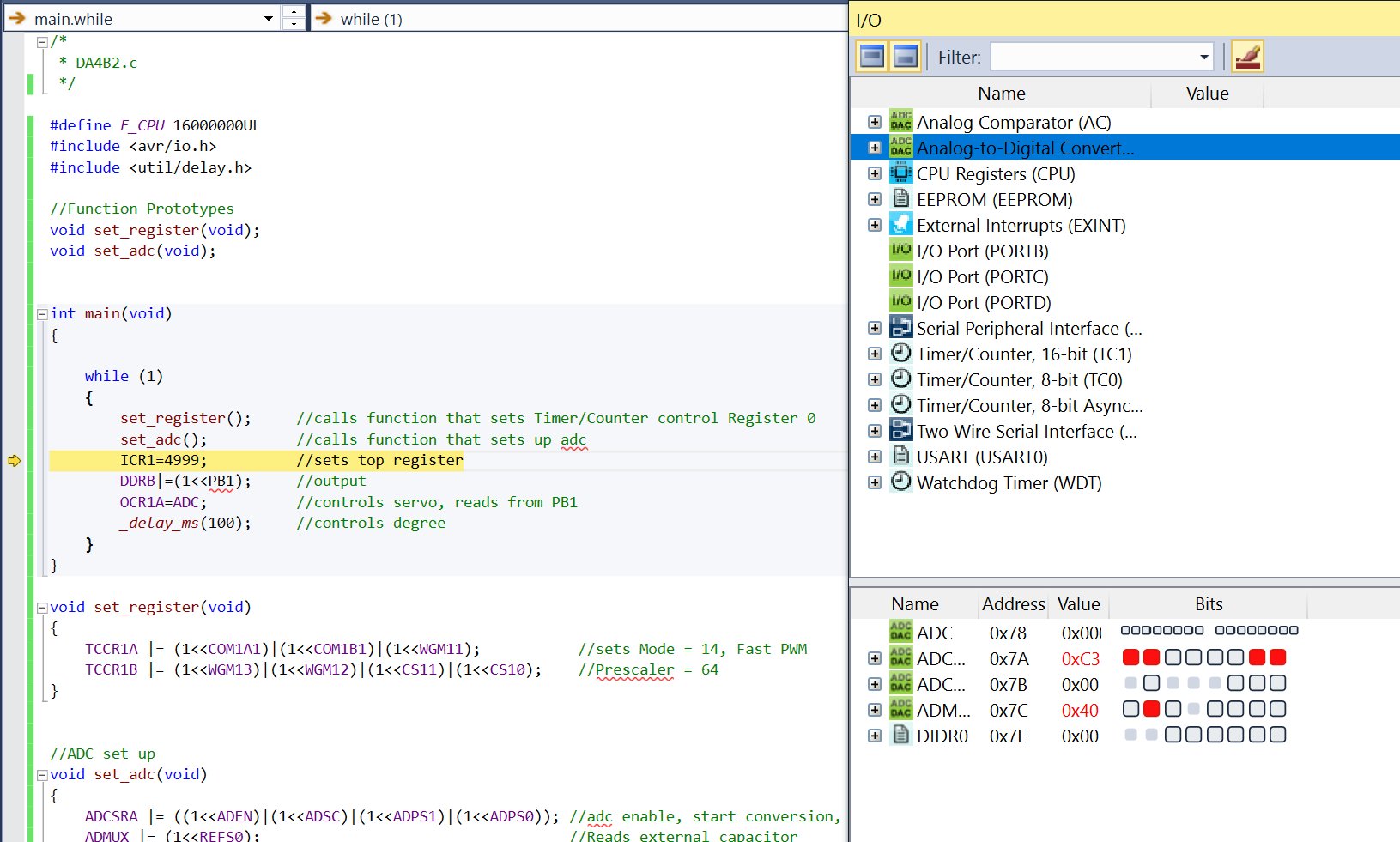


**Task 2**

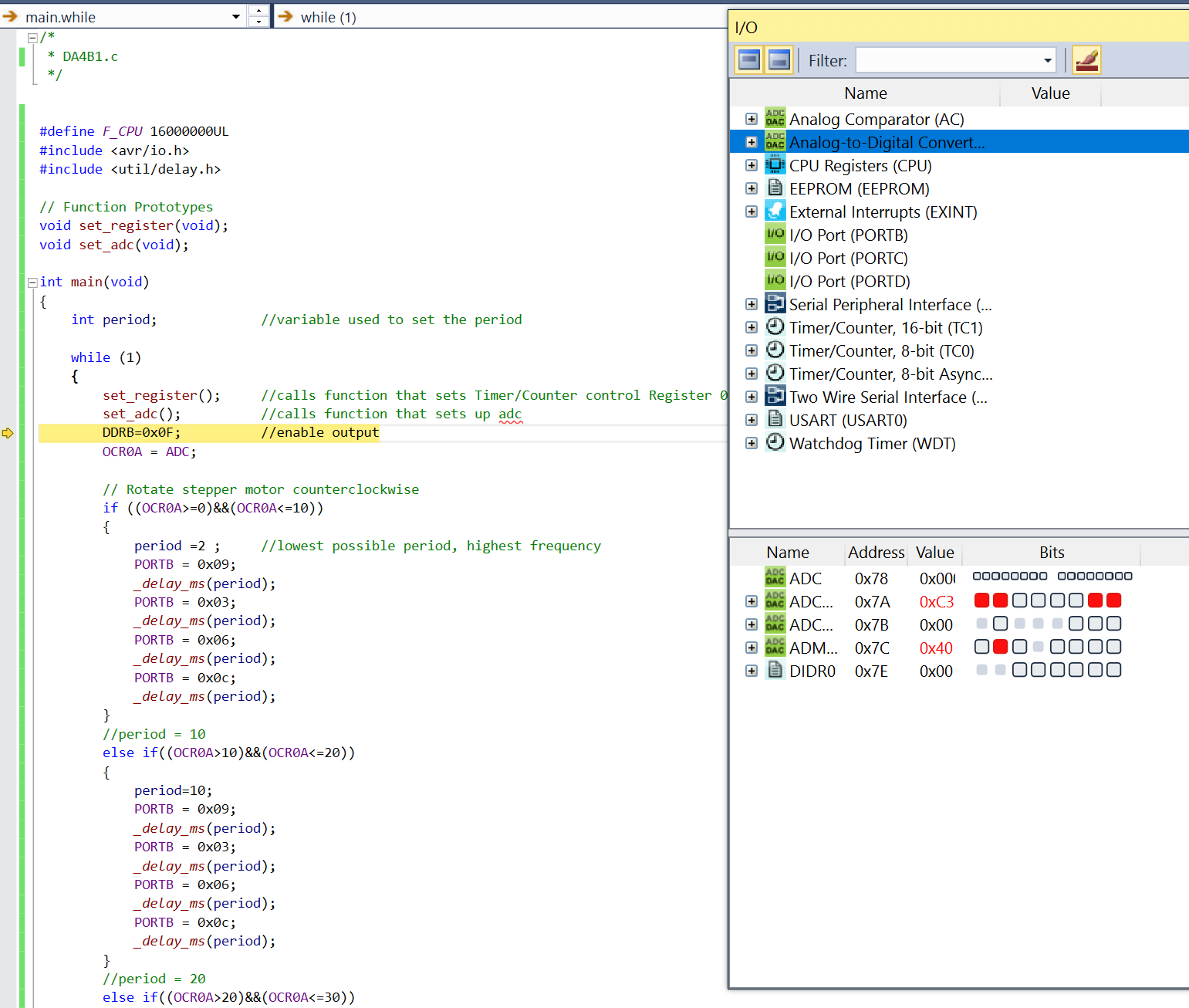


1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**

**Task 1**

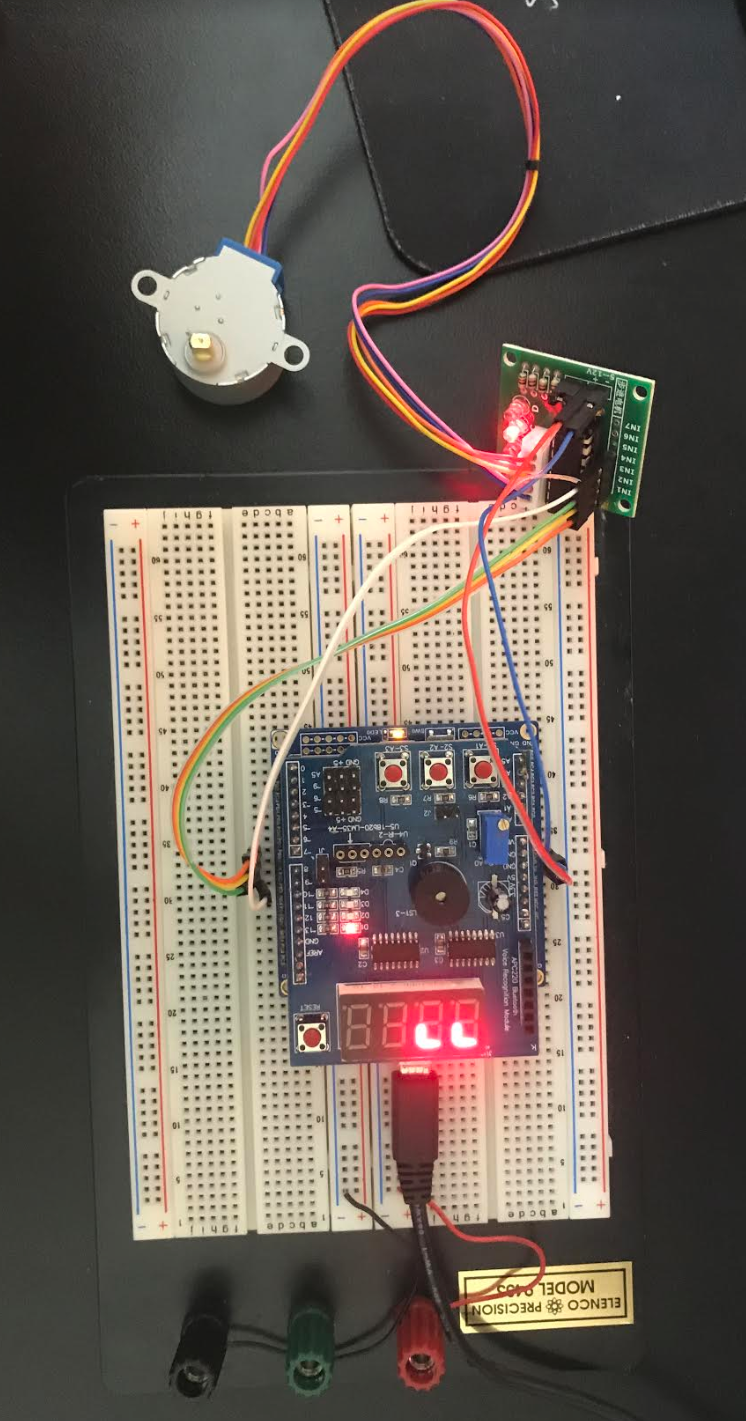


**Task 2**

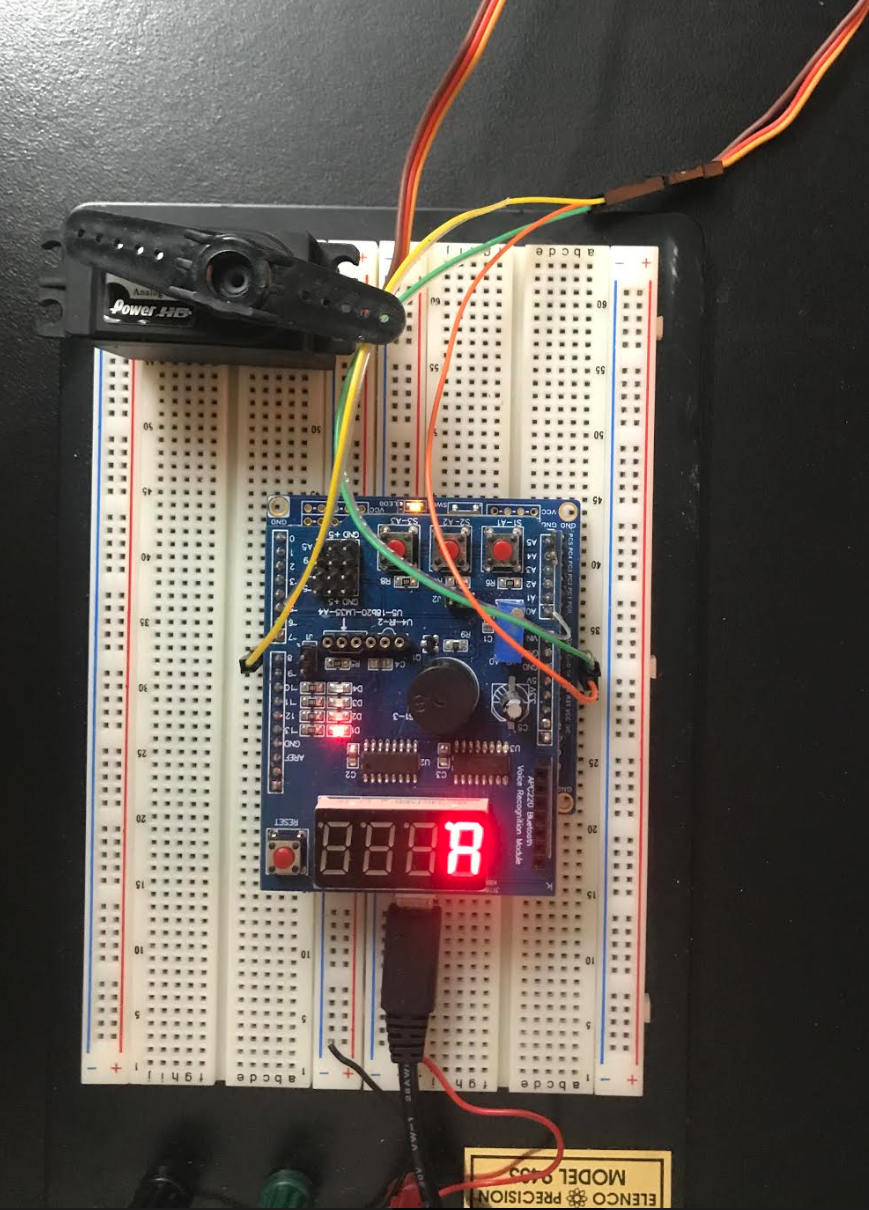


1. **SCREENSHOT OF EACH DEMO (BOARD SETUP)**

**Task 1**



**Task 2**



1. **VIDEO LINKS OF EACH DEMO**

**Task 1**

<https://youtu.be/4_65GW0trZE>

**Task 2**

<https://youtu.be/-jkIV01JFrQ>

1. **GITHUB LINK OF THIS DA**

<https://github.com/acexhp/submission_da.git>

**Student Academic Misconduct Policy**

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

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

Allis Hierholzer