CPE301 – SPRING 2019

Design Assignment 4B

Student Name: Saul Alejandro Mendoza Guzman
Student #: 2000540481
Student Email: mendos1@unlv.nevada.edu
Primary Github address: https://github.com/mendos1/subnission_da
Directory: DA4B

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 Xplained MINI
- Usb cable
- External power supply
- Jumper wires
- Potentiometer
- Servo motor
- Sepper mtor
- Atmel studio 7

2. INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1/B

```
#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>

void adc_int(void);
void timer_init(void);
volatile unsigned int speed; // Value of delay AKA speed
volatile int stop = 0;                 // Variable used to stop operation

int main(void)
{
	DDRB = 0x0F;		// Set PD0 - PD3 as outputs for stepper motor
	adc_int();              // Initialize ADC
	TCCR1B = 0x0D;		// Set CTC mode and 1024 prescaler
	while(1){
		while((ADCSRA&(1<<ADIF))==0); // wait for ADC conversion

		if (ADC <= 4)     {stop = 0; speed = 1;}
		if (ADC <= 85)    {stop = 0; speed = 2;}
		if (ADC <= 170)   {stop = 0; speed = 3;}
		if (ADC <= 255)   {stop = 0; speed = 4;}
		if (ADC <= 340)   {stop = 0; speed = 5;}
		if (ADC <= 425)   {stop = 0; speed = 6;}
		if (ADC <= 510)   {stop = 0; speed = 7;}
		if (ADC <= 595)   {stop = 0; speed = 8;}
		if (ADC <= 680)   {stop = 0; speed = 9;}
		if (ADC <= 765)   {stop = 0; speed = 10;}
		if (ADC <= 850)   {stop = 0; speed = 11;}
		if (ADC <= 935)   {stop = 0; speed = 12;}
		if (ADC <= 1015)  {stop = 0; speed = 13;}
		if (ADC >= 1016)   {stop = 1;}

		OCR1A = speed; // set OCR1A to the determined speed
		TCNT1 = 0x00;  // reset the clock
		if(stop == 0){
			// If the ADC value is not at its MAX value then step with desired delay
```

```
                        while((TIFR1 & 0x2) != 0x2);
                        PORTB = 0x09;
                        TIFR1 |= (1<<OCF1A);
                        while((TIFR1 & 0x2) != 0x2);
                        PORTB = 0x08;
                        TIFR1 |= (1<<OCF1A);
                        while((TIFR1 & 0x2) != 0x2);
                        PORTB = 0X0C;
                        TIFR1 |= (1<<OCF1A);
                        while((TIFR1 & 0x2) != 0x2);
                        PORTB = 0X04;
                        TIFR1 |= (1<<OCF1A);
                        while((TIFR1 & 0x2) != 0x2);
                        PORTB = 0x06;
                        TIFR1 |= (1<<OCF1A);
                        while((TIFR1 & 0x2) != 0x2);
                        PORTB = 0x02 ;
                        TIFR1 |= (1<<OCF1A);
                        while((TIFR1 & 0x2) != 0x2);
                        PORTB = 0X03;
                        TIFR1 |= (1<<OCF1A);
                        while((TIFR1 & 0x2) != 0x2);
                        PORTB = 0X01;
                        TIFR1 |= (1<<OCF1A);
                }
        }
}

void adc_int(void){
        ADMUX = (0<<REFS1)|    //      Reference Selection Bits
                        (1<<REFS0)|    //      AVcc-external cap at AREF
                        (0<<ADLAR)|    //      ADC     Left Adjust     Result
                        (0<<MUX3) |
                        (0<<MUX2) |    //      ANalogChannel Selection       Bits
                        (0<<MUX1) |    //      ADC0 (PC0)
                        (0<<MUX0);

        ADCSRA = (1<<ADEN)|    //      ADC     ENable
                        (1<<ADSC) |    //      ADC     Start Conversion
                        (1<<ADATE)|    //      ADC     Auto Trigger Enable
                        (0<<ADIF) |    //      ADC     Interrupt Flag
                        (1<<ADIE) |    //      ADC     Interrupt Enable
                        (1<<ADPS2)|    //      ADC     PrescalerSelectBits
                        (1<<ADPS1)|
                        (1<<ADPS0);
}
```

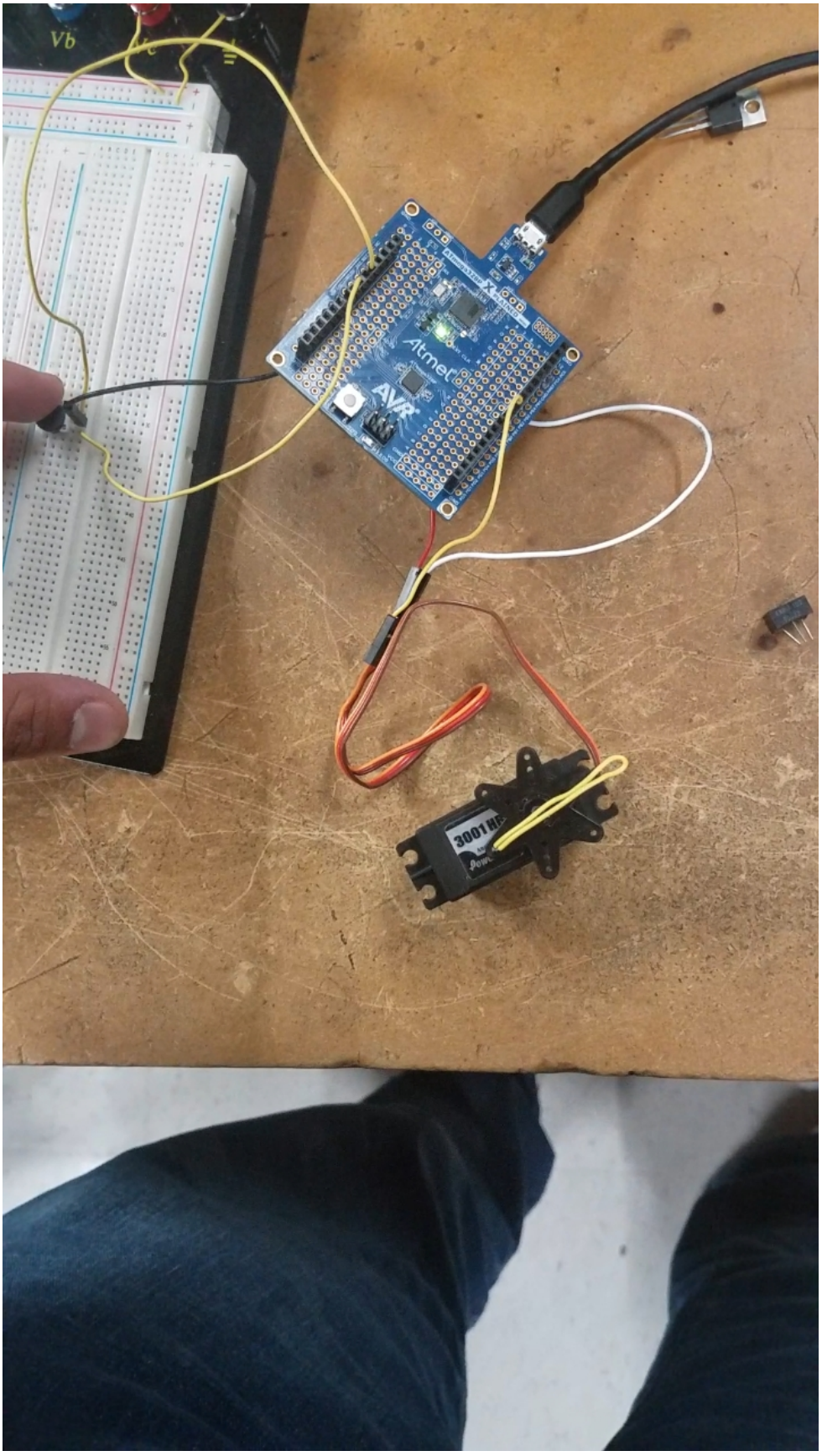
3. DEVELOPED CODE OF TASK 2/B

```
Insert only the modified sections here
#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>
```

```
#include <avr/interrupt.h>
int check = 0;
int main(void)

{


    DDRB = 0xFF; //DDRB as an output
        DDRD = 0xFF;
        TCCR1B |= (1<<WGM13)  | (1<<WGM12)  | (1<<CS11)  | (1<<CS10);
        TCCR1A |= (1<<COM1A1) | (1<<COM1B1) | (1<<WGM11);
        ICR1=4999;
        ADMUX = 0x60;
        ADCSRA = 0xA6;

    while (1)
    {
                ADCSRA |= ( 1 << ADSC); //start conversion
                while((ADCSRA & (1 << ADIF))== 0);
                check = ADCH;                   //temp value

                if(check == 0) // minimum value
                {
                        OCR1A = 0; //turn 0 deg
                        _delay_ms(500);
                }
                if(check == 255) // maximum pot value
                {
                        //PORTB = ( 1 << PORTB2);
                        OCR1A = 535;  //turn 180
                        _delay_ms(500);
                }
    }
}
```

4. SCHEMATICS

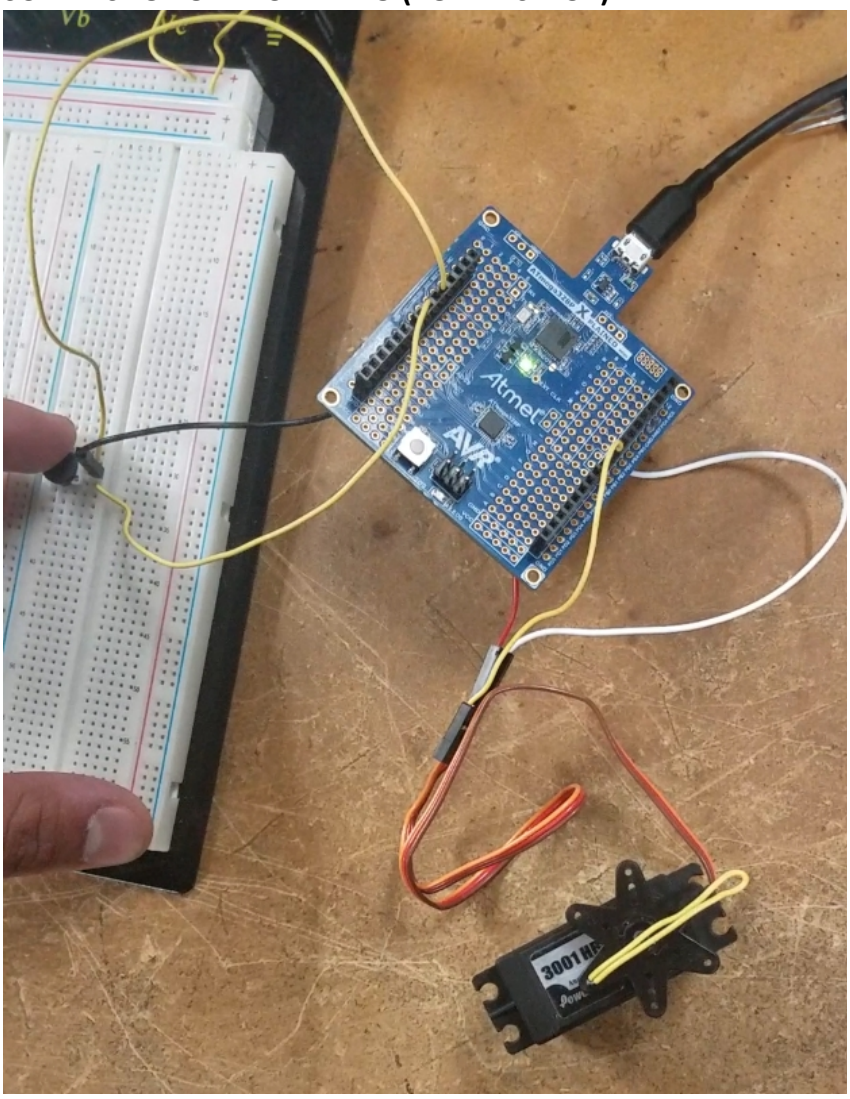
Use fritzing.org


Vb
Atmel
AVR

5. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)

None for this assignment

6. SCREENSHOT OF EACH DEMO (BOARD SETUP)



7. VIDEO LINKS OF EACH DEMO

STEPPER: https://www.youtube.com/watch?v=uRQB7bCZkSw
SERVO: https://www.youtube.com/watch?v=eugV6I61uaw

8. GITHUB LINK OF THIS DA

https://github.com/mendos1/subnission_da/tree/master/DA4B

Student Academic Misconduct Policy

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

"*This assignment submission is my own, original work*".

NAME OF THE STUDENT