

Design Assignment 4B

Student Name: Armon Latifi

Student #: 2000698173

Student Email: latifa1@unlv.nevada.edu

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

Directory: https://github.com/armonlatifi/sub_da/tree/master/DA4B

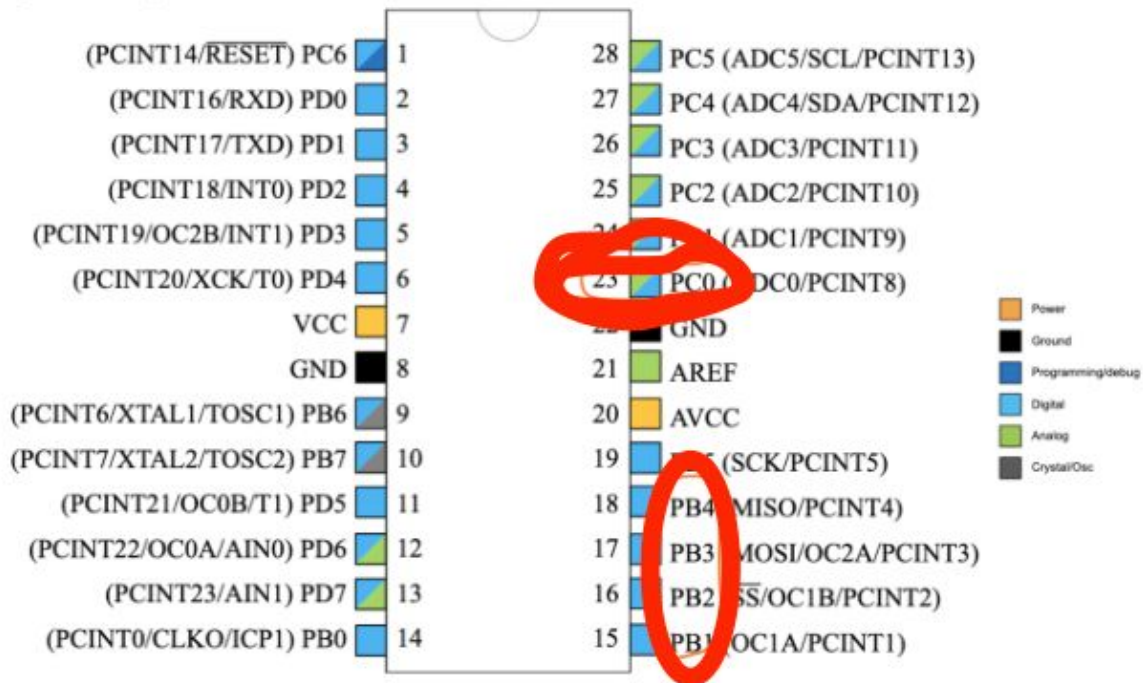
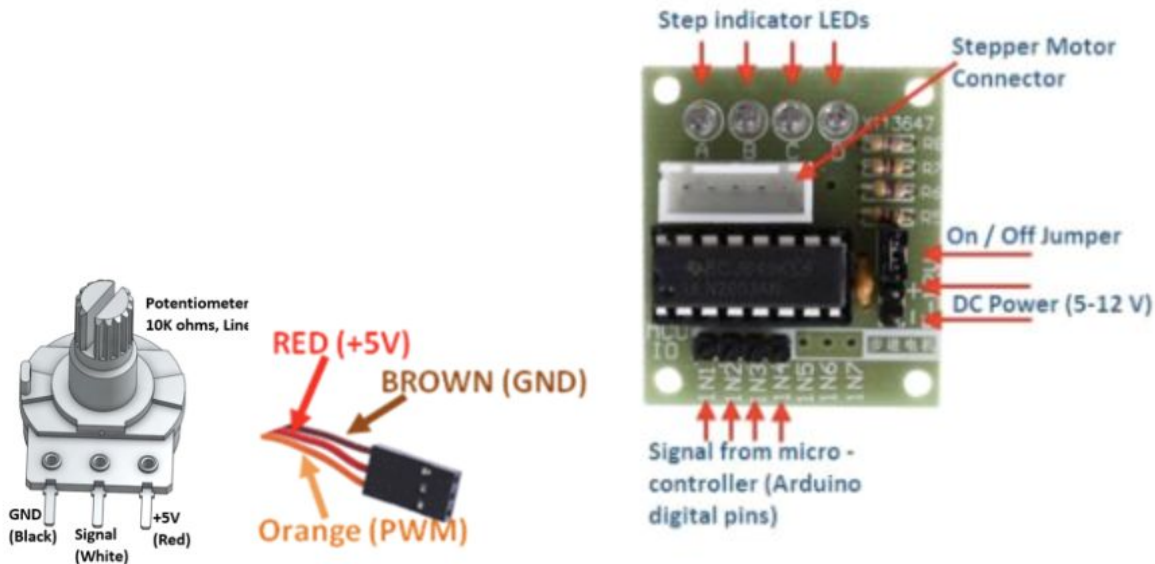
Submit the following for all Labs:

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

List of Components used:

- Assembler
- Simulator
- Debugger
- Breadboard
- Atmega328P
- Wires
- servo motor
- stepper motor
- driver
- potentiometer
- power supply
- Xplained mini
- Microusb cord
- Atmel studio 7



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

part_1.c

```
#define F_CPU 1000000UL //set clock speed
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
```

```
volatile uint8_t value; //volatile variable
```

```

ISR(ADC_vect)
{
    value = ADCH; //read and set pot
}

void delayFunct(void) //enable CTC mode
{
    TCNT1 = 0;
    OCR1A = value;
    TCCR1B |= ( 1<< WGM12) | (1 << CS12) | (1<< CS10);
    TCCR1A |= (1 << COM1A0);
    while(TIFR1 & ( 1 << OCF1A));

    TIFR1 |= ( 1 << OCF1A);
}

int main(void)
{
    DDRB = 0xFF;
    ADMUX = (0<<REFS1)|
            (1<<REFS0)|
            (0<<ADLAR)|
            (0<<MUX2)|
            (1<<MUX1)|
            (0<<MUX0);

    ADCSRA = (1<<ADEN) // enable adc
            (0<<ADSC)|
            (0<<ADATE)|
            (0<<ADIF)|
            (0<<ADIE)|
            (1<<ADPS2)|
            (0<<ADPS1)|
            (1<<ADPS0);

    sei(); //interrupt

    while(1)
    {
        PORTB |= (1<< PORTB0); //stepper
        delayFunct();           //CTC to timer
        PORTB |= (1<< PORTB1);
        delayFunct();
        PORTB |= (1<< PORTB2);
        delayFunct();
        PORTB |= (1<< PORTB3);
        delayFunct();

    }
}

```

part_2.c

```
#define F_CPU 1000000UL //set clock speed
```

```

#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>

int check = 0;

int main(void)
{
    DDRB = 0xFF; //port d as output
    DDRD = 0xFF;
    TCCR1B=3; //set a prescaler
    TCCR1A=0x83; //enable fast PWM mode
    ADMUX = 0x60;
    ADCSRA = 0xE6;

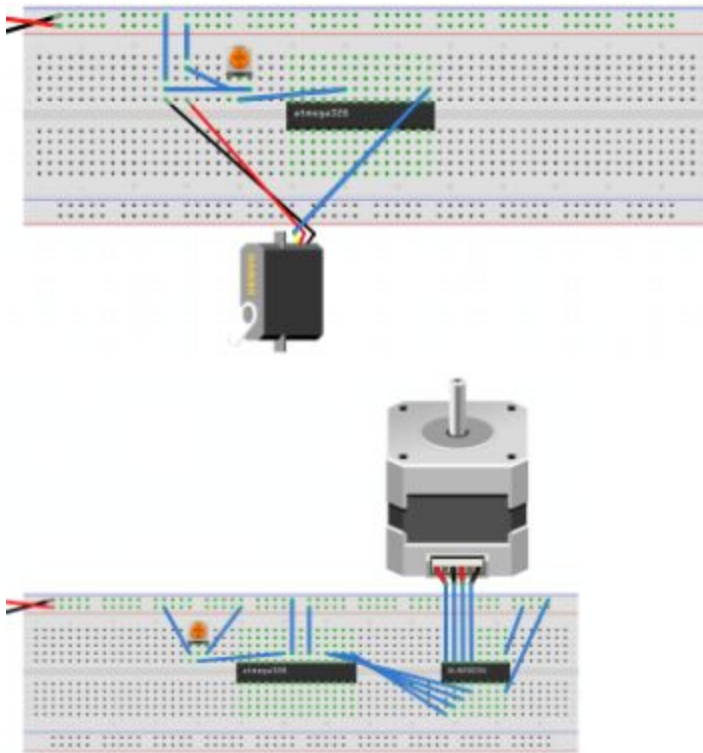
    while (1)
    {
        ADCSRA |= ( 1 << ADSC); //start conversion
        while((ADCSRA & (1 << ADIF))== 0);
        check = ADCH; //temporary variable

        if(check == 0) //minimum
        {
            OCR1A = 15;
            _delay_ms(1000);
        }
        else if(check == 255) //maximum
        {
            OCR1A = 30;
            _delay_ms(1000);
        }
        else;
    }
}

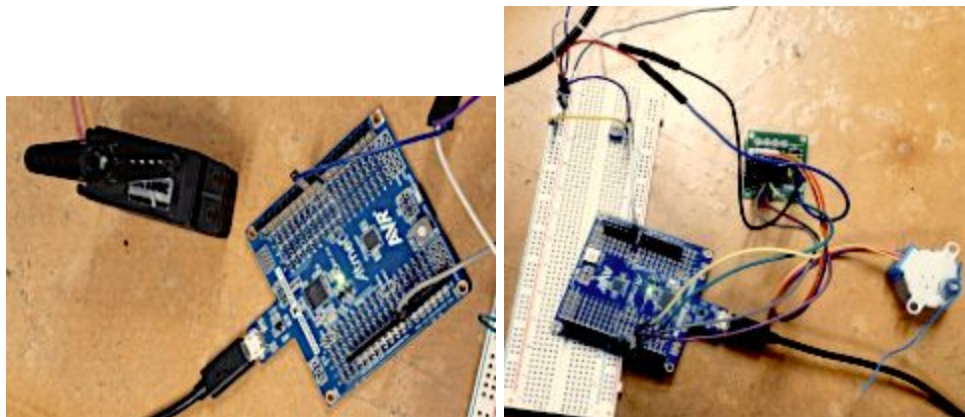
```

3. SCHEMATICS

Use fritzing.org



4. SCREENSHOT OF EACH DEMO (BOARD SETUP)



5. GITHUB LINK OF THIS DA

https://github.com/armonlatifi/sub_da/tree/master/DA4B

Student Academic Misconduct Policy

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

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

Armon Latifi