CPE301 – SPRING 2019

Design Assignment DA4B

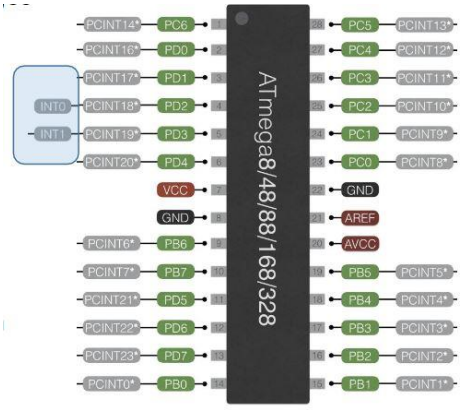
Student Name: Jett Guerrero
Student #: 5002377713
Student Email: guerrj1@unlv.nevada.edu
Primary Github address: https://github.com/guerrj1/Submission_DA.git
Directory: DA4B - https://github.com/guerrj1/Submission_DA/tree/master/DA4B

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

-ATMega328p
-Arduino Shield
-Male to male wires
-Male to female wires
-ULN2003A Driver
-Stepper motor
-Servo motor



Atmega328P using PB0, PB1, PB2, PB3 as output and PC0 for potentiometer

2. DEVELOPED CODE OF TASK 1 C CODE

```
//DA4BT1

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

volatile unsigned int ADCVal;               //variable that the ADC value is stored in
void adc_init(void);

int main(void)
{
	DDRB = 0x0F;                                                    //PB0-PB3 as output
	DDRC = 0;                                                               //set portc as input
	PORTB = 0;                                                              //turns off portb
	//Timer
	TCCR1B = (1 << WGM12) | (1 << CS11); //set prescalar to 8

	adc_init();

	while(1)
	{
		ADCSRA |= (1 << ADSC);                      //start adc conversion
		while((ADCSRA&(1<<ADIF))==0);//wait for conversion to finish
		ADCVal = ADC & 0x03FF;                      //read ADCH and ADCL
		OCR1A = 10*ADCVal;                          //duty cycle for PWM

		PORTB = 0x09;
		while(!(TIFR1 & (1<<OCF1A)));  //delay
```

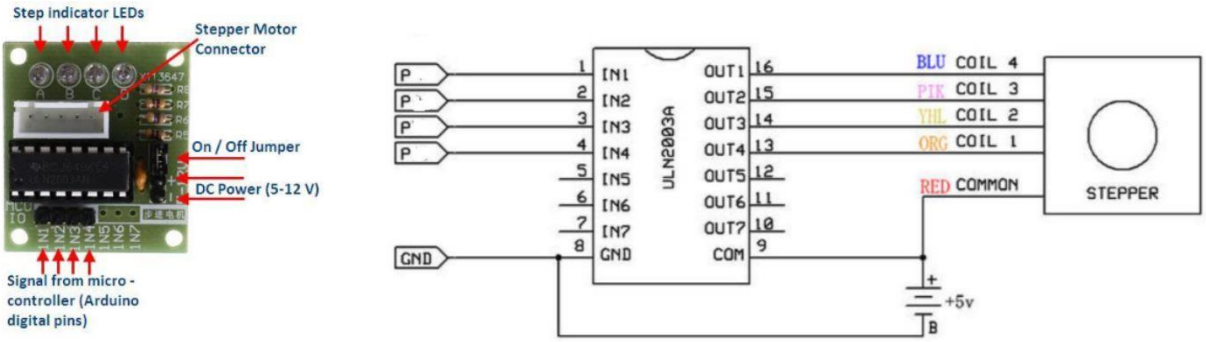
```
                TIFR1 |= (1 << OCF1A);          //reset flag
                PORTB = 0x03;
                while(!(TIFR1 & (1<<OCF1A)));   //delay
                TIFR1 |= (1 << OCF1A);          //reset flag
                PORTB = 0x06;
                while(!(TIFR1 & (1<<OCF1A)));   //delay
                TIFR1 |= (1 << OCF1A);          //reset flag
                PORTB = 0x0C;
                while(!(TIFR1 & (1<<OCF1A)));   //delay
                TIFR1 |= (1 << OCF1A);          //reset flag
        }
}

void adc_init (void)
{
        DIDR0 = 0x1;                                            //digital input disable
        ADMUX = (1<<REFS0);                                     //reference selection; AVcc

        ADCSRA |= (1<<ADEN) |                    //enable ADC
                  (1<<ADPS2) |                   //prescalar 128
                  (1<<ADPS1) |
                        (1<<ADPS0);
        ADCSRB = 0x0;                                           //adc control and status
register free running mode
}
```

3. DEVELOPED CODE OF TASK 2 C CODE

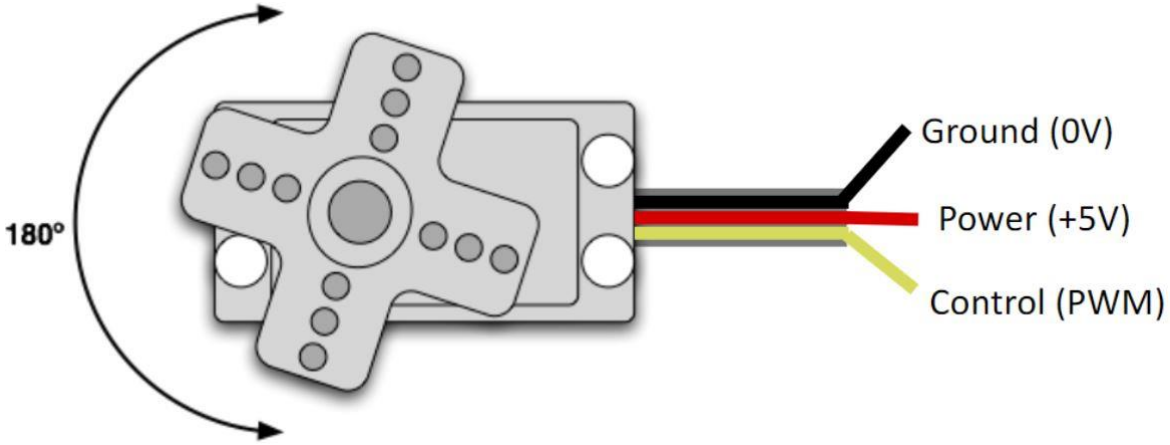
```
//DA2BT2

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

void adc_init(void);

int main(void)
{

        while(1)
        {
                //Timer1
                TCCR1A = (1 << COM1A1) | (1<<COM1B1) | (1<<WGM11);
                TCCR1B = (1<<WGM13) | (1<<WGM12) | (1<<CS11) | (1<<CS10);     //set
prescalar to 64

                adc_init();

                ICR1=4999;              //50Hz
                DDRB |= (1<<PB1);           //output pin to the servo motor
                OCR1A = ADC;            //for potentiometer
                _delay_ms(50);          //delay between pot input and output of servo motor
        }
}
```

```
void adc_init (void)
{
	ADMUX = (1<<REFS0);                                    //reference selection; AVcc

	ADCSRA |= (1<<ADEN) |                  //enable ADC
		  (1<<ADSC) |
			(1<<ADPS2) |                    //prescalar 128
		  (1<<ADPS1) |
		  (1<<ADPS0);
	ADCSRB = 0x0;                                          //adc control and status
register free running mode
}
```

4. SCHEMATICS



Stepper motor Connection Schematic. IN1-IN4 connected to PB0-PB3.

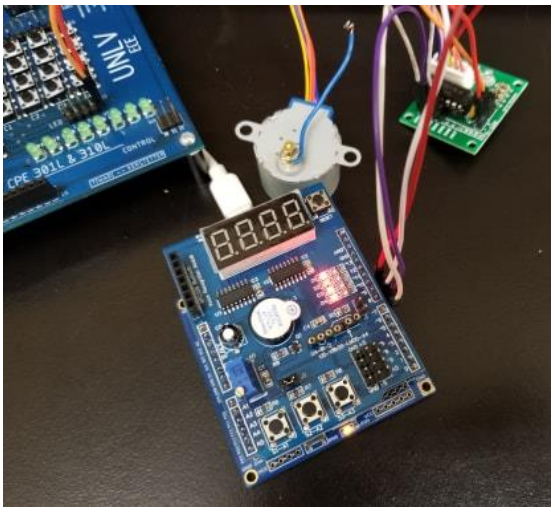


Servo motor connection schematic. Control wire connected to PB1.

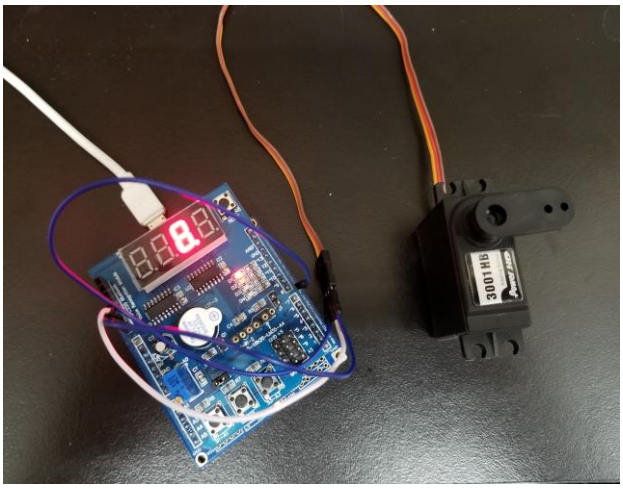
5. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)

N/A

6. SCREENSHOT OF EACH DEMO (BOARD SETUP)



Stepper motor board set up with Atmega328p



Servo motor board set up with Atmega328p

7. VIDEO LINKS OF EACH DEMO

Task 1: Stepper Motor
https://youtu.be/HDzUFmH0Z0A

Task 2: Servo Motor
https://youtu.be/miwpquWKJE8

8. GITHUB LINK OF THIS DA

https://github.com/guerrj1/Submission_DA/tree/master/DA4B

Student Academic Misconduct Policy
http://studentconduct.unlv.edu/misconduct/policy.html

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