CPE301 – SPRING 2019

Design Assignment 4B

Student Name: Ivan Soto

Student #: 2000921825

Student Email: sotoi2@unlv.nevada.edu

Primary Github address: <https://github.com/sotoi2/submission_da>

Directory: <https://github.com/sotoi2/submission_da/tree/master/ESD301/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

Block diagram with pins used in the Atmega328P

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

Insert initial code here

Task 1

/\*

\* DA4B.c

\* Author : Ivan

\*/

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <util/delay.h>

volatile unsigned int ADCnumber;

int main(void)

{

timer\_init();

ADC\_init();

while(1)

{

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

while((ADCSRA&(1<<ADIF))==0); //get the value and poll

ADCnumber = ADC & 0x03FF; //get the bits from the reg

OCR1A = 10\*ADCnumber; // make the DC

PORTB = 0x09;

while (!(TIFR1 & (1<<OCF1A)));

TIFR1 |= (1<<OCF1A);

PORTB = 0x03;

while (!(TIFR1 & (1<<OCF1A)));

TIFR1 |= (1<<OCF1A);

PORTB = 0x06;

while (!(TIFR1 & (1<<OCF1A)));

TIFR1 |= (1<<OCF1A);

PORTB = 0x0C;

while (!(TIFR1 & (1<<OCF1A)));

TIFR1 |= (1<<OCF1A);

}

}

void timer\_init(void)

{

DDRB = 0xFF; //make port b outputs

DDRC = 0; //make the port c an input

PORTB = 0; // turn off port b

TCCR1B = (1<< WGM12) | (1<<CS11); //pre scalar is 8

}

void ADC\_init(void) // ADC conversion

{

DIDR0 = 0x1;

ADMUX = (1<<REFS0);

ADCSRA |= (1<<ADEN)|(1<<ADPS2)|

(1<<ADPS1)|(1<<ADPS0);

ADCSRB = 0x0;

}

1. **DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A**

Insert only the modified sections here

Task 2:

/\*

\* DA4B.2.c

\*\* Author : Ivan

\*/

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <util/delay.h>

void adc\_init(void);

int main(void)

{

while(1) // set up our timer

{

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

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

adc\_init();

ICR1=5000; //50Hz

DDRB |= (1<<PB1); //use portb.1 as out

OCR1A = ADC; //potentiometer

*\_delay\_ms*(50);

}

}

void adc\_init (void)

{

ADMUX = (1<<REFS0); //avcc

ADCSRA |= (1<<ADEN) | //enable our ADC

(1<<ADSC) |

(1<<ADPS2) | //prescalar 128

(1<<ADPS1) |

(1<<ADPS0);

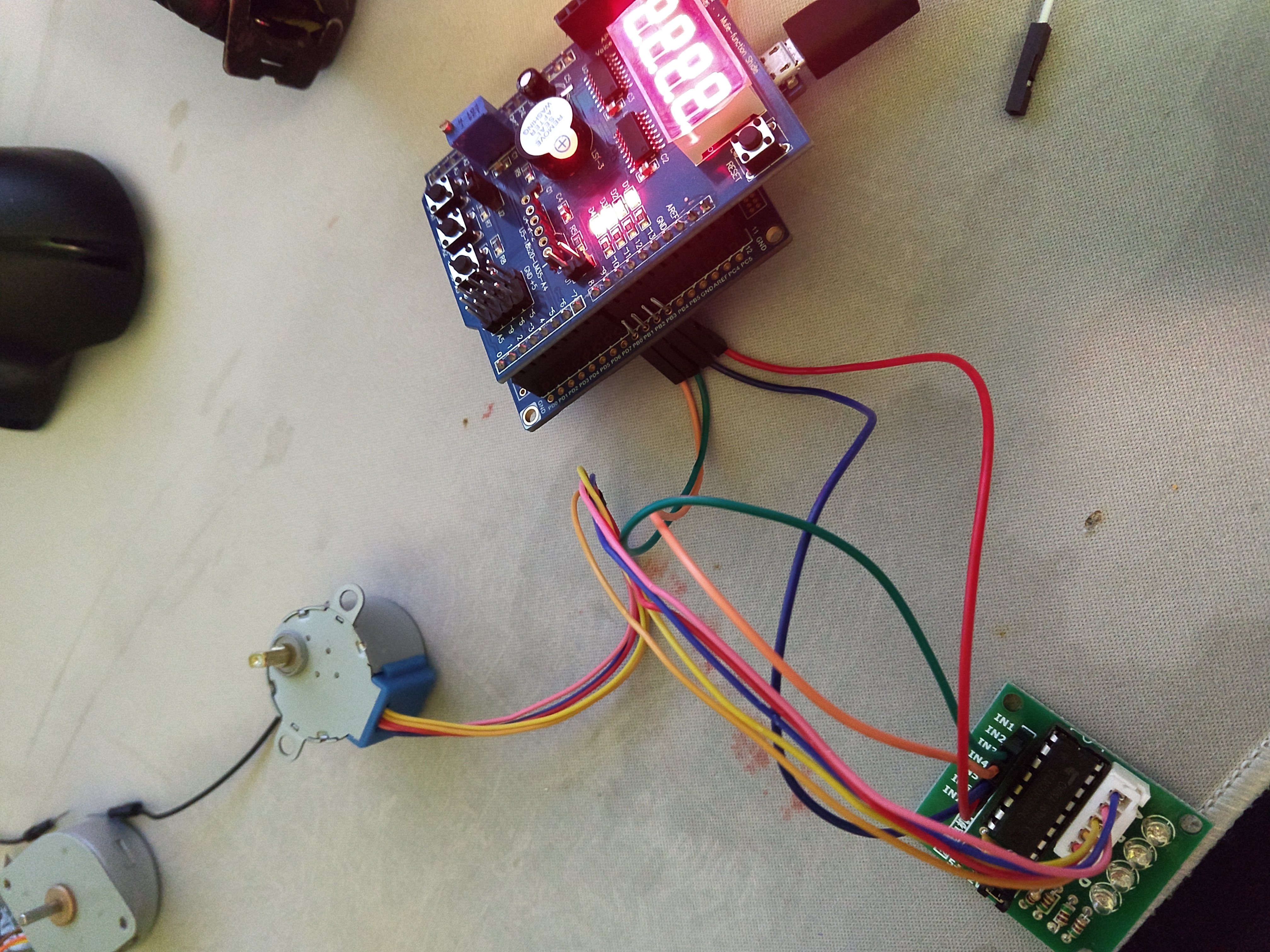
ADCSRB = 0x0; //free run mode

}

1. **SCHEMATICS**

Use fritzing.org

1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**
2. **SCREENSHOT OF EACH DEMO (BOARD SETUP)**



1. **VIDEO LINKS OF EACH DEMO**

Task 1: unable to get motor moving, running out of time.

T2: <https://youtu.be/2AIoCqfI8Pw>

1. **GITHUB LINK OF THIS DA**

<https://github.com/sotoi2/submission_da/tree/master/ESD301/DA4B>

**Student Academic Misconduct Policy**

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

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

Ivan Soto