CPE301 – SPRING 2019

Design Assignment 3B

Student Name: Chris Barr

Student #: 2000682859

Student Email: barrc1@unlv.nevada.edu

Primary Github address: https://github.com/BarrChris

Directory: https://github.com/BarrChris/submission\_da.git

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).

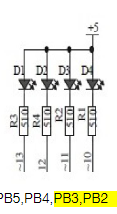
1. **COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS**

List of Components used

* Atmega328P
* Multi-Function Shield
* FTDI Basic
* 4 M-M wires
* 3 F-F wires
* LM34

Block diagram with pins used in the Atmega328P





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

/\*

\* task1.c

\*

\* Created: 4/20/2019 3:36:08 PM

\* Author : Chris

\*/

#define F\_CPU 16000000UL

#include <avr/io.h>

#include <util/delay.h>

volatile unsigned int ADCVal**;** // Holds ADC Register value

int main**(**void**)**

**{**

int period**;**

timer\_init**();**

ADC\_init**();**

**while(**1**)**

**{**

ADCSRA **|=** **(**1 **<<** ADSC**);** // Starts conversion

**while((**ADCSRA**&(**1**<<**ADIF**))==**0**);** // Waits for conversion

ADCVal **=** ADC **&** 0x03FF**;** // Takes right 10-bits of ADC register

OCR1A **=** 10**\***ADCVal**;** // OCR1A value for duty cycle

PORTB **=** 0x09**;** // Forward the step

**while(!(**TIFR1 **&** **(**1**<<**OCF1A**)));** // Delays using CTC

TIFR1 **|=** **(**1 **<<** OCF1A**);** // Resets flag, now it repeats 3 times

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**)**

**{**

//Port directions

DDRB **=** 0x0F**;** // Set PORTB3:0 as outputs

DDRC **=** 0**;** // PORTC as input

PORTB **=** 0**;** // Initially turns off all PB5:0

//Timer1

//TCCR1A = (1 << COM1A1);

TCCR1B **=** **(**1 **<<** WGM12**)** **|** **(**1 **<<** CS11**);** // Prescaler = 8

**}**

void ADC\_init **(**void**)**

**{**

// ADC initialization

DIDR0 **=** 0x1**;** // disable digital input on ADC0 pin

ADMUX **=** **(**1**<<**REFS0**);** // Reference = Aref, ADC0 (PC.0) used as analog input

// right-justified data

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

**(**1**<<**ADPS1**)** **|** **(**1**<<**ADPS0**);** // enable ADC, system clock used for ADC

ADCSRB **=** 0x0**;** // free running mode

**}**

/\*

\* task2.c

\*

\* Created: 4/20/2019 3:36:08 PM

\* Author : Chris

\*/

#define F\_CPU 16000000UL

#include <avr/io.h>

#include <util/delay.h>

void timer\_init**(**void**);**

void ADC\_init**(**void**);**

int main**(**void**)**

**{**

timer\_init**();**

ADC\_init**();**

**while(**1**)**

**{**

ADC\_init**();**

ICR1 **=** 4999**;** // fPWM = 50Hz

DDRB **|=** **(**1**<<**PB1**);** // Enables PB1 as output

OCR1A **=** ADC**;** // Grabs value from potentiometer to timer1

\_delay\_ms**(**30**);** // Delays from potentiometer to relay

**}**

**}**

void timer\_init**(**void**)**

**{**

//Timer1

TCCR1A **=** **(**1 **<<** COM1A1**)** **|** **(**1**<<**COM1B1**)** **|** **(**1**<<**WGM11**);** // Non-inverting Mode, Fast PWM Mode

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

**}**

void ADC\_init **(**void**)**

**{**

// ADC initialization

ADMUX **=** **(**1**<<**REFS0**);** // Reference = Aref, ADC0 (PC.0) used as analog input

// right-justified data

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

**(**1**<<**ADPS1**)** **|** **(**1**<<**ADPS0**);** // enable ADC, system clock used for ADC

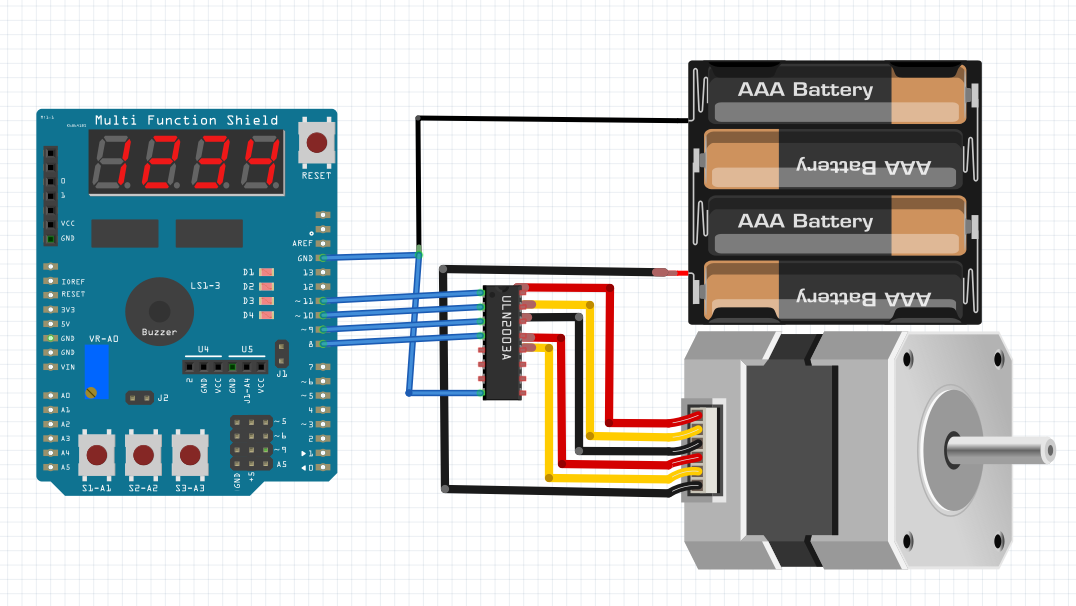
ADCSRB **=** 0x0**;** // free running mode

**}**

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

None modified

1. **SCHEMATICS**

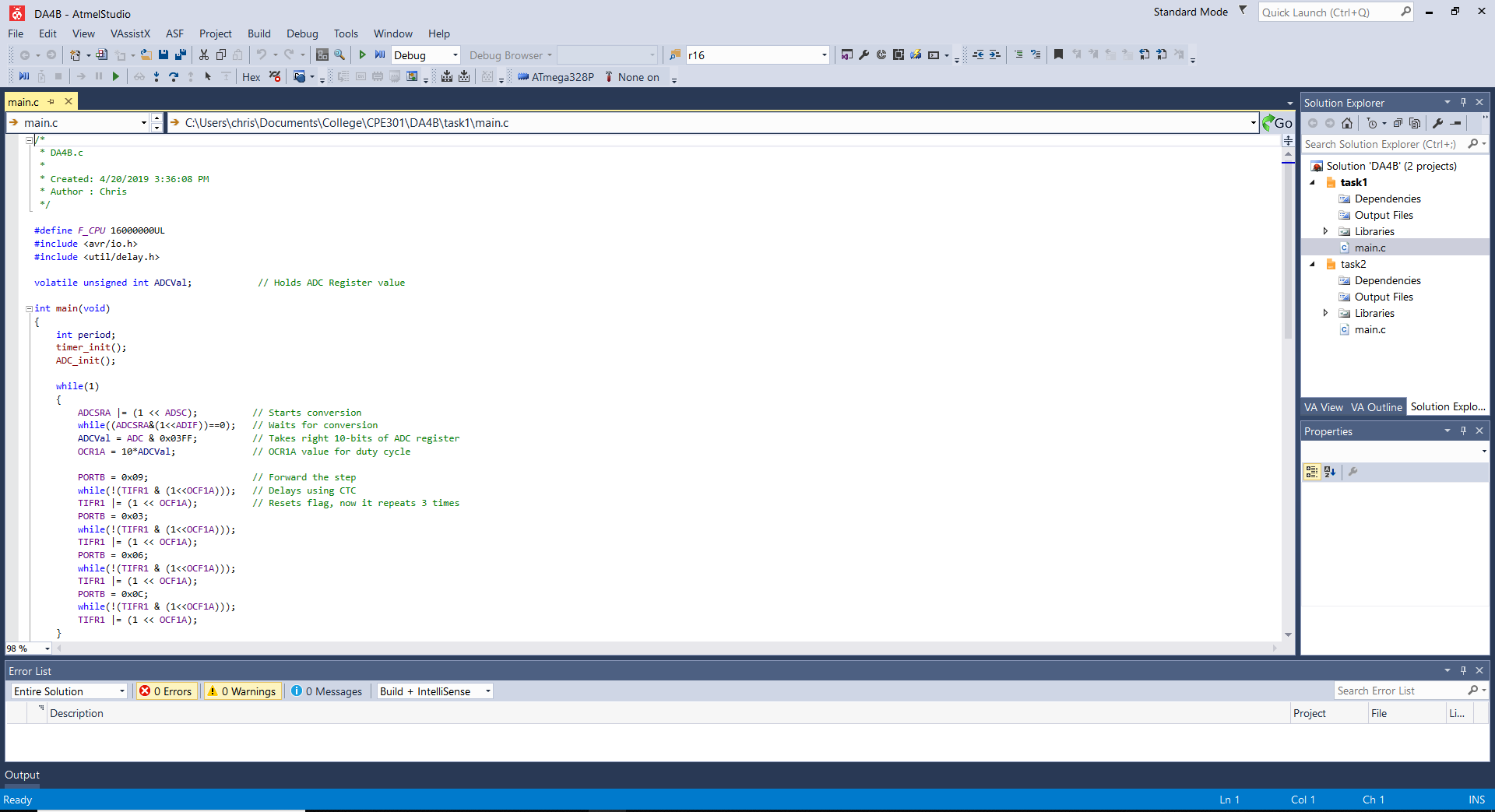


Stepper Motor



Servo Motor (Note: Couldn’t find exact motor, but essentially the same setup)

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



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



Task 1



Task 2

1. **VIDEO LINKS OF EACH DEMO**

Task 1: <https://www.youtube.com/watch?v=9LII_X_6B8Y>

Task 2: <https://www.youtube.com/watch?v=XcqiHuqZ2b4>

1. **GITHUB LINK OF THIS DA**

<https://github.com/BarrChris/submission_da/tree/master/DesignAssignments/DA4B>

**Student Academic Misconduct Policy**

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

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

Chris Barr