Damian Cisneros

CPE301 – SPRING 2018

Design Assignment 4

**DO NOT REMOVE THIS PAGE DURING SUBMISSION:**

The student understands that all required components should be submitted in complete for grading of this assignment.

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **SUBMISSION ITEM** | **COMPLETED (Y/N)** | **MARKS**  **(/MAX)** |
| 1 | COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS |  |  |
| 2. | INITIAL CODE OF TASK 1/A |  |  |
| 3. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 2/B |  |  |
| 3. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 3/C |  |  |
| 3. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 4/D |  |  |
| 3. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 5/E |  |  |
| 4. | SCHEMATICS |  |  |
| 5. | SCREENSHOTS OF EACH TASK OUTPUT |  |  |
| 5. | SCREENSHOT OF EACH DEMO |  |  |
| 6. | VIDEO LINKS OF EACH DEMO |  |  |
| 7. | GOOGLECODE LINK OF THE DA |  |  |
|  |  |  |  |
|  |  |  |  |

Damian Cisneros

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

List of Components used:

ATmega328p

LM34

Breadboard

10kΩ resistor

DC motor

Stepper motor

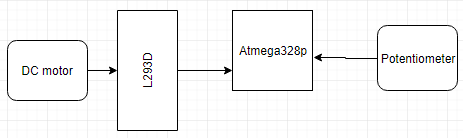
Servo motor

Pushbutton

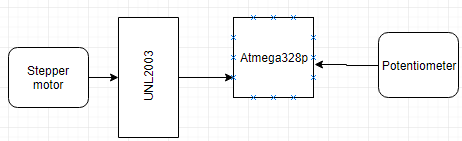
L293D

ULN2003

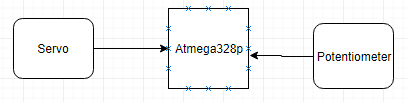
10k potentiometer



Block diagram with for Task 1(DC motor)



Block diagram with for Task 2(Stepper motor)



Block diagram with for Task 3(Servo motor)

1. **INITIAL/DEVELOPED CODE OF TASK 1/C Code**

/\*

/\*

\* DA4\_Task1.c

\*

\* Created: 4/11/2018 5:07:49 PM

\* Author : Damian Cisneros

\*/

#define F\_CPU 8000000UL

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

int temp**=**0**;**

ISR**(**INT0\_vect**){**

EIFR **|=** **(**INTF0 **<<** 1**);** // clear INT0 flag

\_delay\_ms**(**10**);**

//toggle on and off

**if(**temp **==** 0**)**

temp **=** 1**;**

**else** **if(**temp **==** 1**)**

temp **=** 0**;**

**}**

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

**{**

DDRB **|=** **(**1 **<<** PINB1**);** //use PINB1 as output for input of motor driver

DDRD **|=** **(**1 **<<** PIND0**);** //use PIND0 as output for input of driver enable

EIMSK **=** **(**1 **<<** INT0**);** //activate external interrupt

EICRA **=** **(**1 **<<** ISC01 **)** **|** **(**1 **<<** ISC00**);** //activate external interrupt on rising edge

TCCR1B **|=** **(**1 **<<** WGM12**)** **|** **(**1 **<<** CS12**)** **|** **(**0 **<<** CS11**)** **|** **(**1 **<<** CS10**);** //prescale to 1024

TCCR1A **|=** **(**1 **<<** COM1B1**)** **|** **(**1 **<<** COM1A1**)** **|** **(**1 **<<** WGM11**)** **|** **(**1 **<<** WGM10**);** //set mode to PWM with TOP 1024

//set channel to take input for ADC0(changed to ADC1),left adjust,AVcc with external cap at AREF

ADCSRA **|=** **(**1 **<<** ADEN**)** **|** **(**1 **<<** ADPS2**)** **|** **(**1 **<<** ADPS1**)** **|** **(**1 **<<** ADSC**)** **|** **(**1 **<<** ADATE**);**

ADMUX **|=** **(**1 **<<** MUX0**);** //ADC prescaler 64, enable ADC,Start conversion,enable auto trigger (free running mode)

sei**();**

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

**{**

volatile int ad;

while ((ADCSRA & (1 << ADIF)) == 0); //check if ADIF is 1, if not stay here

ADCSRA |= (1 << ADIF); //set ADIF 1

ad = ADC << 2; //read conversion from ADC register

OCR1A = ad;

if(temp == 1) //if on turn on enable

PORTD = 1;

else if(temp == 0) //if off turn off enable

PORTD = 0;

}

}

1. **INITIAL/DEVELOPED CODE OF TASK 2/C Code**

#define F\_CPU 8000000UL

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

int temp**;** //variable to hold ADC value

ISR**(**ADC\_vect**){**

ADCSRA **|=** **(**1 **<<** ADIF**);** //set ADIF 1

temp **=** ADCH**;** //read conversion from ADCH register

//PORTD = temp; //output ADC value

//OCR1A = temp;

**}**

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

**{**

DDRD **=** 0b00001111**;** //set PIND0-PIND3 to output

//set to CTC mode

TCCR1A **=** 0**;** //set WGM11 = 0, WGM10 = 0

TCCR1B **=** 13**;** //set WGM12 = 1, WGM13 = 0, prescale = 1024

OCR1A **=** 23000**;** //set Output Compare Register to calculated count(compared

ADCSRA **|=** **(**1 **<<** ADEN**)** **|** **(**1 **<<** ADPS2**)** **|** **(**1 **<<** ADPS1**)** **|** **(**1 **<<** ADSC**)** **|** **(**1 **<<** ADATE**)** **|** **(**1 **<<** ADIE**);**//enable ADC interrupt set channel to take input for ADC0(changed to ADC1),left adjust,AVcc with external cap at AREF (You can ignore connecting external capacitor at AREF)

ADMUX **|=** **(**1 **<<** ADLAR**)** **|** **(**1 **<<** REFS0**)** **|** **(**1 **<<** MUX0**);** //ADC prescaler 64, enable ADC,Start conversion,enable auto trigger (free running mode)

sei**();**

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

**{**

//if((TIFR1 & (1 << OCF1A)) == 0){ //check if flag is set

**for(**int i**=**0**;**i**<**temp**;**i**++)**

\_delay\_ms**(**1**);**

PORTD **=** 1**;** //black

**for(**int i**=**0**;**i**<**temp**;**i**++)**

\_delay\_ms**(**1**);**

PORTD **=** 2**;** //red

**for(**int i**=**0**;**i**<**temp**;**i**++)**

\_delay\_ms**(**1**);**

PORTD **=** 4**;** //green

**for(**int i**=**0**;**i**<**temp**;**i**++)**

\_delay\_ms**(**1**);**

PORTD **=** 8**;** //blue

TIFR1 **|=** **(**1 **<<** OCF1A**);** //clear flag

**}**

**}**

1. **INITIAL/DEVELOPED CODE OF TASK 3/C Code**

#define F\_CPU 8000000UL

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

int temp**=**0**;**

void Wait**()**

**{**

uint8\_t i**;**

**for(**i**=**0**;**i**<**1**;**i**++)**

**{**

\_delay\_loop\_2**(**0**);**

\_delay\_loop\_2**(**0**);**

\_delay\_loop\_2**(**0**);**

**}**

**}**

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

**{**

DDRB **|=** **(**1 **<<** PINB1**);** //OCR1A as output to servo

TCCR1A **=** 0x82**;** //non inverted PWM

TCCR1B **=** 0x1A**;** //prescale 8 Fast PWM mode

ICR1 **=** 19999**;** //(8Mhz/(8\*50Hz)-1) = 20ms period

ADCSRA **|=** **(**1 **<<** ADEN**)** **|** **(**1 **<<** ADPS1**)** **|** **(**1 **<<** ADPS0**)** **|** **(**1 **<<** ADSC**)** **|** **(**1 **<<** ADATE**)** **;**//enable ADC interrupt set channel to take input for ADC0(changed to ADC1),left adjust,AVcc with external cap at AREF (You can ignore connecting external capacitor at AREF)

ADMUX **|=** **(**1 **<<** REFS0**)** **|** **(**1 **<<** MUX0**);** //ADC prescaler 8, enable ADC,Start conversion,enable auto trigger (free running mode)

sei**();**

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

**while** **((**ADCSRA **&** **(**1 **<<** ADIF**))** **==** 0**);** //check if ADIF is 1, if not stay here

ADCSRA **|=** **(**1 **<<** ADIF**);** //set ADIF 1

temp **=** ADC**;** //read conversion from ADCH register

//divide = 1/1023 = .000977517;

temp **=** 1800**\*(**1**-(**0.000977517**\***temp**));**

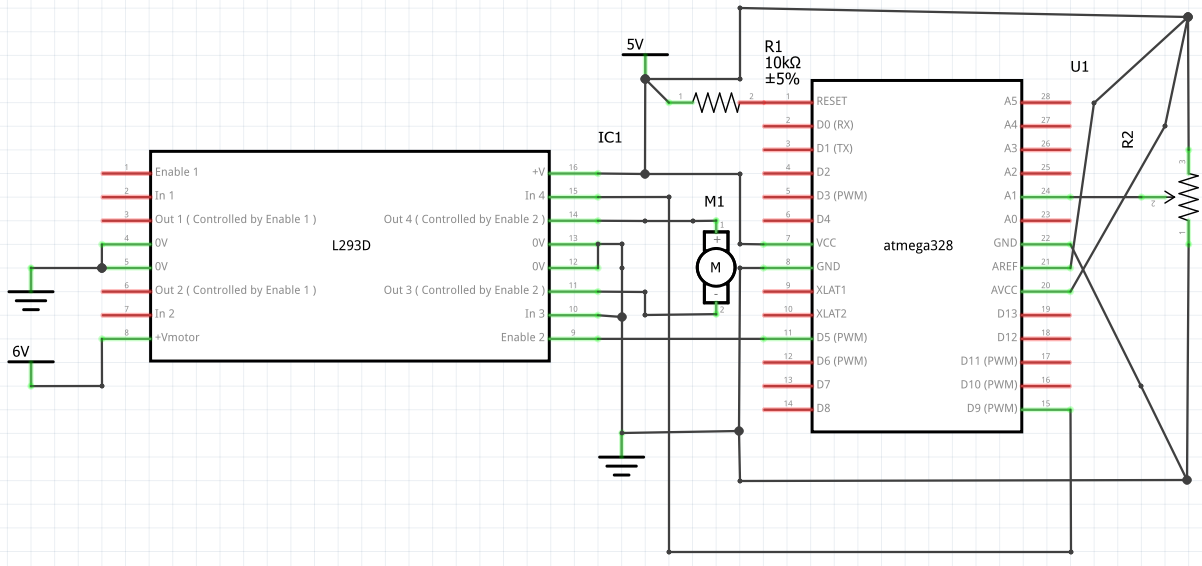
OCR1A **=** temp**+**400**;**

Wait**();**

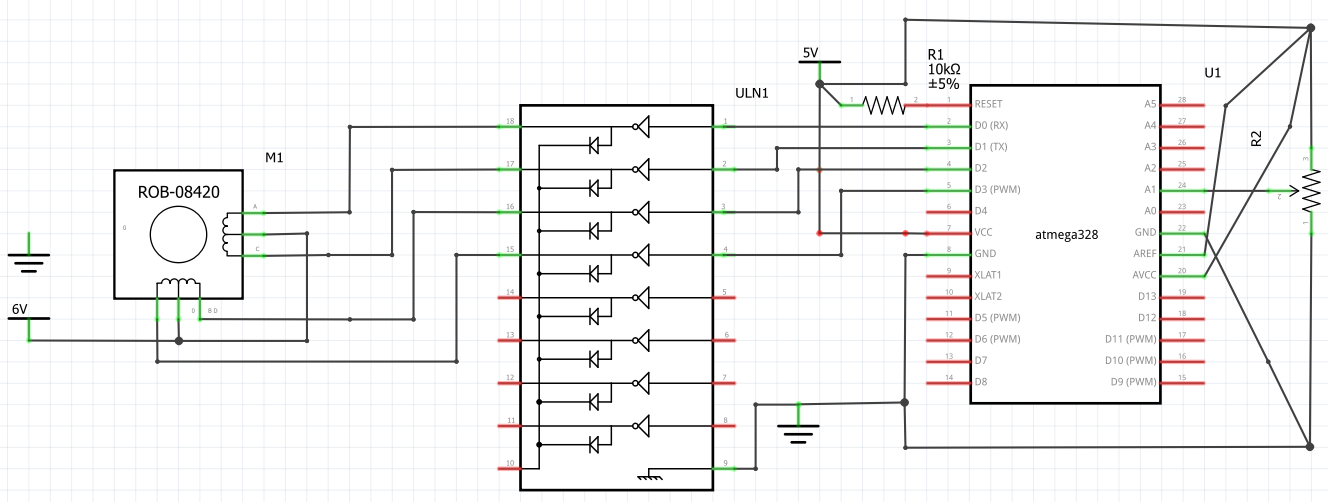
**}**

**}**

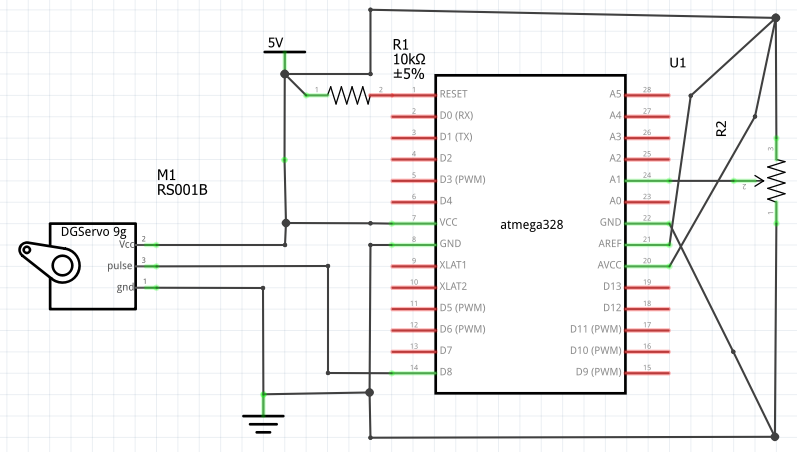
1. **SCHEMATICS**



Task 1 C Board Schematic



Task 2 C Board Schematic



Task 3 C Board Schematic

1. **VIDEO LINKS OF EACH DEMO**

Playlist - <https://www.youtube.com/playlist?list=PL5RuXbzEXweu1vdesRJxvBKYlDok19OfU>

Task 1/C Code - <https://youtu.be/KyEDR6F6uZ4>

Task 2/C Code - <https://youtu.be/cxfENl8muf0>

Task 3/C Code - <https://youtu.be/xWqpKz2-9J0>

1. **GITHUB LINK OF THIS DA**

[https://github.com/cisned2/DA](https://github.com/cisned2/DA3)4

**Student Academic Misconduct Policy**

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

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