Ashim Ghimire

CPE301 – SPRING 2014

Design Assignment 5

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

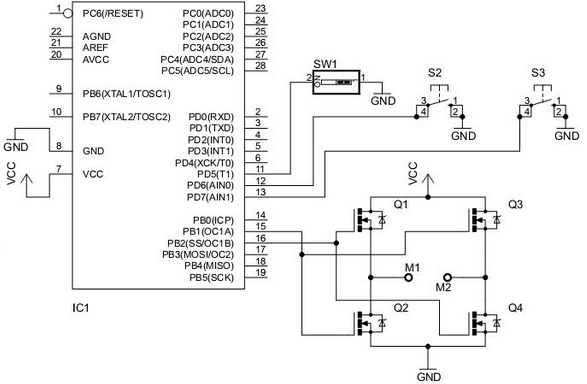
The student understands that all required components should be submitted in complete for grading of this assignment.

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **SUBMISSION ITEM** | **COMPLETED (Y/N)** | **MARKS**  **(/MAX)** |
| 0. | COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS |  |  |
| 1. | INITIAL CODE OF TASK 1/A |  |  |
| 2. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 2/B |  |  |
| 3. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 3/C |  |  |
| 4. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 4/D |  |  |
| 5. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 5/E |  |  |
| 6. | SCHEMATICS |  |  |
| 7. | SCREENSHOTS OF EACH TASK OUTPUT |  |  |
| 8. | SCREENSHOT OF EACH DEMO |  |  |
| 9. | VIDEO LINKS OF EACH DEMO |  |  |
| 10. | GOOGLECODE LINK OF THE DA |  |  |
|  |  |  |  |
|  |  |  |  |

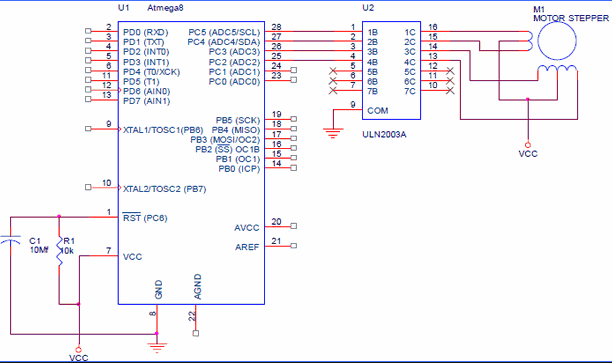
|  |  |  |  |
| --- | --- | --- | --- |
| 0. | COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS |  |  |

* Atmega 328P
* DC motor
* Stepper motor
* Servo motor
* TIP120
* ILD 74 Opto-Isolator
* ULN2003A
* Capacitor = 0.1 uF
* Diode, 1N4004
* Resistors = 330, 10k, 100k ohms
* Potentiometer

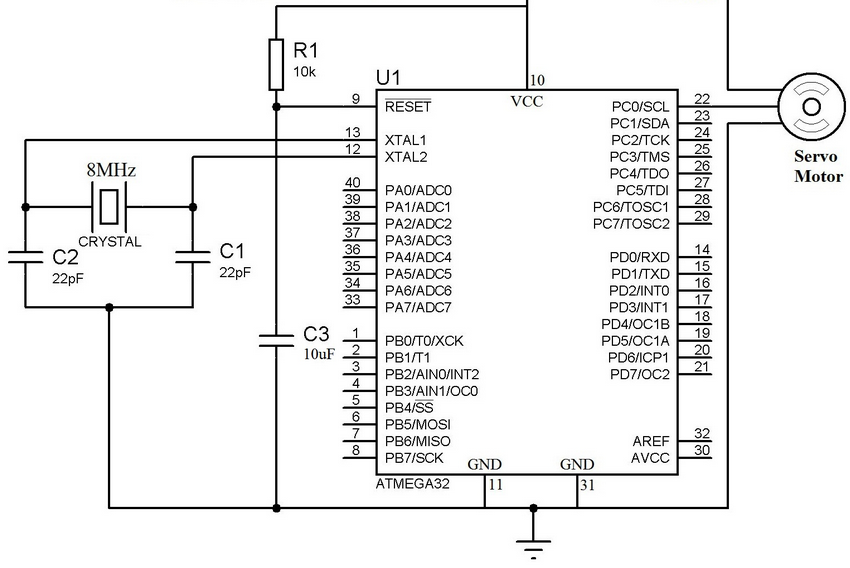
DC Motor:



Stepper Motor:



Servo Motor: 5V Motor Supply



|  |  |  |  |
| --- | --- | --- | --- |
| 1. | INITIAL CODE OF TASK 1/A: DC MOTOR |  |  |

/\*

\* DA5\_DC.c

\*

\* Created: 4/14/2015 11:43:41 AM

\* Author: ghimirea

DC Motor

\*/

#define F\_CPU 8000000UL //XTAL = 8MHZ

#include <avr/io.h>

#include <util/delay.h>

//#define ENABLE 0

void adc\_init()

{

ADCSRA = (1<<ADEN) | (1<<ADPS1); //enable ADC

//ADMUX = 0X00;

}

void adc\_read()

{

ADCSRA |= (1<<ADSC); //analog to digital

while ((ADCSRA&(1<<ADIF))== 0)

{ //wait for the conversion

}

}

void delay\_pot()

{

while ((TIFR1 & (1<<OCF1A)) == 0) //control delay

{

}

TIFR1 |= (1<<OCF1A);

}

int main()

{

DDRB = 0xFF; //make PORTB output pin

DDRC = 0x00; //make PORTC input

TCCR1B = 0x0D; //sets WGM12 = 1; CS12 = 1 and CS10 = 1

adc\_init(); //initialize PC.0 as input

while(1)

{

adc\_read(); //converts analog to digital

OCR1A = 950; //set a constant value for output compare register

delay\_pot(); //delay

if (ADC<90){

ADC = 90; //set min to 90

}

if (ADC>500){

ADC = 500; //set max to 500

}

OCR1A = ((ADC-80)\*3.8) + 280; //value from to potentiometer

PORTB ^= 0x01; //Toggle PB.0

delay\_pot(); //delay

PORTB ^= 0x01; //Toggle PB.0

}

return 0 ;

}

|  |  |  |  |
| --- | --- | --- | --- |
| 2. | TASK B – STEPPER MOTOR |  |  |

/\*

\* DA5\_STEPPER.c

\*

\* Created: 4/18/2015 1:36:05 PM

\* Author: ghimirea

\*/

#define F\_CPU 8000000UL //XTAL = 8MHZ

#include <avr/io.h>

#include <stdio.h>

#include <avr/interrupt.h>

#include <util/delay.h>

void adc\_init()

{

ADCSRA = (1<<ADEN) | (1<<ADPS1); //enable ADC

//ADMUX = 0X00;

}

void adc\_read()

{

ADCSRA |= (1<<ADSC); //analog to digital

while ((ADCSRA&(1<<ADIF))== 0)

{ //wait for the conversion

}

}

void delay\_pot()

{

while ((TIFR1 & (1<<OCF1A)) == 0) //control delay

{

}

TIFR1 |= (1<<OCF1A);

}

int main(void)

{

DDRB = 0xFF; //make PORTB output pin

DDRC = 0x00; //make PORTC input

TCCR1B = 0x0D; //sets WGM12 = 1; CS12 = 1 and CS10 = 1

adc\_init();

while(1)

{

adc\_read(); //converts analog to digital

if (ADC<90){

ADC = 90; //set min to 90

}

if (ADC>500){

ADC = 500; //set max to 500

}

OCR1A = ((ADC-80)\*3.8) + 280; //value from to potentiometer

PORTB = 0X66; //01100110 sequence

delay\_pot();

PORTB = 0X33; //00110011 sequence

delay\_pot();

PORTB = 0X99; //10011001 sequence

delay\_pot();

PORTB = 0XCC; //11001100 sequence

delay\_pot();

}

return 0;

}

|  |  |  |  |
| --- | --- | --- | --- |
| 3. | TASK C – SERVO MOTOR |  |  |

/\*

\* DA5\_SERVO.c

\*

\* Created: 4/20/2015 2:24:31 PM

\* Author: ghimirea

\*/

#define F\_CPU 8000000UL //XTAL = 8MHZ

#include <avr/io.h>

#include <stdio.h>

#include <avr/interrupt.h>

#include <util/delay.h>

void adc\_init()

{

ADCSRA = (1<<ADEN) | (1<<ADPS1); //enable ADC

ADMUX = 0X00;

}

void adc\_read()

{

ADCSRA |= (1<<ADSC); //analog to digital

while ((ADCSRA&(1<<ADIF))== 0)

{ //wait for the conversion

}

}

int main(void)

{

DDRB = 0xFF; //make PORTB output pin

DDRC = 0x00; //make PORTC input

//TOP = ICR1;

//output compare OC1A 8 bit non inverted PWM

//Clear OC1A on Compare Match, set OC1A at TOP

//Fast PWM

//ICR1 = 19999 defines 50Hz pwm

ICR1=19999;

TCCR1A|=(0<<COM1A0)|(1<<COM1A1)|(0<<COM1B0)|(0<<COM1B1)|(0<<FOC1A)|(0<<FOC1B)|(1<<WGM11)|(0<<WGM10);

TCCR1B|=(0<<ICNC1)|(0<<ICES1)|(1<<WGM13)|(1<<WGM12)|

(0<<CS12)|(1<<CS11)|(0<<CS10);

adc\_init(); //enables ADC

while(1)

{

adc\_read(); //convert analog to digital

if (ADC<90){

ADC = 620; //set min to 620

}

if (ADC>500){

ADC = 2100; //set max to 2100

}

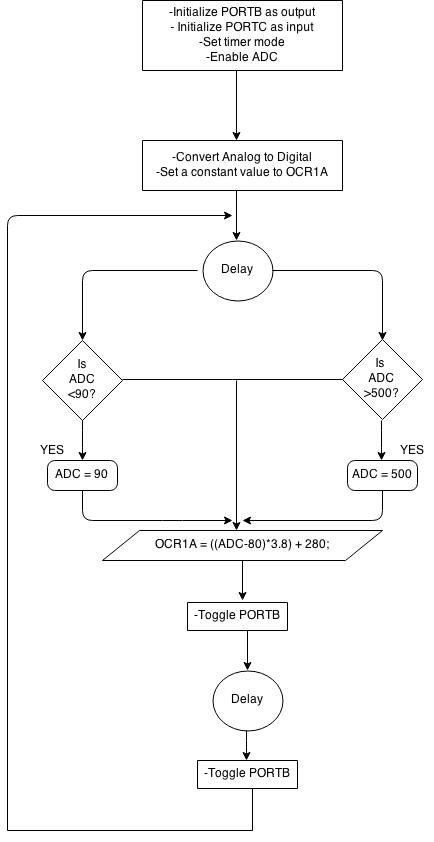
OCR1A = ((ADC-80)\*3.8) + 620; //value from to potentiometer

}

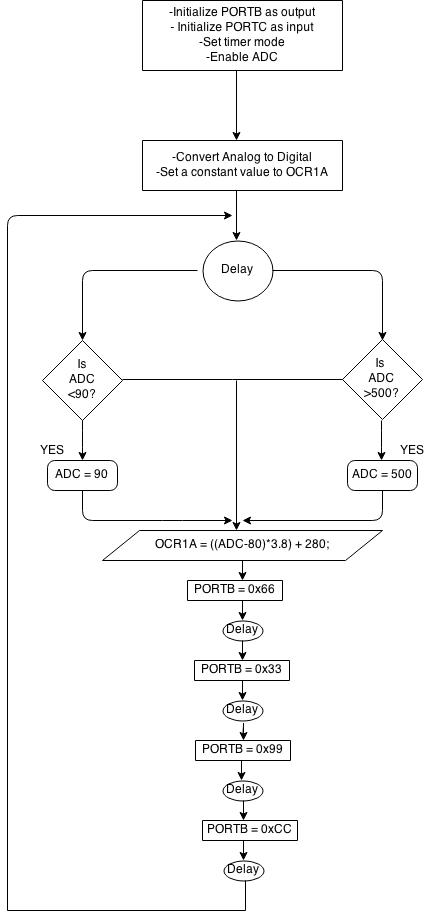
}

|  |  |  |  |
| --- | --- | --- | --- |
| 7. | FLOWCHART |  |  |

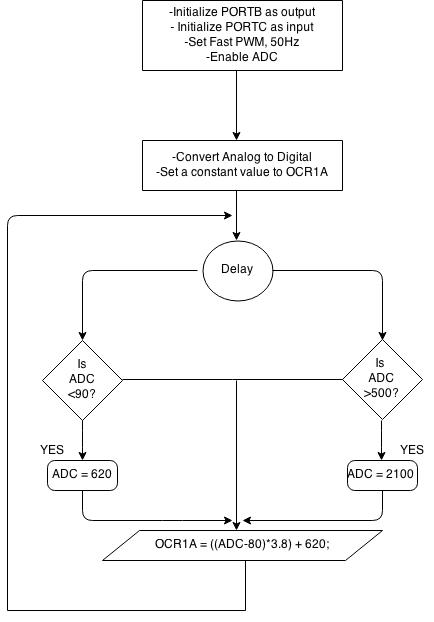
DC motor:



STEPPER MOTOR:

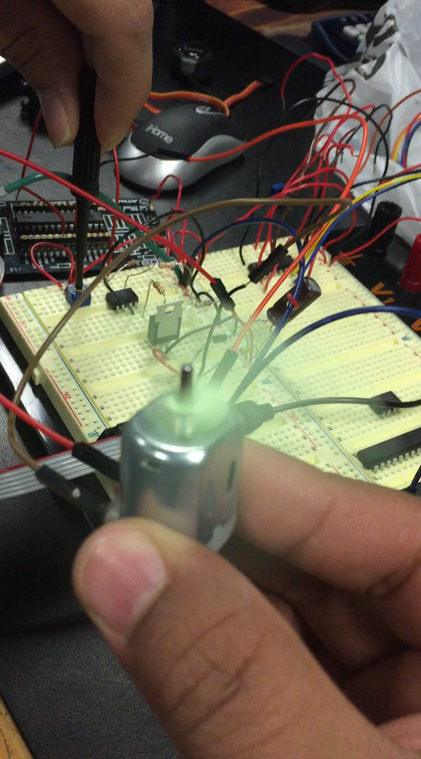


SERVO MOTOR:

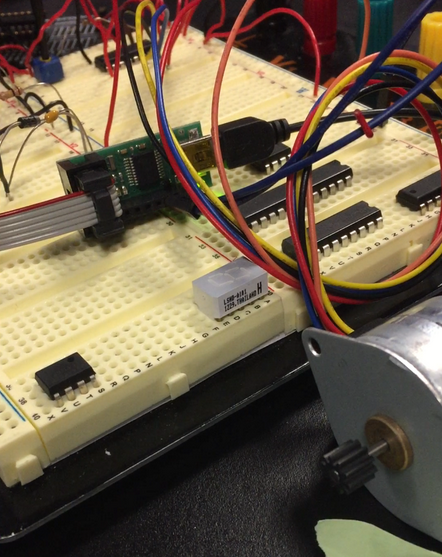


|  |  |  |  |
| --- | --- | --- | --- |
| 8. | SCREENSHOT OF EACH DEMO |  |  |

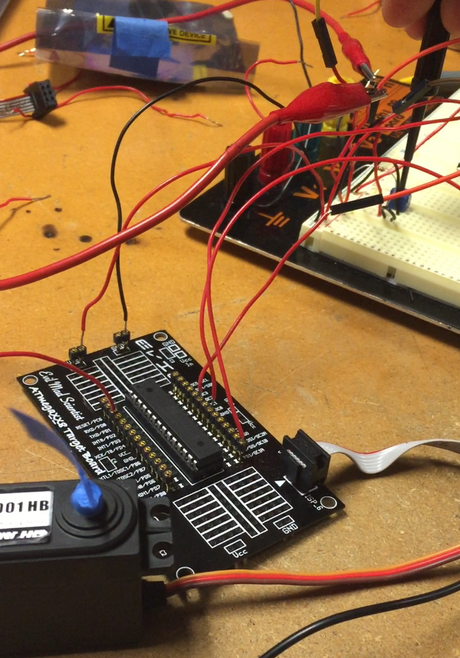
TASK 1/A: DC MOTOR



TASK B: STEPPER MOTOR



TASK C: SERVO MOTOR



|  |  |  |  |
| --- | --- | --- | --- |
| 9. | VIDEO LINKS OF EACH DEMO |  |  |
| DC : <https://www.dropbox.com/s/xhj3h97jvtnq89q/DC_video.mov?dl=0>  STEPPER: <https://www.dropbox.com/s/c51y8c67lwohilq/Stepper_video.mov?dl=0>  SERVO: <https://www.dropbox.com/s/rt3x2wu42ehdc6d/Servo_video.mov?dl=0> | | | |
| 10. | GITHUB LINK OF THE DA |  |  |
| <https://github.com/Ashim-Ghimire/301_GHIMIRE.git> | | | |

**Student Academic Misconduct Policy**

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

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

ASHIM GHIMIRE