Derek Nhan

CPE301 – SPRING 2016

Design Assignment 5

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

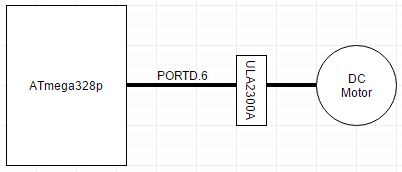
The student understands that all required components should be submitted in complete for grading of this assignment.

|  |  |  |  |
| --- | --- | --- | --- |
| **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. | INITIAL CODE OF TASK 2/B |  |  |
| 3. | INITIAL CODE OF TASK 3/C |  |  |
| 4. | SCHEMATICS |  |  |
| 5. | SCREENSHOTS OF EACH TASK OUTPUT |  |  |
| 6. | SCREENSHOT OF EACH DEMO |  |  |
| 7. | VIDEO LINKS OF EACH DEMO |  |  |
| 8. | GOOGLECODE LINK OF THE DA |  |  |
| 9. |  |  |  |
| 10. |  |  |  |
|  |  |  |  |
|  |  |  |  |

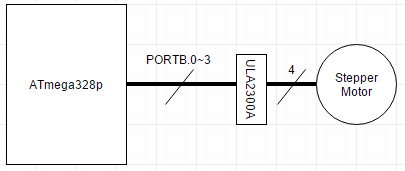
|  |  |  |  |
| --- | --- | --- | --- |
| 0. | COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS |  |  |

* Atmega328p
* Stepper motor
* DC motor
* Servo motor
* ULN2003A Transistor Array
* Breadboard

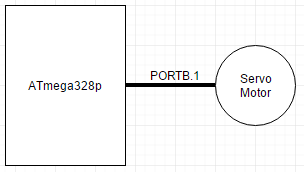
Task 1/A:



Task 2/B:



Task 3/C:



|  |  |  |  |
| --- | --- | --- | --- |
| 1. | INITIAL CODE OF TASK 1/A |  |  |

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

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

volatile unsigned int ADCvalue;

void init\_328(void)

//subroutine to initialize all counter/adc used in the application

{

ADMUX|=(1<<REFS0); //use Vcc as ref (0.1uF cap attached)

ADCSRA|=(1<<ADSC)|(1<<ADEN)|(1<<ADIE)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0)|(1<<ADATE);

//start conversion; enable adc; enable interrupts; select 128 prescaler;

//auto trigger enable; ADC0 used

ADCSRB=0;

//adc in free running mode

TCCR0A|=(1<<COM0A1)|(1<<WGM01)|(1<<WGM00); //use timer0 in fast PWM mode

//set OC0A on bottom and clear on up counting

TCCR0B|=(1<<CS00); //prescaler of 1 selected

}

ISR (ADC\_vect)

{

ADCvalue=ADC; //reads all 16-bits of ADCH:ADCL (0-1023 from pot)

ADCvalue=ADCvalue/4; //divide ADC value by 4 to get 0-225 value

OCR0A=ADCvalue; //set this value to the OCR0A register

*\_delay\_ms*(10); //delay to allow OCR0A to hold

}

int main()

{

DDRC = 0x00; //make PORTC input pins

DDRD = 0xFF; //make PORTB output pin

init\_328(); //initialize 328 (ADC/timers)

sei(); //enable global interrupts

while(1)

{

};

return 0;

}

|  |  |  |  |
| --- | --- | --- | --- |
| 2. | INITIAL CODE OF TASK 2/B |  |  |

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

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

volatile unsigned int ADCvalue;

void delay(unsigned int d)

//delay subroutine

{

unsigned n;

for(n=d; n>0; n--) //as long as n>0, decrease value

*\_delay\_ms*(1); //1 ms delay, thus n ms delay is used

}

void init\_328(void)

//subroutine to initialize all counter/adc used in the application

{

ADMUX|=(1<<REFS0); //use Vcc as ref (0.1uF cap attached)

ADCSRA|=(1<<ADSC)|(1<<ADEN)|(1<<ADIE)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0)|(1<<ADATE);

//start conversion; enable adc; enable interrupts; select 128 prescaler;

//auto trigger enable; ADC0 used

ADCSRB=0; //adc in free running mode

}

ISR (ADC\_vect)

{

ADCvalue=ADC; //reads all 16-bits of ADCH:ADCL (0-1023 from pot)

PORTB = 0x66; //output first step of stepper motor

delay(ADCvalue); //hold this step for the delay (the larger the slower the stepper)

PORTB = 0xCC; //output second step

delay(ADCvalue); //hold for same value

PORTB = 0x99; //output third step

delay(ADCvalue); //hold for same value

PORTB = 0x33; //output fourth step

delay(ADCvalue); //stay for same value

}

int main()

{

DDRC = 0x00; //make PORTC input pins

DDRB = 0xFF; //make PORTB output pin

init\_328(); //initialize adc

sei(); //enable global interrupts

while(1)

{

};

return 0;

}

|  |  |  |  |
| --- | --- | --- | --- |
| 3. | INITIAL CODE OF TASK 3/C |  |  |

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

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

volatile unsigned int ADCvalue;

void init\_328(void)

//subroutine to initialize all counter/adc used in the application

{

ADMUX|=(1<<REFS0); //use Vcc as ref (0.1uF cap attached)

ADCSRA|=(1<<ADSC)|(1<<ADEN)|(1<<ADIE)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0)|(1<<ADATE);

//start conversion; enable adc; enable interrupts; select 128 prescaler;

//auto trigger enable; ADC0 used

ADCSRB=0;

//adc in free running mode

TCCR1A|=(1<<COM1A1)|(1<<WGM11); //timer1 in fast pwm mode, OCR1A clear on up, set down

TCCR1B|=(1<<WGM12)|(1<<WGM13)|(1<<CS11); //prescaler of 1, fast pwm, ICR1 as top

ICR1=20000; //set to 20000 (50Mhz, T=20m)

}

ISR (ADC\_vect)

//ADC interrupt subroutine

{

ADCvalue=ADC; //reads all 16-bits of ADCH:ADCL (0-1023 from pot)

ADCvalue=(3\*ADCvalue)+1150; //multiply value by 3 and add to 1150 to obtain servo duty

OCR1A=ADCvalue; //set OCR1A as new value to create duty cycle

}

int main()

{

DDRC = 0x00; //make PORTC input pins

DDRB = 0xFF; //make PORTB output

init\_328(); //initizlize timer1 and adc

sei(); //set global interrupts

while(1)

{

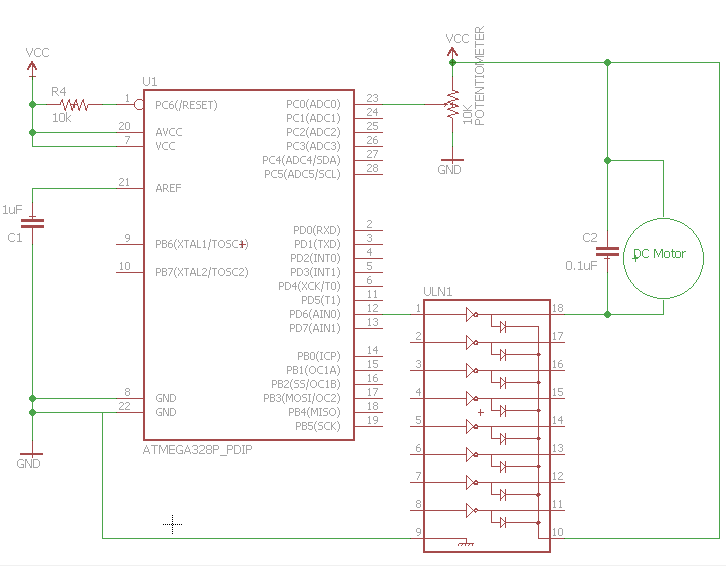
};

return 0;

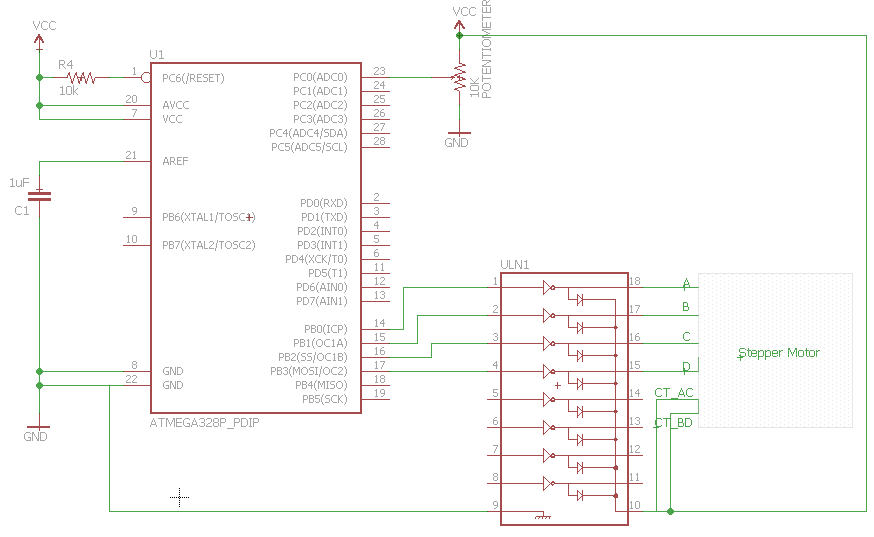
}

|  |  |  |  |
| --- | --- | --- | --- |
| 4. | SCHEMATICS |  |  |

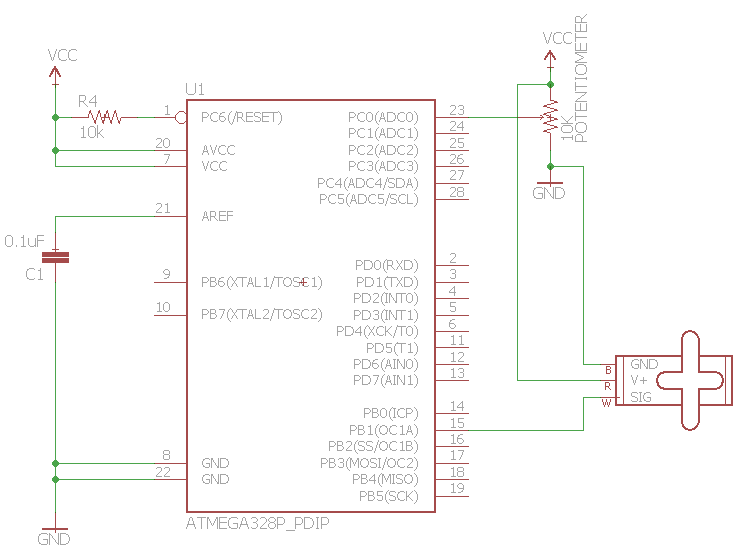
TASK 1/A:



TASK 2/B:



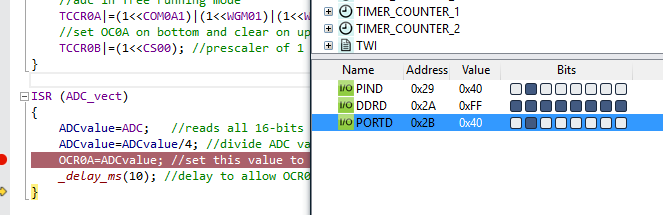
TASK 3/C:



|  |  |  |  |
| --- | --- | --- | --- |
| 5. | SCREENSHOTS OF EACH TASK OUTPUT |  |  |

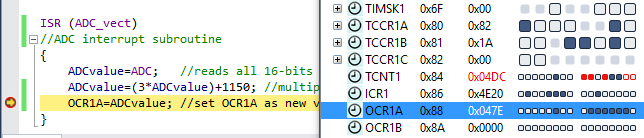
TASK 1/A:

Verity OCR0A is sending an output signal to PORTD.6



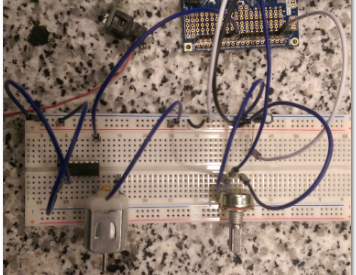
TASK 3/C:

Verify OCR1A is updating with change in ADC value

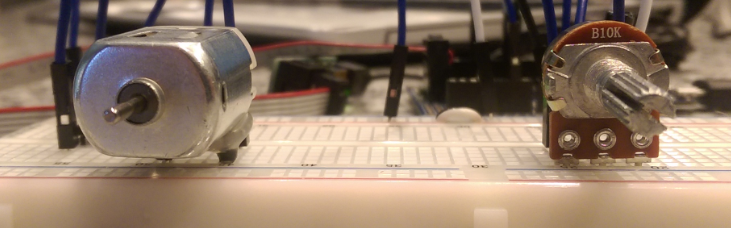


|  |  |  |  |
| --- | --- | --- | --- |
| 6. | SCREENSHOT OF EACH DEMO |  |  |

TASK 1/A:

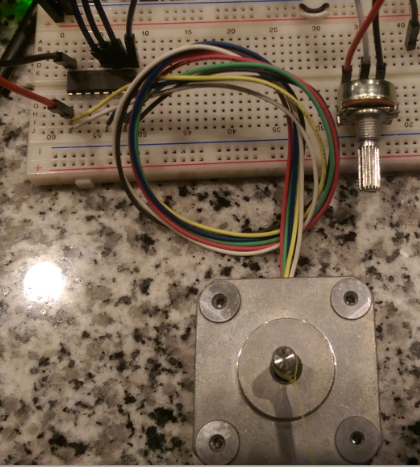
DC motor connected to ULN2003A and the potentiometer all wired up to the ATmega328p xplained board.  


Front view of DC motor and potentiometer.

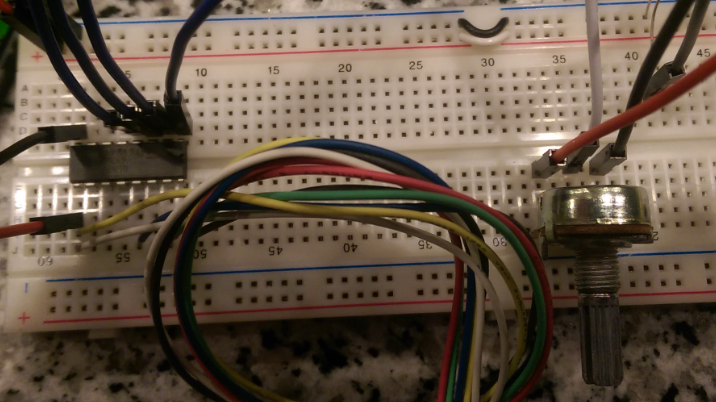


TASK 2/B:

Stepper motor and potentiometer wired up to control stepper speed (see video).

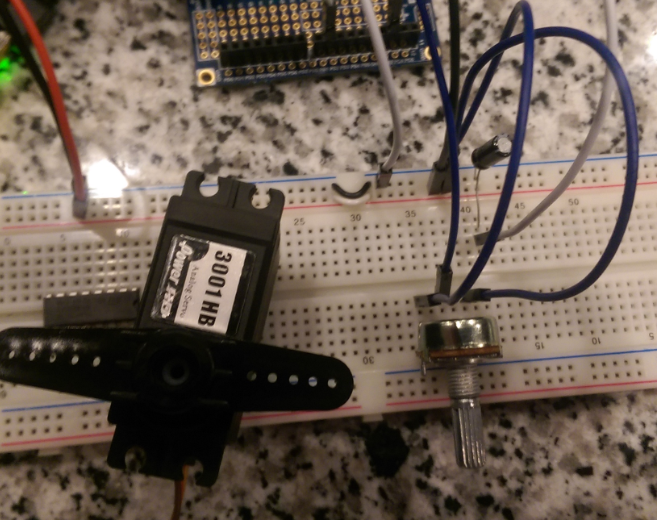


Close up of stepper motor wired up to the transistor array.



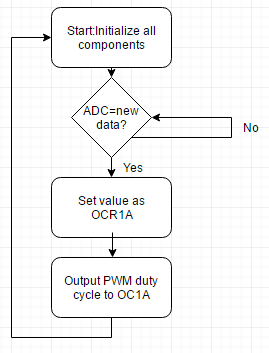
TASK 3/C:

Servo motor attached to Vcc, GND, PORTB.1 of the ATmega328p. Potentiometer attached to PC.0 for analog input. (see video for demo)

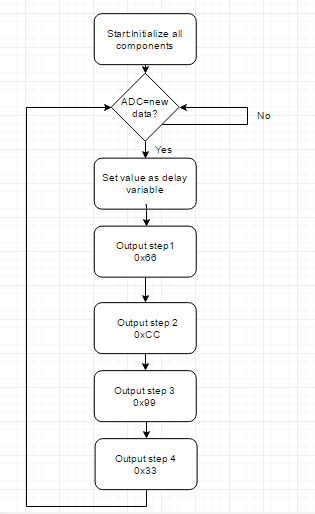


|  |  |  |  |
| --- | --- | --- | --- |
| 7. | FLOW CHART |  |  |

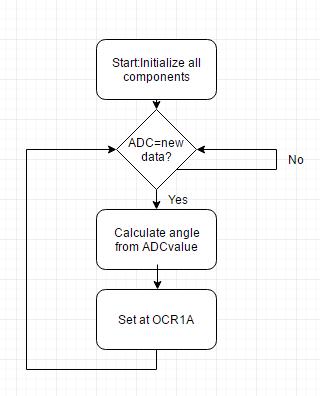
TASK 1



TASK 2



TASK 3



|  |  |  |  |
| --- | --- | --- | --- |
| 7. | VIDEO LINKS OF EACH DEMO |  |  |
| TASK1 - <https://www.youtube.com/watch?v=Z10W6opUYFA>  TASK2 - <https://www.youtube.com/watch?v=nAnN8p-ctWA>  TASK 3 - https://www.youtube.com/watch?v=l4EAQfZGHIs | | | |
| 8. | GOOGLECODE LINK OF THE DA |  |  |
| https://github.com/nhand2/CPE301S16 | | | |

**Student Academic Misconduct Policy**

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

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

Derek Nhan