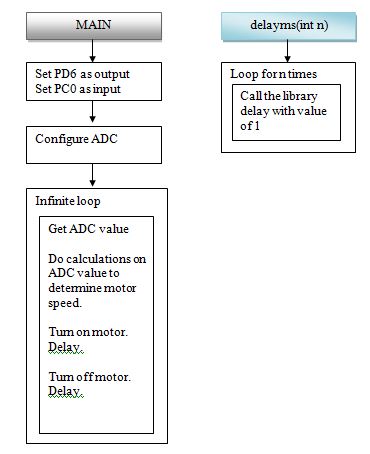
CPE 301 – SPRING 2015

DESIGN ASSIGNMENT 5

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **SUBMISSION ITEM** | **COMPLETED (Y/N)** | **MARKS**  **(/MAX)** |
| 0. | Flowchart of code - DC motor | Y |  |
| 1. | AVR C Code that compiles and works – DC motor | Y |  |
| 2. | Flowchart of code - Stepper motor | Y |  |
| 3. | AVR C Code that compiles and works – Stepper motor | Y |  |
| 4. | Flowchart of code – Servo motor | Y |  |
| 5. | AVR C Code that compiles and works – Servo motor | Y |  |
| 6. | Schematics: DC motor | Y |  |
| 7. | Schematics: Stepper motor | Y |  |
| 8. | Schematics: Servo motor | Y |  |
| 9. | Snapshot of board with connected components: DC motor | Y |  |
| 10. | Snapshot of board with connected components: Stepper motor | Y |  |
| 11. | Snapshot of board with connected components: Servo motor | Y |  |
| 12. | Links to YouTube Videos | Y |  |

**0 – Flowchart of code: DC Motor**

****

**1 - AVR C Code that compiles and works: DC Motor**

/\*

Emmanuel Sanchez

CPE 301 - DA5: DC motor control using potentiometer

\*/

#define F\_CPU 16000000UL //clock = 16 MHz

#include <avr/io.h>

#include <util/delay.h> //needed for delays

void delayms(int); //delay function. Uses milliseconds

int main(void)

{

DDRD |= (1<<PD6); //OC0A output PWM

DDRC &= ~(1<<PC0); //potentiometer analog input

//configure ADC

ADCSRA = 0x87; //ADC enable, no interrupts, div factor 128

ADMUX = 0x60; //Reference voltage = AVCC with external capacitor at AREF pin, left justified

uint8\_t pot; //holds value of pot from analog channel

float motor\_speed; //used to determine delays

while(1)

{

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

while((ADCSRA &(1<<ADIF)) == 0); //wait for conversion to finish

pot = ADCH; //take value from upper byte of ADC

motor\_speed = pot \* 5.0 / 1024.0; //scale the potentiometer value

PORTD |= (1<<PD6); //turn on motor

delayms(motor\_speed \* 30); //delay is determined by pot

PORTD &= ~(1<<PD6); //turn off motor

delayms((1 - motor\_speed) \* 30); //time off = (1 - time on)

}

}

void delayms(int n) //delay function

{

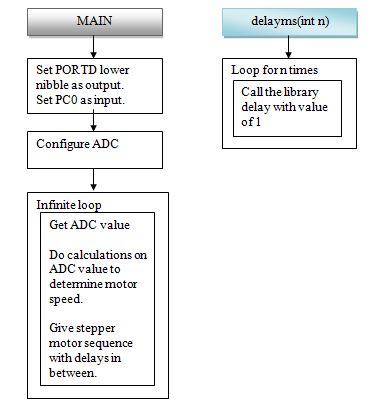
int i;

for(i=0; i<n; i++) //loop for n milliseconds

\_delay\_ms(1);

};

**2 – Flowchart of code: Stepper Motor**

****

**3 - AVR C Code that compiles and works: DC Motor**

/\*

Emmanuel Sanchez

CPE 301 - DA5: stepper motor speed control using potentiometer

\*/

#define F\_CPU 16000000UL //XTAL = 16MHZ

#include <avr/io.h>

#include <util/delay.h>

void delayms(int); //delay function

int main()

{

DDRD = 0x0F; //set lower nibble of PORTD as output

DDRC &= ~(1<<PC0); //potentiometer analog input

//configure ADC

ADCSRA = 0x87; //ADC enable, no interrupts, div factor 128

ADMUX = 0x60; //Reference voltage = AVCC with external capacitor at AREF pin, left justified

uint8\_t pot; //holds value of pot from analog channel

float motor\_speed; //used to determine delays

while(1)

{

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

while((ADCSRA &(1<<ADIF)) == 0); //wait for conversion to finish

pot = ADCH; //take value from upper byte of ADC

motor\_speed = pot \* 5.0 / 1024.0; //scale the potentiometer value

//sequence for stepper motor

PORTD = 0X66;

delayms(motor\_speed \* 200); //delay is proportional to pot value

PORTD = 0x33;

delayms(motor\_speed \* 200); //delay is proportional to pot value

PORTD = 0x99;

delayms(motor\_speed \* 200); //delay is proportional to pot value

PORTD = 0xCC;

delayms(motor\_speed \* 200); //delay is proportional to pot value

}

return 0;

}

void delayms(int n) //delay function

{

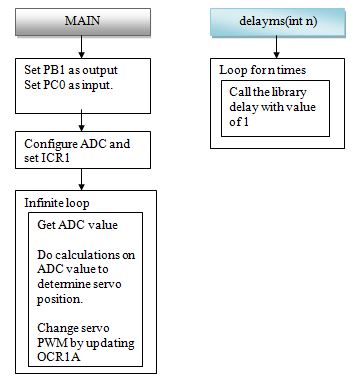
int i;

for(i=0; i<n; i++) //loop for n milliseconds

\_delay\_ms(1);

}

**4 – Flowchart of code: Servo Motor**

****

**5 - AVR C Code that compiles and works: Servo Motor**

/\*

Emmanuel Sanchez

CPE 301 - DA5: servo motor control using potentiometer

\*/

#define F\_CPU 16000000UL //clock = 16 MHz

#include <avr/io.h>

int main()

{

DDRB |= (1<<PB1); //PWM pin (OC1A)

DDRC &= ~(1<<PC0); //potentiometer analog input

//configure timer1

TCCR1A |= (1<<COM1A1) | (1<<WGM11); //fast PWM (mode 14), non-inverting

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

ICR1 = 4999; //fPWM = 50 Hz (period = 20ms)

//configure ADC

ADCSRA = 0x87; //ADC enable, no interrupts, div factor 128

ADMUX = 0x60; //Reference voltage = AVCC with external capacitor at AREF pin, left justified

uint8\_t pot; //holds value of pot from analog channel

float servo\_pos; //holds value to determine servo position

while(1)

{

//get ADC value

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

while((ADCSRA &(1<<ADIF)) == 0); //wait for conversion to finish

pot = ADCH<<1; //take value from upper byte of ADC

servo\_pos = pot \* 5 / 1024.0; //scale the potentiometer value

//determine servo position

//0 degrees on my servo is when OCR1A is 150.

//180 degrees is when OCR1A is 535.

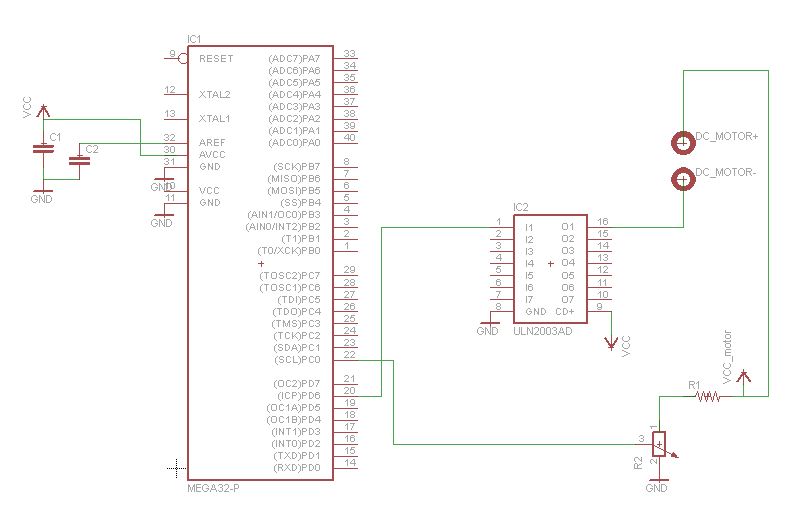
servo\_pos = (servo\_pos \* 385) + 150; //equation to determine position. Min = 150, Max = 535

OCR1A = (int)servo\_pos; //update servo position

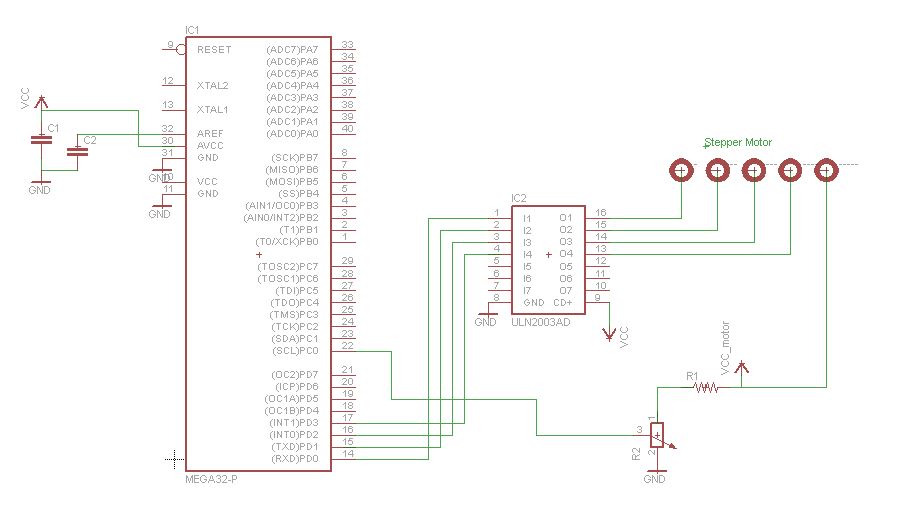
}

}

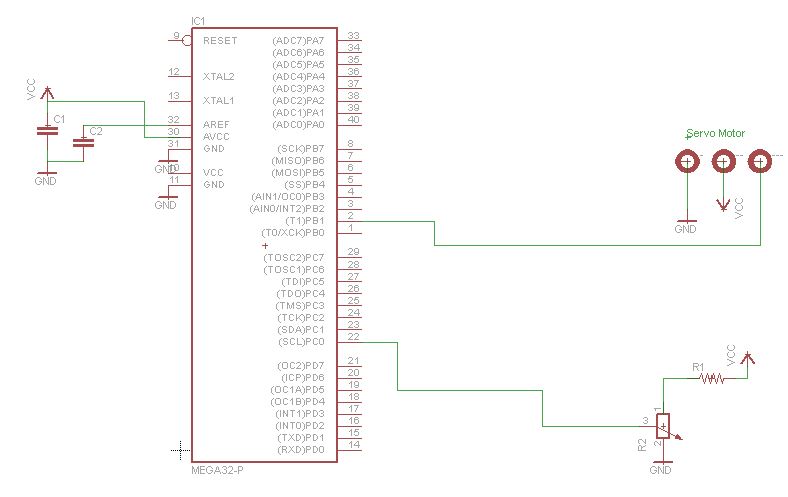
**6 – Schematics: DC Motor**

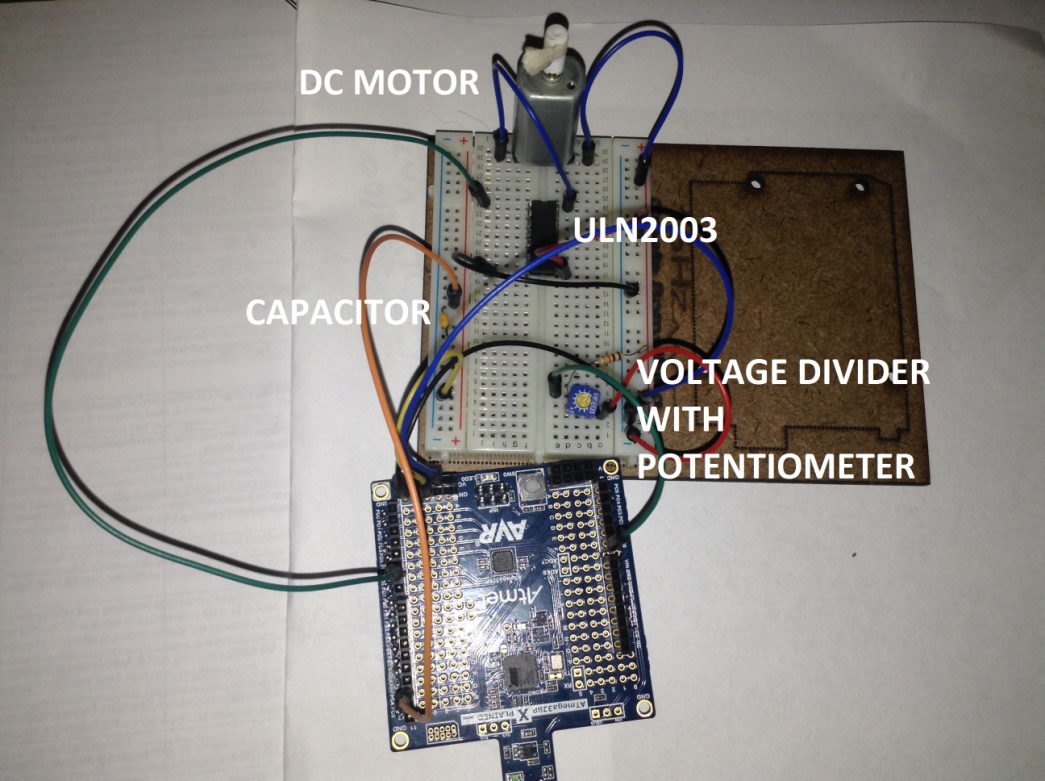
****

**7 – Schematics: Stepper Motor**

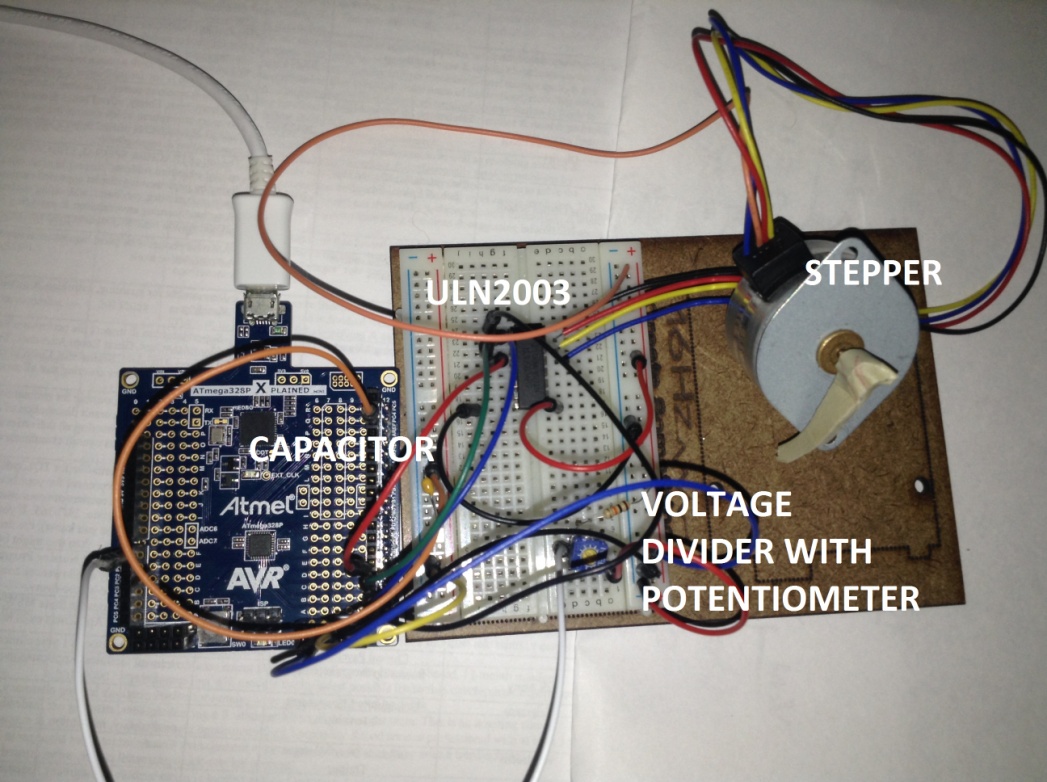
****

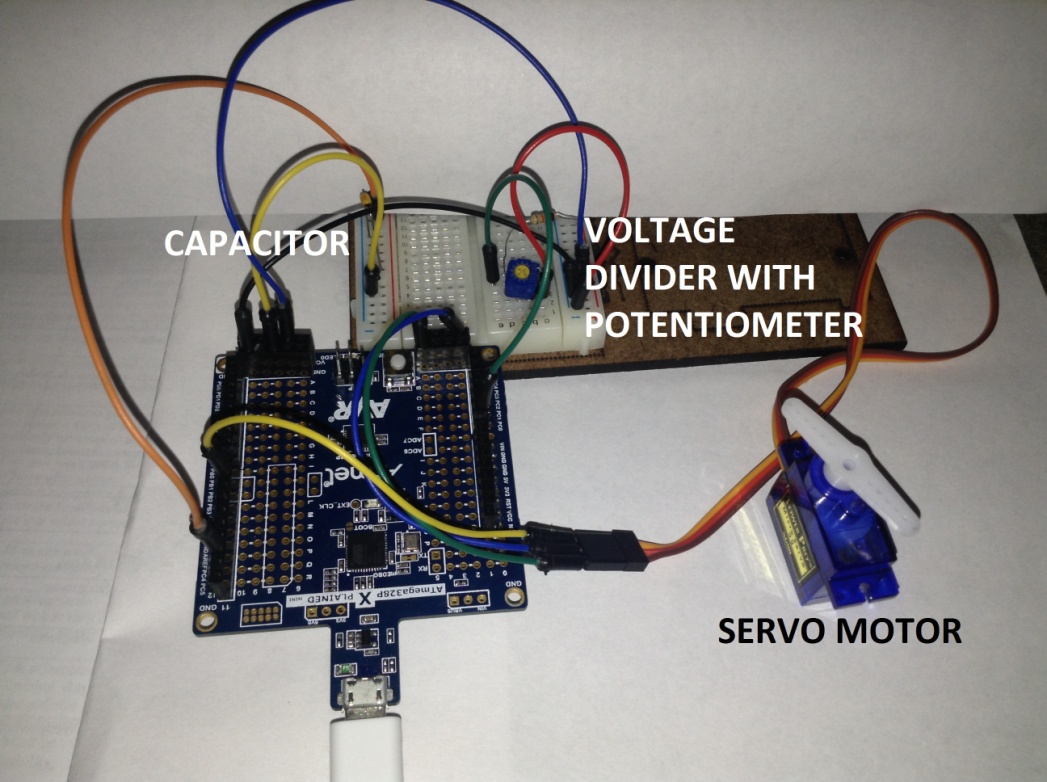
**8- Schematics: Servo Motor**

****

**9 – Snapshot of board with connected components: DC Motor**

**10 – Snapshot of board with connected components: Stepper Motor**

****

**11 – Snapshot of board with connected components: Servo Motor**

**12 – Links to YouTube Videos:**

**DC Motor:**

<https://www.youtube.com/watch?v=Yf1n8og6tKA&feature=youtu.be>

**Stepper Motor:**

<https://www.youtube.com/watch?v=CAcE81VF8iE&feature=youtu.be>

**Servo Motor:**

<https://www.youtube.com/watch?v=ZOKYE8QVthA&feature=youtu.be>

**Student Academic Misconduct Policy**

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

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

Emmanuel Sanchez