RALPH MAGO

CPE301 – SPRING 2016

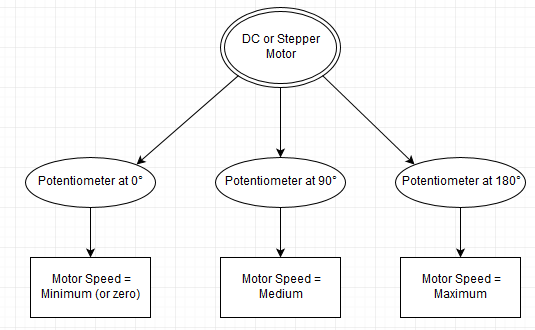
Design Assignment 5

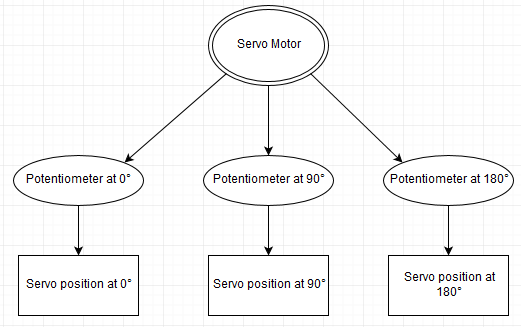
**DO NOT REMOVE THIS PAGE DURING SUBMISSION:**

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. | FLOWCHART OF THE CODE | Y |  |
| 1. | COMPONENTS LIST | Y |  |
| 2. | INITIAL CODE OF TASK 1/A | Y |  |
| 3. | INCREMENTAL CODE OF TASK 2/B | Y |  |
| 4. | INCREMENTAL CODE OF TASK 3/C | Y |  |
| 5. | SCHEMATICS | Y |  |
| 6. | SCREENSHOTS OF EACH TASK OUTPUT | Y |  |
| 7. | SCREENSHOT OF EACH DEMO | Y |  |
| 8. | VIDEO LINKS OF EACH DEMO | Y |  |
| 9. | GITHUB LINK OF THE DA | Y |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 0. | FLOWCHART OF THE CODE |  |  |





|  |  |  |  |
| --- | --- | --- | --- |
| 1. | COMPONENTS LIST |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atmel Xplained Mini (328P) | | 1 | | |
| Potentiometer – 5KΩ | | 1 | | |
| TI ULN2003 Motor Driver | | 1 | | |
| Resistor – 1KΩ | | 1 | | |
| 3-wire HITEC Servo Motor | | 1 | | |
| 5-wire Unipolar Stepper Motor | | 1 | | |
| 2-wire DC Motor | | 1 | | |
| 2. | INITIAL CODE OF TASK 1/A | |  |  |

/\*

\* DA5T1.c

\*

\* Created: 4/20/2016 1:48:54 AM

\* Author : r

\*/

//TASK 1: Control the speed of the DC Motor using a potentiometer

//connected to any analog port.

#define *F\_CPU* 16000000UL //clock frequency

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

#include <avr/io.h>

void motor\_delay\_ms(int); //delay function in ms

void motor\_delay\_ms(int n) //delay function

{

int i;

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

*\_delay\_ms*(1);

};

int main()

{

*uint8\_t* trimpot\_val; //8-bit int type for potentiometer values

float motorSpeed; //holds value to determine speed of DC motor

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

DDRC |= (0<<PC0); //potentiometer analog input

//configure ADC

ADCSRA |= (1<<ADEN) | (1<<ADPS2) | (1<<ADPS1) | (1<<ADPS0); //ADCSRA=0b10000111 => division factor 128

ADMUX |= (0<<REFS1) | (1<<REFS0) | (1<<ADLAR); //VRef = AVCC with external capacitor at AREF pin.

// ADLAR high, ADCH = 0b00000000 and ADCL = 0b00xxxxxx

// https://sites.google.com/site/qeewiki/books/avr-guide/analog-input

while(1)

{

ADCSRA |= (1<<ADSC); //ADC conversion of potentiometer value to servo position

while((ADCSRA &(1<<ADIF)) == 0); //loop until ADC done

trimpot\_val = ADCH << 1; //potentiometer value = ADC

motorSpeed = trimpot\_val \* 5.0 / 1024.0; //scale the potentiometer value

//5 Volts divided by 2^10 (10 bit ADC)

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

motor\_delay\_ms(motorSpeed \* 30); //delay determined by potentiometer

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

motor\_delay\_ms((1 - motorSpeed) \* 30); //stop motor (1 minus speed of motor)

}

}

|  |  |  |  |
| --- | --- | --- | --- |
| 3. | INCREMENTAL CODE OF TASK 2/B |  |  |

/\*

\* DA5T2.c

\*

\* Created: 4/20/2016 12:23:43 AM

\* Author : r

\*/

//TASK 2: Control the speed of the Stepper Motor using a potentiometer

//connected to any analog port.

#define *F\_CPU* 16000000UL //clock frequency

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

#include <avr/io.h>

void motor\_delay\_ms(int); //delay function in ms

void motor\_delay\_ms(int n) //delay function

{

int i;

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

*\_delay\_ms*(1);

};

int main()

{

*uint8\_t* trimpot\_val; //8-bit int type for potentiometer values

float motorSpeed; //holds value to determine speed of DC motor

DDRD |= (1<<PD0) | (1<<PD1) | (1<<PD2)| (1<<PD3); //(OC0A)

DDRC |= (0<<PC0); //potentiometer analog input

//configure ADC

ADCSRA |= (1<<ADEN) | (1<<ADPS2) | (1<<ADPS1) | (1<<ADPS0); //ADCSRA=0b10000111 => division factor 128

ADMUX |= (0<<REFS1) | (1<<REFS0) | (1<<ADLAR); //VRef = AVCC with external capacitor at AREF pin.

// ADLAR high, ADCH = 0b00000000 and ADCL = 0b00xxxxxx

// https://sites.google.com/site/qeewiki/books/avr-guide/analog-input

while(1)

{

ADCSRA |= (1<<ADSC); //ADC conversion of potentiometer value to servo position

while((ADCSRA &(1<<ADIF)) == 0); //loop until ADC done

trimpot\_val = ADCH << 1; //potentiometer value = ADC

motorSpeed = trimpot\_val \* 5.0 / 1024.0; //scale the potentiometer value

//5 Volts divided by 2^10 (10 bit ADC)

PORTD = 0b0110; //0x06, PD1 and PD2 HIGH

motor\_delay\_ms(motorSpeed \* 200); //stepping speed times 200 ms delay

PORTD = 0b0011; //0x03, PD0 and PD1 HIGH

motor\_delay\_ms(motorSpeed \* 200); //stepping speed times 200 ms delay

PORTD = 0b1001; //0x09, PD0 and PD3 HIGH

motor\_delay\_ms(motorSpeed \* 200); //stepping speed times 200 ms delay

PORTD = 0b1100; //0x0c, PD3 and PD4 HIGH

motor\_delay\_ms(motorSpeed \* 200); //stepping speed times 200 ms delay

}

return 0;

}

|  |  |  |  |
| --- | --- | --- | --- |
| 4. | INCREMENTAL CODE OF TASK 3/C |  |  |

/\*

\* DA5T3.c

\*

\* Created: 4/19/2016 9:17:46 PM

\* Author : r

\*/

//TASK 3: Control the position of the Servo motor using a potentiometer

//connected to any analog-in. Pot value = 0, Servo at 0 degrees.

//Pot value = MAX (5V), Servo at 180 degrees.

#define *F\_CPU* 16000000UL //clock frequency

#define preScaler 64 //prescaler value

#include <avr/io.h>

int main()

{

*uint8\_t* trimpot\_val; //8-bit int type for potentiometer values

float servoPosition; //holds value to determine servo position

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

DDRC |= (0<<PC0); //potentiometer analog input

//configure timer1

TCCR1A |= (1<<COM1A1); //COM1A1=1 and COM1A0=0 (Non-inverted mode) (HIGH at bottom, LOW on match)

TCCR1A |= (1<<WGM11) | (0<<WGM10); //WGM13=1, WGM12=1, WGM11=1, WGM10=0

TCCR1B |= (1<<WGM13) | (1<<WGM12); //1 1 1 0 => fast PWM using ICR1 (mode 14)

//FAST PWM is more suited for servo

TCCR1B |= (0<<CS12) | (1<<CS11) | (1<<CS10); //0 1 1 => prescaler 64

//we want T1 to repeat with a period of 20ms or 50Hz.

ICR1 = ((double)*F\_CPU*) / (preScaler \* 50); //value of ICR1 = [(F\_CPU)/(prescaler \* 50Hz)]

//configure ADC

ADCSRA |= (1<<ADEN) | (1<<ADPS2) | (1<<ADPS1) | (1<<ADPS0); //ADCSRA=0b10000111 => division factor 128

ADMUX |= (0<<REFS1) | (1<<REFS0) | (1<<ADLAR); //VRef = AVCC with external capacitor at AREF pin.

// ADLAR high, ADCH = 0b00000000 and ADCL = 0b00xxxxxx

// https://sites.google.com/site/qeewiki/books/avr-guide/analog-input

while(1)

{

ADCSRA |= (1<<ADSC); //ADC conversion of potentiometer value to servo position

while((ADCSRA &(1<<ADIF)) == 0); //loop until ADC done

trimpot\_val = ADCH << 1; //store high byte of ADC

servoPosition = trimpot\_val \* 5 / 1024.0; //scale the potentiometer value

//5 Volts divided by 2^10 (10 bit ADC)

//OCR1A = 150 => 0 degrees (min) 150 ohms

//OCR1A = 535 => 180 degrees (max) 535 ohms

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

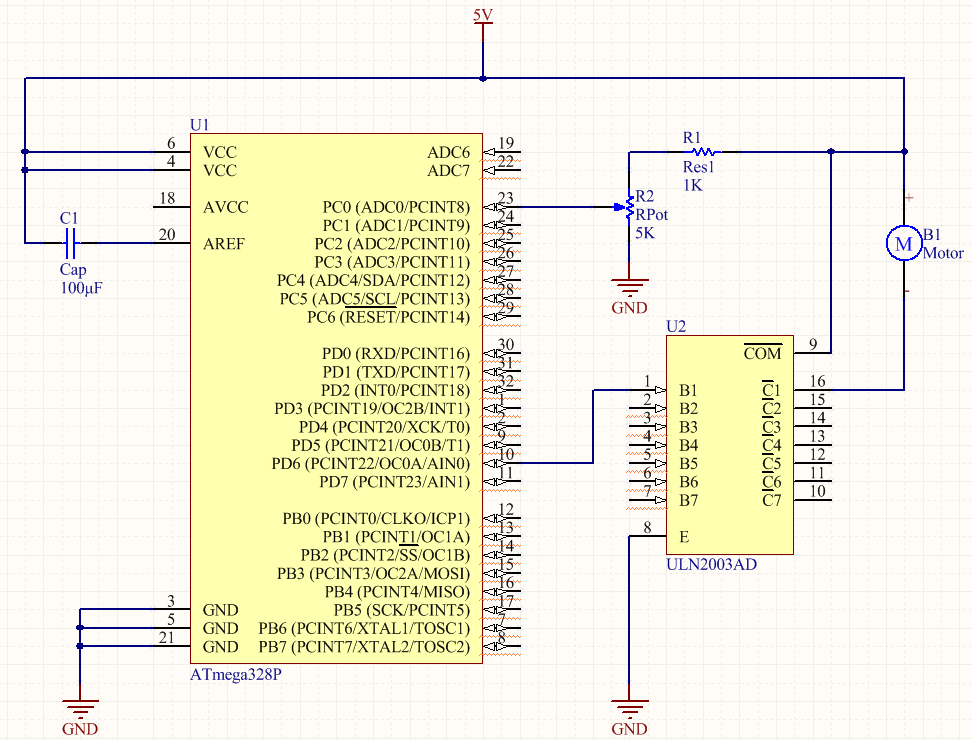
OCR1A = (int)servoPosition; //update servo position

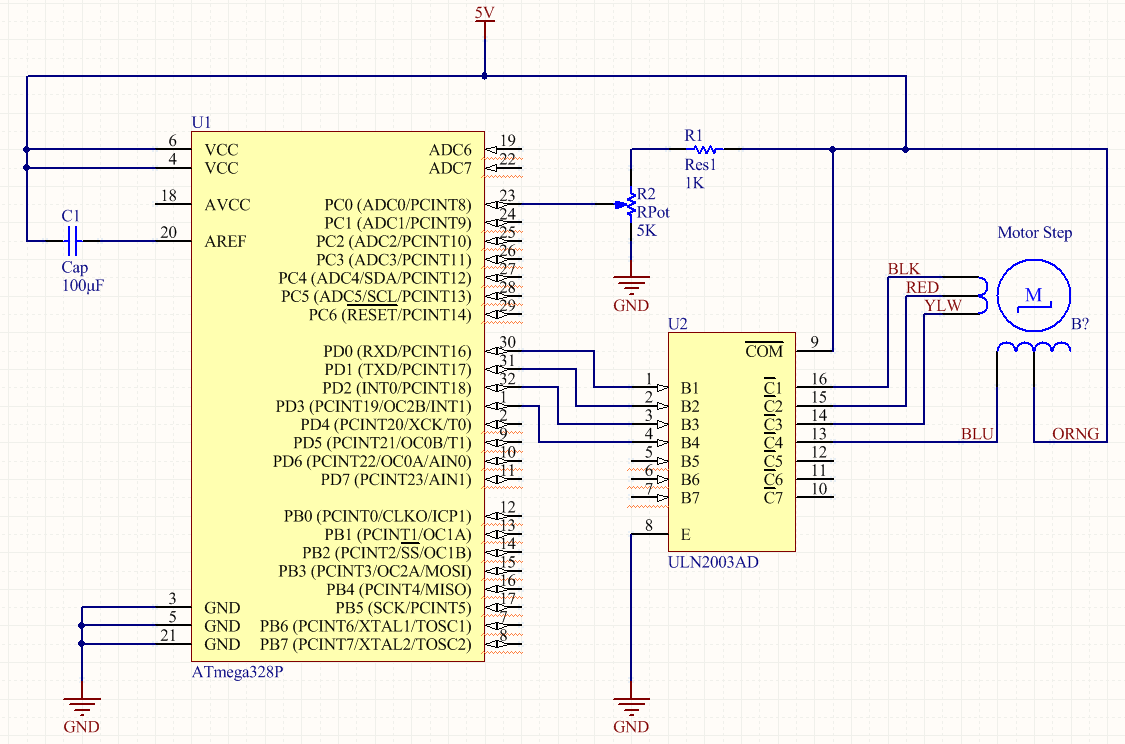
}

}

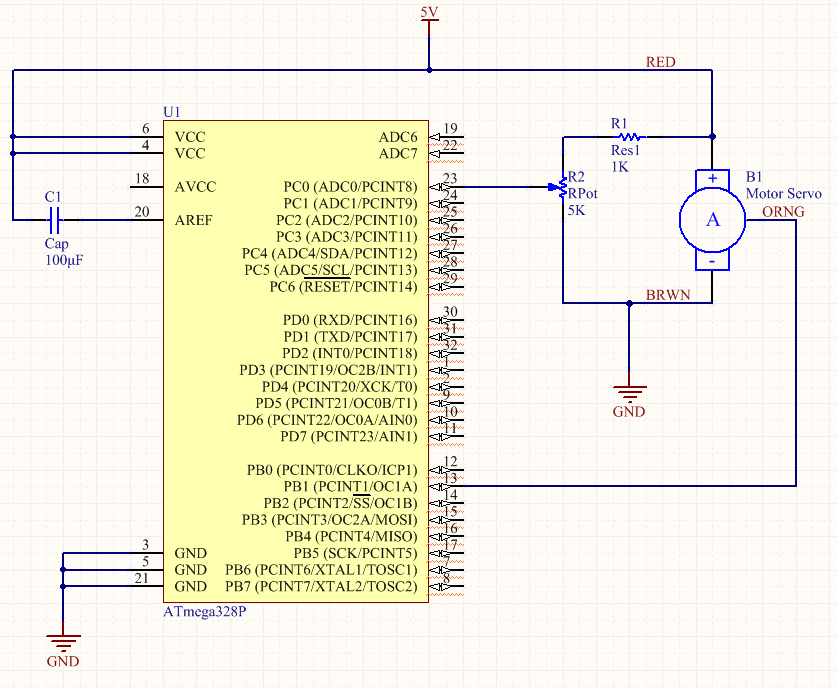
|  |  |  |  |
| --- | --- | --- | --- |
| 5. | SCHEMATIC |  |  |

DC Motor:

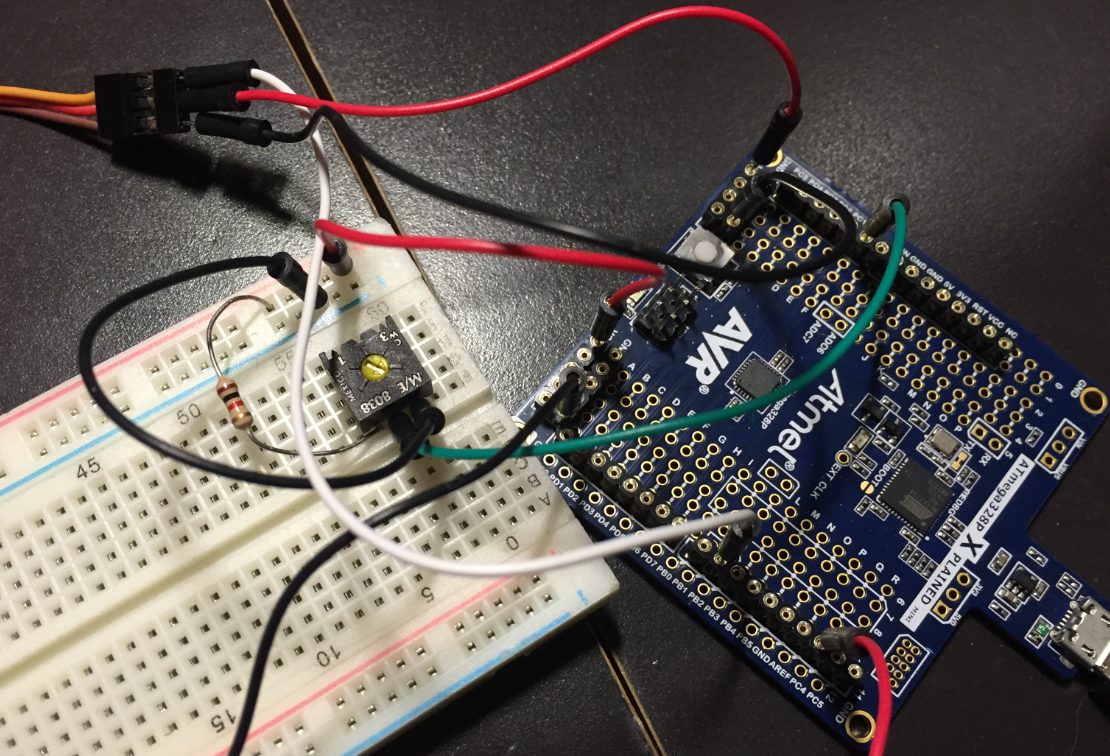


Stepper Motor: 

Servo Motor:

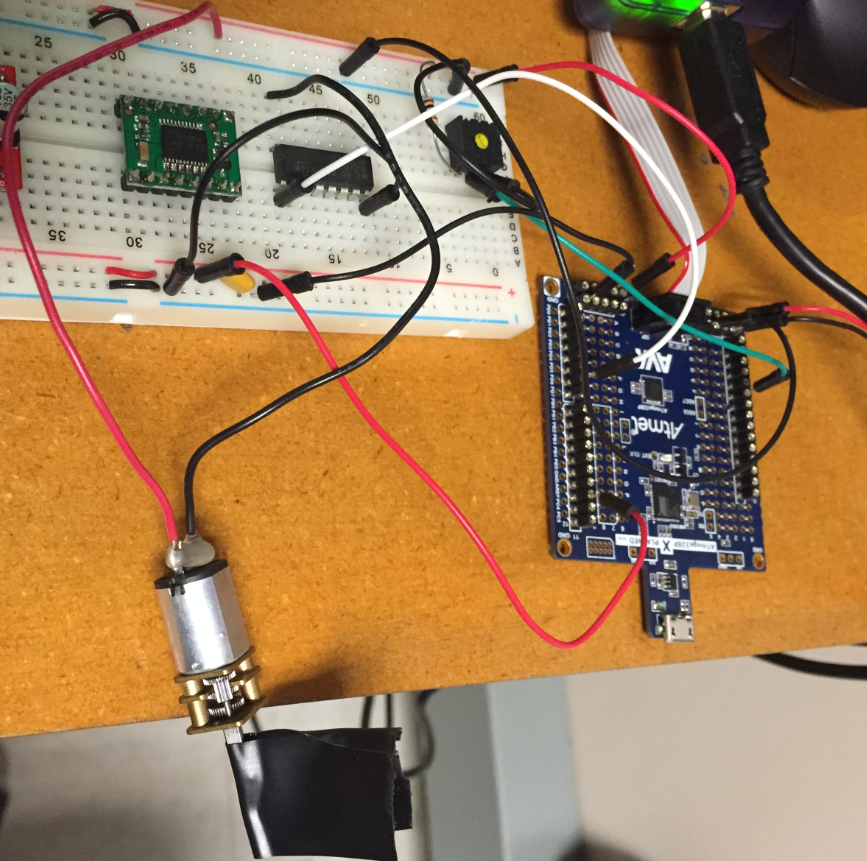


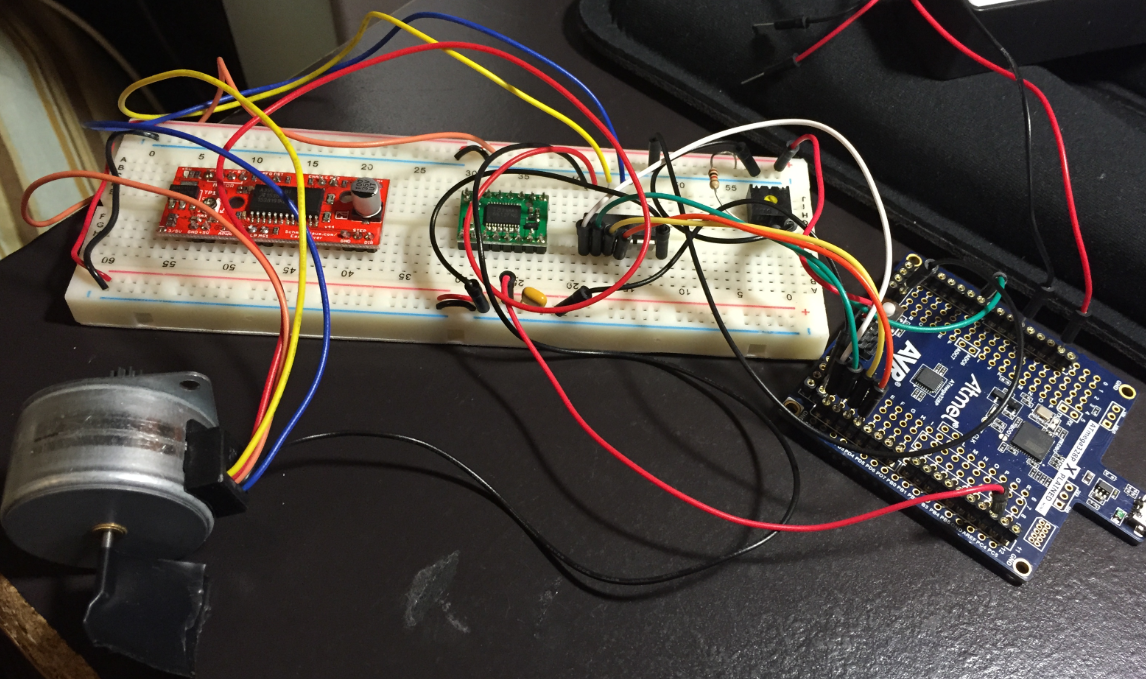
|  |  |  |  |
| --- | --- | --- | --- |
| 6. | SCREENSHOTS OF EACH TASK OUTPUT |  |  |

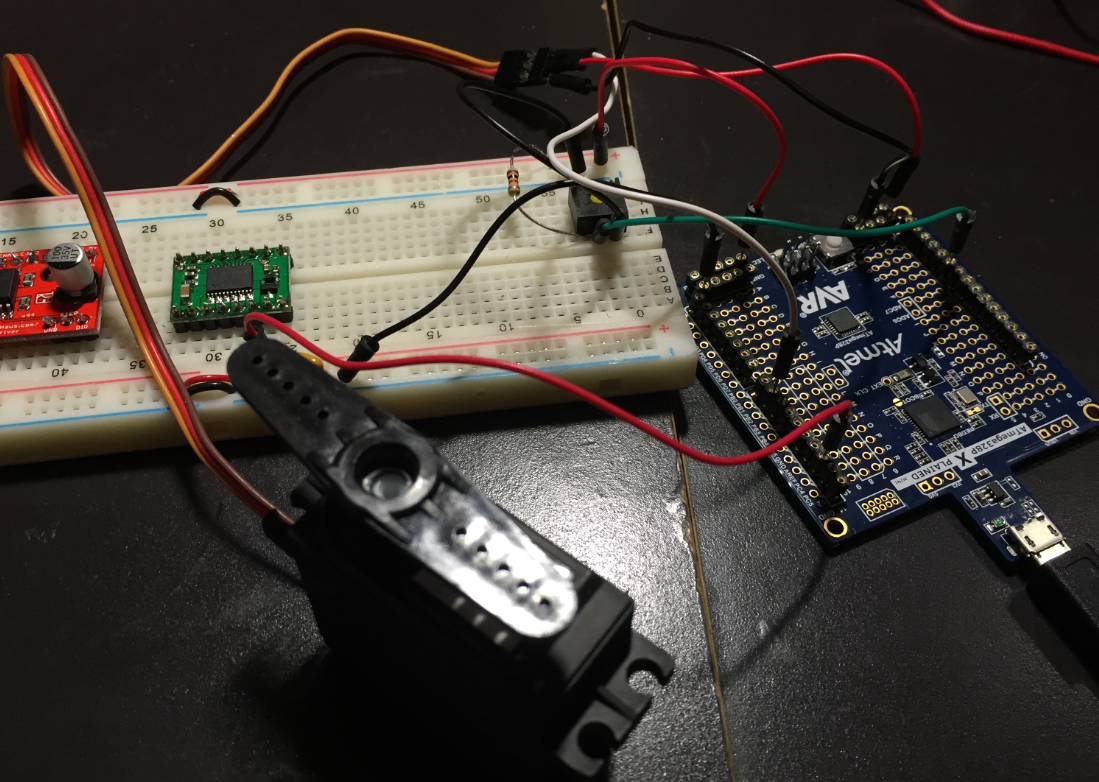
For each task output, a potentiometer is used to modify motor speed or motor position. Adjusting the knob clockwise or counterclockwise adjusts the resistance/voltage.

|  |  |  |  |
| --- | --- | --- | --- |
| 7. | SCREENSHOT OF EACH DEMO |  |  |

On the DC Motor and Stepper Motor, an electrical tape “flag” is added to show movement and position.

DC Motor: 

Stepper Motor: 

Servo Motor: 

|  |  |  |  |
| --- | --- | --- | --- |
| 8. | VIDEO LINK OF DEMO |  |  |
|  | <https://www.youtube.com/watch?v=G7qyfxZUcYY> | DC |  |
|  | <https://www.youtube.com/watch?v=M8q3btu3_LY> | STEPPER |  |
|  | <https://www.youtube.com/watch?v=SFeGjCNPtE4> | SERVO |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 9. | GITHUB LINK OF THE DA |  |  |
|  | https://github.com/magor1/embedded-design-VM/tree/master/DA5 |  |  |

**Student Academic Misconduct Policy**

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

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

RALPH MAGO