CPE301 – SPRING 2020

MIDTERM 4B

Student Name: Xianjie Cao

Student #:5004222179

Student Email: [caox2@unlv.nevada.edu](mailto:caox2@unlv.nevada.edu)

Primary Github address: <https://github.com/c1029324620/Mocha.git>

Directory: Mocha/DesignAssignments/LAB4/DA4B

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

Atmel Studio 7: Assembler, simulator, and Debugger

Atmega328pb-Xmini PC: multi-functional shield and breadboard

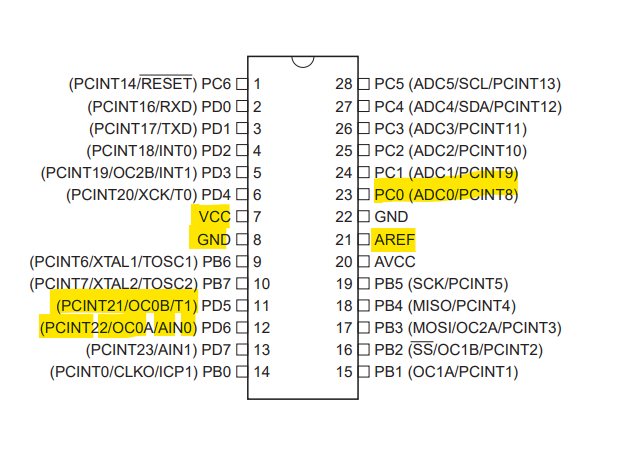
Step motor

Servo motor

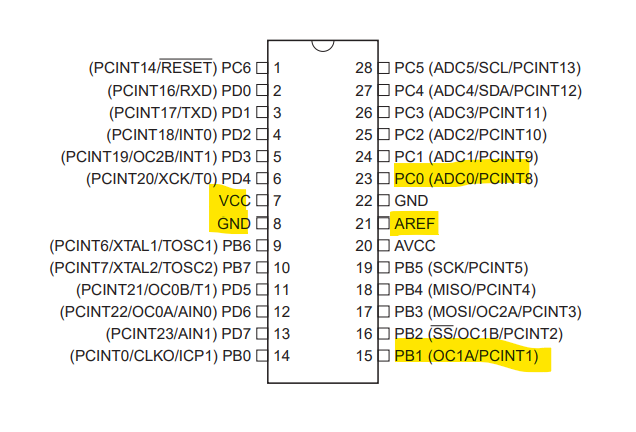
TB6612FNG

ULN2003AN

Task 1:



Task 2:



1. **INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1/A**

/\*

\* DA4BT1.c

\*

\* Created: 4/6/2020 12:33:28 PM

\* Author : c1029

\*/

#define *F\_CPU* 16000000UL

#include <avr/io.h>

unsigned int adc\_value;

void adc\_int()

{

ADMUX = (0<<REFS1) | //AVref

(1<<REFS0) |

(1<<ADLAR) | //Left justified

(0<<MUX2) | //ADC0 channel 0

(0<<MUX1) |

(0<<MUX0);

ADCSRA = (1<<ADEN) | //Enable ADC

(0<<ADSC) |

(0<<ADATE)|

(0<<ADIF) |

(0<<ADIE) |

(1<<ADPS2)| //prescaler of 128

(1<<ADPS1)|

(1<<ADPS0);

}

void adc\_read()

{

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

while(ADCSRA & (1<< ADSC)); //wait

adc\_value = ADCH; // potentiometer value stored in adc\_value

}

//control delay by timer0 in CTC mode

void my\_delay()

{

OCR0A = adc\_value;

int cycle = 100;

while(cycle != 0)

{

cycle--;

while((TIFR0 & 0x02) == 0);

TIFR0 = 1<<OCIE0A;

}

}

int main(void)

{

/\* Replace with your application code \*/ adc\_int();

TCCR0A |= (1<<WGM01); //CTC mode

TCCR0B |= (0<< WGM02)|(1<<CS02) |(1<<CS00); //prescaler of 1024

TCNT0 = 0x00;

PORTC |= (1<<0); //Pull Up Resistor Activated PC0

DDRD = 0xF0; //make PORTD upper pins as output

while (1)

{

//counter-clockwise and full step

PORTD=0x90;

adc\_read(); //read potentiometer value

my\_delay(); //set the potentiometer value as OCRA value

PORTD=0x30;

adc\_read();

my\_delay();

PORTD=0x60;

adc\_read();

my\_delay();

PORTD=0xC0;

adc\_read();

my\_delay();

}

}

1. **DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A**

/\*

\* DA4BT2.c

\*

\* Created: 4/6/2020 4:50:31 PM

\* Author : c1029

\*/

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <util\delay.h>

unsigned int adc\_value;

void adc\_int()

{

ADMUX = (0<<REFS1) | //AVref

(1<<REFS0) |

(1<<ADLAR) | //Left justified

(0<<MUX2) | //ADC0 channel 0

(0<<MUX1) |

(0<<MUX0);

ADCSRA = (1<<ADEN) | //Enable ADC

(0<<ADSC) |

(0<<ADATE)|

(0<<ADIF) |

(0<<ADIE) |

(1<<ADPS2)| //prescaler of 128

(0<<ADPS1)|

(1<<ADPS0);

}

void adc\_read()

{

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

while(ADCSRA & (1<< ADSC)); //wait

adc\_value = ADCH; // potentiometer value stored in adc\_value

}

int main(void)

{

adc\_int(); //Initial ADC

PORTC |= (1<<0); //Pull Up Resistor Activated PC0

//Configuration of Timer1

TCCR1A |= (1<<COM1A1)|(1<<COM1B1);

TCCR1A |= (1<<WGM11); //CTC Mode

TCCR1B |= (1<<WGM12)|(1<<WGM13); //CTC Mode

TCCR1B |= (1<<CS10) |(1<<CS11); //Prescaler 64

ICR1 = 4999; //freq = 50Hz, Period = 20ms

DDRB |= (1<<DDRB1);

while (1)

{

adc\_read();

if(adc\_value < 64)

{

OCR1A = 97; //0 degree.

}

else if(adc\_value >= 64 && adc\_value < 128)

{

OCR1A = 316; //90 degree.

}

else if(adc\_value >= 126 && adc\_value < 192)

{

OCR1A = 426; //135 degree

}

else

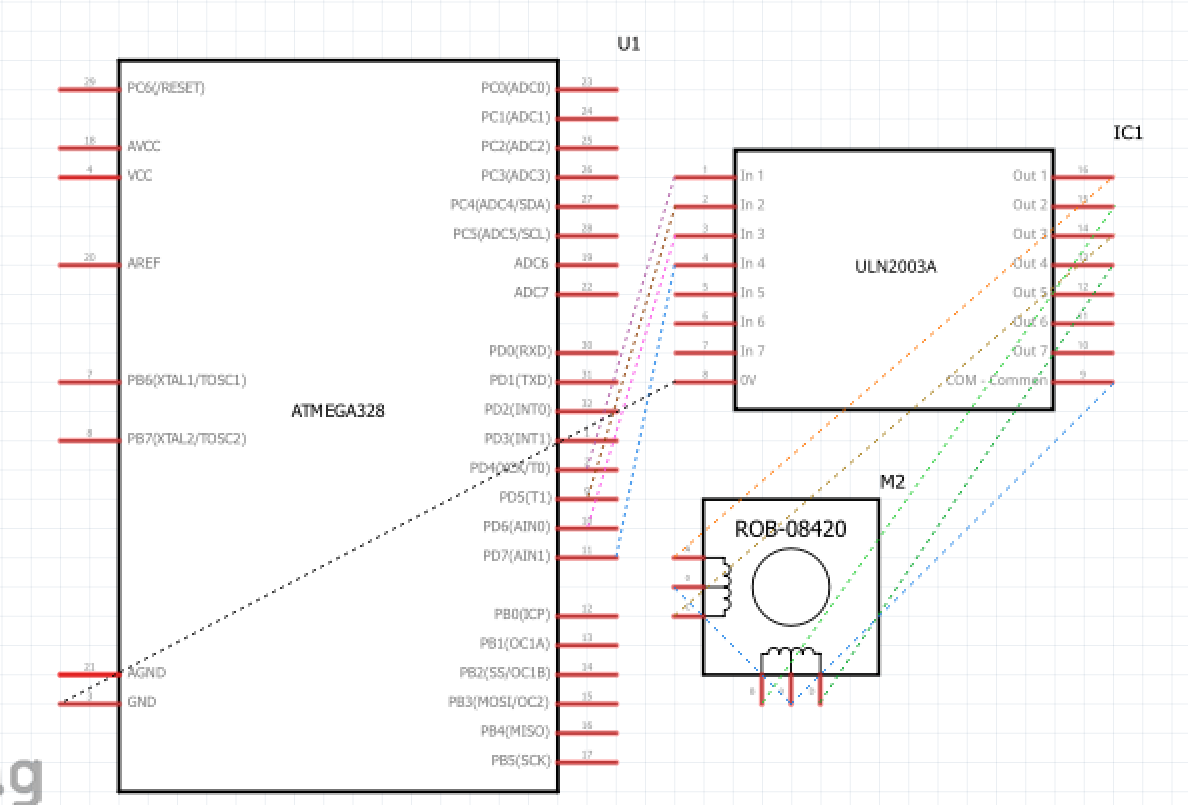
OCR1A = 535; //180 degree

}

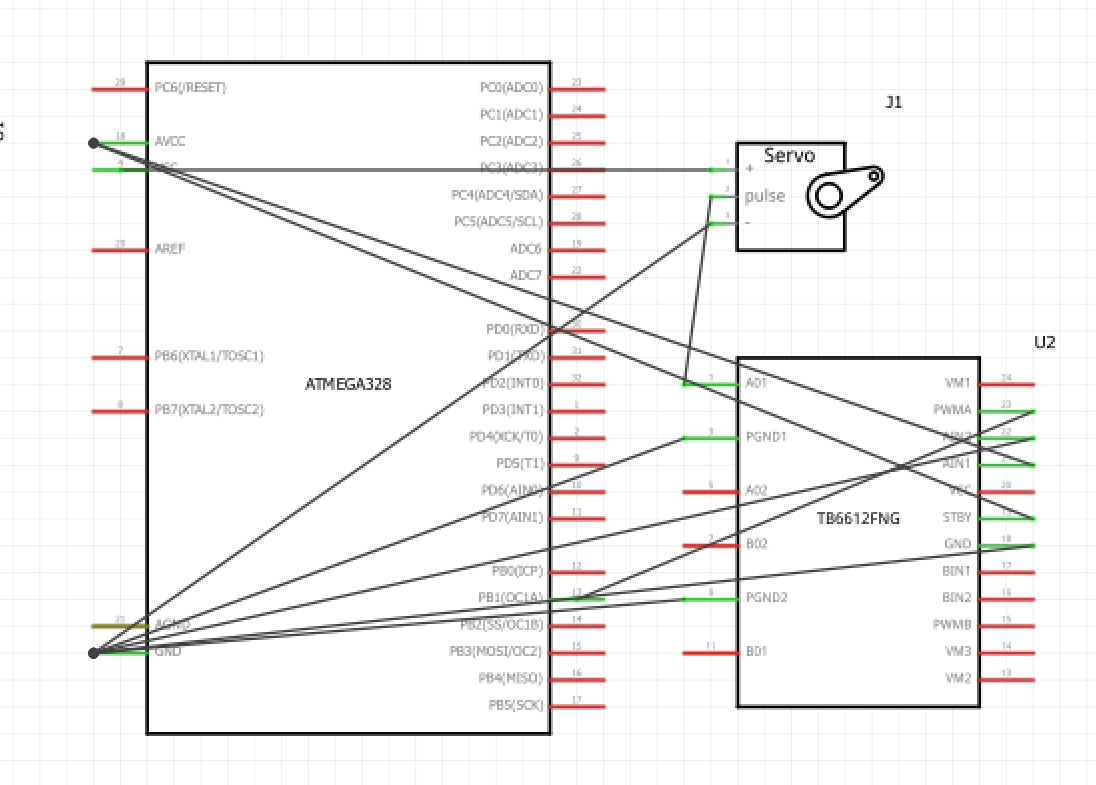
}

1. **SCHEMATICS**

Task 1:



Task 2:

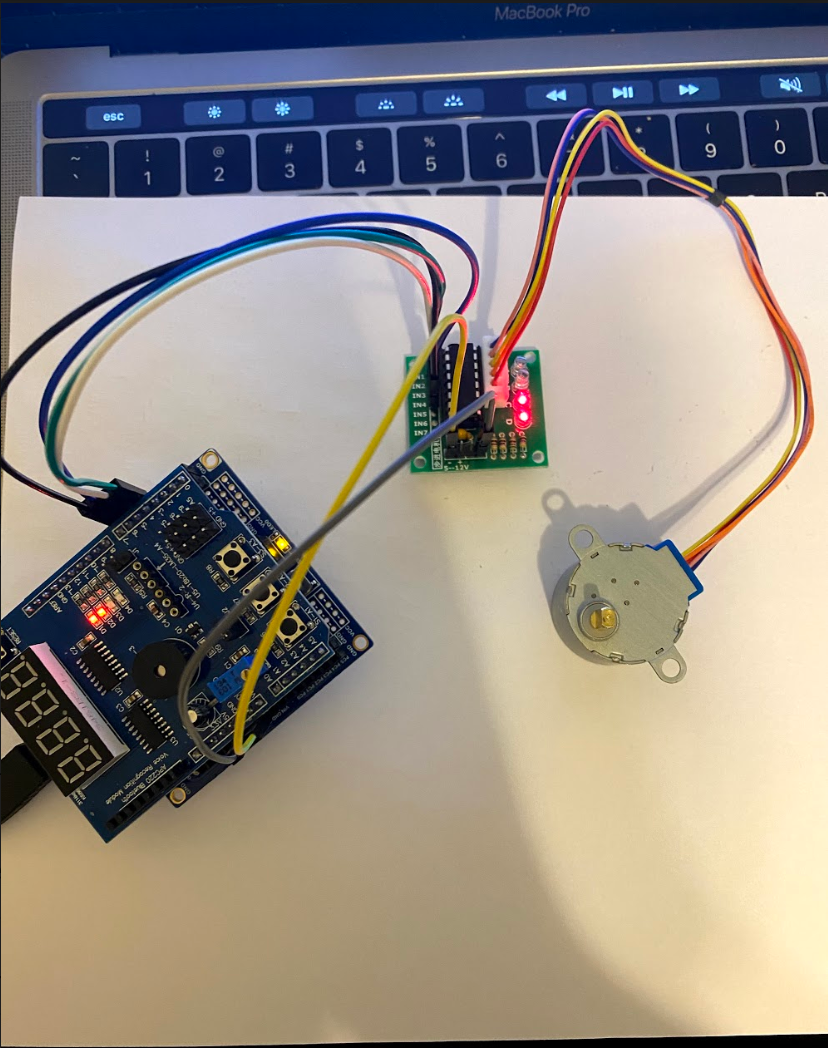


1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**

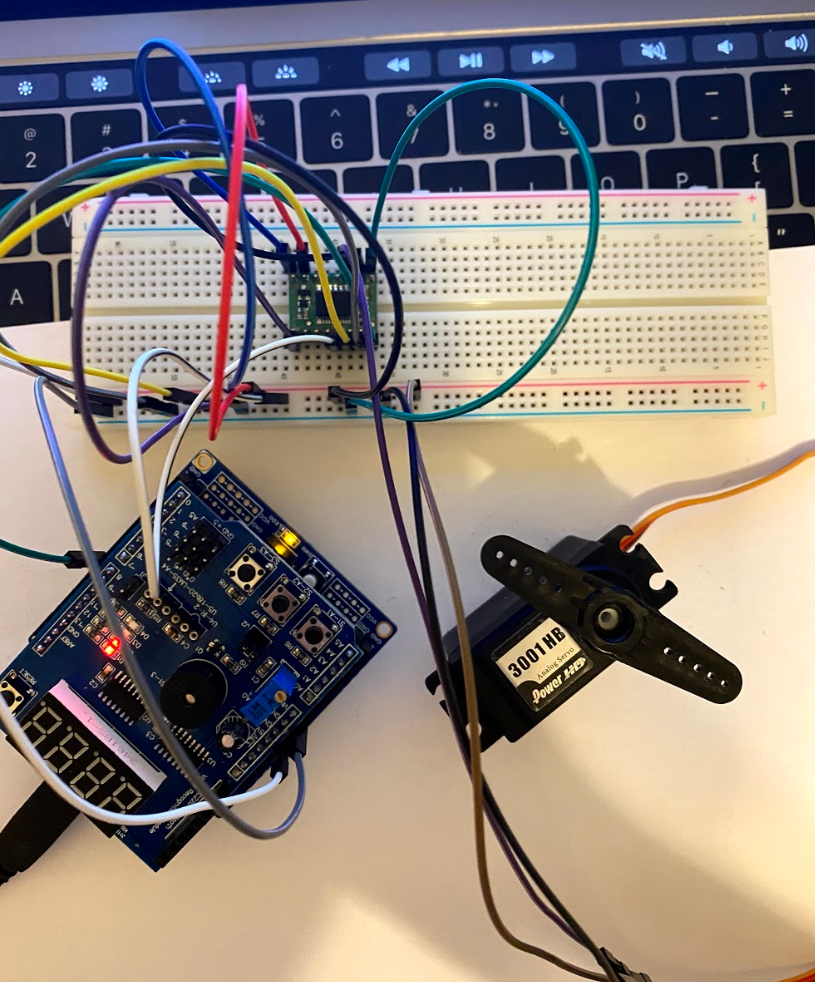
N/A

1. **SCREENSHOT OF EACH DEMO (BOARD SETUP)**

Task 1:



Task 2:



1. **VIDEO LINKS OF EACH DEMO**

Task 1: <https://youtu.be/6kh1by7SA_A>

Task 2: <https://youtu.be/dV_hGSUhvfo>

1. **GITHUB LINK OF THIS DA**

**Student Academic Misconduct Policy**

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

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

Xianjie Cao