CPE301 – SPRING 2021

Design Assignment 4B

Student Name: Brian Wolak

Student #: 2000509437

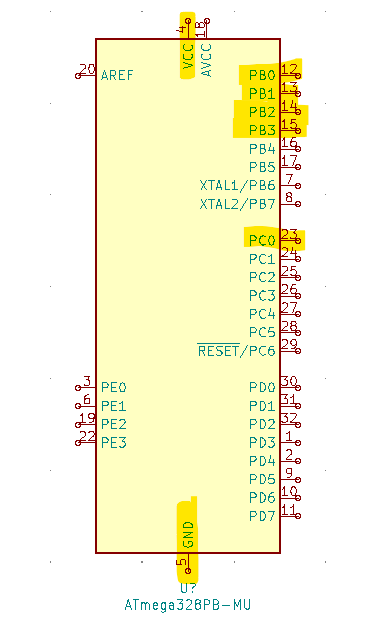
Student Email: wolak@unlv.nevada.edu

Primary Github address: <https://github.com/brianwolak/submission_da.git>

Directory: [submission\_da/DA\_4B at main · brianwolak/submission\_da (github.com)](https://github.com/brianwolak/submission_da/tree/main/DA_4B)

**Task 1:**

Task one for this design assignment involves writing a C code to interface a stepper motor and control its speed using a potentiometer connected and read from pin PCO.



*Design Assignment 4B Task #1 Ports Used*

**Video Link:**

<https://youtu.be/VSgTz4x6sms>

**C Code:**

#define *F\_CPU* 16000000UL

#define BAUD 9600

#include <avr/io.h>

#include <util/setbaud.h>

#include <avr/interrupt.h>

#include <util/delay.h>

#include <stdio.h>

volatile unsigned int pot; //temporary read value from ADC

int delay = 0; //OCR2A manipulation value

int loop = 0; //timer looping value

char display1[20]; //display output string

//usart initialize

void USART\_init(void)

{

UBRR0H = *UBRRH\_VALUE*;

UBRR0L = *UBRRL\_VALUE*;

UCSR0C = \_BV(UCSZ01) | \_BV(UCSZ00);

UCSR0B = \_BV(RXEN0) | \_BV(TXEN0);

}

//ADC initialize

void adc\_init(void){

ADMUX = (0 << REFS1) | (1 << REFS0) | (0 << ADLAR) | (0 << MUX3) | (0 << MUX2) |

(0 << MUX1) | (0 << MUX0);

ADCSRA = (1 << ADEN) | (0 << ADSC) | (0 << ADATE) | (0 << ADIF) | (0 << ADIE) | (1

<< ADPS2) | (1 << ADPS1) | (1 << ADPS0);

}

//read ADC value

void read\_adc(void){

ADCSRA |= (1 << ADSC);

while((ADCSRA & (1 << ADIF))==0); //wait for ADIF flag

ADCSRA |= (1 << ADIF); //reset ADIF flag

pot = ADC; //read value

delay = (pot / 4); //modify pot value for OCR2A

if(delay < 20){ //check min pot value

delay = 20; //set min value if below 20

}

OCR2A = delay; //set OCR2A to delay value

}

//timer setup

void timer\_set(void){

TCCR2A |= (0 << COM2A1) | (0 << COM2A0) | (1 << WGM21) | (0 << WGM20); //mode 2 CTC top 0xFF non-invert

TCCR2B |= (0 << WGM22) | (1 << CS22) | (1 << CS21) | (1 << CS20); //1024 prescale set

}

void motor\_delay(){

TCNT2 = 0; //reset timer2

while (OCR2A != TCNT2); //compare timer2 with OCR2A delay

loop++; //increment loop

}

int main(void)

{

DDRB = 0x0F; //set PORTD low bits to output

int position = 0; //motor position variable

//initialize function call

USART\_init();

//initialize function call

adc\_init();

//initialize function call

timer\_set();

//turn on interrupts

sei();

while (1)

{

read\_adc(); //read pot value call

motor\_delay(); //motor delay call

if(loop == 4){

switch(position){

case 0: //motor case 0

PORTB = 0x03; //pin 0 and 1 high

position++; //increment position

loop = 0; //reset loop

break;

case 1: //motor case 1

PORTB = 0x06; //pin 1 and 2 high

position++; //increment position

loop = 0; //reset loop

break;

case 2: //motor case 2

PORTB = 0x0C; //pin 3 and 4 high

position++; //increment position

loop = 0; //reset loop

break;

case 3: //motor case 3

PORTB = 0x09; //pin 0 and 3 high

position = 0; //reset position

loop = 0; //reset loop

break;

}

}

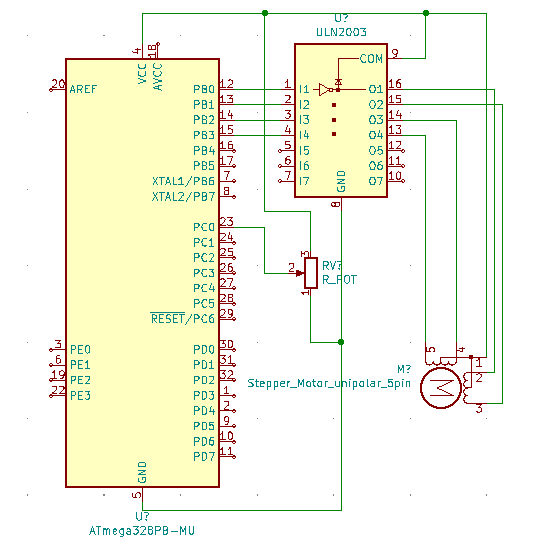
}

}

A picture containing text

Description automatically generated

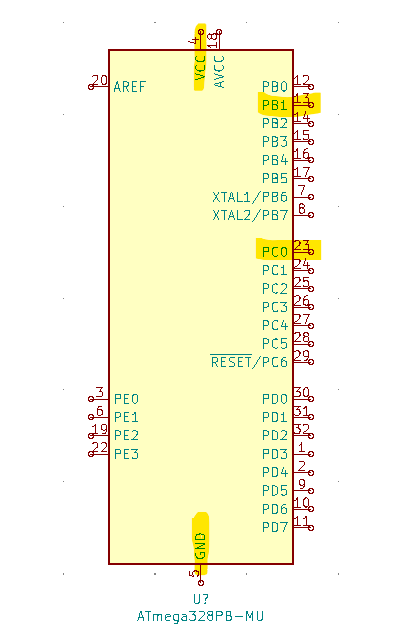
*Design Assignment 4B Task 1 Board Setup*



*Design Assignment 4B Task #1 Circuit*

**Task 2:**

Task two for this design assignment involves writing a C code to interface a servo motor and control its position using a potentiometer connected and read from pin PCO. The servo motor will perform a smooth 1800 sweep when a sweep of the potentiometer is read with potentiometer value of zero corresponding to 00 and 5V to the 1800 position.



*Design Assignment 4B Task #2 Ports Used*

**Video Link:**

<https://youtu.be/VaR87Lpuei0>

**C Code:**

#define *F\_CPU* 16000000UL /\* Define CPU Frequency 16MHz \*/

#include <avr/io.h> /\* Include AVR std. library file \*/

#include <util/delay.h> /\* Include Delay header file \*/

volatile unsigned int pot; //temporary read value from ADC

//ADC initialize, PC0 input

void adc\_init(void){

ADMUX = (0 << REFS1) | (1 << REFS0) | (0 << ADLAR) | (0 << MUX3) | (0 << MUX2) |

(0 << MUX1) | (0 << MUX0);

ADCSRA = (1 << ADEN) | (0 << ADSC) | (0 << ADATE) | (0 << ADIF) | (0 << ADIE) | (1

<< ADPS2) | (1 << ADPS1) | (1 << ADPS0);

}

//read ADC value

void read\_adc(void){

ADCSRA |= (1 << ADSC);

while((ADCSRA & (1 << ADIF))==0); //wait for ADIF flag

ADCSRA |= (1 << ADIF); //reset ADIF flag

pot = ADC; //read value

OCR1A = 50 + pot / 2; //set pot value to OCR1A with

}

//timer1 setup function

void t1\_setup(void){

TCCR1A|=(1<<COM1A1)|(1<<COM1B1)|(1<<WGM11); //NON Inverted PWM

TCCR1B|=(1<<WGM13)|(1<<WGM12)|(1<<CS11)|(1<<CS10); //PRESCALER=64 MODE 14(FAST PWM)

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

}

int main(void)

{

t1\_setup(); //timer1 intialize

adc\_init(); //initialize ADC

DDRB|=(1<<PINB1); //PWM output on PB1

while(1)

{

read\_adc(); //read ADC

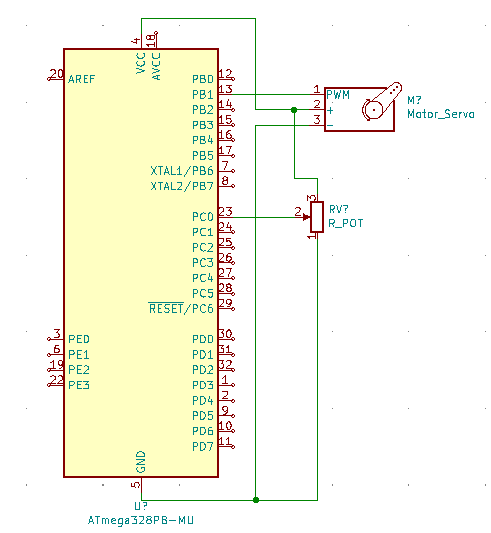
}

}

A picture containing floor, indoor, connector

Description automatically generated

*Design Assignment 4B Task 2 Board Setup*



*Design Assignment 4B Task #2 Circuit*