CPE301 – SPRING 2021

Design Assignment 4A

Student Name: Brian Wolak

Student #: 2000509437

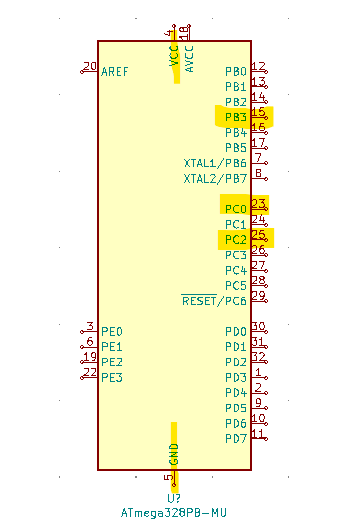
Student Email: wolak@unlv.nevada.edu

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

Directory:

**Task 1:**

The purpose of this design assignment is to create an AVR C code that will control the speed of a motor using a potentiometer connected to PC0. Using an interrupt connected to PC2 the motor can be started and stopped with each button press. The minimum motor sped should be between 0 and 95% with an absolute maximum PWM speed of 95% read by the potentiometer.



*ATMEGA328pb Ports Used in Design Assignment 4A*

**Video Link:**

<https://youtu.be/QWDmSaGsQq0>

**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; //temp read value from ADC

float pot2 = 0; //OCR2A manipulation 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);

}

//initialize interrupt function

void interrupt\_set(void){

PCICR |= (1<<1); //enable pin change interrupt 1

PCMSK1 |= (1<<PCINT10); //setup pin change interrupt PC2

TCCR0A = 0x83;

TCCR0B = 0x01;

TIMSK0 = 0x02;

}

//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);

}

//usart transmit function

void USART\_TX\_string(char \*data){

while (\*data != '\0'){ //while data DNE 0

while (!(UCSR0A & (1<<UDRE0))); //while UNDRE0 DNE 1

UDR0 = \*data; //UDR0 gets data value

data++; //next data value

}

}

//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

}

//timer setup

void timer\_set(void){

TCCR2A |= (1 << COM2A1) | (1 << WGM21) | (1 << WGM20); //mode 3 fast PWM top 0xFF non-invert

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

TIMSK2 |= (1 << OCIE2A);

PCMSK1 |= (1 << PCINT10); //turn on pin change interrupt PC2

PCICR |= (1 << PCIE1); //PCINT1 interrupt

}

int main(void)

{

DDRB |= (1<<3); //set PORT B output PIN3

PORTC |= (1<<2); //set pull-ups PINC 0 & 2

//initialize function call

USART\_init();

//initialize function call

adc\_init();

//initialize function call

timer\_set();

//turn on interrupts

sei();

USART\_TX\_string("Device is Connected..\r\n");

while (1)

{

read\_adc(); //read pot value

*sprintf*(display1, "%d", pot); //convert pot to string

*\_delay\_ms*(100); //10ms delay

//terminal print statements for potentiometer value

USART\_TX\_string("Potentiometer value is: ");

USART\_TX\_string(display1);

USART\_TX\_string("\n");

if (pot > 972){

pot = 972;

}

pot2 = (pot / 972.0) \* 255; //modify pot value for OCR2A

OCR2A = pot2; //set OCR2A to pot2 value

}

}

//PORTC pin2 interrupt

ISR(PCINT1\_vect){

if(PINC & (1 << 2)){ //if PINC 2

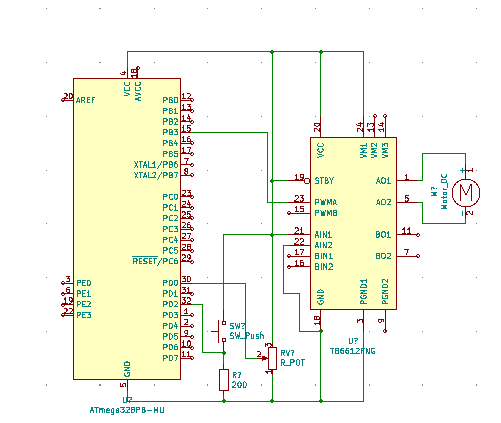
TCCR2A ^= (1 << COM2A1); //toggle COM2A1

}

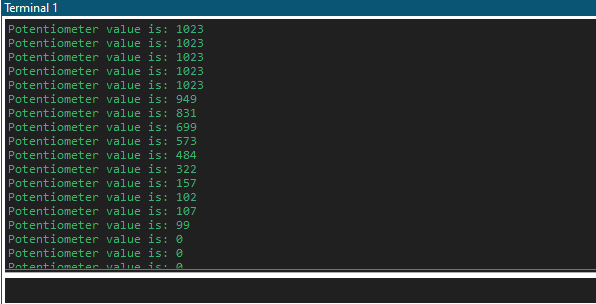
}

ISR(BADISR\_vect){

}



*Design Assignment 4A Circuit*



*Design Assignment 4A sweep of potentiometer read in terminal*