CPE301 – SPRING 2019

Design Assignment 4A

Student Name: Brian West

Student #: 5003032874

Student Email: westb2@unlv.nevada.edu

Primary Github address: <https://github.com/westbrian2/Spring2019>

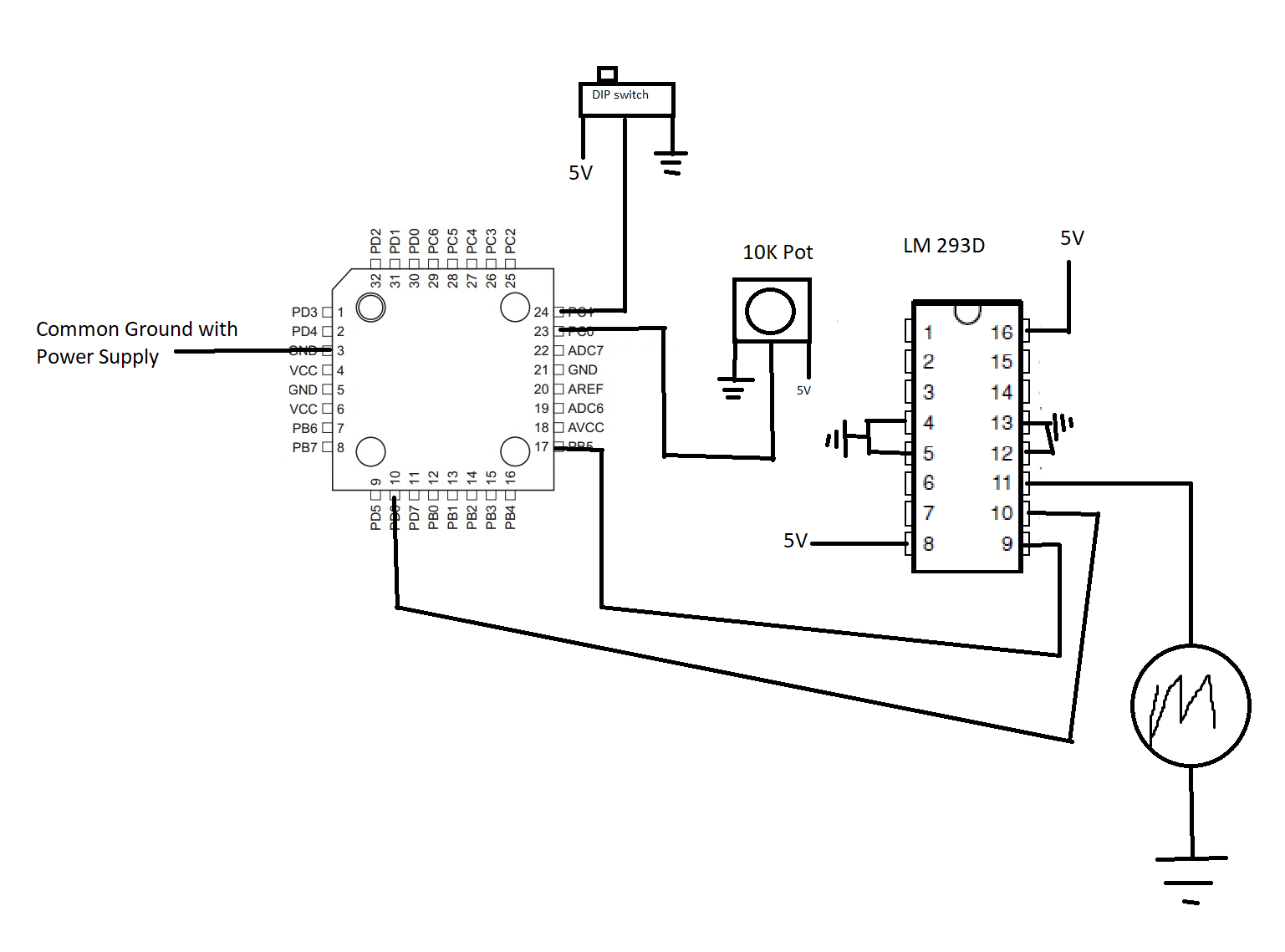
Directory: Spring2019/DesignAssignments

Submit the following for all Labs:

1. In the document, for each task submit the modified or included code (only) with highlights and justifications of the modifications. Also, include the comments.
2. Use the previously create a Github repository with a random name (no CPE/301, Lastname, Firstname). Place all labs under the root folder ESD301/DA, sub-folder named LABXX, with one document and one video link file for each lab, place modified asm/c files named as LabXX-TYY.asm/c.
3. If multiple asm/c files or other libraries are used, create a folder LabXX-TYY and place these files inside the folder.
4. The folder should have a) Word document (see template), b) source code file(s) and other include files, c) text file with youtube video links (see template).

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

Mini Xplained board, power supply, L293DNE, DC motor, switch, potentiometer(5K?)



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

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <avr/interrupt.h>

#include <util/delay.h>

void read\_adc(void); //gets value from potentiometer

volatile unsigned int adc\_temp;

int main(void){

int speed=0;

//set up output/input

DDRD=(1<<6); //setting pinD6 to output

DDRB=(1<<5); //setting pinb5 to output

DDRC=(0<<1); //setting pinc1 as input (PinC0 is ADC)

PORTB=(0<<5); //setting enable pin low

//setting up pin change interrupt

PCICR=(1<<PCIE1); //enabling pins 8-14

PCMSK1=2; //enabling pcint9 (PC1)

//Set up ADC

ADMUX=((0<<REFS1)|(1<<REFS0)|(1<<ADLAR)|(0<<MUX3)|(0<<MUX2)|(0<<MUX1)|(0<<MUX0)); //reference Vcc, left shift(8bits), and PC0 is input

ADCSRA=((1<<ADEN)|(0<<ADSC)|(0<<ADATE)|(0<<ADIF)|(0<<ADIE)|(1<<ADPS2)|(0<<ADPS1)|(1<<ADPS0)); //prescaler=32, enable with no interrupts

//Set up timer0 for PWM

OCR0A=0x88; //set for ~50 duty cycle

TCCR0A|=((1<<COM0A0)|(1<<WGM01)|(1<<WGM00)); //setting to fast PWM, top of 0xFF, and no prescaler

TCCR0B|=((1<<WGM02)|(1<<CS00)); //starts timer

sei();

while(1){

//poll the adc value and update speed based on value

read\_adc(); //gets value from pot

if(adc\_temp>=0xF2)

OCR0A=0xF2;// max value is 95% of top

else

OCR0A=adc\_temp; //assign value

}

}

void read\_adc(void) {

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

while(ADCSRA & (1<<ADSC)); //waiting for coversion to finish

adc\_temp= ADC;

}

ISR(PCINT1\_vect) { //when pinC1 is hit

PORTB^=(1<<5); //toggle enable

*\_delay\_ms*(100); //debouncing

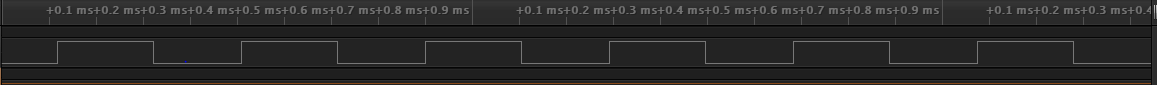
}

1. **SCHEMATICS**

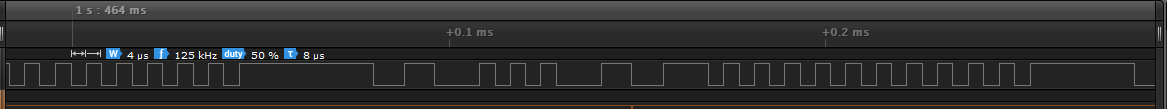


1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**

**Wave form from midpoint on potentiometer**



Quickly turning back and forth



1. **VIDEO LINKS OF EACH DEMO**

<https://www.youtube.com/watch?v=YV1AaTyAUWc&feature=youtu.be>

1. **GITHUB LINK OF THIS DA**

<https://github.com/westbrian2/Spring2019>/DesignAssignments/DA4A\_submission

**Student Academic Misconduct Policy**

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

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

Brian West