CPE301 – FALL 2019

Design Assignment 4A

Student Name: Cody Jones

Student #: 5002863328

Student Email: Jonesc30@unlv.nevada.edu

Primary Github address: https://github.com/Jonesc30/Submission

Directory: Submission

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**

Atmega328PB and shield, FTDI, PWM DC Motor, wires and DC Motor Driver

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>

volatile unsigned int ADC\_Val; // holds value of ADC

int toggle = 0; // used to toggle

int main(void)

{

DDRD |= 0x40; //PWM output

DDRC = 0x02;

// PC0 as input and PC1 is now an output

PORTC |= (1<<PC1);

DDRB |= (1<<2); //PB1 as output to stop the motor on and off

PORTB |= (1<<2); //Stops and turns it on

TCCR0A = 0x83;

// set none-inverting mode

// set fast PWM Mode for timer0 max height is 256

TCCR0B = 0x05;

// set prescaler to 1024 and starts PWM

PCICR = (1<<PCIE1); //Enable PCINT

PCMSK1 = (1<<PCINT9); // Set interupt for PC1

//Set up ADC

ADMUX = (1<<REFS0); // PC0 is input for ADC

ADCSRA |= (1<<ADEN) | (1<<ADSC) | (1<<ADATE) | (1<<ADPS2) | (1<<ADPS1) | (1<<ADPS0); // enable ADC, system clock used for ADC

sei(); //enable interrupts

while (1);

}

//ISR for PC1

ISR(PCINT1\_vect)

{

if(!(PINC & (1 << PINC1)))

{

if(toggle == 0)

{

OCR0A = 0;

PORTB &= ~(1 << 2); //turn off OCR0A

}

else if(toggle == 1)

{

PORTB |= (1 << 2); //turn on OCR0A

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

while((ADCSRA&(1<<ADIF))==0)

{ // wait for conversion to finish

ADC\_Val = ADC & 0x03FF; // we only need first 10 bits

OCR0A = 10\*ADC\_Val; // Calculate duty cycle

}

}

toggle ^= 1; //toggle

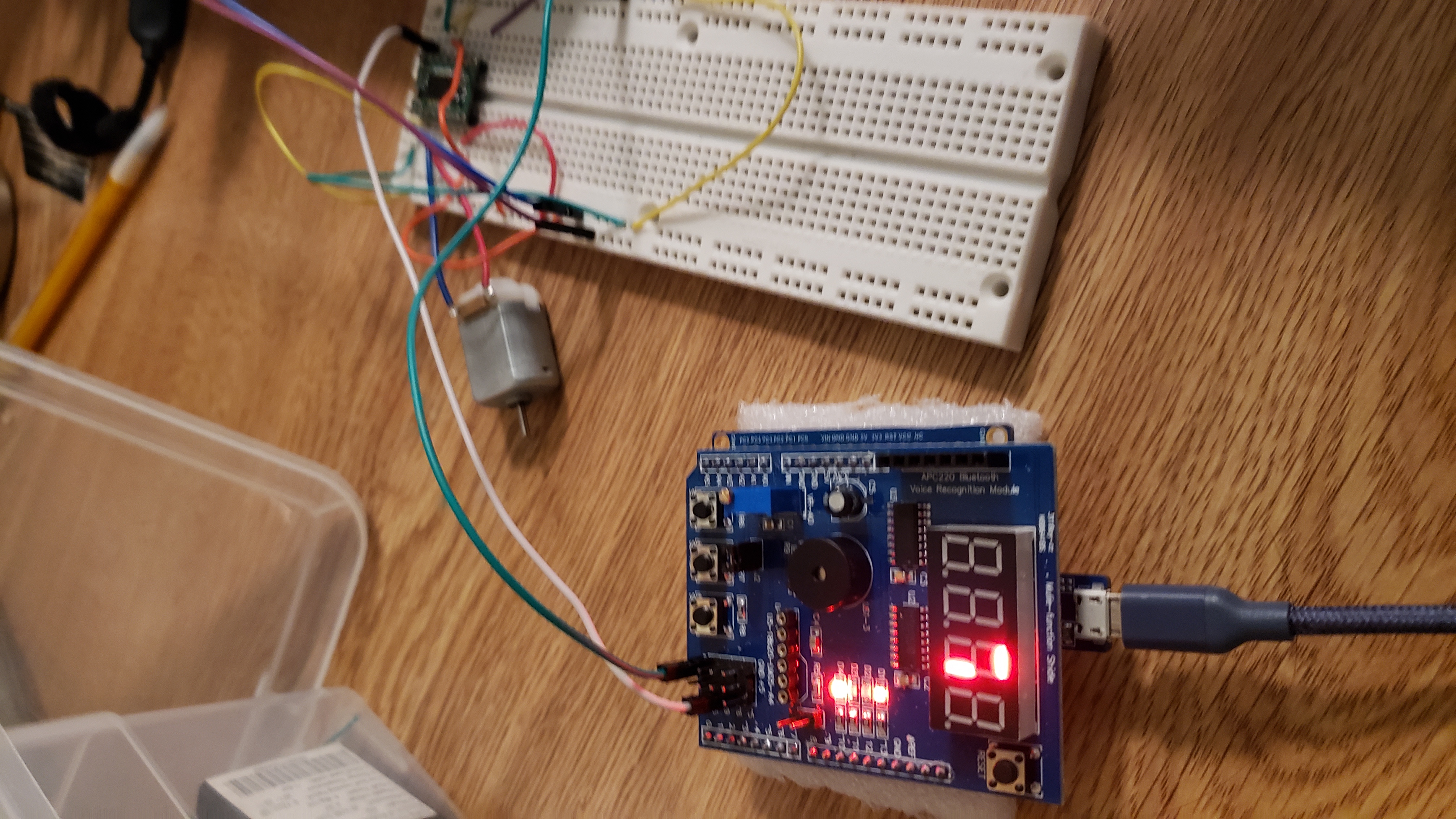
}

}

1. **SCHEMATICS**

Use fritzing.org

1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**
2. **SCREENSHOT OF EACH DEMO (BOARD SETUP)**



1. **VIDEO LINKS OF EACH DEMO**

<https://www.youtube.com/watch?v=YOYFHud2kk4>

1. **GITHUB LINK OF THIS DA**

<https://github.com/Jonesc30/Submission/tree/master/DesignAssignments>

**Student Academic Misconduct Policy**

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

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

Cody Jones