CPE301 – SPRING 2019

Design Assignment 4A

Student Name: Moriah Wingrove

Student #: 5000156247

Student Email: Wingrove@unlv.nevada.edu

Primary Github address: <https://github.com/windew/Tiny_Dragons.git>

Directory:

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**

Atmega 328pb

pushbutton

HiLetgo Multi-nction Shield ProtoShield Multi-functional Expansion Board Sensor Shield Module witn

DC Toy Motor

**INITIAL/MODIFIED/DEVELOPED CODE OF TASK 3B**

**C code**

/\*

\* Design\_Assignment\_4A.c

\*

\* Created: 10/29/2019 7:55:13 PM

\* Author : Moriah Wingrove

\*/

//Task One: Write an AVR C program to control the speed of the DC Motor using a potentiometer connected to PC0.

// Use an interrupt on a button (PC1/2/3) to stop and start the motor at each click. The minimum speed

// of the motor should be 0 when pot is at minimum position and at maximum position should be 95%

// of PWM value

#define *F\_CPU* 8000000UL // clock is 8MHz

#include <avr/io.h>

#include <avr/interrupt.h>

#include <util/delay.h>

void init\_adc(void);

int control; // controlling the motor with potentiometer

int toggle = 0; // to toggle the motor with the switch

int main()

{

DDRC = 0xFB; // set Port C as outputs, leave INT1(PC3) as interrupt

PORTC = 0x00; // pull-up pins

EIMSK = 0x02; // enable INT1

EIFR = 0x02; // enable interrupt flag

EICRA = 0X0C; // set interrupt on rising edge

sei(); // enable interrupts

init\_adc(); // initiate adc

TCCR0A=0x83; // set fast PWM // clear OCR0A on MATCH

TCCR0B=0x03; // set prescaler to 1024

while (1)

{

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

control = ADC\*80/100; // ADC Conversion

OCR0A = control; // Output to converted value to 0CR0A

if(toggle == 0) // when switch is NOT pressed

{

PORTC = 0x00; //

}

}

}

void init\_adc(void) // Initiate ADC function

{

ADMUX = (1<<REFS0); // Reference voltage at Aref

ADCSRA = (1<<ADEN)|(1<<ADSC)|(1<<ADATE)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0);

// from ADCRSA we Enable ADC, Start Conversion, Set prescalar as 128

}

ISR(INT1\_vect)

{

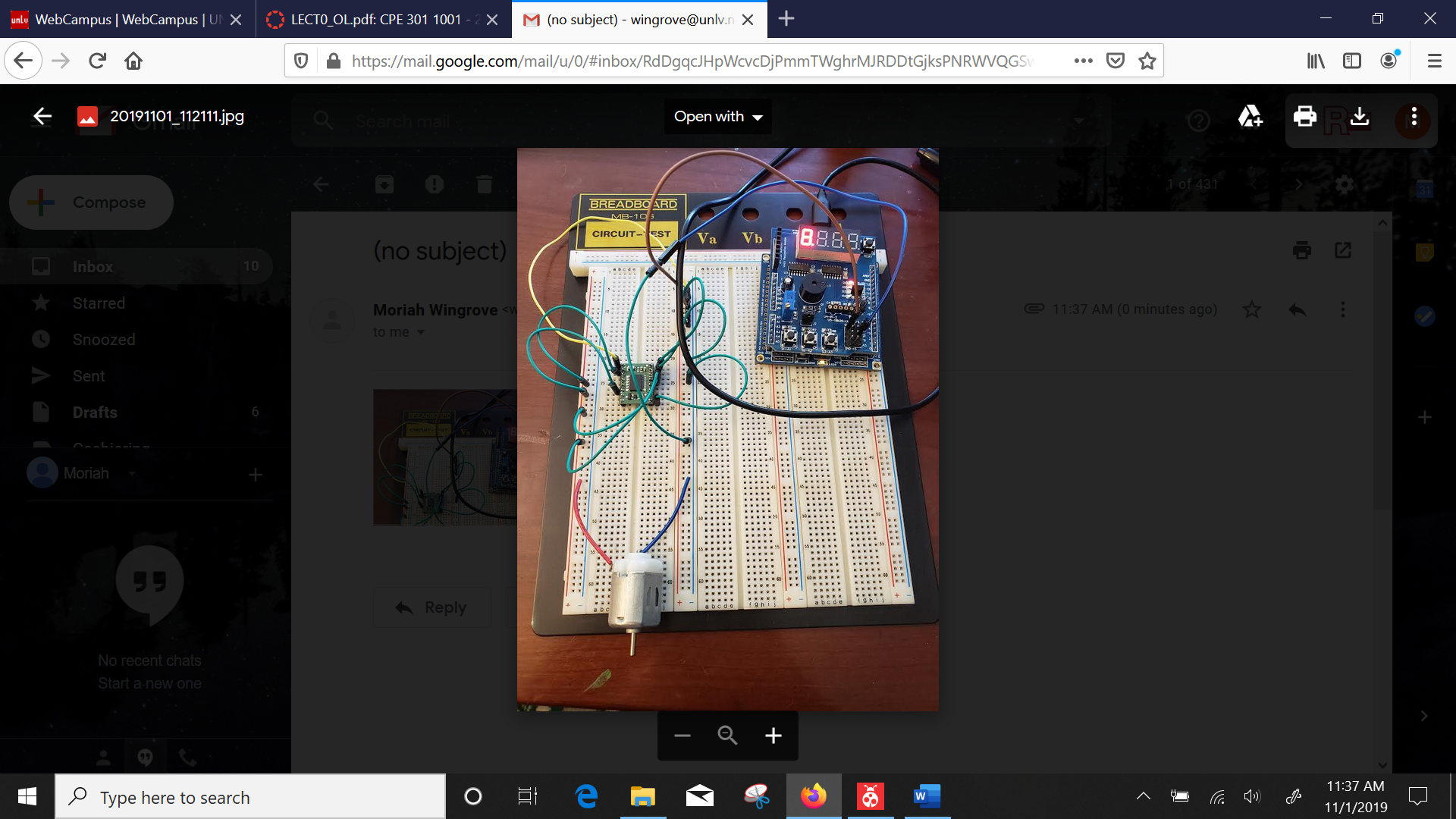
toggle ^= 1; // toggle switch on INT1 Interrupt

}

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**
2. **GITHUB LINK OF THIS DA**

<https://github.com/windew/Tiny_Dragons>

**Student Academic Misconduct Policy**

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

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

NAME OF THE STUDENT