CPE301 – SPRING 2019

Design Assignment 4A

Student Name: Henry Mesa

Student #: 1008099273

Student Email: mesah1@unlv.nevada.edu

Primary Github address: https://github.com/mesah1

Directory: https://github.com/mesah1/submissions

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

List of Components used:

ATmega328PB

Multifunction Shield

Breadboard

DC Motor

MD08A

Block diagram with pins used in the Atmega328PB:



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

/\*

\* Assignment4A.c

\*

\* Created: 11/4/2019 8:01:41 AM

\* Author: Henry Mesa

\*/

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <util/delay.h>

#include <stdio.h>

/\* When switch (SW) goes LOW motor will stop

SW is connected to PC1 always HIGH until SW is

pressed \*/

#define SW (PINC |= (1 << 1))

void read\_adc(void);

void adc\_init(void);

volatile float adc;

int main( ){

adc\_init();

PORTC |= 0b00000011; //pull-up enable PC0 & PC1

DDRB |= (1 << DDB1); // PB1 is an output

ICR1 = 0xFFFF; //set TOP to 16bit

TCCR1A |= (1 << COM1A1)|(1 << COM1B1); // set to non-inverting mode

TCCR1A |= (1 << WGM11);

TCCR1B |= (1 << WGM12)|(1 << WGM13); // set Fast PWM mode using ICR1 as TOP

TCCR1B |= (1 << CS10); //no prescaler

while (SW)

{

read\_adc(); //call function

*\_delay\_ms*(50);

if ((adc >= 62258) && (adc < 65535 )) //keeping pwm at 95% max

{

OCR1A = 62258;

*\_delay\_ms*(20);

}

else if ((adc < 62257) && (adc >= 3277)) //increasing pwm as resistor

{ //value increases

OCR1A = adc + 30000;

*\_delay\_ms*(20);

}

else

OCR1A = 0x00; //if resistor value is less than 5% then pwm is 0%

}

}

/\* INITIALIZING ADC \*/

void adc\_init(void){

//ADC ENABLE AND SET-UP

ADMUX = (0 << REFS1)| // external cap at AREF

(1 << REFS0)| // AVcc - Reference Selection INTERNAL

(1 << ADLAR)| // ADC Left Adjust Result

(0 << MUX2)| // Analog Channel Selection Bits

(0 << MUX1)| // ADC0 (PC0 PIN23)

(0 << MUX0);

ADCSRA = (1 << ADEN)| // ADC ENable

(0 << ADSC)| // ADC Start Conversion

(0 << ADATE)| // ADC Auto Trigger Enable

(0 << ADIF)| // ADC Interrupt Flag

(0 << ADIE)| // ADC Interrupt Enable

(1 << ADPS2)| // ADC Prescaler Select Bits 128

(0 << ADPS1)|

(1 << ADPS0);

}

/\* READ ADC PINS \*/

void read\_adc(void){

unsigned char i = 4;

adc = 0;

while (i--)

{

ADCSRA |= (1 << ADSC);

while(ADCSRA & (1 << ADSC));

adc+= ADC;

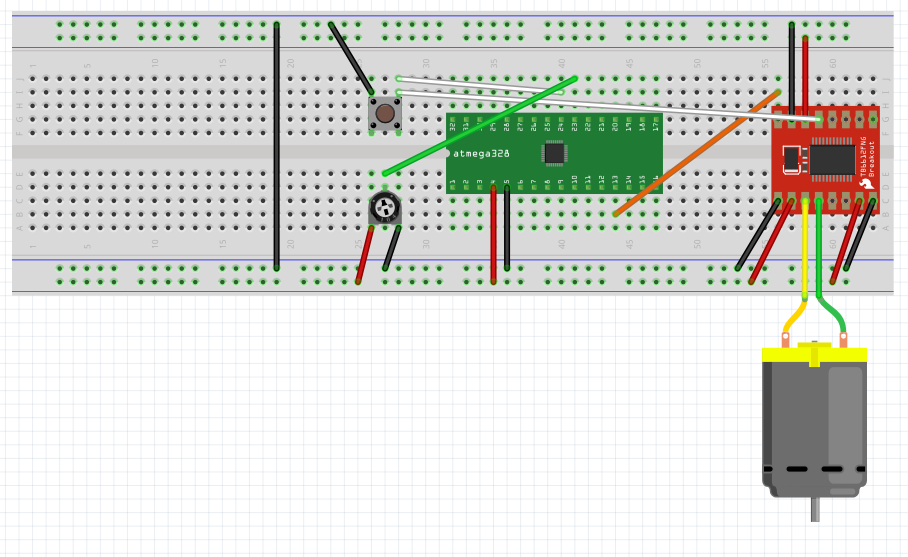
*\_delay\_ms*(50);

}

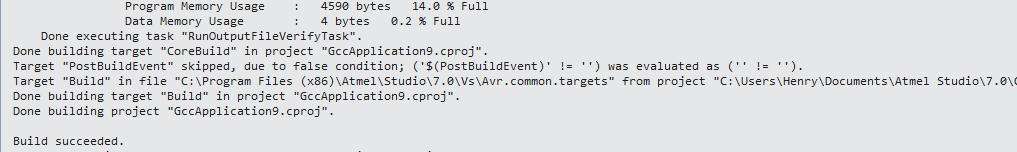
adc = adc / 8; // Average a few samples

}

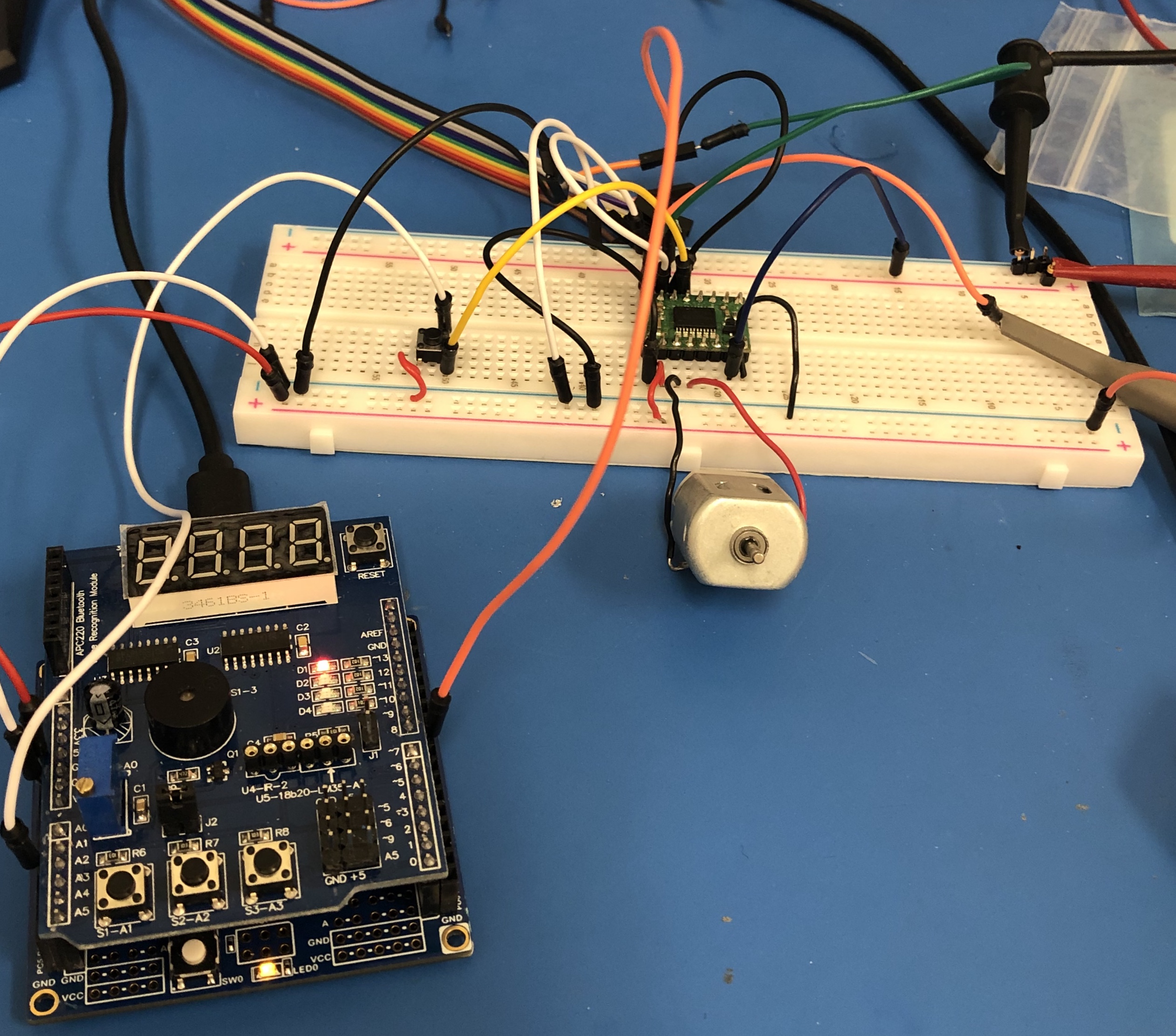
1. **SCHEMATICS**



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



1. **SCREENSHOT OF EACH DEMO (BOARD SETUP)**



1. **VIDEO LINKS OF EACH DEMO**

<https://youtu.be/jgq1-BrRsHk>

1. **GITHUB LINK OF THIS DA**

<https://github.com/mesah1/submissions/tree/master/DA4A>

**Student Academic Misconduct Policy**

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

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

Henry Mesa