Design Assignment 4A

Student Name: Francisco Mata Carlos
Student #: 1012593607
Student Email: matacarl@unlv.nevada.edu
Primary Github address: https://github.com/chicosisco/da_sub.git
Directory: repository/cpe301/DesignAssignments/DA4A

1.   COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

The components used for this assignment are the next:
   a.   Atmega328p Xplained Mini
   b.   Multi-functional Shield
   c.   Atmel Studio 7
   d.   DC motor
   e.   TB6612FNG Driver IC for DC motor
   f.   Breadboard
   g.   CPE310L Lab board

Block diagram with pins used in the Atmega328P
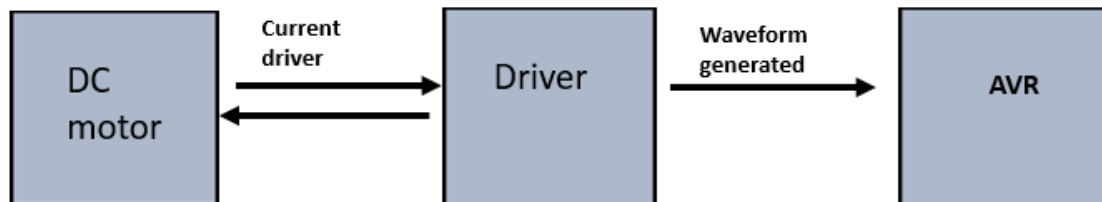
2. INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1/A

Task 1_A
1. 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 minimum, and maximum should be 95% of PWM value.

```c
#define F_CPU 16000000UL    /* clock runs at 16 MHz*/

#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#include <stdio.h>
#define MTR_1 5             //
#define SW (PINC&(1<<2))    // defining SW as the PINC 2, connected to switch 2

int adc_value;      //declaring variable
int motor=0;        //declaring integer to declare motor on or off

void adc_init(void);  // function declared for ADC set up


int main(void)
{


    PORTC |=(1<<2);     //enable pull-up
    DDRD |=(1<<DDD6)|(1<<DDD5); // PD5, PD6 as outputs

    OCR0A = 250;  // setting counter top=250

    // Timer/counter control register is set as follows:
    // clear OC0A on compare match
    // clear OC0B on compare match
    // mode=7, Fast PWM
    // prescaler = 256
    TCCR0A |= (1<<COM0A1)|(0<<COM0A0)|(1<<COM0B1)|(0<<COM0B0)|(1<<WGM01)|(1<<WGM00);
    //TCCR0B |= (1<<CS01);
    TCCR0B |= (1<<WGM02)|(1<<CS02)|(0<<CS01)|(0<<CS00);
    /* Replace with your application code */
    while (1)
    {
            while (!motor)     // while motor is off the next follows
            {
                    if(!SW)    // if PC2 is high or push button was pressed down
                    {
                            _delay_ms(100);
                            adc_init();     // initialize ADC set up

                            {                           //clockwise rotation
                                    OCR0B = ADC;    // adc values read from pot are being
sent to
                                                            // the compare register
```

```c
                                        DDRD = 0xFF;     //setting all D as outputs
                                        PORTD |= (1 <<DDD5);
                                         motor=1; // setting motor to 1 to get out of while
loop
                            }
                        }
                }
                //motor on
                while (motor)
                {
                        if(!SW)
                    {
                                _delay_ms(100);
                                //clockwise rotation
                                // turning motor off
                                    PORTD &= ~(1<<DDD5);
                                    DDRD = 0;
                                    motor=0;
                        }

                }
            }
}

// funtion to initilize ADC set up
void adc_init(void)
{
        ADCSRA |= ((1<<ADEN)|(1<<ADSC)|(0<<ADPS2)|(0<<ADPS1)|(1<<ADPS0));
        ADMUX |= (1<<REFS0);
}
```
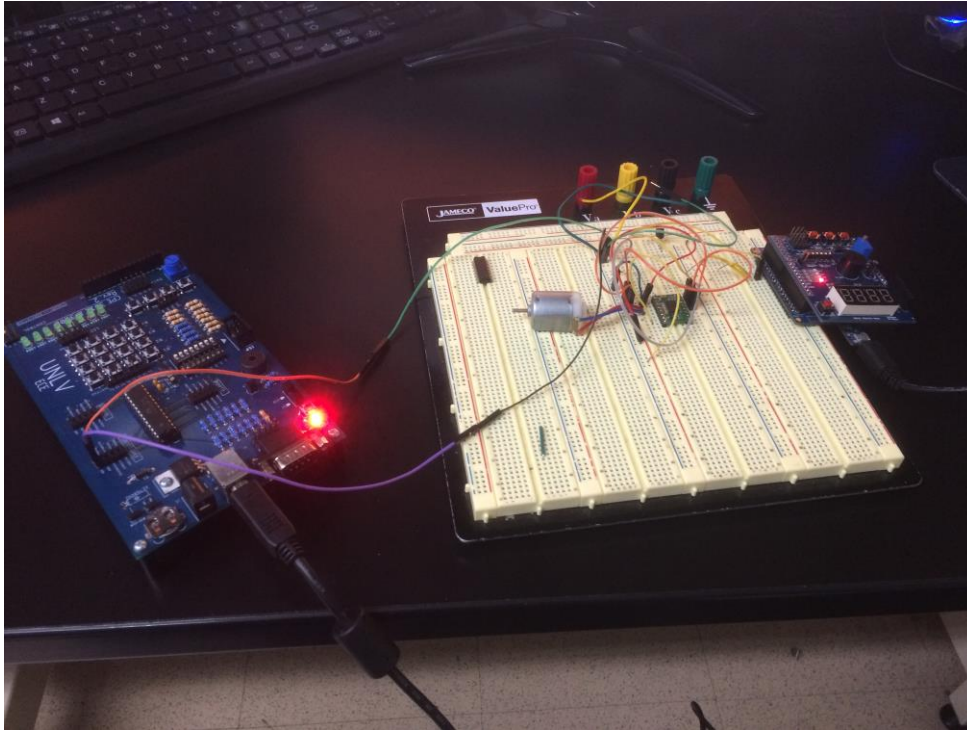
3.   DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A

Same as above

4.   SCHEMATICS

5.   SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)

6.   SCREENSHOT OF EACH DEMO (BOARD SETUP)
Task 1

7.   VIDEO LINKS OF EACH DEMO
https://youtu.be/0_1ULBimppg

8.   GITHUB LINK OF THIS DA

https://github.com/chicosisco/da_sub.git

Student Academic Misconduct Policy
http://studentconduct.unlv.edu/misconduct/policy.html

"*This assignment submission is my own, original work*".
Francisco Mata Carlos