CPE301 – SPRING 2019

Design Assignment 4B

Student Name: Mat Tomljenovic
Student #: 1012254710
Student Email: tomljeno@unlv.nevada.edu
Primary Github address: https://github.com/matcroatia/
Directory: https://github.com/matcroatia/submission_da

1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

Atmega328, Dual Motor Driver TBN6612FNG, several wires, bread board and small 5V motor.

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

```
/*
 * DA4B.c
 *
 * Created: 4/17/2019 5:02:52 PM
 * Author : Mat Tomljenovic
 */

#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>

void adc_funct(void);
int motor_speed;
int motor_stop = 0;



int main()
{
	DDRD = 0xFB;		//make port D as output
	EIMSK = 0x01;		//enable INT0
	EIFR = 0x01;		//enable interrupt flag
	EICRA = 0x03;		//set interrupt on rising edge
	TCCR0A= 0x83;		//set timer 0 for fast PWM and clear OCR0A on clear and
compare match
	TCCR0B= 0x05;		//set pre-scaler to 1024
	sei();			//enable global interrupts
	adc_funct();		//function declaration

	while (1)
```

```
	{
		while((ADCSRA &( 1<<ADIF )) == 0);
		motor_speed = ADC*190/800;		//controls speed of the motor
		OCR0A = motor_speed;			//speed is set to change on clear
and compare match
		if(motor_stop == 0)
		{
			PORTD = 0x01;				//motor will rotate
clockwise
		}
		else
		{
			PORTD = 0x00;				//motor will rotate counter
clockwise
		}

	}
}
ISR(INT0_vect)
{
	motor_stop ^= 1;				//toggles motor to stop
}

void adc_funct(void)
{
	ADMUX = 0x40;				//used to enable ADC
	ADCSRA = 0xE7;				//used to start the ADC conversion
process
}
```

3. DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A

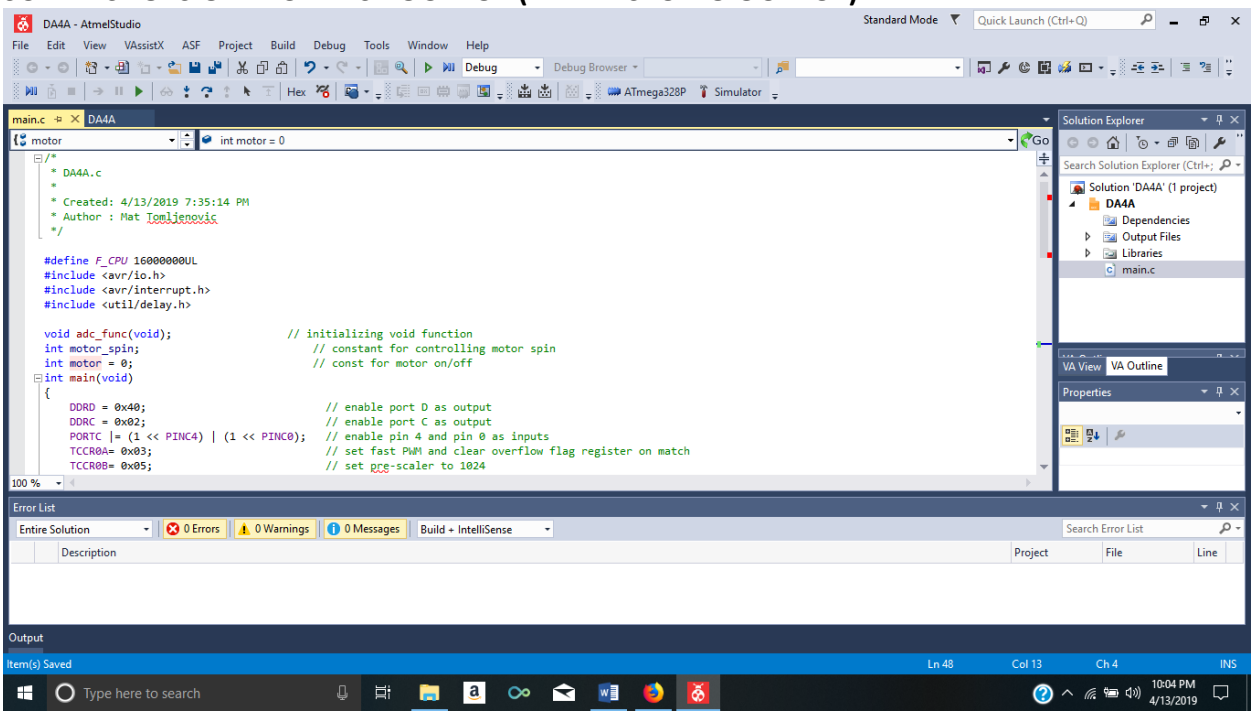
```
/*
 * DA4B2.c
 *
 * Created: 4/18/2019 5:40:39 PM
 * Author : Mat Tomljenovic
 */

#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>

void adc_funct(void);
int motor_rotate;

int main(void)
{
	DDRD = 0xFF;		//enable port D as output
	TCCR0A = 0x83;			//set timer 0 for fast PWM and clear OCR0A on
clear and compare match
	TCCR0B = 0x05;			//set pre-scaler to 1024
	adc_funct();		//function declaration
	while (1)
	{
		while((ADCSRA & ( 1<<ADIF )) == 0); //while motor is on
```

```
                motor_rotate = (ADC /33 +1);                //rotate motor 180 degrees
                OCR0A = motor_rotate;                               //set rotate on clear and
match register

        }
}

void adc_funct(void)
{
        ADMUX = ADMUX = 0x40;                                       //used to enable ADC;

        ADCSRA = 0xEF;
}
```

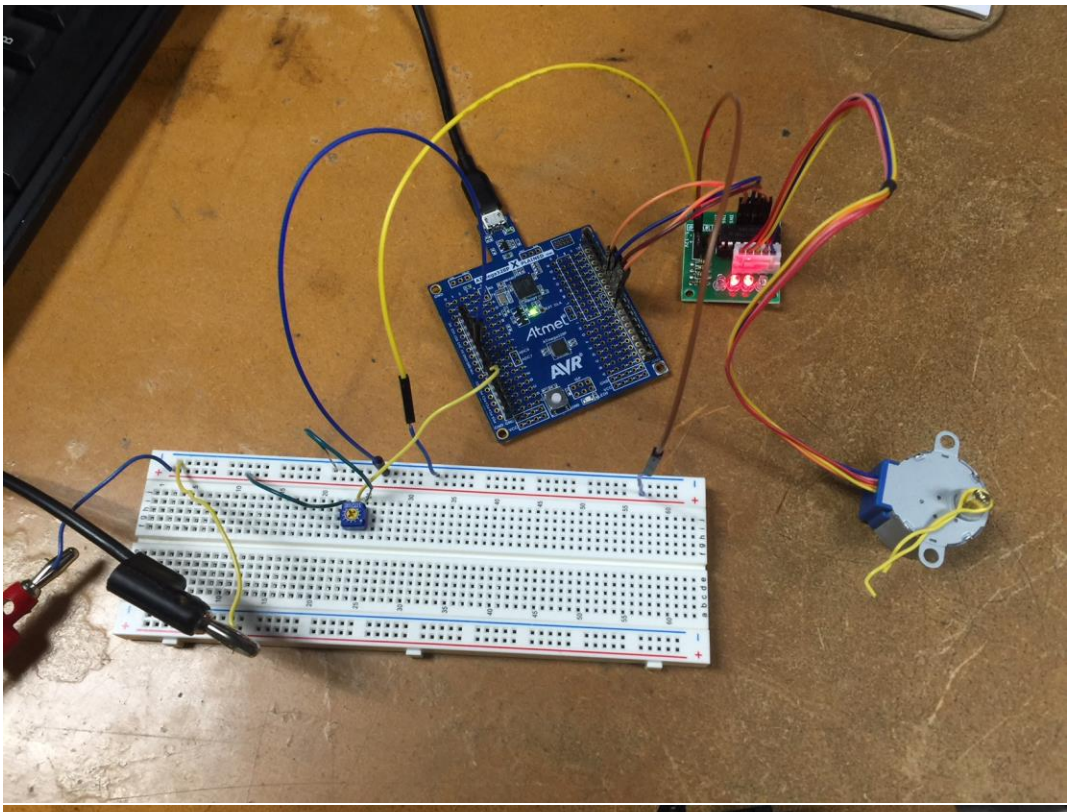
4. SCHEMATICS

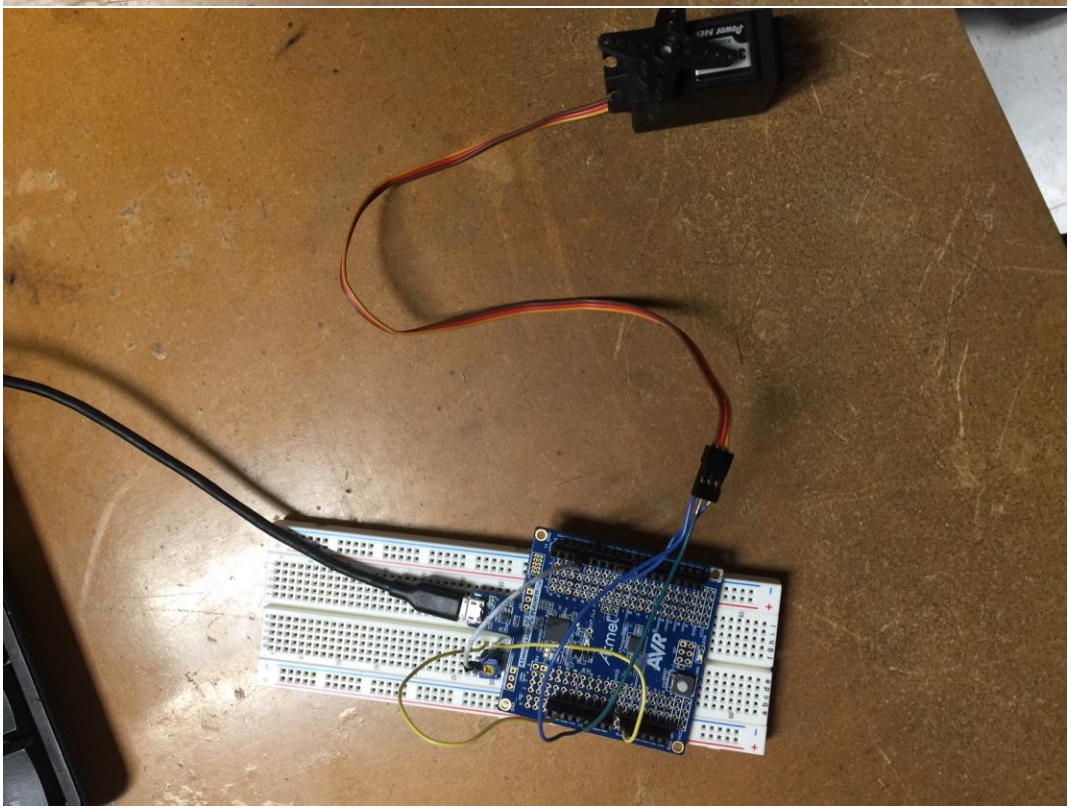
Use fritzing.org

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



6. SCREENSHOT OF EACH DEMO (BOARD SETUP)


Atmel
AVR


Atmel
AVR

7. VIDEO LINKS OF EACH DEMO

https://youtu.be/yY70O9BluZs
https://youtu.be/sp5UeuEem9w

8. GITHUB LINK OF THIS DA

https://github.com/matcroatia/submission_da

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

"*This assignment submission is my own, original work*".
Mat Tomljenovic