CPE301 – SPRING 2019

Design Assignment 4A

Student Name: Kyungseo Yun

Student #: 2001091216

Student Email: yunk93@unlv.nevada.edu

Primary Github address: <https://github.com/biscuit0x/submission_yun.git>

Directory: submission\_yun/DesignAssignments/DA4A/

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 used

atmega328p, TB6612FNG, DC motor, multi-shield

2. C code

#define F\_CPU 16000000UL

#include <avr/io.h>

#include <avr/interrupt.h>

#include <util/delay.h>

volatile int val, power = 0; //val value and power on/off state

void init\_pwm(void) {

DDRD |= (1<<PORTD1);

DDRC |= (1<<PORTC1);

TCCR0A |=

(1<<WGM01)|

(1<<WGM00)|

(1<<COM0A1);

PCMSK1 |= (1<<PCINT9);

PCICR |= (1<<PCIE1);

TCCR0B |=

(1<<CS02)| // Prescaler = 1024

(1<<CS00);

}

void init\_adc(void) {

DDRC &= (0<<PORTC0); //adc initialization

PORTC |= (1<<PORTC1);

ADMUX |= (1<<REFS0);

ADCSRA |=

(1<<ADEN)|

(1<<ADPS2)|

(1<<ADPS1)|

(1<<ADPS0); // prescaler = 128

sei();

}

int main(void)

{

init\_adc(); // initialization

init\_pwm();

while (1)

{

if (power == 1) {

ADCSRA |= (1<<ADSC);

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

ADCSRA |= (1<<ADIF); //reset

val = ADC; // store ADC to val for debugging

OCR0A = (224/1024)\*val; //max speed = 95% of PMW

}

else

OCR0A = 0;

}

}

//PC1 = power button

ISR(PCINT1\_vect) {

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

\_delay\_ms(100);

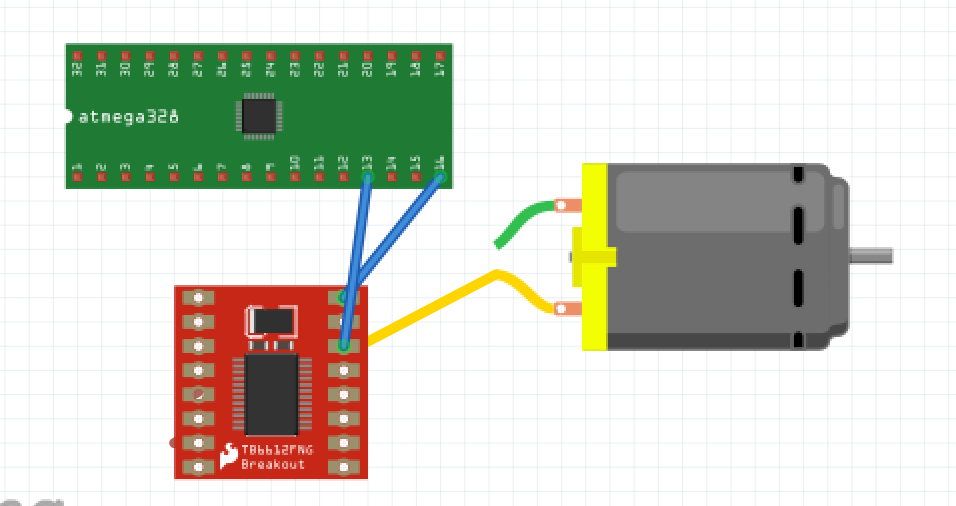
while(!(PINC&(1<<PINC1)));

power ^= 1; //toggle power

}

}

3.



PD1 = output

PC1 = interrupt