CPE301 – SPRING 2019

Design Assignment 4A

Student Name: Ron Joshua Recrio
Student #: 5003825419
Student Email: recrio@unlv.nevada.edu
Primary Github address: https://github.com/recrio/submissions
Directory: /DesignAssignments/DA4A

Submit the following for all Labs:

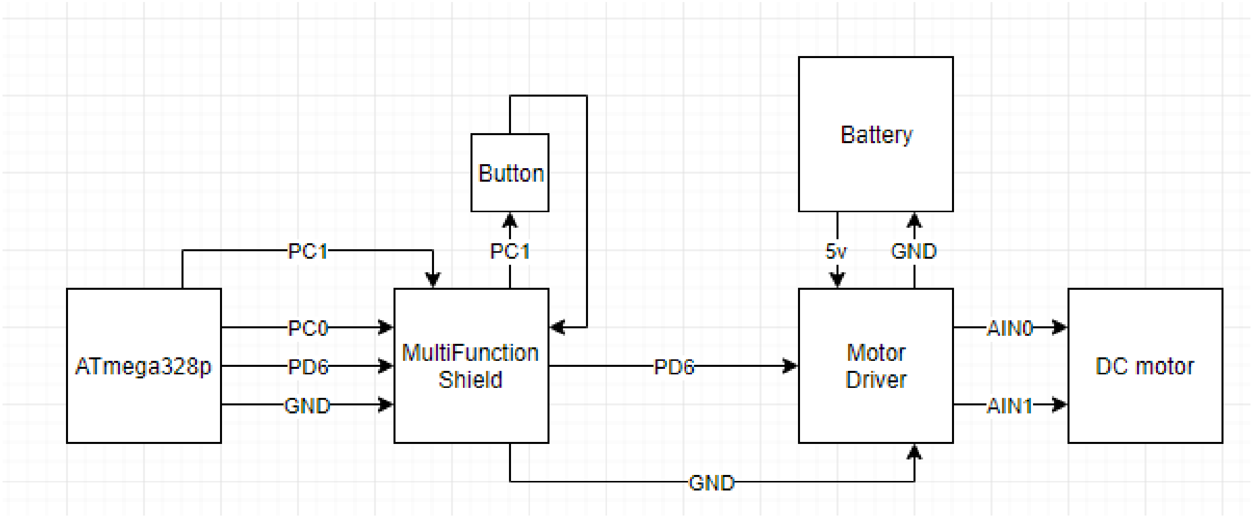
1. In the document, for each task submit the modified or included code (only) with highlights and justifications of the modifications. Also, include the comments.
2. Use the previously create a Github repository with a random name (no CPE/301, Lastname, Firstname). Place all labs under the root folder 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 LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

List of Components used:

ATmega328p Xplained Mini
Multifunction Shield
TB6612FNG Dual Motor Driver Carrier
DC motor

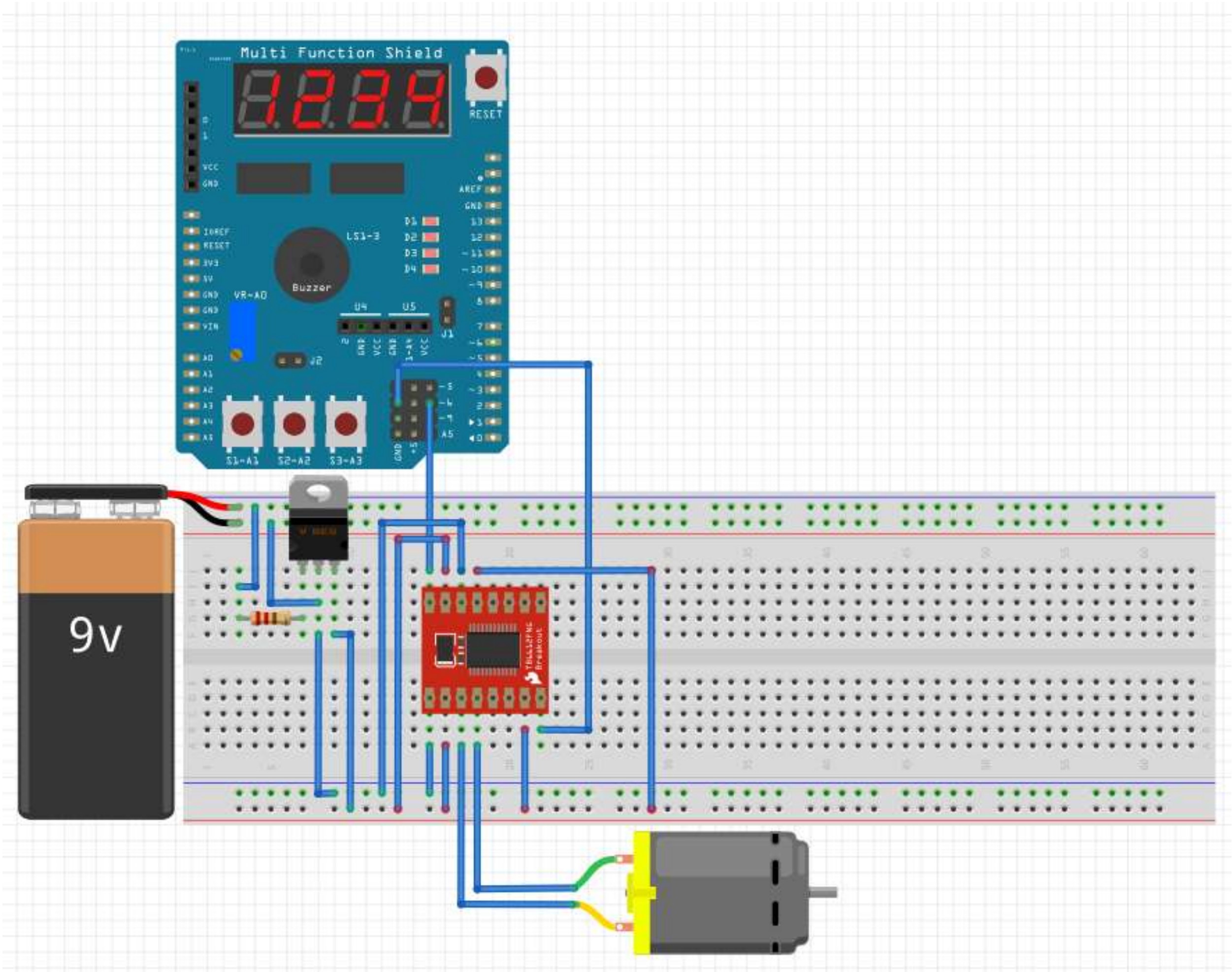
Block diagram with pins used in the Atmega328P



2. INITIAL CODE OF TASK 1/A

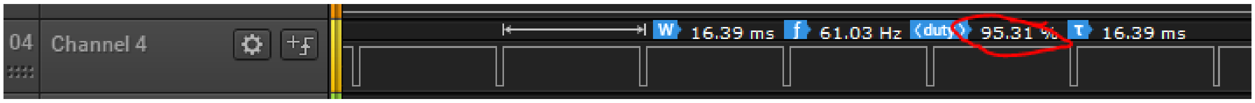
```
#define F_CPU 16000000UL
#define MAXSPEED 244
#define CONVERSION .238 // 244/1024

#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>


void PWM_init(void);
void ADC_init(void);

volatile int ADCvalue; // holds the value of ADC
volatile int motor = 0; // State of motor 0 = off, 1 = on
int main(void)
{
    ADC_init(); // initializes ADC
       PWM_init(); // initiales PWM
    while (1)
    {
              if (motor == 1) {
                     ADCSRA |= (1<<ADSC); // Start conversion
                     while((ADCSRA&(1<<ADIF))==0); // while not done converting do
nothing
                     ADCSRA |= (1<<ADIF); // reset converter
                     ADCvalue = ADC; // store ADC to ADCvalue for debugging
                     OCR0A = CONVERSION*ADCvalue; // 224/1024 is the ratio of the speed
so that max speed is 95% of the pwm
              }
              if (motor == 0) {
                     OCR0A = 0;
              }
    }
}

ISR(PCINT1_vect) {
       if(!(PINC & (1 << PINC1))) { // if Button pressed
              _delay_ms(100); // debouncing
              while(!(PINC&(1<<PINC1))); // extra debouncing
              motor ^= 1; // Toggle motor
       }
}

void PWM_init(void) {
       DDRD |= (1<<PORTD6); // PD6 as output
       DDRC |= (1<<PORTC1); // PC1 as output for interrupt
       TCCR0A |=
       (1<<WGM01)| // Fast
       (1<<WGM00)| // PWM
       (1<<COM0A1);// Clear OC0A on compare/set OC0A as bottom (non-inverting)
       PCMSK1 |= (1<<PCINT9); // Enable PCINT9
       PCICR |= (1<<PCIE1); // Enable PCMSK1
       TCCR0B |=
       (1<<CS02)| // Prescaler
       (1<<CS00); // 1024 and start pwm
}

void ADC_init(void) {
       DDRC &= (0<<PORTC0); // PC0 as input
       PORTC |= (1<<PORTC1); // Pull up resistor
       ADMUX |= (1<<REFS0); // REFERENCE VOLTAGE AT AREF
       ADCSRA |=
       (1<<ADEN)| // ADC enable
       (1<<ADPS2)| // ADC
       (1<<ADPS1)| // Prescaler
       (1<<ADPS0); // 128
       sei();
}
```

3. SCHEMATICS



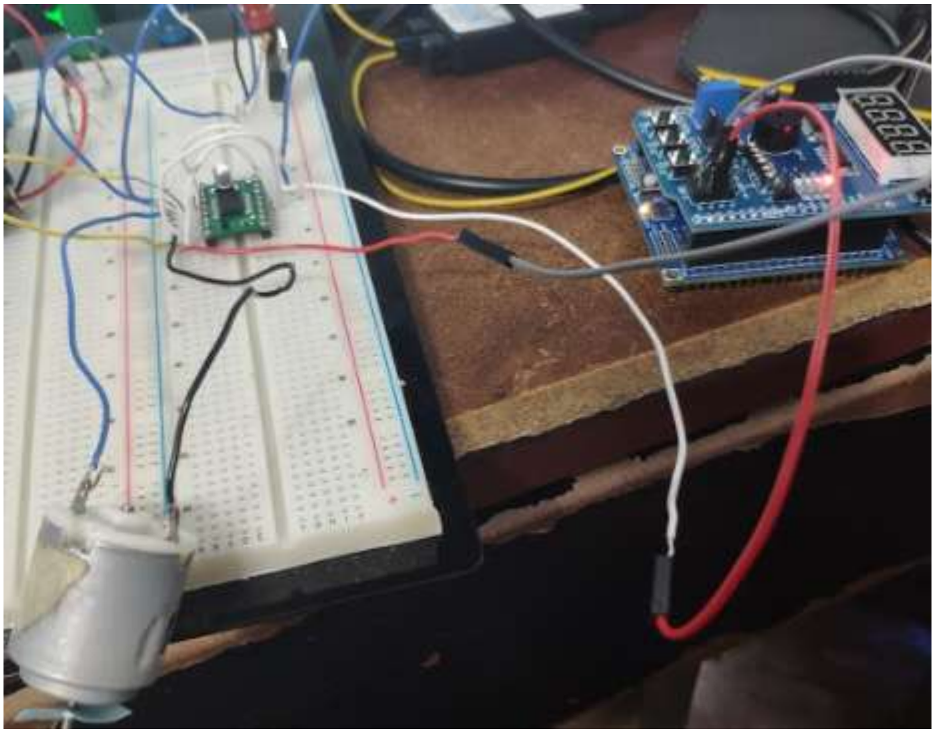
4. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)

Logic Analyzer showing 95% duty cycle at max speed:



W 16.39 ms f 61.03 Hz duty 95.31 % T 16.39 ms

5. SCREENSHOT OF EACH DEMO (BOARD SETUP)



6. VIDEO LINKS OF EACH DEMO

https://youtu.be/PS56JLzTxOk

7. GITHUB LINK OF THIS DA

https://github.com/recrio/submissions/tree/master/DesignAssignments/DA4A

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

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