CPE301 – SPRING 2019

Design Assignment DA4A

Student Name: Rimon Sawa

Student #: 5001529281

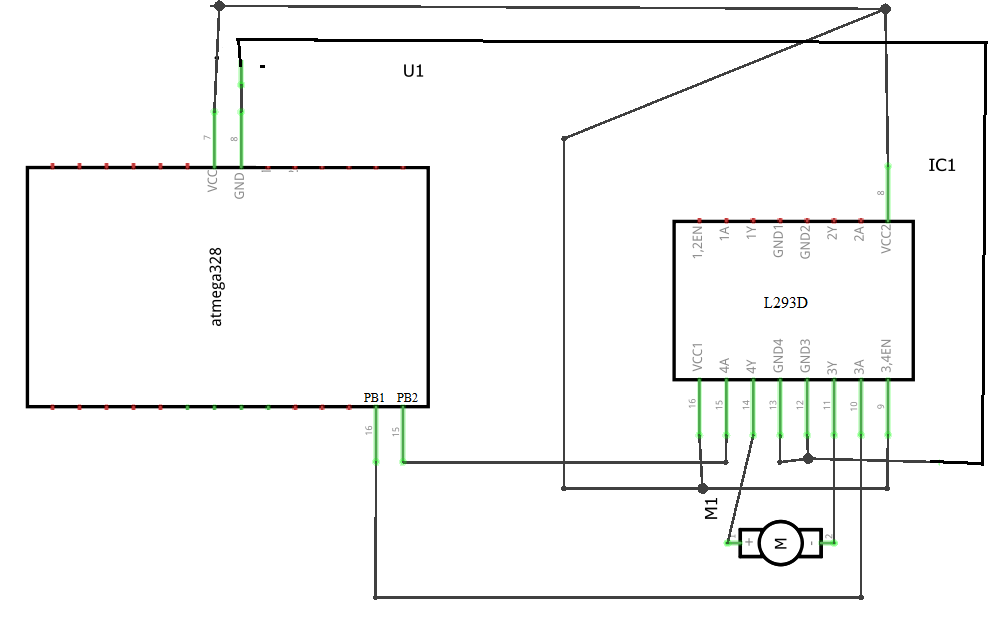
Student Email: sawar1@unlv.nevada.edu

Primary Github address: <https://github.com/sawar1/UNLV301>

Directory: DA4A

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

DC motor. The DC motor will be connected to the l293D chip.



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

/\*

\* GccDA4A.c

\*

\* Created: 4/9/2019 7:38:46 PM

\* Author : Home

\*/

#include <avr/io.h>

#define *F\_CPU* 8000000UL

#include <util/delay.h>

#include <avr/interrupt.h>

// Variable Declaration

volatile unsigned int adcVal; // holds value of ADC

volatile unsigned int pressed;

int main(void){

// port initialization

DDRB = (1<<1)|(1<<2); // PB.1-2 (OC1A & OC1B) as output to generate PWM

DDRC = 0; // PORTC as input

DDRD |= 0xFF; //PD2 Input

//PORTD |= (1 << 2);

// timer 1 initialization - generate 50Hz PWM

TCCR1A |= (1<<COM1A1)|(1<<COM1B1)|(1<<WGM11); // enable PWM for OC1A & OC1B

// Fast PWM, Non-inverted mode

TCCR1B |= (1<<WGM13)|(1<<WGM12)|(1<<CS10); // 8 prescaling

ICR1 = 65535; // top value for timer1

OCR0A = 122; //OCR0A is set compare register to 128

TCCR0B=(1 << CS02) | (1 << CS00); //TCCR0B sets prescaler to None

TCCR0A=0x83; //TCCR0A sets WGM00 and WGM01 to 1 which sets Fast PWM as well as COM0A1 and COM0B1 to 1 which clears OCR0A when compare match.

TIMSK0 |= (1 << TOIE0);

// initialize ADC

DDRC = 0; // Set PORTC as input for adc

DIDR0 = 0x1; // Disable digital input on ADC0 pin

ADMUX = 0; // ADC0 (PC.0) used as analog input

ADMUX |= (1 << REFS0); // use AVcc as the reference

ADMUX |= (1 << ADLAR); // Right adjust for 8 bit resolution

ADCSRA = 0x87; // Enable ADC, system clock, 10000111

ADCSRB = 0x0; // Free running mode

EIMSK = 1<<INT0; // Enable INT0

EICRA = 1<<ISC01 | 1<<ISC00; // Trigger INT0 on rising edge

sei();

while (1){

ADCSRA |= (1 << ADSC); // start conversion

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

adcVal = ADCH; // extract right 8-bits of ADC register

if(pressed) {

OCR1A = 257\*adcVal; // OCR1A value for duty cycle

}

if(!pressed) {

OCR1A = 0; // OCR1A value for duty cycle

}

}

return 0;

}

ISR(TIMER0\_OVF\_vect) {

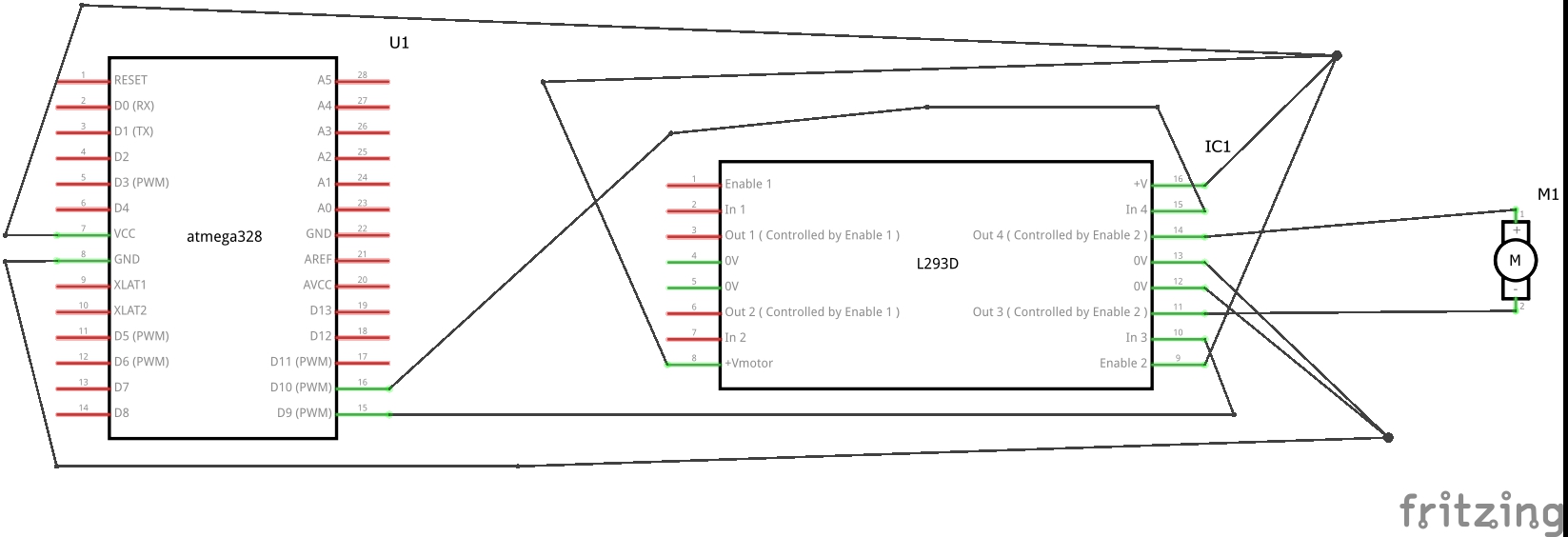
if((PIND & (1<< PIND0))!=0){

pressed = !pressed;

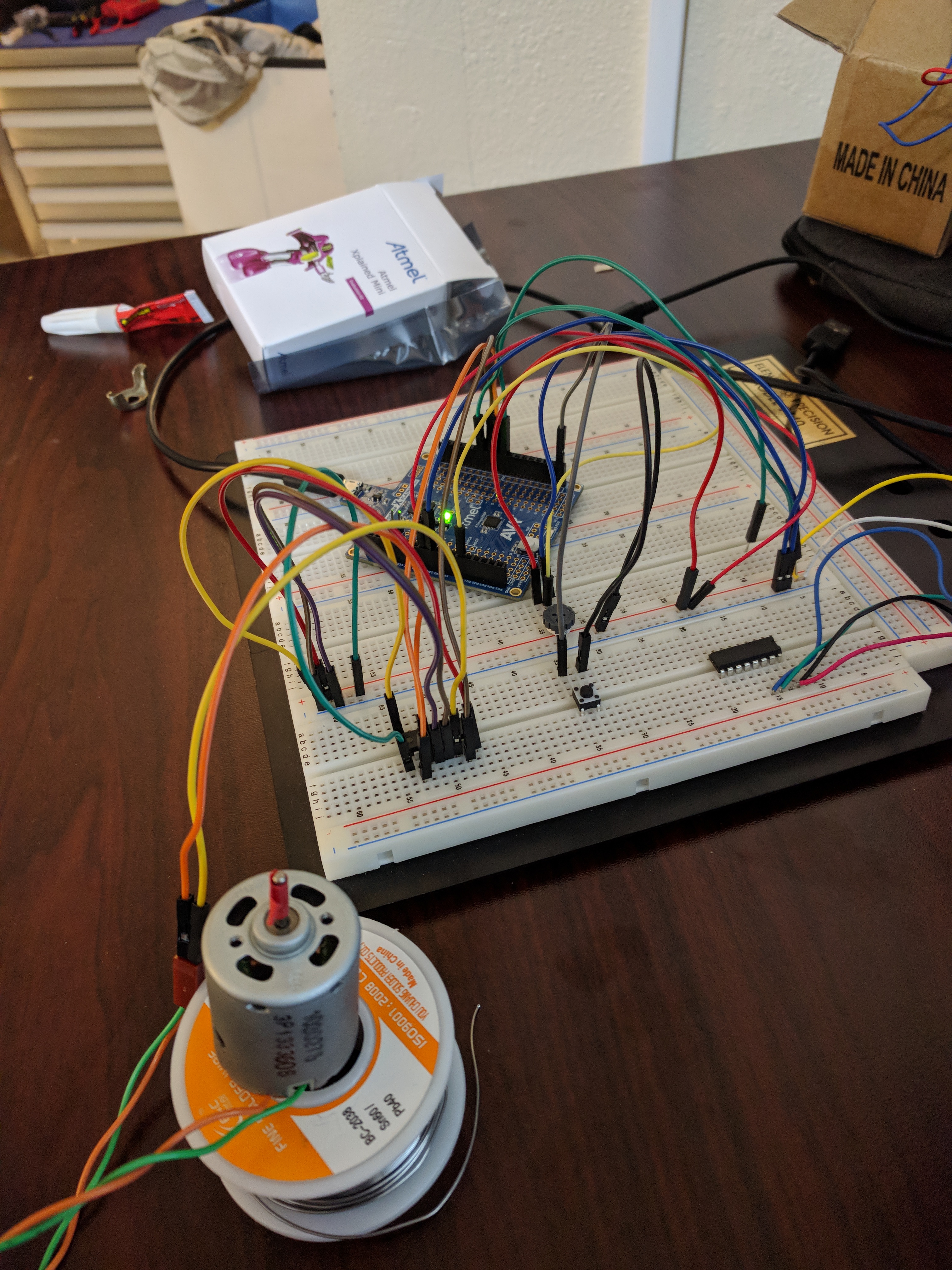
}

}

1. **Schematic**



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



1. **VIDEO LINKS OF EACH DEMO**

<https://youtu.be/GHoU_qti7EI>

1. **GITHUB LINK OF THIS DA**

<https://github.com/sawar1/UNLV301>

**Student Academic Misconduct Policy**

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

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

RIMON SAWA