CPE301 – FALL 2019

Design Assignment 4A

Student Name: Cameron Kirk

Student #: 2000660135

Student Email: kirkc1@unlv.nevada.edu

Primary Github address: <https://github.com/kirkster96/submission_da>

Directory: <https://github.com/kirkster96/submission_da/tree/master/DesignAssignment/DA4_a>

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

Atmega328PB

Hilelgo Multi-Function Shield Module for Arduino

Pololu TB6612FNG

DC motor

No pins were used for this solution

1. **AVR C DEVELOPED CODE OF TASK 1**

#define *F\_CPU* 16000000UL

#define ENABLE 3

#define MTR\_1 1

#define MTR\_2 2

#define SW (PINC&(1<<1))

#define BAUD\_RATE 9600

#define BAUD\_PRESCALE (((*F\_CPU* / (BAUD\_RATE \* 16UL))) - 1)

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

#include <stdlib.h>

#include <stdio.h>

//UART for debug purposes on terminal

void usart\_init (void);

void usart\_send (char ch);

void UART\_sendString(const char \*str);

int main (void)

{

usart\_init();

//configure the PWM

DDRD |= (1<<DDD6); //PD6 is now a output

OCR0A = 127;//set PWM for 50% dutycycle;

TCCR0A |= (1<<COM0A1)|(1<<WGM01) |(1<<WGM00);

TCCR0B |=(1<<CS01);

//initialize the ADC

ADMUX = (0<<REFS1)| //Reference selection bits

(1<<REFS0)| //AVcc - external cap at AREF

(0<<ADLAR)| //Left adjust most significant bit

(0<<MUX2)| //Analog channel selection bits

(0<<MUX1)| //ADC0 (PC0 PIN23)

(0<<MUX0);

ADCSRA = (1<<ADEN)| // Enable ADC

(0<<ADSC)| //ADC start conversion

(0<<ADATE)| //ADC Autotrigger enable

(0<<ADIF)| //ADC Interrupt Flag

(0<<ADIE)| //ADC Interrupt Enable

(1<<ADPS2)| //ADC prescaler bits

(0<<ADPS1)|

(1<<ADPS0);

DDRB |= (1<<DDB1)|(1<<DDB2)|(1<<DDB3);

//PB1,PB2, and PB3 is now an output

PORTC |= (1<<0)|(1<<1)|(1<<2)|(1<<3);

//Enable PC1 as input

PORTB &= ~(1<<ENABLE);

PORTB &= ~(1<<MTR\_1);

PORTB &= ~(1<<MTR\_2);

PCMSK1 |= (1<<PCINT9)| //enable PC1 as External Interrupt

(1<<PCINT10)| //enable PC2 as External Interrupt

(1<<PCINT11); //enable PC3 as External Interrupt

//enable interrupts

PCICR |= (1<<PCIE1);

sei();

//configure to trigger on falling edge

EICRA |= (1<<ISC11);

EICRA &= ~(1<<ISC10);

PORTB |= (1<<ENABLE);

volatile unsigned char TEMP [4];

while(1){

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

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

ADCSRA |= (1<<ADIF);

//output adc results to terminal

int c = ADCL;

c = c | (ADCH<<8);

c = (c/1024.0)\*5000/10;

*itoa*(c,TEMP,10);

UART\_sendString(TEMP);

UART\_sendString("\r\n");

//c<11 -> Duty Cycle 0%

//c>490 -> Duty Cycle 95%

OCR0A = c/2;

*\_delay\_ms*(20);

PORTB &= ~(1<<MTR\_2);

PORTB |= (1<<MTR\_1);

}

}

ISR (PCINT1\_vect){

PORTB ^= (1<<ENABLE);

*\_delay\_ms*(200);

//double click button to toggle

}

void usart\_init(void){

//turn on Rx, TX

UCSR0B |= (1<<RXEN0)|(1<<TXEN0);

//8bit char sizes

UCSR0C |= (1<<UCSZ01)|(1<<UCSZ00);

//set baud rate

UBRR0H = (BAUD\_PRESCALE >> 8);

UBRR0L = BAUD\_PRESCALE;

//Enable the USART Receive interrupt

UCSR0B |= (1<<RXCIE0);

//globally enable interrupts

//sei();

}

void usart\_send(char ch){

while(! (UCSR0A & (1<<UDRE0))); //wait until UDR0 is empty

UDR0 = ch; //transmit ch

}

void UART\_sendString(const char \*str){

while(\*str){

usart\_send(\*str++);

}

}

1. **VIDEO LINKS OF EACH DEMO**

AVR C Task 1 Demo

<https://drive.google.com/file/d/1Fz7bXZy9ctsbqd9-kqUmJKVTWVpLv8Kr/view?usp=sharing>

1. **GITHUB LINK OF THIS DA**

<https://github.com/kirkster96/submission_da/tree/master/DesignAssignment/DA4_a>

**Student Academic Misconduct Policy**

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

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

Cameron Kirk