**CPE301 – SPRING 2019**

Design Assignment 3B

Student Name: Prachi Patel

Student #: 500280222

Student Email: patelp3@unlv.nevada.edu

Primary Github address: <https://github.com/prachi173/>

Directory: <https://github.com/prachi173/da_sp18/>

Submit the following for all Labs:

* In the document, for each task submit the modified or included code (only) with highlights and justifications of the modifications. Also, include the comments.
* 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.
* If multiple asm/c files or other libraries are used, create a folder LabXX-TYY and place these files inside the folder.
* 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)
* **COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS**

ATmega328p Board given in CPE310 LAB

LM35 Temperature Sensor

FTDI BASIC module for USART

AVR Programmer



* **INITIAL/MODIFIED/DEVELOPED CODE OF TASK 3/A**

/\*

\* DA3A.c

\*

\* Created: 3/27/2019 9:37:20 PM

\* Author : patel

\*/

#include <avr/io.h>

#include <util/delay.h>

#include <stdio.h>

#include <avr/interrupt.h>

#define *F\_CPU* 8000000UL

#define BAUDRATE 9600

#define BAUD\_PRESCALLER (((*F\_CPU* / (BAUDRATE \* 16UL)))-1)

void USART\_init(unsigned int ubrr);

void USART\_tx\_string(char \*data);

volatile int OVFCount; //declare global overflow counter

char outs[80]; //char limit

int x; //integer

float n; //floating number

char str[] = "hello world!"; //string

char empty[] = " "; //empty space

int main(void)

{

OVFCount = 0; //setup CTC timer

OCR0A = 240;

TCCR0A |= (1<<WGM01);

DDRC = 0X00;

PORTC = 0XFF;

DDRB = 0xFF;

TIMSK0 |= (1<<OCIE2A);

TCCR0B |= (1<<CS00) | (1<<CS02);

sei(); //interrupt enable

USART\_init(BAUD\_PRESCALLER); //initialize the USART

*\_delay\_ms*(250);

USART\_tx\_string("\r\nConnected!\r\n"); //we're alive!

while (1)

{

while(OVFCount >= 33){ //when 1 second

USART\_tx\_string(str); //print string

USART\_tx\_string(empty); //print empty

x = *rand*(); //genrate random integer

*snprintf*(outs,sizeof(outs),"%2d\r\n",x); //printing number

USART\_tx\_string(outs);

USART\_tx\_string(empty);

n = 1.1234; //the floating number

*dtostrf*(n, 4, 4, outs); //assign decimal limits

USART\_tx\_string(outs); //print it

USART\_tx\_string(empty);

OVFCount = 0; //reset counter

}

}

}

void USART\_init(unsigned int ubrr){

//initializing USART RS-232 using the given code from professor

UBRR0H = (unsigned char)(ubrr>>8);

UBRR0L = (unsigned char)ubrr;

UCSR0B = (1 << TXEN0);

UCSR0C = (3 << UCSZ00);

}

void USART\_tx\_string(char \*data){

//send a string to the RS-232

while ((\*data != '\0')){

while(!(UCSR0A & (1 << UDRE0)));

UDR0 = \*data;

data++;

}

}

ISR(TIMER0\_COMPA\_vect) //external timer initializing

{

OVFCount++;

}

* **DEVELOPED MODIFIED CODE OF TASK 3/B from TASK 3/A**

/\*

\* DA3B.c

\*

\* Created: 3/28/2019 9:50:20 PM

\* Author : patel

\*/

#ifndef *F\_CPU*

#define *F\_CPU* 8000000UL

#endif

#ifndef BAUD

#define BAUD 9600

#endif

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

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

#include <stdio.h>

//defining ADC0 as 0 to initialize

//using PC0 to read in temperature

#define ADC0 0x00

#define ADC1 0x01

#define AVCC 5000

volatile int OVFCount; //declare global overflow counter

//initializing and calling functions

void USART\_init(void);

void USART\_send(*uint8\_t* ch);

void USART\_print(*uint8\_t*\* str);

*uint8\_t* USART\_Receive(void);

void ADC\_init(void);

*uint16\_t* read\_adc(*uint8\_t* adc\_temp);

void main(void)

{

//in main function, the data from read\_adc function will be received and converted into readable data. Also initialize Timer0

OVFCount = 0; //setup CTC timer

OCR0A = 240;

TCCR0A |= (1<<WGM01);

TIMSK0 |= (1<<OCIE2A);

TCCR0B |= (1<<CS00) | (1<<CS02);

sei(); //interrupt enable

*uint16\_t* temp; //define unsigned 16 bit value for voltage reading from adc

*uint8\_t* t1, t2, t3; //define unsigned 8bit value for three digits

USART\_init(); //initialize the USART

ADC\_init(); //setup analog to digital converter

*\_delay\_ms*(500);

while(1){

while(OVFCount >= 33){

temp = read\_adc(ADC0); //set temp as ADC value read from read\_adc function

temp = (((5000/1024)\*temp)/10) + 2; //convert temp using prescaler of 1024

if(temp >= 100){

t1 = temp / 100;

t2 = temp % 100;

t3 = t2 % 10;

t2 = t2 / 10;

}

else{

t1 = 0;

t2 = temp / 10;

t3 = temp % 10;

}

//transfer from ASCII to decimal value and send to print

USART\_send(t1+48);

USART\_send(t2+48);

USART\_send(t3+48);

USART\_send('C');

USART\_send('\n');

USART\_send('\r');

OVFCount = 0;

}

//delay for a second

}

}

void USART\_init(void){

//initialize the USART FTDI Basic

//Set Baud Rate and frame format 8N1

//enable receiver and transmitter

UBRR0H = (*uint8\_t*)(ubrr >> 8);

UBRR0L = (*uint8\_t*)ubrr;

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

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

}

void ADC\_init(void)

{

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

ADCSRA |= (1<<ADPS2) | (1<<ADPS1) | (1<<ADPS0) | (1<<ADEN);

//ADC prescaler and ADC enable

}

*uint16\_t* read\_adc(*uint8\_t* adc\_temp){

ADMUX = (ADMUX & 0xF0) | (adc\_temp & 0x0F);

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

while(ADCSRA & (1<<ADSC)); //wait for conversion to finish

return ADC; //return the value in ADC to where the function is called (in temp)

}

void USART\_send(*uint8\_t* ch)

{

//wait until UDR0 is empty and then transmit ch

while(!(UCSR0A & (1<<UDRE0)));

UDR0 = ch;

}

*uint8\_t* USART\_Receive(void){

//wait until receiver is not busy then receive UDR0

while(!(UCSR0A & (1<<RXC0)));

return UDR0;

}

void USART\_print(*uint8\_t*\* str){

//receive string str and then send to USART\_send function to print

while(\*str){

USART\_send(\*str);

str++;

}

}

ISR(TIMER0\_COMPA\_vect) //external timer initializing

{

OVFCount++;

}

* **SCHEMATICS**

Use fritzing.org



* **SCREENSHOTS OF THE TASK OUTPUT (ATMEL STUDIO OUTPUT)**



* **SCREENSHOT OF THE DEMO (BOARD SETUP)**

A picture containing indoor, table, floor

Description automatically generated

* **VIDEO LINKS OF THE DEMO**

<https://youtu.be/53eyFBhClHM>

* **GITHUB LINK OF THIS DA**

<https://github.com/prachi173/da_sp18/tree/master/Design%20Assignments/DA3B>

**Student Academic Misconduct Policy**

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

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

PRACHI PATEL