CPE301 – Fall 2019

Design Assignment 3A

Student Name: Jacob Patrick Reed

Student #: 1008448895

Student Email: reedj35@unlv.nevada.edu

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

Directory: <https://github.com/reedjacobp/submission_da>

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**

ATmega328PB Xplained Mini

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

/\*

\* DA3A.c

\*

\* Created: 10/16/2019 9:43:39 PM

\* Author : jreed

\*/

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <avr/interrupt.h>

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#define BAUDRATE 9600

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

void USART\_init(void);

unsigned char USART\_receive(void);

void USART\_send(unsigned char data);

void USART\_send\_rand(int n);

void USART\_putstring(char\* StringPtr);

float pi = 3.14;

char String[] = "DA3A";

char newline[] = "\n";

char buffer[5];

char pi\_buffer[4];

*uint8\_t* OVF\_COUNT = 0;

*uint8\_t* OVF\_LIMIT = 250;

int main(void)

{

USART\_init();

TCCR0A = 0x00; //normal operation

TCCR0B |= (1 << CS02); //set prescalar to 256

TCNT0 = 16; //TOP = 256-250 = 16

TIMSK0 |= (1 << TOIE0);

sei();

while (1)

{

}

}

void USART\_init(void)

{

UBRR0H = (*uint8\_t*)(BAUD\_PRESCALLER >> 8);

UBRR0L = (*uint8\_t*)(BAUD\_PRESCALLER);

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

UCSR0C = (3 << UCSZ00);

}

unsigned char USART\_receive(void)

{

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

return UDR0;

}

void USART\_send(unsigned char data)

{

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

UDR0 = data;

}

void USART\_putstring(char\* StringPtr)

{

while(\*StringPtr != 0x00)

{

USART\_send(\*StringPtr);

StringPtr++;

}

}

// C program for implementation of ftoa()

#include<stdio.h>

#include<math.h>

// reverses a string 'str' of length 'len'

void reverse(char \*str, int len)

{

int i=0, j=len-1, temp;

while (i<j)

{

temp = str[i];

str[i] = str[j];

str[j] = temp;

i++; j--;

}

}

// Converts a given integer x to string str[]. d is the number

// of digits required in output. If d is more than the number

// of digits in x, then 0s are added at the beginning.

int intToStr(int x, char str[], int d)

{

int i = 0;

while (x)

{

str[i++] = (x%10) + '0';

x = x/10;

}

// If number of digits required is more, then

// add 0s at the beginning

while (i < d)

str[i++] = '0';

reverse(str, i);

str[i] = '\0';

return i;

}

// Converts a floating point number to string.

void ftoa(float n, char \*res, int afterpoint)

{

// Extract integer part

int ipart = (int)n;

// Extract floating part

float fpart = n - (float)ipart;

// convert integer part to string

int i = intToStr(ipart, res, 0);

// check for display option after point

if (afterpoint != 0)

{

res[i] = '.'; // add dot

// Get the value of fraction part upto given no.

// of points after dot. The third parameter is needed

// to handle cases like 233.007

fpart = fpart \* *pow*(10, afterpoint);

intToStr((int)fpart, res + i + 1, afterpoint);

}

}

ISR (TIMER0\_OVF\_vect)

{

OVF\_COUNT++;

if (OVF\_COUNT == OVF\_LIMIT)

{

USART\_putstring(String);

USART\_putstring(newline);

*itoa*(*rand*(), buffer, 10);

USART\_putstring(buffer);

USART\_putstring(newline);

ftoa(pi, pi\_buffer, 2);

USART\_putstring(pi\_buffer);

USART\_putstring(newline);

USART\_putstring(newline);

OVF\_COUNT = 0;

}

TCNT0 = 16;

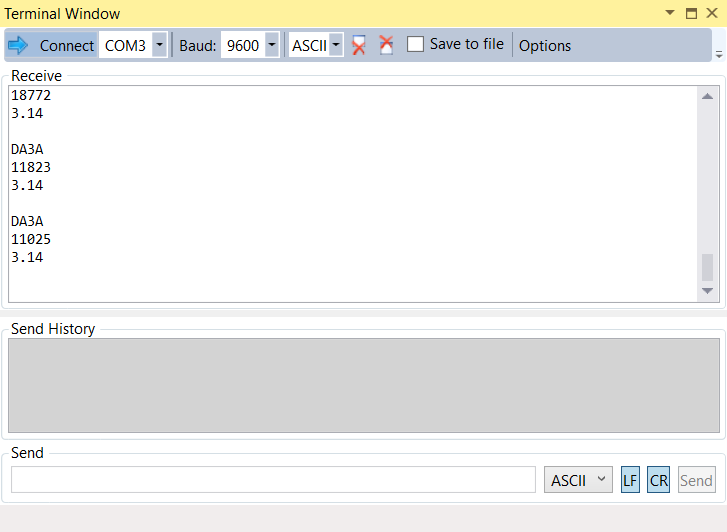
1. }**DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A**

N/A

1. **SCHEMATICS**

N/A

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



1. **SCREENSHOT OF EACH DEMO (BOARD SETUP)**

N/A

1. **VIDEO LINKS OF EACH DEMO**
2. **GITHUB LINK OF THIS DA**

**Student Academic Misconduct Policy**

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

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

NAME OF THE STUDENT