CPE301 – FALL 2019

Design Assignment 3A

Student Name: Andrew Buchanan

Student #:5003154346

Student Email: buchaa2@unlv.nevada.edu

Primary Github address: <https://github.com/buchaa2/103EPC>

Directory: https://github.com/buchaa2/103EPC/tree/master/DA3A

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

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <stdlib.h>

#include <math.h>

#include <avr/interrupt.h>

#include <stdio.h>

#define BAUDRATE 9600

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

void USART\_init(void); //initialize USART

void USART\_send(unsigned char data); //send through USART

unsigned char USART\_receive(void); //receive through USART

void USART\_putstring(char\* StringPtr); //scan through string

char String[] = "Hello\_World"; //string i chose

char newline[] = "\n"; //make output pretty

float nu = 9.87; //number i chose

char buffer[5]; //buffer for random int number

char nu\_buffer[4]; //buffer for float number

*uint8\_t* OVF\_COUNT = 0; //initialize the overflow count for interrupt

*uint8\_t* OVF\_LIMIT = 250; //set the over flow limit for 1 sec delay

int main(void)

{

USART\_init(); //initialize USART

TCCR0A = 0x00; //set normal operation

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

TCNT0 = 6;

TIMSK0 |= (1 << TOIE0);

sei(); //set interrupt

while (1)

{ //wait for interrupt

}

}

void USART\_init(void)

{

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

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

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

UCSR0C = (3 << UCSZ00);

}

void USART\_send(unsigned char data)

{

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

UDR0 = data;

}

unsigned char USART\_receive(void)

{

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

return UDR0;

}

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--;

}

}

void USART\_putstring(char\* StringPtr)

{

while(\*StringPtr != 0x00)

{

USART\_send(\*StringPtr);

StringPtr++;

}

}

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

{

int i = 0;

while (x)

{

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

x = x/10;

}

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 decimal

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

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

}

}

ISR (TIMER0\_OVF\_vect)

{

OVF\_COUNT++;

if (OVF\_COUNT == OVF\_LIMIT) //check to see if the limit equals count

{

USART\_putstring(String); //print string to the terminal window

USART\_putstring(newline); //go to next line

*itoa*(*rand*(), buffer, 10); //convert random int number (base 10) to a string

USART\_putstring(buffer); //print string

USART\_putstring(newline); //go to next line

ftoa(nu, nu\_buffer, 2); //convert float number to a string

USART\_putstring(nu\_buffer); //print string

USART\_putstring(newline); //go to next line

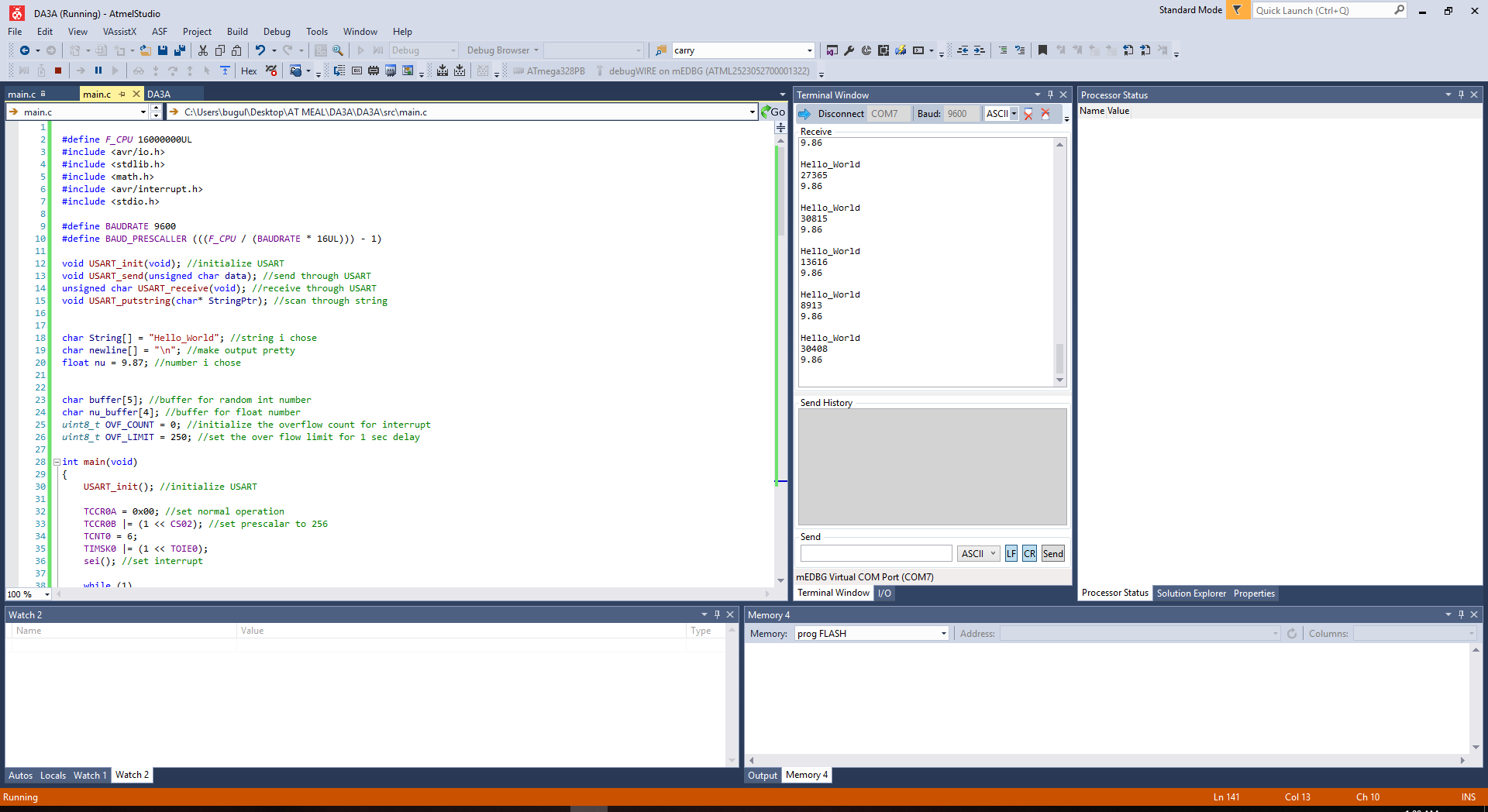
USART\_putstring(newline); //go to next line

OVF\_COUNT = 0; //reset overflow counter

}

TCNT0 = 6; //reset

}

1. **DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A**
2. **SCHEMATICS**
3. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)** 
4. **SCREENSHOT OF EACH DEMO (BOARD SETUP)**
5. **VIDEO LINKS OF EACH DEMO**

https://youtu.be/YyKzeDlmp-M

1. **GITHUB LINK OF THIS DA**

https://github.com/buchaa2/103EPC/tree/master/DA3A

**Student Academic Misconduct Policy**

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

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

Andrew Buchanan