CPE301 – SPRING 2019

Design Assignment 3A

Student Name: Rocky Yasuaki Gonzalez

Student #: 5003229733

Student Email: gonzar14@unlv.nevada.edu

Primary Github address: https://github.com/rockyg1995/ihswppdar.git

Directory: C:\Users\rocky\Documents\CpE 301+L - Embedded Systems Design\CpE

301\Repository\DesignAssignments\DA3A

Submit the following for all Labs:

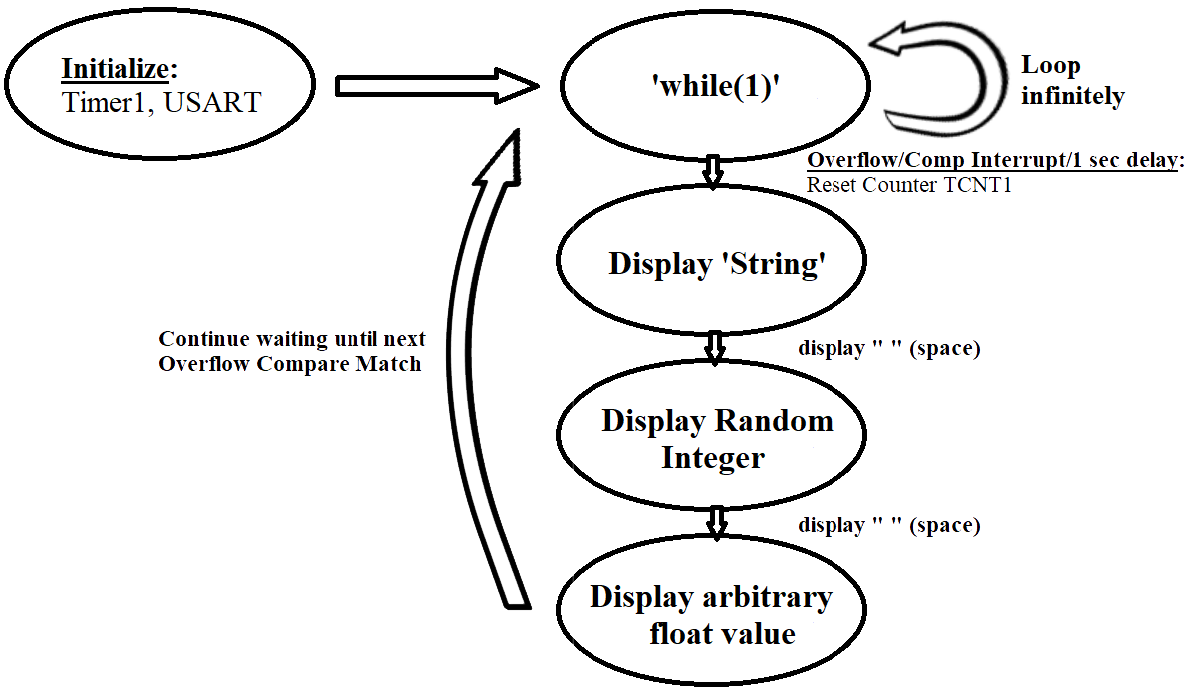
1. In the document, for each task submit the modified or included code (only) with highlights and justifications of the modifications. Also, include the comments.
2. Use the previously create a Github repository with a random name (no CPE/301, Lastname, Firstname). Place all labs under the root folder 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**

Atmega328PB Xplained Mini

Micro USB Cable (Power Supply)

Block Diagram N/A (Everything done through Atmega328PB Xplained Mini)



*Figure 1 – Flow Chart for Coding Algorithm*

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

/\*

\* DA3A.c

\*

\* Created: 3/25/2019 12:05:50 AM

\* Author: rocky

\*/

#define *F\_CPU* 16000000UL // Frequency of Xplained Mini (16MHz)

#include <avr/io.h> // Standard AVR Library

#include <stdio.h> // AVR library containing printf functions

#include <avr/interrupt.h> // AVR library containing interrupt functions

#include <util/delay.h> // AVR library containing \_delay\_ms() function

#define BAUDRATE 9600 // Baudrate in Bits per second (bps)

#define BAUD\_PRESCALLER ((*F\_CPU* / (BAUDRATE \* 16UL)) - 1) // Baudrate Prescaler UBRR0

#define MAX\_1s 15624 // 1s delay: OCR1A = (16MHz\*1000ms/prescaler) -1

//Declaration of our functions

void Timer1\_init(void); // Function to initialize Timer1

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

unsigned char USART\_receive(void); // Function to receive Serial data from UDR0

void USART\_send( unsigned char data); // Function to send individual char into UDR0

void USART\_putstring(char\* StringPtr); // Function to break string into chars and send

char String[] = "DA3A"; // 'String[]' is an array treated as a string when between " "

char outs[20]; // 'outs[]' to store integer/float values into array of chars

char t[10]; // 't[]' use

unsigned int rand\_int; // contains integer representing random value

float rand\_float; // contains float representing arbitrary random value

int main(void) {

TIMSK1 |= (1 << OCIE1A); // Set Interrupt on Compare Match

sei(); // Enable Global Interrupts

Timer1\_init(); // Call the Timer1 initialization code

USART\_init(); // Call the USART initialization code

while (1) { // Infinite loop

if (TCNT1 == OCR1A) { // Display a String when Timer1 Matches OCR1A

rand\_int = *rand*(); // Randomize Integer Value

rand\_float = 4.716; // Arbitrary Float Value chosen

USART\_putstring(String); // Pass 'String' to send serial of chars

USART\_send(' '); // Puts blank space into terminal monitor

*snprintf*(outs,sizeof(outs),"%3d", rand\_int); // Stores integer

// 'rand\_int' into the string 'outs'

USART\_putstring(outs); // Pass 'outs' to to send serial of chars

USART\_send(' '); // Puts blank space into terminal monitor

*dtostrf*(rand\_float, 3, 5, t); // Store float readable to

// 'snprintf'

*snprintf*(outs,sizeof(outs),"%s", t); // Stores 't' (float

// 'rand\_float') into the string 'outs'

USART\_putstring(outs); // Pass 'outs' to send serial of chars

USART\_send('\n'); // Puts new line into terminal monitor

}

}

return 0;

}

//-----------------------------------------------------------------------

void Timer1\_init(void) { // Function to Initialize Timer1 properties

OCR1A = MAX\_1s; // Let the OCR1A be the Max Value used for 1s delay

TCCR1A = (0<<COM1A1)|(0<<COM1A0); // Normal Operation, Disconnected OC1A COM1A1:0

TCCR1A = (0<<COM1B1)|(0<<COM1B0); // Normal Operation, Disconnected OC1B COM1B1:0

TCCR1A = (0<<WGM11)|(0<<WGM10); // Set Timer1 to CTC Mode 'WGM1:0'

TCCR1B = (0<<WGM13)|(1<<WGM12); // Set Timer1 to CTC Mode 'WGM3:2'

TCCR1B = (1<<CS12)|(0<<CS11)|(1<<CS10); // Set Clock Select for Prescaler '1024'

}

ISR(TIMER1\_COMPA\_vect) { // Function to Truncate Timer1 to delay properly for 1 second

TCNT1 = 0; // Restart Timer1 from the Beginning

return; // Resume code from where interrupt left off

}

//-----------------------------------------------------------------------

void USART\_init(void) { // Function to Initialize USART properties

UBRR0H = (*uint8\_t*)(BAUD\_PRESCALLER >> 8); // Store Upper Baudrate into UBRR0H

UBRR0L = (*uint8\_t*)(BAUD\_PRESCALLER); // Store Lower Baudrate into UBRR0L

UCSR0B = (1 << RXEN0) | (1 << TXEN0); // Enable Receiver and Enable Transmitter

UCSR0C = (3 << UCSZ00); // Set UCSZ02:1 as 8-bit character data

}

unsigned char USART\_receive(void) { // Function to receive ASCII value from UDR0

while (!(UCSR0A & (1 << RXC0))); // Keep Checking until RXC0 is 'High' to break

return UDR0; // Return received serial into unsigned char

}

void USART\_send(unsigned char data) { // Function to transmit ASCII value into UDR0

while (!(UCSR0A & (1 << UDRE0))); // Check until UDRE0 data register 'High'

UDR0 = data; // Store unsigned char serial data into UDR0

}

void USART\_putstring(char\* StringPtr) { // Function to break string into chars, and send

while (\*StringPtr != 0x00) { // Keep Looping until String Completed (null)

USART\_send(\*StringPtr); // Send char value pointed by the string pointer

StringPtr++; // Increment pointer to next char array location

}

}

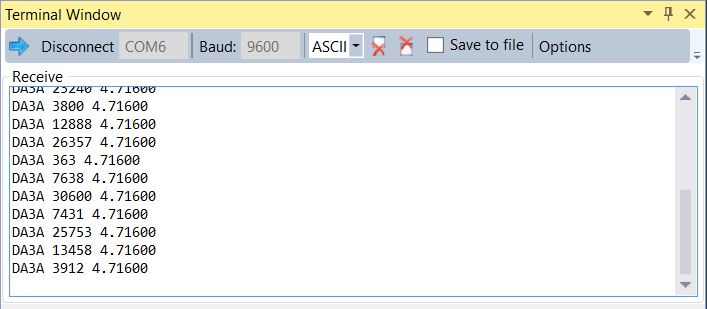
//-----------------------------------------------------------------------

1. **SCHEMATICS**

N/A (Everything done through Atmega328PB Xplained Mini)

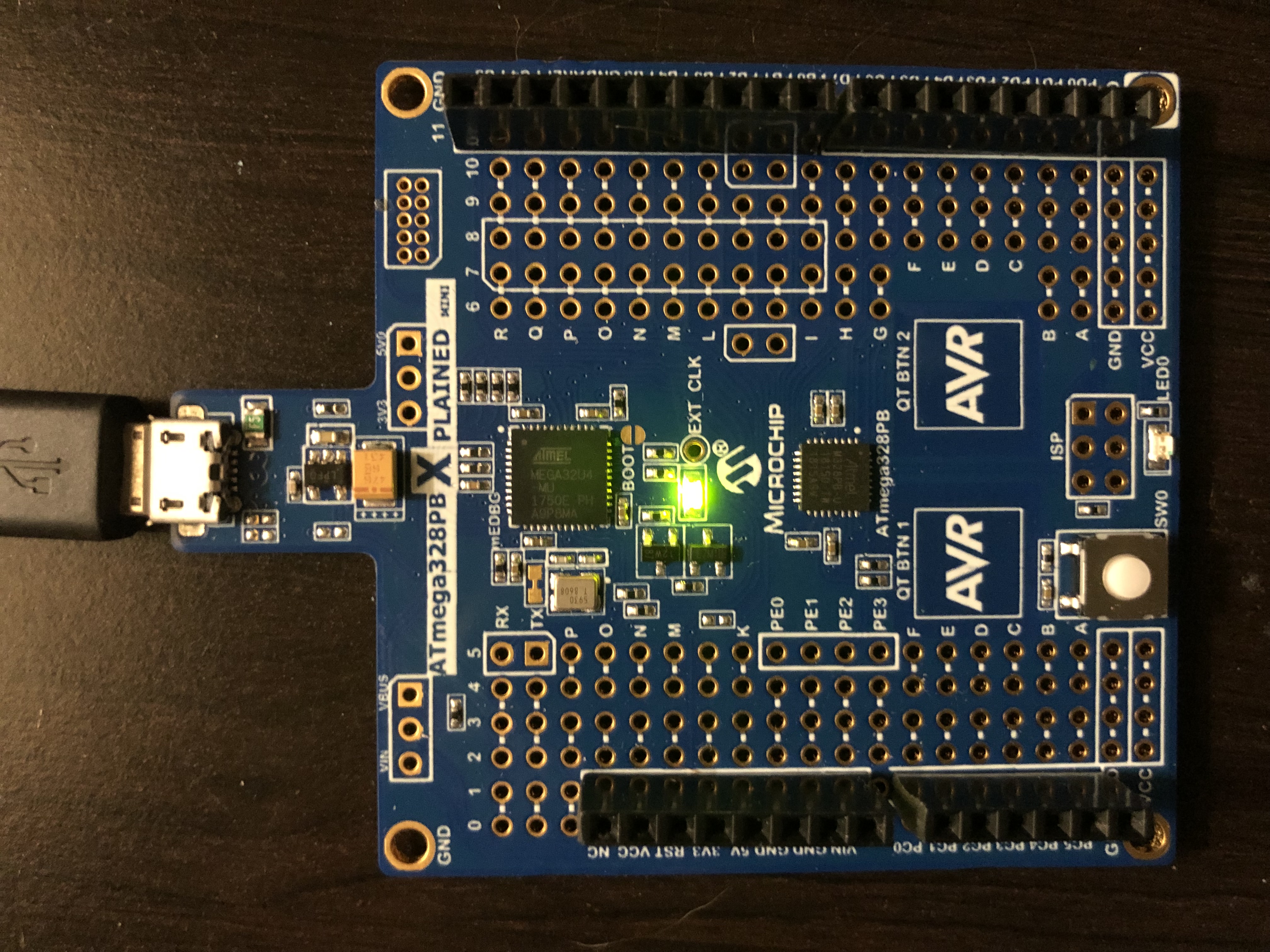
1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**

Since the Atmega328P/PB Xplained Mini have a built in AVR RX/TX module, we can send our serial data into a terminal which is able to display the string of chars, a random integer, and arbitrarily chosen float as shown in *Figure 3*:



*Figure 2 – Output Terminal of Serial Data*

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



*Figure 3 – Xplained Mini only for Serial Communication*

1. **VIDEO LINKS OF EACH DEMO**

N/A (Not required for assignment)

1. **GITHUB LINK OF THIS DA**

https://github.com/rockyg1995/ihswppdar/tree/master/DesignAssignments/DA3A

**Student Academic Misconduct Policy**

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

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

Rocky Gonzalez