CPE301 – SPRING 2020

Design Assignment 3A

Student Name: Minsung Cho

Student #: 2001446442

Student Email: chom3@unlv.nevada.edu

Primary Github address: <https://github.com/cho-minsung>

Directory: <https://github.com/cho-minsung/assignment3a>

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

ATMEGA328PB board TeraTerm ATMEL Studio

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

#ifndef F\_CPU

#define F\_CPU 16000000UL

#endif

#ifndef BAUD

#define BAUD 9600

#endif

//#include <stdlib.h>

#include <avr/io.h>

#include <stdio.h>

#include <util/delay.h>

#include <util/setbaud.h>

extern FILE uart\_output;

extern FILE uart\_input;

//int uart\_putchar(char c, FILE \*stream);

//int uart\_getchar(FILE \*stream);

//void uart\_init(void);

void uart\_init(void) {

UBRR0H = UBRRH\_VALUE;

UBRR0L = UBRRL\_VALUE;

UCSR0C = \_BV(UCSZ01) | \_BV(UCSZ00); /\* 8-bit data \*/

UCSR0B = \_BV(RXEN0) | \_BV(TXEN0); /\* Enable RX and TX \*/

}

int uart\_putchar(char c, FILE \*stream) {

if (c == '\n') {

uart\_putchar('\r', stream);

}

loop\_until\_bit\_is\_set(UCSR0A, UDRE0);

UDR0 = c;

return 0;

}

int uart\_getchar(FILE \*stream) {

loop\_until\_bit\_is\_set(UCSR0A, RXC0);

return UDR0;

}

FILE uart\_output = FDEV\_SETUP\_STREAM(uart\_putchar, NULL, \_FDEV\_SETUP\_WRITE);

FILE uart\_input = FDEV\_SETUP\_STREAM(NULL, uart\_getchar, \_FDEV\_SETUP\_READ);

int main(void) {

uart\_init();

stdout = &uart\_output;

stdin = &uart\_input;

while (1) {

/\* Blink led by toggling state of PORTB5 (Arduino digital 13). \*/

PORTB ^= \_BV(PORTB5);

puts("Hello world!");

\_delay\_ms(500);

}

return 0;

}

Module8A\_3.c code given in the Canvas.

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

#ifndef *F\_CPU*

#define *F\_CPU* 16000000UL

#endif

#ifndef BAUD

#define BAUD 9600

#endif

#include <avr/io.h>

#include <util/delay.h>

#include <util/setbaud.h>

#include <stdio.h>

void USART\_init(void);

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

int main(void)

{

USART\_init();

int num; //integer that will be randomly generated.

char snum1[8]; //character array that will be converted to string

char snum2[10]; //this one is for floating value.

while (1)

{

num = *rand*(); //gives a random value to the integer.

*itoa*(num, snum1, 5); //converts the integer to string.

float dec = ((*rand*() \* 2.2)); //give the floating random value.

*snprintf*(snum2, sizeof(snum2), "%f", dec); //converts the float to string.

USART\_tx\_string("This is how much I love Embedded Systems: \n");

USART\_tx\_string(snum1);

USART\_tx\_string("\n");

USART\_tx\_string(snum2);

USART\_tx\_string("\n");

*\_delay\_ms*(1000); //delay of 1 second.

}

}

/\* INIT USART (RS-232) \*/

void USART\_init( void)

{

UBRR0H = *UBRRH\_VALUE*;

UBRR0L = *UBRRL\_VALUE*;

UCSR0B = (1<<TXEN0); // enable TX & RX interrupt

UCSR0C = (3<<UCSZ00); // asynchronous 8 N 1

}

/\* SEND A STRING TO THE RS-232 \*/

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

{

while((\*data != '\0')) // while there's still data

{

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

UDR0 = \*data; // data is written into UDR0

data++;

}

}

code 3c.1 printing out random integer with delay function.

#ifndef *F\_CPU*

#define *F\_CPU* 16000000UL

#endif

#ifndef BAUD

#define BAUD 9600

#endif

#include <avr/io.h>

#include <util/delay.h>

#include <util/setbaud.h>

#include <avr/interrupt.h>

#include <stdio.h>

void USART\_init(void);

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

int main(void)

{

USART\_init(); //pretty much the same algorithm but this time it uses the timer interrupt.

int num;

char snum1[8];

char snum2[10];

TCCR0A = 0x00;

TCCR0B = 0x05;

TCNT0 = 0;

while (1)

{

num = *rand*();

*itoa*(num, snum1, 5);

float dec = ((*rand*() \* 2.2));

*snprintf*(snum2, sizeof(snum2), "%f", dec);

USART\_tx\_string("This is how much I love Embedded Systems: ");

USART\_tx\_string(snum1);

USART\_tx\_string(" ");

USART\_tx\_string(snum2);

USART\_tx\_string("\n");

//calculated with the cycle value estimate of 1 second.

for (int i = 0; i <= 61; i++)

{

while (TCNT0 != 255)

{

}

TCNT0 = 0;

}

}

}

/\* INIT USART (RS-232) \*/

void USART\_init( void)

{

UBRR0H = *UBRRH\_VALUE*;

UBRR0L = *UBRRL\_VALUE*;

UCSR0B = (1<<TXEN0); // enable TX & RX interrupt

UCSR0C = (3<<UCSZ00); // asynchronous 8 N 1

}

/\* SEND A STRING TO THE RS-232 \*/

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

{

while((\*data != '\0')) // while there's still data

{

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

UDR0 = \*data; // data is written into UDR0

data++;

}

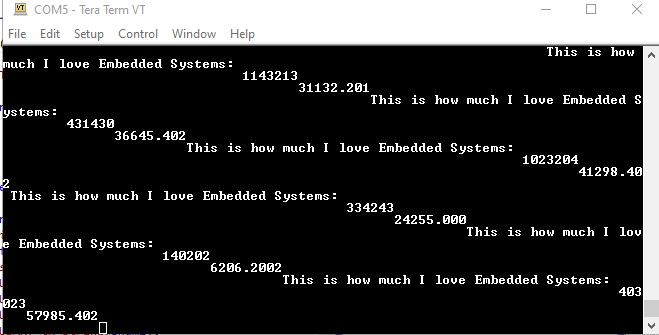
}

code with timer interrupt.

1. **SCHEMATICS**

Only connection was between the board and the computer.

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



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

the board setup was not necessary because there was nothing done to the board.

1. **VIDEO LINKS OF EACH DEMO**

https://youtu.be/i6Ry8QSXBIY

1. **GITHUB LINK OF THIS DA**

<https://github.com/cho-minsung/assignment3a>

**Student Academic Misconduct Policy**

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

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

Minsung Cho