CPE301 – SPRING 2019

Design Assignment 3A

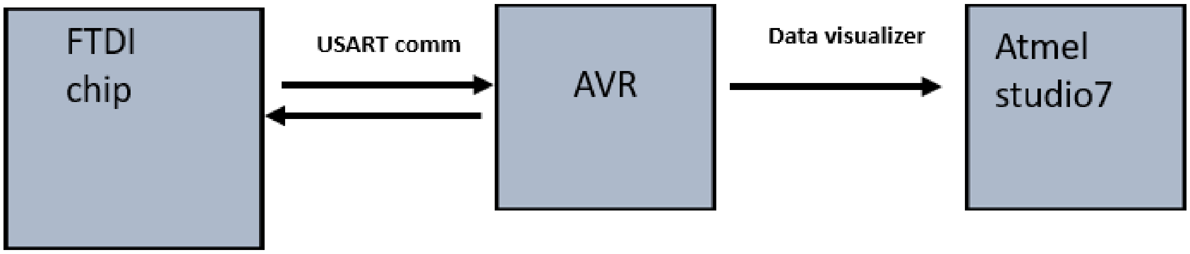
Student Name: Francisco Mata Carlos
Student #: 1012593607
Student Email: matacarl@unlv.nevada.edu
Primary Github address: https://github.com/chicosisco/da_sub.git
Directory: repository/cpe301/DesignAssignments/DA3A

1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

The components used for this assignment are the next:

a. Atmega328p Xplained Mini
b. Atmel Studio 7
c. FTDI chip

Block diagram with pins used in the Atmega328P



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

Task 1_A

1. Write a C AVR program that will display a string, random integer and floating point values on the serial terminal every 1 sec. Use a timer with interrupt for the 1 sec delay. Use a FTDI chip for serial to USB conversion

```
/*
 * DA3A.c
 *
 * Created: 3/28/2019 2:24:27 AM
 * Author : Francisco Mata carlos
 */

#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#include <stdio.h>

#define BAUDRATE 9600
#define BAUD_PRESCALLER (((F_CPU / (BAUDRATE * 16UL)))-1)

//Function declarations
volatile int Count;
void USART_init( unsigned int ubrr );
void USART_TX_string(char *data);
char outs[30];
int x_num;                               //creates variable for random number
float due_date;                          //creates variable for decimal point
char str[] = "July_4th_1776";             //creates string
char empty[] = " ";                      //creates space

int main(void)
{
       Count = 0;
       TIMSK0 |= (1<<TOIE0);              //sets interrupt when overflow occurs
       sei ();                           //set global overflow
       TCCR0A = 0;
       //normal mode
       TCCR0B |= (1<<CS02)|(1<<CS00);                                //prescaler = 1024
       USART_init(BAUD_PRESCALLER);                                  //baud prescaler
       USART_TX_string("connection_successful\r\n"); //prints the word connection
successful when connected and running
       while (1);
}
// USART (RS-232)
void USART_init( unsigned int ubrr ){
       UBRR0H = (unsigned char)(ubrr>>8);
       UBRR0L = (unsigned char)ubrr;
       UCSR0B = (1 << TXEN0);
       UCSR0C = (3 << UCSZ00);
}
//sending string to TEH RS-232
void USART_TX_string(char *data) {
       while (*data != '\0') {
```

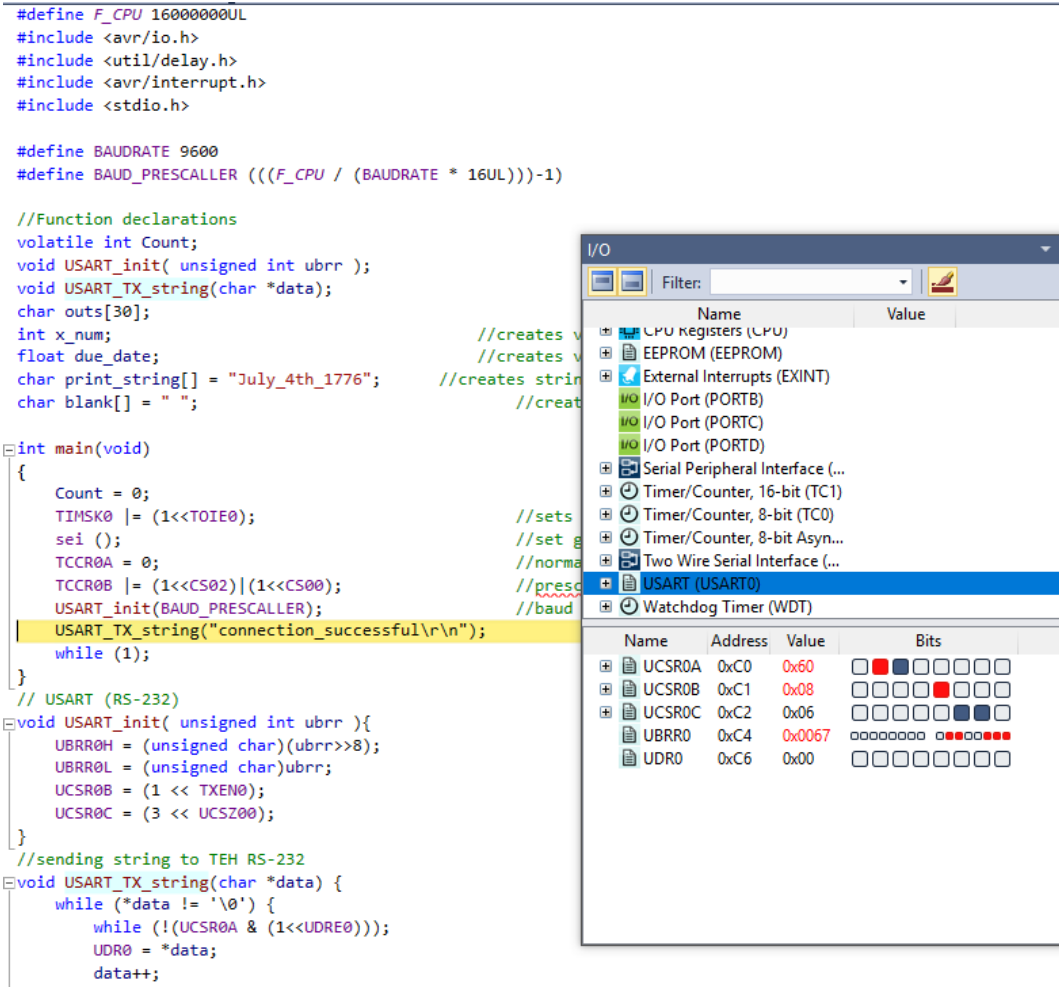
```
                while (!(UCSR0A & (1<<UDRE0)));
                UDR0 = *data;
                data++;
        }
}
ISR (TIMER0_OVF_vect){                          //timer0 overflow interrupt call
        while (Count < 61){
                if ((TIFR0 & 0x01) == 1){       //checks for overflow flag
                        TIFR0 = 0X01;           //reset overflow
                        Count++;
                }
        }
        if (Count > 60){
                USART_TX_string(str);           //prints string
                USART_TX_string(empty);         //prints space

                x_num = rand();                 // random number
                due_date = 327.19;              //creates floating value; due date of assignment

                snprintf(outs, sizeof(outs), "%3d\r\n", x_num);
                USART_TX_string(outs);
                USART_TX_string(empty);

                sprintf(outs, "%f", due_date);
                USART_TX_string(outs);
                USART_TX_string(empty);
                Count = 0;
        }
}
```

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

Same as above

4. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)

Task 1_A C code



```c
#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#include <stdio.h>

#define BAUDRATE 9600
#define BAUD_PRESCALLER (((F_CPU / (BAUDRATE * 16UL)))-1)

//Function declarations
volatile int Count;
void USART_init( unsigned int ubrr );
void USART_TX_string(char *data);
char outs[30];
int x_num;                                              //creates v
float due_date;                                         //creates v
char print_string[] = "July_4th_1776";              //creates strin
char blank[] = " ";                                         //creat

int main(void)
{
    Count = 0;
    TIMSK0 |= (1<<TOIE0);                                   //sets
    sei ();                                                 //set g
    TCCR0A = 0;                                             //norma
    TCCR0B |= (1<<CS02)|(1<<CS00);                          //presc
    USART_init(BAUD_PRESCALLER);                            //baud
    USART_TX_string("connection_successful\r\n");
    while (1);
}
// USART (RS-232)
void USART_init( unsigned int ubrr ){
    UBRR0H = (unsigned char)(ubrr>>8);
    UBRR0L = (unsigned char)ubrr;
    UCSR0B = (1 << TXEN0);
    UCSR0C = (3 << UCSZ00);
}
//sending string to TEH RS-232
void USART_TX_string(char *data) {
    while (*data != '\0') {
        while (!(UCSR0A & (1<<UDRE0)));
        UDR0 = *data;
        data++;
```

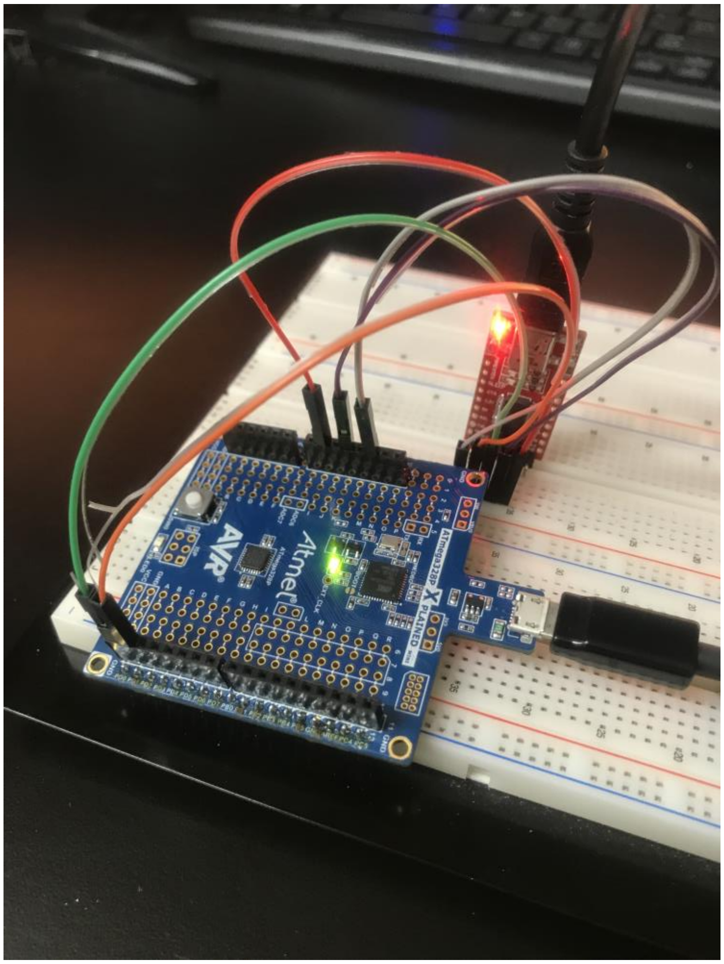
I/O
Filter:
Name
Value
CPU Registers (CPU)
EEPROM (EEPROM)
External Interrupts (EXINT)
I/O Port (PORTB)
I/O Port (PORTC)
I/O Port (PORTD)
Serial Peripheral Interface (...
Timer/Counter, 16-bit (TC1)
Timer/Counter, 8-bit (TC0)
Timer/Counter, 8-bit Asyn...
Two Wire Serial Interface (...
USART (USART0)
Watchdog Timer (WDT)

Name	Address	Value	Bits
UCSR0A	0xC0	0x60	□■■□□□□□
UCSR0B	0xC1	0x08	□□□□■□□□
UCSR0C	0xC2	0x06	□□□□□■■□
UBRR0	0xC4	0x0067	□□□□□□□□ □■■□□■■■
UDR0	0xC6	0x00	□□□□□□□□

5. SCREENSHOT OF EACH DEMO (BOARD SETUP)

Task 1

Photo below shows the set up



6. VIDEO LINKS OF EACH DEMO

DA3A

https://youtu.be/vQLKcaiOqFk

7. GITHUB LINK OF THIS DA

Student Academic Misconduct Policy

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

"*This assignment submission is my own, original work*".

Francisco Mata Carlos