CPE301 – SPRING 2019

Design Assignment 3, Part A

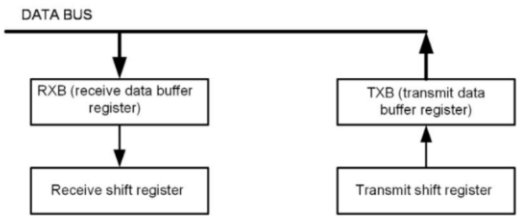
Student Name: James Skelly
Student #: 2000945485
Student Email: skellj1@unlv.nevada.edu
Primary Github address: https://github.com/skellj1/submission_da
Directory: skellj1/submission_da

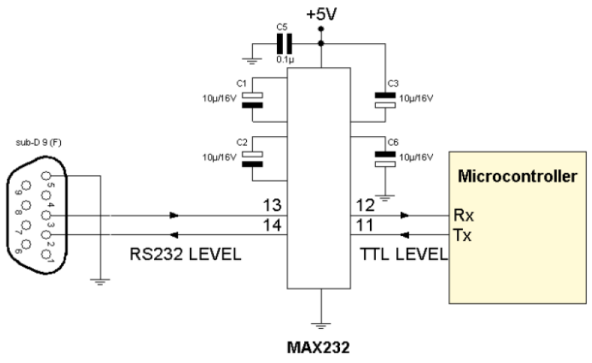
Submit the following for all Labs:

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

Components used include FTDI chip and board for UART communication, Atmel Studio 7, Tektronix TDS 2014 oscilloscope, scope probe, Xplained Mini board, Atmel Data Visualizer, jumper wires for connections from mini to FTDI chip and board, breadboard, iphone for recording.





From CPE 301 Slides Module 8

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

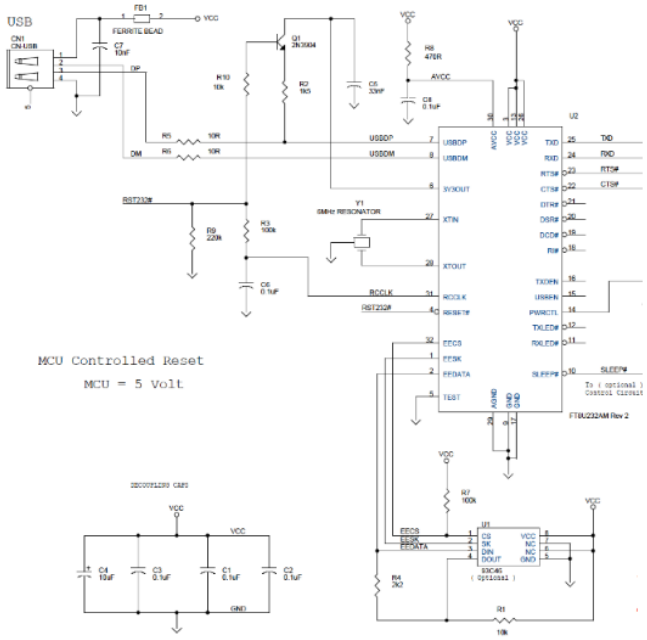
```
// DA3A_C.c
// Created: 3/25/2019 11:51:23 AM
// Author : James Skelly
// CPE 301 Design Assignment 3A: UART

// Define board frequency
#define F_CPU 16000000UL
// Include necessary headers for operation
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#include <stdio.h>

// Define Baud Rate and Baud Rate Prescaler
#define BAUDRATE 9600
#define BAUD_PRESCALLER (((F_CPU / (BAUDRATE * 16UL)))-1)

// Define the functions used in the program
volatile int Counter;
void USART_init( unsigned int ubrr );
void USART_TX_string(char *data);

// Define variables used in the program
float phi;
char outs[30];
int random_number;

// Define string to be output and white space variable

char str[] = "= The Golden Ratio";              // Initialize string "= The Golden Ratio"
char space[] = " ";                             // Initiliaze string for white space

int main(void)
{
          Counter = 0;                                                  // Initialize counter variable to 0
          TIMSK0 |= (1<<TOIE0);                                         // Set timer overflow interrupt enable bit
          sei ();                                                       // Enable interrupts
          TCCR0A = 0x00;                                                // Set timer 0 to normal mode operation
          TCCR0B |= (1<<CS02)|(1<<CS00);          // Set prescaler to 1024
          USART_init(BAUD_PRESCALLER);            // Initialize Baud prescaller in program
          USART_TX_string("Connected\r\n");       // Prints connected once the USART is connected
          while (1);                                        // wait here for interrupt to occur

}
          // INT USART (RS-232)
void USART_init( unsigned int ubrr )              // USART initial function header
{
          UBRR0H = (unsigned char)(ubrr>>8);      // Set baud rate reg. high bits
          UBRR0L = (unsigned char)ubrr;           // set baud rate reg. low bits
          UCSR0B = (1 << TXEN0);                                        // Enable the Tx interrupt
          UCSR0C = (3 << UCSZ00);                                       // Asynchronous 8 N 1
}
void USART_TX_string(char *data)                                        // USART string initial function header
{
          while (*data != '\0')                                         // while conj data is not equal to zero
          {
                    while (!(UCSR0A & (1<<UDRE0)));                     // while not conditions met
                    UDR0 = *data;                                       // set UDR0 to data
                    data++;                                             // increment data variable
          }

ISR (TIMER0_OVF_vect){                                  // Header for timer overflow interrupt subroutine
          while (Counter < 61){                         // Loop while count is less than 61
                    if ((TIFR0 & 0x01) == 1){           // If timer interrupt flag bit 0 is high...
                              TIFR0 = 0X01;             // Clear flag
                              Counter++;                // Increment count variable
                    }
          }
          if (Counter > 60){                            // If counter variable is greater than 60...
                    USART_TX_string(str);               // Print string character
                    USART_TX_string(space);             // Print out blank space
                    random_number = rand();             // generate a random number in random number variable
                    phi = 1.618034;                     // stores the golden ratio, 1.618034, in phi variable
                    snprintf(outs, sizeof(outs), "%3d\r\n", random_number);               // print out random number
                    USART_TX_string(outs);
                    USART_TX_string(space);
                    sprintf(outs, "%f", phi);           // Print out floating point value
                    USART_TX_string(space);
                    Counter = 0;                        // Reset counter to 0
          }
}
```

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

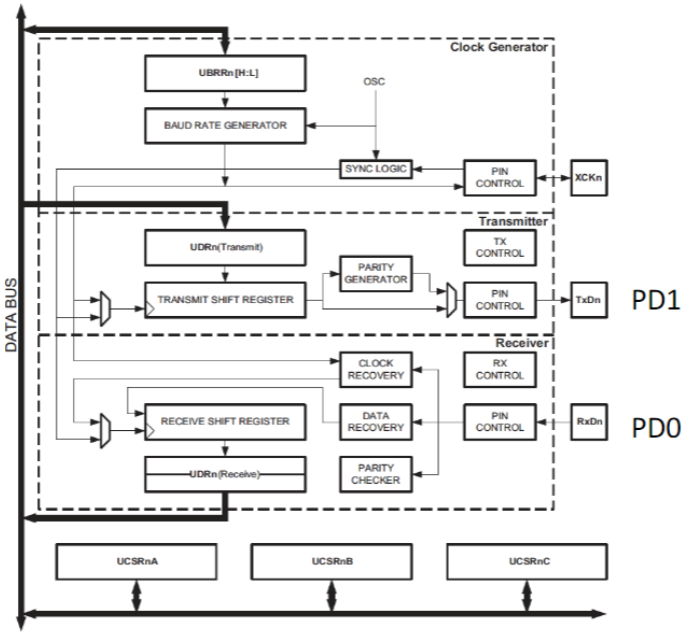
Not Applicable for this assignment.

4. SCHEMATICS

Below is the schematic of the microcontroller to USB connections to the FTDI chip used for the UART communication in this assignment, taken from the CPE 310L Module 8 slides.

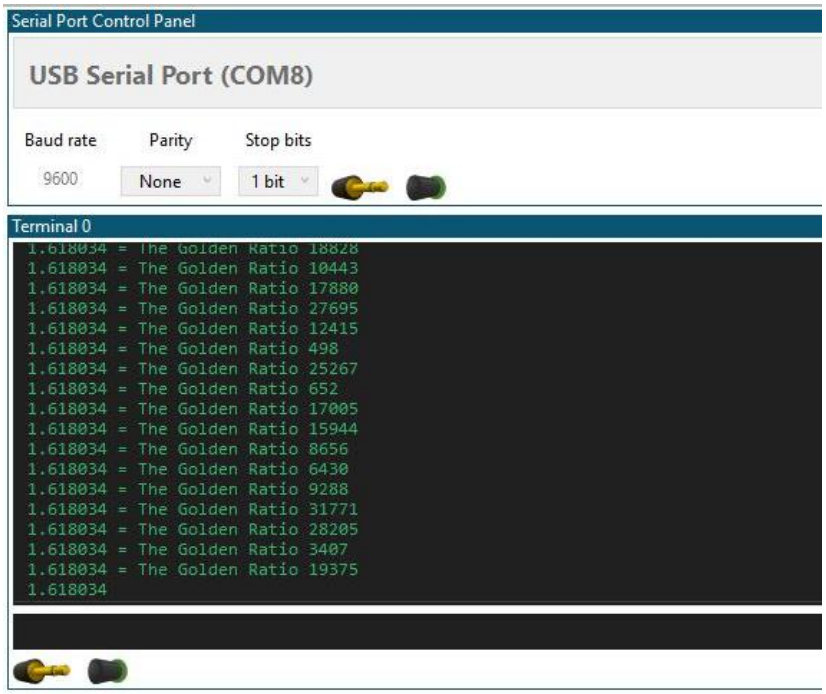


The schematic below is the schematic for the UART internal to the ATmega328P.

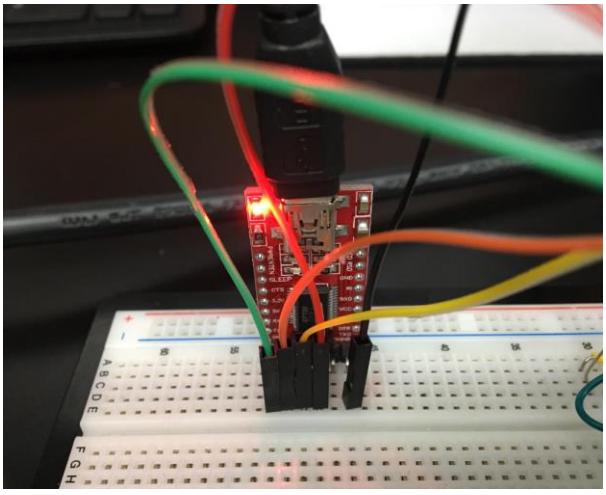


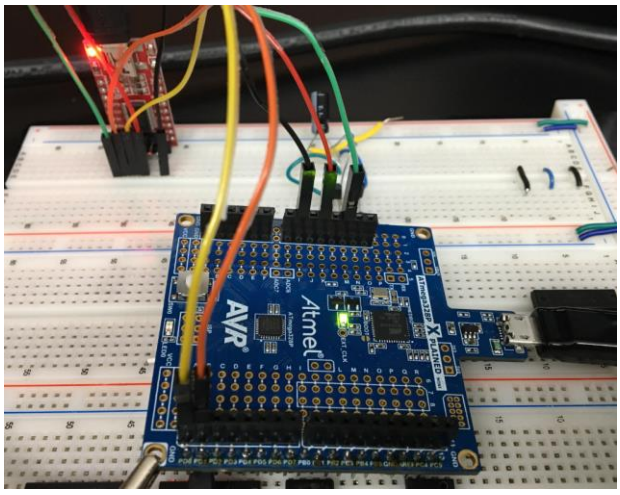
5. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)

Below is the output on the USB Serial Port on Terminal 0 in Atmel Studio 7. The floating point value that I chose is 1.618034, also known as the Golden Ratio. I made my string "= The Golden Ratio" so that every second, the terminal output will consist of the value of phi, the Golden Ratio string, and a random number, obviously generated randomly.



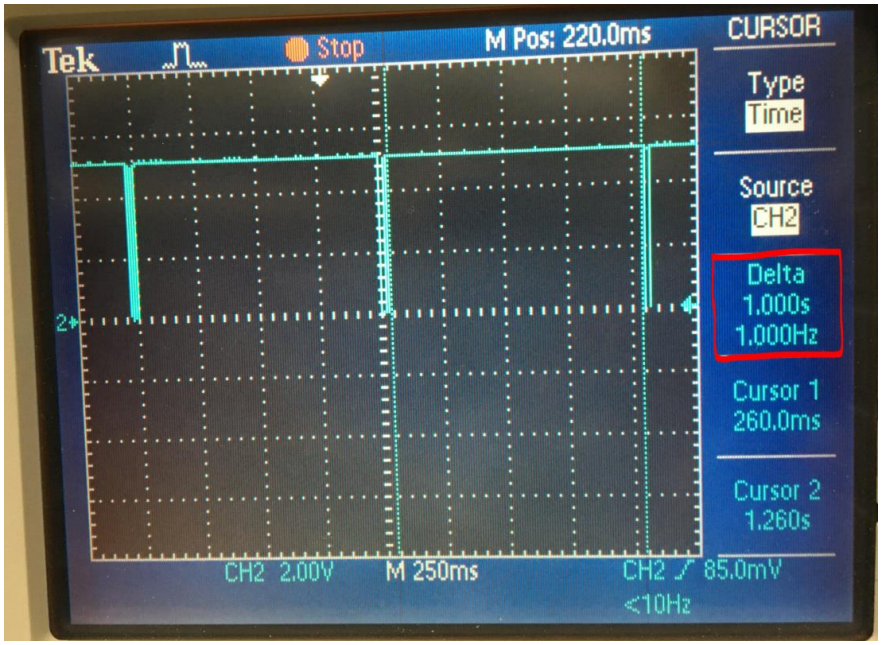
6. SCREENSHOT OF EACH DEMO (BOARD SETUP)





Above we see the jumper cables routed from the Xplained Mini to the FTDI board, which is connected to our breadboard. The cables routed consist of two data cables on PORTD.0/1, GND, RESET, and 5V VCC.

Below is the oscilloscope output waveform, which as we can see from the photo, pulses once per second, in accordance with the instructions. The data outputs at a rate of one line per second.



7. VIDEO LINKS OF EACH DEMO

https://www.youtube.com/watch?v=FYAaaMG2Dtk

8. GITHUB LINK OF THIS DA

https://github.com/skellj1/submission_da

Student Academic Misconduct Policy
http://studentconduct.unlv.edu/misconduct/policy.html

"This assignment submission is my own, original work".
James W. Skelly