

# Design Assignment DA3a

---

Student Name: Daniel Senda

Student #: 5002362633

Student Email: sendad1@unlv.nevada.edu

Primary Github address: <https://github.com/dsenda/Smiles>

Directory: 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

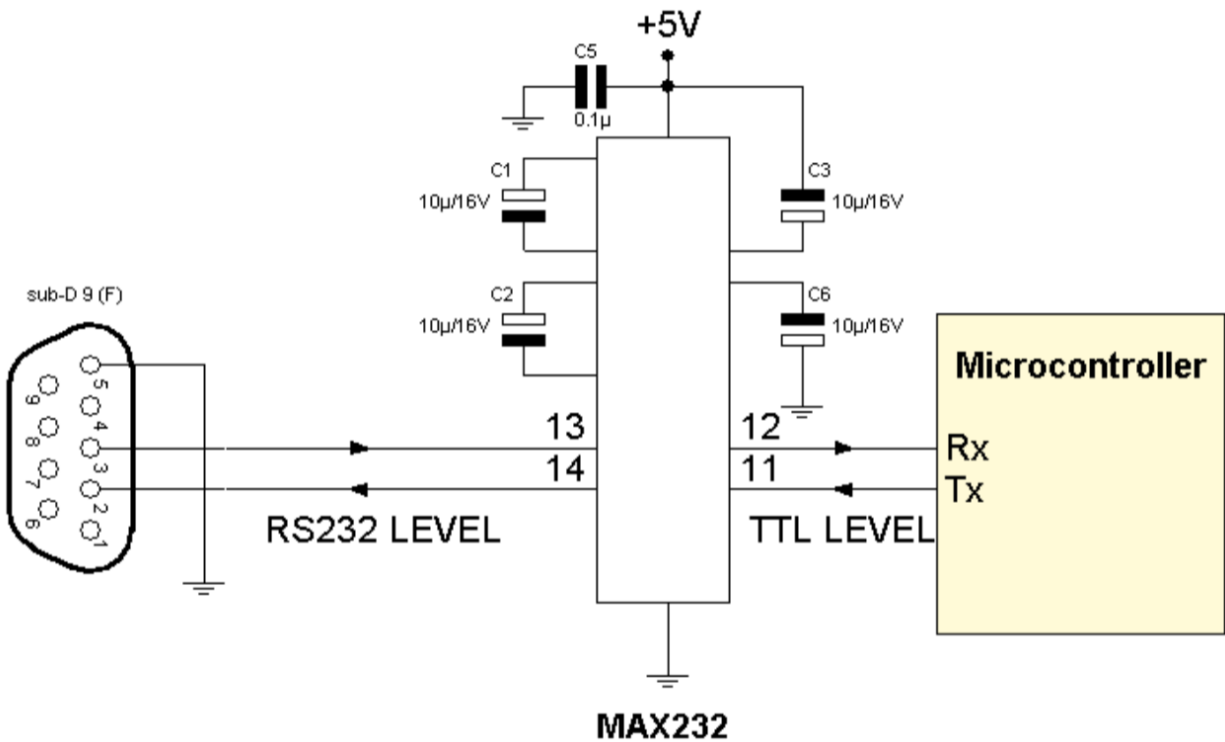
*List of Components used:*

Atmel Studio 7

ATmega328P Xplained Mini

FTDI chip

*Block diagram about the transmission connections:*



## 2. INITIAL/MODIFIED/DEVELOPED CODE OF TASK DA3a

Not applicable, there is no modified code.

## 3. DEVELOPED MODIFIED CODE OF TASK DA3a

The following is the code that accomplishes the desired tasks that are required.

```
// DA3a_C_code_string_serial2usb.c
// Daniel Senda
// Displays a string, random integer and floating-point values on the serial terminal
// every 1s.
// Uses the FTDI chip for serial to USB conversion.

#define F_CPU 16000000UL
#define BAUDRATE 9600
#define BAUD_PRESCALLER (((F_CPU / (BAUDRATE * 16UL)))-1)
#include <avr/io.h>
```

```

#include    <util/delay.h>
#include    <avr/interrupt.h>
#include    <stdio.h>

// Defines functions used for this code.
volatile int Count;
void USART_init( unsigned int ubrr );
void USART_TX_string(char *data);
char outs[30];
int n;
float num_year = 0.001998;           // Declares the float and assigns a value.
char str[] = "Toyota Supra 2JZ-GTE"; // Declares a string.
char empty[] = " ";                  // Declares empty string for space.

int main(void)
{
    Count = 0;                        // Sets count to 0.
    TIMSK0 |= (1<<TOIE0);             // Sets interrupts when overflow flag is set.
    sei ();                           // Sets the global interrupt.
    TCCR0A = 0x00;                    // Sets normal mode for timer0.
    TCCR0B |= (1<<CS02)|(1<<CS00);     // Sets 1024 prescaler.
    USART_init(BAUD_PRESCALLER);      // Sets the BAUD prescaler.
    USART_TX_string("Connected!\r\n"); // Prints "Connected" to serial.
    while (1);                        // Loop that waits for interrupt.
}

// INT USART (RS-232)
void USART_init( unsigned int ubrr ){
    UBRR0H = (unsigned char)(ubrr>>8);
    UBRR0L = (unsigned char)ubrr;
    UCSR0B = (1 << TXEN0);            // Enables Tx interrupt.
    UCSR0C = (3 << UCSZ00);           // Asynchronous 8 N 1.
}

// Sends a string to the RS-232
void USART_TX_string(char *data) {
    while (*data != '\0') {
        while (!(UCSR0A & (1<<UDRE0)));
        UDR0 = *data;
        data++;
    }
}

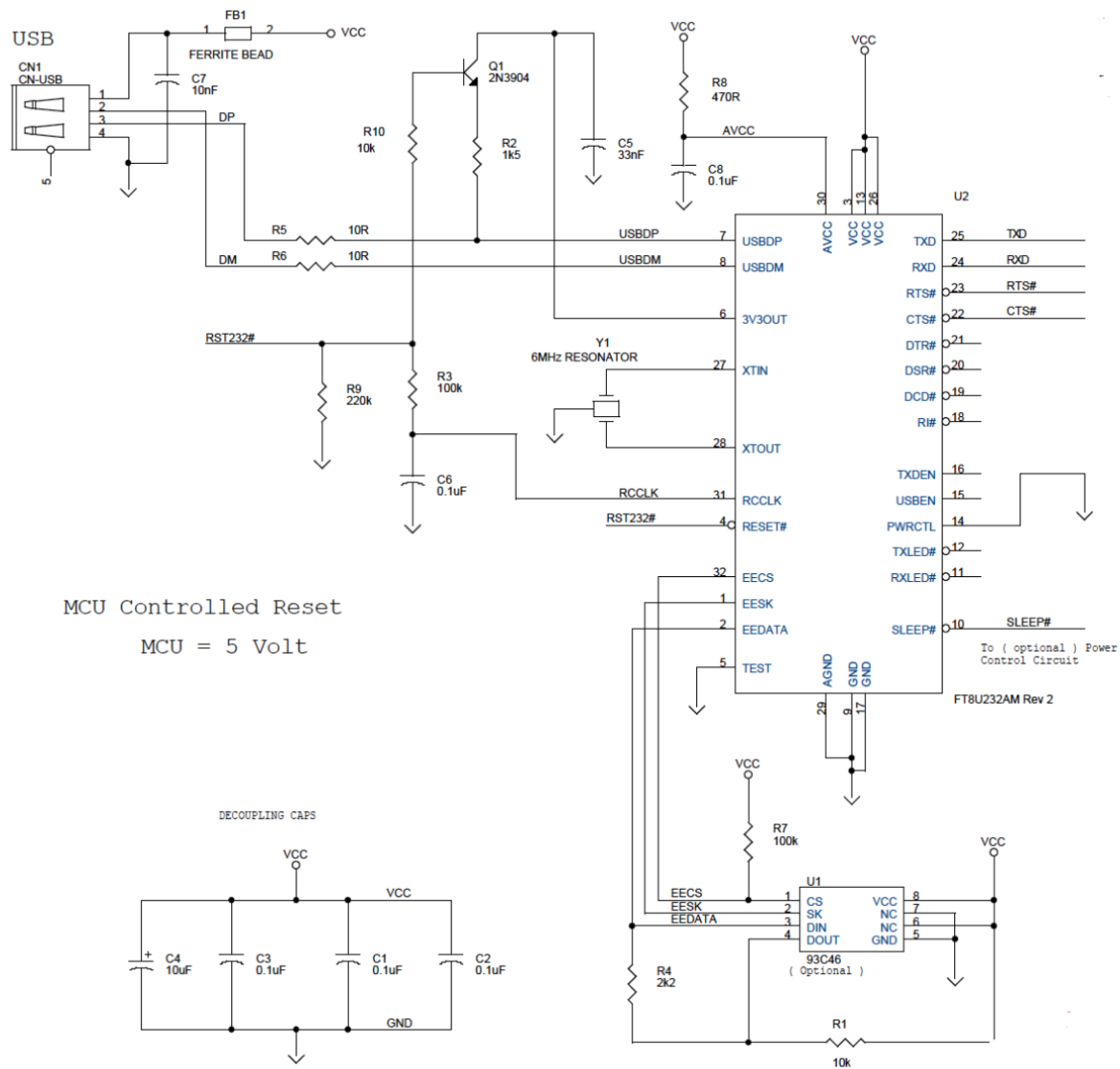
ISR (TIMER0_OVF_vect){
    while (Count < 61){                // TV0V flag interrupt.
        if ((TIFR0 & 0x01) == 1)      // Loops while Count value is less than 76.
        {                             // If TV0V flag is set, do the following.
            TIFR0 = 0x01;              // Clears the flag.
            Count++;                    // Increments the count.
        }
    }
    if (Count > 60){                   // Executes the following if Count is less than
        USART_TX_string(str);          // Prints the string "Toyota Supra 2JZ-GTE".
        USART_TX_string(empty);        // Prints an empty space.

        n = rand();                    // Generates a random number and stores in n.
    }
}

```

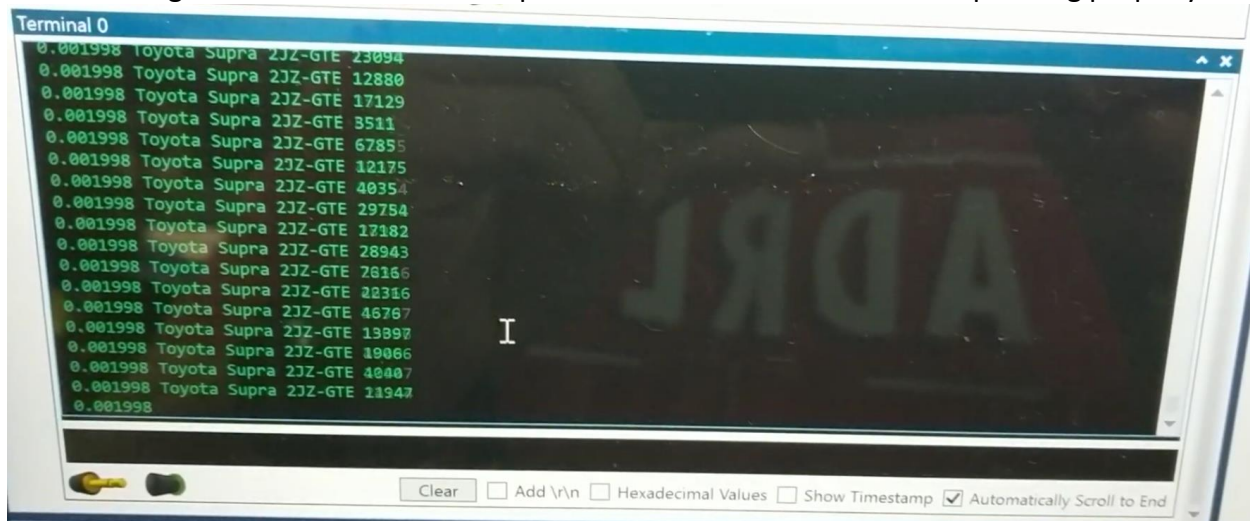
}

The following schematic from the slides describe the connections that are made relatively well.

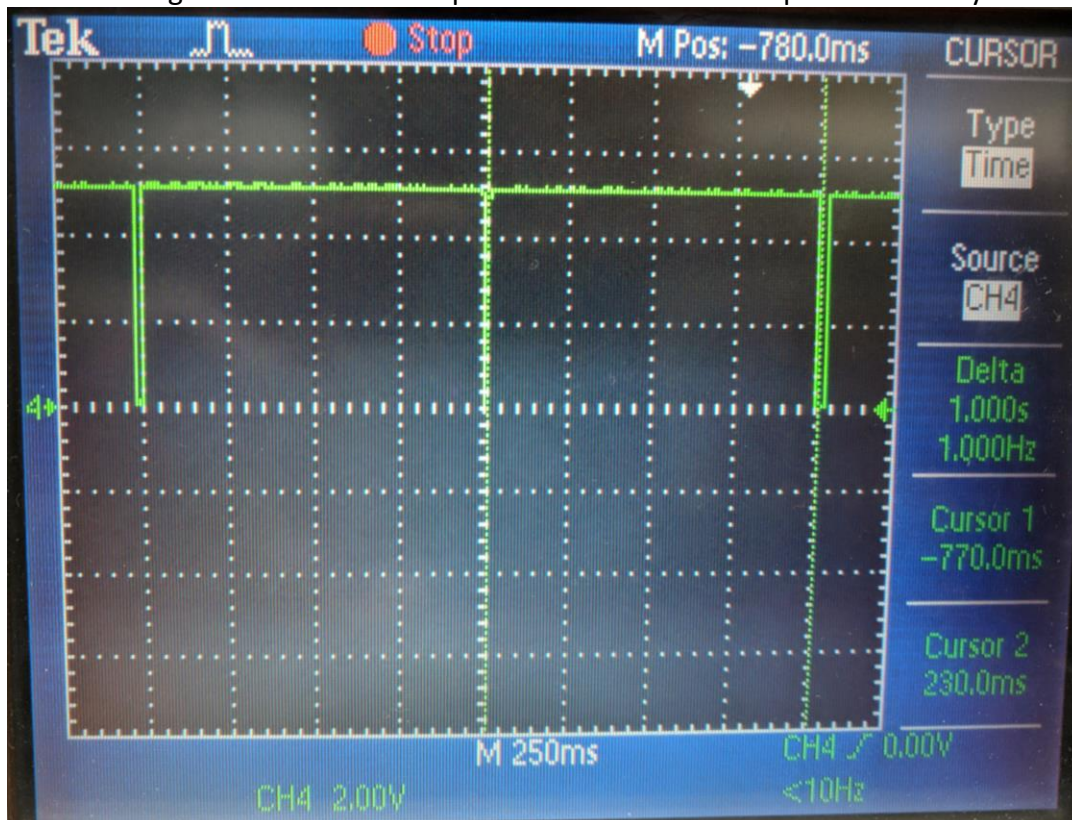


## 5. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)

The following is a screenshot of the output which demonstrates that it is operating properly.

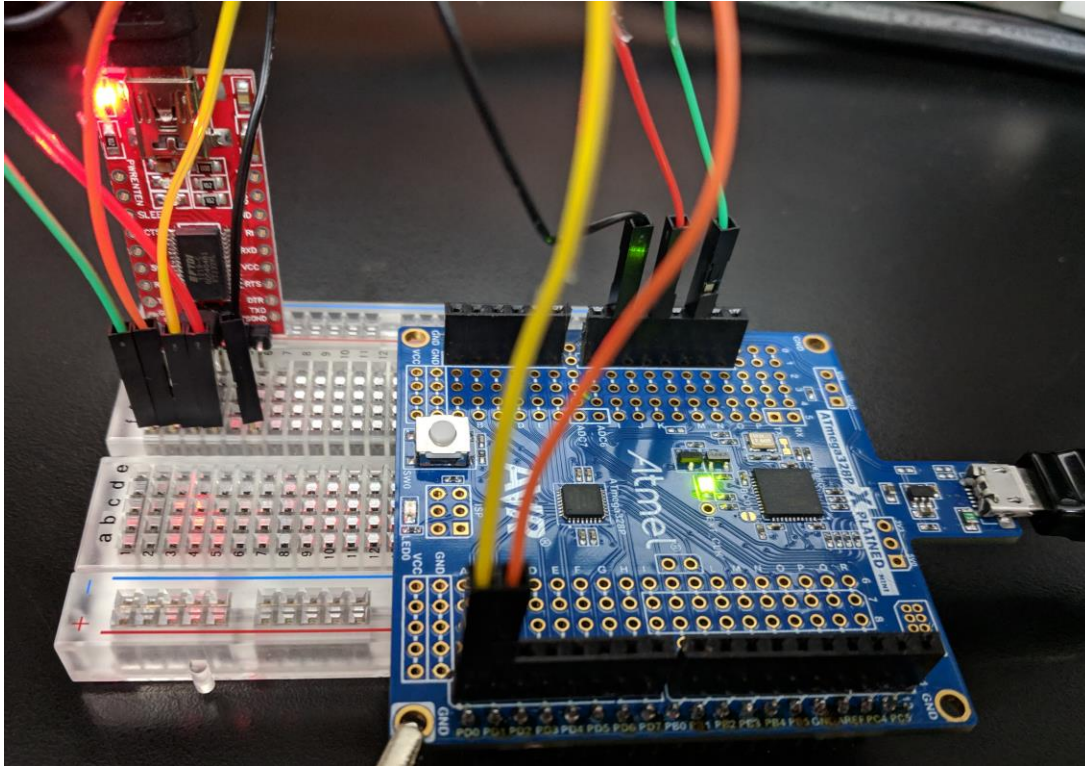


The following is the waveform output that shows that it outputs data every 1s.



## 6. SCREENSHOT OF EACH DEMO (BOARD SETUP)

The following is a picture of the board setup:



## 7. VIDEO LINKS OF EACH DEMO

<https://youtu.be/IPxbpYkhnuA>

## 8. GITHUB LINK OF THIS DA

<https://github.com/dsenda/Smiles/tree/master/DA3a>

**Student Academic Misconduct Policy**

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

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

Daniel Senda