

Design Assignment 3A

Student Name: Dillon Archibald

Student #: 5004439916

Student Email: Archid1@unlv.nevada.edu

Primary Github address: <https://github.com/Dil-bert/Alabaster.git>

Directory: Alabaster/DesignAssignments/DA3a/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

- 7x Wires
- 1x atmega328P
- 1x Logic Analyzer
- 1x FTDI ft232rl usb to serial
- 2x USB mini-b cables
- 1x USB micro-b cable
- 1x Mini bread board

Block diagram with pins used in the Atmega328P

Will Resubmit with Block Diagram

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

```
//provided via Dr.Venki
#define BAUD 9600
#define F_CPU 16000000UL
#include <avr/io.h>
#include <stdio.h>
#include <util/delay.h>
#include <avr/interrupt.h>

void USART_send(char data); // Used to send integer to terminal
void USART_putstr(char* StringPtr); // Used to take in every character in the string and sends
it to the terminal
void USART_init(void); // Initializes the analog to digital functions, as well as OVF interrupt

char stringtype[] = "String: "; // Declaring the string value on screen
char inttype[] = "Integer: "; // Declaring the integer value on screen
char floattype[] = "Floating Point: "; // Declaring the floating point value on screen
char Space[] = "\n"; // Used to create the next line

char String[] = "This is my DA3A"; //String[] is the variable to output into terminal
char outs[20]; // Allocating memory space to contain the float value
volatile float adc_temp = 74.744; // Sets the float value

int main(void)
{
    USART_init(); // Initializes the analog to digital functions as well as OVF interrupt

    while(1)
    {
        // main loop
    }
}

ISR (TIMER1_OVF_vect)
{
```

```

    USART_putstrstring(Space); // creates next line
    USART_putstrstring(stringtype); // LABEL PRINT "String: "
    USART_putstrstring(String); // prints string "This is my DA3A"
    USART_putstrstring(Space); // creates next line
    USART_putstrstring(inttype); // LABEL PRINT "Integer: "
    USART_send('5'); // prints value 5
    USART_putstrstring(Space); // creates next line
    USART_putstrstring(floattype); // LABEL PRINT "Floating Point: "
    snprintf(outs, sizeof(outs), "%f\r\n", adc_temp); // the floating point characters are stored
in outs
    USART_putstrstring(outs); // transmits outs to UART
    USART_putstrstring(Space); // creates next line
    TCNT1 = 49911; // Reset timer
}

void USART_init( void )
{
    UBRROH = 0;
    UBRROL = F_CPU/16/BAUD - 1; // Used for the BAUD prescaler
    UCSROC = _BV(UCSZ01) | _BV(UCSZ00); /* 8-bit data */
    UCSROB = _BV(RXEN0) | _BV(TXEN0); /* Enable RX and TX */
    TCCR1B |= 5; //(1 << CS12) | (1 << CS10); // Sets prescaler to 1024
    TIMSK1 = (1 << TOIE1); // Enables overflow flag
    TCNT1 = 49911; // 1 second delay = (0xFFFF) - TCNT = 65535 - 15624 = 49911
    sei();
}

void USART_send(char data)
{
    while (!(UCSROA & (1 << UDRE0))); // Until UDRE0 goes high, it will keep looping
    UDRO = data; // UDRO register grabs the value given from the parameter
}

void USART_putstrstring(char *StringPtr)
{
    while ((*StringPtr != '\0')){ // Until it reaches the end of the line, it will keep looping
        while (!(UCSROA & (1 << UDRE0))); // Until UDRE0 goes high, it will keep looping
        UDRO = *StringPtr; // UDRO register grabs the value given from the parameter
        StringPtr++; // but it does it by every character as shown here
    }
}

```

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

Modified by me version

```

#define BAUD 9600
#define F_CPU 16000000UL
#include <avr/io.h>
#include <stdio.h>
#include <stdlib.h>
#include <util/delay.h>
#include <avr/interrupt.h>

void USART_send(char data); // Used to send integer to terminal

```

```

void USART_putstr(char* StringPtr); // Used to take in every character in the string and sends
it to the terminal
void USART_init(void); // Initializes the analog to digital functions, as well as OVF interrupt
void USART_tx_int(int x); // Convert a random integer to a string

char stringtype[] = "String: "; // Declaring the string value on screen
char inttype[] = "Integer: "; // Declaring the integer value on screen
char floatype[] = "Floating Point: "; // Declaring the floating point value on screen
char Space[] = "\n"; // Used to create the next line

char String[] = "DA3A"; //String[] is the variable to output into terminal
char outs[20]; // Allocating memory space to contain the float value
char outints[40]; // Allocating memory space to contain the random int value
volatile float adc_temp = 69.6969696969; // Sets the float value
volatile int r; // Global Variable for the random int (So I don't have to pass it)

int main(void)
{
    USART_init(); // Initializes the analog to digital functions as well as OVF interrupt
    srand(5); // Random number seed
    while(1)
    {
        // main loop
        r = rand(); // making and setting a random int
    }
}

ISR (TIMER1_OVF_vect)
{
    USART_putstr(Space); // creates next line
    USART_putstr(stringtype); // LABEL PRINT "String: "
    USART_putstr(String); // prints string "DA3A"
    USART_putstr(Space); // creates next line
    USART_putstr(inttype); // LABEL PRINT "Integer: "
    snprintf(outints, sizeof(outints), "%d", r); // the int characters are stored in outints
    USART_putstr(outints); // transmits outints to UART
    USART_putstr(Space); // creates next line
    USART_putstr(floatype); // LABEL PRINT "Floating Point: "
    snprintf(outs, sizeof(outs), "%f\r\n", adc_temp); // the floating point characters are stored
in outs
    USART_putstr(outs); // transmits outs to UART
    USART_putstr(Space); // creates next line
    TCNT1 = 49911; // Reset timer
}

void USART_init( void )
{
    UBRROH = 0;
    UBRROL = F_CPU/16/BAUD - 1; // Used for the BAUD prescaler
    UCSROC = _BV(UCSZ01) | _BV(UCSZ00); /* 8-bit data */
    UCSROB = _BV(RXEN0) | _BV(TXEN0); /* Enable RX and TX */
    TCCR1B |= 5; //(1 << CS12) | (1 << CS10); // Sets prescaler to 1024
    TIMSK1 = (1 << TOIE1); // Enables overflow flag
    TCNT1 = 49911; // 1 second delay = (0xFFFF) - TCNT = 65535 - 15624 = 49911
    sei();
}

void USART_send(char data)

```

```

{
    while (!(UCSROA & (1 << UDRE0))); // Until UDRE0 goes high, it will keep looping
    UDRO = data; // UDRO register grabs the value given from the parameter
}

void USART_putstr(char *StringPtr)
{
    while ((*StringPtr != '\0')){ // Until it reaches the end of the line, it will keep looping
        while (!(UCSROA & (1 << UDRE0))); // Until UDRE0 goes high, it will keep looping
        UDRO = *StringPtr; // UDRO register grabs the value given from the parameter
        StringPtr++; // but it does it by every character as shown here
    }
}

// ***** my Original transmit function (much more complex than needed) *****

/*void USART_tx_int(int x){
    char buffer[64];
    int ret = sprintf(buffer, sizeof buffer, "%d", x);
    USART_send(buffer);
    if(ret < 0){
        return EXIT_FAILURE;
    }
    if(ret >= sizeof buffer){
        //Result was truncated - resize the buffer and retry
        USART_send("\r\nResult was truncated - resize the buffer and retry\r\n");
    }
}
*/

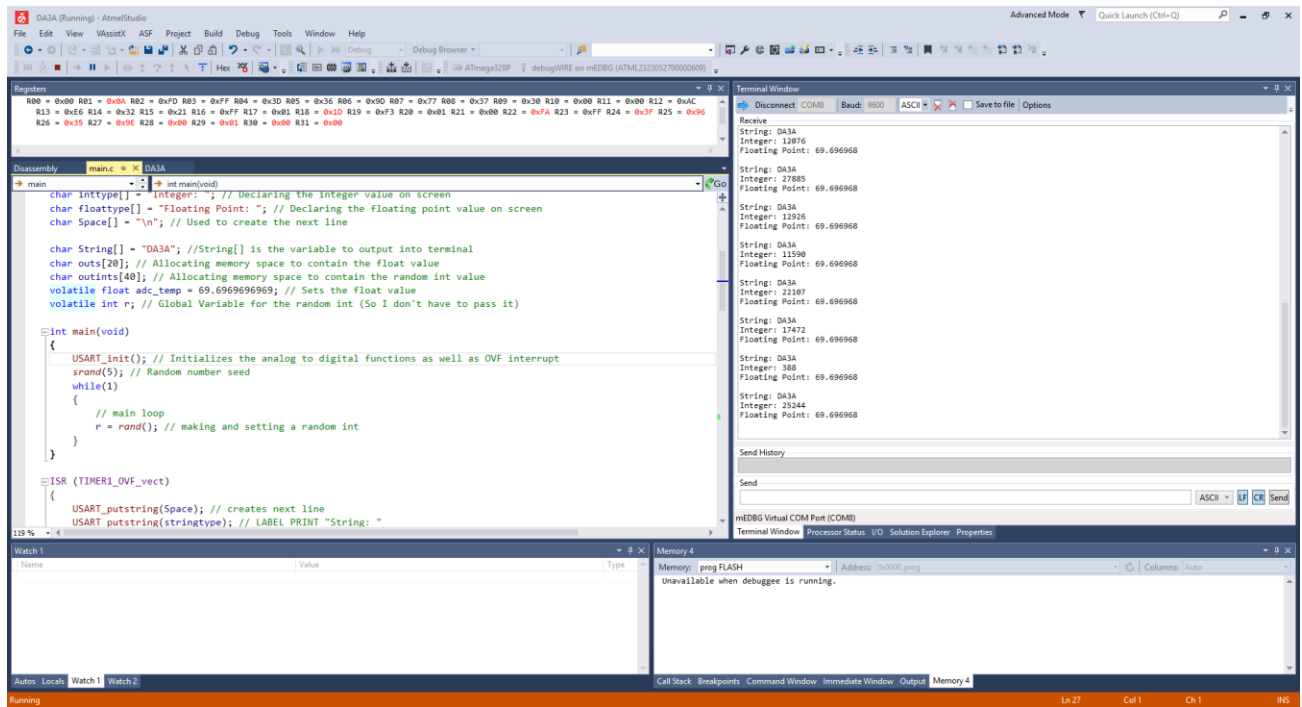
```

4. SCHEMATICS

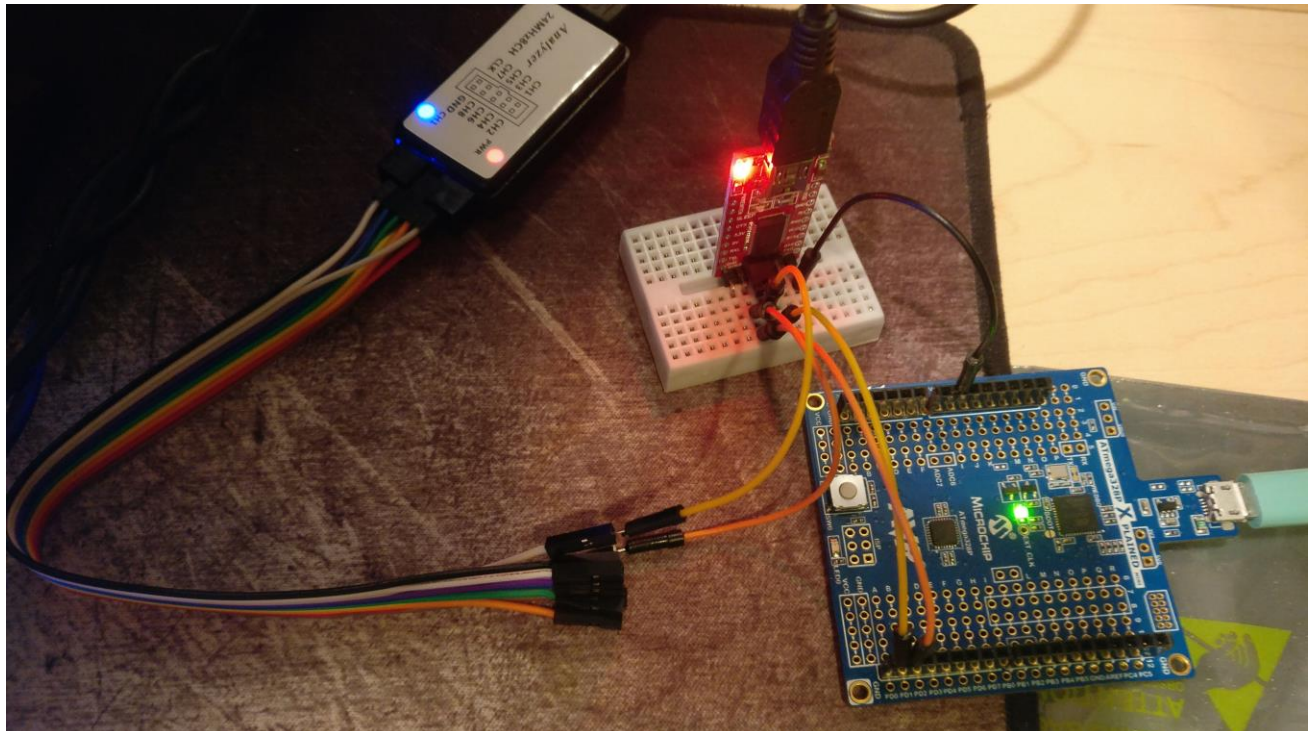
Use fritzing.org

Will Resubmit with Schematic
(midterm for 300L tomorrow morning, must sleep)

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



6. SCREENSHOT OF EACH DEMO (BOARD SETUP)



7. VIDEO LINKS OF EACH DEMO

<https://youtu.be/OPko-VRbBvI>

8. GITHUB LINK OF THIS DA

Dil-bert/Alabaster/DesignAssignments/DA3a/DA3A

Student Academic Misconduct Policy

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

"This assignment submission is my own, original work".
Dillon Archibald