CPE301 – SPRING 2019

Design Assignment 3A

Student Name: Tyler Gardenhire
Student #: 8000450294
Student Email: gardenhi@unlv.nevada.edu
Primary Github address: gardenhi@unlv.nevada.edu
Directory: https://github.com/tylergardenhire/submission_projects.git

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

Atmel Studio 7 w/ AVR assembly and simulator and Atmega328p board used.

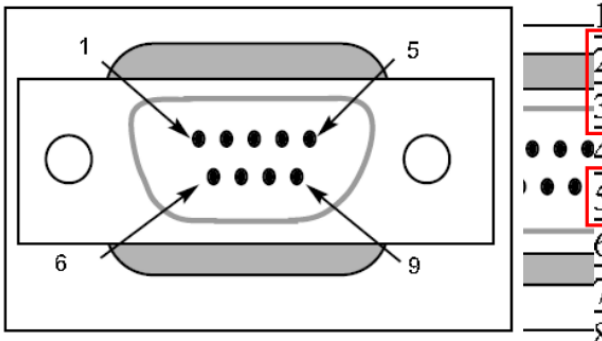
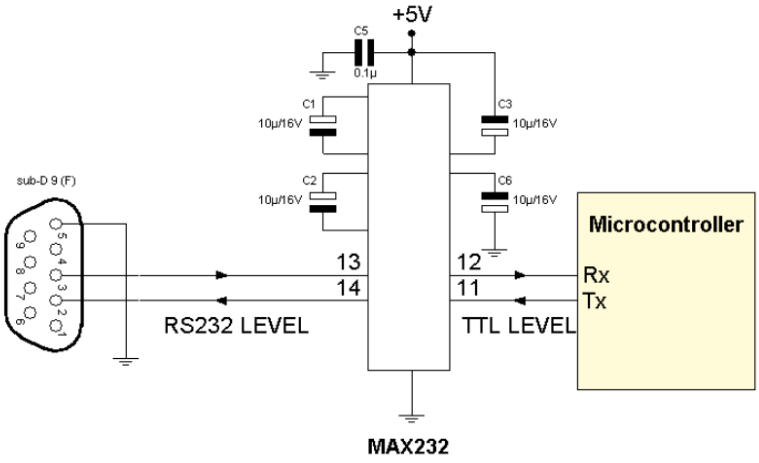
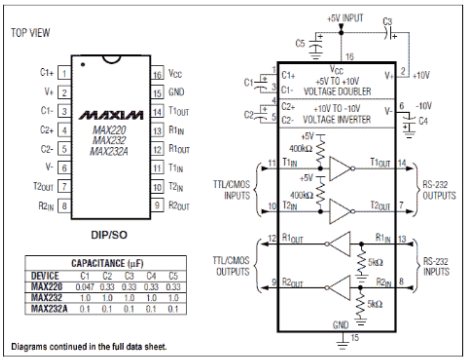


Figure 11-5. DB-9 9-Pin Connector

Pin	Description
1	Data carrier detect (DCD)
2	Received data (RxD)
3	Transmitted data (TxD)
4	Data terminal ready (DTR)
5	Signal ground (GND)
6	Data set ready (DSR)
7	Request to send (RTS)
8	Clear to send (CTS)
9	Ring indicator (RI)





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

Task 1 C code:

```
#define F_CPU 8000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <util/setbaud.h>
#define BAUD 9600
#include <stdio.h>

void read_adc(void);// Function Declarations
void adc_init(void);
void USART_init( unsigned intubrr);
void USART_tx_string( char *data );

volatile unsigned intadc_temp;
char outs[20];

//initializes the USART (RS232 interface)
void USART_init(void) {
	UBRR0H = (uint8_t)(BAUD_PRESCALLER >> 8);
	UBRR0L = (uint8_t)(BAUD_PRESCALLER);
	UCSR0B = (1 << TXEN0)|(1<<RXEN0); // Enable receiver, transmitter & RX interrupt
                                                                //asynchronous 8 N 1
	UCSR0C = (0 << UMSEL1) |
		(0 << UMSEL0) | // 00 async operation, 01 synch operation
		(0 << UPM10) | // Parity -0 Disabled, 0 Reserved, 1 Enabled Even, 1 Enabled
		(0 << UPM00) | // Parity -0 Disabled, 1 Reserved, 0 Enabled Even, 1 Enabled
		(0 << USBS0) |// stop Bits -0 = 1bit 1 = 2bit
		(1 << UCSZ10) |// 8 Data bits
		(1 << UCSZ00) |//
		(0 << UCPOL0);// for Synch Mode only -clock polarity
}


//send some data to the serial port
void USART_tx_string( char *data ) {
	while ((*data != '\0')) {
		while (!(UCSR0A & (1 <<UDRE0)))
			UDR0 = *data;
		data++;
	}
}


unsigned char USART_Receive( void )
{
	//wait for data to be received
	while ( !(UCSR0A & (!<<RXC)))
	//get and return received data from buffer
	return UDR0;
}

void adc_init(void){
	/** Setup and enable ADC **/
	ADMUX = (0<<REFS1)|   // Reference Selection Bits
		(1<<REFS0)|  // AVcc-external cap at AREF
```

```
              (0<<ADLAR)| // ADC Left Adjust Result
              (0<<MUX2)|  // ANalogChannel Selection Bits
              (1<<MUX1)|  // ADC2 (PC2 PIN25)
              (1<<MUX0);
       ADCSRA = (1<<ADEN)|// ADC ENable
              (0<<ADSC)| // ADC Start Conversion
              (0<<ADATE)| // ADC Auto Trigger Enable
              (0<<ADIF)|// ADC Interrupt Flag
              (0<<ADIE)| // ADC Interrupt Enable
              (1<<ADPS2)| // ADC PrescalerSelect Bits
              (0<<ADPS1)|
              (1<<ADPS0);   // Select Channel
}

void read_adc(void) {
       unsigned char i=4;
       adc_temp= 0;
       while (i--) {
              ADCSRA |= (1<<ADSC);
              while(ADCSRA & (1<<ADSC));
              adc_temp+= ADC;
              _delay_ms(50);
       }
       adc_temp= adc_temp/ 4;// Average a few samples
}


int main(void)
{
       adc_init(); //initialize adc
       USART_init(UBRR_9600); //initialize usart
    _delay_ms(250); //wait 0.25 seconds
       USART_tx_string("\r\nConnecteed!\r\n"); //display connected
       _delay_ms(250);      //wait 0.25 seconds

    while (1) //do this until finished or broken
    {
              USART_tx_string("Hello World!!");  //display string
              _delay_ms(1000);     //wait 1 second
              read_adc();
              snprintf(outs, sizeof(outs), "%f\r\n", adc_temp);      //print floating
              USART_tx_string(outs);
              _delay_ms(1000);     //wait 1 second
              read_adc();
              snprintf(outs, sizeof(outs), "%3d\r\n", adc_temp);     //print integer
              USART_tx_string(outs);
              _delay_ms(1000);
    }
}
```

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

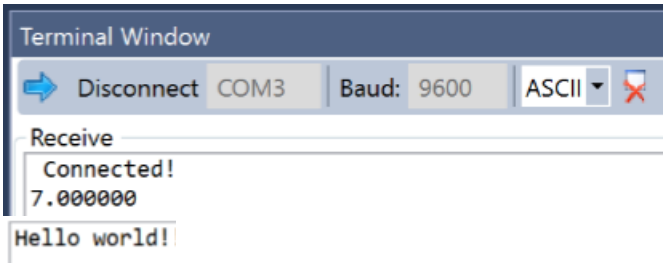
N/A

4. SCHEMATICS

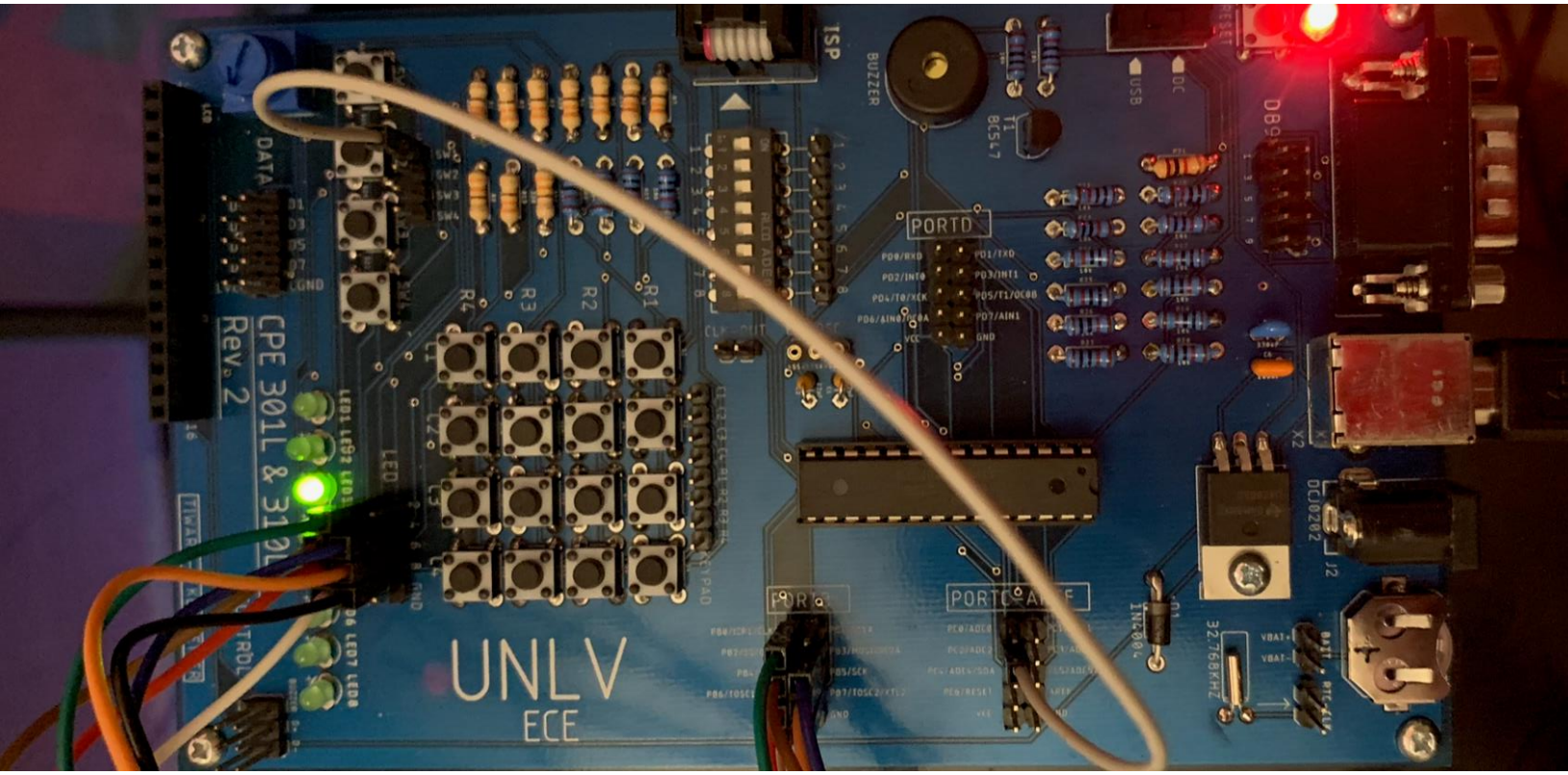
N/A

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

Task 1:



6. SCREENSHOT OF EACH DEMO (BOARD SETUP)



7. VIDEO LINKS OF EACH DEMO

N/A

8. GITHUB LINK OF THIS DA

https://github.com/tylergardenhire/submission_projects.git

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

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