Design Assignment 4

Student Name: Johnathan Widney
Student #: 5000354900
Student Email: widney@unlv.nevada.edu
Primary Github address: https://github.com/JackOfSpades-7/UNLV-Embeded-Systems
Directory: https://github.com/JackOfSpades-7/UNLV-Embeded-Systems/tree/main
Video Playlist:
https://www.youtube.com/playlist?list=PLoASw0sToF2VSxvcRVV4onrNnoOzf-Dnl

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).
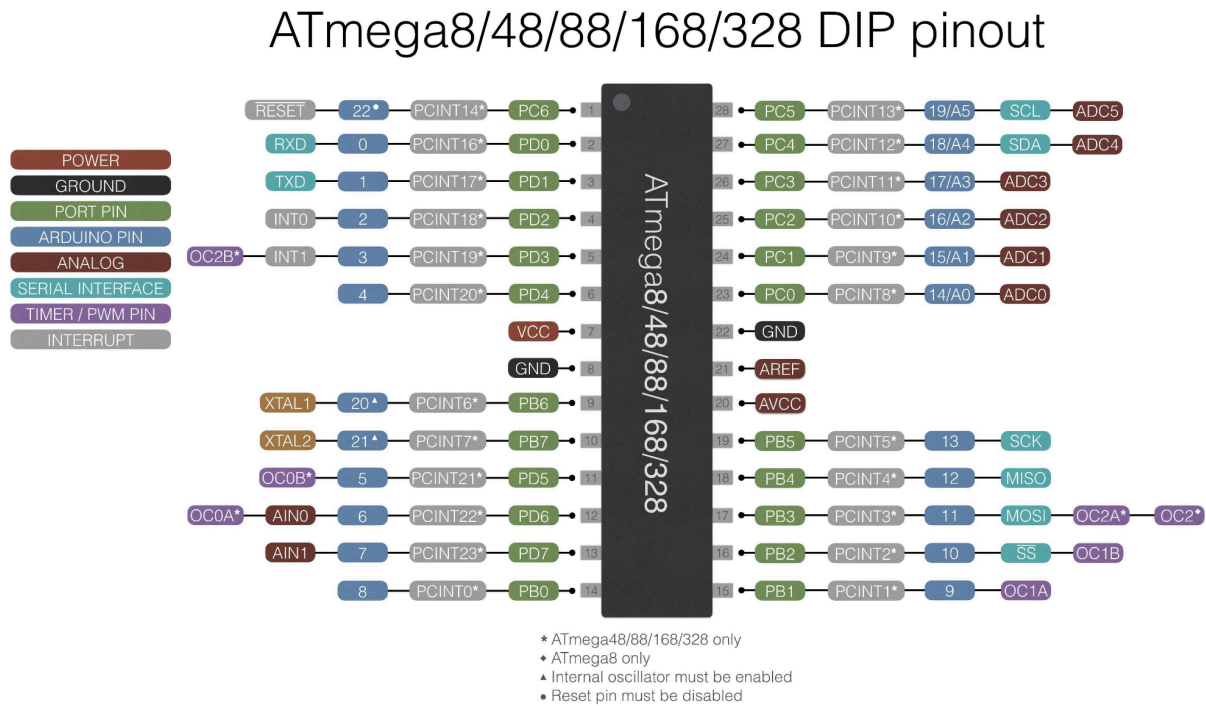
1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

List of Components used
Block diagram with pins used in the Atmega3PB (only)
- Atmega328PB Xplained mini microcontroller board
- Arduino compatible external multifunction development shield
- Male-to-male jumper cables
- Logic analyzer
- Female-to-female ribbon cable
- PC

Block diagrams and pins:

Atmega328PB Micro controller:

Arduino compatible multifunction development shield:

Arduino Header

BlueTooth Module or EXT serial-USB Module

GV - TR - -

PB5,PB4,PB3,PB2

1=PD1=TXD
0=PD0=RXD

UART

For assignments dealing with LED - use the pins PB5,PB4,PB3,PB2. For assignments dealing with switches, pin interrupts use pins PC1,PC2,PC3
For assignments in PWM use ~5/PD5 (T0B), ~6/PD6 (T0A), ~9/PB1 (T1A), ~10/PB2 (T1B-LED), ~11/PB3(T2A-LED),~3/PD3(T2B/Buzzer)
For assignments with analog input use A0/PC0-Potentiometer, A4/PC4/LM3X, or EXT @ PC5. PD2 is INT0 pin (external interrupt)

2.   INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1

```c
/* This code contains all of task 1 */

#define F_CPU 16000000UL
#include<avr/io.h>
#include<util/delay.h>
#define BAUD 9600
#define MYUBRR F_CPU/16/BAUD-1
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <avr/interrupt.h>

int adcResult; // global integer for the adc result

void transmit_str(const char *str);
void USART_Init(unsigned int ubrr);
void usart_print(char* str);
void USART_putstring(char* StringPtr);
```

```c
char out_str[51];

int main (void)
{
        memset(out_str, ' ', 49);
        out_str[50] = '\0';
        ADC_Init(); // Initialize ADC
        DDRB = 0xFF; // Portb as outputs
        sei(); // Enable Global interrupts

        USART_Init(MYUBRR); //Initializing USART

        while (1) {
                ADCSRA |= (1<<6); // start conversion
                transmit_str("* \n"); // Print star to the serial terminal
                _delay_ms(100); // Delay 0.1s before printing again
        }

        return 0;
}

ISR (TIMER1_OVF_vect)
{
        ADCSRA |= (1 << ADSC);
        while((ADCSRA&(1<<ADIF))==0);

        ADCSRA |= (1<<ADIF);

        int a = ADCL;
        a = a | (ADCH<<8);
        int8_t index = a/20.48;
        out_str[index] = ' ';
        TCNT1 = 49911;
}

void ADC_Init() {
        ADMUX = 0b01100000; //AVCC reference voltage (5V), ADC0 input, result left adjusted
        ADCSRA = 0b10001111; //ADC enabled, interrupt enabled, prescaler 128
        ADCSRB = 0x00; //free running mode
}


ISR(ADC_vect) //ADC conversion complete interrupt
{
        adcResult = ADCH; // outputting to 8 LEDS so only taking top 8 bits
        PORTB = adcResult; // outputting ADC result to LEDs
}

void USART_Init(unsigned int ubrr)
{
        // Set baud rate
        UBRR0H = (unsigned char)(ubrr>>8);
```
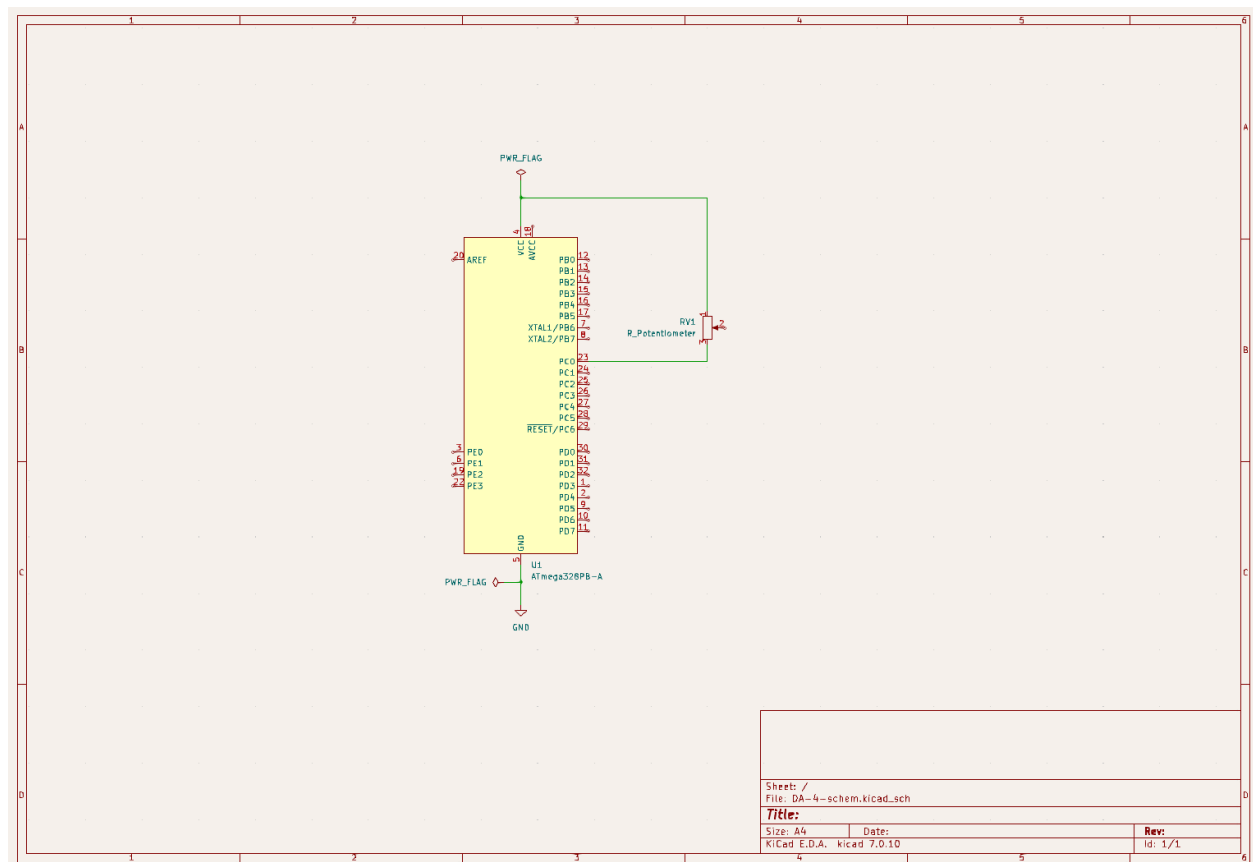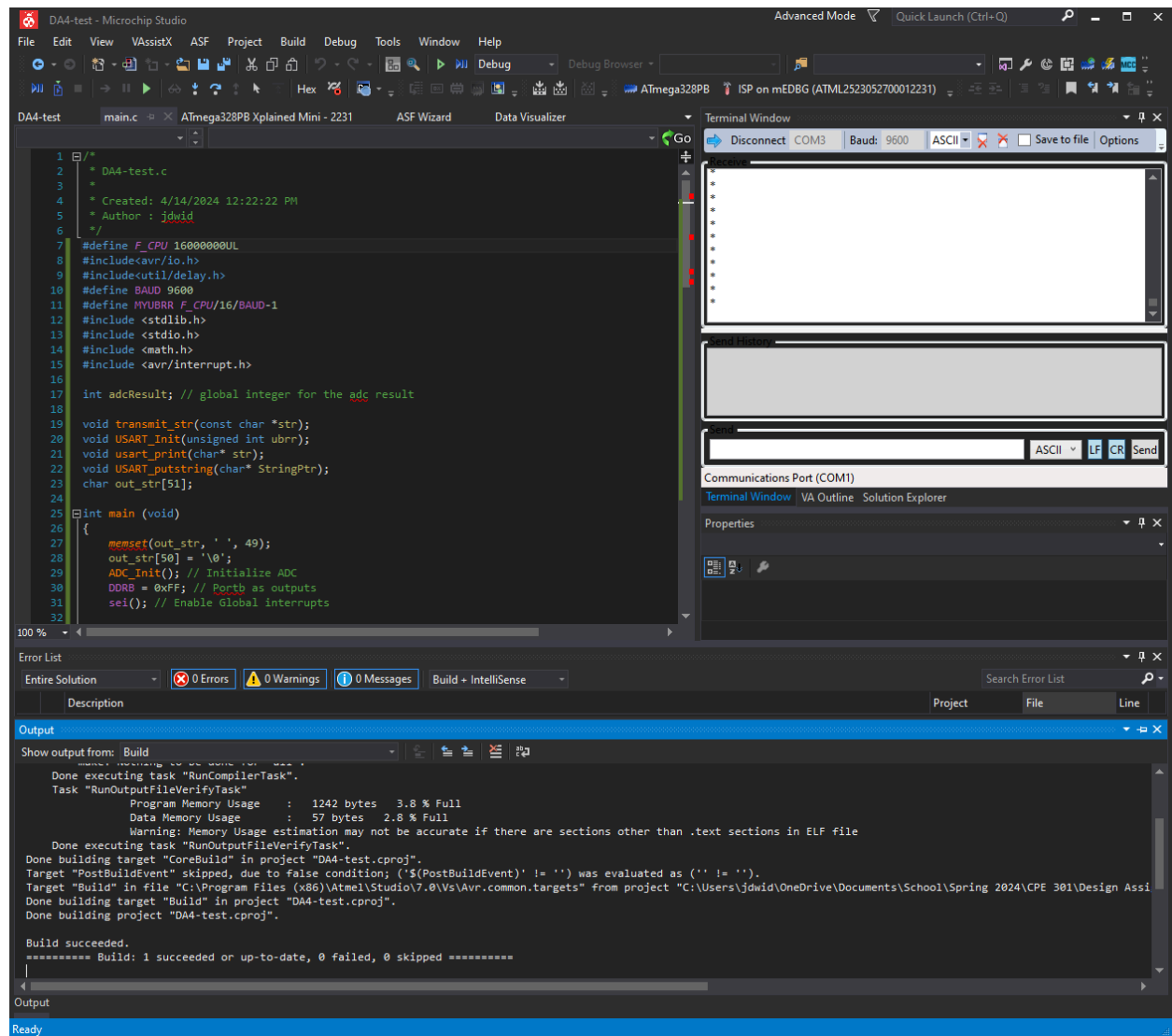
```
        UBRR0L = (unsigned char)ubrr;
        // Enable receiver and transmitter
        UCSR0B = (1<<RXEN0)|(1<<TXEN0);
        // Set frame format: 8data, 1stop bit
        UCSR0C = (1 << UCSZ01) | (1 << UCSZ00);
}


void transmit_str(const char *str)
{

        while(*str) // writes each character of string until the string ends
        {
                while (!(UCSR0A & (1<<UDRE0))); // wait for transmit buffer
                UDR0 = *str; // current character transferred to data register
                ++str; // Increment character pointer
        }

}
```

3.   SCHEMATICS

Use KICAD schematics only (not required for DA1 simulation)

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

Task 1:

5.   SCREENSHOT OF EACH DEMO (BOARD SETUP)

Task 1:
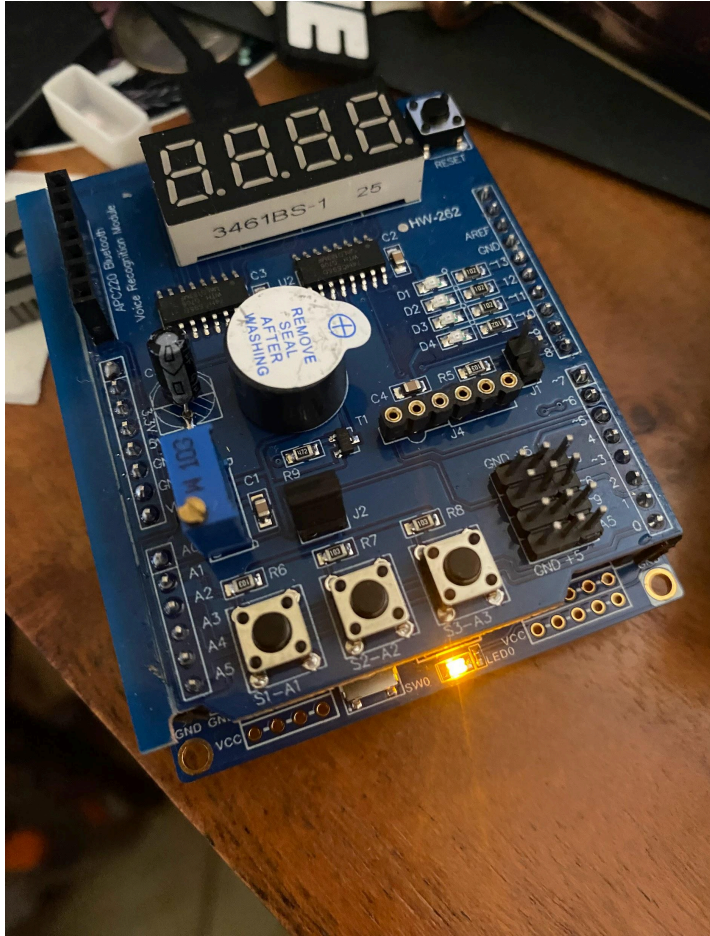
6.   VIDEO LINKS OF EACH DEMO

Task 1: https://youtu.be/dLk8oB640_M

7.   GITHUB LINK OF THIS DA

Task 1:
https://github.com/JackOfSpades-7/UNLV-Embeded-Systems/tree/main/Design%20Assignment%204

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