CPE301 – SPRING 2024

Design Assignment 3

Student Name: Abraham Garcia
Student #: 5005262049
Student Email: garci11@unlv.nevada.edu
Primary Github address: https://github.com/SON-Abe/submission_da.git
Directory: submission_da/Design_Assignments/DA3
Video Playlist: DA3

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**

- Microchip Studio Debugger
- Microchip Studio Simulator
- Female-to-Male Wire
- ATmega328PB Microcontroller
- Logic Analyzer
- Saleae Logic Analyzer Software

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

TASK 1:

```
/*
 * main.c
 *
 * Created: 3/22/2024 5:15:35 PM
 *  Author: abrah
 */
#define F_CPU 16000000UL
#include<avr/io.h>
#include<util/delay.h>

int main(void)
{
    int counter = 0;                    // INITIALIZE COUNTER
    DDRB |= 0X20;                        // SET PB5 AS AN OUTPUT
    TCCR0A = 0X00;                       // NORMAL OPERATION
    TCNT0 = 231;                        // TIMER COUNTER 0 SET TO 0 ->
231
    TCCR0B |= 0X03;                      // PRESCALE 64
    while(1)
    {
        while((TIFR0 & 0X01) == 0);      // WAIT FOR THE OVERFLOW EVENT
        if(counter == 10000)            // WHEN COUNTER REACHES 1 SECOND
        {
            PORTB ^= (1 << DDB5);           // TOGGLE PORTB LED
            counter = 0;                // RESET COUNTER
        }
        counter ++;                      // INCREMENT COUNTER
        TCNT0 = 231;                    // RESET TCNT0
        TIFR0 = 0X01;                    // CLEAR FLAG OVERFLOW
```

```
    }
    return 0;
}
```

TASK 2:

```
#define F_CPU 16000000UL
#include <avr/io.h>
#include <avr/interrupt.h>

ISR(TIMER1_COMPA_vect)                 // TIMER1 ISR CTC MODE
{
    PORTB ^= 0X10;                     // TOGGLE  PORTB4
}

int main(void)
{
    DDRB |= 0x10;                     // PB4 AS OUTPUT
    TCCR1A |= 0X00;                    // SET TCCR1A REGISTER TO 0
    TCCR1B |= 0X0D;                    // SET TIMER1 TO CTC MODE &
PRESCALE 1024
    OCR1A = 49152;                     // INITIALIZE COMPARE VALUE
    TIMSK1 |= 0X02;                    //ENABLE TIMER/COUNTER 1 OUTPUT
COMPARE A MATCH INTERRUPT
    sei();                             // ENABLE GLOBAL INTERRUPT
    while(1)
    {
        if(TIFR1 & (1 << TOV1))        // CHECK IF TIMER 1 OVERFLAG IS 1
        TIFR1 |= (1 << TOV1);    // CLEAR OVERFLOW FLAG
    }
}
```

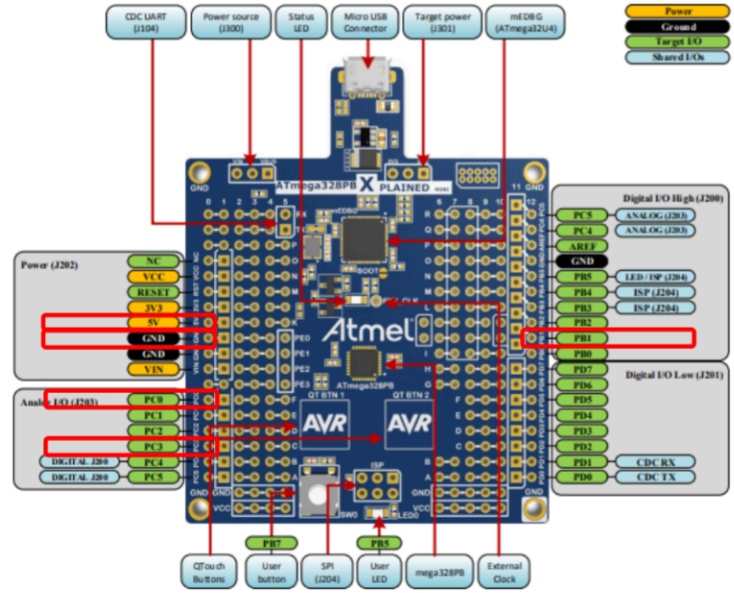
TASK 3:

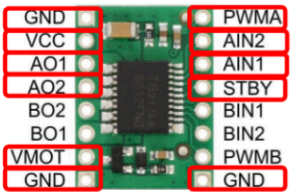
```
#define F_CPU 16000000UL
#include <avr/io.h>
#include <avr/interrupt.h>

ISR(TIMER2_OVF_vect)
{
    static uint16_t counter = 0;    // STATIC COUNTER HOLDS VALUE BETWEEN
ISR
```

```
    counter++;                          // INCREMENT COUNTER
    if(counter == 2000)                 // IF COUNTER REACHES 2 SECONDS
    {
        PORTB ^= 0x08;                  // TOGGLE PORTB LED
        counter = 0;                   //RESET COUNTER
    }
}

int main(void)
{
    DDRB |= 0X08;                      // PB3 IS AN OUTPUT
    TCCR2A |= 0X00;                     // SET REGISTER TCCR2A
    TCCR2B |= 0X04;                     // SET PRESCALAR TO 64 AND STARTS
PWM
    TIMSK2 |= 0X01;                     // SET INTERRUPT ON OVERFLOW IN
TIMER/COUNTER 2
    TCNT2 = 192;                       //  TIMER COUNTER 0 SET TO 0 ->  192
    sei();                              // ENABLE GLOBAL INTERRUPTS
    while(1);                          // INTERRUPT HANDLES LED
    return 0;
}
```

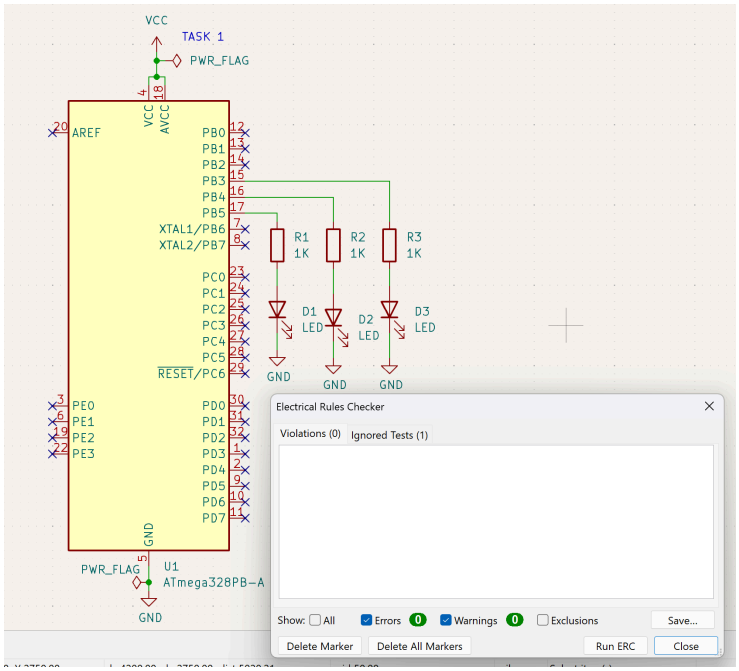
3. **DEVELOPED/MODIFIED CODE OF TASK 3/A**

4. **SCHEMATICS**





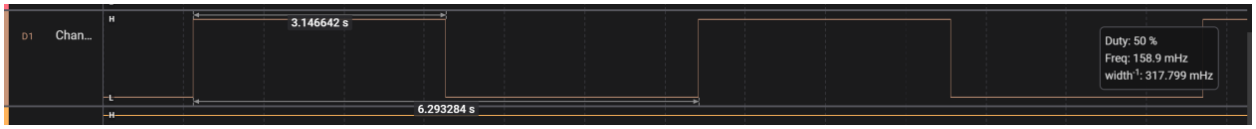
TASK 1:



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

TASK 1:

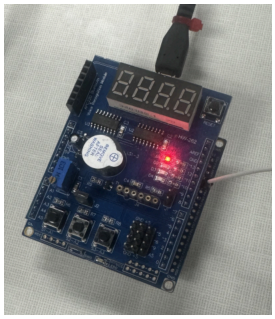
TASK 2:



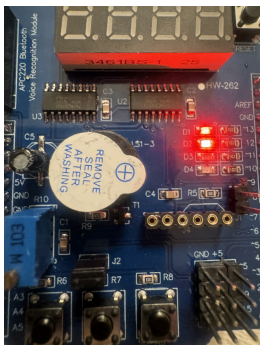
TASK 3:

6. SCREENSHOT OF EACH DEMO (BOARD SETUP)

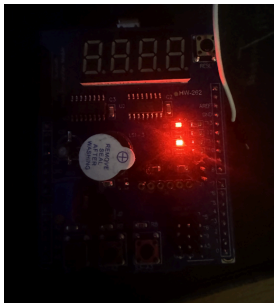
Task 1:



Task 2:



Task 3:



7. VIDEO LINKS OF EACH DEMO

DA3_1 DA3_2 DA3_3

8. **GITHUB LINK OF THIS DA**

DA3

Student Academic Misconduct Policy

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

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

Abraham Garcia