CPE301 – SPRING 2019

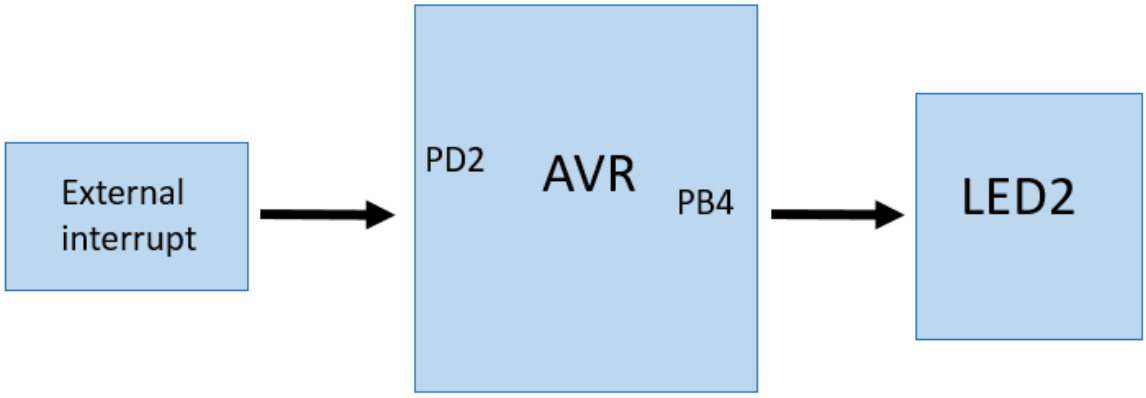
Design Assignment 2B

Student Name: Francisco Mata Carlos
Student #: 1012593607
Student Email: matacarl@unlv.nevada.edu
Primary Github address: https://github.com/chicosisco/da_sub.git
Directory: repository/cpe301/DesignAssignments/DA2B

1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

a. Atmega328p Xplained Mini
b. Multi-functional Shield
c. Atmel Studio 7

Block diagram with pins used in the Atmega328P



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

C Code

```
/* This C code creates an interruption on PORTD2 of 1.25s, and this is shown
 * by the LED2 turning off for about 1.25s. The assignment is requesting for
 * LED on PB2 to turn on, however LED on PB2 is burnt and not working, which is
 * the reason for using LED on PB4
 * DA2B_C.c
 *
 * Created: 3/9/2019 12:50:59 AM
 * Author : Francisco Mata Carlos
 */

#define F_CPU 16000000UL // clock
#include <avr/io.h>   //header files
```

```c
#include <avr/interrupt.h>
#include <util/delay.h>

int main(void)
{

	DDRB = (1<<4) ; //PB4 as an output
	DDRB = 0XFF;
	PORTB = 0xFF;  // setting portb high, which turns led off
	PORTD = 1<<2; //pull-up activated
	EICRA = 0x2;  //make INT0 falling edge triggered

	EIMSK= (1<<INT0);  // enable external interrupt 0
	sei ();    // enable interrupt


while (1)
	{	//waiting for interruption
	}
	}
	ISR (INT0_vect)  // ISR for external interrupt 0
{
	PORTB ^= (1<<4);   //toggle PORTB4
	_delay_ms(1250);   // 1.25s delay
}
```

Assembly Code

```
;
; DA2B_Assem.asm
;
; Created: 3/9/2019 2:06:22 AM
; Author : Francisco Mata Carlos
;

.include <m328pdef.inc>
.ORG 0  ;location for reset
JMP MAIN
.ORG 0x02 ;location external interrupt 0
JMP EX0_ISR
MAIN:
	SBI DDRB, 5 ;setting PB5 as output
	SBI PORTB, 5 ;settng PB5 high so led is off
	LDI R20, HIGH(RAMEND)
	OUT SPH, R20
	LDI R20, LOW(RAMEND)
	LDI R20, 0X2  ;initializing stack

	LDI R20, 0X2 ;make INT0 falling edge triggered
	STS EICRA, R20
	SBI DDRB, 4  ;set PORTB4 as output
	SBI PORTB, 4 ;set PORTB4 high to keep led off until interrupt

	SBI PORTD, 2 ;pull-up activated
	LDI R20, 1<<INT0 ; enable INT0
	OUT EIMSK, R20 ;masking specific bit
```

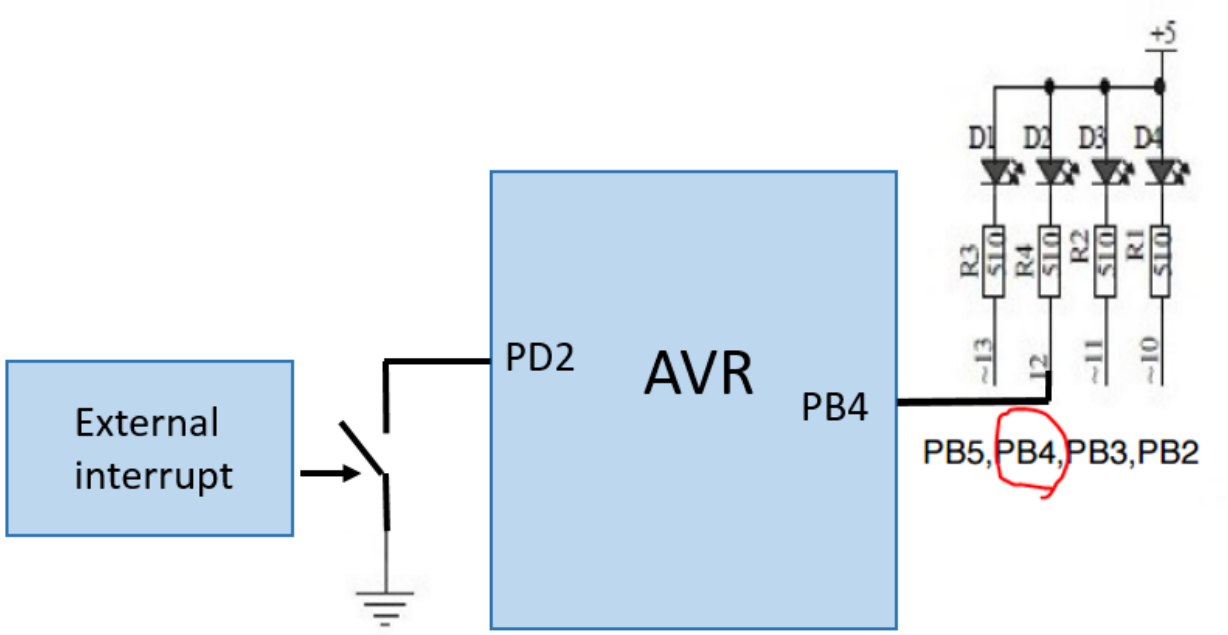
```
       SEI ; enable interrupts

HERE: JMP HERE

EX0_ISR:
       IN R21, PORTB ;PORTB an input
       LDI R22, (1<<4)  ;00100000
       EOR R21, R22 ;toggl led
       OUT PORTB, R21 ;turn on led
       rcall delay_1s
       rcall delay_100ms
       rcall delay_100ms
       rcall delay_50ms
       RETI ;return to next instruction after interrupt

;=================================================================================================
; Below is the time delay subroutine used for the code above
;
;=================================================================================================
delay_1s:
       rcall delay_100ms
       rcall delay_100ms
       rcall delay_100ms
       rcall delay_100ms
       rcall delay_100ms
       rcall delay_100ms
       rcall delay_100ms
       rcall delay_100ms
       rcall delay_100ms
       rcall delay_100ms
       ret

delay_100ms:
       rcall delay_10ms
       rcall delay_10ms
       rcall delay_10ms
       rcall delay_10ms
       rcall delay_10ms
       rcall delay_10ms
       rcall delay_10ms
       rcall delay_10ms
       rcall delay_10ms
       rcall delay_10ms
       ret

delay_50ms:
    rcall delay_10ms
       rcall delay_10ms
       rcall delay_10ms
       rcall delay_10ms
       rcall delay_10ms
       ret

delay_10ms:
       rcall delay_1ms
       rcall delay_1ms
```

```
        rcall delay_1ms
        rcall delay_1ms
        rcall delay_1ms
        rcall delay_1ms
        rcall delay_1ms
        rcall delay_1ms
        rcall delay_1ms
        rcall delay_1ms
        ret

delay_1ms: ;1ms per loop
        rcall delay_1ms_16
        rcall delay_1ms_16
        rcall delay_1ms_16
        rcall delay_1ms_16
        ret

delay_1ms_16:
        push r16 ;save the value in r16
        ldi  r16, 249 ;3984 cycles
delay_1a:
        nop    ;1 cycle
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        dec  r16
        brne delay_1a ;2 cycles
        pop  r16 ;restore the value in r16
        ret
```

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

Same as above

4. SCHEMATICS



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

TASK1A_Code

1. **Implement Design Assignment 2A.2 using INT0 (PD2 pin) interrupt mechanism.**

The snapshot below shows the time delay of about 1.25s. A time delay line was used above the while loop in order to inspect the delay being used.

```
 * Created: 3/9/2019 12:50:59 AM
 * Author : Francisco Mata Carlos
 */

#define F_CPU 16000000UL // clock
#include <avr/io.h>   //header files
#include <avr/interrupt.h>
#include <util/delay.h>

int main(void)
{

    DDRB = (1<<4) ; //PB4 as an output
    DDRB = 0XFF;
    PORTB = 0xFF;  // setting portb high, which turns led off
    PORTD = 1<<2; //pull-up activated
    EICRA = 0x2;  //make INT0 falling edge triggered
    _delay_ms(1250);   // 1.25s delay
    EIMSK= (1<<INT0);  // enable external interrupt 0
    sei ();    // enable interrupt


while (1)
    {        //waiting for interruption
    }
    }
    ISR (INT0_vect)  // ISR for external interrupt 0
{
    PORTB ^= (1<<4);   //toggle PORTB5
    _delay_ms(1250);   // 1.25s delay
}
```

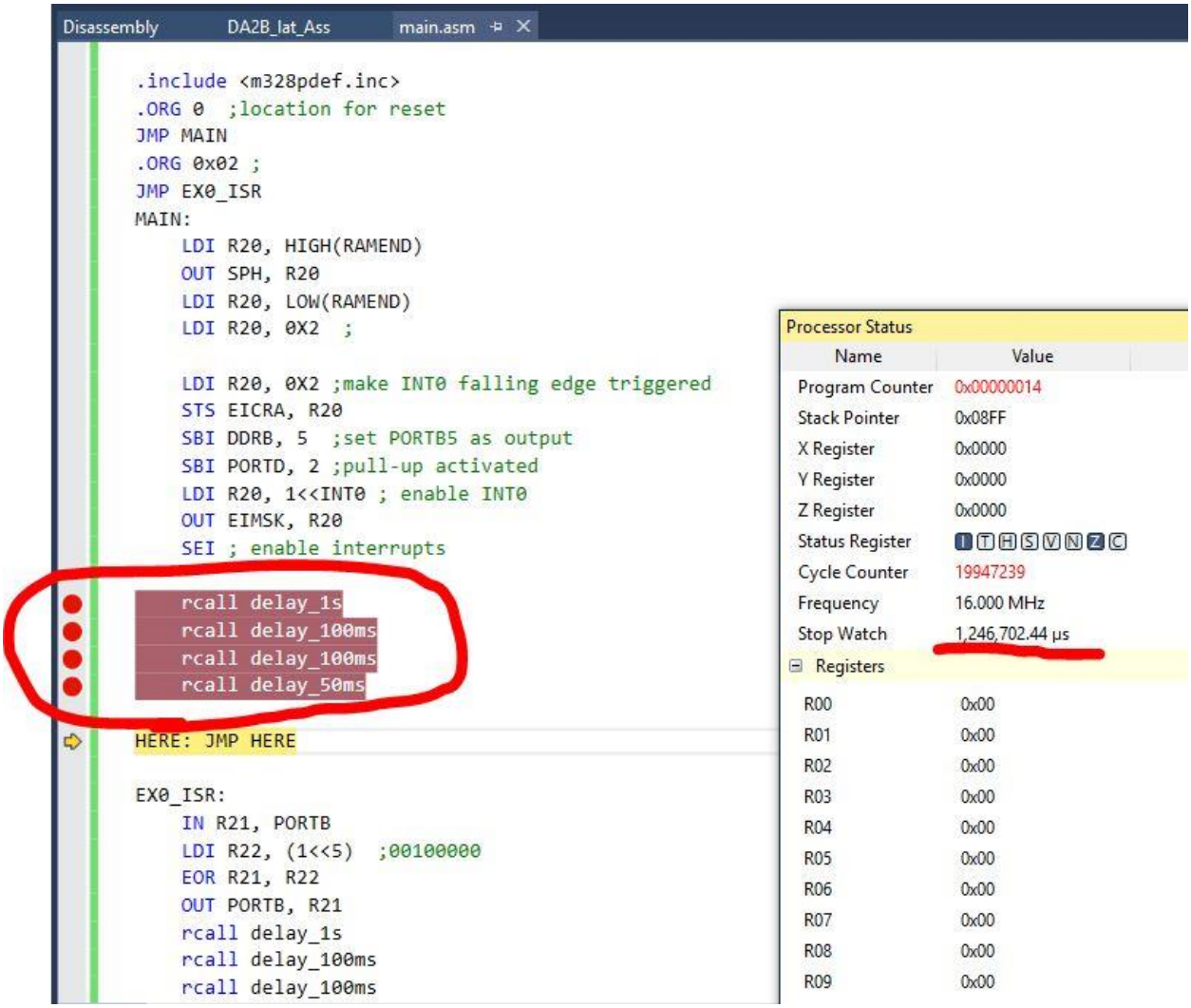
Processor Status

Name	Value
Program Counter	0x00000053
Stack Pointer	0x08FD
X Register	0x0000
Y Register	0x08FF
Z Register	0x0000
Status Register	I T H S V N Z C
Cycle Counter	20000022
Frequency	16.000 MHz
Stop Watch	1,250,001.38 µs
Registers	
R00	0x00
R01	0x00
R02	0x00
R03	0x00
R04	0x00
R05	0x00
R06	0x00
R07	0x00
R08	0x00

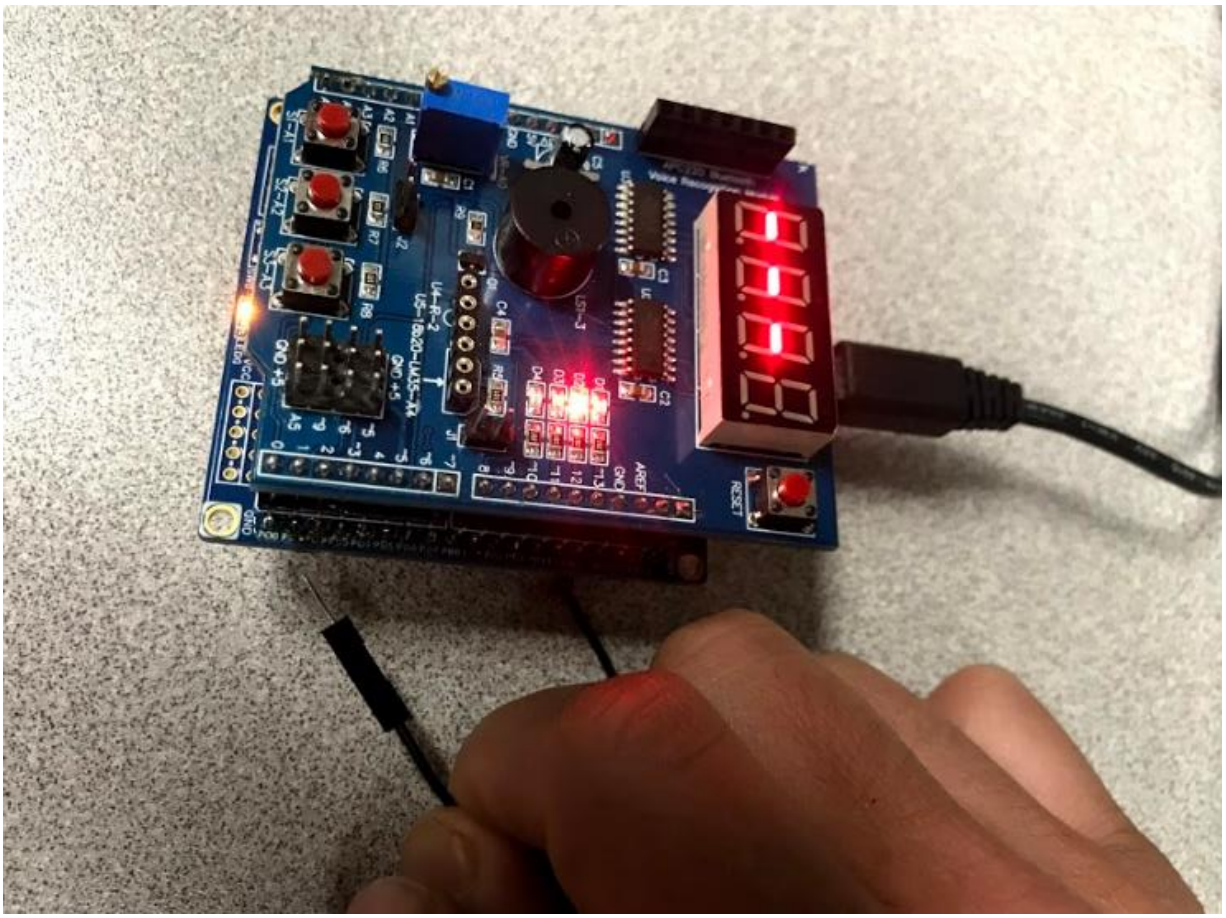
TASK1B_Assembly

2. **Implement Design Assignment 2A.2 using INT0 (PD2 pin) interrupt mechanism**.

The snapshot below shows the time delay of about 1.25s. A subroutine time-delay was placed before the while loop in order to inspect the delay being used.



6. SCREENSHOT OF EACH DEMO (BOARD SETUP)



7. VIDEO LINKS OF EACH DEMO

Assembly Code

https://youtu.be/YsqnW_ZT4O8

C Code

https://youtu.be/PUeKfl3emqU

8. GITHUB LINK OF THIS DA

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

"*This assignment submission is my own, original work*".
NAME OF THE STUDENT