CPE301 – SPRING 2019

Design Assignment 2B

Student Name: Ron Joshua Recrio
Student #: 5003825419
Student Email: recrio@unlv.nevada.edu
Primary Github address: https://github.com/recrio/submissions
Directory: /DesignAssignments/DA2B

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).

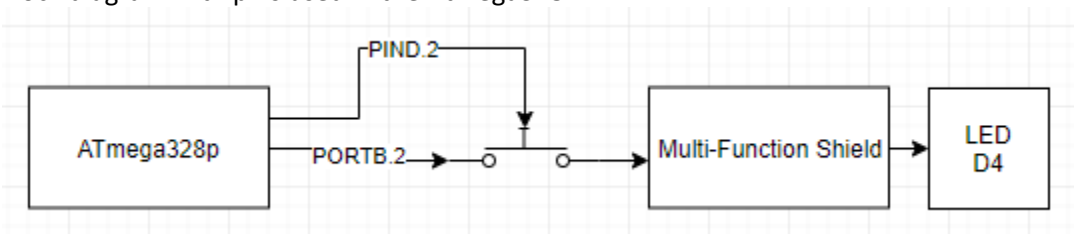
1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

List of Components used:

ATmega328p Xplained Mini

Multifunction Shield

Block diagram with pins used in the Atmega328P



2. INITIAL CODE OF TASK 2/A

```
.ORG 0
init:
 CBI DDRC, 2          ; PC2 as input, Clears DDRC2
 LDI R16, 4           ; PB2
 OUT DDRB, R16        ; SET PB2 as output
 LDI R17,0            ; toggle for PC2
 LDI R20,5            ; 0b101 for prescaler 1024
 STS TCCR1B, R20      ; Set prescaler to 1024
 LDI R20, 0           ; For counter/timer reset
 EOR R17, R16         ; Toggle with XOR so that it starts as OFF

main:
 OUT PORTB, R17       ; Turn off LED
 SBIC PINC, 2         ; Skip if PC2 is pressed
 RJMP keep_checking   ; if not then jump

 EOR R17, R16         ; Toggle again
 OUT PORTB, R17       ; Turn on LED
 EOR R17, R16         ; Toggle back
 STS TCNT1H, R20      ; Reset
 STS TCNT1L, R20      ; Timer/Counter

 RCALL delay          ; call delay of 1.25s

 RJMP main            ; Loop again

keep_checking:
 SBI PORTB, 2         ; PB2 = 1, set it high so it will not light up
 RJMP main            ; Loop back to check for button press

delay:                ; TCNT1 should be 19,531 or 0x4C4B
 LDS R27, TCNT1H      ; high byte of timer/counter
 LDS R26, TCNT1L      ; low byte of timer/counter
 CPI R26,0x4B         ; check if low byte matches

 BRSH check_high      ; if matching, then check high byte

 RJMP delay           ; else keep checking

check_high:
 CPI R27,0x4C         ; check high byte
 BRLT delay           ; if not matching , keep checking
 RET                  ; if timer/counter is finally matching, return
```

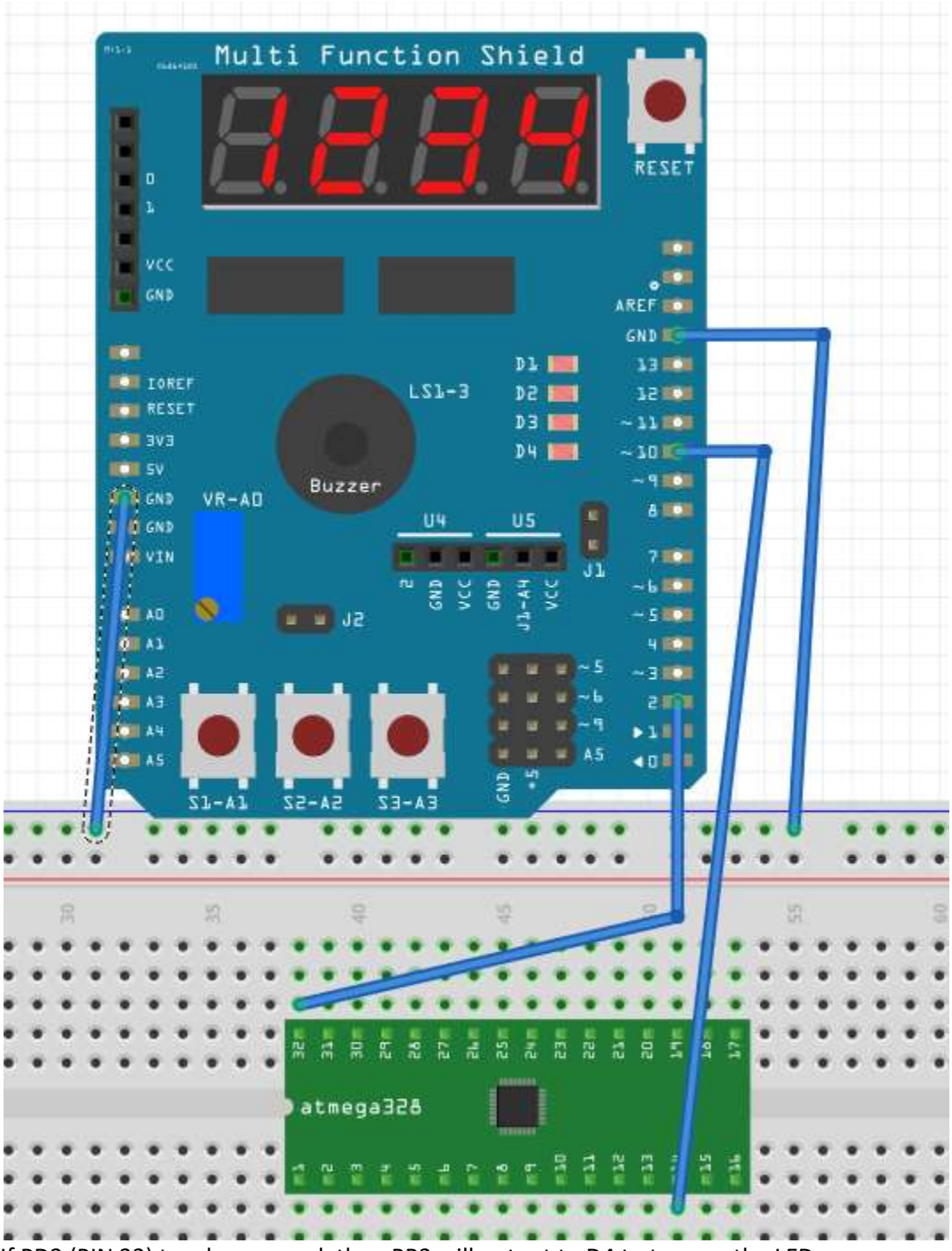
3. DEVELOPED MODIFIED CODE OF TASK 1/B from TASK 2/A

```
.include <m328pdef.inc>
.ORG 0 ; main code location
JMP MAIN
.ORG 0x02 ; interrupt 0 code location
JMP EX0_ISR

MAIN:
       // Initialize stack pointer
       LDI R20, HIGH(RAMEND)
       OUT SPH, R20
       LDI R20, LOW(RAMEND)
       OUT SPL, R20

       LDI R22, (1<<2) ; for toggling PB2

       SBI PORTD, 2  ; Pull-UP activated
       LDI R20, 0x00 ; setting to make
       STS EICRA, R20       ; INT0 low generate an interrupt
       SBI DDRB, 2          ; PORTB.2 set to output
       LDI R20, 1<<INT0; enable
       OUT EIMSK, R20       ; INT0
       SEI                          ; enable global interrupt
HERE:
       OUT PORTB, R22       ; Turn off LED
       JMP HERE             ; infinite loop to catch interrupts

EX0_ISR:
       IN R21, PORTB ; Take in status of PORTB
       EOR R21, R22  ; Toggle it
       OUT PORTB, R21       ; Output to LED
       RCALL DELAY          ; Call a delay of 1.25s
       RETI                 ; Return with interrupt flag on

DELAY:                      ; Delay adds up to 20 million clock cycles
    LDI  r18, 102
    LDI  r19, 118
    LDI  r20, 194
L1: DEC  r20
    BRNE L1
    DEC  r19
    BRNE L1
    DEC  r18
    BRNE L1
       RET
```

4. VERIFICATION OF TASK 1/B USING C

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

int main(void)
{
    DDRB |= (1<<2); // PB2 set to output
       PORTB |= (1<<2); // PB2 set to HIGH which means LED OFF
       DDRD &= (0<<2);      // PD2 set to input
       PORTD |= (1<<2); // PD2 Turn Up Resistor

       EICRA = 0x00; //Low mode
       EIMSK = 1<<INT0; // Masking
       sei(); // Turn on global interrupt

    while (1)
    {
              PORTB |= (1<<2); // Turn off LED
    }
}

ISR (INT0_vect)
{
       PORTB ^= (1<<2); // Toggle PB2
       _delay_ms(1250); // Delay for 1.25s
}
```

5. SCHEMATICS

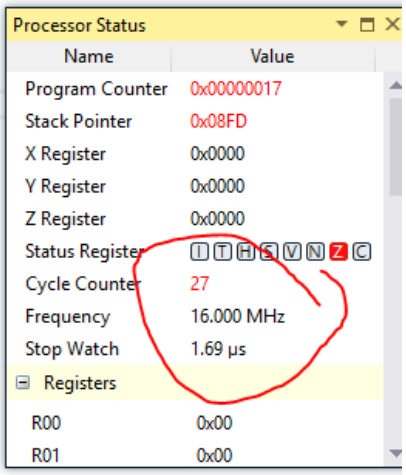


If PD2 (PIN 32) touches ground, then PB2 will output to D4 to turn on the LED

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

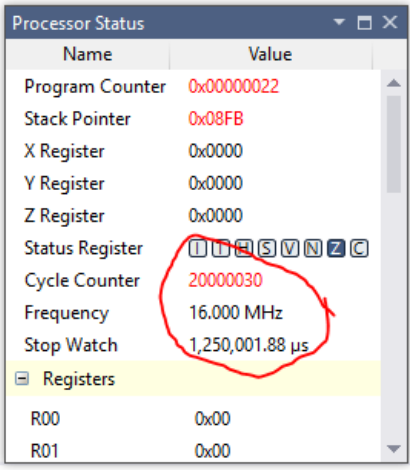
Before:

```
EX0_ISR:
    IN R21, PORTB    ; Take in status of PORTB
    EOR R21, R22     ; Toggle it
    OUT PORTB, R21   ; Output to LED
    RCALL DELAY      ; Call a delay of 1.25s
    RETI             ; Return with interrupt flag on

DELAY:               ; Delay adds up to 20 million clock cycles
    LDI  r18, 102
    LDI  r19, 118
    LDI  r20, 194
L1: DEC  r20
    BRNE L1
    DEC  r19
    BRNE L1
    DEC  r18
    BRNE L1
    RET
```



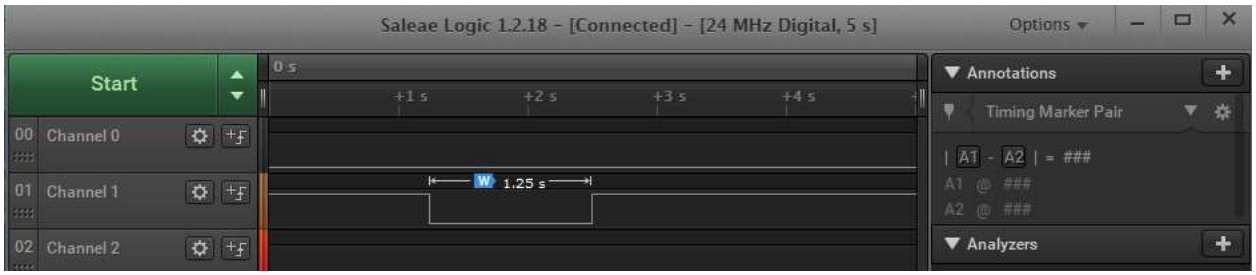
After:

```
EX0_ISR:
    IN R21, PORTB    ; Take in status of PORTB
    EOR R21, R22     ; Toggle it
    OUT PORTB, R21   ; Output to LED
    RCALL DELAY      ; Call a delay of 1.25s
    RETI             ; Return with interrupt flag on

DELAY:               ; Delay adds up to 20 million clock cycles
    LDI  r18, 102
    LDI  r19, 118
    LDI  r20, 194
L1: DEC  r20
    BRNE L1
    DEC  r19
    BRNE L1
    DEC  r18
    BRNE L1
    RET
```

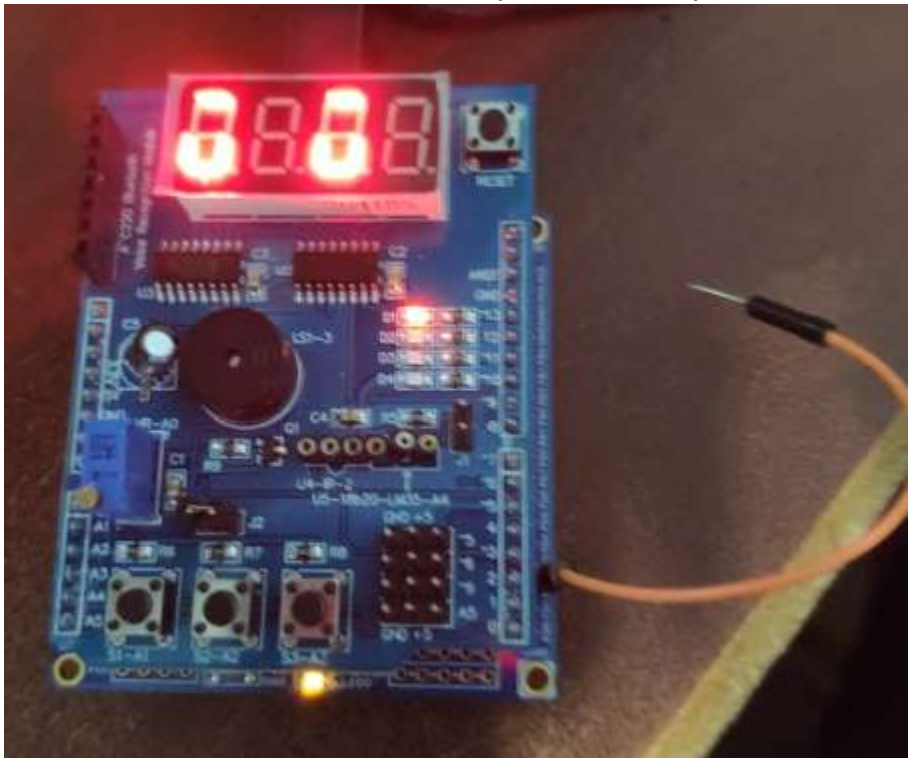


Using Logic Analyzer :

Channel 1 is connected to PB2



7. SCREENSHOT OF EACH DEMO (BOARD SETUP)



8. VIDEO LINKS OF EACH DEMO

https://youtu.be/l4KSmy9SAN4

9. GITHUB LINK OF THIS DA

https://github.com/recrio/submissions/tree/master/DesignAssignments/DA2B

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

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