

# Design Assignment 2A

---

Student Name: Rocky Yasuaki Gonzalez

Student #: 5003229733

Student Email: gonzar14@unlv.nevada.edu

Primary Github address: <https://github.com/rockyg1995/ihswwppdar.git>

Directory: C:\Users\rocky\Documents\CpE 301+L - Embedded Systems Design\CpE  
301\Repository\DesignAssignments\DA2A

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

Multifunction Shield  
Atmega328PB Xplained Mini  
Green LED  
Micro USB Cable (Power Supply)

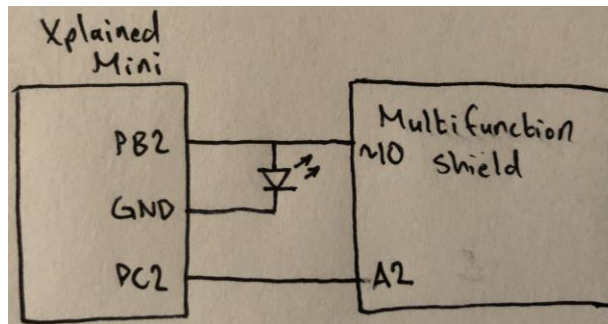


Figure 1 – Block Diagram/Pin Connections

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

```
; DA2A_T1.asm
;
; Created: 3/1/2019 10:08:49 PM
; Author: rocky

.ORG 0

    LDI R16, (1<<2)      ; Load '0000 0100' into R16
    OUT DDRB, R16        ; Set PB2 as an Output
    OUT PORTB, R16       ; Set PB2 as 'High'

main:
    ; Loop LED Toggle, Period of 735ms (60% High, 40% Low)
    CALL delay435ms      ; Subroutine to Delay program 435ms
    CBI PORTB, 2         ; Toggle LED 'Low' at PORTB.2
    CALL delay290ms      ; Subroutine to Delay program 290ms
    SBI PORTB, 2         ; Toggle LED 'High' at PORTB.2
    RJMP main            ; Repeat LED Squarewave Period (725ms)

delay435ms:
    ; Delay Program '290ms'
    Push R16
    Push R17
    Push R18
    LDI R18, 15
    ;
L3:   LDI R17, 200
    ; 15
L2:   LDI R16, 232
    ; 200*15
L1:   NOP
    ; 232*200*15
    NOP
    ; 232*200*15
    NOP
    ; 232*200*15
    NOP
    ; 232*200*15
    NOP
    ; 232*200*15
```

```

        NOP                ; 232*200*15
        NOP                ; 232*200*15
        DEC R16            ; 232*200*15
        BRNE L1           ; 232*200*15
        DEC R17            ; 200*15
        BRNE L2           ; 200*15
        DEC R18            ; 15
        BRNE L3           ; 15
        Pop R18            ; Restore R18 from Stack
        Pop R17            ; Restore R17 from Stack
        Pop R16            ; Restore R16 from Stack
        RET               ; Complete Subroutine

delay290ms:                ; Delay Program '290ms'
        Push R16           ; Store R16 into Stack
        Push R17           ; Store R17 into Stack
        Push R18           ; Store R18 into Stack
        LDI R18, 10        ;
M3:      LDI R17, 200       ; 10
M2:      LDI R16, 232      ; 200*10
M1:      NOP              ; 232*200*10
        NOP              ; 232*200*10
        NOP              ; 232*200*10
        NOP              ; 232*200*10
        NOP              ; 232*200*10
        NOP              ; 232*200*10
        NOP              ; 232*200*10
        NOP              ; 232*200*10
        DEC R16           ; 232*200*10
        BRNE M1          ; 232*200*10
        DEC R17           ; 200*10
        BRNE M2          ; 200*10
        DEC R18           ; 10
        BRNE M3          ; 10
        Pop R18           ; Else, Restore R18 from Stack
        Pop R17           ; Restore R17 from Stack
        Pop R16           ; Restore R16 from Stack
        RET               ; Complete Subroutine

/*
 * DA2A_T1_C.c
 *
 * Created: 3/2/2019 2:30:06 AM
 * Author: rocky
 */

#include <avr/io.h>          // Standard AVR Library
#include <util/delay.h>      // AVR library containing _delay_ms() function

#define LED 0b00000100      // Modify LED Bit here (Currently PB2)

int main(void) {           // Main Function
    DDRB = LED;             // Set PB2 as an Output
    PORTB = LED;            // Set PB2 as 'High'

    while (1) {             // Loop LED Toggle, Period of 735ms (60% High, 40% Low)
        _delay_ms(435*16);  // Delay Program by 435ms (Multiply '16' because 16MHz)
        PORTB &= ~LED;      // Turn OFF LED (High-to-Low)
    }
}

```

```

        _delay_ms(290*16);    // Delay Program by 290ms (Multiply '16' because 16MHz)
        PORTB |= LED;        // Turn ON LED (Low-to-High)
    }
}

```

### 3. INITIAL/DEVELOPED CODE OF TASK 2/A

```

; DA2A_T2.asm
;
; Created: 3/2/2019 9:11:02 AM
; Author: rocky

```

```

.ORG 0

```

```

    LDI R16, (1<<2)          ; Load '0000 0100' into R16
    LDI R17, (0<<2)          ; Load 'xxxx x0xx' into R17
    CLR R18
    OUT DDRB, R16             ; Set PB2 as an Output
    OUT PORTB, R17            ; Set PB2 as 'Low'
    OUT DDRC, R17             ; Set PC2 as an Input
    Out PORTC, R16            ; Activate Pull-up in PC2 (resistor connected to VCC)

```

```

check:                                ; Poll: Check if PC2 is 'High' or 'Low'
    IN R18, PINC                 ; Read in PC2 to 'R18'
    ANDI R18, 0x04                ; Mask Out all bits except bit 2
    BRNE check                    ; If Bit 2 is 'Low', continue reading PC2
    SBI PORTB, 2                  ; Else, Toggle LED 'High' at PB2
    Call delay1250ms              ; Subroutine to Delay program 1250ms
    CBI PORTB, 2                  ; Toggle LED 'Low' at PB2
    RJMP check                    ; Go back to Poll

```

```

delay1250ms:                          ; Delay Program '290ms'
    Push R16                       ; Store R16 into Stack
    Push R17                       ; Store R17 into Stack
    Push R18                       ; Store R18 into Stack
    LDI R18, 32                    ;
L3:    LDI R17, 250                  ; 32
L2:    LDI R16, 250                  ; 250*32
L1:    NOP                          ; 250*250*32
        NOP                        ; 250*250*32
        NOP                        ; 250*250*32
        NOP                        ; 250*250*32
        NOP                        ; 250*250*32
        NOP                        ; 250*250*32
        NOP                        ; 250*250*32
        NOP                        ; 250*250*32
        DEC R16                     ; 250*250*32
        BRNE L1                     ; 250*250*32
        DEC R17                     ; 250*32
        BRNE L2                     ; 250*32
        DEC R18                     ; 32
        BRNE L3                     ; 32
        Pop R18                     ; Restore R18 from Stack
        Pop R17                     ; Restore R17 from Stack
        Pop R16                     ; Restore R16 from Stack
        RET                         ; Complete Subroutine

```

```

/*
 * DA2A_T2_C.c
 *
 * Created: 3/2/2019 9:50:45 AM
 * Author : rocky
 */

#include <avr/io.h>           // Standard AVR Library
#include <util/delay.h>       // AVR library containing _delay_ms() function

#define LED 0b00000100       // Modify LED bit here (PB2 in program)
#define SWITCH 0b00000100    // Modify SWITCH bit here (PC2 in program)

int main(void) {             // Main Function
    DDRC = (0<<2);           // Set/Mask PC2 as an Input
    PORTC = (1<<2);           // Activate Pull-up on PC2 (resistor connected to VCC)
    DDRB = (1<<2);           // Set/Mask PB2 as an Output
    PORTB = (0<<2);           // Set/Mask PB2 as 'Low'

    while (1) {               // Poll: Check to read if PC2 is 'High'
        if ((~PINC & SWITCH) == SWITCH) { // If 'High' Turn LED ON for 1250ms
            PORTB |= LED;      // Set/Mask PB2 as 'High'
            _delay_ms(1250*16); // Delay Program by 1250ms
            PORTB &= ~LED;     // Set/Mask PB2 as 'Low'/End Program
        }
    }
}

```

#### 4. SCHEMATICS

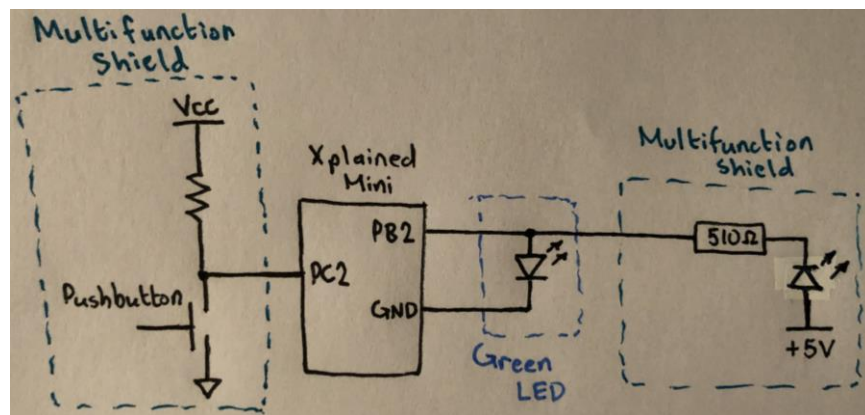
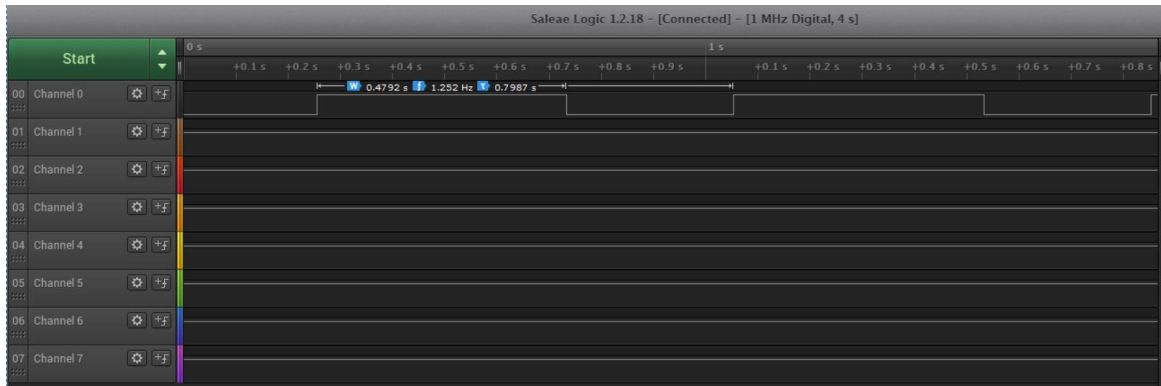


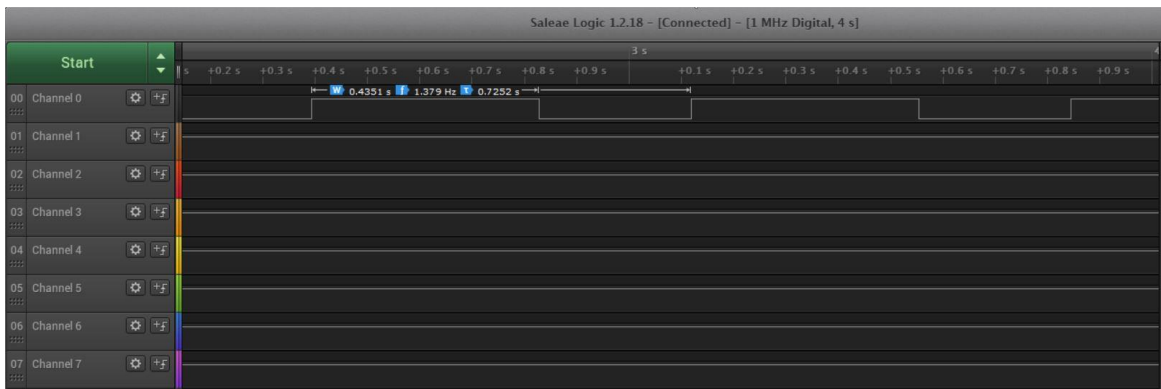
Figure 2 – Schematic Connections for Tasks 1 and 2

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

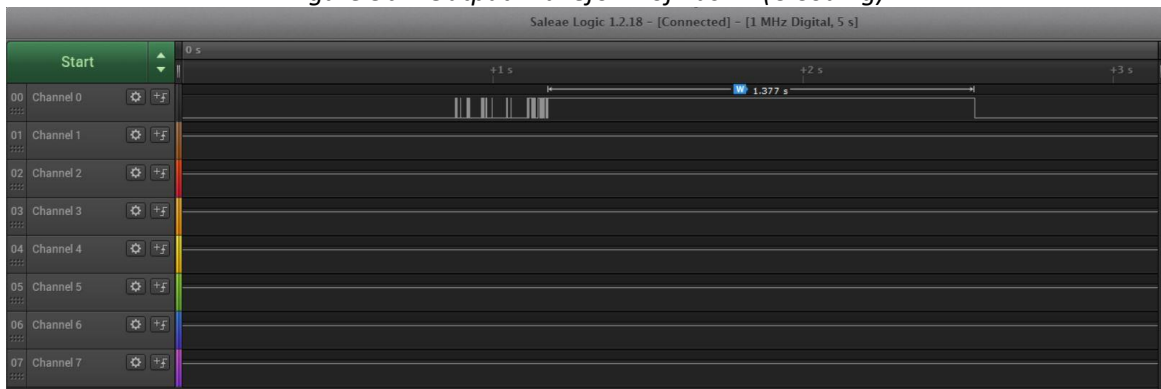
The following are the output waveforms Tasks 1 and 2 using Assembly and C Coding:



*Figure 3a – Output Waveform of Task 1 (Assembly Coding)*



*Figure 3b – Output Waveform of Task 1 (C Coding)*



*Figure 3c – Output Waveform of Task 2 (Assembly Coding)*



*Figure 3d – Output Waveform of Task 2 (C Coding)*

## 6. SCREENSHOT OF EACH DEMO (BOARD SETUP)

Each Demo utilizes the same set up which includes a Shield Attachment placed on top of the Xplained Mini PB and a green LED connected to the actual Xplained Mini PB. This set up is shown in Figure 2:

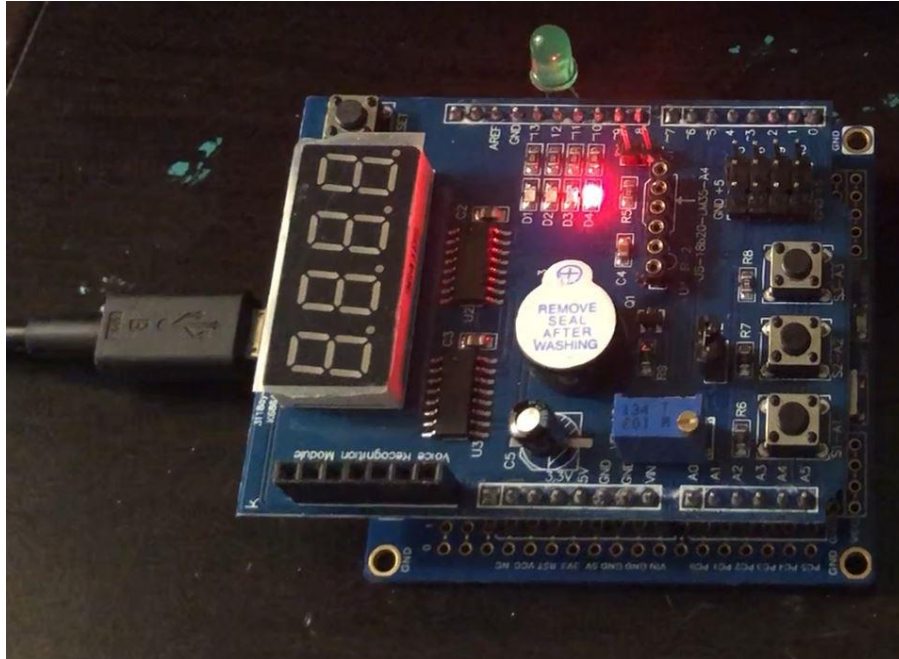


Figure 4 – Green LED and Shield Attached onto Atmega328PB Xplained Mini

## 7. VIDEO LINKS OF EACH DEMO

<https://youtu.be/oliSMltYldA>

## 8. GITHUB LINK OF THIS DA

<https://github.com/rockyg1995/ihswwppdar/tree/master/DesignAssignments/DA2A>

## Student Academic Misconduct Policy

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

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

Rocky Gonzalez