

Design Assignment 2

Student Name: Johnathan Widney

Student #: 5000354900

Student Email: widney@unlv.nevada.edu

Primary Github address: <https://github.com/JackOfSpades-7/UNLV-Embedded-Systems>

Directory:

<https://github.com/JackOfSpades-7/UNLV-Embedded-Systems/tree/main/Design%20Assignment%202>

Video Playlist:

<https://www.youtube.com/playlist?list=PLoASw0sToF2XNlstCsNQZPPxGsU5po0ET>

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

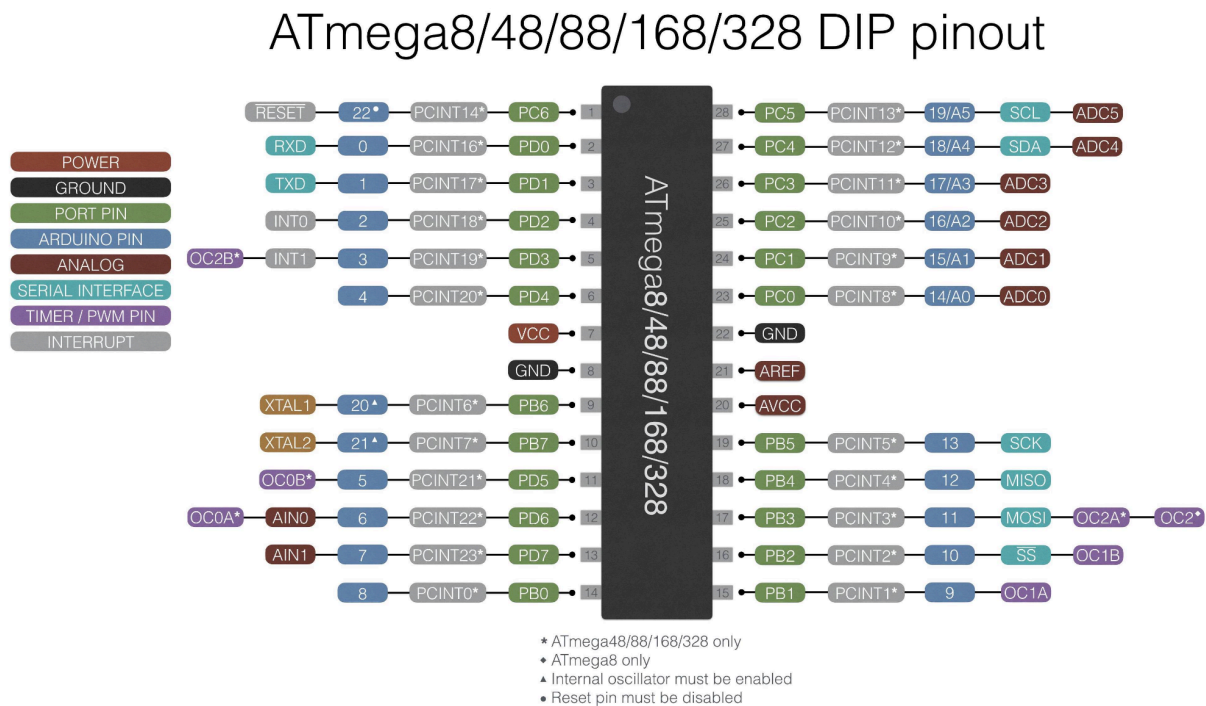
List of Components used

Block diagram with pins used in the Atmega328PB (only)

- Atmega328PB Xplained mini microcontroller board
- Arduino compatible external multifunction development shield
- Mini solderless breadboard
- 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:


```

brne DelayLoop
dec r18
brne DelayLoop
ret

```

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

The following code is the C code variant for this task.

```

#define F_CPU 16000000UL // 8 Mhz clock
#include <avr/io.h> // main libraries
#include <avr/interrupt.h> // interrupt libraries
#include <util/delay.h> // delay libraries

// short delay subroutine
void ShortDelay() {
    _delay_ms(333); // delay of 0.333 seconds
}

int main(void)
{
    DDRB |= (1 << PINB5); // set Port B5 as output
    DDRC &= (0 << PINC2); // set Port C as input
    PORTC |= (1 << PINC2); // enables pull-ups for SW2 (pushbutton)

    while (1) {
        if (!(PINC & (1 << PINC2))) {
            PORTB &= ~(1 << PINB5); // turn on PB5
            for (int i=0; i<6; i++) {
                ShortDelay(); // loop will run 6 times, results in 2 second delay
            }
        }
        else {
            PORTB |= (1 << PINB5);
        }
    }
    return 0;
}

```

The following code segment is in the AVR Assembly variant of the task.

```

.org 0x00

; defining constants
.equ F_CPU = 16000000
.equ DELAY_MS = 333

; short delay subroutine
ShortDelay:
    ldi r18, DELAY_MS/4

```

```

    ldi r19, DELAY_MS/4
    ldi r20, DELAY_MS/4
DelayLoop:
    dec r20
    brne DelayLoop
    dec r19
    brne DelayLoop
    dec r18
    brne DelayLoop
    ret

; label for main - for commenting purposes
main:
    ; set Port B5 as output
    sbi DDRB, 5

    ; set Port C2 as input and enable pull-up
    cbi DDRC, 2
    sbi PORTC, 2

Loop:
    ; check if PINC2 is low (button pressed)
    sbic PINC, 2
    rjmp ButtonReleased

    ; turn on PB5
    sbi PORTB, 5

    ; running ShortDelay (6 times for a 2-second delay)
    ldi r22, 6
DelayLoopMain:
    call ShortDelay
    dec r22
    brne DelayLoopMain

    ; button released, turn off PB5
ButtonReleased:
    cbi PORTB, 5

    ; infinite loop to keep checking for button press
    rjmp Loop

```

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

The following code is the C code variant for this task.

```

#define F_CPU 16000000UL // 8 Mhz clock
#include <avr/io.h> // standard library
#include <avr/interrupt.h> // interrupt library

```

```

#include <util/delay.h> // delay library

// short delay subroutine
void ShortDelay() {
    _delay_ms(333); // delay of 0.333 seconds
}

ISR(INT1_vect) {
    cli();
    PORTB &= ~(1 << PINB4); // turns on LED2
    for (int i=0; i<5; i++) {
        ShortDelay(); // delay cycle to achieve 3 seconds while accounting for interrupt activation
    }
    sei(); // enable global interrupt
}

int main(void)
{
    DDRB = 0xff; // set PB4 as output
    PORTD |= (1 << PIND3); // enables pull-ups for PD3
    EICRA |= (1 << ISC10); // using falling edge to generate interrupt
    EIMSK |= (1 << INT1); // only using INT1
    sei(); // global interrupt
    while (1) {
        PORTB = ~(0x00);
    }
    return 0;
}

```

The following code segment is in the AVR Assembly variant of the task.

```

.cseg
.org 0x0000 ; Reset vector
rjmp main ; Jump to main program

cli ; Disable global interrupts
sbi PORTB, PINB4 ; Turn on LED2 (set PB4 high)
ldi r16, 5 ; Initialize loop counter to 5
ShortDelayLoop:
    call ShortDelay ; Call the ShortDelay subroutine
    dec r16 ; Decrement loop counter
    brne ShortDelayLoop ; Repeat until loop counter is not zero
    sei ; Enable global interrupts
    reti ; Return from interrupt

main:
    ldi r16, 0xFF ; Set DDRB to output (all pins)
    out DDRB, r16
    ldi r16, (1 << PIND3) ; Enable pull-up on PD3
    out PORTD, r16
    ldi r16, (1 << ISC11)

```

```

        ori r16, (1 << ISC10) ; Set both ISC11 and ISC10
    sts EICRA, r16
    ldi r16, (1 << INT1) ; Enable INT1
    out EIMSK, r16
    sei ; Enable global interrupts

```

mainLoop:

```

    ldi r16, 0x00 ; Set PORTB to all zeros
    out PORTB, r16
    rjmp mainLoop ; Infinite loop

```

ShortDelay:

```

    ldi r17, 111 ; Load delay count (approximately 333 ms)

```

DelayLoop:

```

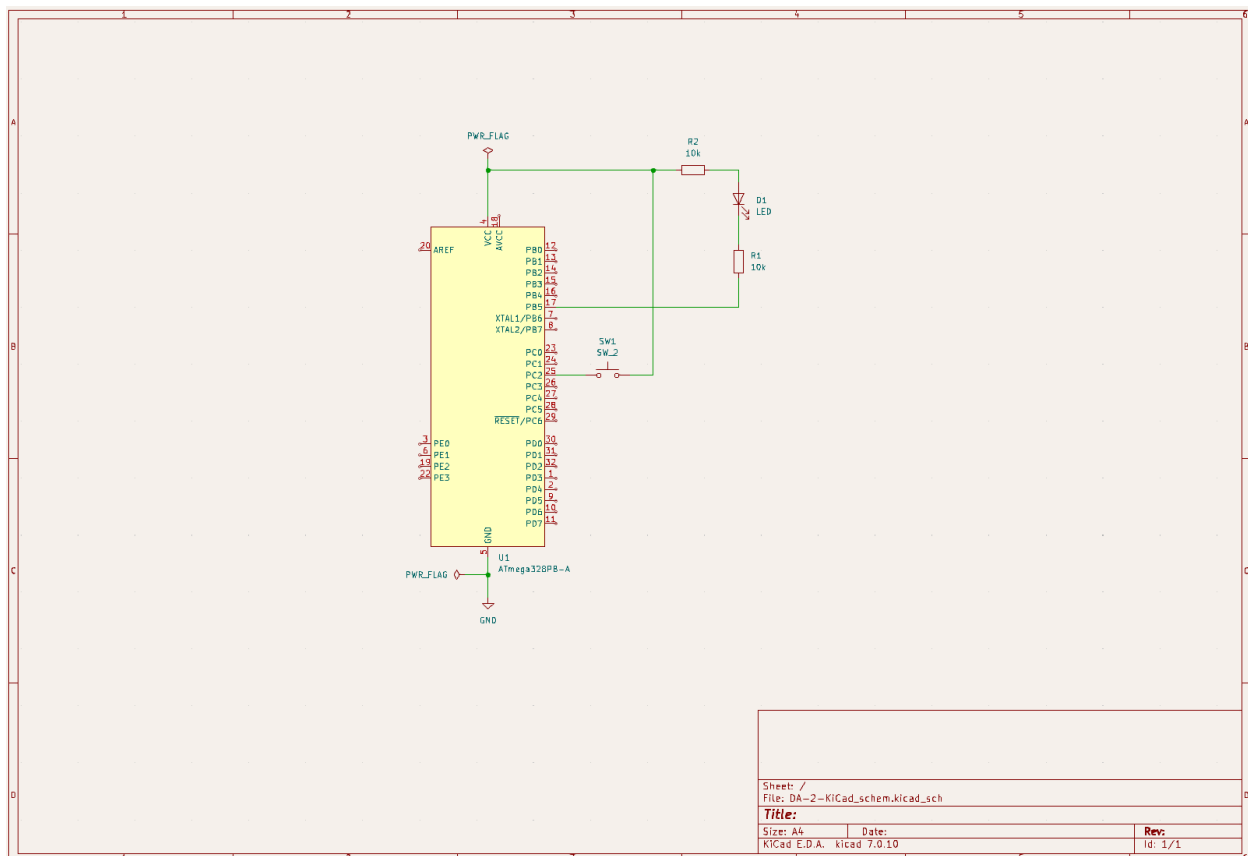
    dec r17 ; Decrement delay count
    brne DelayLoop ; Repeat until delay count is zero
    ret ; Return from subroutine

```

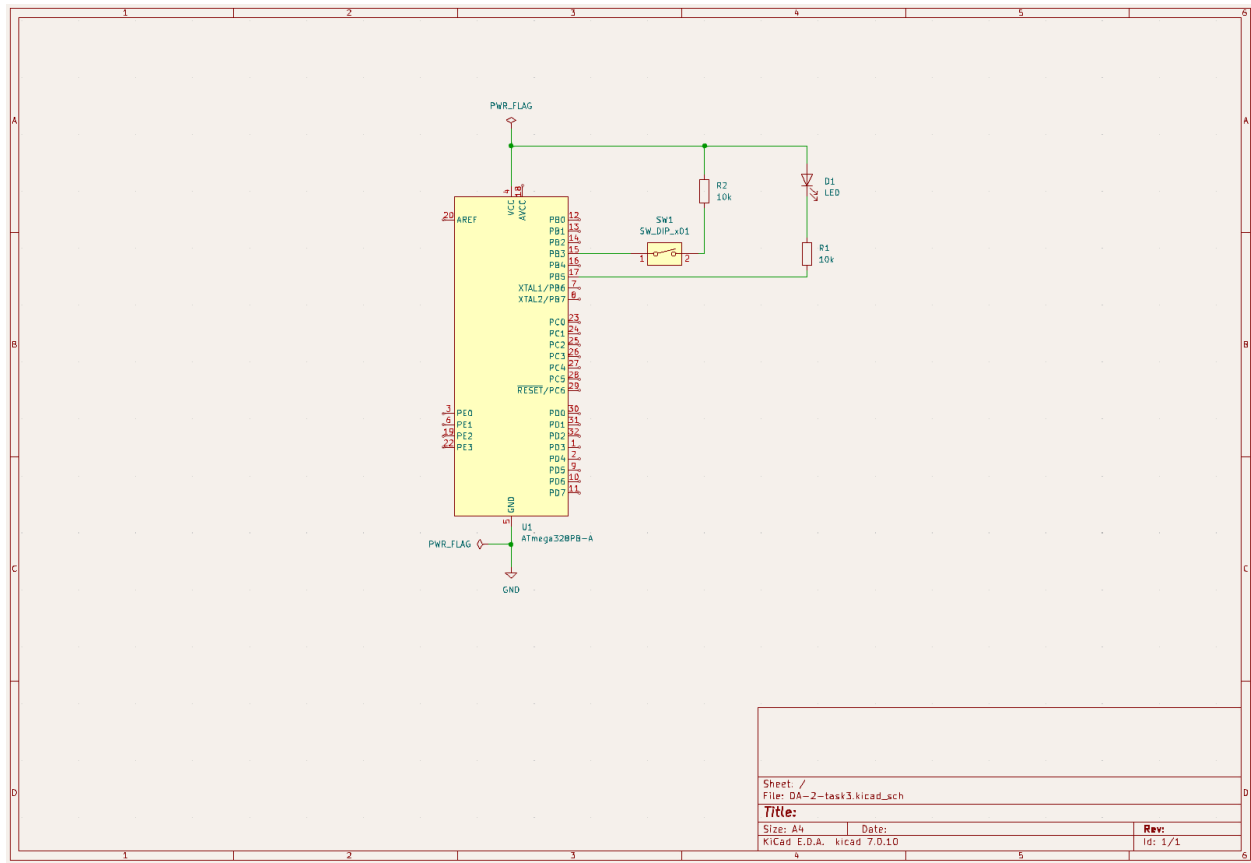
5. SCHEMATICS

The following will be KiCad schematics of tasks 2 and 3, displaying the pinouts and component connections for each task.

Task 2:

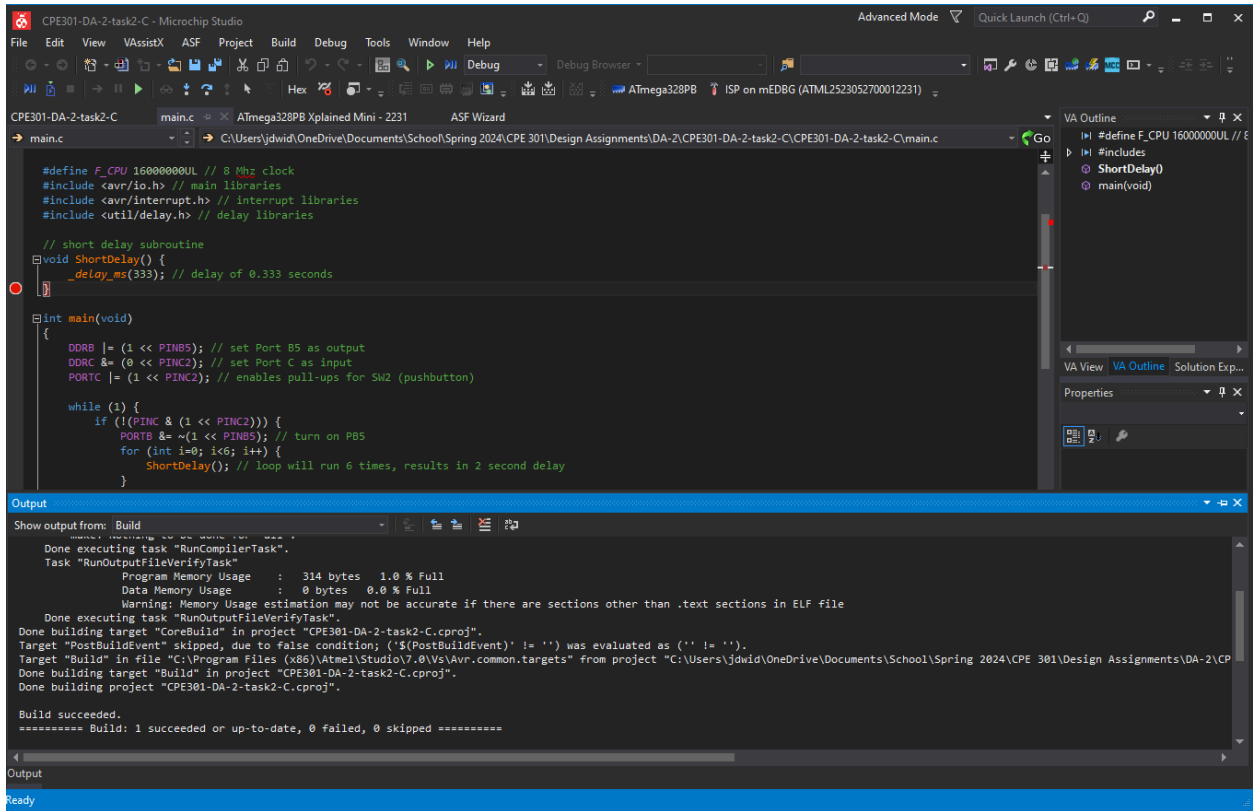


Task 3:



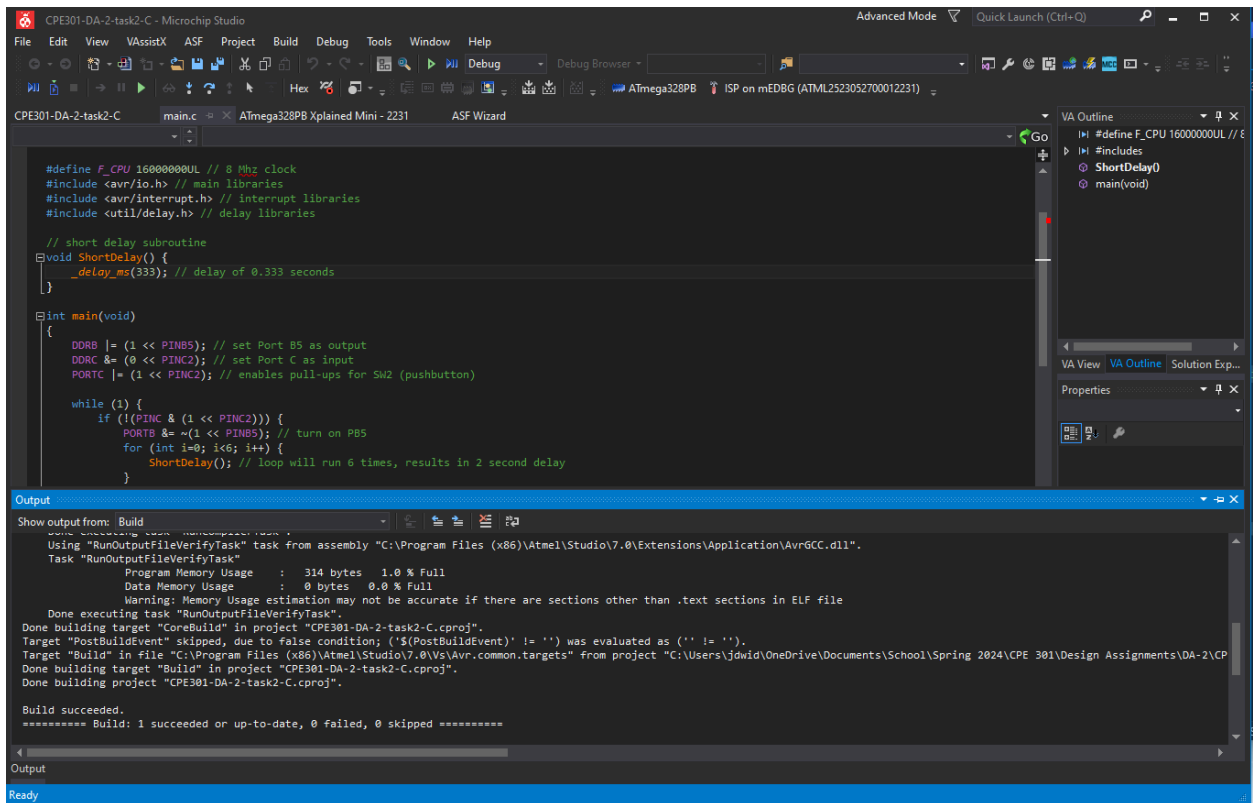
6. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)

Task 1:

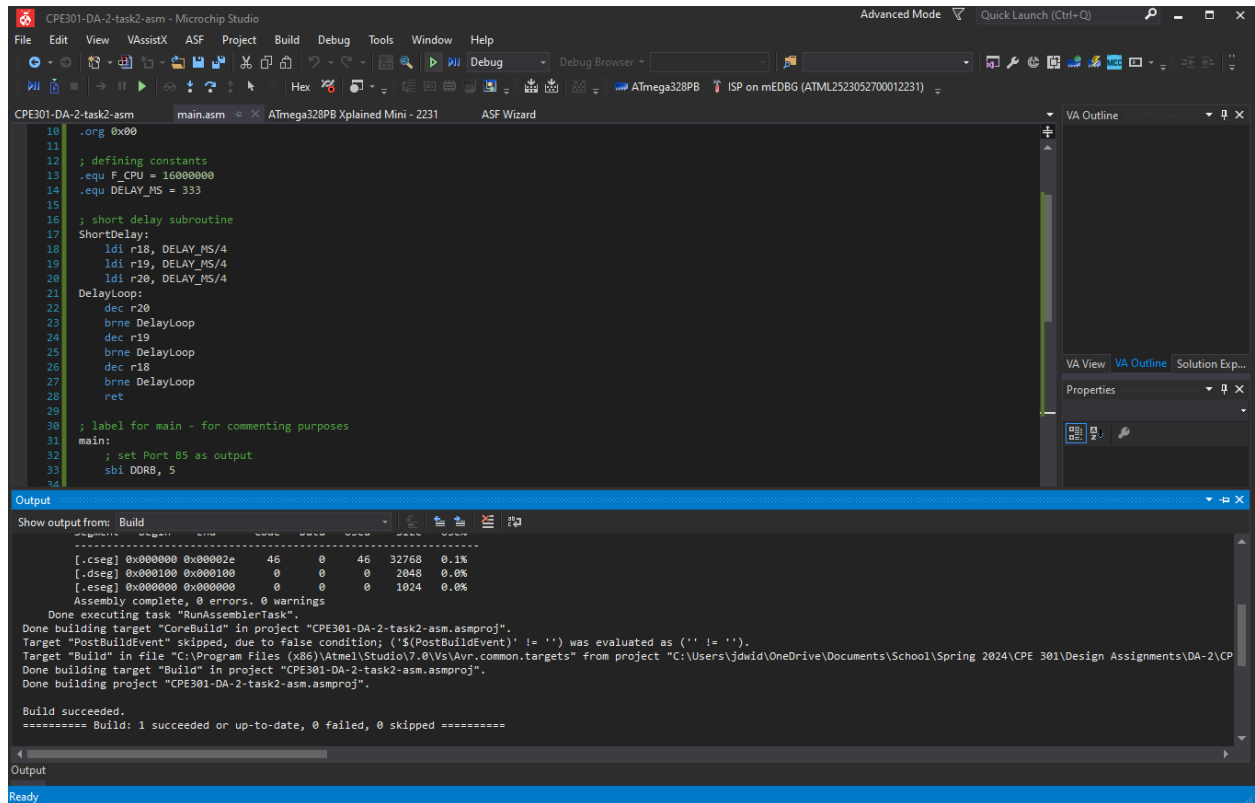


Task 2:

C code:



ASM code:



The screenshot displays the Microchip Studio IDE interface. The main editor window shows the assembly file `main.asm` for the `ATmega328PB Xplained Mini - 2231` target. The code includes constants for CPU frequency and delay, a short delay subroutine, and a main routine that sets Port B5 as output.

```
10 .org 0x00
11
12 ; defining constants
13 .equ F_CPU = 16000000
14 .equ DELAY_MS = 333
15
16 ; short delay subroutine
17 ShortDelay:
18     ldi r18, DELAY_MS/4
19     ldi r19, DELAY_MS/4
20     ldi r20, DELAY_MS/4
21 DelayLoop:
22     dec r20
23     brne DelayLoop
24     dec r19
25     brne DelayLoop
26     dec r18
27     brne DelayLoop
28     ret
29
30 ; label for main - for commenting purposes
31 main:
32     ; set Port B5 as output
33     sbi DDRC, 5
34
```

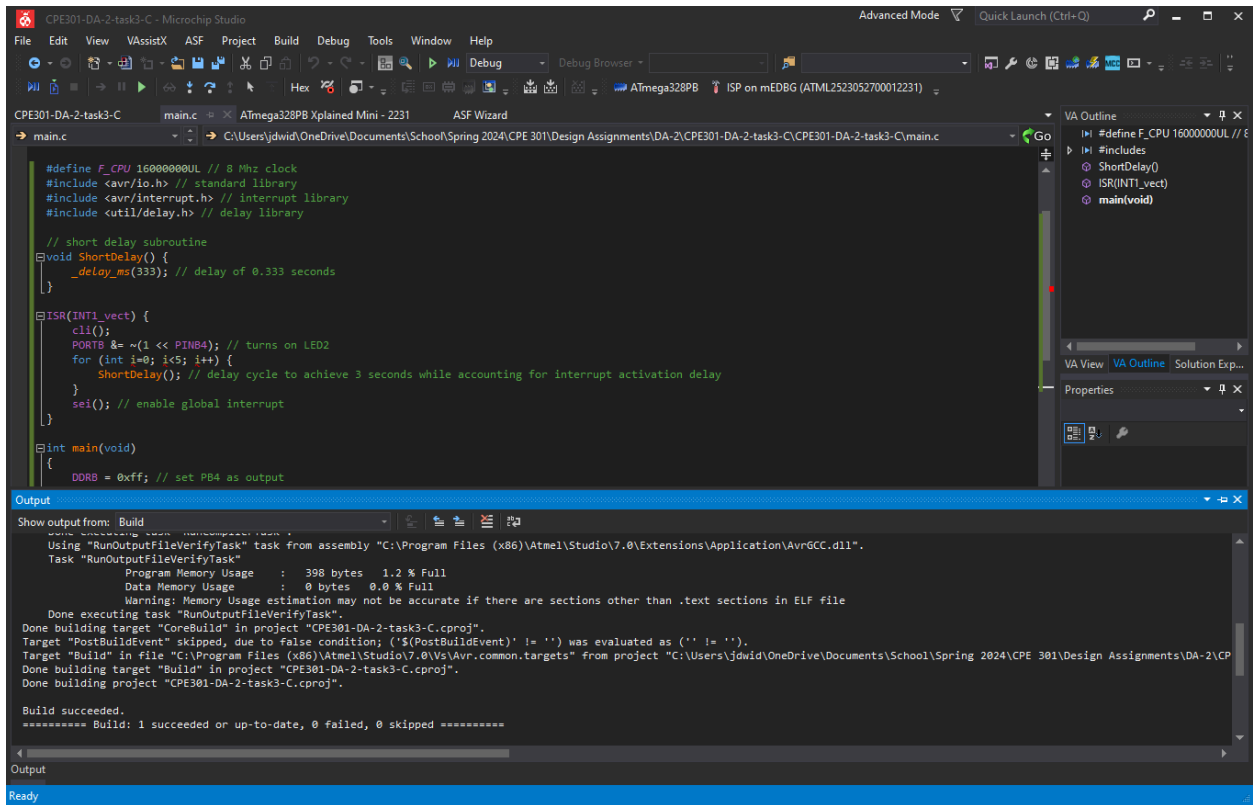
The Output window at the bottom shows the build process results:

```
Show output from: Build
-----
[.cseg] 0x000000 0x00002e 46 0 46 32768 0.1%
[.dseg] 0x000100 0x000100 0 0 0 2048 0.0%
[.eseg] 0x000000 0x000000 0 0 0 1024 0.0%
Assembly complete, 0 errors, 0 warnings
Done executing task "RunAssemblerTask".
Done building target "CoreBuild" in project "CPE301-DA-2-task2-asm.asmproj".
Target "PostBuildEvent" skipped, due to false condition; ('$(PostBuildEvent)' != '') was evaluated as ('' != '').
Target "Build" in file "C:\Program Files (x86)\Atmel\Studio\7.0\Vs\Avr.common.targets" from project "C:\Users\jdwild\OneDrive\Documents\School\Spring 2024\CPE 301\Design Assignments\DA-2\CPE301-DA-2-task2-asm.asmproj".
Done building target "Build" in project "CPE301-DA-2-task2-asm.asmproj".
Done building project "CPE301-DA-2-task2-asm.asmproj".

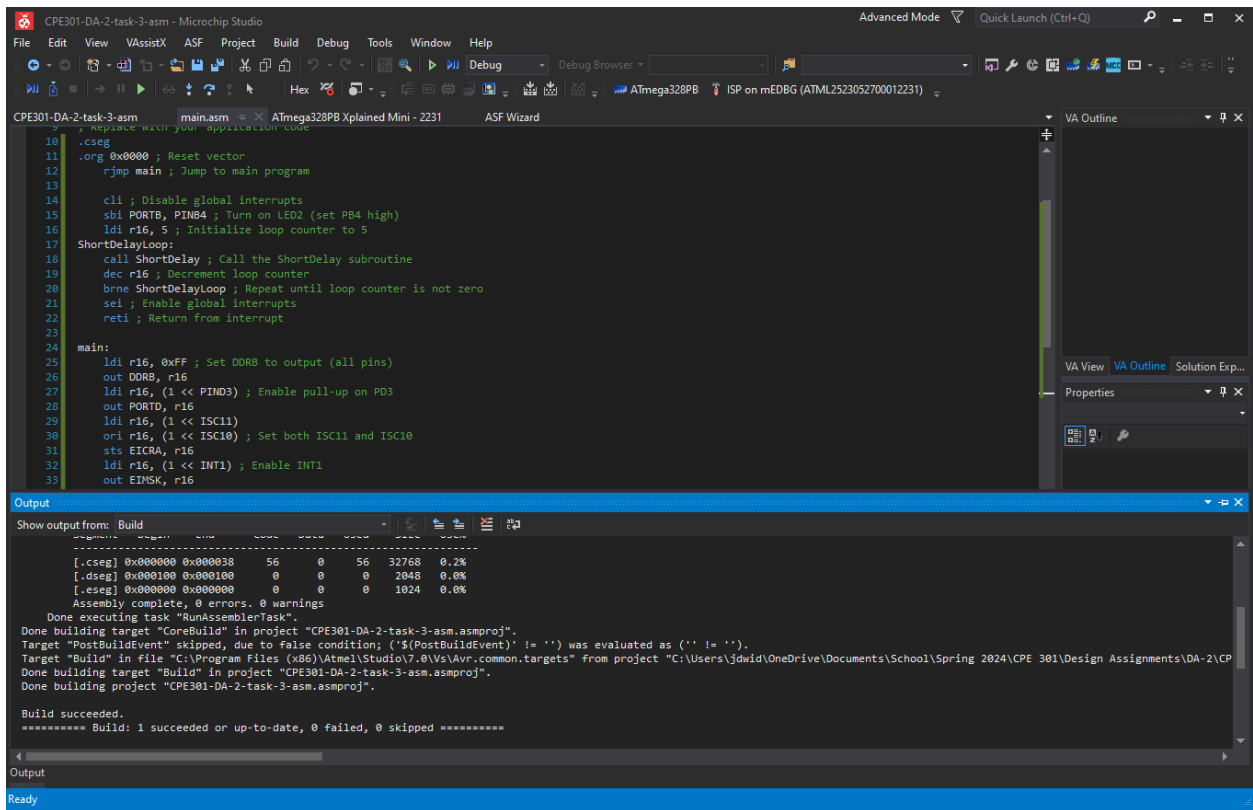
Build succeeded.
***** Build: 1 succeeded or up-to-date, 0 failed, 0 skipped *****
```

The status bar at the bottom indicates the IDE is **Ready**.

Task 3: C code:



ASM code:

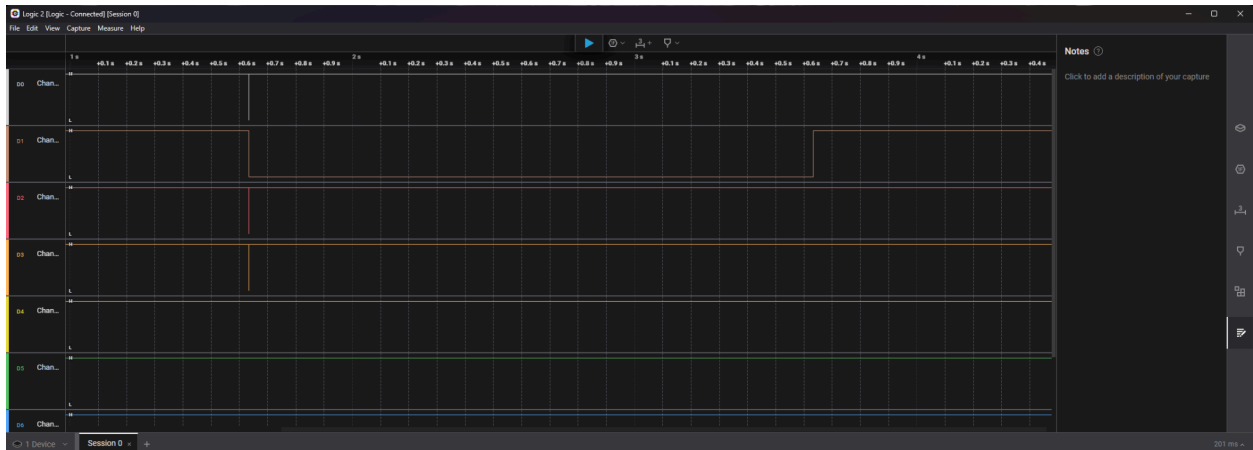


7. SCREENSHOT OF EACH DEMO (BOARD SETUP)

Task 1: N/A

Task 2:

Logic analysis:

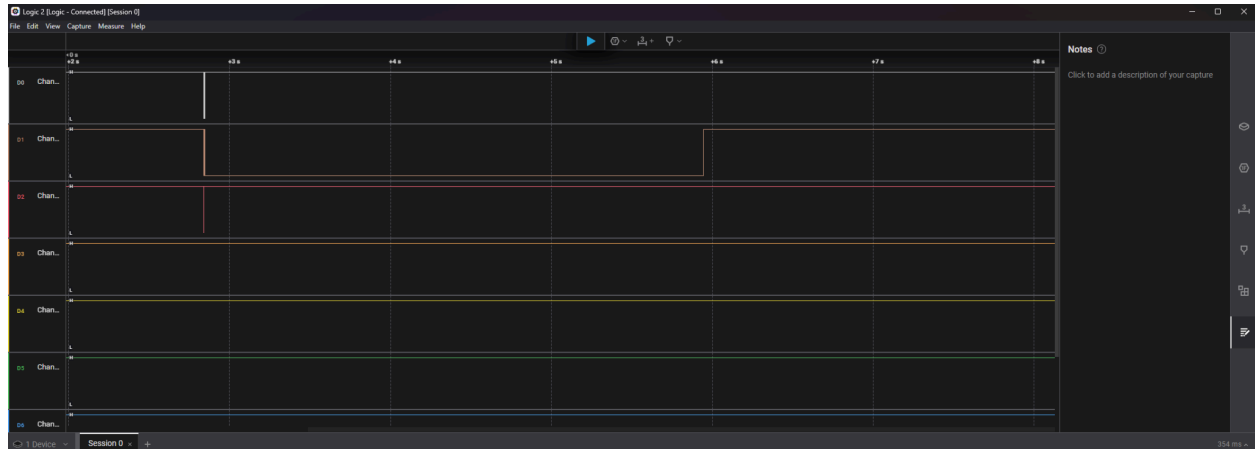


Board layout:

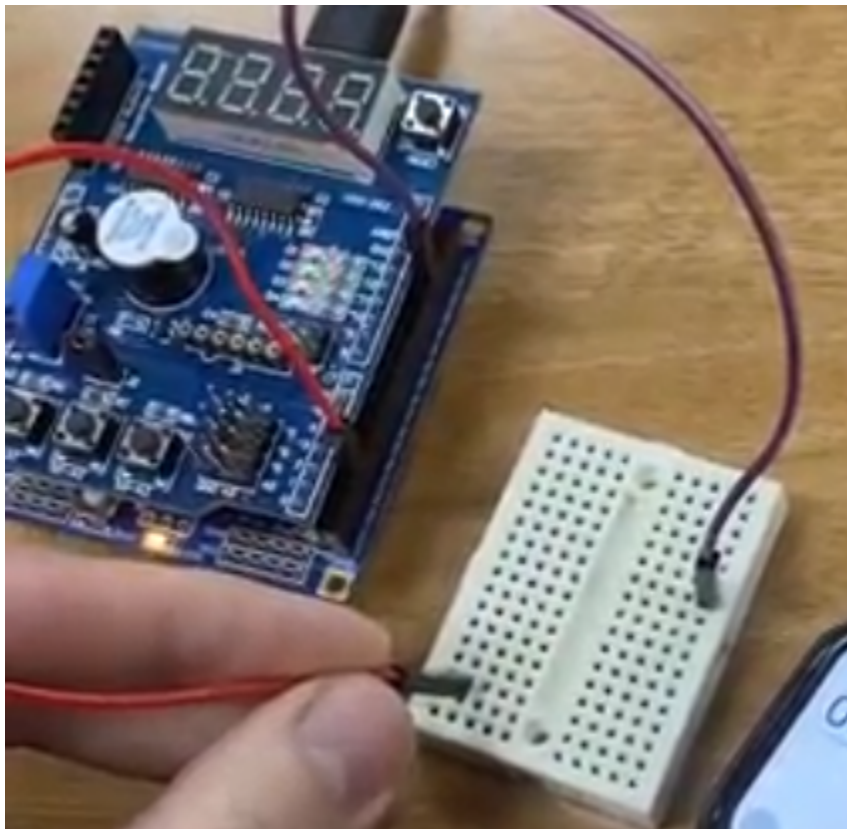


Task 3:

Logic analysis:



Board layout:



8. VIDEO LINKS OF EACH DEMO

Task 1: N/A

Task 2: <https://youtube.com/shorts/vcl5-gKW0n8?feature=share>

Task 3: <https://youtube.com/shorts/SML3IMhItPE?feature=share>

9. GITHUB LINK OF THIS DA

Task 1:

C code:

<https://github.com/JackOfSpades-7/UNLV-Embedded-Systems/blob/main/Design%20Assignment%202/D A-2%20Task%201-2%20C%20code>

ASM code:

<https://github.com/JackOfSpades-7/UNLV-Embedded-Systems/blob/main/Design%20Assignment%202/D A-2%20Task%201-2%20ASM%20code>

Task 2:

C code:

<https://github.com/JackOfSpades-7/UNLV-Embedded-Systems/blob/main/Design%20Assignment%202/D A-2%20Task%201-2%20C%20code>

ASM code:

<https://github.com/JackOfSpades-7/UNLV-Embedded-Systems/blob/main/Design%20Assignment%202/D A-2%20Task%201-2%20ASM%20code>

Task 3:

C code:

<https://github.com/JackOfSpades-7/UNLV-Embedded-Systems/blob/main/Design%20Assignment%202/D A-2%20Task%203%20C%20code>

ASM code:

<https://github.com/JackOfSpades-7/UNLV-Embedded-Systems/blob/main/Design%20Assignment%202/D A-2%20Task%203%20ASM%20code>

Student Academic Misconduct Policy

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

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

Johnathan Widney