

# Design Assignment 3

---

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%203>

Video Playlist:

<https://www.youtube.com/playlist?list=PLoASw0sToF2WLAyQSglq1SsK2MMI-Ypez>

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

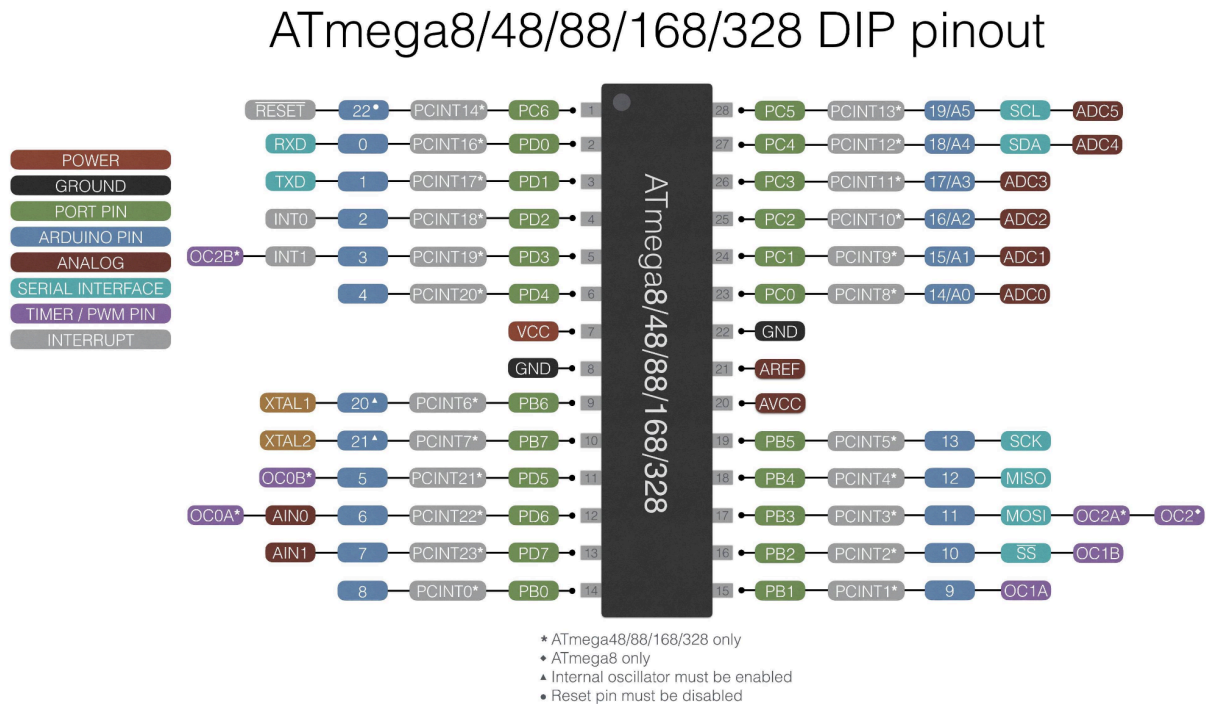
List of Components used

Block diagram with pins used in the Atmega328PB (only)

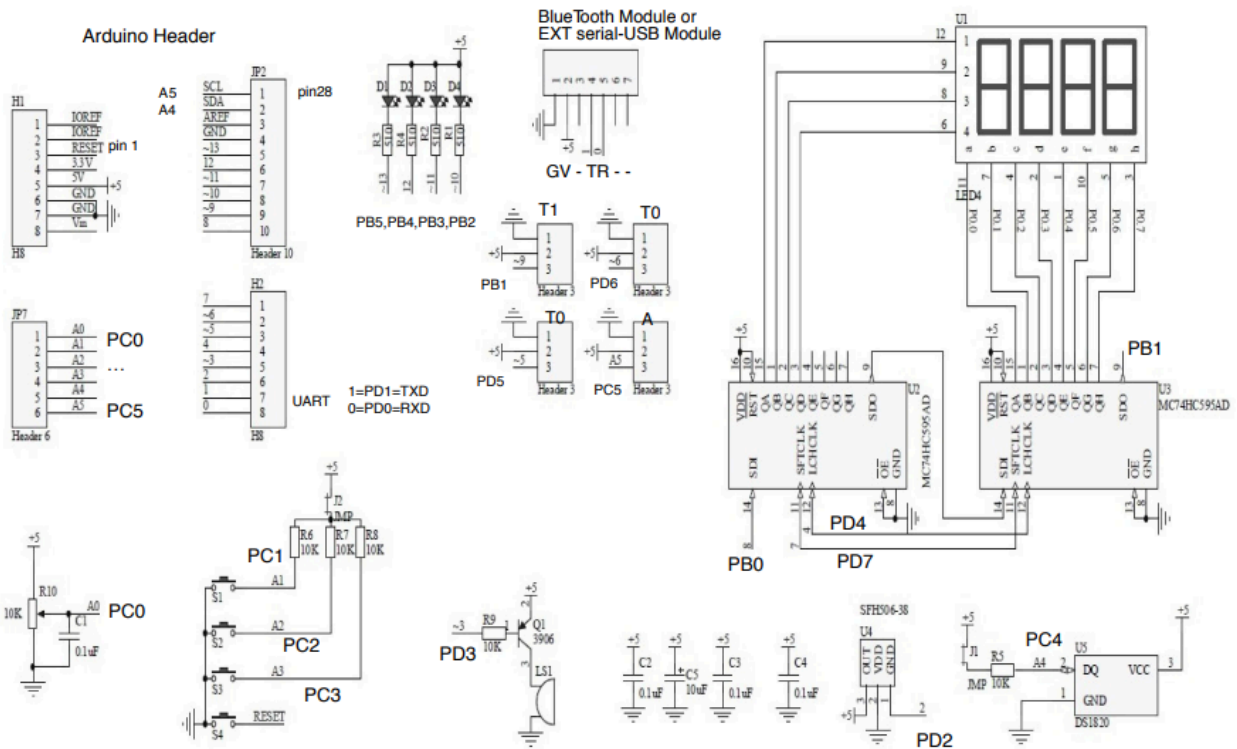
- Atmega328PB Xplained mini microcontroller board
- Arduino compatible external multifunction development shield
- Male-to-male jumper cables
- Logic analyzer
- Female-to-female ribbon cable
- PC

Block diagrams and pins:

Atmega328PB Micro controller:



Arduino compatible multifunction development shield:



For assignments dealing with LED - use the pins PB5,PB4,PB3,PB2. For assignments dealing with switches, pin interrupts use pins PC1,PC2,PC3  
 For assignments in PWM use ~5/PD5 (T0B), ~6/PD6 (T0A), ~9/PB1 (T1A), ~10/PB2 (T1B-LED), ~11/PB3(T2A-LED),~3/PD3(T2B/Buzzer)  
 For assignments with analog input use A0/PC0-Potentiometer, A4/PC4/LM3X, or EXT @ PC5. PD2 is INT0 pin (external interrupt)

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

/\* This code contains all subtasks of task 1, as I have all 3 timers running simultaneously

Task 1.a is accomplished by Timer 0

Task 1.b is accomplished by Timer 1

Task 1.c is accomplished by Timer 2

Review these code pieces for each respective task

\*/

```
#include <avr/io.h>
```

```
#include <avr/interrupt.h>
```

```
// Define LED pins
```

```
#define LED0_PIN PB5
```

```
#define LED1_PIN PB4
```

```
#define LED2_PIN PB3
```

```
// Define counters for LED toggle delays
```

```
volatile uint16_t counter0 = 0;
```

```
volatile uint16_t counter1 = 0;
```

```

volatile uint16_t counter2 = 0;

// Initialize Timer 0 in normal mode (no interrupts)
void initTimer0() {
    // Set prescaler to 64 (CS02 = 0, CS01 = 1, CS00 = 1)
    TCCR0B |= (1 << CS01) | (1 << CS00);
    // Set compare match value for 0.1 ms delay
    OCR0A = 25; // (16 MHz / 64) * 0.1 ms = 25
}

// Initialize Timer 1 in CTC mode with interrupt
void initTimer1() {
    // Set CTC mode (WGM12 = 1)
    TCCR1B |= (1 << WGM12);
    // Set prescaler to 256 (CS12 = 1, CS11 = 0, CS10 = 0)
    TCCR1B |= (1 << CS12);
    // Set compare match value for 0.5 ms delay
    OCR1A = 125; // (16 MHz / 256) * 0.5 ms = 125
    // Enable Timer 1 compare match interrupt
    TIMSK1 |= (1 << OCIE1A);
}

// Initialize Timer 2 in normal mode with interrupt
void initTimer2() {
    // Set prescaler to 64 (CS22 = 1, CS21 = 0, CS20 = 1)
    TCCR2B |= (1 << CS22) | (1 << CS20);
    // Set compare match value for 0.25 ms delay
    OCR2A = 63; // (16 MHz / 64) * 0.25 ms = 63
    // Enable Timer 2 compare match interrupt
    TIMSK2 |= (1 << OCIE2A);
}

// Timer 1 compare match ISR
ISR(TIMER1_COMPA_vect) {
    counter1++;
    if (counter1 >= 6000) { // 3 seconds / 0.5 ms = 6000
        PORTB ^= (1 << LED1_PIN); // Toggle LED1
        counter1 = 0;
    }
}

// Timer 2 compare match ISR
ISR(TIMER2_COMPA_vect) {
    counter2++;
    if (counter2 >= 8000) { // 2 seconds / 0.25 ms = 8000
        PORTB ^= (1 << LED2_PIN); // Toggle LED2
        counter2 = 0;
    }
}

int main() {
    // Initialize LEDs as outputs

```

```

DDRB |= (1 << LED0_PIN) | (1 << LED1_PIN) | (1 << LED2_PIN);

// Initialize timers
initTimer0();
initTimer1();
initTimer2();

// Enable global interrupts
sei();

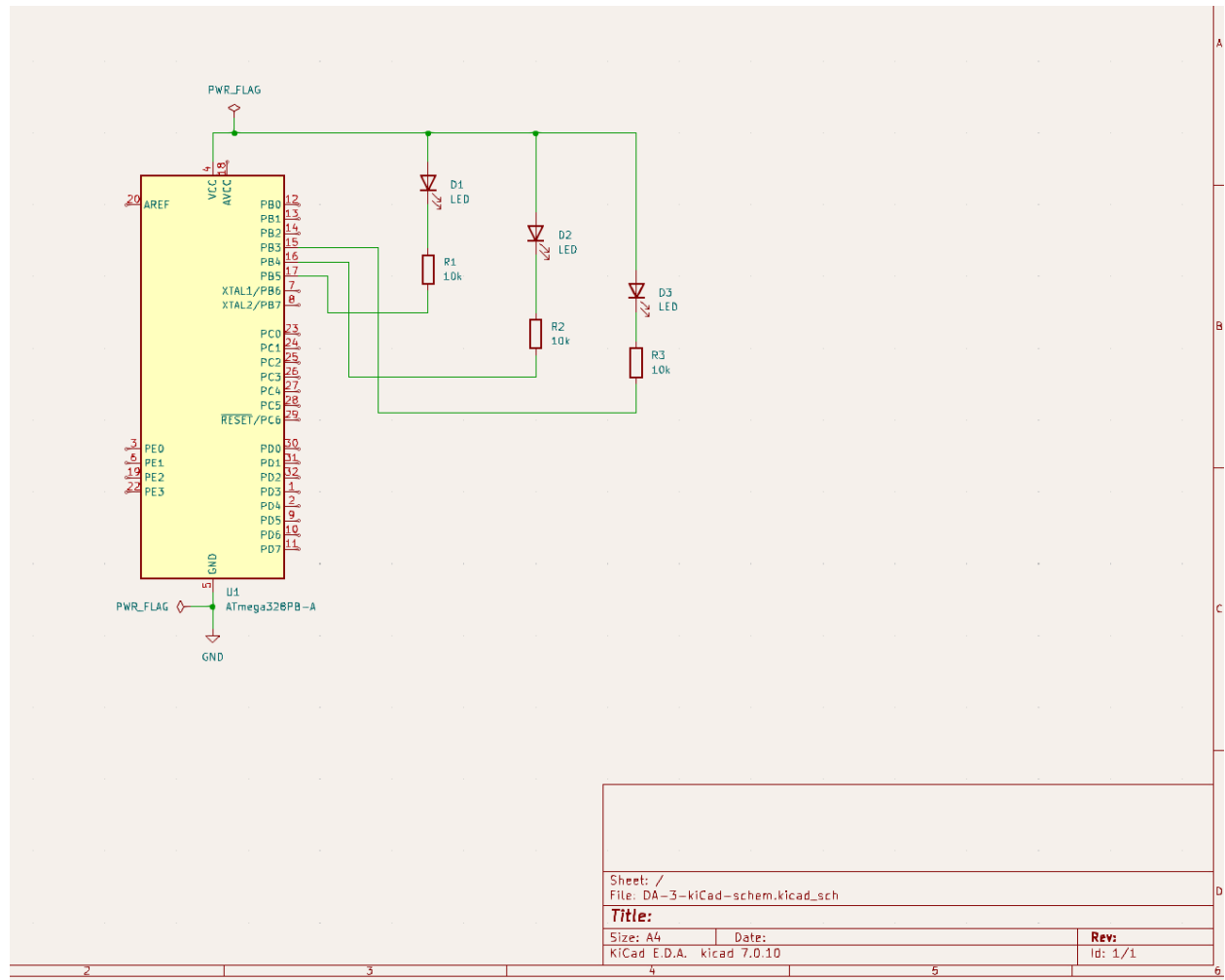
while (1) {
    // Check Timer 0 (0.1 ms delay)
    if (TIFR0 & (1 << OCF0A)) {
        counter0++;
        if (counter0 >= 10000) { // 1 second / 0.1 ms = 10000
            PORTB ^= (1 << LED0_PIN); // Toggle LED0
            counter0 = 0;
        }
        TIFR0 |= (1 << OCF0A); // Clear the flag
    }
}

return 0;
}

```

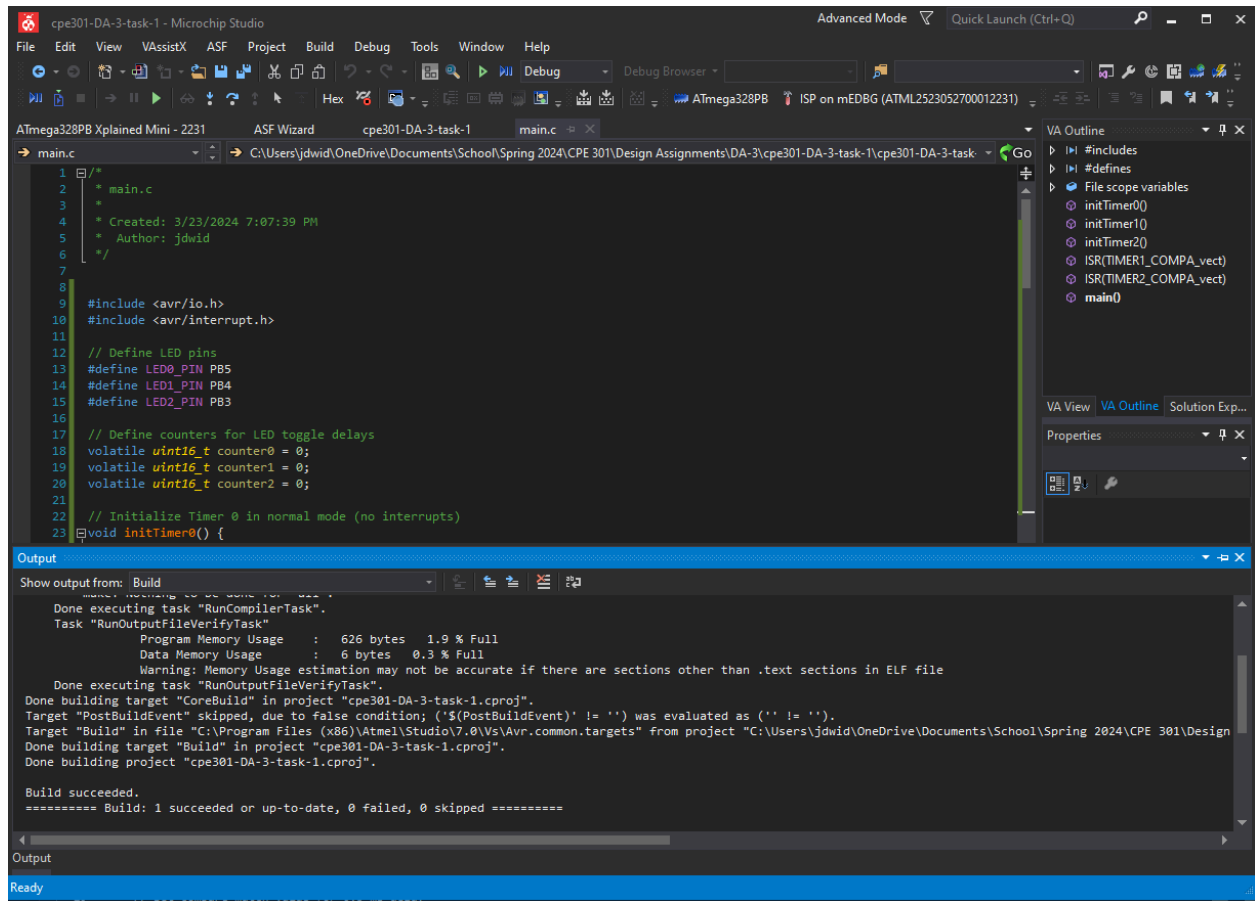
### 3. SCHEMATICS

Use KICAD schematics only (not required for DA1 simulation)



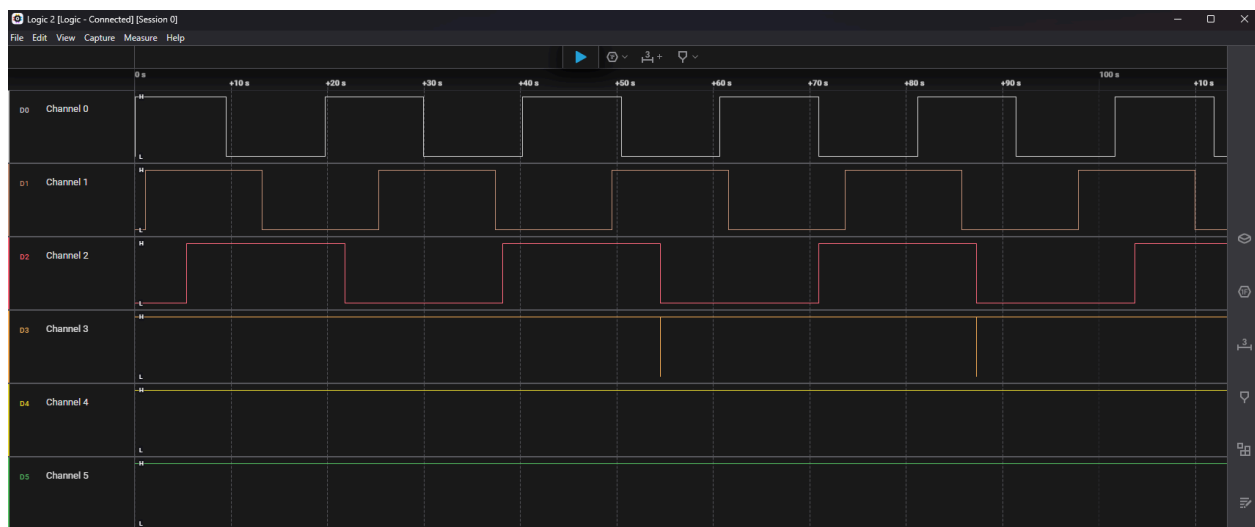
#### 4. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)

## Task 1:



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

### Task 1:



## 6. VIDEO LINKS OF EACH DEMO

Task 1: <https://www.youtube.com/playlist?list=PLoASw0sToF2WLAyQSglq1SsK2MMI-Ypez>

## **7. GITHUB LINK OF THIS DA**

Task 1:

<https://github.com/JackOfSpades-7/UNLV-Embeded-Systems/blob/main/Design%20Assignment%203/Task-1-code.c>

### **Student Academic Misconduct Policy**

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

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

Johnathan Widney