

# Design Assignment 2A

---

Student Name: Dillon Archibald

Student #: 5004439916

Student Email: Archid1@unlv.nevada.edu

Primary Github address: <https://github.com/Dil-bert/Alabaster.git>

Directory: Alabaster/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

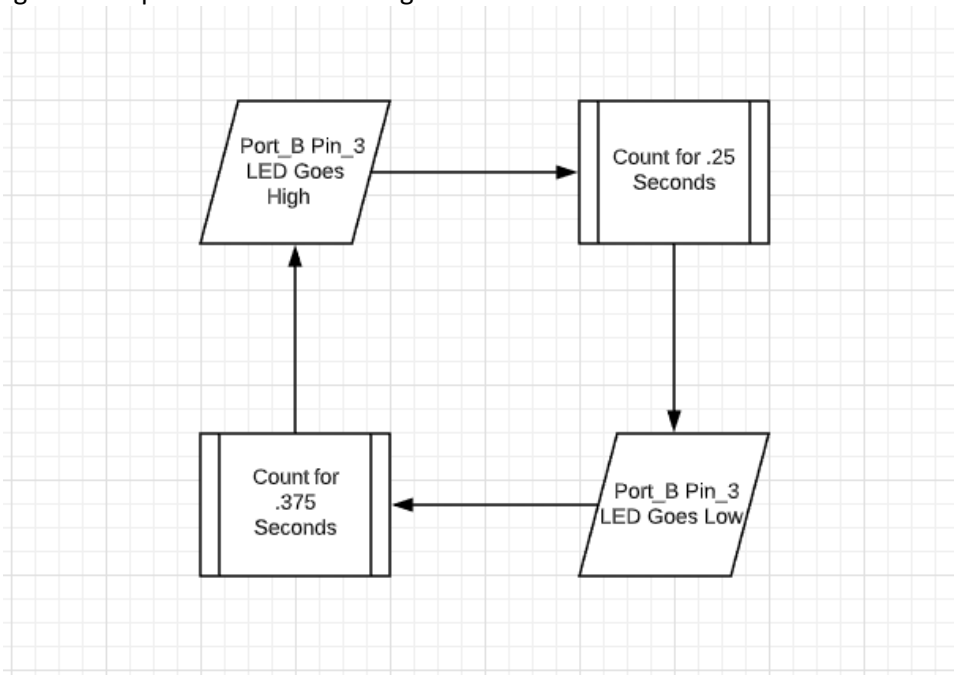
List of Components used

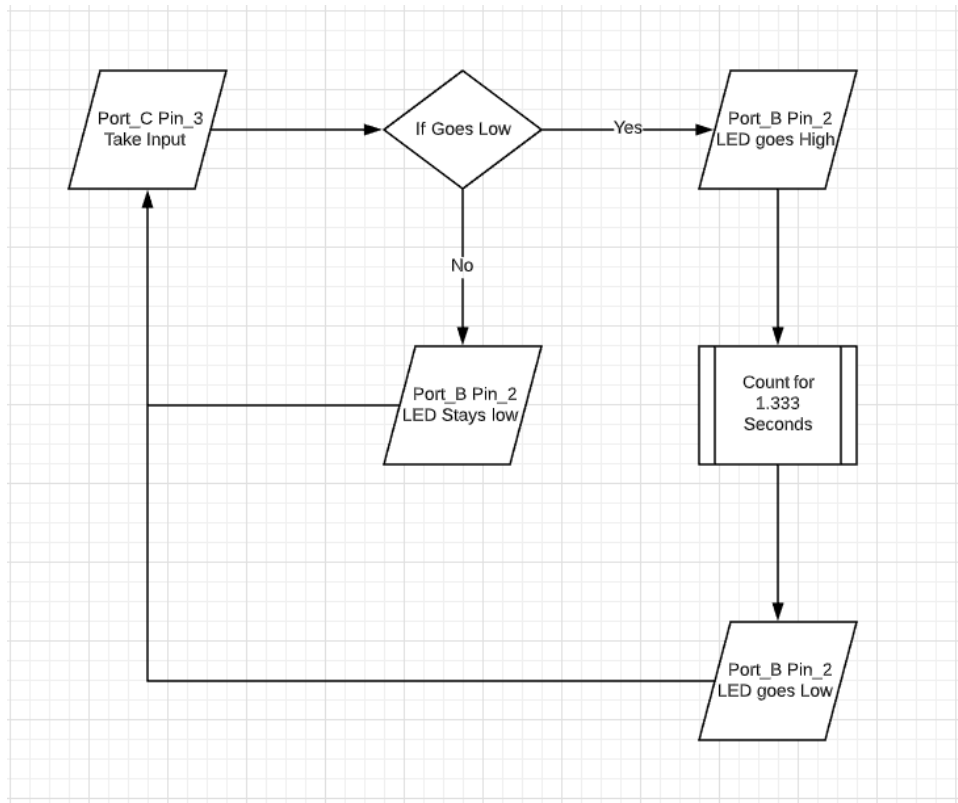
1 X ATmega328P Xplained mini

1 X Wire

1 X LED

Block diagram with pins used in the Atmega328P





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

N/A

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

```

; DA2A_1_assemble.asm
;
; Created: 9/29/2019 8:16:28 PM
; Author : Dilbert
;
.org 0x00                                ; Start of program at reset interrupt
ldi r16, 0x08                            ; Load register R16 with 0b'00001000
ldi r17, 0x00                            ; Load register R17 with 0x00
out ddrb, r16                            ; Set Port B Pin 3 as an output
out portb, r17                           ; Set all of Port B to Low
start:                                   ; Start Location after each period
    ldi r18, 0x04                        ; Load register R18 with 0x04 (This is the duty cycle percent) e.g. 40%
    ldi r19, 0x06                        ; Load register R19 with 0x06 (This is the off time of the period) e.g. 60%
    out portb, r16                       ; set Port B Pin 3 High
loop_one:                                ; Start the Loop for the duty cycle on time
    call delay                           ; Call the Delay function, the delay runs for one tenth of the total
    period.
    dec r18                              ; Decrement R18
    brne loop_one                        ; If R18 is not zero go back to Loop_one:
    out portb, r17                       ; Set all of Port B to Low
  
```

```

loop_two:                                ; Start the Loop for the off time of the duty cycle
call delay                               ; Call the Delay function, the delay runs for one tenth of the total
period.                                  ;
dec r19                                  ; Decrement R19
brne loop_two                            ; If R19 is not zero go back to Loop_two:
rjmp start                               ;****End Of Main Prog***** Jump back up to start
;-----

```

```

delay:                                   ;*****Begin Of Delay Function*****
ldi r23, 0x0A                            ; Load R23 With 10
d2:                                       ; Loop d2
ldi r24, 0x64                            ; Load R22 With 100
d1:                                       ; Loop d1
ldi r25, 0x64                            ; Load R21 With 100
d0:                                       ; Loop d0
nop                                       ; No operation
nop                                       ; No operation
nop                                       ; No operation
nop                                       ; No operation
nop                                       ; No operation
nop                                       ; No operation
nop                                       ; No operation
dec r25                                  ; Decrement R25
brne d0                                  ; Branch to d0 if R25 is not 0
dec r24                                  ; Decrement R24
brne d1                                  ; Branch to d1 if R24 is not 0
dec r23                                  ; Decrement R23
brne d2                                  ; Branch to d0 if R23 is not 0
ret                                       ; Return to calling function
    rjmp start                           ; Unused but would return to beginning of the cycle if ret was
skipped for some reason

```

```

/*
 * DA2A_1_C.c
 *
 * Created: 9/29/2019 10:55:18 PM
 * Author : Dilbert
 */
#define F_CPU 16000000UL

```

```

#include <avr/io.h>
#include <util/delay.h>

```

```

int main(void)
{
    DDRB |= (1 << 3);                    // Set port B pin 3 as an output
    PORTB = (0 << 3);                    // Set port B pin 3 output to zero
    while (1)                            // Infinite loop
    {

```

```
; DA2A_2_assemble.asm
;
; Created: 9/28/2019 9:31:49 PM
; Author : Dilbert
;
```

[illegible]

```

nop                ;      No operation
nop                ;      No operation
nop                ;      No operation
dec r25            ;      Decrement R25
brne d0            ;      Branch to d0 if R25 is not zero
dec r24            ;      Decrement R24
brne d1            ;      Branch to d1 if R24 is not zero
dec r23            ;      Decrement R23
brne d2            ;      Branch to d0
ret                ;      Return to Main Prog

```

```

/*
 * DA2A_2_C.c
 *
 * Created: 9/29/2019 11:23:02 PM
 * Author : Dilbert
 */

```

```

#define F_CPU 16000000UL

```

```

#include <avr/io.h>
#include <util/delay.h>

```

```

int main(void)
{
    DDRB |= (1 << 2);           // Set port B pin 2 as output
    PORTB = (0 << 2);           // Set port B pin 2 Low
    DDRC |= (0 << 3);           // Set port C pin 3 as input
    PORTC = (1 << 3);           // Set port C pin 3 as normal high (Enable Pull up resistor)
    while (1)                   // Infinite loop (polling)
    {
        if(!(PINC & (1 << 3))) { // IF Pin 3 on Port C is Low, enter statement, else loop
            PORTB |= (1 << 2);    // Turn on LED
            _delay_ms(1333);      // Delay for 1.333 sec
            PORTB = (0 << 2);    // Turn off LED
        }
    }
}

```

#### 4. SCHEMATICS

Use fritzing.org  
 Don't know exactly what is wanted here. I will ask in class

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

## DA2A\_1\_Assembly

Registers

R00 = 0x00 R01 = 0x00 R02 = 0x00 R03 = 0x00 R04 = 0x00 R05 = 0x00 R06 = 0x00 R07 = 0x00 R08 = 0x00 R09 = 0x00 R10 = 0x00 R11 = 0x00 R12 = 0x00 R13 = 0x00 R14 = 0x00 R15 = 0x00 R16 = 0x00  
R17 = 0x00 R18 = 0x00 R19 = 0x00 R20 = 0x00 R21 = 0x00 R22 = 0x00 R23 = 0x00 R24 = 0x00 R25 = 0x00 R26 = 0x00 R27 = 0x00 R28 = 0x00 R29 = 0x00 R30 = 0x00 R31 = 0x00

Disassembly

main.asm DA2A\_1\_assemble

```
;
;
.org 0x0000 ; Start of program at reset interrupt
ldi r16, 0x08 ; Load register R16 with 0b'00001000
ldi r17, 0x00 ; Load register R17 with 0x00
out ddrb, r16 ; Set Port B Pin 3 as an output
out portb, r17 ; Set all of Port B to Low
start: ; Start Location after each period
ldi r18, 0x04 ; Load register R18 with 0x04 (This is the duty cycle percent) e.g. 40%
ldi r19, 0x06 ; Load register R19 with 0x06 (This is the off time of the period) e.g. 60%
out portb, r16 ; Set Port B Pin 3 High
loop_one: ; Start the Loop for the duty cycle on time
call delay ; Call the Delay function, the delay runs for one tenth of the total period.
dec r18 ; Decrement R18
brne loop_one ; If R18 is not zero go back to Loop_one:
out portb, r17 ; Set all of Port B to Low
loop_two: ; Start the Loop for the off time of the duty cycle
call delay ; Call the Delay function, the delay runs for one tenth of the total period.
dec r19 ; Decrement R19
brne loop_two ; If R19 is not zero go back to Loop_two:
rjmp start ;****End Of Main Prog**** Jump back up to start
;-----
delay: ;*****Begin Of Delay Function*****
ldi r23, 0x0A ; Load R23 With 10
loop_d2:
```

Processor Status

| Name            | Value         |
|-----------------|---------------|
| Program Counter | 0x00000004    |
| Stack Pointer   | 0x00FF        |
| X Register      | 0x0000        |
| Y Register      | 0x0000        |
| Z Register      | 0x0000        |
| Status Register | 0x00000000    |
| Cycle Counter   | 10030418      |
| Frequency       | 16,000 MHz    |
| Stop Watch      | 626,901.13 µs |

Registers

| Register | Value |
|----------|-------|
| R00      | 0x00  |
| R01      | 0x00  |
| R02      | 0x00  |
| R03      | 0x00  |
| R04      | 0x00  |
| R05      | 0x00  |
| R06      | 0x00  |
| R07      | 0x00  |
| R08      | 0x00  |
| R09      | 0x00  |
| R10      | 0x00  |
| R11      | 0x00  |
| R12      | 0x00  |
| R13      | 0x00  |
| R14      | 0x00  |
| R15      | 0x00  |
| R16      | 0x08  |
| R17      | 0x00  |
| R18      | 0x00  |

Autos

| Name | Value | Type |
|------|-------|------|
|------|-------|------|

Memory 4

Memory: prog FLASH Address: 0x0,prog

|             |   |
|-------------|---|
| prog 0x0000 | 08 e0 10 e0 04 b9 15 b9 24 e0 36 e0 05 b9 0e 94 11 00 2a 95 e1 f7 15 b9 0e 94 |
| prog 0x0010 | 95 e1 f7 f3 cf 7a e0 e6 94 e6 00 00 00 00 00 00 00 00 00 00 00 00 00 9a       |
| prog 0x003A | 8a 95 a1 f7 7a 95 89 f7 08 95 e1 cf ff ff ff ff ff ff ff ff ff ff ff ff ff    |
| prog 0x0057 | ff       |
| prog 0x0074 | ff       |
| prog 0x0091 | ff       |
| prog 0x00AE | ff       |
| prog 0x00CB | ff       |
| prog 0x00E8 | ff       |

Stopped

Ln 13

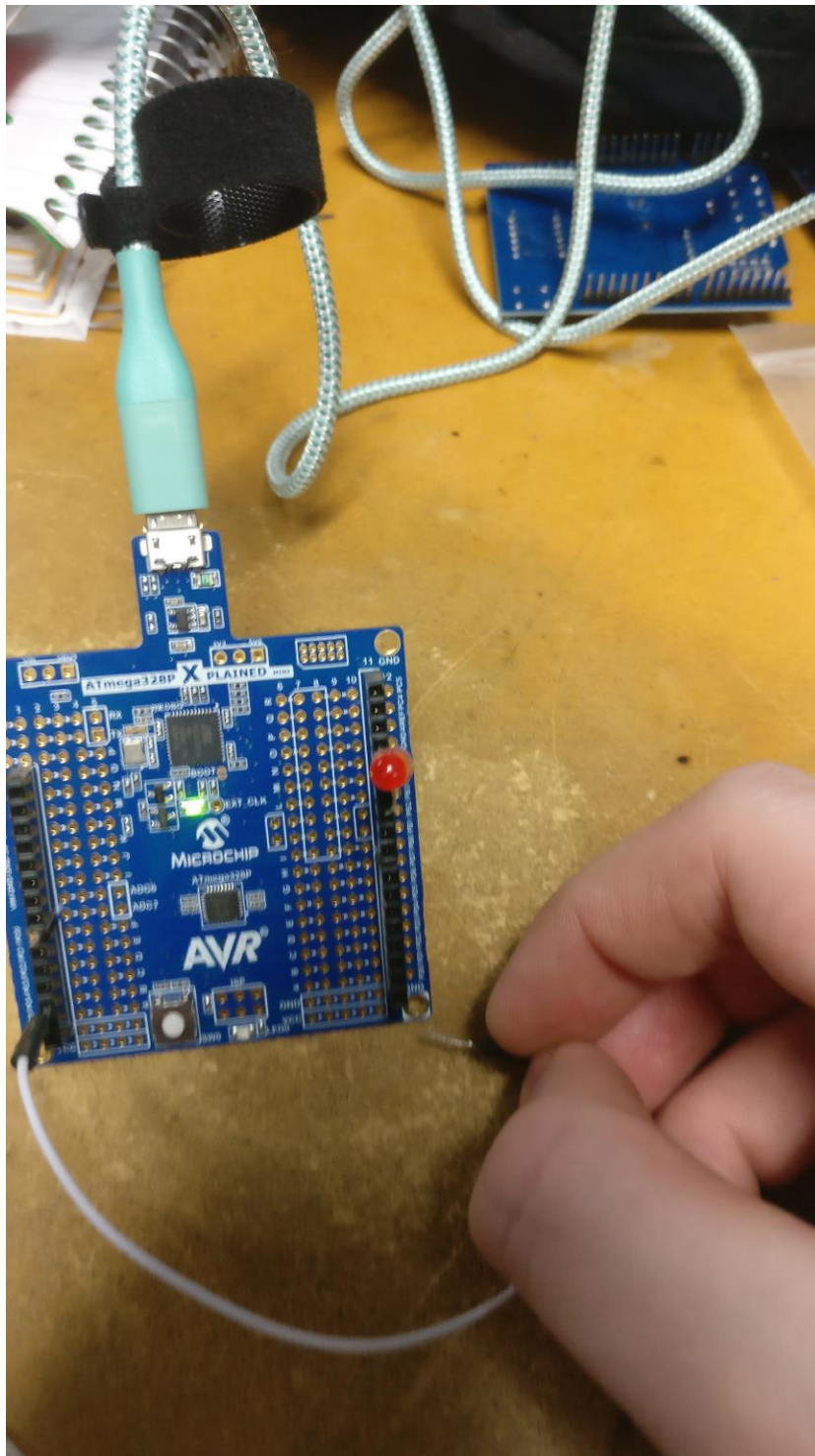
## DA2A\_2\_Assembly





DA2A\_2\_C





**7. VIDEO LINKS OF EACH DEMO**

<https://youtu.be/Rkil6zo1Xk0>  
<https://youtu.be/qAl0zXpwOu8>  
<https://youtu.be/D1yCu39f1JY>  
<https://youtu.be/uUT0h4v8vXQ>

**8. GITHUB LINK OF THIS DA**

<https://github.com/Dil-bert/Alabaster.git>

**Student Academic Misconduct Policy**

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

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

Dillon Archibald