CPE301 – FALL 2019

Design Assignment 2B

Student Name: Dillon Archibald
Student #: 5004439916
Student Email: Archid1@unlv.nevada.edu
Primary Github address: https://github.com/Dil-bert/Alabaster.git
Directory: DA2B

Submit the following for all Labs:

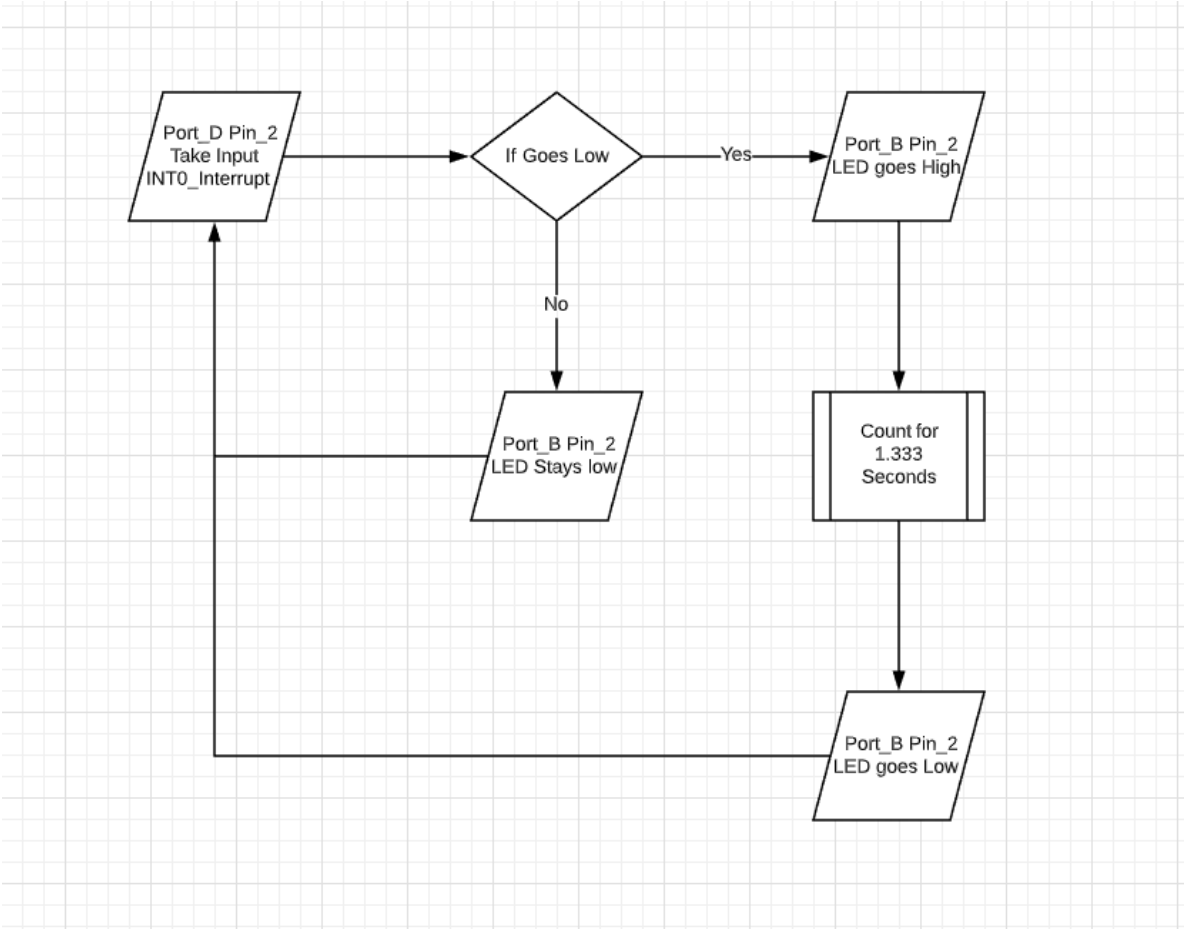
1. In the document, for each task submit the modified or included code (only) with highlights and justifications of the modifications. Also, include the comments.
2. Use the previously create a Github repository with a random name (no CPE/301, Lastname, Firstname). Place all labs under the root folder 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

1x Breadboard
1x Wire
1x 330 ohm Resistor
1x LED (Red = Assembly/Green = C)
1xAtmega328P Xplained Mini

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

ASSEMBLY

```
;
; DA2b_2_assembly.asm
;
; Created: 10/2/2019 6:20:09 PM
; Author : Dilbert
```

```
;
.org 0x00                                   ; Restart interupt insertion location
jmp main                            ;       Jump to main on restart
.org 0x02                                   ; INT0 Interupt insertion location
jmp ext_int                         ;       Jump to Interupt Service Request Function
main:                                       ; Main prog function
LDI R20, HIGH(RAMEND)       ;       *Initalize stack*
   OUT SPH, R20                     ;       *Initalize stack*
LDI R20, LOW(RAMEND)        ;       *Initalize stack*
OUT SPL, R20                        ;       *Initalize stack*
SBI DDRB, 2                         ;       Set PORTB.2 as an output
CBI DDRD, 2                         ;       Set PORTD.2 as an input
SBI PORTD, 2                        ;       Set PORTD.2 as active Low(enable pull up resistor)
LDI R20,1<<INT0                     ;       Load the INT0 Interupt pin location into R20
OUT EIMSK,R20                       ;       Activate the INT0 interupt enable
SEI                                         ;       Activate the Global interupt enable
start:                                      ; Start infinite loop
rjmp start                          ;       Go to infinite loop start location
   rjmp main                        ; Back up jump, in case of skip (unnessesary)
;*****************************End of main prog*********************************
;------------------------------------------------------------------------------
;*****************************INT0 Interupt Service Request Function************
ext_int:                                    ; ISR label
LDI R20, 1<<INTF0           ;       Set Interupt Flag location on R20
OUT EIFR, R20                       ;       Set Interupt Flag
IN R21, PORTB                       ;       load value of PORTB
LDI R22,0X04                        ;       Load 0x04 into R22 = pin 2
EOR R21,R22                         ;       EXOR PORTB.2 and 0000'0100; Load result into R21
OUT PORTB,R21                       ;       Output the result in R21 to PORTB = PORTB.2(HIGH)
CALL delay                          ;       Call Delay function
CBI PORTB,2                         ;       Clear PORTB.2
RETI                                ;       RETURN TO INSTRUCTION AFTER INTERUPT CALL
;*****************************End Interupt Service Request Function*************
;------------------------------------------------------------------------------
;*****************************Begin of Delay Function**************************
delay:
ldi r23, 0x19               ;       Load R23 with 25
d2:                                         ; Loop d2
ldi r24, 0xFF               ;       Load R22 with 255
d1:                                         ; Loop d1
ldi r25, 0xFF               ;       Load R21 with 255
d0:                                         ; Loop d0
nop                                 ;       No operation
nop                                 ;       No operation
nop                                 ;       No operation
nop                                 ;       No operation
nop                                 ;       No operation
nop                                 ;       No operation
```

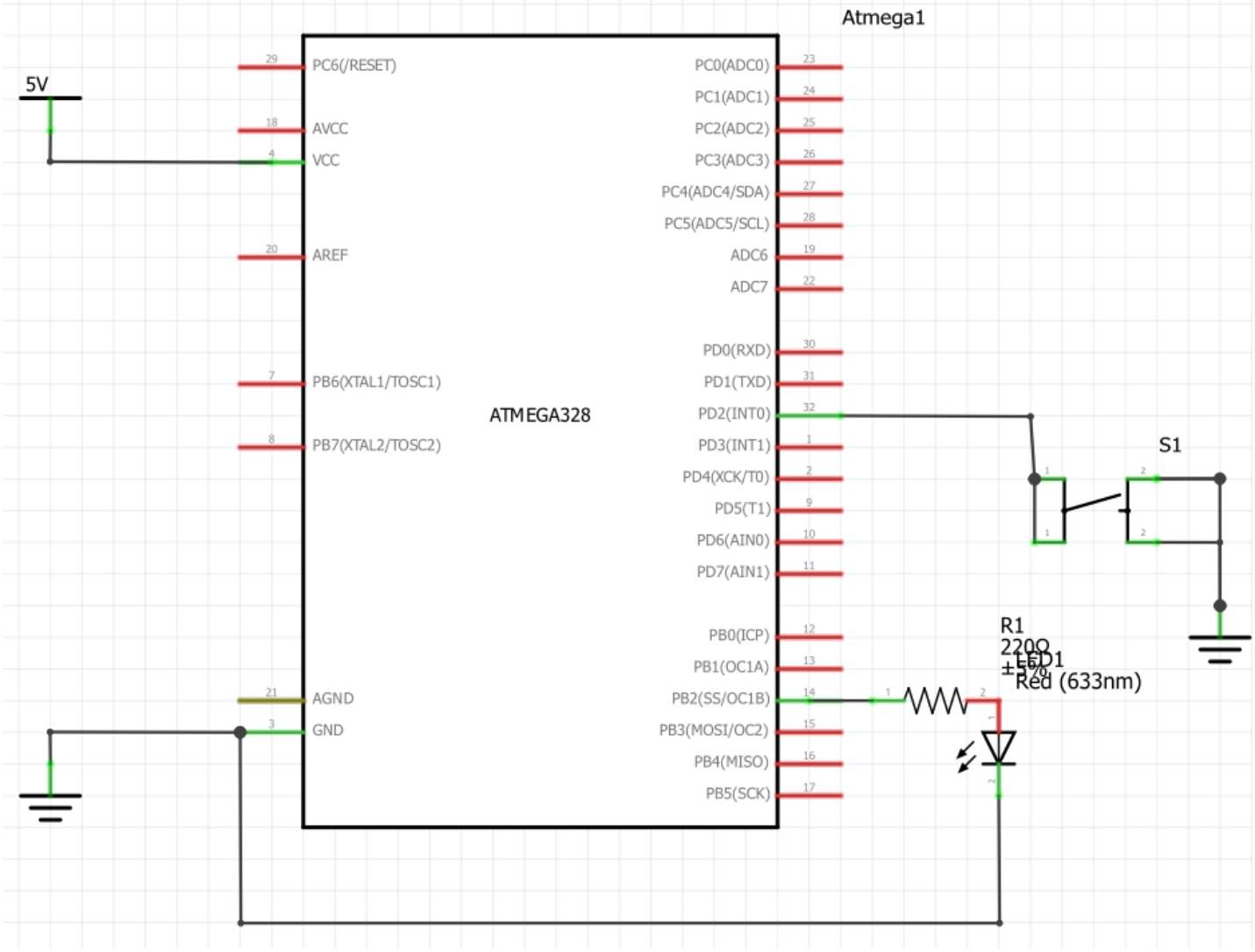
```
nop                                     ;       No operation
nop                                     ;       No operation
nop                                     ;       No operation
nop                                     ;       No operation
dec r25                         ;       Decrement R25
brne d0                         ;       Branch to d0 if R25 is not 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
```

C program

```
/*
* DA2B_2_C.c
*
* Created: 10/3/2019 11:36:25 AM
* Author : Dilbert
*/
#define F_CPU 16000000UL

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

//#define cli() asm volatile("cli"::)
//#define sei() asm volatile("sei"::)
int main(void)
{
DDRB = 1<<2;            //      PB.2 set as output
PORTD = 1<<2;           //      Enable Pull up resistor on PORTD.2
EICRA = 0X2;            //      Set as a falling edge interrupt

EIMSK = (1<<INT0);      //      THIS ENABLES THE INTERUPT "INT0" (note: INT0 is a global constant
from <avr/interrupt.h>)
sei();                          //      set the global interrupt enable
   while (1)                    // General program loop for what ever is needed.
   {
   }
}

ISR (INT0_vect)                 // Interrupt Sub Routine for external interrupt 0
{
PORTB = 1<<2;           // PORTB.2 output high
_delay_ms(1333);        // Delay for 1.333 seconds
PORTB = (0<<2);                 // PORTB.2 output low
}
```

4. SCHEMATICS

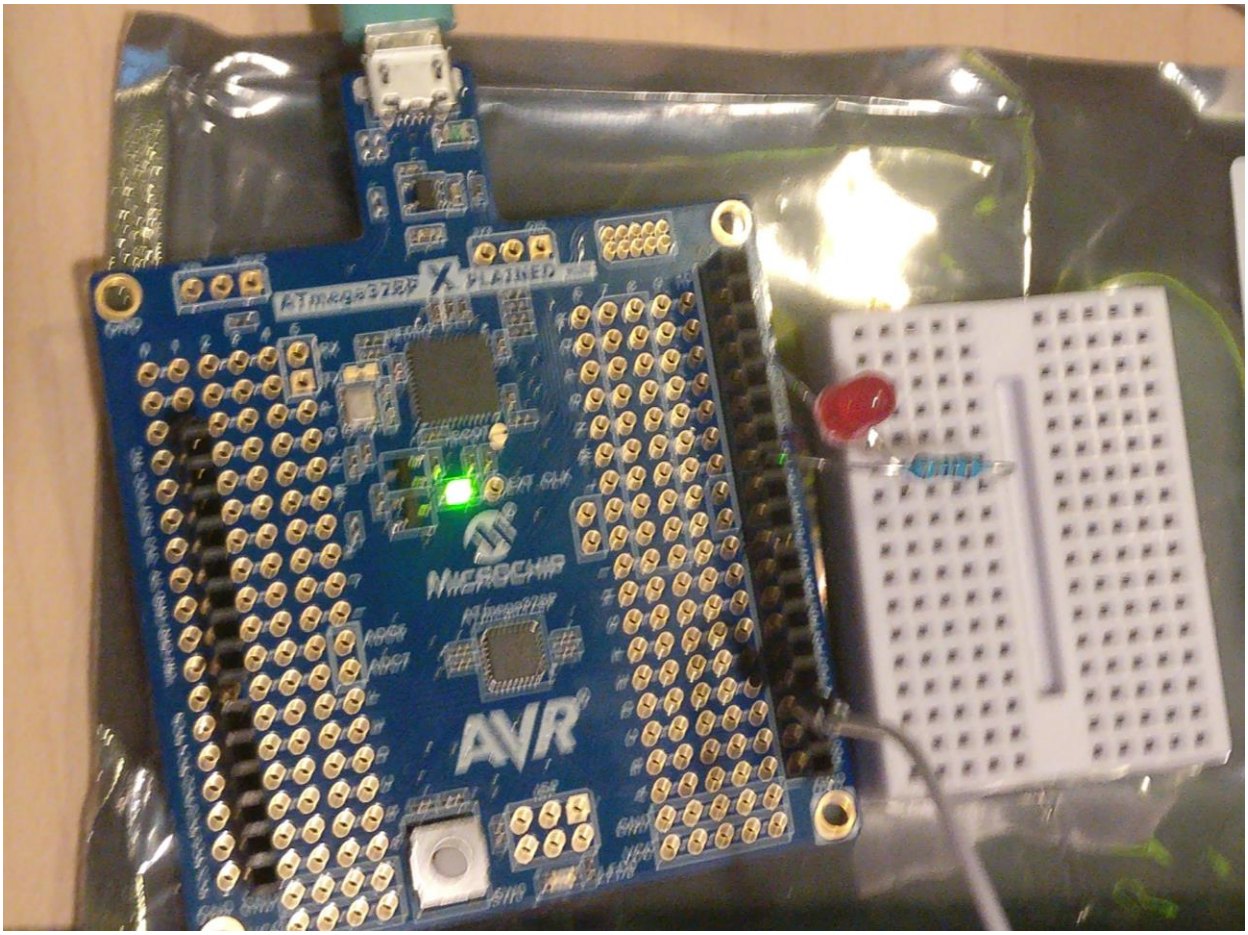
Use fritzing.org



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

N/A?

6. SCREENSHOT OF EACH DEMO (BOARD SETUP)



7. VIDEO LINKS OF EACH DEMO

Assembly example
https://youtu.be/HoMLZ39W4Vs

C-code example
https://youtu.be/R7j6KqwL818

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