CPE301 – SPRING 2024

Design Assignment 2

Student Name: Carlos Funes

Student #: 5007995832

Student Email: [funesc2@unlv.nevada.edu](mailto:funesc2@unlv.nevada.edu)

Primary Github address: <https://github.com/CaFu0320>

Directory:<https://github.com/CaFu0320/submission_da/tree/main/DesignAssignments/DesignAssignment2>

Video Playlist: <https://www.youtube.com/playlist?list=PLa3xS6s509hiXVF-aiEOqfY5oCL0WiYr7>

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**

* ATMEGA328PB MINI BOARD
* FEMALE TO FEMALE CABLE
* MALE TO MALE CABLE
* LOGIC ANALYZER
* ARDUINO SHIELD

1. **C CODE TASK 2/A**

/\*

\* main.c

\*

\* Created: 3/8/2025 3:15:20 PM

\* Author: Carlos Funes

\*/

#define *F\_CPU* 8000000UL

#include <avr/io.h>

#include <util/delay.h>

void delay\_150ms(void) {

volatile *uint16\_t* i = 0;

volatile *uint16\_t* j = 0;

while(i < 660) { //outer loop

j = 0;

while(j < 200) { //inner loop

j++;

}

i++;

}

} //this loop creates 660\*200 = 132k iterations

int main(void) { // main function entry point

DDRC &= ~(1 << 1); //set pc1 for input

PORTC |= (1 << 1); //enable pull-up resistor for pc1

DDRB |= (1 << 5); //set pb5 for output

PORTB &= ~(1 << 5); //make sure led on pb5 is initially off

while(1) { // start infinite loop

if (!(PINC & (1 << 1))) { // check if button at pc1 is pressed (active low)

PORTB |= (1 << 5); // turn on led on pb5

*uint8\_t* count = 10; // we want a total delay of 1.5 seconds (10 x 150 ms)

while(count--) { // loop 10 times

delay\_150ms(); // delay occurs here (~150 ms delay)

}

PORTB &= ~(1 << 5); // turn off led on pb5 after delay

} else {

PORTB &= ~(1 << 5); // ensure led remains off when button is not pressed

}

}

return 0; // end of main function

}

**A screenshot of a computer

AI-generated content may be incorrect.**

**ASSEMBLY PART 1**

;

; DA2PART1ASSEMBLY.asm

;

; Created: 3/20/2025 4:22:28 PM

; Author : Carlos Funes

;

.INCLUDE <M328PBDEF.INC>

.ORG 0 ;RESET LOCATION

JMP MAIN ;JUMPING TO MAIN

DELAY\_150MS: ;DELAY SUBROUTINE

LDI R17, 0x30 ;THIS IS OUTER LOOP

L1:

LDI R18, 0x40 ;MIDDLE LOOP

L2:

LDI R19, 0x60 ;INNER LOOP

L3:

DEC R19 ;DECREMENTING R19

CPI R19, 0 ;

BRNE L3 ;BRANCH TO L3 IF R19 /= 0

DEC R18 ;DECREMENTING R18

CPI R18, 0

BRNE L2 ;BRANCH TO L2 IF R18 /= 0

DEC R17 ;DECREMENTING R17

CPI R17, 0

BRNE L1 ;BRANCH TO L1 IF R17 /= 0

RET ;RETURN TO THE PROGRAM

MAIN:

LDI R20, HIGH(RAMEND) ;STARTING STACK POINTER

OUT SPH, R20

LDI R20, LOW(RAMEND)

OUT SPL, R20

SBI DDRB, 5 ;PORTB5 AS OUTPUT

CBI PORTB, 5 ;PORTB5 INITIALIZED TO OFF

CBI DDRC, 1 ; PORTC1 AS INPUT

SBI PORTC, 1 ;ENABLING PULL UP RESISTORS FOR PUSHBUTTON

WHILE:

SBIC PINC, 1 ;CHECKING IF PUSHBUTTON PC1 IS PRESSED

RJMP CHECKER ;IF BUTTON IS RELEASED

RJMP WHILE ;LOOPING FOREVER IF BUTTON IS NOT PRESSED

CHECKER:

SBIS PINC, 1 ;CHECK IF BUTTON IS NOT PRESSED

RJMP LED\_ON ;IF BUTTON IS PRESSED, JUMP TO LED

RJMP CHECKER ;ONCE BUTTON IS PRESSED, STOP THE LOOP

LED\_ON:

SBI PORTB, 5 ;TURN ON PB5 LED

LDI R16, 0x14 ;1.5 SECOND DELAY

L0:

CALL DELAY\_150MS ;CALLING DELAY SUBROUTINE

DEC R16 ;DECREMENT R16

CPI R16, 0 ;CHECK IF R16 = 0

BRNE L0 ;CONTINUE TO LOOP IF R16 /= 0

CBI PORTB, 5 ;AFTER 1.5 SECONDS, TURN OFF LED

RJMP WHILE ;JUMPING BACK TO THE MAIN LOOP

**A close-up of a computer screen

AI-generated content may be incorrect.**

Insert initial code here

1. **C CODE TASK 3/A**

/\*

\* main.c

\*

\* Created: 3/20/2025 3:15:20 PM

\* Author: Carlos Funes

\*/

#define *F\_CPU* 8000000UL

#include <avr/io.h>

#include <avr/interrupt.h>

#include <util/delay.h>

void delay\_150ms(void) {

volatile *uint16\_t* i = 0;

volatile *uint16\_t* j = 0;

while(i < 660) { // outer loop remains at 660 iterations

j = 0;

while(j < 200) { // inner loop: 200 iterations

j++;

}

i++;

}

}

ISR(INT0\_vect) {

// Interrupt-based handling: turn on LED and delay for 3 sec.

PORTB |= (1 << 5); // Turn on LED at PB5

for(int k = 0; k < 10; k++){

delay\_150ms();

} // Delay 3 seconds (blocking delay)

PORTB &= ~(1 << 5); // Turn off LED

}

int main(void) {

// Configure LED output

DDRB |= (1 << 5); // Set PB5 as output

PORTB &= ~(1 << 5); // Make sure LED is initially off

// Configure switch on INT0 (PD2)

DDRD &= ~(1 << 2); // Set PD2 as input

PORTD |= (1 << 2); // Enable internal pull-up on PD2

// Configure INT0 to trigger on a rising edge (active high)

EICRA |= (1 << ISC01) | (1 << ISC00); // Rising edge trigger for INT0

EIMSK |= (1 << INT0); // Enable external interrupt INT0

sei(); // Enable global interrupts

while(1) {

// this is just to keep the microcontroller running

}

return 0;

}

**A screenshot of a computer program

AI-generated content may be incorrect.**

**ASSEMBLY PART 2**

;

; DA2PART2ASSEMBLY.asm

;

; Created: 3/20/2025 5:06:26 PM

; Author : Carlos Funes

;

; Replace with your application code

.INCLUDE <M328PBDEF.INC>

.ORG 0 ;RESET LOCATION

JMP MAIN ;JUMPING TO MAIN

.ORG 0x02 ;INT0 VECTOR

JMP ISR\_INT0VECTOR

DELAY\_150MS: ;DELAY SUBROUTINE

LDI R17, 0x30 ;THIS IS OUTER LOOP

L1:

LDI R18, 0x40 ;MIDDLE LOOP

L2:

LDI R19, 0x60 ;INNER LOOP

L3:

DEC R19 ;DECREMENTING R19

CPI R19, 0 ;

BRNE L3 ;BRANCH TO L3 IF R19 /= 0

DEC R18 ;DECREMENTING R18

CPI R18, 0

BRNE L2 ;BRANCH TO L2 IF R18 /= 0

DEC R17 ;DECREMENTING R17

CPI R17, 0

BRNE L1 ;BRANCH TO L1 IF R17 /= 0

RET ;RETURN TO THE PROGRAM

MAIN:

LDI R20, HIGH(RAMEND) ;STARTING STACK POINTER

OUT SPH, R20

LDI R20, LOW(RAMEND)

OUT SPL, R20

LDI R20, 0x03 ;MAKING INT0 BE TRIGGERED AT RISING EDGE, LAST 2 BITS MAKE THIS HAPPEN

STS EICRA, R20 ;STORING R20 INTO EICRA

SBI EIMSK, INT0 ;ENABLING INT0 INTERRUPT

SEI

SBI DDRB, 5 ;PORTB5 AS OUTPUT

CBI PORTB, 5 ;PORTB5 INITIALIZED TO OFF

CBI DDRD, 2 ; PORTD2 AS INPUT

SBI PORTD, 2 ;ENABLING PULL UP RESISTORS FOR PUSHBUTTON

WHILE:

CBI PORTB, 5 ;LED IS OFF WHILE IN THE LOOP

RJMP WHILE ;LOOP FOREVER

ISR\_INT0VECTOR:

SBI PORTB, 5 ;TURN ON PB5 LED

LDI R16, 0x14 ;1.5 SECOND DELAY

LJ:

CALL DELAY\_150MS ;CALLING DELAY SUBROUTINE

DEC R16 ;DECREMENT R16

CPI R16, 0 ;CHECK IF R16 = 0

BRNE LJ ;CONTINUE TO LOOP IF R16 /= 0

CBI PORTB, 5 ;AFTER 1.5 SECONDS, TURN OFF LED

RETI ;RETURNING FROM INTERRUPT

**A screenshot of a computer

AI-generated content may be incorrect.**

1. **C CODE TASK 2 AND 3**

/\*

\* DA2PART3CCODE.c

\*

\* Created: 3/20/2025 4:13:42 PM

\* Author : Carlos Funes

\*/

//Question 3)

#define *F\_CPU* 8000000UL

#include <avr/io.h>

#include <avr/interrupt.h>

#include <util/delay.h>

void delay\_150ms(void) {

volatile *uint16\_t* i = 0;

volatile *uint16\_t* j = 0;

while(i < 660) { //outer loop remains at 660 iterations

j = 0;

while(j < 200) { //inner loop: 200 \* 20 = 4000 iterations

j++;

}

i++;

}

}

ISR(INT0\_vect) {

//interrupt-based handling: turn on LED and delay for 3 sec.

PORTB |= (1 << 5); //turn on LED at PB5

for(int k = 0; k < 10; k++ ){

delay\_150ms();

}

//delay 3 seconds (blocking delay)

PORTB &= ~(1 << 5); //turn off LED

}

int main(void) {

DDRB |= (1 << 5); //set PB5 as output

PORTB &= ~(1 << 5); //make sure LED is initially off

DDRC &= ~(1 << 1); //set pc1 for input

PORTC |= (1 << 1); //enable pull-up resistor for pc1

DDRD &= ~(1 << 2); //set PD2 as input

PORTD |= (1 << 2); //enable internal pull-up on PD2

EICRA |= (1 << ISC01) | (1 << ISC00); //rising edge trigger for INT0

EIMSK |= (1 << INT0); //enable external interrupt INT0

sei(); //enable global interrupts

while(1) { //start infinite loop

if (!(PINC & (1 << 1))) { //check if button at pc1 is pressed (active low)

PORTB |= (1 << 5); //turn on led on pb5

*uint8\_t* count = 10; //we want a total delay of 1.5 seconds (10 x 150 ms)

while(count--) { //loop 10 times

delay\_150ms(); //delay occurs here (~150 ms delay)

}

PORTB &= ~(1 << 5); //turn off led on pb5 after delay

} else {

PORTB &= ~(1 << 5); //ensure led remains off when button is not pressed

}

}

return 0;

}

**A screenshot of a computer

AI-generated content may be incorrect.**

**ASSEMBLY PART 3**

;

; DA2PART3ASSEMBLY.asm

;

; Created: 3/20/2025 5:16:02 PM

; Author : Carlos Funes

;

.INCLUDE <M328PBDEF.INC>

.ORG 0 ;RESET LOCATION

JMP MAIN ;JUMPING TO MAIN

.ORG 0x02 ;INT0 VECTOR

JMP ISR\_INT0VECTOR

DELAY\_150MS:

LDI R17, 0x30 ;THIS IS OUTER LOOP

L1:

LDI R18, 0x40 ;MIDDLE LOOP

L2:

LDI R19, 0x60 ;INNER LOOP

L3:

DEC R19 ;DECREMENTING R19

CPI R19, 0 ;

BRNE L3 ;BRANCH TO L3 IF R19 /= 0

DEC R18 ;DECREMENTING R18

CPI R18, 0

BRNE L2 ;BRANCH TO L2 IF R18 /= 0

DEC R17 ;DECREMENTING R17

CPI R17, 0

BRNE L1 ;BRANCH TO L1 IF R17 /= 0

RET ;RETURN TO THE PROGRAM

MAIN:

LDI R20, HIGH(RAMEND) ;STARTING STACK POINTER

OUT SPH, R20

LDI R20, LOW(RAMEND)

OUT SPL, R20

LDI R20, 0x03 ;MAKING INT0 BE TRIGGERED AT RISING EDGE, LAST 2 BITS MAKE THIS HAPPEN

STS EICRA, R20 ;STORING R20 INTO EICRA

SBI EIMSK, INT0 ;ENABLING INT0 INTERRUPT

SEI

SBI DDRB, 5 ;PORTB5 AS OUTPUT

CBI PORTB, 5 ;PORTB5 INITIALIZED TO OFF

CBI DDRD, 2 ; PORTD2 AS INPUT

SBI PORTD, 2 ;ENABLING PULL UP RESISTORS FOR PUSHBUTTON

CBI DDRC, 1 ; PORTC1 AS INPUT

SBI PORTC, 1 ;ENABLING PULL UP RESISTORS FOR PUSHBUTTON

LDI R20,(1<<ISC01)|(1<<ISC00) ;SETTING INT0 RISING EDGE TRIGGER

STS EICRA,R20 ;STORING TO INT CONTROL REGISTER

SBI EIMSK,INT0 ;ENABLING INT0 INTERRUPT

SEI ;GLOBAL INTERRUPT

POLL\_LOOP:

SBIC PINC,1 ;CHECKING IF BUTTON PC1 PRESSED

RJMP POLL\_LOOP ;KEEP POLLING IF NOT PRESSED

CALL DELAY\_150MS ;DEBOUNCE DELAY 0.15 SEC

WAIT\_RELEASE:

SBIS PINC,1 ;CHECK IF BUTTON RELEASED

RJMP WAIT\_RELEASE ;WAIT UNTIL BUTTON RELEASED

SBI PORTB,5 ;TURN ON LED

LDI R16,20 ;SET 1.5 SEC DELAY

DELAY\_1\_5S:

CALL DELAY\_150MS ;CALLING 0.15 SEC DELAY

DEC R16 ;DECREMENT COUNTER

BRNE DELAY\_1\_5S ;REPEAT 20 TIMES

CBI PORTB,5 ;TURN OFF LED

RJMP POLL\_LOOP ;RETURN TO POLLING LOOP

ISR\_INT0VECTOR:

SBI PORTB,5 ;TURN ON LED ON INTERRUPT

PUSH R16 ;SAVE REGISTER STATE

LDI R16,20 ;SET 3 SEC DELAY

DELAY\_3S:

CALL DELAY\_150MS ;CALLING 0.15 SEC DELAY

DEC R16 ;DECREMENT COUNTER

BRNE DELAY\_3S ;REPEATING 20 TIMES

CBI PORTB,5 ;TURN OFF LED

POP R16 ;RESTORING REGISTER STATE

RETI ;RETURN FROM INTERRUPT

**A screenshot of a computer

AI-generated content may be incorrect.**

1. **SCHEMATICS**

A computer screen shot of a computer

AI-generated content may be incorrect.

A diagram of a circuit board

AI-generated content may be incorrect.

1. **SCREENSHOTS OF EACH TASK OUTPUT (LOGIC ANALYZER)**

Task 2:

A black screen with many colored lines

AI-generated content may be incorrect.

Task 3:

A black screen with many colored lines

AI-generated content may be incorrect.

Task 2 & 3 Polling and Interrupt method

A black screen with many colored lines

AI-generated content may be incorrect.

1. **SCREENSHOT OF EACH DEMO (BOARD SETUP)**

Task 2:

A hand holding a blue circuit board

AI-generated content may be incorrect.

Task 3:

A person holding a wire

AI-generated content may be incorrect.

1. **VIDEO LINKS OF EACH DEMO**

Task 2:

<https://www.youtube.com/watch?v=6eriUlZqST4&list=PLa3xS6s509hiXVF-aiEOqfY5oCL0WiYr7&index=3&ab_channel=CarlosGordfgg>

Task 3:

<https://www.youtube.com/watch?v=prNirN60sFA&list=PLa3xS6s509hiXVF-aiEOqfY5oCL0WiYr7&index=1&ab_channel=CarlosGordfgg>

Task 2 & 3 Polling and Interrupt method:

<https://www.youtube.com/watch?v=uCc1aDGF9xg&list=PLa3xS6s509hiXVF-aiEOqfY5oCL0WiYr7&index=3&ab_channel=CarlosGordfgg>

1. **GITHUB LINK OF THIS DA**

Task 2 c code: <https://github.com/CaFu0320/submission_da/tree/main/DesignAssignments/DesignAssignment2/DA2PART1CCODE>

Task 2 assembly code: <https://github.com/CaFu0320/submission_da/tree/main/DesignAssignments/DesignAssignment2/DA2PART1ASSEMBLY>

Task 3 c code: <https://github.com/CaFu0320/submission_da/tree/main/DesignAssignments/DesignAssignment2/DA2PART2CCODE>

Task 3 assembly code:

<https://github.com/CaFu0320/submission_da/tree/main/DesignAssignments/DesignAssignment2/DA2PART2ASSEMBLY>

Task 2 and 3 c code: <https://github.com/CaFu0320/submission_da/tree/main/DesignAssignments/DesignAssignment2/DA2PART3CCODE>

Task 2 and 3 assembly:

<https://github.com/CaFu0320/submission_da/tree/main/DesignAssignments/DesignAssignment2/DA2PART3ASSEMBLY>

**Student Academic Misconduct Policy**

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

“This assignment submission is my own, original work”.

NAME OF THE STUDENT