CPE301 – SPRING 2019

Design Assignment X

Student Name: Shaquille Regis

Student #: 2000686590

Student Email: regis@unlv.nevada.edu

Primary Github address:

Directory:

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

ATmega328P-XMINI Microcontroller

Arudino Multi-function shield

1. **INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1/A**

Insert initial code here (Assembly)

;

; DA2A.asm

;

; Created: 3/2/2019 1:21:47 PM

; Author : regis

;

; Replace with your application code

start:

.org 0

LDI R16,4

SBI DDRB, 0x2 ;PB2 as output

LDI R17,0 ;needed to toogle led

out PORTB,R17

LDI R20,5 ;to set prescaler

STS TCCR1B,R20 ;Prescaler: 1024

LDI R18, 0

begin:

RCALL delay ;calling timer to wait for 1 sec

EOR R17,R16 ;XOR to toogle led

out PORTB,R17

RCALL delay2 ;calling timer to wait for 1 sec

EOR R17,R16 ;XOR to toogle led

out PORTB,R17

RJMP begin ;repeating i.e, while(1)

delay:

LDS R29, TCNT1H ;loading upper bit of counter to R29

LDS R28, TCNT1L ;loading lower bit of counter to R28

CPI R28,0x8B ;comparing if lower is 0xC6

BRSH body

RJMP delay

body:

CPI R29,0x1A

BRSH done

RJMP delay

delay2:

LDS R29, TCNT1H ;loading upper bit of counter to R29

LDS R28, TCNT1L ;loading lower bit of counter to R28

CPI R28,0xB2 ;comparing if lower is 0xC6

BRSH body2

RJMP delay2

body2:

CPI R29,0x11

BRSH done

RJMP delay2

done:

LDI R20,0x00

STS TCNT1H,R20 ;resetting the counter to 0 for next round

LDI R20,0x00

STS TCNT1L,R20 ;resetting the counter to 0 for next round

RET

Code in C:

#include <avr/io.h> //This contains definitions for all the registers locations and some

// other things, must always be included

#define *F\_CPU* 16000000UL //F\_CPU tells the compiler that our crystal is an 16Mhz one so it

// can generate an accurate delay, must be declared above delay so

// delay knows what is the value of F\_CPU

#include <util/delay.h> //Contains some delay functions that will generate accurate delays

// of ms and us

int main(void){ //In ANSI C, the main function as always an int return and using

// void will give you an warning

DDRB |= (1<<PB2); //Define PORTB2 as an output so we can blink our led

while(1){

PORTB |= (1<<PB2); //Turn led on

*\_delay\_ms*(290);//290); //Wait

PORTB &= ~(1<<PB2); //Turn led off

*\_delay\_ms*(435); //Wait

}

return 1;

}

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

Insert only the modified sections here

#include <avr/io.h>

#define *F\_CPU* 16000000UL

#include <util/delay.h>

int main(void)

{

DDRB |= (1<<2); //Define PORTB2 as an output so we can blink our led

DDRC &= (0<<2);

PORTC |= (1<<2);

while(1){

//PORTB |= (1<<PB2); //Turn led off

//\_delay\_ms(290);//290); //Wait 1 second

if(PINC & 0x2){

*\_delay\_ms*(1250); //Wait another second

PORTB |= (1<<PB2);

}

else

PORTB &= ~(1<<PB2); //Turn led on

}

return 0;

}

start:

SBI DDRB, 0x4

CBI DDRC, 0x2

LDI r16, 0x00

OUT PORTB, r16

loop:

IN r18, PINC

CP r16, r18

BRNE lightOn

rjmp loop

lightOn:

LDI r18, 0xff

OUT PORTB, r18

RCALL delay

rjmp start

delay:

LDS R29, TCNT1H ;loading upper bit of counter to R29

LDS R28, TCNT1L ;loading lower bit of counter to R28

CPI R28,0x8B ;comparing if lower is 0xC6

BRSH body

RJMP delay

body:

CPI R29,0x1A

BRSH done

RJMP delay

1. **SCHEMATICS**

Use fritzing.org

1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**
2. **SCREENSHOT OF EACH DEMO (BOARD SETUP)**

See youtube videos for Board setup.

1. **VIDEO LINKS OF EACH DEMO**

Part 1:

https://www.youtube.com/watch?v=VDNGyuaauIk

Part 2:

https://www.youtube.com/watch?v=W3H7n59MvTM

1. **GITHUB LINK OF THIS DA**

**Student Academic Misconduct Policy**

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

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

NAME OF THE STUDENT