CPE301 – SPRING 2019

Design Assignment 2C

Student Name: Armon Latifi
Student #: 2000698173
Student Email: latifa1@unlv.nevada.edu
Primary Github address: https://github.com/armonlatifi
Directory: https://github.com/armonlatifi/sub_da/tree/master/DA2C

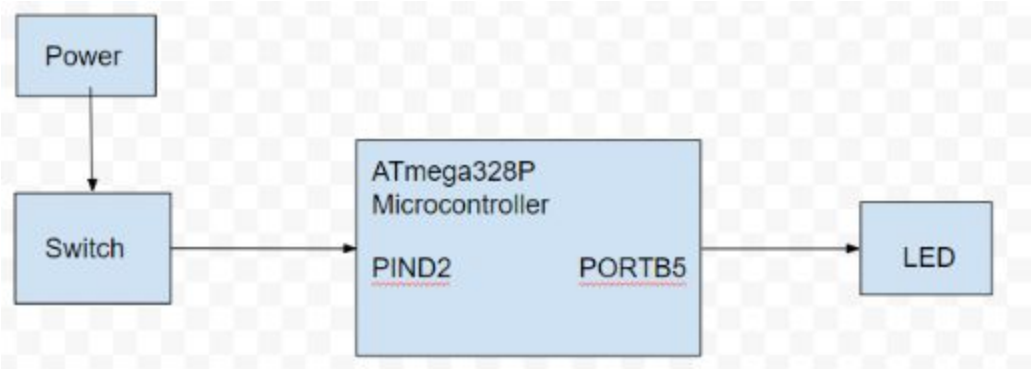
Submit the following for all Labs:

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:

- Assembler
- Simulator
- Debugger
- Breadboard
- Atmega328P
- Wires
- Microusb cord
- Atmel studio 7
- Arudino Multi-function shield



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

part_1.asm

```
.org 0
        LDI R16, 32
        SBI DDRB, 5                 ;set port b to output
        LDI R17, 0
        OUT PORTB, R17 ;initialize
        LDI R20, 5
        STS TCCR0B, R20             ;prescaler --> 1024
        LDI R20, 0xF3
        STS OCR0A, R20

loop_1:
        RCALL d_loop                ;delay loop
        EOR R17, R16
        OUT PORTB, R17              ;set port b to output
        RJMP loop_1

d_loop:
        LDS R19, TCNT0
        CPI R19, 0xF3           ;equality check
        BRSH done
        RJMP d_loop                 ;jump back
```

```
done:
	LDI R20, 0x00
	STS TCNT0, R20			;reset counter to 0
	RET
```

part_1.c

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

int main(void)
{
	DDRB = 32;			//port b is output
	TCCR0B = 13;	//prescaler --> 1024
	TCNT0 = 0;
	OCR0A = 0x00F3;		//top value

	while (1)
	{
		if((TIFR0 & 0b00000001) == 0b00000001) //overflow condition
		{
			PORTB = 0xFF;		//port b is high
			_delay_ms(250);
			TCNT0 = 0;
		}
		else
		PORTB = 0x00;
	}
}
```

part_2.asm

```
.ORG 0x00
	JMP MAIN
.ORG 0x20
	JMP ISR_loop

MAIN:
	LDI R20, HIGH(RAMEND)
	OUT SPH, R20			;high address
	LDI R20, LOW(RAMEND)
	OUT SPL, R20			;low address
	LDI R17, 0
	SBI DDRB, 5				;port b is output
	LDI R20, 13
	STS TCCR0B, R20			;prescaler --> 1024
	LDI R20, 71
	STS OCR0A, R20			;top value
	LDI R20, (1 << TOIE0)
	OUT TIFR0, R20
	SEI						;interrupt

begin:
	RJMP begin

ISR_loop:
```

```
        LDI R20, (1 << TOIE0)    ;flag bit
        OUT TIFR0, R20                  ;clear the flag
        LDI R16, 32

        EOR R17, R16
        OUT PORTB, R17                  ;port be is output
        LDI R18, 0xF3                   ;loop

loop_1:
        SUBI R18, 1
        CPI R18, 0                      ;R18 = 0
        BRNE loop_1

        LDI R20, 0x00
        STS TCNT0, R20
        RETI
```

part_2.c

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

ISR(TIMER0_OVF_vect) //interrupt
{
        PORTB ^= 0xFF;                  //port b is output
        TCNT0 = 65292l;                 //reset
}

int main(void)
{
        DDRB = 0xFF;                    //port b set to high
        TIMSK0 = (1 << TOIE0);
        TCCR0B = 0x05;                  //prescaler --> 1024
        TCNT0 = 65292;
        sei();
        while(1)
        {
        }
}
```

part_3.asm

```
.ORG 0x00
        JMP main_loop_1
.ORG 0x06
        JMP ex0_ISR

main_loop_1:
        LDI R20, HIGH(RAMEND)
        OUT SPH, R20                    ;high addy
        LDI R20, LOW(RAMEND)
        OUT SPL, R20                    ;low addy
        SBI DDRB, 5                     ;port b set to output

        LDI R17, 0
        LDI R20, (1 << INT0)
```

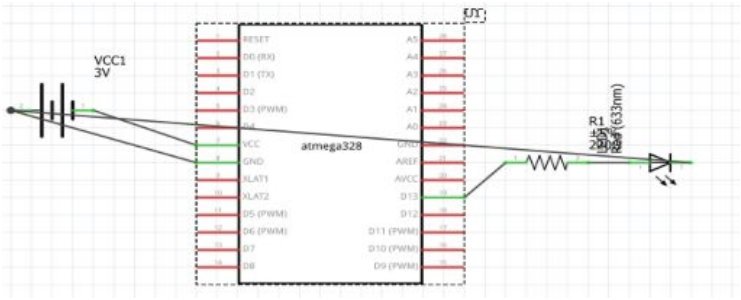
```
        OUT EIMSK, R20                          ;flag for interrupt
        SEI

here:
        JMP here

ex0_ISR:
        LDI R20, (1 << INTF0)    ;clear
        LDI R16, 32
        EOR R17, R16
        OUT PORTB, R17                          ;port b is output
        LDI R18, 0xF3

loop_1:
        SUBI R18, 1
        CPI R18, 0x00                   ;check with zero
        BRNE loop_1
        LDI R20, 0x00
        STS TCNT0, R20
        RETI
```

part_3.c

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

ISR(TIMER0_COMPA_vect)
{
        PORTB = 0xFF;  //set port b to max
        _delay_ms(250);
        PORTB = 0x00;  //clear port b to zero
}


int main(void)
{
        DDRB = 0xFF;    //port b is set to output
        EICRA = 0x03;   //rising edge
        EIMSK = (1 << INT0);
        EIFR = (1 << INTF0);       //clear flag
        sei();                     //set interrupts

        while (1)
        {
        }

}
```

3. SCHEMATICS

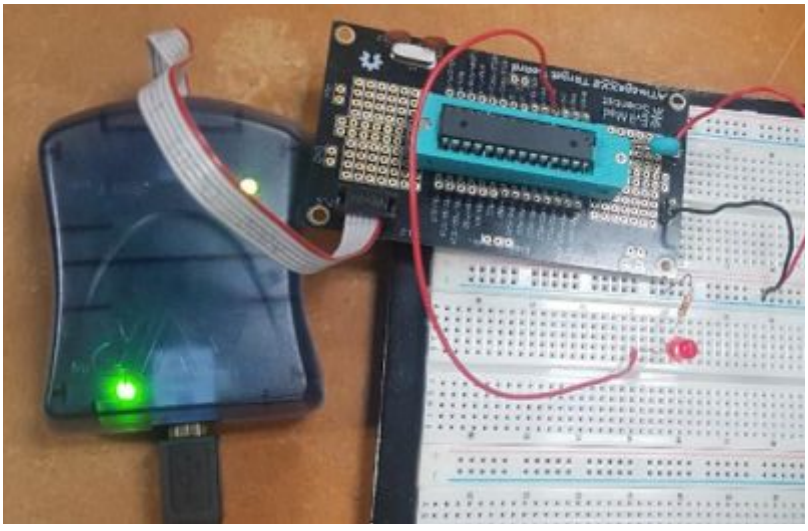
Use fritzing.org

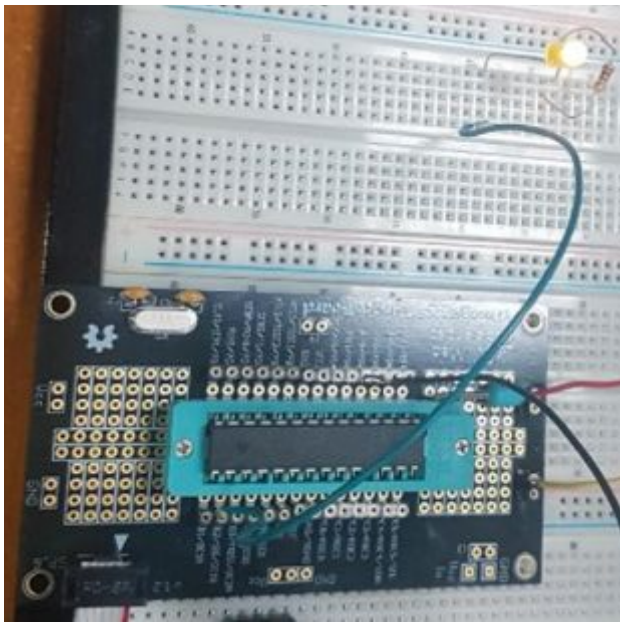


4. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)

Cycle Counter	12	Cycle Counter	12
Frequency	16.000 MHz	Frequency	16.000 MHz
Stop Watch	0.75 µs	Stop Watch	0.75 µs

5. SCREENSHOT OF EACH DEMO (BOARD SETUP)





6. GITHUB LINK OF THIS DA

https://github.com/armonlatifi/sub_da/tree/master/DA2C

Student Academic Misconduct Policy

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

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

Armon Latifi