CPE301 – SPRING 2019

Design Assignment 2A Task 1

Student Name: Saul Alejandro Mendoza Guzman

Student #: 2000540481

Student Email: mendos1@unlv.nevada.edu

Primary Github address: https://github.com/mendos1/subnission\_da

Directory: /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

Block diagram with pins used in the Atmega328P

* Atmega328p xplained mini
* Multifunction shield

LED

Multi

Function

shield

328p xplained

mini

POWER   
SUPPLY

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

.org 0x0000

;Initialize the microcontroller stack pointer

ldi r16, 0x24 ; setting pd2

out DDRB, r16 ; enableing pd2 as output

ldi r16, 0x04 ; setting pd2

ldi r17, 0x20 ; used to set/reset pd5

//out PORTB, r17 ; setting port to off

ldi r20, 0x04 ; setting up TCCR1B = 00000101 for 256

sts TCCR1B, r20

begin:

ldi r20, 0x00 ; setting up timer counter to 0

sts TCNT1H, r20 ; Timer counter is 16 bits

sts TCNT1L, r20 ; Thus, requires 2 8 bit regs

rjmp delay\_onnnnn ; routine for on delay (remember multi function

; table is reverse logic

returnOne:

eor r17, r16 ; xor to toggle LED

out PORTB, r17 ; enable pd5

ldi r20, 0x00 ; setting up timer counter to 0

rjmp delay\_offffff ; routine call for 4 second delay see below

returnTwo:

eor r17, r16 ; xor to toggle LED

out PORTB, r17 ; enable pd5

rjmp begin ; repeat main loop

delay\_onnnnn:

lds r29, TCNT1H ; load upper bytes of timer counter to r29

lds r28, TCNT1L ; load lower bytes of timer counter to r28

cpi r28, 0xCC ; check to see if lower 8 bits of timer counter are 0x08

brsh body1

rjmp delay\_onnnnn ; otherwise keep checking lower bytes

body1:

cpi r29, 0x46 ; check if upper timer counter have reached desired value

brlt delay\_onnnnn ; otherwise recheck the lower bytes

rjmp returnOne

delay\_offffff:

lds r29, TCNT1H ; load upper bytes of timer counter to r29

lds r28, TCNT1L ; load lower bytes of timer counter to r28

cpi r28, 0x33 ; check to see if lower 8 bits of timer counter are 0x08

brsh body2

rjmp delay\_offffff ; otherwise keep checking lower bytes

body2:

cpi r29, 0x6A ; check if upper timer counter have reached desired value

brlt delay\_offffff ; otherwise recheck the lower bytes

rjmp returnTwo

1. **C VERIFICATION**

Insert only the modified sections here

#define F\_CPU 16000000UL

#include <avr/io.h>

#include <avr/interrupt.h>

int main(void){

DDRB = 0X24; //Set the PB2 and PB5 as an outputs

PORTB |= (1<<5); // turn off PB5

TCCR1B =4; // set prescalar to 256

while(1){

TCNT1 = 0; // reset timer counter

while(TCNT1 != 27187){} // on delay

PORTB ^= 0X04; // toggle LED

TCNT1 = 0; // reset timer counter

while(TCNT1 != 18124){} // off delay

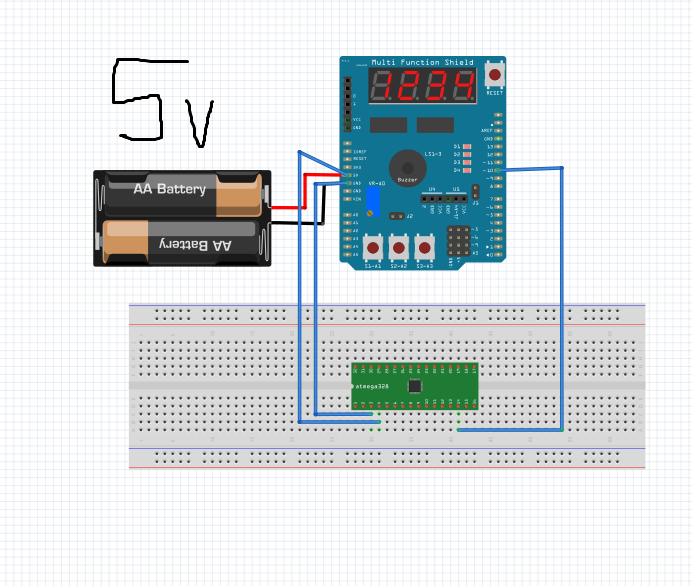
PORTB ^= 0X04; // toggle LED

}

}

1. **SCHEMATICS**

Use fritzing.org



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

<https://www.youtube.com/watch?v=uebhY2LG9Is>

1. **GITHUB LINK OF THIS DA**

<https://github.com/mendos1/subnission_da>

**Student Academic Misconduct Policy**

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

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

NAME OF THE STUDENT