CPE301 – SPRING 2019

Design Assignment 2C

Student Name: Saul Alejandro Mendoza Guzman

Student #: 2000540481

Student Email: mendos1@unlv.nevada.edu

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

Directory: 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**

* Atmel studio 7
* Atmega328P xplained mini
* Multifunction shield

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

**#define F\_CPU 16000000UL**

**#include <avr/io.h>**

**// THIS IS TASK1 PART1**

**#include <util/delay.h>**

**#include <avr/interrupt.h>**

**int main(void){**

**DDRB = 36;**

**PORTB |= (1<<5);**

**TCCR0B = 5;**

**while(1){**

**TCNT0 = 0;**

**int TACO = 0;**

**while(TACO != 27){**

**while(TCNT0 != 255);**

**TCNT0 = 0;**

**TACO++;**

**}**

**PORTB ^= (1<<2);**

**TACO = 0;**

**TCNT0 = 0;**

**while(TACO != 18){**

**while(TCNT0 != 255);**

**TCNT0 = 0;**

**TACO++;**

**}**

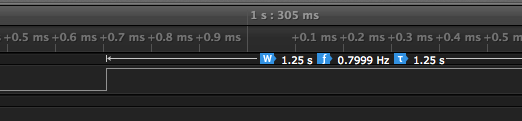
**PORTB ^= (1<<2);**

**TACO = 0;**

**TCNT0 = 0;**

**}**

**}**

****

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

#define F\_CPU 16000000UL

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

// global variable to count the number of overflows

volatile uint8\_t tof\_detected;

// TIMER0 overflow interrupt service routine

// called whenever TCNT0 overflows

ISR(TIMER0\_OVF\_vect){

// keep a track of number of overflows

tof\_detected++;

}

void timer0\_init(){

// set up timer with prescaler = 1024

TCCR0B |= (1 << CS02)|(1 << CS00);

// initialize counter

TCNT0 = 0;

TIMSK0 |= (1 << TOIE0);

sei();

tof\_detected = 0;

}

void of\_funct(unsigned int of\_detection\_num, unsigned int tcount){

while(tof\_detected != 0){

// check if no. of overflows = of\_detection\_num

if (tof\_detected >= of\_detection\_num){

// NOTE: '>=' is used

// check if the timer count reaches tcount

if (TCNT0 >= tcount){

PORTB ^= (1 << 2); // toggles the led

TCNT0 = 0; // reset counter

tof\_detected = 0; // reset overflow detection counter

}

}

}

}

int main(void){

// connect led to pin PB2

DDRB |= (1 << 2);

DDRB |= (1 << 5);

PORTB ^= (1 << 5); // toggles the led off

// initialize timer

timer0\_init();

// loop forever

while(1){

if(!(PINC & (1<<PINC1))){

PORTB ^= (1 << 2); // toggles the led off

of\_funct(78, 165);

}

else

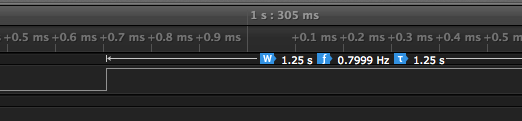
PORTB |= (1<<2); // OTHERWISE STAY OFF

//of\_funct(26, 165);

//of\_funct(17, 195);

}

}

****

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

**#define F\_CPU 16000000UL**

**#include <avr/io.h>**

**#include <util/delay.h>**

**#include <avr/interrupt.h>**

**// global variable to count the number of overflows**

**volatile uint8\_t tof\_detected;**

**// TIMER0 overflow interrupt service routine**

**// called whenever TCNT0 overflows**

**ISR(TIMER0\_OVF\_vect){**

**// keep a track of number of overflows**

**tof\_detected++;**

**}**

**void timer0\_init(){**

**// set up timer with prescaler = 1024**

**TCCR0B |= (1 << CS02)|(1 << CS00);**

**// initialize counter**

**TCNT0 = 0;**

**TIMSK0 |= (1 << TOIE0);**

**sei();**

**tof\_detected = 0;**

**}**

**void of\_funct(unsigned int of\_detection\_num, unsigned int tcount){**

**// check if no. of overflows = of\_detection\_num**

**if (tof\_detected >= of\_detection\_num){**

**// NOTE: '>=' is used**

**// check if the timer count reaches tcount**

**if (TCNT0 >= tcount){**

**PORTB ^= (1 << 2); // toggles the led**

**TCNT0 = 0; // reset counter**

**tof\_detected = 0; // reset overflow detection counter**

**}**

**}**

**}**

**int main(void){**

**// connect led to pin PB2**

**DDRB |= (1 << 2);**

**DDRB |= (1 << 5);**

**PORTB ^= (1 << 5); // toggles the led off**

**// initialize timer**

**timer0\_init();**

**// loop forever**

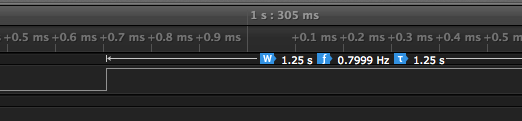
**while(1){**

**of\_funct(26, 165);**

**of\_funct(17, 195);**

**}**

**}**

****

1. **INITIAL/MODIFIED/DEVELOPED CODE OF TASK 2/B**

**#define F\_CPU 16000000UL**

**#include <avr/io.h>**

**#include <util/delay.h>**

**#include <avr/interrupt.h>**

**// global variable to count the number of overflows**

**volatile uint8\_t tof\_detected;**

**// TIMER0 overflow interrupt service routine**

**// called whenever TCNT0 overflows**

**ISR(TIMER0\_OVF\_vect){**

**// keep a track of number of overflows**

**tof\_detected++;**

**}**

**void timer0\_init(){**

**// set up timer with prescaler = 1024**

**TCCR0B |= (1 << CS02)|(1 << CS00);**

**// initialize counter**

**TCNT0 = 0;**

**TIMSK0 |= (1 << TOIE0);**

**sei();**

**tof\_detected = 0;**

**}**

**void of\_funct(unsigned int of\_detection\_num, unsigned int tcount){**

**while(tof\_detected != 0){**

**// check if no. of overflows = of\_detection\_num**

**if (tof\_detected >= of\_detection\_num){**

**// NOTE: '>=' is used**

**// check if the timer count reaches tcount**

**if (TCNT0 >= tcount){**

**PORTB ^= (1 << 2); // toggles the led**

**TCNT0 = 0; // reset counter**

**tof\_detected = 0; // reset overflow detection counter**

**}**

**}**

**}**

**}**

**int main(void){**

**// connect led to pin PB2**

**DDRB |= (1 << 2);**

**DDRB |= (1 << 5);**

**PORTB ^= (1 << 5); // toggles the led off**

**// initialize timer**

**timer0\_init();**

**// loop forever**

**while(1){**

**if(!(PINC & (1<<PINC1))){**

**PORTB ^= (1 << 2); // toggles the led off**

**of\_funct(78, 165);**

**}**

**else**

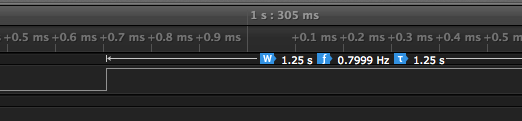
**PORTB |= (1<<2); // OTHERWISE STAY OFF**

**//of\_funct(26, 165);**

**//of\_funct(17, 195);**

**}**

**}**

****

1. **INITIAL/MODIFIED/DEVELOPED CODE OF TASK 3/A**

**#include <avr/io.h>**

**#include <util/delay.h>**

**#include <avr/interrupt.h>**

**volatile uint8\_t tof\_detected;**

**ISR( TIMER0\_COMPA\_vect ){**

**TCNT0 = 0;**

**int TACO = 0;**

**while(TACO != 27){**

**while(TCNT0 != 255);**

**TCNT0 = 0;**

**TACO++;**

**}**

**PORTB ^= (1<<2);**

**TACO = 0;**

**TCNT0 = 0;**

**while(TACO != 18){**

**while(TCNT0 != 255);**

**TCNT0 = 0;**

**TACO++;**

**}**

**PORTB ^= (1<<2);**

**TACO = 0;**

**TCNT0 = 0;**

**}**

**void timer0\_init(){**

**// set up timer with prescaler = 1024**

**TCCR0B |= (1 << CS02)|(1 << CS00);**

**// initialize counter**

**TCNT0 = 0;**

**TIMSK0 |= (1 << OCIE0A);**

**sei();**

**tof\_detected = 0;**

**}**

**int main(void){**

**DDRB |= (1 << 2);**

**DDRB |= (1 << 5);**

**PORTB ^= (1 << 5); // toggles the led off**

**timer0\_init();**

**while(1);**

**}**

1. **INITIAL/MODIFIED/DEVELOPED CODE OF TASK 3/B**

#define F\_CPU 16000000UL

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

volatile uint8\_t tof\_detected;

ISR( TIMER0\_COMPA\_vect ){

TCNT0 = 0;

int counter = 0;

if(!(PINC & (1<<PINC1) ) ){

while(counter < 77){

while((TIFR0 & (1<<OCF0A) ) == 0){

PORTB &= ~(1<<2);

}

TCNT0 = 0;

TIFR0 |= (1<<OCF0A);

counter++;

}

counter = 0;

}

else{

PORTB |= (1<<2);

}

}

void timer0\_init(){

// set up timer with prescaler = 1024

TCCR0B |= (1 << CS02)|(1 << CS00);

// initialize counter

TCNT0 = 0;

TIMSK0 |= (1 << OCIE0A);

sei();

tof\_detected = 0;

}

int main(void){

DDRB |= (1 << 2);

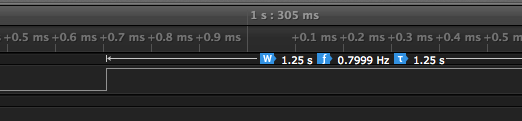
DDRB |= (1 << 5);

PORTB ^= (1 << 5); // toggles the led off

timer0\_init();

while(1);

}

****

1. **SCHEMATICS**

LED

Multi

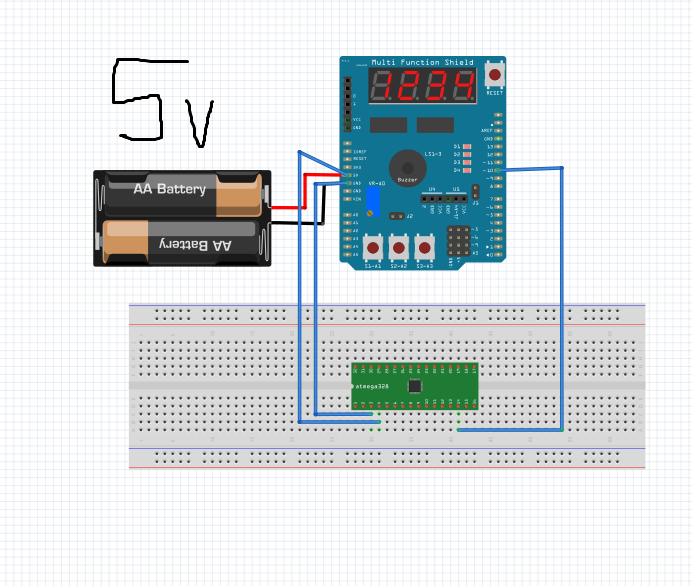
Function

shield

328p xplained

mini

POWER   
SUPPLY



1. **VIDEO LINKS OF EACH DEMO**

<https://www.youtube.com/watch?v=aP703kmuqv4>

<https://www.youtube.com/watch?v=VfGHUE9jQb8>

1. **GITHUB LINK OF THIS DA**

<https://github.com/mendos1/subnission_da/DA2C>

**Student Academic Misconduct Policy**

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

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

NAME OF THE STUDENT