CPE301 – SPRING 2019

Design Assignment A2A

Student Name: Alfonso Contreras

Student #: 1012383150

Student Email: contra2@unlv.nevada.edu

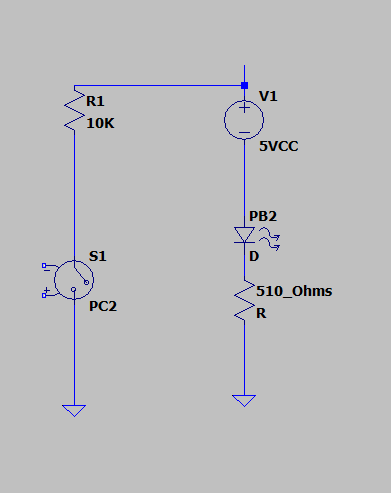
Primary Github address: https://github.com/alf8420/Submit.git

Directory: alf8420/Submit

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**



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

; This code was implemented by Alfonso Contreras

; for CPE 301 Dr.Venkatesan, and it creates a waveform of 60% DutyCycle

; of with a period of 0.725ms with 0.435ms in one state and 0.290 ms in the

;other state.

.org 0

LDI R16,04 ;representing PB2

OUT DDRB, R16 ;enable PB2 as output

LDI R17,0 ;used to set or reset PB2

LDI R20,5 ;set clock prescaler to 1024

STS TCCR1B,R20

begin:

LDI R20,0x00 ;resetting the counter to 0

STS TCNT1H,R20

STS TCNT1L,R20

RCALL delay2 ;calling timer to wait for 1 sec

EOR R17,R16

OUT PORTB, R17

LDI R20,0x00 ;resetting the counter to 0

STS TCNT1H,R20

STS TCNT1L,R20

RCALL delay1 ;XOR to toggle led

EOR R17,R16

OUT PORTB, R17

RJMP begin ;repeat main loop

delay1: ; I'm using as OFF state

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

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

CPI R28,0x8C ;comparing if lower 8 bits of timer is 0x08

BRSH body ;if lower bits of timer have reached desired amount, check the upper bits

RJMP delay1 ;otherwise, keep checking lower bits

body:

CPI R29,0x1A ;check to see if upper timer bits have reached the desired value

BRLT delay1 ;if not, recheck the lower bits

RET

delay2: ; I'm using as ON state

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 8 bits of timer is 0x08

BRSH body2 ;if lower bits of timer have reached desired amount, check the upper bits

RJMP delay2 ;otherwise, keep checking lower bits

body2:

CPI R29,0x11 ;check to see if upper timer bits have reached the desired value

BRLT delay2 ;if not, recheck the lower bits

RET ;once the timer reached the desired value, toggle the LED

1. **SCHEMATICS**

1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <avr/interrupt.h>

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

int main(void)

{

/\*Set up for PIN C as an input and PORTB

as an output \*/

DDRB |= (1<<2)

PORTB |= (1<<2);

DDRC &= (0<<2);

PORTC |= (1<<2);

/\* led is lit up for 1.25s once input pinc is pressed\*/

while (1)

{

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

{

PORTB &= ~(1<<2);

*\_delay\_ms*(1250)

}

else

PORTB |= (1<<2);

}

return 0;

//PROGRAM 2

/\*

\* BLINKING.cpp

\*

\* Created: 3/2/2019 3:11:06 PM

\* Author : ALFONSO cONTRERAS

\*/

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <avr/interrupt.h>

#include <util/delay.h>

int main(void)

{

// Pin B activated as OUTPUT FOR WAVESQUARE

DDRB |= (1<<2);

PORTD |= (1<<2); //ACTIVATED AS PULL-UP

EICRA = 0X2;

EIMSK = (1<<AD2C);

sei();

while (1)

{

PORTB ^= (1<<2);

*\_delay\_ms*(1250);

}

}

ISR (AD2C\_vect)//ISR FOR EXTERNAL INTERRUPT 0

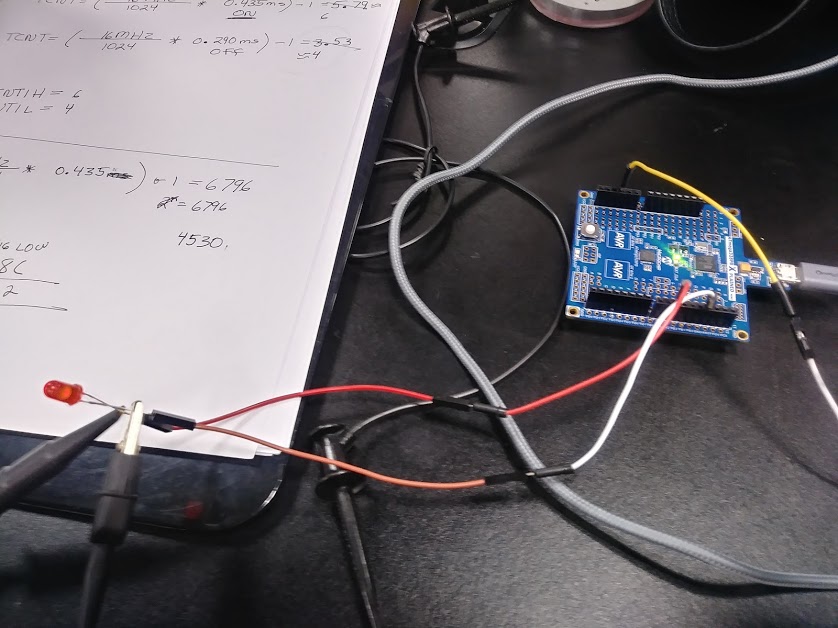
{

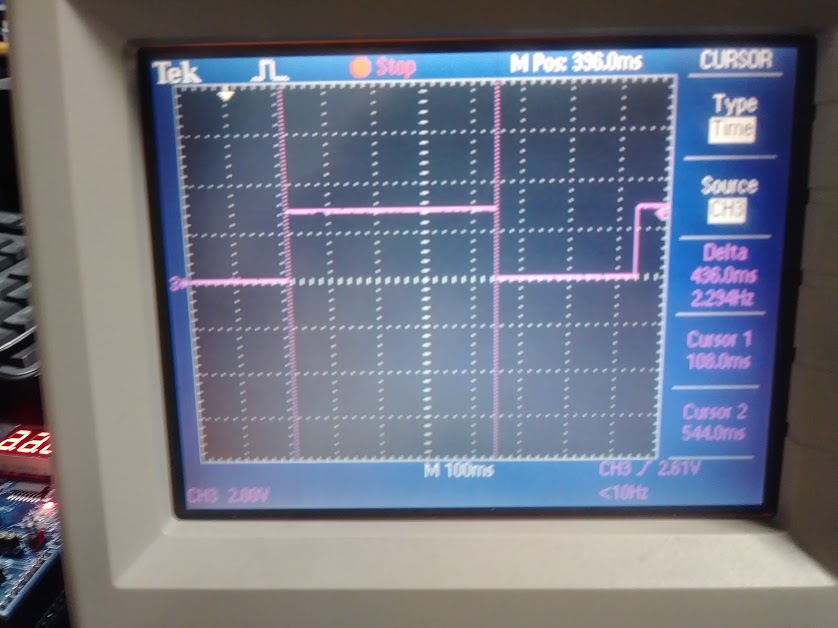
PORTB ^=(1<<5);

*\_delay\_ms*(1250);

}

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

****

****

1. **VIDEO LINKS OF EACH DEMO**

<https://youtu.be/q4RqfATAVLI>

1. **GITHUB LINK OF THIS DA**

https://github.com/alf8420/DA2.git

**Student Academic Misconduct Policy**

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

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

NAME OF THE STUDENT