CPE301 – SPRING 2019

Design Assignment 2B

Student Name: Tanner Tindall

Student #: 8000733733

Student Email: tindat1@unlv.nevada.edu

Primary Github address: <https://github.com/TannerTindall51>

Directory: <https://github.com/TannerTindall51/tindalltannerm_submission/tree/master/Design_Assignments/DA2B>

Design Assignment 2B:

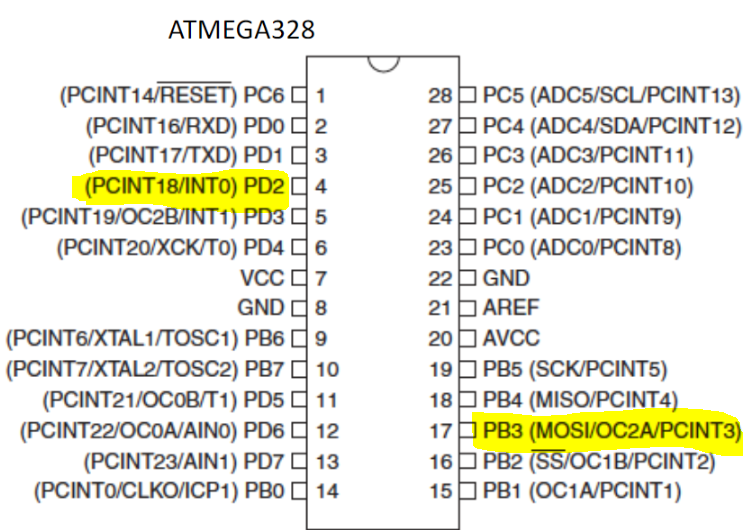
The goal of the assignment is use GPIO Interrupts:

1. Implement Design Assignment 2A.2 using INT0 (PD2 pin) interrupt mechanism.

1. **COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS**

* Atmel Studio 7.0 (Assembler, Simulator, & Debugger)
* Atmega328PB-Xmini
* PC Multi-Function Shield
* Logic Analyzer
* Switches
* LEDs

PB3 = LED

PD2 = Interrupt

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

|  |
| --- |
|  |

Assembly Variant:

;

;Design Assignment 2B ASM Code Variant

;Tanner Tindall

;

.org 0x00

jmp Interrupt\_Handler

.org 0x02

jmp Interrupt

Interrupt\_Handler: ;utilizing a stack to handle interrupts

ldi r25, HIGH(RAMEND) ;initilizing high end of the stack

out SPH, r25

ldi R20, LOW(RAMEND) ;initilizing low end of the stack

out SPL, r25

ldi r25, 0x02 ;loading value to determine trigger

sts EICRA, r25 ;interrupt control register set to trigger on 0x2 = falling edge

sbi DDRB, 3 ;sets DDRB3 as output

sbi PORTB, 3 ;sets PORTB3 to logic level HIGH

sbi PORTD, 2 ;sets INT0 flag

cbi DDRC, 3 ;sets DDRC3 as input

sbi PORTC,3 ;sets PORTC3 to logic level HIGH

ldi r25, 1<<INT0 ;loading a logic level HIGH value into INT0

out EIMSK, r25 ;loads the flag INT0 into EIMSK reggister with logic level HIGH

sei ;turns on interrupt

Duty\_Cycle: ;primary loop for blinking LED

cbi PORTB, 3 ;turns on the LED at PORTB.3

ldi r16, 28 ;register values for delay1

ldi r17, 101

ldi r18, 233

delay1: ;delay1 = 337.5ms

dec r18

brne delay1

dec r17

brne delay1

dec r16

brne delay1

nop

sbi PORTB, 3 ;turn off the LED at PORTB.3

ldi r19, 34 ;register values for delay2

ldi r20, 124

ldi r21, 86

delay2: ;delay2 = 412.5ms

dec r21

brne delay2

dec r20

brne delay2

dec r19

brne delay2

jmp Duty\_Cycle ;otherwise jump back to top and repeat

Interrupt: ;loop which executes interrupt instruction

in r26, PORTB ;loads value of PORTB into r26

ldi r27, (1<<2) ;loading value into r27 for XOR operation below

eor r26, r27 ;XOR operation with value of PORTB and r27

out PORTB, r26 ;load value in r26 into PORTB

ldi r22, 163 ;register values for delay3

ldi r23, 87

ldi r24, 3

delay3: ;delay3 = 2s

dec r24

brne delay3

dec r23

brne delay3

dec r22

brne delay3

nop

reti ;reti = returns from "Interrupt" loop

C Variant:

//

//Design Assignment 2B C Code Variant

//Tanner Tindall

//

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <avr/interrupt.h>

#include <util/delay.h>

int main(void)

{

DDRB = (1<<3); //setting PORTB.3 as an output

PORTB = (1<<3); //setting LED off (active low)

PORTD = (1<<2); //turn on pull-up port (PD2)

EIMSK = (1<<INT0); //turns on INT0 (external interrupt)

EICRA = 0x2; //INT0 triggers on 0x2 = falling edge

sei (); //turn on interrupt

while (1)

{

PORTB &= ~(1<<3); //turns LED on

*\_delay\_ms*(412.5); //delay for 55% of .75s period

PORTB |= (1<<3); //turns LED off

*\_delay\_ms*(337.5); //delay for 45% of .75s period

}

}

ISR (INT0\_vect) // ISR for external interrupt INT0

{

PORTB &= ~(1<<3); //set LED at PORTB.3 on

*\_delay\_ms*(2000); //delay for 2 sec

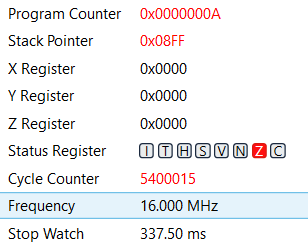
}

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

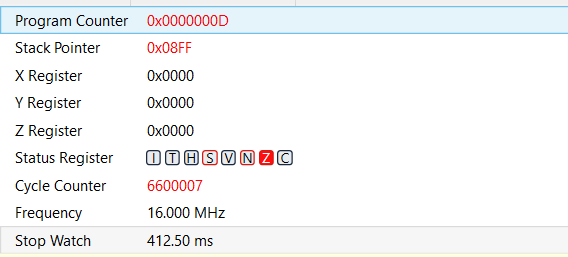
N/A

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

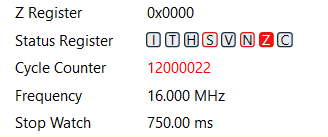
Delay Loop 1 = 337.5ms



Delay Loop 2 = 412.5ms



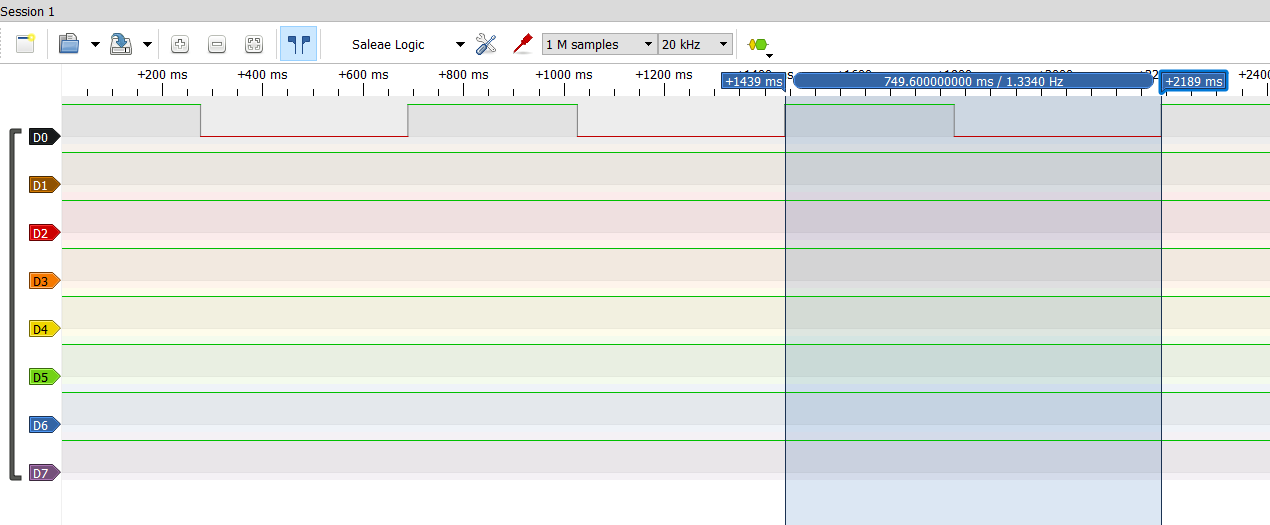
Delay Loop 3 = 2s



337.5ms + 412.5ms = 750ms

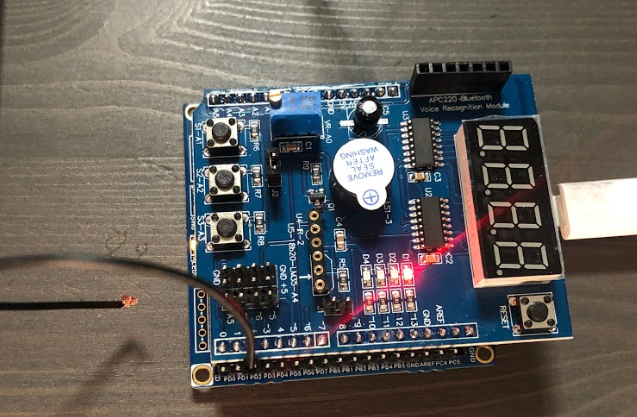
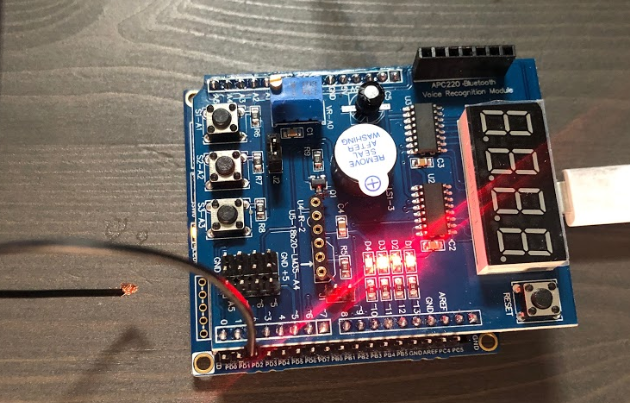
Therefore, these 3 delay loops are responsible for producing a duty cycle of 55% in the given time period.

The image below displays the output waveform of the logic analyzer displaying the length (in time) of one period (750ms).



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

The images below show the proper LED in both the “ON” and “OFF” phase of the duty cycle. The black wire is used as the interrupt when connected to ground.



1. **VIDEO LINKS OF EACH DEMO**

Demo Video Link: https: //youtu.be/M\_SBl5HkUq8

1. **GITHUB LINK OF THIS DA**

<https://github.com/TannerTindall51/tindalltannerm_submission/tree/master/Design_Assignments/DA2B>

**Student Academic Misconduct Policy**

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

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

Tanner Tindall