CPE301 – SPRING 2019

Design Assignment 2A

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/DA2A>

Design Assignment 2A: The goal of the assignment is use GPIO and delays:

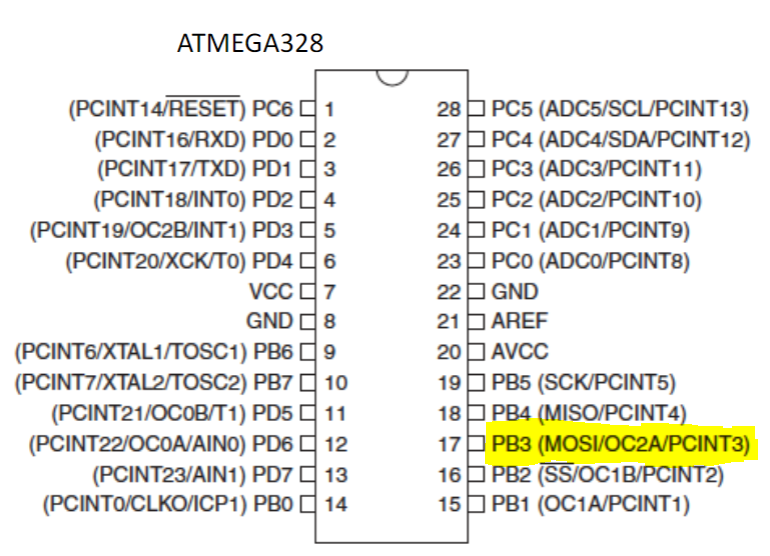
1. Design a delay subroutine to generate a waveform on PORTB.3 with 55% DC and 0.75 sec period.

2. Connect a switch to PORTC.3 (active high - turn on the pull up transistor) to poll for an event to turn on the led at PORTB.2 for 2 sec after the event.

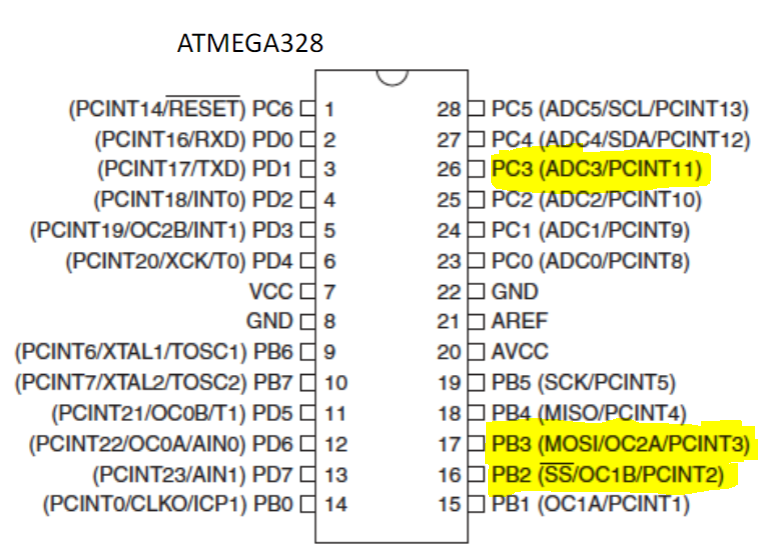
1. **COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS**

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

Ports used in 2A



Ports used in 2B



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

|  |
| --- |
|  |

1A Assembly:

;

;Design Assignment 2A ASM

;Tanner Tindall

;

ldi r19, 28 ;initializing register values

ldi r20, 101 ;for delay loops below

ldi r21, 233

ldi r16, 34

ldi r17, 124

ldi r18, 86

sbi DDRB, 3

Duty\_Cycle: ;primary loop

sbi PORTB, 3 ;sets portB.3 high

call delay1 ;delay for 337.5ms

cbi PORTB, 3 ;clears portB setting it low

call delay2 ;delay for 412.5ms

jmp Duty\_Cycle ;jump back to primary loop and restart

delay1: ;delay1 = 337.5ms

dec r21

brne delay1

dec r20

brne delay1

dec r19

brne delay1

nop

ret

delay2: ;delay2 = 412.5ms

dec r18

brne delay2

dec r17

brne delay2

dec r16

brne delay2

ret

1A C:

//

//Design Assignment 2A C

//Tanner Tindall

//

#include <avr/io.h>

#define F\_CPU 16000000UL

#include <util/delay.h>

int main(void)

{

DDRB = 0xFF; //sets all ports to output

PORTB = 0b00001000; //sets portB.3 high

while (1)

{

PORTB = 0b00001000; //sets portB.3 high, LED on

\_delay\_ms(337.5); //delay for 337.5ms

PORTB = 0x00; //sets portB low, LED off

\_delay\_ms(412.5); //delay for 412.5ms

}

return 1;

}

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

1B Assembly:

;Design Assignment 2B ASM

;Tanner Tindall

.org 0x00

ldi r16, 28 ;initializing register values

ldi r17, 101 ;for delay loops below

ldi r18, 233

ldi r19, 34

ldi r20, 124

ldi r21, 86

ldi r22, 163

ldi r23, 87

ldi r24, 3

ldi R25, 0xFF ;loading all 1's into r25

out DDRB, r25 ;portB is set to output sonce r25 = 0xFF

out PORTB, r25 ;intiliazes all LED's to turn off

ldi r25, 0x00 ;loading 0 into r25 for next step

out DDRC, r25 ;sets the switches to inputs

Duty\_Cycle: ;primary loop

cbi PORTB, 3 ;turn on port needed

delay1: ;delay1 = 337.5ms

dec r18

brne delay1

dec r17

brne delay1

dec r16

brne delay1

nop

sbi PORTB, 3 ;turns off port

delay2: ;delay2 = 412.5ms

dec r21

brne delay2

dec r20

brne delay2

dec r19

brne delay2

sbis PINC, 3 ;if button is pressed, then continue otherwise skip next instruction

rjmp Button\_Press ;go to button\_press loop

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

Button\_Press:

cbi PORTB, 2 ;turns on LED D4

delay3: ;delay3 = 2s

dec r24

brne delay3

dec r23

brne delay3

dec r22

brne delay3

nop

sbi PORTB, 2 ;turns off LED D4

rjmp Duty\_Cycle ;go back to primary loop and start over

2B C:

//Design Assignment 2B C

//Tanner Tindall

#include <avr/io.h>

#define F\_CPU 16000000UL

#include <util/delay.h>

int main(void)

{

DDRB = 0xFF; //sets all to output

PORTB = 0xFF; //sets all to output

DDRC = 0x00; //sets all to input

while (1)

{

PORTB |= (1<<3); //turns on led 3

\_delay\_ms(337.5); //delay for 337.5ms

PORTB &= ~(1<<3); //turns off led 3

\_delay\_ms(412.5); //delay for 412.5ms

if(!(PINC & (1<<3))) //enters loop if button is pressed

{

PORTB |= (1<<3);

PORTB &= ~(1<<2);

\_delay\_ms(2000);

PORTB |= (1<<2);

}

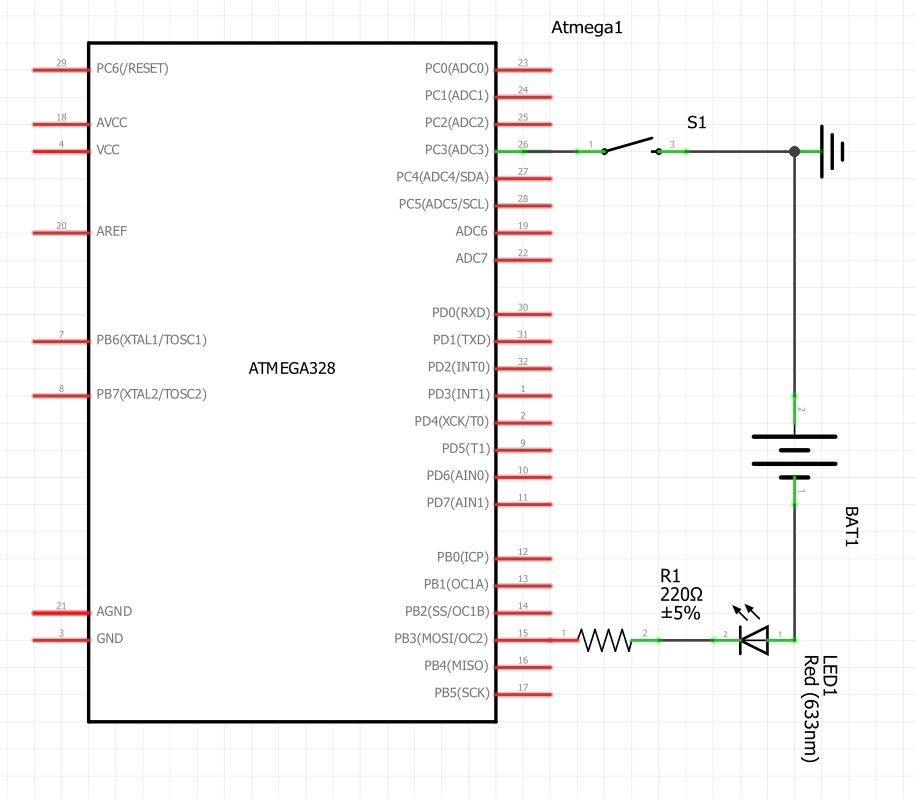
}

return 0;

}

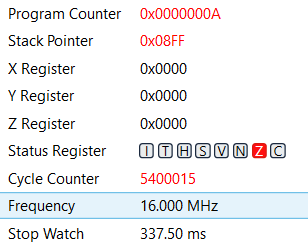
1. **SCHEMATICS**

Obtained using *Fritzing,* shows the schematic for Task 1A/B

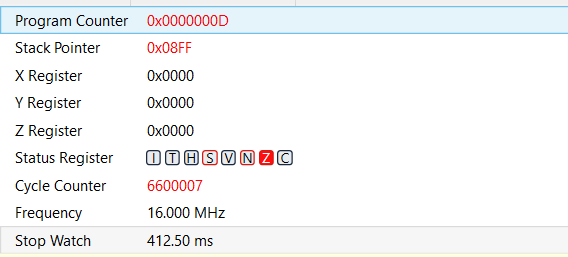


1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**

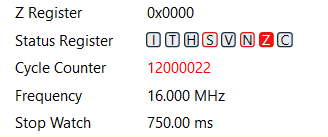
Delay Loop 1 = 337.5ms



Delay Loop 2 = 412.5ms



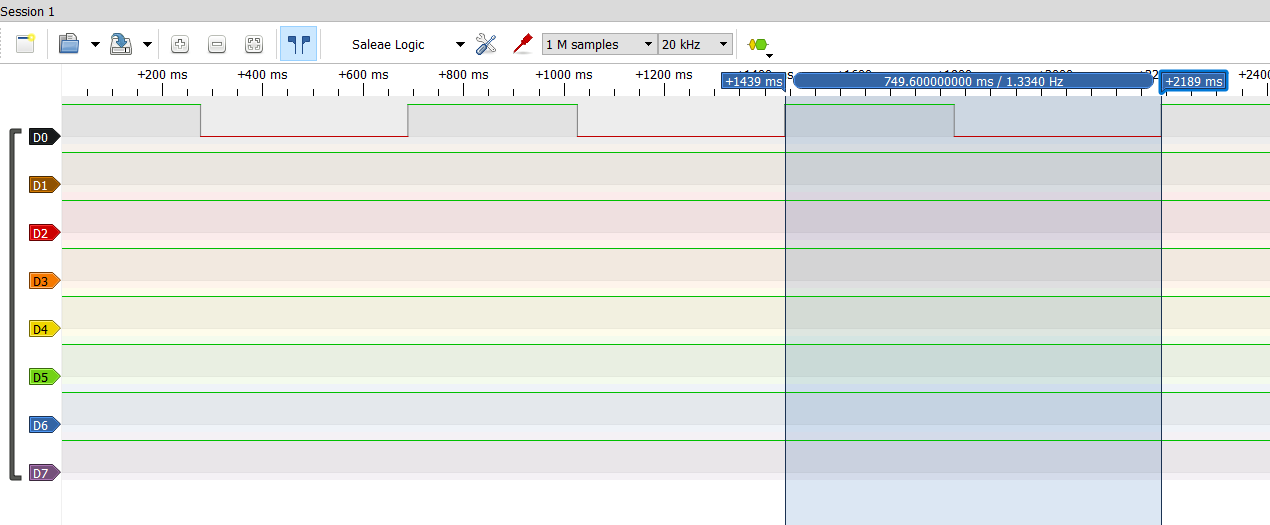
Delay Loop 3 = 2s



337.5ms + 412.5ms = 750ms

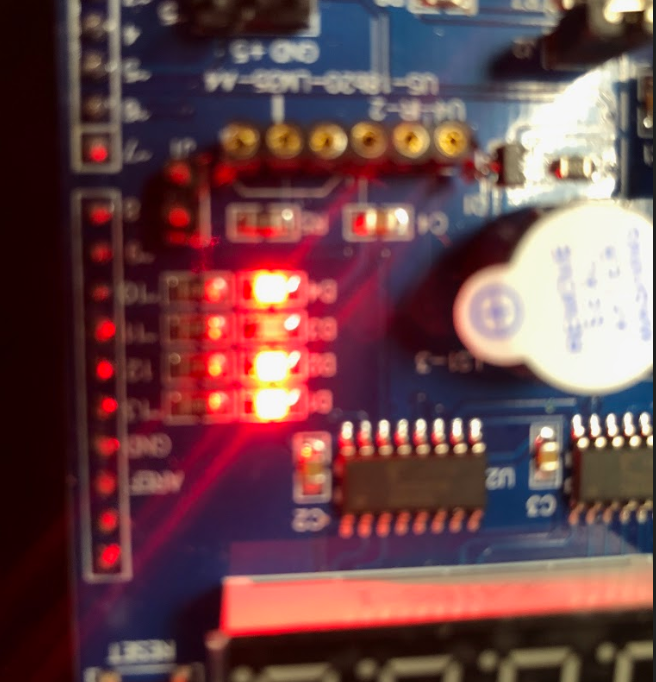
Therefore, these 3 delay loops are responsible for producing a duty cycle of 55% in the given time period.

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.





1. **VIDEO LINKS OF EACH DEMO**

Assembly Program Demo Video: <https://youtu.be/DzeufDcAsrQ>

C program Demo Video: <https://youtu.be/HES49L17CvM>

1. **GITHUB LINK OF THIS DA**

<https://github.com/TannerTindall51/tindalltannerm_submission/tree/master/Design_Assignments/DA2A>

**Student Academic Misconduct Policy**

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

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

Tanner Tindall