CPE301 – SPRING 2019

Design Assignment 2b

Student Name: Do Le

Student #: 2001183621

Student Email: led2@unlv.nevada.edu

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

Directory: <https://github.com/DoVietLe/assignments>

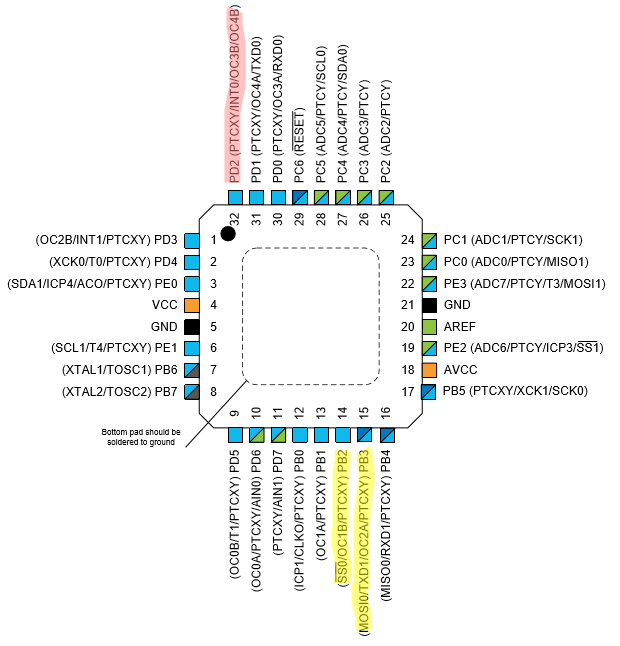
The goal of the assignment is use GPIO Interrupts:

1. Implement Design Assignment 2A.2 using INT0 (PD2 pin) interrupt mechanism.
2. **COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS**

• ATmega328PB Xplained Mini board

• Atmel Studios 7.0

* Assembler
* Debugger

• Arduino Multifunction Shield

PB2 & PB3 used to control LEDs.

PD2 used for button.

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

**Assembly Code**

.EQU A = 236 ; Calculated time for 37.5ms delay.

.EQU B = 254

.EQU A2 = 254 ; Calculated time for 2s delay.

.EQU B2 = 252

.EQU C2 = 50

.ORG 0x0000

JMP start

.ORG 0x0002 ; Sets a JMP instruction to the ISR at

JMP flash\_led ; INT0 location in the interrupt V.T.

; Replace with your application code

start:

LDI R16, HIGH(RAMEND); Initializes the stack.

OUT SPH, R16

LDI R16, LOW(RAMEND)

OUT SPL, R16

; Sets up interrupt.

LDI R16, 0x02

STS EICRA, R16 ; Sets INT0 to interrupt on falling edge.

LDI R16, 0x01

OUT EIMSK, R16 ; Enabled interrupt on INT0 pin.

SEI ; Enabled the global interrupt.

; Sets up ports.

LDI R17, (0x01<<2) ; Makes sure LED4 starts off.

LDI R16, 0x00

OUT DDRC, R16 ; Sets PORTC.3 as an input port.

OUT DDRD, R16

LDI R16, (0x01<<3)

OUT PORTC, R16 ; Enabled the pull-up resistor on PORTC.3.

LDI R16, (0x01<<2 | 0x01<<3)

OUT DDRB, R16 ; Sets PORTB.2 and PORTB.3 as an output pin.

OUT PORTB, R16 ; Turns LED off.

poll:

ORI R17, (0x01<<3) ; Sets the output waveform to high.

OUT PORTB, R17

LDI R16, 11

CALL delay\_37500us ; Waits ~0.4125s.

ANDI R17, ~(0x01<<3)

OUT PORTB, R17 ; Sets the output waveform to low.

LDI R16, 9

CALL delay\_37500us ; Waits ~0.3375s.

RJMP poll ; Keep polling.

end:

RJMP end

; Interrupt subroutine that flashes the LED on for 2 seconds, then turns it off.

flash\_led:

ANDI R17, ~(0x01<<2)

OUT PORTB, R17 ; Sets the output of PORTB.2 to low (turns LED on).

CALL delay\_2s ; Waits ~2s.

ORI R17, (0x01<<2) ; Sets the output of PORTB.2 to high (turns LED off).

OUT PORTB, R17

RETI ; Return from interrupt.

; ~2 delay

; Delay subroutine.

delay\_2s:

PUSH R16

PUSH R17 ; Stores the value of R17 and R18 AND R16 on the stack,

PUSH R18 ; so values aren't overwrited.

LDI R16, C2

delay\_loop0:

LDI R17, A2 ; Reloads the value of A.

delay\_loop1:

LDI R18, B2 ; Reloads the value of B.

delay\_loop2:

NOP ; Do nothing.

NOP

NOP

NOP

NOP

NOP

NOP

DEC R18 ; Decrement nested loop counter.

BRNE delay\_loop2; Loop again when counter has not reached zero.

DEC R17 ; Decrement nested loop counter.

BRNE delay\_loop1; Loop again when counter has not reached zero.

DEC R16 ; Decrements counter.

BRNE delay\_loop0; Loop again when counter has not reached zero.

POP R18 ; Pops the values of R18 and R17 AND R16 back from the stack.

POP R17

POP R16

RET ; Exit the subroutine.

; ~37.5ms delay

; Delay subroutine. The delay runs for about 600,000 cycles. The value of R16 determines

; how many times the delay is run (kind of).

delay\_37500us:

PUSH R17 ; Saves values on the stack.

PUSH R18

delay\_again:

LDI R17, A ; Outter loop.

loop0:

LDI R18, B ; Nested loops.

loop1:

NOP ; Do nothing.

NOP

NOP

NOP

NOP

NOP

NOP

DEC R18

BRNE loop1

DEC R17

BRNE loop0

DEC R16

BRNE delay\_again

POP R18 ; Returns values from the stack.

POP R17

RET ; Exits the subroutine.

**C Code**

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h> // Includes the interrupt file.

int main(void)

{

// Sets up the interrupt.

EICRA = 0x02; // Sets the interrupt to occur on falling edges.

EIMSK = 0x01; // Enables INT0 interrupt.

sei();

// Sets up the ports.

DDRC = 0x00; // Sets PORTC3 as input port.

PORTC = (0x01<<PORTC3); // Activates the pull up resistor on PORTC3.

DDRB = (0x01<<PORTB2 | 0x01<<PORTB3); // Sets PORTB2 as an output port.

PORTB = (0x01<<PORTB2); // Output 1 on PORTB2 (Turns off LED since

// LED is connected to Vcc).

while (1)

{

// Generates the waveform.

PORTB |= (0x01<<PORTB3); // Turns PORTB3 high for 55% duty cycle.

*\_delay\_ms*(412.5);

PORTB = ~(0x01<<PORTB3); // Turns PORTB3 low for 45% of the period.

*\_delay\_ms*(337.5);

}

}

ISR(INT0\_vect) {

PORTB &= ~(0x01<<PORTB2);

*\_delay\_ms*(2000);

PORTB |= (0x01<<PORTB2);

}

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

There is no task 2 for this assignment.

1. **SCHEMATICS**

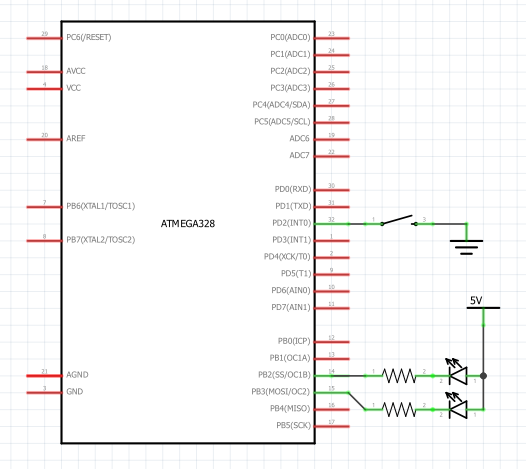


Figure : Schematic

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

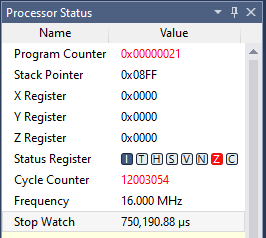


Figure : 0.75s period square wave

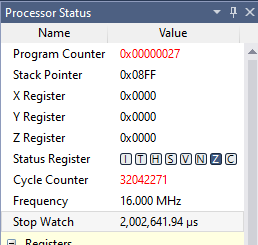


Figure : 2s that LED is on

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

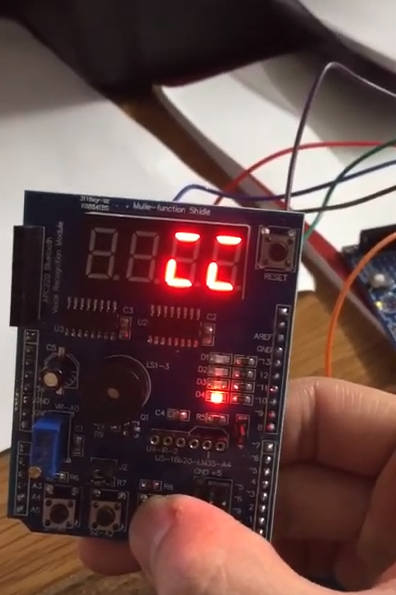


Figure : LED setup

1. **VIDEO LINKS OF EACH DEMO**

Assembly Implementation:

https://youtu.be/GpjJw-UB58o

C Implementation:

https://youtu.be/c0sM925\_ZoU

1. **GITHUB LINK OF THIS DA**

<https://github.com/DoVietLe/assignments/tree/master/ESD301/LAB02b>

**Student Academic Misconduct Policy**

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

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

Do Viet Le