CPE301 – SPRING 2019

Design Assignment 2B

Student Name: Mohamad Jundi

Student #: 8000321867

Student Email: jundi@unlv.nevada.edu

Primary Github address: https://github.com/MohamedJundi1994/Submission\_DA.git

Directory: Documents\School\CPE 301\Repository\CPE\_301\DesignAssignments\DA2B

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

For Assembly and C,

PORTB connection => Resistor => PB5 LED

When PD2 is Grounded => PORTD => Resistor => Ground => LED off for delay.

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

Assembly code:

.include <m328pdef.inc>

.ORG 0 // This is the location for the reset

JMP MAIN // Jump to Main loop

.ORG 0x02 // This is the location for the external interrupt 0

JMP EX0\_ISR // Jump to EX0\_ISR loop

MAIN:

LDI R16, HIGH(RAMEND) // Load high bits from last memory location in SRAM

OUT SPH, R16 // Store high bits from R20 to Stack Pointer High

LDI R20, LOW(RAMEND) // Load low bits from last memory location in SRAM

OUT SPL, R16 // Store high bits from R20 to Stack Pointer High

LDI R16, 0X2 // Load the value 0x2 to R20

STS EICRA, R16 // 0x2 set to the register so falling edge of INT0

SBI DDRB, 5 // Store bit 5 to light up PB5

LDI R17, 0XFF // Load value 0xFF to R17

OUT PORTB, R17 // Used to turn off all LEDs

SBI PORTD, 2 // Store bit 2 to use PD2 as pin

LDI R16, 1<<INT0 // Enable INT0 to R20

OUT EIMSK, R16 // Enable external interrupt zero

SEI // Used to enable interrupts

HERE: JMP HERE // Jump to HERE

EX0\_ISR: // Start of Loop EX0\_ISR

IN R21, PORTB // Input from portB

LDI R22, (1<<5) // PB5 is the output LED loaded into R22

EOR R21, R22 // XOR both the registers

OUT PORTB, R21 // Output value from R21 to PORTB

RCALL delay\_4\_1\_25s

RETI

// Begin code for all delays, used to achieve a delay of 1.25s

delay\_4\_1\_25s:

RCALL delay\_4\_100ms

RCALL delay\_4\_100ms

RCALL delay\_4\_100ms

RCALL delay\_4\_100ms

RCALL delay\_4\_100ms

RCALL delay\_4\_100ms

RCALL delay\_4\_100ms

RCALL delay\_4\_100ms

RCALL delay\_4\_100ms

RCALL delay\_4\_100ms

delay\_4\_100ms:

RCALL delay\_4\_10ms

RCALL delay\_4\_10ms

RCALL delay\_4\_10ms

RCALL delay\_4\_10ms

RCALL delay\_4\_10ms

RCALL delay\_4\_10ms

RCALL delay\_4\_10ms

RCALL delay\_4\_10ms

RCALL delay\_4\_10ms

RCALL delay\_4\_10ms

delay\_4\_10ms:

RCALL delay\_s

RCALL delay\_s

RCALL delay\_s

RCALL delay\_s

delay\_s:

RCALL delay\_4\_2\_5ms

RCALL delay\_4\_2\_5ms

RCALL delay\_4\_2\_5ms

RCALL delay\_4\_2\_5ms

RCALL delay\_4\_2\_5ms

RCALL delay\_4\_2\_5ms

RCALL delay\_4\_2\_5ms

RCALL delay\_4\_2\_5ms

RCALL delay\_4\_2\_5ms

RCALL delay\_4\_2\_5ms

RCALL delay\_4\_2\_5ms

RCALL delay\_4\_2\_5ms

RCALL delay\_4\_2\_5ms

RCALL delay\_4\_2\_5ms

RCALL delay\_4\_2\_5ms

RCALL delay\_4\_2\_5ms

delay\_4\_2\_5ms:

PUSH R19 // Push R19 on stack

LDI R19, 243 // load value of 243, number of times to loop

delay\_spread1:

NOP //

NOP //

NOP // NO OPERATION, used to waste time

NOP //

NOP //

DEC R19 // Decrement R19 value

BRNE delay\_spread1 // Branch back to begining of loop

POP R19 // Pop R19 from stack

RET

C code:

#define *F\_CPU* 16000000UL // CPU Frequency set to 16MHz

#include <avr/io.h> // Header file for AVR

#include <avr/interrupt.h> // Header file allowing use of interrupts

#include <util/delay.h> // Header file allowing use of delays

int main(void)

{

DDRB = (1<<5); // Set PB4 as output

PORTD = 1<<2; // Set Pin PD2 as input

EICRA = 0x2; // 0x2 set to the register so falling edge of INT0 generates an interrupt request

EIMSK = (1<<INT0); // Enable external interrupt zero

sei (); // Used to enable interrupts

while (1) // While loop for infinite loop

{

// No content in loop, used as a standby until interrupt

}

}

ISR (INT0\_vect) // Interrupt service routine for external interrupt

{

PORTB ^= (1<<5); // XOR output on PB3

*\_delay\_ms*(1250); // Given delay time to use in ms

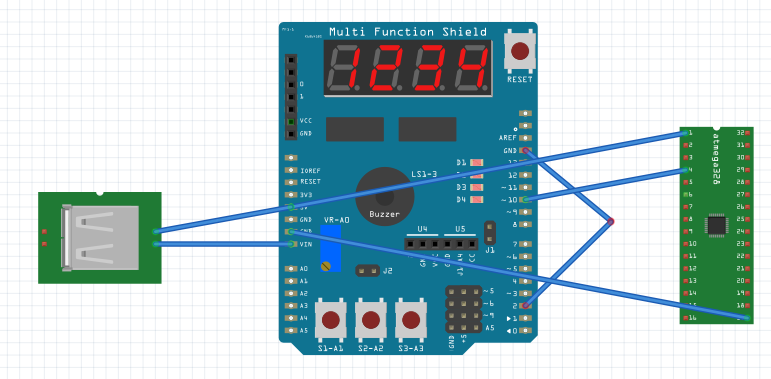
}

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

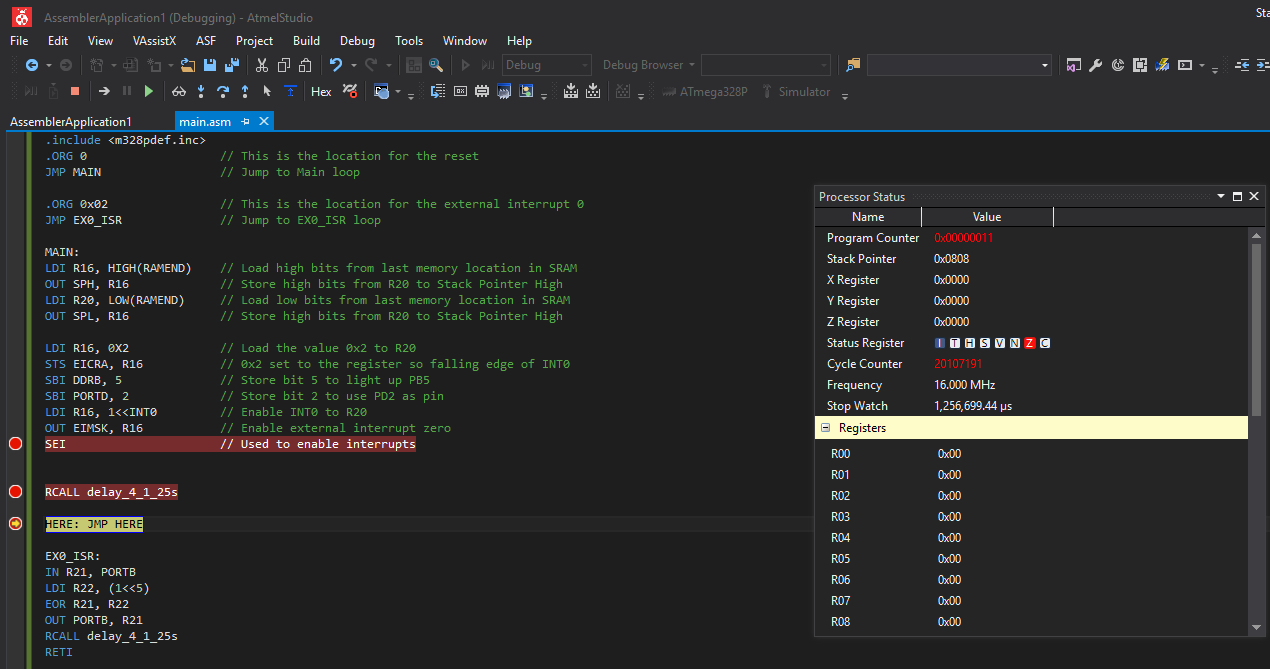
All code is in number 2.

1. **SCHEMATICS**

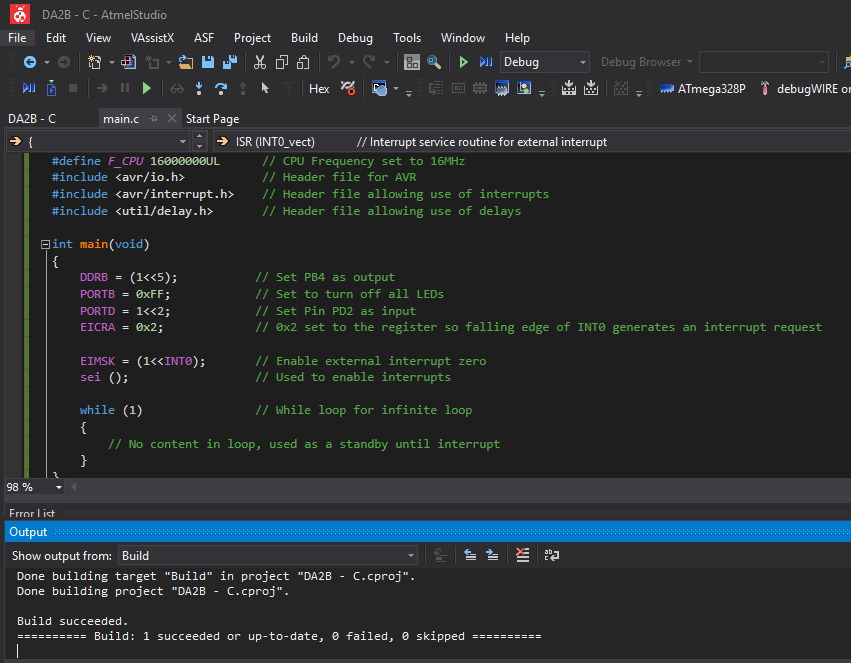
Schematic built using Fritzing,

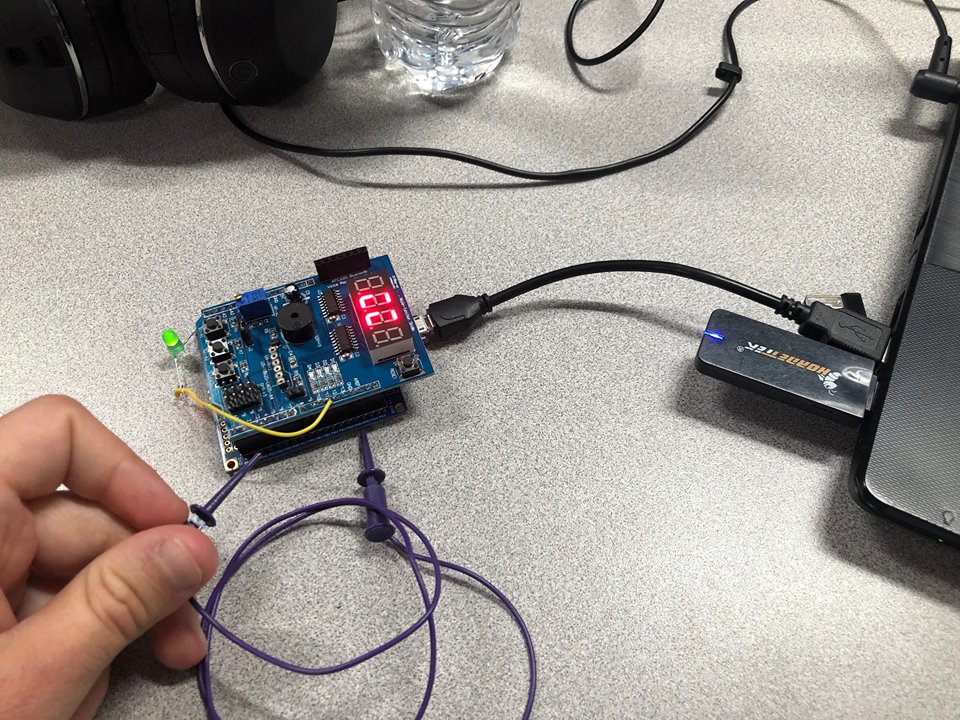


1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**

Assembly output screenshot:

C on next page =>

C output screenshot:

1. **SCREENSHOT OF EACH DEMO (BOARD SETUP)**
2. **VIDEO LINKS OF EACH DEMO**

Assembly video:

<https://www.youtube.com/watch?v=X8YcX13Q-9o&feature=youtu.be&fbclid=IwAR31-FcTJFBSMSoBt4DPEjMShf4XGq_YYB1C4pVUBTZ9FAZHAUoi0PvQKjk>

C video:

<https://www.youtube.com/watch?v=XgS5OEAl_8s&feature=youtu.be&fbclid=IwAR1dCE0k2U2VMFa1EvJ45GkCWJ4JkbBAVs-jHhaj5h9x4tMWglbwQGZuB90>

1. **GITHUB LINK OF THIS DA**

Link: https://github.com/MohamedJundi1994/Submission\_DA.git

This assignment submission is my own, original work.

MOHAMAD JUNDI