Student no: 2020-01144-MN-0 **Date Started: Feb. 28, 2023**

Name: Jewel Anne Reyes **Date Submitted: Mar. 2, 2023**

Machine Problem # 4

Sleep Instruction

Objective:

- To be familiar with the operation of Seven Segment Display
- To be familiarize with controlling a Common Anode or Common Cathode Seven Segment Display through Digital Control.
- To create a Binary Counter using Seven Segment Display.
- To be familiar with Input/Output Interfacing technique of PIC16F84A
- To configure the PORTS of MCU for I/O Interfacing.
- To enhance skills of Assembly Programming using MPLAB and Proteus.
- To improve analyzing and setting of sleep instructions of PIC16F84A

Background:

SLEEP mode offers a very low current power-down mode. The user can wake-up from SLEEP through external RESET, Watchdog Timer Time-out or through an interrupt. Several oscillator options are provided to allow the part to fit the application. The RC oscillator option saves system cost while the LP crystal option saves power. A set of configuration bits are used to select the various options.

A device may be powered down (SLEEP) and later powered up (wake-up from SLEEP). The Power-down mode is entered by executing the SLEEP instruction. If enabled, the Watchdog Timer is cleared (but keeps running), the PD bit (STATUS<3>) is cleared, the TO bit (STATUS<4>) is set, and the oscillator driver is turned off. The I/O ports maintain the status they had before the SLEEP instruction was executed (driving high, low, or hi-impedance). For the lowest current consumption in SLEEP mode, place all I/O pins at either VDD or VSS, with no external circuitry drawing current from the I/O pins, and disable external clocks. I/O pins that are hi-impedance inputs should be pulled high or low externally to avoid switching currents caused by floating inputs. The T0CKI input should also be at VDD or VSS. The contribution from on-chip pull-ups on PORTB should be considered. The MCLR pin must be at a logic high level (VIHMC). It should be noted that a RESET generated by a WDT time-out does not drive the MCLR pin low.

WAKE-UP FROM SLEEP

The device can wake-up from SLEEP through one of the following events:
1. External RESET input on MCLR pin.
2. WDT wake-up (if WDT was enabled).
3. Interrupt from RB0/INT pin, RB port change, or data EEPROM write complete.

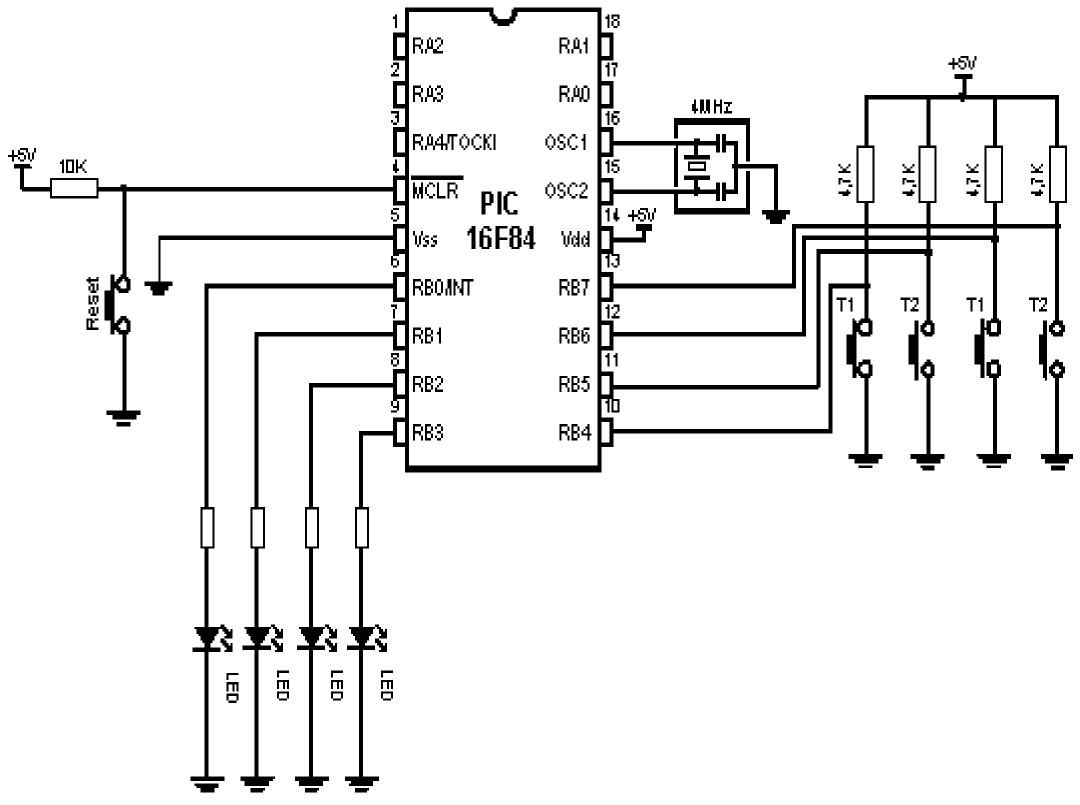
Materials:

PIC16F84A	PC
PIC Programmer	Power Supply
Test Jig	

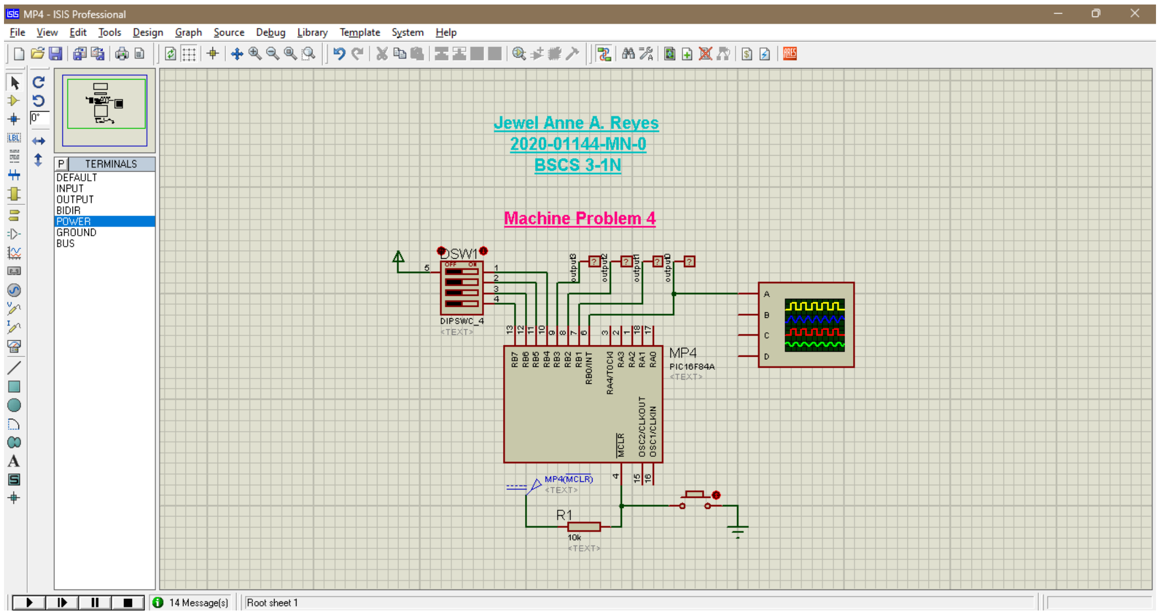
Procedure:

Perform the following Setup and provide a code for the said setup.

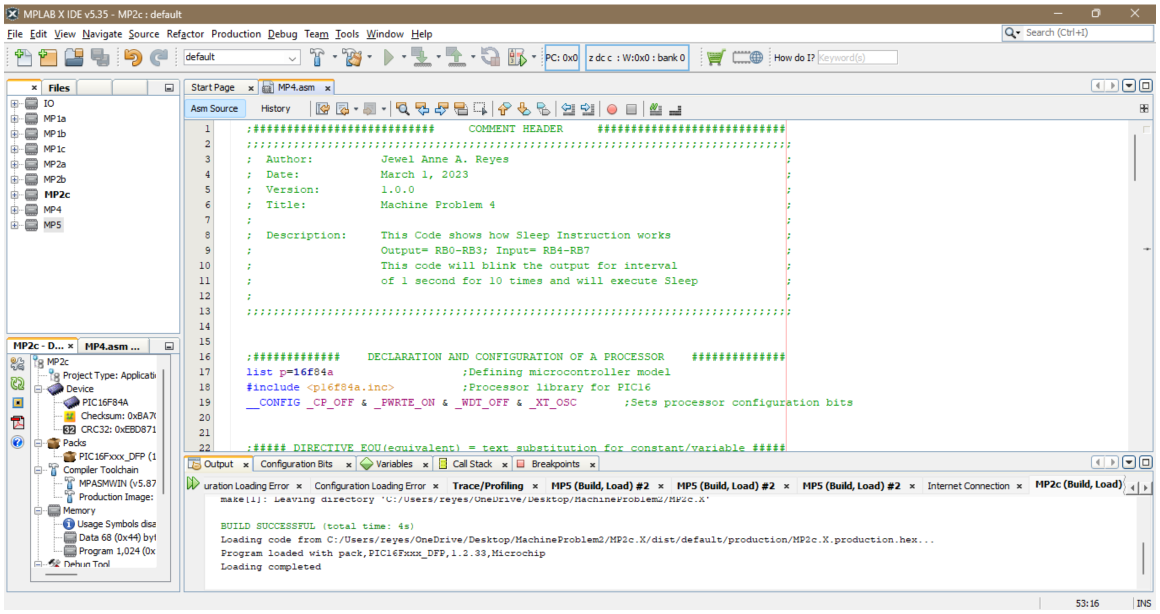
Set RB0 to RB3 as output, and RB4 to RB7 as input. Create a Code that will Blink RB0 to RB3 for an interval of 1 sec. for 10 times, and will execute the Sleep instruction. Perform the Wake-up from Sleep using the interrupt on changes in RB Port.



Schematic:



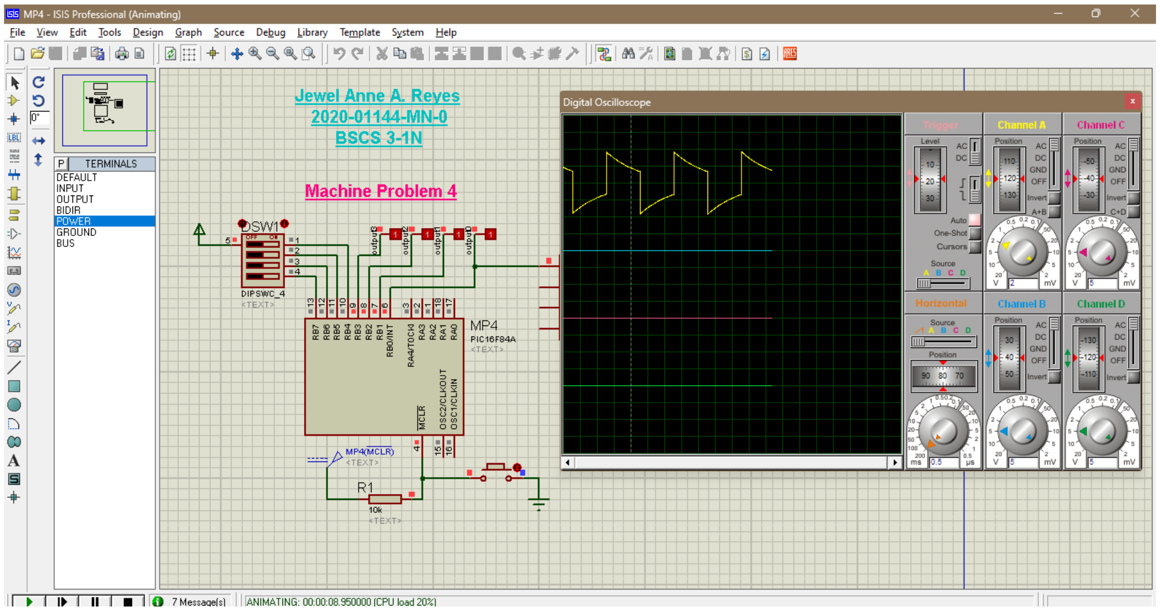
Program Build Successful:



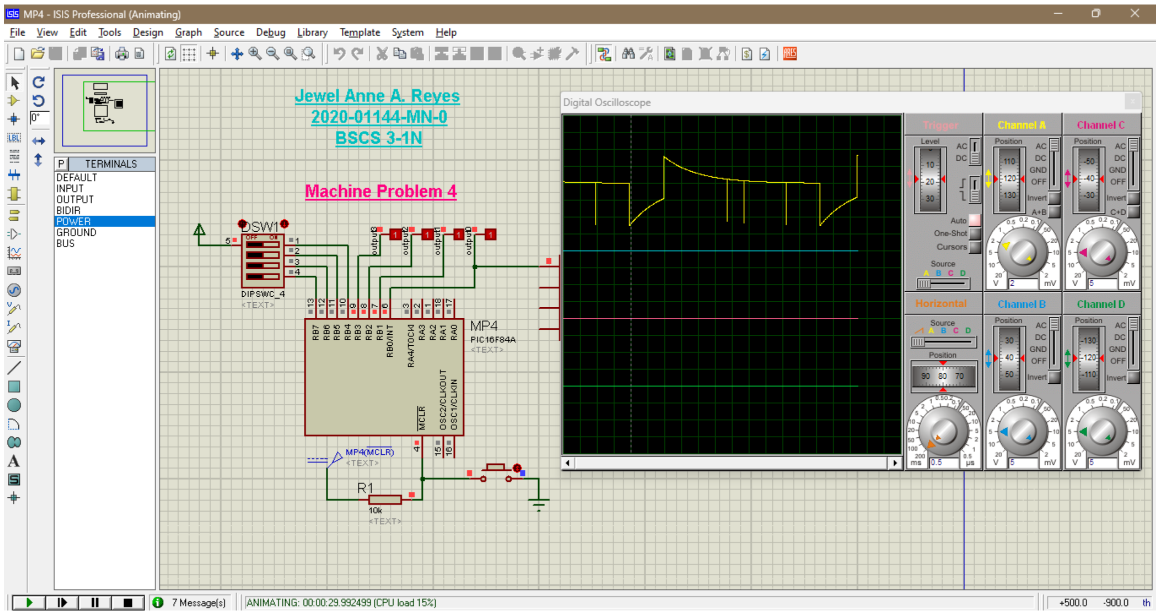
```
;##########################     COMMENT HEADER     ############################
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;  Author:          Jewel Anne A. Reyes                                        ;
;  Date:            March 1, 2023                                              ;
;  Version:         1.0.0                                                      ;
;  Title:           Machine Problem 4                                          ;
;                                                                              ;
;  Description:     This Code shows how Sleep Instruction works                ;
;                   Output= RB0-RB3; Input= RB4-RB7                            ;
;                   This code will blink the output for interval               ;
;                   of 1 second for 10 times and will execute Sleep            ;
;                                                                              ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


;#############    DECLARATION AND CONFIGURATION OF A PROCESSOR    ##############
list p=16f84a                   ;Defining microcontroller model
#include <p16f84a.inc>          ;Processor library for PIC16
__CONFIG _CP_OFF & _PWRTE_ON & _WDT_OFF & _XT_OSC       ;Sets processor configuration bits


;##### DIRECTIVE EQU(equivalent) = text substitution for constant/variable #####
```

```
make[1]: Leaving directory 'C:/Users/reyes/OneDrive/Desktop/MachineProblem2/MP2c.X'

BUILD SUCCESSFUL (total time: 4s)
Loading code from C:/Users/reyes/OneDrive/Desktop/MachineProblem2/MP2c.X/dist/default/production/MP2c.X.production.hex...
Program loaded with pack,PIC16Fxxx_DFP,1.2.33,Microchip
Loading completed
```

Outputs:



ANIMATING: 00:00:08.950000 (CPU load 20%)



ANIMATING: 00:00:29.992499 (CPU load 15%)

Observation:

I have observed that every time the light or LED blinks, there are a certain illustration of the digital pulse. It rises then falls into a slope then set to low. This automatically give a loop in the pulse unless it becomes inactive thus activating the sleep. In the setup, after blinking for 10 times, the PIC will activate the sleep and will continue execution if there are any inputs from the user.

I noticed that when the watchdog timer is off, you need to manually wake the PIC in order for it to continue. While when the watchdog timer is on, it can automatically wake the PIC in a certain set of time.

The SLEEP instruction is an important feature of the PIC16F84A microcontroller. It allows the microcontroller to enter a low power consumption mode, where the CPU and peripherals are turned off, and the oscillator is stopped. This helps in saving power and extends battery life in battery-powered applications.

The SLEEP mode can be used in applications where the microcontroller needs to wait for an external event or a specific amount of time before resuming normal operation. It is commonly used in applications such as battery-powered sensors, data loggers, and remote controllers.

In addition to saving power, the SLEEP instruction can also help in reducing EMI (electromagnetic interference) emissions from the microcontroller. When the oscillator is stopped, the clock signals that are a source of EMI are also stopped, reducing the EMI noise generated by the microcontroller.

Conclusion/ Recommendation:

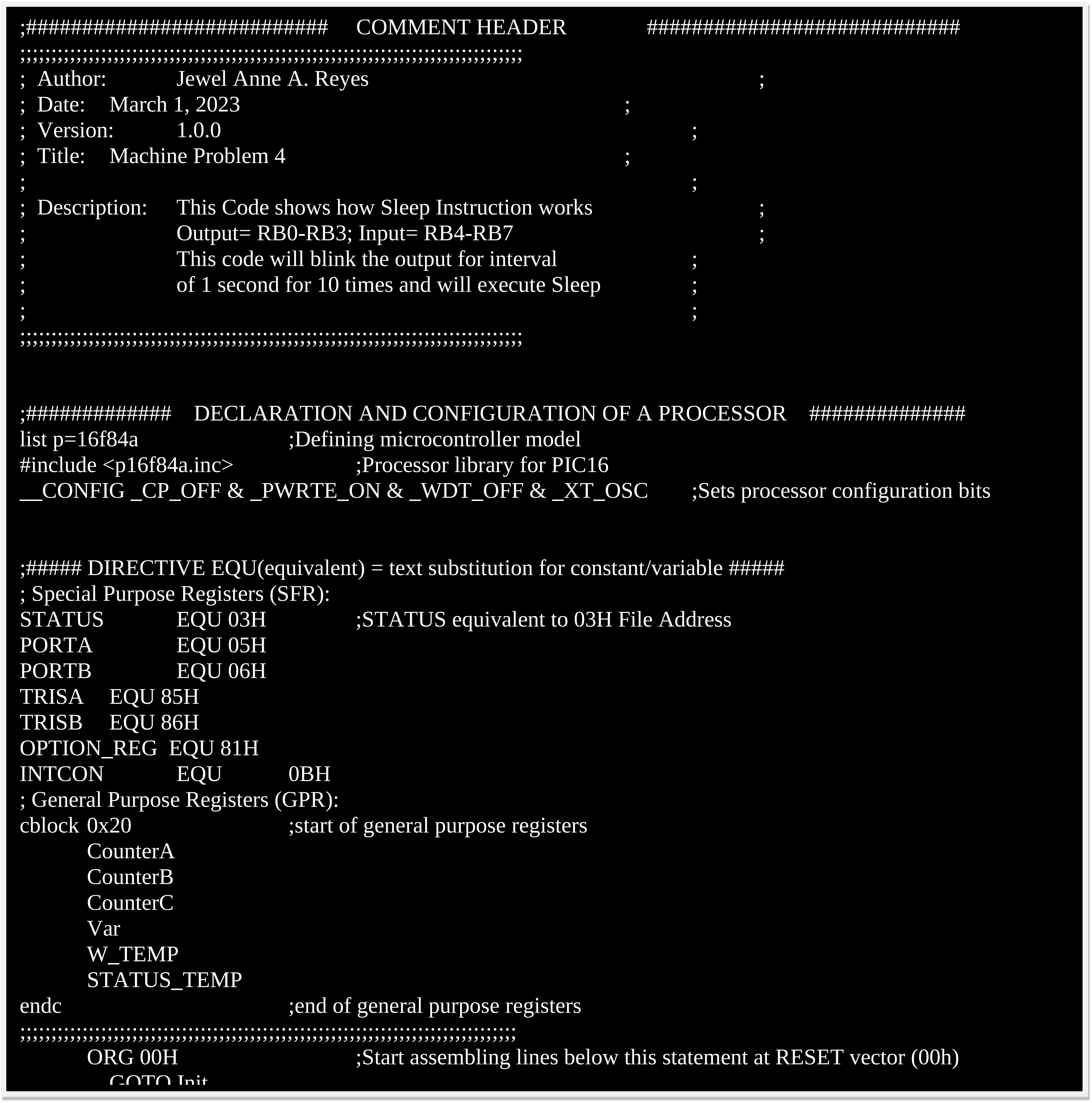
In PIC16F84A, the Sleep mode and the Watchdog Timer (WDT) are related in the sense that the WDT is used to wake up the microcontroller from Sleep mode after a specified period of time has elapsed. With this, it can be easier when implementing the sleep function as it will automatically wake the PIC.

It is recommended to try it using different setups like blinking lights or maybe a pomodoro timer that are used for focusing. It would be cool to try it also with 7 segment display.

Power saving sleep mode is added in the chip that generates a low current power down mode. The sleep mode can be removed using an interrupt, watchdog timer or external reset. In this setup, I used an interrupt and tried to integrate the watchdog timer. It is recommended to try different ways on removing the sleep mode with different setups.

Source Code:

(Please Attach the asm file)

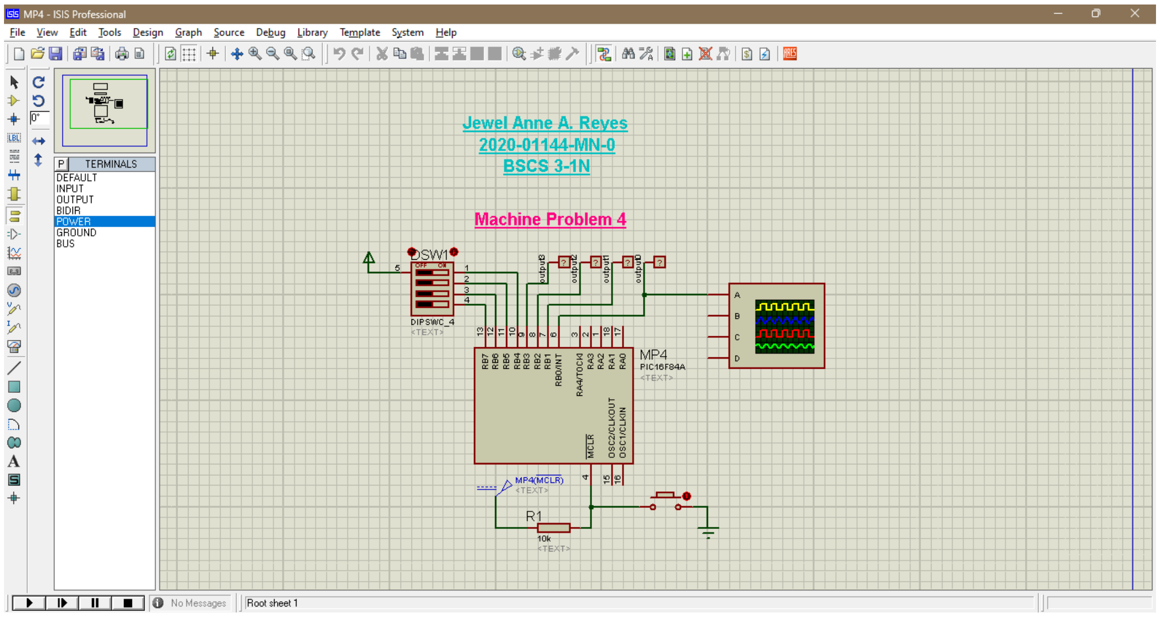


```
;##########################     COMMENT HEADER            ############################
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;  Author:          Jewel Anne A. Reyes                                               ;
;  Date:    March 1, 2023                                           ;
;  Version:         1.0.0                                                   ;
;  Title:    Machine Problem 4                                      ;
;                                                                           ;
;  Description:     This Code shows how Sleep Instruction works                      ;
;                   Output= RB0-RB3; Input= RB4-RB7                                   ;
;                   This code will blink the output for interval                ;
;                   of 1 second for 10 times and will execute Sleep             ;
;                                                                           ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


;#############    DECLARATION AND CONFIGURATION OF A PROCESSOR    ##############
list p=16f84a                   ;Defining microcontroller model
#include <p16f84a.inc>                  ;Processor library for PIC16
__CONFIG _CP_OFF & _PWRTE_ON & _WDT_OFF & _XT_OSC       ;Sets processor configuration bits


;##### DIRECTIVE EQU(equivalent) = text substitution for constant/variable #####
; Special Purpose Registers (SFR):
STATUS          EQU 03H         ;STATUS equivalent to 03H File Address
PORTA           EQU 05H
PORTB           EQU 06H
TRISA    EQU 85H
TRISB    EQU 86H
OPTION_REG  EQU 81H
INTCON          EQU         0BH
; General Purpose Registers (GPR):
cblock 0x20                     ;start of general purpose registers
        CounterA
        CounterB
        CounterC
        Var
        W_TEMP
        STATUS_TEMP
endc                            ;end of general purpose registers
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
        ORG 00H                         ;Start assembling lines below this statement at RESET vector (00h)
           GOTO Init
```

```
            GOTO Init
        ORG 04H                            ;Start assembling lines below this statement at Peripheral Interrupt Vector (04H)

            PUSH
                    MOVWF W_TEMP
                    SWAPF STATUS, 0
                    MOVWF STATUS_TEMP

            ISR
                    BCF INTCON, 0

            POP
                    SWAPF STATUS_TEMP, 0
                    MOVWF STATUS
                    SWAPF W_TEMP, 1
                    SWAPF W_TEMP, 0

        RETFIE                             ;Return from interrupt
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
Init
        MOVLW 0x0A
        MOVWF Var
        BSF STATUS,5                       ;Bit Set F (goes to Bank 1)
        MOVLW 0xF0
        MOVWF TRISB                        ;Transfer W Register literals to F Register 86H
        BSF OPTION_REG, 7                  ;Enables internal pull-up resistors on PORTB
        BCF STATUS, 5                      ;Bit Clear F (goes to Bank 0)
        BSF INTCON, 7                      ;GIE: Enable all un-masked interrupts
        BSF INTCON, 3                      ;RBIE: Enables RB Port change interrupt

   ;Clearing of bits in PORTB register
        BCF PORTB, 0
      BCF PORTB, 1
      BCF PORTB, 2
      BCF PORTB, 3

        goto Start
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
Start
   ;Setting the bits into high
        BSF PORTB, 0
      BSF PORTB, 1
      BSF PORTB, 2
      BSF PORTB, 3

      CALL Delay                   ;1 second delay
   ;Clearing of bits in PORTB register
        BCF PORTB, 0
      BCF PORTB, 1
      BCF PORTB, 2
      BCF PORTB, 3

        CALL Delay              ;1 second delay

      decfsz Var, 1             ;Decrement F, Skip if 0
      goto Start
      goto loopx
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
loopx
      NOP                       ;No operation
      Sleep
        BCF INTCON, 7                   ;GIE: Disables all interrupts
        MOVLW 0x0A
        MOVWF Var
        goto Start
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
Delay
   MOVLW D'6'                           ;1 us
   MOVWF CounterC               ;1 us
   MOVLW D'24'                          ;1 us
   MOVWF CounterB               ;1 us
   MOVLW D'167'                 ;1 us
   MOVWF CounterA                       ;1 us
loop
   DECFSZ CounterA,1                    ;167 x 1 cycle + 1 cycle = 168 us
   GOTO loop                    ;166 * 2 us = 332 us
   DECFSZ CounterB,1                    ;24 x 1 cycle + 1 cycle = 25 us
   GOTO loop                    ;24 * 2 us = 48 us
   DECFSZ CounterC,1                    ;6 x 1 cycle + 1 cycle = 7 us
   GOTO loop                    ;5 * 2 us = 10 us
   NOP                                  ;1 us
                                ;TOTAL = 597 us
                                ;3A + 770B + 197,122C - 197,879
                                ;3(167) + 770(24) + 197,122(6) - 197,879
                                ;501 + 18,480 + 1,182,732 - 197,879
                                ;1,201,713 - 197,879
                                ;DELAY = 1,003,834 us == 1.003 seconds
   RETURN                       ;2 us
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
end
```

Additionals:

(Please attach any changes in the setup including images and schematics)


MP4 - ISIS Professional
File View Edit Tools Design Graph Source Debug Library Template System Help
TERMINALS
DEFAULT
INPUT
OUTPUT
BIDIR
POWER
GROUND
BUS
Jewel Anne A. Reyes
2020-01144-MN-0
BSCS 3-1N
Machine Problem 4
DSW1
DIPSWC_4
output3
output2
output1
output0
MP4
PIC16F84A
R1
10k
MP4(MCLR)
No Messages
Root sheet 1