Student no: _2020-01144-MN-0 **Date Started: Feb. 28, 2023**

Name: _Jewel Anne Reyes____ **Date Submitted: Mar. 2, 2023**

Machine Problem # 5

Watchdog Timer (WDT)

Objective:

- To be familiar with the operation of Watchdog Timer
- To use the WDT to wake up MCU from Sleep.
- To enhance skills in using bit-oriented file register operations.
- To practice setting the Watchdog Timer Prescaler/Postscaler with bit and rate.
- To Perform Sequential Display using MCU processing.
- To utilize branching and Looping technique in Assembly Language
- To enhance skills of Assembly Programming using MPLAB and Proteus.
- To be familiar with Timing analysis in Assembly language and to use the 1-second routine.

Background:

The Watchdog Timer is a free running On-Chip RC Oscillator which does not require any external components. This RC oscillator is separate from the RC oscillator of the OSC1/CLKIN pin. That means that the WDT will run even if the clock on the OSC1/CLKIN and OSC2/CLKOUT pins of the device has been stopped, for example, by execution of a SLEEP instruction. During normal operation, a WDT time-out generates a device RESET. If the device is in SLEEP mode, a WDT wake-up causes the device to wake-up and continue with normal operation. The WDT can be permanently disabled by programming configuration bit WDTE as a '0'

The WDT has a nominal time-out period of 18 ms, (with no prescaler). The time-out periods vary with temperature, VDD and process variations from part to part (see DC specs). If longer time-out periods are desired, a prescaler with a division ratio of up to 1:128 can be assigned to the WDT under software control by writing to the OPTION_REG register. Thus, time-out periods up to 2.3 seconds can be realized.
The CLRWDT and SLEEP instructions clear the WDT and the postscaler (if assigned to the WDT) and prevent it from timing out and generating a device RESETcondition. The TO bit in the STATUS register will be cleared upon a WDT time-out.

Materials:

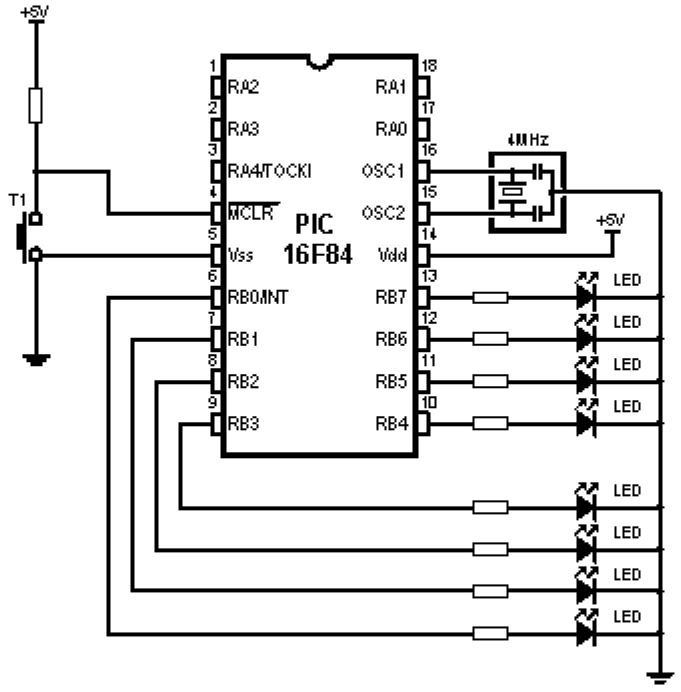
PIC16F84A	PC
PIC Programmer	Power Supply
Test Jig	

PIC16F84A Laboratory Machine Problems

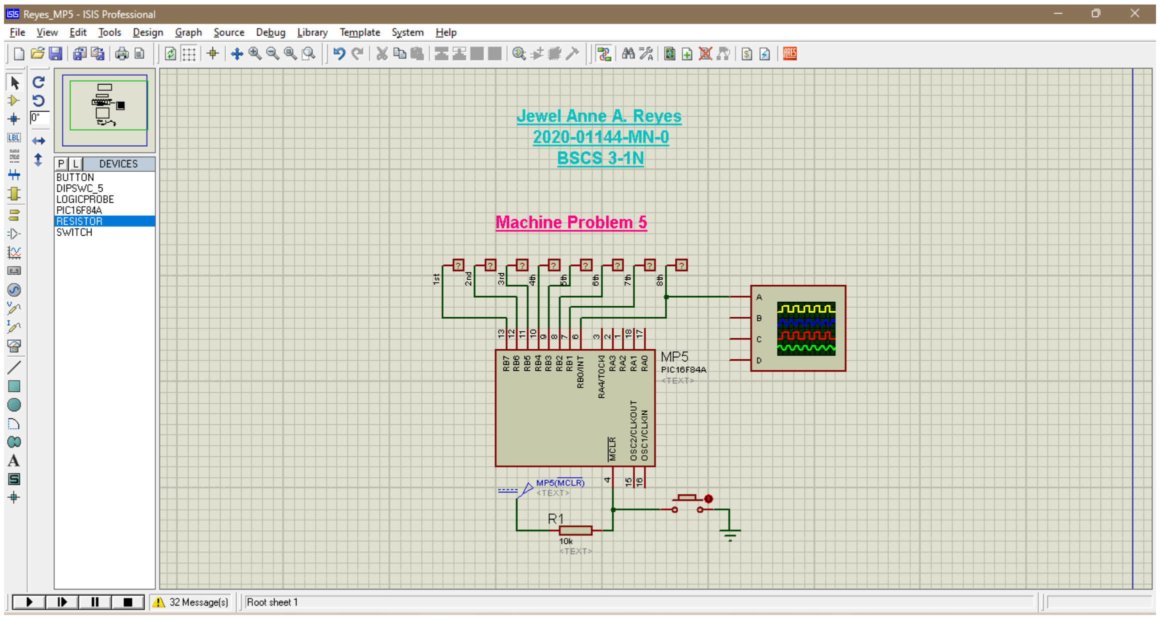
Procedure:

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

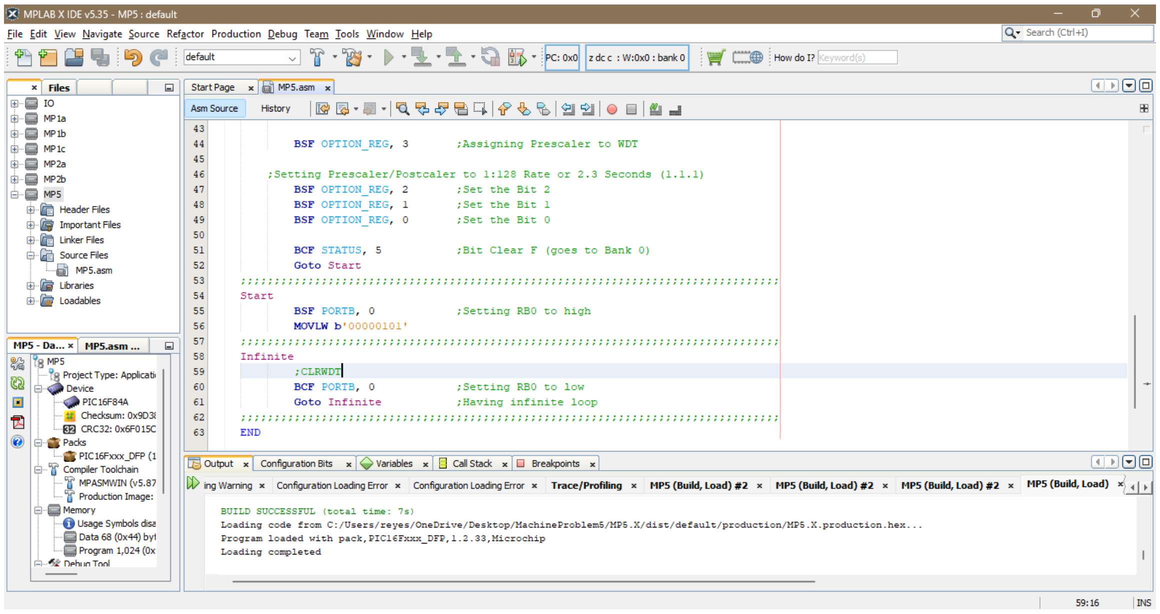
Set PORTB as output. Create a Code that will run light on PORTB for an interval of 1 sec. in a vise versa direction. Use the WDT to reset the display. Use CLRWDT instruction to reset the WDT to run the code properly. Set the Prescaler/Postscaler with 1:128 for 2.3 sec of watchdog timeout. Eventually delete the CLRWDT instruction and observe the response of the PORTB



Schematic:



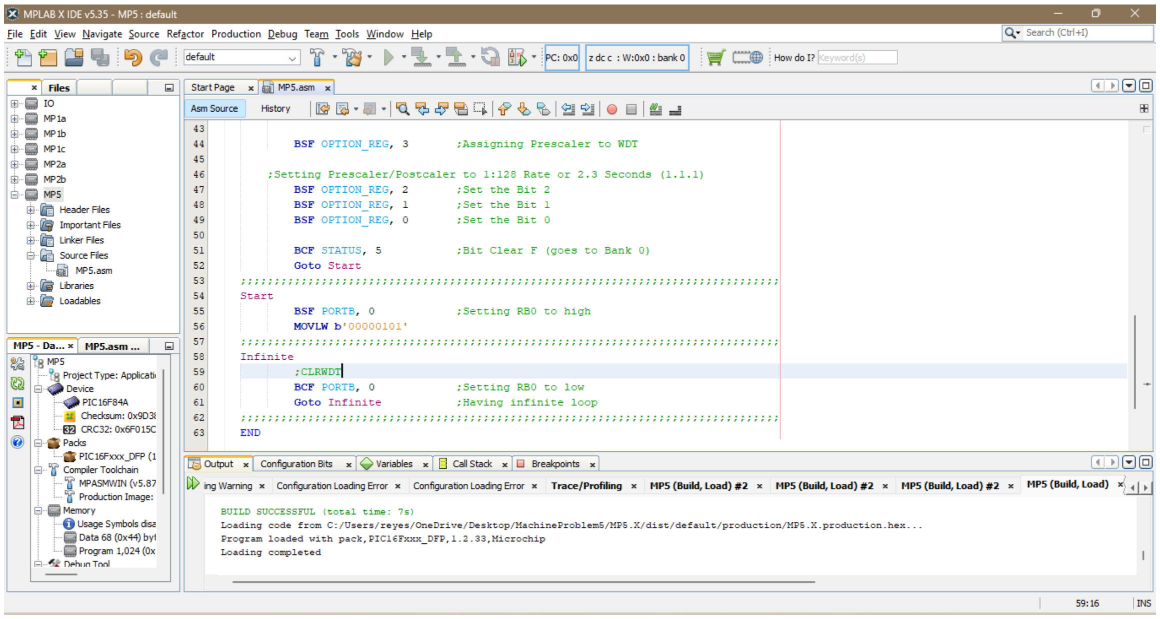
Source code without CLRWDT:



```
            BSF OPTION_REG, 3           ;Assigning Prescaler to WDT

        ;Setting Prescaler/Postcaler to 1:128 Rate or 2.3 Seconds (1.1.1)
            BSF OPTION_REG, 2           ;Set the Bit 2
            BSF OPTION_REG, 1           ;Set the Bit 1
            BSF OPTION_REG, 0           ;Set the Bit 0

            BCF STATUS, 5               ;Bit Clear F (goes to Bank 0)
            Goto Start
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
Start
            BSF PORTB, 0                ;Setting RB0 to high
            MOVLW b'00000101'
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
Infinite
            ;CLRWDT
            BCF PORTB, 0                ;Setting RB0 to low
            Goto Infinite               ;Having infinite loop
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
END
```

```
BUILD SUCCESSFUL (total time: 7s)
Loading code from C:/Users/reyes/OneDrive/Desktop/MachineProblem5/MP5.X/dist/default/production/MP5.X.production.hex...
Program loaded with pack,PIC16Fxxx_DFP,1.2.33,Microchip
Loading completed
```

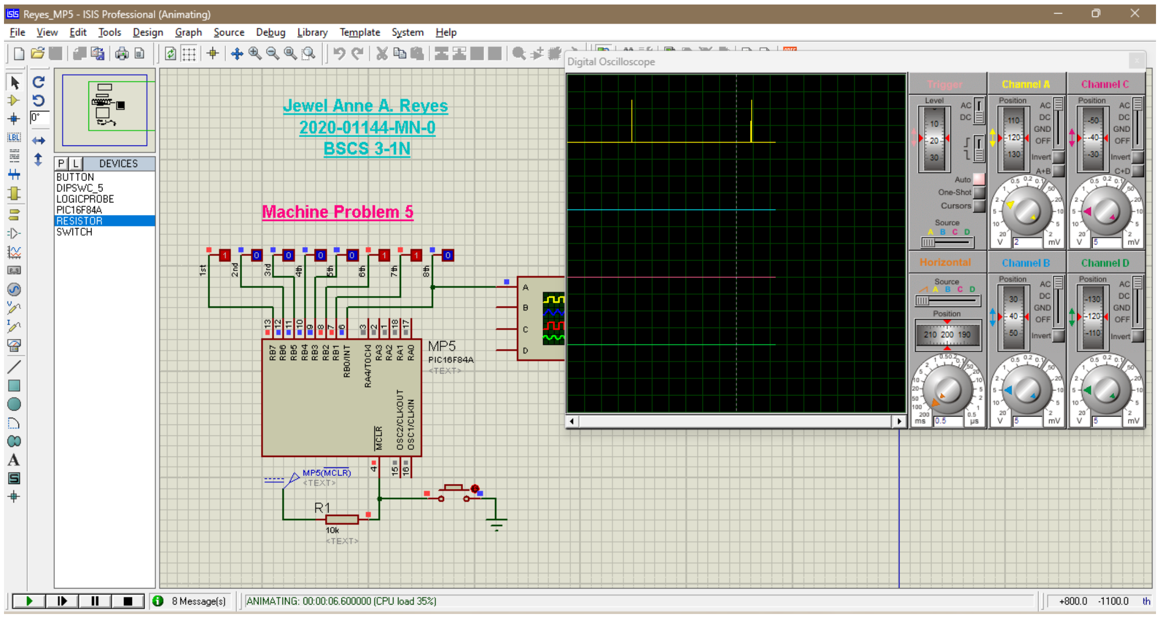
Source code with CLRWDT:

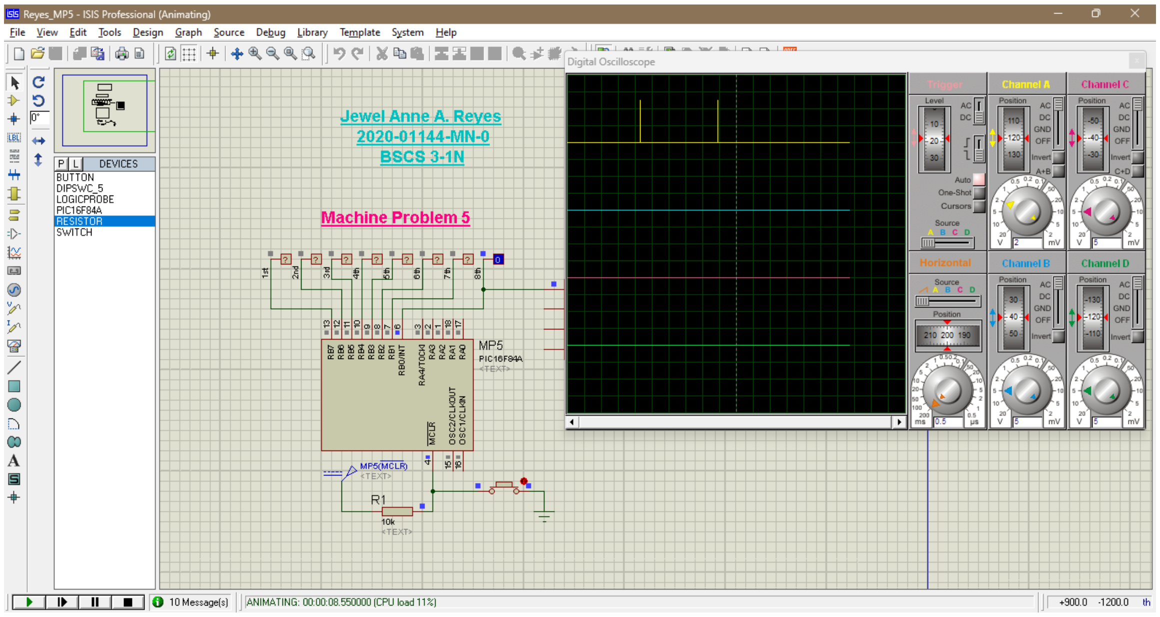


```
        BSF OPTION_REG, 3       ;Assigning Prescaler to WDT

    ;Setting Prescaler/Postcaler to 1:128 Rate or 2.3 Seconds (1.1.1)
        BSF OPTION_REG, 2       ;Set the Bit 2
        BSF OPTION_REG, 1       ;Set the Bit 1
        BSF OPTION_REG, 0       ;Set the Bit 0

        BCF STATUS, 5           ;Bit Clear F (goes to Bank 0)
        Goto Start
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
Start
        BSF PORTB, 0            ;Setting RB0 to high
        MOVLW b'00000101'
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
Infinite
        ;CLRWDT
        BCF PORTB, 0            ;Setting RB0 to low
        Goto Infinite           ;Having infinite loop
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
END
```

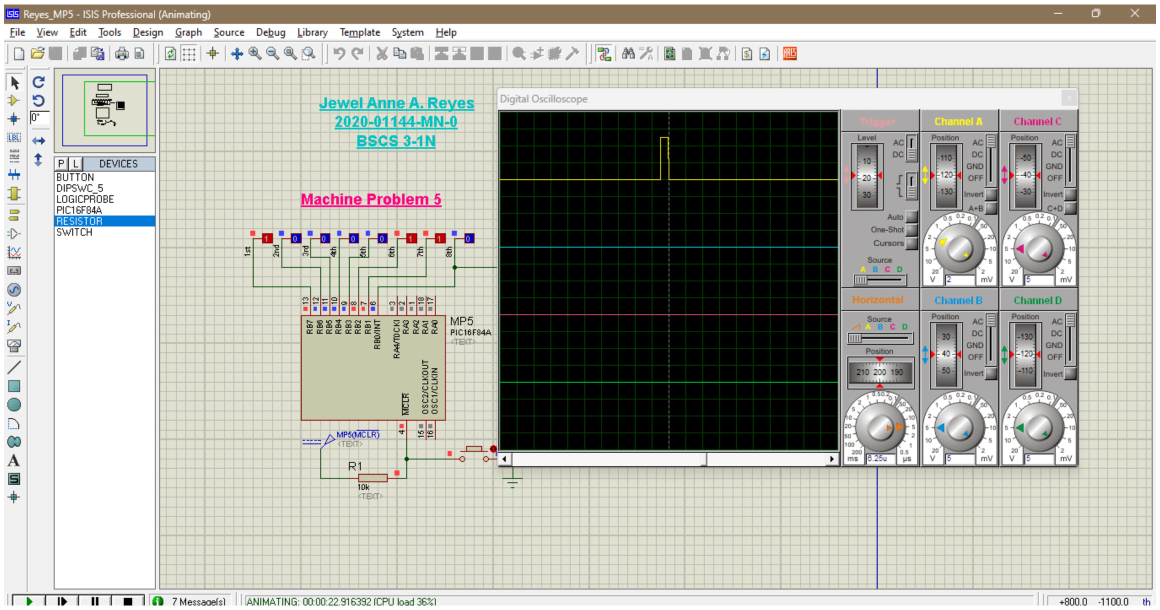
```
BUILD SUCCESSFUL (total time: 7s)
Loading code from C:/Users/reyes/OneDrive/Desktop/MachineProblem5/MP5.X/dist/default/production/MP5.X.production.hex...
Program loaded with pack,PIC16Fxxx_DFP,1.2.33,Microchip
Loading completed
```

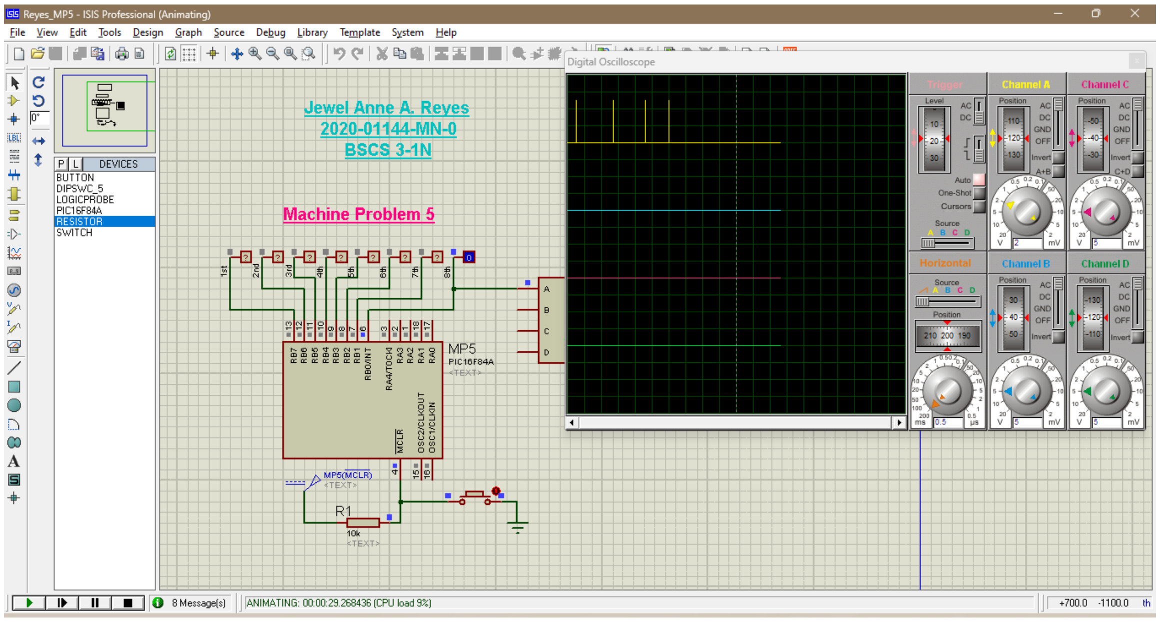
Output without CLRWDT:





Output with CLRWDT:





Observation:

I have observed that having the CLRWDT instruction has a huge effect in the result. The WDT is a hardware timer with its own clock that (if enabled) forces the PIC to reset rather like you had powered it off and on again. It can't be used as a method of creating a delay because there is no way of resuming when time is up, it always goes back to the reset vector. To prevent it resetting the PIC you have to stop it reaching maximum count by using the CLRWDT instruction within the timeout period.

I have also observed that the rate of prescaler/postscaler have a total impact especially in the time of Watchdog Timer. I tried using different settings of bits PS0, PS1, PS2 to have different WDT time and rate. As for me, I prefer the 1:1 rate or 18 mS because I can see clearly the digital clock in a bigger picture compared to the 2.3 seconds or 1:128 rate which has a setting of 1.1.1 bit of PS0, PS1, and PS2. This is because I cannot fully count or see the ratio on every rising and falling of signals on the oscilloscope.

At first, I wondered how to set the bits to have a specific rate or WDT time. I have tried using the BCF instruction set and noticed that it does not go according to my plan. Turns out, I overlooked it and used BCF instead of BSF when setting the bits to high.

In terms of using the oscilloscope, I have noticed that whenever the running light was lit at RB0 where the oscilloscope was connected, the digital clock was having two pulses and a curve which was weird. It was in a loop and whenever I pushed the button connected to the master clear, it has the same pulse.

Conclusion/ Recommendation:

The watchdog timer is essentially a counter that is constantly counting up. If the microcontroller is running normally, it periodically resets the watchdog timer, preventing it from reaching its maximum count value. However, if the microcontroller gets stuck or stops functioning for any reason, it will not be able to reset the watchdog timer, and it will eventually reach its maximum count value. When this happens, the watchdog timer will generate a reset signal, which will reset the microcontroller and allow it to start over from the beginning.

It is recommended to use different rates and WDT time to have a thorough observation about their differences. Also, it would be nice to use WDT with different setups especially the ones using loops to prevent being stuck in it. It is really helpful and a new experience for me as I have not noticed much about WDT when learning about circuits and microcontrollers when I am in high school.

Also, try using different delays to observe how it affected the instruction cycle. I think it will be great to try experimenting with the WDT as it monitors the system operations to detect if there are any faults or errors. It checks for software bugs, hardware errors, and other issues that could cause the system to malfunction.

Source Code:

(Please Attach the asm file)

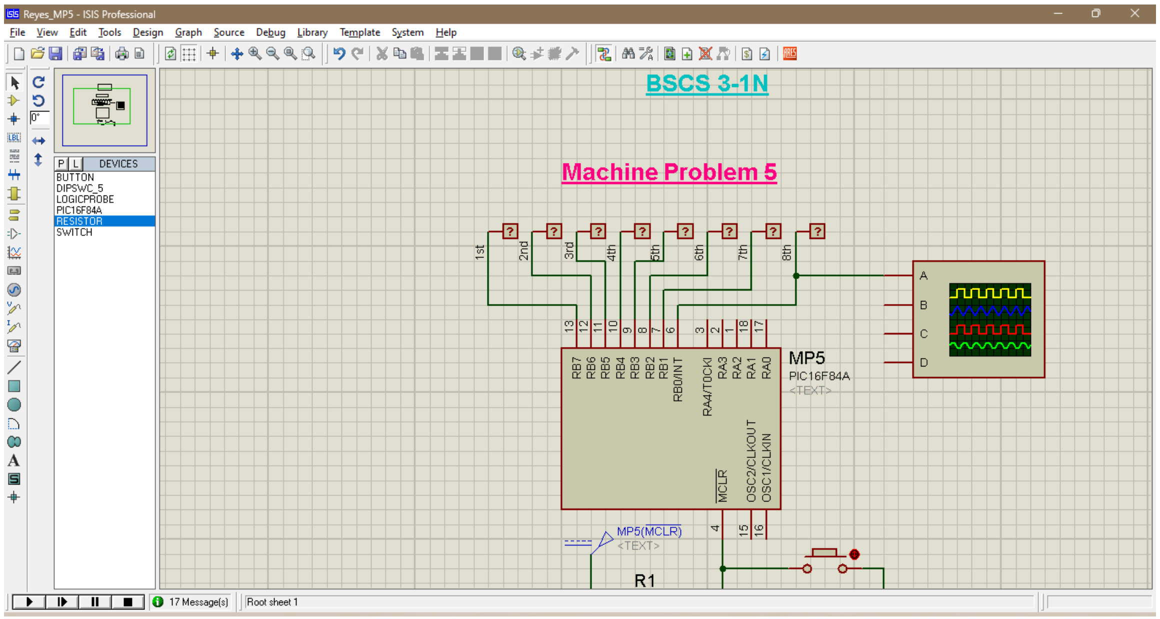
```
;###########################     COMMENT HEADER          #############################
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;  Author:          Jewel Anne A. Reyes                                     ;
;  Date:     February 28, 2023                              ;
;  Version:         1.0.0                                          ;
;  Title:    Machine Problem 5                              ;
;                                                                  ;
;  Description:     This Code shows the running light of           ;
;                   1 LED or logic probe with 1-sec interval.      ;
;                   It uses a Watchdog Timer with Prescaler/Postscaler      ;
;                   rate of 1:128 for 2.3 seconds.                 ;
;                                                                  ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


;#############    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_ON & _XT_OSC        ;Sets processor configuration bits
```

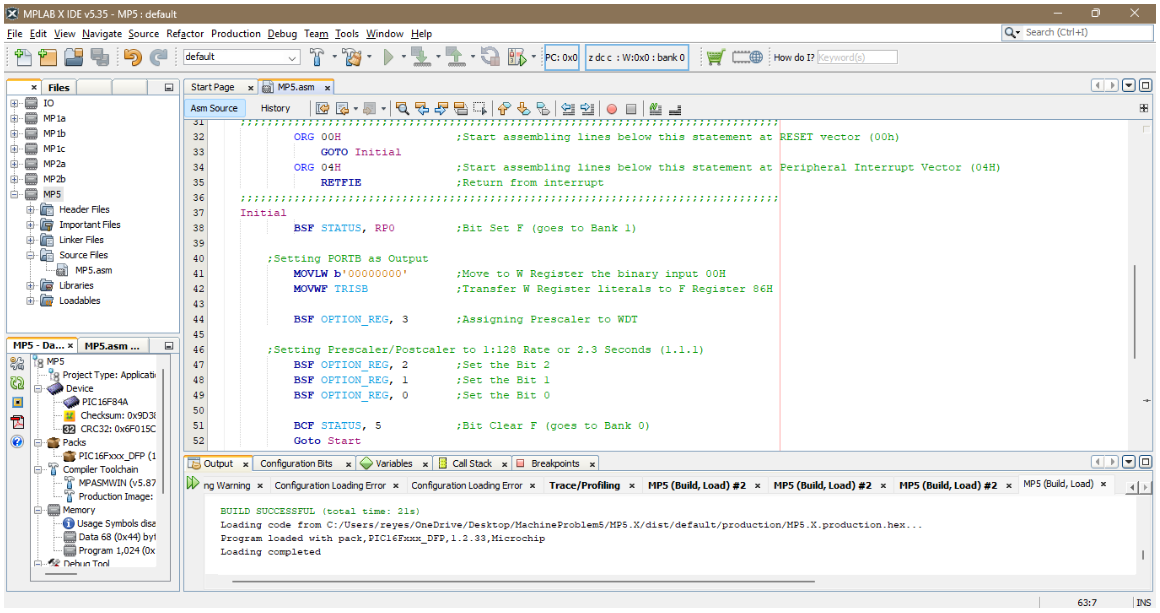
```
;##### DIRECTIVE EQU(equivalent) = text substitution for constant/variable #####
; Special Purpose Registers (SFR):
STATUS          EQU 03H             ;STATUS equivalent to 03H File Address
PORTA           EQU 05H
PORTB           EQU 06H
TRISA   EQU 85H
TRISB   EQU 86H
OPTION_REG  EQU 81H                 ;this is optional since it is already in p16f84a.inc

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
        ORG 00H                     ;Start assembling lines below this statement at RESET vector (00h)
          GOTO Initial
        ORG 04H                     ;Start assembling lines below this statement at Peripheral Interrupt Vector (04H)
          RETFIE            ;Return from interrupt
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
Initial
        BSF STATUS, RP0             ;Bit Set F (goes to Bank 1)

  ;Setting PORTB as Output
        MOVLW b'00000000';Move to W Register the binary input 00H
        MOVWF TRISB                 ;Transfer W Register literals to F Register 86H

        BSF OPTION_REG, 3           ;Assigning Prescaler to WDT

  ;Setting Prescaler/Postcaler to 1:128 Rate or 2.3 Seconds (1.1.1)
        BSF OPTION_REG, 2           ;Set the Bit 2
        BSF OPTION_REG, 1           ;Set the Bit 1
        BSF OPTION_REG, 0           ;Set the Bit 0

        BCF STATUS, 5               ;Bit Clear F (goes to Bank 0)
        Goto Start
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
Start
        BSF PORTB, 0                ;Setting RB0 to high
        MOVLW b'00000101'
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
Infinite
        CLRWDT                      ;Reset WDT
        BCF PORTB, 0                ;Setting RB0 to low
        Goto Infinite       ;Having infinite loop
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
END
```

Additionals:

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

The images below shows the schematic diagram of PIC16F84A, logic probes from RB0 to RB7 with oscilloscope, and source code in which I changed certain lines indicated in my observation.





```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
        ORG 00H                 ;Start assembling lines below this statement at RESET vector (00h)
            GOTO Initial
        ORG 04H                 ;Start assembling lines below this statement at Peripheral Interrupt Vector (04H)
            RETFIE              ;Return from interrupt
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
Initial
        BSF STATUS, RP0         ;Bit Set F (goes to Bank 1)

    ;Setting PORTB as Output
        MOVLW b'00000000'       ;Move to W Register the binary input 00H
        MOVWF TRISB             ;Transfer W Register literals to F Register 86H

        BSF OPTION_REG, 3       ;Assigning Prescaler to WDT

    ;Setting Prescaler/Postcaler to 1:128 Rate or 2.3 Seconds (1.1.1)
        BSF OPTION_REG, 2       ;Set the Bit 2
        BSF OPTION_REG, 1       ;Set the Bit 1
        BSF OPTION_REG, 0       ;Set the Bit 0

        BCF STATUS, 5           ;Bit Clear F (goes to Bank 0)
        Goto Start
```

```
BUILD SUCCESSFUL (total time: 21s)
Loading code from C:/Users/reyes/OneDrive/Desktop/MachineProblem5/MP5.X/dist/default/production/MP5.X.production.hex...
Program loaded with pack,PIC16Fxxx_DFP,1.2.33,Microchip
Loading completed
```