

Labpic06 Watchdog

วัตถุประสงค์

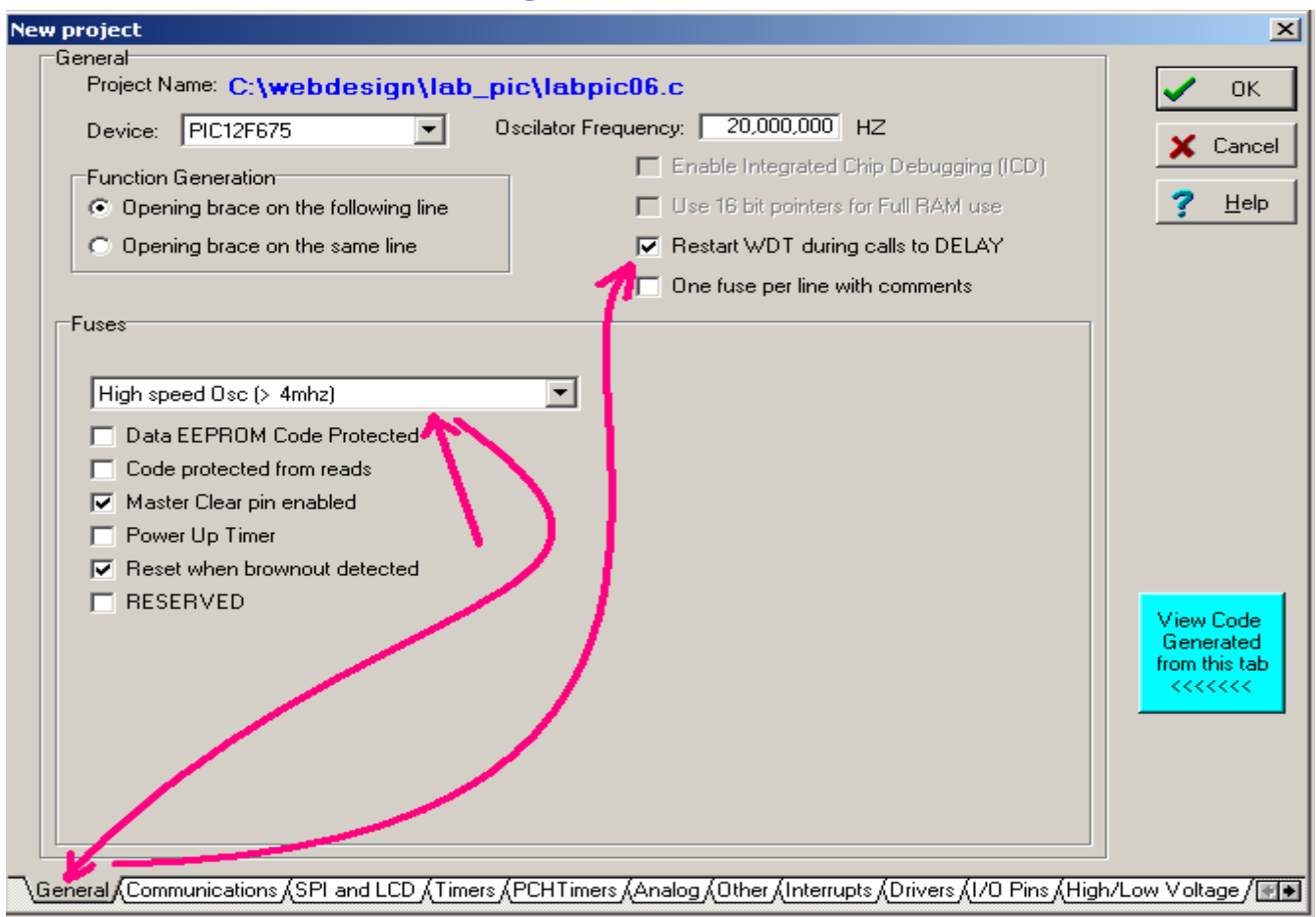
เพื่อให้เข้าใจโครงสร้างของโปรแกรมที่มีระบบตรวจสอบความผิดปกติของ **microcontroller**

ทำไมต้องมีระบบ **Watchdog**

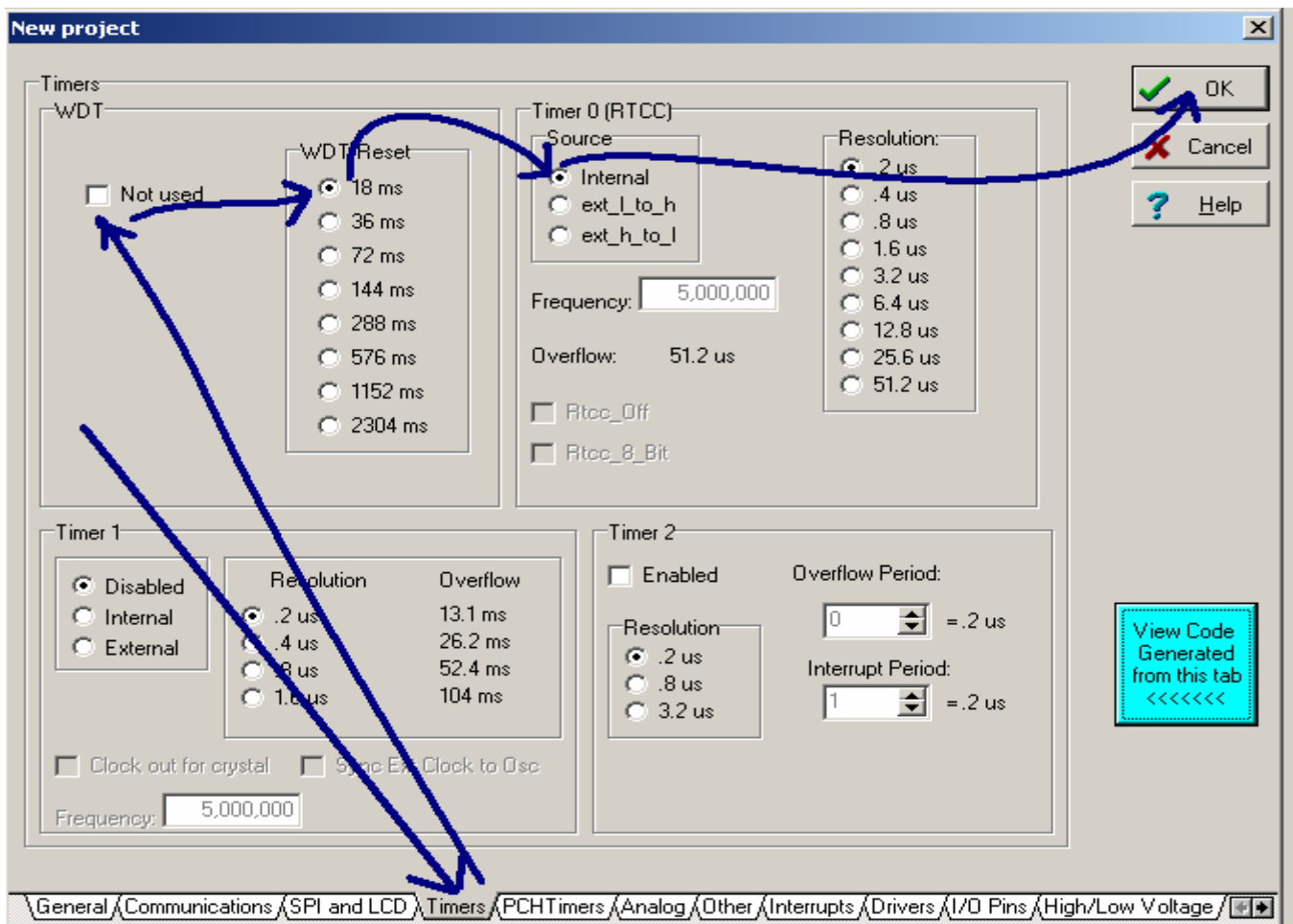
ในระบบการทำงาน ไม่ว่าจะเป็นใน เครื่องจักร รถยนต์กลไก หรือ แม้แต่ในหน่วยงานใด ๆ ก็ตาม ความผิดพลาดหรือความผิดปกติ เกิดขึ้นได้เสมอ บางครั้งถึงกับระบบต้องหยุดชงักลงเพื่อรอการตัดสินใจหรือตรวจสอบระบบ ซึ่งเหตุการณ์เหล่านี้ถ้าอยู่ในระบบที่ไม่ก่อให้เกิดความเสียหายมากนักก็คงไม่เป็นไร แต่ในบางระบบความผิดปกติเหล่านี้อาจก่อให้เกิดความเสียหายอย่างใหญ่หลวง ให้คุณลองนึกดูซิว่าถ้าคนขับรถของคุณเกิดหลับขณะที่ยารถกำลังวิ่งด้วยความเร็ว 240 กิโลเมตร/ชั่วโมง หรือ **ECU ใน Turbo Jet** ในเครื่องบินที่อยู่บนน่านฟ้าหยุดการทำงานมันจะเกิดอะไรขึ้นกับคุณ หรือ ผู้โดยสารบนเครื่อง

ระบบ **Watchdog** มีหน้าที่คอยดูแลการทำงานของ **CPU** เมื่อเกิดเหตุการณ์ผิดปกติที่ทำให้การทำงานใน **CPU** ขาดความต่อเนื่องระบบ **watchdog** จะสั่งระบบ **Reset** ให้ทำงานทันที

การสร้างโปรแกรมที่มีระบบ **watchdog**



รูป lab06_1 tab general define



📌 lab06_2 tab Timer Define

```

PCW C Compiler IDE
File Project Edit Options Compile View Tools Debug Help
Microchip 14 bit
labpic06.c
#include "C:\webdesign\lab_pic\labpic06.h"

void main()
{
    setup_adc_ports(NO_ANALOGS|USS_UDD);
    setup_adc(ADC_OFF);
    setup_timer_0(RTCC_INTERNAL);setup_wdt(WDT_18MS);
    setup_timer_1(T1_DISABLED);
    setup_comparator(NC_NC);
    setup_vref(UREF_LOW|-2);
}

```

📌 lab06_3 code from wizard

Copy file "C:\webdesign\lab_pic\labpic06.h" & replace on line 1

```
1#include <12F675.h>
2#device adc=8
3#FUSES WDT, HS, NOCPD, NOPROTECT, MCLR, NOPUT, BROWNOUT
4#use delay(clock=20000000,RESTART_WDT)

5void main()
6 {

7   setup_adc_ports(NO_ANALOGS|VSS_VDD);
8   setup_adc(ADC_OFF);
9   setup_timer_0(RTCC_INTERNAL);setup_wdt(WDT_18MS);
10  setup_timer_1(T1_DISABLED);
11  setup_comparator(NC_NC);
12  setup_vref(VREF_LOW|-2);

13 }
```

จากโปรแกรมส่วนที่คุณต้องทำความเข้าใจเพิ่มคือใน line 3 คือ “FUSES WDT” และใน line 4 #use delay(clock=20000000,RESTART_WDT) อธิบายได้คร่าว ๆ คือ มีการกำหนดให้ register ที่เกี่ยวข้องกับระบบ watchdog และ ให้มีการ reset watchdog ในส่วนของ sub program delay

และใน line 9 ใช้ timer0 เป็นตัวจับเวลาในความผิดพลาดของ cpu (หยุดทำงาน >18 msec) โดยใช้ฐานเวลาจาก internal clock นั้นหมายถึงว่าถ้ามีเหตุการณ์ผิดพลาดทำให้ cpu หยุดทำงานมากกว่า 18 msec ระบบ reset จะทำงานทันที

เพราะฉะนั้นโปรแกรมที่มีการเรียกใช้ watchdog ในส่วนของโปรแกรมที่มีการวนลูปที่ใช้เวลามากกว่า setup_wdt(WDT_18MS); จะต้องมีการสั่ง restart_wdt(); เสมอ

หมายเหตุ ให้ทำความเข้าใจเพิ่มจาก datasheet pic ในส่วนของ reset block diagram, watchdog blockdiagram และ จากโปรแกรมตัวอย่างต่อไปนี้

Exemple program

```
1 #include <12F675.h>
2 #device adc=8
3 #FUSES WDT, HS, NOCPD, NOPROTECT, MCLR, NOPUT, BROWNOUT
4 #use delay(clock=20000000,RESTART_WDT)

5 #define led pin_a0

6 void main()
7 {

8     setup_adc_ports(NO_ANALOGS|VSS_VDD);
9     setup_adc(ADC_OFF);
10    setup_timer_0(RTCC_INTERNAL);setup_wdt(WDT_18MS);
11    setup_timer_1(T1_DISABLED);
12    setup_comparator(NC_NC);
13    setup_vref(VREF_LOW|-2);
14    //=====
15    output_high(led);
16    delay_ms(1000);
17    output_low(led);
18    delay_ms(1000);
19    //-----
20    while(1)
21    {

22        restart_wdt();
23        output_high(led);

24    }
25 }
```

เนื่องจากโปรแกรมนี้อยู่ในโปรเจกต์ที่ไม่สามารถตรวจสอบการทำงานของโปรแกรมนั้นได้โดยใช้ soft ware proteus ได้ จำเป็นต้องใช้ hardware จริง โดยให้ทดสอบจากความแตกต่างระหว่าง มี กับ ไม่มี คำสั่ง line20 (restart_wtd();)