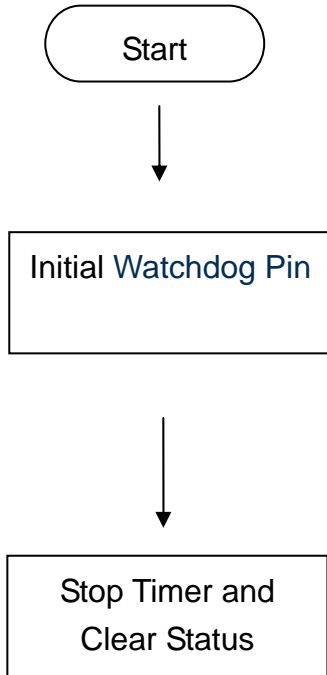




Watch Dog Timer Configuration

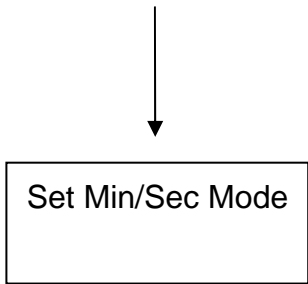
The WDT is used to generate a variety of output signals after a user programmable count. The WDT is suitable for use in the prevention of system lock-up, such as when software becomes trapped in a deadlock. Under these sort of circumstances, the timer will count to zero and the selected outputs will be driven. Under normal circumstance, the user will restart the WDT at regular intervals before the timer counts to zero.

CODE FLOW-CHART

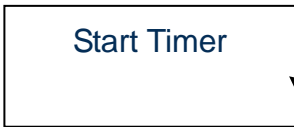


```
1. Initial Watchdog Pin
(1)set -> Logical Device: Global
        CR: 0x2D
        Bit: 0 = 0  WDTO.
(2)set -> Logical Device: 0x09
        CR: 0X30
        Bit:3 = 1 activate GPIO5
        CR: 0xE0, 0xE1, 0xE2
        Bit: 6, Bit: 7, Bit: 6
            = 0 GPIO port is programmed as an output port.
```

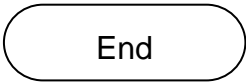
```
1. Stop timer
(1)set -> Logical Device: 0x08
        CR: 0xF6
        Bit: 7~0 = 0x00 Stop timer
(2)Set -> Logical Device 0x08
        CR 0x30
        Bit:0 = 1, activate WDTO
2. Clear Status
(1)set -> Logical Device: 0x08
        CR: 0xF7
        Bit: 4 = 0 Timer counting
```



1. Set Mode
(1)set -> Logical Device: 0x08
CR: 0xF5
Bit: 3
= 0 By second
= 1 By minute



1. Set Mode
(1)set -> Logical Device: 0x08
CR: 0xF6
Bit: 7~0
= 0x01 Time- out occur after 1
second/minute
.
.
.
= 0xFF Time- out occur after 255 second/minute



SAMPLE CODE:

This code and information is provided "as is" without warranty of any kind, either expressed or implied, including but not limited to the implied warranties of merchantability and/or fitness for a particular purpose. This program just for MB with Winbond W83267DHG LPC I/O

1. InitWatchdogPin:

```
outportb(0x2E,0x87);      outportb(0x2E,0x87);      outportb(0x2E,0x87);
outportb(0x2E,0x87);      outportb(0x2E,0x87);      outportb(0x2E,0x87);
outportb(0x2E,0x07);      outportb(0x2E,0x07);      outportb(0x2E,0x07);
outportb(0x2F,0x01);      outportb(0x2F,0x09);      outportb(0x2F,0x09);
outportb(0x2E,0x2D);      outportb(0x2E,0xE0);      outportb(0x2E,0xE2);
outportb(0x2F,byBit);      outportb(0x2F,byBit);      outportb(0x2F,byBit);
outportb(0x2E,0xAA);      outportb(0x2E,0xAA);      outportb(0x2E,0xAA);
```

```
outportb(0x2E,0x87);      outportb(0x2E,0x87);
outportb(0x2E,0x87);      outportb(0x2E,0x87);
outportb(0x2E,0x07);      outportb(0x2E,0x07);
outportb(0x2F,0x09);      outportb(0x2F,0x09);
outportb(0x2E,0x30);      outportb(0x2E,0xE1);
outportb(0x2F,byBit);      outportb(0x2F,byBit);
outportb(0x2E,0xAA);      outportb(0x2E,0xAA);
```

2. Stop Timer

```
outportb(0x2E,0x87);  
outportb(0x2E,0x87);  
outportb(0x2E,0x07);  
outportb(0x2F,0x08);  
outportb(0x2E,0xF6);  
outportb(0x2F,byBit);  
outportb(0x2E,0xAA);
```

```
outportb(0x2E,0x87);  
outportb(0x2E,0x87);  
outportb(0x2E,0x07);  
outportb(0x2F,0x08);  
outportb(0x2E,0x30);  
outportb(0x2F,byBit);  
outportb(0x2E,0xAA);
```

```
outportb(0x2E,0x87);  
outportb(0x2E,0x87);  
outportb(0x2E,0x07);  
outportb(0x2F,0x08);  
outportb(0x2E,0xF7);  
outportb(0x2F,byBit);  
outportb(0x2E,0xAA);
```

3. Set Min/Sec Mode

```
outportb(0x2E,0x87);  
outportb(0x2E,0x87);  
outportb(0x2E,0x07);  
outportb(0x2F,0x08);  
outportb(0x2E,0xF5);  
outportb(0x2F,byBit);  
outportb(0x2E,0xAA);
```

4. Start Timer

```
outportb(0x2E,0x87);  
outportb(0x2E,0x87);  
outportb(0x2E,0x07);  
outportb(0x2F,0x08);  
outportb(0x2E,0xF6);  
outportb(0x2F,byBit);  
outportb(0x2E,0xAA);
```