

## CHAPTER 1

### INTRODUCTION

The PCI watch dog timer2 adapter is your systems and peripheral applications protector! PCI watch dog timer2 adapter provides user selectable refresh timer to be used inside your PC, which prevents the computer to hang by automatically resetting the system of your computer. When the PCI watch dog timer2 adapter is enable, the system software or application program must refresh watch dog before selected time slice is reached, otherwise the computer will be reset, and the power of power control box that connects to peripheral devices will also be reset.

The PCI watch dog timer2 adapter is a 32 bits PCI bus board with Plug and Play (PnP) features; it is a programmable I/O interface card for Pentium, or compatibles. The PnP features let hardware configuration for IRQ and I/O address is detected by BIOS automatically, you don't need set switch and jumper.

In the distribution diskette, we provide software for Windows/95, Windows/98, Windows/NT4.0 workstation, and Windows/2000 operating systems.

#### **The features of the PCI watch dog timer2 adapter are:**

- Provides user selectable refresh timer.
- 32 bits PCI bus with Plug and Play (PnP) features.
- When the application program or computer system does not generate refresh signal, the computer will be reset automatically.
- The maximum distance of RG59 cable from card to power control box is up to 500m.

- Available with 110V and 240V (10A) Plug type UK, Europe, USA and Austria type selectable for extension power control box.
- I/O address selectable.
- Pentium hardware compatibles.
- Suitable for Windows/95, Windows/98, Windows/NT, Windows/2000 operating systems.

### ***1.1 Applications***

On industrial application, lots of computers are automatically operating itself once the power turns on. In case the application got hang surely the computer and its operation hang also. If you don't boot your computer again it will not go back to its regular operation, but not with the watchdog! Because once the watchdog was not refresh, it will reset the computer automatically!

While using your computer, unexpectedly your modem got hang. One way of connecting it again is to switch off then turn on your modem to continue on working. But now, it's the new way around, PCI watch dog timer2 will be automatically reset the modem power. It will protect your application software system.

### ***1.2 The Difference between Watch Dog I and II***

The advantage of watch dog II over the watch dog I is that the watch dog II has an expansion power control box that can control external component. When reset signal activate of the watch dog II, it will also reset the power of power control box.

User can also reset the power of power control box by watch dog II directly, without reset whole the computer system.

## CHAPTER 2

### UNPACKING INFORMATION

☞ Check that your watch dog package includes one of the following items:

#### PCI WATCHDOG TIMER2

- PCI watch dog timer2 adapter.
- Watch dog expansion power control box.
- User manual.
- Power cable.
- RG59AU Connection cable (1.5m).
- Reset cable.
- Utility software.

## CHAPTER 3

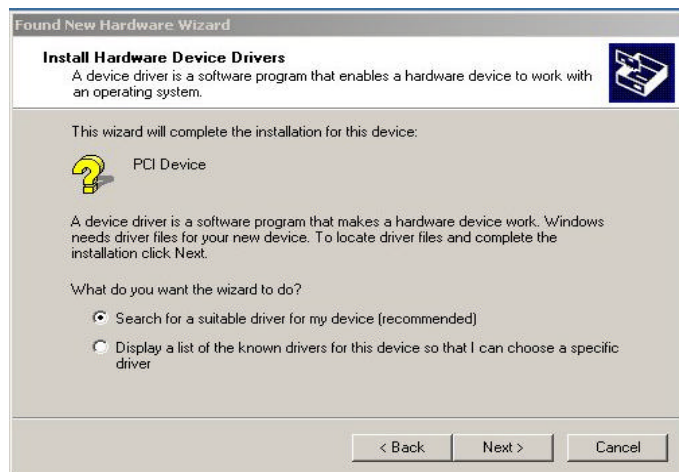
### HARDWARE INSTALLATION

The steps for installation are shown as follows:

1. Turn off the Power of your PC, insert the PCI Watchdog card to the PCI expansion Slot or any available PCI slot of you motherboard.
2. Connect the J1 to the reset switch of motherboard.
3. Connect the J2 to the reset switch of PC panel.
4. Connect RG59AU connector to expansion power control box, and then connect the power control box to the power plug. Also connect peripheral power-to-power control box if necessary.
5. Turn on the power of your PC, since you are installing a PCI card, your Windows 2000 will automatically detect the newly installed card and will look for a driver as shown below. To continue, click Next.



6. The windows is now ready to search for a driver, click Next.



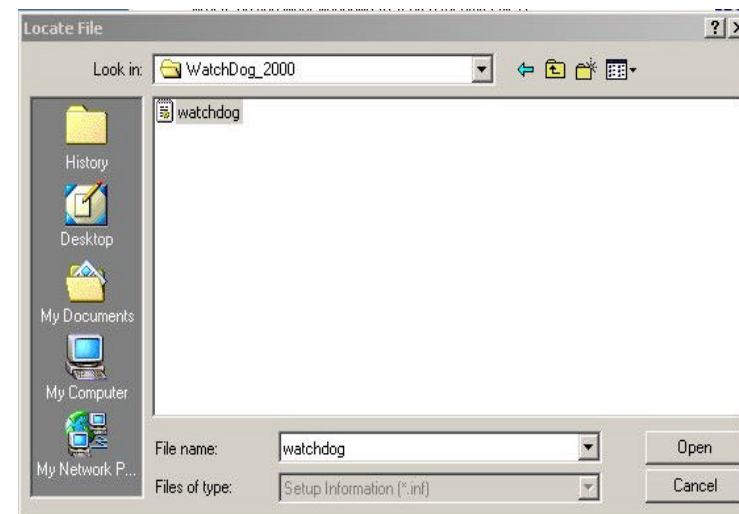
7. You will have to specify the driver location of WatchDog 2000, then click Next.



8. After clicking the Next button, you will be prompt to a dialog box to browse the location of the Watchdog 2000 driver. Click OK.



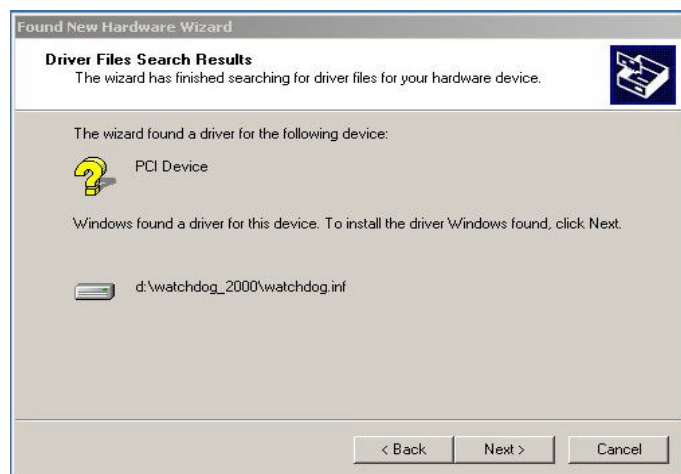
9. To locate the WatchDog2000 driver, point your directory where in the driver is copied.



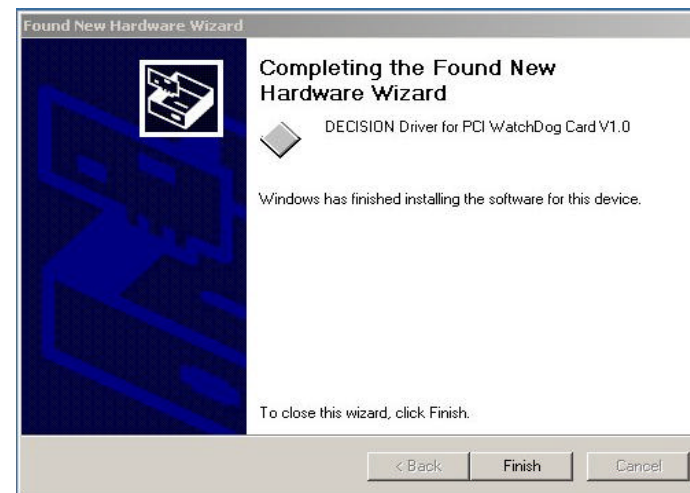
10. After you have located the directory of the driver, just click the OK button.



11. The search results will be display on screen, indication or presence of Watchdog 2000 driver should be displayed. To continue, click Next



12. The Watchdog 2000 driver installation is already complete, to continue, click Finish



## CHAPTER 4

### HARDWARE CONFIGURATION

#### 4.1 Addressing and Jumper Settings

Be sure to connect J1 and J2 at reset signal of main board and to reset push button, then get the correct I/O address from plug and play function.

##### 1. I/O address

| ADDRESS    | SPECIFICATION                         |
|------------|---------------------------------------|
| Base + 00H | Enable or Refresh Watch Dog (READ)    |
| Base + 01H | Disable Watch Dog (READ)              |
| Base + 01H | Turn ON/OFF power control box (WRITE) |

When you install PCI card, the base address will be assigned from the system automatically.. It is easily to read the corresponding address to trigger the enable (refresh) or disable watch dog adapter. To turn OFF power control box, just write "1" to (Base + 01H), to turn ON power control box, write "0" to (Base + 01H).

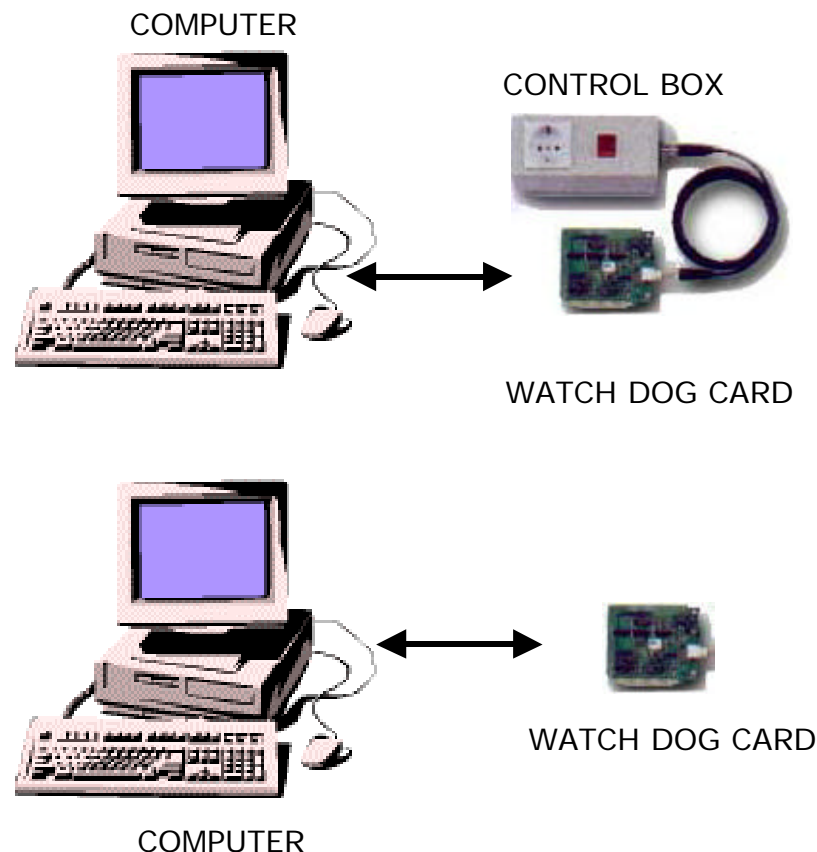
##### 2. J1 and J2

Please connect J1 to reset connector of your CPU main board, and connect J2 to reset push button on the front panel of your PC.

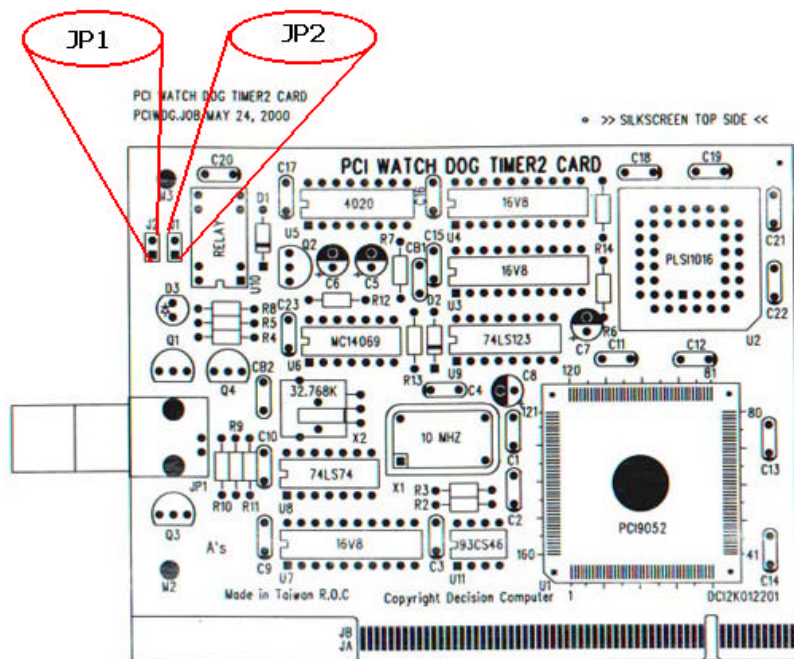
##### 3. RG59AU Connector

The RG59AU connector of Watch Dog is used to connect to

expansion power control box.

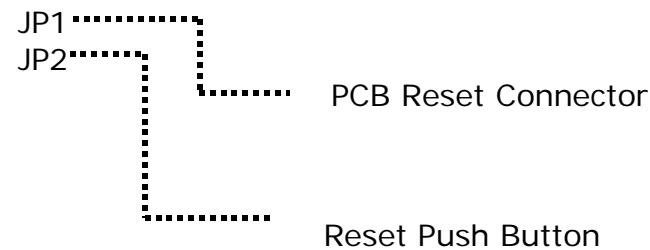


## PCI WATCH DOG II



## 4.2 Hardware Configurations

### JP1 and JP2 Connection for watch dog2



## CHAPTER 5

### SOFTWARE

In the distribution diskette, we support the device driver for Windows/95, Windows/98, Windows/2000, and Windows/NT to access adapter, and provide auto or manual refresh control.

When watch dog software is installed, user may select auto refresh or manual refresh mechanism. To select auto refresh, the watch dog software will refresh watch dog timer adapter automatically, except the system hang (when system hang, the watch dog adapter will reset the computer system). To select manual refresh, user must refresh watch dog adapter by application program on time; otherwise the system will be reset. Under multitasking (multiprogramming, multithread, ... etc.) operating system, the time schedule can not control accuracy by application program, so that if you use manual refresh, please refresh the watch dog more quicker than selected time slice.

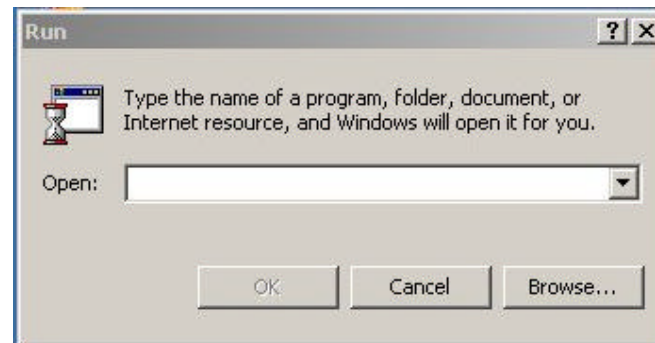
After watch dog software is installed and set to manual mode, user need enable watch dog timer, then refresh it, or disable the watch dog timer. The enhanced functions of watch dog will be used to turn OFF/ON power control box.

## 5.1 Software Installation

1. Click the Start button, select Run and click it.

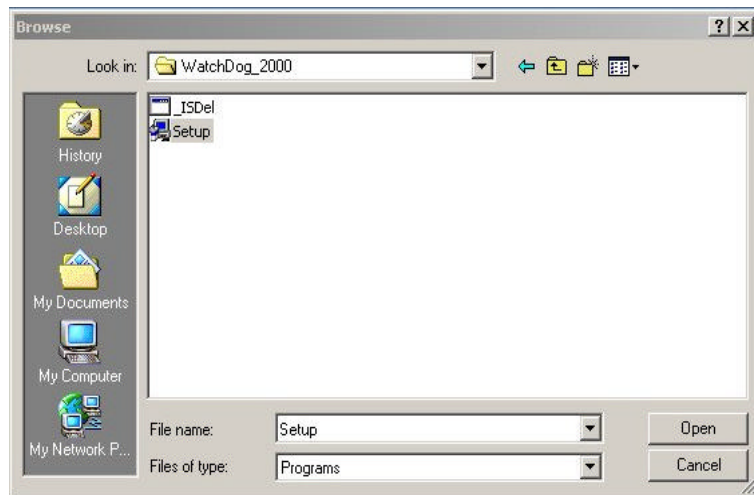


2. After you click the RUN, a dialog box will come up on the screen, click the Browse button.

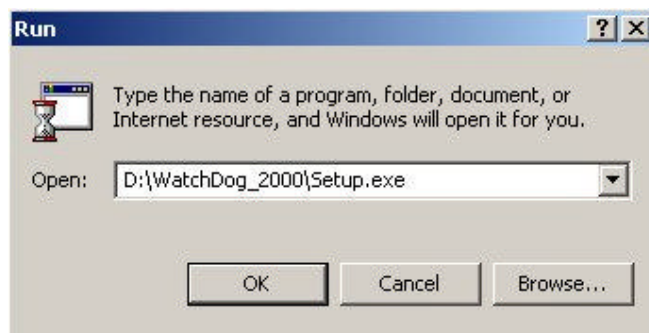




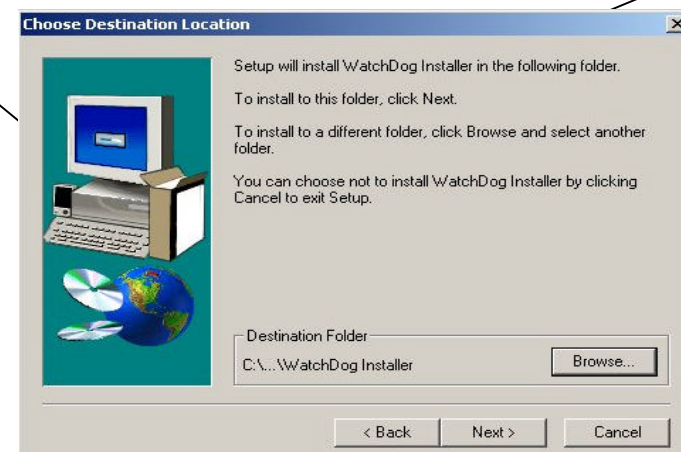
- After you click the Browse button, Browse directory will come up on the screen, in order for you to select the location of the driver of the Watchdog2000. Click the Setup then click Open button



- After clicking the Open button, Setup.exe will be on the Run dialog box, to execute, click OK



- The Setup.exe will now run, and will show something like the window below. The installation setup will ask for a Destination folder, to select the default folder, just click Next.

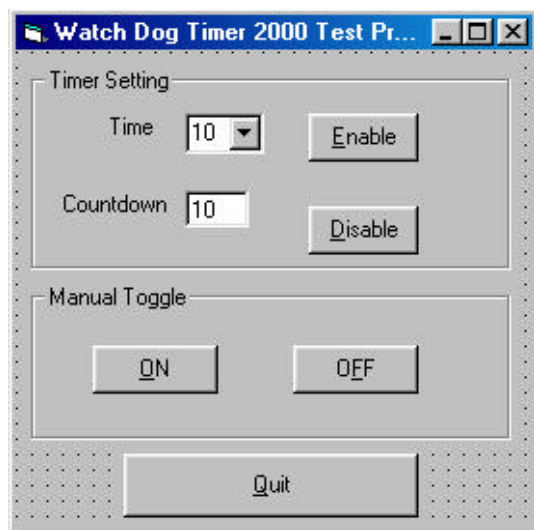


- To select you desired program folder, type the name or select the default name and click Next.





7. After the installation made, click the Start button, then Programs, then select WatchDog Timer.



## 5.2 Developing Software

Developing software on watch dog is very simple. Below are the sample programs under different programming languages:

### 1. EXAMPLE PROGRAM USING BASIC:

```
REM
REM WDSIMPLE.BAS -> Simple Program for Watchdog (I and
II)
REM Program by Edgardo A Regodon Jr(SrSE)
REM of Decision Asia Inc. (Phil.)
REM
```

```
DIM UserChoice AS STRING * 1      ' Variable Declaration
DIM WDAAddress AS INTEGER
DIM status AS STRING * 1
DIM Feedback AS STRING * 1
DIM Counter1 AS INTEGER
DIM counter2 AS INTEGER
```

```
status = "F"
```

```
CLS                                ' Get The Watch Dog Address
PRINT "Please Enter the address of Watch Dog Card (I/II) in
decimal"
PRINT "Address Range from 0 to 4094"
INPUT "Watch Dog address is: ", WDAAddress
```

```
Inquire:
CLS                                ' Main Menu
PRINT ("Menu Choices")
PRINT ("1] Enable/Disable")
PRINT ("2] Refresh")
PRINT ("3] Exit")
INPUT "Please enter your choices: ", UserChoice
```

```

SELECT CASE UserChoice
CASE "1"                                ' Enable Disable Watch Dog
    IF status = "F" THEN
        status = "T"
        value = INP(WDAddress)
        PRINT "The Watch Dog has been enabled. . ."
    ELSE
        status = "F"
        value = INP(WDAddress + 1)
        PRINT "The Watch Dog has been disabled. . ."
    END IF
    INPUT "Press ENTER key to continue", temp
    GOTO Inquire
CASE "2"                                ' Refresh Watch Dog
    IF status = "T" THEN
        value = INP(WDAddress + 1)
        value = INP(WDAddress)
        PRINT "Watch Dog Refresh"
    ELSE
        PRINT "The Watch Dog is currently not enable. . ."
        PRINT "You have to enable the Watch Dog first to refresh it"
    END IF
    INPUT "Press ENTER key to continue", temp
    GOTO Inquire
CASE "3"                                ' QUIT Watch Dog
    value = INP(WDAddress + 1)
    END
CASE ELSE                                ' Wrong selection
    PRINT "Please select 1, 2 or 3 only"
    INPUT "Press ENTER key to continue", temp
    GOTO Inquire
END SELECT

```

## 2. EXAMPLE PROGRAM USING PASCAL

```

{
WDSimple.PAS -> Simple Program for Watchdog (I and II)
Program by Edgardo A Regodon Jr (SrSE) of Decision Asia Inc.
(Phil.)
Jan 1998
}

uses crt,dos;

label
Inquire;

var
UserChoice,ans:char;
WDAddress,Value:integer;
Status:Boolean;

BEGIN
Status := FALSE;
clrscr;
writeln('Please Enter the address of Watch Dog Card (I/II) in
hexadecimal');
writeln('Address Range from 0 decimal to 4094 decimal');
write('Wathc Dog address is: ');
read(WDAddress);

Repeat
clrscr;
writeln('Menu Choices');
writeln("");
writeln('1] Enable/Disable');

```

```
writeln('2] Refresh');
writeln('3] Exit');
write('Please enter your choices: ');
UserChoice := readkey;
delay(70);

case UserChoice of
'1':
    if (Status = FALSE) then
        begin
            Status := TRUE;
            value:=port[WDAAddress];
            writeln('The Watch Dog has been enabled. . .');
            ans := readkey;
        end
    else
        begin
            Status := FALSE;
            value:=port[WDAAddress+1];
            writeln('The Watch Dog has been disabled. . .');
            ans := readkey;
        end;
'2':
    if (Status = TRUE) then
        begin
            value:=port[WDAAddress+1];
            value:=port[WDAAddress];
            writeln("");
            write('Watch Dog Refresh');
            delay(500);
        end
    else
        begin
```

```
writeln('The Watch Dog is currently not enable. . .');
write('You have to enable the Watch Dog first to refresh it');
ans:=readkey;
end;
'3':
    value:=port[WDAAddress+1];
else
    writeln("");
    writeln('Please select 1, 2 or 3 only');
    readkey;
end;
until (UserChoice = '3');

end.
```

### 3. EXAMPLE PROGRAM USING C++

```

/*
WDSIMPLE.CPP -> Simple Program for Watchdog (I and II)
Program by Edgardo A Regodon Jr(SrSE)
of Decision Asia Inc. (Phil.)
*/

#include <stdio.h>           /* Headers Declaration */
#include <conio.h>
#include <dos.h>
#include <string.h>

void main()
{
    char UserChoice;         /* Variable Declaration */
    unsigned int WAddress;
    char Boolean;
    Boolean = 'F';

    clrscr();                /* Inquire for the Address */
    printf("Please Enter the address of Watch Dog Card (I/II) in  
hexadecimal\n");
    printf("Address Range from 0 Hex to 0FFE Hex\n");
    printf("Wathc Dog address is: ");
    scanf("%x", &WAddress);

    Inquire:
    clrscr();                /* Main menu */
    printf("Menu Choices\n\n");
    printf("1] Enable/Disable\n");
    printf("2] Refresh\n");
    printf("3] Exit\n\n");
}

```

```
printf("Please enter your choices: ");

UserChoice = getche();
delay(70);

switch(UserChoice) {          /* Action to do based on choices */
case '1':                    /* Enable Disable function */
    if (Boolean == 'F')
    {
        Boolean = 'T';        /* Enable Watch Dog */
        inportb((unsigned int)WDAddress);
        printf("\nThe Watch Dog has been enabled. . .");
        getch();
    }
    else                    /* Disable Watch Dog */
    {
        Boolean = 'F';
        inportb((unsigned int)WDAddress+1);
        printf("\nThe Watch Dog has been disabled. . .");
        getch();
    }
    goto Inquire;
case '2':                    /* Refresh function */
    if (Boolean == 'T')
    {
        /* If Watch Dog is enabled */
        inportb((unsigned int)WDAddress+1);
        inportb((unsigned int)WDAddress);
        printf("\nWatch Dog Refresh");delay(500);
    }
    else                    /* If Watch Dog is disabled */
    {
        printf("\nThe Watch Dog is currently not enable. . \n");
        printf("You have to enable the Watch Dog first to refresh it");
    }
}
```

```

        getch();
    }
    goto Inquire;
case '3':          /* Quit the program */
    inportb((unsigned int)WDAddress+1);
    break;
default:           /* error in selection */
    printf("\nPlease select 1, 2 or 3 only");
    getch();
    goto Inquire;
}
}

```

### 5.3 Files on Windows

|              |  |
|--------------|--|
| WATCHDOG.C   | : The sample file to test watch dog card |
| WATCHDOG.EXE | : The test program for watch dog card    |
| WATCHDOG.H   | : The include file for test program      |
| WATCHDOG.RC  | : The example file for DLL               |
| WDGWIN95.C   | : The example file for DLL               |
| WDGWIN95.DLL | : The DLL for development                |
| WATCHDOG.DLL | : The additional DLL for development     |
| WDGWIN95.H   | : The include file for DLL               |
| WDGWIN95.LIB | : The LIB for development                |

EXAMPLE OF USING WATCHDOG.DLL

### 1. Under Visual Basic 5.0

#### Module Declaration under VB

Option Explicit

Declare Function WDEnable Lib "WATCHDOG.DLL" (ByVal  
Addr As Integer) As Integer  
Declare Function WDDisable Lib "WATCHDOG.DLL" (ByVal  
Addr As Integer) As Integer  
Declare Function WDRrefresh Lib "WATCHDOG.DLL" (ByVal  
Addr As Integer) As Integer

**Inside the form in VB with 3 Command named cmdDisable,  
cmdEnable and cmdRefresh**

```

Private Sub cmdDisable_Click()
Dim FeedBack As Integer
    FeedBack = WDDisable(&H240) ' Disable Watch Dog
End Sub

```

```

Private Sub cmdEnable_Click()
Dim FeedBack As Integer
    FeedBack = WDEnable(&H240) ' Enable Watch Dog
End Sub

```

```

Private Sub cmdRefresh_Click()
Dim FeedBack As Integer
    FeedBack = WDRrefresh(&H240) ' Refresh Watch Dog
End Sub

```

## 2. Under Visual C++ 5.0

### Under initialization:

```

BOOL CWDTestDlg::OnInitDialog()
{
    CDialog::OnInitDialog();
    .....
    ....
    // TODO: Add extra initialization here
    // initialization goes here
    // if watchdog.dll is already been loaded, terminate this
function
    if (gLibMyDLL != NULL)
    {
        MessageBox("The WatchDog.DLL dll has already been
loaded.");
        return(0);
    }
    // Load the WatchDog.DLL dll.
    gLibMyDLL = LoadLibrary("WATCHDOG.DLL");
    // if the DLL was not loaded successfully
    if (gLibMyDLL == NULL)
    {
        char msg[300];
        strcpy(msg, "Cannot load the WATCHDOG.DLL DLL.");
        strcat(msg, "Make sure that the file WATCHDOG.DLL");
        strcat(msg, "is in your \\WINDOWS\\SYSTEM
directory");
        MessageBox( msg );
    }
    // Get the address of the functions on the DLL

```

```

        WDEnable = (WDENABLE)GetProcAddress(gLibMyDLL,
"WDEnable");
        WDDisable = (WDDISABLE)GetProcAddress(gLibMyDLL,
"WDDisable");
        WDRrefresh = (WDREFRESH)GetProcAddress(gLibMyDLL,
"WDRrefresh");
        // end of (loading) initialization
        return TRUE; // return TRUE unless you set the focus to
a control
    }

```

### Codes on Buttons:

```

void CWDTestDlg::OnButtonEnable()
{
    // TODO: Add your control notification handler code here
    if (gLibMyDLL == NULL)
    {
        MessageBox("WATCHDOG.DLL NOT LOADED.");
        return;
    }
    WDEnable(0x240); // Enable Watchdog with 240 hex address
}

void CWDTestDlg::OnButtonDisable()
{
    // TODO: Add your control notification handler code here
    if (gLibMyDLL == NULL)
    {
        MessageBox("WATCHDOG.DLL NOT LOADED.");
        return;
    }
    WDDisable(0x240); // Disable Watchdog with 240 hex

```



```
address
{

void CWDTestDlg::OnButtonRefresh()
{
    // TODO: Add your control notification handler code here
    if (gLibMyDLL == NULL)
    {
        MessageBox("WATCHDOG.DLL NOT LOADED.");
        return;
    }
    WDRefresh(0x240);    // Refresh Watchdog with 240 hex
address
}
```

## CHAPTER 6

### ACTIVE X CONTROL

#### 6.1 Methods

The watch dog library provides OCX functions to let user to use Active X Control to develop application program, the methods are shown in the belows.

##### 1. Long Disable ( )

###### Syntax

Long Disable()

###### Description

This is use to disable the watchdog countdown of the counter.

###### Usage

Long=Watchdog.disable()

##### 2. Long enable PCI Reset (bool status)

###### Syntax

Watchdog.EnablePCIReset(True)

###### Description

This option is use to specify whether the card will be affected by the PCI reset signal from the PC.

###### Usage

Long=Watchdog. EnablePCIReset(True)

### 3. Long EnableorLoad ( )

#### *Syntax*

Watchdog.EnableorLoad

#### *Description*

This is to enable or load the counter. This will start the countdown of the counter

#### *Usage*

Long= Watchdog.EnableorLoad ( )

### 4. Long Getcounter ( )

#### *Syntax*

Watchdog.GetCounter

#### *Description*

This will return the current value of the counter.

#### *Usage*

Long= Watchdog.GetCounter()

### 5. Long PowerOff ( )

#### *Syntax:*

WatchDog.PowerOFF

#### *Description:*

This will trigger the Power Remote Box kit to turn OFF.

#### *Usage:*

Long= WatchDog.PowerOFF()

### 6. Long PowerON ( )

#### *Syntax:*

Watchdog.PowerON

#### *Description*

This will trigger the Remote Power Box Kit to turn ON.

#### *Usage:*

Long= Watchdog.PowerON()

### 7. Long SetCounter (Long counter)

#### *Syntax:*

Watchdog.SetCounter(counter)

#### *Description:*

This is used to specify the counter.

#### *Usage:*

Long = SetCounter(long)

### 8. SetTimeUnit(long Timeunit)

#### *Syntax:*

Watchdog.SetTimeUnit(Long counter)

#### *Description*

This is used to specify the time Unit or the Frequency of the timer countdown.

#### *Usage:*

Long = Watchdog.SetTimeUnit(Long)

#### Example of Timer Countdown

0 = 1 / 2048

1 = 1 / 1024

2 = 1 / 512

3 = 1 / 256

4 = 1 / 128

5 = 1 / 64

6 = 1 / 32

7 = 1 / 16

8 = 1 / 8

9 = 1 / 4

10 = 1 / 2

#### 6.2 Sample Source Code

```
Private Sub cmdDisable_Click()  
Timer1.Enabled = False  
End Sub
```

```
Private Sub cmdEnable_Click()  
Watchdog.SetTimeUnit (Val(cboTime.Text)) 'Set the Time unit  
Watchdog.SetCounter (Val(txtCountDown.Text)) 'Set the  
counter  
Watchdog.EnableorLoad  
Timer1.Enabled = True  
End Sub
```

```
Private Sub cmdOff_Click()  
Watchdog.PowerOFF  
End Sub
```

```
Private Sub cmdOn_Click()  
Watchdog.PowerON  
End Sub
```

```
Private Sub Command1_Click()  
End  
End Sub
```

```
Private Sub Timer1_Timer()  
txtCountDown.Text = Watchdog.GetCounter  
If txtCountDown.Text = 0 Then  
txtCountDown.Text = 0  
Timer1.Enabled = False  
End If  
End Sub
```

## CHAPTER 7

### EXTENSION POWER CONTROL BOX

#### 7.1 Features

1. Miniature high power designed for mounting on P.C. Board.
2. High contact rating (10 - 30A), high shock / vibration resistance.
3. High reliability and long life.
4. High temperature design, "F" class + 155 degree C is available.

#### 7.2 Specification

##### COIL RATING

| RATE<br>VOLTAGE<br>(VDC) | COIL<br>RESISTANCE | RATED<br>CURRENT |
|--------------------------|--------------------|------------------|
|--------------------------|--------------------|------------------|

|     |       |     |
|-----|-------|-----|
| 5   | 27    | 185 |
| 6   | 40    | 93  |
| 12  | 155   | 77  |
| 18  | 380   | 47  |
| 24  | 660   | 36  |
| 46  | 2300  | 21  |
| 110 | 13400 |     |

#### CONTINUATION OF THE CHART:

| RATE<br>VOLTAGE<br>(VDC)               | RATE<br>VOLTAGE<br>(VDC) | MUST<br>DROPOUT<br>VOLTAGE | MAX.<br>VOLT | POWER<br>COMSUMPTION<br>(W) |
|--|--------------------------|----------------------------|--------------|-----------------------------|
| % OF RATE VOLTAGE (A1 + 20C )<br>(20C) |                          |                            |              |                             |

|     |        |        |         |             |
|-----|--------|--------|---------|-------------|
| 5   | 75 Max | 10 Min | 120 Max | 0.9 Approx. |
| 6   |        |        |         |             |
| 12  |        |        |         |             |
| 18  |        |        |         |             |
| 24  |        |        |         |             |
| 46  |        |        |         |             |
| 110 |        |        |         |             |

### 7.3 Characteristics

|                       |  |
|-----------------------|--|
| Contact Arrangement   | SPST(1 Form A), SPDT(1 Form 0)   |
| Contact Material      | AGODO  |
| Contact Resistance    | 50 m ohm Max   |
| Switching Voltage     | DC 125V Max. AC 250 V Max.   |
| Operate Time          | <= 10 ms.  |
| Release Time          | <= 8 ms.   |
| Insulation Resistance | 1000M ohm min (500V DC)  |
| Dielectric Strength   | 1400 VAC 60 Hz, 1 min. between open contact<br>2800 VAC 50 Hz, 1 min. between coil and contact. "H" type<br>2500 VAC |
| Shock Resistance      | 10g Approx   |
| Vibration Resistance  | 1.65 mm, excursions from 10 - 55Hz.<br>10 - 55 Hz  |
| Ambient Temperature   | Storage: -55°C to + 130°C Operating:<br>-55°C to + 85°C  |
| Humidity              | 220 to 85% R.H   |
| Operation Life        | Mechanical: 10 <sup>7</sup> Electrical: 10 <sup>5</sup> (at rated load)  |
| Weight                | 22 gr. Approx(Open Type) 28 gr. Approx(Sealed Type)  |

Note: Specifications are subject to change without notice

### UL/CSA RATING TYPE:

|                        | Form A                        | Form B          | Form C                          |                |
|------------------------|-------------------------------|-----------------|---------------------------------|----------------|
|                        | 50/60 Hz                      | 50/60 Hz        | 50/60 Hz                        |                |
|                        | NO                            | NC              | NO                              | NC             |
| Resistive              | 10A, 240V<br>AC               | 5A, 240 V<br>AC | 10A, 240V<br>AC                 | 5A, 240V<br>AC |
| Tungsten               | 5A, 240V<br>AC                | 3A, 240V<br>AC  | 5A, 240V<br>AC                  | 3A, 240V<br>AC |
| HP                     | 1HP, 125V AC, 2HP,<br>240V AC |                 | 1/2HP, 125V AC, 1HP,<br>240 VAC |                |
| Coil Rating: 5-120V DC |                               |                 |                                 |                |

|                          | Form A                             | Form B                        | Form C                     |  | Cycles |
|--------------------------|------------------------------------|-------------------------------|----------------------------|--|--------|
|                          | 50/60<br>Hz                        | 50/60 Hz                      | 50/60 Hz                   |  |        |
|                          | NO                                 | NC                            | NO                         | NC                                     |        |
| Resi<br>stive            | 30 A,<br>14<br>VDC /<br>240V<br>AC | 20 A,<br>14VDC /<br>240VAC    | 30A,<br>14VDC /<br>240 VAC | 30A,<br>14VDC<br>, 20A /<br>240VA<br>C | 100k   |
| HP                       | 1 HP/<br>16<br>FLA/<br>120V        | 30 LRA /<br>10FLA/12<br>0V 30 | 1hp/<br>16FLA/<br>120V     | 30LRA<br>/<br>10FLA/<br>120V           |        |
|                          | 2 HP/<br>12 FLA<br>240V            | LRA/<br>10FLA/24<br>0V        | 2hp/<br>12FLA/<br>240V     | 30LRA<br>/<br>10FLA/<br>240V           |        |
| Coil Ratings: 5 – 120VDC |                                    |                               |                            |  |        |

## APPENDIX A

### WARRANTY INFORMATION

#### A.1 Copyright

Copyright 1999, 2000 DECISION COMPUTER INTERNATIONAL CO., LTD. All rights reserved. No part of WATCH DOG software and manual may be reproduced, transmitted, transcribed, or translated into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without the prior written permission of DECISION COMPUTER INTERNATIONAL CO., LTD.

Each piece of WATCH DOG package permits user to use WATCH DOG only on a single computer, a registered user may use the program on a different computer, but may not use the program on more than one computer at the same time.

Corporate licensing agreements allow duplication and distribution of specific number of copies within the licensed institution. Duplication of multiple copies is not allowed except through execution of a licensing agreement. Welcome call for details.

#### A.2 Warranty Information

DECISION warrants that for a period of one year from the date of purchase (unless otherwise specified in the warranty card) that the goods supplied will perform according to the specifications defined in the user manual. Furthermore that the WATCH DOG product will be supplied free from defects in materials and workmanship and be fully functional under normal usage.

In the event of the failure of a WATCH DOG product within the specified warranty period, DECISION will, at its option, replace or repair the item at no additional charge. This limited warranty does not cover damage resulting from incorrect use, electrical interference, accident, or modification of the product.

All goods returned for warranty repair must have the serial number intact. Goods without serial numbers attached will not be covered by the warranty.

Transportation costs for goods returned must be paid by the purchaser. Repaired goods will be dispatched at the expense of WATCH DOG.

To ensure that your WATCH DOG product is covered by the warranty provisions, it is necessary that you return the Warranty card.

Under this Limited Warranty, DECISION's obligations will be limited to repair or replacement only, of goods found to be defective as specified above during the warranty period. DECISION is not liable to the purchaser for any damages or losses of any kind, through the use of, or inability to use, the WATCH DOG product.

DECISION reserves the right to determine what constitutes warranty repair or replacement.

**Return Authorization:** It is necessary that any returned goods are clearly marked with an RA number that has been issued by DECISION. Goods returned without this authorization will not be attended to.