

# *PC Watchdog Timer Card* CE

**PCIRWDT PCIXWDT**  
**Documentation Number PCIxWDT-1303**

*This product designed and manufactured in Ottawa, Illinois USA  
of domestic and imported parts by*

## **B & B Electronics**

### ***International Headquarters***

B&B Electronics Mfg. Co. Inc.  
707 Dayton Road -- P.O. Box 1040 -- Ottawa, IL 61350 USA  
Phone (815) 433-5100 -- General Fax (815) 433-5105  
**Home Page: [www.bb-elec.com](http://www.bb-elec.com)**  
Sales e-mail: [orders@bb-elec.com](mailto:orders@bb-elec.com) -- Fax (815) 433-5109  
Technical Support e-mail: [support@bb-elec.com](mailto:support@bb-elec.com) -- Fax (815) 433-5104

### ***European Headquarters***

B&B Electronics Ltd.  
Westlink Commercial Park, Oranmore, Co. Galway, Ireland  
Phone +353 91-792444 -- Fax +353 91-792445  
**Home Page: [www.bb-europe.com](http://www.bb-europe.com)**  
Sales e-mail: [orders@bb-europe.com](mailto:orders@bb-europe.com)  
Technical Support e-mail: [support@bb-europe.com](mailto:support@bb-europe.com)

© B&B Electronics – Revised March 2003

# Table of Contents

<b>CHAPTER 1: GENERAL INFORMATION.....</b>	<b>1</b>
INTRODUCTION.....	1
FEATURES .....	1
SPECIFICATIONS .....	1
<b>CHAPTER 2: SETUP AND INSTALLATION.....</b>	<b>2</b>
INSPECTION .....	2
INSTALLING THE HARDWARE.....	4
CONNECTING THE PCIRWDT MOTHERBOARD RESET OPTION .....	4
CONNECTING THE ATX RESET OPTION FOR THE PCIXWDT .....	5
INSTALLING THE SOFTWARE .....	6
<b>CHAPTER 3: OPERATION.....</b>	<b>7</b>
COMMUNICATING WITH THE WDT.....	7
PINOUTS .....	8
<b>CHAPTER 4: SOFTWARE .....</b>	<b>10</b>
THE APPLICATION PROGRAM INTERFACE .....	10
<i>Using the API with Borland C++ .....</i>	<i>10</i>
<i>Using the API with Visual C++ .....</i>	<i>11</i>
<i>Using the API with Visual BASIC .....</i>	<i>11</i>
COMMAND REFERENCE.....	11
<i>BBWatchdog_ReadRegister .....</i>	<i>11</i>
<i>BBWatchdog_Reset .....</i>	<i>12</i>
<i>BBWatchdog_Shutdown .....</i>	<i>13</i>
<i>BBWatchdog_Startup .....</i>	<i>13</i>
<b>CHAPTER 5: DISTRIBUTING WATCHDOG SOFTWARE.....</b>	<b>14</b>
<b>CHAPTER 6: SPECIAL FEATURES FOR WIN 2000 AND NT .....</b>	<b>19</b>
AUTO LOGON IN WINDOWS NT .....	19
AUTO LOGON IN WINDOWS 2000.....	22
SOFTWARE SHUTDOWN IN WINDOWS NT AND WINDOWS 2000.....	24
<b>APPENDIX A: DECLARATION OF CONFORMITY .....</b>	<b>A-1</b>

# Chapter 1: GENERAL INFORMATION

## Introduction

B&B Electronics' Watchdog Timer (WDT) Cards, Models PCIRWDT and PCIXWDT, are hardware devices designed to overcome the dangers or annoyances associated with a PC "locking up." These WDT's are designed for the PCI bus. Your software periodically resets the WDT. If the watchdog does not receive the reset trigger within a software selectable timeout period, the WDT resets the computer. The PCIRWDT uses the reset pins from the motherboard to reset the computer; where as, the PCIXWDT uses the ATX power supply to turn off the computer, then turn it back on 10 seconds later. A form C relay output is also provided on both models to reset an external device if desired. This relay may be set to energize when timeout is reached or during normal operations.

## Features

- No address or IRQ setup.
- Software selectable timeout period from 0.5 to 512 seconds.
- Software enable and disable.
- Form C relay output to control external device.
- Write back timeout verification.
- Reset flag for PCIRWDT.
- Non-splicing connection to motherboard reset pins for the PCIRWDT.
- Non-splicing connection to the ATX power supply for the PCIXWDT.
- Supported OS: Win 98, 2000, ME, NT

## Specifications

Dimensions:	2.5"H x 6.9"L (6.35 x 17.5 cm)
Bus:	PCI
Power Consumption:	1.5 W maximum
Output specs:	Form C relay output, maximum ratings 24 VDC at 1.0 A 115 VAC at 0.5 A
Connector:	DB-9 male

## Chapter 2: SETUP AND INSTALLATION

### Inspection

The WDT has been tested for proper operation. It should be in perfect electrical and mechanical condition upon receipt.

Remove the card from its protective packaging, practicing proper grounding techniques. Handle the card only by its edges being careful not to touch the gold connection fingers. Save the packaging for storage or shipping if the card needs repair.

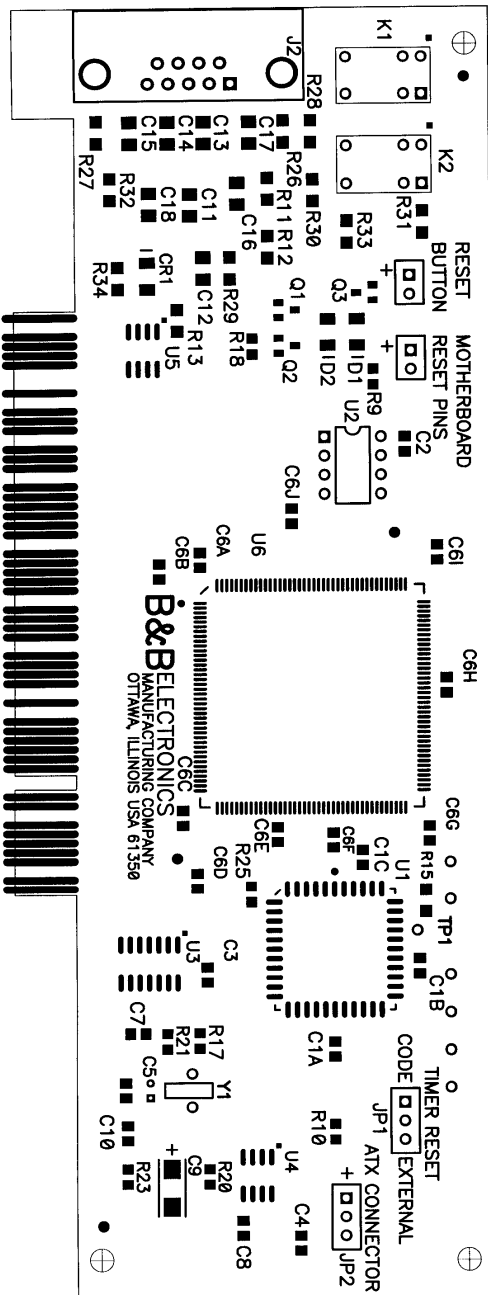


Figure 1. PCIRWDT/PCIXWDT PCB Silk Screen

## Installing the Hardware

***Before installing the new watchdog card in your computer, make sure that you are grounded.***

1. Turn the power to your computer off and unplug from the power outlet.
2. Remove the cover of the computer.
3. Pick an unused PCI slot.
4. Remove the expansion PCI slot cover. Save the mounting screw for installation of the WDT.
5. Install the WDT into the unused PCI slot. Be certain that the card is inserted completely into the slot.
6. Secure the card with the mounting screw.

## Connecting the PCIRWDT Motherboard Reset Option

To use the PCIRWDT to reboot the host computer, a connection must be made from the PCIRWDT to the motherboard. The jumper wire supplied with the PCIRWDT must be connected between the *MOTHERBOARD RESET PINS* jumper on the PCIRWDT and the RESET pins on the host motherboard. If your computer has a RESET switch, the switch's jumper wires should already be connected to the motherboard reset pins. Remove the switch wires from the motherboard pins and connect them to the *RESET BUTTON* jumper on the PCIRWDT. Next, connect the jumper wire to the motherboard-reset pins. The other end is connected to the *MOTHERBOARD RESET PINS* jumper on the PCIRWDT. These pins are simply in parallel with *RESET BUTTON* jumper and will allow the PCIRWDT or your RESET switch to reboot the host computer.

The polarity of the *RESET JMPR* is indicated on the PCIRWDT PCB, but in most cases, not on the motherboard or reset switch. If you are unsure of which way to connect the PCIRWDT to the motherboard, use the following procedure to verify the polarity of the connections.

1. Disconnect the reset switch wires from the motherboard.
2. Install the PCIRWDT in an empty PCI slot in your computer. Leave the computer cover off.
3. Connect the jumper wire provided with the PCIRWDT to the motherboard-reset pins. Leave the other end unconnected.
4. Power up the computer.
5. After the computer has booted, connect the other end of the jumper wire to *MOTHERBOARD RESET PINS* jumper of the PCIRWDT.
6. Remove the wires from *MOTHERBOARD RESET PINS* jumper.
7. If your computer does not reboot, the polarity was correct. If the computer did reboot, reverse the connection made to *MOTHERBOARD RESET PINS* jumper of the PCIRWDT.
8. Follow the same procedure to connect the RESET switch wires to *RESET BUTTON* jumper of the PCIRWDT.

**NOTE:** Use caution when operating the computer with the case open. Be sure to ground yourself by touching the metal case of the computer before you handle any components inside the machine.

## Connecting the ATX Reset Option for the PCIXWDT

The PCIXWDT uses the ATX power supply to reset the computer. The PCIXWDT sends a 5 volt, 10 second signal on the PS-ON data line to turn off the computer for 10 seconds. Then when this signal goes back to zero the computer turns back on. The ATX daughter board also allows the computer to bypass the ON/OFF button. Setting the jumper to **ON/OFF** will allow the ON/OFF button to work as normal. Setting the jumper to **ALWAYS ON** will bypass the ON/OFF button, leaving the computer always on.

1. Unplug the computer from the outlet socket.
2. Press the release tab on the connector to disconnect the power cable from the motherboard.
3. Set the Jumper to **ON/OFF** or **ALWAYS ON**.
4. Plug the 20-pin connector of the jumper wire into the motherboard's power connector.
5. Plug the 3-pin connector of the jumper wire into *ATX CONNECTOR* jumper of the PCIXWDT.
6. Plug the power cable into the header on the jumper wire.

## Installing the Software

The floppy disk supplied with the Watchdog card contains all the necessary software needed to install the card. After installing the card and rebooting the machine, place the software floppy disk in the disk drive.

**In a Windows NT machine**, after installing the hardware, restart the computer and logon as a user that has administrator rights, or ask your system administrator to install the software. After logging on, run the Setup.exe program from the NT40 folder located on the floppy disk provided. The Setup.exe program will direct the user through the software installation. After software installation, remove the floppy disk from the disk drive and restart the computer.

**In Windows 98, Windows ME, and Windows 2000 machines** the computer will automatically recognize that new hardware has been installed and will search for the appropriate drivers. With the disk in the floppy drive, follow the directions to install the software. After software installation, remove the floppy disk from the disk drive and restart the computer.

**Sample programs** can be copied onto your PC's hard drive or run from the floppy drive, and used to demonstrate the function of the watchdog time. The sample programs can be used on all of the above operating systems. Create a folder on your root drive and copy the "Samples" folder from the floppy drive into that folder. There are sample programs on the disk that are written in \Bc5 (Borland C++ 5.01), \Vb6 (Visual Basic 6.0), \Vc5 (Visual C++ 5.0).

## Chapter 3: OPERATION

### Communicating with the WDT

The WDT uses a four-bit latch to start and stop the timers and set the timeout period.

Regardless of the mode of operation of the WDT, a "1" must be written to the least significant bit of the latch to start the timers. Writing a "0" to the least significant bit of the latch at any time will stop the timers. Note that since the least significant bit is used to start and stop the counters, any even number written to the latch will result in the counters being turned off.

The remaining three bits of the latch are used to select one of eight timeout periods. Table 3 shows the possible timeouts.

**Table 3. WDT Timeouts**

<b>Binary Latch Value</b>	<b>Hex Latch Value</b>	<b>Approximate Timeout (seconds)</b>
<b>0001</b>	1	0.5
<b>0011</b>	3	2
<b>0101</b>	5	6
<b>0111</b>	7	14
<b>1001</b>	9	30
<b>1011</b>	B	126
<b>1101</b>	D	254
<b>1111</b>	F	510

The fifth bit is a "don't care" when you write to it. The sixth bit enables the relay. A <1> disables the relay, so the WDT has no control of the relay. A <0> allows the WDT to control the relay. The seventh bit sets the relays position when the card is running. A <0> will cause the relay to be normally discharged, energizing only when the timeout is reached. A <1> will cause the relay to be normally energized when the card is enabled and the timeout is not reached. The WDT will discharge when the timeout is reached, the card is disabled, or power is cut from the computer. The eighth bit is used to turn on and off an optional relay.

To use the WDT from your program, the latch value corresponding to the desired timeout and control bits must be periodically written to the latches. After the counters are started, they are reset every time the WDT is addressed. As long as the WDT is addressed before the end of the selected timeout period, no reset will occur. **Be certain that a <0> is written to the WDT whenever your program is terminated.**

The four-bit latch values are also readable. Reading these bits is recommended to verify the setup of the WDT. In the PCIRWDT the fifth bit is the reset flag. It will be a 1 when the PCIRWDT resets the computer and a 0 if the computer has not been reset by the PCIRWDT. This bit will be cleared if the computer is turned off or after the bit has been read.

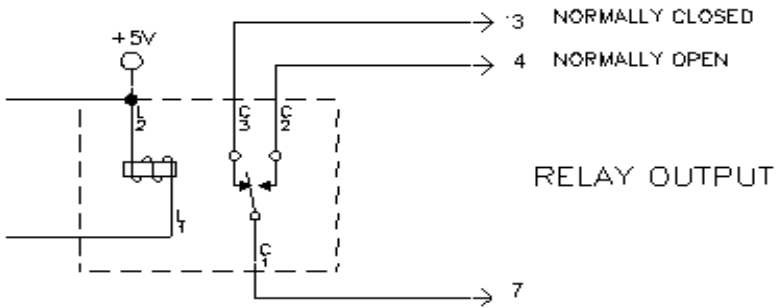
**Table 4. Registers**

Write Register	Read Register
L0	L0
L1	L1
L2	L2
L3	L3
X	Reset Flag for ATR X for ATX
Relay Enable	Relay Enable
Charged/Discharged	Charged/Discharged
Optional Relay	Optional Relay

X = Don't Care

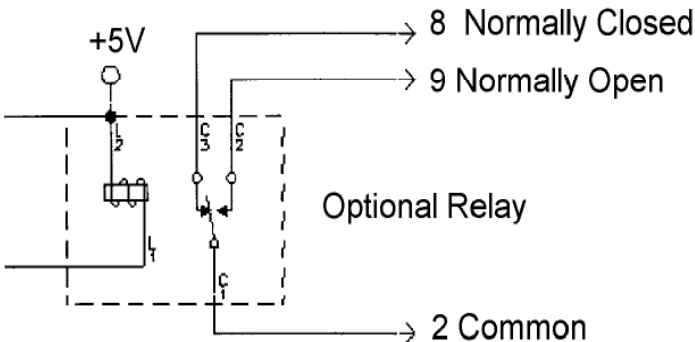
## Pinouts

Three pins of the male DB9 connector are used for the WDT. Pin seven is the common for the relay. Pins 4 and 7 are the external device reset relay contacts (Relay 1) and are non-polarized, normally open contacts. Pins 3 and 7 are for the non-polarized, normally closed contacts. Figure 3.1 is a schematic of the output portion of the WDT.



**Figure 3.1 Output Schematic**

If the WDT includes the optional relay (Relay 2), three more pins of the male DB-9 connector are used. Pin 2 is the common for the optional relay. Pins 2 and 9 are non-polarized, normally open contacts and pins 2 and 8 are non-polarized, normally closed contacts. Figure 3.2 is a schematic of the optional relay for WDT equipped with the optional relay.



**Figure 3.2 Output Schematic with Optional Relay**

## Chapter 4: Software

### The Application Program Interface

The application program interface (API) is a dynamic-link library (DLL) that provides a common set of functions to access the watchdog timer under, Windows 98, Windows NT, Windows 2000 and Windows ME. The API supports the PCIRWDT and the PCIXWDT. The function names in this version of the API are explained in the Command Reference section of this chapter.

#### Using the API with Borland C++

In the `\BC5` directory is an example of using the Watchdog Timer API with Borland C++ 5.01. To use the API, the import library, `WatchdogApi.lib`, must be added to the project file and the include file, `WatchdogApi.h`, must be in a `#include` statement in the applications source code.

## Using the API with Visual C++

In the `\VC5` directory is an example of using the Watchdog Timer API with Visual C++ 5.0. To use the API, the import library, `WatchdogApi.lib`, must be added to the project file and the include file, `Watchdog.h`, must be in a `#include` statement in the applications source code.

## Using the API with Visual BASIC

In the `\VB4`, `\VB5` and `\VB6` directories are examples of using the Watchdog Timer API with Visual BASIC. To use the API, the function definitions in `WatchdogApi.bas` must be included in the project.

## Command Reference

### BBWatchdog\_ReadRegister

---

Purpose:	Read the register value from the watchdog timer.
C/C++:	<pre>BOOL BBWatchdog_ReadRegister(HANDLE hDev, BYTE *pDataValue);</pre>
Visual BASIC:	<pre>Public Declare Function BBWatchdogReadRegister Lib "WatchdogApi.dll" Alias "BBWatchdog_ReadRegister" (ByVal hDev As Long, pDataValue As Integer) As Byte</pre>
Remarks:	<p><i>hDev</i> is the handle to the watchdog timer that was returned by <code>BBWatchdog_Startup</code>. <i>pDataValue</i> is a pointer to a variable to hold the register value.</p> <p>See <code>BBWatchdog_Reset</code> for the meaning of the data returned in <i>pDataValue</i>.</p>
Return Value:	TRUE (non-zero) if successful; otherwise, FALSE (zero).

## BBWatchdog\_Reset

**Purpose:** Reset the timeout of the watchdog timer.

**C/C++:** `BOOL BBWatchdog_Reset (HANDLE hDev, BYTE bResetValue);`

**Visual BASIC:** `Public Declare Function BBWatchdogReset Lib "WatchdogApi.dll" Alias "BBWatchdog_Reset" (ByVal hDev As Long, ByVal bResetValue As Integer) As Byte`

**Remarks:** *hDev* is the handle to the watchdog timer that was returned by `BBWatchdog_Startup`.

*bResetValue* is the reset value that is sent to the watchdog timer. The values are:

Bit	Hex	Decimal	Description
7	80h	128	Optional external relay. 0 = off, 1 = on
6	40h	64	0 = turn relay off until timeout. 1 = turn relay on until timeout.
5	20h	32	0 = turn on externals 1 = turn off externals
4	10h	16	Timeout occurred.
3	08h	8	Timeout duration. (See Table 4.1)
2	04h	4	
1	02h	2	
0	01h	1	0 = Disable Timer 1 = Enable Timer

**Return Value:** TRUE (non-zero) if successful; otherwise, FALSE (zero).

## BBWatchdog\_Shutdown

---

Purpose:	Terminate communications with the watchdog timer.
C/C++:	<code>BOOL BBWatchdog_Shutdown(HANDLE hDev);</code>
Visual BASIC:	<code>Public Declare Function BBWatchdogShutdown Lib "WatchdogApi.dll" Alias "BBWatchdog_Shutdown" (ByVal hDev As Long) As Byte</code>
Remarks:	<i>hDev</i> is the handle to the watchdog timer that was returned by <code>BBWatchdog_Startup</code> .
Return Value:	TRUE (non-zero) if successful; otherwise, FALSE (zero).

## BBWatchdog\_Startup

---

Purpose:	Begin communications with the watchdog timer.
C/C++:	<code>HANDLE BBWatchdog_Startup(UINT nID);</code>
Visual BASIC:	<code>Public Declare Function BBWatchdogStartup Lib "WatchdogApi.dll" Alias "BBWatchdog_Startup" (ByVal nID As Integer) As Long</code>
Remarks:	<p>This function sets up communications with the watchdog device identified by <i>nID</i>. Under normal operation, the value of <i>nID</i> should be one.</p> <p>A value of zero for <i>nID</i> is a special value that tells the API to attempt to find an ISA Watchdog Timer using the old device driver. If it cannot find an ISA Watchdog Timer, then the API changes the value of <i>nID</i> to one and attempts to find a PCI Watchdog Timer.</p>
Return Value:	On error, <code>INVALID_HANDLE_VALUE</code> is returned; otherwise, a handle to the watchdog timer is returned.

## Chapter 5: Distributing the Watchdog Software

If the watchdog timer is distributed with other software, the following files must be included with it.

### For Windows NT:

C:\WinNT\System32

- WatchdogApplet.cpl
- WatchdogApi.dll
- WatchdogService.exe

C:\WinNT\System32\Drivers

- BBWatchdog.sys

In addition to installing these files, the installation program must add the appropriate registry keys and values to tell Windows NT or Windows 2000 that the device driver and service exist. Below are the keys and values that must be added under `HKEY_LOCAL_MACHINE \ System \ CurrentControlSet \ Services`. After these modifications are made to the registry, Windows NT or Windows 2000 must be restarted for the changes to take effect.

BBWatchdog

- REG\_DWORD ErrorControl = 1
- REG\_SZ Group = "Extended Base"
- REG\_DWORD Start = 2
- REG\_DWORD Type = 1

## BBWatchdogService

- REG\_SZ DisplayName = "B&B Electronics' Watchdog Service"
- REG\_DWORD ErrorControl = 1
- REG\_SZ ImagePath = "%SystemRoot%\System32\WatchdogService.exe"
- REG\_SZ ObjectName = "LocalSystem"
- REG\_DWORD Start = 2
- REG\_DWORD Type = 16

## BBWatchdogService\Parameters

- REG\_SZ DeviceName = "\\.\WATCHDOG1"
- REG\_DWORD NtShutdownDelayValue = 11

## EventLog\System\BBWatchdog

- REG\_SZ ErrorMessageFile = "%SystemRoot%\System32\Drivers\BBWatchdog.sys"
- REG\_DWORD TypesSupported = 7

## For Windows 2000 / 98 / ME:

A:\

- BBWatchdog.exe
- Colnstaller.dll
- Pciwdt.dll
- Pciwdt.inf
- Pciwdt.sys
- WatchdogAPI.dll
- WatchdogApplet.cpl

These files should be copied to a folder and shipped with any programs distributed along with the Watchdog Timer card.

### Automatic Logon in Windows NT

A common way to have applications start up automatically is to put them into the StartUp folder; however, these applications are only started after a user logs on. Only specially written applications, known as services, are allowed to run before a user logs on. For information about writing Windows NT services, see the Win32 SDK documentation and the sample program included with the Win32 SDK under the directory \MSTools\Samples\Win32\WinNT\Service. In order for an application program in the StartUp folder to start up automatically, automatic logon must be set up. Windows NT does not provide an easy way to enable automatic logon, but it can be setup by using the registry editor, REGEDT32.EXE.

1. Logon to Windows NT 4.0 as a user that is a member of the Administrator group. Start the User Manager (Programs | Administrative Tools (Common) | User Manager under the Start Button). Create a new user by choosing New User from the User menu. Setup a user named “AutoLogon” and uncheck the option, “User Must Change Password at Next Logon” and check the options “User Cannot Change Password” and “Password Never Expires”. When you are finished, the dialog box should look like Figure 6.1. For this example, we will leave the password blank.

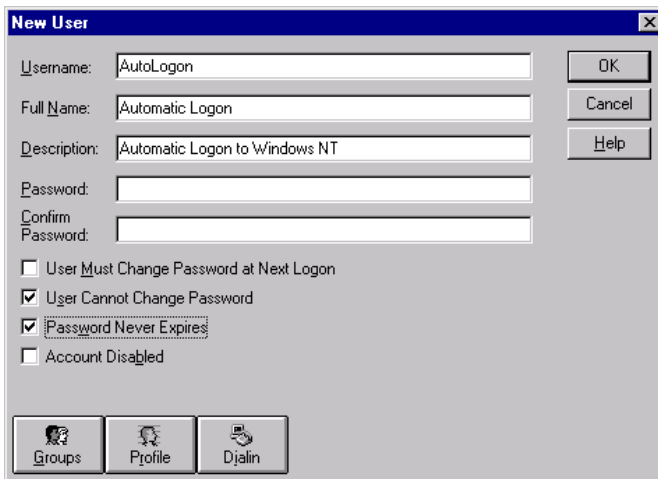
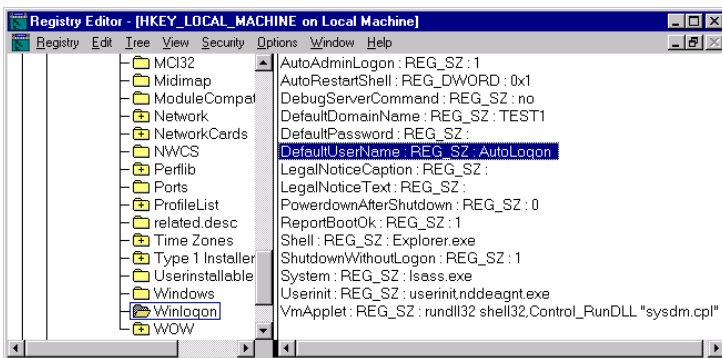


Figure 6.1

Note that assigning access rights to users and groups is beyond the scope of this document, but for security reasons, the AutoLogon account should only have the minimum access rights needed to run your application. Anyone with physical access to the computer will have the access rights for the automatic logon account. For this example, we will leave the new account as a member of the users group.

2. Test the new account by logging off and logging on as the user AutoLogon. While logged on, add your application to the StartUp Folder (Settings | Taskbar | Start Menu Programs | Add under the Start Button). Logoff and logon again as AutoLogon to verify that your application starts up automatically.
3. Logoff and logon a user that is a member of the Administrator Group. Start the registry editor (select Run from the Start Button and type "REGEDT32" for the application to open). Be careful when using the registry editor, because changes in the registry effect the setup of Windows NT. Incorrect changes may make Windows NT unusable.
4. Select the "HKEY\_LOCAL\_MACHINE on Local Machine" window. Before making any changes, save the current values of HKEY\_LOCAL\_MACHINE by choosing Registry | Save Subkey As. Then open the following keys: HKEY\_LOCAL\_MACHINE | Software | Microsoft | Windows NT | Current Version | Winlogon (see Figure 6.2). On the right-hand side of the window, double-click on the value, DefaultUserName, and change it to "AutoLogon". The value, DefaultDomainName, should be set to the correct value; however, if it is not, double-click on it and change it to the correct domain name.



**Figure 6.2**

5. Windows NT must know the password for the AutoLogon account in order to use it. Added a value to the Winlogon key, by

choosing `Add Value` from the Edit menu. Type "DefaultPassword" for the name of the value. The data type for DefaultPassword is REG\_SZ. For this example, leave the password blank; however, if a password was assigned to the AutoLogon user when it was created in step 1, type that password here.

6. Now Windows NT must be informed that automatically logon should be used. Add another value to the Winlogon key. Type "AutoAdminLogon" for the name of the value and the data type is REG\_SZ. Type "1" for the value of the string. (Note: do not include quotes around the 1.)
7. The final step in creating the AutoLogon account is to tell Windows NT not to update the values in the Winlogon key. Normally, the values of DefaultUserName and DefaultDomainName are updated when a user logs on. In order to keep NT from changing these values, access rights must be changed. Select Winlogon on the left-hand side of the window. Choose `Permissions` from Security menu. Select each group or user shown and change the type of access to "Read". This will give everyone read-only access to the key. If the values of the Winlogon key need to be changed in the future, change the type of access for the Administrators group back to "Full Access", but remember to change it back to "Read" when you are finished making changes.
8. Now restart Windows NT to use the automatic logon account. If you want to logon as a different user, shutdown with the "Close all programs and log on as a different user" option selected and **hold the SHIFT key down during the entire logoff process**; otherwise, it will use the automatic logon account to automatically logon.

## Automatic Logon in Windows 2000

This describes how to set the automatic logon feature for Windows 2000 Professional. Note that this option is not available for Windows 2000 Server or Advanced Server.

To prevent the password prompt in a non-domain system:

- In Control Panel, double-click **Users and Passwords**.
- Click to clear **Users must enter a user name and password to use this computer check box**.

Note: This option does not appear on a system that is a member of a domain.

- Click the Advanced tab.
- Click to clear the Require users to press Ctrl-Alt-Del before logging on check box.

Automatic logon is not supported when you are logging on to a domain. You will need to join a workgroup if you want the automatic logon feature.

This can be turned on in a Windows 2000 Professional domain system by editing the registry.

**Warning:** This procedure will allow the user to log on automatically to the domain. This could cause a serious security problem. Anyone booting this system with this registry modification will be logged on to the system with the user's security credentials.

1. Start Regedit32.exe and locate the following registry key:

HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows  
NT\CurrentVersion\Winlogon

2. Establish your domain name, account name, and password, using the values you would normally type when logging on. You should assign the following values: DefaultDomainName, DefaultUserName, and DefaultPassword.

**NOTE:** The DefaultDomainName and DefaultUserName values may already exist. The DefaultPassword value may not. If it does not, create it:

- a. On the Edit menu, click Add Value.
- b. In the Value Name box, type DefaultPassword.
- c. In the Data Type box, click REG\_SZ.
- d. Click OK
- e. In the String Editor box, type your password.
- f. Click OK and save your changes.

**NOTE:** If no DefaultPassword string is specified, Windows automatically changes the value of the AutoAdminLogon key from 1 (true) to 0 (false). This disables the AutoAdminLogon feature.

3. On the **Edit** menu, click **Add Value**. Type **AutoAdminLogon** in the **Value Name** box. Click **REG\_SZ** in the **Data Type** box. Enter 1 in the **String** box. Save your changes.
4. Quit Regedt32.exe.
5. Shut down Windows and turn off the computer.
6. Restart your computer and Windows 2000. You should be able to log on automatically now.

**NOTE:** To bypass the AutoAdminLogon process and logon as a different user, hold down the SHIFT key after a logoff or after a windows restart.

This is a by-design security feature. The resolution involves editing the registry. Editing the registry is not supported.

## Software Shutdown in Windows NT and 2000

Windows NT and Windows 2000 are protected-mode operating systems and I/O instructions, such as IN and OUT, are privileged instructions. These instructions can only be accessed from within a device driver. The watchdog device driver, BBWATCHDOG.SYS, starts running when Windows boots. It detects the watchdog timer hardware on the PCI bus and configures it for use in Windows. NOTE: the watchdog device driver does not attempt to detect the ISA version of the watchdog timer hardware; it only detects the PCI version.

WATCHDOGAPI.DLL, the watchdog application program interface (API), is a group of functions for use in application programs. These functions provide an easy way of communicating with the watchdog device driver. Descriptions of these functions are in the Application Program Interface Section of this manual.

The watchdog service, WATCHDOGSERVICE.EXE, and the watchdog **Control Panel** applet, WATCHDOGAPPLET.CPL, are new additions to this release of the watchdog software. The service communicates with the watchdog device driver and allows Windows to have a chance to flush the disk cache and shutdown properly before the watchdog timer hardware resets the computer. The watchdog **Control Panel** applet configures the watchdog service. It can be started by double-clicking the watchdog timer icon (see Figure 6.3) in the **Control Panel**.



**Figure 6.3: Watchdog Timer Icon**

When the applet starts, the dialog box in Figure 6.4 is displayed. This dialog box allows enabling or disabling of the **software shutdown** used by the watchdog service. If software shutdown is disabled, the service will not attempt to shutdown Windows before the watchdog hardware reboots the computer. When Windows is not shutdown properly, the data in the write-behind disk cache will be lost.



**Figure 6.4**

### **Operation in Windows NT and Windows 2000.**

When software shutdown is enabled, the device driver will use these parameters to extend the hardware shutdown and allow Windows NT and Windows 2000 time to shutdown normally. If Windows does not shutdown within this extended amount of time, the watchdog timer hardware will reboot the computer.

Whether software shutdown is enabled or disabled, it is recommended that all hard drives be formatted as NTFS (NT File System) volumes, rather than FAT (File Allocation Table) volumes, because NTFS provides better protection from disk corruption. Note that, if all hard drives are formatted as NTFS, they cannot be accessed from DOS, Windows 95 or Windows 98 by using NT's multiboot option.

To determine when to attempt to shutdown Windows, the watchdog service issues a shutdown notification request to the device driver. When the device driver determines that a hardware timeout will occur within 1 second, it writes the extended timeout value and relay states, which were specified in the watchdog control panel applet, to the watchdog timer hardware. Then the device driver notifies the **service** that it wants to shutdown Windows. In response to this notification, the service displays a warning message for 20 seconds, allowing the user a chance to save their work. If the extended timeout value is less than 20 seconds, the warning message is not displayed and Windows begins the shutdown procedure. When the extended timeout duration has expired, the watchdog timer hardware reboots the computer. Table 6.1 shows the hardware and software timeout values and durations.

Latch Value			Timeout (Seconds)	
Binary	Decimal	Hexadecimal	Software	Hardware
0001	1	01	-	½
0011	3	03	1	2
0101	5	05	5	6
0111	7	07	13	14
1001	9	09	29	30
1011	11	0B	125	126
1101	13	0D	253	254
1111	15	0F	509	510

**Table 6.1: Hardware and Software Timeouts Values**

# Appendix A: Declaration of Conformity Statement

## DECLARATION OF CONFORMITY

Manufacturer's Name: B&B Electronics Manufacturing Company  
Manufacturer's Address: P.O. Box 1040  
707 Dayton Road  
Ottawa, IL 61350 USA  
Model Numbers: PCIRWDT, PCIXWDT  
Description: PC Watchdog Timer for Windows NT  
Type: Light industrial ITE equipment  
Application of Council Directive: 89/336/EEC  
Standards: EN55022  
EN50082-1:  
EN61000 (-4-2, -4-3, -4-4, -4-6)



William H. Franklin III, Director of Engineering