

congatec Application Note



| | |
|---------------------------------|---|
| Applicable Products | All congatec modules |
| Application Note Subject | congatec Watchdog features and implementation |
| Document Name | AN3_Watchdog |
| Usage Designation | External |

Application Note 3

Revision 1.4

Revision History

| Revision | Date (dd.mm.yy) | Author | Changes |
|----------|-----------------|--------|--|
| 0.1 | 09.05.05 | EHA | Document created |
| 1.0 | 25.06.06 | HCH | Initial Release |
| 1.1 | 26.10.05 | GDA | Added note about S3 (save to Ram) to Notes and Cautions section 6. |
| 1.2 | 18.01.06 | GDA | Added information about the timeout values. Added information to section 5.3.1. Added note about Watchdog ACPI Event to section 6. |
| 1.3 | 25.08.06 | HCH | Changed File name to match data base requirements |
| 1.4 | 22.09.06 | GDA | Removed sections 5.3 and 5.3.1. |

Preface

This application note describes the Watchdog feature found on congatec AG modules. It explains the full feature set, however some items may not be applicable for systems which do not support all Watchdog features.

Disclaimer

The information contained within this Application Note, including but not limited to any product specification, is subject to change without notice.

congatec AG provides no warranty with regard to this Application Note or any other information contained herein and hereby expressly disclaims any implied warranties of merchantability or fitness for any particular purpose with regard to any of the foregoing. congatec AG assumes no liability for any damages incurred directly or indirectly from any technical or typographical errors or omissions contained herein or for discrepancies between the product and the Application Note. In no event shall congatec AG be liable for any incidental, consequential, special, or exemplary damages, whether based on tort, contract or otherwise, arising out of or in connection with this Application Note or any other information contained herein or the use thereof.

Intended Audience

This Application Note is intended for technically qualified personnel. It is not intended for general audiences.

Symbols

The following symbols may be used in this Application Note:



Warning

Warnings indicate conditions that, if not observed, can cause personal injury.



Caution

Cautions warn the user about how to prevent damage to hardware or loss of data.



Note

Notes call attention to important information that should be observed.

Copyright Notice

Copyright © 2005, congatec AG. All rights reserved. All text, pictures and graphics are protected by copyrights. No copying is permitted without written permission from congatec AG.

congatec AG has made every attempt to ensure that the information in this document is accurate yet the information contained within is supplied "as-is".

Trademarks

Intel and Pentium are registered trademarks of Intel Corporation. Expresscard is a registered trademark of Personal Computer Memory Card International Association (PCMCIA). PCI Express is a registered trademark of Peripheral Component Interconnect Special Interest Group (PCI-SIG). I²C is a registered trademark of Philips Corporation. CompactFlash is a registered trademark of CompactFlash Association. Winbond is a registered trademark of Winbond Electronics Corp. AVR is a registered trademark of Atmel Corporation. ETX is a registered trademark of Kontron AG. AMICORE8 is a registered trademark of American Megatrends Inc. XpressROM is a registered trademark of Insyde Technology, Inc. Microsoft®, Windows®, Windows NT®, Windows CE and Windows XP® are registered trademarks of Microsoft Corporation. VxWorks is a registered trademark of WindRiver. conga, congatec and XTX are registered trademarks of congatec AG. All product names and logos are property of their owners.

Contents

| | |
|---|----|
| 1 Watchdog Terminology Definitions..... | 6 |
| 1.1 Watchdog..... | 6 |
| 1.2 Watchdog Timeout..... | 6 |
| 1.3 Watchdog Event..... | 6 |
| 1.4 Watchdog Trigger..... | 6 |
| 1.5 BIOS Power On Self Test (POST)..... | 6 |
| 1.6 Runtime..... | 6 |
| 1.7 CMOS Setup Utility..... | 6 |
| 1.8 CGOS API..... | 6 |
| 2 Runtime Watchdog..... | 7 |
| 2.1 Runtime Watchdog Modes..... | 7 |
| 2.1.1 Single Event Mode..... | 7 |
| 2.1.2 Repeated Event Mode..... | 7 |
| 2.1.3 Single Trigger Mode..... | 7 |
| 2.1.4 Watchdog Timing Diagram..... | 8 |
| 2.2 Watchdog Events..... | 9 |
| 2.2.1 NMI / IRQ..... | 9 |
| 2.2.2 ACPI..... | 9 |
| 2.2.3 Reset..... | 9 |
| 2.2.4 Power Button..... | 9 |
| 3 Post Watchdog..... | 10 |
| 4 Watchdog Configuration, Initialization, and Lifetime..... | 11 |
| 4.1 Watchdog Configuration via the CMOS Setup Utility..... | 11 |
| 4.2 Watchdog Configuration via the CGOS API..... | 11 |
| 4.3 Initialization and Lifetime..... | 11 |
| 5 Watchdog Triggering..... | 12 |
| 5.1 Watchdog Triggering via the CGOS API..... | 12 |
| 5.2 External Trigger Method..... | 12 |
| 6 Notes and Cautions..... | 13 |

1 Watchdog Terminology Definitions

The following section defines the terminology used within this document.

1.1 Watchdog

A *Watchdog* is a combination of system means that support automatic recovery from an error condition such as deadlocks and system hangups.

1.2 Watchdog Timeout

A *Watchdog Timeout* defines the time period after which the *Watchdog* generates a *Watchdog Event* if there is no longer a response from the system.

1.3 Watchdog Event

If there is no system response within a defined time period, then the *Watchdog* generates a *Watchdog Event*. Usually this is a hardware signal such as a non-maskable interrupt or a reset signal.

1.4 Watchdog Trigger

This is the system response that forces the *Watchdog* to reload its timeout counter, i.e. triggering the *Watchdog* prevents a *Watchdog Event*.

1.5 BIOS Power On Self Test (POST)

This is the amount of time needed for system initialization between power-up and the start of the loading of the operating system.

1.6 Runtime

The phase of normal system operation starting with the loading of the operating system, i.e. after the POST has finished.

1.7 CMOS Setup Utility

This is the system configuration tool built into the BIOS. It controls the CMOS RAM used to save the system configuration.

1.8 CGOS API

This is the abbreviation for the congatec 32-bit uniform operating system application programming interface.

2 Runtime Watchdog

The *Runtime Watchdog* is available during normal system operation and is used to recover from malfunctioning operating systems, application software or system expansions like add-in hardware or peripheral devices. It supports up to three stages. For every stage a separate timeout value and event type can be specified. The granularity of the timeout values is one millisecond and the watchdog timer may have a maximum deviation of 2%.

2.1 Runtime Watchdog Modes

The *Watchdog Mode* defines what happens when the *Watchdog* generates the event of the last defined stage. When there are several stages defined, then the *Watchdog* switches to the next stage after generating an event. The selected *Watchdog Mode* defines how the *Watchdog* behaves after it has generated the last defined event. Below is a list of the possible modes.

2.1.1 Single Event Mode

In the *Single Event Mode*, the *Watchdog* switches off after generating the event of the last defined stage.

2.1.2 Repeated Event Mode

When in the *Repeated Event Mode*, and after generating the event of the last defined stage, the *Watchdog* stays in the last stage and restarts the timeout counter.

2.1.3 Single Trigger Mode

The *Single Trigger Mode* is a variant of the *Single Event Mode*. It also switches off after generating the event of the last defined stage. Additionally, it switches off when it gets triggered the first time.

2.1.4 Watchdog Timing Diagram

Figure 1: single stage / single event mode

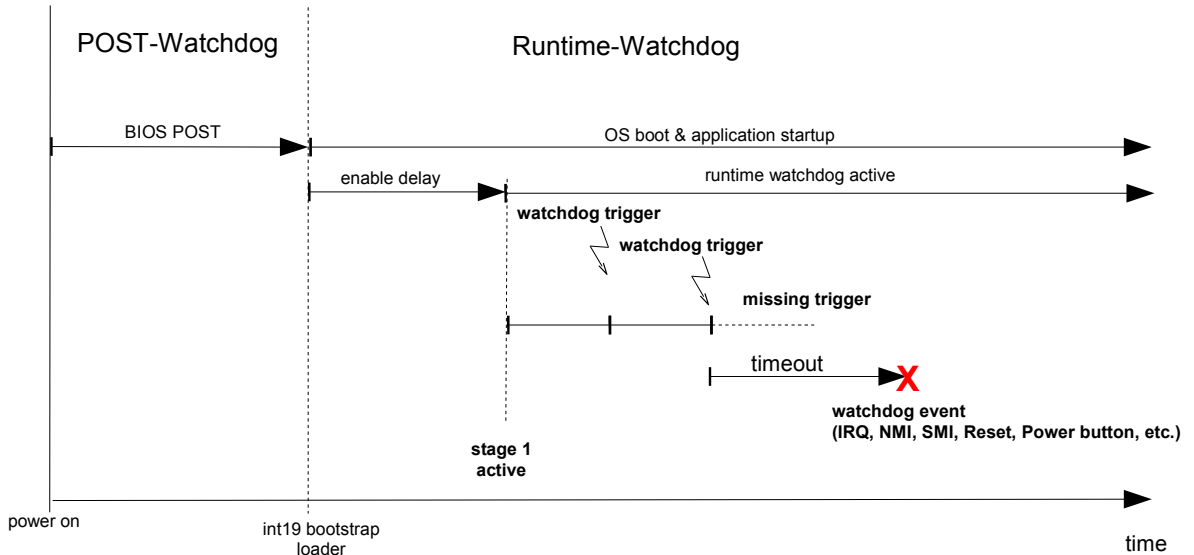
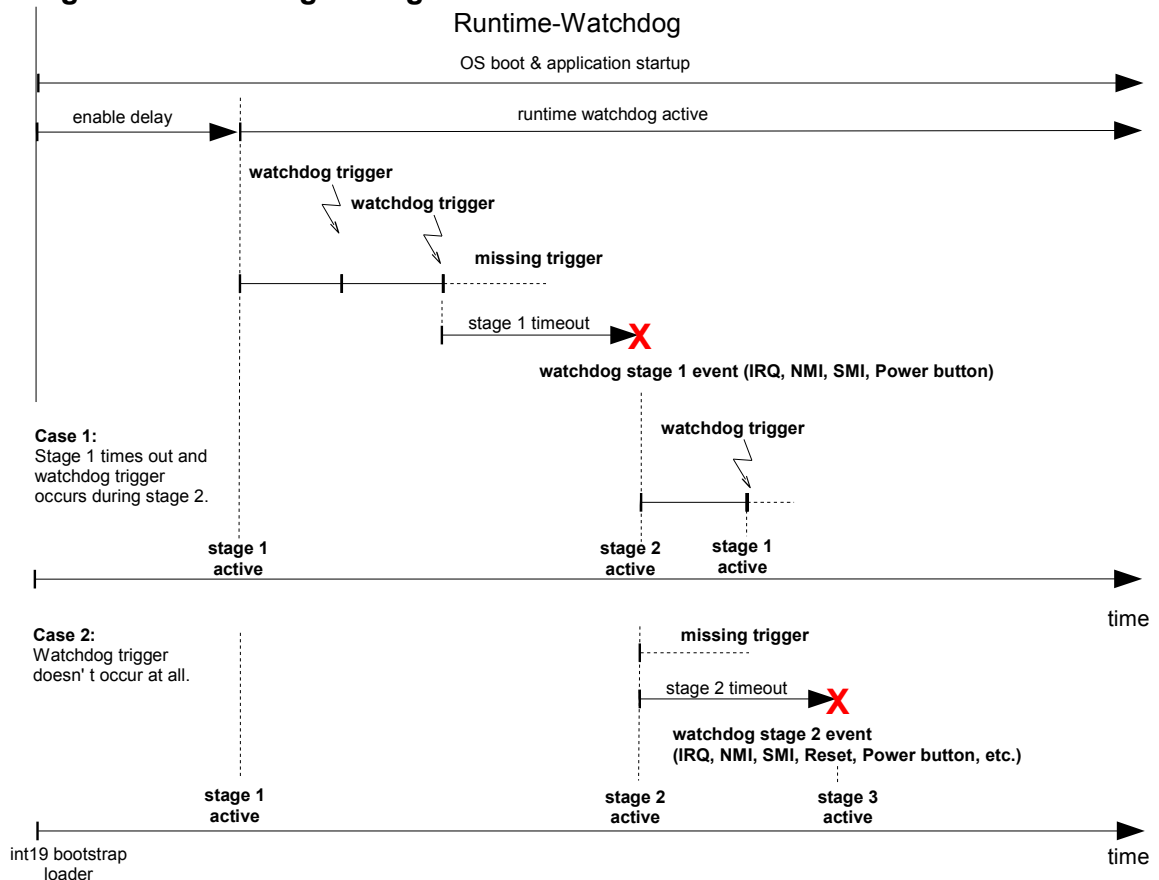


Figure 2: multi stage / single event mode



2.2 Watchdog Events

The following is a description of possible *Watchdog Events*.

2.2.1 NMI / IRQ

This *Watchdog Event* generates an interrupt. Depending on the system implementation this may be a non-maskable interrupt (NMI) or a normal interrupt request (IRQ).

2.2.2 ACPI

This *Watchdog Event* generates a system management interrupt. Depending on the system management implementation (ACPI/APM) this may be an SCI or an SMI. See section 6 “Notes and Cautions” for more information about the ACPI Event.

2.2.3 Reset

This *Watchdog Event* generates a reset signal. Depending on the system implementation this may reset the whole system, part of it, or just the CPU. In any case after generating the reset signal the *Runtime Watchdog* gets switched off and no further *Watchdog* stages will be processed.

2.2.4 Power Button

This *Watchdog Event* generates a power button signal. Depending on the system implementation this can invoke a system shutdown, switch off the system or power up the system.

3 Post Watchdog

The *POST Watchdog* is available during the system initialization process and is used to recover from a malfunction of system expansions like add-in hardware or peripheral devices. If enabled the *POST Watchdog* is started immediately after system power up and automatically switched off when the POST is finished and the system is ready to load the operating system. If the system does not finish the POST within the time period defined by the *POST Watchdog* timeout, then the *Watchdog* generates a reset signal to reboot the system. The granularity of the timeout value is one millisecond and the watchdog timer may have a maximum deviation of 2%.

4 Watchdog Configuration, Initialization, and Lifetime

4.1 Watchdog Configuration via the CMOS Setup Utility

The setup program for the BIOS provides a CMOS setup screen that is used to configure the *Watchdog*. Any changes done in the CMOS setup screen will take effect as soon as the new values have been saved to CMOS RAM and the system is restarted.



See the BIOS setup program description of your particular congatec AG module for a detailed list of the options.

4.2 Watchdog Configuration via the CGOS API

congatec AG modules provide the CGOS API to configure and initialize the *Runtime Watchdog*. Changing the parameters via the CGOS API will take effect immediately. Please keep in mind that any *Runtime Watchdog* configuration done via the CGOS API will be overwritten by the *Watchdog* parameters that have been set using the BIOS setup program when the system reboots.



See the CGOS API documentation (*CGOSAPIm1x.pdf*), which can be found on the congatec website at www.congatec.com, for a more detailed description.

4.3 Initialization and Lifetime

If the *POST Watchdog* is enabled, then it is initialized and started every time the system powers up or reboots. It stays active until the system reaches the end of POST. The *POST Watchdog* is switched off automatically before the system starts loading the operating system. Additionally, the *POST* watchdog is switched off automatically when invoking the CMOS setup utility or when entering a BIOS boot menu.

If the *Runtime Watchdog* is enabled via the CMOS setup utility, then it is initialized and started automatically at the end of POST. Additionally, it can be initialized and started at any time during runtime via the CGOS API. Except for when in the *Single Trigger Mode*, the *Runtime Watchdog* stays active as long as it gets triggered and the system continues to run. The *Watchdog* can be switched off at any time during runtime via the CGOS API. The *Watchdog* switches off automatically when being triggered in single trigger mode or after generating a *RESET EVENT*.

5 Watchdog Triggering

Triggering the *Watchdog* within the *Watchdog Timeout* interval prevents the *Watchdog* from generating an event. When there are several *Watchdog* stages defined, then triggering the *Watchdog* also forces the *Watchdog* back to the first stage. There are different methods of triggering the *Watchdog*. Below you will find a description of each trigger method.

5.1 Watchdog Triggering via the CGOS API

The usual method of triggering the watchdog is through the use of the CGOS API.

 **Note**

See the *CGOS API documentation (CGOSAPIm1x.pdf)*, which can be found on the *congatec website at www.congatec.com*, for a more detailed description.

5.2 External Trigger Method

Some congatec AG modules support triggering the *Watchdog* via an external trigger signal called WDTRIG.

 **Note**

See the *user's guide of your particular congatec AG module* for details about which connector the *WDTRG* is located on, pin number and signal requirements.

6 Notes and Cautions

Note

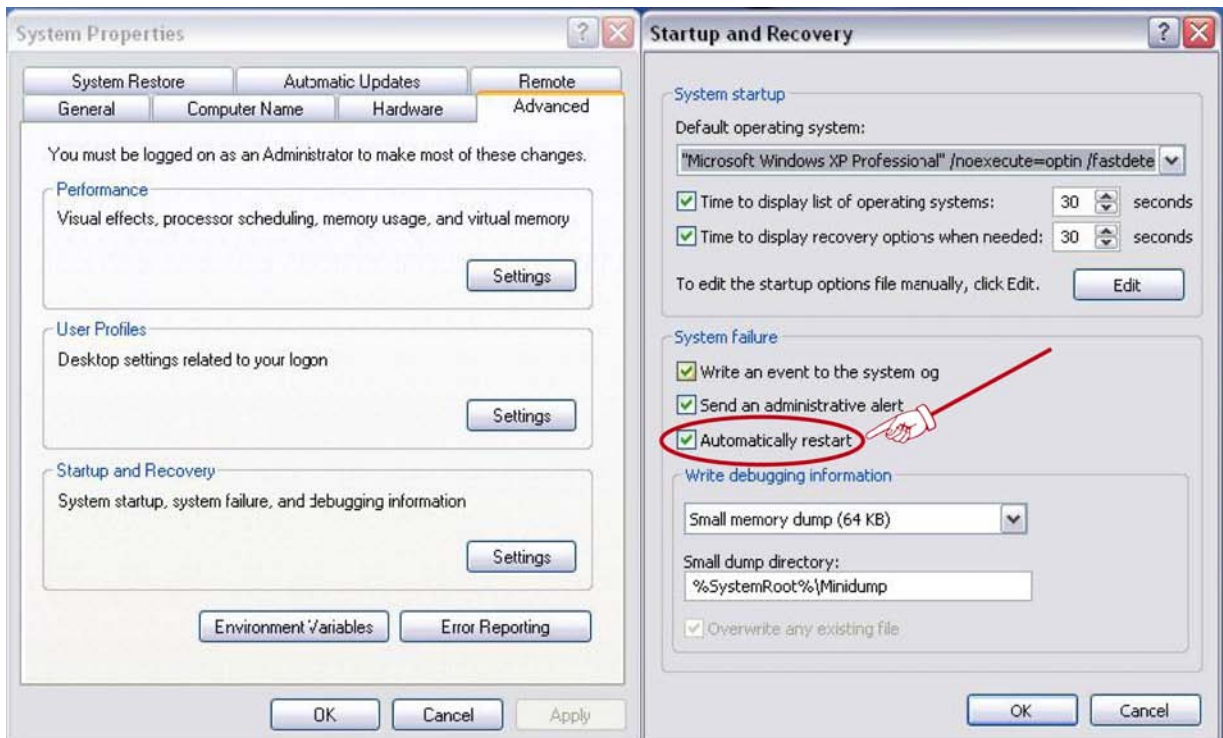
In ACPI mode it is not possible for a "Watchdog ACPI Event" handler to directly restart or shutdown the OS. For this reason the congatec BIOS will do one of the following:

For Shutdown: An over temperature notification is executed. This causes the OS to shut down in an orderly fashion.

For Restart: An ACPI fatal error is reported to the OS.

It depends on your particular OS as to how this reported fatal error will be handled when the Restart function is selected. If you are using Windows XP/2000 there is a setting that can be enabled to ensure that the OS will perform a restart when a fatal error is detected. After a very brief blue-screen the system will restart.

You can enable this setting by going to the "System Properties" dialog box and choosing the "Advanced" tab. Once there choose the "Settings" button for the "Startup and Recovery" section. This will open the "Startup and Recovery" dialog box. In this dialog box under "System failure" there are three check boxes that define what Windows will do when a fatal error has been detected. In order to ensure that the system restarts after a 'Watchdog ACPI Event' that is set to 'Restart', you must make sure that the check box for the selection "Automatically restart" has been checked. If this option is not selected then Windows will remain at a blue-screen after a 'Watchdog ACPI Event' that has been configured for 'Restart' has been generated. The following is a Windows screen-shot showing the proper configuration.



 **Note**

By using several Watchdog stages it is possible to escalate the Watchdog actions. For example the Watchdog could generate an interrupt as a first event giving some interrupt handler of the application the chance to recover from an error condition. If this handler also fails to trigger the Watchdog, then the Watchdog may generate a reset signal to restart the system.

**Caution**

Be careful when selecting a POST Watchdog timeout value. It should be taken into account that the power up time of peripheral devices may vary or option ROMs, such as LAN boot ROMs, may elongate the POST process. Choosing a POST Watchdog timeout value that is too short may be counterproductive. Instead of ensuring that only a recovery from a true malfunction is implemented, the system may reset periodically without a valid reason as a result of an incorrect Watchdog Timeout value.

 **Note**

It doesn't make any sense to select Watchdog Event RESET together with Repeated Event Mode because the Watchdog switches off immediately after generating the first reset signal due to the fact that a repeated reset signal is not supported.

 **Note**

Under normal circumstances it is not necessary to trigger the POST Watchdog. However it is possible. This may be helpful when writing option ROMs, which need to delay the POST in special situations. The CGOS API is not available in that case, therefore the fast or external trigger methods can only be used.

 **Note**

It's possible that two Watchdog stages with Power Button Events could be used to configure defined system on/off times.

 **Note**

The Single Trigger Event may be useful for application software, which cannot use the CGOS API but still want to ensure that the operating system boots completely and starts the application code. In that case the CMOS setup utility must be used to configure the Runtime Watchdog in Single Trigger Mode with one stage and event RESET. Together with a POST watchdog this guarantees that the system is restarted until it makes it to the application code. The only thing that the application code has to do then is to switch off the watchdog via the fast or external trigger method.

 **Note**

Some congatec AG modules support the power state S3 (suspend to RAM). On these particular modules the watchdog configuration is not saved when entering the S3 state nor is it restored when resuming from the S3 state. Generally the watchdog is switched off after resuming from the S3 state. Applications which allow the S3 power state but also require watchdog operation must reinitialize the watchdog when the system resumes from S3. For this it may be necessary to register with the S3 resume notification of the operating system. A similar problem arises with operating systems supporting the power state S4 (suspend to disk) also known as hibernation mode. However, in this case when resuming from the S4 state the runtime watchdog will continue to operate using the parameters that were defined with the CMOS setup utility. Any re-configurations done by an application before entering the S4 state are lost. Again the application must register with the resume notification of the operating system if this behavior is not acceptable.