Watchdog HowTo

Using a watchdog on Fujitsu's OEM mainboards

Rainer König

Watchdog HowTo
Using a watchdog on Fujitsu's OEM mainboards

Author Rainer König *Rainer.Koenig@ts.fujitsu.com*

This document explains how to use the watchdog features of current Fujitsu mainboards with various Linux distributions.

Preface

1. Document Conventions

This manual uses several conventions to highlight certain words and phrases and draw attention to specific pieces of information.

1.1. Typographic Conventions

Four typographic conventions are used to call attention to specific words and phrases. These conventions, and the circumstances they apply to, are as follows.

`Mono-spaced Bold`

Used to highlight system input, including shell commands, file names and paths. Also used to highlight keys and key combinations. For example:

> To see the contents of the file **`my_next_bestselling_novel`** in your current working directory, enter the **`cat my_next_bestselling_novel`** command at the shell prompt and press **`Enter`** to execute the command.

The above includes a file name, a shell command and a key, all presented in mono-spaced bold and all distinguishable thanks to context.

Key combinations can be distinguished from an individual key by the plus sign that connects each part of a key combination. For example:

> Press **`Enter`** to execute the command.
>
> Press **`Ctrl`**+**`Alt`**+**`F2`** to switch to a virtual terminal.

The first example highlights a particular key to press. The second example highlights a key combination: a set of three keys pressed simultaneously.

If source code is discussed, class names, methods, functions, variable names and returned values mentioned within a paragraph will be presented as above, in **`mono-spaced bold`**. For example:

> File-related classes include **`filesystem`** for file systems, **`file`** for files, and **`dir`** for directories. Each class has its own associated set of permissions.

Proportional Bold

This denotes words or phrases encountered on a system, including application names; dialog-box text; labeled buttons; check-box and radio-button labels; menu titles and submenu titles. For example:

> Choose **System** → **Preferences** → **Mouse** from the main menu bar to launch **Mouse Preferences**. In the **Buttons** tab, select the **Left-handed mouse** check box and click **Close** to switch the primary mouse button from the left to the right (making the mouse suitable for use in the left hand).
>
> To insert a special character into a **gedit** file, choose **Applications** → **Accessories** → **Character Map** from the main menu bar. Next, choose **Search** → **Find…** from the **Character Map** menu bar, type the name of the character in the **Search** field and click **Next**. The character you sought will be highlighted in the **Character Table**. Double-click this highlighted character to place it in the **Text to copy** field and then

> click the **Copy** button. Now switch back to your document and choose **Edit** → **Paste** from the **gedit** menu bar.

The above text includes application names; system-wide menu names and items; application-specific menu names; and buttons and text found within a GUI interface, all presented in proportional bold and all distinguishable by context.

Mono-spaced Bold Italic or ***Proportional Bold Italic***

Whether mono-spaced bold or proportional bold, the addition of italics indicates replaceable or variable text. Italics denotes text you do not input literally or displayed text that changes depending on circumstance. For example:

> To connect to a remote machine using ssh, type **ssh *username@domain.name*** at a shell prompt. If the remote machine is **example.com** and your username on that machine is john, type **ssh john@example.com**.
>
> The **mount -o remount *file-system*** command remounts the named file system. For example, to remount the **/home** file system, the command is **mount -o remount /home**.
>
> To see the version of a currently installed package, use the **rpm -q *package*** command. It will return a result as follows: ***package-version-release***.

Note the words in bold italics above: username, domain.name, file-system, package, version and release. Each word is a placeholder, either for text you enter when issuing a command or for text displayed by the system.

Aside from standard usage for presenting the title of a work, italics denotes the first use of a new and important term. For example:

> Publican is a *DocBook* publishing system.

1.2. Pull-quote Conventions

Terminal output and source code listings are set off visually from the surrounding text.

Output sent to a terminal is set in **mono-spaced roman** and presented thus:

```
books        Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests  Desktop1  downloads      images  notes  scripts  svgs
```

Source-code listings are also set in **mono-spaced roman** but add syntax highlighting.

1.3. Notes and Warnings

Finally, we use three visual styles to draw attention to information that might otherwise be overlooked.

Note

> Notes are tips, shortcuts or alternative approaches to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.

Important

Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring a box labeled "Important" will not cause data loss but may cause irritation and frustration.

Warning

Warnings should not be ignored. Ignoring warnings will most likely cause data loss.

2. We Need Feedback!

If you find a typographical error in this manual, or if you have thought of a way to make this manual better, we would love to hear from you!

If you have a suggestion for improving the documentation, try to be as specific as possible when describing it. If you have found an error, please include the section number and some of the surrounding text so we can find it easily.

Introduction

1.1. General information

Fujitsu industrial mainboards provide a watchdog device that reboots the system in case that the operating system crashes.

This HowTo describes the steps to setup the watchdog and how to use it.

1.2. Required prerequisites and skills

To do the configuration steps described in this document you need *root access* for your Linux operating system.

Important

All shell commands must be executed with root privileges.

If your distribution kernel does not provide the required device driver module then you need to compile it by yourself. Therefore you need to know how to compile kernel modules for your distribution kernel. This also requires that your system has the complete toolchain for building kernel modules installed.

You need to know how to locate and install packages for your Linux distribution.

What is a watchdog?

In general a watchdog is a hardware device that is a sort of kitchen timer. You can define a period of time and after that time is counted down to zero the watchdog triggers an event. On the Fujitsu industrial mainboards that event will result in a reset of the system.

On Fujitsu's industrial mainboards that offer a watchdog device the timer of the watchdog is defined in the BIOS setup. The range is from 1 minute up to 255 minutes.

The operating system needs to provide a process that frequently resets that watchdog. Every reset of the watchdog means that the timer is restarted. So, as long as that process is running the watchdog will never be counted down to zero. If the system crashes, then the process will no longer restart the watchdog, it will countdown to zero and then the system will reset and restart.

Modern systems can have several kind of watchdogs, e.g. modern CPUs provide watchdog devices for every CPU core that will trigger a NMI interrupt when the watchdog goes off. Those watchdogs are handled by the Linux kernel already, so this document focuses on the specific watchdog provided by the ASIC on the specific mainboard.

Fujitsu mainboard overview

This table shows you which mainboards you can use and what monitoring ASIC devices are used on those mainboards.

Table 3.1. Fujitsu mainboards

Board	ASIC	Comment
D3003-Sx	SCH5627	eOntario chipset
D307x-Sx	Theseus	CougarPoint chipset
D323x-Sx	Theseus	LynxPoint chipset
D3243-Sx	Theseus	LynxPoint chipset
D3313-Sx	SCH5627	eKabini chipset
D3348-Ax	Teutates	Grantley chipset
D3348-Bx	Teutates	Grantley chipset
D3358-Ax	Teutates	Grantley chipset
D3402-Ax/Bx	Teutates	Skylake chipset
D3417-Ax	Teutates	Skylake chipset
D3423-Ax	Teutates	Skylake chipset
D3427-Ax	Teutates	Skylake chipset
D3432-Ax	Teutates	Skylake chipset
D3433-Sx	Teutates	Skylake chipset
D3441-Sx	Teutates	Skylake chipset
D3445-Sx	Teutates	Skylake chipset
D3446-Sx	Teutates	Skylake chipset

From this table you can see what ASIC is used for system monitoring.

Kernel drivers details

4.1. Southbridge drivers

The most common driver for the SMBus interface of the southbridge is i2c_i801 which comes with the kernel.

You can check if your hardware is supported with the follwoing steps:

First execute an 'lspci' and search for the string 'smb':

```
# lspci | grep -i smb
00:1f.3 SMBus: Intel Corporation 82801JI (ICH10 Family) SMBus Controller
```

Then you need to look a bit closer at this device by using the bus adress as a parameter:

```
# lspci -s 1f.3 -vvv
00:1f.3 SMBus: Intel Corporation 82801JI (ICH10 Family) SMBus Controller
 Subsystem: Fujitsu Technology Solutions Device 114d
 Control: I/O+ Mem+ BusMaster- SpecCycle- MemWINV- VGASnoop- ParErr+ ...
 Status: Cap- 66MHz- UDF- FastB2B+ ParErr- DEVSEL=medium >TAbort- ...
 Interrupt: pin B routed to IRQ 17
 Region 0: Memory at f7307000 (64-bit, non-prefetchable) [size=256]
 Region 4: I/O ports at 1c00 [size=32]
 Kernel driver in use: i801_smbus
```

As you see in the last line of this example in this case the driver that is used is called i801_smbus.

When you have an output for 'Kernel driver in use' then you're halfway done. There is still a chance, that your driver doesn't work because newer systems assign also an ACPI node to that SMBus interface and then you might get a resource conflict between the PCI resources required by the driver and the resources reserved by ACPI.

To find out if you're affected by such an issue have a look at the dmesg output and search for "conflicts".

```
    # dmesg | grep conflicts
```

If you see a line that looks like

> ACPI Warning: SystemIO range 0x000000000000F040-0x000000000000F05F conflicts with OpRegion 0x000000000000F040-0x000000000000F04F (_SB_.PCI0.SBUS.SMBI) (20150619/utaddress-254)

then you are affected by this problem. In that case you need the following boot parameter as a work around: *acpi_enforce_resources=lax*

See the explanation of this boot parameter:

```
acpi_enforce_resources= [ACPI]
   { strict | lax | no }
   Check for resource conflicts between native drivers
```

```
and ACPI OperationRegions (SystemIO and SystemMemory
only). IO ports and memory declared in ACPI might be
used by the ACPI subsystem in arbitrary AML code and
can interfere with legacy drivers.
strict (default): access to resources claimed by ACPI
is denied; legacy drivers trying to access reserved
resources will fail to bind to device using them.
lax: access to resources claimed by ACPI is allowed;
legacy drivers trying to access reserved resources
will bind successfully but a warning message is logged.
no: ACPI OperationRegions are not marked as reserved,
no further checks are performed.
```

If you use lax, then the kernel is still warning, but your PCI driver is allowed to use the resources and you can access the SMBus.

4.2. ASIC drivers

The Fujitsu mainboards use different ASICs for system monitoring as you can see in the *mainboard table*.

4.2.1. SCH5627 driver

The SCH5627 chip uses the following kernel modules:

- sch5627.ko
- sch56xx-common.ko

The driver was first introduced into the 3.1 kernel.

If you don't know the kernel version of your distribution then you can execute the following command:

```
# uname -r
```

If the version that you get from the above command is lower than 3.1 then you should see if the driver is maybe backported to your older distribution kernel. Execute the following command:

```
# find /lib/modules/`uname -r` -name "sch5627.ko"
```

If the search is successful you will probably get the location of the driver module under **`/kernel/drivers/hwmon/5627.ko`**. If you don't get a result with the search above, then you need to compile the driver from the sources.

The sources for the driver are provided on the SCH56XX homepage (see *Section 6.3, "Homepage for the SCH5627 & SCH5636 (Theseus) driver"*).

4.2.2. Theseus driver

The Theseus chip uses the following kernel modules:

- sch5636.ko
- sch56xx-common.ko

The driver was first introduced into the 3.1 kernel.

If you don't know the kernel version of your distribution then you can execute the following command:

```
# uname -r
```

If the version that you get from the above command is lower than 3.1 then you should see if the driver is maybe backported to your older distribution kernel. Execute the following command:

```
# find /lib/modules/`uname -r` -name "sch5636.ko"
```

If the search is successful you will probably get the location of the driver module under **`/kernel/drivers/hwmon/5636.ko`**. If you don't get a result with the search above, then you need to compile the driver from the sources.

The sources for the driver are provided on the SCH56XX homepage (see *Section 6.3, “Homepage for the SCH5627 & SCH5636 (Theseus) driver”*).

4.2.3. Teutates driver

The Theseus chip uses the following kernel modules:

- ftsteutates.ko

The ftsteutaes driver is not yet (as of April 2016) part of the mainstream kernel.

The sources for ftsteutates are available from Fujitsu via FTP (see *Section 6.2, “Download adress for the ftsteutates driver”*). If you have obtained the sources you need to build the kernel module using the standard procedure for external kernel modules.

Using the watchdog

5.1. Installing the driver and software

First step is to check if the required kernel module is loaded. The following is what you get on boards with the Theseus chip. On boards with the SCH5627 chip the driver listed will be "sch5627".

```
linux-o2ln:~ # lsmod
sch5636                 17416  0
sch56xx_common          14462  1 sch5636
```

Important

You should add the driver sch5636 (or sch5627)to the file **/etc/modules** or add a configuration file in **/etc/modules-load.d/** to make sure that the driver is loaded when the system boots.

Now you need to install the watchdog package. Some Linux distributions like Debian provide binary packages, others don't. If your distribution doesn't provide this package you need to compile it yourself from the source package.

Important

At this point you should **reboot** your system, because so you make sure that you will find the correct watchdog device with the next step. If you don't reboot then there is a risk, that watchdog devices are enumerated differently after the next boot.

After the reboot you can see what watchdog devices your system offers:

```
# >ls -l /dev/watch*
crw------- 1 root root  10, 130 Apr 22 08:52 /dev/watchdog
crw------- 1 root root 250,   0 Apr 22 08:52 /dev/watchdog0
```

Important

On systems with the SCH5627 chip the watchdog device will not be registered until the watchdog is enabled in the BIOS setup. So for those boards it is necessary to setup the watchdog in the BIOS first (set it to a reasonable time, e.g. 60 minutes) and then do all the configuration steps described. If you don't see your watchdog then check the dmesg output for lines like "sch56xx_common: Watchdog not enabled by BIOS, not registering".

In the screenshot above you see that we get 2 files (**`/dev/watchdog`** and **`/dev/watchdog0`** listed. So now the important question is: What is the watchdog device associated with our ASIC driver? The answer is in the major number of the driver. The SCH5627 or SCH5636 ASIC drivers register themselves as a "misc" device which has the major number 10. So in our case **`/dev/watchdog`** is the correct device node for our ASIC's watchdog.

If your system uses the Teutates ASIC then its a bit different. The ftsteutates driver registers a watchdog device. If you look at all the watchdog devices under **`/dev/`** you might get something like that:

```
# ls -l /dev/watchdog*
crw-------. 1 root root  10, 130 Apr 26 08:54 /dev/watchdog
crw-------. 1 root root 251,   0 Apr 26 08:54 /dev/watchdog0
crw-------. 1 root root 251,   1 Apr 26 08:54 /dev/watchdog1
crw-------. 1 root root 251,   2 Apr 26 08:54 /dev/watchdog2
```

So the question is, which of the 4 devices is mapped to your watchdog. Lets try **`/dev/watchdog0`** for an example. We see a major number of 251 and a minor number of 0. If we look at the **`/sys/dev/char/`** directory we see those major:minor combination again.

```
# ls /sys/dev/char/25
250:0/ 250:1/ 251:0/ 251:1/ 251:2/ 252:0/
```

Note: We pressed the **TAB** key after the 25 to see the possible completions.

Next step is to change into the device directory inside the **`/sys/`** tree.

```
# cd /sys/dev/char/251\:0/device/
# ls
cpu_throttling  fan5_input   in3_input       temp14_fault  temp5_input
driver          fan5_source  misc            temp14_input  temp6_fault
fan1_fault      fan6_fault   modalias        temp15_fault  temp6_input
fan1_input      fan6_input   name            temp15_input  temp7_fault
fan1_source     fan6_source  overall_uptime  temp16_fault  temp7_input
fan2_fault      fan7_fault   power           temp16_input  temp8_fault
fan2_input      fan7_input   subsystem       temp1_fault   temp8_input
fan2_source     fan7_source  temp10_fault    temp1_input   temp9_fault
fan3_fault      fan8_fault   temp10_input    temp2_fault   temp9_input
fan3_input      fan8_input   temp11_fault    temp2_input   uevent
fan3_source     fan8_source  temp11_input    temp3_fault   watchdog
fan4_fault      hwmon        temp12_fault    temp3_input
fan4_input      in0_input    temp12_input    temp4_fault
fan4_source     in1_input    temp13_fault    temp4_input
fan5_fault      in2_input    temp13_input    temp5_fault
# cat name
ftsteutates
```

The **`cat name`** command showed, that the driver name for this device is indeed ftsteutates so we were lucky and found our device node on the first try.

Now you have to modify the configuration file for the watchdog which is stored at **`/etc/watchdog.conf`**. Below there is a listing of an example configuration file.

```
#ping   = 172.31.14.1
```

```
#ping   = 172.26.1.255
#interface  = eth0
#file   = /var/log/messages
#change   = 1407

# Uncomment to enable test. Setting one of these values to '0' disables it.
# These values will hopefully never reboot your machine during normal use
# (if your machine is really hung, the loadavg will go much higher than 25)
#max-load-1  = 24
#max-load-5  = 18
#max-load-15  = 12

# Note that this is the number of pages!
# To get the real size, check how large the pagesize is on your machine.
#min-memory  = 1

#repair-binary  = /usr/sbin/repair
#repair-timeout  =
#test-binary  =
#test-timeout  =

watchdog-device = /dev/watchdog

# Defaults compiled into the binary
#temperature-device =
#max-temperature = 120

# Defaults compiled into the binary
#admin   = root
#interval  = 1
#logtick                = 1
#log-dir  = /var/log/watchdog

# This greatly decreases the chance that watchdog won't be scheduled before
# your machine is really loaded
realtime  = yes
priority  = 1

# Check if rsyslogd is still running by enabling the following line
#pidfile  = /var/run/rsyslogd.pid
```

The watchdog device is configured in the line that defines "watchdog-device" by setting **`/dev/watchdog`** as the watchdog device. On some distributions you just have to uncomment this line because its there but preceded by a "#".

After you have defined the watchdog in **`/etc/watchdog.conf`** you can check if the watchdog identifies itself as the sch56xx watchdog.

The watchdog packages provides an utiltiy that is called **`wd_identify`**. Execute this command to find out if the device configured in the config file matches the sch56xx watchdog:

```
# wd_identify
sch56xx watchdog
```

If **`wd_identify`** confirms that you have configured the correct watchdog device then you can now go on with setting up the process that frequently resets the watchdog.

Warning

During our tests we encountered the situation that the Teutates watchdog registered as **`/dev/watchdog2`** directly after the first **`modprobe ftsteutates`**. On the next reboot the device was registered as **`/dev/watchdog0`**, probably because the kernel enumerated the watchdog devices differently. So schedule one reboot and check if your watchdog moved to another device node after the reboot.

The watchdog package usually comes with 2 daemons:

- watchdog: A fully fledged watchdog daemon that not only periodically updates your ASIC's watchdog but also checks for the other conditions defined in **`/etc/watchdog.conf`** (which is not part of this HowTo).
- wd_keepalive: A very basic daemon whose only purpose is to periodically update your ASIC's watchdog. This daemon conflicts with thewatchdog daemon, so its not started until the watchdog daemon is stopped.

Now restart your system and check if one of the daemons described above is started automatically (some distributions start it automatically, some don't). If not you should either

- add a line in **`/etc/init.d/boot.local`** to start it during boot.
- see if a startup script for the daemon exists in **`/etc/init.d/`** (on Linux systems that still use the System V boot method) and add enable this script (e.g. with the **`chkconfig`** command).
- enable the watchdog service with **`systemctl`** on Linux systems that use the systemd boot method.

Now you are ready to start using the watchdog.

5.2. Using the watchdog

After the correct watchdog device is configured in **`/etc/watchdog.conf`** and the daemon that keeps it periodically updated is started as a daemon on system boot you can configure a watchdog timeout in the BIOS setup.

Reboot your system and enter the BIOS setup by pressing the **F2** key. Then choose **Advanced** → **System monitoring** → **Watchdog timeout** to enter a timeout (in minutes) after which the system is rebooted automatically. The default value is **`0`** which means that the watchdog timer is disabled. Enter a reasonable amount of minutes here.

Important

Be careful not to chose to small values for the watchdog timer. Keep in mind that sometimes your system might do maintanance jobs during bootup before the watchdog daemon started from the init scripts (e.g. a filesystem check that might take some time. If your timeout value is too low you risk that the system reboots during that maintanance work.

Now press **ESC** and leave the BIOS setup with **`Save changes & reset`**.

5.3. Trying out the watchdog

Now your system should boot and run. No reboots should occur as the **`wd_keepalive`** process is alive and frequently resetting the watchdog timer.

Important

Do the test only if your system can be rebooted without data loss, so don't have any unsafed files or other users logged into your system.

If you want to test the watchdog function just execute the following commands to force a kernel panic:

```
# echo 1 > /proc/sys/kernel/sysrq
# echo c > /proc/sysrq-trigger
```

Executing the 2 commands above will force a kernel panic, so you will see a log screen of the kernel panic and the system is locked up.

Now just wait until your system reboots. It will happen because there is no longer any alive process which resets the watchdog.

Online resources

6.1. Actual version of this HowTo

The actual version of this HowTo can be downlaoded from *ftp://ftp.ts.fujitsu.com/tobedefined*

6.2. Download adress for the ftsteutates driver

As long as the ftsteutaes driver is not part of the Linux kernel it can be downloaded from *ftp://ftp.ts.fujitsu.com/tobedefined*

6.3. Homepage for the SCH5627 & SCH5636 (Theseus) driver

The sources for the driver are provided at *https://fedorapeople.org/~jwrdegoede/sch56xx/*.

6.4. Watchdog package home page

http://watchdog.sourceforge.net/

Appendix A. Revision History

Revision 0.9-0 2014-01-17 Rainer König

First draft

Revision 1.0-0 2014-07-31 Rainer König

Updates for D3313-Sx board

Revision 1.1-0 2016-04-22 Rainer König

Added preface with conventions and chapter for southbridge and ASIC drivers. Udated the usage chapter to reflect modern distributions that use systemd.

Revision 1.1-1 2016-04-25 Rainer König

Added warning for the ftsteutates watchdog device node which can change after installing the module and rebooting.