

# **Watchdog on Fujitsu mainboards**

---

Copyright © 2014 Fujitsu

---

**COLLABORATORS**

	<i>TITLE :</i> Watchdog on Fujitsu mainboards		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Rainer König	July 31, 2014	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME
0.9	2014-01-17	First draft	Rainer König
1.0	2014-07-31	Updates for D3313-Sx board	Rainer König

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	General information . . . . .	1
1.2	Required prerequisites and skills . . . . .	1
<b>2</b>	<b>What is a watchdog?</b>	<b>2</b>
<b>3</b>	<b>Fujitsu OEM mainboard overview</b>	<b>3</b>
<b>4</b>	<b>Using the watchdog</b>	<b>4</b>
4.1	Installing the driver and software . . . . .	4
4.2	Using the watchdog . . . . .	6
4.3	Trying out the watchdog . . . . .	6
<b>5</b>	<b>Online resources</b>	<b>7</b>
5.1	Watchdog package home page . . . . .	7

---

# List of Tables

3.1	Fujitsu mainboards . . . . .	3
-----	------------------------------	---

## **Abstract**

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

# Chapter 1

## 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.

---

## Chapter 2

# 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 ASCI on the specific mainboard.

---



## Chapter 3

# Fujitsu OEM mainboard overview

This table shows you which mainboards provide a hardware watchdog and which kernel driver is needed to use it.

<b>Board</b>	<b>ASIC</b>	<b>Driver</b>
D3076-Sx	Theseus	sch5636
D3236-Sx	Theseus	sch5636
D3313-Sx	SCH5627	sch5627

Table 3.1: Fujitsu mainboards

From this table you can see what ASIC is used for the watchdog device.

## Chapter 4

# Using the watchdog

### 4.1 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".

```
1 linux-o2ln:~ # lsmod
2 sch5636                17416  0
3 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:

```
1 linux-o2ln:~ # ls /dev/watchdog*
2 /dev/watchdog /dev/watchdog0 /dev/watchdog1 /dev/watchdog2
```



#### 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".

---

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.

```
1 #ping      = 172.31.14.1
2 #ping      = 172.26.1.255
3 #interface  = eth0
4 #file       = /var/log/messages
5 #change     = 1407
6
7 # Uncomment to enable test. Setting one of these values to '0' disables it.
8 # These values will hopefully never reboot your machine during normal use
9 # (if your machine is really hung, the loadavg will go much higher than 25)
10 #max-load-1  = 24
11 #max-load-5  = 18
12 #max-load-15 = 12
13
14 # Note that this is the number of pages!
15 # To get the real size, check how large the pagesize is on your machine.
16 #min-memory  = 1
17
18 #repair-binary  = /usr/sbin/repair
19 #repair-timeout =
20 #test-binary    =
21 #test-timeout  =
22
23 watchdog-device = /dev/watchdog1
24
25 # Defaults compiled into the binary
26 #temperature-device =
27 #max-temperature  = 120
28
29 # Defaults compiled into the binary
30 #admin            = root
31 #interval         = 1
32 #logtick          = 1
33 #log-dir          = /var/log/watchdog
34
35 # This greatly decreases the chance that watchdog won't be scheduled before
36 # your machine is really loaded
37 realtime         = yes
38 priority         = 1
39
40 # Check if rsyslogd is still running by enabling the following line
41 #pidfile         = /var/run/rsyslogd.pid
```

The watchdog device is configured in line 23 by setting `/dev/watchdog1` as the watchdog device. This value is just a guess, your hardware watchdog can be any of the watchdog devices that are listed above. So we have to find out if this device really is the sch56xx watchdog.

The watchdog packages provides an utility that is called **wd\_identify**. Execute this command to find out if the device configured in the config file matches the sch56xx watchdog:

```
1 linux-o2ln:/etc # wd_identify
2 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.

This program is called **wd\_keeplive** and is part of the watchdog package.

Now restart your system and check if **wd\_keeplive** is started automatically (some distributions start it automatically, some don't). If not you should add a line in `/etc/init.d/boot.local` to start it during boot.

Now you are ready to start using the watchdog.

## 4.2 Using the watchdog

After the correct watchdog device is configured in `/etc/watchdog.conf` and **wd\_keepalive** 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. Go to the **Advanced** menu and select **System monitoring**.

In this screen you can select the value for the **Watchdog timeout**. The default value is **0** which means that the watchdog timer is disabled. Enter the amount of minutes here after which the watchdog should trigger the reboot.



### Important

Be careful not to choose too small values for the watchdog timer. Keep in mind that sometimes your system might do maintenance jobs during bootup before **wd\_keepalive** is 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 maintenance work.

---

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

## 4.3 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 unsaved 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:

```
1 linux-o21n:~ # echo 1 > /proc/sys/kernel/sysrq
2 linux-021n:~ # 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.

---

## Chapter 5

# Online resources

### 5.1 Watchdog package home page

<http://watchdog.sourceforge.net/>