Sensors HowTo

Hardware monitoring on Fujitsu mainboards

Rainer König

Sensors HowTo
Hardware monitoring on Fujitsu mainboards

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

This document explains how to use the hardware monitoring 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!

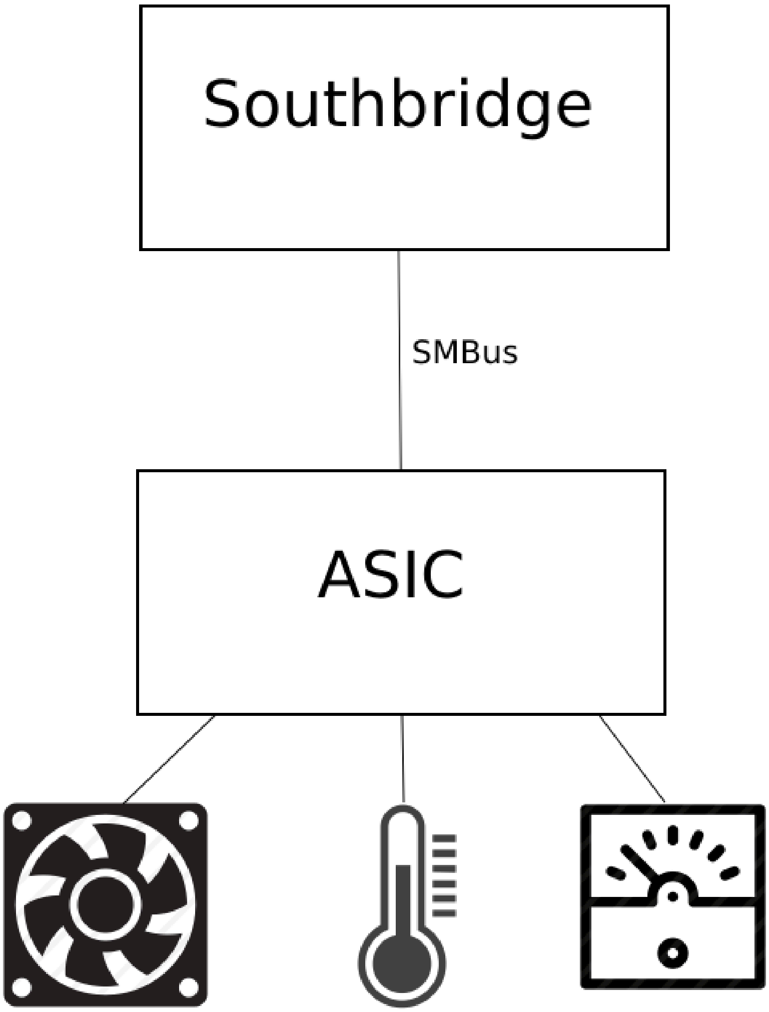
If you have a suggestion for improving the documentation, try to be as specific as possible when describing it. If you have found an error, please include the section number and some of the surrounding text so we can find it easily.

Introduction

1.1. General hardware information

Fujitsu mainboards provide hardware monitoring functions by using an ASIC that can be read out by a Linux kernel device driver.

The following picutre shows how the ASIC is connected.



The ASIC is connected to the southbridge using an SMBus (System Management Bus) which is usually an I2C bus. The ASIC is capable of monitoring the fan speeds, the thermal sensors and the voltages of your systemboard.

1.2. General software information

If we look at the software then we need to distinguish between kernel drivers and userspace applications.

1.2.1. Kernel drivers needed to make the sensors work

To access the sensors you need the following drivers on the kernel side:

- A driver for the SMBus interface of your Southbridge. Without such a driver you cannot establish a bus connection to the SMBus and therefore you can't access the ASIC at all.

- A driver for the ASIC that can read out all sensors.

1.2.2. Userspace applications

The userspace application to read out the sensor data is called lm-sensors. This application is usually shipped with all current Linux distributions.

There are some desktop applets for displaying sensors around. The best supported GUI frontend as of today is GKrellM which is also avaiable in most Linux distributions. Other graphical frontends like the sensors plugin for gDesklets for Gnome seem to be no longer maintained.

1.3. 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.

Fujitsu mainboard overview

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

Table 2.1. 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

3.1. 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.

3.2. ASIC drivers

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

3.2.1. 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 5.2, “Homepage for the SCH5627 & SCH5636 (Theseus) driver”*).

3.2.2. 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 5.2, “Homepage for the SCH5627 & SCH5636 (Theseus) driver”*).

3.2.3. 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 5.1, “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.

Chapter 4.

The userspace application lm-sensors

The main userspace application for system monitoring with Linux is lm-sensors.

4.1. Installation

lm-sensosrs is part of the most Linux distributions. So you can easily install it using your distributions package manager. If its not available for your distribution you can obtain the source on the homepage and compile it yourself.

4.2. Hardware detection and kernel configuration

lm-sensors provides a hardware detection script that aims to scan all I2C bus interfaces and look for known ASICs. The script is executed with the following command:

```
# sensors-detect
```

sensors-detect will ask you what parts of your hardware should be scanned for manamgenent devices. Use the defaults for scanning, there is no need to try out other options.

When sensors-detect is finished it might ask you to activate a boot service or to write configurations. Do that and then reboot the system to get everything in place.

After rebooting verify that the kernel drivers are loaded with the following command:

```
# lsmod
```

You should see the kernel drivers listed here.

4.3. Troubleshooting sensors detect

sensors-detect is just a Perl script that can suffer from two problems:

- It complains that your SMBus interface is not supported
- It can't identify the ASIC on your mainboard

The root cause for the first problem (unsupported SMBus interface) is that the script maintains its own list of PCI adapters where it holds the vendor and device IDs. Unfortunately this list is a bit out-of-date and does not contain the Skylake SMBus interface.

The root cause for the second problem (can't identify ASIC) is that the script does not yet know about the Teutates chip. There is a patch for the sensors-detect script available in the sources of the Teutates driver that fixes this problem, but you also might suffer from the first problem too on Skylake chipsets.

A workaround is that you add the needed ASIC modules to the boot configuration of your distribution. Usually this can be done by adding a configuration file in "/etc/modprobe.d/ that tells the kernel to load one driver. For the teutates you could simply add a file "teutates.conf" that looks like that:

```
# modprobe ftsteutates
```

This ensures that the ftsteutats driver will be loaded on the next boot.

Important

The ASIC drivers usually can't autoload since they are no plug&play drivers that are automatically identified during boot. So you need to list them as additional modules to load.

So if you added the drivers manually to your boot configuration then just reboot and look if the drivers really got loaded.

4.4. Reading the sensors

The next step is to verify if you can read out sensors. Execute the following command:

```
# sensors
```

Now you should get a list of all your sensor values. This list is using generic labels as long as you don't configure the ASIC data for lm-sensors.

This means you need to provide a configuration file that maps the generic labels of the driver to the real labels on your mainboard.

4.4.1. Confiruation for the SCH5627 chip

To get a translation table between the generic labels and what the sensors are really used for you need to put them into a file named **`/etc/sensors.d/SCH5627.conf`**:

```
chip "SCH5627-*"
    label in0 "3.3V"
    label in1 "VDimm"
    label in2 "VBAT"
    label in3 "3VSB"
    label in4 "VCore"
    label fan1 "CPU-Fan"
    label fan2 "Case-Fan"
    ignore fan3
    ignore fan4
    label temp1 "CPU"
    label temp2 "SIO"
    label temp3 "PCIe"
    ignore temp4
    ignore temp5
    ignore temp6
    ignore temp7
    ignore temp8
```

If you run sensors again your list should show the correct labels for your sensors.

4.4.2. Configuration for the Theseus chip

To get a translation table between the generic labels and what the sensors are really used for you need to put them into a file named **`/etc/sensors.d/theseus.conf`**:

```
chip "theseus-*"
  label fan1 "CPU Fan"
  label fan2 "Rear Fan"
  label fan3 "Fan3"
  label fan4 "Front Fan"
  label temp1 "CPU"
  label temp2 "Super I/O"
  label temp3 "Memory"
  label temp4 "Power Regulator"
  label temp5 "PCH"
  label temp6 "PCIe"
```

If you run sensors again your list should show the correct labels for your sensors.

4.4.3. Configuration for the Teutates chip

The ftsteutates driver comes with a set of config files for the different mainboards that use the Teutates chip. You just need to copy the file for your mainbord number to the **`/etc/sensors.d/`** directory to enable the mapping.

Online resources

5.1. Download adress for the ftsteutates driver

As long as the ftsteutaes driver is not yet part of the Linux kernel it can be downloaded from the Fujitsu FTP site.

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

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

5.3. Homepage of lm-sensors

The lm-sensors source can be found on this GitHub repository: *https://github.com/groeck/lm-sensors*

5.4. GKrellM homepage

http://freecode.com/projects/gkrellm

Appendix A. Revision History

Revision 0-9 **2013-12-04** **Rainer König**

First draft

Revision 1-0 **2013-08-01** **Rainer König**

Updates for D3003-Sx and D3313-Sx boards.

Revision 1-1 **2015-02-10** **Rainer König**

Changed link to sch56xx driver website.

Revision 1-2 **2016-04-12** **Rainer König**

Complete rework of the document to make it better..