NAME

fancontrol − automated software based fan speed regulation

SYNOPSIS

fancontrol *[configfile]*

DESCRIPTION

fancontrol is a shell script for use with lm_sensors. It reads its configuration from a file, then calculates fan speeds from temperatures and sets the corresponding PWM outputs to the computed values.

WARNING

Please be careful when using the fan control features of your mainboard, in addition to the risk of burning your CPU, at higher temperatures there will be a higher wearout of your other hardware components, too. So if you plan to use these components in 50 years, **maybe** you shouldn't use fancontrol at all. Also please keep in mind most fans aren't designed to be powered by a PWMed voltage.

In practice it doesn't seem to be a major issue, the fans will get slightly warmer, just be sure to have a temperature alarm and/or shutdown call, in case some fan fails, because you probably won't hear it anymore ;)

CONFIGURATION

For easy configuration, there's a script named **pwmconfig**(8) which lets you interactively write your configuration file for **fancontrol**. Alternatively you can write this file yourself using the information from this manpage.

Since most of you are going to use **pwmconfig**(8) script, the config file syntax will be discussed last. First I'm going to describe the various variables available for changing **fancontrol**'s behaviour:

INTERVAL
This variable defines at which interval in seconds the main loop of **fancontrol** will be executed

DEVPATH
Maps hwmon class devices to physical devices. This lets **fancontrol** check that the configuration file is still up-to-date.

DEVNAME
Records hwmon class device names. This lets **fancontrol** check that the configuration file is still up-to-date.

FCTEMPS
Maps PWM outputs to temperature sensors so **fancontrol** knows which temperature sensors should be used for calculation of new values for the corresponding PWM outputs.

FCFANS
Records the association between a PWM output and a fan input. Then **fancontrol** can check the fan speed and restart it if it stops unexpectedly.

MINTEMP
The temperature below which the fan gets switched to minimum speed.

MAXTEMP
The temperature over which the fan gets switched to maximum speed.

MINSTART
Sets the minimum speed at which the fan begins spinning. You should use a safe value to be sure it works, even when the fan gets old.

MINSTOP

The minimum speed at which the fan still spins. Use a safe value here, too.

MINPWM

The PWM value to use when the temperature is below MINTEMP. Typically, this will be either 0 if it is OK for the fan to plain stop, or the same value as MINSTOP if you don’t want the fan to ever stop. If this value isn’t defined, it defaults to 0 (stopped fan).

MAXPWM

The PWM value to use when the temperature is over MAXTEMP. If this value isn’t defined, it defaults to 255 (full speed).

AVERAGE

How many last temperature readings are used to average the temperature. It can be used to smoothen short temperature peaks. If this value isn’t defined, it defaults to 1 (no averaging).

The configuration file format is a bit strange:

```
VARIABLE=chip/pwmdev=value chip/pwmdev2=value2
VARIABLE2=...
```

Each variable has its own line. The variable name is followed by an equal sign and the device=value pairs. These consist of the path to the pwm output for which the value is valid, equal sign followed by the value and are separated by a blank. Path can be absolute or relative (from /sys/bus/i2c/devices or /sys/class/hwmon depending on the kernel version). Example:

```
MINTEMP=hwmon0/device/pwm1=40 hwmon0/device/pwm2=54
```

You have to play with the temperature values a bit to get happy. For initial setup I recommend using the **pwmconfig** script. Small changes can be made by editing the config file directly following the rules above.

Upon starting, fancontrol will make sure that all referenced devices do exist and match what they were at configuration time, and that all referenced sysfs files do exist. If not, it will quit immediately, upon the assumption that the configuration file may be out-of-sync with the loaded kernel drivers.

THE ALGORITHM

fancontrol first reads its configuration, writes it to arrays and loops its main function. This function gets the temperatures and fanspeeds from kernel driver files and calculates new speeds depending on temperature changes, but only if the temp is between MINTEMP and MAXTEMP. After that, the new values are written to the PWM outputs. Currently the speed increases linearly with rising temperature. This way you won’t hear your fans most of the time at best.

SEE ALSO

pwmconfig(8), sensors(1).

AUTHOR

Marius Reiner <marius.reiner@hdev.de>