

How to config PC - fan in ubuntu

Enthaltene Funktionen

- Sensoren auslesen mit lm - sensors
- Lüfterregelung erstellen mit fancontrol
- Automatischen Start einrichten
- Drehzahl beim Beenden steuern
- Lösung über die GUI

Voraussetzung

- PWM - Steuerung für Lüfter

1. Sensoren auslesen und als Konfiguration speichern

- Voraussetzung für die Lüfterregelung ist eine Erfassung der Temperatur- und Drehzahlsensoren am Mainboard. Hierfür wird das Paket lm-sensors benötigt, welches mit: `sudo apt-get install lm-sensors` installiert wird. Nach der Installation müssen dann die verfügbaren Sensoren abgefragt werden. Dies erfolgt mit dem Befehl:

`sudo sensors-detect`

Es folgen nun je nach Anzahl der gefundenen Sensoren je eine Meldung, ob eine Abfrage erfolgen soll. Diese Meldungen können alle mit der Angabe von <yes> oder einfach mit {Enter} bestätigt werden. Am Ende der Abfragen erscheint dann folgende Meldung:

“Do you want to add these lines automatically to /etc/modules? (yes/NO)“

Diese muss mit <yes> bestätigt werden. Es werden dann alle gefundenen Sensorangaben in die Datei (modules) eingetragen. Als nächstes muss die Konfiguration neu geladen werden, mit: `sudo service module-init-tools start` oder für ältere Ubuntu - Versionen mit: `sudo /etc/init.d/module-init-tools start`

Nach dieser Konfiguration können nun die Sensoren ausgelesen werden. Hierfür einfach den Befehl: `sensors` angeben. Es werden dann alle Sensoren gelistet ausgegeben

```

user@server:~$ sensors
coretemp-isa-0000
Adapter: ISA adapter
Core 0:      +46.0Â°C  (high = +86.0Â°C, crit = +100.0Â°C)
Core 1:      +42.0Â°C  (high = +86.0Â°C, crit = +100.0Â°C)

w83627dhg-isa-0290
Adapter: ISA adapter
Vcore:      +1.10 V  (min = +0.00 V, max = +1.74 V)
in1:        +0.91 V  (min = +0.26 V, max = +0.51 V)  ALARM
AVCC:       +3.44 V  (min = +2.98 V, max = +3.63 V)
+3.3V:      +3.44 V  (min = +2.98 V, max = +3.63 V)
in4:        +1.72 V  (min = +0.32 V, max = +0.14 V)  ALARM
in5:        +1.74 V  (min = +0.53 V, max = +0.51 V)  ALARM
in6:        +1.91 V  (min = +1.28 V, max = +1.42 V)  ALARM
3VSB:       +3.52 V  (min = +2.98 V, max = +3.63 V)
Vbat:       +3.34 V  (min = +2.70 V, max = +3.30 V)  ALARM
fan1:        0 RPM  (min = 10546 RPM, div = 128)  ALARM
fan2:       1054 RPM (min = 795 RPM, div = 8)
fan3:        0 RPM  (min = 10546 RPM, div = 128)  ALARM
fan4:        0 RPM  (min = 1318 RPM, div = 128)  ALARM
fan5:        0 RPM  (min = 10546 RPM, div = 128)  ALARM
temp1:       +35.0Â°C  (high = -110.0Â°C, hyst = -94.0Â°C)  ALARM sensor = thermistor
temp2:       +34.0Â°C  (high = +80.0Â°C, hyst = +75.0Â°C)  sensor = thermistor
temp3:       +43.0Â°C  (high = +80.0Â°C, hyst = +75.0Â°C)  sensor = thermistor
cpu0_vid:    +0.000 V

```

2. Lüfterregelung erstellen

- nachdem nun die Sensoren ausgelesen und als Konfiguration gespeichert sind, kann jetzt die Regelung des CPU - Lüfters erfolgen. Als erstes muss das Paket fancontrol mit: `sudo apt-get install fancontrol` installiert werden. Jetzt kann man mit dem Befehl `sudo pwmconfig` die Konfiguration starten. Das Programm versucht dann festzustellen, an welchem Anschluss ein Lüfter angeschlossen ist und regelt die Drehzahl automatisch. Jeder Anschluss wird mit einer Meldungsausgabe abgeschlossen, welche man mit <y> oder <n> bestätigen muss. D.h., wenn z.B. an Anschluss 2 ein Lüfter vorhanden ist und dieser wird kurzzeitig angesteuert, bestätigt man die Meldung mit <y>, ansonsten mit <n> und somit weiß das Programm, welcher Anschluss belegt ist und welcher nicht. Wenn man möchte, kann man zudem eine detaillierte Ansteuerung des Lüfters veranlassen indem die Meldung: „Would you like generate a detailed correlation (y)“ mit y bestätigt. Es wird dann Stufenweise der Lüfter angesteuert, wobei man sich nun einen Wert notieren sollte, z.B. 135 für die max. Umdrehung (der Wert wird später benötigt). Das ganze könnte dann wie folgt aussehen

```
the pwm outputs connected to the fan connectors,
check out the hardware database on http://www.almico.com/forumindex.php)

Did you see/hear a fan stopping during the above test (n)? n

Testing pwm control hwm0n1/device/pwm2 ...
hwm0n1/device/fan2_input ... speed was 2636 now 1430
It appears that fan hwm0n1/device/fan2_input
is controlled by pwm hwm0n1/device/pwm2
Would you like to generate a detailed correlation (y)? y
Note: If you had gnuplot installed, I could generate a graphical plot.
PWM 255 FAN 2721
PWM 240 FAN 2678
PWM 225 FAN 2445
PWM 210 FAN 2280
PWM 195 FAN 2280
PWM 180 FAN 1939
PWM 165 FAN 1704
PWM 150 FAN 1704
PWM 135 FAN 1371
PWM 120 FAN 1081
PWM 105 FAN 1081
PWM 90 FAN 1004
```

Am Ende, wenn alle Sensoren abgefragt wurden, wird folgende Meldung ausgegeben:
“What should be the path to your fancontrol config file (/etc/fancontrol)”

```
The fancontrol script can automatically respond to temperature changes
of your system by changing fanspeeds.
Do you want to set up its configuration file now (y)? y
What should be the path to your fancontrol config file (/etc/fancontrol)?
Loading configuration from /etc/fancontrol ...

Select fan output to configure, or other action:
1) hwm0n1/device/pwm2   3) Just quit           5) Show configuration
2) Change INTERVAL    4) Save and quit

select (1-n):
```

Die abgeschlossene Konfiguration muss in dieser Datei gespeichert werden. Sofern diese nicht bereits vorhanden ist, wird mit der Bestätigung {Enter} eine neue Datei autom. erstellt. Im Beispiel wurde der Sensor pwm2 gefunden, nun beginnt die Konfiguration der Lüfterregelung an diesem Sensor in einzelnen Schritten. Hierfür muss die Nummer des Sensors angegeben werden (im Beispiel also: 1). Jetzt erfolgt die Angabe der Temperatur für die Abschaltung sowie die Werte für die Drehzahlen des Lüfters

Im Beispiel wurden 30° für die Abschaltung des Lüfters eingetragen, d.h. wenn die CPU-Temperatur diesen Wert unterschreitet, schaltet der Lüfter ab, für die maximale Drehzahl wurden 50° vorgegeben, d.h. wenn die CPU-Temperatur 50° überschreitet, wird die Drehzahl des Lüfters auf das Maximum erhöht. Als Letztes gibt man nun die max. Drehzahl für den Lüfter an, also die Drehzahl, welche bei Überschreitung der 50° erfolgt. Im Beispiel wurden hier 20, also eine niedrige Drehzahl verwendet, da der Server keine Großen Aufgaben übernimmt

```
Select a temperature sensor as source for hwmon1/device/pwm2:
1) hwmon0/device/temp2_input
2) hwmon0/device/temp3_input
3) hwmon1/device/temp1_input
4) hwmon1/device/temp2_input
5) hwmon1/device/temp3_input
6) None (Do not affect this PWM output)
select (1-n): 1

Enter the low temperature (degree C)
below which the fan should spin at minimum speed (20): 30

Enter the high temperature (degree C)
over which the fan should spin at maximum speed (60): 50

Enter the PWM value (0-255) to use when the temperature
is over the high temperature limit (255): 20

Select fan output to configure, or other action:
1) hwmon1/device/pwm2  3) Just quit          5) Show configuration
2) Change INTERVAL    4) Save and quit
select (1-n): █
```

Abschließend speichert man nun die Konfiguration durch Eingabe mit 4. Mit folgenden Befehl startet man jetzt die Steuerung: `sudo /etc/init.d/fancontrol start`
Jetzt sollte der Lüfter auch mit den eingestellten Werten angesteuert werden.

3. Automatischen Start einrichten

- damit nun die Lüfterregelung beim Neustart des Systems greift, muss in der Datei `/etc/rc.local` die entsprechende Zeile gesetzt werden. Dies könnte dann wie folgt aussehen:

```
# By default this script does nothing.
/etc/init.d/fancontrol start
exit 0
```

4. Drehzahl beim Beenden der Steuerung steuern

Beim Beenden der Lüftersteuerung, z.B. wenn der Computer herunter gefahren wird, kann es auf einigen Systemen vorkommen, dass der Lüfter dann mit voller Drehzahl betrieben wird. Um dies zu unterbinden, kann man am Skript selber eine kleine Anpassung vornehmen. Dazu öffnet man mit einem Editor die Datei `/usr/sbin/fancontrol` und kommentiert in der Methode `restorefans()` die Zeile mit dem Eintrag `pwmdisable $pwmo` aus und fügt anschließend folgende neue Zeile ein:

```
echo `expr ${AFCMAXPWM[$fcvcount]} - 25` > $pwmo
```

Die Änderung sollte dann wie folgt aussehen:

```
# pwmdisable $pwmo
echo `expr ${AFCMAXPWM[$fcvcount]} - 25` > $pwmo
```

Sie bewirkt, dass der Lüfter beim Beenden der Lüftersteuerung mit dem in der `/etc/fancontrol` angegebenen Maximalwert minus 25 angesteuert wird.