

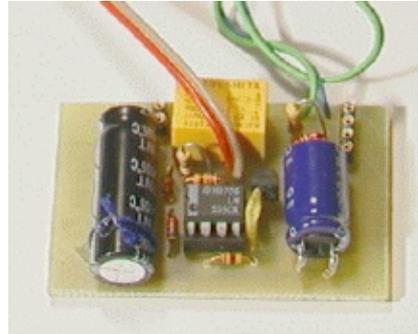
Eine Hardware Watchdog mit Taste zum Herunterfahren des Rechners



by Guido Socher ([homepage](#))

About the author:

Guido mag Linux, weil es immer interessant ist, zu sehen, wie Computer wirklich funktionieren. Linux mit seinem modularen Aufbau ist das beste System für ein "ich entdecke meinen Computer Abenteuer".



Abstract:

Der [LCD Kontrollfeld Artikel](#) zeigte, wie man eine kleine Microcontroller-basierte LCD Anzeige mit enormen Möglichkeiten baut. Manchmal braucht man aber all diese Möglichkeiten gar nicht. Die Schaltung, die wir hier entwickeln, ist viel billiger (das LCD Kontrollfeld war schon preiswert) und enthält nur zwei wichtige Funktionen:

- Ein Taster zum Herunterfahren des Servers
- Eine Watchdog, um den Server zu überwachen

Unsere hier vorgestellte Hardware besteht nur aus überall verfügbaren Bauteilen. Du wirst keine Probleme haben, die Teile zu bekommen. Alles zusammen kostet etwa 5 Euro.

Was ist eine Watchdog?

Eine Watchdog ist in der Computersprache ein sehr zuverlässiges Stück Hardware, das sicherstellt, daß der Server immer läuft. Man findet solche Dinger im Mars Pathfinder (wer möchte schon zum Mars reisen, um eine Resettaste zu drücken?) oder in sehr teuren Servern.

Die Idee, die hinter solch einer Watchdog steckt ist sehr einfach: Der Computer muß von Zeit zu Zeit "hallo" zur Watchdog sagen, um ihr mitzuteilen, daß er noch am Leben ist. Falls der Computer das nicht macht, bekommt er einen Hardware-Reset.

Beachte, daß ein Linuxserver normalerweise ununterbrochen für mehrere Monate arbeiten sollte, im Mittel vielleicht 1-2 Jahre, ohne sich aufzuhängen. Wenn du einen Rechner hast, der sich jede Woche aufhängt, dann ist etwas faul und eine Watchdog ist die falsche Lösung. Du solltest dann lieber die Hardware

überprüfen. Defektes RAM (siehe memtest86.com) überhitzte CPU, zuuuuu lange IDE Kabel ...

Wenn Linux so zuverlässig ist, daß es ein Jahr ohne Probleme läuft, warum braucht man dann eine Watchdog? Nun die Antwort ist einfach: Um es noch zuverlässiger zu machen. Es gibt außerdem ein menschliches Problem. Ein Server, der nie Probleme gemacht hat, ist dem Servicepersonal unbekannt. Wenn der Server dann versagt, weiß niemand, wo er steht. Es kann auch sein, daß der Server genau im Weihnachtsurlaub stehen bleibt und niemand da ist. In all diesen Fällen ist eine Watchdog die richtige Lösung.

Eine Watchdog löst jedoch nicht alle Probleme. Sie ist kein Schutz gegen defekte Hardware. Wenn man eine Watchdog in den Server einbaut, dann sollte man auch darauf achten, das der Rest stimmt (nicht immer die neusten Boards + BIOS Fehler nehmen, gute Kühlung).

Wie benutzt man eine Watchdog?

Die Watchdog stellt lediglich sicher, daß normale Programme (user space) immer noch ausgeführt werden können. Um ein wirklich zuverlässiges System zu haben, muß man seine Applikationen (Web-server, Datenbank) und System Resources (Plattenplatz, vielleicht CPU Temperatur) überwachen. Diese Aufgabe können andere Programme übernehmen (crontab). All das ist schon in dem [LCD Kontrollfeld Artikel](#) beschrieben. Ich gehe daher hier nicht weiter ins Detail.

Beispiele? Hier ist ein [kleines Skript](#), das das Netzwerk, den Swap space und den Plattenplatz überwacht.

```
#!/bin/sh
PATH=/bin:/usr/bin:/usr/local/bin
export PATH
#
# Monitor the disk
# -----
# check if any of the partitions are more than 80% full.
# (crontab will automatically send an e-mail if this script
# produces some output)
df | egrep ' (8.%|9.%|100%) '
#
# Monitor the swap
# A server should normally be dimensioned such that it
# does not swap. Swap space should therefore be constant
# and limited.
# -----
# check if more than 6 Mb of swap are used
swpfree=`free | awk '/Swap:/{ print $3 }'`
if expr $swpfree \> 6000 > /dev/null ; then
    echo "$0 warning! swap usage is now $swpfree"
    echo " "
    free
    echo " "
    ps auxw
fi
#
# Monitor the network
# -----
```

```

# your _own_ IP addr or hostname:
hostn="linuxbox.your.supercomputer"
#
if ping -w 5 -qn -c 1 $hostn > /dev/null ; then
    # ok host is up
    echo "0" > /etc/pingfail
else
    # no answer count up the ping failures
    if [ -r /etc/pingfail ]; then
        pingfail=`cat /etc/pingfail`
    else
        # we do not handle the case where the
        # pingfail file is missing
        exit 0
    fi
    pingfail=`expr "$pingfail" "+" 1`
    echo "$pingfail ping failures"
    echo "$pingfail" > /etc/pingfail
    if [ $pingfail -gt 10 ]; then
        echo "more than 10 ping failures. System reboot..."
        /sbin/shutdown -t2 -r now
    fi
fi
# ---- end of monitor script ----

```

Man kann dieses Skript mit einem crontab Eintrag kombinieren, der das Skript alle 15 Minuten ausführt:

```
1,15,30,45 * * * * /where/the/script/is
```

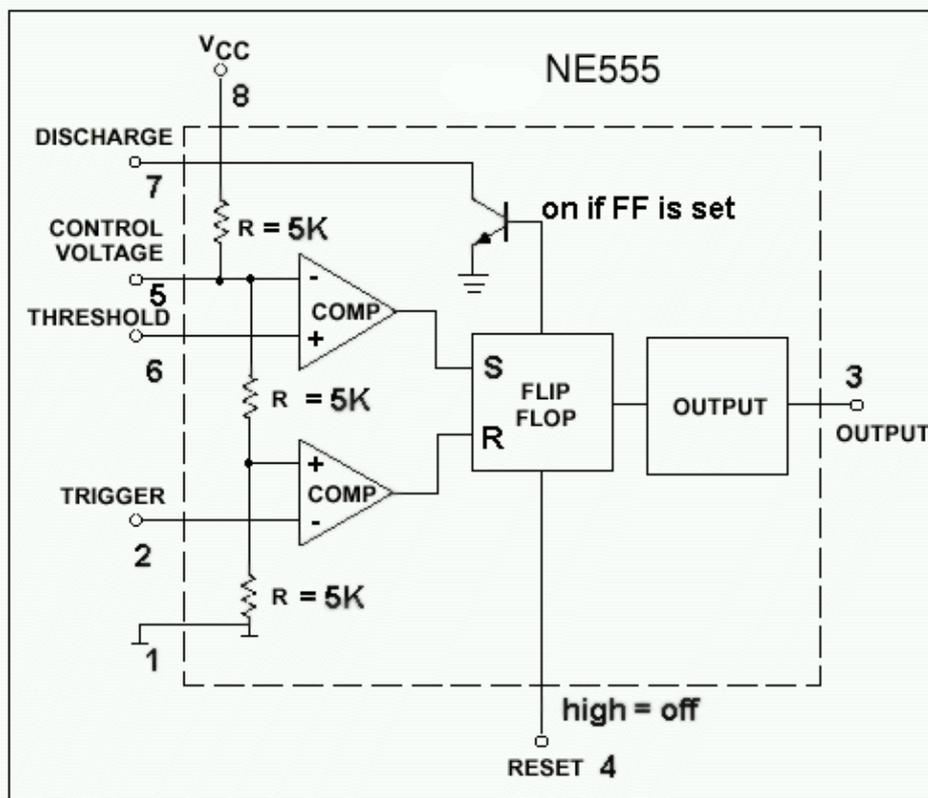
Die Watchdog Hardware

Es gibt kein Standard Relai. Jeder Hersteller hat seine eigenen Bauformen. Bei unserer Schaltung ist es von Bedeutung, wie hoch der Innenwiderstand der Spule ist. Hier sind deshalb zwei Schaltungen ausgeführt, eine für ein 5V, 500 Ohm Relai und die andere für ein 5V, 120 Ohm Relai. Frag nach der Impedanz oder messe sie mit einem Ohmmeter, bevor du das Relai kaufst. Man kann auf die Schaltpläne klicken, um ein größeres Bild zu erhalten.

120 Ohm Relai

Wie die Schaltung funktioniert

Die Watchdog Schaltung baut auf den NE555 Timer Chip auf. Dieser Chip enthält 2 Komparatoren, eine Flipflop und 3x 5K Ohm Widerstände, um eine Referenz für die Komparatoren zu haben. Wenn der Pin namens threshold (6) über 2/3 der Versorgungsspannung kommt, dann wird das Flipflop gesetzt (Zustand ein).



Nun ein Blick auf unsere Schaltung: Wir benutzen den RTS Anschluß von der seriellen Schnittstelle als Versorgungsspannung. Die Spannung an der RS232 Schnittstelle ist +/- 10V, daher brauchen wir eine Diode vor Kondensator C1. C1 wird sehr schnell geladen und dient als Energiespeicher, der in der Lage sein muß, das Relai für einen Moment zum Anziehen zu bringen.

Kondensator C2 wird langsam über den 4.7M Widerstand geladen. Der Transistor T1 entlädt C2 immer, wenn er einen kurzen Puls über den RS232 DTR pin erhält. Falls die Pulse nicht kommen, weil der Computer sich aufgehängt hat, dann wird C2 irgendwann über 2/3 der Versorgungsspannung geladen und das Flipflop geht auf "an". Das Laden von C2 dauert ca. 40 Sekunden.

Der Kondensator C1, der Widerstand R2, die LED und das Relai müssen so dimensioniert sein, daß das Relai kurz aus der Energie des Kondensators zum Anziehen gebracht werden kann, aber es darf nicht permanent gehalten werden, selbst wenn RTS auf 10V liegt. Wir wollen schließlich erreichen, daß der "Resettaster" nur für 1-2 Sekunden "gedrückt" wird.

Die LED bleibt an, bis der Server nach einem Reset wieder hoch kommt.

Wie man im Schaltbild sehen kann, gibt es noch einen Shutdown Taster, der mit Pin CD verbunden ist. Wird dieser Taster für 15 Sekunden (oder länger) gedrückt, wird die Treibersoftware "shutdown -h now" ausführen und den Server herunterfahren. Das ist für normale Wartungsarbeiten gedacht und hat nichts mit der Watchdog zu tun.

Die Treibersoftware

Die Treibersoftware ist ein kleines C Programm, das über ein `/etc/init.d/` Skript gestartet werden kann. Es schaltet den RS232 Pin RTS permanent ein und schickt dann alle 12 Sekunden einen Puls an DTR (der Timeout der Watchdog ist 40 Sekunden). Falls man den Computer normal herunterfährt, dann wird das Programm zuerst RTS ausschalten und dann einen letzten Puls an DTR geben. Der Effekt ist, das Kondensator C1 schon entladen ist, bevor der Watchdogtimeout kommt. Die Watchdog schlägt also in diesem Fall nicht zu. Um die Software zu installieren, mußt du einfach `linuxwd-0.3.tar.gz` auspacken. `linuxwd-0.3.tar.gz` ist auf der Download Seite zu haben. Mit

```
make
```

kompiliert man die Software. Das Programm `linuxwd` kann nun nach `/usr/sbin/linuxwd` kopiert werden.

Editire das Skript `linuxwd_rc` (für redhat/mandrake, oder `linuxwd_rc_anydist` für andere Distributionen) und setze den Port der seriellen Schnittstelle, die du benutzt (`ttyS1=COM2` oder `ttyS0=COM1`), ein. Kopiere das `rc`-Skript dann nach

```
/etc/rc3.d/S21linuxwd
```

und nach

```
/etc/rc5.d/S21linuxwd
```

Das ist alles.

Testen

Wenn man alles zusammengelötet hat, sollte man die Schaltung zuerst testen. Verbinde den Pin, der später an RTS angeschlossen wird, mit 9–10V Gleichspannung und warte 40–50 Sekunden. Du solltest einen kleinen Klick hören, wenn das Relai anzieht und die LED sollte angehen. Das Relai sollte nicht permanent anbleiben. Die LED sollte anbleiben, bis man auch DTR mit +10V verbunden hat.

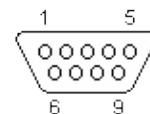
Nach diesem Test kann man die Schaltung an den Computer anschließen. Das Programm `linuxwd` hat einen Testmode, in dem es ausgibt, was es gerade macht und nach einiger Zeit aufhört, Pulse an DTR zu senden. Damit wird ein Versagen des Systems simuliert. Mit dem Befehl

```
linuxwd -t /dev/ttyS0
```

führt man `linuxwd` im Testmode aus. Die Hardware ist hier an `ttyS0=COM1` angeschlossen.

Hardware Installation

Das RS232 Interface hat folgende Pinbelegung:



9 PIN D-SUB MALE at the Computer.

9 PIN-connector	25 PIN-connector	Name	Dir	Description
1	8	CD	input	Carrier Detect
2	3	RXD	input	Receive Data
3	2	TXD	output	Transmit Data
4	20	DTR	output	Data Terminal Ready
5	7	GND	---	System Ground
6	6	DSR	input	Data Set Ready

7	4	RTS	output	Request to Send
8	5	CTS	input	Clear to Send
9	22	RI	input	Ring Indicator

Das Anschließen der Schaltung an die RS232 Schnittstelle sollte ganz einfach sein. Um die CPU Reset Leitung anzuschließen, muß man das Kabel finden, das im Computer zu dem Reset Taster führt. Parallel dazu schließen wir das Relai unserer Schaltung an.

Zusammenfassung

Eine Watchdog ist keine 100%ige Garantie für ein zuverlässiges System, aber es ist ein weiterer Schritt in Richtung Ausfallsicherheit. Ein Problem kann es geben, wenn der Filesystem-check nach einem Reset mit einem Fehler stehen bleibt. Ein Journaling Filesystem könnte hier helfen, aber ich habe damit noch nicht experimentiert. Die Watchdog, die hier vorgestellt wurde, ist preiswert und einfach zu bauen, zugleich ist sie fast so gut wie teure kommerzielle Produkte.

Links

- Die linuxwd Teibersoftware: [Software Download Seite](#)
- Datenblatt für den NE555 [NE555.pdf 140K](#)

<p><u>Webpages maintained by the LinuxFocus Editor team</u> © Guido Socher "some rights reserved" see linuxfocus.org/license/ http://www.LinuxFocus.org</p>	<p>Translation information: en --> -- : Guido Socher (homepage) en --> de: Guido Socher (homepage)</p>
---	--