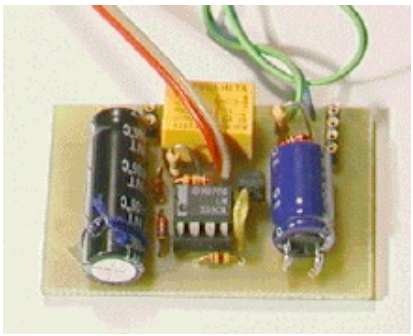
LinuxFocus article number 239
http://linuxfocus.org

A Hardware watchdog and shutdown button



by Guido Socher (homepage)

About the author:

Guido loves Linux because it is always interessting to discover how computers really work. Linux with its modularity and open design is the best system for such adventures.

Abstract:

The LCD control panel article explained how to build a small microcontroller based LCD panel with enormous possibilities. Sometimes you don’t need all those features. The hardware which we design in this article is a lot cheaper (the LCD panel was already a bargain) and includes just 2 important features from the LCD panel:

- A button to shutdown the server
- A watchdog to supervise the server

The hardware consist only of widely available parts. You will have no problems to get those parts. All parts together will cost you about 5 Euro.

What is a watchdog?

A watchdog in computer terms is a very reliable hardware which ensures that the computer is always running. You find such devices in the Mars Pathfinder (who wants to send a person to the mars to press the reset button?) or in some extra expensive servers.

The idea behind such a watchdog is very simple: The computer has to "say hello" from time to time to the watchdog hardware to let it know that it is still alive. If it fails to do that then it will get a hardware reset.

Note that a normal Linux server should be able to run uninterrupted for several month, in average probably 1-2 years without locking up. If you have machine that locks up every week then there is

something else wrong and a watchdog is not the solution. You should check for defect RAM (see memtest86.com) overheated CPUs, too long IDE cables ...

If Linux is so reliable that it will run for a year without any problems then why do you need a watchdog? Well the answer is simple to: make it even more reliable. There is as well a human problem related to that. A server that made no trouble for a year is basically unknown to the service personal. If it fails then nobody knows where it is? It might as well lock up just before Christmas when everybody is at home. In all such cases a watchdog can be very useful.

A watchdog does however not solve all of the problems. It is no protection against defect hardware. If you include a watchdog in your server then you should also ensure that you have well dimensioned (probably not the latest BIOS bugs and chipset bugs, properly cooled hardware).

How to use the watchdog?

The watchdog we design here only ensures that user space programs are still executing. To have a truly reliable system you still have to monitor your applications (web-servers, databases) and your system resources (disk space, perhaps CPU temperature). You can do this via other user space applications (crontab). All this is already described in the LCD control panel article. Therefore I will not go into further details here.

Examples? Here is a small script that can monitor networking, swap usage and disk usage.

```
#!/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
```

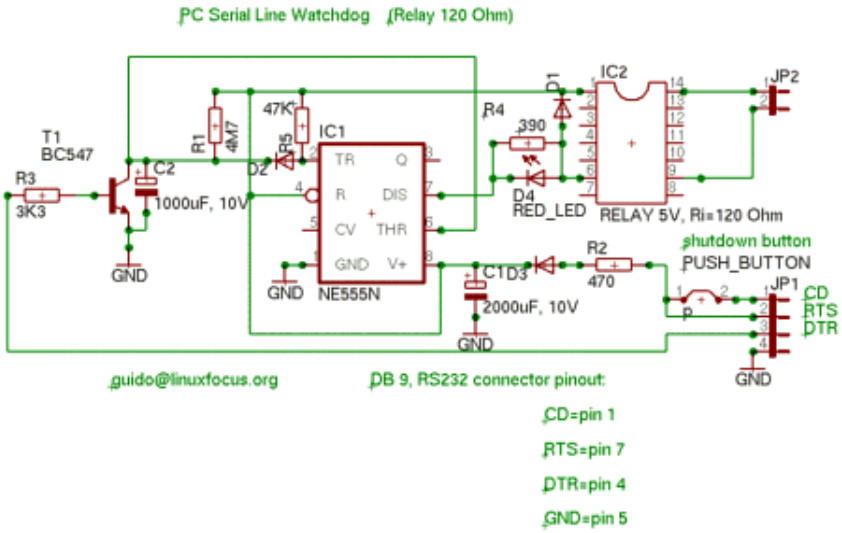
```
fi
#
# Monitor the network
# -------------------
# your _own_ IP addr or hostname:
hostn="linuxbox.your.supercomputer"
#
if ping -w 5 -qn -c 1 $hostn > /dev/null ; then
   # ok host is up
   echo "0" > /etc/pingfail
else
   # no answer count up the ping failures
   if [ -r /etc/pingfail ]; then
      pingfail=‘cat /etc/pingfail‘
   else
      # we do not handle the case where the
      # pingfail file is missing
      exit 0
   fi
   pingfail=‘expr "$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 ---
```

You can combine this with a crontab entry that will run the script every 15 minutes:

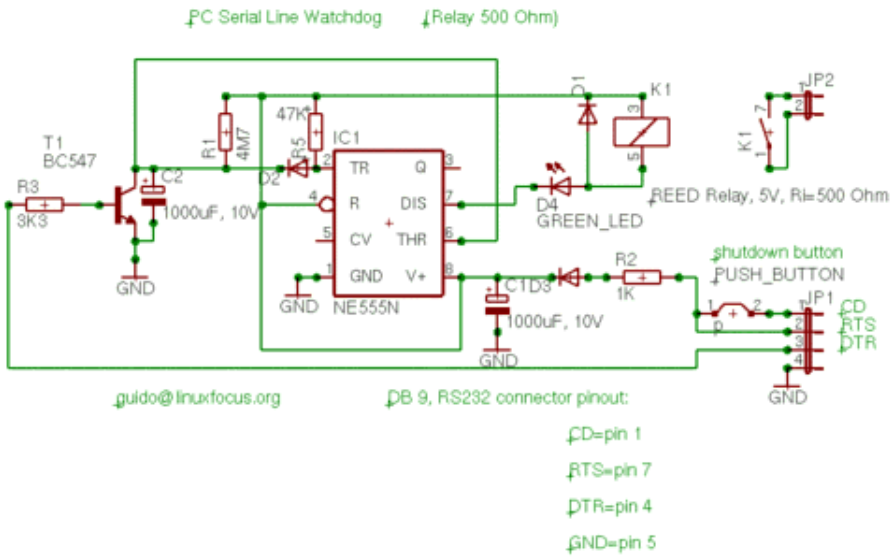
```
1,15,30,45 * * * * /where/the/script/is
```

The watchdog hardware

There is no standard relay. Every manufacture has it’s own design. For our circuit it matters very much what the inner resistance of the coil is. Therefore you find below 2 circuits one for a 5V, 500 Ohm relay and one for a 5V, 120 Ohm relay. Ask for the impedance of the relay or measure it with a Ohmmeter before you buy it. You can click on the schematic for a bigger picture.
120 Ohm relay:



500 Ohm relay:



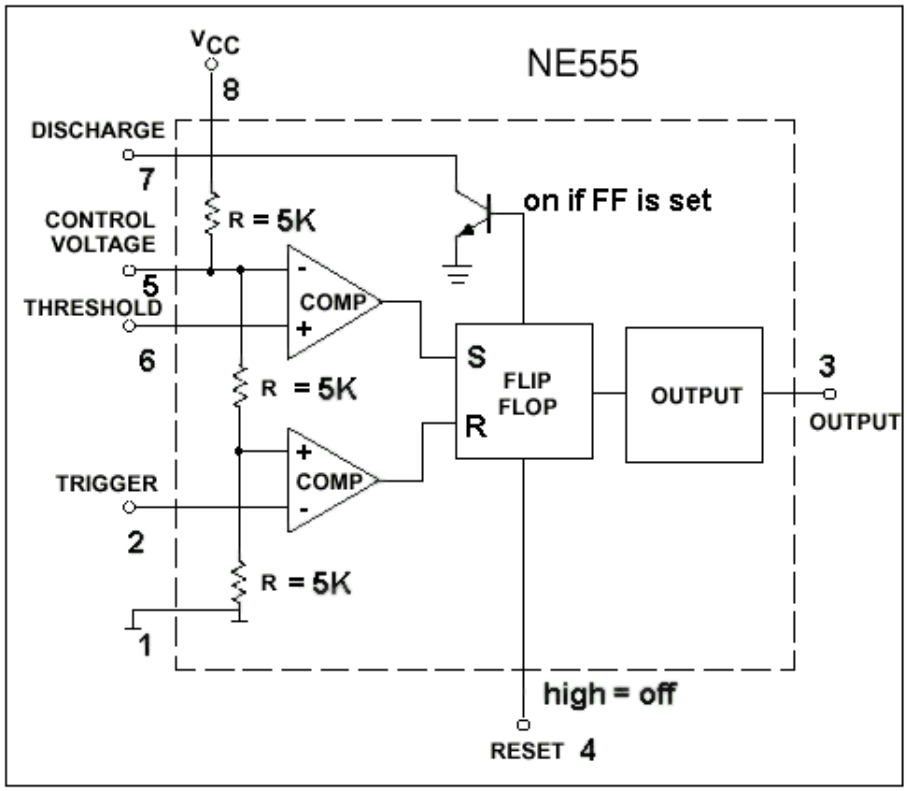
The shutdown button is a push button that connects RTS and CD when pressed. It looks a bit strange in the schematic because Eagle does not have a better symbol.

I don’t include a part list in this article. You can see what you need in the above schematic (Don’t forget the DB9 connector for the serial line). For the diodes you can use any diode, e.g 1N4148. Personally I believe that the circuit with the 500 Ohm relay is better because you do not need R4 and do not need a 2000uF (or 2200uF) capacitor. You can use a smaller 1000uF capacitor for C1.

Note: That for the 120 Ohm circuit you need a Red LED and for the 500 Ohm Relay a green LED. This is not joke. The voltage drop over a green LED is higher than over a red LED.
Board layout, eagle files and postscript files for etching the board are included in the software package which you can download at the end of the article. The Eagle CAD software for Linux is available from cadsoftusa.com.

How the circuit works

The watchdog circuit is build around the NE555 timer chip. This chip includes 2 comparators, a Flipflop and 3 resistors 5K Ohm each to have a reference for the comparators. Whenever the pin named threshold (6) goes above 2/3 of the supply voltage then the Flipflop is set (state on).



Now look at the schematic of our circuit: We use the RTS pin from the serial line as supply voltage. The voltages on the RS232 serial line interface are +/- 10V therefore we need a diode before capacitor C1. The capacitor C1 is charged very quickly and serves as a energy storage to be able to switch on the relay for a moment. Capacitor C2 is charged very slowly over the 4.7M resistor. The transistor T1 discharges the capacitor C2 if it gets a short pulse via the RS232 DTR pin. If the pulses are not coming (because the computer has locked up) then the capacitor C2 will eventually (the time is about 40 seconds) be charged above 2/3 of the supply voltage and the Flipflop goes to "on".

The capacitor C1, resistor R2 the LED and the relay have to be dimensioned such that the relay is switched on shortly from the energy in capacitor C1 but there is not enough current to keep the relay on all the time. We want the "reset button" to be "pressed" just for a second or two.

The LED will stay on until the server comes up again after a reset.

As you can see in the schematic there is as well a shutdown button connected to pin CD. If you press it for a short while (15 sec) then the driver software will run "shutdown -h now" and shutdown the server. This is for normal maintenance operations and has nothing to do with the watchdog.

The driver software

The driver software is a small C program that can be started from the /etc/init.d/ scripts. It will permanently switch on the RS232 pin RTS and then send pulses to DTR every 12 seconds (the timeout of our watchdog is 40 seconds). If you shut down your computer normally then the program will switch off RTS and give a last pulse to DTR. The effect is that the supply voltage capacitor (C1) will already be discharged before the timeout comes. Therefore the watchdog will not hit under normal operations. To install the software unpack the linuxwd-0.3.tar.gz file which you can get from the download page. Then unpack it and run
make
to compile. Copy the resulting linuxwd executable to /usr/sbin/linuxwd. Edit the provided linuxwd_rc script (for redhat/mandrake, or linuxwd_rc_anydist for any other distribution) and enter the right serial port where the hardware is connected (ttyS1=COM2 or ttyS0=COM1). Copy the rc script then to
/etc/rc3.d/S21linuxwd
and
/etc/rc5.d/S21linuxwd
That's it.

Testing

When you have soldered everything together you should test first the circuit before connecting it to the computer. Connect the pin that will later connect to the RTS line of the serial port to a 9-10V DC power supply and wait 40-50 seconds. You should hear a little click when the relay is switched on and the LED should go on. The relay should not stay permanently on. The LED will stay on until you connect as well the line that will later go to DTR to +10V.
When you have verified that this works you can connect it properly to the computer. The linuxwd program has a test mode where it produces some printouts and stops after some time to sent pulses over DTR to simulate a locked up system. Run the command

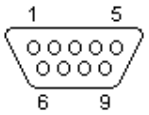
linuxwd -t /dev/ttyS0

to run linuxwd in test mode (use /dev/ttyS1 if you have the hardware on COM2).

Hardware installation

The RS232 interface has the following pinout:

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

Connecting the circuit to the RS232 should be straight forward. To connect the CPU reset line with the relay you need to locate the wires that go to the reset button on your computer. Connect the relay from our circuit in parallel to the reset button.

Conclusion

A watchdog is certainly not a 100% guarantee to have reliable system but it adds another level of security. A problem can be a situation where the file system check does not complete after a hardware reset. The new journaling filesystems might help here but I have not tried them out yet. The watchdog presented here is inexpensive, not too complex to build and almost as good as most commercial products.

References

- The linuxwd driver software: software download page
- Datasheet for the NE555 NE555.pdf 140K

2005-01-14, generated by lfparser_pdf version 2.51