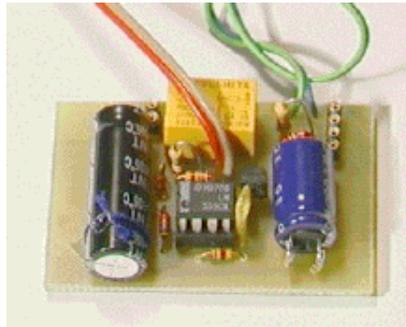


## "Chien-de-garde" matériel et bouton d'arrêt



par Guido Socher ([homepage](#))



### *L'auteur:*

Guido aime Linux parce que la manière dont fonctionne réellement un ordinateur le passionne. La modularité et l'ouverture des systèmes Linux en font un compagnon idéal pour de telles expérimentations.

*Traduit en Français par:*  
Paul Delannoy ([homepage](#))

### *Résumé:*

L'article un panneau de contrôle LCD expliquait comment construire un affichage LCD à microcontrôleur offrant de très nombreuses possibilités. Vous n'avez généralement besoin que de peu d'entre elles. Le matériel que je vais décrire dans cet article est moins cher (l'afficheur était déjà une affaire) et utilise seulement deux des fonctions principales de l'afficheur LCD :

- Un bouton d'arrêt général (shutdown) du serveur
- Un 'chien-de-garde' pour sa surveillance

Les pièces nécessaires sont disponibles pratiquement partout. Vous n'aurez pas de difficulté à les trouver. Elles vous coûteront au total environ 5 Euros.

---

## Qu'est ce qu'un 'chien-de-garde'?

En jargon informatique ce terme désigne un système matériel très fiable qui contrôle le fonctionnement continu d'un ordinateur. De tels matériels se trouvent à bord de la mission Mars Pathfinder (qui voudrait envoyer quelqu'un sur Mars simplement pour appuyer sur 'Reset' ?) et dans certains ordinateurs très chers.

L'idée de base est extrêmement simple : l'ordinateur va "parler", à intervalles réguliers, au chien-de-garde matériel pour lui montrer qu'il est toujours en fonctionnement. S'il échoue dans cette mission, le chien-de-garde redémarrera l'ordinateur.

Notez qu'un système Linux est normalement capable de fonctionner en permanence pendant des mois, en moyenne probablement 1 à 2 ans, sans aucun "plantage". Si votre machine "plante" chaque semaine, quelque chose ne va pas et le chien-de-garde n'est pas la solution. Cherchez une RAM défectueuse (voir [memtest86.com](http://memtest86.com)), un processeur en surchauffe, des câbles IDE trop longs,...

Mais puisque Linux est si fiable, étant capable de fonctionner un an sans problème, pourquoi lui adjoindre un chien-de-garde ? La réponse est simple : pour le rendre encore plus fiable. Une question humaine se cache derrière ce raisonnement. Un serveur qui ne pose aucun problème pendant un an est potentiellement inconnu des personnels de service. S'il se "plante" et que personne ne sait où il se trouve ? Il peut même décider de s'arrêter la veille de Noël, quand tout le monde est chez soi. Dans ces situations un chien-de-garde devient très utile.

Pourtant un chien-de-garde ne résout pas tous les problèmes. Il ne protège pas de la panne d'un composant matériel. Inclure un chien-de-garde dans un serveur suppose que celui-ci a été correctement préparé (mises à jour du BIOS et du chipset, ventilation correcte).

## Comment utiliser le chien-de-garde ?

Le système présenté ici vérifie uniquement le fonctionnement des programmes des utilisateurs. Pour augmenter encore la fiabilité d'une machine vous devez surveiller vos applications (serveurs web, bases de données) et vos ressources système (espace disque, peut-être la température CPU). D'autres applications (comme cron) vous y aideront. Tout ceci est déjà décrit dans l'article un panneau de contrôle LCD aussi je n'y reviens pas en détail ici.

Exemples? Voici un petit script pour surveiller le réseau, l'usage de la partition d'échange et du disque.

```
#!/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 ---

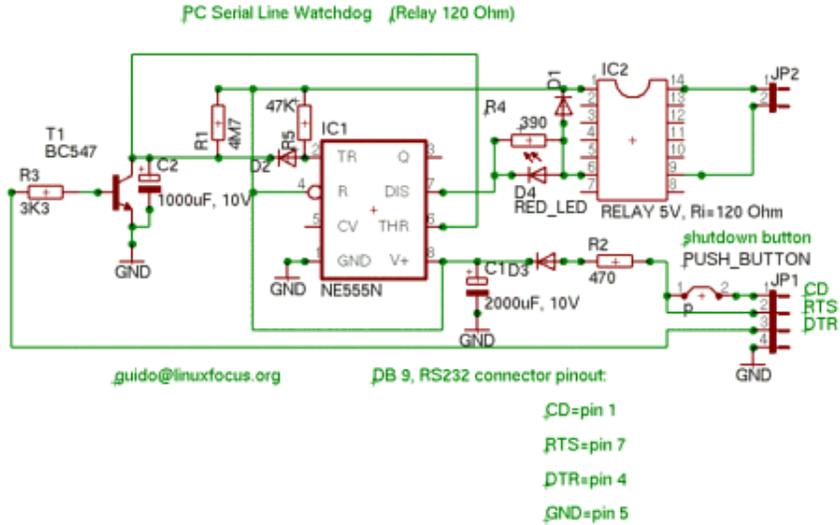
```

Ce script peut être lancé toutes les 15 minutes en ajoutant l'entrée suivante dans le fichier crontab :

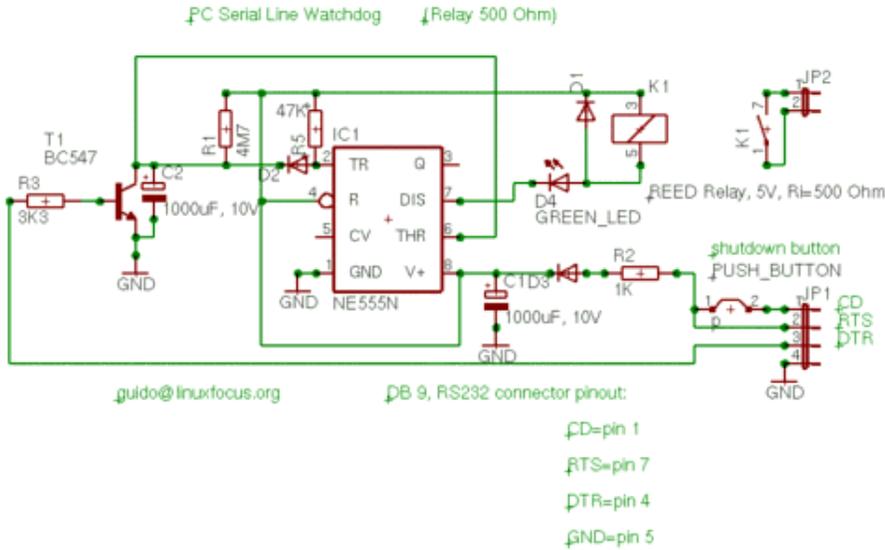
```
1,15,30,45 * * * * /chemin/du/script
```

## Le matériel pour le chien-de-garde

Il n'existe pas de relais normalisé, chaque fabricant propose sa propre conception. L'important pour nous est la valeur de la résistance interne du bobinage. Aussi je vous propose deux schémas, l'un pour un relais 5V, 500 Ohm, l'autre pour un relais 5V, 120 Ohm. Demandez l'impédance du relais ou mesurez-la avec un Ohmmètre avant de l'acheter. Vous pouvez agrandir le schéma en cliquant dessus. Relais 120 Ohm :



Relais 500 Ohm :



Le bouton d'arrêt met en contact les lignes RTS et CD lorsqu'il est pressé. Il a un aspect un peu étrange sur le schéma parce que Eagle ne propose pas de meilleur symbole.

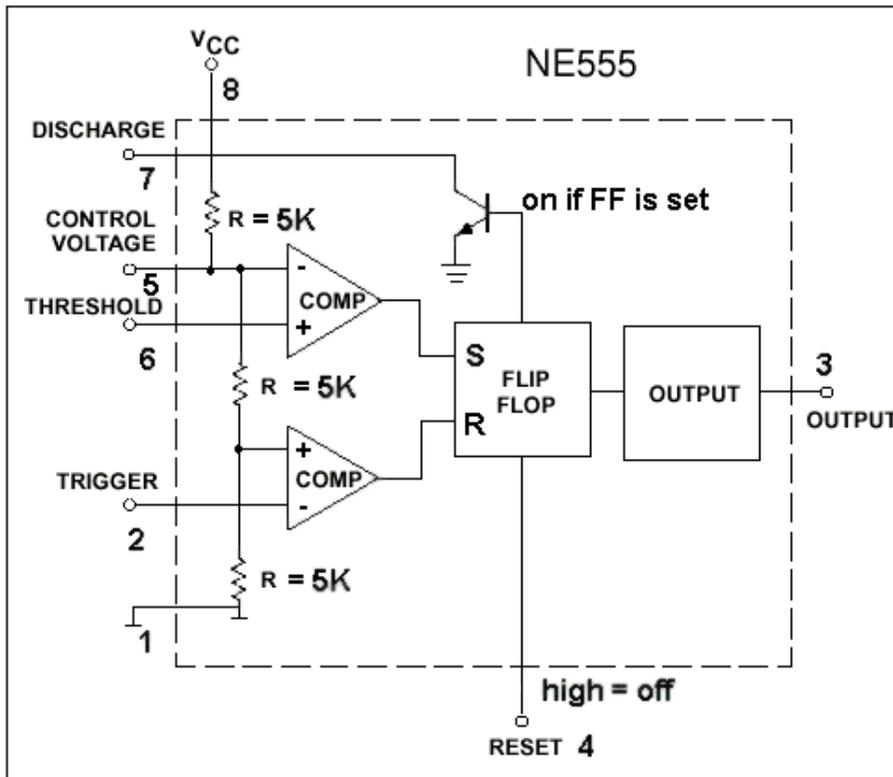
Je ne donne pas de liste de composants dans cet article. Le nécessaire est visible sur le schéma ci-dessus (N'oubliez pas le connecteur DB9 pour la ligne série). Vous pouvez aussi utiliser tout type de diode, 1N4148 par exemple. Personnellement, je crois que le relais 500 Ohm est plus adapté puisque vous n'avez besoin ni de R4, ni de condensateur 2000uF (ou 2200uF). Un condensateur plus petit (1000uF) est suffisant pour C1.

Note : Vous utiliserez une LED rouge dans le circuit 120 Ohm et une LED verte pour le relais 500 Ohm. Ce n'est pas une plaisanterie ! Le voltage supporté par une LED verte est plus élevé que celui supporté par une LED rouge.

Le plan de la carte, les fichiers Eagle et postscript pour sa gravure sont inclus dans le logiciel proposé en téléchargement à la fin de cet article. Le logiciel de CAD (Computer Aided Design) Eagle version Linux est disponible sur [cadssoftusa.com](http://cadssoftusa.com).

## Comment ça marche ?

Le coeur du circuit du chien-de-garde est une puce timer NE555. Cette puce contient 2 comparateurs, une bascule Flipflop et 3 résistances de 5KOhm chacune, qui servent de référence aux comparateurs. Chaque fois que la broche nommée seuil (6) descend sous les 2/3 de la valeur de l'alimentation, la bascule est activée (état on).



Regardons maintenant le schéma du circuit : le signal RTS de la ligne série est utilisé comme tension d'alimentation. Selon la norme RS232 cette tension est de +/- 10V, c'est pourquoi une diode est nécessaire avant le condensateur C1. Ce condensateur C1 se charge très rapidement et sert à stocker l'énergie nécessaire à l'activation temporaire du relais. Le condensateur C2 se charge très lentement au travers de la résistance de 4.7MOhms. Le transistor T1 décharge le condensateur C2 lorsqu'il reçoit une courte impulsion par la broche DTR de la prise RS232. Si cette impulsion ne lui parvient pas (parce que l'ordinateur s'est arrêté) le condensateur C2 peut-être chargé (il faut environ 40 secondes) au-delà des 2/3 de la tension d'alimentation et le FlipFlop est mis en position "on".

Le condensateur C1, la résistance R2, la LED et le relais doivent avoir des valeurs telles que l'énergie fournie par C1 soit suffisante pour activer le relais mais insuffisante pour le garder activé en permanence. Le but est que l'on presse le bouton "reset" seulement une ou deux secondes.

La LED restera allumée jusqu'à ce que le serveur ait terminé son redémarrage.

Comme vous le voyez sur le schéma, il existe également un bouton d'arrêt connecté à la broche CD. Si vous le maintenez enfoncé un court moment (15 sec.) le pilote logiciel lancera la commande "shutdown -h now" pour arrêter le serveur. Ceci concerne la maintenance normale et n'a aucun lien avec le

fonctionnement du chien-de-garde.

## Le pilote logiciel

C'est un court programme en langage C qui peut être lancé par les scripts du répertoire /etc/init.d/. Il active en permanence le signal RTS de la prise RS232 puis envoie une impulsion sur DTR toutes les 12 secondes (la valeur "timeout" de ce chien-de-garde est de 40 secondes). Lors d'un arrêt normal de votre ordinateur le programme va couper le signal RTS et envoyer une dernière impulsion sur DTR. Ainsi le condensateur d'alimentation (C1) sera déchargé avant que l'on dépasse le délai du "timeout". Le chien-de-garde ne réagit donc pas lors d'opérations normales. Pour installer ce pilote téléchargez le fichier linuxwd-0.3.tar.gz depuis la page de téléchargement. Puis décompressez-le et tapez :

```
make
```

pour le compiler. Copiez l'exécutable linuxwd obtenu dans /usr/sbin/linuxwd. Editez le script linuxwd\_rc (pour redhat/mandrake, ou linuxwd\_rc\_anydist pour une autre distribution) pour indiquer le port RS232 sur lequel est connecté le circuit (ttyS1=COM2 ou ttyS0=COM1). Copiez alors le script dans

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

et dans

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

C'est tout.

## Tests

Lorsque vous avez fini de souder tous les composants, vous devez tester votre circuit avant de le connecter à l'ordinateur. Connectez la broche prévue pour être reliée ultérieurement à la ligne RTS du port série, à une source continue de 9-10V et attendez 40-50 secondes. Vous devriez entendre le clic du relais qui s'enclenche et voir la LED s'allumer. Le relais ne doit pas rester collé. La LED ne s'éteindra que lorsque vous brancherez la ligne qui va vers DTR à +10V.

Lorsque cette vérification est terminée vous pouvez brancher le circuit sur votre ordinateur. Le programme linuxwd propose un mode de test dans lequel il effectue des sorties sur imprimante et un certain nombre d'arrêts pour simuler des arrêts système. La commande

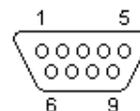
```
linuxwd -t /dev/ttyS0
```

lance le pilote dans ce mode (avec /dev/ttyS1 si le circuit est branché sur COM2).

## Installation matérielle

Le brochage de l'interface RS232 est le suivant :

9 broches D-SUB MALE sur l'ordinateur.



connecteur 9 broches	connecteur 25 broches	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

Connecter le circuit à la prise RS232 doit être très simple. Pour connecter la ligne 'reset' du CPU au relais, vous devez repérer les fils qui vont au bouton 'reset' de votre ordinateur. Connectez le relais du circuit en parallèle au bouton 'reset'.

## Conclusion

Un tel chien-de-garde n'apporte pas une fiabilité à 100% mais il donne un niveau supplémentaire de sécurité. Un problème peut survenir si la vérification du système de fichiers ne se termine pas après un 'reset' matériel. Les nouveaux systèmes de fichiers journalisés peuvent améliorer les choses, mais je ne les ai pas encore essayés. Le chien-de-garde présenté ici n'est pas cher, n'est pas trop complexe à construire et pratiquement aussi efficace que beaucoup de produits du commerce.

## Références

- Le pilote linuxwd : Téléchargement de logiciel
- Fiche technique de la puce NE555 NE555.pdf 140K

Site Web maintenu par l'équipe d'édition LinuxFocus © Guido Socher "some rights reserved" see <a href="http://linuxfocus.org/license/">linuxfocus.org/license/</a> <a href="http://www.LinuxFocus.org">http://www.LinuxFocus.org</a>	Translation information: en --> -- : Guido Socher (homepage) en --> fr: Paul Delannoy (homepage)
--	--