

RaspberryPi et synchronisation horaire GPS

Comme on en avait parlé sur un fil sur le Raspi, voila l'avancée des recherches. J'en profite pour vous expliquer ce que j'ai compris de la gestion du temps sur les raspi, n'hésitez pas à corriger, compléter ou poser des questions sur les points qui vous sont obscurs, je ferai de mon mieux pour compléter !

<https://www.hisse-et-oh.com/articles/2395-raspberrypi-et-synchronisation-horaire-gps>

Quelques détails sur la gestion de l'heure sur le RaspberryPi

Le RaspberryPi ne dispose pas d'une horloge matérielle, comprendre par là qu'il est incapable de conserver l'heure lorsqu'il est éteint, contrairement à un PC traditionnel. L'astuce employée par les développeurs de raspbian (la distribution linux/debian majoritairement employée sur les raspberry) est d'utiliser un script qui s'appelle fakehwclock, qui se contente d'enregistrer la dernière heure connue lorsque le système s'éteint et de s'en servir comme heure de référence au redémarrage. C'est très pratique d'un point de vue système (ça permet tout plein de checks notamment sur les systèmes de fichiers), mais ça ne fait absolument pas l'affaire lorsqu'on aime avoir une machine à l'heure (ce qui est bien pratique, voire le strict minimum si on s'en sert de traceur).

Pour pallier ce problème, plusieurs solutions:

- lorsque le Raspi est connecté à un réseau, il peut utiliser des serveurs de temps pour synchroniser son horloge. Si le raspi a accès à internet, c'est encore plus simple, puisque des serveurs de temps sont déjà configurés et le système se synchronise automatiquement dessus. Problème, sur le bateau, on n'a pas forcément internet.

- on peut rajouter une horloge matérielle au raspi. Ça s'appelle une RTC (Real Time Clock), ça ne vaut que quelques euros, et ça se présente sous la forme d'un petit circuit intégré et d'une petite pile (pour garder l'heure quand le raspi est éteint), et ça se connecte aux GPIOs (les deux rangées de contacts électroniques sur le bord de la carte). C'est une très bonne solution, d'autant qu'elle ne coûte pratiquement rien et que c'est "relativement" facile à mettre en place. Problème cependant : en l'absence d'une source de temps "fiable" pour l'actualiser, ces horloges (de faible qualité, mais vu le prix, peut-on vraiment s'en plaindre?) ont tendance à dériver. La dérive est plus ou moins importante suivant la calibration initiale de la puce, la précision du quartz qui l'anime, et surtout la température, mais elle est surtout tout à fait imprévisible... Il faut donc accepter de devoir remettre l'horloge du système à jour régulièrement, ou de connecter le tout à internet pour que ce soit fait automatiquement.

- on peut se servir d'une source de temps externe pour synchroniser l'heure du système. Dans le cas présent, un GPS. Ce n'est pas une solution parfaite, en ce qu'elle nécessite un GPS (!) ,

et surtout, que le GPS ait réussi à capter au moins un satellite. Le système ne sera donc pas à l'heure tant que le GPS n'aura pas au moins un satellite de capturé (en réalité, ça prend un peu plus de temps, j'y reviens tout de suite !). De plus, la précision de l'heure est médiocre (de l'ordre de la seconde) par rapport à une synchronisation réseau, même si dans notre cas, c'est plus de précision qu'il n'en faut.

Nous y voilà donc.

Installer GPSD

Pour le cas présent, j'ai choisi d'utiliser GPSD. Pour faire simple, disons que gpsd est une couche d'abstraction pour GPS. Il se connecte au(x) gps présent(s) sur le système, interprète les données et les rend disponibles pour qui (les applications) en a besoin.

Allons-y donc pour l'installation de gpsd !

Dans un terminal, tapez:

```
sudo apt-get update && sudo apt-get install gpsd gpsd-clients
```

répondez oui aux questions, laissez mouliner, et bientôt, vous devriez avoir gpsd d'installé.

Il faut maintenant le configurer. Pour cela, tapez:

```
sudo leafpad /etc/default/gpsd
```

cela devrait vous ouvrir un éditeur de texte avec le contenu du fichier de configuration. Voici le contenu du fichier:

```
# Default settings for gpsd.  
# Please do not edit this file directly - use `dpkg-reconfigure gpsd' to  
# change the options.  
START_DAEMON="true"  
GPSD_OPTIONS="-n"  
DEVICES="/dev/ttyS0"  
USBAUTO="true"
```

Petite explication du contenu : START_DAEMON=true signifie que vous voulez que GPSD se lance comme service système (et c'est le cas). GPSD_OPTIONS="-n" signifie que vous voulez que le GPS cherche des satellites dès son lancement, c'est très important pour synchroniser l'heure. DEVICES="/dev/ttyS0" est spécifique (dans une certaine mesure) au système de chacun et doit pointer sur le GPS. Le mien est branché sur le port série, d'où le /dev/ttyS0. S'il est en usb, ce sera probablement /dev/ttyUSB0. USBAUTO="true" est de moindre importance.

A partir de là, il est conseillé de tester que gpsd fonctionne en le lançant avec la commande :

```
sudo service gpsd restart
```

puis, une fois qu'il est lancé,

```
gpsmon
```

ce qui ouvrira un outil de gestion du GPS qui indique les GPS captés, les coordonnées GPS si ce dernier a réussi à la calculer, etc... Si à ce niveau là ça ne marche pas, recommencez étape par étape ou criez à l'aide, ce n'est pas normal. Si tout va bien et que *gpsmon* vous indique l'heure et votre emplacement, bravo, vous avez le droit de passer à l'étape d'après !

Installer NTP

NTP , ou plutôt *ntp* en minuscules, est l'outil qui permet sous linux de synchroniser l'horloge système avec les horloges distantes (sur le réseau). C'est un outil obscur mais relativement puissant qui va nous permettre de régler automatiquement l'heure système.

Allons-y donc pour son installation:

```
sudo apt-get install ntp
```

Ensuite, sa configuration:

```
sudo leafpad /etc/ntp.conf
```

Le contenu de ce fichier est assez inintéressant (on y trouve les serveurs par défaut de synchro horaire, entre autres) , aussi allez directement à la fin et rajoutez:

```
#GPS TIME SOURCE
server 127.127.28.0 minpoll 4 maxpoll 4
fudge 127.127.28.0 refid GPS
```

En très gros, 127.127.28.0 est une IP spéciale qui fait comprendre à NTP qu'il doit aller piocher les informations horaires dans la mémoire système partagée. En l'occurrence dans la mémoire partagée de GPSD (qui, si vous avez suivi, centralise les informations gps, dont l'heure et la date !). On sauve et on quitte.

On relance *ntp*:

```
sudo service ntp restart
```

On vérifie que notre modification a bien été prise en compte avec la commande:

```
ntpq -p
```

qui devrait afficher une ligne commençant par:

```
SHM(0) .GPS.
```

Une fois encore, n'allez pas plus loin si vous n'arrivez pas au moins à ça.

A ce stade, normalement, vous avez GPSD qui tourne dans un coin et qui vous fournit l'heure (vous l'avez testé avec *gpsmon* et ça marchait, rappelez-vous) et maintenant, vous avez *ntp* qui fonctionne aussi et qui lit la mémoire de *gpsd*. On peut s'en assurer en regardant la colonne "reach" de la commande précédente (*ntpq -p*). Si celle-ci comporte un chiffre supérieur à 0, c'est que tout fonctionne. Si elle affiche 0, patientez un peu (le temps de rafraîchissement est d'environ 16 secondes) et réessayez dans quelques minutes. Si c'est toujours 0, criez à l'aide.

Automatiser le tout

La configuration de raspbian (et de debian de façon générale) est fourbe sur ce point. En effet, par défaut, GPSD est lancé au démarrage, mais "dormant". Il utilise un système qui lui permet de ne se lancer que si un logiciel en fait la demande. C'est très économe en ressources pour la machine, mais dans notre cas, c'est contraignant : nous n'aurons aucune donnée GPS (et donc pas l'heure) tant qu'aucune application n'aura fait appel à gpsd. Pas un problème me direz-vous puisque NTP se sert de GPSD. Oui, mais non. Il n'utilise pas directement GPSD, mais sa mémoire partagée, ce qui signifie qu'il ne "lance" pas le service lui-même, mais se contente de regarder discrètement ce que GPSD fait. Du coup, tant que vous n'avez pas lancé opencpn, gpsmon ou un autre, point de synchronisation horaire.

La solution nous vient du site du créateur de gpsd

(<http://www.catb.org/gpsd/troubleshooting.html> section Troubleshooting Start at Boot) :

Il faut créer un nouveau fichier de configuration pour gpsd pour le forcer à se lancer au démarrage, même si personne ne fait appel à lui :

```
sudo leafpad /etc/systemd/system/gpsd.service
```

Et le contenu essentiel dudit fichier :

```
[Unit]
Description=GPS (Global Positioning System) Daemon
Requires=gpsd.socket
# Needed with chrony SOCK refclock
After=chronyd.service

[Service]
EnvironmentFile=-/etc/default/gpsd
EnvironmentFile=-/etc/sysconfig/gpsd
ExecStart=/usr/sbin/gpsd -N $GPSD_OPTIONS $DEVICES

[Install]
WantedBy=multi-user.target
Also=gpsd.socket
```

Je ne vais pas m'amuser à vous décrire les lignes une par une, mais sachez que la ligne vraiment importante, c'est `ExecStart`.

On recharge les services avec la nouvelle configuration:

```
sudo systemctl daemon-reload
sudo systemctl reenale gpsd.service
```

On redémarre son RaspberryPi, et normalement, tout le monde fonctionne ! Vous pouvez tester successivement en lançant `gpsmon` et `ntpq -p` pour voir que tout est comme à l'étape 2.

Et voilà, votre système se synchronise tout seul sur l'heure donnée par le GPS !!