

Voici le mien..

16 Mai 2020



Pourquoi?

Parce-que c'est rigolo, parce-que j'avais un minitel dans un coin, parce-que d'autres l'ont fait, parce-que dans les années 80 j'utilisais beaucoup le minitel, parce-que une console série pour GNU/Linux c'est le b.a.-ba, et surtout parce-que c'est inutile donc totalement indispensable!

Sur le web on peut trouver de nombreux exemples dont je me suis largement inspiré. Le site incontournable est le site [d'Alexandre Montaron](#) qui est l'auteur de mntl.ti, le fichier terminfo utilisé . Faites-y un tour si le sujet vous intéresse. Sur sa page web il donne aussi 13 liens (à ce jour) vers des réalisations similaires, dont l'autre site incontournable qui est le [blog de Pyla](#).

Je ne vais pas décrire ici tout le détail, pour cela je vous renvoie aux sites listés ci-dessus. Je vais surtout parler que de ce qui est spécifique à ma réalisation.

La démarche que j'ai suivi:

Je veux obtenir un Pi 100% opérationnel, full Raspbian, accessible en ssh, vnc et qui soit une console sur le minitel en cas d'ultime secours (par ex: plus de réseau).

Je ne souhaite pas utiliser un adaptateur "USB to serial" pour dialoguer avec le Minitel comme de nombreuses autres réalisations l'ont fait mais utiliser le port serie natif du Pi. Je veux aussi intégrer physiquement le Pi dans le Minitel pour avoir un tout homogène.

C'est avec un Minitel 1B que j'ai fait ce projet, sans doute le modèle le plus répandu.



Attention spécifique Raspberry Pi3

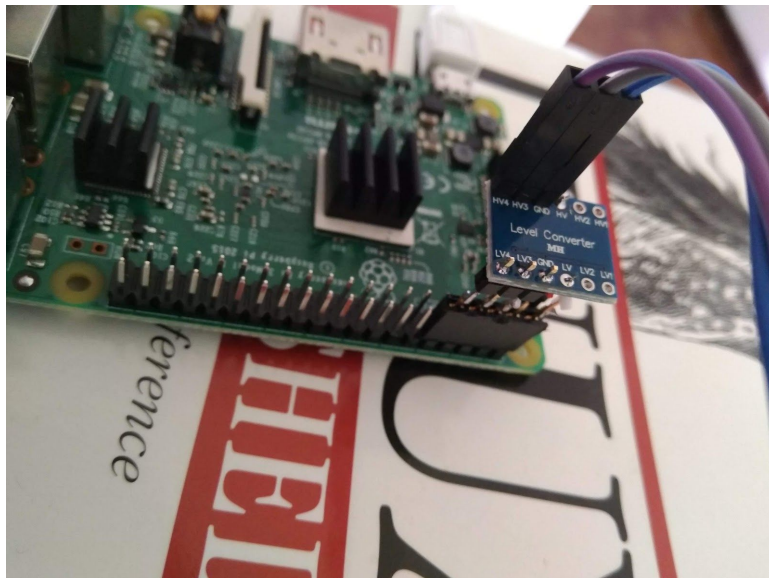
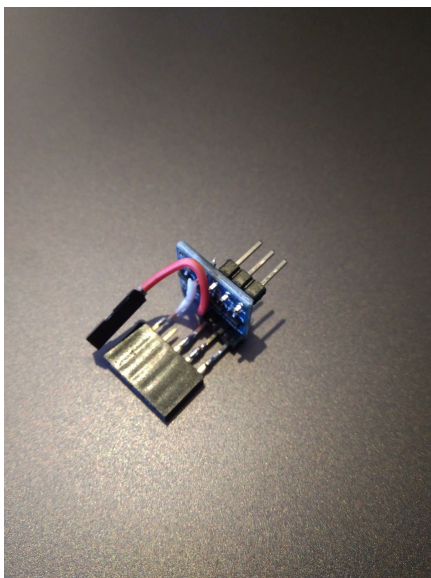


J'ai utilisé un Pi3, problème sur le Pi3 de base ttyAMA0 est utilisé pour le bluetooth et c'est le mini-uart qui est dispo sur les pins 8 et 10 pour la console série. Or le mini-uart a plusieurs limitations comme par exemple: pas de parité. Le Minitel utilise une parité paire c'est donc bloquant, j'ai essayé sans succès. Il faut donc changer la config pour avoir ttyAMA0 sur serial0 [comme expliqué ici](#) ça se résume à ces 2 lignes dans /boot/config.txt:

```
enable_uart=1  
dtoverlay=pi3-disable-bt
```

Faisable aussi depuis raspi-config, voir la doc.

Donc à ce stade nous avons ttyAMA0 dispo sur les pins 8 et 10 (gpio 14 et 15). Comme expliqué de partout le Pi est en 3.3V et le Minitel en 5V j'ai fait le choix d'utiliser un petit circuit convertisseur "level shifter" [comme celui ci](#) que j'ai monté sur le Pi de cette façon:



Fiche DIN du Minitel

Raspberry Pi	Module Level Shifter	Minitel
Pin 1 : 3.3V	LV	
Pin 2 : 5V	HV	
Pin 6 : Gnd	Gnd	Fiche Din Pin 2
Pin 8 : Tx	LV3	
Pin 10 : Rx	LV4	
	HV3	Fiche Din Pin 1
	HV4	Fiche Din Pin 3

Sur le Minitel j'ai soudé les fils sur la face intérieur de la fiche DIN pour que rien ne soit visible de l'extérieur et laisser la prise accessible au cas où. Sur une Raspbian de base il y a : **console=serial0** dans **/boot/cmdline.txt**, il faut changer pour **console=serial0,4800e7** pour dire 4800 bauds, parité paire (even) et 7 bits de data. Bien sur il faut aussi passer le Minitel en mode 80 colonnes, désactiver l'écho local et passer à 4800 bauds, soit:

Fonction T puis A, Fonction T puis E, Fonction P puis 4

A ce stade si l'on boot le Pi on a tous les messages de boot qui s'affiche sur le Minitel. A noter que bien sur cela ralentit le boot car c'est seulement à 4800 bauds. Mais pour 1980 c'est du rapide ;-) Il est toujours possible d'ajouter "quiet" dans **/boot/cmdline.txt**. Mais pour ma part j'aime bien avoir les messages du kernel au boot et puis Minitel c'est rétro :-)

Raspbian dispose de fichiers terminfo pour Minitel: **ls -l /usr/share/terminfo/m/minit*** pour les voir. Pour pouvoir se logger correctement il faut configurer systemd, il faut modifier le fichier **/lib/systemd/system/serial-getty@.service** la ligne **ExecStart** devient:

ExecStart=-/sbin/getty -i -L -l "\033\143" %i 4800 minitel1b-80

Maintenant on peut se logger, mais le login prompt est moche, et tous les messages de systemd de démarrage des services n'apparaissent pas, il y a seulement une colonne blanche sur la droite de l'écran. Pour faire mieux on va télécharger le dernier fichier terminfo d'Alexandre Montaron, le compiler avec tic et l'installer dans **/etc/terminfo/**

wget canal.chez.com/mntl.ti; tic -o /etc/terminfo/ mntl.ti

Et aussi pour avoir un meilleur prompt son fichier prompt 7 bits, que l'on range dans /etc/

wget canal.chez.com/issue.minitelG; mv issue.minitelG /etc/

Pour prendre en compte le dernier terminfo et le login prompt. La ligne **ExecStart** de **/lib/systemd/system/serial-getty@.service** devient:

ExecStart=-/sbin/getty -f /etc/issue.minitelG -L -l "\033\143" %i 4800 m1b-x80

Personnellement je n'ai pas fouillé et pas vu de différence entre minitel1b-80 et m1b-x80 peut-être en verrais-je à l'usage.

Il restait encore les messages des services de systemd qui polluent l'écran avec la colonne blanche sur la droite pour les supprimer il faut ajouter à **/boot/cmdline.txt**

rd.systemd.show_status=false

mais ça ne suffit pas, sur Raspbian plymouth force systemd à les afficher je l'ai donc masqué:

systemctl mask plymouth-start.service

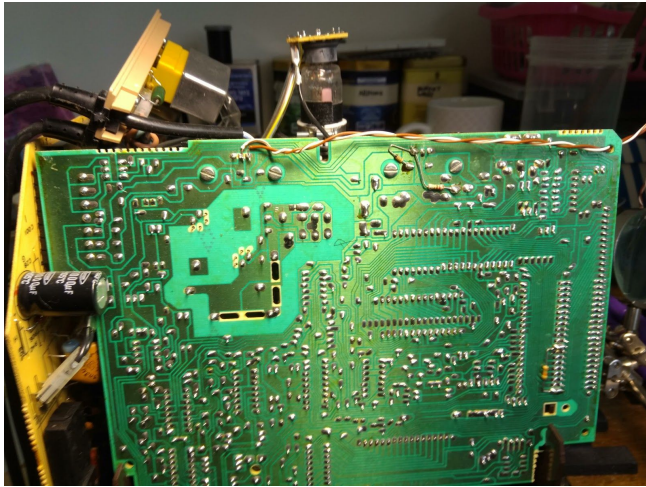
Maintenant ça y est j'ai ce que je voulais un Pi qui boot proprement sur un Minitel!

Pour info si comme moi vous utilisez gpm pour avoir la souris en console, il faudra désactiver ce service qui de toute façon ne fonctionne pas sur la console Minitel et qui de plus gêne le fonctionnement de MidnightCommander (mc).

Il y aussi le fichier inputrc pour bash proposé par Alexandre Montaron sur son site qui ajoute les touches de fonction, c'est encore un petit plus bien sympa.

Et maintenant un peu de hardware

Mon idée de départ était d'utiliser le 5V du minitel pour alimenter le Pi. Cela n'a pas été possible, pas assez de puissance! J'ai essayé de soutirer le 5V à la sortie du régulateur:



Ca marche mais le Pi détecte des sous-tensions. Je n'aime pas ça ce n'est pas propre:

```
moi@raspi:~$ zgrep -i volt /var/log/kern.log*
```

```
/var/log/kern.log.1:May 14 14:07:05 raspi kernel: [ 38.717340] Under-voltage detected! (0x00050005)
```

```
/var/log/kern.log.1:May 14 14:07:29 raspi kernel: [ 63.757297] Voltage normalised (0x00000000)
```

D'autre part je veux pouvoir mettre le Pi en et hors tension indépendamment du Minitel. Si j'avais pu utiliser ce 5V interne Minitel j'aurais de toute façon mis un interrupteur pour le Pi ne serait ce que pour pouvoir passer les commandes Fcn T,A Fcn T,E Fcn P, 4 au Minitel avant de booter le Pi. Bon je re-essayerais à partir du 8.5V-1A dispo sur la broche 5 de la fiche DIN du Minitel. Je n'ai pas de régulateur 5V sous la main pour le moment mais en théorie je devrais pouvoir tirer 1.7A (8.5W/5V). A tester. Donc pour le moment je reste avec une alim séparée pour le Pi. Mais en fait cette solution à aussi l'avantage de pouvoir laisser le Pi, avec ses fonctions serveur, sous tension 24/7 et le Minitel hors tension. Le Minitel consomme 140mA sous 240V que l'écran soit en veille ou pas c'est donc non négligeable face au Pi qui consomme entre 7 et 4 fois moins suivant son activité.

Il y a de la place pour installer le Pi proprement dans le Minitel. J'ai utilisé la petite plaque démontable au dessus de la prise péri-informatique. Je l'ai découpée et j'ai installé le Pi de manière à laisser accessible le port réseau et les connecteurs USB.



J'ai même suffisamment de place, si besoin, pour passer un câble audio et/ou hdmi sur les connecteurs qui sont à l'intérieur. Le Pi peut sortir facilement la plaque est juste clipsée.

Et enfin pour preuve de concept de l'utilité d'un tel montage j'ai fait un NAS avec [OpenMediaVault](#) et un disque USB externe. Ca marche plutôt bien et en cas de gros pépin la console Minitel serait là comme ultime secours.



That's all folks!