

Le module transceiver VHF DRA818V

Cet article est dans la continuité d'aider à débiter dans la « bidouille » avec ce module très séduisant. On le trouve tout monté par exemple sur le site de SV1AFN pour une prix très modique de la trentaine d'euros.

Mais reste à trouver avec lui un moyen de dialoguer avec son port série pour le configurer.

Mon souhait, toujours dans des configurations minimalistes, est d'utiliser de raspberry pi zero, à la place ou en complément des arduinos. Utiliser un Pi4 me semble bien du gaspillage ici. Mais que ce soit avec un pi zero ou un autre modèle de Pi, je me suis tout de suite planté sur l'utilisation des outils les plus évidents. Ainsi minicom ne fonctionne pas avec ce module. La raison est certainement qu'il ne gère pas correctement des octets série en temps réels du module DRA818V qui gère certainement sa liason série d'une manière très rudimentaire. Avec le pi zero, on reste en mode console avec la version légère de raspbian, c'est à dire sans mode graphique.

Sur Internet, je me suis aperçu bien sur que je n'étais pas le premier à m'être planté avec minicom. La solution est un script en python qui fonctionne.

Le but ici est de configurer ce module, rien d'autre.

Ce module se branche sans autre composant sur le port GPIO du raspberry. On prend soin de brancher l'alimentation du module sur la broche 1 du GPIO qui est en 3,3 V, et on branche de même en direct en croisé les fils Tx et Rx du module et du pizero. Une LED s'allume sur le module quand on met sous tension le raspberry.

1- Configurer une carte SD avec Raspbian:

A ce stade, si ce pi zero doit servir pour travailler avec ce module comme avec APRX ou tout autre utilisation en AX25, mieux vaut partir d'une image « neuve ». Je ne décris pas ici la manip trop bien connue.

Mais veiller à faire des mises à jours de raspbian, c'est quand même la chose minimale à faire.

2 – Installer python et ses librairies:

Ici, on installe python en 3.7, mais la version 2.7 s'installe aussi pour garantir une compatibilité des scripts plus anciens.

Surtout on a besoin de la librairie « python-serial », puisqu'il s'agit d'entrées-sorties séries. Elle est en version python 2.7 seulement.

Utiliser la commande : `apt-get install python-serial`

ou

`aptitude install python-serial.`

3- Le script en python:

```
#!/usr/bin/env python2.7import time
import serial
ser= serial.Serial(port='/dev/ttyAMA0',
baudrate = 9600,
parity=serial.PARITY_NONE,
bytesize=serial.EIGHTBITS,
stopbits=serial.STOPBITS_ONE,
)
commande=raw_input(« Entrez votre commande: »)
print commande
ser.write(commande+ »\r\n »)
x=ser.readline()
print x
```

Faire un `chmod +x` de ce script pour le rendre exécutable.

Pour lancer le script, faire sous root donc avec `sudo` si l'utilisateur root n'est pas validé:

Allezr dans le répertoire où réside le script, et faire:

```
$ sudo ./nom_du_script
```

A l'exécution, on vous demande d'entrer la commande à envoyer au module. Par exemple si on tape:

```
AT+DMOCONNECT
```

le module répond:

```
+DMOCONNECT:0
```

S'il y a un problème, on obtient:

```
+DMOERROR
```

3 – Configurer le module;

Maintenant, on va configurer à minima le module. Pour cela, il faudra saisir dans la commande demandée:

```
AT+DMOSETGROUP=GBW,TFV, RFV,Tx_CTCSS,SQ,Rx_CTCSS
```

Ci-dessus, après le signe « = », ce sont des valeurs numériques qu'il faut entrer.

Par exemple on veut utiliser ce module en APRS donc la fréquence 144,800 Mhz

- GBW est la largeur de bande et des canaux. « 0 » configure en 12,5 khz, et 1 en 25 khz
- TFV est la fréquence en émission, attention, il faut 4 chiffres après le point;
- RFV de même pour la fréquence en réception;
- Tx_CTCSS est la fréquence en hertz du signal CTCSS éventuel. la valeur 00000 enlève le CTCSS en émission;
- SQ est la valeur du squelch, la valeur « 0 » enlève le squelch, c'est à dire ouvre entièrement le récepteur;
- Rx_CTCSS est la valeur à recevoir du CTCSS, la valeur 00000 enlève la réception du CTCSS.

Donc la commande exacte à envoyer est:

```
AT+DMOSETUPGROUP=0,144.8000,144.8000,0000,4,0000
```

- « 0 » indique une bande passante de 12,5 khz;
- 144.8000 la valeur de la fréquence en émission et en réception;
- « 0000 » enlève l'émission du CTCSS, il faut 4 chiffres;
- « 4 » est la valeur du squelch;
- « 0000 » enlève le CTCSS en réception, il faut 4 chiffres.

On a ainsi configuré le module, la notice technique donne la manière du PTT etc.

Cette notice donne aussi la façon de faire varier le volume du

signal BF de sortie en réception.

C'est tout pour cet article qui n'a d'autre prétention de vous permettre de bricoler avec ce module pour l'usage que vous voudrez.

Remarque :

Ce script est très facilement adaptable aux modules TX et RX de chez Radiometrix, la parité chez ces bidules est toujours 8 bits, 1 ou 2 stop bits (une émission avec 2 stop bits est lue par une réception configurée avec 1 stop bits). Le changement de vitesse est évident à la lecture du script.

Il est vrai que les raspberry peuvent poser problème quand on veut dialoguer avec beaucoup de modules, python permet d'y remédier.

MariaDB dans Debian

Cet article a pour but de dépanner les OMs qui peuvent trouver des difficultés avec mariadb qui remplace MySQL dans debian depuis pas mal de temps. Mariadb est totalement compatible avec mysql, c'est un fork depuis que MySQL a été racheté, de manière à rester totalement libre, car MySQL depuis son rachat ne peut pas être considéré comme tel.

Nonobstant une petite modification, donc mariadb reste compatible pour tous les site web qui utilisent mysql, mais il y a une petite modification dans l'installation qui peut dérouter pas mal de webmasters amateurs, à savoir l'invalidation du root mysql.

Vous avez certainement remarqué que sous l'utilisateur root système, le client console mysql, par la commande mysql, se lance sans mot de passe avec l'utilisateur root mysql, ce qui rend invalide certains installateurs de sites PHP par exemple. C'est totalement déroutant.

De même, depuis debian 9 au moins, phpmysqladmin n'est plus dans les dépôts debian. On peut le remplacer par le package « adminer ». Ce package bien sur a besoin de l'utilisateur root mysql.

1 – Configurer root mysql:

Sur votre serveur debian ou raspbian (pour un serveur raspbian, on préconise de donner un mot de passe à root et d'enlever l'utilisateur pi), se connecter en root. Surtout pour un serveur, un raspberry ne doit pas garder l'utilisateur pi, car cela est une faille de sécurité inadmissible pour un appareil de toute manière connecté à internet.

Puis lancer la commande: # mysql_secure_installation par la suite, répondre « Y » à toutes les propositions. Cela a pour effet de rendre possible l'utilisation de l'utilisateur root mysql et d'enlever l'utilisateur anonyme qui existe avec l'installation de mariadb.

Enfin, lancer le client mysql toujours sous root: # mysql

Vous êtes donc connecté encore en anonyme mariadb sans mot de passe à root.

Faire:

```
mariadb > SET PASSWORD FOR 'root'@'localhost' =  
PASSWORD('votre_mot_de_passe');
```

```
mariadb > flush privileges;
```

Vous pouvez taper quit pour sortir du client mysql.

Vous pouvez enfin vous connecter dans mysql avec l'utilisateur root.

2- Installer adminer:

Avec apt-get ou aptitude installer adminer.

Aller dans /usr/share/adminer sous utilisateur root;

faire:

```
# ./compile.php
```

```
# mv adminer_*.php adminer.php (le caractère * ici est à  
remplacer par la version de adminer obtenue).
```

```
# a2enconf adminer.conf
```

```
puis: # systemctl reload apache2
```

A partir de là, en vous connectant avec:

```
http://votre_serveur/adminer.php
```

Vous arriver à la page de login de « adminer ». Vous vous logez naturellement et vous faites vos manip sur votre ou vos bases de données.

Cours linux et raspberry

Vous avez au moins tous entendus parler des raspberry, et même la plupart d'entre vous en avez au moins un chez lui. Dans beaucoup de stations il y a de plus en plus de ces petits ordinateurs, l'APRS et APRX en sont l'exemple le plus fréquent. Mais bien d'autres usages sont à la portée de ces minis ordinateurs. Son port GPIO est conçu pour permettre la gestion de signaux TTL de toutes sortes, en entrée comme en sortie. En serveurs, ils sont aussi irremplaçables, et en radio numériques, ils sont de plus en plus indispensables.

L'ARALA propose à ses adhérents des formations sur linux Raspbian, formations linux donc très orientées raspberry. Plusieurs niveaux sont proposés: débutants et perfectionnement.

Le niveau débutant, permet d'apprendre à récupérer une image de carte SD, et de la copier sur une carte SD soit à partir d'un PC windows microsoft, soit d'un PX sous linux, puis d'utiliser la commande « raspi-config » c'est à dire par elle, de configurer un raspberry pour le rendre opérationnel,

Une description des différents modèles de raspberry, avec leur usages respectifs le plus adapté.

L'approche du mode graphique et du mode console (affichage de type windows ou de type DOS).

La prise en main du raspberry pour les commandes linux les plus courantes, et une formation initiale à sa configuration réseau.

Au niveau perfectionnement:

- entre autres, formations sur:
- réseaux TCP/IP;
- serveur apache;
- serveur d'imprimantes, et de scanner;
- réseau AX25 pour l'APRS avec direwolf;
- et tout sujet demandé par les intéressés.

Ces formations resteront à des niveaux raisonnables, sans prétention à être au niveau professionnel. En effet, un serveur personnel et un serveur professionnel, ce n'est pas la même chose, de même que les réseaux. Mais un minimum de connaissances reste à la portée d'un OM, et cela lui permet de comprendre les articles de technique informatique de radioamateurs qui sont de plus en plus fréquents, et de les réaliser.

Pour s'inscrire:

Bien sur rien ne commencera avant la fin du confinement covid 19. Pour cela, si vous êtes intéressés par ces formations, envoyez un e-mail à:

f6daa@ganmon.fr

73+ à tous.

Enlever l'utilisateur pi d'un raspberry

Ce court article est destiné aux OMs qui utilisent cet ordinateur minuscule qui est connecté sur des réseaux divers. Le nombre de raspberry en service plus ou moins permanents

commence à devenir considérable, surtout chez les radioamateurs.

Raspbian, dérivé de Debian est particulièrement fiable, mais sa plus grosse faille de sécurité est quand même que dans trop de cas, ce mini PC travaille sous l'utilisateur « pi ». Maintenant, quand vous passez par « raspi-config » pour l'initialiser à votre cas particulier, il vous est demandé de changer le mot de passe de l'utilisateur « pi ». C'est déjà ça, mais c'est quand même très insuffisant si l'on en reste là.

Quelle est la faille?

Déjà, les raspberry sont faciles à détecter dans un réseau, en raison de leur carte réseau:

Par exemple, vous avez peut-être remarqué qu'avec nmap, quand vous scannez un réseau LAN, vous obtenez aussi l'adresse ethernet dite « MAC adress » (sans rapport avec Mac Intosh). Or la première partie de ces adresses ethernet est répertoriée par fabricants de cartes réseau. Donc le Pi1, Pi2, Pi3 et Pi4 maintenant, ont leur adresse ethernet qui répond à cette convention. Il est donc aisé de détecter d'éventuels raspberrys sur un LAN.

Pour se connecter à distance sur un raspberry, on utilise un client SSH. En dehors de Raspbian et d'OS comme Ubuntu et autres, l'utilisateur root est connectable avec un mot de passe. Pour Raspbian et Ubuntu, l'utilisateur root est invalidé en accès direct, les droits étendus se font avec un utilisateur particulier et qui est présent par défaut, avec la

commande « sudo ». Si dans Ubuntu, lors de son installation, le premier utilisateur entré sera l'utilisateur sudo, avec les raspberry ce rôle est d'office attribué à l'utilisateur « pi ». On voit que Raspbian a ce défaut de l'utilisateur pi connu de tous avec son mot de passe « raspberry » par défaut.

Il n'est pas rare de voir des raspberry qui restent configurés avec l'utilisateur « pi » et son mot de passe par défaut: « raspberry ». Une première mesure prise par les développeurs de Raspbian, a été de demander dès l'ouverture initiale de la commande « raspi-config », de changer le mot de passe de « pi ». C'est quand même un peu mince comme mesure de sécurité.

En effet, un hacker n'a pas dans le cas de Raspbian à deviner le nom de l'utilisateur aux droits de « root ». Il sait que c'est « pi ». Il reste seulement à trouver son mot de passe. Une fois fait, le pirate a tous les droits dans le raspberry. Comme indiqué précédemment, en SSH on ne peut pas se connecter directement avec comme nom d'utilisateur « root » pour un linux debian, redhat etc. Il faut d'abord se connecter avec un nom d'utilisateur ordinaire. Ce n'est qu'une fois qu'on est connecté sur la console via SSH que cet utilisateur ordinaire passera en « root ». Donc avec un utilisateur linux ordinaire, il y a les difficultés:

- trouver un nom d'utilisateur qui existe dans le système à pénétrer;
- trouver son mot de passe;
- enfin trouver le mot de passe de root.

Avec Raspbian, le seul défi est de trouver le mot de passe de « pi ».

Donner un mot de passe à « root »:

Pour cela, quand on est connecté au raspberry par l'utilisateur « pi », il faut entrer un mot de passe dans « root ». De ce fait, on rend accessible « root » en connexion. Alors que sans mot de passe, Raspbian est configuré pour ne pas être accessible.

Pour donner un mot de passe à « root »:

```
$ sudo passwd root
```

Vous entrez donc le mot de passe de « root »

Vous créez un nouvel utilisateur:

```
$ sudo adduser toto
```

Vous entrez son mot de passe.

Vous supprimez l'utilisateur « pi »:

Avec la commande:

```
$ exit
```

Vous vous déconnectez. Puis vous vous connectez avec l'utilisateur « root ».

Sous « root » vous faites:

```
# deluser pi
```

Cela efface l'utilisateur « pi », puis vous effacer son répertoire:

```
# rm -Rf /home/pi
```

Il faut effectivement effacer ce répertoire pour enlever toute trace de « pi ».

Voilà, l'utilisateur « pi » n'existe plus, et l'utilisateur « root » est disponible. La commande « sudo » n'est plus à utiliser et n'est plus utilisable. Pour certaines commandes qui ont besoin d'être exécutées avec les droits de « root », il faut passer sous l'utilisateur « root ».