

Mise en œuvre d'une liaison série RS232

Objectifs :

Valider une liaison série RS232 entre deux ETTD dans les environnements Linux et Windows.

Moyens : PC sous Windows et Linux, travail en binôme.

Important :**Vous devez :**

☞ **Notez sur votre compte-rendu personnel :**

- **tous les éléments importants du TP ;**
- **Les réponses aux questions posées ;**
- **Les problèmes que vous avez rencontrés ;**
- **La solution que vous avez apportée éventuellement à chacun des problèmes rencontrés.**

Présentation :

La liaison série RS232 reste présente dans certains équipements de communication (switches, routeurs, systèmes embarqués, ...). Elle permet un échange de données de type « caractère par caractères » de façon très simple entre deux équipements (peer to peer).

Pas de trame, pas de réseau, transmission en bande de base sans codage ... on est dans le fondamental. C'est la genèse de la communication, dont les principes s'appliquent à toutes les liaisons de type « série » créées ensuite (usb, lan, ...).

Vocabulaire :

ETTD : Equipement terminal de transmission de données : un PC, un Terminal Wyse.

ETCD : Equipement terminal de circuit de données : un modem RTC.

Full Duplex : communication dans les deux sens simultanément.

PARTIE 1 : Validation d'une liaison RS232

On dispose d'un ordinateur équipé d'un port série.

Le système d'exploitation nous permet déjà de tester le bon fonctionnement de ce port série.

L'objectif est de valider la partie technique de l'installation.

Identifiez le(s) connecteur(s) série disponible(s) sur votre machine :

Il s'agit généralement d'une prise SUB-DB9 (9 broches).

La norme ayant pas mal évolué, vous rencontrerez peut être sur certains dinosaures des SUB-DB25(25 broches) plus anciennes...

Femelles ou mâles ? La norme est floue à ce sujet. Par prudence, il faut consulter la notice du fabricant. Pour un PC, ce sera toujours une DB9 mâle.

Le câble de liaison doit respecter un principe : la sortie (TXD) d'un ETTD va vers l'entrée de l'autre (RXD). Si les 2 ETTD sont identiques, il faudra donc un câble croisé.

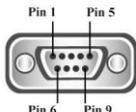
Au minimum, une liaison rs232 bi-directionnelle n'a besoin que de 3 broches : RX, TX, GND. Les autres broches sont des informations de contrôle utilisées à l'origine pour dialoguer avec des ETCD (modems).

Une liaison rs232 se caractérise par des paramètres de communication :

Vitesse (en baud), format de la donnée (7 ou 8 bits), paramètres de synchronisation (start/stop bit), paramètre de contrôle de cohérence (parité), paramètre de gestion de flux (DTR/ Xon-Xoff).

☞ 2 appareils reliés doivent évidemment avoir la même configuration pour bien se comprendre.

RS232

Pin 1	DCD	
Pin 2	RXD	
Pin 3	TXD	
Pin 4	DTR	
Pin 5	GND	
Pin 6	DSR	
Pin 7	RTS	
Pin 8	CTS	
Pin 9	RI	

1-1 Paramétrer l'écran Wyse (si vous n'avez pas de Wyse, consultez la note à la fin du §) :

Un terminal Wyse est un appareil assez simple qui affiche sur son écran les caractères reçus par un de ses ports série (par sa broche RXD), et qui envoie ce qu'on tape sur son clavier sur ce même port série (sa broche TXD). C'est l'équivalent d'un logiciel comme HyperTerminal, Tera Term, puTTY, ...

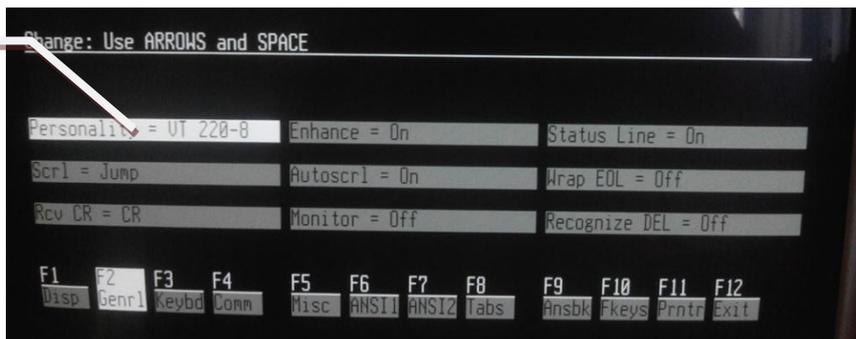
Simple ne veut pas dire qu'il n'y a rien à faire. Il faut adapter le terminal à l'ETTD auquel il est relié. Outre les paramètres de communication, il y a toute une série de paramètres propres aux terminaux et qui concernent sa compatibilité d'affichage (Jeu de caractères, émulation, ...).

Pour configurer l'écran Wyse : accédez au menu par <SHIFT> <SELECT> sur le clavier. Explorez les menus avec les touches de fonction, les flèches et la barre d'espace.

Reproduisez les réglages suivants :

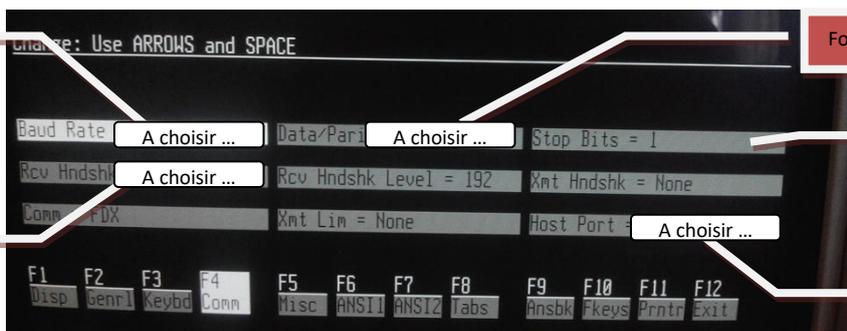


Emulation



Vitesse

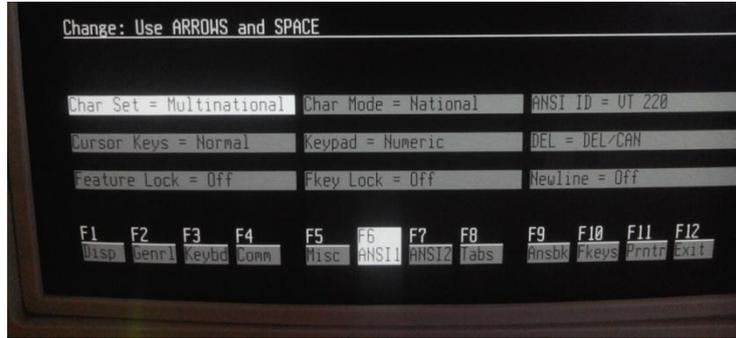
Format/Contrôle données



Synchronisation

Contrôle de flux

Choix du connecteur



Pour sortir et enregistrer : F12, puis <ESPACE> pour indiquer Yes à la question posée, puis F12.

NOTE : Si vous ne disposez pas de terminal Wyse, vous pouvez relier 2 PC ensemble. Celui qui jouera le rôle du Wyse utilisera (sous Windows) un logiciel comme TeraTerm, puTTY, ou (sous Linux), CuteCom (interface graphique) ou MiniCom (mode texte).

Ce logiciel doit aussi être paramétré (COM, vitesse, flux ...).

1-2 Tests dans l'environnement Windows

Les ports séries se nomment généralement COM1, COM2, ...
La commande **mode** permet de les configurer.

Site à consulter (ou l'aide en ligne de la commande) :

<http://www.computerhope.com/modehelp.htm>

- a) Afficher le réglage actuel du port COM choisi (Voir le Gestionnaire de périphérique pour connaître le numéro de COM présent sur votre machine)
- b) Modifier les paramètres du port COM comme suit : 19200baud, 8bits, pas de parité. Vérifiez que le réglage est bien appliqué.

Connexion des deux ETTDs :

- c) Vérifiez le câble de connexion entre votre port COM et le terminal Wyse. Configurez le terminal de la même façon que le COM du PC.
- d) Envoyer des caractères du PC vers le terminal :
Utilisez la redirection. Ex : DIR > COM1
- e) Utilisez un logiciel comme TeraTerm, terminal, ou PuttY pour échanger en full duplex.
- f) Avec un multimètre (Ohm-mètre ou testeur de continuité), établissez le schéma du câble.

1-3 Tests dans l'environnement Linux

Les ports séries (COM1, COM2, ...) se nomment généralement /dev/ttyS0, /dev/ttyS1, ...

Si vous utilisez un adaptateur USB, vous pouvez connaître le nom du port série en affichant les derniers messages du système avec la commande : `dmesg`

La commande **stty** permet de configurer un port série.

Site à consulter (ou l'aide en ligne de la commande) :

<http://pwet.fr/man/linux/commandes/stty>

- a) Afficher le réglage actuel du port COM choisi (Ex. pour le COM1 : `stty -a < /dev/ttyS0`)
- b) Utilisez les paramètres suivants : 19200baud, 8bits, pas de parité. Vérifiez que le réglage est bien appliqué (ex : `stty 19200 < /dev/ttyS0`)

Connexion des deux ETTDs :

- c) Utilisez le câble fourni pour la connexion entre votre port COM et le terminal Wyse (ou le 2^e PC).
- d) Envoyer des caractères du PC vers le terminal pour un test rapide :
Utilisez la redirection. Ex : `ls -l > /dev/ttyS0`
- e) Utilisez un logiciel CuteCom ou MiniCom pour effectuer des tests en Full Duplex.
- f) Avec un oscilloscope, visualisez la forme du signal.

1-4 Mise en évidence d'une erreur de configuration

Modifiez le réglage de la transmission sur un seul des ETTD : 9600 bauds.
Constatez l'effet sur la communication.

PARTIE 2: Exemple d'utilisation d'une liaison RS232 sous Linux : Une deuxième console.**Petite information sur les dessous de Linux :**

Un bref séjour sur Wikipédia vous en apprendra un peu plus sur les origines d'Unix et de Linux, et notamment des normalisations¹ adoptées au fil du temps.

Linux suit actuellement les recommandations de la norme POSIX et utilise le mécanisme **SystemD**² pour gérer tous les services (*daemons*) à lancer au démarrage (SystemD utilise lui-même un *daemon* appelé *systemd*).

Notion de RUN-LEVEL (Niveau d'exécution)

Linux possède plusieurs modes de fonctionnement : monoposte, multiposte, avec réseau, avec interface graphique...

Ils sont repérés par un numéro :

- 0 : arrêt
- 1 : monoposte (maintenance)
- 2 : multipostes sans réseau
- 3 : multipostes avec réseau
- 5 : multipostes réseau et interface graphique
- 6 : reboot

La commande **init** permet de passer d'un niveau à l'autre.

Généralement, on démarre en mode 3 (serveurs) ou 5 (postes de travail).

A chaque *runlevel* correspond une liste de services à démarrer.

Gestion manuelle des services :

Pour établir la liste des services (*daemons*) à démarrer, on utilise des instructions³ dont la principale est **systemctl**.

Systemctl ou service ?

Vous connaissez sans doute la commande linux **service** (start, stop, status, reload) qui permet de gérer certains services qui sont répertoriés dans le dossier `/etc/init.d` Mais cela ne modifie pas la liste des services à démarrer au lancement de la machine ; c'est juste une action temporaire.

Systemctl permet de faire comme **service** mais son but principal est de déterminer quel service doit être démarré à chaque lancement de la machine (mots clés *enable* et *disable*)

Il utilise le dossier : `/etc/systemd/system` qui contient des liens (regroupés dans des sous-dossiers) vers des *UNITS*, c'est-à-dire des fichiers de configuration des services.

Ces fichiers *unit* sont à l'origine dans le dossier : `/lib/systemd/system` (ou `/usr/lib/systemd/system` sur les dernières versions). Il faut les copier dans `/etc/systemd/system`, les modifier selon nos besoins, et utiliser la commande **systemctl** pour les activer.

Bien sûr, on ne le fera pas nous même : on utilisera la commande **systemctl** comme indiqué ci-après.

¹ http://fr.wikipedia.org/wiki/UNIX_System_V

² <http://fr.wikipedia.org/wiki/Systemd>

³ <https://doc.opensuse.org/documentation/leap/reference/html/book.opensuse.reference/cha.systemd.html>

Travail : Ajouter un service de login sur un terminal en mode texte sur port Série

Après avoir lu ce qui précède, voici la description du travail à effectuer :

Le fichier *unit* correspondant à ce service est **serial-getty@.service**

Comme ce fichier est générique, il faut **lui indiquer** le nom du port série à utiliser. Il créera ensuite le module de démarrage.

Quel est le nom du port série que vous utilisez sur votre Linux ?

Admettons que se soit le pour `/dev/ttyS0` :

```
systemctl enable serial-getty@ttyS0.service
```

Note : Le `ttyS0` positionné après le `@` sera récupéré comme paramètre `%I` dans le fichier de configuration.

Il faut ensuite **activer la modification au démarrage** :

```
systemctl start serial-getty@ttyS0.service
```

Le “start” peut être remplacé par “stop”, “status”, comme avec la commande *service*. Mais l’action est permanente.

Pour un terminal Wyse, paramétrer le mode vt220-8 (Menu F2, *Personality*)

Si vous avez correctement fait le travail, vous devriez avoir le login sur le Wyse ou sur le terminal.

A LA FIN DU TP : Remettez la configuration du port COM dans l’état où vous l’avez trouvé.

Réf : <http://Opointer.de/blog/projects/serial-console.html>