

Platine Terminal AXEL

AX3000 modèles TCP/IP

*Mise en œuvre sous
UNIX*

La reproduction et la traduction de ce manuel, ou d'une partie de ce manuel, sont interdites. Pour tout renseignement complémentaire, s'adresser à :



Zone d'activité d'Orsay-Courtabœuf
16 Avenue du Québec
BP 728
91962 LES ULIS Cedex
Tél. : 01.69.28.27.27
Fax : 01.69.28.82.04

Les informations contenues dans ce document ne sont données qu'à titre indicatif ; elles peuvent être modifiées sans préavis. AXEL ne peut, en aucun cas, être tenu responsable des erreurs qui pourraient s'y être glissées.

© - 1994-1997 - AXEL - Tous droits réservés

1 - PRESENTATION	1
1.1 - GENERALITES.....	2
Liaison asynchrone.....	2
Liaison réseau.....	3
Liaisons logiques multiples.....	3
1.2 - FONCTIONNALITES.....	4
1.3 - EXEMPLE DE CONFIGURATION.....	5
2 - INSTALLATION ET CONFIGURATION	7
2.1 - PRESENTATION DU SET-UP TCP/IP.....	8
2.2 - DECLARATION DE L'ENVIRONNEMENT RESEAU.....	10
2.2.1 - Adresse Ethernet.....	11
2.2.2 - Adresse IP et protocole RARP.....	11
2.2.3 - Gestion des serveurs.....	13
2.2.4 - Gestion des routeurs.....	13
2.3 - PARAMETRAGE DU MULTI-SESSION.....	14
2.3.1 - Paramétrage des sessions.....	15
2.3.2 - Ligne status.....	15
2.3.3 - Association Vue/Serveur.....	15
2.4 - PARAMETRAGE DU PROTOCOLE.....	16
2.4.1 - Option 'mss'.....	17
2.4.2 - Option 'window'.....	17
2.4.3 - Option 'ttl'.....	18
2.4.4 - Option 'noise'.....	18
2.4.5 - Option 'tcp port'.....	19
2.5 - STATISTIQUES.....	20
2.5.1 - Option 'Ethernet'.....	20
2.5.2 - Option 'ARP'.....	21
2.5.3 - Option 'TCP Client'.....	21
2.5.3 - Option 'TCP Server'.....	21
2.6 - PARAMETRAGE DES EMULATIONS.....	22
Exemple 1.....	24
Exemple 2.....	25
3 - CONFIGURATION DU SYSTEME UNIX	27
3.1 - DECLARATIONS RESEAU.....	28
3.2 - DECLARATIONS GENERALES.....	29
3.2.1 - Généralités.....	29
3.2.2 - La variable TERM.....	29
3.2.3 - La gestion scancode.....	30
3.3 - TUNING DU SYSTEME D'EXPLOITATION.....	32

3.3.1 - UNIX SCO 3.2v4	32
3.4 - SERVEUR RARP	33
4 - UTILISATION DU MULTI-SESSION.....	35
4.1 - UTILISATION AVEC LES ASSOCIATIONS VUE/SERVEUR.....	36
4.2 - UTILISATION PAR LE SET-UP	38
4.2.1 - Ouvrir une session.....	39
4.2.2 - Changement de session	39
4.2.3 - Fermer une session	40
5 - LES SERVEURS D'IMPRESSION STANDARD.....	41
5.1 - CONFIGURATION DE LA PLATINE	42
5.1.1 - Choix du service d'impression	43
5.1.2 - Configuration du port AUX1 ou AUX2	44
5.1.3 - Sortie du set-up	44
5.2 - CONFIGURATION DU SYSTEME UNIX.....	45
5.2.1 - Utilisation de lpd.....	45
5.3.1 - Utilisation de rcmd.....	47
6 - LE SERVEUR DE TERMINAUX	49
6.1 - SET-UP TERMINAL	50
6.1.1 - Choix du service associé	50
6.1.2 - Configuration du port AUX1 ou AUX2	51
6.1.3 - Sortie du set-up	52
6.2 - SET-UP TCP/IP	52
7 - LE SERVEUR AXEL DE VOIES SERIE.....	55
7.1 - INTRODUCTION	56
7.2 - CONFIGURATION DE LA PLATINE	57
7.2.1 - Set-up Terminal.....	58
7.2.2 - Set-up TCP/IP	59
7.2.3 - A la mise sous tension.....	60
7.3 - CONFIGURATION SOUS UNIX	60
7.3.1 - Installation.....	60
7.3.2 - Mise en œuvre et utilisation.....	61
7.3.3 - Désinstallation	63
8 - LES SERVEURS RESEAU PROPRIETAIRES AXEL	65
8.1 - INTRODUCTION	66
8.1.1 - le concept.....	66
8.1.2 - Trois serveurs : printd, ttyd et rprint	67
8.1.3 - Device : pseudo-terminal ou named pipe ?	68
8.1.4 - Notion de Port TCP	68

8.1.5 - En résumé.....	69
8.1.6 - Précaution d'utilisation.....	72
8.1.7 - Mise en œuvre.....	72
8.2 - CONFIGURATION DE LA PLATINE.....	73
8.2.1 - Choix du service.....	73
8.2.2 - Configuration du port AUX1 ou AUX2.....	74
8.2.3 - Sortie du set-up.....	75
8.3 - CONFIGURATION DU SYSTEME UNIX.....	75
8.3.1 - Association Platine/Device.....	76
8.3.2 - IBM AIX - Association Platine/Imprimante.....	78
9 - LE MULTI-SHELL.....	81
9.1 - INSTALLATION SOUS UNIX.....	83
9.2 - PARAMETRAGE DE LA PLATINE.....	83
9.3 - UTILISATION.....	84
9.3.1 - Lancement.....	84
9.3.2 - Création et changement de vue.....	85
9.3.3 - Suppression de vue.....	85
9.3.4 - Sortie du logiciel.....	85
9.4 - LES OPTIONS.....	86
9.4.1 - Option b : choix des pttys.....	86
9.4.2 - Option c : bordure d'écran.....	86
9.4.3 - Option h : aide en ligne.....	87
9.4.4 - Option L : ligne status sur la 26 ^{ème} ligne.....	88
9.4.5 - Option l : ligne status sur la 25 ^{ème} ligne.....	88
9.4.6 - Option m : mode muet.....	90
9.4.7 - Option n : nombre de vues.....	90
9.4.8 - Option p : préaffectation des pseudo-terminaux.....	91
9.4.9 - Option S : choix du shell.....	91
9.4.10 - Option x : buffer circulaire.....	92
9.5 - LIMITATIONS DE LA NORME ANSI.....	93
10 - CONFIGURATION A DISTANCE.....	94
10.1 - INTRODUCTION.....	95
10.2 - CONFIGURATION D'UNE PLATINE.....	96
10.3 - ACQUISITION D'UNE CONFIGURATION.....	97
10.4 - MESSAGES D'ERREUR.....	98
10.5 - FORMAT DU FICHER DE CONFIGURATION.....	102
Entête.....	103
Paramètres 'Ethernet'.....	104
Paramètres 'TCP/IP'.....	104
Paramètres 'Ports AUX1 et AUX2'.....	105
Paramètres 'Port parallèle'.....	107

Paramètres 'Multi-session'	107
Paramètres 'Terminal'	107
Paramètres 'Session'	108
Fin de fichier	111
11 - TELECHARGEMENT DU FIRMWARE	113
11.1 - RAPPEL SUR LA COMMANDE RCMD.....	114
11.2 - TELECHARGEMENT PAR TFTP SEUL	115
11.2.1 - Administration à distance.....	115
11.2.2 - Exécution depuis le set-up.....	117
11.3 - TELECHARGEMENT PAR BOOTP ET TFTP.....	118
11.3.1 - Administration à distance.....	120
11.3.2 - Exécution depuis le set-up.....	121
11.4 - PROBLEMES POSSIBLES ET SOLUTIONS.....	122
11.4.1 - Phase bootp	122
11.4.2 - Phase tftp.....	122

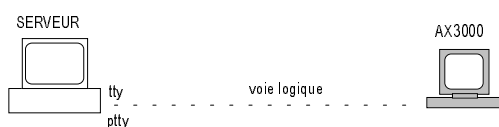
**- 1 -
PRESENTATION**

Ce chapitre présente le concept d'une liaison Terminal sous UNIX et les principales fonctionnalités de la Platine Terminal AXEL Ethernet TCP/IP.

1.1 - GENERALITES

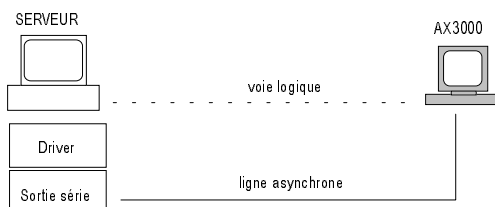
Sous Unix, la liaison entre un serveur et un terminal est réalisée au travers d'un device appelé **tty** (ou **ptty** pour les pseudo-terminals).

Un device tty ou pty permet à l'application d'échanger des données avec un terminal. L'application ne connaît pas le support physique utilisé (ligne asynchrone ou synchrone) pour transporter les données. Le device tty ou pty représente donc une voie logique entre le serveur et le terminal :



Liaison asynchrone

Le schéma suivant présente le principe d'une liaison asynchrone :

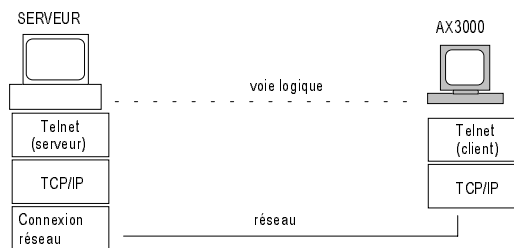


Du côté serveur, la liaison asynchrone est réalisée au travers d'un 'driver' gérant une carte Entrée/Sortie série (COM1, carte multivoie...). Un driver de ce type gère autant de devices tty que de ports série disponibles sur une même carte (un device par voie).

Le terminal possède une électronique lui permettant d'envoyer et de recevoir des caractères sur une ligne asynchrone.

Liaison réseau

Le schéma suivant illustre la communication entre une Platine Terminal AXEL et un Serveur au travers d'un réseau TCP/IP Ethernet :

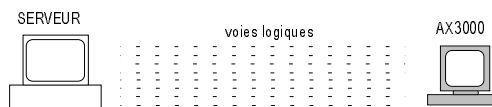


Du côté serveur, la liaison est réalisée au travers d'un 'package' réseau gérant une carte Ethernet selon les protocoles Telnet (serveur) et TCP/IP. Les devices auxquels sont attachés les terminaux connectés au serveur sont des pseudo-terminaux (pty).

Le terminal AXEL possède également une carte Ethernet intégrant les protocoles Telnet (client) et TCP/IP.

Liaisons logiques multiples

Le schéma suivant présente la notion de liaisons multiples :



Une des caractéristiques essentielle de la Platine, est la possibilité d'avoir HUIT voies logiques pour une SEULE voie physique. Cette fonctionnalité, mise en œuvre au travers d'un utilitaire dans le cas d'une liaison asynchrone (mscreen ou axmscreen), est directement utilisable dans le cas d'une liaison réseau et permet jusqu'à huit connexions indépendantes sur des serveurs différents (multi-connexion).

1.2 - FONCTIONNALITES

Comme il a été présenté plus haut, l'un des points forts des Platines AX3000 TCP/IP est de fournir **huit terminaux virtuels** simultanés et totalement INDEPENDANTS.

Cette notion de terminaux virtuels permet aux Platines d'être multi-connexion. En effet, chaque terminal virtuel offre à l'utilisateur :

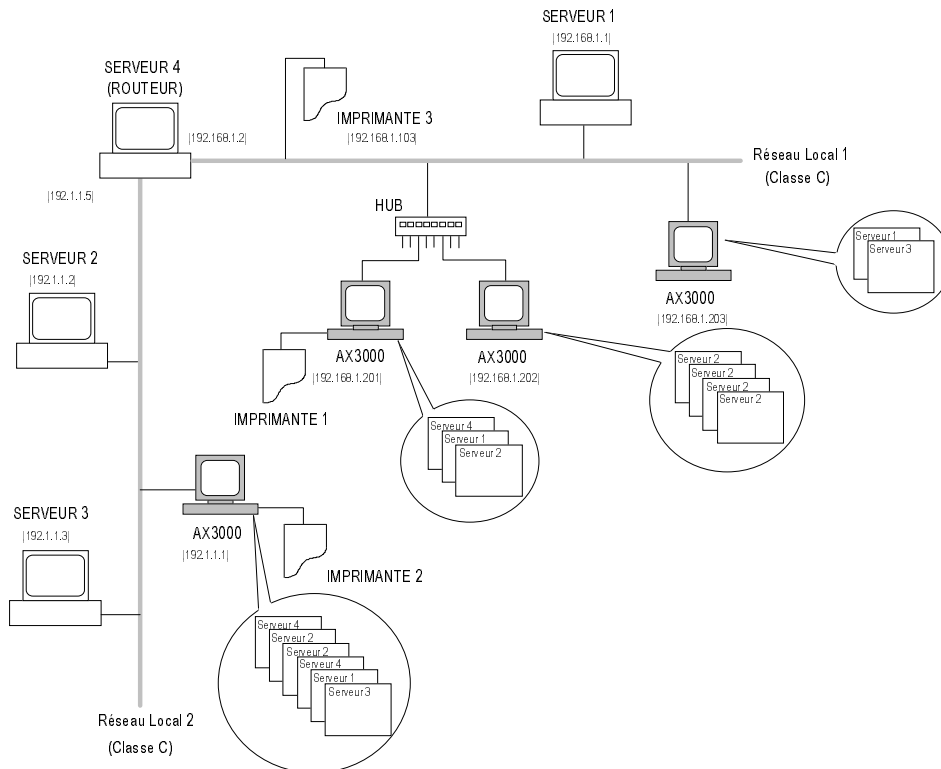
- une connexion à n'importe quel serveur du réseau et,
- des paramètres set-up propres (émulation, jeu de touches de fonctions, coloriage des attributs vidéo...).

De plus, il est possible, par déclaration dans le set-up, d'associer à chacun des huit terminaux virtuels (appelés aussi 'sessions' ou 'vues' dans la suite de ce document), l'un des serveurs du réseau, auquel se connectera automatiquement la Platine. Cette fonctionnalité offre l'avantage, non seulement d'éviter de lancer manuellement la procédure de connexion, mais aussi de figer, session par session, une configuration.

Le nombre de terminaux virtuels par Platine est paramétrable (de 1 à 8).

Les Platines AX3000 TCP/IP intègrent la fonction de **serveurs d'impression et de tty**. Cela signifie par exemple, qu'une imprimante connectée à un port auxiliaire (série ou parallèle) de la Platine est vue comme une **imprimante système** et est accessible à tous les utilisateurs du réseau.

1.3 - EXEMPLE DE CONFIGURATION



Ici, deux réseaux locaux sont représentés. Le serveur 1 appartient au réseau 1 (classe C), les serveurs 2 et 3 appartiennent au réseau 2 (classe C). Le serveur 4 appartient aux réseaux 1 et 2 et sert de routeur entre ceux-ci.

Les nombres indiqués entre parenthèses représentent les adresses IP de chaque élément du réseau.

Dans cet exemple, chacune des quatre Platines peut se connecter indifféremment et simultanément à un ou à plusieurs serveurs.

Les imprimantes 1 et 2, attachées directement aux Platines, sont accessibles à tous les utilisateurs du réseau au même titre que l'imprimante 3.

**- 2 -
INSTALLATION ET
CONFIGURATION**

Ce chapitre fournit les informations nécessaires à l'installation et la configuration des Platines AX3000 TCP/IP au sein d'un réseau Ethernet.

Les principales étapes de la configuration d'une Platine terminal sont :

- affectation d'une adresse IP compatible avec l'environnement réseau existant,
- déclaration de l'environnement réseau accessible à partir de cette Platine (serveurs telnet, routeurs...),
- association d'un serveur à chacune des sessions de la Platine,
- définition de l'environnement (émulation, touches de fonctions...) pour chacune des sessions disponibles sur la Platine.

Les trois premières étapes sont réalisées au travers du set-up TCP/IP décrit ci-dessous. La 4^{ème} étape s'effectue par le set-up Terminal.

Note : la suite de ce document ne concerne que l'installation et l'utilisation de la Platine AX3000 TCP/IP. Il est supposé que le package TCP/IP du système UNIX soit déjà installé et fonctionne correctement.

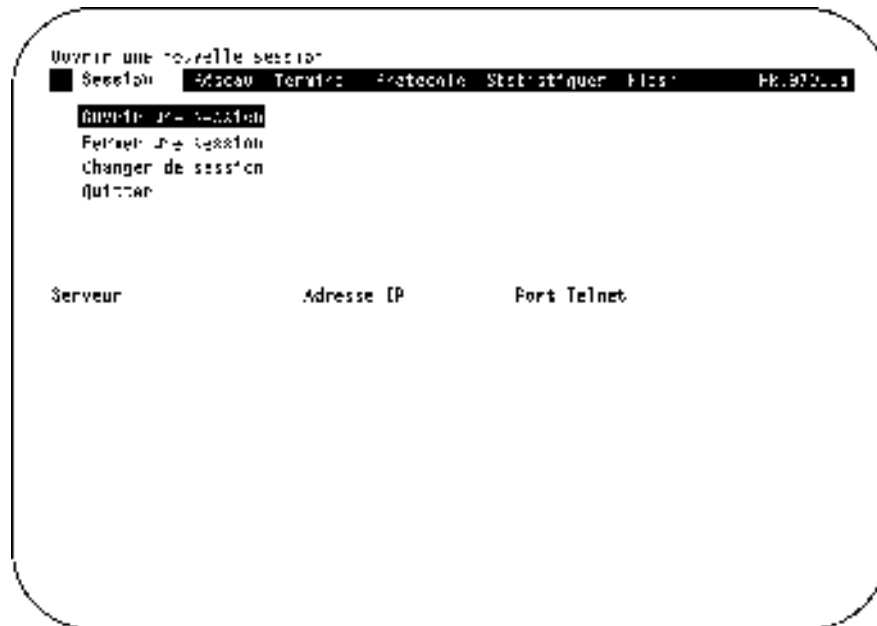
2.1 - PRESENTATION DU SET-UP TCP/IP

La combinaison de touches suivante permet d'accéder au set-up TCP/IP :



Note : utilisez la touche <*> du pavé numérique.

Le menu général du set-up TCP/IP est le suivant :



Il est présenté sous forme de menus horizontaux et déroulants. Les flèches de direction permettent le déplacement dans ces menus (des informations complémentaires associées à la sélection courante sont automatiquement affichées dans la partie inférieure de l'écran).

L'activation de la sélection courante s'effectue au moyen de **<RC>**.

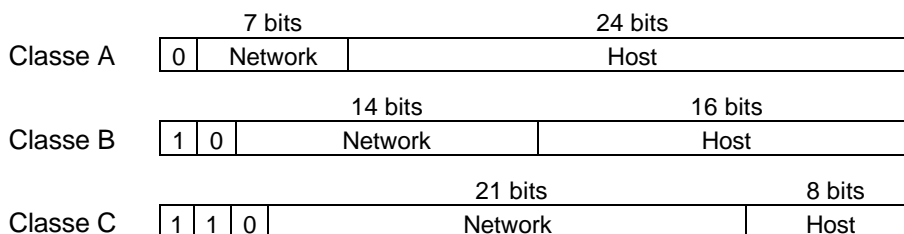
La sortie de ce menu s'effectue, soit au moyen de l'option **'Quitter'** du menu **'Session'**, soit au moyen de la touche **<Echap>**.

Les différentes options du menu **'Session'** seront détaillées au chapitre 4 (Utilisation du multi-session).

2.2 - DECLARATION DE L'ENVIRONNEMENT RESEAU

Tout périphérique connecté à un réseau Ethernet doit posséder une adresse IP **unique** dans le réseau. Cette adresse IP, toujours exprimée en décimal, est constituée de 4 octets séparés par des 'points' (caractère '.').

Pour mémoire, trois classes d'adresse IP sont disponibles :



Les 4 octets d'une adresse IP codent en fait deux adresses :

- une adresse Network (Réseau),
- et une adresse Hosts (Serveur).

ATTENTION : tous les périphériques attachés à un même réseau local doivent appartenir à la même classe et posséder la même adresse Network.

Exemple : pour accéder au serveur 192.168.1.40 (classes C : trois octets pour l'adresse Network), les trois premiers octets de l'adresse IP d'une Platine devront être également 192.168.1 (le quatrième octet étant différent de 40).

Le menu '**Réseau**' du set-up TCP/IP permet :

- de définir l'adresse IP de la Platine,
- de consulter l'adresse Ethernet de la Platine,
- d'entrer la liste des serveurs et des routeurs accessibles.

```
Ajouter serveur
Enlever serveur
Ajouter routeur
Enlever routeur
AX3000 IP
AX3000 Ethernet
```

2.2.1 - Adresse Ethernet

Tout périphérique réseau possède une adresse 'hardware' définie par le constructeur : l'adresse Ethernet.

L'adresse Ethernet, unique pour chaque Platine, n'est pas modifiable. Cette adresse, toujours exprimée en hexadécimal, est constituée de 6 octets séparés par des 'deux points' (caractère ':').

Le format de l'adresse Ethernet des Platinas est toujours le suivant :
00:A0:34:xx:xx:xx

L'adresse Ethernet de la Platine est affichée lors de la mise sous tension ou peut être consultée à tout moment en sélectionnant l'option '**AX3000 Ethernet**'.

2.2.2 - Adresse IP et protocole RARP

Tout périphérique réseau possède une adresse IP unique dans le réseau (voir définition en début de chapitre).

L'adresse IP de la Platine peut être définie :
- par le set-up TCP/IP au moyen de l'option '**AX3000 IP**',
- automatiquement à la mise sous tension de la Platine (protocole RARP).

Rappel sur le protocole RARP :

Le protocole RARP permet à un périphérique de récupérer son adresse IP à partir de son adresse Ethernet. Cette opération est effectuée par l'envoi d'une requête RARP à l'ensemble des composants réseau (broadcast). Si un composant réseau connaît l'adresse IP associée à l'adresse Ethernet du demandeur, il la communique à celui-ci. Cette affectation automatique de l'adresse IP est très utile pour la configuration à distance (voir chapitre 10).

Protocole RARP sur Unix :

Consultez le chapitre 3.4 pour permettre à une machine Unix d'être serveur RARP (et de communiquer les adresses IP des périphériques demandeurs).

Protocole RARP sur la Platine :

Si l'adresse IP d'une Platine n'est pas valide (i.e. égale à 0.0.0.0), le protocole RARP peut être lancé lors de l'installation de cette Platine. L'écran suivant apparaît à la mise sous tension :

```
PREMIERE MISE SOUS TENSION

Appuyez sur <Ctrl><Alt><*> pour entrer dans le set-up

ou

appuyez sur <Espace> pour rechercher l'adresse IP (protocole RARP)

Platine Terminal AX3000 TCP/IP
FK3.BV1.1a/TCP.FR.9701.a
00:A0:34:xx:xx:xx
```

Appuyez sur <Espace> pour lancer le protocole RARP. La Platine émet en broadcast 8 requêtes RARP espacées de 3 secondes.

Si un host répond à l'une de ces requêtes :

La phase RARP s'arrête, l'adresse IP de la Platine est automatiquement mise à jour et le message suivant est affiché :

Adresse IP trouvée : xxx.xxx.xxx.xxx

Appuyez sur une touche pour continuer.

Si aucune réponse n'est reçue pour ces 8 requêtes :

Le message Adresse IP non trouvée est affiché. L'adresse IP doit être saisie par l'intermédiaire du set-up TCP/IP (ou un serveur RARP doit être déclaré et la Platine éteinte puis allumée).

2.2.3 - Gestion des serveurs

Un serveur est une machine à laquelle la Platine Terminal va pouvoir se connecter au travers du réseau Ethernet via les protocoles TCP/IP et TELNET.

Un serveur au niveau de la Platine est identifié par trois paramètres :

- **Serveur** : le nom du serveur (chaîne alphanumérique commençant nécessairement par un caractère alphabétique).
- **Adresse IP** : l'adresse IP du serveur (respectant la syntaxe définie précédemment).
- **Port telnet** : le port TCP utilisé par le service telnet du serveur Unix. Par défaut ce port est le 23.

Sélectionnez l'option '**Ajouter serveur**' du set-up TCP/IP de la Platine et renseignez ces trois paramètres pour déclarer un serveur. A la différence du nom du serveur qui est local à la Platine (et donc éventuellement différent de celui du fichier `/etc/hosts`), il est indispensable que l'adresse IP saisie soit identique à celle du fichier `/etc/hosts`.

L'option '**Enlever serveur**' permet de supprimer un serveur. Il suffit de se placer sur la ligne du serveur à supprimer et de taper **<RC>**.

2.2.4 - Gestion des routeurs

Il est possible qu'un serveur ne soit pas directement accessible à la Platine. C'est le cas si la Platine et le serveur appartiennent à deux réseaux distincts (les deux réseaux étant alors reliés par un ou plusieurs routeurs). Il faut alors définir le routeur, **appartenant au même réseau que la Platine**, par lequel les trames vont transiter (entre la Platine et le serveur).

La déclaration d'un routeur s'effectue par l'option '**Ajouter routeur**'. Il est nécessaire de saisir les trois paramètres suivants :

- **Adresse IP routeur** : l'adresse IP du routeur qui appartient au même réseau local que la Platine.
- **Adresse IP distant** : adresse IP d'un des serveurs de ce nouveau réseau.
- **Masque** : masque logique de sélection sur l'**adresse IP distant**. La valeur par défaut de ce masque permet d'isoler la partie Network (par exemple 255.255.255.0 pour la classe C).

L'**adresse IP distant** peut être égale au mot clé '**default**'. Ainsi toutes les trames à destination d'un serveur non accessible par un des routeurs définis seront expédiées au routeur 'default'

Note : le routeur 'default' doit toujours être présent dans la liste des routeurs. Le masque associé est toujours 0.0.0.0.

Exemple à partir de la configuration du chapitre 1.3 : déclaration du routeur au niveau de la Platine d'adresse IP 192.168.1.201 :

```
Adresse IP routeur : 192.168.1.2 (le routeur)
Adresse IP distant : 192.1.1.3 (le serveur 3)
Masque             : 255.255.255.0 (Classe C)
```

L'option '**Enlever routeur**' permet de supprimer un routeur. Il suffit de se placer sur la ligne du routeur à supprimer et de taper **<RC>**.

2.3 - PARAMETRAGE DU MULTI-SESSION

La fonctionnalité de multi-connexion qu'offre l'AX3000 TCP/IP, permet jusqu'à huit accès simultanés répartis sur **un ou plusieurs** serveurs du réseau Ethernet.

La déclaration de ces paramètres (nombre de session et définition des associations Vue/Serveur) s'effectue au moyen du menu '**Terminal**' du set-up TCP/IP :

```
Session
Ligne status
Associer sessions
Associer AUX1
Associer AUX2
Dissocier
```

2.3.1 - Paramétrage des sessions

La Platine TCP/IP est capable de gérer jusqu'à 8 contextes d'écran. Cette ressource est à partager entre :

- un **nombre de sessions** (nombre d'accès simultanés à des serveurs du réseau),
- un **nombre de pages par session**.

Ainsi, il est possible d'avoir par exemple :

- 8 sessions avec 1 page par session
- 4 sessions avec 2 pages par session
- 3 sessions avec 1 page par session
- ...

Sélectionnez l'option '**Session**' et entrez le nombre de sessions puis le nombre de pages par session.

Note : il est impossible de modifier ces paramètres si une session est connectée.

2.3.2 - Ligne status

Ce paramètre est utilisé pour activer ou désactiver la ligne status.

Cette ligne status, située sur la 26^{ème} ligne, permet d'identifier la session active et de visualiser le nombre de sessions connectées.

Un label de 10 caractères maximum est affecté à chaque session connectée. Ce label contient soit le nom du host Unix sur lequel la session est connectée, soit une chaîne de caractères entrée depuis le set-up TCP/IP (voir ci-dessous).

Note: certains moniteurs ne supportent pas les fréquences utilisées pour une affichage en 26 lignes.

2.3.3 - Association Vue/Serveur

Il est possible d'associer à chacune des vues disponibles l'un des serveurs déclarés. Ainsi, lors du premier accès à une vue donnée, la connexion au serveur choisi est automatiquement établie. Ceci a le double avantage de ne

pas avoir à lancer manuellement la procédure de connexion, et de permettre à l'utilisateur de la Platine de 'figer', vue par vue, sa configuration.

Pour déclarer une association Vue/Serveur, sélectionnez l'option '**Associer sessions**'. Saisissez les paramètres suivants :

- **Session** le numéro de la session à associer (de 1 à 8).
- **Serveur** entrez le nom de l'un des hosts UNIX précédemment déclarés avec l'option '**Ajouter serveur**' du menu '**Réseau**' (cf. chapitre 2.2.3).
- **TERM** la valeur de la variable d'environnement TERM. Si aucune valeur n'est pas précisée la valeur par défaut est précisée au chapitre 3.2.
- **Label status** ce paramètre n'apparaît que si la ligne status est active (voir ci-dessus) et représente le label associé à cette session. Si ce champ est laissé vide, le label utilisé est le nom du host Unix associé.
- **1ère conn. auto.** '**o**' : lors de la mise sous tension de la Platine TCP/IP la connexion sera automatiquement ouverte.
'**n**' : il est nécessaire d'utiliser <Alt><Fx> pour ouvrir la connexion.
- **Reconn. auto.** '**o**' : après une déconnexion (<Ctrl><D>), une nouvelle connexion est immédiatement ouverte.
'**n**' : il est nécessaire d'utiliser <Alt><Fx> pour ouvrir une nouvelle connexion.

L'utilisation de la Platine ainsi paramétrée est décrite au chapitre 4.1.

L'option '**Dissocier**' supprime une association. Il suffit de se placer sur l'association à supprimer et de taper <RC>.

2.4 - PARAMETRAGE DU PROTOCOLE

Cinq paramètres permettent d'optimiser le fonctionnement du protocole TCP/IP selon les différents environnements d'utilisation.

Note : des valeurs par défaut sont associées à ces paramètres. Elles peuvent être modifiées, mais il est nécessaire pour cela d'avoir une bonne connaissance du protocole TCP/IP.

Le menu '**Protocole**' du set-up TCP/IP permet de visualiser ou de modifier ces paramètres :

```
mss
window
ttl
noise
TCP port
```

2.4.1 - Option 'mss'

Cette option est utilisée pour choisir la valeur du paramètre 'Maximum Segment Size'.

Ce paramètre définit la taille maximum (en octets) d'un segment de la couche TCP. La valeur par défaut est 512 octets.

Il est possible d'affecter une valeur quelconque à ce paramètre. Par contre, la taille maximum d'un 'datagram' de la couche IP (Maximum Transfert Unit) est fixée à 1200 octets. Il est donc inutile que le paramètre 'mss' ait une valeur supérieure à 1200.

Dans le cas où le nombre de trames rejetées est important, la valeur du paramètre 'mss' peut être diminuée.

La nouvelle valeur de ce paramètre sera prise en compte lors des prochaines connexions (inutile de mettre hors tension la Platine).

2.4.2 - Option 'window'

Ce paramètre définit la taille (en octets) de la fenêtre TCP. La valeur par défaut est 800 octets.

Cette valeur doit être supérieure à la valeur du paramètre 'mss'.

Dans le cas où le nombre de trames rejetées est important, la valeur du paramètre 'window' peut être diminuée.

La nouvelle valeur de ce paramètre sera prise en compte lors des prochaines connexions (inutile de mettre hors tension la Platine).

2.4.3 - Option 'ttl'

Cette option permet de définir la valeur du paramètre 'Time To Live'.

La Platine Terminal peut accéder à un serveur au travers d'un certain nombre de réseaux reliés entre eux par des routeurs (un routeur faisant office de 'charnière' entre deux réseaux).

Pour éviter que des trames 'perdues' ne circulent indéfiniment dans les réseaux, on définit, par la variable 'ttl', le nombre maximum de serveurs ou routeurs que peut traverser une trame.

Toute trame envoyée sur le réseau contient la variable 'ttl'. A chaque fois que la trame 'traverse' la couche IP d'un serveur ou d'un routeur cette variable est décrétementée de 1. Lorsque sa valeur devient nulle, la trame est détruite.

La valeur par défaut de cette variable est égale à 255.

La nouvelle valeur de ce paramètre sera prise en compte lors des prochaines connexions (inutile de mettre hors tension la Platine).

2.4.4 - Option 'noise'

Cette option est utilisée pour définir le 'seuil de rejet au bruit' pour les Platinas 10BaseT (connectique RJ45).

Le seuil de rejet représente la sensibilité de la Platine aux changements d'état des signaux du câble réseau. Les deux valeurs possibles sont '**bas**' (la valeur par défaut) et '**haut**'.

Un seuil de rejet 'haut' permet à la Platine d'être plus sensible en réception. Ainsi, en cas d'une atténuation anormale des signaux (câblage de mauvaise

qualité, câble trop long...) ce paramètre permet à la Platine de recevoir et d'interpréter normalement les trames réseau.

Note : le changement du seuil de rejet doit être effectué avec précaution car un seuil de rejet haut accroît la sensibilité de la Platine en réception, mais accroît aussi sa sensibilité aux perturbations de l'environnement extérieur.

La nouvelle valeur de ce paramètre sera prise en compte à la prochaine mise sous tension de la Platine.

2.4.5 - Option 'tcp port'

Une session TCP/IP (telnet, serveur d'impression, serveur de terminal...) est identifiée par deux paramètres :

- l'adresse IP de la Platine (commune à toutes les sessions),
- un identifiant unique par session appelé Port TCP.

Par exemple : session 1 (192.168.1.241 / 1024)
session 2 (192.168.1.241 / 1025)

Les ports TCP associés à chaque session, peuvent être déterminés de manière aléatoire ou être fixes. Le paramètre '**tcp port**' indique la méthode choisie :

- **aléatoire** (valeur par défaut) : à chaque mise sous tension de la Platine, une valeur aléatoire **x** est calculée. Les 14 ports TCP utilisés par la Platine sont inclus dans l'intervalle variant de x à x+13.
- **fixe** : les 14 ports TCP utilisés par la Platine sont inclus dans l'intervalle variant de 1024 à 1037.

La nouvelle valeur de ce paramètre sera prise en compte à la prochaine mise sous tension de la Platine.

2.5 - STATISTIQUES

Le menu '**Statistiques**' permet la visualisation de différents paramètres concernant la gestion de la Platine Terminal au sein du réseau Ethernet (nombre de trames échangées, rejetées...) :

```
Ethernet
ARP
IP
ICMP
TCP Client
TCP Server
```

Ces statistiques n'étant utilisées que dans des cas très spécifiques (diagnostic d'erreurs, optimisation...), seules sont décrites ici les options Ethernet, ARP, TCP Client et TCP Server.

2.5.1 - Option 'Ethernet'

Cette option permet de visualiser le contenu de 16 compteurs associés aux paramètres gérés par la carte Ethernet de la Platine.

En voici les principaux :

- **intr** : nombre d'interruptions reçues par la Platine. En général, une interruption correspond à une trame, mais il est possible de recevoir plusieurs trames pour une seule interruption.
- **short** : nombre de trames reçues dont l'entête est trop courte.
- **dribble** : nombre de trames reçues dont l'entête est incorrecte.
- **crc** : nombre de trames reçues dont le CRC est incorrect.
- **good** : nombre de trames reçues valides.
- **unktype** : nombre de trames reçues de type non IP ou non ARP (trames IPX par exemple).
- **output** : nombre de trames émises par la Platine.
- **drop** : nombre de trames 'jetées' à cause d'un manque de place dans la file d'attente.
- **nomem** : nombre de trames 'jetées' à cause d'un manque de mémoire.

2.5.2 - Option 'ARP'

Cette option permet de visualiser le nombre de requêtes 'ARP' (Address Resolution Protocol) envoyées ou reçues par la Platine :

- **received** : nombre de trames ARP reçues. Ce nombre est la somme de :
 - **replies** : réponse à une requête ARP émise par la Platine.
 - **badtype** : trame erronées,
 - **request-in** : requêtes ARP reçues,
- **request out** : nombre de requêtes ARP émises par la Platine (demandes de connexion).

A la suite de ces paramètres est affichée la liste des correspondances '**adresse IP / adresse Ethernet**' connues par la Platine.

2.5.3 - Option 'TCP Client'

Cette option permet de visualiser les valeurs de paramètres liés aux connexions telnet :

- **conout** : nombre total de demande de connexions telnet émises par la Platine depuis sa mise sous tension.
- **reset-out**, **runt** et **chksum-err** signalent les erreurs éventuelles.
- **bdcsts** : nombre de 'broadcasts' reçus par la Platine.

A la suite de ces paramètres est affichée la liste des connexions telnet en cours sur la Platine.

2.5.3 - Option 'TCP Server'

Cette option permet de visualiser les valeurs de paramètres liés aux connexions des serveurs d'impression ou de tty :

- **conin** : nombre de demande de connexions reçues par la Platine (chaque demande de connexion est générée par un serveur d'impression ou de tty sur la machine UNIX).
- **reset-out**, **runt** et **chksum-err** signalent les erreurs éventuelles.
- **bdcsts** : nombre de 'broadcasts' reçus par la Platine.

A la suite de ces paramètres est affichée la liste des connexions imprimante ou tty en cours sur la Platine.

2.6 - PARAMETRAGE DES EMULATIONS

L'AX3000 TCP/IP est multi-connexion et multi-host (chaque connexion est indépendante). Il est possible d'utiliser sur une même Platine Terminal autant d'émulations différentes que de sessions disponibles (par exemple, l'émulation **ANSI RS6000** sur une vue et l'émulation **VT220** sur une autre).

Cette fonctionnalité est liée à la capacité de la Platine de pouvoir gérer huit terminaux virtuels autonomes et indépendants. Chaque terminal gère sa propre connexion et possède son propre environnement (variable TERM, émulation, jeu de touches de fonction, nombre de lignes...).




La mise à jour des paramètres environnement de chacun de ces terminaux virtuels s'effectue par l'intermédiaire du set-up Terminal de la Platine (accessible par la combinaison de touches **<Ctrl><Alt><Echap>**).



Note : l'appel du set-up Terminal ne peut être effectué que sur une session active (présence du login ou du prompt).

La description complète de ce set-up est donnée dans le *Guide de l'utilisateur*. L'initialisation d'une session s'effectue simplement par les seules opérations suivantes :

   (entrée dans le set-up)

 (option set-up prédéfini)

  puis  (choix du set-up prédéfini selon le système UNIX installé)

 puis  (propagation éventuelle et validation de cette configuration)

Les set-up prédéfinis (option <F10>) initialisent complètement une session en positionnant automatiquement tous les paramètres de fonctionnement du terminal. Ces paramètres (nombre de lignes, de colonnes, touches de fonctions...) peuvent être repris individuellement, si nécessaire, pour être ajustés à une configuration précise.

Lorsque toutes les sessions d'une Platine doivent être paramétrées de façon identique, la configuration de la session active peut être reportée automatiquement à TOUTES les autres vues (propagation par l'option <F12>).

Note : les modifications de configuration sont effectives immédiatement sur la vue active, alors qu'elles ne seront prises en compte pour les autres vues que lors des prochaines connexions.

A chacun des set-up prédéfinis correspond un système d'exploitation comme le montre le tableau suivant :

Système d'exploitation	Set-up prédéfini	TERM
Unix SCO 3.2v2	UNIX SCO 3.2.2	ansi
Unix SCO 3.2v4	UNIX SCO 3.2.4	ansi
Open Server SCO	SCO OPENSERVER	ansi
Xenix SCO	XENIX SCO	ansi
IBM AIX 3.1	ANSI RS6000	hft-c
IBM AIX 3.2	ANSI RS6000	hft-c-old
IBM AIX 4.x	ANSI RS6000 ou HFT	hft-c-old hft-c
Unix Ware	ANSI SVR4	AT386
Unix Interactive	ANSI INTERACTIVE	AT386
SUN - Solaris 2.x	ANSI SVR4	AT386
OS/400	VT AS400	vt220
Autres systèmes	ANSI	ansi
	VT220	vt220

Ce tableau indique aussi pour chaque système d'exploitation la valeur conseillée de la variable TERM (voir chapitre suivant).

Exemple 1

Supposons que les connexions sur chacune des vues soient opérationnelles et que la Platine doit être configurée de la manière suivante :

- vue 1 : serveur Unix SCO 3.2v4,
- vue 2 : serveur IBM AIX.

Voici les opérations à effectuer pour initialiser chacune des sessions :

- <Alt><F1> activation de la vue 1
- <Ctrl><Alt><Echap> entrée dans le set-up Terminal
- <F10> choix du set-up prédéfini UNIX SCO 4.0
- <F12> sauvegarde des modifications
- <Alt><F2> activation de la vue 2
- <Ctrl><Alt><Echap> entrée dans le set-up Terminal
- <F10> choix du set-up prédéfini ANSI RS6000

Exemple 2

Supposons que la vue 1 soit opérationnelle et que la Platine doit être configurée de la manière suivante :

- vue 1 : serveur IBM AIX,
- vue 2 : serveur IBM AIX,
- vue 3 : serveur IBM AIX,
- vue 4 : serveur IBM AIX.

Voici les opérations à effectuer :

- <Alt><F1> activation de la vue 1
- <Ctrl><Alt><Echap> entrée dans le set-up Terminal
- <F10> choix du set-up prédéfini ANSI RS6000
- <F12> sauvegarde des modifications et
PROPAGATION aux autres sessions

- 3 -
CONFIGURATION DU
SYSTEME UNIX

Ce chapitre concerne les déclarations à effectuer sur le serveur Unix pour l'intégration d'un terminal TCP/IP.

La déclaration d'une Platine Terminal TCP/IP au sein d'un serveur Unix se décompose en :

- déclarations spécifiques réseau,
- déclarations standards terminal.

La configuration standard de certains UNIX peut être mal adaptée à l'utilisation de terminaux TCP/IP, il est alors nécessaire d'effectuer un 'tuning' de ces systèmes d'exploitation (cf. chapitre 3.3).

3.1 - DECLARATIONS RESEAU

Un seul fichier est à modifier au niveau du serveur : `/etc/hosts`.

Ce fichier contient la liste des périphériques réseau accessibles (terminaux, imprimantes... mais aussi serveurs). Un périphérique est défini dans ce fichier par deux paramètres :

- une adresse IP,
- et un nom.

Exemple :

192.1.1.200	vangogh
192.1.1.240	axe11
192.1.1.241	axe12

La mise à jour du fichier `/etc/hosts` n'est pas obligatoire pour une Platine utilisée comme simple terminal mais elle est nécessaire pour utiliser la fonctionnalité de serveurs d'impression et de tty (cf. chapitres 5 et 7).

3.2 - DECLARATIONS GENERALES

3.2.1 - Généralités

Les déclarations nécessaires au fonctionnement d'un terminal TCP/IP sont identiques à celles d'un terminal asynchrone.

Par contre, à la différence d'un périphérique série, un périphérique réseau n'est pas attaché à un device précis. Le device (qui est en fait un pseudo-terminal) est attribué dynamiquement selon l'ordre de connexion des périphériques.

Ceci interdit l'utilisation de fichiers de configuration qui établissent des associations entre un nom de device et une fonctionnalité (par exemple `/etc/ttytype` ou `/etc/default/mapchan` sur Unix SCO).

3.2.2 - La variable TERM

L'un des avantages de la Platine AX3000 est de pouvoir se comporter exactement comme la console principale d'un système d'exploitation donné.

Pour cela, la variable TERM, qui définit l'environnement du terminal (nombre de lignes et de colonnes, valeur des touches de fonctions...) au travers des fichiers `/etc/termcap` et `/usr/lib/terminfo`, doit être renseignée.

La valeur de cette variable est définie au niveau du set-up TCP/IP de la Platine (**[Terminal]**→**[Associer]**) où une session est associée à un serveur UNIX et à une valeur de TERM.

En effet, chacune des 8 sessions de la Platine étant un terminal virtuel, il est possible d'affecter une variable TERM différente par session.

Lors d'une connexion, la Platine TCP/IP communique au système UNIX la valeur TERM requise. Ainsi cette variable est mise à jour dynamiquement pour chaque connexion.

Si la valeur de la variable TERM n'est pas précisée au niveau du set-up TCP/IP ou si la connexion est établie depuis le set-up TCP/IP (option **[Session]→[Ouvrir une session]**), la valeur proposée au serveur dépend de l'émulation de la session :

- émulation de type ANSI : TERM=ANSI
- émulation de type VT : TERM=DEC-VT220

Notez que seules deux valeurs de la variable TERM (exprimées en majuscules) sont disponibles. Ceci est dû à une normalisation du protocole TELNET.

Note : le système d'exploitation peut aussi modifier la valeur de TERM. Ainsi IBM AIX, transforme 'ANSI' en 'ansi' et 'DEC-VT220' en 'vt220' (cf. `termdef`) alors que UNIX SCO convertit ces deux chaînes de caractères en minuscules.

3.2.3 - La gestion scancode

Pour un terminal, deux gestions clavier sont disponibles :

- ASCII : l'appui d'une touche renvoie le code ASCII représenté sur la touche.
- Scancode : à chaque touche est associé un code géographique. Ce code est envoyé à chaque fois qu'une touche est appuyée ou relâchée.

La gestion du clavier (ASCII ou scancode) peut être positionnée depuis le set-up Terminal de la Platine ou par séquences escape.

Il est préférable de que ce soit le système UNIX qui positionne la gestion clavier par séquence escape. En effet il est INDISPENSABLE que le système d'exploitation et le terminal utilise la même gestion.

Les deux séquences escape de gestion clavier sont :

- mode ASCII : `Esc [<1A`
- mode scancode : `Esc [<0A`

Les versions d'UNIX SCO 3.2v4 (et supérieures) permettent une gestion scancode des terminaux. La suite de ce chapitre est donc dédiée à la mise en œuvre de cette fonctionnalité sous ce système.

Pour passer d'une gestion ASCII à une gestion scancode, l'émulation **ansi** utilisée doit contenir les directives `smsc` et `rmsc`. Si ce n'est pas le cas cette émulation doit être modifiée de la manière suivante :

Créez un fichier `ansi` au format 'terminfo' :

```
# cd /usr/lib/terminfo
# infocmp ansi > ansi.src
```

Ajoutez la ligne suivante à ce fichier (ni au début, ni à la fin du fichier) :

```
smsc=\E[<0A, rmsc=\E[<1A, xonc=e, xoffc=g,
```

Sortez de l'éditeur en sauvegardant ces modifications, puis lancez la commande suivante :

```
# tic ansi
```

L'émulation **ansi** ainsi modifiée accepte alors le passage de la gestion ASCII à la gestion scancode.

Note : les directives `xonc` et `xoffc` permettent un contrôle de flux XPC, mais celui-ci n'est pas utilisé pour les terminaux TCP/IP.

Pour activer la gestion scancode sur une session, lancez les commandes suivantes depuis le shell de cette session :

```
$ tput smsc ; stty isscancode xscancode
$ mapkey /usr/lib/keyboard/ps.ibm.fra
```

Note : la commande `mapkey` permet de positionner la nationalité du clavier. Tous les fichiers claviers disponibles sont regroupés dans le répertoire `/usr/lib/keyboard`.

Le passage en ASCII s'effectue par la commande suivante :

```
$ tput rmsc ; stty -isscancode
```

Le passage en gestion scancode peut nécessiter la resynchronisation des leds du clavier. Une séquence escape est disponible (CSI < p1;p2;p3 O). Pour plus de précisions, consulter le *Guide de l'utilisateur*.

3.3 - TUNING DU SYSTEME D'EXPLOITATION

Le package TCP/IP d'un système d'exploitation peut ne pas avoir des valeurs par défaut correctes pour le fonctionnement de terminaux TCP/IP. Cet ajustement des paramètres système s'appelle un 'tuning'.

Note : pour analyser les paramètres à modifier, il est nécessaire de laisser la machine UNIX fonctionner quelques heures pour avoir des résultats significatifs.

3.3.1 - UNIX SCO 3.2v4

L'analyse du comportement du package TCP/IP de UNIX SCO s'effectue au moyen de la commande suivante :

```
# netstat -m
```

Cette commande liste les caractéristiques de tous les classes de buffers intermédiaires utilisées par TCP/IP.

Les deux paramètres à contrôler sont le nombre de buffers par classe (colonne *configure*) et le nombre d'échecs (colonne *fail*).

Si une classe possède un nombre de 'fail' non nul, il est nécessaire d'augmenter le nombre de buffers pour cette classe.

Pour modifier le paramétrage d'une des classes, lancez *sysadmsh* et sélectionnez les options suivantes :

[System]→[Configure]→[Kernel]

Dans le menu proposé, choisissez '**Parameters - Tune kernel params**'. Un second menu apparaît, choisissez l'option 11 : **Streams Data**.

Ajustez les valeurs des paramètres.

Le noyau doit être 'relié' après ces modifications.

3.4 - SERVEUR RARP

Le protocole RARP permet à un périphérique de récupérer son adresse IP à partir de son adresse Ethernet. Cette affectation automatique de l'adresse IP est très utile pour la configuration à distance (voir chapitre 9).

Deux conditions sont requises pour qu'une machine Unix devienne serveur RARP et communique les adresses IP des périphériques émettant des requêtes RARP :

- le process `/etc/rarpd` doit être lancé,
- le fichier `/etc/ethers` doit contenir la liste des associations 'adresse Ethernet / nom de périphérique'.
Le nom du périphérique est celui référencé dans le fichier `/etc/hosts` (voir chapitre 3.1).

Note : avec une configuration de base de UNIX SCO, le démon `rarpd` n'est pas lancé. Il est donc nécessaire de modifier le fichier `/etc/tcp` pour que ce démon soit lancé à chaque démarrage de la machine.

Exemple de fichier `/etc/ethers` :

00:01:34:22:00:44	vangogh
00:A0:34:02:45:03	axel1
00:A0:34:02:45:04	axel2

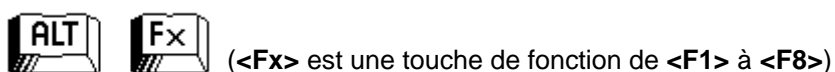
- 4 -

UTILISATION DU MULTI-SESSION

Ce chapitre décrit l'utilisation du multi-session de la Platine AX3000 TCP/IP et les associations Vue/Serveur.

Une des fonctionnalités importantes de la Platine est la connexion **multiple et simultanée**. Cela signifie qu'après avoir effectué une connexion à un serveur, il est possible d'établir d'autres connexions à d'autres serveurs (ou sur le même). Ces différentes sessions s'exécutent simultanément.

Le changement de vue (i.e. de session) s'effectue par la combinaison de touches suivante :



Note : il est possible de modifier la séquence de changement de vue au moyen de l'option Multi-session du set-up de la Platine (voir le *guide de l'utilisateur*).

L'établissement d'une connexion est différente selon qu'il a été déclaré, ou non, des associations Vue/Serveur dans le set-up TCP/IP.

4.1 - UTILISATION AVEC LES ASSOCIATIONS VUE/SERVEUR

L'utilisation de la Platine Terminal est simplifiée dans le cas où les vues sont associées à un ou plusieurs serveurs. Les connexions sont automatiquement établies lors de la mise sous tension (voir chapitre 2.3.2 pour le paramétrage set-up) et lors du changement de vue.

Ces associations Vue/Serveur présentent l'avantage, d'une part de figer une Platine dans une configuration donnée, et d'autre part de rendre inutile l'accès au set-up TCP/IP pour initialiser une connexion.

Lors de la mise sous tension de la Platine, toutes les sessions associées avec le paramètre '**1ère conn. auto.**' positionné à '**o**' (oui) sont automatiquement établies.

Si aucune session n'est ainsi paramétrée, un message d'aide apparaît résumant les associations définies. Par exemple :

```
vue 1 → Site1  ansi
vue 2 → Site1  vt220
vue 3 → Site2  hft-c
```

Il suffit alors d'utiliser la combinaison de touches associée pour se connecter au serveur voulu.

Lors d'un changement de vue (**<Alt><Fx>**), la connexion est automatiquement établie sur la nouvelle vue si cela n'était pas déjà le cas.

Le comportement de la Platine lors de la déconnexion de la session courante (par **<Ctrl><D>** par exemple) dépend du paramètre '**Reconn. auto.**'.

Si ce paramètre est positionné à '**o**' (oui), une nouvelle connexion est automatiquement et immédiatement établie sur cette session.

Si ce paramètre est positionné à '**n**' (non), la Platine affiche la première session parmi celles encore actives. Si aucune session n'est active, le menu résumant les associations est affiché. Il suffit alors d'utiliser la combinaison de touches associée pour se connecter à un serveur.

4.2 - UTILISATION PAR LE SET-UP

Si aucune association Vue/Serveur n'est définie, lors de la mise sous tension de la Platine, le message suivant apparaît :

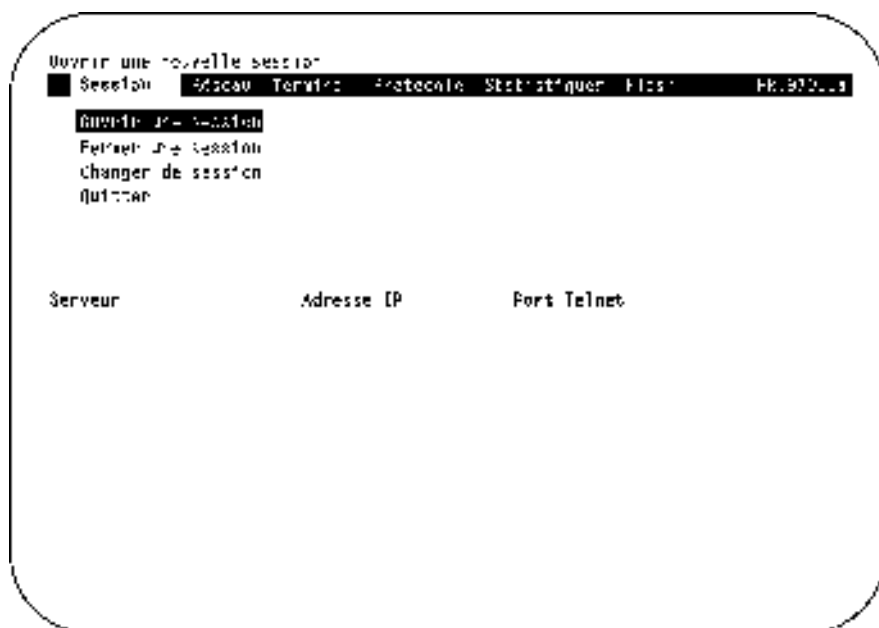
Pas d'association Vue/Serveur

Une connexion à un serveur doit être effectuée manuellement par l'intermédiaire du set-up TCP/IP.

L'accès au set-up TCP/IP se fait par la combinaison de touches suivante :



Le menu général du set-up TCP/IP est le suivant :



Il est présenté sous forme de menus horizontaux et déroulants. Les flèches de direction permettent le déplacement dans les menus. Les sélections sont validées par la touche **<RC>**.

La sortie de ce menu s'effectue, soit au moyen de l'option '**Quitter**' du menu '**Session**', soit au moyen de la touche **<ESC>**.

Le menu '**Session**' du set-up TCP/IP contient les options suivantes :

Ouvrir une session
Fermer une session
Changer de session
Quitter

4.2.1 - Ouvrir une session

La connexion à un serveur se fait au moyen de l'option '**Ouvrir une session**' qui permet de choisir un serveur dans la liste proposée.

Dès que le serveur est choisi, l'écran s'efface, la procédure de connexion est lancée et le login apparaît dès l'établissement de cette dernière (la valeur TERM négociée est la valeur par défaut selon l'émulation liée à la vue).

Note : si la procédure de connexion à un serveur n'aboutit pas (machine éteinte par exemple), il est possible d'annuler la session correspondante en sélectionnant l'option '**Fermer une session**'.

4.2.2 - Changement de session

Il est possible d'avoir jusqu'à 8 sessions (i.e. connexions ou vues) actives simultanément. Le changement de vue s'effectuant par la combinaison de touches **<Alt><Fx>** (hors du set-up TCP/IP).

Mais, il est également possible de commuter de session par l'option '**Changer de session**'. Les sessions sont affichées sous la forme d'une liste (une étoile se trouve devant la session active). Il suffit alors de choisir une autre session courante dans cette liste.

A la sortie du set-up, la nouvelle vue est affichée.

4.2.3 - Fermer une session

En général, la déconnexion se fait au moyen de la combinaison de touches (<Ctrl><D>).

Il est aussi possible de se déconnecter au moyen de l'option '**Fermer une session**'. Il suffit de choisir dans la liste proposée la session à fermer et de valider par <RC>.

Après une déconnexion, la Platine affiche la première vue parmi celles encore actives. S'il n'en existe pas (i.e. la déconnexion s'est effectuée sur la dernière vue active), le message suivant apparaît :

Pas d'association Vue/Serveur

- 5 -
LES SERVEURS D'IMPRESSION
STANDARD

Ce chapitre décrit la gestion d'imprimantes connectées à la Platine.

Les utilitaires `lpd` et `rcmd` permettent **une gestion standard** d'imprimantes distantes au sein d'un réseau. La suite de ce chapitre décrit la connexion d'une ou plusieurs imprimantes à une Platine AXEL TCP/IP et la mise en œuvre de ces deux utilitaires.

Note : la gestion des ports auxiliaires de la Platine peut être aussi assurée par des utilitaires propriétaires AXEL. Ces utilitaires, écrits en langage C, doivent être recompilés sur le système UNIX utilisé. Leur installation et utilisation sont décrites au chapitre 7.

La mise en œuvre de `lpd` et `rcmd` doit être effectuée à deux niveaux :

- le set-up Terminal de la Platine,
- le système UNIX.

5.1 - CONFIGURATION DE LA PLATINE

Le set-up Terminal permet de définir, d'une part, le mode de gestion utilisé pour chacun des ports (`lpd`, `rcmd` ou autres), et d'autre part, la configuration des ports auxiliaires série (vitesse, contrôle de flux...).

Entrez dans le set-up terminal de la Platine (**<Ctrl><Alt><Echap>**) à partir d'une session active.

5.1.1 - Choix du service d'impression

Appuyez sur <F5> pour appeler l'écran '**Mode de fonctionnement**'.

Dans cet écran, les trois paramètres suivants permettent d'associer les services lpd et/ou rcmd à chacun des ports auxiliaires disponibles :

- **Service associé au port AUX1**
- **Service associé au port AUX2**
- **Service associé au port parallèle**

Sélectionnez le paramètre avec les flèches verticales et appuyez sur <ESPACE>. Si le service actuel est printd, rtty, rcmd ou lpd la boîte de dialogue ci-dessous apparaît :

```
Service : printd
Port TCP : 2048
Filtre : aucun
```

Appuyez alors sur <ESPACE> pour sélectionner le service désiré :

```
Aucun
printd
rtty
lpd
rcmd
rtelnet
```

Note : ce menu apparaît directement si le service actuel est aucun ou rtelnet.

Utilisez les **flèches verticales** et la touche <RC> pour choisir le service lpd ou rcmd. Après sélection du service désiré la boîte de dialogue suivante apparaît :

```
Service : lpd
Nom IMP : parallel
Filtre : NL=CR+NL
```

Dans cette boîte de dialogue, utilisez les **flèches verticales** pour sélectionner un paramètre, la touche **<ESPACE>** pour modifier la valeur d'un paramètre et la touche **<RC>** pour sortir.

Le paramètre **Nom IMP** est le nom affecté au port auxiliaire. Ce nom est aussi celui de l'imprimante gérée par `lpd`.

Le paramètre **Filtre** (disponible pour `lpd` seulement) permet de transformer le caractère NL (0Ah) en caractères CR NL (0Dh 0Ah).

Note : le paramètre '**Port auxiliaire par défaut**' de cet écran 'Mode de fonctionnement' permet de d'utiliser le port sélectionné pour la hardcopie ou pour la gestion par séquences escape (CSI 4i et CSI 5i par exemple).

5.1.2 - Configuration du port AUX1 ou AUX2

Si un **port série** est utilisé, il est nécessaire de spécifier le mode de fonctionnement et les paramètres de communication de ce port.

Appuyez sur **<F3>** pour configurer le port AUX1 (ou sur **<F4>** pour AUX2).

Le paramètre '**Fonctionnement**' permet de définir le mode de fonctionnement du port auxiliaire choisi. Appuyez sur **<ESPACE>** pour sélectionner le mode **Imprimante**.

Le choix d'un mode de fonctionnement met automatiquement à jour les paramètres de communication. Il est possible de reprendre individuellement chacun de ces paramètres pour les ajuster au périphérique utilisé.

5.1.3 - Sortie du set-up

Appuyez sur **<F12>** puis **<RC>** pour sauvegarder cette configuration et sortir du set-up. La configuration de la Platine est terminée.

5.2 - CONFIGURATION DU SYSTEME UNIX

Deux possibilités pour gérer les imprimantes connectées à la Platine :

- `lpd` : l'imprimante est directement déclarée et utilisée comme une imprimante système (gérée par `lp`, `lpstat` et `cancel`) mais certaines options de la commande `lp` ne sont pas disponibles.
- `rcmd` : l'imprimante est accédée via la commande `rcmd` (insérée dans un script d'impression ou associée à un pseudo-terminal pour permettre la déclaration et l'utilisation d'une imprimante système).

5.2.1 - Utilisation de lpd

a - Généralités

Déclarez, au moyen de l'utilitaire d'administration système, une imprimante distante. Généralement deux paramètres sont demandés :

- le nom de la machine distante : entrez le nom de la Platine (référencé dans le fichier `/etc/hosts`),
- le nom de l'imprimante : `Nom IMP` saisi dans le set-up de la Platine.

Utilisez la commande standard `lp` pour imprimer.

Note : certaines options de la commande `lp` (nombre de copies, bannière...) ne peuvent être utilisées car la Platine n'est pas une machine UNIX et donc ne spooles pas les demandes d'impressions.

b - Mise en œuvre sous IBM AIX V3.x

Utilisez la commande `smit mkrque` pour créer des imprimantes distantes. La saisie de quatre paramètres est obligatoire :

- **Nom de la file d'attente à ajouter** : nom de l'imprimante (niveau AIX)
- **Syst. hôte cible pour les travaux éloignés** : nom de la Platine (référencé dans le fichier `/etc/hosts`)
- **Nom file d'attente de l'imprimante éloignée** : `Nom IMP` du port `lpd` (au niveau du set-up de la Platine)
- **Nom de l'unité à ajouter** : unité de l'imprimante (au niveau AIX)

Après confirmation, l'imprimante peut être utilisée par la commande `lp`.

c - Mise en œuvre sous IBM AIX V4.x

Utilisez la commande `smit spooler` puis sélectionnez les options 'Ajout d'une file d'attente' puis 'Remote' pour créer des imprimantes distantes.

Deux type d'imprimantes distantes peuvent être créées :

- **avec traitement standard.** Le système AIX transmet un fichier brut à la Platine. C'est à la Platine d'effectuer un traitement éventuel (filtre CR-NL au niveau du set-up).

Pour ce traitement 3 paramètres sont demandés :

- **Nom de la file d'attente à ajouter** : nom de l'imprimante (au niveau AIX)
- **Nom hôte du serveur éloigné** : nom de la Platine (référéncé dans le fichier `/etc/hosts`)
- **Nom de la file d'attente du serveur éloigné** : Nom IMP du port `lpd` (au niveau du set-up de la Platine)

- **avec filtrage local.** Dans ce mode la Platine doit être transparente (aucun filtre au niveau set-up). Le système AIX formate le fichier pour le type d'imprimante déclaré.

Le premier renseignement demandé est le type d'imprimante. Choisissez celui de votre imprimante dans la liste proposée. Ensuite renseignez les paramètres suivants

- **Noms des files d'attente d'impression à ajouter** : nom de l'imprimante (au niveau AIX). Complétez la zone qui convient au type d'imprimante (ASCII, GL emulation, PCL emulation).
- **Nom du système hôte du serveur éloigné** : nom de la Platine (référéncé dans le fichier `/etc/hosts`)
- **Nom de la file d'attente du serveur éloigné** : Nom IMP du port `lpd` (au niveau du set-up de la Platine)

Conseil : utilisez de préférence le traitement avec filtrage local.

Après confirmation, l'imprimante peut être utilisée par la commande `lp`.

d - Mise en œuvre sous UNIX SCO

Lancez l'utilitaire `scoadmin` et sélectionnez [Imprimante]→[Gestionnaire d'impression] puis [Imprimante]→[Ajouter Distant...]→[UNIX].

Les renseignements suivants sont demandés :

- **serveur** : nom de la Platine (référéncé dans le fichier `/etc/hosts`)
- **imprimante** : Nom IMP du port lpd (au niveau du set-up de la Platine)

Après confirmation, l'imprimante peut être utilisée par la commande `lp`.

5.3.1 - Utilisation de rcmd

La commande `rcmd` peut être utilisée pour imprimer, sur un des ports auxiliaires de la Platine, des données arrivant du 'standard input' (stdin).

Note : sur certains UNIX, la commande `rcmd` est remplacée par `rsh`.

La commande suivante est utilisée pour imprimer le contenu d'un fichier :

```
$ cat fic | rcmd ax3000 nomport
```

- où : - `ax3000` est le nom de la Platine dans le fichier `/etc/hosts`,
- `nomport` est le Nom IMP du port (affecté dans le set-up de la Platine).

En cas de besoin, le paramètre `onlcr` permet d'exécuter cette commande avec un filtre du type `NL=CR+NL` :

```
$ cat fic | rcmd ax3000 nomport onlcr
```

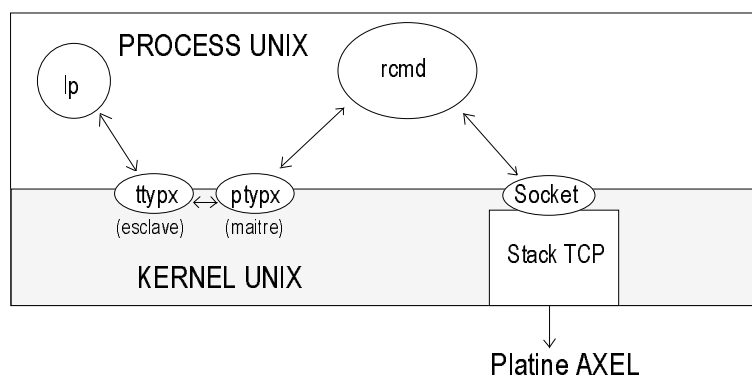
Il est possible de se servir de cette commande `rcmd` en association avec un pseudo-terminal. Ceci permet de déclarer une imprimante attachée à un pseudo-terminal et donc d'accéder à cette imprimante via la commande `lp`.

Pour cela, il est nécessaire de lancer la commande suivante en background :

```
$ rcmd ax3000 nomport < /dev/ptypx &
```

- ax3000 est le nom de la Platine dans le fichier `/etc/hosts`,
- nomport est le nom TCP du port (affecté dans le set-up de la Platine),
- `/dev/ptypx` est la **partie maître** d'un pseudo-terminal.

L'imprimante doit être attachée à la **partie esclave** du pseudo-terminal. Par exemple, si `/dev/ptypf` est utilisé dans la commande `rcmd`, l'imprimante doit être attachée à `/dev/ttypf`). Le schéma suivant illustre les mécanismes mis en jeu :



Note : il est possible d'insérer cette commande `rcmd` dans un script de boot dans le répertoire `/etc/rc2.d`.

- 6 -

LE SERVEUR DE TERMINAUX

Ce chapitre décrit la gestion de terminaux série connectés à la Platine.

La Platine AX3000 possède deux ports auxiliaires série bi-directionnels. Ces ports peuvent être utilisés pour la connexion de divers périphériques (imprimantes, douchettes, balance...) mais aussi pour des terminaux série.

Cette gestion est complètement automatisée grâce au service `rtelnet` qu'il est possible d'associer aux ports AUX1 ou AUX2.

La mise en œuvre de cette fonctionnalité est réalisée au travers du set-up terminal et du set-up TCP/IP de la Platine.

6.1 - SET-UP TERMINAL

Le set-up Terminal permet de définir, d'une part, le mode de gestion utilisé pour chacun des ports (`lpd`, `rcmd`, `rtelnet` ou autres), et d'autre part, la configuration des ports auxiliaires série (vitesse, contrôle de flux...).

Entrez dans le set-up terminal de la Platine (**<Ctrl><Alt><Echap>**) à partir d'une session active.

6.1.1 - Choix du service associé

Appuyez sur **<F5>** pour appeler l'écran '**Mode de fonctionnement**'.

Dans cet écran, les deux paramètres suivants permettent d'associer le service `rtelnet` à chacun des ports auxiliaires série disponibles :

- **Service associé au port AUX1**
- **Service associé au port AUX2**

Sélectionnez le paramètre avec les flèches verticales et appuyez sur **<ESPACE>**. Si le service actuel est `printd`, `rtty`, `rcmd` ou `lpd` la boîte de dialogue ci-dessous apparaît :

```
Service : printd
Port TCP : 2048
Filtre : aucun
```

Appuyez alors sur **<ESPACE>** pour sélectionner le service désiré :

```
Aucun
printd
rtty
lpd
rcmd
rtelnet
tty
```

Note : ce menu apparaît directement si le service actuel est `aucun`, `tty` ou `rtelnet`.

Utilisez les **flèches verticales** et la touche **<RC>** pour choisir le service `rtelnet`.

6.1.2 - Configuration du port AUX1 ou AUX2

Il est nécessaire de spécifier le mode de fonctionnement et les paramètres de communication du ou des ports auxiliaire série utilisés.

Appuyez sur **<F3>** pour configurer le port AUX1 (ou sur **<F4>** pour AUX2).

Le paramètre '**Fonctionnement**' permet de définir le mode de fonctionnement du port auxiliaire choisi. Appuyez sur **<ESPACE>** pour sélectionner le mode **Périphérique bi-directionnel**.

Le choix d'un mode de fonctionnement met automatiquement à jour les paramètres de communication. Il est possible de reprendre individuellement chacun de ces paramètres pour les ajuster au périphérique utilisé.

6.1.3 - Sortie du set-up

Appuyez sur **<F12>** puis **<RC>** pour sauvegarder cette configuration et sortir du set-up.

6.2 - SET-UP TCP/IP

Une fois le service `rtnet` associé à un ou plusieurs port série, il est nécessaire d'associer le ou les ports concernés à un host UNIX (de la même manière que les sessions peuvent être associées à un host UNIX).

Entrez dans le set-up TCP/IP de la Platine (**<Ctrl><Alt><*>**) et appelez le menu **Terminal** :

```
Session
Ligne status
Associer sessions
Associer AUX1
Associer AUX2
Dissocier
```

Dans ce menu sélectionnez l'option **Associer AUX1** ou **Associer AUX2** selon le port choisi.

Note : si le service `rtnet` n'est pas associé au port choisi (dans le set-up terminal), cette option n'est pas disponible.

L'association d'un port nécessite la saisie des paramètres suivants :

- **Serveur** entrez le nom de l'un des hosts UNIX précédemment déclarés avec l'option '**Ajouter serveur**' du menu '**Réseau**' (cf. chapitre 2.2.3).
- **TERM** la valeur de la variable d'environnement TERM. Si aucune valeur n'est pas précisée la valeur par défaut est donnée au chapitre 3.2.

- **1ère conn. auto.** 'o' : lors de la mise sous tension de la Platine TCP/IP la connexion sera automatiquement ouverte.
'n' : il est nécessaire d'appuyer sur une touche du clavier du terminal série pour ouvrir la connexion.
- **Reconn. auto.** 'o' : après une déconnexion (<Ctrl><D>), une nouvelle connexion est immédiatement ouverte.
'n' : il est nécessaire d'appuyer sur une touche du clavier du terminal série pour ouvrir une nouvelle connexion.

L'option '**Dissocier**' supprime une association. Il suffit de se placer sur l'association à supprimer et de taper <RC>.

**- 7 -
LE SERVEUR AXEL
DE VOIES SERIE**

Ce chapitre présente le concept et la mise en œuvre du serveur AXEL de voies série pour UNIX.

7.1 - INTRODUCTION

Le principe du serveur AXEL de voies série est de réaliser une association entre un device Unix (pseudo-terminal) et un port auxiliaire d'une Platine. Ainsi, les données écrites sur le device sont émises sur le port auxiliaire et les données reçues par le port auxiliaire peuvent être lues via le device.

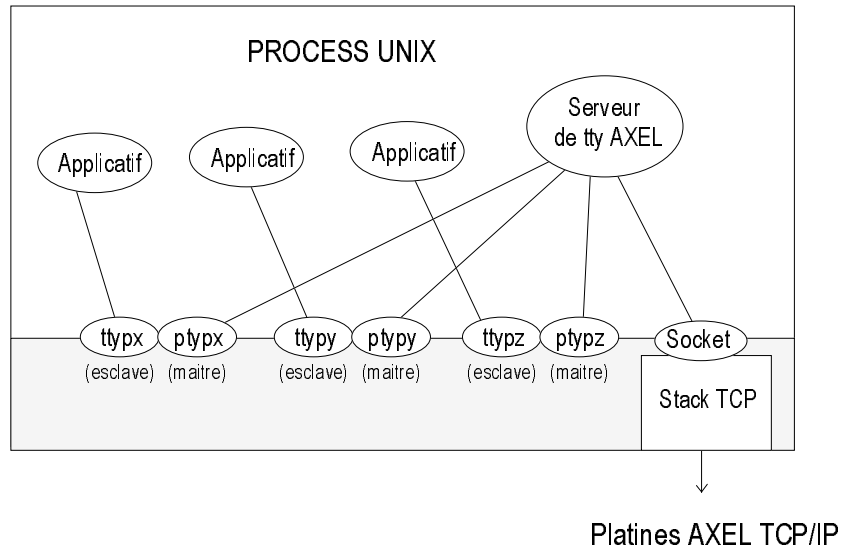
Note : si le périphérique connecté est un terminal asynchrone, reportez-vous au chapitre précédent pour utiliser la fonction 'serveur de terminaux'.

Conditions de fonctionnement :

- Du côté UNIX : le serveur AXEL doit être lancé. Un fichier paramètre donne à ce process la liste des ports auxiliaires des Platinas associés aux pseudo-terminaux Unix.
- Du côté Platine : la Platine doit être configurée avec le service `tty` associé à la machine Unix cible.

Lorsqu'une Platine configurée ainsi est mise sous tension, une connexion automatique est lancée sur le serveur Unix sélectionné. Si cette connexion est acceptée par le serveur de voies série AXEL, le port auxiliaire et le pseudo-terminal sont associés. Un applicatif peut alors utiliser le pseudo-terminal pour accéder au port auxiliaire de la Platine.

Le schéma suivant montre les mécanismes mis en jeu au niveau d'UNIX :



7.2 - CONFIGURATION DE LA PLATINE

La configuration de la Platine s'effectue à deux niveaux :

- set-up terminal : attribution du service `ttty` à chacun des ports auxiliaires utilisés et, si le port auxiliaire est un port série, configuration du port (vitesse, contrôle de flux...),
- set-up TCP/IP : association 'port auxiliaire / host Unix'

7.2.1 - Set-up Terminal

Entrez dans le set-up terminal de la Platine (<Ctrl><Alt><Echap>) à partir d'une session active.

Appuyez sur <F5> pour appeler l'écran '**Mode de fonctionnement**'.

Dans cet écran, les paramètres suivants permettent d'associer le services `tty` à chacun des trois ports auxiliaires :

- Service associé au port AUX1
- Service associé au port AUX2
- Service associé au port parallèle

Sélectionnez le paramètre avec les flèches verticales et appuyez sur <ESPACE>. La boîte de dialogue ci-dessous apparaît :

```
Service : lpd
Nom IMP : aux1
Filtre  : aucun
```

Appuyez alors sur <ESPACE> pour faire apparaître le menu suivant (selon le service actuel, ce menu peut apparaître directement et la boîte de dialogue précédente n'est pas affichée) :

```
Aucun
printd
rtty
lpd
rcmd
rtelnet
tty
```

Sélectionnez le service `tty` avec les flèches verticales et appuyez sur <RC>.

Si le port choisi est un port série, appuyez sur <F3> pour configurer le port AUX1 (ou sur <F4> pour AUX2).

Le paramètre '**Fonctionnement**' permet de définir le mode de fonctionnement du port auxiliaire choisi. Appuyez sur **<ESPACE>** pour sélectionner le mode approprié :

- **Imprimante** : gestion uni-directionnelle (la Platine ne reçoit que les caractères de contrôle de flux),
- **Périphérique bi-directionnel** : gestion bi-directionnelle du port (pour imprimante postscript, écran tactile, douchette...),

Le choix d'un mode de fonctionnement met automatiquement à jour les paramètres de communication. Il est possible de reprendre individuellement chacun de ces paramètres pour les ajuster au périphérique utilisé.

Appuyez sur **<F12>** puis **<RC>** pour sauvegarder cette configuration et sortir du set-up. La configuration de la Platine est terminée.

7.2.2 - Set-up TCP/IP

Entrez dans le set-up TCP/IP de la Platine (**<Ctrl><Alt><*>**).

Sélectionnez le menu vertical **[Terminal]**. Dans ce menu, sélectionnez l'option **[Associer AUX1]** (ou [Associer AUX2] ou [Associer parallèle]).

Dans la zone de saisie proposée, saisissez les paramètres suivants :

- **Serveur** entrez le nom de l'un des hosts UNIX précédemment déclarés avec l'option '**Ajouter serveur**' du menu '**Réseau**' (cf. chapitre 2.2.3).
- **Reconn. auto.** **'o'** : si la connexion se termine (machine unix rebootée par exemple), une nouvelle connexion est automatiquement établie.
 'n' : si la connexion se termine, il est nécessaire d'éteindre puis d'allumer la Platine pour ouvrir une nouvelle connexion.

Quittez le set-up par l'option **[Session]→[Quitter]**.

Note : pour les tests, il est conseillé de ne pas utiliser l'option de reconnexion automatique. En effet si le host Unix n'est pas accessible ou si la configuration du host unix refuse la connexion de la Platine, celle-ci générera des demandes

de connexion régulièrement. Il est donc préférable d'utiliser la reconnexion automatique quand la configuration est validée.

7.2.3 - Ouverture d'une connexion...

Une connexion du service tty est automatiquement générée sur le host Unix :

- à chaque mise sous tension de la Platine
- après chaque déconnexion du service tty si l'option reconnexion automatique est sélectionnée. Cette reconnexion est effectuée après 12 secondes.

Cette connexion est effectuée sur le port TCP 2048 du host UNIX. Si le serveur AXEL de voie série (qui 'écoute' sur ce port TCP) accepte la connexion, la Platine envoie automatiquement une chaîne de caractère contenant son adresse IP et le nom du port auxiliaire (séparé par un caractère NUL).

Le serveur AXEL identifie ainsi le couple 'Platine/Port auxiliaire'. Si celui-ci ne fait pas partie de la liste des connexions autorisées (voir chapitre suivant), le serveur AXEL se déconnecte automatiquement. Sinon la liaison entre le device Unix et le port auxiliaire de la Platine est établie.

7.3 - CONFIGURATION SOUS UNIX

Pour qu'un host UNIX accepte les connexions de ports auxiliaires de Platine, il est nécessaire que le serveur AXEL de voie série (axttyd) soit actif et référence la liste des couples 'Platine/Port auxiliaire' autorisés à ce connecter.

7.3.1 - Installation

Placez vous dans le répertoire /tmp, copiez les fichiers de la disquette AXEL sur disque et placez vous dans le répertoire ttyserver ainsi créé :

```
# cd /tmp <RC>
# tar xvf /dev/fdx <RC>
# cd ttyserver <RC>
```

(fdx représente le device du lecteur de disquette)

Copiez dans le répertoire `/etc` le fichier binaire du serveur AXEL correspondant au système d'exploitation utilisé (`axttyd.XXX`). Par exemple pour IBM AIX :

```
# cp axttyd.AIX /etc/axttyd <RC>
```

Note : les fichiers source (`axttyd.c` et `makefile`) sont présents sur la disquette. Il est donc possible de recompiler le serveur AXEL si le binaire correspondant au système n'est pas disponible.

Copiez dans le répertoire `/etc` le fichier d'associations du serveur AXEL :

```
# cp axfile /etc <RC>
```

Pour que le serveur AXEL soit lancé à chaque boot de la machine, copiez le fichier `S91axel` dans le répertoire `/etc/rc2.d` (ou `/etc/rcd.3` selon les systèmes) :

```
# cp S91axel /etc/rc2.d <RC>
```

Le fichier `S91axel` contient simplement la ligne de commande permettant de lancer le serveur AXEL. Pensez à mettre à jour ce fichier si d'autres paramètres que ceux prévus par défaut sont utilisés.

7.3.2 - Mise en œuvre et utilisation

Le serveur de voie série AXEL est basé sur un fichier de configuration qui décrit les associations entre les ports auxiliaires des Platinas et les pseudo-terminals Unix.

Chaque entrée de ce fichier est composée de 4 paramètres :

- nom de la Platine (référéncé dans `/etc/hosts`)
- port auxiliaire de la Platine. Un des trois mots clé possibles (en minuscule) : `aux1`, `aux2`, `parallel` (attention en anglais),
- la partie maître du pseudo-terminal associé (`/dev/ptypx`)
- la partie esclave du pseudo-terminal associé (`/dev/ttypx`)

Exemple de fichier :

```
#Fichier d'association du serveur AXEL
axell      aux1      /dev/ptyp12  /dev/ttyp12
axell      aux2      /dev/ptyp13  /dev/ttyp13
ax3000     parallel /dev/ptyp0   /dev/ttyp0
```

Note : les lignes commençant par le caractère '#' sont ignorées.

Le fichier de configuration est pris en compte au lancement du serveur AXEL.

Les erreurs éventuelles (syntaxe incorrecte, nom de Platine inconnu, pseudo-terminal indisponible...) sont reportées dans un fichier journal .

La syntaxe de lancement du serveur AXEL est la suivante :

```
/etc/axttyd [-f file] [-l log] [-n port] &
```

- option -f : le nom du fichier paramètre (/etc/axfile par défaut).
- option -l : le nom du fichier journal (/tmp/axttylog par défaut).
- option -n : numéro de port TCP des connexions (par défaut 2048).

Note : ne pas oublier le caractère '&' à la fin de la ligne de commande.

Le serveur AXEL peut être lancé depuis le prompt Unix ou par l'intermédiaire d'un fichier de boot (/etc/rc2.d/S91axel).

La liste des associations prises en compte ainsi que les différentes connexions ou déconnexions sont reportées dans le fichier journal.

Modification du fichier paramètre :

Si le contenu du fichier paramètre est modifié, il est nécessaire d'envoyer le signal 16 au process AXEL pour forcer la relecture de ce fichier. Exemple :

```
# kill -16 pid <RC>
```

- pid est le numéro de process AXEL.

7.3.3 - Désinstallation

Supprimez les fichiers copiés dans le répertoire `/etc` et `/etc/rc2.d` puis 'tuez' le process AXEL par la commande `kill` avec le signal 15 :

```
# kill -15 pid <RC>
```

-pid est le numéro de process AXEL.

- 8 -

**LES SERVEURS RESEAU
PROPRIETAIRES AXEL**

Ce chapitre présente le concept, la mise en œuvre et l'utilisation des serveurs d'impression et de tty AXEL.

Ce chapitre est dédié au trois serveurs UNIX propriétaires AXEL permettant de gérer divers périphériques connectés à la Platine TCP/IP. Ces serveurs, écrits en langage C, doivent être recompilés, si nécessaire, sur le système UNIX utilisé.

Note : le chapitre 5 décrit la mise en œuvre de deux serveurs d'impression standard UNIX : `lpd` et `rcmd`.

8.1 - INTRODUCTION

8.1.1 - le concept

La Platine AX3000 Ethernet TCP/IP possède 2 port auxiliaires série et 1 port auxiliaire parallèle. Les périphériques connectés à la Platine peuvent être gérés au moyen de séquences escape comme sur un terminal traditionnel, ou via des serveurs réseaux AXEL fournis avec la Platine.

Le principe de ces serveurs AXEL est d'associer un **device** à un port auxiliaire de la Platine. Ainsi, l'accès au port auxiliaire se fait au travers du device sans bloquer ni même ralentir l'affichage au niveau de la Platine.

Cette fonctionnalité permet, par exemple, de déclarer une imprimante connectée à la Platine, comme **imprimante système** (accessible par la commande `lp`).

Chacun des ports auxiliaires peut être géré par un serveur réseau AXEL. Il est ainsi possible d'avoir 3 imprimantes système connectées à une Platine.

8.1.2 - Trois serveurs : printd, ttyd et rprint

Pour répondre aux différents besoins, AXEL a développé trois serveurs réseaux :

`printd` serveur d'impression uni-directionnel (AX3000⇒périphérique). Ce serveur utilise un **protocole spécifique** (envoi et attente d'accusé de réception) pour établir la connexion et communiquer avec la Platine.

Le service `printd` doit être activé dans le set-up de la Platine.

Le serveur `printd` est un process en background qui reçoit les données au travers d'un device (pipe ou pseudo-terminal).

Lorsque `printd` n'utilise pas son port, celui-ci reste accessible à d'autres programmes (impressions locales ou hardcopies).

`ttyd` serveur de tty bi-directionnel (AX3000⇔périphérique)
Pas de protocole de communication spécifique (le serveur `ttyd` est donc **totalelement transparent** pour la communication entre le système et le périphérique).

Le service `rtty` doit être activé dans le set-up de la Platine.

Le serveur `ttyd` est un process en background qui reçoit les données au travers d'un pseudo-terminal.

Lorsque `ttyd` est connecté à un port de la Platine terminal, ce port est **inaccessible aux autres programmes** jusqu'à la déconnexion du `ttyd`.

`rprint` serveur d'impression uni-directionnel (AX3000⇒périphérique).
Pas de protocole de communication spécifique (le serveur `rprint` est donc **totalelement transparent** pour la communication entre le système et le périphérique).

Le service `rtty` doit être activé dans le set-up de la Platine.

Le serveur `rprint` est une version simplifiée du serveur `ttyd`. Les données sont lues depuis 'stdin' (et non d'un device).

`rprint` n'est pas un démon mais un simple programme.

Lorsque `rprint` n'utilise pas son port, celui-ci reste accessible à d'autres programmes (impressions locales ou hardcopies).

8.1.3 - Device : pseudo-terminal ou named pipe ?

Le device associé à une Platine peut être soit un **named pipe**, soit un **pseudo-terminal**. Lequel choisir ?

- L'utilisation d'un named pipe est plus souple car il est possible de créer autant de named pipes que nécessaire puisque ceux-ci ne sont pas attachés à un 'driver' UNIX.
- Les pseudo-terminaux sont des couples de devices (parties maître et esclave). Créés et gérés par le système Unix, leur utilisation est donc plus 'standard' que les named pipes. Par contre ce type de ressource est limité en nombre et d'autres logiciels peuvent entrer en concurrence avec le serveur AXEL pour leur obtention.

Notes :

- Le serveur `printd` gère les named pipes et les pseudo-terminaux,
- le serveur `ttyd` ne gère que les pseudo-terminaux,
- le serveur `rprint` n'utilise aucun device.

8.1.4 - Notion de Port TCP

Un serveur AXEL pouvant gérer un ou plusieurs ports d'une même Platine, il est nécessaire de différencier chacun de ces ports.

Ceci s'effectue par affectation d'un numéro à chacun des ports auxiliaires. Ce numéro est appelé numéro de port TCP. Ainsi un port auxiliaire est repéré par l'adresse IP de la Platine et son numéro de Port TCP.

Note : les ports d'une même Platine doivent avoir des numéros de Port TCP différents.

IMPORTANT : le numéro de Port TCP d'un port auxiliaire d'une Platine doit être renseigné à deux niveaux :

- le serveur AXEL,
- le set-up Terminal de la Platine.

8.1.5 - En résumé

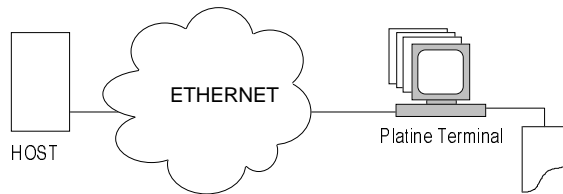
Le tableau suivant résume les caractéristiques des trois serveurs AXEL disponibles :

Serveur AXEL	Service AX3000	Liaison	Gestion UNIX	Partage du port	Device UNIX
printd	printd	uni-directionnelle	démon ou Piobe (*)	Oui	pseudo-terminal ou named pipe (*)
ttyd	rtty	bi-directionnelle	démon	Non	pseudo-terminal
rprint	rtty	uni-directionnelle	programme	Oui	aucun

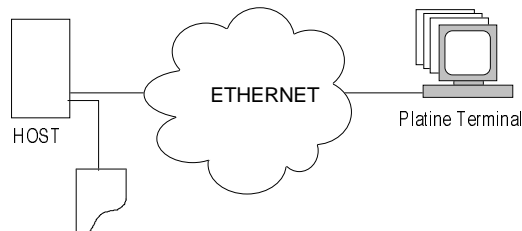
(*) IBM AIX autorise une gestion par démon ou par script Piobe.
 Dans le cas d'une gestion Piobe, aucun device UNIX n'est nécessaire.

La déclaration d'une imprimante connectée à une Platine AXEL comme imprimante système est possible car le système UNIX 'voit' l'imprimante comme si elle était connectée à un port local :

Configuration réelle :



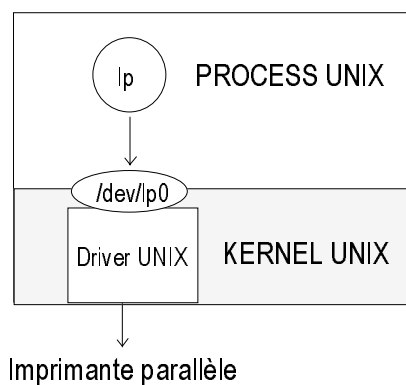
Configuration vue par le système :



Pour illustrer les mécanismes mis en jeu par les serveurs AXEL, voici une suite de schémas reprenant les principales configurations :

a) L'imprimante est connectée sur le port parallèle de la machine UNIX.

Le système UNIX (en l'occurrence le programme `lp`) accède à l'imprimante via un device. Ce device, fourniture standard de l'Unix, gère la carte parallèle de la machine.



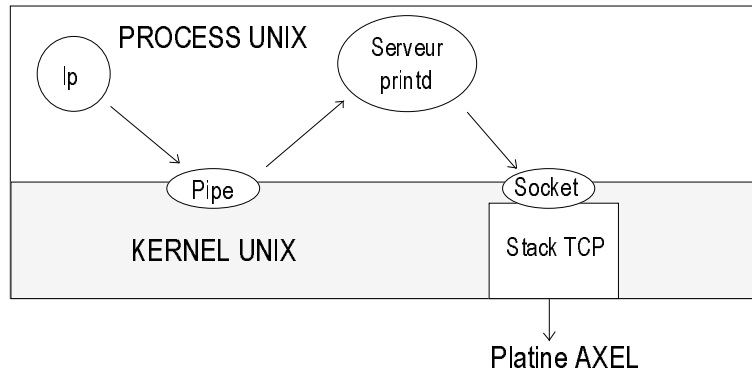
b) L'imprimante est connectée à la Platine (port série ou parallèle)

Le principe est de permettre à UNIX d'accéder à l'imprimante connectée à la Platine via un device (comme dans l'exemple précédent).

Pour cela, trois méthodes sont possibles :

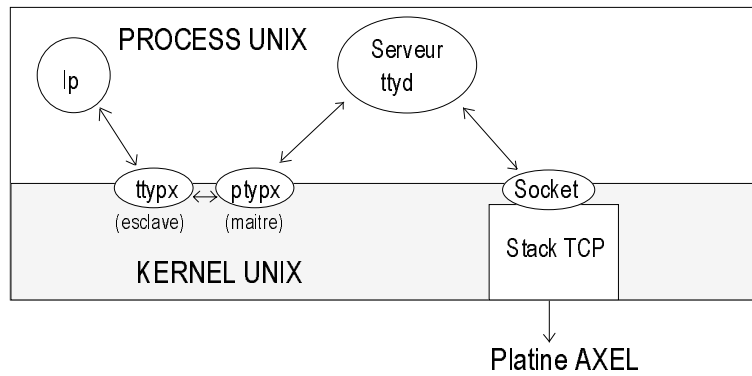
- gestion par `printd` (avec un process en background),
- gestion par `ttysd` (avec un process en background),
- gestion par `printd` (script Piobe, IBM AIX seulement).

Gestion par printd :



Le device utilisé ici est un named pipe (device uni-directionnel) mais cela pourrait être aussi un pseudo-terminal.

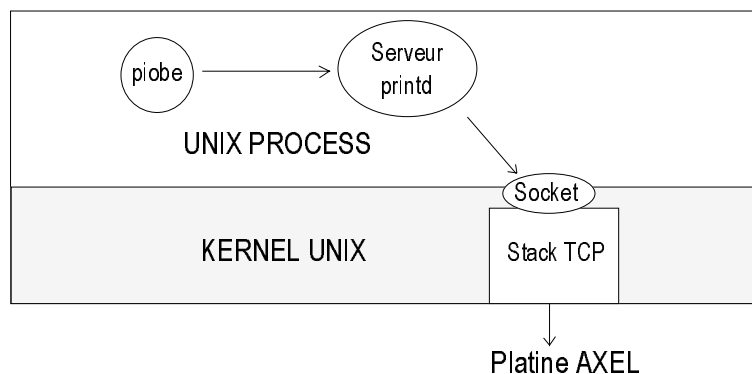
Gestion par ttyd :



Le device utilisé est un pseudo-terminal (device bi-directionnel). Ceci permet de connecter une imprimante postscript éventuellement.

A noter la notion de parties maître et esclave d'un pseudo-terminal.

Gestion par Piobe (sur AIX seulement) :



L'avantage de cette gestion réside dans l'absence de device UNIX. A noter que le serveur rprint utilise le même principe que la gestion par Piobe.

8.1.6 - Précaution d'utilisation

La Platine n'étant plus utilisée comme un simple terminal, mais comme un serveur d'impression et/ou de tty, il est nécessaire de prendre quelques précautions quant à la mise hors tension de celle-ci.

L'idéal est de ne jamais mettre hors tension la Platine (comme s'il s'agissait d'un serveur réseau). Par contre, il est bien sûr possible de mettre hors tension le moniteur.

Si la Platine doit être mise hors tension, vérifiez qu'aucune impression n'est en cours.

8.1.7 - Mise en œuvre

La mise en œuvre d'un serveur AXEL comporte deux phases :

- configuration de la Platine (set-up Terminal),
- configuration du système Unix au travers d'un configurateur fourni avec la Platine.

8.2 - CONFIGURATION DE LA PLATINE

Le set-up Terminal permet de définir, d'une part, le mode de gestion utilisé pour chacun des ports (`printd`, `rtty` ou autres), et d'autre part, la configuration des ports auxiliaires série (vitesse, contrôle de flux...).

Entrez dans le set-up terminal de la Platine (**<Ctrl><Alt><Echap>**) à partir d'une session active.

8.2.1 - Choix du service

Appuyez sur **<F5>** pour appeler l'écran '**Mode de fonctionnement**'.

Dans cet écran, les trois paramètres suivants permettent d'associer les services `lpd` et `rcmd` à chacun des ports auxiliaires disponibles :

- **Service associé au port AUX1**
- **Service associé au port AUX2**
- **Service associé au port parallèle**

Sélectionnez le paramètre avec les flèches verticales et appuyez sur **<SPACE>**. Si le service actuel est `printd`, `rtty`, `rcmd` ou `lpd` la boîte de dialogue ci-dessous apparaît :

```
Service : lpd
Nom IMP : aux1
Filtre  : aucun
```

Appuyez alors sur **<SPACE>** pour sélectionner le service désiré :

```
Aucun
printd
rtty
lpd
rcmd
rtelnet
tty
```

Note : ce menu apparaît directement si le service actuel est `aucun`, `tty` ou `rtelnet`.

Utilisez les **flèches verticales** et la touche **<RC>** pour choisir le service `printd` ou `rty`. Après sélection du service désiré la boîte de dialogue suivante apparaît :

Service	: printd
Port TCP	: 2048
Filtre	: aucun

Dans cette boîte de dialogue, utilisez les **flèches verticales** pour sélectionner un paramètre, la touche **<ESPACE>** pour modifier la valeur d'un paramètre et la touche **<RC>** pour sortir.

Le paramètre **Port TCP** est le numéro affecté au port auxiliaire. Ce nombre est compris entre 1 et 65535.

Le paramètre **Filtre** permet de transformer le caractère NL (0Ah) en caractères CR NL (0Dh 0Ah).

Note : le paramètre '**Port auxiliaire par défaut**' de cet écran 'Mode de fonctionnement' permet d'utiliser le port sélectionné pour la hardcopie ou pour la gestion par séquences escape (CSI 4i et CSI 5i par exemple).

8.2.2 - Configuration du port AUX1 ou AUX2

Si un **port série** est utilisé, il est nécessaire de spécifier le mode de fonctionnement et les paramètres de communication de ce port.

Appuyez sur **<F3>** pour configurer le port AUX1 (ou sur **<F4>** pour AUX2.)

Le paramètre '**Fonctionnement**' permet de définir le mode de fonctionnement du port auxiliaire choisi. Appuyez sur **<ESPACE>** pour sélectionner le mode approprié :

- **Imprimante** : gestion uni-directionnelle (la Platine ne reçoit que les caractères de contrôle de flux),
- **Périphérique bi-directionnel** : gestion bi-directionnelle du port (pour imprimante postscript, écran tactile...).

Le choix d'un mode de fonctionnement met automatiquement à jour les paramètres de communication. Il est possible de reprendre individuellement chacun de ces paramètres pour les ajuster au périphérique utilisé.

8.2.3 - Sortie du set-up

Appuyez sur **<F12>** puis **<RC>** pour sauvegarder cette configuration et sortir du set-up. La configuration de la Platine est terminée.

8.3 - CONFIGURATION DU SYSTEME UNIX

Cette configuration système doit être effectuée en mode MULTI-USER avec le service impression UNIX actif.

Placez vous dans le répertoire `/tmp`, copiez les fichiers de la disquette AXEL sur disque et placez vous dans le répertoire `printserver` ainsi créé :

```
# cd /tmp <RC>
# tar xvf /dev/fdx <RC>
# cd printserver <RC>
```

(fdx représente le device du lecteur de disquette)

Lancez le script d'installation fourni :

```
# ./install <RC>
```

Ce script permet de sélectionner le langage utilisé (français ou anglais) et le système UNIX sur lequel le serveur d'impression sera utilisé (les deux serveurs AXEL et le configurateur sont recopiés dans le répertoire `/etc`).

Cette opération étant terminée, le configurateur (`/etc/axconf`) est exécuté automatiquement. Le menu suivant apparaît :

1) Aide
2) Configuration actuelle
3) Créer une association Platine/Device
4) Créer une imprimante système
5) Supprimer une association Platine/Device
6) Supprimer une imprimante système
7) Retour au menu principal

Note : si le système UNIX utilisé est IBM AIX, le menu est légèrement différent car une seconde gestion est disponible (cf. chapitre 8.3.3).

8.3.1 - Association Platine/Device

La création d'une association Platine/Device (choix 3) nécessite cinq paramètres :

- Serveur AXEL :

Sélectionnez le type de serveur requis (`printd` ou `ttyd`) en restant cohérent avec les choix du set-up Terminal de la Platine.

- Platine AXEL :

Saisissez le nom de la Platine cible. Si ce nom ne fait pas parti du fichier de configuration `/etc/hosts`, le configurateur demande l'adresse IP de la Platine pour compléter le fichier.

- Numéro de Port TCP :

Saisissez le numéro de Port TCP (affecté au niveau du set-up Terminal de la Platine) du port auxiliaire choisi.

- Device :

Entrez d'abord le type du device (pseudo-terminal ou named pipe), puis saisissez son nom.

Note : si le device est un pseudo-terminal, entrez le nom de la partie maître (`/dev/ptypx`).

- Fichier journal :

Entrez le nom du fichier où seront reportées toutes les opérations effectuées par le serveur AXEL. Pour ne pas utiliser de fichier journal, tapez **<RC>** directement.

Après confirmation de ces informations, cette nouvelle association Platine/Device est ajoutée dans un fichier utilisé au boot de la machine (/etc/rc.axel pour IBM AIX, /etc/rc2d/S90axel pour les autres UNIX).

Pour éviter d'avoir à relancer la machine, il est possible de lancer le démon AXEL immédiatement.

Une fois le démon (printd ou ttyd) actif, le serveur AXEL est une véritable passerelle entre le device et la Platine AXEL.

Note : pour détruire une association Platine/Device, utilisez le choix 5 du configurateur AXEL.

Le système UNIX peut maintenant accéder au périphérique attaché à la Platine par l'intermédiaire du device géré par le serveur AXEL.

Note : si ce device est un pseudo-terminal, le nom du device est celui de la partie esclave (/dev/ttypx).

a) Utilisation d'une imprimante

L'utilisation la plus simple est de déclarer une imprimante attachée au device géré par le serveur AXEL. Cette création d'imprimante est réalisée de manière standard par l'utilitaire système UNIX (sysadm, sysadmsh, smit...)

Note : une imprimante système peut aussi être créée ou supprimée par les options 4 et 6 du configurateur AXEL. Utilisez ces options si le device imprimante est un named pipe. Utilisez l'utilitaire système UNIX si le device imprimante est un pseudo-terminal.

Note: avec Solaris 1 et 2, il est obligatoire d'attacher l'imprimante système à un pseudo-terminal.

b) Autres utilisations

D'autres périphériques que des imprimantes peuvent être attachés au device géré par le serveur AXEL. Il est notamment possible de gérer des périphériques bi-directionnels (imprimantes postscript, terminaux série, écrans tactiles...) au moyen du serveur `ttvd`.

La gestion de ces périphériques est totalement standard puisque l'accès s'effectue par le device UNIX.

8.3.2 - IBM AIX - Association Platine/Imprimante

Le menu du configurateur AXEL est le suivant :

1) Aide
2) Configuration actuelle
Gestion par DEMON
3) Créer une association Platine/Device
4) Créer une imprimante système
5) Supprimer une association Platine/Device
6) Supprimer une imprimante système
Gestion par PIOBE
7) Créer une association Platine/Imprimante
8) Supprimer une association Platine/Imprimante
9) Retour au menu principal

La gestion par démon est celle décrite dans le chapitre précédent. IBM AIX apporte une gestion supplémentaire : **la gestion PIOBE**.

Cette gestion associe non plus une Platine à un device, mais une Platine à une imprimante. L'avantage de cette méthode est que le serveur AXEL n'est plus un démon mais un simple programme d'interface PIOBE.

La création d'une association Platine/Imprimante au travers du choix 7 du configurateur AXEL entraîne la saisie des renseignements suivants :

- Platine AXEL :

Saisissez le nom de la Platine. Si ce nom ne fait pas partie du fichier de configuration `/etc/hosts` le configurateur peut le rajouter.

- Numéro de Port TCP :

Saisissez le numéro de Port TCP (affecté au niveau du set-up Terminal de la Platine) du port auxiliaire choisi.

Note : ce port doit être associé au service `printd`.

- Fichier journal :

Nom d'un fichier où seront reportées toutes les opérations effectuées par le serveur AXEL. Pour ne pas utiliser de fichier journal, tapez **<RC>** directement.

- Imprimante :

Descriptif de l'imprimante :

- nom,
- modèle (**4201-3** par défaut),
- type (**asc** par défaut).

Note : pour connaître les modèles et les types d'imprimantes disponibles sur le système, consultez les fichiers présents dans le répertoire `/usr/lpd/pio/predef`.

L'imprimante ainsi créée est accédée au moyen de la commande `lp` :

```
$ lp -d nom_prn fichier
```

Pour supprimer une association Platine/Imprimante, utilisez le choix 8 du configurateur AXEL.

**- 9 -
LE MULTI-SHELL**

Ce chapitre décrit l'installation et l'utilisation du logiciel de multi-shell fourni avec la Platine.

Une des principales caractéristiques des Platinas TCP/IP est la multi-connexion. Ceci permet d'obtenir jusqu'à 8 connexions simultanées et indépendantes sur un ou plusieurs host Unix.

Lorsque plusieurs sessions se connectent sur le même host avec le même nom utilisateur, il n'est pas utile que chacune de ces sessions soient obtenues par l'intermédiaire d'un 'login'.

Le logiciel AXEL de multi-shell pour Platine TCP/IP évite de multiplier les logins en permettant de dupliquer le shell d'une session. Pour cela, il suffit de lancer le logiciel AXEL depuis l'interpréteur de commandes UNIX pour obtenir jusqu'à 8 exemplaires de la session courante.

Exemple : il est possible d'ouvrir deux connexions sur deux hosts Unix différents et d'avoir pour chacune de ces connexions 4 shells indépendants. Des combinaisons de touches différentes permettent de changer de session TCP/IP (<Alt><Fx>) et de changer de shell à l'intérieur d'une session TCP/IP (<Ctrl><Fx>).

9.1 - INSTALLATION SOUS UNIX

Loggez-vous comme utilisateur `root`, placez vous dans le répertoire `/tmp`, copiez les fichiers de la disquette AXEL sur disque et placez vous dans le répertoire `multishell` ainsi créé :

```
# cd /tmp <RC>
# tar xvf /dev/fdx <RC>
# cd multishell <RC>
```

(`fdx` représente le device du lecteur de disquette)

Les fichiers exécutables suivants sont copiés sur disque :

- `tcpmsh.AIX` : IBM AIX
- `tcpmsh.SCO` : XENIX et UNIX SCO
- `tcpmsh.SVR4` : UNIX AT&T et UNIXWARE
- `tcpmsh.SNX` : SINIX

Copiez dans le répertoire `/usr/bin` le fichier binaire du serveur AXEL correspondant au système d'exploitation utilisé (`tcpmsh.XXX`) en enlevant le suffixe.

Exemple pour AIX :

```
# cp tcpmsh.AIX /usr/bin/tcpmsh <RC>
```

Le logiciel est immédiatement opérationnel, car étant dédié à la Platine Terminal AXEL, aucun fichier de configuration n'est nécessaire.

9.2 - PARAMETRAGE DE LA PLATINE

L'utilisation de plusieurs vues multi-shell au sein d'une même session telnet doit être paramétrée au niveau du set-up de la Platine.

En effet, la Platine gère 8 contextes d'écran. Ces 8 contextes sont à partager entre les sessions telnet et les vues du multi-shell (appelées dans le set-up `page/session`).

La configuration usine de la Platine est 8 sessions telnet et 1 page/session.

Pour modifier le paramétrage de la Platine il est nécessaire d'entrer dans le set-up TCP/IP (tapez <Ctrl><Alt><*>) et dans le menu proposé de sélectionner l'option **[Terminal]**→**[Session]**.

Cette option permet de mettre à jours deux paramètres :

- nombre de session : nombre de sessions telnet
- nombre de page/session : nombre de vues multi-shell

Dans l'exemple donné en introduction du chapitre, le nombre de sessions est 2 et le nombre de page/session est 4.

Quittez le set-up par l'option **[Session]**→**[Quitter]**.

Notes : Pour modifier ces deux paramètres, aucune session telnet ne doit être connectée.

Pour rester cohérent avec les ressources de la Platine, le nombre de sessions multiplié par le nombre de pages/session ne doit pas être supérieur à 8.

9.3 - UTILISATION

La commande `tcpmsh` peut être lancée, soit directement à partir du prompt Unix, soit insérée dans un fichier script.

9.3.1 - Lancement

La syntaxe de lancement de `tcpmsh` est la suivante :

```
$ tcpmsh [-options] <RC>
```

Note : La suite du chapitre détaille chacune des options disponibles.

Après lancement du logiciel, une bannière est affichée où apparaissent notamment la version de `tcpmsh` et le nombre maximum de vues allouées.

Le logiciel crée ensuite la vue 1 en lui attribuant un pseudo-terminal.

La vue ainsi créée est une copie exacte du shell d'où a été lancé `tcpmsh`. Aussi le fichier `.profile` associé au login de ce shell est de nouveau exécuté (il en sera de même pour les autres vues).

9.3.2 - Création et changement de vue

Pour créer une nouvelle vue ou pour changer de vue, utilisez les combinaisons de touches suivantes :

- **<Ctrl><F1>** : vue 1
- **<Ctrl><F2>** : vue 2
- ...
- **<Ctrl><F8>** : vue 8

Si la vue accédée n'est pas déjà créée, le logiciel attribue un pseudo-terminal et initialise cette nouvelle vue (exécution du fichier `.profile`).

Le nombre maximum de vues accessibles est paramétrable par une option de lancement du logiciel.

9.3.3 - Suppression de vue

Pour supprimer une vue, passez sur cette vue (**<Ctrl><Fx>**) et tapez la commande `exit` (ou tapez **<Ctrl><D>**) à l'interpréteur de commande UNIX. Le logiciel libère le pseudo-terminal associé à cette vue et passe automatiquement sur la première vue active.

Si la vue supprimée était la dernière, le logiciel `tcpmsh` se termine.

9.3.4 - Sortie du logiciel

La sortie du logiciel s'effectue en supprimant la dernière session.

9.4 - LES OPTIONS

La syntaxe de lancement de `tcpmsh` est la suivante :

```
$ tcpmsh [-bchLLmnpSx] <RC>
```

La suite du chapitre détaille chacune des options disponibles.

9.4.1 - Option b : choix des pttys

Le logiciel `tcpmsh` est basé sur les pseudo-terminaux (pty). Un pseudo-terminal est un fichier dont le nom varie selon les systèmes d'exploitation.

Pour pallier d'éventuels problèmes, l'option `-b` permet de paramétrer le nom générique des pseudo-terminaux du système d'exploitation utilisé.

L'option `-b` doit être suivie d'un paramètre (trois valeurs possibles) :

- b 0 : utilisation des `ttypx` (x exprimé en décimal, varie de 0 à 63),
- b 1 : utilisation des `ttypx` (x en hexadécimal, varie de 0 à 3F),
- b 2 : utilisation des `ttypx`, `ttypq`, `ttypx` et `ttypx` (x exprimé en hexadécimal, varie de 0 à F).

Par défaut, le comportement du logiciel est celui de l'option `-b 0`.

Note : cette option n'est pas disponible pour la version SVR4 et est incompatible avec l'utilisation de l'option `-p` (voir chapitre 9.4.9)

9.4.2 - Option c : bordure d'écran

Une fois le logiciel `tcpmsh` activé, la ligne status permet éventuellement d'identifier la vue courante.

Une autre méthode pour identifier la vue courante, consiste à associer à chacune des vues une couleur de bordure d'écran. Ceci est obtenu grâce à l'utilisation de l'option `-c`.

Le tableau suivant donne la correspondance entre numéro de vue et couleur de bordure d'écran :

Vue	Couleur
1	Bleu
2	Jaune brillant
3	Vert
4	Cyan

Vue	Couleur
5	Rouge
6	Magenta
7	Jaune clair
8	Blanc

9.4.3 - Option f : commandes initiales

Il est possible de lancer des commandes initiales sur chacune des vues. Cette liste de commandes (une par vue) est stockée dans un fichier dont le nom est passé en paramètre avec l'option '-f'.

Ce fichier de commandes contient 8 lignes maximum. Chaque ligne contient la commande affectée à sa vue :

1^{ère} ligne → vue 1
2^{ème} ligne → vue 2
... ..
8^{ème} ligne → vue 8

Si une vue n'a pas de commande initiale, la ligne qui lui est affectée doit commencer par un tiret (caractère 2Dh).

Exemple : le fichier **cmd**

```
smit  
-  
client
```

La prise en compte de ce fichier est effectué par la commande :

```
# tcpmsh -f cmd <RC>
```

Dans cet exemple, l'utilitaire smit est lancé sur la vue 1 et l'application client sur la vue 3. Les vues 2, 4, 5, 6, 7 et 8 n'ont pas de commande initiale (c'est le shell qui est actif).

9.4.4 - Option h : aide en ligne

Cette option affiche la syntaxe de lancement du logiciel `tcpmsh`, ainsi qu'une description sommaire de chacune des options.

9.4.5 - Option L : ligne status sur la 26^{ème} ligne

L'option `-L` permet au logiciel `tcpmsh`, d'utiliser la ligne status gérée par la multi-connexion de la Platine.

Rappel sur la ligne status TCP/IP :

A chaque session TCP/IP, est affectée une cellule (10 caractères) de la ligne status. A chaque cellule est associé un libellé qui est soit le nom du host (sur lequel est connectée la session) soit un label (entré au set-up ou positionné par séquence escape). La cellule de la session courante est affichée en inverse vidéo alors que les autres sont affichées en mode normal.

A la suite de l'option `-L` doit être ajoutée une chaîne de caractère qui représente le label associé à la cellule. La longueur de cette chaîne est de 6 caractères maximum car les 2 caractères `/n` sont concaténés à cette chaîne (`n` représente le numéro de la vue courante).

Modification du set-up :

Ce choix d'utiliser la 26^{ème} ligne de l'écran comme ligne status doit être précisé au niveau du set-up. Tapez `<Ctrl><Alt><*>` et dans le menu proposé sélectionnez l'option **[Terminal]→[Ligne status]** pour activer la ligne status.

Exemple :

```
$ tcpmsh -n 4 -L unix <RC>
```

la cellule de la ligne status contient `unix/1` pour la vue 1, `unix/2` pour la vue 2, `unix/3` pour la vue 3 et `unix/4` pour la vue 4.

9.4.6 - Option l : ligne status sur la 25^{ème} ligne

L'option `-l` (L minuscule) permet d'utiliser le logiciel `tcpmsh` avec une ligne status au bas de l'écran indiquant à la fois, le nombre de vues utilisées et la vue courante.

A la suite de cette option, une chaîne de caractères peut être ajoutée. Cette chaîne de caractères représente un commentaire qui sera affiché sur la droite de la ligne status.

Exemple :

```
$ NOM=`tty` <RC>
$ tcpmsh -n 4 -l $NOM <RC>
```

Cette ligne status étant située sur la 25^{ème} ligne, il est important que la variable TERM pointe sur une émulation décrivant une taille d'écran de 24 lignes.

Si le système d'exploitation UNIX ne contient pas en standard une émulation de ce type, il est nécessaire de modifier le fichier `terminfo` utilisé.

Modification du fichier terminfo :

La description des émulations se situe, en général, dans un fichier suffixé par `.src` ou `.ti` dans le répertoire `/usr/lib/terminfo`. Si la description de l'émulation n'est pas disponible tapez la commande suivante :

```
# infocmp emul > /usr/lib/terminfo/emul.src <RC>
```

où `emul` est l'émulation utilisée (ansi ou at386 par exemple)

Editez le fichier de description avec l'éditeur `vi`. Une fois la description de l'émulation trouvée, positionnez la variable fixant le nombre de lignes (`li#` ou `lines#` selon l'Unix) à 24. Sauvegardez ces modifications.

Le fichier ainsi modifié doit être recompilé. Lancez la commande suivante (où `nomfic` est le nom du fichier terminfo modifié) :

```
# tic /usr/lib/terminfo/emul.src <RC>
```

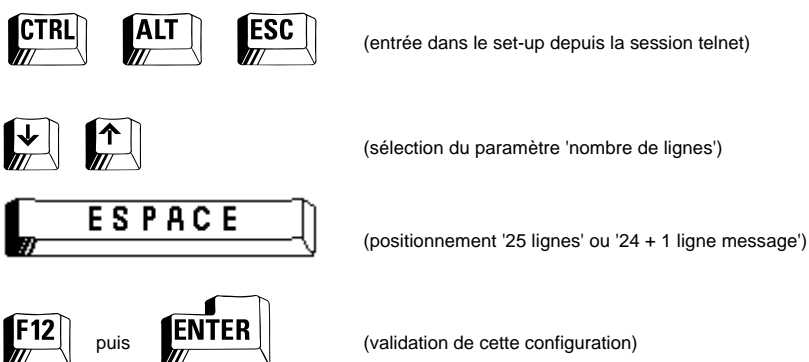
Modification du set-up :

Ce choix d'utiliser la 25ème ligne de l'écran comme ligne status doit être en concordance avec la valeur du paramètre 'nombre de lignes' du set-up de la Platine Terminal qui peut avoir deux valeurs :

- 25 lignes,
- 24 + 1 ligne message.

Le lancement du logiciel `tcpmsh` avec l'option `-l`, force la mise à jour de ce paramètre à '24 + 1 ligne message'. Par contre en sortie du logiciel, l'ancienne valeur n'est pas restaurée.

Voici les opérations à effectuer pour modifier la valeur de ce paramètre :



9.4.7 - Option m : mode muet

Lors du lancement de `axmscreen`, des messages d'aide sont automatiquement affichés (nombre de vues, touches actives...).

Il est possible de supprimer ces messages en utilisant l'option `-m`.

9.4.8 - Option n : nombre de vues

Par défaut, 2 vues au maximum sont disponibles. Ce nombre peut être modifié par l'option `-n x` (où `x` varie de 1 à 8).

Exemple :

```
$ tcpmsh -n 4 <RC>
```

9.4.9 - Option p : préaffectation des pseudo-terminaux

L'option `-p` permet de choisir par avance les pseudo-terminaux à utiliser pour chaque vue du multiscreen. Cette liste de pseudo-terminaux est stockée dans un fichier dont le nom est passé en paramètre.

Le fichier de pseudo-terminaux contient 8 lignes maximum. Chaque ligne contient le pty affecté à sa vue :

```
1ère ligne → vue 1  
2ème ligne → vue 2  
... ..  
8ème ligne → vue 8
```

Exemple : le fichier predef

```
ptyp2  
ptyp5  
ptyp6  
ptyp8
```

La prise en compte de ce fichier est effectué par la commande :

```
$ tcpmsh -p predef <RC>
```

Avec ce fichier `predef`, `ptyp2` est affecté à la vue 1, `ptyp5` à la vue 2, `ptyp6` à la vue 3 et `ptyp8` à la vue 4.

Note : cette option n'est pas disponible pour la version SVR4.

9.4.10 - Option S : choix du shell

Sous Unix, plusieurs shells sont disponibles (`sh`, `ksh`, `rsh`...). L'option `-s` permet de choisir un autre shell que celui affecté par défaut.

Pour cela, il suffit de faire suivre l'option `-s` du nom du shell désiré.

Exemple :

```
$ tcpmsh -S ksh <RC>
```

9.4.11 - Option x : buffer circulaire

Une vue inactive est une vue sur laquelle l'utilisateur ne travaille pas. Dans une session, si n vues sont utilisées, il y a, à tout moment, une seule vue active (la vue courante) et $n-1$ vues inactives.

Quel est le comportement d'une vue inactive ?

Par défaut, l'affichage de caractères sur une vue inactive est impossible. En effet, une vue inactive qui reçoit des caractères à afficher est rapidement bloquée (i.e. XOFF). Elle sera débloquée lorsqu'elle deviendra active.

Pour éviter ces blocages, il est prévu d'affecter à chacune des vues un buffer circulaire au moyen de l'option `-x`. Ainsi les caractères que reçoit une vue inactive sont stockés dans le buffer circulaire correspondant ; au moment où la vue est réactivée, le contenu du buffer est affiché.

La taille par défaut de ces buffers circulaires est de 2 Ko par vue. Il est possible de spécifier la taille au moyen de la syntaxe `-x n` ou `n` est la taille en Ko des buffers par vue (n varie de 1 à 9).

A noter que cette gestion par buffer circulaire ne conserve que les dernières données envoyées à une vue inactive. La réactivation d'une vue peut donc faire apparaître un affichage incomplet (cf. chapitre suivant).

9.5 - LIMITATIONS DE LA NORME ANSI

L'émulation ANSI n'est pas réentrante. Cela signifie qu'une séquence escape 'coupée' par une autre séquence escape ou par des caractères quelconques est perdue ou provoque des anomalies d'affichage. En ANSI, les caractères d'une séquence escape non reconnue (car coupée ou incomplète) sont affichés.

Cette spécification de l'émulation ANSI peut perturber l'utilisation du logiciel `tcpmsh` dans les deux cas suivants :

- **Le changement de vue** : si la séquence de changement de vue (combinaison `<Ctrl><Fx>`) coupe une séquence escape qui s'affiche sur une vue au moment où on la rend inactive, l'affichage peut être perturbé. La solution consiste à changer de vue lorsque l'écran est 'stable'.
- **L'utilisation de buffers (option '-x')** : après un changement de vue, le logiciel vide le buffer affecté à la nouvelle vue active. Ce buffer étant géré comme un buffer circulaire, il est possible que les premiers caractères de la première séquence escape à afficher soient absents. Les caractères de la deuxième moitié de cette séquence escape coupée seront alors affichés tel quels. Ceci peut se produire si les buffers circulaires utilisés ont une taille insuffisante. La solution consiste donc à augmenter cette taille au moyen de l'option `-x n`.

**- 10 -
CONFIGURATION A DISTANCE**

Ce chapitre présente la notion de configuration à distance des Platines AX3000 TCP/IP.

10.1 - INTRODUCTION

Cette fonction d'administration permet de configurer à distance une Platine TCP/IP. La configuration d'une Platine s'effectue au moyen de la commande `rcmd` et d'un fichier texte qui décrit chacun des paramètres set-up de la Platine.

Ce fichier texte peut contenir la totalité ou une partie des paramètres set-up disponibles. Un fichier de configuration peut être obtenu de deux manières :

- création par un éditeur de texte (`vi` par exemple),
- acquisition de la configuration d'une Platine déjà configurée au moyen de la commande `rcmd`.

Ceci permet par exemple :

- l'installation rapide de plusieurs Platines : configuration de ces Platines à partir d'un même fichier de configuration,
- la maintenance d'un site : archivage du fichier de configuration de chacune des Platines,
- la modification à distance de la configuration d'une Platine.

Note : la commande `rcmd` est aussi appelée `rsh` sur certains systèmes d'exploitation.

Les paramètres de la commande `rcmd` sont :

- un nom de Platine : ce nom est contenu dans le fichier `/etc/hosts` qui associe un nom à une adresse IP.

- une commande : pour l'administration à distance d'une Platine, trois commandes sont disponibles :

- `setup_send` : mise à jour d'une configuration,
- `setup_get` : acquisition d'une configuration,
- `ax_reboot` : ré-initialisation de la Platine.

Exemple :

```
# rcmd nom_ax3000 ax_reboot <RC>
```

Note : pour pouvoir accéder une Platine TCP/IP via une commande `rcmd`, il est impératif que cette Platine ait une adresse IP valide. Une adresse IP peut être mise à jour de deux manières :

- par le set-up TCP/IP.
- ou automatiquement par le protocole RARP (voir le chapitre 2.2.2).

10.2 - CONFIGURATION D'UNE PLATINE

La configuration à distance d'une Platine s'effectue par la commande '`rcmd setup_send`' qui permet de télécharger le fichier de configuration :

```
# cat /tmp/fic | rcmd nom_ax3000 setup_send <RC>
```

Note : le fichier de configuration est, soit créé par un éditeur de texte, soit obtenu par la commande '`rcmd setup_get`' (voir chapitre suivant). Consultez le chapitre 10.5 pour une description complète du fichier de configuration.

Si la commande `rcmd` est prise en compte par la Platine à configurer, le message '`Store set-up in progress...`' est affiché sur le poste de l'opérateur sinon la commande sort en erreur après quelques secondes.

Si la configuration s'est bien passée le message '`OK, set-up updated`' est affiché sur le poste de l'opérateur (si d'autres messages sont affichés, consultez le chapitre 10.4).

Pour prendre en compte cette nouvelle configuration la Platine doit être ré-initialisée. Ceci peut être effectué en éteignant puis allumant la Platine ou en utilisant la commande suivante :

```
# rcmd nom_ax3000 ax_reboot <RC>
```

Le message 'Reboot in progress...' est affiché sur le poste de l'opérateur, la Platine est ré-initialisée et le nouveau set-up est pris en compte. En cas de problème sur la commande `rcmd`, un message d'erreur système est affiché sur le poste de l'opérateur.

10.3 - ACQUISITION D'UNE CONFIGURATION

L'acquisition de la configuration d'une Platine s'effectue au travers de la commande '`rcmd setup_get`' qui génère l'affichage de tous les paramètres set-up (sur le standard output). Cet affichage peut être redirigé dans un fichier :

```
# rcmd nom_ax3000 setup_get > /tmp/fic <RC>
```

Note : si la commande `rcmd` est prise en compte par la Platine à interroger, le message 'Load set-up in progress...' est affiché sur le poste de l'opérateur sinon la commande sort en erreur après quelques secondes.

Si l'acquisition du set-up s'est bien passée le message 'OK, set-up sent' est affiché sur le poste de l'opérateur (voir le chapitre 10.4 si d'autres messages sont affichés).

Le fichier ASCII généré contient les valeurs de tous les paramètres set-up de la Platine. Consultez le chapitre 10.5 pour une description complète du fichier de configuration.

10.4 - MESSAGES D'ERREUR

Ce chapitre liste les messages d'erreur possibles durant l'exécution de la commande `rcmd` (affichés sur le standard error). Pour chacun de ces messages sont spécifiés : le nom de la commande utilisée, la cause de l'erreur, les conséquences et les vérifications ou solutions possibles.

Les erreurs sont réparties en deux groupes :

- erreur fatales : numéro d'erreur 100 à 106
- erreurs d'alertes (warning) : numéro d'erreur 200 à 213

10.4.1 - Erreurs fatales

Une erreur fatale interrompt l'exécution de la commande `rcmd` en cours. Les erreurs fatales possibles sont les suivantes :

ERR 101: header label expected.

Commande : `setup_send`
Cause : l'entête AXEL (`BEGIN_AX_SETUP`) n'est pas la première ligne du fichier de configuration.
Conséquence : la commande `rcmd` est interrompue, le set-up de la Platine n'est pas modifié.
Solution : vérifiez que le fichier utilisé est un fichier de type AXEL.

ERR 102: bad header label protocol version.

Commande : `setup_send`
Cause : la version de protocole intégrée dans l'entête ne peut pas être traitée par le firmware de la Platine.
Conséquence : la commande `rcmd` est interrompue, le set-up de la Platine n'est pas modifié.
Solution : utilisez une version de protocole compatible avec le firmware de la Platine.

ERR 103: trailer label not found. Default factory set-up reloaded

Commande : `setup_send`

Cause : le libellé de fin (`END_AX_SETUP`) n'a pas été trouvé (car il ne fait pas partie du fichier ou que la connexion `rcmd` a été interrompue).

Conséquence : le set-up usine de la Platine a dû être rechargé car l'absence du libellé de fin de configuration entraîne une configuration incomplète.

Solution : vérifiez que ce libellé fasse partie du fichier et relancez la commande `rcmd`.

ERR 104: Can not get complete set-up (`rcmd` command aborted).

Commande : `setup_get`

Cause : le set-up n'a pas été récupéré complètement (la commande `rcmd` a été interrompue ou un composant réseau a été mis hors tension ou débranché durant la commande `rcmd`).

Conséquence : le fichier récupéré est invalide (le label de fin n'est pas présent).

Solution : relancez la commande `rcmd`.

ERR 105: invalid `rcmd` command.

Commande : `setup_send`, `setup_get` ou `ax_reboot`

Cause : la commande associée à `rcmd` est inconnue.

Conséquence : la commande `rcmd` n'a aucun effet.

Solution : utilisez `setup_send`, `setup_get` ou `ax_reboot`.

ERR 106: `cmos` busy.

Commande : `setup_send`

Cause : la commande `rcmd` ne peut accéder au set-up de la Platine car une autre commande `rcmd` (`setup_send` ou `setup_get`) est en cours sur cette Platine ou le set-up de cette Platine est en cours d'utilisation.

Conséquence : la commande `rcmd` n'a aucun effet.

Solution : attendez quelques secondes la fin de la première commande `rcmd` ou sortez du set-up sur la Platine à configurer.

10.4.2 - Erreurs d'alerte (warning)

Une erreur de type warning n'interrompt pas l'exécution de la commande en cours mais signale un problème. Les erreurs de type warning possibles sont les suivantes :

ERR 201: xxx: unknown parameter name.

Commande : `setup_send`

Cause : le paramètre set-up xxx n'est pas reconnu par la Platine (ex : `s1_screencolumn` au lieu de `s1_screen_column`).

Conséquence : ce paramètre set-up du fichier de configuration est ignoré, l'ancienne valeur est conservée.

Solution : vérifiez la syntaxe du paramètre en question (chapitre 10.5).

ERR 202: xxx: invalid value.

Commande : `setup_send`

Cause : la valeur affectée au paramètre set-up xxx n'est pas reconnue par la Platine (exemple : `multi_nbpage=10`).

Conséquence : ce paramètre set-up du fichier de configuration est ignoré, l'ancienne valeur est conservée.

Solution : vérifiez les valeurs autorisées pour ce paramètre (chapitre 10.5).

ERR 203: xxx: no place to store udk value.

Commande : `setup_send`

Cause : le paramètre `udk_xxx` correspond à une touche programmable (`udk` signifie User Defined Key, une touche de fonction par exemple). Il est impossible de mettre à jour ce paramètre à cause d'un manque de mémoire.

Conséquence : la valeur par défaut de la touche est restaurée.

ERR 204: line(s) after the trailer label ignored.

Commande : `setup_send`

Cause : `END_AX_SETUP` n'est pas la dernière ligne du fichier.

Conséquence : les lignes situées après le label de fin sont ignorées. Le set-up de la Platine est correctement mis à jour.

Solution : supprimez les lignes après ce label.

ERR 211: service name/number error. Service(s) may be reset.

Commande : setup_send

Cause : plusieurs services d'impression (ou de voie série) ont les mêmes noms (ou numéros de port TCP).

Conséquence : pour obtenir une configuration cohérente, certains services ont été annulés.

Solution : vérifiez les valeurs des identifiants (noms ou port TCP) affectés à chacun des services.

ERR 212: host error. Host(s) may be reset.

Commande : setup_send

Cause : le paramétrage d'un ou plusieurs hosts du set-up est incomplet ou deux hosts du set-up ont le même nom.

Conséquence : pour obtenir une configuration cohérente, les noms de ces hosts ont été supprimés du set-up.

Solution : vérifiez les déclarations des hosts.

ERR 213: too many sessions or pages. Page number set to 1.

Commande : setup_send

Cause : la Platine supporte 8 'contextes d'écran'. Cette ressource est partagée entre les sessions et les pages par session (i.e. $\text{multi_nb_session} * \text{multi_nb_page} \leq 8$).

Conséquence : pour obtenir une configuration cohérente, le paramètre multi_nbpage a été positionné à 1.

Solution : vérifiez les déclarations des sessions et des pages.

10.5 - FORMAT DU FICHIER DE CONFIGURATION

Le fichier de configuration contient les valeurs d'une partie ou de la totalité des paramètres set-up d'une Platine AX3000 TCP/IP.

Le fichier doit toujours commencer par l'entête `BEGIN_AX_SETUP` et se terminer par le label `END_AX_SETUP`.

Exemple de fichier :

```
BEGIN_AX_SETUP V1.1
# ceci est un commentaire
tcp_host1_name=vangogh
...
END_AX_SETUP
```

Note : les lignes du fichier commençant par un dièse sont des commentaires.

Les paramètres set-up sont classés par groupe :

Ethernet :	paramètre liés à l'interface Ethernet,
TCP/IP :	environnement réseau,
Port AUX1 :	service et paramètres de communication associés,
Port AUX2 :	service et paramètres de communication associés,
Port Parallèle :	service associé,
Multi-session :	nbre de sessions, touches de changement de session...
Terminal :	paramètres généraux (type de clavier, d'écran...),
Session 1 :	paramètres spécifiques à la session 1 ,
...
Session 8 :	paramètres spécifiques à la session 8.

Une valeur affectée à un paramètre set-up répond à un format de donnée. Les formats possibles sont les suivants :

Liste de valeurs :	les valeurs possibles sont listées et séparées par le caractère ' '. Exemples : (yes no). (none xon-xoff xpc dtr)
Numérique :	Les bornes inférieures et supérieures sont données.

- Chaîne de caractères : La longueur maximale est spécifiée.
Note : un caractère peut être représenté par son code ASCII hexadécimal précédé par un anti-slash (ex. : \1B pour Escape et \\ pour un anti-slash).
- Format adresse IP : Le format d'une adresse IP est xxx.xxx.xxx.xxx.
- Format 8 couleurs : Les valeurs disponibles sont `black`, `red`, `green`, `brown`, `blue`, `magenta`, `cyan` et `white`.
- Format 16 couleurs : 8 valeurs supplémentaires sont disponibles : `grey`, `lt-red`, `lt-green`, `yellow`, `lt-blue`, `lt-magenta`, `lt-cyan` et `hi-white`.

Dans les listes qui suivent, le format ou les valeurs possibles d'un paramètre set-up sont données, après ce paramètre, entre parenthèses.

Un fichier de configuration peut être créé au moyen d'un éditeur de texte ou récupéré à partir d'une Platine déjà configurée (commande `rcmd setup_get`). Lorsque le fichier de configuration est obtenu par cette commande tous les paramètres set-up non significatifs (hosts non définis, service d'impression non utilisé, colorisation sur un écran monochrome...) sont exprimés sous forme de commentaire.

Dans la suite de ce chapitre, la présentation du fichier de configuration est conforme à celle obtenue par la commande `rcmd setup_get`.

Entête

```
BEGIN_AX_SETUP V1.1
#####
#          TCP/IP AX3000 Platine Terminal          #
#                                                  #
# Ethernet address: 00:A0:34:00:27:10             #
# Firmware: FK3.BV1.1a/TCP.FR.9701.a            #
#####
#
# RESET_CMOS
```

Note : la commande `RESET_CMOS` permet de ré-initialiser tous les paramètres set-up de la Platine (mis à part l'adresse IP) en début de configuration. Lorsque

le fichier de configuration est obtenu par commande `rcmd`, la commande `RESET_CMOS` est présente en début de fichier sous forme de commentaire.

Paramètres 'Ethernet'

```
#ethernet_ipaddr=192.168.1.241 (format adresse IP)
ethernet_noise=low           (low | high)
```

Note : Lorsque le fichier de configuration est obtenu par commande `rcmd`, le paramètre `ethernet_ipaddr` apparaît sous forme de commentaire (afin d'utiliser ce fichier pour configurer plusieurs Platines).

Paramètres 'TCP/IP'

```
tcp_host1_name=vangogh      (chaîne de 15 caractères max.)
tcp_host1_ip=192.168.1.252  (format adresse IP)
tcp_host1_telnetp=23       (numérique)
tcp_host2_name=picasso     (chaîne de 15 caractères max.)
tcp_host2_ip=192.168.1.248 (format adresse IP)
tcp_host2_telnetp=23       (numérique)
tcp_host3_name=pablo       (chaîne de 15 caractères max.)
tcp_host3_ip=192.168.1.249 (format adresse IP)
tcp_host3_telnetp=23       (numérique)
tcp_host4_name=vangogh     (chaîne de 15 caractères max.)
tcp_host4_ip=192.168.1.250 (format adresse IP)
tcp_host4_telnetp=23       (numérique)
#tcp_host5_name=           (chaîne de 15 caractères max.)
#tcp_host5_ip=             (format adresse IP)
#tcp_host5_telnetp=23     (numérique)
#tcp_host6_name=           (chaîne de 15 caractères max.)
#tcp_host6_ip=             (format adresse IP)
#tcp_host6_telnetp=23     (numérique)
#tcp_host7_name=           (chaîne de 15 caractères max.)
#tcp_host7_ip=             (format adresse IP)
#tcp_host7_telnetp=23     (numérique)
#tcp_host8_name=           (chaîne de 15 caractères max.)
#tcp_host8_ip=             (format adresse IP)
#tcp_host8_telnetp=23     (numérique)
tcp_router1_ip=192.168.1.252 (format adresse IP)
tcp_router1_target=192.1.1.252 (format adresse IP | default)
```



```
tcp_router1_mask=255.255.255.0 (format adresse IP)
tcp_router2_ip=192.168.1.253 (format adresse IP)
tcp_router2_target=192.18.1.2 (format adresse IP | default)
tcp_router2_mask=255.255.255.0 (format adresse IP)
tcp_router3_ip=0.0.0.0 (format adresse IP)
tcp_router3_target=default (format adresse IP | default)
tcp_router3_mask=0.0.0.0 (format adresse IP)
#tcp_router4_ip= (format adresse IP)
#tcp_router4_target= (format adresse IP | default)
#tcp_router4_mask= (format adresse IP)
#tcp_router5_ip= (format adresse IP)
#tcp_router5_target= (format adresse IP | default)
#tcp_router5_mask= (format adresse IP)
#tcp_router6_ip= (format adresse IP)
#tcp_router6_target= (format adresse IP | default)
#tcp_router6_mask= (format adresse IP)
#tcp_router7_ip= (format adresse IP)
#tcp_router7_target= (format adresse IP | default)
#tcp_router7_mask= (format adresse IP)
#tcp_router8_ip= (format adresse IP)
#tcp_router8_target= (format adresse IP | default)
#tcp_router8_mask= (format adresse IP)
tcp_mss=512 (numérique inférieur à 2049)
tcp_window=1000 (numérique inférieur à 2049)
tcp_ttl=255 (numérique inférieur à 256)
tcp_setport=fixed (fixed | random)
```

Note : Lorsque le fichier de configuration est obtenu par commande `rcmd`, les `hosts` et les routeurs non définis apparaissent sous forme de commentaire.

Notes sur les routeurs :

- le paramètre `tcp_router%_ip` est l'adresse IP du routeur (sur le même réseau que la Platine), le paramètre `tcp_router%_target` est l'adresse IP du host ou du réseau à atteindre et le paramètre `tcp_router%_mask` est le masque réseau à appliquer à l'adresse IP du réseau à atteindre.
- le routeur par défaut du set-up de la Platine est codé par le paramètre `tcp_router%_ip` égal à `0.0.0.0`, le paramètre `tcp_router%_target` égal à `default` et le paramètre `tcp_router%_mask` égal à `0.0.0.0`.

Paramètres 'Ports AUX1 et AUX2'

Pour faciliter la lecture de ce document, le caractère '%' présent dans les paramètres suivants varie de 1 à 2 et représente le numéro du port auxiliaire. Mais dans le fichier de configuration, chaque port auxiliaire à ses propres paramètres.

```

aux%_service=none           (none | rtty | printd | lpd
                           rcmd | rtelnet | tty)
aux%_service_name=aux1     (chaîne de 8 caractères max.)
aux%_service_num=2048      (numérique)
aux%_filter_nl=no          (yes | no)
aux%_associate_host=vangogh (chaîne de 8 caractères max.)
aux%_associate_term=ansi   (chaîne de 8 caractères max.)
aux%_associate_autoconn=yes (yes | no)
aux%_associate_reconn=yes  (yes | no)
aux%_use=bi-directional    (printer | bi-directional)
aux%_speed=38400           (300 | 600 | 1200 | 2400 | 4800
                           9600 | 19200 | 38400 | 57600
                           115200)
aux%_data                   (7-1-none | 7-1-odd | 7-1-even
                           8-1-none | 8-1-odd | 8-1-even)
aux%_tx_hdsk=dtr           (none | xon-xoff | xpc | dtr)
aux%_rx_hdsk=dtr           (none | xon-xoff | xpc | dtr)
aux%_detect=none          (none | cts)
    
```

Note : selon le service associé à un port auxiliaire (aux%_service) certains paramètres ne sont pas significatifs : les paramètres non cochés dans le tableau suivant apparaissent donc sous forme de commentaire :

paramètre aux%...	Services disponibles						
	none	printd	rtty	lpd	rcmd	rtelnet	tty
service_name				✓	✓		
service_num		✓	✓				
filter_nl		✓	✓	✓			
associate_host						✓	✓
associate_term						✓	
associate_autoconn						✓	
associate_reconn						✓	✓

Paramètres 'Port parallèle'

```
parallel_service=none          (none | rtty | printd |
                               lpd | rcmd)
parallel_service_name=parallel (chaîne de 8 caractères max.)
parallel_service_num=2050     (numérique)
parallel_filter_nl=no        (yes | no)
```

Note : selon le service associé au port parallèle (parallel_service) certains paramètres ne sont pas significatifs : les paramètres non cochés dans le tableau suivant apparaissent donc sous forme de commentaire :

	Services disponibles					
	none	printd	rtty	lpd	rcmd	tty
parallel_service_name				✓	✓	
parallel_service_num		✓	✓			
parallel_filter_nl		✓	✓	✓		
parallel_associate_host						✓
parallel_associate_reconn						✓

Paramètres 'Multi-session'

```
multi_nbsession=4            (numérique compris entre 1 et 8)
multi_nbpage=2              (numérique compris entre 1 et 8)
multi_statusline=yes       (yes | no)
multi_intro=alt            (alt | shift | ctrl - alt-shift
                           ctrl-shift | ctrl-alt
                           ctrl-alt-shift)

multi_s1=112               (numérique entre 2 et 123)
multi_s2=112               (numérique entre 2 et 123)
multi_s3=112               (numérique entre 2 et 123)
multi_s4=112               (numérique entre 2 et 123)
multi_s5=112               (numérique entre 2 et 123)
multi_s6=112               (numérique entre 2 et 123)
multi_s7=112               (numérique entre 2 et 123)
multi_s8=112               (numérique entre 2 et 123)
```

Paramètres 'Terminal'

```
term_screen=color          (black-white | color)
```

```

                                grey-levels | paper-white)
term_screen_enhanced=no        (no | underline | doublesize)
term_keyboard=french           (french | german | italian
                                spanish | belgian | english
                                american | portuguese | dutch
                                swiss-german | swiss-french |
                                turk-q | turk-f | iceland)
term_screensaver=yes          (no | yes | yes-kbd-only)
term_screensaver_delay=2      (numérique inférieur à 31)
term_numlock=on               (on | off)
term_capslock=on              (on | off)
term_beeper=long              (no | long | short)
term_keydelay=mediun          (low | medium | high)
term_keyspeed=low             (low | medium | high)
term_language=french          (french | english)
term_defaultport=aux1         (none | aux1 | aux2 | parallel)

```

Note : lorsque le fichier de configuration est obtenu par commande `rcmd`, le paramètre `term_screensaver_delay` apparaît sous forme de commentaire si le paramètre `term_screensaver` est égal à `no`.

Paramètres 'Session'

Pour faciliter la lecture de ce document, le caractère '%' présent les paramètres suivants varie de 1 à 8 et représente le numéro de session. Mais dans le fichier de configuration, chaque session à ses propres paramètres.

```

s%_associate_host=vangogh      (chaîne de 15 caractères max.)
s%_associate_term=ansi         (chaîne de 15 caractères max.)
s%_associate_label=view 1     (chaîne de 10 caractères max.)
s%_associate_autoconn=yes     (yes | no)
s%_associate_reconn=yes       (yes | no)
s%_predefined_setup=ansi      (ansi | ansi dos | unix sco 3.2.2
                                unix sco 3.2.4 | sco openserver
                                xenix sco | unix svr4 | ansi mos
                                ansi interactive | ansi rs 6000
                                ansi data general | vt220
                                vt as400 | sm9400 | sm9412
                                ato300 | hft)

```

```

s%_screen_enhanced=no          (no | yes)
s%_screen_crlf=yes            (yes | no)
s%_screen_line=25             (25 | 24+1)
s%_screen_column=80           (80 | 132)
s%_screen_codepage=437        (437 | 850 | 860 | 8859 | 8859-sg
                               dec-multi | iso-7 | sm9400
                               ato300 | greek | 861 | 857 )
s%_screen_scroll=yes          (yes | no)
s%_screen_wrap=yes            (yes | no)
s%_screen_cursor=block        (line | half-block | block)
s%_screen_overscan=09         (no | nbr compris entre 0 et 63)
s%_fctn_monitor=no            (no | symbol-mode | hexa-mode)
s%_fctn_termprg=no            (yes | no)
s%_fctn_fctnbold=bold         (bold | doublesize)
s%_fctn_endprn=\1B[4i         (chaîne de 6 caractères max.)
s%_kbd_code=scancode           (ascii | scancode)
s%_kbd_capsmode=caps-lock     (caps-lock | shift-lock)
s%_kbd_localcompose=no        (no | remote | local)
s%_color_mode=yes             (yes | no)
s%_normal_foregrnd=lt-green   (format 16 couleurs)
s%_normal_backgrnd=black      (format 8 couleurs)
s%_reverse_foregrnd=lt-red    (format 16 couleurs)
s%_reverse_backgrnd=white     (format 8 couleurs)
s%_underscore_foregrnd=yellow (format 16 couleurs)
s%_underscore_backgrnd=black  (format 8 couleurs)
s%_graphics_foregrnd=hi-white (format 16 couleurs)
s%_graphics_backgrnd=magenta  (format 8 couleurs)
s%_tab=      X      X      X      X      X      X      X      X
                               (chaîne de 132 car., X=tabu.)
s%_udk_f1=                     (chaîne de 32 caractères maximum)
s%_udk_f2=                     (chaîne de 32 caractères maximum)
s%_udk_f3=                     (chaîne de 32 caractères maximum)
s%_udk_f4=                     (chaîne de 32 caractères maximum)
s%_udk_f5=                     (chaîne de 32 caractères maximum)
s%_udk_f6=                     (chaîne de 32 caractères maximum)
s%_udk_f7=                     (chaîne de 32 caractères maximum)
s%_udk_f8=                     (chaîne de 32 caractères maximum)
s%_udk_f9=                     (chaîne de 32 caractères maximum)
s%_udk_f10=                    (chaîne de 32 caractères maximum)

```

s%_udk_f11=	(chaîne de 32 caractères maximum)
s%_udk_f12=	(chaîne de 32 caractères maximum)
s%_udk_f13=	(chaîne de 32 caractères maximum)
s%_udk_f14=	(chaîne de 32 caractères maximum)
s%_udk_f15=	(chaîne de 32 caractères maximum)
s%_udk_f16=	(chaîne de 32 caractères maximum)
s%_udk_f17=	(chaîne de 32 caractères maximum)
s%_udk_f18=	(chaîne de 32 caractères maximum)
s%_udk_f19=	(chaîne de 32 caractères maximum)
s%_udk_f20=	(chaîne de 32 caractères maximum)
s%_udk_f21=	(chaîne de 32 caractères maximum)
s%_udk_f22=	(chaîne de 32 caractères maximum)
s%_udk_f23=	(chaîne de 32 caractères maximum)
s%_udk_f24=	(chaîne de 32 caractères maximum)
s%_udk_f25=	(chaîne de 32 caractères maximum)
s%_udk_f26=	(chaîne de 32 caractères maximum)
s%_udk_f27=	(chaîne de 32 caractères maximum)
s%_udk_f28=	(chaîne de 32 caractères maximum)
s%_udk_f29=	(chaîne de 32 caractères maximum)
s%_udk_f30=	(chaîne de 32 caractères maximum)
s%_udk_f31=	(chaîne de 32 caractères maximum)
s%_udk_f32=	(chaîne de 32 caractères maximum)
s%_udk_f33=	(chaîne de 32 caractères maximum)
s%_udk_f34=	(chaîne de 32 caractères maximum)
s%_udk_f35=	(chaîne de 32 caractères maximum)
s%_udk_f36=	(chaîne de 32 caractères maximum)
s%_udk_f37=	(chaîne de 32 caractères maximum)
s%_udk_f38=	(chaîne de 32 caractères maximum)
s%_udk_f39=	(chaîne de 32 caractères maximum)
s%_udk_f40=	(chaîne de 32 caractères maximum)
s%_udk_f41=	(chaîne de 32 caractères maximum)
s%_udk_f42=	(chaîne de 32 caractères maximum)
s%_udk_f43=	(chaîne de 32 caractères maximum)
s%_udk_f44=	(chaîne de 32 caractères maximum)
s%_udk_f45=	(chaîne de 32 caractères maximum)
s%_udk_f46=	(chaîne de 32 caractères maximum)
s%_udk_f47=	(chaîne de 32 caractères maximum)
s%_udk_f48=	(chaîne de 32 caractères maximum)
s%_udk_ins=	(chaîne de 32 caractères maximum)

```
s%_udk_end=                (chaîne de 32 caractères maximum)
s%_udk_dn=                  (chaîne de 32 caractères maximum)
s%_udk_pgdn=                (chaîne de 32 caractères maximum)
s%_udk_left=                (chaîne de 32 caractères maximum)
s%_udk_five=                (chaîne de 32 caractères maximum)
s%_udk_right=               (chaîne de 32 caractères maximum)
s%_udk_home=                (chaîne de 32 caractères maximum)
s%_udk_up=                  (chaîne de 32 caractères maximum)
s%_udk_pgup=                (chaîne de 32 caractères maximum)
s%_udk_minus=               (chaîne de 32 caractères maximum)
s%_udk_plus=                (chaîne de 32 caractères maximum)
s%_udk_del=                 (chaîne de 32 caractères maximum)
s%_udk_esc=                 (chaîne de 32 caractères maximum)
s%_udk_dot=                  (. | ,)
s%_udk_backspace=          (\08 | \7F | ansi | sm9400)
```

Notes :

- Le paramètre `s%_predefined_setup` permet de positionner directement la valeur de TOUS les autres paramètres relatifs au comportement terminal de cette session.
- Lorsque le fichier de configuration est obtenu par commande `rcmd` :
 - Les paramètres des sessions non valides (voir paramètre `multi_nbssession`) apparaissent sous forme de commentaires.
 - Les paramètres concernant la colorisation apparaissent sous forme de commentaire si le paramètre `s%_color_mode` est égal à `no`.
 - Les touches de fonction dont la valeur est celle par défaut (par rapport à `s%_predefined_setup`), apparaissent sous forme de commentaire.

Fin de fichier

Le fichier de configuration doit impérativement être terminé par le libellé suivant :

```
END_AX_SETUP
```

Si ce libellé n'est pas trouvé lors d'une commande de configuration, le set-up usine est rechargé (ERR 103 : toutes les modifications sont perdues).

- 11 -
TELECHARGEMENT DU FIRMWARE

La fonctionnalité de téléchargement permet de mettre à jour le firmware d'une Platine via le réseau.

Le téléchargement d'un firmware peut être utilisé par exemple pour mettre à jour les fonctionnalités d'une Platine ou pour ajouter de nouvelles fonctionnalités (émulations disponibles sur demande, nouveaux paramètres set-up...).

IMPORTANT : la condition de fonctionnement est que la Platine soit équipée d'une **flash memory**. Si ce n'est pas le cas (équipement PROM), la fonction de téléchargement n'est pas disponible au niveau du set-up de la Platine.

Deux méthodes sont disponibles pour télécharger le firmware d'une Platine :

- le protocole `tftp` : la localisation du fichier à télécharger doit être renseignée par l'opérateur.
- les protocoles `bootp` et `tftp` : la procédure est automatique, tous les paramètres nécessaires doivent être préalablement renseignés au niveau d'Unix.

Quel que soit la méthode utilisée, le principe est que la Platine télécharge le fichier de code du firmware depuis une machine du réseau (appelée machine `tftp`). A la suite de ce transfert, la Platine se ré-initialise automatiquement avec le nouveau firmware.

Chacune de ces deux méthodes peut être exécutée :

- soit par la commande `rcmd` lancée depuis n'importe quel point du réseau (administration à distance),
- soit depuis le set-up TCP/IP de la Platine à mettre à jour.

IMPORTANT : généralement la configuration de base d'un UNIX ne lance pas les protocoles `tftp` et `bootp`. Modifiez le fichier `/etc/inetd.conf` en

supprimant les commentaires des lignes concernant `tftp` et/ou `bootps` puis rebootez la machine (ou envoyez le signal 1 au process `inet`).

11.1 - RAPPEL SUR LA COMMANDE RCMD

La commande `rcmd` permet l'accès à distance de périphérique réseau. La syntaxe générale de cette commande est la suivante :

```
$ rcmd name keyword <RC>
```

- `name` : nom du périphérique réseau à accéder. Ce nom est référencé dans le fichier `/etc/hosts` (certains systèmes d'exploitation acceptent aussi l'adresse IP du périphérique).
- `keyword` : mot clé.

Note : la commande `rcmd` est aussi appelée `rsh` sur certains systèmes d'exploitation.

Pour pouvoir accéder une Platine TCP/IP via une commande `rcmd`, il est impératif que cette Platine ait une adresse IP valide. Une adresse IP peut être attribuée de deux manières :

- par le set-up TCP/IP.
- ou automatiquement par le protocole RARP (voir le chapitre 2.2.2).

11.2 - TELECHARGEMENT PAR TFTP SEUL

Pour effectuer un téléchargement il est nécessaire de préciser la localisation du fichier firmware (nom du fichier et adresse IP de la machine `tftp`).

11.2.1 - Administration à distance

La syntaxe de la commande `rcmd` utilisée est la suivante :

```
# rcmd axname ax_download file tftpIP routeIP <RC>
```

- `axname` : nom de la Platine à mettre à jour. Ce nom est référencé dans le fichier `/etc/hosts`,

- `ax_download` : mot clé pour le téléchargement du firmware,
- `file` : chemin et nom du fichier à télécharger,
- `tftpIP` : nom ou adresse IP de la machine tftp (le nom doit être référencé dans le set-up de la Platine cible)
- `routeIP` (paramètre optionnel) : nom ou adresse IP du routeur pour accéder à la machine tftp (le nom doit être référencé dans le set-up de la Platine cible). Ce routeur est directement accessible par la Platine.

Exemple 1 :

```
# rcmd ax3001 ax_download /usr/firm 192.168.1.249 <RC>
```

Exemple 2 : au travers d'un routeur

```
# rcmd ax3002 ax_download /usr/firm vangogh 192.1.1.200 <RC>
```

Messages d'erreur possibles (affichés sur la console de l'opérateur) :

- ERR 105: `invalid rcmd command` : le mot clé est absent de la commande ou est mal libellé.
- ERR 108: `invalid number of parameters` : le nombre de paramètres de la commande `rcmd` est incorrect.
- ERR 109: `invalid file length` : le nom du fichier firmware est trop long (plus de 31 caractères).
- ERR 110: `invalid server` : le nom de la machine tftp est inconnu.
- ERR 111: `invalid router` : le nom du routeur tftp est inconnu.

Note : d'autres messages d'erreur système peuvent apparaître (time-out de connexion par exemple). Consultez le manuel du système d'exploitation utilisé.

Si le téléchargement peut s'effectuer (Platine et machine tftp accessibles, fichier firmware présent...), le message `Downloading in progress...` est affiché sur la console de l'opérateur. Après quelques secondes, la Platine se ré-initialise automatiquement pour prendre en compte le nouveau firmware.

Note : Durant le téléchargement du fichier firmware, des messages s'affichent sur l'écran de la Platine cible (voir le chapitre 11.2.2).

Pour vérifier que le téléchargement s'est bien effectué, il est possible d'utiliser la commande 'rcmd ax_version' pour faire afficher la version du firmware d'une Platine :

```
# rcmd ax3001 ax_version <RC>
```

A la suite de cette commande la version du firmware s'affiche sur la console de l'opérateur. Par exemple :

```
FK3.BV1.1a/TCP.FR.9701.a
```

11.2.2 - Téléchargement depuis le set-up

Entrez dans le set-up TCP/IP de la Platine (<Ctrl><Alt><*) et sélectionnez le menu 'Flash'. Dans ce menu, sélectionnez l'option 'Télécharger (tftp)'.

Plusieurs renseignements doivent être saisis au clavier :

- Nom du fichier : chemin et nom du fichier à télécharger,
- AX3000 IP : cette adresse est utilisée durant la phase de transfert de fichier et peut être différente de l'adresse IP habituelle de la Platine,
- Serveur tftp IP : nom ou adresse IP de la machine tftp,
- Routeur tftp IP (paramètre optionnel) : nom ou adresse IP du routeur éventuel pour accéder à la machine tftp. Ce routeur est directement accessible par la Platine.

Lorsque le dernier champ de saisie est validé, la procédure de téléchargement de fichier est lancée et les messages suivants s'affichent sur l'écran de la Platine :

```
AX TFTP V1.1a  
Flash Key 3
```

Si le téléchargement peut s'effectuer (la machine tftp est accessible, le fichier à télécharger est présent et est un fichier de firmware Axel...), le firmware actuel est effacé :

```
Erasing code ...
```

Le nouveau firmware est ensuite téléchargé (chaque point qui s'affiche correspond à un bloc de 512 octets) :

```
Loading code
.....
.....
.....
.....
.....
code loaded
```

La Platine se ré-initialise automatiquement pour prendre en compte ce nouveau firmware.

IMPORTANT : la couche IP du protocole tftp de la Platine AX3000 ne gère pas la fragmentation/défragmentation. Ainsi, si un routeur par exemple fragmente les trames (mtu inférieur à 600 octets), il sera impossible d'effectuer le téléchargement.

En cas de problème, consultez le chapitre 11.4 pour en connaître la cause et les solutions possibles.

11.3 - TELECHARGEMENT PAR BOOTP ET TFTP

Cette procédure est composée de deux phases :

- `bootp` : demande d'information concernant le fichier à télécharger,
- `tftp` : transfert et mise à jour du firmware.

Le protocole `bootp` permet à une Platine d'interroger le réseau pour récupérer les informations concernant le code à télécharger (adresse IP de la machine `tftp`, nom du fichier...).

Les renseignements concernant la localisation du fichier à télécharger doivent être donc précisés au niveau de la machine `bootp`.

Note : la machine `bootp` et la machine `tftp` peuvent être dissociées.

Conditions requises à bootp :

- Le process `bootpd` doit être lancé sur la machine `bootp`.
- Le fichier paramètre du process `bootpd` (`/etc/bootptab` par défaut) doit contenir une entrée pour chaque Platine susceptible de demander le téléchargement de son firmware.
- La machine `bootp` est en général directement accessible par la Platine (sur le même réseau) sinon il est nécessaire de configurer une machine accessible avec un protocole `bootp relay`.

Les principales entrées du fichier paramètre du process `bootpd` sont les suivantes :

- `tc` : description du réseau
- `ht` : type du réseau
- `sa` : adresse IP de la machine où se trouve le fichier à télécharger
- `sm` : masque du réseau
- `gw` : routeur éventuel
- `vm` : magic number (utilisé en cas de routeur)
- `ha` : adresse Ethernet de la Platine à mettre à jour
- `ip` : adresse IP de la Platine à mettre à jour (cette adresse IP est utilisée seulement durant le transfert du code à télécharger)
- `bf` : nom du code à télécharger

Exemples de fichier de configuration de bootp :

a) Platine et serveur tftp sur le même réseau

```
default:hn=df=/etc/btdump:ht=ethernet:sa=192.168.1.252:to=auto:
axell:tc=default:ht=ethernet:ha=00A034000001:ip=192.168.1.242:bf
=/tmp/axel:
```

b) Platine et serveur tftp sur deux réseaux distincts

```
default:hn=df=/etc/btdump:ht=ethernet:sa=192.1.1.243:to=auto:
net1:tc=default:sm=255.255.255.000:gw=192.168.1.252:
axell:tc=net1:ht=ethernet:vm=rfc1048:ha=00A034000001:ip=192.168.
1.242:bf=/usr/axel/firm9645:
```

IMPORTANT : si un problème est intervenu lors d'un précédent téléchargement de firmware (quelque soit la méthode utilisée), la Platine en question ne possède plus de firmware valide. Pour récupérer un firmware, les protocoles bootp et tftp sont automatiquement lancés lors de la mise sous tension de cette Platine. Et ceci jusqu'au succès de l'opération de téléchargement de firmware.

11.3.1 - Administration à distance

La syntaxe de la commande `rcmd` utilisée est la suivante :

```
# rcmd axname ax_download <RC>
```

- axname : nom de la Platine à mettre à jour. Ce nom est référencé dans le fichier `/etc/hosts`,
- ax_download : mot clé pour le téléchargement du firmware.

Exemple :

```
# rcmd ax3001 ax_download <RC>
```

Messages d'erreur pouvant survenir durant l'exécution de `rcmd` (affichés sur la console de l'opérateur) :

- ERR 105: invalid rcmd command : le mot clé est absent de la commande ou est mal libellé,

Note : d'autres messages d'erreur système peuvent apparaître (time-out de connexion par exemple). Consultez le manuel du système d'exploitation utilisé.

Si le téléchargement peut s'effectuer (Platine machine tftp accessible, fichier firmware présent...), le message suivant est affiché 'Downloading in progress...'. Après quelques secondes, la Platine se ré-initialise automatiquement pour prendre en compte le nouveau firmware.

Note : Durant le téléchargement du fichier firmware, des messages s'affichent sur l'écran de la Platine (voir le chapitre 11.3.2).

Pour vérifier que le téléchargement s'est bien effectué, il est possible d'utiliser la commande 'rcmd ax_version' pour faire afficher la version du firmware d'une Platine :

```
# rcmd ax3001 ax_version <RC>
```

A la suite de cette commande la version du firmware s'affiche. Par exemple :

```
FK3.BV1.1a/TCP.FR.9701.a
```

11.3.2 - Exécution depuis le set-up

Pour télécharger un firmware par cette méthode, entrez dans le set-up TCP/IP de la Platine (<Ctrl><Alt><*>) et sélectionnez le menu 'Flash'. Dans ce menu, sélectionnez l'option 'Télécharger (bootp+tftp)'.

Pour lancer la procédure de téléchargement, répondez 'o' (oui) à la question 'Téléchargement ?'. Le message suivant s'affiche sur l'écran de la Platine :

```
AX BOOTP V1.1a  
Flash Key 3
```

Si une réponse à la requête de la Platine est reçue, les renseignements transmis par la machine bootp s'affichent :

```
AX3000 IP: 192.168.1.242  
bootp server name: vangogh  
bootp relay IP: 0.0.0.0  
tftp server IP: 192.1.1.254  
file name: /axel/firm9645  
tftp routeur IP: 192.168.1.252
```

Ensuite, la Platine se connecte automatiquement sur la machine tftp et lance le téléchargement (voir chapitre précédent).

En cas de problème, consultez le chapitre 11.3 pour en connaître la cause et les solutions possibles.

11.4 - PROBLEMES POSSIBLES ET SOLUTIONS

Ce chapitre donne la liste des erreurs qui peuvent être rencontrées durant le téléchargement d'un firmware. Cette liste n'est pas exhaustive. Si une erreur non référencées ici apparaît, contactez votre distributeur AXEL.

Après une erreur, il est nécessaire d'éteindre puis d'allumer la Platine pour reprendre la main.

11.4.1 - Phase bootp

En cas de problème durant la phase bootp, un message d'erreur s'affiche :

```
Bootp errno: xxx
```

- xxx est le numéro d'erreur.

Erreurs possibles :

- 0 : Problème relatif à la carte Ethernet.
- 1 : Pas de réponse à la requête bootp (la machine bootp est inaccessible ou est mal configurée).

Note : une erreur n'entraîne pas la perte du précédent firmware.

11.4.2 - Phase tftp

Les messages d'erreur durant la phase tftp peuvent provenir soit de la machine tftp soit de la Platine.

Selon la nature de l'erreur, il est possible que le firmware actuel soit effacé alors que le nouveau ne soit pas encore opérationnel. Dans ce cas, lors de la prochaine mise sous tension de la Platine, le téléchargement du firmware sera automatiquement déclenché (protocoles bootp et tftp).

Messages d'erreur issus de la machine tftp :

```
tftp errno: xxx label
```

- xxx est le numéro d'erreur,
- label est le libellé de l'erreur. Ce libellé est transmis par la machine tftp et varie selon le système d'exploitation de cette machine.

Erreurs possibles :

- 1 : File not found
- 2 : Access violation
- 3 : Disk full or allocation exceeded
- 4 : Illegal TFTP operation
- 5 : Unknown Transaction Identifier
- 6 : File already exists
- 7 : Illegal TFTP operation
- 0 : Autres erreurs, libellé varie selon l'erreur

Pour plus de précisions, sur la signification de ces erreurs, consultez la documentation de la machine `tftp`.

Messages d'erreur Platine, problème tftp :

```
tftp errno: xxx
```

- xxx est le numéro d'erreur.

Erreurs possibles :

- 10 : Problème relatif à la carte Ethernet.
- 11 : Pas de réponse à la requête tftp (la machine tftp est inaccessible ou est mal configurée).
- 12 : Erreur réseau (routeur ou hub éteint ou inaccessible).
- 13 : le fichier à télécharger n'est pas un firmware de Platine AXEL ou ne correspond pas au modèle de Platine.
- 14 : la taille du fichier à télécharger est inférieure à 256 octets. Ce n'est donc pas un fichier de firmware de Platine AXEL.
- 15 : checksum du fichier à télécharger incorrect.

NOTES PERSONNELLES

A large, empty rectangular box with a thin black border, intended for personal notes. The box is centered on the page and occupies most of the lower half of the document.



Zone d'activité d'Orsay-Courtabœuf
16 Avenue du Québec - BP 728 - 91962 LES ULIS Cedex
Tél. : 01.69.28.27.27 - Fax : 01.69.28.82.04