

Deuxième partie

Le logiciel de communication du NANORESEAU

Les échanges dans le NANORESEAU

Dans ce chapitre, on décrira la manière dont se font les échanges d'informations entre les différents éléments du réseau : postes et central. Cette description se fera à un niveau logique. Il n'est pas nécessaire d'aborder les aspects électroniques de ces échanges pour pouvoir écrire une application ou un programme pour le réseau.

Théoriquement, il n'y a pas de différence entre les postes et le central en ce qui concerne le protocole d'échange. Toutes les fonctionnalités décrites ci-dessous existent dans les postes comme au central. D'ailleurs, un poste peut être considéré, dans certains cas, comme le serveur d'un autre poste. C'est le cas, par exemple, lorsqu'on demande le programme ou l'écran d'un autre poste grâce à l'instruction CLONE de BASIC.

En fait, il existe une certaine dissymétrie entre les postes et le central dans la mesure où ce dernier gère l'ensemble du réseau et n'a pas le même microprocesseur que les postes.

Chaque élément du réseau porte un numéro fixe compris entre 0 et 31. Le central porte le numéro 0 et chaque poste porte un numéro déterminé par les interrupteurs 1 à 5 qui se trouvent à l'arrière du boîtier de communication NANORESEAU. Le numéro des postes est donc laissé à la discrétion de l'utilisateur. Il faut toutefois que deux postes n'aient pas le même numéro et qu'aucun poste n'ait le numéro 0.

Les cinq messages élémentaires et la mise en attente

Les communications se font toujours entre deux éléments du réseau : l'appelant et l'appelé. Toute communication commence par l'envoi par l'appelant d'un message de 12 à 60 octets, que nous appelons une "**consigne**". Cette consigne décrit complètement la requête de l'appelant. Ce premier envoi est nommé "**appel initial**".

Le poste appelé peut très bien ne pas répondre à cet appel si ses interruptions sont masquées ou si, tout simplement, il n'existe pas dans le réseau... Dans ce cas, le poste demandeur fait un traitement d'erreur approprié. Si l'appelé répond, deux cas peuvent se présenter :

Premier cas:

L'appelant devient "l'esclave" de l'appelé. Ce dernier a alors la maîtrise totale de l'échange et agit sur l'appelant comme il l'entend. Nous dirons que l'appelant est

"**en attente**" de l'appelé. Ce cas est le plus fréquent; c'est ce qui se passe notamment lorsqu' un poste demande quelque chose au central.

Second cas:

L'appelé est en attente de l'appelant. Cela se produit, par exemple, lorsque le central envoie un langage ou une application vers un poste (option "DIFFUSION" du menu au central).

Examinons en détail cette notion d'attente qui est fondamentale pour les échanges dans le réseau.

Le central ne peut pas être mis en attente par un poste. Seuls les postes peuvent être mis en attente par le central ou par un autre poste.

L'état d'attente d'un poste est caractérisé par un léger bruit de métronome qu'on entend dans le haut-parleur du moniteur.

Pour faciliter l'exposé qui suit, supposons que le poste A soit en attente de B qui peut être un poste ou le central.

Dans ce cas, le poste A ne peut pas entrer en communication avec un troisième interlocuteur. Un autre poste essayant de lui parler (c'est-à-dire de lui envoyer un appel initial) ne recevra aucune réponse. Ceci garantit l'intégrité des échanges entre les deux interlocuteurs.

Sous attente, le poste A peut recevoir quatre types de messages de la part de B :

- Le poste A peut recevoir une consigne. Tout comme pour l'appel initial, cette consigne est un message de 12 à 60 octets. Cet envoi est nommé "**appel sous attente**". De façon générale, il sert à contrôler la suite des échanges (spécifier l'adresse et/ou la longueur des données qui vont être lues ou écrites, transmettre un compte rendu, provoquer l'exécution dans le poste d'une courte séquence de code 6809, etc.).
- Le poste A peut recevoir des données. Il faut pour cela que B lui envoie un ordre que nous appelons "**vas-y reçois**". L'adresse où ranger ces données et leur longueur sont déterminées par la dernière consigne reçue par le poste A. Les données reçues ont une longueur quelconque comprise entre 1 octet et 64 kilo-octets.
- Le poste A peut également émettre des données. Il faut pour cela que B lui envoie un ordre "**vas-y émet**". Comme pour l'ordre "vas-y reçois", l'adresse dans le poste et la longueur des données sont déterminées par la dernière consigne reçue par le poste A. Les données envoyées ont une longueur quelconque comprise entre 1 octet et 64 kilo-octets. Le fonctionnement des deux ordres "vas-y reçois" et "vas-y émet" est tout à fait semblable.

- Le poste A ne peut sortir de cette attente que lorsque B lui en donne la permission en lui envoyant un très court message que nous appelons un ordre "déconnecte-toi".

La constitution d'un jeu de primitives

Les cinq types de messages élémentaires que l'on vient de décrire permettent de construire des échanges complexes entre les postes et le central.

Prenons un exemple.

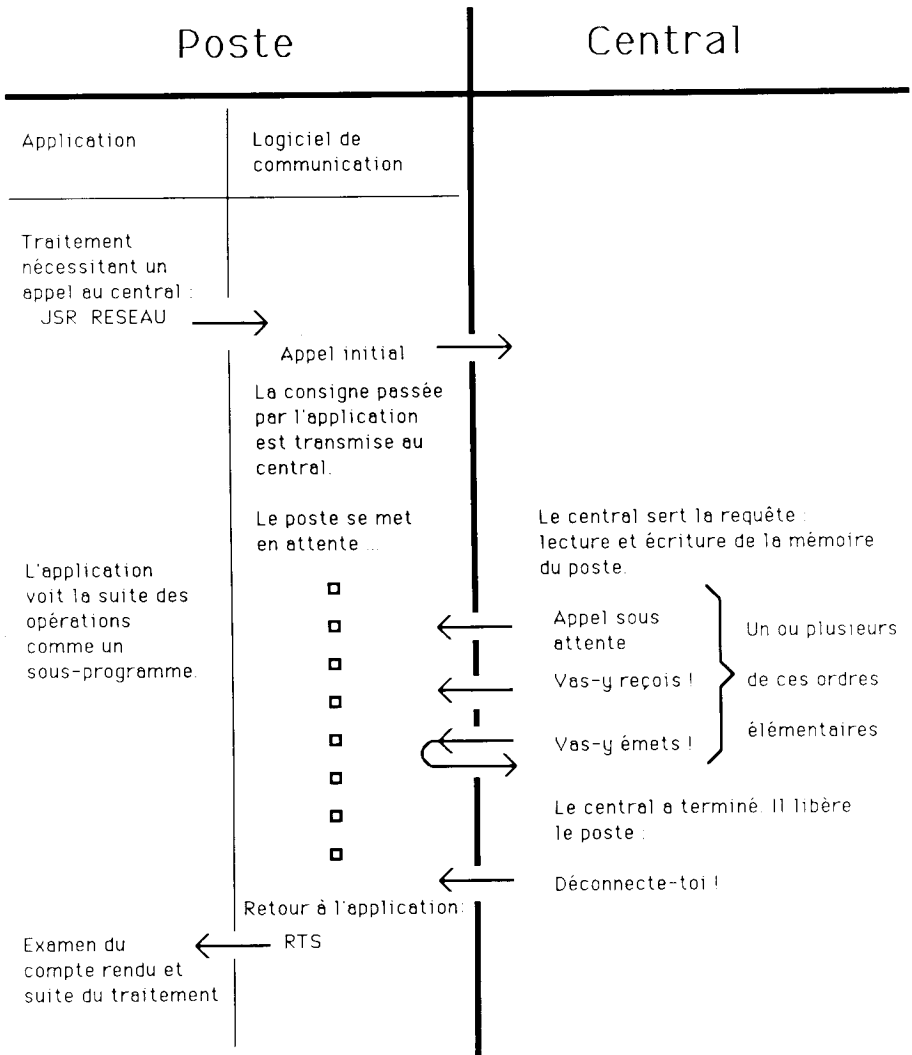
Examinons le comportement d'une application dans un poste du réseau. Au cours d'un traitement particulier, cette application doit faire une requête au central : elle prépare une consigne proprement formatée et envoie cette consigne vers le central par un appel initial. Pour l'application, cela se traduit par l'appel d'un sous-programme d'émission du logiciel de communication :

JSR` RESEAU

Cette consigne est envoyée au central et le poste se met en attente; le micro-processeur du poste "tourne" dans le logiciel de communication tant que l'échange n'est pas terminé. Pendant cette attente, le central peut lire ou écrire dans la mémoire du poste par le jeu des ordres élémentaires "vas-y émet", "vas-y reçois" et "appel sous attente". Lorsque la requête est servie, le central envoie un ordre "déconnecte-toi" qui fait sortir le poste de son attente et qui équivaut pour l'application au retour du sous-programme RESEAU qu'elle a appelé.

Ce mécanisme est décrit par le schéma numéro 3.

Schéma 3 - Les échanges dans le NANORESEAU



Pour l'application, toute l'opération s'est déroulée comme un unique sous-programme. A l'entrée de ce sous-programme, les informations nécessaires au bon fonctionnement de la requête sont passées par l'intermédiaire de la zone consigne. A la sortie, des informations se trouvent dans une zone spéciale dite zone de "**compte rendu**". De façon systématique, le premier octet de ce compte rendu est nul si l'opération s'est bien déroulée; dans le cas contraire, il contient le code de l'erreur survenue.

A partir de ce mécanisme d'échange entre les postes et le central, nous avons développé un jeu de primitives qui mettent à la disposition de chaque poste du réseau un système d'exploitation aussi puissant que MS-DOS.

Structure des consignes

Examinons à présent la structure des consignes. Cette structure est commune aux appels initiaux et aux appels sous attente. Chaque consigne se compose d'au moins 11 octets dont la position et la signification sont figées, suivis d'un nombre variable d'octets dont la signification dépend du contexte :

Offset	Nombre d'octets	Signification
+0	[1]	Longueur de la consigne
+1	[1]	Destinataire
+2	[1]	Code tâche réseau
+3	[1]	Code tâche application
+4	[2]	Nombre d'octets du message
+6	[1]	Page (généralement nulle)
+7	[2]	Adresse du message
+9	[1]	Ordinateur
+10	[1]	Application
+11	[?]	Octets dépendant du contexte

La **longueur** de la consigne est comptée à partir de l'octet "destinataire". Ce doit être un multiple de 4.

Le **destinataire** de l'appel est un nombre compris entre 0 et 31. Il est nul si l'appel est destiné au central; sinon, c'est le numéro du poste appelé.

Le "**code tâche réseau**" et le "**code tâche application**" permettent de caractériser la requête contenue dans la consigne. Soyons plus précis :

Les codes tâche réseau compris entre 0 et 6 sont réservés au logiciel de communication dans les postes (à l'exception du code 0 qui a une signification particulière lorsqu'il est envoyé vers le central).

0	Pas d'opération (au poste) Demande d'initialisation d'un poste (au central)
1	Mise en attente
2	Exécution de code 6809 dans le poste
3	Affichage sur l'écran du poste
4	Envoi de l'écran d'un poste
5	Envoi de mémoire
6	Recopie du compte rendu

Ces codes tâche réseau sont décrits en détail dans le chapitre suivant.

Les codes tâche réseau supérieurs à 6 désignent des ensembles de fonctions évoluées; le code tâche application permet alors de désigner la fonction invoquée à l'intérieur de chaque ensemble.

Un certain nombre de codes tâche réseau ont été déjà attribués :

- 7 Utilisé par BASIC 1.0
- 8 Réserve - non utilisé pour l'instant
- 9 Le système de gestion des fichiers du NANORESEAU
- 10 Réserve - non utilisé pour l'instant
- 11 Le gestionnaire d'imprimante

Le système de gestion des fichiers et le gestionnaire d'imprimante constituent l'essentiel du système d'exploitation NANORESEAU.

L'attribution de nouveaux codes tâche réseau pour certaines applications spécifiques est faite par la société Léanord.

Le nombre d'octets du message, l'adresse du message et la page permettent au poste de savoir où se trouvent les données à envoyer ou à recevoir lors du prochain "vas-y émet" ou "vas-y reçois".

Le logiciel de gestion du réseau dans le MO5 et dans le TO7 gère le basculement des deux pages écran (forme et couleur) et donne à l'utilisateur la possibilité d'inclure la gestion d'autres pages dans le traitement standard. Nous conseillons cependant aux développeurs de mettre l'octet page à 0 et d'effectuer les changements de page dans le poste avant de faire appel au réseau.

L'octet "**ordinateur**" de la consigne est initialisé par le logiciel de communication lui-même. Il prend les valeurs suivantes :

- 0 = TO7
- 1 = MO5
- 2 = TO7/70

L'octet "**application**" est également initialisé par le logiciel de communication et prend les valeurs suivantes :

- 0 = application non référencée (la majorité)
- 1 = BASIC 1.0
- 2 = LOGO
- 3 = LSE

C'est la copie d'un octet de la RAM réseau initialisé par le central lors du chargement des langages ou des applications dans les postes.

Cet octet "application" est utilisé au cours des échanges entre postes. Grâce à cet octet, l'application qui reçoit une requête peut déterminer si celle-ci s'adresse bien à elle et refuser de la traiter dans le cas contraire. Si un utilisateur travaillant sous BASIC demande le programme d'un poste travaillant sous LSE, il ne se

passer rien ! En effet, LSE teste l'octet application de la consigne et refuse de répondre dans ce cas.

L'attribution d'un code pour des applications futures est faite par la société Léanord. Il n'est pas utile que toutes les applications disposent d'un code particulier; cela ne se justifie que pour celles qui procèdent à des échanges de données entre les postes du réseau.

Enfin, les **octets dépendant du contexte** n'ont pas une signification figée. Ils contiennent des informations supplémentaires nécessaires au traitement de la fonctionnalité correspondant au code tâche réseau et au code tâche application. Ce peut être du code à exécuter par le microprocesseur du poste, un compte rendu, le nom d'un fichier à créer sur le disque du central, etc.

Le logiciel de communication dans les postes

Le boîtier NANORESEAU qui se met à l'arrière du poste a les mêmes caractéristiques que le contrôleur de disque pour MO5 ou TO7. Il contient un logiciel de communication de 2 kilo-octets environ, situé aux adresses réservées au contrôleur de disque, et il utilise les mêmes ports d'entrées/sorties que ce dernier.

Il en résulte qu'un poste ne peut pas avoir en même temps un contrôleur disque et un contrôleur NANORESEAU. Le disque et le réseau sont exclusifs l'un de l'autre.

Le logiciel de communication se trouve dans une EPROM; celle-ci contient en fait deux versions de ce logiciel : une version pour le MO5 et une version pour le TO7. Il n'existe qu'un seul modèle de boîtier NANORESEAU valable pour le MO5 comme pour le TO7. L'interrupteur numéro 7, à l'arrière du boîtier, permet de le mettre dans le mode TO7 (position OPEN) ou dans le mode MO5.

La RAM réseau

Initialement, le MO5 et le TO7 n'ont pas été conçus pour fonctionner en réseau. Le réseau a été développé après coup à une époque où les spécifications de ces machines étaient figées depuis longtemps et où un certain nombre d'applications existaient déjà. Le réseau a besoin d'une mémoire de travail de 150 octets environ mais rien n'était prévu pour lui dans l'organisation de la mémoire du MO5 ou du TO7. Certains octets normalement utilisés par le disque ont été tout naturellement récupérés par le réseau, mais c'était insuffisant.

La mémoire écran du MO5 et du TO7 se compose de deux pages de 8 K situées aux mêmes adresses : une page forme et une page couleur. L'écran de ces machines contient 25 lignes de 40 caractères et chaque caractère est défini par 64 pixels (soit 8 octets), ce qui fait un total de $25 \times 40 \times 8 = 8\,000$ octets. Le moniteur n'utilise donc que 8 000 octets dans la page forme et dans la page couleur pour la gestion de l'écran. Il reste $8 \times 1024 - 8\,000 = 192$ octets libres dans chacune des deux pages. Le réseau s'est emparé des 192 octets non utilisés de la page écran forme pour ses besoins propres. Nous appellerons cette mémoire de travail du réseau la "RAM réseau". Elle est située aux adresses \$1F40 à \$1FFF pour le MO5 et \$5F40 à \$5FFF pour le TO7.

Pour accéder à cette mémoire, il faut prendre la précaution de la rendre visible au microprocesseur. Il suffit pour cela de positionner un bit d'un PIA à l'aide du sous-programme suivant :

PORT	EQU	\$A7C0	; pour le MO5
PORT	EQU	\$E7C3	; pour le TO7
	LDB	PORT	
	ORB	#1	
	STB	PORT	
	RTS		

Ce qui donne, en BASIC et pour le MO5 :

POKE &HA7C0 , PEEK (&HA7C0) OR 1

Initialisation des postes

Que se passe-t-il à l'allumage d'un poste dans le réseau ? Le mécanisme est immuable : le moniteur initialise sa page 0 et les entrées/sorties de la machine, teste la présence du contrôleur de disque puis passe la main à l'application.

Les adresses réservées au contrôleur de disque sont comprises entre \$A000 et \$A7BF pour le MO5 et entre \$E000 et E7BF pour le TO7. Les trois premiers octets de cette plage mémoire sont "DKC" ou "DKD". La présence de ces trois lettres et de l'octet suivant qui est une somme de contrôle indique au moniteur que la machine est connectée à un contrôleur de disque; il positionne alors le booléen DKFLAG à \$FF pour le signaler à l'application.

Comme nous l'avons déjà dit, le contrôleur NANORESEAU occupe les mêmes adresses que le contrôleur de disque. Les quatre premiers octets de cette zone mémoire sont "REC" et la somme de contrôle appropriée. Lors de l'initialisation du poste, le moniteur croit être en présence d'un contrôleur de disque et positionne le booléen DKFLAG en conséquence avant de passer la main à l'application.

Reprenons l'examen de notre MO5 ou de notre TO7 connecté à un lecteur de disques. L'application qui vient d'être lancée par le moniteur a la possibilité d'aller chercher sur la disquette qui se trouve dans le lecteur 0 un complément de code. C'est ainsi que BASIC commence par lire sur la disquette système le complément de BASIC qui gère les fichiers sur disque. Un point d'entrée spécial du contrôleur de disque permet d'effectuer ce chargement de code. Il s'agit de DKBOOT situé à l'adresse \$A007 pour le MO5 et \$E007 pour le TO7.

Si on considère maintenant le même MO5 ou TO7 connecté à un réseau, l'application (BASIC en particulier), croyant avoir un lecteur de disques, ira tout naturellement lire un complément de code en faisant appel au point d'entrée DKBOOT. Et c'est ainsi que le logiciel de communication prend la main !

En effet, il existe dans le contrôleur NANORESEAU un sous-programme situé à la même adresse que DKBOOT et portant le même nom. Ce sous-programme initialise les circuits de communication, met certains pointeurs de la RAM réseau à des valeurs standard et tente d'entrer en communication avec le central. Il

envoie au central les 32 octets d'identification de l'application présente dans le poste (adresses \$EFE0 à \$EFFF pour le MO5 et \$0000 à \$001F pour le TO7). Ces 32 octets d'identification sont appelés la "signature" de l'application (terminologie qu'on retrouve dans l'utilitaire CONFIGNR).

Si le logiciel de communication dans le poste ne reçoit aucune réponse de la part du serveur, il met l'octet DKFLAG à 0 et relance l'application par son point d'entrée à froid (JMP [\$EFFE] pour le MO5 et JMP [\$001E] pour le TO7). L'application se retrouve donc à la case départ. Tout se passe comme si le moniteur venait de lui donner la main dans une machine sans contrôleur de disque ou de réseau (avec le booléen DKFLAG à 0).

Dans le cas contraire, le central a bien reçu le message provenant du poste et deux possibilités sont alors à envisager. Si le central trouve dans le fichier NR3.DAT l'identification de l'application, il télécharge dans le poste le fichier binaire associé à cette identification et en lance l'exécution. Sinon, il envoie au poste un ordre "déconnecte-toi", ce qui provoque un simple retour du sous-programme DKBOOT.

Examinons en détail les différents cas pouvant se présenter en fonction de la configuration matérielle et logicielle au niveau du poste.

Cas du MO5 avec cartouche de RAM

A l'allumage de la machine, c'est le BASIC résident qui est visible. Il procède à certaines initialisations et consulte DKFLAG qui a été mis à \$FF par le moniteur. Croyant être en présence d'un lecteur de disques, il passe la main au point d'entrée DKBOOT.

Le logiciel de communication initialise alors le poste dans le réseau et envoie l'identification du BASIC résident au central.

Si le central ne répond pas, l'octet DKFLAG est remis à zéro et le BASIC est relancé par son point d'entrée à froid. Dans ce cas, il considère qu'il n'y a pas de réseau et termine ses initialisations. Cela explique qu'on soit parfois en présence d'un poste sous BASIC ne communiquant pas avec le central (c'est-à-dire d'un BASIC ne disposant pas des primitives fichiers adaptées au réseau). Il suffit alors de s'assurer que le central fonctionne correctement et de donner l'ordre "DOS" pour relancer le processus d'initialisation du poste dans le réseau.

Sinon, le central télécharge dans la mémoire du poste le fichier binaire associé à l'identification envoyée. Ce peut être :

- Le complément de BASIC adapté au réseau (le fichier BASIC.MO5). Dans ce cas, la cartouche de RAM n'est pas utilisée.
- Un menu présentant un choix de différents langages ou applications. C'est le contenu du MENU.MO5 qui se met dans la mémoire non dédoublée du MO5. Ici aussi, la cartouche de RAM n'est pas utilisée.

- Un langage ou une application qui se charge dans la cartouche de RAM.

Le fichier binaire qui est ainsi téléchargé dans le poste au démarrage peut être spécifié par l'utilisateur grâce à l'utilitaire CONFIGNR.

Comme on le constate, dans le cas d'un MO5 avec une cartouche de RAM, le BASIC a un double rôle de *bootstrap* du système et de langage en tant que tel. C'est pour s'accommoder d'une situation antérieure à la conception du réseau qu'il en est ainsi.

Une application destinée à fonctionner dans la cartouche de RAM doit procéder à certaines initialisations. Celles-ci sont décrites plus loin et dans le chapitre "Ecrire une application pour le NANORESEAU".

Cas du MO5 avec cartouche de ROM

Dans ce cas, c'est le logiciel contenu dans cette ROM qui est visible à l'allumage de la machine. Deux sous-cas peuvent se présenter :

- L'application dans la cartouche de ROM est autonome et sait gérer le réseau. Son identification ne doit pas figurer dans le fichier NR3.DAT. Elle doit faire un appel au sous-programme DKBOOT pour initialiser le poste. Le central ne reconnaissant pas son identification provoquera dans le poste un simple retour de ce sous-programme et l'application reprendra la main à l'instruction suivant l'appel au sous-programme DKBOOT.

- L'application dans la cartouche de ROM exige un complément. Ce cas est analogue à celui du BASIC résident. L'identification de cette application doit se trouver dans le fichier NR3.DAT avec le nom du fichier binaire contenant ce complément. L'application doit faire un saut au point d'entrée DKBOOT pour initialiser le poste.

- Si le central ne répond pas, l'application sera relancée par son point d'entrée à froid avec l'octet DKFLAG mis à zéro.

- Sinon, le central reconnaissant l'identification de l'application téléchargera le complément dans la mémoire du poste et lancera le poste à l'adresse d'exécution contenue dans le fichier binaire. Nous ne conseillons pas aux développeurs d'utiliser cette possibilité (cartouche de ROM + complément RAM) car cela implique la livraison d'une ROM, d'une disquette et la nécessité pour l'utilisateur de modifier son fichier NR3.DAT pour y mettre l'identification de l'application et le nom du fichier binaire correspondant.

Cas du MO5 sans cartouche de RAM ni cartouche de ROM

Dans ce cas, le BASIC résident est toujours visible. On peut bien sûr demander le complément de BASIC et travailler sous BASIC dans le réseau.

On peut également demander le chargement d'un langage ou d'une application qui se contente des 32 K de RAM compris entre \$2000 et \$9FFF (la mémoire non dédoublée du MO5). C'est le cas de MENU et de NR-DOS.

Cas du TO7 avec cartouche de ROM

Ce cas est semblable à celui du MO5 avec cartouche de ROM. Le BASIC n'est pas résident dans le TO7 et se retrouve au même niveau que les autres applications.

Les pointeurs de la RAM réseau à initialiser

Pour pouvoir communiquer correctement avec le réseau, toute application doit initialiser trois pointeurs dans la RAM réseau. Cette initialisation doit être faite juste après le DKBOOT dans le cas d'une application en cartouche de ROM, ou immédiatement après le chargement dans le cas d'une application en cartouche de RAM.

Voici la description des pointeurs à initialiser, les adresses correspondantes étant données en annexe :

ADCRDU doit contenir l'adresse d'une zone mémoire structurée de la façon suivante :

ADCRDU -->	[1]	Nombre d'octets à recopier, soit n
	[n]	La place pour la recopie du compte rendu

Cette zone de recopie du compte rendu doit se trouver dans la partie de la mémoire de l'application qui n'est pas soumise à la commutation des banques de mémoire.

ADTRCTR doit contenir l'adresse de la routine qui traite les codes tâche réseau inconnus du logiciel de communication. Cela concerne plus particulièrement les applications qui gèrent les échanges entre postes. Pour les autres, il suffira de mettre dans ADTRCTR l'adresse d'une instruction RTS.

ADCTRLC doit pointer sur le traitement de reprise en main. De façon plus précise, lors d'un échange sur le réseau, le logiciel de communication procède à 256 essais successifs. Si ces 256 essais restent infructueux (le correspondant ne répond pas) ou si l'utilisateur appuie sur la touche <Ctrl-C> entre-temps, le logiciel de communication branche le micro-processeur sur la routine de reprise en main.

Le sous-programme RESEAU

C'est certainement le sous-programme le plus utilisé du logiciel de communication. Il permet de lancer un appel initial en direction d'un élément quelconque du réseau (central ou autre poste).

A l'entrée de ce sous-programme, le registre X du 6809 doit pointer sur une zone mémoire contenant une consigne correctement formatée. Cette consigne ne sera

pas modifiée par la suite des opérations. La structure des consignes permettant d'utiliser le système d'exploitation du NANORESEAU est donnée en annexe.

Le logiciel de communication essaie d'entrer en relation avec le destinataire de la consigne. Il procède à 256 essais successifs. Si ces essais sont infructueux, il branche le microprocesseur sur la routine de reprise en main pointée par ADCTRLC. Il fait de même si l'utilisateur a appuyé sur <Ctrl-C> pendant ces essais.

Si le correspondant répond, le poste appelant se met "en attente". Le poste appelé sert alors la requête du poste appelant. Dans la majorité des cas, cela se termine par l'envoi par le poste appelé d'un compte rendu et d'un ordre "déconnecte-toi". Le compte rendu est recopié par le logiciel de gestion du réseau dans la zone mémoire pointée par ADCRDU. Au retour du sous-programme RESEAU, l'application doit consulter la zone de compte rendu pour savoir comment l'opération s'est déroulée.

Le fonctionnement décrit ci-dessus est celui de presque toutes les primitives du système d'exploitation du NANORESEAU. Il faut toutefois signaler que les deux primitives CHBIN et SYSTEM ne se terminent pas par l'envoi d'un compte rendu mais par le téléchargement d'un module binaire et le lancement du microprocesseur à une adresse déterminée.

Une autre exception au fonctionnement général décrit ci-dessus se produit si l'échange a lieu entre deux postes et si le code tâche réseau envoyé par le poste appelant est le code de mise en attente. Dans ce cas, c'est le poste appelé qui se retrouve en attente et le poste appelant est libéré immédiatement et devient le maître de l'échange.

Les sous-programmes pour contrôler un autre poste sous attente

Cette section concerne essentiellement les échanges entre postes. On se place donc dans le cas où deux postes sont en communication, l'un étant en attente de l'autre.

- Cela se produit en particulier lorsqu'un appel initial venant du réseau contient un code tâche réseau inconnu du logiciel de communication. Dans ce cas, le contrôle est passé à l'application dans le poste appelé (plus précisément à la routine pointée par ADTRCTR).

- Cela se produit également si le poste appelant a mis le poste appelé en attente par un appel de RESEAU avec un code tâche réseau de mise en attente.

Le maître de l'échange dispose de quatre sous-programmes pour agir sur le poste en attente : EMAP, EMVE, EMVR et EMDISC décrits ci-après. Ces quatre routines ne protègent aucun registre. Au retour, le bit C du registre d'état du 6809 (carry) est positionné en cas d'erreur fatale.

Au cas où le traitement effectué entre deux appels à ces routines prendrait plus de 150 micro-secondes, il est **impératif** de procéder de la manière suivante :

- Mettre l'octet d'adresse \$2058 (MO5) ou \$6058 (TO7) à une valeur non nulle. Cela interdit d'éventuels appels initiaux venant des autres postes.
- Relâcher la ligne par un appel à RELACH.
- Effectuer le traitement qui prend plus de 150 micro-secondes.
- Lancer la procédure de prise de ligne par un appel à PRLGN.
- Remettre l'octet d'adresse \$2058 (MO5) ou \$6058 (TO7) à sa valeur initiale.
- Lancer la routine d'émission choisie (EMAP, EMVE, EMVR ou EMDISC).

EMAP

Ce sous-programme envoie vers le poste en attente un "appel sous attente". La consigne qui est transmise est celle qui se trouve dans la RAM réseau à l'adresse de la consigne courante (\$1F5D pour le MO5 et \$5F5D pour le TO7). Au retour de EMAP, les deux postes en dialogue ont la même consigne courante.

Cet appel sous attente peut servir à donner au poste en attente l'adresse et la longueur de la prochaine plage mémoire à émettre ou à recevoir, à envoyer une courte séquence de code à exécuter, à envoyer un compte rendu à recopier, etc.

EMVE

Ce sous-programme envoie vers le poste en attente un ordre "vas-y émet". Le poste en attente va envoyer des données vers le poste maître. Pour chacun des deux postes, l'adresse des données transmises est déterminée par l'adresse et la page de la consigne courante (\$1F65/\$5F65 pour l'adresse et \$1F64/\$5F64 pour la page). Bien entendu, ces adresses peuvent ne pas être identiques pour les deux postes. La longueur des données transmises est déterminée par la longueur de la consigne courante (\$1F62/\$5F62). La longueur doit être la même pour les deux postes, sinon l'échange ne peut pas se dérouler correctement.

EMVR

Ce sous-programme envoie vers le poste en attente un ordre "vas-y reçois". Le poste en attente va recevoir des données de la part du poste maître. Les considérations sur l'adresse, la page et la longueur des données sont les mêmes que pour le sous-programme EMVE.

EMDISC

Ce sous-programme envoie vers le poste en attente un ordre "déconnecte-toi" qui libère le poste de son attente et termine l'échange.

Le sous-programme de transfert d'écran

Le sous-programme OEE (adresse \$A03E pour le MO5 et \$E03E pour le TO7) permet de saisir l'écran d'un autre poste. Il suffit de mettre dans A le numéro du poste (de 1 à 31) dont on veut prendre l'écran et d'appeler ce sous-programme. Si le poste visé n'existe pas ou si l'utilisateur appuie sur <Ctrl-C>, le logiciel de communication branche à l'adresse de reprise en main pointée par ADCTRLC. Signalons que ce sous-programme réalise correctement les échanges d'écran entre MO5 et TO7 : si les machines sont de type différent, un traitement est fait sur la page écran couleur du poste récepteur pour que l'écran reçu soit identique à l'écran envoyé.

Le rôle de la consigne courante

Il existe dans la RAM réseau une zone mémoire appelée "consigne courante" qui joue un rôle important dans les échanges avec le réseau. Cette zone mémoire se trouve aux adresses \$1F5D à \$1F9F pour le MO5 et \$5F5D à \$5F9F pour le TO7. C'est dans cette zone mémoire que sont rangées les consignes reçues du réseau ou envoyées vers le réseau, que ces consignes proviennent d'un appel initial ou d'un appel sous attente. En d'autres termes, on trouve dans la zone consigne courante la dernière consigne reçue ou envoyée par le poste.

Ce sont les informations contenues dans les variables "longueur du message", "page" et "adresse du message" qui sont utilisées lors de l'exécution des ordres "vas-y émet" ou "vas-y reçois". Notons qu'après l'exécution d'un tel ordre, les variables "adresse du message" et "longueur du message" sont mises à jour de façon à permettre le chargement ou la lecture d'une plage mémoire par morceaux.

Réception d'une consigne par le logiciel de communication

Un poste peut recevoir une consigne par un appel initial ou par un appel sous attente. Dans les deux cas, il range cette consigne dans la consigne courante puis examine le code tâche réseau.

Si le code tâche réseau est compris entre 0 et 6, il est exécuté immédiatement par le logiciel de communication.

Si le code tâche réseau est compris entre 7 et 127, le logiciel de communication passe la main à l'application à l'adresse de la routine pointée par ADTRTCR.

Si le bit 7 du code tâche réseau est à 1, il n'est pris en compte qu'après la réception de l'ordre "déconnecte-toi" suivant. De façon plus précise, à la

réception d'un ordre "déconnecte-toi", le logiciel de communication examine le code tâche réseau qui se trouve dans la consigne courante. Si son bit 7 est à 1, il exécute le code tâche réseau formé par les bits 0 à 6 selon la procédure précédente.

Le numéro du poste qui est l'émetteur de la consigne n'est pas dans la consigne courante. On le trouve à l'adresse DESTI (\$2055 pour le MO5 et \$6055 pour le TO7).

Les codes tâche réseau connus du logiciel de communication

Pas d'opération (code = 0)

Ce code est un code neutre. Quand le logiciel de communication le reçoit, il ne fait rien. Il sert essentiellement à modifier la consigne courante du poste en attente et en particulier les variables "longueur du message", "page" et "adresse du message" utiles au prochain "vas-y émet" ou "vas-y reçois".

Prenons un exemple. Imaginons un poste A qui demande des données à un poste B. Il fera un appel initial en direction de A (RESEAU) et se mettra en attente de B. Après cet appel initial, la consigne courante du poste A sera la même que la consigne courante du poste B et contiendra en particulier l'adresse et la longueur du buffer de réception des données dans le poste A. Si le poste B s'aperçoit que les données à envoyer sont moins longues que le buffer de réception de A, il devra envoyer vers A un appel sous attente (EMAP) pour modifier la longueur contenue dans la consigne courante de A. Le code tâche réseau sera alors le code nul : après avoir rangé la consigne venant de B à l'adresse de la consigne courante, le logiciel de communication dans A se remettra en attente de B sans rien faire d'autre.

Notons que ce code tâche réseau prend une signification différente au central. Il est envoyé à ce dernier par le sous-programme DKBOOT et constitue donc une demande d'initialisation du poste. Les octets dépendant du contexte dans la consigne contiennent alors les 32 octets d'identification de l'application dans le poste (la signature de l'application) et permettent au central de déterminer l'action à faire en fonction du contenu du fichier NR3.DAT. Cette action consiste soit à ne rien faire, soit à télécharger un fichier binaire dans la mémoire du poste et à en lancer l'exécution.

Mise en attente (code = 1)

Ce code tâche réseau permet de mettre le poste appelé en attente. Il n'est effectif qu'en direction d'un poste du réseau; le central ne peut pas être mis en attente. Il n'a de sens que lors d'un appel initial, c'est-à-dire un appel par l'intermédiaire du

sous-programme RESEAU. Dans ce cas, le poste appelant est libéré immédiatement.

De façon plus précise, prenons l'exemple de deux postes A et B du réseau. Le poste A cherche à rentrer en communication avec le poste B. Pour cela, il utilise tout naturellement le sous-programme RESEAU. S'il envoie au poste B un code tâche réseau égal à 1 (mise en attente), c'est le poste B qui sera mis en attente et A sera le maître de l'échange. Dans le cas contraire, le poste A sera mis en attente et le poste B sera maître.

Ce code tâche réseau est utilisé par le central lors de la diffusion d'un programme ou d'une application en direction des postes.

Exécution de code dans le poste (code = 2)

Ce code tâche réseau permet de forcer le poste en attente à exécuter une courte séquence de code (48 octets au maximum) constituée par les octets dépendant du contexte de la consigne. De façon précise, le code à exécuter est placé dans la consigne courante à l'adresse \$1F69 pour le MO5 et \$5F69 pour le TO7.

Lors du téléchargement d'un langage ou d'une application dans un poste, c'est ce code tâche réseau qui permet au central d'effectuer le basculement des banques de mémoire et de lancer le microprocesseur à l'adresse d'exécution à la fin de l'opération.

Affichage sur l'écran (code = 3)

Ce code tâche réseau permet d'afficher sur l'écran du poste un message dont l'adresse se trouve dans la consigne courante ("adresse du message") et qui est terminé par le code \$04.

Envoi de l'écran (code = 4)

Ce code tâche réseau est utilisé de façon interne par le logiciel de communication. Il existe un point d'entrée spécial qui effectue les transferts d'écran entre les postes du réseau. C'est le sous-programme OEE situé à l'adresse \$A03E (MO5) ou \$E03E (TO7).

Envoi de mémoire (code = 5)

Ce code tâche réseau est utilisé de façon interne par le logiciel de communication.

Recopie du compte rendu (code = 6)

A la réception de ce code tâche réseau, le logiciel de communication recopie les octets dépendant du contexte de la consigne courante dans la zone de compte rendu de l'application. De façon plus précise, ADCRDU contient l'adresse d'une zone mémoire CRDU dont le premier octet est le nombre d'octets à recopier, la recopie se faisant à partir de l'adresse CRDU+1.

ADCRDU -->

CRDU [1]

nombre d'octets à recopier

[1]

premier octet du compte rendu

[1]

deuxième octet du compte rendu

...

etc.

Le logiciel de gestion du réseau au central

Les fichiers constituant le gestionnaire de réseau

Le nom du logiciel de gestion du NANORESEAU est NRxx.EXE, dans lequel xx désigne un numéro de version. La version 3.3 de ce logiciel s'appelle donc NR33.EXE. C'est un programme écrit en Assembleur 8088/86 sous le système d'exploitation MS-DOS avec contrôleur d'écran ANSI.

Le NANORESEAU est conçu pour fonctionner sur des centraux de marques différentes. Un point commun à toutes ces machines est leur compatibilité avec l'ordinateur personnel IBM qui constitue actuellement le standard du marché. Cette compatibilité a cependant des limites. Si la majeure partie logicielle de gestion du réseau est indépendante de la marque du central grâce, en particulier, au système d'exploitation qui est MS-DOS pour tout le monde, il n'en est pas de même dès qu'il s'agit de commander la carte d'interface NANORESEAU du central. Le pilotage de cette carte se fait à la micro-seconde près et les différences entre les machines entraînent des différences dans le logiciel qui gère les communications.

Le seul module qui dépend du central est donc le module de communication. Sur la disquette système NANORESEAU se trouvent plusieurs modules de communication correspondant à plusieurs centraux différents. Pour utiliser une machine donnée comme central, il suffit de copier le module de communication de cette machine sous le nom CTNR3.SYS (cf. le chapitre "Configuration du NANORESEAU"). Le fichier CTNR3.SYS est donc le seul fichier qui dépend du central.

Pour lancer le réseau, on doit avoir, sur le disque par défaut, les fichiers suivants:

NRxx.EXE	le gestionnaire de réseau
CTNR3.SYS	le module de communication
NR3.DAT	le fichier de configuration du réseau
PRINTER.SYS	le fichier de configuration du gestionnaire d'impression

Lorsqu'on lance le programme NRxx.EXE, le module de communication est intégré dynamiquement au système, puis les informations de configuration sont lues dans les fichiers NR3.DAT et PRINTER.SYS.

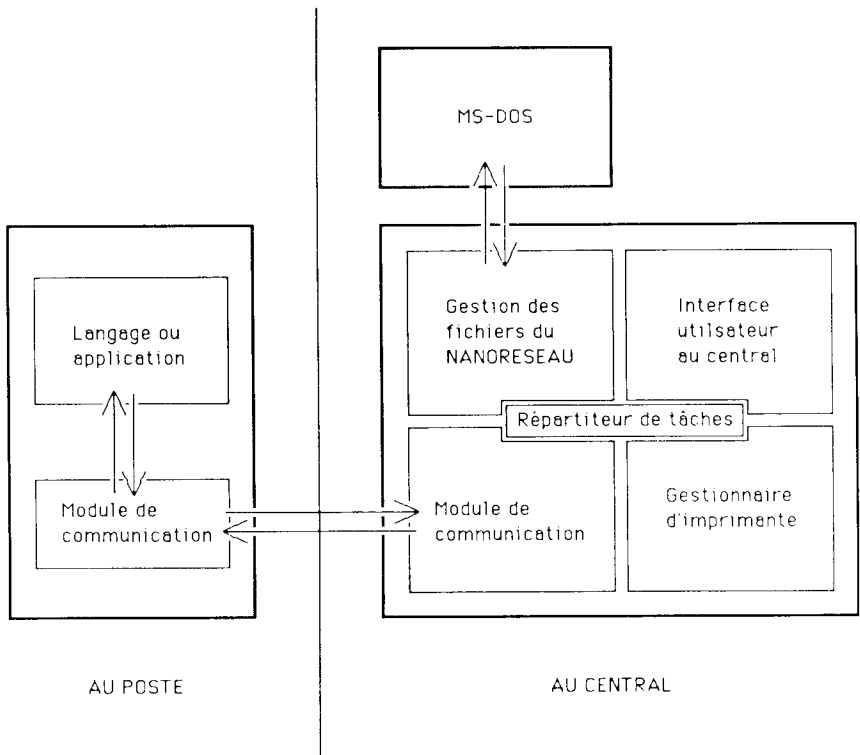
Notons que le fichier SPOOL.SYS n'est pas indispensable au lancement du système. Ce fichier contient la file d'attente des fichiers à imprimer. S'il est absent, le gestionnaire d'impression crée automatiquement un nouveau fichier SPOOL.SYS qui ne contient alors aucun fichier à imprimer.

Architecture logicielle au central

Le gestionnaire de réseau au central est logiquement constitué de cinq modules (cf. schéma numéro 4) :

- Le module de communication.
- Le module de gestion des fichiers NANORESEAU.
- Le module de gestion de l'imprimante.
- Le module d'interface avec l'utilisateur au central.
- Le répartiteur de tâches.

Schéma 4 - Architecture logicielle du NANORESEAU



Le module de communication

Ce module de communication est le pendant de celui qui se trouve dans les postes (dans l'EPROM du boîtier NANORESEAU) et offre les mêmes fonctionnalités. Il se compose de deux parties :

- Une partie qui reçoit les consignes venant du réseau sous interruption et qui les range dans une pile réservée à cet usage. Chaque requête est ensuite retirée de cette pile puis passée au module chargé de la traiter. Les consignes sont traitées dans l'ordre d'arrivée au central (Mode FIFO : *First In, First Out*).

- Une partie qui exécute les ordres d'émission ou de réceptions demandés par les autres modules au central. Comme dans les postes, on peut envoyer des appels initiaux, des appels sous attente, des ordres "vas-y émet", "vas-y reçois" et "deconnecte-toi" en direction des postes.

Il faut toutefois noter que, contrairement aux postes, le central ne peut pas être mis en attente.

Le gestionnaire des fichiers

Ce module sert d'interface entre les postes et MS-DOS. Il exécute tous les ordres fichiers venant des postes et prend en charge l'initialisation des postes dans le réseau.

C'est ce module qui réalise le système d'exploitation du NANORESEAU par le jeu d'un ensemble de primitives. Il partage MS-DOS (qui est un système d'exploitation mono-utilisateur) entre les différents postes du réseau.

La description des primitives du système d'exploitation du NANORESEAU est amplement faite dans cet ouvrage.

Le module de gestion de l'imprimante

Il traite toutes les requêtes venant des postes en direction de l'imprimante partagée. L'impression se fait en différé au travers de fichiers intermédiaires qui sont placés dans une file d'attente. Grâce à ce module, on peut imprimer des fichiers de texte avec différents transcodages et faire des copies d'écrans graphiques.

Ce module réalise également les quatre primitives de gestion de l'imprimante qui font partie du système d'exploitation du NANORESEAU.

L'utilisation pratique de l'imprimante du système est décrite dans le chapitre "La gestion de l'imprimante".

Le module d'interface avec l'utilisateur au central

C'est ce module qui affiche sur l'écran du central les actions sur le réseau proposées à l'utilisateur au central (Catalogue, Fichiers, Diffusion, Système, Impression et Quitter). Il lui donne ainsi le contrôle sur le réseau.

Les possibilités offertes par ce module sont décrites dans le chapitre "Ce qu'on peut faire depuis le central".

Le répartiteur de tâches

Ce petit module est en quelque sorte le cœur du gestionnaire de réseau dans la mesure où c'est dans ce module que le microprocesseur du central passe le plus clair de son temps. Ce module répète inlassablement les actions suivantes :

- Il teste si une consigne émanant d'un poste est arrivée au central. Dans l'affirmative, il passe cette consigne au module chargé de la traiter (gestionnaire de fichiers ou gestionnaire d'impression). A la fin de l'exécution de cette consigne, la main est rendue au répartiteur de tâches.
- Il passe la main au gestionnaire d'impression pour lui permettre d'envoyer quelques caractères vers l'imprimante, s'il y a lieu.
- Il teste si le clavier a été touché. Dans l'affirmative, il donne le caractère reçu au module d'interface avec l'utilisateur au central.

Grâce au répartiteur de tâches, le central peut donc faire plusieurs choses en même temps : gérer le réseau, gérer l'imprimante et répondre aux sollicitations de l'utilisateur devant son clavier.