

## 1. PROCOLES de type B.S.C.

### Basic Synchronous Communication

Ces protocoles anciens puisqu'ils datent de 1965-1966 pour la plupart s'insèrent mal dans le Modèle d'Interconnexion des Systèmes Ouverts de l'O.S.I. si on veut mettre en oeuvre toutes leurs fonctions. Légèrement réduits, ils se placent au niveau 2/OSI : Liaison de données.

La normalisation en ce domaine est insuffisamment contraignante et ne permet l'interconnexion de systèmes hétérogènes que si l'on utilise la version particulière d'un constructeur. En ce sens on utilisera souvent les procédures 2780 ou 3780 d'IBM disponibles sur de nombreux systèmes d'autres constructeurs. (Ces protocoles sont souvent à l'heure actuelle le seul moyen d'interconnecter ces systèmes..).

Nous citerons en référence :

- BSC norme ECMA 16 , ECMA 24, etc;
- AFNOR Z-66
- Bisync : 2780 , 3780 (IBM)
- VIP (Honeywell)
- DDC-MP (DEC)
- TMM-RB (Bull)
- etc.

Ce type de protocole a servi de modèle pour créer des protocoles de liaisons de données spécifiques, utilisant un alphabet de référence, sur des liaisons arytmiques. Ces protocoles sont utilisés, en particulier, dans des logiciels de communications entre des microordinateurs ou sur des "réseaux locaux industriels" de première génération.

### 1.1. Service physique requis.

service normal avec :

circuit de données **synchrone**  
**demi-duplex (ou duplex)**

Le coupleur doit en outre signaler la reconnaissance de caractères spéciaux de synchronisation (Syn), leur insertion automatique dans les blocs et l'extraction automatique correspondante pour gérer la fonction de "bourrage" mise en oeuvre lorsque la fonction émission ne peut assurer un débit suffisant de données au coupleur. Pour pouvoir gérer le mode "transparent" cette fonction devra pouvoir insérer et extraire deux caractères spéciaux.

## 1.2. Service fourni

Les protocoles de type BSC fournissent un service de **Liaison de Données point-à-point ou multipoint à commande centralisée**.

Ils offrent donc :

un transfert de données

bidirectionnel à l'alternat  
avec acquittement de chaque trame  
et contrôle de flux

un contrôle d'erreur

par parité croisée  
ou par code cyclique (CRC16)

des fonctions de connexion / déconnexion sous forme de **scrutations** (polling)  
des stations esclaves par la station maître :

Invitation à recevoir ou sélection  
Invitation à émettre ( ou polling)

une fonction de **synchronisation** et de contrôle du synchronisme.

Eventuellement

un transfert **transparent** peut être assuré.

## 1.3. Alphabet de référence

### 1.3.1. Introduction

Les protocoles BSC sont basés sur un alphabet de référence dans lequel sont codés les champs de l'information de contrôle de protocole (PCI) de chaque trame transmise.

Les données utilisateurs peuvent être codées dans cet alphabet ou, si l'on passe en mode **transparent**, dans n'importe quel autre.

L'alphabet de référence peut être quelconque mais doit réserver une **dizaine de caractères spéciaux** (au minimum) pour pouvoir assurer les fonctionnalités du protocole.

La normalisation requiert l'utilisation de l'alphabet international numéro 5 (CCITT T50, EIA 5, ECMA 6, ex-code ASCII). Cet alphabet réserve 10 caractères pour les transmissions notés TC1 à TC10.

déci m al		0	16	32	48	64	80	96	112
	hexa	0	1	2	3	4	5	6	7
+0	0	Nul	TC7 (DLE)	SP	0	@	P	`	p
+1	1	TC1 (SOH)	DC1 (Xon)	!	1	A	Q	a	q
+2	2	TC2 (STX)	DC2 (Tapeon)	"	2	B	R	b	r
+3	3	TC3 (ETX)	DC3 (Xoff)	# (£)	3	C	S	c	s
+4	4	TC4 (EOT)	DC4 (tapeoff)	\$ (¤)	4	D	T	d	t
+5	5	TC5 (ENQ)	TC8 (NAK)	%	5	E	U	e	u
+6	6	TC6 (ACK)	TC9 (SYN)	&	6	F	V	f	v
+7	7	BEL	TC10 (ETB)	'	7	G	W	g	w
+8	8	FE0 (BS)	CAN	(	8	H	X	h	x
+9	9	FE1 (HT)	EM	)	9	I	Y	i	y
+10	A	FE2 (LF)	SUB	*	:	J	Z	j	z
+11	B	FE3 (VT)	ESC	+	;	K	nat : 	k	nat : {
+12	C	FE4 (FF)	IS4 (FS)	,	<	L	nat : \	l	nat : 
+13	D	FE5 (CR)	IS3 (GS)	-	=	M	nat : 	m	nat : }
+14	E	S0	IS2 (RS)	.	>	N	nat : ^	n	nat : ~
+15	F	SI	IS1 (US)	/	?	0	_	o	• DEL

D'autres caractères sont réservés pour la mise en pages (FE0 à FE5 par exemple saut de ligne, retour chariot), à la segmentation des données utilisateurs (IS1 à IS4) ou au contrôle des terminaux (DC1 à DC4 par exemple xon : DC1 , x-off : DC3).

### 1.3.2. Fonctions et caractères spéciaux associés

Synchronisation: SYN code 16<sub>h</sub>

Connexion, Déconnexion, Scrutation ENQ code 05<sub>h</sub>  
EOT code 04<sub>h</sub>

Transfert de données	
Trames de données	SOH code 01 <sub>h</sub> STX code 02 <sub>h</sub> ETB code 17 <sub>h</sub> (fin de bloc) ETX code 03 <sub>h</sub> (dernier bloc d'un message)
Trames d'acquittement	ACK code 06 <sub>h</sub> NAK code 15 <sub>h</sub>
Préfixe pour modification de fonction DLE	code 10 <sub>h</sub>

L'utilisation de ces caractères sera explicitée sur des exemples lors de l'étude des fonctions.

### 1.3.3. Transparence en code

Ce problème sera étudié en fin de chapitre. Cette fonction permet de transporter des données utilisateurs indépendamment de l'alphabet de référence. Dans ce cas l'en-tête, écrite dans l'alphabet de référence se termine par la chaîne de 2 caractères : DLE STX et le champ de données utilisateur par DLE ETX ou DLE ETB. Le caractère DLE est traité de manière spécifique (technique du double DLE).

## 1.4. Etude fonctionnelle

### 1.4.1. Synchronisation

La fonction de synchronisation se situe à trois niveau :

porteuse  
bit  
caractère

Ces trois niveau sont du ressort de la couche physique. Seule la synchronisation caractère a une incidence sur la couche Liaison de Données. On se rapportera au chapitre "Modes de transmission".

Pour assurer la synchronisation caractère chaque trame, qu'elle soit de données ou de service doit être précédée d'une séquence de synchronisation, chaîne de 1 à 7 caractères SYN selon les protocoles.

En général on placera au moins deux caractères SYN. (protocole Bisync)

Le dispositif matériel de synchronisation émet automatiquement ces caractères et les extrait à la réception en le signalant au logiciel par un mot d'état convenable.

Si, en cours de transmission de transmission, l'émetteur ne peut fournir au coupleur ses données suffisamment vite, celui-ci insérera des caractères SYN en "bourrage" pour éviter la perte de synchronisation. Ces caractères sont automatiquement retirés à la réception et ne sont pas soumis au contrôle d'erreurs.

Si les trames sont longues, la synchronisation peut être contrôlée en plaçant des caractères SYN à des positions particulières de la trame ou à des intervalles de temps réguliers. Ceci permet aussi éventuellement de trouver une synchronisation tardive et de reconnaître la fin de trame pour pouvoir rejeter celle-ci.

### 1.4.2. Scrutations

Les protocoles de type BSC sont de type maître-esclave, la station maîtresse scrutant la station esclave pour l'**inviter à recevoir** (sélection) ou pour l'**inviter à émettre** (polling). Ceci est obligatoire pour une liaison multipoint à commande centralisée et se révèle simple à gérer pour des liaisons point-à-point entre une système central et des terminaux.

Pour une liaison point-à-point entre deux systèmes équivalents on peut parfois utiliser un mode de transmission symétrique.

#### 1.4.2.1. Invitation à recevoir :

La station maître ou primaire sollicite la ou une station secondaire qui accepte ou refuse la connexion en fonction des ressources dont elle dispose.

En fin de transmission la station primaire déconnecte la station secondaire.

En cours de transfert, la station secondaire peut demander une "suspension positive" du transfert de donner si elle ne dispose plus des ressources (buffers) nécessaires. Cette suspension ne peut être demandée qu'après la réception d'un bloc de données valide.

#### 1.4.2.2.. Invitation à émettre :

Deux modes de fonctionnement peuvent être utilisés :

\* dans le premier cas, la station secondaire, invitée à émettre, envoie immédiatement son premier bloc de données

\* dans le second, la station secondaire envoie une trame d'acceptation et ne transmet les données utilisateurs qu'après réception d'une trame d'envoi de données. Ce mode est

utile en particulier quand la station maîtresse joue le rôle de commutateur entre deux stations esclaves : elle envoie une invitation à émettre à la première, puis une invitation à recevoir à la seconde, enfin la commande d'envoi à la première ; elle est alors sûre de pouvoir acheminer les données de bout en bout.

Pour interrompre les échanges (et par exemple scruter d'autres stations) la station primaire peut suspendre à tout moment le transfert soit par une "suspension positive" si elle a reçu un bloc valide, soit par une "suspension négative" en cas de défaut de transmission.

### 1.4.3. Transfert de données

Le transfert de données de fait à l'alternat, chaque **trame de données** émise étant acquittée **positivement** si elle est bien reçue ou **négativement** dans le cas contraire.

Cette technique permet de réaliser le **contrôle de flux** :

La station source ne peut normalement émettre une nouvelle trame de données tant que l'accusé de réception de la trame précédente ne lui est pas parvenu. Une temporisation de surveillance permet de réaliser une reprise sur défaut lorsqu'une trame d'acquiescement est perdue.

Elle est utilisée pour la **correction des erreurs** :

Toutes les trames de **données** sont séquencées (modulo 8 ou modulo 2). A la réception on en contrôle la validité (redondance, numéro de séquence) et, en case d'erreur, on en demande la répétition par une trame d'accusé négative (NAK) portant le numéro de séquence de la **dernière trame bien reçue**. Lorsque la trame de données est bien reçue on sollicite l'envoi de la trame suivante par un accusé positif (ACK) comportant le numéro de la trame **bien reçue**.

Le contrôle d'erreurs est en général réalisé par un calcul de parité croisée portant sur tous les caractères de la trame sauf le premier ( et les caractères de bourrage). En cas de transfert transparent on utilise un code cyclique calculé à partir du polynôme CRC16.

Toute trame de données courante est terminée par le caractère ETB suivi du caractère de contrôle d'erreurs. La dernière trame est terminée par ETX suivi du caractère de contrôle d'erreurs.

### 1.4.4. Déconnexion

Après réception d'une trame finale (terminée par ETX) la station collecteur se déconnecte en émettant une trame composée du caractère EOT ou de la chaîne DLE EOT pour une liaison commutée que l'on veut "raccrocher".

## 1.5. Protocole ( exemple )

Un exemple de protocole est illustré sur les quatre schémas ci-dessous. Ils correspondent au protocole ancien TMM-RB de Bull et ne sont donnés que de manière indicative.

Le premier diagramme illustre une "**invitation à recevoir**". La station primaire émet une séquence de sélection (S) vers la station secondaire A. Si la station A ne peut recevoir de données, elle répond par une séquence négative (Nak), sinon elle répond positivement. La station primaire commence alors à transmettre sa trame 1 qui est acquittée positivement puis la trame 2, qui mal reçue, est rejetée par la station secondaire qui indique que la dernière trame bien reçue est la trame 1. La trame 2 est répétée. Ces trames sont terminées par Etb qui indique des trames à suivre. La dernière trame de la phase est terminée par Etx. Si la station secondaire ne peut plus recevoir de données elle émet une séquence de **suspension positive** Dle Ack.

Le second diagramme illustre une "**invitation à émettre**". La station primaire émet une séquence de polling (P) vers la station secondaire A. Celle-ci termine la phase de "polling" si elle n'a aucune donnée à transmettre (Eot); sinon elle commence à émettre ses données. La première trame porte le numéro 1. Celle-ci est acquittée positivement par la station primaire. La seconde trame est alors émise; elle est rejetée par la station primaire puis répétée. La terminaison de la trame 3 par ETX indique que la station secondaire n'a plus de données à émettre.

La station primaire peut demander une suspension positive mais aussi une suspension négative dans le cas où elle reçoit des données incohérentes et veut arrêter la communication.

Les deux diagrammes suivant reprennent ces phases de sélection et d'invitation à émettre, en y ajoutant le traitement des anomalies. On peut y observer qu'une trame de données mal transmise est répétée  $q$  fois. En cas de non réponse  $p$  essais sont effectués. Dans tous les cas la décision finale est laissée à la station primaire.

Diagramme d'invitation  
à recevoir :

Diagramme d'invitation à recevoir :

## 1.6. Mode transparent

Lorsque l'on veut transmettre des données utilisateurs binaires ou codées dans un alphabet différent de l'alphabet de référence on doit utiliser une procédure spéciale. En effet les codes réservés peuvent dans les données utilisateur et conduire à de fausses interprétations.

Souvent on utilise un codage hexadécimal. Un octet est décomposé en 2 quadruplet, chacun étant envoyé sous forme d'un caractère le représentant. Par exemple l'octet 04<sub>h</sub> est transmis par la chaîne "30<sub>h</sub> 34<sub>h</sub>" soit les caractères 0 et 4. Le rendement de la transmission est réduit de moitié. On peut améliorer ce rendement à 66% ou 75% par un codage plus sophistiqué à partir de mots de 16 ou 24bits codés sur 3 ou 4 caractères (par exemple "uuencode").

Pour avoir un rendement de l'ordre de 98 ou 99%, on utilise la technique du "**double DLE**".

Le champ de données est précédé du champ DLE STX et se termine par DLE ETB pour une trame courante ou DLE ETX pour la dernière trame d'une séquence.

Entre ces deux chaînes les données utilisateur peuvent être quelconques.

**A l'émission :**

Le protocole analyse les données à émettre et s'il rencontre un caractère codé  $10_h$  (DLE) il le fait suivre du même caractère DLE. A la place de DLE on utilise parfois le caractère \ codé  $5C_h$

Si un "bourrage" est nécessaire il sera d'insérer automatiquement la séquence DLE SYN.

**A la réception :**

On utilise l'algorithme ci-dessous :

Réception du caractère C.

Si  $C \neq \text{DLE}$  ranger C sinon recevoir caractère suivant K.

Si  $K = \text{DLE}$  ranger K sinon si  $K = \text{ETB}$  si  $K = \text{ETX}$  fin de trame

si  $K = \text{SYN}$  ignorer

sinon Défaut.

**Exemple :**

Chaîne à coder représentée dans l'alphabet de référence :

A B 1 ETB C STX 2 DLE 3 SYN D DLE DLE 4 ETX

Chaîne émise (les caractères gras sont ajoutés )

**DLE STX A B 1 ETB C STX 2 DLE DLE 3 SYN D DLE DLE DLE DLE 4 DLE SYN ETX F I N DLE ETX**

Réception :

DLE STX début de texte transparent A B 1 ETB C STX 2

ranger ces caractères DLE DLE

ranger DLE 3 SYN D

ranger ces caractères DLE DLE

ranger DLE DLE DLE

ranger DLE 4

ranger 4 DLE SYN

bourrage, ignorer ETX F I N

ranger ces caractères DLE ETX

Fin de texte transparent