

Les Transmissions séries asynchrones



Les transmissions séries asynchrones

1) Introduction:

Les communications entre différents systèmes informatiques se fait le plus souvent par des liaisons de types séries afin de limiter le câblage.

Ainsi, il y a lieu de transformer (nous disons *translater*) les informations "séries" en informations "parallèles" lorsque le microprocesseur reçoit des données d'une ligne de transmission et translater les informations parallèles en informations séries lorsque le microprocesseur doit émettre des données vers un périphérique. Des circuits intégrés réalisent de manière élégante ce type d'interfaçage.

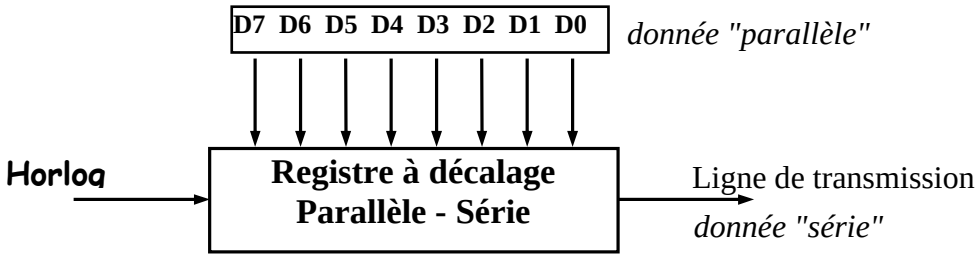
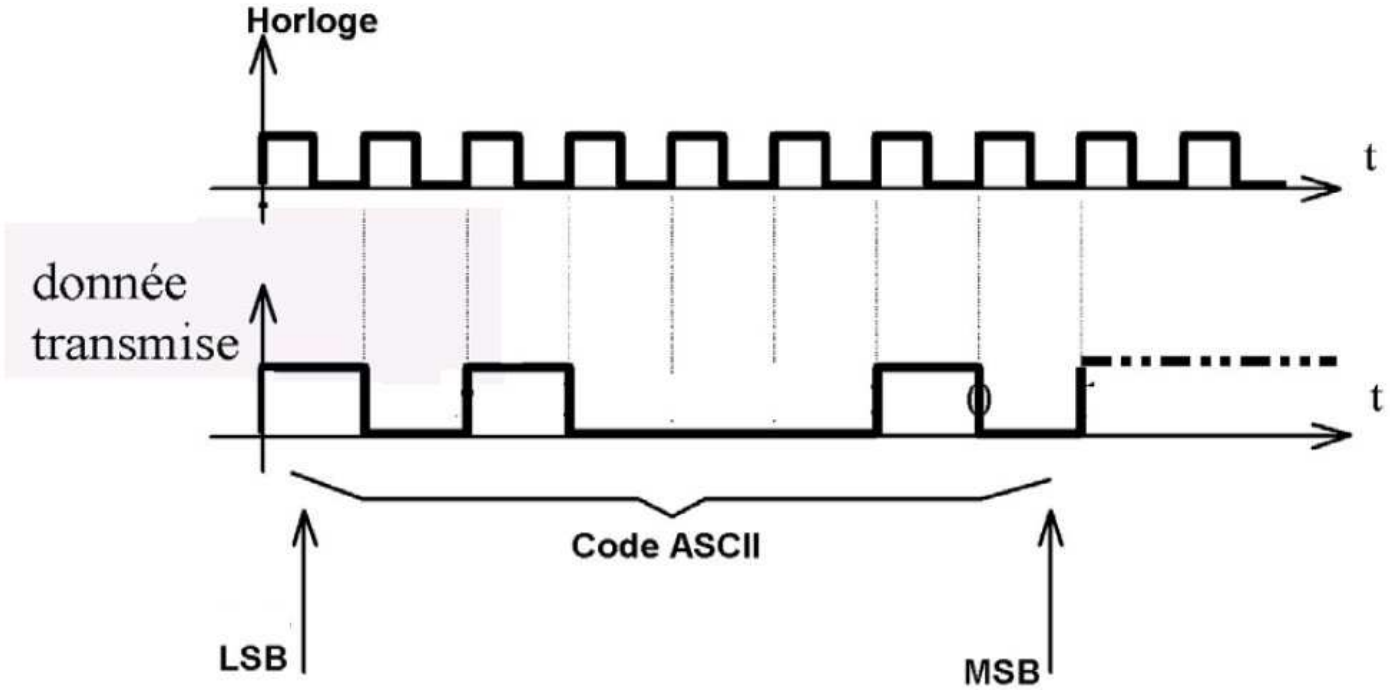


figure 8-02: principe de la translation d'une donnée parallèle

Considérons le cas de la transmission série du code ASCII de «E», soit l'octet $A2 10100010):



Suivant les constructeurs, pour des problèmes de licences, ils possèdent des appellations différentes:

- ✓ ACIA: Asynchronous Communications Interface Adapter chez Motorola,
- ✓ UART / USART: Universal Synchronous/Asynchronous Receiver Transmitter chez Intel,
- ✓ SIO: Serial Input Output chez Zilog …..

2) liaison série :

2.1) généralités:

En, un code normalisé associe aux différents caractères d'écriture un code binaire sur un octet, c'est le code ASCII[1].Tous les caractères imprimables, ainsi que les codes de contrôle (retour à la ligne, descente d'une ligne, début de texte, etc.) sont codés au format ASCII. Or, avec 8 bits (1 octet) nous pouvons représenter 256 caractères différents. Les 128 premiers sont internationaux et les 128 suivants dépendent du pays (page de code) et plus précisément de la langue.
Les 128 premiers caractères sont codés ainsi:

Table des codes ASCII (hexadécimal)

00 NUL	01 SOH	02 STX	03 ETX	04 EOT	05 ENQ	06 ACK	07 BEL
08 BS	09 HT	0A NL	0B VT	0C NP	0D CR	0E SO	0F SI
10 DLE	11 DC1	12 DC2	13 DC3	14 DC4	15 NAK	16 SYN	17 ETB
18 CAN	19 EM	1A SUB	1B ESC	1C FS	1D GS	1E RS	1F US
20 SP	21 !	22 "	23 #	24 $	25 %	26 &	27 '
28 (	29)	2A *	2B +	2C ,	2D -	2E .	2F /
30 0	31 1	32 2	33 3	34 4	35 5	36 6	37 7
38 8	39 9	3A :	3B ;	3C <	3D =	3E >	3F ?
40 @	41 A	42 B	43 C	44 D	45 E	46 F	47 G
48 H	49 I	4A J	4B K	4C L	4D M	4E N	4F O
50 P	51 Q	52 R	53 S	54 T	55 U	56 V	57 W
58 X	59 Y	5A Z	5B [	5C \	5D]	5E ^	5F _
60 `	61 a	62 b	63 c	64 d	65 e	66 f	67 g
68 h	69 i	6A j	6B k	6C l	6D m	6E n	6F o
70 p	71 q	72 r	73 s	74 t	75 u	76 v	77 w
78 x	79 y	7A z	7B {	7C |	7D }	7E ~	7F DEL

Tableau 8-02: code ASCII

Si on reprend l'exemple d'un «SOS», il faut transmettre en hexadécimal $(534F53)_{16}$.
Soit en binaire: $(0101\ 0011\ 0100\ 1111\ 0101\ 0011)_2$

[1] American Standard Code for Information Interchange créé en 1966.

3.1) Liaison SIMPLEX:

La liaison *simplex* est *toujours unidirectionnelle*. Le poste A émet et le poste B reçoit.



figure 8-06: liaison SIMPLEX

C'est le cas de certaines imprimantes. L'ordinateur envoie les caractères à imprimer et n'attend aucune réponse en retour.

3.2) Liaison SEMI-DUPLEX ou HALF-DUPLEX:

Le poste A émet vers le poste B, et le poste B peut émettre vers le poste A, mais un seul d'entre eux peut émettre à un instant donné.

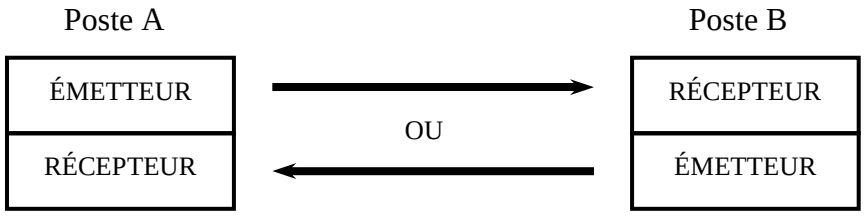


figure 8-07: liaison SEMI-DUPLEX

La liaison semi-duplex est utilisée lorsque le canal de communication ne possède pas une largeur de bande suffisante pour permettre des liaisons bilatérales.

3.3) Liaison DUPLEX:

Les deux postes A et B, peuvent émettre simultanément.

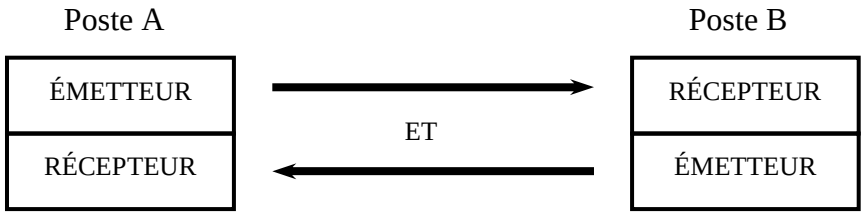


figure 8-08: liaison DUPLEX

4) types de transmission:

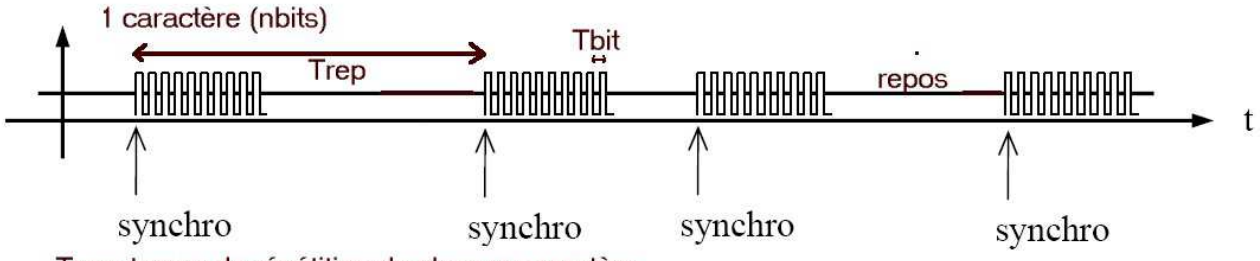
Nous abordons maintenant la question de la *chronologie* de l'émission d'une suite de données, ou "texte", en nous limitant au cas de liaison série. Cette chronologie est toujours

conditionnée par une durée T (période d'une horloge de référence) attachée à l'émission de chaque donnée binaire.

4.1) Transmission asynchrone:

Dans une transmission asynchrone, le rythme n'est respecté que pour l'émission des n bits composant chaque caractère. Les instants de démarrage des différents caractères sont, dans une large mesure, arbitraires. Afin que la réception puisse travailler correctement, il est nécessaire:

- ✓ de spécifier l'état de repos de la ligne;
- ✓ de faire précéder l'émission de chaque caractère par un signal particulier appelé *bit de START*;
- ✓ de maintenir la ligne à son état de repos pendant une durée minimale, appelée *durée de STOP*, entre la fin de la transmission d'un caractère et le bit de START du caractère suivant.



Trep: temps de répétition de chaque caractère
Trep asynchrone
Tbit=constante

trame en liaison asynchrone

En début de transmission, la ligne est au repos. Le récepteur vérifie que cet état se maintient pendant un temps au moins supérieur à la durée de STOP et attend l'arrivée du bit de START. Cette dernière déclenche la génération d'une salve de n bits de période Tbit, qui commandent le processus d'identification des n chiffres binaires du premier caractère. Le récepteur vérifie ensuite que la durée de STOP soit respectée, et il attend l'arrivée du START du caractère suivant ….

Le temps total consacré à la transmission d'un caractère est:

$$T_1 = \text{durée START} + n \,.\, Tbit + \text{durée STOP} \quad [8.01]$$

La seule contrainte sur la chronologie de la transmission des caractères est que le temps séparant la transmission de deux caractères doit être supérieure à un temps dit timeout.

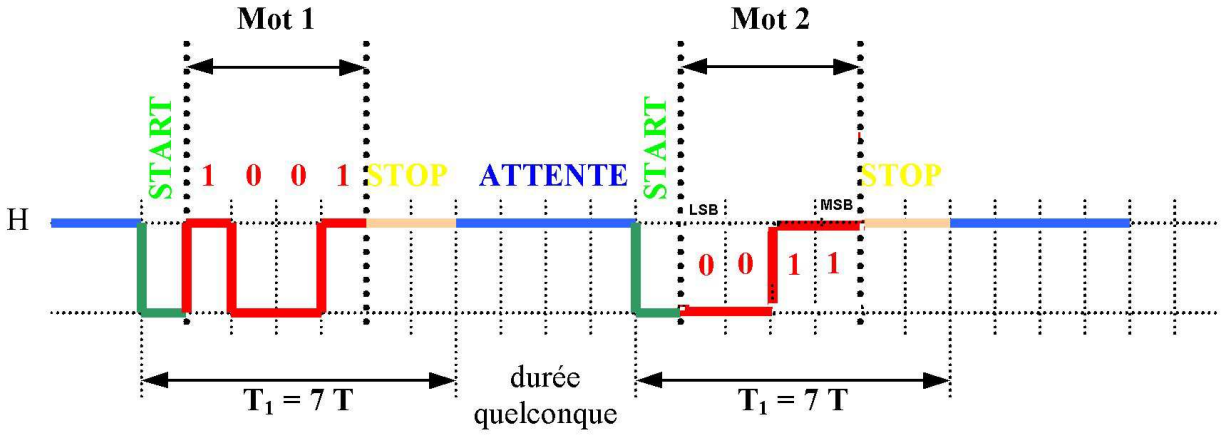
Si nous souhaitons protéger une liaison asynchrone contre les erreurs de transmission, nous effectuons un codage de chaque caractère émis. Comme la longueur des n bits est de seulement quelques unités, la probabilité d'erreur par caractère est faible et nous pouvons nous contenter d'un codage rudimentaire. Nous utilisons couramment le bit de *contrôle de PARITÉ*.

Le bit ***q*** représente la parité du nombre q_1 de bits informatifs ***égaux à 1***. Deux conventions sont utilisées:

- ✓ la parité ***PAIRE***: le bit de contrôle est égal à 0 si ***q1*** est impair, de sorte que le nombre de bits au niveau logique 1 d'un mot complet (bits informatifs + bit de parité) soit pair.
- ✓ la parité ***IMPAIRE***:nous utilisons la convention inverse. Le nombre de niveau logiques 1 du mot complet est impair.

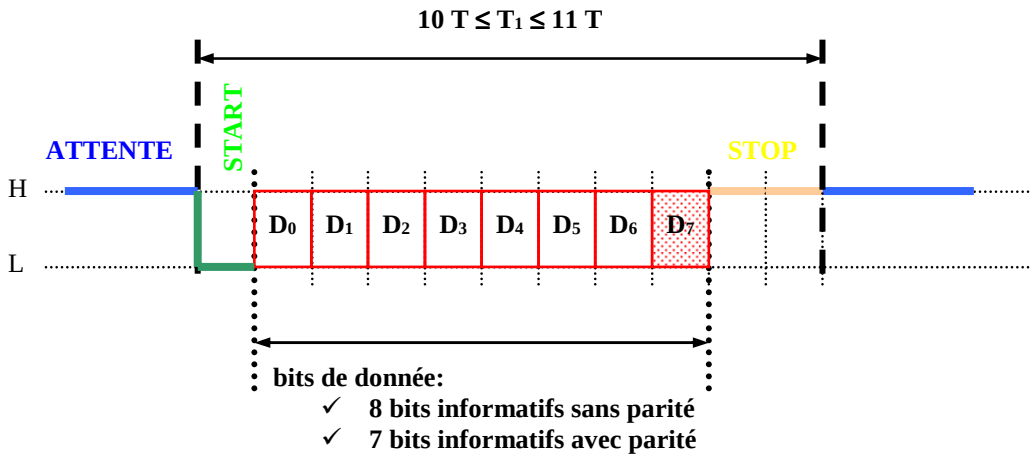
EXEMPLE: Transmission asynchrone de mots de 4 bits: %1001 et %1100 exemples de mots à 4 bits informatifs ou de mots à 3 bits informatifs et un bit de contrôle de parité paire.

codage NRZ en logique positive,
état de repos de la ligne: niveau HAUT,
START: niveau BAS de durée T,
STOP: niveau HAUT de durée 2T.



transmission série asynchrone de deux quartets

Format général d'une transmission d'un caractère:



transmission série asynchrone d'un caractère

Transmission synchrone:

Dans un système de transmission synchrone, l'émetteur et le récepteur sont synchronisés sur la même horloge de référence, de période ***T*** secondes, qui fonctionne pendant toute la durée de l'émission. Ce signal d'horloge est:

- ✓ soit transmis sur la ligne par un fil séparé,
- ✓ soit reconstitué, à la réception, à l'aide d'une séquence de synchronisation.

Reprenons l'exemple de la transmission série asynchrone du paragraphe précédent.

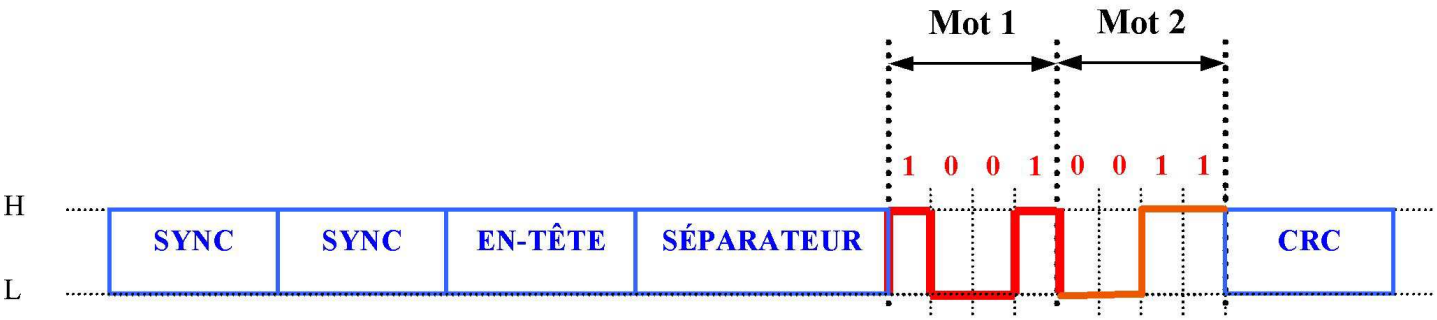


figure 8-12: transmission série synchrone de deux quartets

Le code le plus connu est le BISYNC d'IBM®:

- ✓ avec deux caractères de *synchronisation* pour prévenir le récepteur qu'un message arrive et lui permettre de se synchroniser sur l'horloge de l'émetteur;
- ✓ l'*en-tête* contient en général des informations de commande qui aident le récepteur à déterminer que faire du message. Le compte du nombre total d'octets du message est en général contenu dans l'en-tête;
- ✓ un *séparateur* de message,
- ✓ et une *séquence de contrôle cyclique* (CRC Cyclic Redundancy Check) qui suivent en général le "bloc message".

5) débit de transmission:

Nous avons vu que pour transmettre les données, le temps est divisé en intervalles d'égale durée. Le débit de transmission est mesuré

- ✓ soit en bauds avec: ***1 baud = 1 symbole / seconde.*** [8.02]
- ✓ soit en Bits/s avec: ***1 bits/s = 1 bit / seconde.*** [8.03]

Dans le cas de modulations simples, le débit de symboles correspond au débit de bits donc 1 baud = 1 bits/s. Dans le cas de modulations plus complexes ce n'est plus vrai. Prenons l'exemple d'une modulation par sauts de phase (PSK) de 90°: nous disposons de 4 angles, un par symbole; or avec 4 combinaisons nous pouvons coder les états de 2 bits ($2^2 = 4$). Alors le débit de bits est le double du débit de symboles (600 bauds mais 1200 bits/s).

Voici quelques vitesses de transmission courantes en bauds:

75 110 250 300 600 1200 2400 4800 9600 19200

Voici quelques vitesses de modems normalisés (avis V… du CCITT):

Tableau 8-03: débits des modems normalisés

Type	Modulation	Transmission	Mode	Bits/s	Fréquences émission (Hz)		Fréquences réception (Hz)	
					0	1	0	1
V21	FSK 2 états	Asynchrone	Full Duplex	300	1180	980	1850	1650
V22	PSK 4 états	Sync–Async	Half Duplex	1200				
V23	FSK 2 états	Async–Sync	Half Duplex	1200	1700	1300	1700	1300
V23 BACK	FSK 2 états	Async–Sync	Half Duplex	75	450	390	450	390
V26	PSK 4 états	Synchrone	Half Duplex	2400				
V27	PSK 8 états	Synchrone	Half Duplex	4800				
V29	A et PSK 16 états	Synchrone	Full Duplex	9600				
V32	A et PSK 32 états	Async–Sync	Full Duplex	9600				
V92	16QAM	Async–Sync	Full Duplex	56000				
ADSL	QAM DMT		Full Duplex	8Mb/s 512kb/s				
ADSL2+	QAM DMT		Full Duplex	25Mb/s 1.2Mb/s				

La norme V 23 en mode dissymétrique (1200 – 75 bits/s) correspond au Minitel par exemple. Actuellement les modems utilisent jusqu'à 128 symboles, ce qui permet de transmettre **7 bits par symbole** tout en limitant la bande passante du support de transmission.

6) Protocoles (Contrôle de flux):

Dans une liaison série asynchrone, si les données reçues par le récepteur peuvent être exploitées aussi rapidement qu'elles sont émises, une procédure ordonnant à l'émetteur de suspendre provisoirement son émission n'est pas nécessaire. Mais, la plupart du temps, le récepteur effectue un traitement des données reçues (affichage, impression, etc...) et ne peut suivre la cadence. Il dispose donc d'une mémoire tampon limitée permettant le stockage (avant traitement) au rythme de l'émission. Lorsque le tampon est plein, le récepteur doit demander à l'émetteur de suspendre l'émission. Le récepteur devra informer l'émetteur de la disponibilité de cette mémoire par une demande de reprise d'émission.

Différents protocoles permettent de réaliser cette gestion du flux de données. Il existe des protocoles matériels et des protocoles logiciels (employant des caractères ASCII).

6.1) Protocole RTS/CTS - DTR/DSR: Gestion de flux de données matérielle:

La gestion du flux est réalisée par des signaux autres que ceux de transmission et de réception. Ce sont les signaux RTS/CTS - DTR/DSR.
Request To Send ou *Data Terminal Ready*: Émission prête
Clear To Send ou *Data Set Ready*: Réception prête

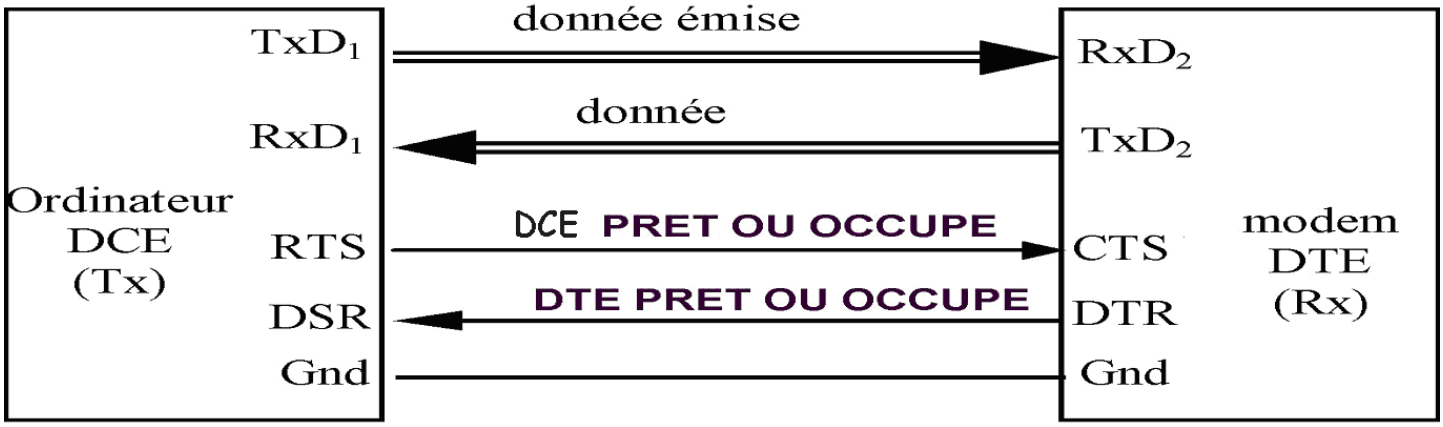


figure 8-13: liaison de base 5 fils DCE - DTE

Le principe de ce protocole est:

- Le *récepteur* (émetteur) informe l'*émetteur* (récepteur) qu'il ne peut plus recevoir de données donc que la transmission doit être suspendue en positionnant le signal *CTS* (DSR) au niveau haut (1L).
- Le *récepteur* (émetteur) signalera une demande de reprise de transmission à l'*émetteur* (récepteur): en positionnant le signal *CTS* (DSR) à l'état bas (0L).

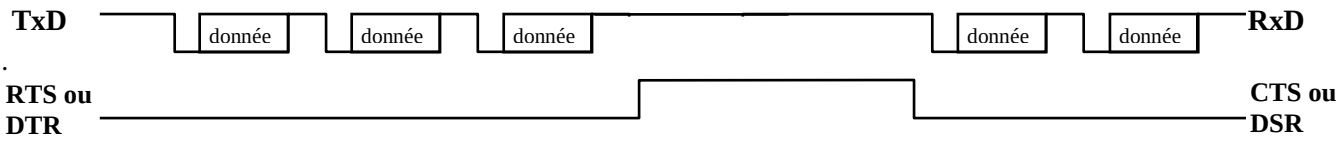


figure 8-14: chronogramme protocole DTR/DSR –

6.2) Protocole Xon/Xoff: Gestion de flux de données logicielle:

Cette gestion de flux logiciel est réalisée par émission des caractères de contrôle XON ($11 en ASCII) et XOFF ($13 en ASCII) sur les lignes de transmission de données.

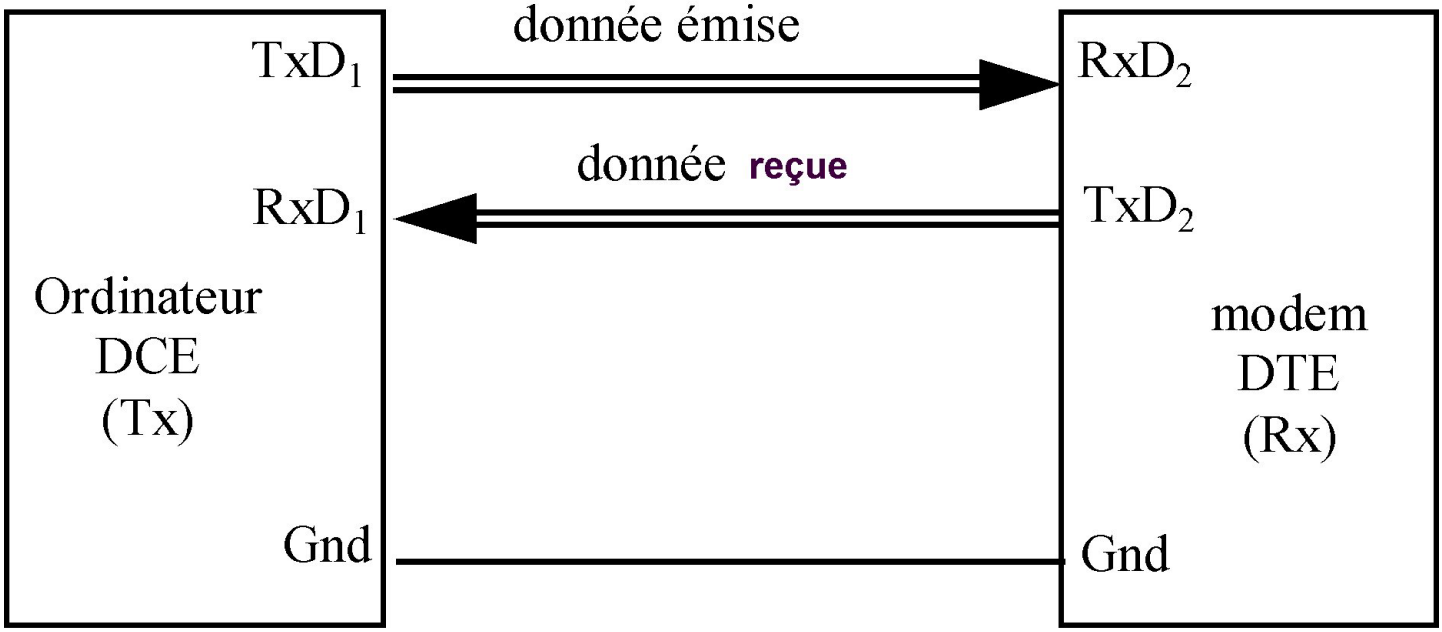


figure 8-15: liaison XON - XOFF

Lorsque la cadence d'émission est trop importante pour le récepteur, celui-ci demande la suspension de l'émission en envoyant le caractère XOFF sur la ligne de transmission. Le caractère XON sera envoyé pour signaler que le récepteur est de nouveau prêt à recevoir. Pour ce protocole, seuls les signaux RxD et TxD sont utilisés (câble de 2 fils + 1 masse).

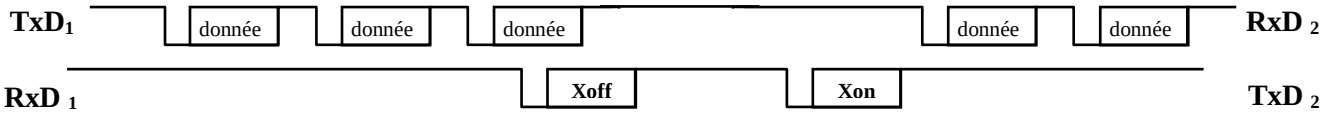


figure 8-16: chronogramme liaison XON - XOFF

Remarque: Ce protocole suppose une analyse par l'émetteur du flux des caractères en retour. De ce fait, les suspensions et reprises sont moins rapides que dans le protocole DTR/DSR.

7) Les normes électriques de transmission:

Différentes normes électriques, mécaniques et fonctionnelles sont équivalentes mais font l'objet de différentes appellations.

Standard	RS 232 C	RS 422	RS 485
Année définition	1962	1975	1984
Mode opératoire	Simple	Différentiel	Différentiel
Nombre d'émetteurs	1	1	32
Nombre de récepteurs	1	10	32
Longueur maxi de la ligne	15 m	1200 m	1200 m
Vitesse maxi de transmission	20 kbits/s	10 Mbits/s	10 Mbits/s
produit vitesse longueur	300 kbits . m/s	120 Mbits . m/s	120 Mbits . m/s

Tableau 8-05: paramètres des standards de la norme EIA

7.1) Norme RS 232C :

La norme RS 232 C est une norme internationale, reconnue essentiellement pour des transmissions séries asynchrones, mais elle dispose de fonctions pour d'éventuelles liaisons série synchrone.

7.1.1) caractéristiques électriques des signaux (Norme V 24 ET V28):

Les signaux numériques à transmettre sont adaptés pour la liaison conformément à la norme V28.

Niveau Logique	Niveau TTL (ou CMOS)	Niveau Norme V28
0	0 volt	12V (+3V < Tension < +25V)
1	5 volts	-12V (-3V < Tension < -25V)

Tableau 8-06: Niveaux de tension V 28

Exemple:

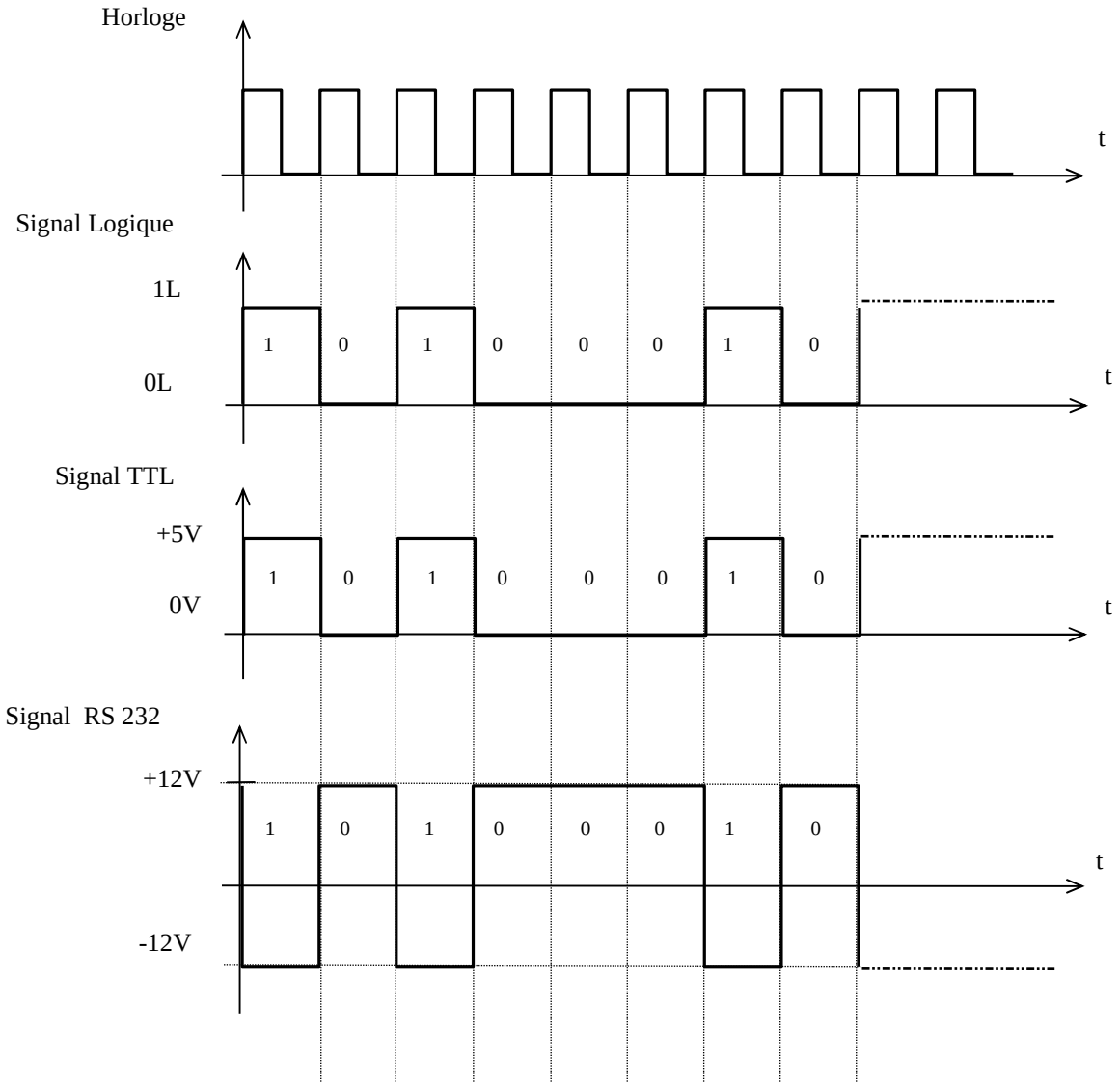


figure 8-17: signal RS 232

Même pour une transmission sans Modem, il y doit avoir une adaptation des niveaux électriques du signal TTL à ceux conformes à la norme RS232 au moyen de circuits spécialisés comme le MAX 232.

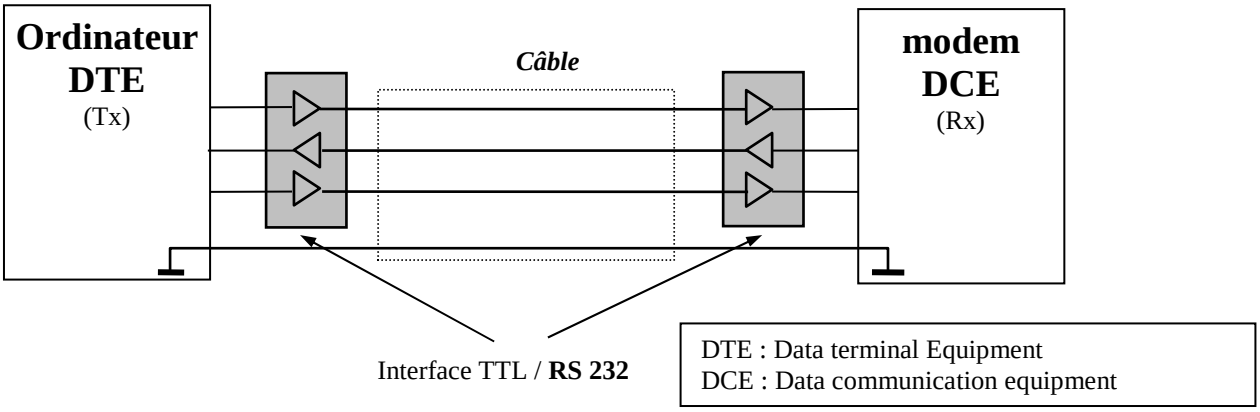
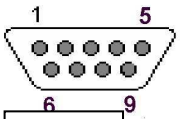


figure 8-18: interface RS 232

7.1.2) caractéristiques mécaniques :

Un connecteur 25 (ou 9) broches permet de réaliser une connexion conforme à la norme RS 232C. Chacune des broches du connecteur permet de transmettre un signal bien précis. Ces différents signaux de contrôle de transmission et de réception sont normalisés et définissent les caractéristiques fonctionnelles de la liaison RS 232C.

PRISE DB9 MALE

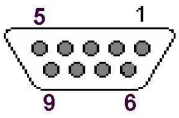


DTE Ordinateur

N°	Nom	Description
3	TD	Donnée transmise
2	RD	Donnée reçue
4	$\overline{DTR}$	sortie qui indique à l'autre équipement que l'on souhaite communiquer
6	$\overline{DSR}$	entrée qui annonce que l'autre équipement est pret
7	$\overline{RTS}$	sortie qui indique à l'équipement opposé de se tenir pret à recevoir
8	$\overline{CTS}$	entrée qui annonce que l'équipement opposée est pret à recevoir
5	SG	Masse

1: DCD

PRISE DB9 FEMELLE

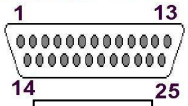


DCE modem

N°	Nom	Description
2	TD	Donnée transmise
3	RD	Donnée reçue
4	$\overline{DTR}$	sortie qui indique à l'autre équipement que l'on souhaite communiquer
6	$\overline{DSR}$	entrée qui annonce que l'autre équipement est pret
8	$\overline{RTS}$	sortie qui indique à l'équipement opposé de se tenir pret à recevoir
7	$\overline{CTS}$	entrée qui annonce que l'équipement opposée est pret à recevoir
5	SG	Masse

1: DCD

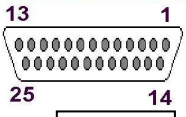
PRISE DB25 MALE



DTE

N°	Nom	Description
2	TD	Donnée transmise
3	RD	Donnée reçue
4	$\overline{RTS}$	*request to send :* demande pour émettre
5	$\overline{CTS}$	clear to send: paré à emettre
6	$\overline{DSR}$	DCE prêt
7	SG	Masse signaux
20	$\overline{DTR}$	DTE prêt

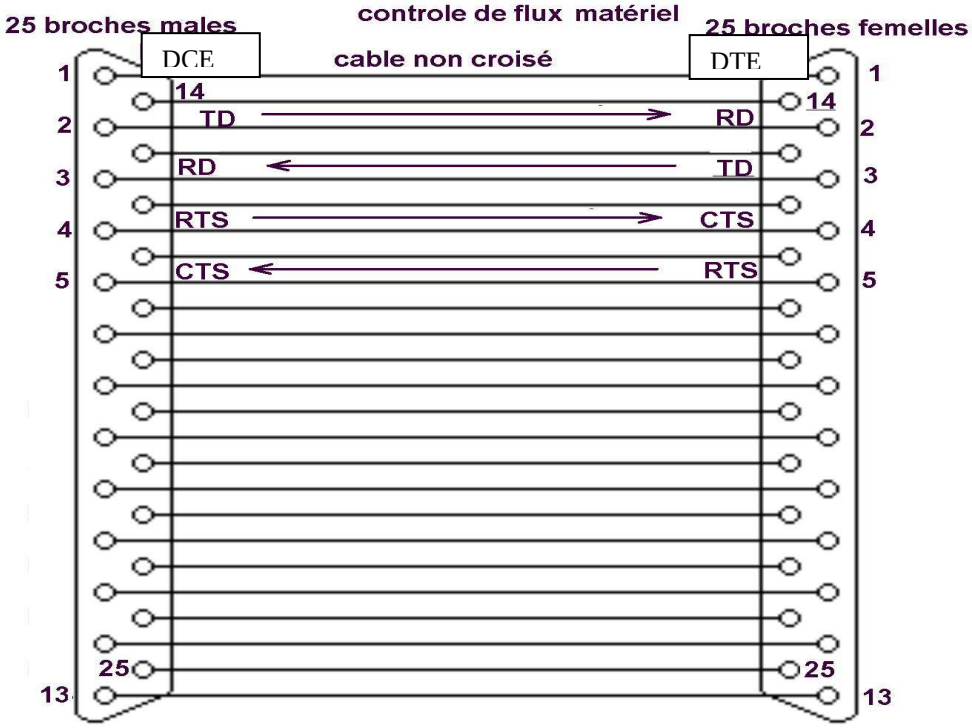
PRISE DB25 FEMELLE



DCE

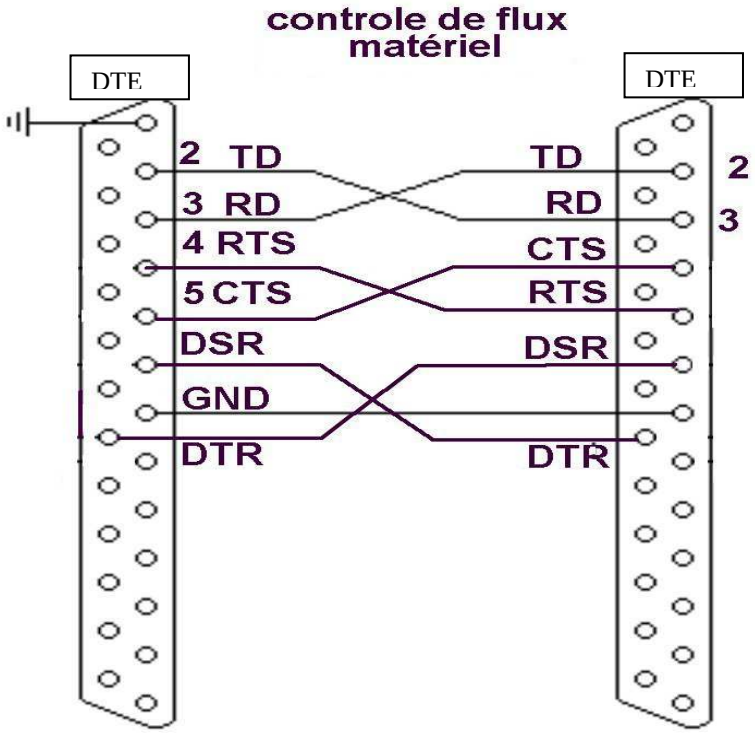
N°	Nom	Description
3	TD	Donnée transmise
2	RD	Donnée reçue
4	$\overline{CTS}$	*request to send :* demande pour émettre
5	$\overline{RTS}$	clear to send: paré à emettre
6	$\overline{DSR}$	DCE prêt
7	SG	Masse signaux
20	$\overline{DTR}$	DTE prêt

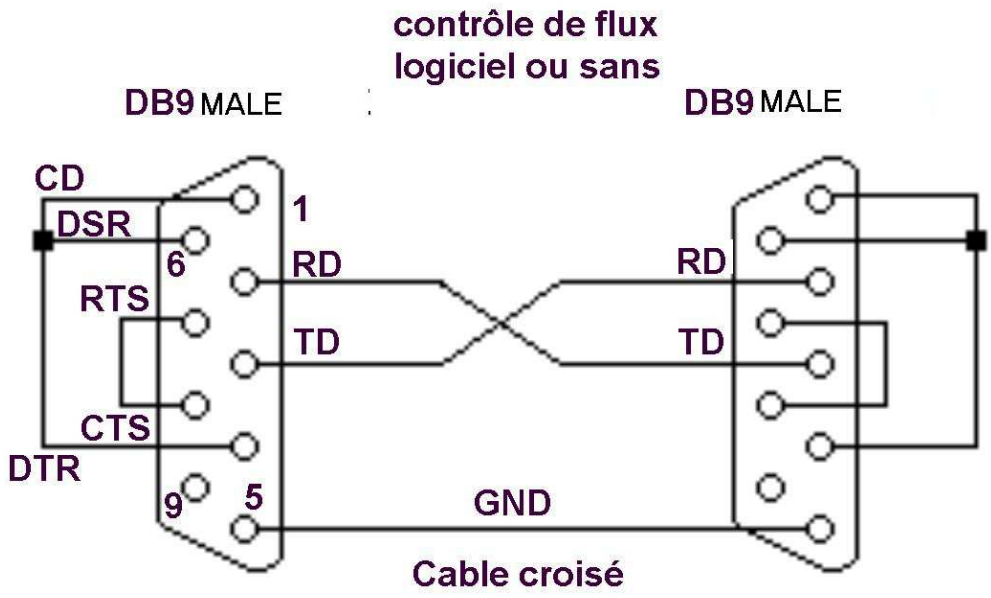
a) une liaison entre un ordinateur et un modem s'effectue de la façon suivante :



Le câble est non croisé

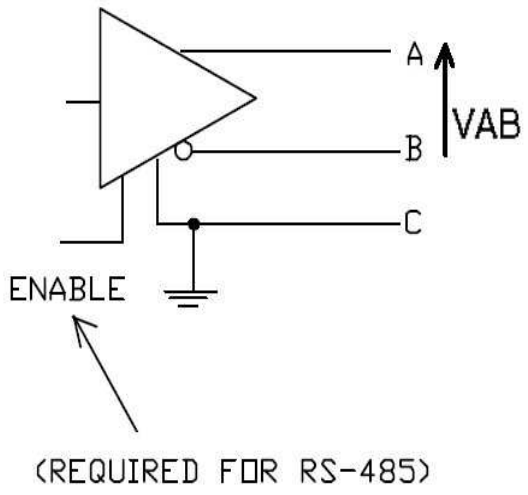
b) une liaison entre un ordinateur et un autre ordinateur s'effectue de la façon suivante :

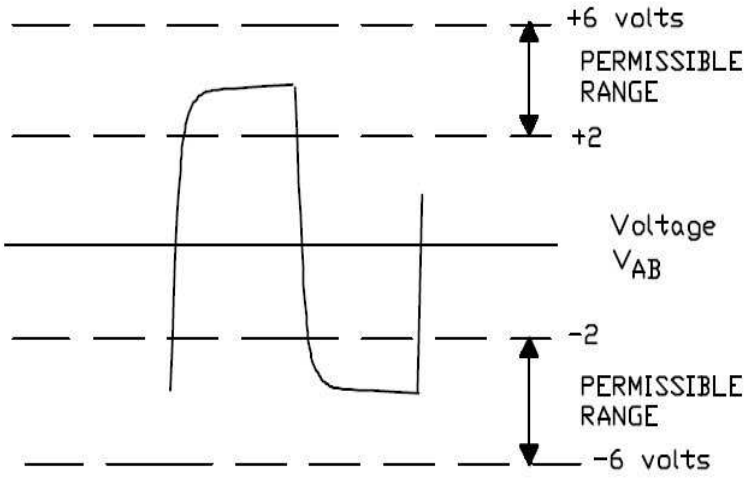



contrôle de flux
logiciel ou sans
DB9 MALE
DB9 MALE
CD
DSR
RTS
CTS
DTR
1
6
9
5
RD
TD
GND
RD
TD
Cable croisé

La norme RS485 ou V11

Liaison utilisée dans les systèmes de contrôle et en mode multipoint, Elle est le support de nombreux "bus de terrain". Généralement, ce système utilise le principe maître/esclave (protocole MODBUS, JBUS, PROFIBUS). Chaque esclave se voit attribuer une adresse et répond lorsqu'un message lui attribué.
Pour des raisons d'immunité au bruit, le signal qui circule sur la ligne est du type différentiel (voir la tension VAB). Comme on peut avoir plusieurs émetteurs sur la ligne et pour éviter les problèmes de collision, un signal de validation (ENABLE) vient connecter chaque émetteur à la ligne (mode haute impédance lorsqu'on est déconnecté).

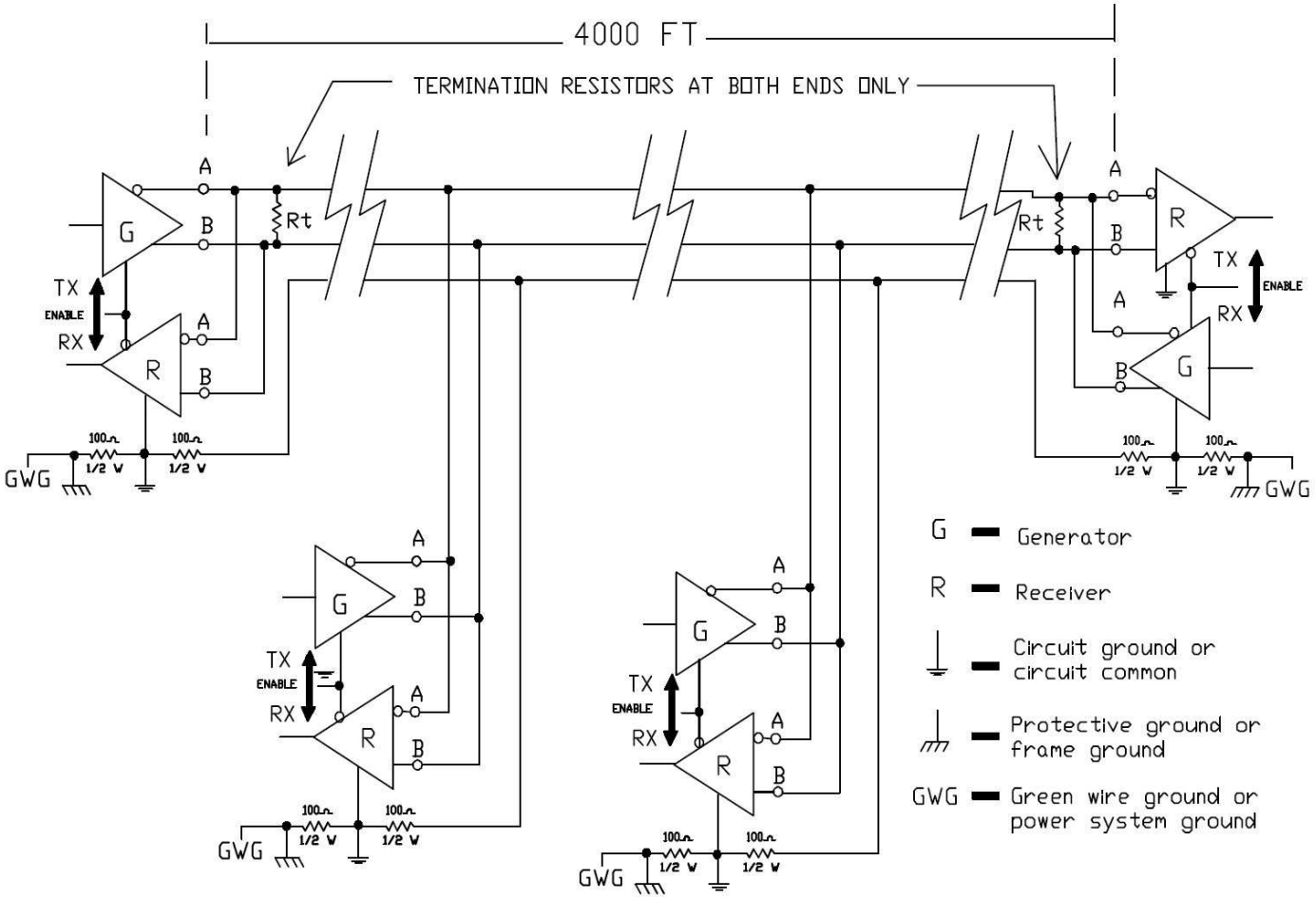




Deux versions existent : 1 ou 2 paires torsadées.

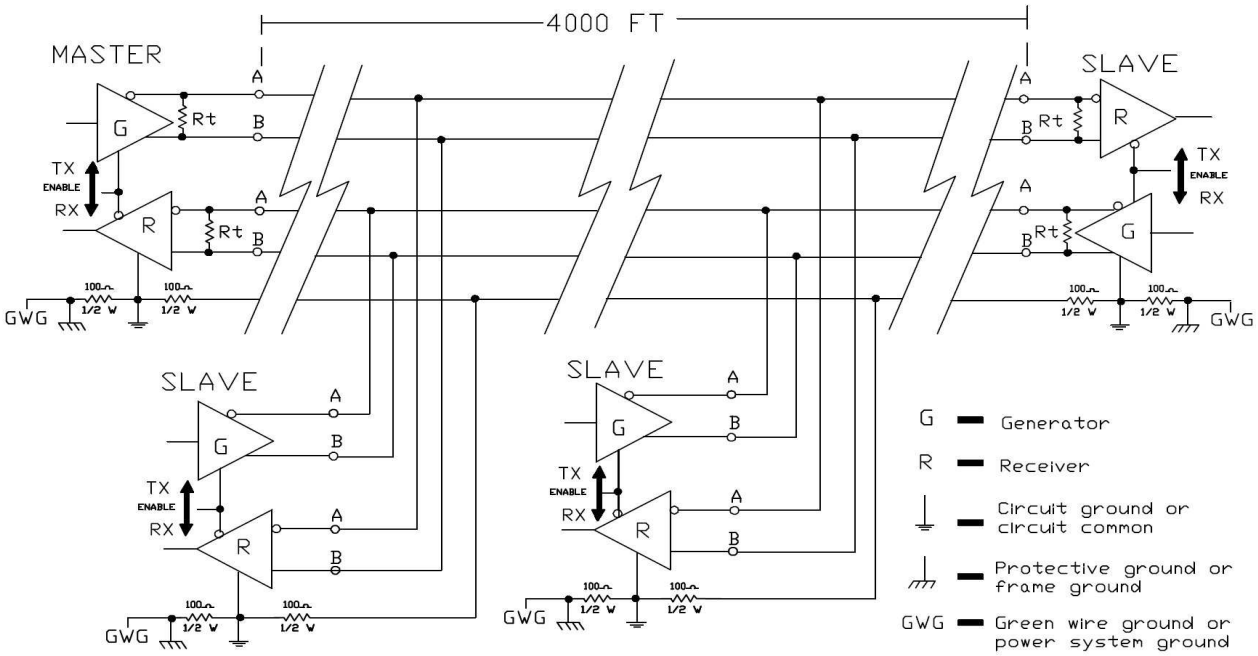
La version 2fils

Sur la première version, 1 paire torsadée, tous les appareils doivent disposer d'une sortie 3 états (actif, inactif, haute impédance) pour ne pas créer de conflits.



La version 4fils

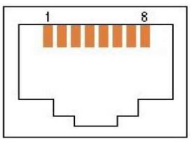
Avec la deuxième version 2 paires torsadées, le maître n'a pas besoin de sortie trois états puisqu'il émet sur une des lignes. L'autre ligne est réservée pour l'émission des esclaves.



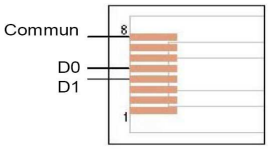
La connectique

Connecteur RJ45

Vue de face

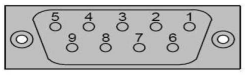


Vue de dessus

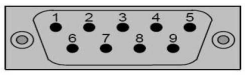


Connecteur Sub-D 9 broches

Femelle



Mâle



RJ45	Sub-DB9	Exigence	2 fils		4 fils		Description
			IDv	EIA/TIA 485	IDv	EIA/TIA 485	
1	8	requis			RxD0	A'	Receiver terminal 0, potentiel Va'
2	4	requis			RxD1	B'	Receiver terminal 1, potentiel Vb'
3	3	optionnel	PMC		PMC		Port Mode Control
4	5	**requis**	D1	B/B'	TxD1	B	Transceiver terminal 1, potentiel V1/Vb
5	9	**requis**	D0	A/A'	TxD0	A	Transceiver terminal 0, potentiel V0/Va
7	2	recommandé	VP				Alimentation positive 5..24VDC
8	1	**requis**	Common	C/C'	Common	C/C'	Commun d'alimentation et de signal

La norme RS422 ou V11

RS422 est utilisé pour les liaisons point à point comme la RS232 (mais avec une liaison du type différentielle).

