

LA LIAISON SERIE RS232

I. Partie 1 : Etude Théorique

1. Présentation

La norme RS-232 (*Recommended Standard 232*) a été définie en 1969 par l'EIA (*Electronic Industries Alliance*) pour le but de l'interfaçage d'équipements DTE (*Data Terminal Equipment*) et d'équipements DCE (*Data Circuit Equipment*) en utilisant l'échange série de données binaires.

L'interface RS 232 spécifie la **méthode de connexion** de deux équipements DTE et DCE. Un équipement DCE (Modem) reçoit les données provenant du DTE (PC ou imprimante) et les fait transmettre à un autre DCE via un support de communication comme le téléphone.

Cette connexion est dite **liaison série** et de **type asynchrone** :

- connexion **série** puisque elle permet la communication entre deux systèmes numériques en limitant le nombre de fils de transmission par rapport à des liaisons parallèles,
- Elle est de type « **asynchrone** », c'est à dire qu'elle ne transmet pas de signal horloge contrairement aux liaisons synchrones.

La liaison série RS 232 est utilisée dans tous les domaines de l'informatique (ex : port de communication com1 et com2 des PC, permettant la communication avec des périphériques tels que modem / scanner / ...), mais également dans le domaine des automatismes pour relier des capteurs aux automates, pour programmer un automate par un PC...etc.

2. Eléments d'un RS 232

Le standard RS 232 consiste en 3 majeures parties :

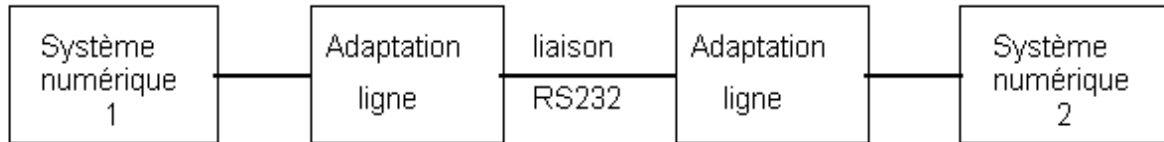
- Partie électrique
- Partie mécaniques de l'interface
- Description fonctionnelle des signaux

2.1. Partie électrique

RS232 définit les caractéristiques du signal électriques, où pour chaque signal le support est un fil référencé par rapport à la masse.

Avant adaptation	Après adaptation
Les niveaux logiques sont les suivants : niveau 0 = 0 V niveau 1 = 5 V	Les niveaux logiques sont les suivants : niveau 0 = +12 V ($3V < v < 25V$) niveau 1 = -12 V ($-25V < v < -3V$)

La norme permet d'adapter les signaux logiques (TTL ou CMOS) à l'aide d'un circuit adaptateur de ligne (ex : MAX232) qui transforme les niveaux logiques issus du système numérique en niveaux logiques compatibles avec les normes RS232 et vice versa.



2.2. Partie Physique

La norme EIA RS-232C définit l'utilisation du connecteur de type Sub-DB25. Afin de simplifier les applications de la norme RS232, elle peut aussi être utilisée avec un connecteur Sub-DB9, il s'agit de la norme EIA/TIA 574. Les possibilités de fonctionnement sont donc nécessairement réduites mais toujours largement suffisante pour des applications de communication standards.

Sur les PC, RS232 disponible et de type Sub-DB9 points. D'autres protocoles séries peuvent être trouvés avec des connectiques PS/2 (clavier/souris) ou Sub-DB15 points (joystick ou instruments de musique : protocole MIDI).

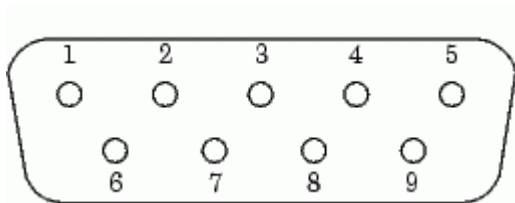


Figure 1 : Sub-DB 9 male (PC), Sub-DB 9 femelle (sur le câble).

2.3. Description fonctionnelle des signaux

Pin	Nom	Direction		Description
		(DTE)	(DCE)	
1	DCD	←		Détection de porteuse (DCD = Data Carrier Detect)
2	RD	←		Réception de données (RD = Receive Data)
3	TD		→	Transmission de données (TD = Transmit Data)
4	DTR		→	Équipement prêt (DTR = Data Terminal Ready)

5	GND	————	Masse du signal (SG = Signal Ground)
6	DSR	←————	Prêt à recevoir (DSR = Data Set Ready)
7	RTS	————→	Demande d'autorisation à émettre (RTS = Request To Send)
8	CTS	←————	Autorisation d'émettre (CTS = Clear To Send)
9	RI	←————	Détection de sonnerie (RI = Ring indicator)

Norme EIA/TIA 574, RS232 sur Sub-DB 9 points

Les broches du standard RS 232 sont divisées en 4 groupes (pour DB25 et DB09)

- lignes de données
- Lignes de contrôle de flux
- Lignes d'intérêt
- Autres lignes
- **GND** : (*Ground*) la masse. Référence nécessaire à toute mesure de tension.
- **Lignes de données**. Elles font passer les données numériques.
 - **RD** : (*Received Data*) Données reçues (par le PC).
 - **TD** : (*Transmitted Data*) Données émises (par le PC).
- **Lignes de contrôle de flux**. le contrôle de flux reste toujours (surtout dans une communication asynchrone) une logique de réception, c'est à dire que c'est la faculté de réception du partenaire qui indique si oui ou non un équipement va émettre.
 - **RTS** : (*Request to Send*) Indique que le DTE (PC) est prêt à émettre, même si la traduction littérale est « demande d'émission ».
 - **CTS** : (*Clear to Send*) Indique que le DCE (modem) est prêt à recevoir, même si la traduction littérale est « autorisé à émettre ».
- **Lignes d'intérêt**. Ces lignes indiquent au correspondant que l'équipement qui les met à '1' s'intéresse à la liaison RS-232. Le fait de faire passer temporairement l'une de ces lignes à '0' constitue un « break ».
 - **DSR** : (*Data Set Ready*) Indique que le DCE (modem) écoute sa liaison RS-232.
 - **DTR** : (*Data Terminal Ready*) Indique que le DTE (PC) écoute la liaison RS-232.
- **Autres lignes d'état du modem** :
 - **DCD** : (*Data Carrier Detect*, aussi nommée RLSD : *Receive Line Signal Detect*)

Indique au DTE (PC) qu'une porteuse de données valide est bien détectée sur le RTC (Réseau Téléphonique Commuté).

- **RI** : (*Ring Indicator*) Indique au DTE (PC) que la ligne téléphonique à laquelle le modem est relié est en train de sonner.

❖ **Remarque :**

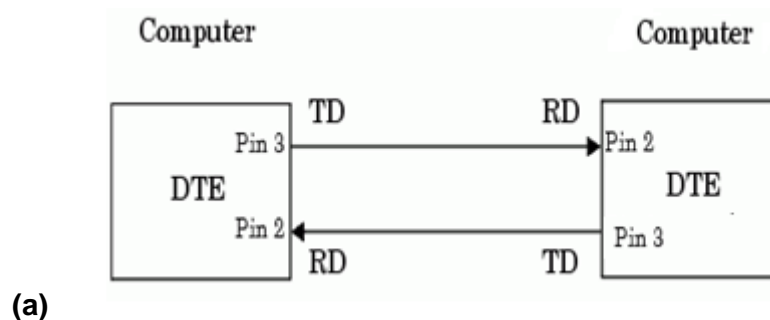
Il existe une relation entre la vitesse de transmission de données et la longueur de la ligne connectant les deux équipements.

Vitesse (Bauds)	Longueur (Mètres)
110	850
300	800
600	700
1200	500
2400	200
4800	100
9600	70
19200	50
115k	20

NB : 1 baud = 1 bit/s

3. La connexion NULL MODEM

Il s'agit de relier directement deux équipements DTE, sans passer par les DCE. La principale différence vient alors du **croisement des fils d'émission/réception (RD/TD)** figure 5.a, ce qui ne doit pas être fait dans le cas de connexions DTE-DCE.



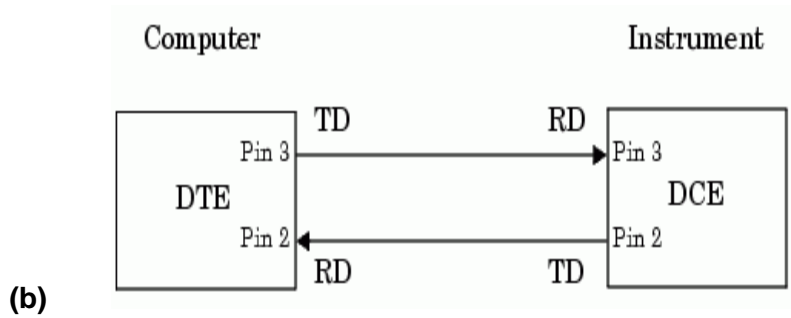


Fig. 5 : Connexion de deux équipements par câble RS 232 :

a) DTE-DTE, b) DTE-DCE

4. Le traitement numérique des données

La mémoire vive des ordinateurs est divisée octet (byte) = 8 bits

0000 0000	→	0	
0000 0001	→	1	
0000 0010	→	2	
0000 0011	→	3	1 octet = 256 nombres ou codes
0000 0100	→	4	(2 ⁸ = 256)
...			
1111 1111	→	255	

Un nombre binaire b6 b5 b4 b3 b2 b1 b0 a la valeur décimale $b6 \cdot 2^6 + b5 \cdot 2^5 + b4 \cdot 2^4 + b3 \cdot 2^3 + b2 \cdot 2^2 + b1 \cdot 2^1 + b0 \cdot 2^0$

Les nombres hexadécimaux = chiffres de 0 à 9 et des lettres A à F :

0, 1, 2, ..., 9, A, B, C, ..., F, 10, 11, 12, ..., 1A, 1B, 1C, ..., 1F, 20, ...

4.1. Le codage des informations

Le morse a été le premier codage. Ce code est composé de points et de tirets comme le codage binaire en quelque sorte.

Dans les années 60, le code ASCII est adopté comme standard. Il permet le codage de caractères sur 8 bits, soit 256 caractères possibles.

- **Qu'est-ce que le code ASCII ?**

La mémoire de l'ordinateur conserve toutes les données sous forme numérique. Il n'existe pas de méthode pour stocker directement les caractères. Chaque caractère possède donc son équivalent en code numérique : c'est le code ASCII (*American Standard Code for Information Interchange*).

Le code ASCII de base représentait les caractères sur 7 bits (c'est-à-dire 128 caractères possibles, de 0 à 127).

4.2. Table des caractères ASCII

Les codes 0 à 31 ne sont pas des caractères. On les appelle *caractères de contrôle* car ils permettent de faire des actions telles que:

- Retour à la ligne : 13 (CR)
- Bip sonore : 07 (BEL)
- Les codes 48 à 57 représentent les chiffres.
- Les codes 65 à 90 représentent les majuscules.
- Les codes 97 à 122 représentent les minuscules.

Code	0	1	2	3	4	5	6	7	8	9
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT
10	LF	VT	NP	CR	SO	SI	DLE	DC1	DC2	DC3
20	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS
30	RS	US	SP	!	"	#	\$	%	&	'
40	()	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[\]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{		}	~	DEL		

Le code ASCII a été mis au point pour la langue anglaise, il ne contient donc pas de caractères accentués, ni de caractères spécifiques à une langue. Pour coder ce type de caractère il faut recourir à un autre code. Le code ASCII a donc été étendu à 8 bits (un octet) pour pouvoir coder plus de caractères (on parle d'ailleurs de code ASCII étendu...)

II. Partie2 : Manipulation

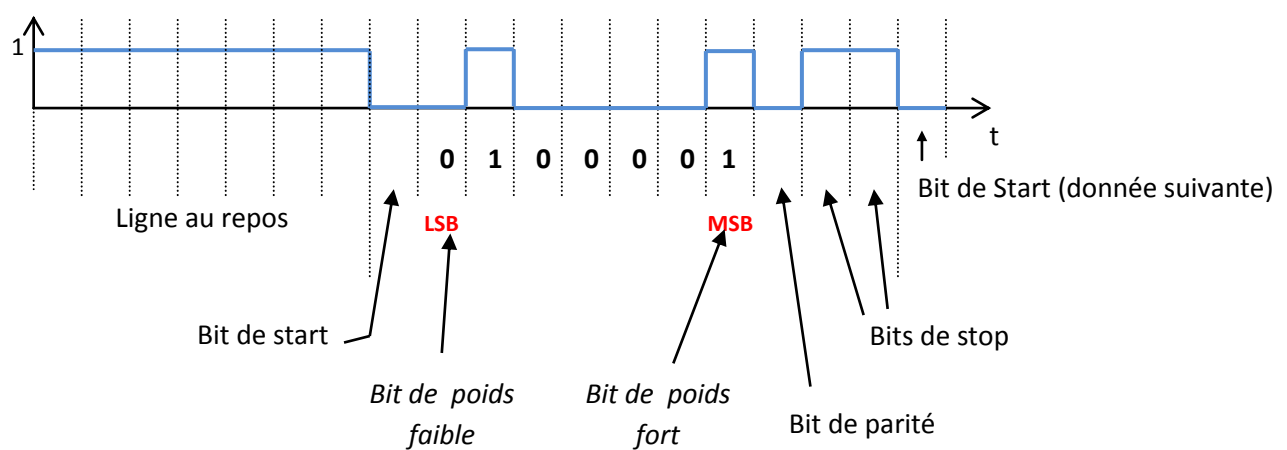
1. Relier deux PC par un câble Null modem
 2. Pour faire dialoguer deux PC, utiliser le logiciel de intégré à Windows : **Hyper Terminal**
 3. Lancement Hyper Terminal :
 Dans le menu Démarrer → Accessoires → Communication → HyperTerminal
 (ou directement en faisant une recherche : *hypertrm.exe*)
 4. Réglage Hyper Terminal :
 - a. Dans la fenêtre qui s'ouvre description de la connexion, donner un nom à la connexion que vous allez créer Par exemple « **master01** », puis cliquer sur OK.
 - b. Dans la fenêtre connexion, choisissez le pays : **Algérie (213)** et ensuite le port sur lequel est branché la session master01 : **COM1**
 - c. Dans la fenêtre *Propriétés* de **COM1**, régler la vitesse à 9600 bits/s et le *contrôle de flux*
- **Appeler un ordinateur distant**
- **Pour envoyer un fichier à un ordinateur distant :**
1. Dans le menu Transférer, cliquez sur Envoyer le fichier.
 2. Dans la zone Nom de fichier, tapez le chemin d'accès et le nom du fichier. Pour utiliser un protocole différent pour le transfert du fichier, sélectionnez le protocole voulu dans la zone Protocole.
 3. Pour envoyer le fichier, cliquez sur Envoyer.

Exemple : Soit à transmettre le caractère B avec le codage ASCII

Longueur du mot : 7 bits (le caractère B en codage ASCII est $100\ 0010_{(2)}$), pas de parité,
2 bits de stop.

La trame sera la suivante :

Niveau ordinateur



Niveau électrique de la ligne norme RS232

