

Les liaisons de données en informatique

1- LA TRANSMISSION DE DONNEES - CODAGE DES INFORMATIONS

Les réseaux de communication permettent la transmission d'informations codées sous la forme d'une suite d'éléments binaires entre des équipements terminaux.

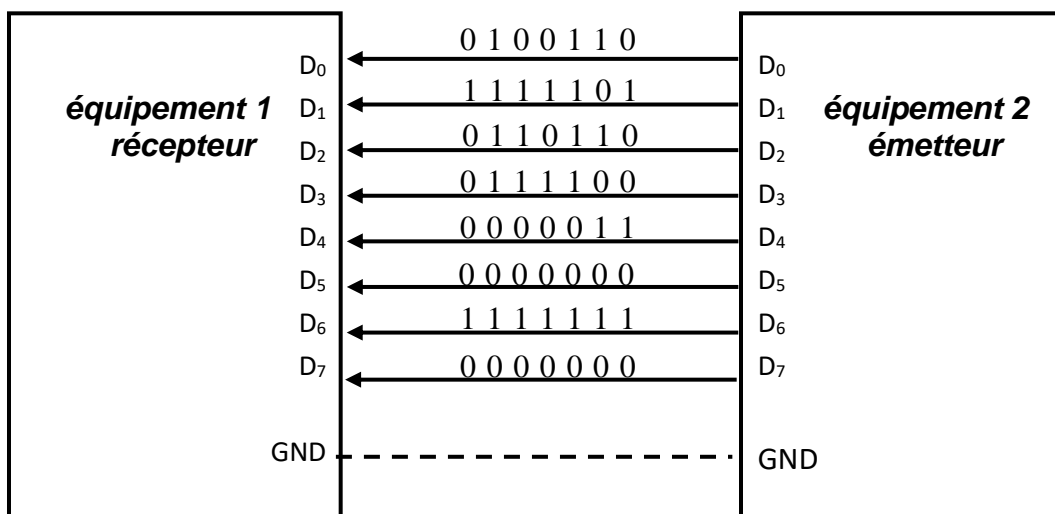
Le code normalisé le plus utilisé est le code ASCII (American Standard Code for Information Interchange). À chaque caractère (lettres, chiffres, symboles, codes de contrôle) correspond un nombre de 7 bits, ce qui permet de coder 128 (2^7) caractères différents représentés et codés en décimal dans le tableau suivant.

000	{nul}	016	► {dle}	032	sp	048	ò	064	@	080	P	096	`	112	p
001	⊕ {soh}	017	◄ {dc1}	033	!	049	1	065	A	081	Q	097	a	113	q
002	⊙ {stx}	018	↑ {dc2}	034	"	050	2	066	B	082	R	098	b	114	r
003	▼ {etx}	019	!! {dc3}	035	#	051	3	067	C	083	S	099	c	115	s
004	⬆ {eot}	020	¶ {dc4}	036	\$	052	4	068	D	084	T	100	d	116	t
005	⚡ {enq}	021	§ {nak}	037	%	053	5	069	E	085	U	101	e	117	u
006	⬆ {ack}	022	— {syn}	038	&	054	6	070	F	086	V	102	f	118	v
007	• {bel}	023	‡ {etb}	039	'	055	7	071	G	087	W	103	g	119	w
008	▣ {bs}	024	↑ {can}	040	(056	8	072	H	088	X	104	h	120	x
009	{tab}	025	↓ {em}	041)	057	9	073	I	089	Y	105	i	121	y
010	{lf}	026	{eof}	042	*	058	:	074	J	090	Z	106	j	122	z
011	♂ {vt}	027	← {esc}	043	+	059	;	075	K	091	[107	k	123	{
012	♀ {np}	028	⌊ {fs}	044	,	060	<	076	L	092	\	108	l	124	
013	{cr}	029	↔ {gs}	045	-	061	=	077	M	093]	109	m	125	}
014	♂ {so}	030	▲ {rs}	046	.	062	>	078	N	094	^	110	n	126	~
015	✱ {si}	031	▼ {us}	047	/	063	?	079	O	095	_	111	o	127	o

Pour remarque : L'encodage UTF-8 utilise les mêmes codes qu'ASCII pour les 127 premiers caractères, et se sert d'octets additionnels pour représenter des caractères spéciaux comme 'É', 'ç' ou '€'.

2 - TRANSMISSION PARALLELE

Les bits sont envoyés simultanément sur des fils distincts pour arriver ensemble à destination. La ligne de transmission comporte n fils d'information plus un fil commun (GND) et plusieurs fils de service suivant le protocole. Ce type de câblage est devenu obsolète pour les imprimantes ou les disques durs.

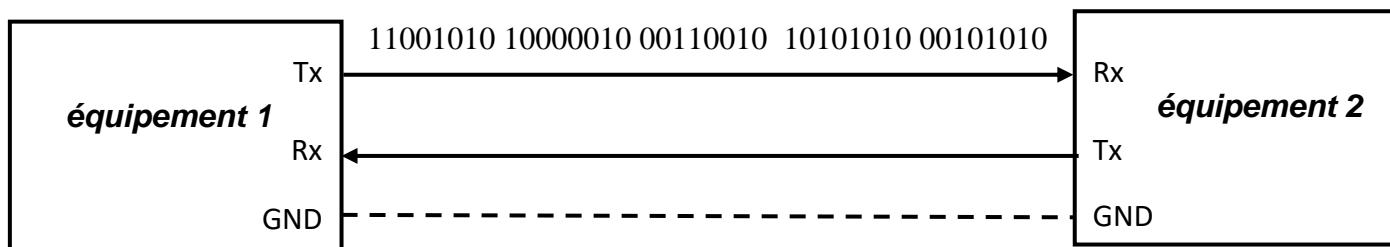


→ Décoder le mot envoyé de 2 vers 1

3 - TRANSMISSION SÉRIE

3-1 – Présentation de la transmission série asynchrone

Les bits sont envoyés les uns derrière les autres sur le même fil. La ligne comporte au moins un fil d'information plus un fil commun. Il est convenu de commencer par envoyer le bit de poids faible (mode little endian) de chaque octet.



→ Décoder le mot envoyé de 1 vers 1

Ce mode de transmission est dit **asynchrone** car l'intervalle de temps entre 2 bits transmis est paramétré sur les 2 équipements. Il existe un mode d'échange **synchrone** qui utilise un signal supplémentaire d'acquittement des données. La vitesse de transmission s'exprime en bauds, bits/s ou bps.

3.2- Protocole de transmission de la norme RS 232

Il est nécessaire d'établir un protocole de transmission afin que les éléments communicants puissent se comprendre. Bien entendu le protocole devra être commun aux deux éléments communicants.

Paramètres rentrant en jeu :

- **Longueur des mots** : 6, 7 ou 8 bits
- **La vitesse de transmission** : les différentes vitesses de transmission sont réglables à partir de 110 bits par seconde (bps) de la façon suivante : 110 bps, 150 bps, 300 bps, 600 bps, 1200 bps, 2400 bps, 4800 bps, 9600 bps ...18,2 kbps ...56 kbps ...
- **Parité** : le mot transmis peut-être complété ou non d'un bit de parité qui sert à détecter les erreurs éventuelles de transmission. Il existe deux types de parité.
 - **parité paire** : le bit ajouté à la donnée est positionné de telle façon que le nombre des états 1 soit paire sur l'ensemble données + bit de parité.
ex : soit la donnée 11001011 contenant 5 état 1, le bit de parité paire est positionné à 1, ramenant ainsi le nombre de 1 à 6.
 - **parité impaire** : le bit ajouté à la donnée est positionné de telle façon que le nombre des états 1 soit impaire sur l'ensemble données + bit de parité
ex : soit la donnée 11001011 contenant 5 état 1, le bit de parité paire est positionné à 0, laissant ainsi un nombre de 1 impaire.
- **Bit de start** : la ligne au repos est à l'état logique 1. Pour indiquer qu'un mot va être transmis la ligne passe à l'état bas avant de commencer le transfert.
- **Bit de stop** : après la transmission, la ligne est positionnée au repos pendant 1, 2 ou 1,5 périodes selon le nombre de bits de stop.

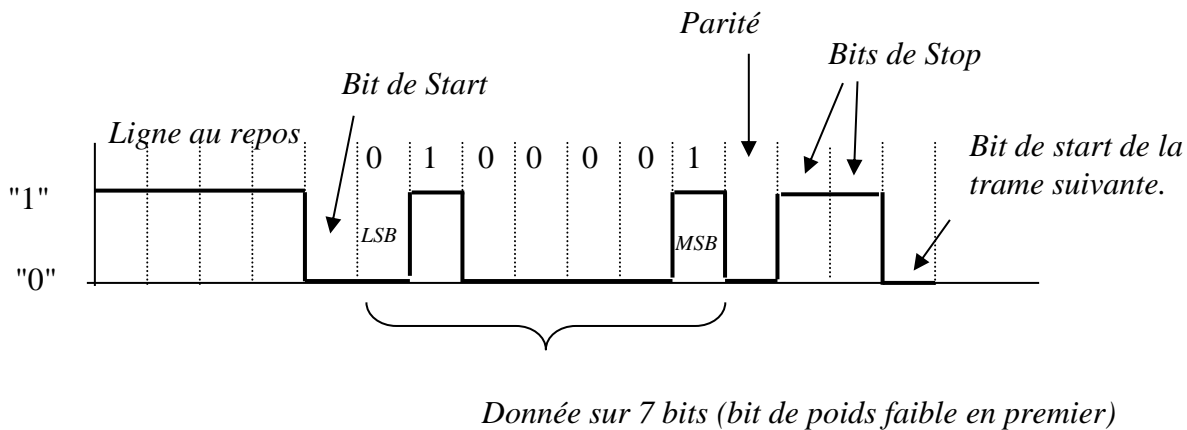
Format (chronogramme, oscillogramme) des trames

Le bit de Start apparaît en premier dans la trame puis les données (poids faible en premier), la parité éventuelle et le (les) bit(s) de stop.

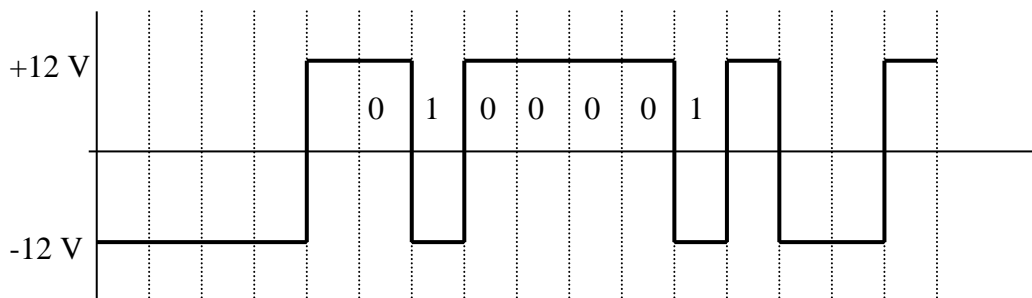
Exemple :

Soit à transmettre en parité paire, 7 bits de données, 2 bits de stop, le caractère "B" dont le codage ASCII est $(42)_{16}$ ou $(1000010)_2$; La trame sera la suivante :

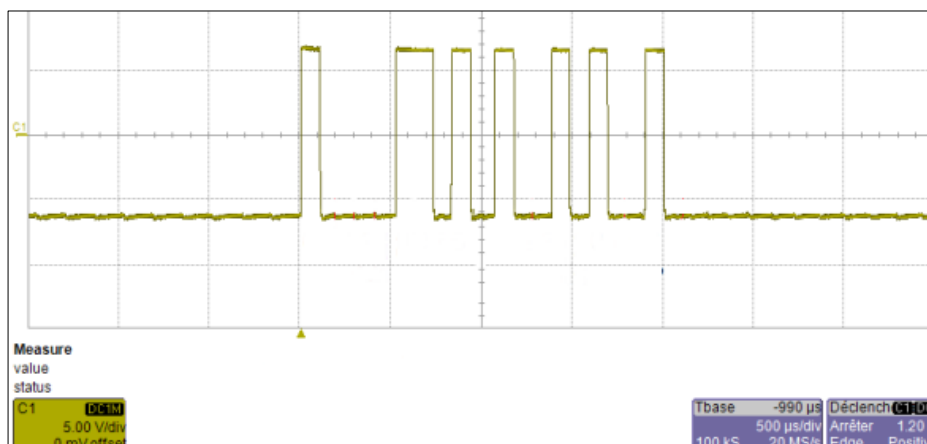
a) D'un point de vue logique (TTL)



b) D'un point de vue RS 232



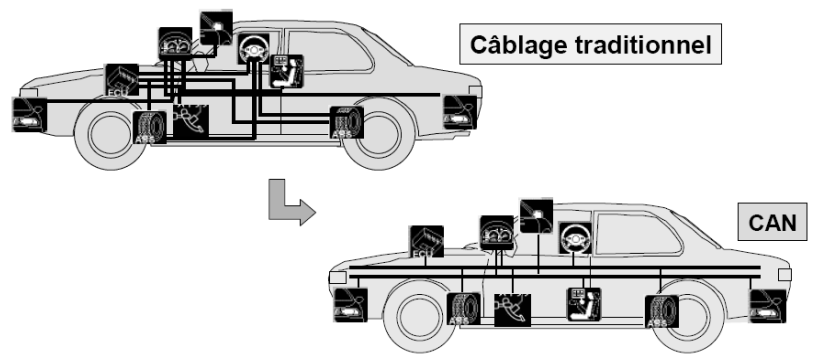
c) Déterminer le débit (en bit/s) ainsi que le message envoyé par la ligne RS232 ci-dessous :



Le protocole est : 1 start, 7 bits de données, parité impaire, 1 stop

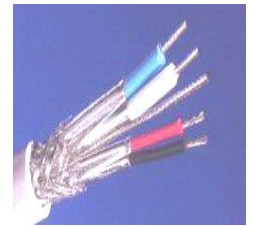
4- Le BUS CAN (Controller Area Network)

Le CAN a été lancé dans les années 1990 pour répondre aux besoins de l'industrie automobile devant la montée de l'électronique embarquée. En 2005 une voiture moyenne comportait une centaine de microcontrôleurs. Pour éviter les 2 km de câblage nécessaire il fallait définir un bus série simplifiant l'intégration des fils dans le châssis.

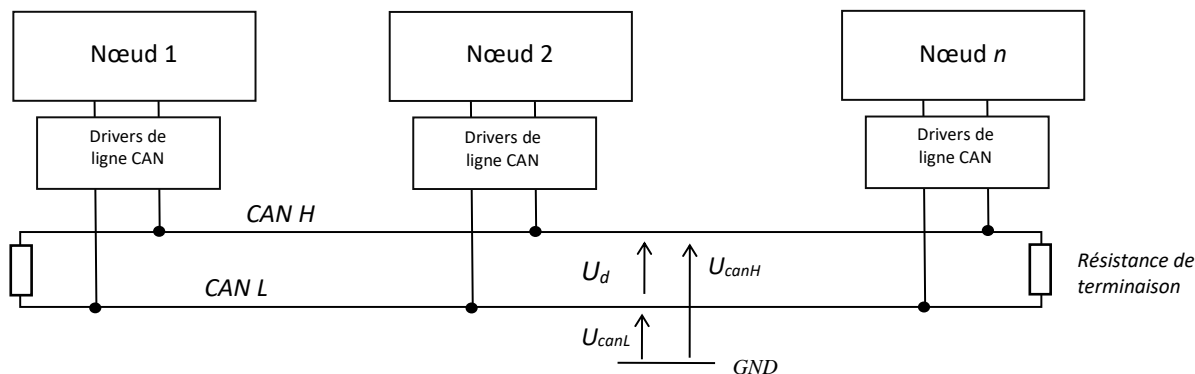


CONSTITUTION

Le bus est constitué de 2 paires de fils. Une première paire transporte l'alimentation électrique (comme pour l'USB) afin d'alimenter directement les appareils de faible consommation. La deuxième paire supporte les signaux de données appelés **CAN L** (CAN LOW) et **CAN H** (CAN HIGH). C'est la différence de tension entre CAN L et CAN H qui matérialise le niveau logique transmis. On appellera "**état dominant**" l'état logique 0, et "**état récessif**" l'état logique 1.



Chaque élément raccordé sur un bus constitue un nœud qui est identifié par une adresse logique unique appelée identificateur. Le protocole autorise chaque élément à démarrer la transmission d'une trame quand le bus est au repos.



Deux modes de fonctionnement possibles

	CAN low speed	CAN high speed
Débit max	125 kb/s	1 Mb/s
Niveau dominant « 0 »	CAN H = 4V CAN L = 1V	$2 \leq V_{CAN\ H} - V_{CAN\ L} \leq 3\ V$
Niveau récessif « 1 »	CAN H = 1,75V CAN L = 3,25V	$V_{CAN\ H} - V_{CAN\ L} \neq 0V$

La longueur maximale du bus est déterminée par la vitesse utilisée :

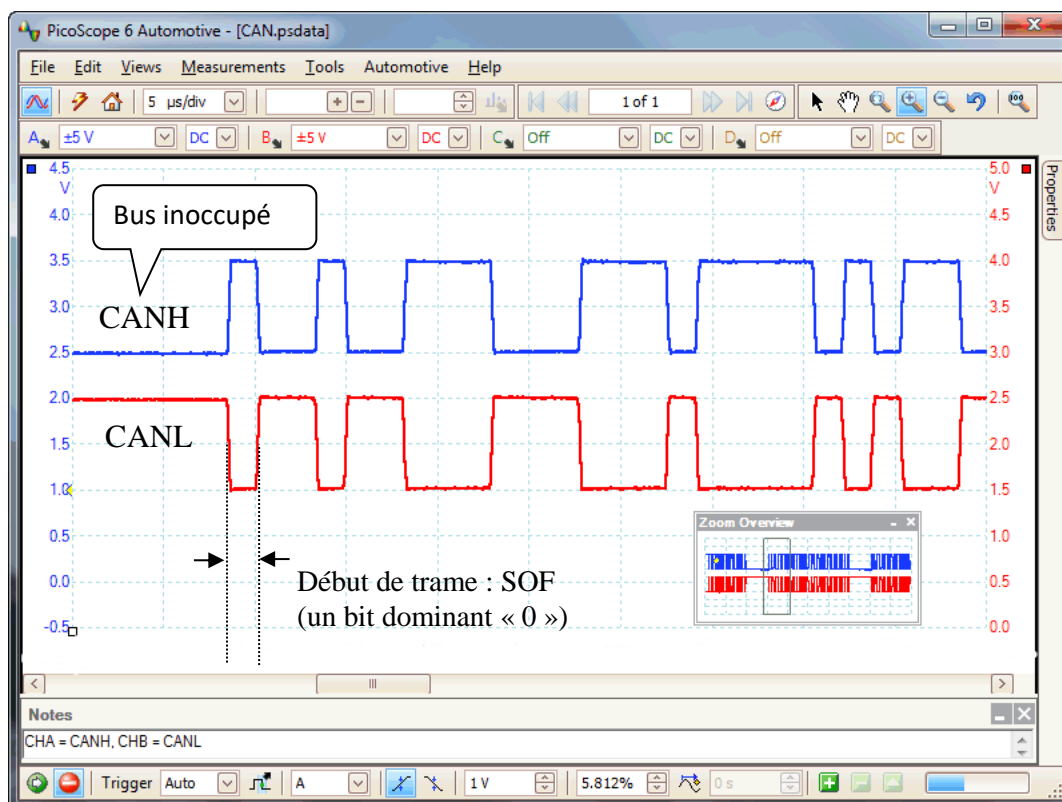
1000 kbit/s → 30 m maximum
800 kbit/s → 50 m maximum
500 kbit/s → 100 m maximum
125 kbit/s → 500 m maximum

LA TRAME d'un BUS CAN

Une trame est composée d'au plus 128 bits utiles repartis sur les champs suivants :

Start of Frame	Champ d'arbitrage	Champ de contrôle	Champ de données	Champ CRC	Champ d'acquittement	End of Frame
1 bit dominant	29 bits + 3bits	6 bits	0 à 8 octets	16 bits	2 bits	7 bits récessifs

Remarque : Lors de la construction d'une trame, si 5 bits sont consécutivement au même état (0 ou 1), un bit supplémentaire (Stuff bit) d'état complémentaire est inséré dans la trame. Exemple: 00000100111110111110



→ Déterminer le mode de fonctionnement.

→ Mesurer la durée de transmission d'un bit, en déduire la vitesse de transmission du bus.

5- Le BUS USB (Universal Serial Bus)

Le bus USB est inspiré de l'architecture vue sur le bus CAN

Nom	Débit par seconde (Gigabits)	Débit par seconde (Megaoctet)	Année de mise sur le marché
USB 1.0	0,012	1,5	1996
USB 2.0	0,48	60	2000
USB 3.0	5,00	625	2009

