Communication par liaison série :
LIAISON SERIE ASYNCHRONE (RS232)

PRESENTATION DE LA LIAISON SERIE ASYNCHRONE

La liaison série asynchrone est un standard de communication permettant de communiquer entre deux équipements.

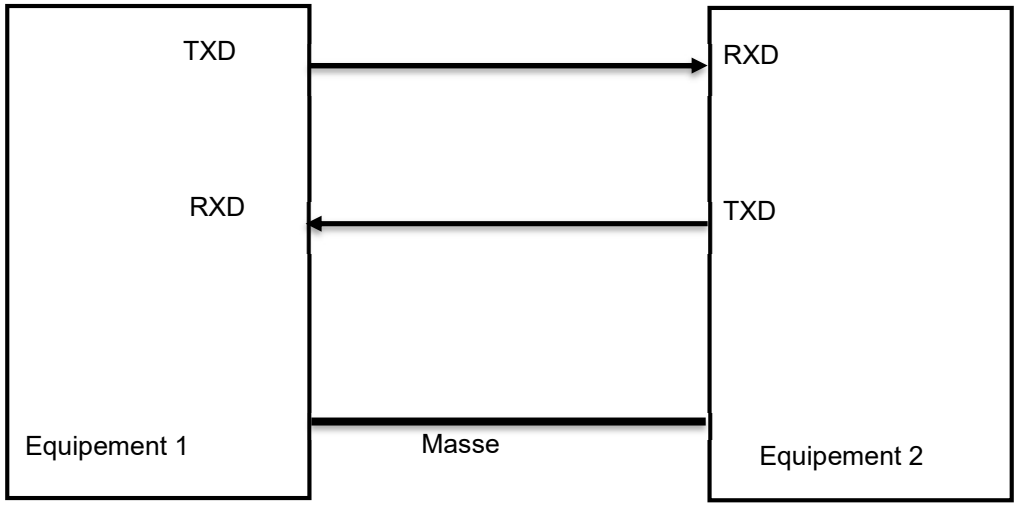
La communication se fait avec 3 fils (TXD, RXD et masse).

TXD : Transmission de données (sortie)
RXD : Réception de données (entrée)

L'équipement 1 peut envoyer des données vers l'équipement 2 via une sortie notée TXD, la réception des données se fait sur la broche RXD de l'équipement 2.

L'équipement 2 peut envoyer des données vers l'équipement 1 via une sortie notée TXD, la réception des données se fait sur la broche RXD de l'équipement 1.

Schéma :



Les données de type octet (8 bits) sont envoyées en série selon le protocole suivant :

- 1 bit de start (toujours à 0),
- 8 bits de données de b0 à b7,
- 1 bit de parité (facultatif),
- bit(s) de stop (toujours à 1 logique).

Communication par liaison série :
LIAISON SERIE ASYNCHRONE (RS232)

Eléments complémentaires :

1 Le débit

Si on veut transmettre un octet sur deux fils, il faut faire un découpage temporel de la donnée.
On définit pour cela la notion de débit en **bit/s** (**BAUD**).
Exemple :
9600 BAUD soit 9600 bits pour une durée de 1s (la durée d'un bit est de :1/9600 soit environ 0,1ms)

2 Le bit de départ : START

Le bit de START permet de « prévenir » le récepteur que des données arrivent.
Le bit de start est toujours à un niveau logique 0.

3 Les bits de DONNEES :

Les données (octets) seront émises du poids faible (B0) au poids fort (B7).

4 Bit (s) de : STOP (1 ou 2 selon la configuration du matériel).

Le bit de stop doit être émis à la fin de chaque octet, afin de permettre la transmission d'un nouvel octet.
Les bits de stop sont toujours à un niveau logique 1.

Exemple d'une transmission de l'équipement 1 vers l'équipement 2:

TXD de l'équipement 1

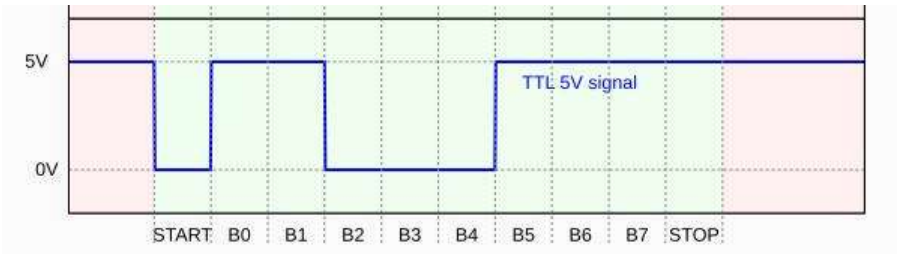


Table ASCII :

La table ASCII établie une correspondance entre un caractère et les bits de données B0 à B7.

Exemple :

Le caractère 'A' correspond à :

41 en base 16 soit 0100 0000 en base 2.

On a donc pour le caractère A :

B6 =1 et tous les autres bits à 0.

Dec	Hex	Name	Char	Ctrl-char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	Null	NUL	CTRL-@	32	20	Space	64	40	@	96	60	`
1	1	Start of heading	SOH	CTRL-A	33	21	!	65	41	A	97	61	a
2	2	Start of text	STX	CTRL-B	34	22	"	66	42	B	98	62	b
3	3	End of text	ETX	CTRL-C	35	23	#	67	43	C	99	63	c
4	4	End of xmit	EOT	CTRL-D	36	24	$	68	44	D	100	64	d
5	5	Enquiry	ENQ	CTRL-E	37	25	%	69	45	E	101	65	e
6	6	Acknowledge	ACK	CTRL-F	38	26	&	70	46	F	102	66	f
7	7	Bell	BEL	CTRL-G	39	27	'	71	47	G	103	67	g
8	8	Backspace	BS	CTRL-H	40	28	(	72	48	H	104	68	h
9	9	Horizontal tab	HT	CTRL-I	41	29	)	73	49	I	105	69	i
10	0A	Line feed	LF	CTRL-J	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	VT	CTRL-K	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	FF	CTRL-L	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage feed	CR	CTRL-M	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	SO	CTRL-N	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	SI	CTRL-O	47	2F	/	79	4F	O	111	6F	o
16	10	Data line escape	DLE	CTRL-P	48	30	0	80	50	P	112	70	p
17	11	Device control 1	DC1	CTRL-Q	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	DC2	CTRL-R	50	32	2	82	52	R	114	72	r
19	13	Device control 3	DC3	CTRL-S	51	33	3	83	53	S	115	73	s
20	14	Device control 4	DC4	CTRL-T	52	34	4	84	54	T	116	74	t
21	15	Neg acknowledge	NAK	CTRL-U	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	SYN	CTRL-V	54	36	6	86	56	V	118	76	v
23	17	End of xmit block	ETB	CTRL-W	55	37	7	87	57	W	119	77	w
24	18	Cancel	CAN	CTRL-X	56	38	8	88	58	X	120	78	x
25	19	End of medium	EM	CTRL-Y	57	39	9	89	59	Y	121	79	y
26	1A	Substitute	SUB	CTRL-Z	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	ESC	CTRL-[	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	FS	CTRL-\	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	GS	CTRL-]	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	RS	CTRL-^	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	US	CTRL-_	63	3F	?	95	5F	_	127	7F	DEL