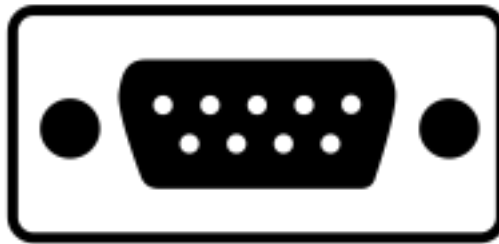


# PROTOCOLE RS232



---

<b>Introduction</b> .....	<b>3</b>
<b>Qu'est-ce que le RS232</b> .....	<b>3</b>
<b>Le protocole RS232</b> .....	<b>4</b>
Vitesse de transmission (Baud Rate).....	4
Niveaux de tensions seuils du récepteur .....	4
Bit de départ .....	4
Bits de données .....	5
Le Bit de parité.....	5
Le Bit d'arrêt.....	5
<b>Comment utiliser le RS232 avec nos interfaces</b> .....	<b>6</b>
Exemple: Démarrer la scène 1 .....	7
Table ASCII .....	8
Connexion RS232 .....	8

## INTRODUCTION

Le RS-232 est un protocole qui normalise un canal de communication de type série. Disponible sur presque tous les PC de 1981 jusqu'au milieu des années 2000, il est communément appelé "port série". Sur les systèmes d'exploitation MS-DOS et Windows, les ports RS-232 sont appelés COM1, COM2, etc. C'est ce qui leur a valu le surnom de "ports COM".

## QU'EST-CE QUE LE RS232

Le RS232 définit un protocole qui détaille comment un flux de données binaires est transmis séquentiellement sur un fil, c'est-à-dire un flux de bits ou d'octets. L'ordre et la signification de chaque bit sont définis par le protocole.

Le RS232 est une norme de protocole de transfert d'informations en série qui définit à la fois le protocole (méthode de transmission des données) et le matériel physique pour le faire.

Il s'agit d'une méthode de transfert de données en série sur un seul fil et unidirectionnelle, les données ne sont transmises que dans un seul sens pour chaque fil, de sorte que pour une communication bidirectionnelle (deux sens), il faut deux fils.

Elle utilise une méthode de transmission en série où les octets de données sont émis avec un bit à la fois sur un seul fil.

RS232 est un protocole de communication asynchrone car il n'y a pas d'horloge transmise du tout entre l'émetteur et le récepteur.

### **Données techniques :**

Il peut transférer un seul octet de données sur un câble série comportant entre 3 et 22 fils de signaux et fonctionnant à des vitesses allant de 100 à 20k bauds. Les vitesses de transmission les plus courantes sont 2,4k, 9,6k, 19,2k. La longueur du câble utilisé peut atteindre 15 mètres.

Pour transférer un bloc de données, des octets individuels sont transmis les uns après les autres.

## LE PROTOCOLE RS232

Les données sont transmises en série, dans une seule direction et sur une paire de fils. Les données sortantes sont marquées Tx (indiquant la Transmission) tandis que les données entrantes sont marquées Rx (indiquant la Réception). Pour créer un système de communication bidirectionnel, un minimum de trois fils sont nécessaires : Tx, Rx et GND (masse électrique). Le croisement des Tx et Rx entre les deux systèmes permet à chaque unité de parler à l'unité opposée.

Chaque octet peut être transmis à tout moment et séquentiellement.

Pour établir une communication efficace en RS-232, il est nécessaire de définir le protocole utilisé et la séquence des données numériques utilisées.

### VITESSE DE TRANSMISSION (BAUD RATE)

Le baud rate est simplement la vitesse de transmission mesurée en bits par seconde. Il définit la fréquence d'envoi d'un bit.

Pour un débit en bauds de 2400 (2400 bps), la fréquence est de 2400Hz et la période d'un bit est de 1/2400 ou 416,6us. C'est la valeur qu'un récepteur utilisera pour récupérer correctement les bits du flux de données.

### NIVEAUX DE TENSIONS SEUILS DU RECEPTEUR

À l'entrée du récepteur, les niveaux de tension minimum ou seuil de détection sont définis comme  $\pm 3V$  et peuvent atteindre  $\pm 25V$ .

C'est-à-dire que pour recevoir un 0 logique, la tension doit être supérieure à 3V.

Pour recevoir un 1 logique, la tension doit être inférieure à -3V.

Cela permet de tenir compte des pertes au fur et à mesure que le signal se propage sur le câble et assure l'immunité au bruit, c'est-à-dire que tout bruit parasite jusqu'à un niveau de  $\pm 3V$  peut être toléré sans qu'il n'ait d'effet sur le récepteur et les données.

### BIT DE DEPART

Au début de chaque transmission, un bit de départ est transmis, indiquant au récepteur qu'un octet de données est sur le point de suivre.

Le bit de départ permet au récepteur de se synchroniser avec les bits de données puisqu'il peut voir le front montant du signal sur la ligne.

Une fois que le bit de départ est trouvé, le récepteur sait où se trouveront les bits suivants, car il dispose de la période d'échantillonnage (dérivée du baud rate) dans le cadre du processus d'initialisation.

**C'est pourquoi vous devez définir les mêmes paramètres pour chaque appareil communiquant.**

C'est-à-dire la vitesse de transmission, le nombre de bits d'arrêt, le nombre de bits de données et le bit de parité (activé ou désactivé).

## BITS DE DONNEES

Les bits de données suivent le bit de départ.

Il y aura généralement sept ou huit bits de données avec le LSB (bit de poids faible) transmis en premier.

La raison pour laquelle vous pouvez choisir entre sept ou huit est que l'ASCII est composé de l'alphabet dans les sept premiers bits (ainsi que des caractères de contrôle).

Le huitième bit étend le jeu de caractères pour les symboles graphiques.

Les autres tailles de bits de données sont 5, 6, 8 et 9 bits. Toutefois, la longueur des bits est généralement fixée à 8 bits, ce qui est très communément utilisé.

## LE BIT DE PARITE

Le bit de parité RS232 est un mécanisme de détection d'erreur minimal. Vous pouvez utiliser soit une parité impaire, soit une parité paire, soit aucune parité du tout.

Au niveau du récepteur, le bit de parité est utilisé pour dire si une erreur s'est produite pendant la transmission.

**Parité paire :** le bit ajouté aux données est positionné de telle sorte que le nombre d'états 1 soit pair sur l'ensemble donné + bit de parité

**Parité impaire :** le bit ajouté aux données est positionné de telle sorte que le nombre d'états 1 soit impair sur l'ensemble donné + bit de parité


## LE BIT D'ARRET

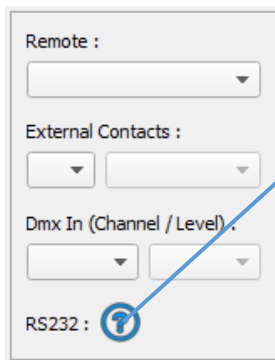
Le bit d'arrêt RS232 donne simplement une période de temps avant que le bit de départ suivant puisse être transmis. C'est le sens inverse du bit de départ et, de ce fait, il permet de voir et détecter ensuite le bit de départ.

S'il n'y avait pas de bit d'arrêt, le dernier bit du flux de données serait le bit de parité (ou le bit de données si la parité n'est pas active).

Le bit d'arrêt peut être défini en choisissant une période de 1, 1,5 ou 2 bits.

## COMMENT UTILISER LE RS232 AVEC NOS INTERFACES

Le mode autonome permet d'utiliser le protocole RS232 comme récepteur pour contrôler l'interface DMX via un autre appareil avec les commandes décrites dans la rubrique d'aide des logiciels. 



### RS232 Protocol

**i** -- Specifications --  
Asynchronous, 9600 bps, No Parity, 8 Data Bit, 2 Stop Bit

-- General --

Start of Text : STX (= 0x02) or \$ (=0x24)  
End of Text : ETX (= 0x03) or & (=0x26)

ZONEX : Set the current Zone (X = A, B, C, D, E)  
ZONEG : General Zone  
SCXXX : Start / Stop the scene XXX of the current zone  
(XXX = 001-255 / SC000 = Black out)

STOPO : Stop  
BLACK : Black Out

-- Commands for the current scene --

PLAY0 : Play  
PAUSE : Pause  
DIM++ : Dimmer +  
DIM-- : Dimmer -  
DIM+X : Set positive dimmer value (X = 0-9 / 0 = default dimmer value)  
DIM-X : Set negative dimmer value (X = 0-9 / 0 = default dimmer value)  
SPD++ : Speed +  
SPD-- : Speed -  
SPD+X : Set positive speed value (X = 0-9 / 0 = default speed value)  
SPD-X : Set negative speed value (X = 0-9 / 0 = default speed value)

-- Colors --

CLRBX : Trigger button X of the color mode (X = 1-8)  
CLRXX : Trigger color X of the predefined colors (X = 00-99)  
CLR00 : Turn off color

-- Example (Start scene 2) --

ASCII Characters : [STX] S C 0 0 2 [ETX]  
Hexadecimal values : 0x02 0x53 0x43 0x30 0x30 0x32 0x03  
Decimal values : 2 83 67 48 48 50 3

OK

Connectez l'émetteur RS232 à l'interface RS232 (Pin Tx, Rx et Masse) et envoyez les lignes de commande ASCII dédiées dont vous avez besoin.

Réglez les bons paramètres de l'émetteur RS232 : Asynchrone, 9600 bps, pas de parité, 8 bits de données, 2 bits d'arrêt.

**Les commandes ASCII doivent être envoyées une seule fois pour être traitées par l'interface.**

EXEMPLE : DEMARRER LA SCENE 1

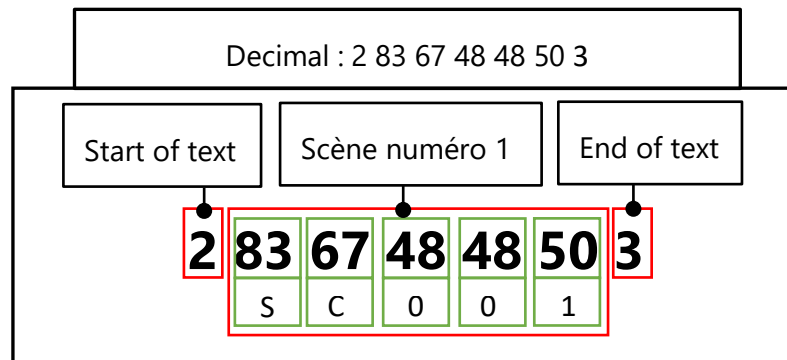
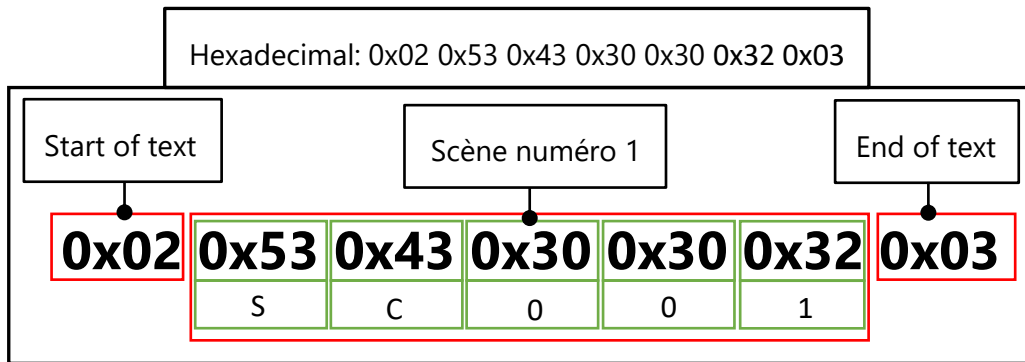
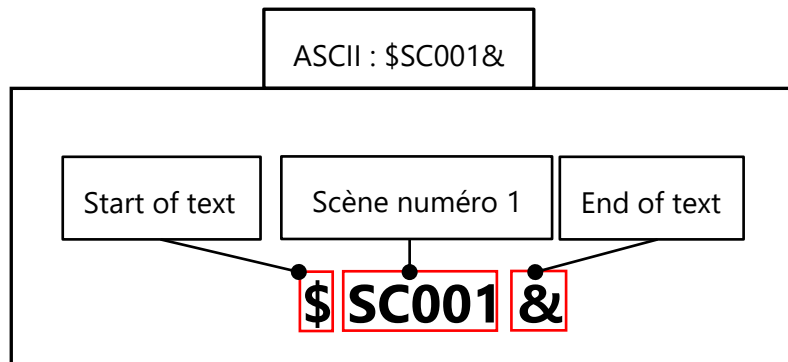


TABLE ASCII

# ASCII TABLE

Decimal	Hexadecimal	Binary	Octal	Char	Decimal	Hexadecimal	Binary	Octal	Char	Decimal	Hexadecimal	Binary	Octal	Char
0	0	0	0	[NULL]	48	30	110000	60	0	96	60	1100000	140	`
1	1	1	1	[START OF HEADING]	49	31	110001	61	1	97	61	1100001	141	a
2	2	10	2	[START OF TEXT]	50	32	110010	62	2	98	62	1100010	142	b
3	3	11	3	[END OF TEXT]	51	33	110011	63	3	99	63	1100011	143	c
4	4	100	4	[END OF TRANSMISSION]	52	34	110100	64	4	100	64	1100100	144	d
5	5	101	5	[ENQUIRY]	53	35	110101	65	5	101	65	1100101	145	e
6	6	110	6	[ACKNOWLEDGE]	54	36	110110	66	6	102	66	1100110	146	f
7	7	111	7	[BELL]	55	37	110111	67	7	103	67	1100111	147	g
8	8	1000	10	[BACKSPACE]	56	38	111000	70	8	104	68	1101000	150	h
9	9	1001	11	[HORIZONTAL TAB]	57	39	111001	71	9	105	69	1101001	151	i
10	A	1010	12	[LINE FEED]	58	3A	111010	72	:	106	6A	1101010	152	j
11	B	1011	13	[VERTICAL TAB]	59	3B	111011	73	;	107	6B	1101011	153	k
12	C	1100	14	[FORM FEED]	60	3C	111100	74	<	108	6C	1101100	154	l
13	D	1101	15	[CARRIAGE RETURN]	61	3D	111101	75	=	109	6D	1101101	155	m
14	E	1110	16	[SHIFT OUT]	62	3E	111110	76	>	110	6E	1101110	156	n
15	F	1111	17	[SHIFT IN]	63	3F	111111	77	?	111	6F	1101111	157	o
16	10	10000	20	[DATA LINK ESCAPE]	64	40	1000000	100	@	112	70	1110000	160	p
17	11	10001	21	[DEVICE CONTROL 1]	65	41	1000001	101	A	113	71	1110001	161	q
18	12	10010	22	[DEVICE CONTROL 2]	66	42	1000010	102	B	114	72	1110010	162	r
19	13	10011	23	[DEVICE CONTROL 3]	67	43	1000011	103	C	115	73	1110011	163	s
20	14	10100	24	[DEVICE CONTROL 4]	68	44	1000100	104	D	116	74	1110100	164	t
21	15	10101	25	[NEGATIVE ACKNOWLEDGE]	69	45	1000101	105	E	117	75	1110101	165	u
22	16	10110	26	[SYNCHRONOUS IDLE]	70	46	1000110	106	F	118	76	1110110	166	v
23	17	10111	27	[ENG OF TRANS. BLOCK]	71	47	1000111	107	G	119	77	1110111	167	w
24	18	11000	30	[CANCEL]	72	48	1001000	110	H	120	78	1111000	170	x
25	19	11001	31	[END OF MEDIUM]	73	49	1001001	111	I	121	79	1111001	171	y
26	1A	11010	32	[SUBSTITUTE]	74	4A	1001010	112	J	122	7A	1111010	172	z
27	1B	11011	33	[ESCAPE]	75	4B	1001011	113	K	123	7B	1111011	173	{
28	1C	11100	34	[FILE SEPARATOR]	76	4C	1001100	114	L	124	7C	1111100	174	
29	1D	11101	35	[GROUP SEPARATOR]	77	4D	1001101	115	M	125	7D	1111101	175	}
30	1E	11110	36	[RECORD SEPARATOR]	78	4E	1001110	116	N	126	7E	1111110	176	~
31	1F	11111	37	[UNIT SEPARATOR]	79	4F	1001111	117	O	127	7F	1111111	177	[DEL]
32	20	100000	40	[SPACE]	80	50	1010000	120	P					
33	21	100001	41	!	81	51	1010001	121	Q					
34	22	100010	42	"	82	52	1010010	122	R					
35	23	100011	43	#	83	53	1010011	123	S					
36	24	100100	44	\$	84	54	1010100	124	T					
37	25	100101	45	%	85	55	1010101	125	U					
38	26	100110	46	&	86	56	1010110	126	V					
39	27	100111	47	'	87	57	1010111	127	W					
40	28	101000	50	(	88	58	1011000	130	X					
41	29	101001	51	)	89	59	1011001	131	Y					
42	2A	101010	52	*	90	5A	1011010	132	Z					
43	2B	101011	53	+	91	5B	1011011	133	[					
44	2C	101100	54	,	92	5C	1011100	134	\					
45	2D	101101	55	-	93	5D	1011101	135	]					
46	2E	101110	56	.	94	5E	1011110	136	^					
47	2F	101111	57	/	95	5F	1011111	137	_					

CONNEXION RS232

