



ETUDE DE LA LIAISON SERIE



1 Mise en situation

Une liaison série est une liaison bidirectionnelle asynchrone.

Q1 : Définir le terme asynchrone.



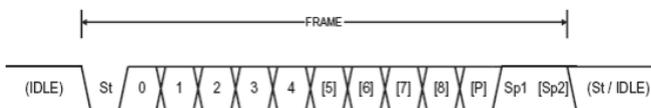
Q2 : Que signifie le terme liaison bidirectionnelle ?



La trame d'une liaison série est rappelée ci-dessous¹ :

L'USART permet de paramétrer la trame d'entrée ou de sortie comme suit :

- un bit de start (st)
- 5,6,7 ou 8 bits de données (nous choisirons 8 bits)
- un bit de parité paire, impaire ou aucun (P)
- un ou 2 bits de stop (Sp1 et Sp2), nous choisirons 1 bit de stop.



Q3 : Donner la valeur du bit de start et du bit de stop :



2 Le code ASCII

Q4 : Avec la [table des codes ASCII](#) encoder la chaîne de caractère 'Bonjour le monde'.



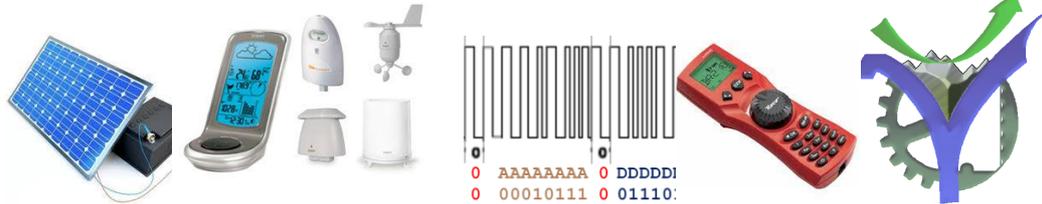
On reçoit d'un équipement une valeur numérique écrite en DCB et ASCII :

\$2B 33 35 2E 34 32 37, (chaque chiffre est codé par sa valeur en ascii).

Q5 : Quelle est la valeur numérique envoyée ?



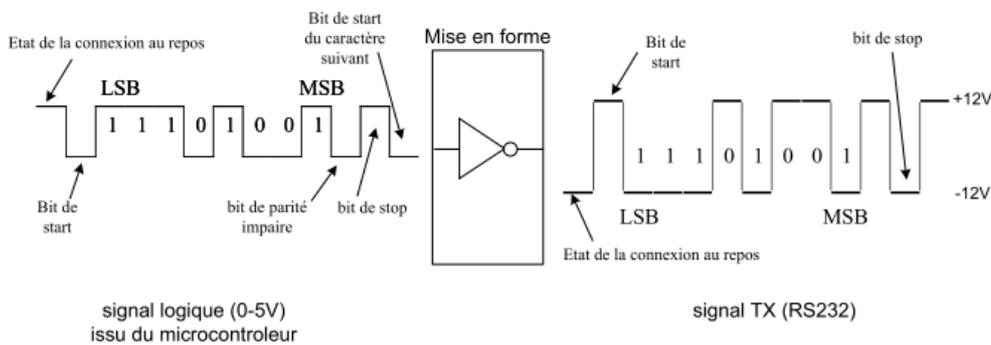
¹ D'après TD8_USART IUT NICE JLS.



3 Mise en œuvre d'une liaison série

Les UART Universal Asynchronous Receiver Transmitter gèrent les échanges au niveau des microcontrôleurs. Ce sont des circuits complexes qui rendent l'utilisation de la liaison série transparente. Ils fonctionnent donc sous des niveaux de tension logique 0-5V ou bien 0-3V.

Pour que ces signaux deviennent conformes à la norme RS232 une adaptation de niveau est nécessaire selon le schéma ci-dessous :



Q6 : Donner à l'examen du schéma ci-dessus la correspondance entre les niveaux logiques 'serial level' TTL et les signaux RS232 :

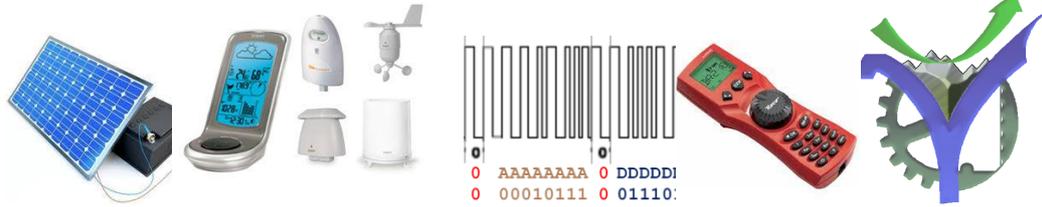


Q7 : Dans l'envoi d'un octet quel est l'ordre d'envoi des bits successifs ?



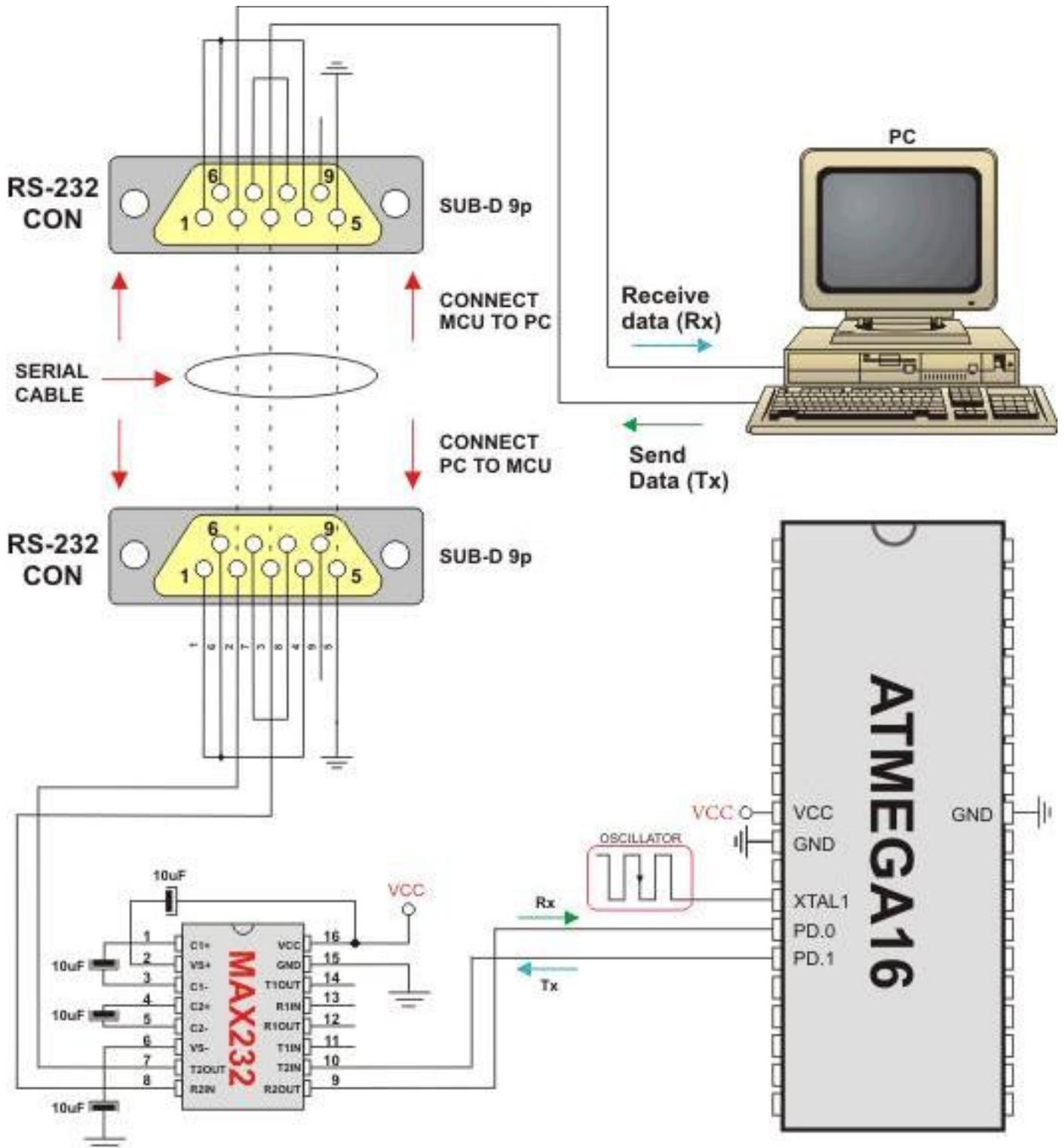
Q8 : Codage Ascii compléter le tableau ci-dessous ([voir table Ascii](#)):

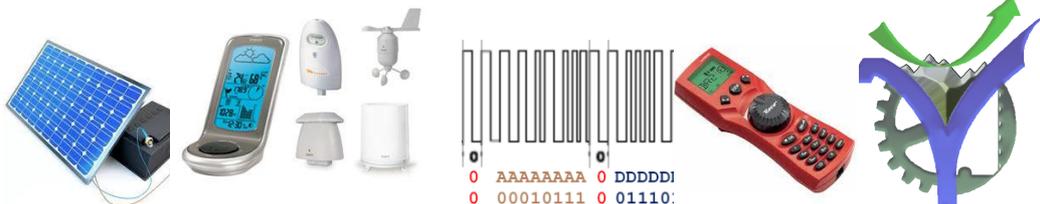
Code Ascii	IDLE	Start	b0	b1	b2	b3	b4	b5	b6	b7	Stop	IDLE
\$ <input type="text"/>		0	0	0	0	0	0	0	1	0	1	
\$ W		0									1	
\$ <input type="text"/>		0	1	1	0	1	1	1	1	0	1	
\$ LF		0									1	



Liaison série RS232 entre un ordinateur de type PC, (ou RaspberryPI ou ...), et un équipement à base de microcontrôleur (ou PSoC).

Q9 : Décrire le cheminement des signaux Tx et Rx en provenance de l'ordinateur.



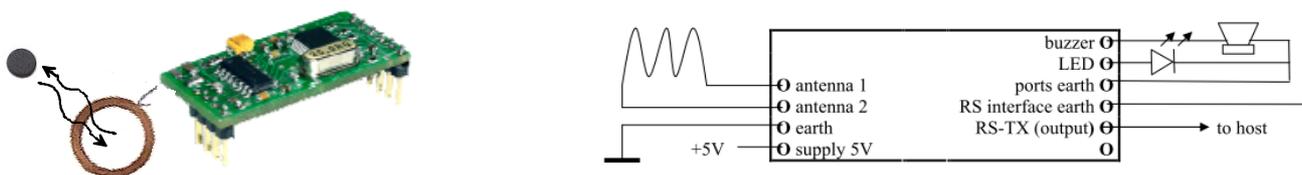


Liaison entre deux circuits ou périphériques 'logiques'

Dans ce cas l'adaptation des signaux échangés à la norme RS232 est inutile, il suffit de faire correspondre directement les UART selon le principe ci-dessous :



Dans cette mise en œuvre les documentations techniques indiquent liaison série 'serial level TTL'. Illustrons cette application avec l'étude de la documentation technique d'un transpondeur RFID fonctionnant sur la fréquence 125 kHz le UM-005 :



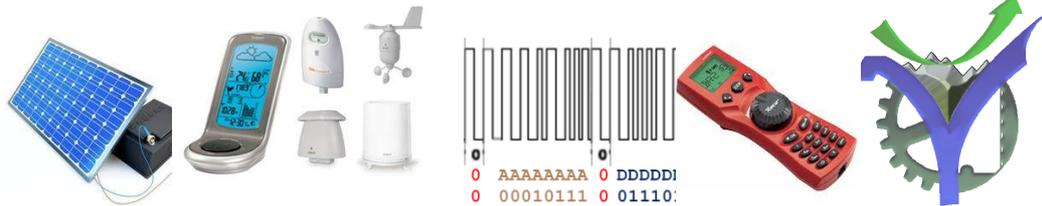
La présence d'un badge ou Tag RFID déclenche l'envoi d'une trame vers le host (circuit principal). La trame est organisée comme suit :

Frame format for serial transmission

Module address	Frame width	Response	Data	Operation code	CRCH	CRCL
1 byte	1 byte	1 byte	n bytes	1 byte	1 byte	1 byte

Where during read-out from transponder:
Module address - 0x01 always
Frame width – total number of response frame bytes = 0x0b
Response - 0x01
Data - ID1...5 – transponder ID (5 bytes)
Operation code – 0xff
CRCH, CRCL - CRC16 MSByte and LSByte respectively

Les spécifications techniques sont données ci-dessous :



Supply voltage Vdd: 4.5...5.5V
 Supply current: 5...55 mA
 Module rated operating radio frequency: 125 kHz
 Modulation type of data received from transponder:..... Manchester
 Baud rate of data received from transponder: RF/64 (1953 b/s)
 Maximum read-out frequency: 2 read-outs/sec
 Output current capacity buzzer, LED and RS-TX: 5 mA
 Transponder read-out distance (depending on used antenna): up to 12 cm
 Antenna inductance 1 mH +-5%
 RS232 transmission:9600 b/s, 8 data bits, 1 stop bit, no parity bit, with voltage levels TTL format compatible.

Q10 : Donner le nombre d'octets envoyés lors d'une détection.



Q11 : L'identificateur ou transpondeur ID est identifié par combien d'octets ?



Un enregistrement des trames reçues a été réalisé avec TestCom le résultat est donné ci-dessous on a présenté devant le détecteur 10 badges successivement donnant lieu à l'émission de 10 trames :

```

11/11: 01 0B 01 01 07 EB 80 92 FF 66 4F<OK>
11/11: 01 0B 01 01 07 EB 80 92 FF 66 4F<OK>
11/11: 01 0B 01 01 07 EB 80 92 FF 66 4F<OK>
11/11: 01 0B 01 01 07 EB 80 92 FF 66 4F<OK>
11/11: 01 0B 01 01 07 EB 8C C2 FF 1D 91<OK>
11/11: 01 0B 01 01 07 EB 8C C2 FF 1D 91<OK>
11/11: 01 0B 01 01 07 EB 8C C2 FF 1D 91<OK>
11/11: 01 0B 01 01 07 EB 8C C2 FF 1D 91<OK>
11/11: 01 0B 01 01 07 EB 8C C2 FF 1D 91<OK>
11/11: 01 0B 01 01 07 EB 8C C2 FF 1D 91<OK>
    
```

Q12 : Combien y a-t-il de badges différents détecté ?



Q13 : Donner l'ID des badges détectés.





0 AAAAAA 0 DDDDDI
0 00010111 0 01110:

Table des codes ASCII

[Retour à la question 2](#)

[Retour à la question 6](#)

MSB \ LSB		0	1	2	3	4	5	6	7
		000	001	010	011	100	101	110	111
0	0000	NUL	DLE	SP	0	@	P	`	p
1	0001	SOH	DC1	!	1	A	Q	a	q
2	0010	STX	DC2	"	2	B	R	b	r
3	0011	ETX	DC3	#	3	C	S	c	s
4	0100	EOT	DC4	\$	4	D	T	d	t
5	0101	ENQ	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	'	7	G	W	g	w
8	1000	BS	CAN	(8	H	X	h	x
9	1001	HT	EM)	9	I	Y	i	y
A	1010	LF	SUB	*	:	J	Z	j	z
B	1011	VT	ESC	+	;	K	[k	}
C	1100	FF	FS	,	<	L	\	l	
D	1101	CR	GS	-	=	M]	m	{
E	1110	SO	RS	.	>	N	^	n	~
F	1111	SI	US	/	?	O	_	o	DEL