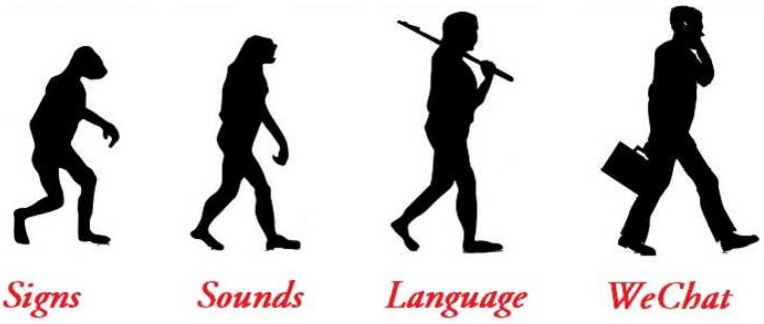
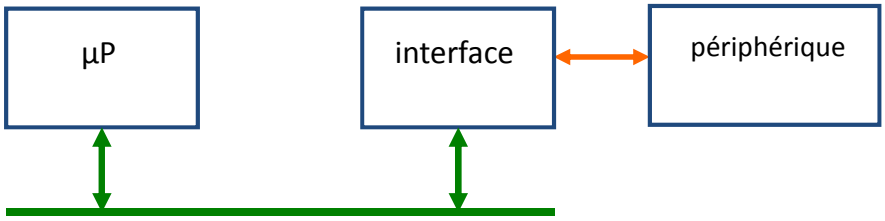
LES MODES DE COMMUNICATION



L'interface série RS232

Les transmissions numériques

- Le µP est amené à transmettre des données vers des périphériques qui sont à proximité ou éloignés.



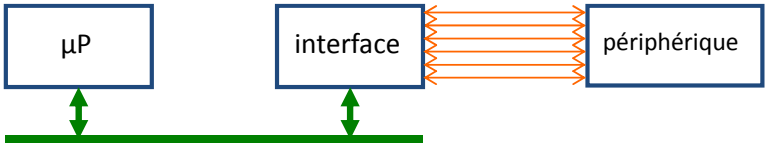
- Le transfert de données doit satisfaire trois exigences :

1. Distance (portée)
2. Vitesse (débit)
3. Sécurité

Les modes de transmission

Parallèle :

- Plusieurs bits sont transmis en même temps
- Débit plus élevée à fréquence égale
- Portée faible



Série :

- Les bits sont transmis l'un après l'autre
- Compense le faible débit par la fréquence
- Portée plus élevée

La transmission série

Synchrone :

- L'horloge est transmise avec les données.
- Des signaux de synchronisation sont intercalés entre les trames.
- Les trames sont longues : plus de données.

Asynchrone :

- L'émetteur et le récepteur ont chacun une horloge.
- Les trames sont courtes.

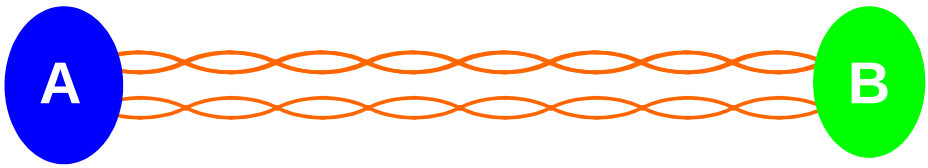
Normes RS232 et RS485

RS232 :

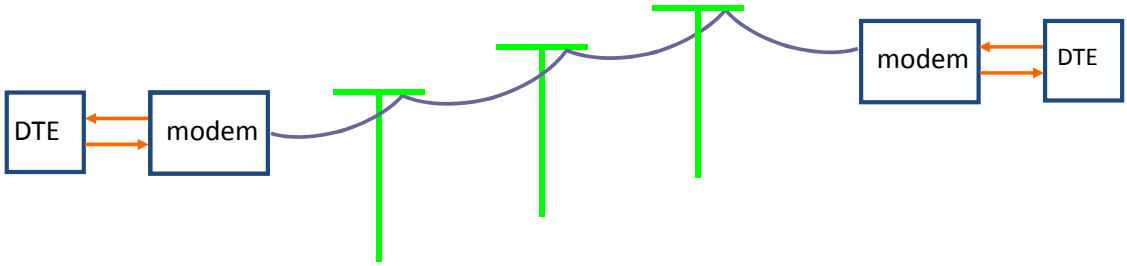
- La plus utilisée
- Portée: 150 m à 9600 Bauds

RS485 :

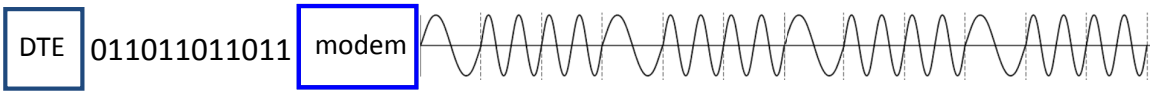
- 2 paires différentielles pour éviter les perturbations
- Distance : autour de 1000 m



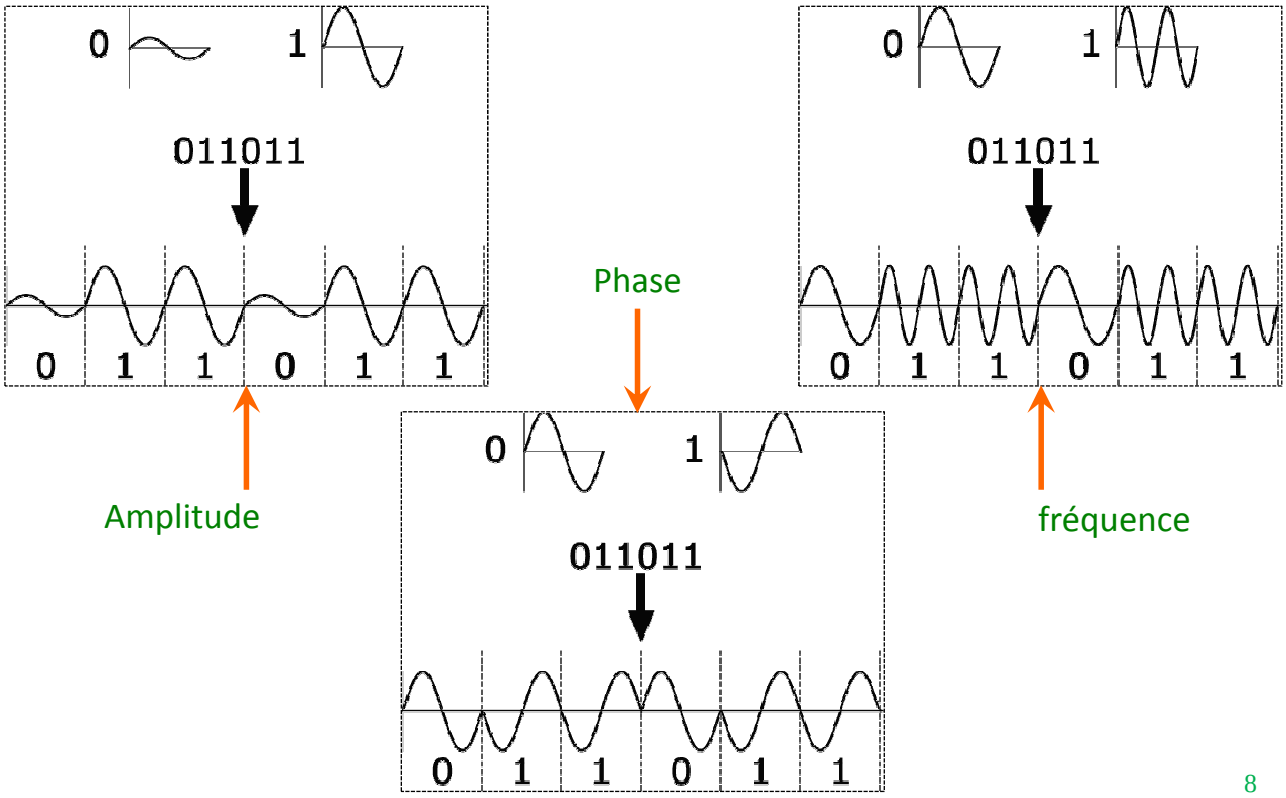
Liaison par modem



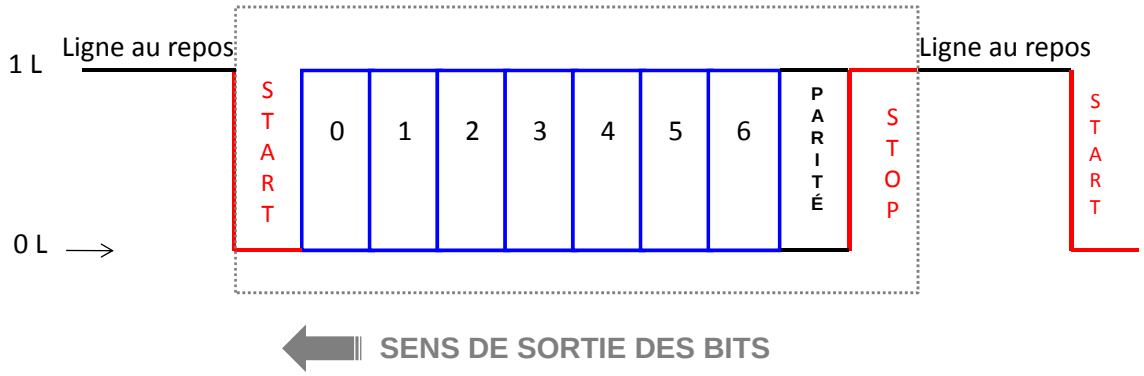
- Le modem transforme les signaux numériques en signaux analogiques au départ.
- A l'arrivée, il fait la transformation inverse.
- La portée est celle du réseau téléphonique (ex: France – Australie).



Principe de la modulation

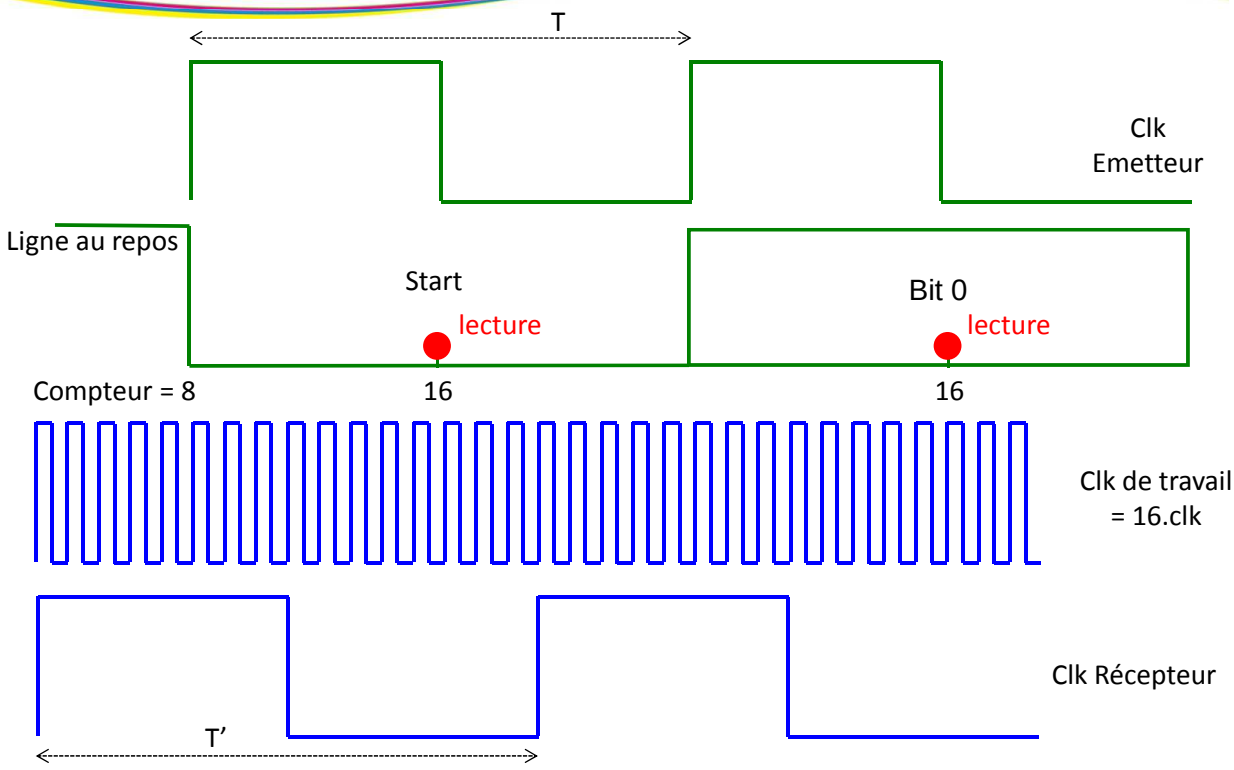


Format d'une trame



- La ligne au repos est au niveau logique 1.
- Le bit de start (zéro logique) signale le début de la trame.
- La donnée est sur 5, 6, 7 ou 8 bits par configuration.
- Le bit de parité (optionnel) permet de détecter une éventuelle erreur
- Le stop (niveau 1 logique) de longueur entre 1 et 2 bits clôture la trame.
- Tous les bits ont la même durée T fixée par l'horloge et son diviseur.

Lecture de la donnée



Sur 10 bits, l'écart maximum des fréquences des horloges est de 5%

Vitesse de transfert

Quelques vitesses normalisées

50 bauds	2400 bauds
75 bauds	4800 bauds
110 bauds	9600 bauds
150 bauds	19200 bauds
300 bauds	38400 bauds
600 bauds	56000 bauds
1200 bauds	115200 bauds

- Chaque bit dure une période T comme c'est déjà annoncé.
- La vitesse en Bauds = nombre de bits transmis par seconde.
- En full duplex, la vitesse est en général la même dans les deux sens.

Le code ASCII de base

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STH	ETH	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	CD2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	spc	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

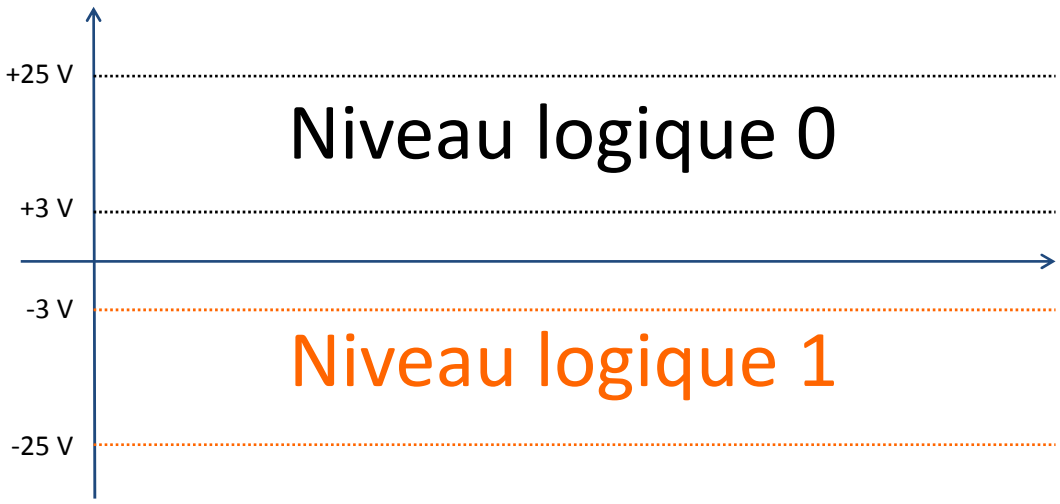
- ASCII = American Standard Code for Information Interchange.
- Les caractères à fond bleu sont des caractères de contrôle.
- Le code ASCII à 7 bits est bien adapté à l'anglais.
- Il n'y a pas de caractères accentués (pb pour les autres langues).

Le code ASCII étendu (ANSI)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STH	ETH	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	CD2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL
8	€	□	‚	ƒ	„	…	†	‡	ˆ	‰	Š	‹	Œ	□	Ž	□
9	□	'	'	"	"	•	–	—	˜	™	š	›	œ	□	ž	Ÿ
A		¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	-	®	¯
B	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

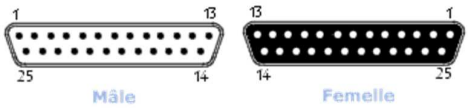
Pour un bit de plus !

Niveaux logiques et électriques



- On travaille en bipolaire ±V pour éviter de véhiculer une valeur moyenne non nulle.
- On adopte la logique négative 1L = -V et 0L = +V.

Les connecteurs DB25



Nomenclature des broches :

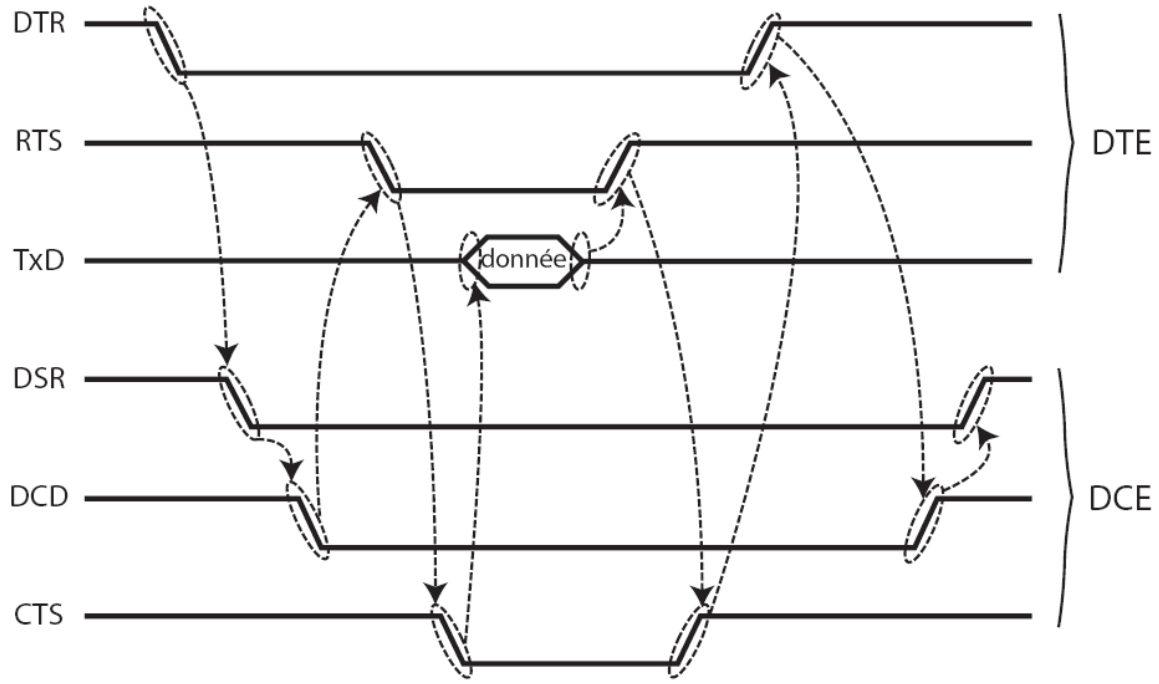
Numéro	Nom	Désignation
2	Transmit Data	Transmission de données
3	Receive Data	Réception de données
4	Request to Send	Demande d'émission
5	Clear to Send	Prêt à émettre
6	Data Set Ready	Données prêtes
7	Signal Ground	Masse logique
8	Carrier Detect	Détection de porteuse
20	Data Terminal Ready	Terminal prêt
22	Ring Indicator	Indicateur de sonnerie

Les connecteurs DB9

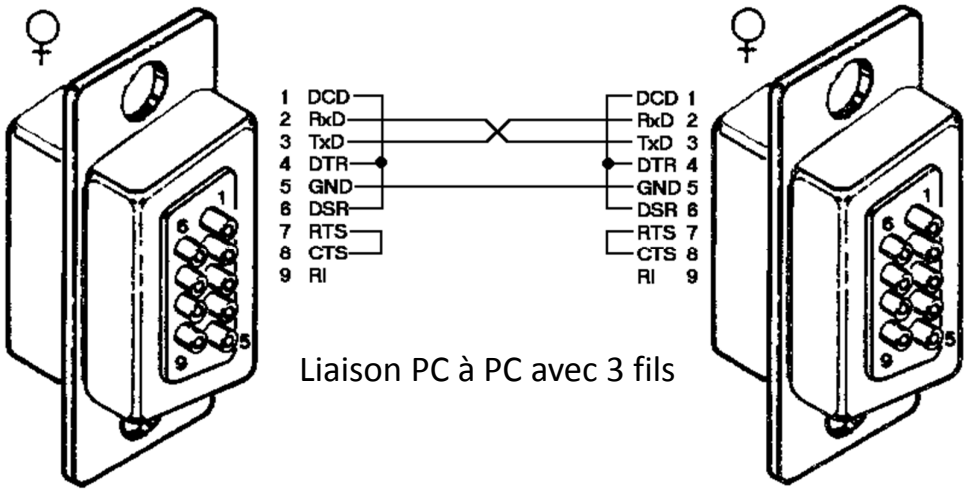
Nomenclature des broches :

Numéro	Nom	Désignation
1	Carrier Detect	Détection de porteuse
2	Receive Data	Réception de données
3	Transmit Data	Transmission de données
4	Data Terminal Ready	Terminal prêt
5	Signal Ground	Masse logique
6	Data Set Ready	Données prêtes
7	Request to Send	Demande d'émission
8	Clear to Send	Prêt à émettre
9	Ring Indicator	Indicateur de sonnerie
	Shield	Blindage

Protocole de communication



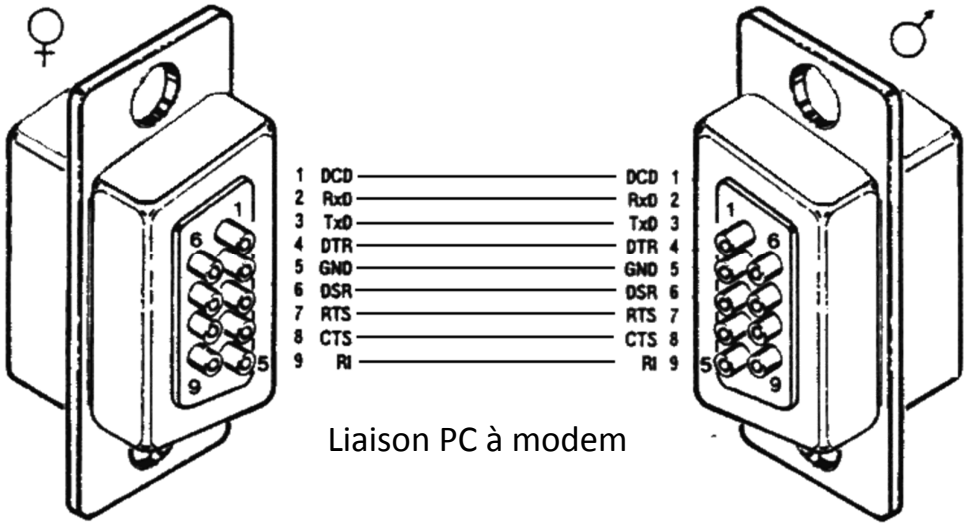
Connexion entre 2 DTE



Liaison PC à PC avec 3 fils

- DTE (Data Terminal Equipment) = Appareil qui génère des données (ex : PC)

Connexion entre DTE et DCE

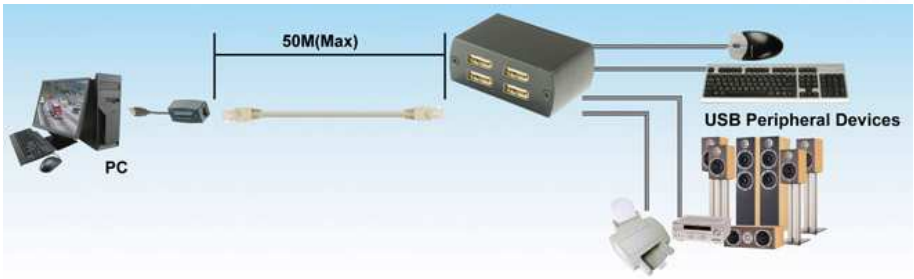


Liaison PC à modem

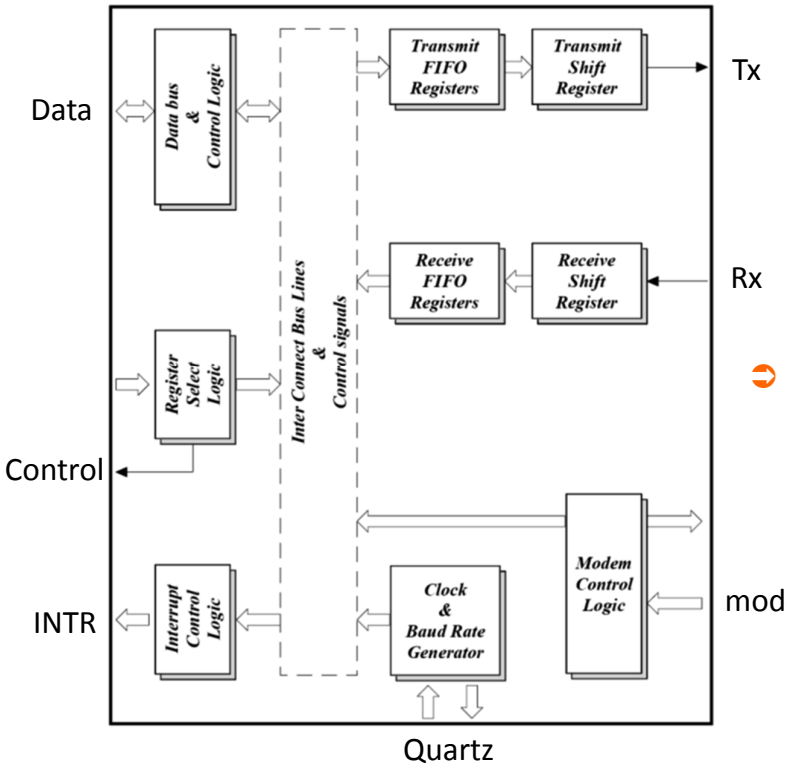
- DCE (Data Communication Equipment) = Appareil qui communique les données (ex : modem)

RS232 et USB

- Le port série RS232 est bien implanté dans le milieu industriel, le port USB équipe toutes les nouvelles machines. Pour communiquer en RS232 avec un port USB, des interfaces intelligentes et transparentes existent.



Universal Asynchronous Receiver Transmitter

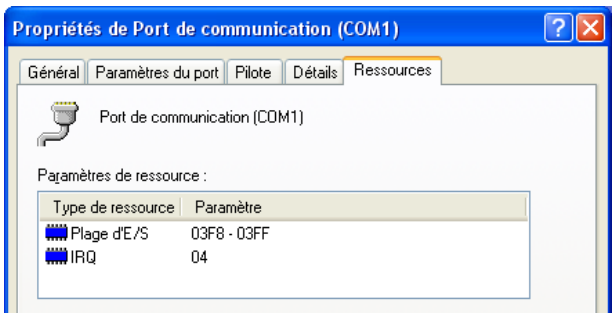


- L'UART est un circuit programmable qui sert d'interface entre le µP et le connecteur qui sert à brancher les périphériques.

Modes d'utilisation

- L'UART peut être utilisé selon deux modes : interruption ou scrutation.
  1. En mode scrutation (polling), le CPU lit en permanence l'état de certains registres pour connaître l'état de la transmission.
  2. En mode interruption, le CPU est averti par interruption à chaque fois que des données sont reçues.

- Lorsqu'un port série est installé sur un PC, il doit être configuré en lui affectant des adresses d'E/S spécifiques et un numéro d'interruption (IRQ). On utilise en général les adresses standard (0x3F8 pour COM1 et 0x2F8 pour COM2).



Le bus I2C

Présentation

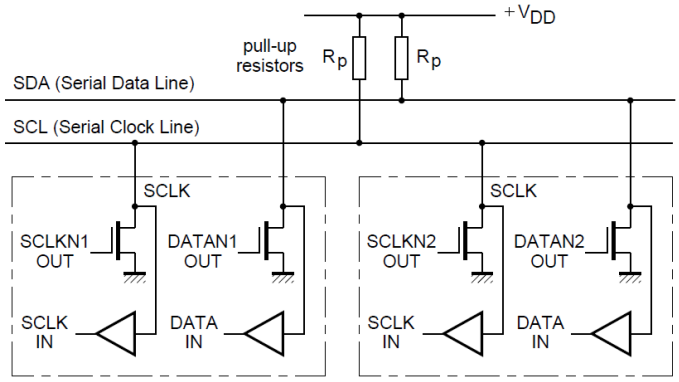
- *I2C : Inter Integrated Circuit.*
- Développé au début des années 80 par « Philips Semiconductor » pour minimiser les liaisons entre les différents circuits numériques utilisés en télévision et dans les appareils HiFi.
- Le but étant de faire communiquer différents circuits intégrés, en utilisant uniquement trois fils :
  1. SCL : horloge qui cadence la transmission des données (bus synchrone)
  2. SDA : signal de données (transmission série en half-duplex)
  3. GND : masse électrique de référence

Caractéristiques

- Deux signaux uniquement pour la transmission de données : SCL et SDA
- Une adresse unique pour chaque périphérique (esclave)
- Bus multi maîtres avec détection des collisions et arbitrage
- Transmission série, 8 bits de données, bidirectionnel, half-duplex
- Bus synchrone, cadencé à 100 Kbps (Sdandard mode) ou 400 Kbps (fast mode) ou 3,2 Mbps (high speed mode)
- Réjection des pics parasites par un filtrage intégré
- Le nombre de périphériques (esclaves) est limité par la capacitance du bus qui ne doit pas dépasser 400 pF et le nombre de bits d'adresse évidemment

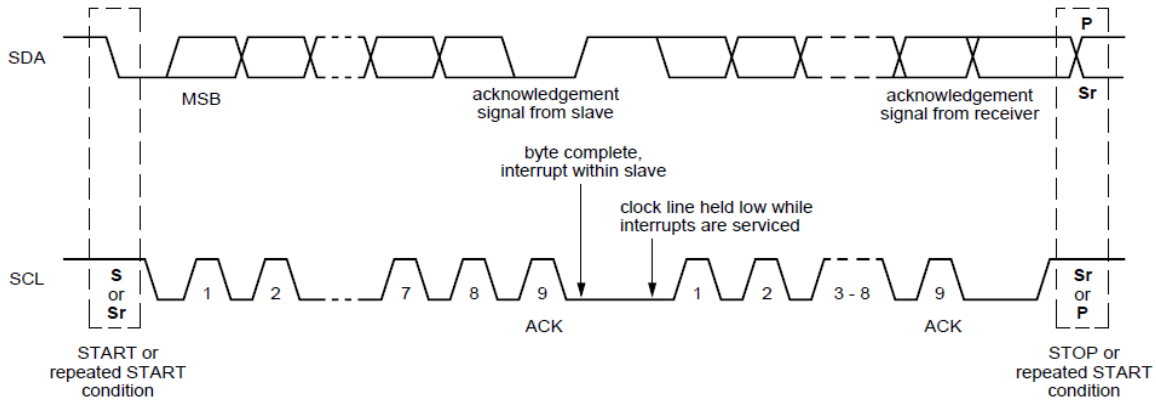
Principe des liaisons électriques

- Les signaux SCL et SDA sont de type « collecteur ouvert » avec des résistances de rappel vers V_{DD}
- Les lignes SCL et SDA ne peuvent être forcées qu'au niveau bas. Pour appliquer un niveau haut, les sorties sont mises en haute impédance
- Ce mode de fonctionnement évite les conflits électriques sur le bus



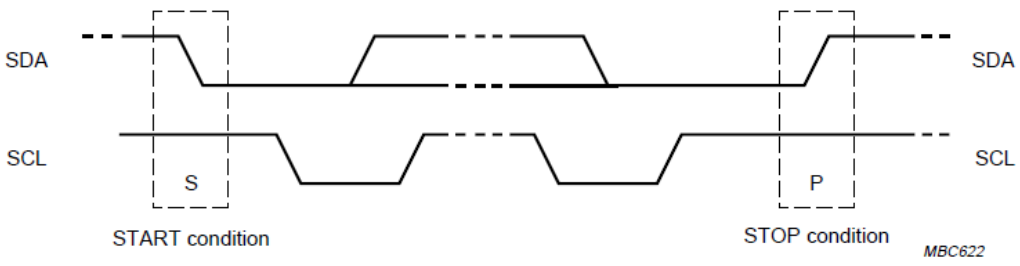
Format d'un transfert

- Au repos les signaux SCL et SDA sont au niveau haut
- Le transfert de données commence par un START et se termine par un STOP
- Un octet est transmis à chaque fois, le bit de poids fort d'abord. Après chaque ACK, un autre octet peut suivre
- Chaque bit est validé par une impulsion de l'horloge SCL



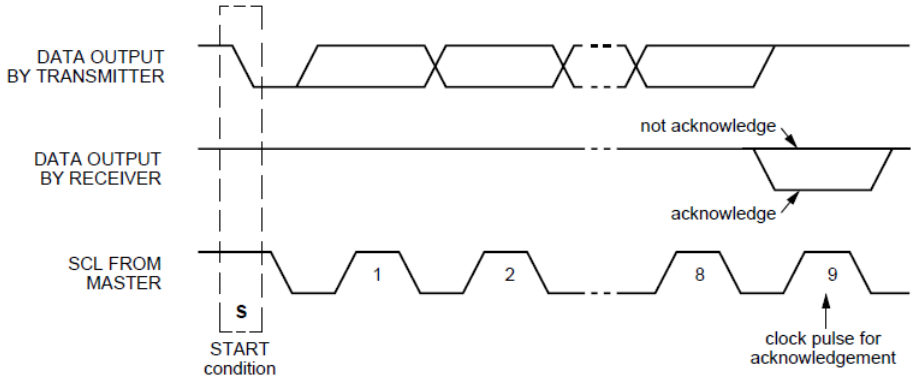
Le START et le STOP

- Le START et le STOP sont générés par le maître
- La condition de START est appliquée lorsque le signal SDA passe à 0 pendant que l'horloge SCL reste à 1
- Après avoir vérifié que le bus est libre et en avoir pris le contrôle, le circuit devient le maître : c'est lui qui génère l'horloge
- La condition de STOP est appliquée lorsque le signal SDA passe à 1 pendant que l'horloge SCL reste à 1



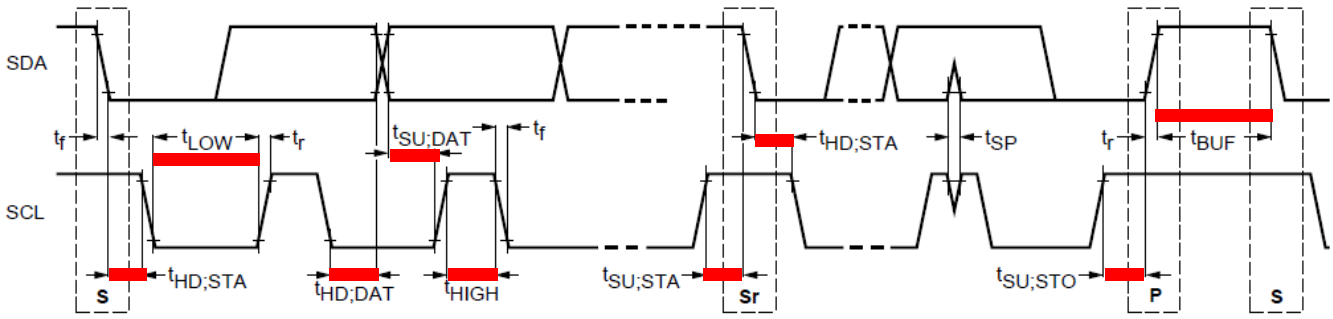
ACK : Acknowledge

- La réponse à tout transfert est obligatoire : Acknowledge
- L'esclave doit signifier son accord en mettant à 0 le signal SDA pendant l'impulsion SCL envoyée par le maître
- Si l'esclave signifie son désaccord en mettant SDA à 1, le maître abandonne le transfert



Spécifications : timing

- Il faut respecter un certain timing dans l'enchainement des signaux pour que le transfert se passe correctement
- La largeur des impulsion d'horloge doit être bien calibrée : t_{LOW} et t_{HIGH}
- Les temps de SETUP et de HOLD (t_{SU} et t_{HD}) doivent respecter certaines valeurs



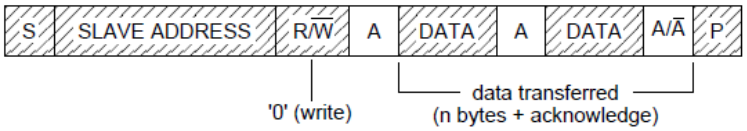
Spécifications : timing

PARAMETER	SYMBOL	STANDARD-MODE		FAST-MODE		UNIT
		MIN.	MAX.	MIN.	MAX.	
SCL clock frequency	f_{SCL}	0	100	0	400	kHz
Hold time (repeated) START condition. After this period, the first clock pulse is generated	$t_{HD;STA}$	4.0	–	0.6	–	µs
LOW period of the SCL clock	t_{LOW}	4.7	–	1.3	–	µs
HIGH period of the SCL clock	t_{HIGH}	4.0	–	0.6	–	µs
Set-up time for a repeated START condition	$t_{SU;STA}$	4.7	–	0.6	–	µs
Data hold time: for CBUS compatible masters (see NOTE, Section 10.1.3) for I²C-bus devices	$t_{HD;DAT}$	 5.0 0(2)	 – 3.45(3)	 – 0(2)	 – 0.9(3)	 µs µs
Data set-up time	$t_{SU;DAT}$	250	–	100(4)	–	ns
Rise time of both SDA and SCL signals	t_r	–	1000	$20 + 0.1C_b$(5)	300	ns
Fall time of both SDA and SCL signals	t_f	–	300	$20 + 0.1C_b$(5)	300	ns
Set-up time for STOP condition	$t_{SU;STO}$	4.0	–	0.6	–	µs
Bus free time between a STOP and START condition	t_{BUF}	4.7	–	1.3	–	µs
Capacitive load for each bus line	C_b	–	400	–	400	pF

Envoi d'une adresse

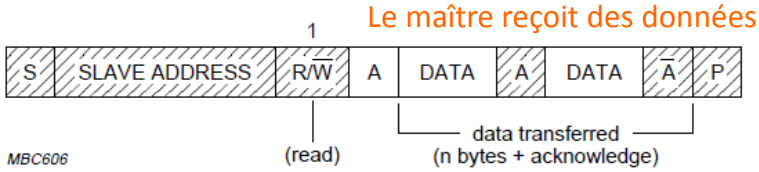
- Après le START, une adresse d'esclave est envoyée sur la ligne
- L'adresse est composée de 7 bits auxquels s'ajoute un bit de lecture/écriture : ce bit vaut 0 pour une écriture et 1 pour une lecture. Pour basculer de l'écriture vers la lecture ou inversement, un nouveau START doit être envoyé.
- Ensuite vient le transfert de données qui se termine par un STOP ou Repeat START pour changer de mode ou d'esclave

Le maître envoie des données



A = acknowledge (SDA LOW)
Ā = not acknowledge (SDA HIGH)
S = START condition
P = STOP condition

Le maître reçoit des données



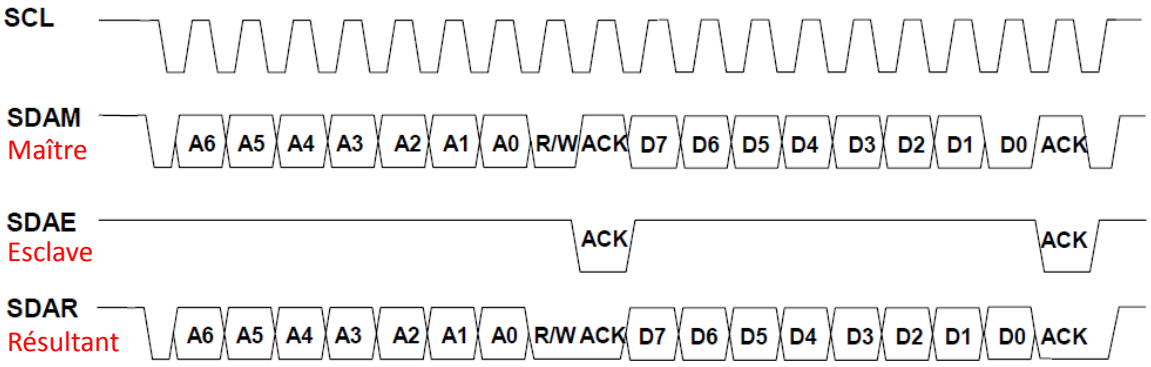
Envoi d'une adresse

- Dans le cas des mémoires, l'espace adressable est élevé : le premier octet sélectionne la mémoire, les octets suivants sélectionnent une case mémoire parmi l'espace adressable
- Les adresses 0000XXX et 1111XXX sont réservées pour des modes particuliers

SLAVE ADDRESS	$\overline{R/W}$ BIT	DESCRIPTION
0000 000	0	General call address
0000 000	1	START byte(1)
0000 001	X	CBUS address(2)
0000 010	X	Reserved for different bus format(3)
0000 011	X	Reserved for future purposes
0000 1XX	X	Hs-mode master code
1111 1XX	X	Reserved for future purposes
1111 0XX	X	10-bit slave addressing

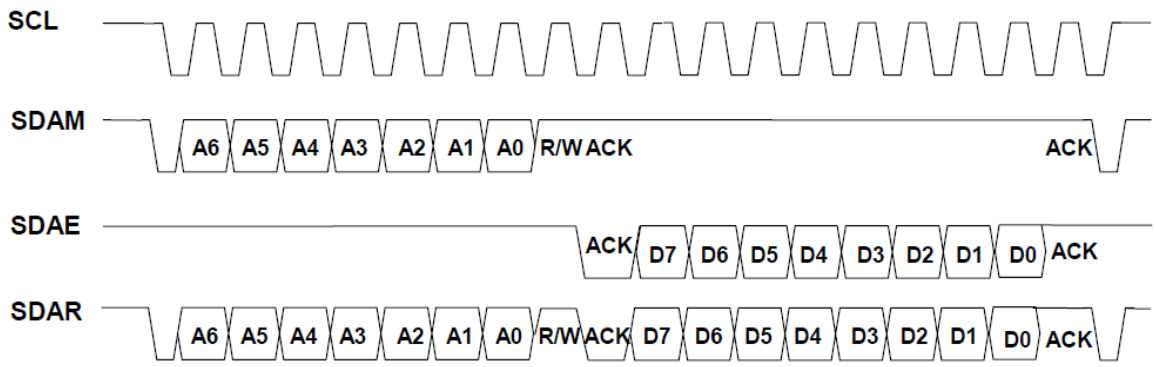
Envoi d'une donnée

- Après le START, le maître envoie l'adresse suivie d'un bit R/W à 0 pour choisir le mode écriture
- Il met ensuite la broche SDA en haute impédance pour laisser l'esclave signifier son accord ou désaccord
- Ensuite viennent les octets suivis chacun d'un Acknowledge



Réception d'une donnée

- Après le START, le maître envoie l'adresse suivie d'un bit R/W à 1 pour choisir le mode lecture
- Il met ensuite la broche SDA en haute impédance pour laisser l'esclave signifier son accord ou désaccord
- Il garde SDA en haute impédance pour lire les données, il termine par un NACK



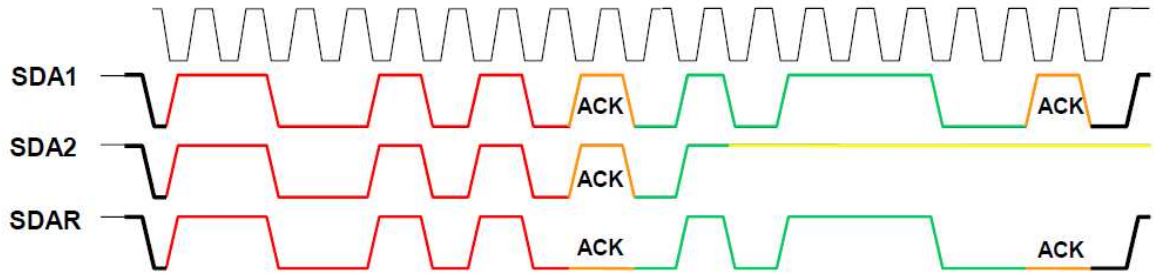
Gestion des conflits

- Le bus I2C autorise qu'il y ait plusieurs maîtres, ce qui crée forcément des conflits au moment de la prise en mains du bus
- Chaque maître peut prendre possession du bus dès que celui-ci est libre. Il est alors possible que deux maîtres veuillent se saisir du bus en même temps
- Il n'y a pas de problème électrique puisque les entrées/sorties sont à collecteur ouvert
- Il y'a par contre un conflit logique et risque de collision des données. Il faut éviter à tout prix la corruption des données

Prise de contrôle du bus

- Vérifier que le bus est libre : SCL et SDA à 1 (repos)
- Le dernier STOP date d'au moins 4,7 µs
- Si plusieurs maîtres, il est nécessaire de vérifier au fur et à mesure l'état des lignes SCL et SDA
- Plusieurs cas peuvent se présenter :
  1. Plusieurs maîtres envoient les mêmes données (extrêmement rare !), il n'y a pas de problème, ils continuent à émettre
  2. Un maître écrit un 1 et lit un 1 sur la ligne SDA, il continue à émettre
  3. Un maître écrit un 1 et lit un 0 sur la ligne SDA : cela signifie qu'un autre maître émet en même temps. Il perd l'arbitrage, il arrête d'émettre.

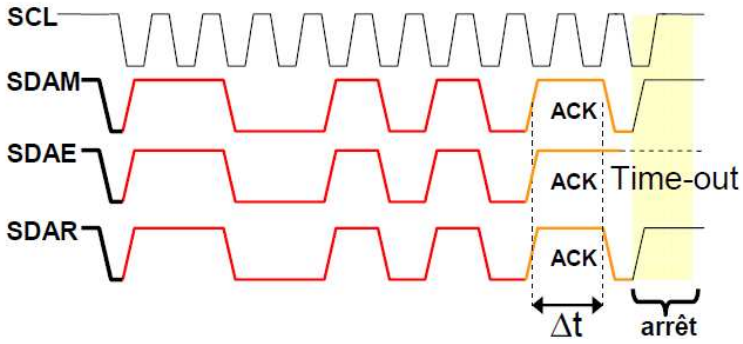
Exemple



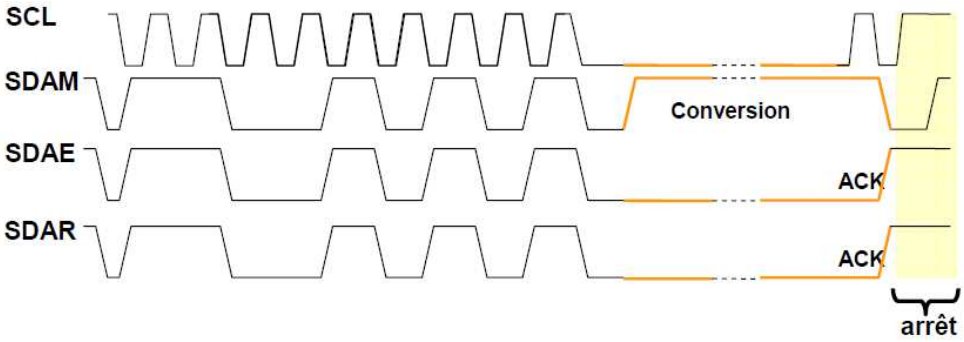
- Le premier octet est transmis normalement, il n'y a pas de conflit, les deux maîtres envoient les mêmes données
- Sur le 3ème bit de l'octet suivant, le maître n°2 envoie un 1 et lit un 0 sur SDAR, il perd l'arbitrage et arrête d'émettre. Il devient esclave
- Le maître n°1 ne voit aucun conflit et continue d'émettre normalement, comme si de rien n'était

Synchronisation

- Dans le cas des esclaves lents, le maître peut s'impatienter et perdre la main avec l'esclave avec qui il dialogue
- Sur la figure suivante, l'esclave qui n'a pas fini son travail (exemple d'un convertisseur CAN) ne donne pas son accord assez rapidement, le maître annule le transfert
- Il faut trouver un moyen pour synchroniser ces deux circuits qui ne travaillent pas manifestement à la même vitesse



Synchronisation



- Après l'octet, l'esclave force SDA à 0 (acquittement) et SCL à 0 et il démarre sa conversion
- Le maître voit que SCL est forcé à 0 par l'esclave, il relâche SCL et SDA et se met dans une boucle d'attente
- Lorsque l'esclave a terminé sa conversion, il relâche la ligne SCL
- Le maître voyant SCL à 1 et SDA à 0 va lire le résultat de la conversion