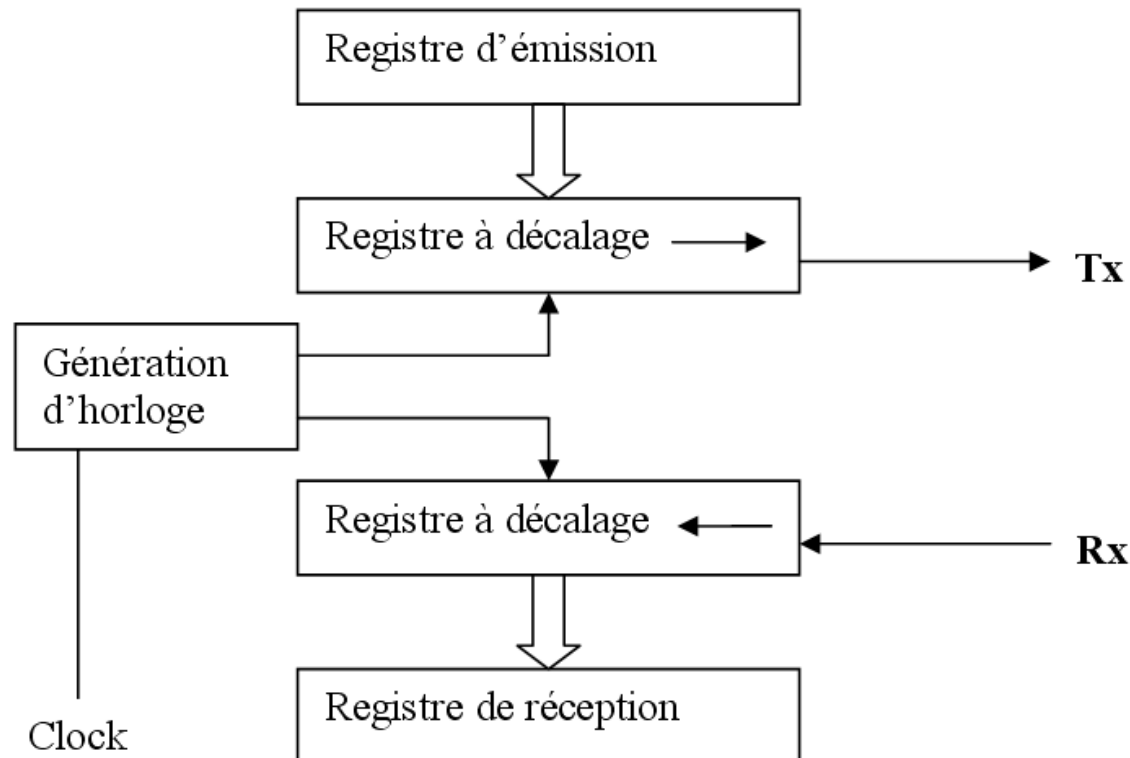


U(S)ART

Communication série synchrone et asynchrone



UART

- UART est un terme qui vient de l'anglais, ce sont les initiales de: *Universal Asynchronous Receiver Transmitter*.
- L'UART est donc un émetteur-récepteur asynchrone universel.
- En langage courant, c'est le composant utilisé pour faire la liaison entre l'ordinateur et le port série .

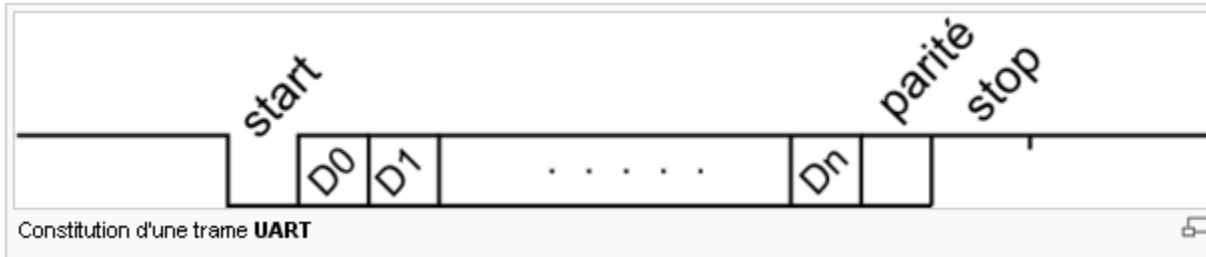
Communication série

- Une communication de données peut être soit série, soit parallèle.
- Une **communication parallèle** transmet plusieurs flux de données simultanés sur de multiples canaux (fil électrique, piste de circuit imprimé, fibre optique, etc.) tandis qu'une communication série ne transmet qu'un seul flux de données sur un seul canal (ou medium de communication).
- En télécommunications et en informatique, la **communication série** est l'action d'envoyer des données bit par bit, séquentiellement, sur un canal de communication ou un bus informatique.
- Les communications séries sont utilisées pour les communications à longue distance, et la plupart des réseaux informatiques car le coût élevé des câbles et des procédés de synchronisation rend impossible l'utilisation de communications parallèles.
- Les bus séries d'ordinateurs sont de nos jours les technologies les plus répandues et évoluées pour des transferts de données rapides.

Communication série

- On aurait tendance à penser au premier abord qu'une liaison série serait moins performante qu'une liaison parallèle, puisqu'elle transmet moins de données sur chaque top d'horloge.
- Cependant, dans la plupart des cas, la fréquence d'horloge d'une liaison série est beaucoup plus élevée que celle d'une liaison parallèle, et au final le taux de données transférées est par conséquent plus élevé. En effet :
 - pas de problème de décalage d'horloge entre les différents canaux (pour les liaisons série *asynchrones*),
 - seulement un câble pour une connexion série, donc un encombrement plus faible qui permet une meilleure isolation des perturbations extérieures.
 - diminution des risques de diaphonie et d'interférences entre plusieurs canaux.
- Généralement, une liaison série reste la meilleure solution car elle est beaucoup moins onéreuse.

UART

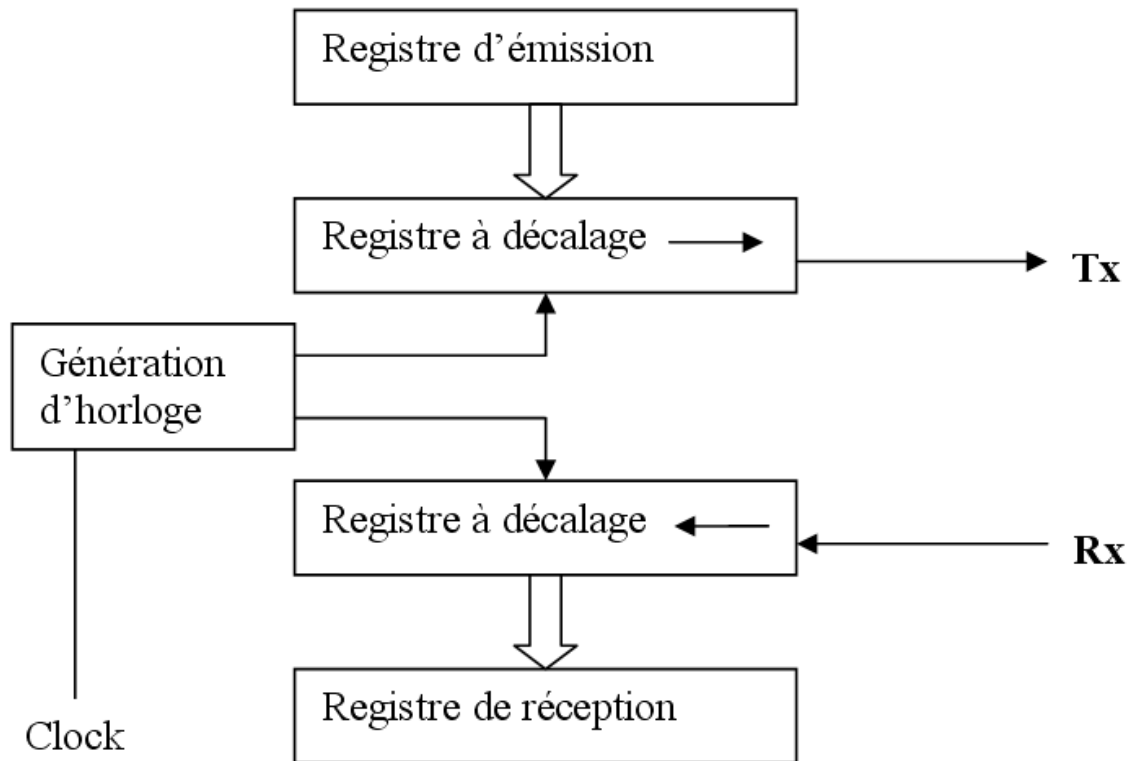


- Une trame UART est constituée des bits suivants :
 - un bit de start toujours à 0 : servant à la synchronisation du récepteur
 - les données : la taille peut varier (généralement entre 5 et 9 bits)
 - éventuellement un bit de parité paire ou impaire
 - et un bit de stop toujours à 1 (la durée peut varier entre 1 et 2 temps bit)
- Le niveau logique de repos est le 1.

Vitesse de transmission

- Afin de faciliter l'interopérabilité entre périphériques (PC, microcontrôleur, modem,...) des vitesses de transmission sont normalisées, l'unité baud correspondant à un temps bit :
 - * 110 bps
 - * 300 bps
 - * 1200 bps
 - * 2400 bps
 - * 4800 bps
 - * 9600 bps
 - * 19.200 bps
 - * 38.400 bps
 - * 57.600 bps
 - * 115.200 bps

UART: Diagramme de fonctionnement général



Exemples d'architectures de communication série

- Alphabet morse (télégraphie)
- RS-232 (ancien, faible coût, vitesse faible)
- EIA-422 (liaison point à point)
- EIA-485 (liaison multi-points)
- Universal Serial Bus (USB) (nouveau, vitesse moyenne, pour la connexion de périphériques)
- FireWire
- Fibre Channel (vitesse élevée, pour la connexion de systèmes de stockage de masse)
- InfiniBand (très haute vitesse, comparable au PCI)
- Serial Attached SCSI
- Serial ATA
- PCI Express

RS-232

- RS-232 (aussi appelé EIA RS-232C ou V.24) est une norme standardisant un port de communication de type série. Disponible sur presque tous les PC jusqu'au milieu des années 2000, il est communément appelé le « port série ».
- Sur les systèmes d'exploitation MS-DOS et Windows, les ports RS-232 sont désignés par les noms COM1, COM2, etc. Cela leur a valu le surnom de « ports COM », encore utilisé de nos jours.
- Le port RS-232 est fréquemment utilisé dans l'industrie pour connecter différents appareils électroniques (automate, appareil de mesure, etc.).
- De même, de nombreux terminaux embarqués (qu'ils soient GPS, modems, terminaux graphiques, etc.) utilisent le RS-232 comme méthode principale de communication avec l'extérieur. Fréquemment, les périphériques réseau (routeurs, commutateurs, etc.) sont équipés d'un port RS-232 au travers duquel il est possible de les configurer.

RS-232

- Bien que ce port de communication ait tendance à être remplacé par l'USB sur les PC, il reste encore très utilisé dans l'industrie, notamment grâce à sa robustesse et à sa simplicité.
- Ainsi, ce port est toujours d'actualité, en particulier dans les systèmes automatisés : le transfert de lignes de programme pour machines-outils à commande numérique s'effectuent toujours par liaison RS-232.
- En cas d'absence de ceux-ci, il existe des adaptateurs USB/port série.
- Ce port peut se présenter sous la forme d'un connecteur 9 ou 25 broches (le nom du connecteur est DB-9 ou DB-25 suivant le nombre de broches). Seule la version 25 broches est vraiment standardisée, la 9 broches est une adaptation d'IBM lors de la création du PC.



USB

- L'USB, *Universal Serial Bus* a été conçu au milieu des années 1990 afin de remplacer les nombreux ports externes d'ordinateur lents et incompatibles, notamment les ports série et parallèle.
- L'USB est une connexion à haute vitesse qui permet de connecter des périphériques externes à un ordinateur (hôte dans la littérature USB). Il supporte le branchement simultané de 127 périphériques, en série.
- L'USB est aujourd'hui présent sur tous les ordinateurs et est généralement utilisé pour brancher les imprimantes, les scanners, les modems et de nombreux appareils stockant des données, dont les clés USB.

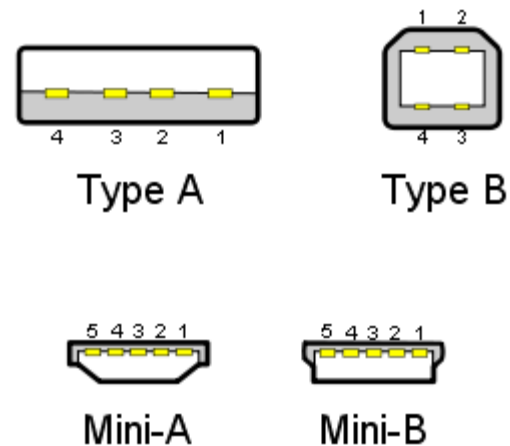


USB

- Le bus supporte les branchements et débranchements à chaud ("hot plug", sans avoir besoin de redémarrer l'ordinateur) et fournit l'alimentation électrique des périphériques, dans la limite des 500 mA et 5V.
- Il existe des composants qui ajoutent facilement une interface USB à tout **microcontrôleur**, typiquement en « traduisant » 8 GPIO parallèles en un port USB.



Types de connecteurs USB, de gauche à droite: micro USB, mini AB, type B, type A femelle et type A male



I2C

- I²C (pour *Inter Integrated Circuit Bus*) est le nom du bus historique, développé par Philips pour les applications de domotique et d'électronique domestique au début des années 1980, notamment pour permettre de relier facilement à un microprocesseur les différents circuits d'une télévision moderne. Ce bus porte parfois le nom de TWI (Two Wire Interface) chez certains constructeurs.
- I²C est un bus qui a émergé de la « guerre des standards » lancée par tous les acteurs du monde électronique. Ainsi, dans votre téléviseur, tous les ensembles sont sur un bus I2C (récepteur de la télécommande, réglages des amplificateurs basses fréquences, tuner, horloge, gestion de la prise péritel, etc.).
- Il existe d'innombrables périphériques exploitant ce bus, il est même implantable **par logiciel** dans n'importe quel microcontrôleur. Le poids de l'industrie de l'électronique grand public a permis des prix très bas aux nombreux composants.
- L'I²C convient tout particulièrement à toutes les applications où la vitesse n'est pas primordiale

I2C

Un bus I²C contient trois fils :

- un signal de donnée (SDA) ;
- un signal d'horloge (SCL) ;
- un signal de référence (masse).

Le périphérique qui gère la communication est le maître, c'est lui qui génère l'horloge (SCL) et qui envoie les données (SDA) mis à part l'*acknowledge* (acquiescement en français).

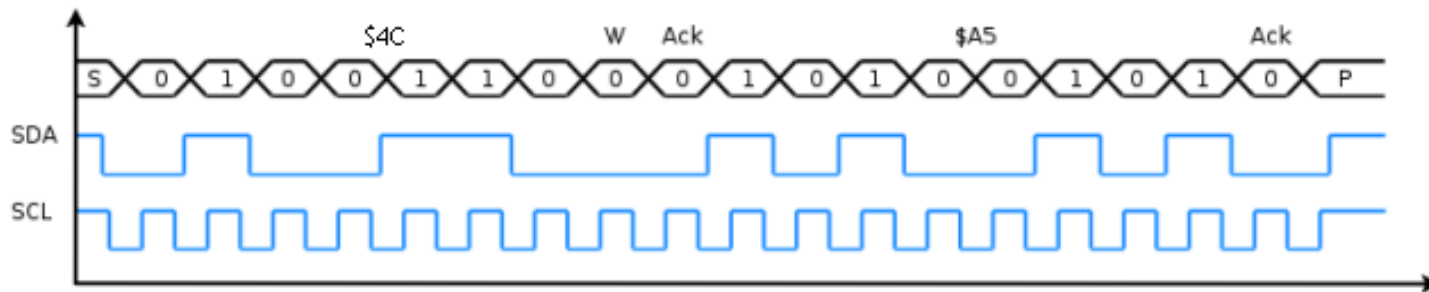
L'*acknowledge* est un bit envoyé par le composant esclave pour indiquer qu'il a bien reçu toutes les données; si c'est le cas l'esclave impose le niveau 0, sinon la résistance de pull-up maintient la ligne à 1, on dit alors qu'il n'y a pas d'*acknowledge*.

Plusieurs composants peuvent être branchés sur le même bus I2C. Pour que l'information aille au bon endroit chaque composant possède **sa propre adresse**.

Elle est composée d'une partie fixe imposée par le constructeur, d'une partie configurable de façon matérielle par l'utilisateur, et du bit de *read/write* qui définit le sens de la transmission (0 pour écriture, 1 pour lecture).

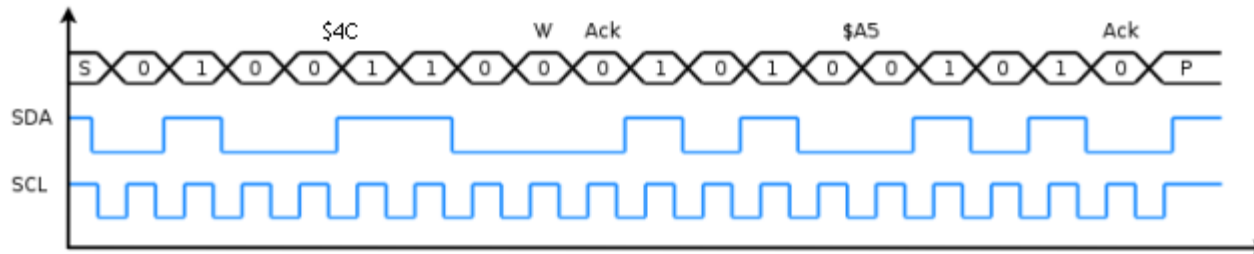
I2C

Exemple de communication I²C:



La communication commence par le StartBit
puis l'adresse, (sur 8bits \$4C) avec bit de read/write à 0
l'acknowledge (Ack)
un octet de données (\$A5)
de nouveau l'acknowledge (Ack)
et enfin le StopBit.

I2C



Au début de la communication SDA passe à 0 alors que SCL reste à 1, c'est le StartBit.

Après avoir imposé la condition de départ, le maître applique sur SDA le bit de poids fort.

Il latch (valide) la donnée en appliquant pendant un instant un niveau #1 sur la ligne SCL.

Lorsque SCL revient à #0, il recommence l'opération avec le bit inférieur jusqu'à ce que l'octet complet soit transmis.

Il redéfinit ensuite SDA comme une entrée et scrute son état ; l'esclave doit alors imposer un niveau #0 pour signaler au maître que la transmission s'est effectuée correctement, c'est l'*acknowledge*, la communication peut donc continuer.

Si l'esclave n'envoie pas l'*acknowledge* les résistances de pull-up maintiennent la ligne à #1.

La communication peut alors être arrêtée, ou reprendre à zéro (dépend de la configuration).

Le premier octet envoyé est l'adresse, il est composé de 7bits variable selon le composant et du bit de *read/write* (0 pour *write*, 1 pour *read*).

Le second octet peut être le *byte* de contrôle sur certains composants, ou directement la donnée.

TRAVAIL PERSONNEL

- Lire et assimiler les chapitres fournis en annexe:
 - Interface série asynchrone
 - Bus I2C

- Articles Wikipédia:
 - USB
 - Duplex (canal de communication)