

Bus I2C

Le bus I2C, dont le sigle signifie *Inter Integrated Circuit* ce qui donne *IIC* et par contraction I2C, a été proposé initialement par Philips mais est adopté de nos jours par de très nombreux fabricants.

C'est un bus de communication de type série.

Présentation

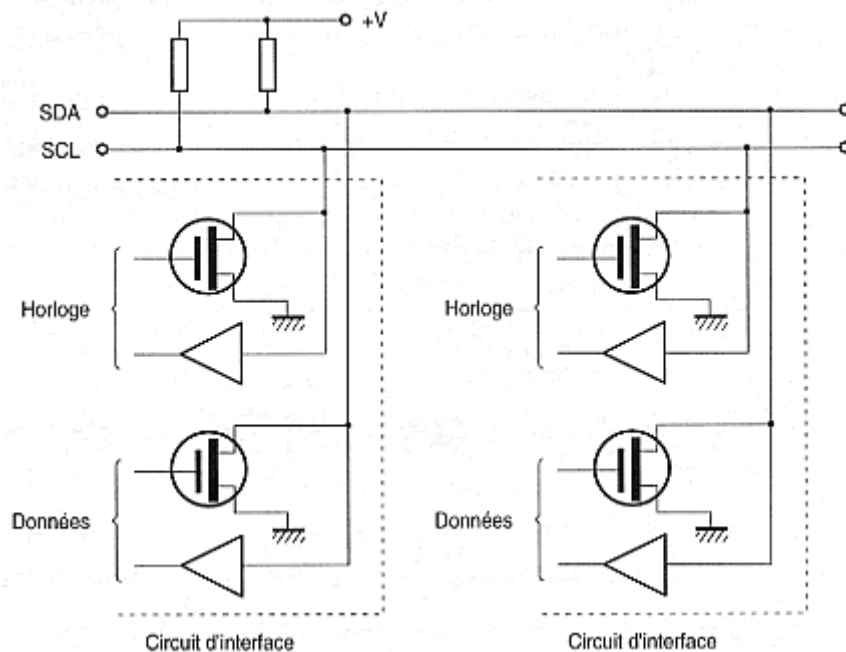
Le bus I2C qui n'utilise que deux lignes de signal (et les masses correspondantes bien sûr) permet à un certain nombre d'appareils d'échanger des informations sous forme série avec un débit pouvant atteindre 100 K bits par seconde ou 400 K bits par seconde pour les versions les plus récentes. Même si ces débits peuvent sembler relativement faibles, les premières applications du bus I2C sont des applications audio ou vidéo pour lesquelles la simplicité de mise en oeuvre est nettement plus importante qu'un débit élevé.

Ceci étant précisé, voici quels sont les points forts du bus I2C :

- c'est un bus série bifilaire utilisant une ligne de données appelée SDA (Serial DATA) et une ligne d'horloge appelée SCL (Serial CLock);
- les données peuvent être échangées dans les deux sens sans restriction ;
- le bus est multi-maîtres ;
- chaque abonné dispose d'une adresse codée sur 7 bits. On peut donc connecter simultanément 128 abonnés d'adresses différentes sur le même bus, sous réserve de ne pas le surcharger électriquement bien sûr;
- un acquittement est généré pour chaque octet de donnée transféré ;
- le bus peut travailler à une vitesse maximum de 100 K bits par seconde (ou 400 K bits par seconde) étant entendu que son protocole permet de ralentir automatiquement l'équipement le plus rapide pour s'adapter à la vitesse de l'élément le plus lent, lors d'un transfert ;
- le nombre maximum d'abonnés n'est limité que par la charge capacitive maximale du bus qui peut être de 400 pF. Ce nombre ne dépend donc que de la technologie des circuits et du mode câblage employés ;
- les niveaux électriques permettent l'utilisation de circuits en technologies CMOS, NMOS ou TTL.

Tout ceci étant précisé, voyons maintenant comment fonctionne réellement ce bus car, à la lecture de ce qui précède vous devez commencer à vous poser bien des questions.

Principe d'un échange de données



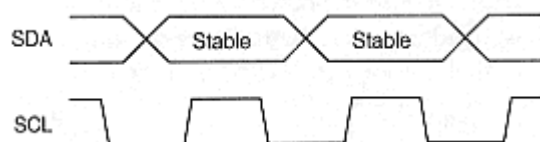
Cette figure montre le principe adopté au niveau des étages d'entrée/sortie des circuits d'interface au bus I2C.

Si la partie entrée n'appelle aucune remarque particulière, on constate que la partie sortie fait appel à une configuration à drain ouvert (l'équivalent en MOS du classique collecteur ouvert) ce qui permet de réaliser des ET câblés par simple connexion, sur la ligne SDA ou SCL, des sorties de tous les circuits.

Aucune charge n'étant prévue dans ces derniers, une résistance de rappel à une tension positive doit être mise en place. Le niveau électrique n'est pas précisé pour l'instant car il dépend de cette tension.

Nous parlerons donc de niveaux logiques hauts ou « 1 » ou bien encore de niveaux logiques bas ou « 0 » étant entendu que l'on travaille en logique positive c'est à dire qu'un niveau haut correspond à une tension plus élevée qu'un niveau bas.

Compte tenu de ce mode de connexion en ET câblé, lorsqu'aucun abonné n'émet sur le bus, les lignes SDA et SCL sont au niveau haut qui est leur état de repos.

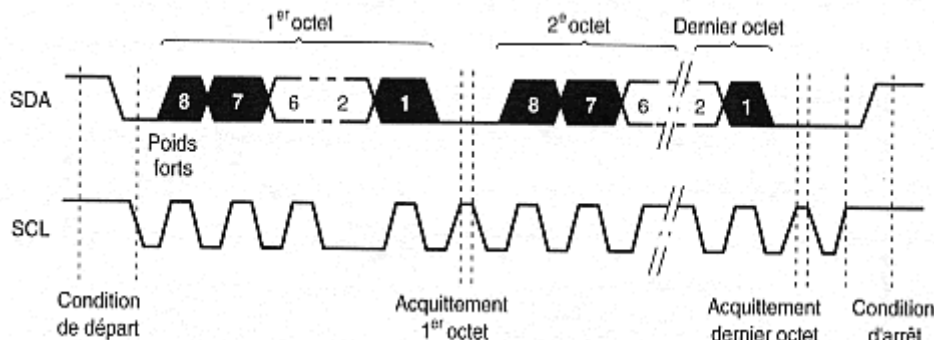


Cette figure résume le principe fondamental d'un transfert à savoir : une donnée n'est considérée comme valide sur le bus que lorsque le signal SCL est à l'état haut.

L'émetteur doit donc positionner la donnée à émettre lorsque SCL est à l'état bas et la maintenir tant que SCL reste à l'état haut (nous verrons des chronogrammes plus précis dans un instant).

Comme la transmission s'effectue sous forme série, une information de début et de fin doit être prévue. L'information de début s'appelle ici condition de départ et l'information de fin condition d'arrêt.

Une condition de départ est réalisée lorsque la ligne SDA passe du niveau haut au niveau bas alors que SCL est au niveau haut. Réciproquement, une condition d'arrêt est réalisée lorsque SDA passe du niveau bas au niveau haut alors que SCL est au niveau haut.



Bien que nous soyons en présence d'un bus série, les données sont envoyées par paquets de huit, même si un octet regroupe en fait huit bits indépendants.

Le bit de poids fort est envoyé le premier.

Chaque octet est suivi par un bit d'acquittement de la part du destinataire et l'ensemble du processus fonctionne comme indiqué sur la figure.

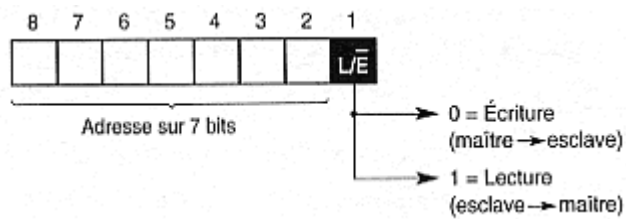
Tout d'abord, sachez que lors d'un échange de ce type, la ligne SCL est pilotée par l'initiateur de l'échange ou maître, quitte à ce que l'esclave agisse également dessus dans certains cas particuliers.

La figure ci-dessus montre tout d'abord une condition de départ, générée par le maître du bus à cet instant. Elle est suivie par le premier octet de données, poids forts en tête. Après le huitième bit, l'émetteur qui est aussi le maître dans ce cas, met sa ligne SDA au niveau haut c'est à dire au repos mais continue à générer l'horloge sur SCL. Pour acquitter l'octet, le récepteur doit alors forcer la ligne SDA au niveau bas pendant l'état haut de SCL qui correspond à cet acquittement, prenant en quelque sorte la place d'un neuvième bit.

Le processus peut alors continuer avec l'octet suivant et se répéter autant de fois que nécessaire pour réaliser un échange d'informations complet. Lorsque cet échange est terminé, le maître génère une condition d'arrêt.

Formats de transmission

Nous savons maintenant comment se déroulent les échanges ; il nous reste à examiner le format des données transmises afin de comprendre comment fonctionne l'adressage, mais aussi la définition du sens de transferts des données.



Cette figure montre le contenu du premier octet qui est toujours présent en début d'échange.

Ses sept bits de poids forts contiennent l'adresse du destinataire du message ce qui autorise 128 combinaisons différentes.

Le bit de poids faible indique si le maître va réaliser une lecture ou une écriture. Si ce bit est à zéro le maître va écrire dans l'esclave ou lui envoyer des données. S'il est à un, le maître va lire dans l'esclave c'est à dire que le maître va recevoir des données de l'esclave.

Lorsqu'un maître désire effectuer plusieurs échanges à destination d'esclaves d'adresses différentes, il n'est pas obligé de terminer le premier échange par une condition d'arrêt mais peut les enchaîner en générant une condition de départ dès la fin d'un échange.

Enfin, il existe une procédure dite d'appel général où l'adresse envoyée par le maître, c'est à dire rappelons-le, les sept bits de poids forts du premier octet, est nulle. Tous les circuits connectés sur le bus capables de répondre à un tel appel général doivent alors le faire et prendre en compte les données qui suivent. Leur attitude dépend du bit de lecture/écriture de ce premier octet.

En effet, si ce bit est à zéro, le deuxième octet revêt une signification particulière, que je ne détaillerais pas ici, mais qui permet, au niveau de chaque esclave, la programmation de son adresse par logiciel ou matériel.

Lorsque ce bit est à un, on est en présence d'un appel général matériel qui est transmis par un maître du bus matériel, tel qu'un circuit d'interface de clavier par exemple. Un tel circuit est incapable de générer l'adresse du destinataire de l'information qu'il a à envoyer.

Dans ces conditions, l'octet suivant contient l'adresse de ce maître matériel afin qu'un esclave « intelligent », généralement un microcontrôleur, le reconnaisse et prenne en compte l'information qu'il émet ensuite.

Texte et images tirés de

<http://www.tavernier-c.com/busi2c.htm>