

A) Présentation du bus I2C

Le bus I²C a été créé au début des années 80 par RTC Philips afin d'apporter une solution simple et peu coûteuse à la communication entre les circuits intégrés numériques à l'intérieur des appareils grand public (téléviseurs, magnétoscopes, etc.). Le principal avantage du bus I²C est de limiter le nombre de liaisons entre circuits intégrés.



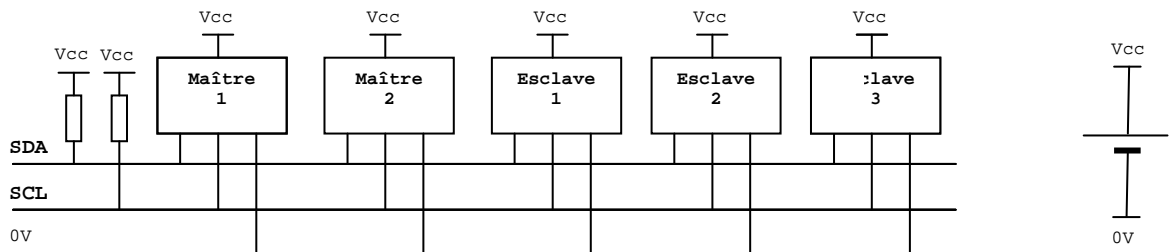
1- Structure et organisation du bus I2C

Le bus I²C est un bus de type série synchrone ne nécessitant que deux signaux.

- SDA (serial data), le signal de donnée.
- SCL (serial clock), le signal d'horloge.

Ce bus permet la communication entre un circuit maître et un circuit esclave. Le montage peut comporter plusieurs maîtres et plusieurs esclaves. **Le maître est le circuit qui émet le signal d'horloge** de synchronisation, un seul maître peut envoyer ce signal. Les données peuvent circuler dans les deux sens sur le fil des données, de sorte que chaque circuit, qu'il soit maître ou esclave peut servir d'émetteur ou de récepteur (de données).

Les différents circuits sont placés en parallèle sur les lignes SDA et SCL comme sur le schéma suivant :

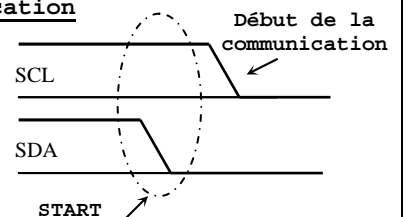


Au repos, c'est à dire lorsqu'aucun circuit n'émet, les signaux SDA et SCL sont au niveau logique haut. Pour éviter les conflits, un maître qui veut émettre doit attendre que le bus soit au repos.

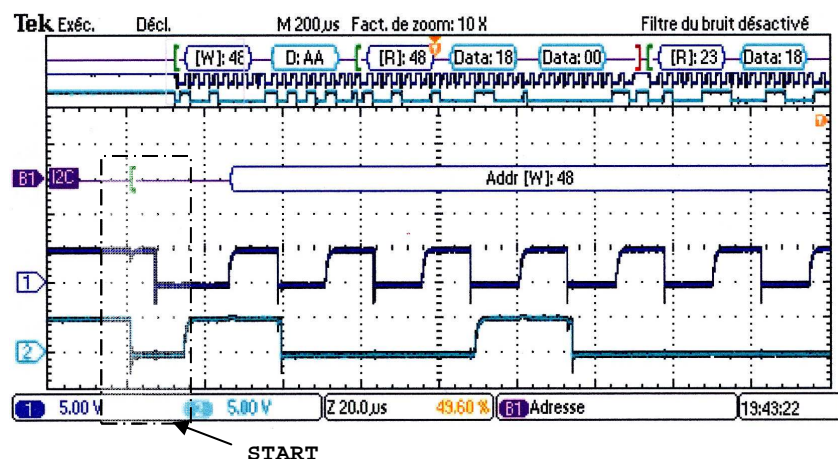
2- Le protocole I2C

2.1) Prise de contrôle du bus par un maître et début de la communication

Le bus étant initialement libre, SDA et SCL sont à 1. Un maître prend le contrôle du bus en effectuant un START : il met SDA à 0, SCL restant à 1. Au cours de la communication, l'horloge SCL est envoyée par le maître et SDA ne peut changer d'état que lorsque SCL est à 0.



Exemple de mesure : START (Oscilloscope TEKTRONIX MSO2014)



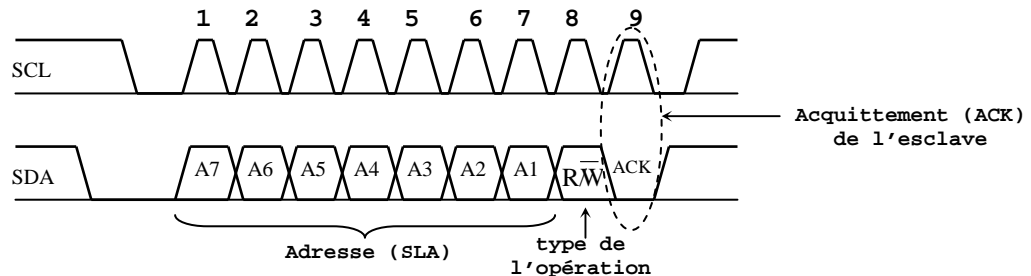
1 : SCL

2 : SDA

2.2) Le maître choisit l'esclave et le type d'opération par un octet de contrôle (SLA+R/W), l'esclave répond !

Pour sélectionner un esclave, lorsqu'il a pris le contrôle du bus, le maître envoie un **premier octet (octet de contrôle noté SLA+R/W)**. Cet octet est constitué :

- d'une adresse (SLA) sur sept bits,
- d'un bit de lecture (Read)/écriture (Write) (R/W) qui indique si le maître souhaite faire une opération de lecture ou d'écriture. Simultanément, le maître génère le signal d'horloge SCL comportant **neuf périodes**.



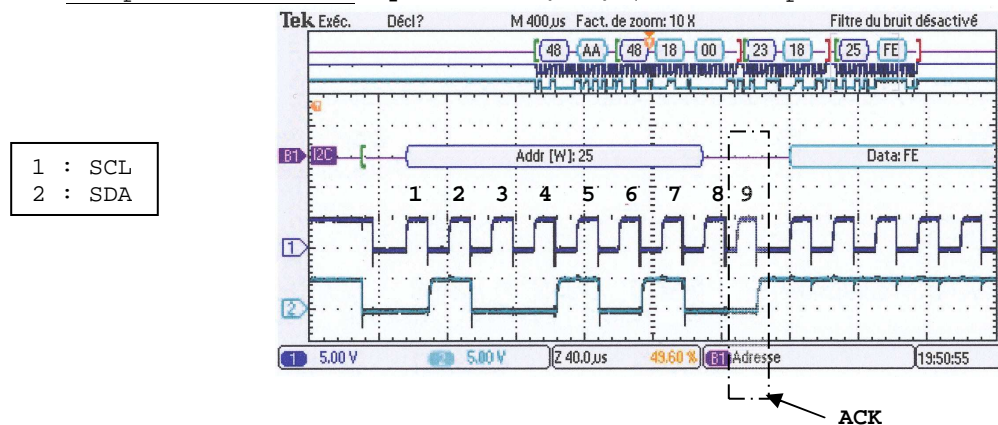
Tous les esclaves observent l'adresse envoyée par le maître. Pendant la **neuvième période** de SCL, l'esclave qui reconnaît son adresse répond au maître en maintenant SDA au niveau logique bas : c'est le bit d'acknowledgment appelé **ACK** (acknowledge). L'esclave est prêt à communiquer avec le maître. Les autres esclaves restent au repos.

Si le bit $\overline{R/W}$ envoyé par le maître est à :

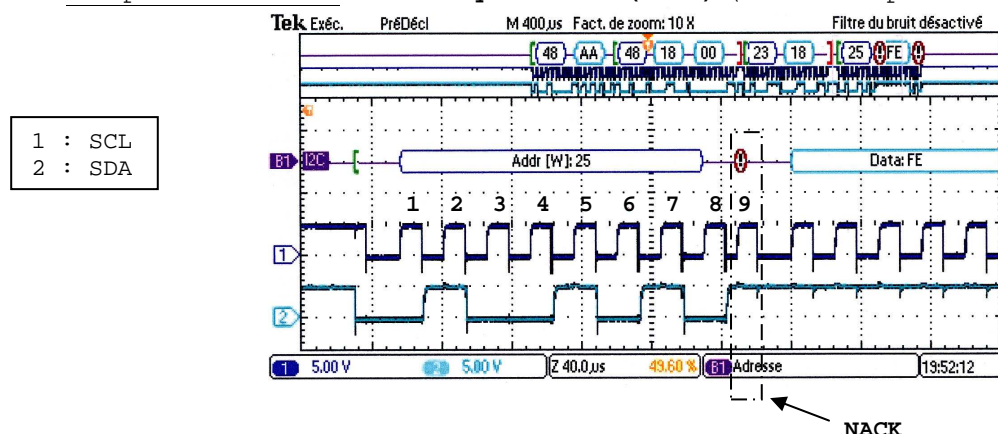
- 1 (lecture), l'esclave prend la parole.
- 0 (écriture), l'esclave se met à l'écoute du maître.

Dans les exemples ci-dessous, l'esclave est adressé avec une SLA = 25₍₁₆₎. Dans le premier chronogramme, il acquitte (ACK) cette adresse car il a été correctement configuré. Dans le second, il est configuré avec une adresse différente de 25₍₁₆₎ : il n'acquitte pas (NACK) !

Exemple de mesure : **Acquittement (ACK)** (Oscilloscope TEKTRONIX MSO2014)



Exemple de mesure : **Pas d'acknowledgment (NACK)** (Oscilloscope TEKTRONIX MSO2014)

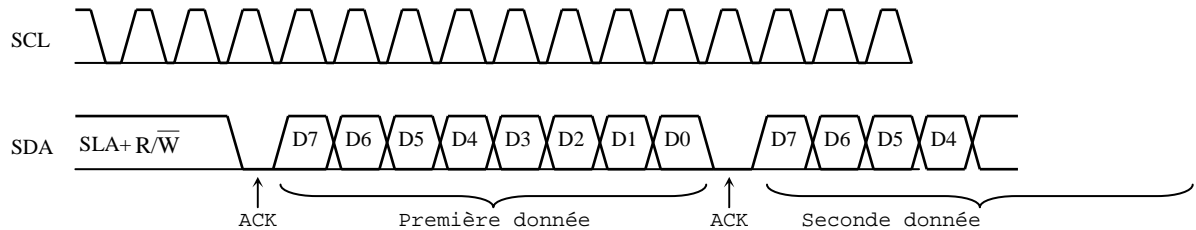


2.3) Envoi d'une ou plusieurs données

A la suite de l'octet de contrôle, chaque donnée est codée sur un octet envoyé bit par bit sur la ligne SDA. Au cours de la communication, plusieurs données peuvent être transmises.

- s'il s'agit d'une opération de **lecture**, les données sont envoyées au maître par l'esclave. **L'esclave est émetteur, le maître est récepteur.**
- s'il s'agit d'une opération d'**écriture**, les données sont envoyées à l'esclave par le maître. **Le maître est émetteur, l'esclave est récepteur.**

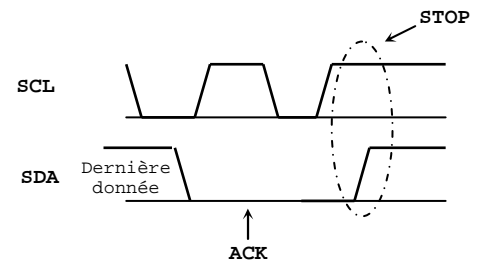
L'émetteur envoie d'abord les huit bits du premier octet puis le récepteur acquitte en maintenant la ligne SDA à 0 (bit ACK). L'émetteur envoie ensuite les huit bits du second octet et le récepteur acquitte. Le même cycle se répète jusqu'à ce que toutes les données aient été envoyées. Pendant la communication, l'horloge SCL est générée par le maître.



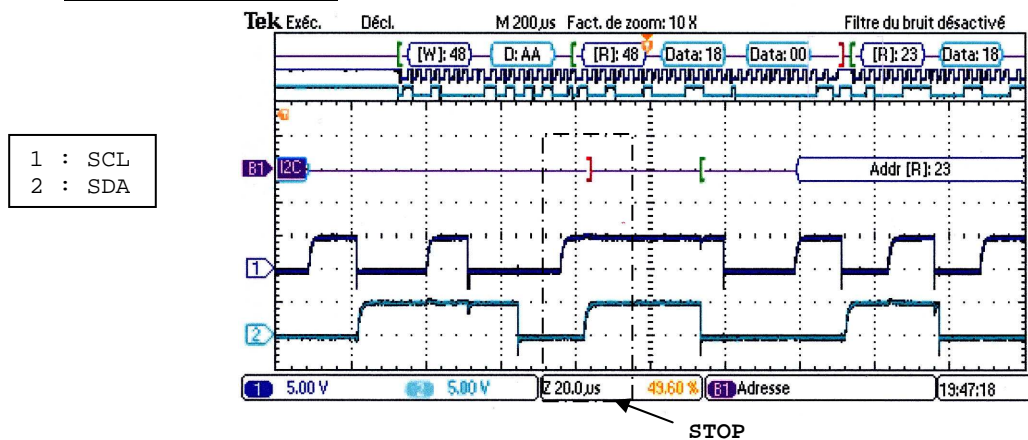
2.4 Fin de communication et libération du bus

Pendant la communication, les changements d'état sur SDA se produisent lorsque SCL est à 0.

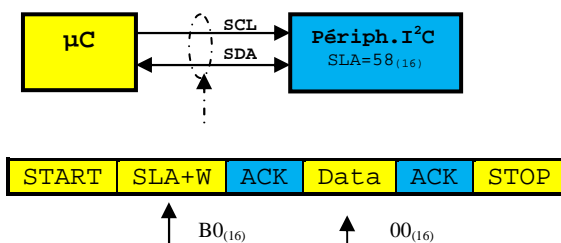
Pour mettre fin à la communication, le maître effectue un **STOP** : il met d'abord SCL à 1 puis ramène SDA à 1. C'est le changement d'état de SDA alors que SCL est à 1 qui met fin à la communication.



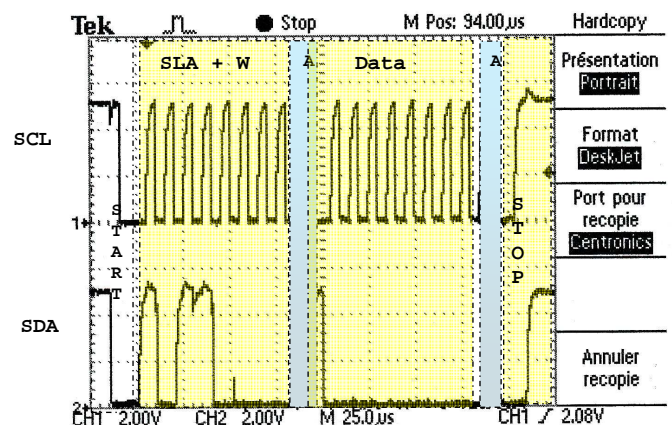
Exemple de mesure : **STOP** (Oscilloscope TEKTRONIX MSO2014)



2.5 Exemple de transaction effectuée entre un microcontrôleur et un périphérique I²C



En envoyant la trame ci-contre, le microcontrôleur écrit la valeur 0 dans un périphérique situé à l'adresse **SLA = 58₍₁₆₎** sur le bus.



2.5 Synchronisation du maître et de l'esclave

Habituellement, la ligne SCL fonctionne en sortie pour un maître et en entrée pour un esclave car c'est le maître qui génère le signal d'horloge. Mais lorsqu'un **esclave veut obliger un maître à ralentir la communication, l'esclave maintient la ligne SCL à 0** pour empêcher le maître de la ramener à 1. La communication reprend son cours lorsque l'esclave libère la ligne SCL. Cette opération s'appelle la **synchronisation**.

2.6 Arbitrage en cas de conflit

Un maître ne peut pas prendre le contrôle du bus si celui-ci est déjà occupé. Il peut arriver que deux maîtres cherchent à prendre le contrôle du bus en même temps, à quelques nanosecondes près. Dans ce cas l'arbitrage se fait sur la ligne SDA. Le premier des deux maîtres qui veut établir la communication doit vérifier que SDA=1 (ligne libre), si cette dernière est à 0, cela signifie que la ligne est occupée, il doit cesser d'émettre et se placer en récepteur au cas où la donnée envoyée par l'autre maître lui serait destinée.

3- Caractéristiques du bus I²C (mode standard)

- Bus série synchrone bi directionnel 8 bits (deux lignes SDA et SCL)
- Bus multi-maître, détection de collision et arbitrage
- Débit : 100kbits/s

4- Exemples de circuits intégrés pour le bus I²C

Le maître est souvent un **microcontrôleur possédant une interface I²C**.

Les esclaves sont des circuits intégrant une interface pour le bus I²C et réalisant la même fonction que des circuits intégrés couramment utilisés en électronique :

- CAN ou CNA : PCF8591
- mémoire vive : PCF8570
- EEPROM : PCF8582
- horloge calendrier : PCF8583
- Port d'entrées / sorties parallèles (8) : PCF8574
- Afficheurs LCD
- ...

5- Bibliographie

"Le bus I2C: Principes et mise en oeuvre" D. PARET DUNOD

6 - Webographie

<http://fr.wikipedia.org/wiki/I2C>

http://artic.ac-besancon.fr/genie_electronique/telech/sonde/doc_bus_i2c.pdf

Remerciements

Ce document a été rédigé en partant du travail de **Jean-Luc Séguret**.
Merci à lui.