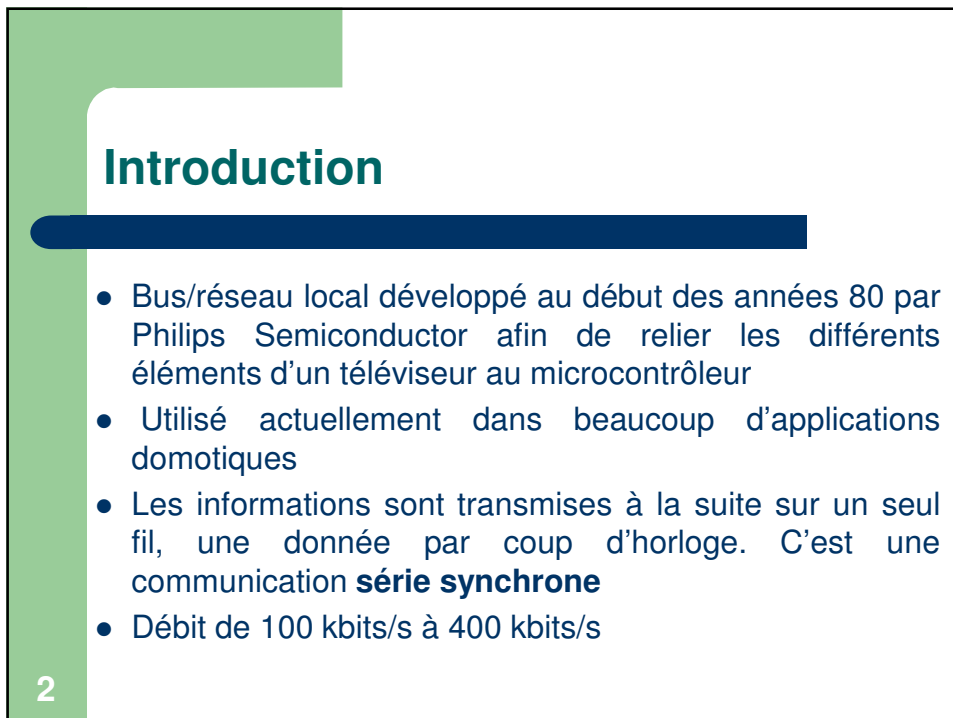


1

LE BUS I2C

Inter Integrated Circuit

A slide with a green background on the left side. The text 'LE BUS I2C' is in large blue letters, and 'Inter Integrated Circuit' is in smaller green letters below it. A dark blue horizontal bar is positioned below the green text. The number '1' is in the bottom left corner.



2

Introduction

- Bus/réseau local développé au début des années 80 par Philips Semiconductor afin de relier les différents éléments d'un téléviseur au microcontrôleur
- Utilisé actuellement dans beaucoup d'applications domotiques
- Les informations sont transmises à la suite sur un seul fil, une donnée par coup d'horloge. C'est une communication **serie synchrone**
- Débit de 100 kbits/s à 400 kbits/s

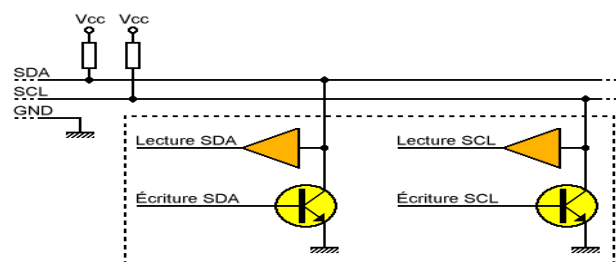
A slide with a green background on the left side. The title 'Introduction' is in green. Below it is a dark blue horizontal bar. A bulleted list follows. The number '2' is in the bottom left corner.

Caractéristiques technologiques

- Le but est de faire communiquer entre eux des composants électroniques grâce à 3 fils:
 - **Signal de données: SDA serial data**
 - **Signal d'horloge: SCL serial clock**
 - **Signal de référence: masse GND**
- Chaque abonné dispose d'une adresse codée sur 7 bits. On peut donc connecter simultanément 128 abonnés.

3

Le support physique utilisé



- Les transistors sont des bipolaires ou des MOS

4

Le support physique utilisé

- Au repos, SCL et SDA sont à 1
- Sorties à collecteur ouvert (ou à drain ouvert pour des circuits CMOS)
- Si une ou plusieurs sorties tentent d'imposer un niveau bas sur une ligne, le ou les transistors associés vont conduire, ce qui entraîne un niveau bas sur la ligne correspondante (ce qui est conforme au résultat de la fonction « ET »)

5

Le protocole I²C

- **Maître:** Circuit qui **prend l'initiative** d'un transfert d'information, qui **génère le signal d'horloge** et qui **termine** le transfert.
- **Esclave:** Circuit qui participe au transfert sur demande du circuit maître.
- Attention: ne pas confondre circuit-maître et circuit-esclave avec circuit-émetteur et circuit-récepteur.

6

Le protocole I²C

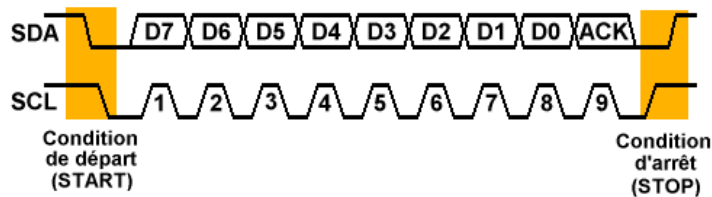
- Pour transmettre des données sur le bus I²C, il faut surveiller deux conditions particulières : la condition de **départ** et la condition **d'arrêt**.
- Condition de départ: SDA passe à 0, SCL reste à 1
- Condition d'arrêt: SDA passe à 1, SCL reste à 1



7

Transmission d'un octet

- A chaque impulsion sur SCL, un bit est transmis correspondant au niveau de SDA
- Cette opération est répétée pour les 8 bits
- Le bit de poids fort est transmis en premier



8

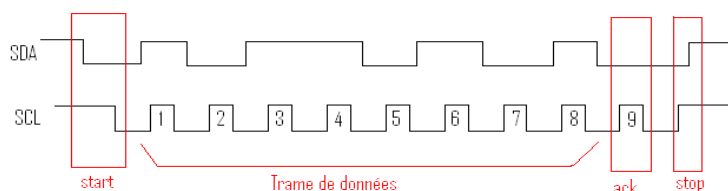
Acquittement

- Une fois les 8 bits transmis, le récepteur acquitte la transmission au 9^{ème} bit
- L'émetteur met sa ligne SDA au niveau haut
- Le récepteur met sa ligne au niveau bas
- Les sorties étant à collecteur ouvert, le récepteur l'emporte, SDA sera au niveau bas
- Au coup d'horloge suivant, 2 possibilités:
 - le maître voit un niveau bas: octet bien reçu
 - Le maître voit un niveau haut: erreur de transmission donc arrêt

9

Exemple de trame

- Envoi de l'octet (B5)h

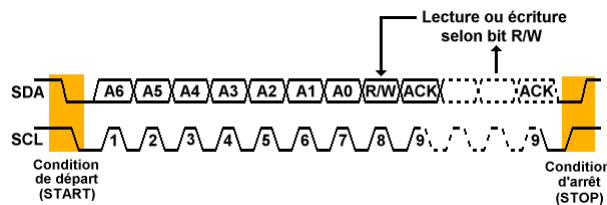


*Le signal SDA doit être stable aussi longtemps que le signal SCL est actif.
Un changement d'état de la ligne SDA pendant que le signal SCL est au niveau haut implique une condition de départ ou d'arrêt*

10

Transmission d'une adresse

- Le premier octet transmis est une adresse correspondant à l'esclave choisi
- Le bit A0 de l'adresse est réservé:
 - A0=1 le maître demande une lecture à l'esclave
 - A0=0 le maître demande une écriture à l'esclave



11

Écriture d'une donnée

- Si le bit R/W de l'adresse est à 0, cela signifie que le maître doit transmettre un ou plusieurs octets de données.
- Après chaque bit ACK valide, le maître peut soit:
 - Continuer d'envoyer des octets à l'esclave
 - Décider de terminer le dialogue par une condition d'arrêt.



- état imposé par le maître
- état imposé par l'esclave

12

Lecture d'une donnée (1)

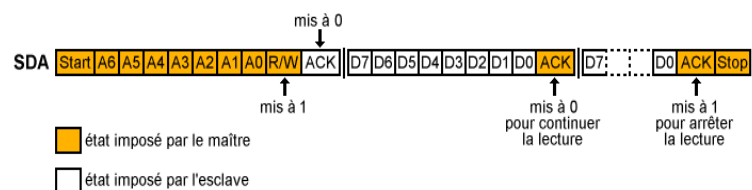
- Si le bit R/W de l'adresse est à 1, cela signifie que le maître veut lire des données issues de l'esclave
- C'est toujours le maître qui va générer le signal d'horloge SCL
- L'esclave prend le contrôle de la ligne SDA
- Après que l'esclave ait transmis les 8 bits de données, c'est le maître, cette fois-ci, qui va générer un bit d'acquittement

13

Lecture d'une donnée (2)

2 possibilités d'acquittement:

- Bit Ack au niveau bas: le maître désire lire des octets supplémentaires
- Bit Ack au niveau haut: le maître décide que la lecture est terminée. Cette fois-ci, bien que le bit ACK soit au niveau 1, cela ne correspond pas à une condition d'erreur mais à une fin de transfert.



14

Restart

- Il est possible d'enchaîner écriture et lecture de l'esclave sans avoir à passer par une condition arrêt
- La commande **repeat start** permet de lire une donnée après écriture sans générer de **stop** entre les 2.

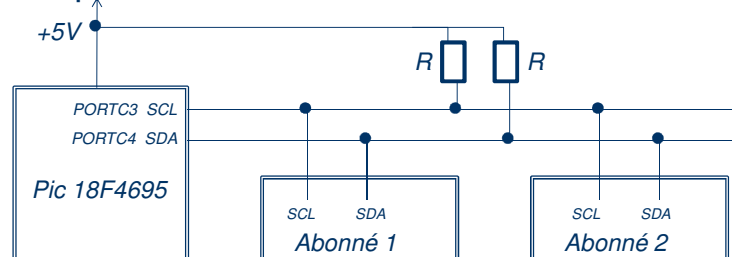
15

APPLICATION DU BUS I2C AU PIC18F4685

16

Présentation et Connexions

- Le PIC18F4685 possède une interface I2C intégrée
- Les 2 lignes SCL et SDA sont connectées respectivement à PORTC3 et PORTC4



17

Différents modes de fonctionnement

- L'interface I2C peut supporter les modes suivants:
 - Mode maitre
 - Mode multimaitre
 - Mode esclave
- Le pic peut donc transmettre ou recevoir des données en mode maitre ou esclave

18

Registres de contrôle

- 2 registres SSPCON1 et SSPCON2 permettent d'établir les transmissions
- Un registre d'état SSPSTAT permet de donner des informations sur la transmission
 - Le bit R/W est témoin d'une transmission en cours d'écriture
- Un buffer SSPBUF contient les adresses ou les données
- SSPADD
 - En mode esclave: définit l'adresse
 - En mode maitre: définit le débit
- Les transmissions consistent donc à régler ces différents registres (8 bits) dans l'ordre du protocole

19

Registre SSPCON2

GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
------	---------	-------	-------	------	-----	------	-----

- GCEN: mis à zéro par défaut, ne sert qu'en mode esclave
- ACKSTAT: acquittement reçu de l'esclave
 - 1 pas d'acquittement
 - 0 acquittement
- ACKDT: acquittement du maitre en mode réception
 - 1 pas d'acquittement
 - 0 acquittement
- ACKEN:
 - 1 mise en place de la séquence d'acquittement du maitre en mode réception. Remise à zéro automatiquement
 - 0 séquence d'acquittement terminée

20

Registre SSPCON2 (suite)

GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
------	---------	-------	-------	------	-----	------	-----

- RCEN:
 - 1 commence une réception de la part du maître
 - 0 réception terminée, l'octet est reçu
- RSEN:
 - 1 commence une séquence de restart
 - 0 restart terminé
- PEN:
 - 1 commence une séquence de stop
 - 0 séquence de stop terminée
- SEN:
 - 1 commence une séquence de start
 - 0 séquence de start terminée

21

Conflits

- Le bus I2C doit pouvoir accueillir plusieurs maîtres
- Il est possible que 2 maîtres prennent la parole en même temps
- Pas de problèmes électriques (collecteur ouvert)
- Problème logique, il faut éviter la corruption des données

22

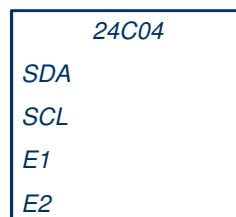
Prise de contrôle du bus

- Le maître relit à chaque fois le bit qu'il cherche à imposer sur le bus
 - S'il impose un 0, il relira obligatoirement un 0
 - S'il impose un 1, 2 cas:
 - Il relit un 1 => il continue à transmettre
 - Il relit un 0 => un autre maître a pris la parole, il **perd** le contrôle
- Si différents maîtres envoient les mêmes données en même temps (cas rare), aucun conflit

23

Exemple: circuit 24C04

- Le circuit 24C04 est une EEPROM série constituée de deux blocs de 256 Ko utilisant le protocole I2C



24

Éléments constitutifs de la trame

- **Octet d'identification (devsel) :**

- C'est le premier octet émis par le maître, il permet de sélectionner le circuit ainsi que la nature et le lieu des opérations à effectuer (R/W, bloc).
- bits 7 à 2 : Identification du circuit constituée d'une partie fixe (bits 7 à 4) égale à 1010, et d'une partie variable (bits 3 et 2) fixée par les états logiques imposés sur les broches E2 et E1.
- bit 1 : Permet l'accès à un des deux blocs de 256 Ko.
- bit 0 : Sélectionne le mode lecture (1) ou écriture (0).

25

Éléments constitutifs de la trame (suite)

- **Exemple :**

- On désire écrire dans le bloc 0 d'un circuit 24C04 dont les pattes E2 et E1 sont connectées respectivement à 1 et à 0. Le premier mot à envoyer sera 10101000.

- **Octet d'adresse (addr) :**

- Ce mot indique l'adresse à laquelle doivent s'effectuer les opérations.

- **Octet de donnée(data):**

- Ce mot indique la donnée à inscrire.

26

Algorithme d'écriture (mode direct)

- La donnée est rangée à l'adresse spécifiée par l'octet addr, la trame envoyée par le circuit maître est la suivante :

start	devsel (R/W = 0)	ack	addr	ack	data	ack	stop
-------	------------------	-----	------	-----	------	-----	------

27

Algorithme d'écriture (mode séquentiel)

- Une fois l'adresse spécifiée, les données sont inscrites aux adresses suivantes tant que la condition de stop n'est pas envoyée

start	devsel (R/W = 0)	ack	addr	ack	data1	ack
-------	------------------	-----	------	-----	-------	-----

data2	ack	data3	ack	...	stop
-------	-----	-------	-----	-----	------

28

Algorithme de lecture (mode direct)

- Pour effectuer une lecture il faut préalablement envoyer l'adresse au circuit en mode écriture. Puis repositionner le circuit en mode lecture afin d'obtenir la (les) donnée(s).



29

Algorithme de lecture (mode séquentiel)

- En mode séquentiel, après réception d'une donnée, le bit ack est positionné à 0 par le maître lorsqu'il désire une nouvelle donnée.



30

Exercice 1

- Soit un circuit 24C04 dont les broches E2 et E1 sont reliées à la masse
- On désire écrire la donnée 0xF9 à l'adresse 00 du bloc 0 en mode direct. Donner la trame.
- On désire écrire les données 0xF9, 0xE0, 0x12 respectivement aux adresses 0x10, 0x11, 0x12 du bloc 1 en mode séquentiel. Donner la trame.

31

Exercice 2

- Soit un circuit 24C04 dont les broches E2 et E1 sont reliées à la masse
- On désire lire la donnée présente à l'adresse 00 du bloc 0 en mode direct. Donner la trame.
- On désire lire les données présentes aux adresses 0x10 et 0x11 du bloc 1 en mode séquentiel. Donner la trame.

32

Sous programmes

- I2c_start
- I2c_restart
- I2c_stop
- I2c_write
- I2c_read

33