

BUS I2C

Table des matières

| | |
|---|---|
| 1. Structure et organisation du bus I2C..... | 2 |
| 2. Le protocole I2C..... | 2 |
| 2.1. Prise de contrôle du bus par un maître et début de la communication..... | 2 |
| 2.2. Choix de l'esclave et du type d'opération par le maître et réponse de l'esclave..... | 2 |
| 2.3. Envoi des données..... | 3 |
| 2.4. Fin de communication et libération du bus..... | 3 |
| 2.5. Synchronisation du maître et de l'esclave..... | 3 |
| 2.6. Arbitrage en cas de conflit..... | 4 |
| 3. Performances et limites d'emploi du bus I2C..... | 4 |
| 4. Exemples de circuits intégrés pour le bus I2C..... | 4 |
| 5. Présentation succincte de la barrière Decma-Park..... | 4 |
| 5.1. Utilisation du bus I2C dans le système Decma-Park..... | 5 |
| 5.2. Fonctionnement du port d'entrées / Sorties PCF8574..... | 6 |
| 5.3. Travail demandé..... | 7 |

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.



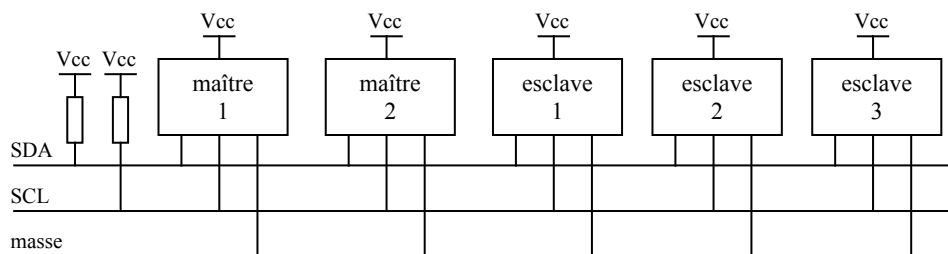
1. Structure et organisation du bus I2C

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 à la fois 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 :

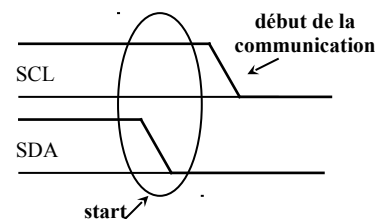


Au repos, c'est à dire lorsque aucun circuit n'émet, les signaux SDA et SCL sont au niveau logique haut. Pour éviter les conflits, un maître qui veut émettre doit attendre que le bus soit au repos.

2. Le protocole I2C

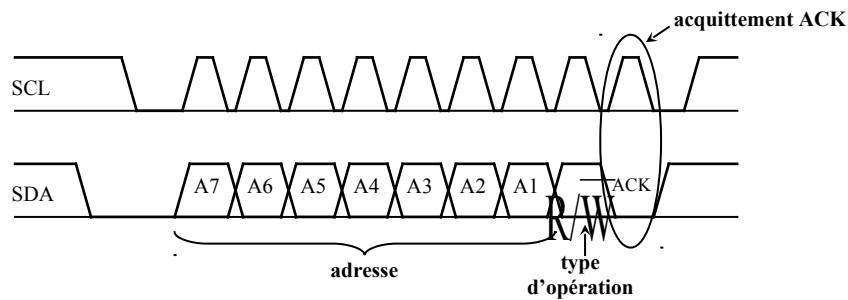
2.1. Prise de contrôle du bus par un maître et début de la communication

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.



2.2. Choix de l'esclave et du type d'opération par le maître et réponse de l'esclave

Après avoir pris le contrôle du bus, le maître envoie une adresse sur 7 bits pour sélectionner un esclave (bit de poids fort en premier). Le huitième bit, appelé R/\overline{W} , indique si le maître souhaite faire une opération de lecture ou d'écriture. Simultanément, le maître génère le signal d'horloge SCL comportant huit créneaux.



Tous les esclaves observent l'adresse envoyée par le maître. L'esclave qui reconnaît son adresse répond au maître en maintenant SDA au niveau logique bas : c'est le bit d'acquiescement appelé **ACK** (acknowledge). L'esclave est prêt à communiquer avec le maître. Les autres esclaves restent au repos. Si le bit R/\bar{W} envoyé par le maître est à :

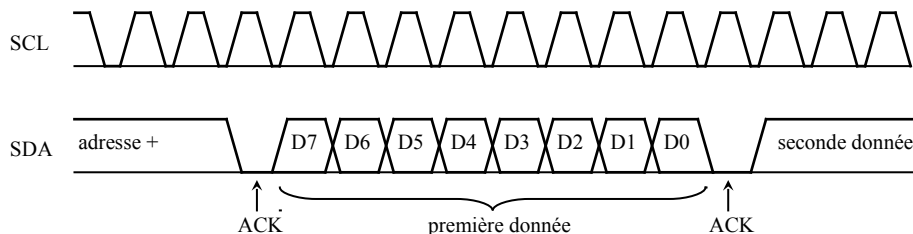
- 1 (lecture), l'esclave prend la parole.
- 0 (écriture), l'esclave se met à l'écoute du maître.

2.3. Envoi des données

Chaque donnée est formée d'un octet envoyé bit par bit sur la ligne SDA. Au cours de la communication, plusieurs octets peuvent être transmis.

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

L'émetteur envoie d'abord les 8 bits du premier octet puis le récepteur fait l'acquiescement en maintenant la ligne SDA à 0 (bit ACK). L'émetteur envoie ensuite les 8 bits du second octet et le récepteur fait l'acquiescement. Le même cycle se répète jusqu'à ce que toutes les données aient été envoyées. Dans les deux cas, 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.

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. Mais 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. Performances et limites d'emploi du bus I2C

Le bus I2C est réservé à de **courtes liaisons** : entre composants sur la même carte ou entre cartes par fils de quelques centimètres.

Les spécifications du bus I2C datant de 1992 imposent :

- l'adressage se fait sur 10 bits soit 1024 adresses possibles.
- la vitesse de transmission est limitée à 400 Kbits/seconde, soit 50 Koctets/seconde.

4. Exemples de circuits intégrés pour le bus I2C

Le maître est souvent un **microcontrôleur possédant une interface I2C**. En l'absence de microcontrôleur, ou si celui-ci n'est pas doté d'un interface I2C, on utilise une interface Bus parallèle / Bus I2C : le circuit intégré PCF8584. Cette interface est alors le maître.

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 :

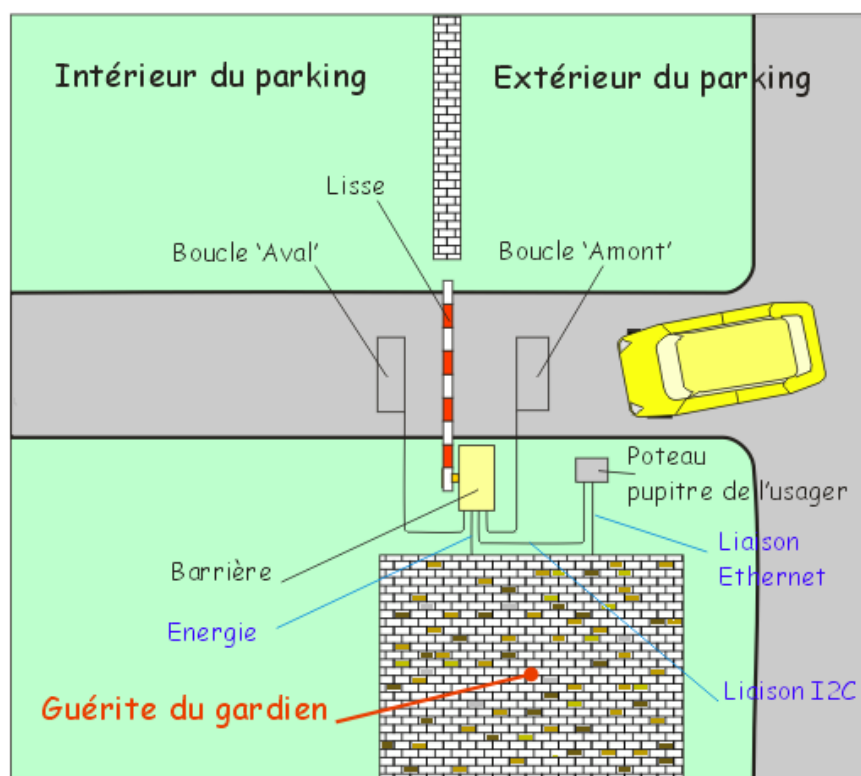
- CAN ou CNA : PCF8591
- mémoire vive : PCF8570
- EEPROM : PCF8582
- horloge calendrier : PCF8576
- Port d'entrées / sorties parallèles (8) : PCF8574
- Afficheurs LCD
-

5. Présentation succincte de la barrière Decma-Park

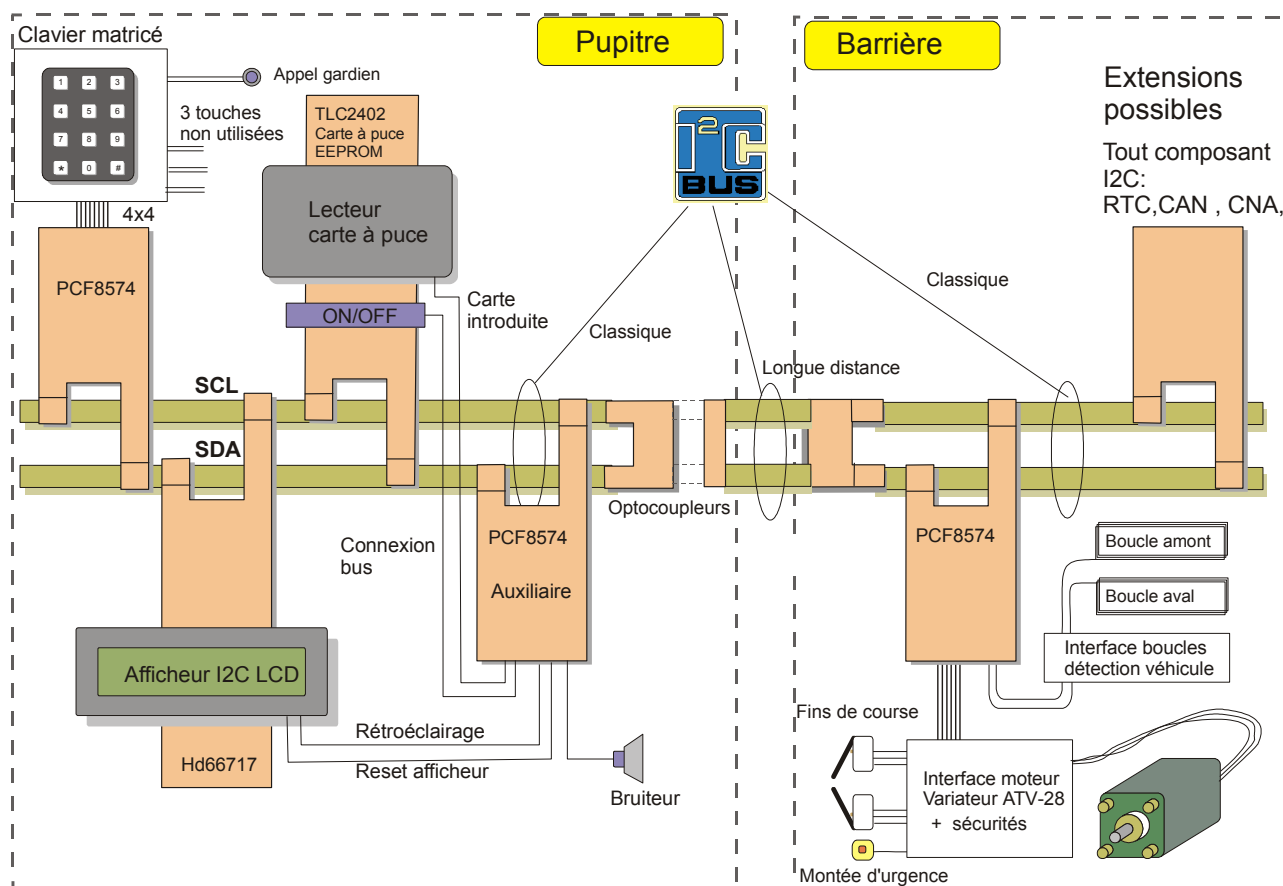
Le système Decma-Park est un système d'accès « conditionnel » à un parking.

Il se compose au minimum des éléments suivants :

- La barrière elle-même, c'est une armoire qui contient les éléments nécessaires à la mise en mouvement de la lisse.
- Le pupitre de l'utilisateur, comprenant un afficheur et un clavier ainsi que la carte ethernet « tini ».
- Un ordinateur de type PC implanté dans la guérite du gardien.



5.1. Utilisation du bus I2C dans le système Decma-Park



L'ensemble est architecturé autour du bus I2C. Le schéma ci-dessus montre l'utilisation intensive des circuits I2C.

Ceux-ci sont au nombre de trois sur la carte côté **pupitre** :

- Un circuit **PCF8574** gérant le **clavier** et l'appel gardien.
- Un **afficheur LCD** nativement I2C géré par un contrôleur HD66717
- Un circuit PCF8574 gérant les **auxiliaires** : détection de l'insertion d'une carte à puce, le couplage du bus sur les contacts de la carte à puce une fois celle-ci détectée (afin d'éviter toute perturbation du bus lors de son insertion), klaxon, rétro éclairage de l'afficheur LCD, reset de l'afficheur.
- Un connecteur pour **carte à puce** nativement I2C (TLC 2402)

Deux autres composants I2C se trouvent également sur le bus, dans la **barrière**. Il s'agit :

- D'un circuit PCF8574 « **mécanisme** » gérant : le variateur (montée, descente), boucles, fins de course...
- D'un circuit PCF8582 **EEPROM I2C** permettant de comptabiliser le nombre d'ouvertures / fermetures de la barrière, et facilite ainsi la maintenance.

5.2. Fonctionnement du port d'entrées / Sorties PCF8574

Le circuit permet de gérer 8 lignes d'entrée / sorties parallèles via le bus I2C.

ADRESSE : 0 1 0 0 A2 A1 A0 R/W : où A2, A1 et A0 sont les niveaux logiques appliqués sur les broches du même nom.

Lorsque le composant reçoit son adresse et que le bit R/W est positionné à 0, alors l'octet suivant l'adresse dans la trame est mémorisé et présenté sur les 8 lignes d'entrée / sorties.

Lorsque le composant reçoit son adresse et que le bit R/W est positionné à 1, alors le 8574 vient lire l'état réel qui est présent sur les 8 lignes d'entrée / sorties et le renvoi sous forme d'un octet placé à la suite de l'adresse dans la trame.

Affectation des broches du port d'entrées / Sorties PCF8574 « Mécanisme »

Remarque : A2, A1 et A0 sont reliées à la masse

| P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 | Caractéristiques |
|----|-----|-----|-----|-----|-----|----|----|---|
| E | E | E | E | E | S | S | S | E= ligne en entrée, S= ligne en sortie |
| | | | | | | 0 | 0 | Arrêt |
| | | | | | | 0 | 1 | Demande de montée |
| | | | | | | 1 | 0 | Demande de descente |
| | | | | | | 1 | 1 | Arrêt |
| | | | | | 0/1 | | | 0 : Balise éteinte 1 : Balise clignotante |
| | | | | 0/1 | | | | 0 : Lisse verticale 1 : Lisse Non verticale |
| | | | 0/1 | | | | | 0 : Lisse horizontale 1 : Lisse non horizontale |
| | | 0/1 | | | | | | 0 : Véhicule détecté sur boucle amont 1 : Pas de véhicule |
| | 0/1 | | | | | | | 0 : Véhicule détecté sur boucle aval 1 : Pas de véhicule |
| NU | | | | | | | | Non utilisée |

5.3. Travail demandé

1. On désire lire le circuit PCF8574 « Mécanisme » afin de connaître l'état de la balise lumineuse.

Complétez le tableau suivant en indiquant l'adresse et l'état du bit R/W :

| MSB | ADRESSE | | | | | LSB | R/W |
|-----|---------|--|--|--|--|-----|-----|
| | | | | | | | |

2. On désire écrire dans le circuit PCF8574 « Mécanisme » afin de faire clignoter la balise lumineuse.

Complétez le tableau suivant en indiquant l'adresse et l'état du bit R/W :

| MSB | ADRESSE | | | | | LSB | R/W |
|-----|---------|--|--|--|--|-----|-----|
| | | | | | | | |

Compléter le tableau suivant en indiquant l'état des bits de la donnée à envoyer :

ATTENTION : On désire uniquement allumer la balise et laisser le reste du système en l'état. Il faut donc recopier l'état des bits que l'on ne désire pas modifier en utilisant le résultat de la lecture précédente : 1110 1000 et modifier uniquement l'état du bit permettant la commande de la balise. Les bits correspondant aux broches en entrée n'ont aucun effet !

| MSB | Donnée | | | | | LSB |
|-----|--------|--|--|--|--|-----|
| | | | | | | |