

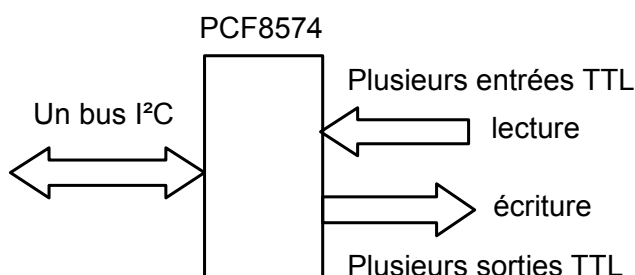
NOTICE PCF 8574

1. Avertissement

Ce texte est une version très réduite de la notice officielle du composant d'expansion du bus I²C

2. Présentation du rôle du composant

Ce composant fait le lien entre des entrées ou des sorties de type logique, au niveau TTL et le transport des informations portées par ces entrées ou ces sorties, par un bus I²C



Le composant est dit quasi bidirectionnel car une broche TTL peut jouer soit le rôle d'une entrée soit le rôle d'une sortie alternativement mais pas les deux à la fois (c'est évident !)

Le composant réalise donc, suivant les cas,

- Une sérialisation, lors de l'entrée, c'est à dire le passage d'informations sous forme parallèle (octet) à une information sous forme série (I²C).
- Une dé-sérialisation, lors des sorties, c'est à dire le passage de série (I²C) à parallèle (octet)

3. Communication avec le composant

La communication se fait de maître à esclave par le bus I²C. Le composant ne peut jouer le rôle de maître.

La communication s'opère à l'aide de l'envoi de deux octets sur la liaison I²C

- un octet d'adresse, qui permet la sélection du composant
- un octet de données qui porte les ordres du maître dans le cas d'une écriture ou l'état des entrées du composant dans le cas d'une lecture

Octets de communication

Adresse	L	H	V	V	A2	A1	A0	RW	le bit V dépend de la version du composant
Donnée	P7	P6	P5	P4	P3	P2	P1	P0	

Lorsque RW est à 0, la donnée sera écrite, lorsque RW est à 1, elle sera lue. Il faudra donc placer les instructions correctes dans le programme.

Les bits P0 à P7 portent l'état du port correspondant

4 . Chronologie d'un échange en écriture

Terminologie

Start : condition de start
SelW : envoi de l'octet de sélection du composant, le bit de sens est en écriture
Ack : acquittement de la part du composant
OctetW : envoi de l'octet de donnée
Stop : condition de stop

Start SelW Ack OctetW Ack Stop

5 . Chronologie d'un échange en lecture

Deux cas peuvent se présenter

- on souhaite lire une seule fois l'état des entrées entre un start et un stop
- on souhaite répéter la lecture dans l'attente d'une modification des entrées entre un start et un stop (scrutation)

Le principe général, valable pour tous les composants I²C, est le suivant :

Toutes les lectures d'octet sont suivies par un Ack sauf la dernière de la série qui est suivie d'un Nack

Chronologie pour la lecture d'un seul octet

Start SelW Ack OctetR Nack Stop

Chronologie pour la lecture de m octets

Start SelW Ack Start SelR Ack OctetR-1 Ack ... OctetR-m Nack Stop

Remarquez la reprise de la condition de Start

6 . Particularité de la constitution de l'octet d'adresse dans un protocole I²C

L'octet d'adresse, le premier qui est envoyé, lors d'une communication, est constitué de deux parties :

- l'adresse proprement dite
- le bit de sens

L'adresse proprement dite est divisée, elle aussi en deux parties

la partie haute de l'adresse est fixée par le constructeur

la partie basse est laissée libre afin que l'utilisateur puisse mettre en œuvre plusieurs composants de même type sur une ligne I²C. Configuration matérielle sur le composant.

Ici on trouvera

0	0	1	1	1	A2	A1	A0	pour le PCF8574 A
0	0	1	0	0	A2	A1	A0	pour le PCF8574

Introduction du bit de sens

Il faut décaler cet octet vers la gauche pour faire de la place pour le bit de sens.

Il suffit de multiplier par deux l'octet ci-dessus puis de lui ajouter le bit de sens souhaité.

Je conseille de conserver l'adresse proprement dite comme constante et d'effectuer les manipulations pour insérer le bit de sens dans le sous-programme d'écriture de l'adresse. Cette astuce permet d'utiliser le même sous-programme d'écriture de l'adresse pour tous les composants I²C présents sur la ligne.

7. Exemples de sous programmes

Avec un PicBasic

```
'----- condition de début de transmission
I2Cstart :
out sda, 1 'par précaution
out scl,1
out sda,0
return
'-----
'----- condition de fin de transmission
I2Cstop :
out sda, 0 'par précaution
out scl,1
out sda,1
return
'-----
'----- Sélection du composant en écriture, la donnée est dans la variable adr (adr est modifiée)
SelW:
gosub I2Cstart 'on peut également expliciter au lieu d'appeler le SP
adr = adr * 2 'décalage de 1 place vers la gauche
adr = adr + 0 'introduction du bit de sens (par cohérence avec la lecture)
shiftout scl, sda, mode, adr 'mode = 2 pour l'attente d'un ack
'-----
'----- Sélection du composant en lecture, la donnée est dans la variable adr (adr est modifiée)
SelR:
gosub I2Cstart 'on peut également expliciter au lieu d'appeler le SP
adr = adr * 2 'décalage de 1 place vers la gauche
adr = adr + 1 'introduction du bit de sens
shiftout scl, sda, mode, adr 'mode = 2 pour l'attente d'un ack
'-----
'----- écriture d'un octet, la donnée est dans la variable don
OctetW:
shiftout scl, sda, mode, don
return
'-----
'----- lecture d'un octet courant
OctetR:
don = shiftin(scl, sda, mode) ' mode est en général égal à 3
out scl, 0 'par précaution
out sda,0 'génération d'un Ack = impulsion sur scl avec sda = 0
out scl, 1
out scl, 0
return
'-----
'----- lecture d'un octet de fin
OctetRF:
don = shiftin(scl, sda, mode) ' mode est en général égal à 3
out scl, 0 'par précaution
out sda,1 'génération d'un Nack = impulsion sur scl avec sda = 1
out scl, 1
out scl, 0
return
```

Avec le CUBLOC, l'écriture du programme est plus simple