

EEPROM 24LC32

1 . Présentation du composant

Une EEPROM est une mémoire qui conserve ses données même en absence de tension d'alimentation. On peut lire son contenu bien entendu, mais on peut également y inscrire des valeurs, les effacer etc en utilisant des signaux de type TTL c'est à dire des signaux électriques¹. Les catégories précédentes de mémoires non volatiles² ne permettaient pas toutes ces opérations à l'aide de signaux TTL.

Sa capacité est 32 Kbits répartis dans 4 Koctets. On ne peut pas s'adresser individuellement à chacun des bits, c'est un argument commercial pour faire croire à une grande capacité. On s'adresse à un octet au minimum.

Le transfert des informations se fait par une liaison I²C c'est à dire que tous transferts d'informations se font sous forme série. La mise en œuvre d'une mémoire nécessite :

- le choix, par son adresse, de l'octet à traiter
- de réaliser une lecture ou une écriture de cet octet.

Pour s'adresser à chacun des 4 Koctets, il faut 4096 adresses différentes soit l'utilisation d'un nombre binaire de 12 chiffres répartis de A0 à A11. Une adresse sera constituée de deux octets, l'octet de poids faible (de A0 à A7) et l'octet de poids fort (de A8 à A11), les autres chiffres de l'octet sont sans signification.

Attention : il ne faut pas confondre l'adresse du composant avec l'adresse de l'octet à traiter, à l'intérieur du composant. L'octet de sélection donne l'adresse du composant .

2 . Les modes d'accès aux données

Le composant possède plusieurs modes d'accès aux données.

Le compteur d'adresse interne de la mémoire s'incrémente automatiquement à chaque opération et il conserve sa valeur entre deux accès. Grâce à cette particularité on peut réaliser plusieurs modes d'accès dont le seul que nous allons étudier : l'accès séquentiel. C'est celui qui est utilisé dans l'application sur laquelle nous travaillons:.

Principe de l'accès séquentiel

Après avoir sélectionné le composant et fourni l'adresse de départ des transferts, on opère une série d'opérations de lecture ou d'écriture en laissant le soin au compteur d'adresse interne de fournir l'adresse de l'octet à traiter. On profite ainsi de l'auto-incrémentation du compteur d'adresse pour simplifier la trame des échanges.

3 . Trame pour une écriture séquentielle de m octets

SeIW : octet de sélection du composant avec le bit de sens en écriture

AdrMSB : écriture de l'octet de poids fort de l'adresse interne

AdrMSB : écriture de l'octet de poids faible de l'adresse interne³

Start SeIW Ack AdrMSB Ack AdrLSB Ack OctetW-1 Ack OctetW-m Ack Stop

1 D'où le premier E du mot EEPROM

2 Qui conservent leur contenu en absence d'alimentation

3 Voir le texte sur le LM75 pour plus d'informations

4 . Trame pour une lecture séquentielle de m octets

SeIR : octet de sélection du composant avec le bit de sens en lecture

Start SeIW Ack AdrMSB Ack AdrLSB Ack (suite ligne suivante)
Start SeIR Ack OctetW-1 Ack OctetW-m Nack Stop

Attention toutes ces instructions devraient être représentées sur la même ligne.

Étudions les particularités de la trame de lecture d'un composant I²C.

première partie **Start SeIW Ack AdrMSB Ack AdrLSB Ack**

cette première partie configure le compteur interne des adresses, elle réalise une opération d'écriture. Les deux octets d'adresse sont transmis l'un à la suite de l'autre

seconde partie **Start SeIR Ack OctetW-1 Ack OctetW-m Nack Stop**

on remarque la répétition de la condition de start, en effet, le protocole I²C précise que le premier octet qui suit une condition de start est une sélection de composant. Ici il faut sélectionner le composant en lecture. À partir d'ici le composant sait que le maitre doit procéder à une lecture des données, à partir de l'adresse définie en première partie.

Ensuite vient la lecture proprement dite des octets. Chaque lecture est suivie par un acquittement de la part du maitre sauf la dernière qui est suivie d'un Nack⁴

5 . Adresse du composant

Comme d'habitude, l'octet de sélection du composant est constitué :

- d'une partie fixe imposée par le constructeur
- d'une partie définie par l'utilisateur selon la configuration matérielle des broches A0, A1, A2
- du bit de sens

ici l'adresse proprement dite est **0 1 0 1 0 A2 A1 A0**

⁴ Voir les feuilles sur PCF8574 pour des exemples de programmation en PicBasic. Le Cubloc possède des instructions toutes faites.