1. Problème :

Comment un microcontrôleur échange-t-il des octets avec un circuit ? L’échange d’octets est ce qu’on appelle une communication.

Pour cela on dispose de plusieurs possibilités :

La communication parallèle :

Sur un microcontrôleur c’est la communication la plus facile. On peut les repérer sur le microcontrôleur 18F2520.

*Fonctionnement* : sur un fil correspond un élément à piloter. Exemple une led, un relais, un transistor…

*Avantages*: simple, rapide

*Inconvénients* :

* grand nombre de broches perdues pour piloter un LCD par exemple
* sur les grandes distances l’information se perd.

La communication série :

Cette communication se fait sur 2 fils. Il en existe deux types. On les repère sur le microcontrôleur.

*Fonctionnement* : deux possibilités :

1. Les données s’envoient les unes derrière les autres et sont synchronisées avec une horloge.
2. Les données sont transmises les unes derrière les autres mais pas l’horloge. L’horloge est indépendante sur chaque circuit.

**On s’intéresse à la communication synchrone avec le protocole I2C ou IIC.**

1. Le protocole IIC (Inter Integrated Circuit):

Présentation :

Ce protocole a été inventé par Philips en 1982.

C’est un bus de communication synchrone. La norme électrique et le protocole d’échange sont très répandus. Dans le commerce il existe de nombreux périphériques disponibles.

La communication se fait sur deux fils + un :

Deux fils pour le signal :

* le signal d’horloge appelé SCL serial clock
* la transmission des données appelée SDA serial data

 Un fil pour la référence.

Le protocole :

Exemple :

***Etude d’une écriture d’un microcontrôleur vers une carte à puce :***

Dans le protocole I2C il y a une notion de maitre et d’esclave. Le maitre ici sera le microcontrôleur et l’esclave la carte à puce.

***Etude d’une écriture d’un microcontrôleur vers une carte à puce :***

* Le maitre demande une communication à l’aide d’un bit appelé START à la carte à puce. Il donne aussi l’adresse d’écriture de la carte à puce.
* L’esclave dit qu’il a reçu l’ordre en envoyant un bit d’acquittement..
* Le maitre envoie alors sa donnée sur 8 bits.
* Puis l’esclave dit qu’il a reçu la donnée en envoyant un acquittement.
* ….
* Jusqu’à ce que le maitre arrête la communication à l’aide d’un bit de STOP.

***Etude d’une lecture d’un microcontrôleur vers une carte à puce :***

Exercice :

1. Soit le schéma ci-après :

µ C

EEPROM

RAM

Afficheur

Adresse : 1 0 1 0 A2 A1 A0 (R/W)

A2 et A1 reliées à +Vcc

A0 reliée à la masse

R/W = 0 Write R/W =1 Read

Adresse : 1 1 1 0 1 1 A0 (R/W)

A0 reliée à la masse

R/W = 0 Write R/W =1 Read

Adresse : 0 1 0 0 1 A1 A0 (R/W)

A1 reliée à +Vcc

A0 reliée à la masse

R/W = 0 Write R/W =1 Read

Le microcontrôleur veut lire dans l’EEPROM la valeur de l’adresse D0h et envoyer cette valeur sur l’afficheur.

Donner l’adresse de lecture et d’écriture de l’EEPROM et de l’afficheur.

Elaborer la trame de cet échange sachant qu’il y a un bit de START et un bit de STOP. Et que les données de l’EEPROM sont :

|  |  |
| --- | --- |
| Adresse | Donnée |
| ED | 01 |
| EE | B6 |
| EF*Les données sont en hexadécimal ainsi que les adresses.* | A0 |
| D0 | 25 |
| D1 | 03 |
| D2 | 42 |
| D3 | 4D |
| D4 | 66 |

1. Donner la trame pour que la donnée soit envoyée à l’afficheur.

3) Le microcontrôleur veut maintenant écrire la valeur 56h à l’adresse 00h de la RAM.

1. Elaborer l’algorigramme nécessaire pour faire cette action.
2. Elaborer la trame sachant qu’il y a un bit de START et un bit de STOP.