I2Cs

D.Moreaux

26 février 2025

1 Le protocole I2C

Le protocole I2C est un protocole prévu pour la communication entre un composant principal (appelé Maître) et des composants secondaires (appelés Esclaves).

Cette communication se fait au travers d'une connexion à 3 fils : la masse, un signal d'horloge (SCL) et un signal de données (SDA). Ces signaux sont tenus au niveau haut par des résistances de pull-up et les différents composants pourront amener ces broches au niveau bas selon les besoins. Seul le Maître peut contrôler le signal d'horloge (SCL).

Tous les échanges sont initiés par le Maître qui commencera par un octet indiquant le numéro du composant destination sur 7 bits suivi du sens de communication (0-écriture/1-lecture du composant ciblé) sur un bit. Certains périphériques prévus pour ne communiquer que dans un sens auront une adresse donnée sur 8 bits, il suffit de la diviser par 2 pour obtenir l'adresse 7 bits correcte.

Pour une écriture sur un Esclave, le Maître envoie d'abord l'adresse du périphérique en écriture puis le numéro de registre et la ou les valeurs à écrire à partir du registre indiqué.

Pour une lecture sur un Esclave, le Maître envoie d'abord un ordre d'écriture pour sélectionner le registre puis envoie l'adresse du périphérique en lecture (dernier bit à 1), libère SDA et continue à générer le signal SCL pendant que l'esclave envoie les données en lecture. Cela s'arrêtera à la demande du Maître une fois le nombre d'octets désirés auront été lus.

On trouve de nombreux périphériques I2C : capteurs, extenseurs d'entrée/sortie, contrôleur de servo, écrans, mémoire EEPROM,...

Un microcontrôleur peut généralement être mis en mode Maître ou Esclave (ce qui permet de faire communiquer plusieurs microcontrôleurs ensemble)

2 Les commandes Arduino

Pour utiliser l'I2C sur Arduino, il faudra utiliser la bibliothèque `Wire.h`. Cette bibliothèque utilisera l'EUART du microcontrôleur et donc les broches SCL/SDA seront fixées. Pour l'Arduino Uno et similaires, ces bornes sont SDA(A4) et SCL(A5). Les bornes A4 et A5 ne peuvent donc pas être utilisées en entrée/sortie si on utilise l'I2C.

2.1 Général

begin() configure la bibliothèque Wire en Maître

begin(adr) configure la bibliothèque Wire en Esclave sur l'adresse adr (7 bits)

end() déconfigure la bibliothèque Wire. Cela libère les broches A4 et A5.

write(val) envoie un octet sur le bus

write(str) envoie une chaîne de caractère sur le bus

write(data[,len)] envoie une série d'octets. `uint8_f data[]`

read() lit un octet du bus I2C

setClock(f) configure la vitesse du bus. f=100000 (normal), 400000 (fast) [1]

available() donne le nombre d'octets disponibles en lecture

Sur Arduino, le buffer de réception est de 32 octets.

2.2 Maître

Le mode Maître est initié par l'instruction `Wire.begin()`. Le microcontrôleur utilisera ensuite les instructions `Wire.beginTransmission(adr)`, `Wire.write(...)` et finalement `Wire.endTransmission()` pour envoyer des informations en écriture et utilisera `Wire.requestFrom(adr,num)` pour demander num octets qui seront lus par `Wire.read()`.

2.3 Esclave

L'arduino utilisera `Wire.onReceive(hnd)` pour enregistrer une fonction prenant un nombre d'octets reçus qui permettra de traiter les données envoyées par le Maître et `Wire.onRequest(hnd)` pour enregistrer une fonction qui enverra des données quand elles sont demandées par le Maître. Les fonctions handler seront appelées automatiquement par la bibliothèque Arduino.

1. Des vitesses de 10000, 1000000 ou 3400000 sont aussi possible si le microcontrôleur et le périphérique les supportent