



## Ce qu'il faut savoir...

Le bus I2C (*Inter Integrated Circuit*) a été développé par *Philips Semiconductor* au début des années 80. Il permet de connecter très simplement divers composants électroniques au travers d'un bus composé de seulement 3 fils (SCL, SDA, GND). Il est aujourd'hui largement utilisé lorsqu'il s'agit de connecter plusieurs périphériques avec un minimum de fils sur de courtes distances.

IOExpander de Microchip, le MCP23016. Il sera présenté plus loin afin de vous permettre de réaliser votre propre carte très facilement.

## Connexion à une carte de développement

Notre carte d'extension I2C comporte un IO-Expander MCP23016 qui sert essentiellement à piloter 2 afficheurs 7 segments avec point. La carte se connecte à la platine de développement à l'aide de grip-fils directement connectés aux connecteurs 40 broches mâles de la carte CPU.

La carte d'extension est alimentée par la carte CPU, nous avons donc besoin de 4 fils au total :

- *rouge* : alimentation de la carte en 5V,
- *noir* : masse,
- *vert* : ligne I2C SDA (pull-up sur la carte CPU),
- *jaune* : ligne I2C SCL (pull-up sur la carte CPU).

Le but est de coder une application qui permette d'afficher une valeur quelconque sur les 2 afficheurs 7 segments. L'application utilisera un driver sous uClinux 2.4 en s'appuyant sur le contrôleur I2C intégré du S3C44.

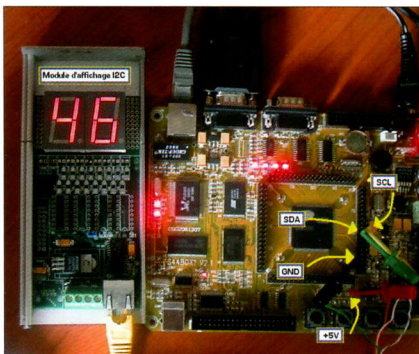


Figure 1. Connexion d'un périphérique I2C à une carte S3C44B0X

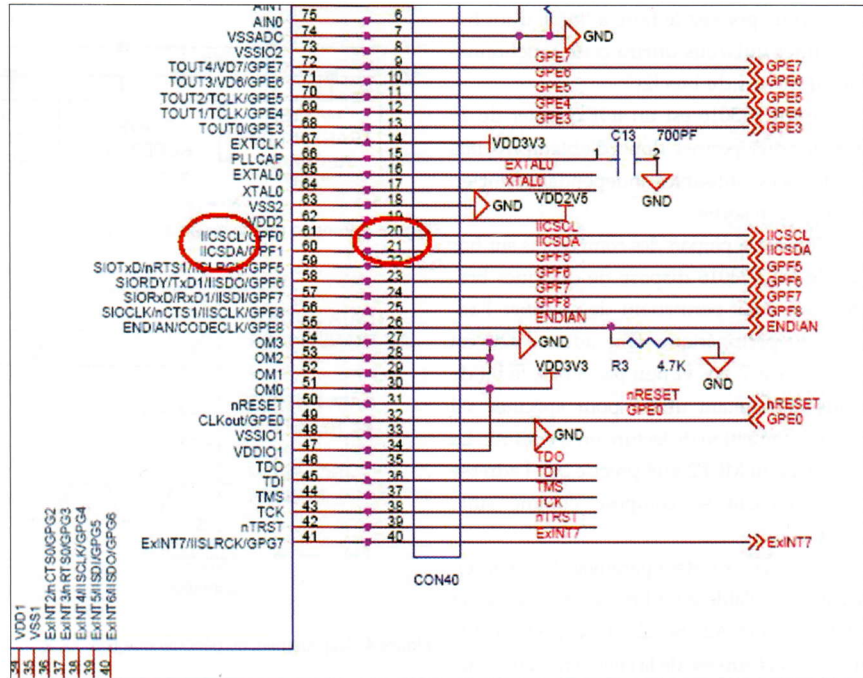


Figure 2. datasheet du S3C44B0X

## Détermination des broches d'extension

Pour connecter votre carte d'extension I2C à la plate-forme vous avez donc besoin de connaître l'emplacement exacte des signaux le schéma électrique de votre carte de développement.

Le document que nous utilisons est la au format PDF. Nous trouvons rapidement le numéro des broches qui sont affectées aux signaux SCL et SDA.

Pour ce qui est des alimentations, nous trouvons sur notre carte un connecteur d'alimentation IDE avec une point à 12V, un autre à 5V et les 2 centraux à GND. Il se trouve juste en dessous du processeur et des connecteurs d'extension. C'est là que nous connecterons les fils rouge et noir de notre carte d'extension I2C.

## Le bus I2C

Le bus i2c est un bus série synchrone, c'est-à-dire que la transmission des données de fait en série sur un seul fil (bit après bit) et qu'un autre signal sert à transmettre l'horloge. Ainsi sur chaque front d'horloge le bit de donnée est échantillonné afin de constituer un ou plusieurs octets.

La vitesse est habituellement de 100KHz, mais la plupart des périphériques actuels s'interfacent à des vitesses allant jusqu'à 400KHz.

Ce bus est dit multi-maître car il permet de connecter plusieurs systèmes I2C sur un même bus de sorte que chaque système a la

possibilité d'émettre (mode *MASTER*) et de recevoir des données (mode *SLAVE*).

Les données sont donc transmises en série d'un maître (émetteur) vers un esclave (le récepteur). Chaque récepteur se doit d'acquiescer la donnée reçue en maintenant la ligne SDA à 0 lors de la fin de la transmission de l'octet de donnée.

Le maître quant à lui initie le transfert à l'aide d'une séquence de signaux dite de *START*, et termine à l'aide d'une séquence *STOP*.

Comme plusieurs esclaves peuvent être connectés sur le bus i2c il convient de transmettre l'adresse du destinataire avant chaque trame transmise. Ainsi un seul et unique esclave sera concerné par la ou les données transmises.

Pour résumer, une communication I2C simple se caractérise par les étapes suivantes :

- *START*,
- Adresse du périphérique I2C,
- Acquiescement du périphérique,
- Transfert des données et acquiescement à l'issue de chaque transfert,
- *STOP*.

## Le périphérique MCP23016

Le composant MCP23016 de la société Microchip fait parti des composants dits *IOExpander*, c'est-à-dire *étendeur de ports* en mauvais français. Cela signifie que si vous souhaitez bénéficier de quelques IO pour piloter des relais ou encore lire des boutons pous-