

	<h2 style="margin: 0;">Le Bus I2C</h2>
<p>Sommaire :</p> <p>I - Le Bus I2C..... 1</p> <p>II - Mise en œuvre du bus I2C avec un ESP32..... 2</p> <p>III - Mise en œuvre du bus I2C avec un Raspberry Pi..... 2</p>	

I - Le Bus I2C

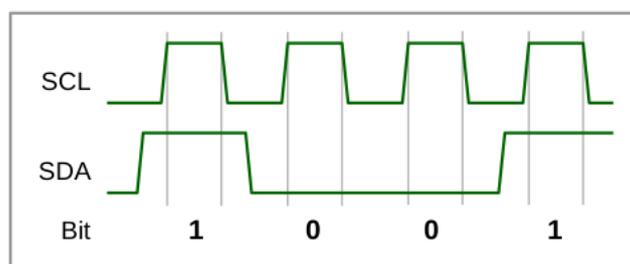
On présente généralement **I²C** (Inter Integrated Circuit) comme le plus simple des bus de communication utilisés dans l'électronique moderne. Il s'appuie simplement sur deux signaux appelés **SDA** (Serial Data) et **SCL** (Serial Clock), sans oublier la masse commune entre les équipements. Il s'agit d'une communication bidirectionnelle half-duplex reposant sur une communication série synchrone, où chacun ne parle qu'à son tour.

Le protocole permet de mettre en communication un composant maître (généralement le microprocesseur) et plusieurs périphériques esclaves. Notons également que le maître peut également envoyer un ordre à tous ses esclaves simultanément avec l'adresse **0x00** (par exemple une mise en sommeil ou une demande de réinitialisation).

La communication s'établit toujours à l'initiative du maître. Celui-ci présente une condition **START** sur la ligne **SDA**, suivie de l'adresse (sur sept bits) de l'esclave avec lequel il souhaite communiquer, puis un bit indiquant le sens (0 = écriture ou 1 = lecture) de la communication.



Pendant toute la durée du créneau haut du signal d'horloge **SCL**, le signal de données **SDA** doit être maintenu au niveau haut ou bas suivant que l'on transmet un 1 ou un 0.



Lorsqu'un périphérique reconnaît son adresse, il doit écrire sur la ligne **SDA** un bit 0, indiquant ainsi un acquittement (**ACK**). Si le message est une écriture vers le périphérique, le maître peut alors envoyer sa commande, octet par octet, l'esclave acquittant chacun d'entre eux par un bit **ACK**.

L'implémentation de l'**I²C** que l'on rencontre sur les systèmes monocartes sous Linux (Raspberry Pi, BeagleBone Black, etc.) est plutôt une spécialisation du protocole **SMBus** (System Management Bus). Ceci explique que de nombreuses fonctions de l'API Linux pour le sous-système **I²C** soient préfixées par **smbus**.

II - Mise en œuvre du bus I2C avec un ESP32

Il suffit de connecter le **capteur I2C** à l'**ESP32** comme décrit ci-dessous :

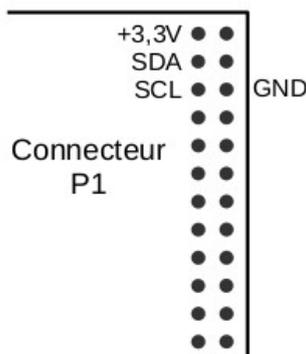
Capteur I2C	ESP32 Pin
3.3V	3.3V
GND	GND
SDA	GPIO21
SCL	GPIO22

III - Mise en œuvre du bus I2C avec un Raspberry Pi

Le protocole **I²C** est supporté par le noyau **Linux** depuis sa version 2.4. L'accès depuis l'espace utilisateur est facilité par le module **i2c-dev** qui rend les bus i²c visibles dans le répertoire **/dev** sous forme de fichier spéciaux représentant des périphériques en mode caractère ("**/dev/i2c-0**" ou "**/dev/i2c-1**").

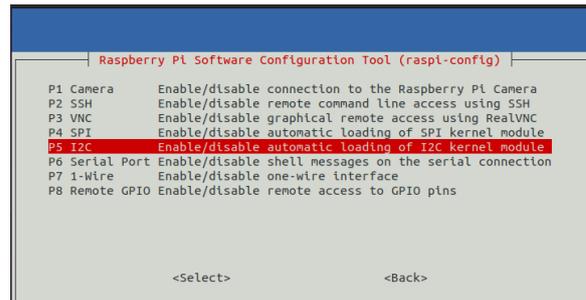
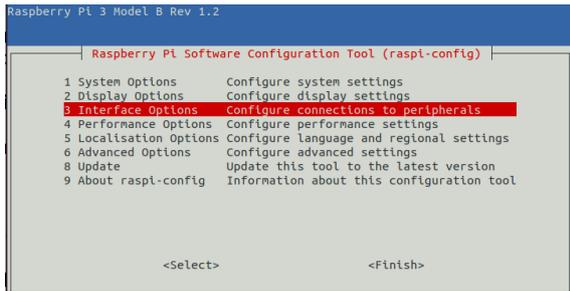
Le **Raspberry Pi** dispose de deux interfaces **I²C**. Le premier bus (numéro **0**) était accessible à travers le connecteur **P5** qui était apparu dans la seconde version du Raspberry Pi modèle B et qui a disparu depuis le modèle B+. Nous allons plutôt nous intéresser au second bus (numéro **1**), accessible via le port d'extension **P1** :

Capteur I2C	RPI Pin
3.3V	3.3V
GND	GND
SDA	broche 3 (signal SDA)
SCL	broche 5 (signal SCL)



Sur le **Raspberry Pi** il faut commencer par activer le bus **I²C** à l'aide du menu « **Interface Options** » de l'utilitaire **raspi-config** :

```
pi@raspberrypi:~ $ sudo raspi-config
```



La manière la plus simple de détecter les périphériques **I²C** est d'employer les utilitaires du package **i2c-tools**. Il faut tout d'abord installer ce dernier, car il n'est généralement pas inclus d'origine dans les distributions courantes :

```
pi@raspberrypi:~# sudo apt-get install i2c-tools
```

Nous pouvons commencer par lister les bus présents, puis scanner celui qui nous intéresse à l'aide de l'outil **i2cdetect** :

```
pi@raspberrypi:~ $ i2cdetect -l
i2c-1 i2c          bcm2835 (i2c@7e804000)    I2C adapter
pi@raspberrypi:~ $ i2cdetect -y 1
 0 1 2 3 4 5 6 7 8 9 a b c d e f
00:  --  - - - - -
10:  - - - - -
20:  - - - - -
30:  - - - - -
40:  - - - - -
50:  - - - - -
60:  - - - - -
70:  - - - - - 77
```

Cela montre qu'il y a un esclave **I²C** connecté et que son adresse est **0x77 (119 en décimal)**.

Afin d'accéder à nos esclaves **I²C** en langage **C++**, il faut installer le package **libi2c-dev** :

```
pi@raspberrypi:~# sudo apt-get install libi2c-dev
```

Pour ce qui concerne le capteur **BMP180**, on trouve différentes bibliothèques sur Internet notamment pour **Raspbian 9 Stretch** (<https://github.com/lexruee/bmp180>) et pour **Raspbian 10 Buster** (<https://github.com/leon-anavi/rpi-examples/tree/master/BMP180/c>).

Pour ce qui concerne le capteur **HDC1080**, on trouve différentes bibliothèques sur Internet notamment <https://github.com/MichaelF1/MMM-HDC1080>.