

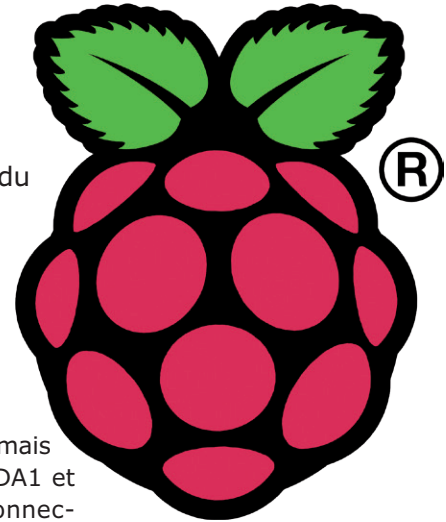
# Raspberry Pi

## part n°5

Tony Dixon  
(Royaume-Uni)

### I<sup>2</sup>C + I<sup>2</sup>C = Π

Trois interfaces série sont disponibles depuis le connecteur d'extension du Raspberry Pi : UART, SPI, et I<sup>2</sup>C. Nous avons déjà mis les deux premières à la sauce framboise, voyons si la dernière se laisse aussi facilement accommoder. Promis, je ne vous servirai pas de salade avec.



#### L'interface I<sup>2</sup>C

Les interfaces série UART et SPI nous sont familières depuis les parts n° 3 et 4 de cette série. Comme le montre le brochage du connecteur d'extension (**tableau 1**), l'interface I<sup>2</sup>C (*Inter Integrated Circuit*) est accessible depuis les broches 3 (SDA) et 5 (SCL).

A l'instar de SPI, I<sup>2</sup>C permet d'établir une interface entre le Raspberry et d'autres périphériques avec un nombre de broches minimal. I<sup>2</sup>C n'utilise que deux lignes bidirectionnelles à collecteur ouvert pour l'accès au bus : SDA (*Serial Data Line*) et SCL (*Serial Clock*). Ces lignes sont typiquement mises au niveau haut à l'aide de résistances, dans le cas du Pi avec deux résistances de rappel de 1,8 kΩ.

I<sup>2</sup>C n'est pas aussi rapide que d'autres bus. Le débit du mode standard est de 100 kbit/s, mais celui du mode rapide atteint tout de même 400 kbit/s.

La puce SoC de Broadcom du Pi possède deux interfaces I<sup>2</sup>C. Seule la première était disponible depuis le connecteur d'extension de la version d'origine du Pi (I2C\_SDA0 et I2C\_SCL0).

La révision 2 de la carte a été équipée d'un petit connecteur d'extension supplémentaire qui a permis l'accès à la seconde interface I<sup>2</sup>C. Détail qui a son importance quand on exploite I<sup>2</sup>C sur le Pi, avec la

carte de révision 2 on accède désormais à la seconde interface I<sup>2</sup>C (I2C\_SDA1 et I2C\_SCL1) depuis le « grand » connecteur, tandis que la première interface I<sup>2</sup>C

Tableau 1. Brochage du connecteur d'extension

nom de la broche	fonction	autre fonction	RPi.GPIO
P1-02	5,0V	-	-
P1-04	5,0V	-	-
P1-06	GND	-	-
P1-08	GPIO14	UART0_TXD	RPi.GPIO8
P1-10	GPIO15	UART0_RXD	RPi.GPIO10
P1-12	GPIO18	PWM0	RPi.GPIO12
P1-14	GND	-	-
P1-16	GPIO23		RPi.GPIO16
P1-18	GPIO24		RPi.GPIO18
P1-20	GND	-	-
P1-22	GPIO25		RPi.GPIO22
P1-24	GPIO8	SPI0_CE0_N	RPi.GPIO24
P1-26	GPIO7	SPI0_CE1_N	RPi.GPIO26

nom de la broche	révision 1 de la carte		révision 2 de la carte	
	fonction	autre fonction	fonction	autre fonction
P1-01	3,3V	-	3,3V	-
P1-03	GPIO0	I2C0_SDA	GPIO2	I2C1_SDA
P1-05	GPIO1	I2C0_SCL	GPIO3	I2C1_SCL
P1-07	GPIO4	GPCLK0	GPIO4	GPCLK0
P1-09	GND	-	GND	-
P1-11	GPIO17	RTS0	GPIO17	RTS0
P1-13	GPIO21		GPIO27	
P1-15	GPIO22		GPIO22	
P1-17	3,3V	-	3,3V	-
P1-19	GPIO10	SPI0_MOSI	GPIO10	SPI0_MOSI
P1-21	GPIO9	SPI0_MISO	GPIO9	SPI0_MISO
P1-23	GPIO11	SPI0_SCLK	GPIO11	SPI0_SCLK
P1-25	GND	-	GND	-



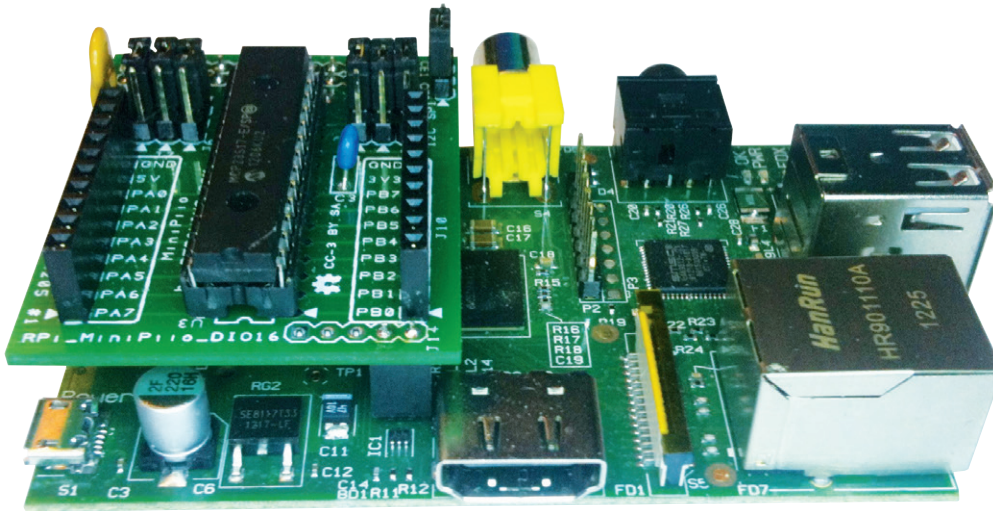


Figure 2. Pi et le MCP23017 sur une carte MiniPiio.

```
sudo apt-get update
sudo apt-get install i2c-tools
```

Nous devons ensuite ajouter un nouvel utilisateur au groupe i2c :

```
sudo adduser pi i2c
```

Redémarrez Pi :

```
sudo reboot
```

Lancez une nouvelle session LXTerminal et, pour vérifier que vous avez bien deux fichiers de périphériques I<sup>2</sup>C (un pour chaque interface I<sup>2</sup>C), entrez :

```
ls /dev/i2c*
```

La sortie de cette commande devrait être :

```
/dev/i2c-0
/dev/i2c-1
```

Assurez-vous que tout est correct en lançant, pour Pi révision 1 :

```
sudo i2cdetect -y 0
```

Et pour Pi revision 2 :

```
sudo i2cdetect -y 1
```

La sortie devrait être proche de celle reproduite sur la **figure 3**.

### Installation de la bibliothèque Python smbus pour I<sup>2</sup>C

Le programme d'exemple de ce projet est écrit en Python. L'interpréteur Python est installé par défaut dans la distribution Raspbian, mais aucune bibliothèque Python pour

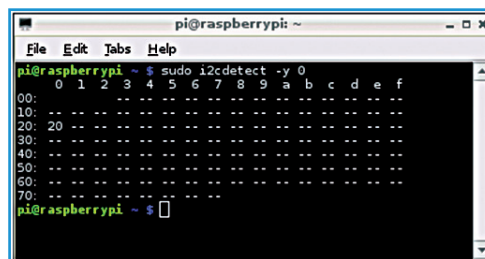


Figure 3. Sortie de *i2cdetect*.

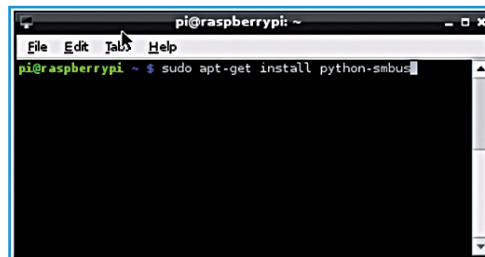


Figure 4. LXTerminal.

I<sup>2</sup>C n'est fournie d'avance. Commençons donc par télécharger et installer le module Python pour I<sup>2</sup>C appelé **smbus** (**fig. 4**) :

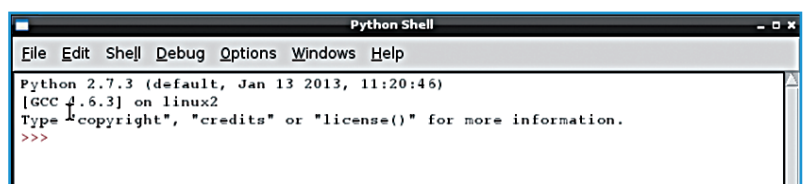
```
sudo apt-get install python-smbus
```

Nous voici prêts à dialoguer avec I<sup>2</sup>C en Python.

### Programme d'exemple : mcp23017.py

Nous allons importer *smbus* dans un court programme chargé d'allumer des LED reliées au duplicateur de ports GPIO.

Figure 5. L'environnement IDLE pour Python.

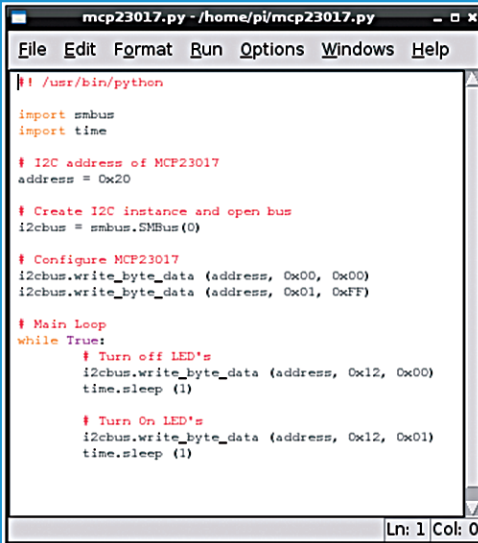


Double-cliquez sur l'icône IDLE du bureau pour lancer l'EDI et la console Python (fig. 5).

Dans le menu *File*, sélectionnez *New Window* pour créer un nouveau programme dans l'éditeur.

Copiez le programme du **listage** dans l'éditeur (fig. 6).

Sauvegardez le programme saisi. Retournez ensuite dans la console Lxterminal, et entrez la commande suivante pour le rendre exécutable :



```
mcp23017.py - /home/pi/mcp23017.py
File Edit Format Run Options Windows Help
#!/usr/bin/python
import smbus
import time

# I2C address of MCP23017
address = 0x20

# Create I2C instance and open bus
i2cbus = smbus.SMBus(0)

# Configure MCP23017
i2cbus.write_byte_data (address, 0x00, 0x00)
i2cbus.write_byte_data (address, 0x01, 0xFF)

# Main Loop
while True:
    # Turn off LED's
    i2cbus.write_byte_data (address, 0x12, 0x00)
    time.sleep (1)

    # Turn On LED's
    i2cbus.write_byte_data (address, 0x12, 0x01)
    time.sleep (1)
Ln: 1 Col: 0
```

Figure 6.  
L'éditeur d'IDLE.

### Listage

```
#!/usr/bin/python

import smbus
import time

# I2C address of MCP23017
address = 0x20

# Create I2C instance and open bus
i2cbus = smbus.SMBus(0)

# Configure MCP23017
i2cbus.write_byte_data(address,0x00,0x00) # Set Bank A to outputs
i2cbus.write_byte_data(address,0x01,0xFF) # Set Bank B to inputs

# Main loop
while True:
    # Turn off LEDs
    i2cbus.write_byte_data (address,0x12,0x00)
    time.sleep(1)

    # Turn on PortA.0
    i2cbus.write_byte_data (address,0x12,0x01)
    time.sleep(1)
```

**Note : Pour les cartes Pi de révision 2, remplacez la ligne :**

`i2cbus = smbus.SMBus(0)`     **par :**     `i2cbus = smbus.SMBus(1)`

```
sudo chmod +x mcp23017.py
```

Et pour exécuter ce programme :

```
sudo ./mcp23017.py
```

## Liens

[1] [ww1.microchip.com/downloads/en/devicedoc/21952b.pdf](http://ww1.microchip.com/downloads/en/devicedoc/21952b.pdf)

[2] [www.dtronixs.com](http://www.dtronixs.com)

Le **Tableau 2** résume les registres de commande du MCP23x17.

(130236)

**Tableau 2. Registres de commande du MCP23x17**

adresse IOCON.BANK = 1	adresse IOCON.BANK = 0	registre	description
0x00 / 0 déc.	0x00 / 0 déc.	IODIRA	I/O Direction Register for Port A
0x10 / 16 déc.	0x01 / 1 déc.	IODIRB	I/O Direction Register for Port B
0x01 / 1 déc.	0x02 / 2 déc.	IPOLA	Input Polarity Port Register for Port A
0x11 / 17 déc.	0x03 / 3 déc.	IPOLB	Input Polarity Port Register for Port B
0x02 / 2 déc.	0x04 / 4 déc.	GPINTENA	Interrupt-n-Change Control Register Port A
0x12 / 18 déc.	0x05 / 5 déc.	GPINTENB	Interrupt-n-Change Control Register Port B
0x03 / 3 déc.	0x06 / 6 déc.	DEFVALA	Default Compare Register for GPINTENA
0x13 / 19 déc.	0x07 / 7 déc.	DEFVALB	Default Compare Register for GPINTENB
0x04 / 4 déc.	0x08 / 8 déc.	INTCONA	Interrupt Control Register for Port A
0x14 / 20 déc.	0x09 / 9 déc.	INTCONB	Interrupt Control Register for Port B
0x05 / 5 déc.	0x0A / 10 déc.	IOCON	I/O Expander Configuration Register
0x15 / 21 déc.	0x0B / 11 déc.	IOCON	I/O Expander Configuration Register
0x06 / 6 déc.	0x0C / 12 déc.	GPPUA	Pull-Up Resistor Configuration Register Port A
0x16 / 22 déc.	0x0D / 13 déc.	GPPUB	Pull-Up Resistor Configuration Register Port B
0x07 / 7 déc.	0x0E / 14 déc.	INTFA	Interrupt Flag Register for Port A
0x17 / 23 déc.	0x0F / 15 déc.	INTFB	Interrupt Flag Register for Port B
0x08 / 8 déc.	0x10 / 16 déc.	INTCAPA	Interrupt Capture Register for Port A
0x18 / 24 déc.	0x11 / 17 déc.	INTCAPB	Interrupt Capture Register for Port B
0x09 / 9 déc.	0x12 / 18 déc.	GPIOA	Port Register for Port A
0x19 / 25 déc.	0x13 / 19 déc.	GPIOB	Port Register for Port B
0x0A / 10 déc.	0x14 / 20 déc.	OLATA	Output Latch Register for Port A
0x1A / 26 déc.	0x15 / 21 déc.	OLATB	Output Latch Register for Port B