**TP3 Raspberry PI**

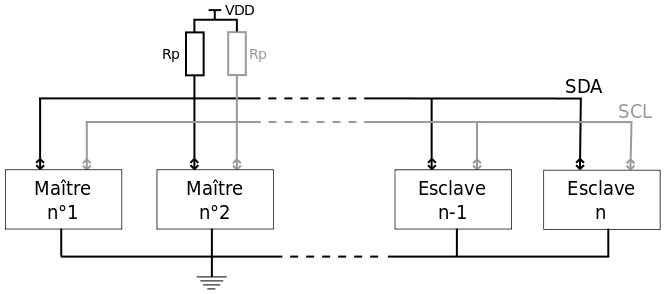
**Utilisation des GPIO : Le bus I2C**

**Objectif :**

Le but du TP est de réaliser un thermomètre numérique à l’aide d’un capteur de température I2C, le TMP102, et le Raspberry

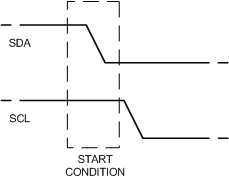
**Rappel sur le bus I2C**

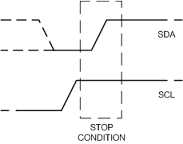
Le bus I2C est un bus local permettant les échanges séries à courte distance entre un microcontrôleur et des périphériques (ADC, DAC, afficheur, mémoire, capteurs etc …) ou un autre microcontrôleur, les échanges sont gérés par protocole avec adressage.  
Le bus I2C est synchrone et half-duplex, sa technologie d’interfaces drains-ouverts le rend très résistant aux courts-circuits et permet une gestion simple des erreurs.



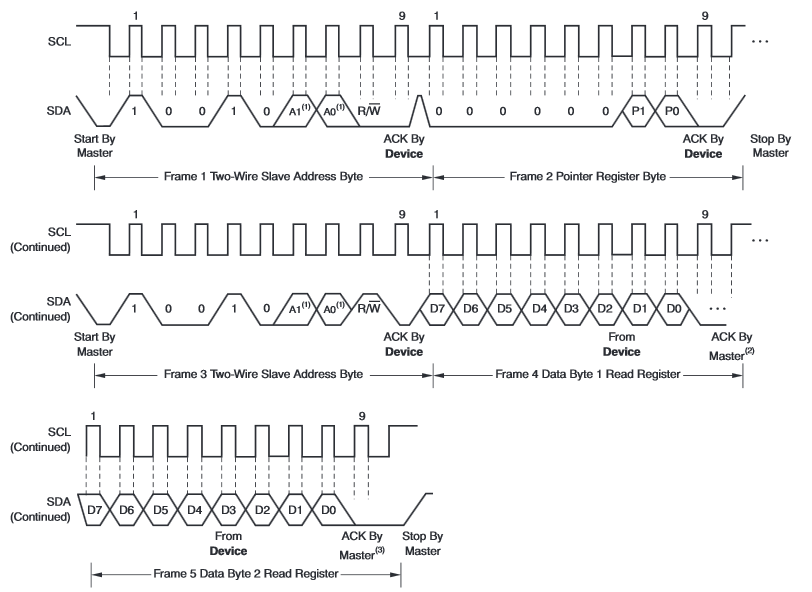
Un échange de données est toujours initié par le maitre (ici Raspberry Pi).

1. Il commence par une Start Condition : SDA passe à 0 puis SCL passe à 0 :

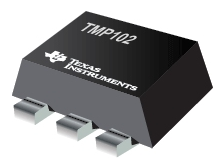


1. Puis l’adresse du destinataire : impaire pour une écriture (LSB R/W=1), paire pour une lecture (LSB R/W=0).
2. Acquittement par le destinataire qui place un niveau bas sur la 9ième impulsion sur SCL.  
   Le sens de l’échange des données dépend de l’application, lecture pour un capteur.
3. L’échange se termine par une condition stop, sda passe à 1 puis SCL passe à 1

Exemple de trame I2C, lecture d’un capteur TMP102



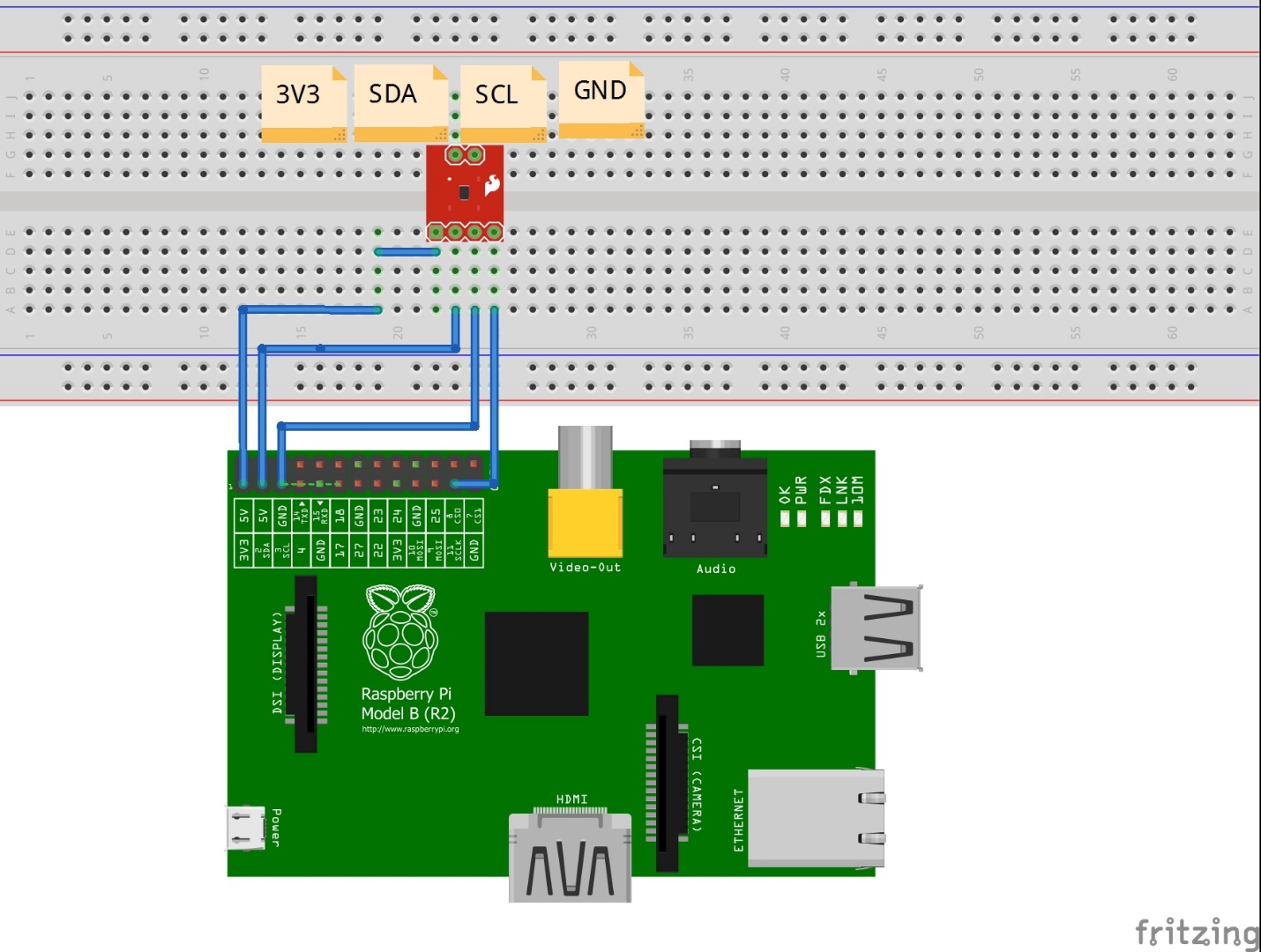
1. **Réalisation du montage**

Le TP met en œuvre le capteur Texas Instrument TMP102 : <http://www.ti.com/product/tmp102>

Ce composant disponible en boitier SOT563 est difficilement implantable sur circuit imprimé sans l’aide de machine de positionnement. Sparkfun propose une petite carte équipée d’un TMP102, des résistances de pull-up du bus I2C et d’un condensateur de découplage : <https://www.sparkfun.com/products/11931>

Le Rasbperry Pi dispose d’une interface I2C et d’une bibliothèque. broche 3 : SDA, broche 5 :SCL  
(les résistances de pull-up sont déjà sur le module Sparkfun)

Réaliser le montage suivant



1. **Programmation d’un thermomètre**
2. Afin d’installer le paquet **i2c-tools** et la bibliothèque **python-smbus**, taper les commandes suivantes :

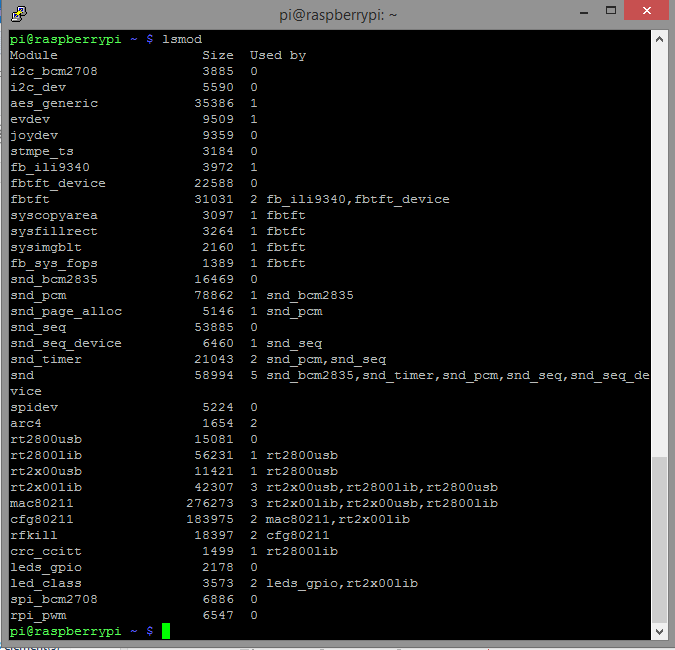
apt-get update

sudo apt-get install python-smbus  
sudo apt-get install i2c-tools

modprobe permet d’ajouter des modules dans le Noyau Linux, comme le module I2C.

1. Taper les commandes suivantes :

sudo modprobe i2c-dev  
sudo modprobe i2c-bcm2708   
modinfo nom\_du\_module 🡪 retourne les informations du module nom\_du\_module



1. Editer le fichier **raspi-blacklist.conf** en tapant la commande :

**sudo nano /etc/modprobe.d/raspi-blacklist.conf**

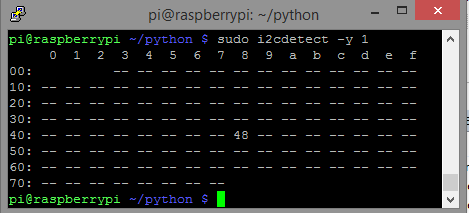
1. Si ce n’est pas le cas, ajouter des dièse(#) devant chacune des deux lignes suivantes :

**spi-bcm2708**

**i2c-bcm2708**

1. Afin de détecter les périphériques I2C connectés au Raspberry, taper la commande suivante :

sudo i2cdetect -y 1



1. Redémarrer le Raspberry pi.

Programme de gestion du capteur I2C tmp102 en python

#!/usr/bin/env python  
import smbus  
import time

bus\_pi = smbus.SMBus(1)

# addresse I2C du capteur TMP102  
addr = 0x48

while True:  
 try:

# x contient les 2 premiers octets transmis par le TMP102

x = bus\_pi.read\_word\_data(addr,0)  
 msb=x>>8  
 lsb=x&0x00FF  
 wtemp=((lsb<<8)|msb)>>4

# **affichage de adrr en hexa sur 2 chiffres et wtemp en hexa sur 4 chiffres**

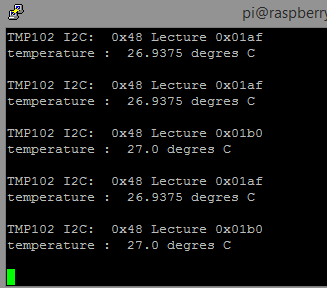
print 'TMP102 I2C: ',hex(addr), 'Lecture ', hex(wtemp),'\n\r'  
 print 'temperature : ',wtemp\*0.0625 ,'degres C\n\r'

# tempo de 0.5s avant une nouvelle lecture

time.sleep(0.5)  
 except:  
 print '...erreur...'  
 break

1. Lancer l’editeur « nano » en tapant la commande : **sudo nano**
2. Saisir le programme ci-dessus
3. Pour enregistrer **ctrl+x** et entrer le nom de fichier **TMP102.py**. Ensuite valider en tapant sur **Y**.
4. Pour tester votre programme, taper la commande **sudo python TMP102.py**

Résultat attendu :



1. Pour sortir du programme, taper sur ctrl+c.