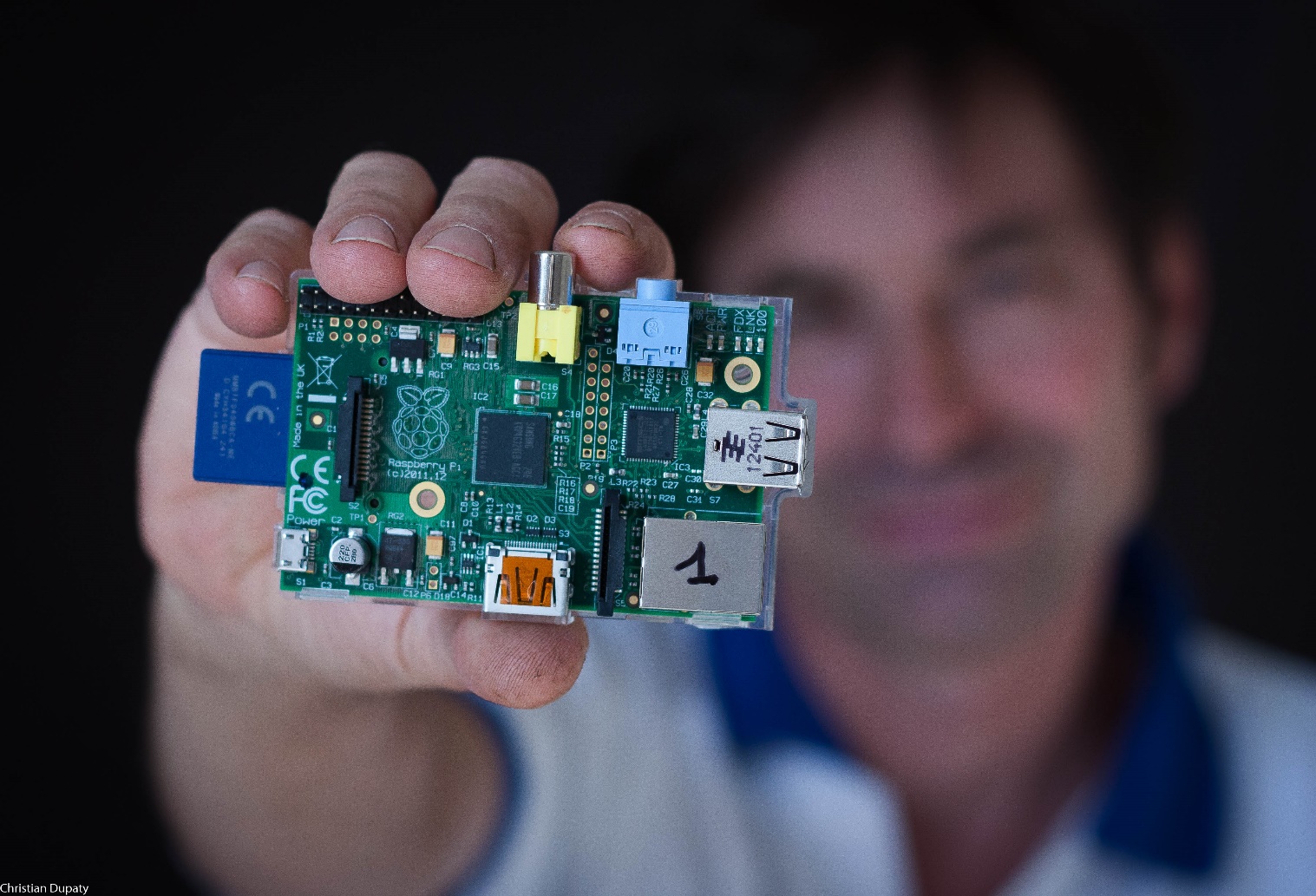
**RASPBERRY PI**

**INSTALLATION-CONFIGURATION  
INTERFACES DE COMMUNICATIONS  
**

I2C

Christian Dupaty

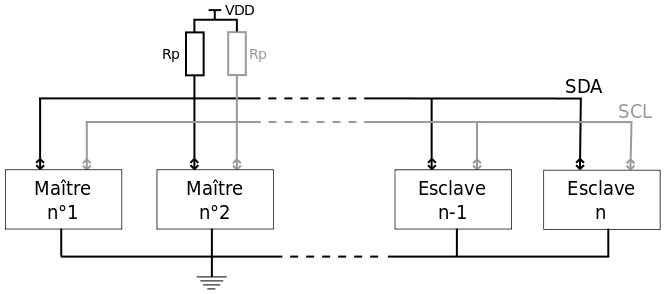
BTS Systèmes Numériques

Lycée Fourcade - Gardanne

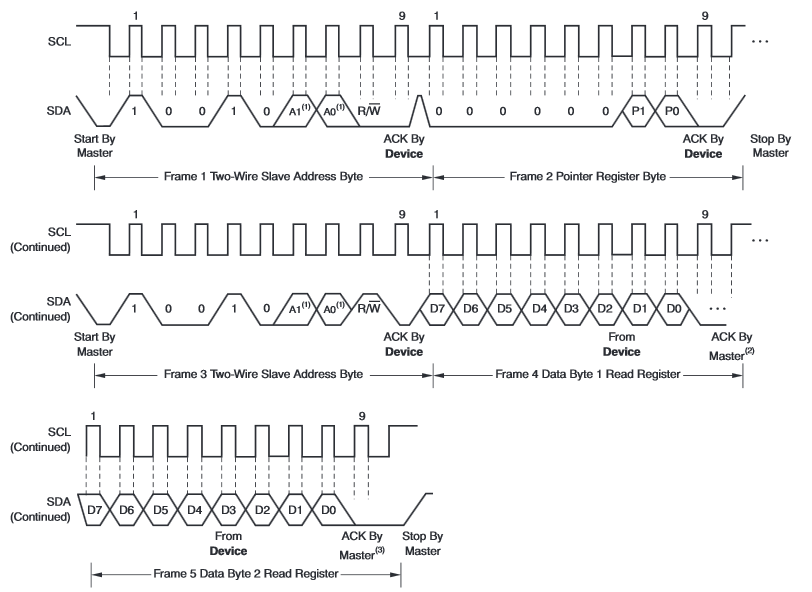
Académie d’Aix-Marseille

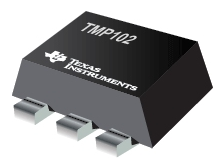
# TP : I2C

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.

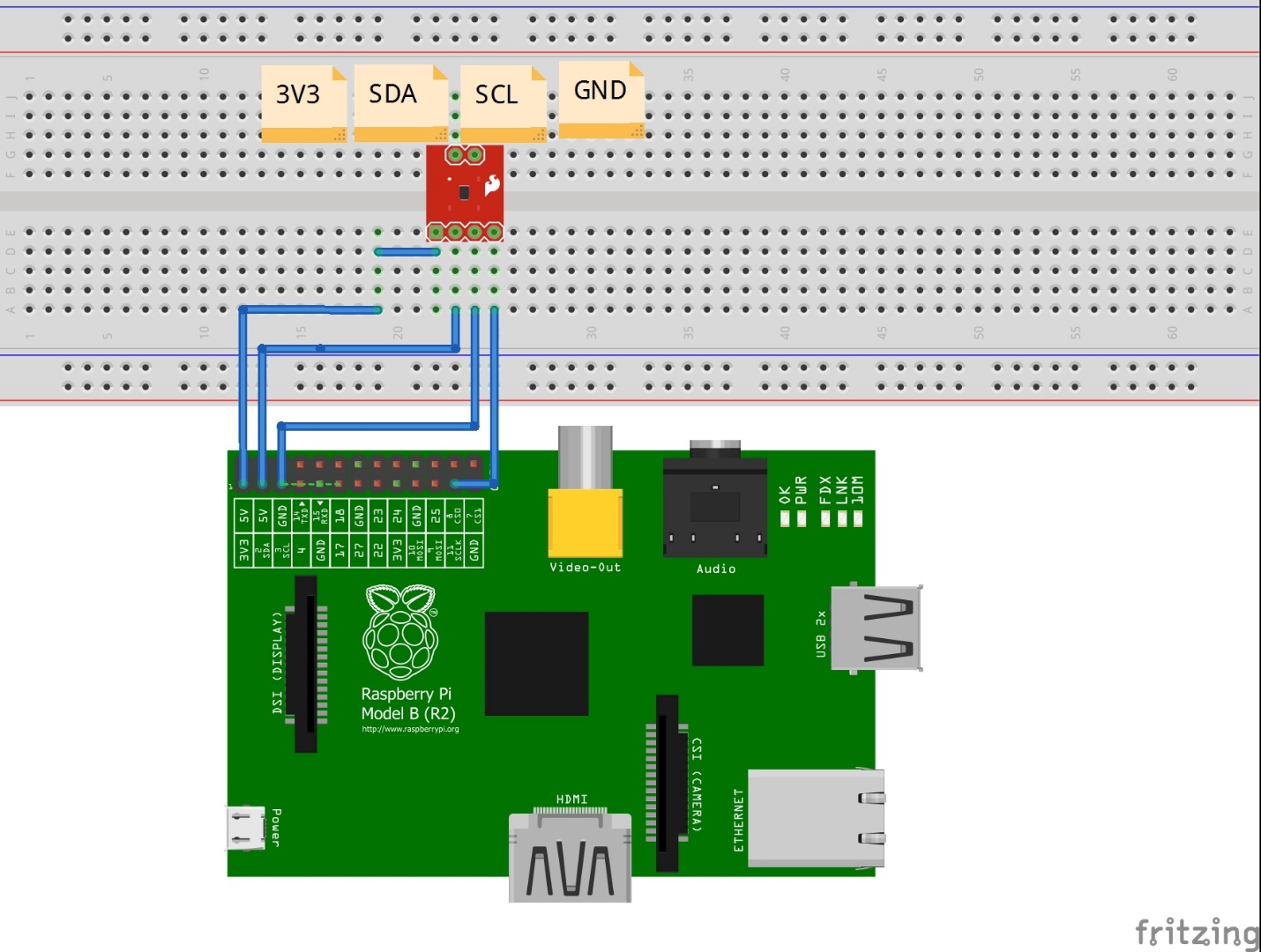
© wilkipedia

Le niveau haut (1) est récessif, le (0) est dominant.  
Un échange de données est toujours initié par le maitre (ici Raspberry Pi).   
Il commence par une Start Condition : SDA passe à 0 puis SCL passe à 0  
Puis l’adresse du destinataire : impaire pour une écriture, paire pour une lecture  
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, écriture pour un afficheur.  
L’échange se termine par une condition stop, sda passe à 1 puis SCL passe à 1  
  
Exemple de trame I2C, lecture d’un TMP102 (Texas-Instruments)



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>



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

**La bibliothèque Python I2C-SMBUS pour Raspberry Pi**

*I2C et SMBUS sont très similaires, les différences résident dans les fréquences d’horloge et les tensions de déclenchement.*

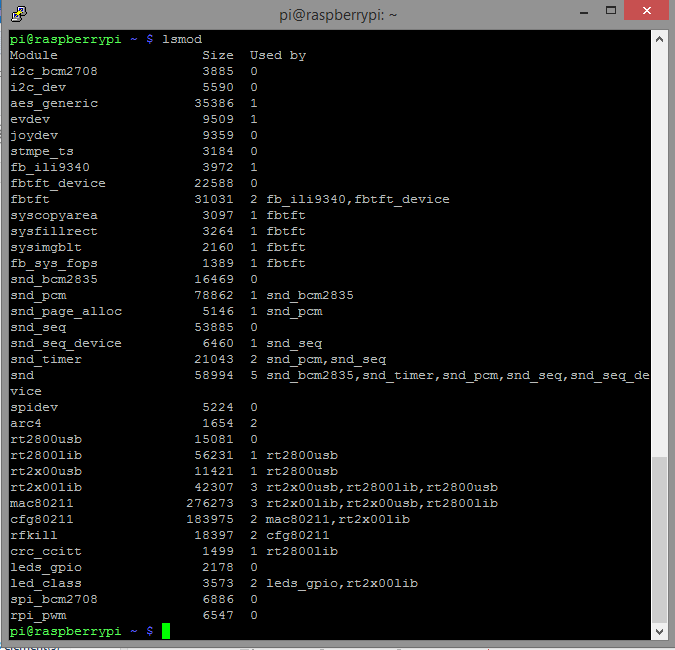
|  |  |  |
| --- | --- | --- |
|  | I²C | SMBus |
| Timeout | No | Yes |
| Minimum Clock Speed | DC | 10kHz |
| Maximum Clock Speed | 100kHz (400kHz and 2MHz also available) | 100kHz |
| VHIGH | 0.7 × VDD, 3.0V Fixed | 2.1V |
| VLOW | 0.3 × VDD, 1.5V Fixed | 0.8V |
| Max I | 3mA | 350µA |
| Clock Nomenclature | SCL | SMBCLK |
| Data Nomenclature | SDA | SMBDAT |

**Programmation d’un thermomètre**

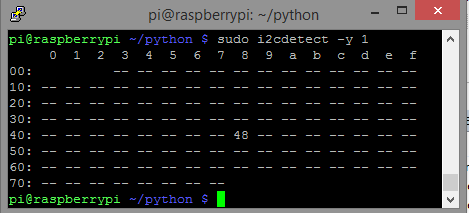
Recuperation des parquets  
apt-get update  
sudo apt-get install i2c-tools

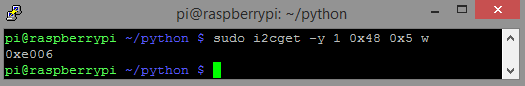
**La commande modprobe**modprobe permet d’ajouter (et d’enlever) des modules dans le Noyau Linux, comme le module I2C.  
sudo modprobe i2c-dev  
sudo modprobe i2c-bcm2708

lsmod retourne la liste des modules actifs  
modinfo nom\_du\_module retourne les informations du module nom\_du\_module



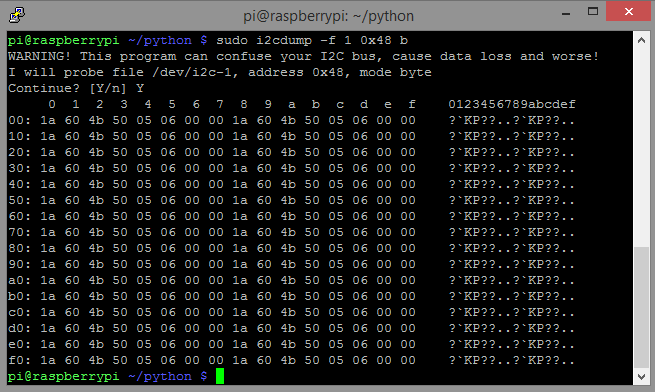
**Commandes de la librairie i2c-dev**sudo i2cdetect -y 1

  
Lit le mot (w) ou l’octet (b) dans l’adresse 5 (rien pour un lire octet unique) du circuit à l’adresse I2C 0x1F   
sudo i2cget -y 1 0x48 0x5 w



Ecrit 55 dans le circuit à l’adresse I2C 0x20 sur le bus 1  
sudo i2cset -y 1 0x20 55

Lit le contenu en octets (b) ou mot (w) du circuit à l’adresseI2C 0x48 sur le bus 1  
i2cdump –f 1 0x48 b

l

**Utilisation de la bibliothèque Python I2C-SMBUS**

Gestion capteur tmp102 en python

#!/usr/bin/env python  
# l acces au bus I2C de la Raspberry Pi necessite :  
#sudo modprobe i2c-dev  
#sudo modprobe i2c-bcm2708  
# pour verifier la presence d'un peripherique i2c :  
#sudo i2cdetect -y 1

import smbus  
import time

bus\_pi = smbus.SMBus(1)

# I2C address for TMP102  
addr = 0x48

while True:  
 try:  
 x = bus\_pi.read\_word\_data(addr,0)  
 msb=x>>8  
 lsb=x&0x00FF  
 wtemp=((lsb<<8)|msb)>>4  
 print 'TMP102 I2C: 0x{0:02x} Lecture 0x{1:04x}'.format( addr,wtemp )  
 print 'temperature : ',wtemp\*0.0625 ,'degres C\n\r'  
 time.sleep(0.5)  
 except:  
 print '...erreur...'  
 break

