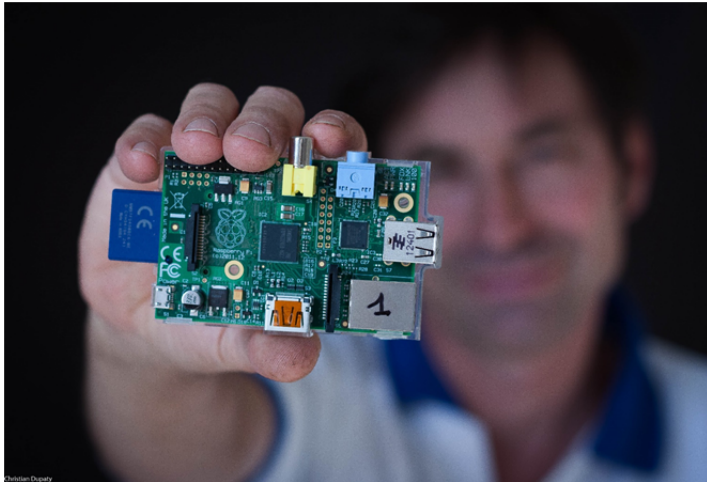


RASPBERRY PI

INSTALLATION-CONFIGURATION

INTERFACES DE COMMUNICATIONS



1 WIRE

Christian Dupaty

BTS Systèmes Numériques

Lycée Fourcade - Gardanne

Académie d'Aix-Marseille

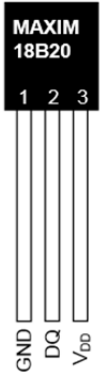


1) TP : 1 WIRE : DS18B20

1 WIRE est une liaison série asynchrone half-duplex avec possibilité de transfert de l'alimentation sur la ligne de donnée développée par Dallas-Semiconductors. Cette liaison série est donc très économique en terme de connexion.

Le bus 1 WIRE : <http://www.maximintegrated.com/products/1-wire/flash/overview/index.cfm>

La ligne DATA du capteur est connectée sur **GPIO4** avec pull-up VDD 3.3v (4.7KΩ)
voir : <http://www.maximintegrated.com/datasheet/index.mvp/id/2812>



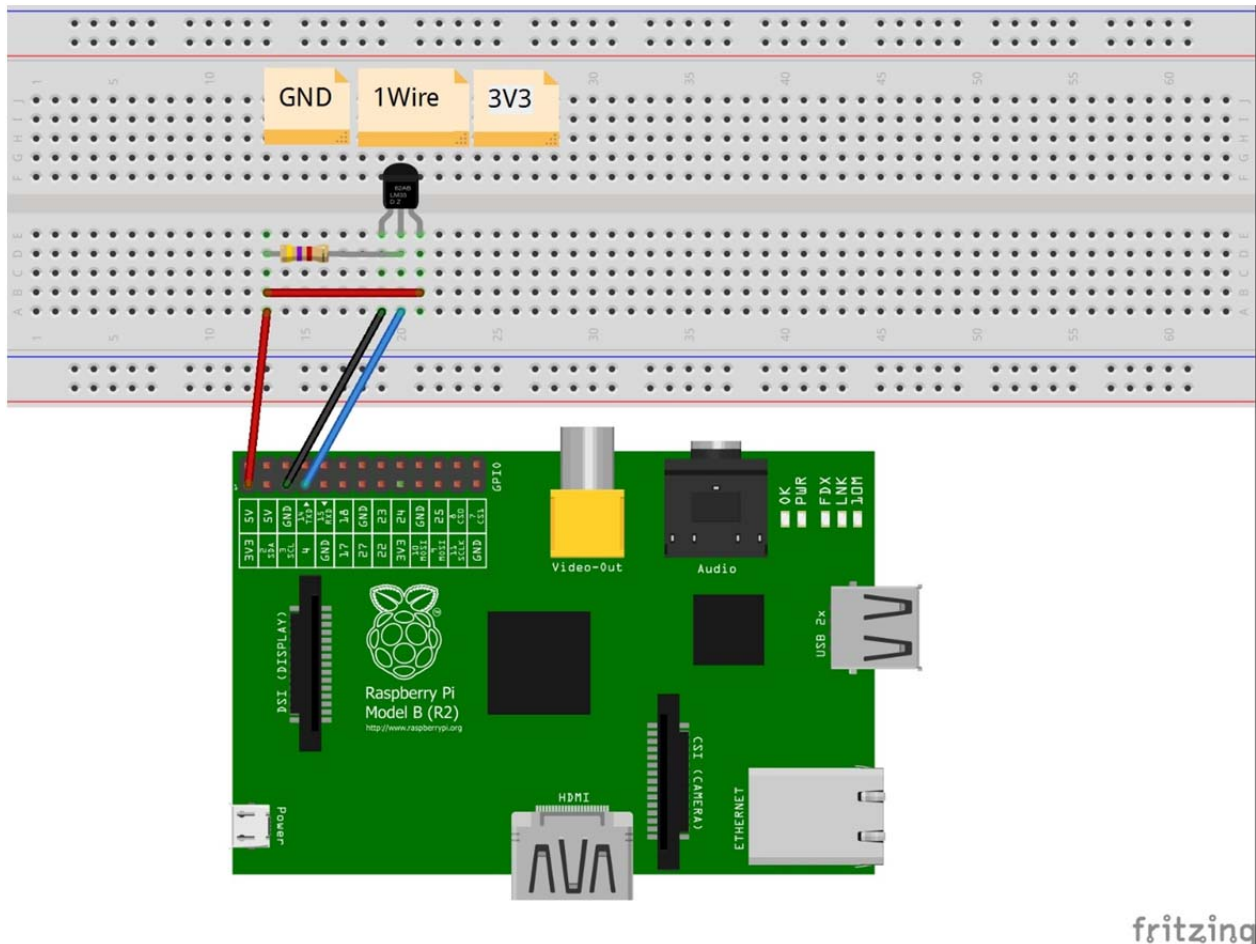
Module w1

<https://www.kernel.org/doc/Documentation/w1/w1.generic>

Le maître (Raspberry Pi) recherche périodiquement de nouveaux esclaves.

Lorsqu'un esclave est trouvé, w1 cherche son driver, si aucun driver n'est trouvé un driver par défaut est chargé.

La famille des drivers pour capteurs de température est w1_therm pour les mémoires : w1_smem.





Test en mode 'normal', attention le capteur DS18B20-PAR ne fonctionne qu'en mode 'parasites', il faut utiliser ici un capteur DS18B20 :

```
sudo modprobe wl-gpio
sudo modprobe wl-therm
```

Pour charger ces modules automatiquement à chaque démarrage :

Ajouter au fichier /etc/modules les lignes

```
wl-gpio
wl-therm
```

```
cd /sys/bus/w1/devices
```

```
ls
```

```
cd 28-xxxxxxxxxxx
```

```
cat w1_slave
```

Cette dernière commande affiche la dernière lecture des registres (SCRATCHPAD) du DS18B20 (format little endian, poids faibles en premiers)

```
pi@raspberrypi: /sys/bus/w1/devices/28-000005a303c4
pi@raspberrypi ~/python $ cd /sys/bus/w1/devices
pi@raspberrypi /sys/bus/w1/devices $ ls
28-000005a303c4 w1_bus_master1
pi@raspberrypi /sys/bus/w1/devices $ cd 28-000005a303c4
pi@raspberrypi /sys/bus/w1/devices/28-000005a303c4 $ ls
driver id name power subsystem uevent w1_slave
pi@raspberrypi /sys/bus/w1/devices/28-000005a303c4 $ cat w1_slave
ba 01 4b 46 7f ff 06 10 d0 : crc=d0 YES
ba 01 4b 46 7f ff 06 10 d0 t=27625
pi@raspberrypi /sys/bus/w1/devices/28-000005a303c4 $ cat w1_slave
b8 01 4b 46 7f ff 08 10 8a : crc=8a YES
b8 01 4b 46 7f ff 08 10 8a t=27500
pi@raspberrypi /sys/bus/w1/devices/28-000005a303c4 $
```

01B8 (pf puis PF): température sur 12bits , résolution 0,0625°C : 01B8=440 : 440x0.0625=27,5 °C

4B46 (PF puis pf): température d'alarme

7F : mesure sur 12bits (Ob01111111)

FF0810 : réservés par MAXIM-IC

8A : CRC

```
# DEMO LECTURE DS18B20 MAXIM-IC
#!/usr/bin/python
import os      # commandes linux (modprobe)
import glob   # recherche de chemins (path)
import time   # pour la boucle principale

os.system('modprobe wl-gpio')    # active le protocole 1WIRE
os.system('modprobe wl-therm')

base_dir = '/sys/bus/w1/devices/' # recherche le numero du capteur
device_folder = glob.glob(base_dir + '28*')[0]
device_file = device_folder + '/w1_slave'

def read_temp_raw():              # lecture
    f = open(device_file, 'r')
    lines = f.readlines()
    f.close()
    return lines

def read_temp():                  # extraction de la temperature
    lines = read_temp_raw()
    while lines[0].strip()[-3:] != 'YES':
        time.sleep(0.2)
```



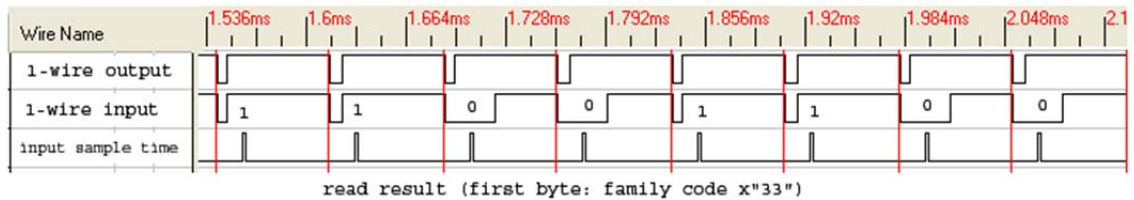
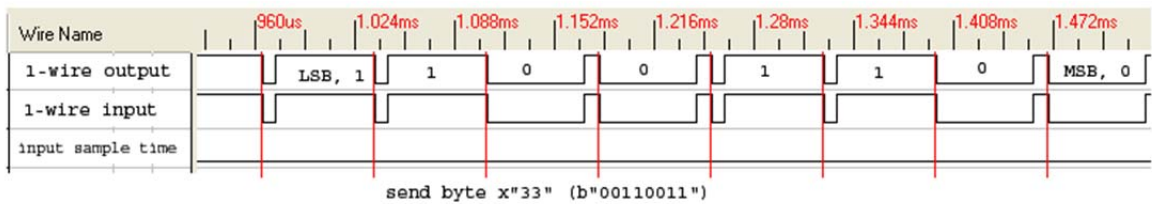
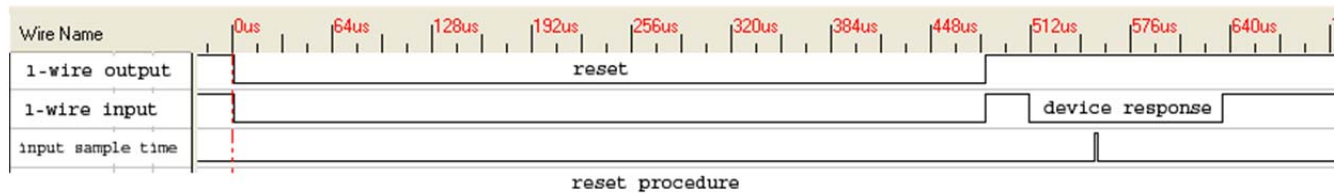
```

lines = read_temp_raw()
equals_pos = lines[1].find('t=')
if equals_pos != -1:
    temp_string = lines[1][equals_pos+2:]
    temp_c = float(temp_string) / 1000.0
return temp_c

while True:
    print('Temperature sur DS18B20 : %f degC' %read_temp())
    time.sleep(1)

```

1 Wire reset, write and read example with DS2432



© WIKIPEDIA