

Lire une sonde BMP085 avec un Raspberry Pi

By Yaug

20 avril
2013

Bonjour à tous,

On continue dans la séries des tutos pour lire des sondes diverses et variée avec votre Raspberry Pi adoré. On va encore partir sur un grand classique : le capteur de pression / température BPM085. Histoire de compléter votre éventuelle station météo maison.

Présentation de la sonde

Bon, ça commence à bien faire tous ces codes ?

C'est quoi encore cette sonde ?

C'est la BMP085 de Bosh, elle permet de mesurer assez précisément température mais surtout pression ! Et, par extension, peut devenir une sorte d'altimètre grâce aux différences de pression avec l'altitude.

Si on lit la description complète ([datasheet](#)), on remarque ces caractéristiques :

- Mesure la pression de 300hPa à 1100hPa (soit de 9000m d'altitude à 500 sous le niveau de la mer)
- Accepte un tension d'entrée de 1,6 à 3,6V
- Consommation de 5 μ A en lecture (toute les secondes).
- Interface I2C
- Lecture de la température entre -40 et 85°C (précision maximale entre 0 et 60°C)
- Résolution de 0.01 hPa et 0.1°C

Précision de 0.2 à 0.5 hPa et de 0.5 à 1°C (en fonction de la température)

Matériel nécessaire

- Un raspberry pi
- Une sonde BMP085
- Quelques câbles de connexion
- Une planche de prototypage

Notez que j'ai reçu un BMP085 sans pins (ou avec les pins non soudés), j'ai donc du trouver des pins et tout un ensemble pour souder (mais bon je comptais m'en resservir de toute manière). Donc attention à ce que vous commandez, sinon vous serez comme moi, 3 semaines à attendre la livraison de votre fer à souder importé de pétaouchnok, sans pouvoir vous servir de votre sonde. C'est un peu con.... donc à prévoir !

Le montage

Hop, la flemme de faire le schéma moi même. De toute manière il est extrêmement simple. On branche la sonde, et même pas de résistance à placer. Donc ... c'est super simple !

On connecte ainsi :

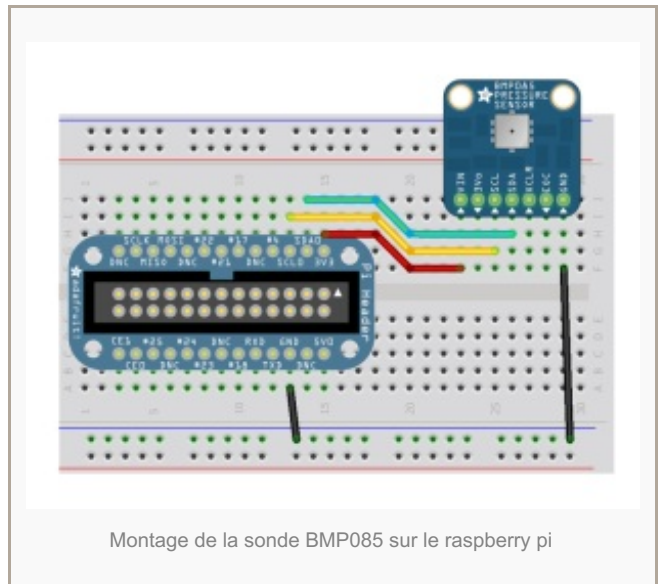
- On alimente en 3V3 le pin VIN

- On connecte le Ground du Raspberry Pi à celui de la sonde
- On utilise le pin SDA (SDA0) du Raspberry Pi qu'on connecte au SDA de la sonde
- Même chose pour le SCL (SCL0) du Raspberry Pi qui file vers celui de la sonde.

Sans doute le montage de sonde le plus simple qui existe enfin ça tombe bien, la vraie difficulté est plus loin.

Mise en place de l'I²C

C'est maintenant que les choses sérieuses commencent. Le montage est extrêmement simple, mais le code demande un peu plus d'attention. Non pas qu'il soit complexe, mais on se base ici sur le protocole I²C pour communiquer avec la sonde, il est donc nécessaire de l'activer sur le Raspberry Pi.



Pour cela il faut ouvrir le fichier de module :

```
sudo nano /etc/modules
```

Vérifiez si les 2 lignes suivantes sont déjà dans le fichier, sinon ajoutez les à la fin :

```
i2c-bcm2708
i2c-dev
```

Votre fichier doit alors ressembler à ça :

```

# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.
# Parameters can be specified after the module name.
snd-bcm2835
i2c-bcm2708
i2c-dev

```

Votre fichier de module

Maintenant, il est temps de faire un reboot pour que ces modifications soient prises en compte.

```
sudo reboot
```

Il peut aussi s'avérer nécessaire de faire d'autres modifications (suivant votre version du système par exemple). Ne craignez rien, j'ai dû passer par ces modifications pour réussir à faire fonctionner le protocole I²C.

Avant toute chose, commencez par mettre à jour votre système. (apt-get update, apt-get upgrade).

Ensuite installez le protocole I²C :

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

Selon votre version du système raspbian, vérifiez si le protocole I²C n'est pas blacklisté.

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

Vérifiez si vous avez les 2 lignes suivantes, si oui, alors il faut les commenter (en ajoutant un #). Si non, alors vous n'avez rien à faire.

```
blacklist spi-bcm2708
blacklist i2c-bcm2708
```

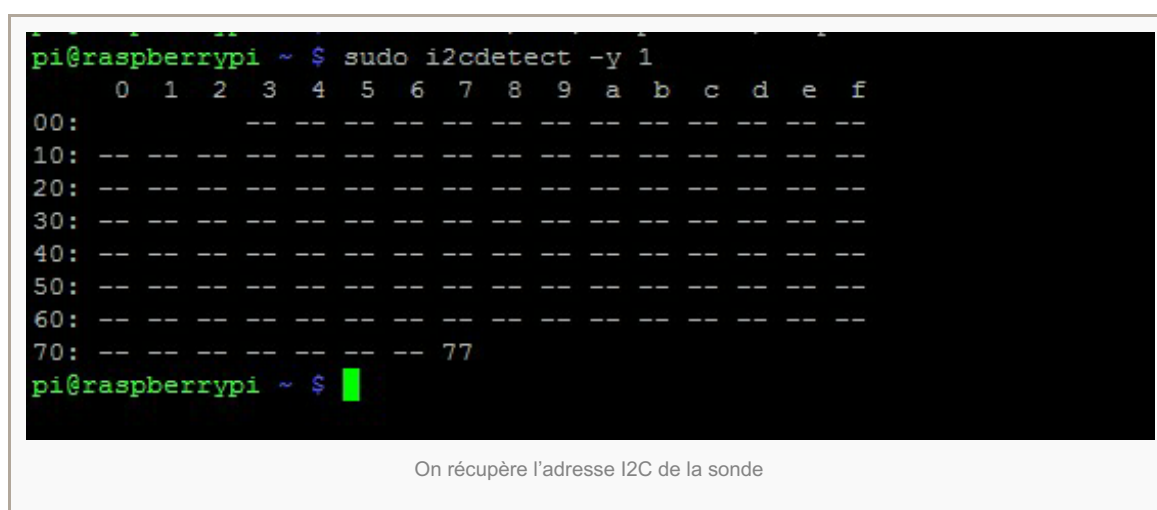
Voilà pour les modifications. Normalement, maintenant tout fonctionne.

Le code

Voilà, normalement, votre protocole I²C est activé et fonctionnel. Tapez la commande suivante pour détecter votre sonde et obtenir sur son adresse (si vous avez un Raspberry Pi modèle A, remplacez le 1 par un 0) :

```
sudo i2cdetect -y 1
```

Vous obtiendrez un écran dans ce genre :



```
pi@raspberrypi ~ $ sudo 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  --  --  --  --  --  --  --
pi@raspberrypi ~ $
```

On récupère l'adresse I2C de la sonde

Dans mon cas (screenshot au dessus), c'est donc l'adresse 77. Gardez l'adresse dans un coin, on peut en avoir besoin. En général, c'est toujours cette adresse qui est utilisée pour ce capteur.

Une fois de plus, on va piller les bibliothèques d'Adafruit (elle a fait un sacré boulot la [Lady Ada](#)). Si vous ne l'avez pas encore fait, importez l'ensemble des bibliothèques adafruit :

```
git clone https://github.com/adafruit/Adafruit-Raspberry-Pi-Python-Code.git
```

Une fois rapportées les bibliothèques, allons dans celle qui nous intéresse :

```
cd Adafruit-Raspberry-Pi-Python-Code
cd Adafruit_BMP085
```

On édite ensuite le fichier **Adafruit_I2C.py** pour mettre les bonnes données.

Si vous utilisez un Raspberry Pi modèle B, modifiez la ligne suivante (ligne 11 normalement) pour obtenir ceci :

```
def __init__(self, address, bus=smbus.SMBus(1), debug=False)
```

Sinon, laissez tel quel !

Il ne vous reste plus qu'à tester le tout en exécutant la commande suivante :

```
sudo python Adafruit_BMP085_example.py
```

Vous obtiendrez un affichage dans ce genre :

```
pi@raspberrypi /var/www/Adafruit-Raspberry-Pi-Python-Code/Adafruit_BMP085 $ sudo python Adafruit_BMP085_example.py
Temperature: 27.10 C
Pressure: 1002.98 hPa
Altitude: 85.19
```

Affichage des données du capteur

Et voilà pour l'utilisation de base. Comme vous pouvez le voir, le calcul de la pression permet de déterminer l'altitude. On se base en effet sur la pression de référence, celle du niveau de la mer qui est de 1013,25hPa.

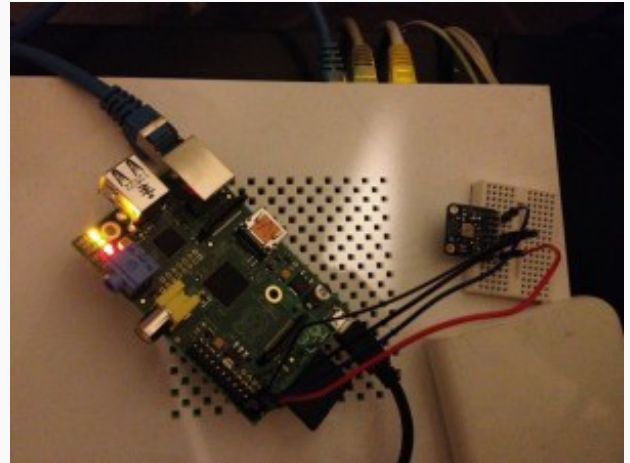
Si vous connaissez l'altitude exacte de l'endroit où vous vous trouvez, vous pouvez adapter le résultat fourni pour calibrer parfaitement votre capteur.

Voilà, on a désormais fini avec la lecture d'un BMP085 en utilisant un Raspberry Pi. Si vous voulez aller plus loin, lisez le datasheet de la sonde, regarder comment fonctionne le code, et si vous en voulez plus, allez faire un tour sur wikipedia, du côté de l'article « [Formule du nivellement barométrique](#) » .

Sinon, avant de partir, voilà à quoi ressemble mon montage :

Et oui, je chauffe mon appartement avec une livebox ! ce qui du coup explique la température relevée plus haut

Voilà, c'est tout. N'hésitez pas si vous avez des questions.



Mon montage du Raspberry Pi et du BMP085

