



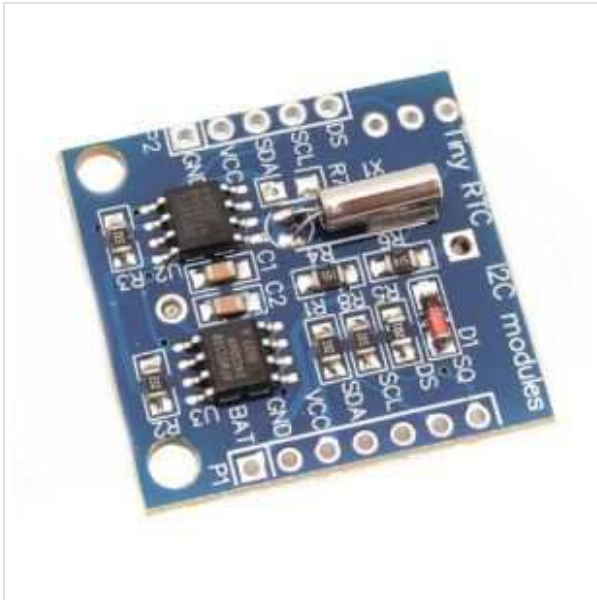
Raspberry Pi : ajouter une horloge RTC en I²C

Dans ce tutoriel, nous allons ajouter un composant particulièrement important et utile pour tout système informatique : une horloge temps-réel, ou Real-Time Clock (que nous appellerons désormais « RTC »). Une RTC permet à un système, et donc notre Raspberry Pi, de toujours être à l'heure, de manière exacte et très précise, sans avoir besoin d'internet. Un module RTC se compose principalement d'un oscillateur à quartz et d'une pile bouton de 3v, qui permettra au module RTC de rester alimenté quand notre Pi sera éteint (et donc de garder l'heure en mémoire). On utilise généralement une RTC dans tout système qui a besoin d'accomplir des tâches dans des timings bien précis (un ordinateur, par exemple).



Nous allons utiliser le module **TinyRC**, un module basé sur le composant DS1307. Ce module fabriqué en Chine présente plusieurs avantages pour nous :

- il est conçu pour arduino mais est compatible Raspberry Pi sans problème,
- facile à câbler et à utiliser,
- précis et fiable (le ds1307 est un composant bien connu),
- utilise le bus I²C (« two-wires »), donc chainable avec d'autres modules I²C sur seulement 2 GPIO (117 périphériques maximum sont possibles sur un seul bus I²C),
- il coûte presque rien du tout : 1.50€ hors frais de port sur [eBay](#) (la pile est à rajouter).



Remarques : au sujet du port P2

et du socket U1 que l'on peut voir sur les photos : notre module semble être prévu pour recevoir un capteur de température DS18B20 optionnel (à cabler dans U1). Nous ne l'utiliserons pas au cours de ce tutoriel, afin de rester compatible avec tous les autres modèles de modules RTC I²C. On peut noter que notre module TinyRC comporte également une puce AT24C32 d'Atmel, qui est une EEPROM de 32kb (une sorte de petite mémoire). Nous ne l'utiliserons pas non plus pour le moment. Voici donc notre checklist :

1. Branchement
2. Activation et configuration du module logiciel I²C
3. Test de l'interface I²C
4. Installation et configuration du module RTC
5. Démarrage automatique

• Branchement

Pour le moment rien de bien compliqué, on pourrait même difficilement faire plus simple, c'est un avantage du bus I²C qui communique sur 2 fils (hors alimentation):

- connectez VCC à une broche +5v du Pi
- connectez GND à une broche GND du Pi
- connectez SDA à la broche SDA0 du Pi
- connectez SCL à la broche SCL0 du Pi

Si vous n'avez pas de pile bouton de 3V sous la main (ce qui est mon cas lors de ce tuto), connectez également 2 piles de 1.5V entre les bornes BAT et GND du module RTC. Sinon, installez la pile bouton dans son logement.

• Activation et configuration du module logiciel I²C

On ajoute les modules au démarrage : **sudo nano /etc/modules** Ajouter ces 2 lignes :

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

On redémarre : **sudo reboot** On installe les outils nécessaires : **sudo apt-get install python-smbus i2c-tools** On supprime les modules I²C de la blacklist (si c'est le cas) : **sudo nano /etc/modprobe.d/raspi-blacklist.conf** les lignes

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

deviennent

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

On redémarre, et le Raspi devrait être prêt à utiliser les périphériques I²C.

• Test de l'interface I²C

On va utiliser la commande **i2cdetect**, qui permet de voir la liste des périphériques I²C reliés au Pi. La commande est différente si vous utilisez un Raspberry Pi Rev 1 ou Rev 2 (l'adresse du bus I²C est différente), donc choisissez la bonne : **sudo i2cdetect -y 0** (pour la Rev 1) **sudo i2cdetect -y 1** (pour la Rev 2)



```
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: 50  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  UU  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
pi@raspberrypi ~ $
```

Si comme sur cette capture, vous voyez votre RTC avec l'ID #68 (il est probable qu'à la place de 'UU', soit indiqué '68', c'est parfaitement normal), tout est correctement câblé et configuré. Dans le cas contraire, il faudra reprendre depuis le début et vérifier chaque étape.

• Installation et configuration du module RTC

Maintenant que notre RTC est correctement câblée et que notre système est prêt à utiliser le bus I²C, nous pouvons passer à la configuration du module. Commençons par lancer le module logiciel RTC : **sudo modprobe rtc-ds1307** On va maintenant activer le module DS1307 dans le module logiciel I²C. Encore une fois, la commande dépend de la version de Raspberry Pi. Ouvrons un bash en tant que root : **sudo bash**
echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-0/new_device (pour la Rev 1)
echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1/new_device (pour la Rev 2)
On ressort du bash : **exit** Nous allons vérifier l'heure de la RTC. En effet, si c'est la première fois que le circuit est utilisé (ou si la pile vient d'être changée), il devrait indiquer le 1er Janvier 2000, et il faudra le mettre à l'heure. **sudo hwclock -r**

```
pi@raspberrypi ~ $ sudo hwclock -r
sam. 01 janv. 2000 01:00:52 CET -0.481737 seconds
pi@raspberrypi ~ $
```

On voit bien sur la capture que le module n'est pas à l'heure... Pour commencer, on va devoir mettre le Raspberry lui-même à l'heure. Le plus simple est de le connecter à internet via câble ou Wifi : l'heure et la date seront ajustés par le réseau. Ensuite on pourra inscrire ces données dans le module RTC. Vérifier l'heure du Raspberry Pi :
date

```
pi@raspberrypi ~ $ date
vendredi 23 août 2013, 02:06:17 (UTC+0200)
pi@raspberrypi ~ $
```

Le Raspi est bien à l'heure Régler l'heure de la RTC, et vérifier : **sudo hwclock -w**
sudo hwclock -r

```
pi@raspberrypi: ~  
pi@raspberrypi ~$ sudo hwclock -w  
pi@raspberrypi ~$ sudo hwclock -r  
ven. 23 août 2013 02:06:55 CEST -0.986051 seconds  
pi@raspberrypi ~$
```

Félicitations, le module RTC est maintenant à l'heure !

- Démarrage automatique

Nous allons maintenant rajouter le module logiciel RTC au fichier **/etc/modules**, afin qu'il soit chargé au démarrage de la machine. **sudo nano /etc/modules** Ajouter à la fin du fichier la ligne :

```
rtc-ds1307
```

Il faut maintenant créer le périphérique DS1307 au démarrage en modifiant le fichier **/etc/rc.local** :

```
sudo nano /etc/rc.local
```

Ajouter, avant la ligne « exit 0 » : pour la Rev 1 :

```
echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-0/new_device  
sudo hwclock -s
```

pour la Rev 2 :

```
echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1/new_device  
sudo hwclock -s
```

Vous pouvez également y rajouter un **hwclock -r** pour vérifier visuellement l'heure du module RTC au démarrage... C'est terminé, désormais votre Raspberry Pi sera constamment à l'heure, qu'il soit connecté à internet ou pas !



Tweeter



Partager 7



submit



E-mail



Posted by: Captain Stouf // Hardware, Les guides, Linux, Raspberry Pi, Software // ds1307, raspberry pi, real time clock, RTC, tutorial, tutoriel // août 23, 2013 [http://hardware-libre.fr/2013/08/raspberry-pi-ajouter-une-horloge-rtc-en-i%C2%B2C/]



About Captain Stouf

Spécialiste en systèmes informatiques, Développeur matériel et logiciel, Inventeur, Maker : électronique, Systems on Chip, micro-controlleurs, Internet of Things, Modélisation / Scan / Impression 3D, Imagerie...

[View all posts by Captain Stouf →](#)

2 thoughts on “Raspberry Pi : ajouter une horloge RTC en I²C”



fred

04/01/2015 at 15 h 17 min

Info: la carte en photo nécessite des accus type LIR2032 (3.6V).

J'ai bêtement mis une CR2032 sans voir qu'il y avait un circuit de charge rudimentaire (D1/R5-R6).

Du coup avec une CD232 faut enlever D1 et R4 (du pont R6/R4) inutile car on est cette fois bien en dessous de 3.5V. On peut aussi mettre une 00hm sur R6.

Pour un système avec alimentation quasi permanente je pense qu'une pile tiendra plus qu'un accus (qui débite en permanence sur le pont).