

Module RTC

Pour notre belle raspberry Pi, nous avons besoin d'un module RTC (Real Time Clock). Il nous permettra d'avoir la date et l'heure exactes sur les photos.

Modèle utilisé :

<http://www.banggood.com/RTC-Real-Time-Clock-Module-For-Raspberry-Pi-p-965976.html>

Pour communiquer avec notre rpi, on utilisera une liaison i2c.

Source :

<http://thepihut.com/blogs/raspberry-pi-tutorials/17209332-adding-a-real-time-clock-to-your-raspberry-pi>

Le branchement :



On utilise les pins suivantes :

- 1 : 3,3v**
- 2 : GPIO2**
- 5 : GPIO3**
- 7 : GPIO4**
- 9 : GND**

Réglage du module RTC

On se connecte en ssh ou même en liaison série sur la rpi et on va exécuter les commandes suivantes :

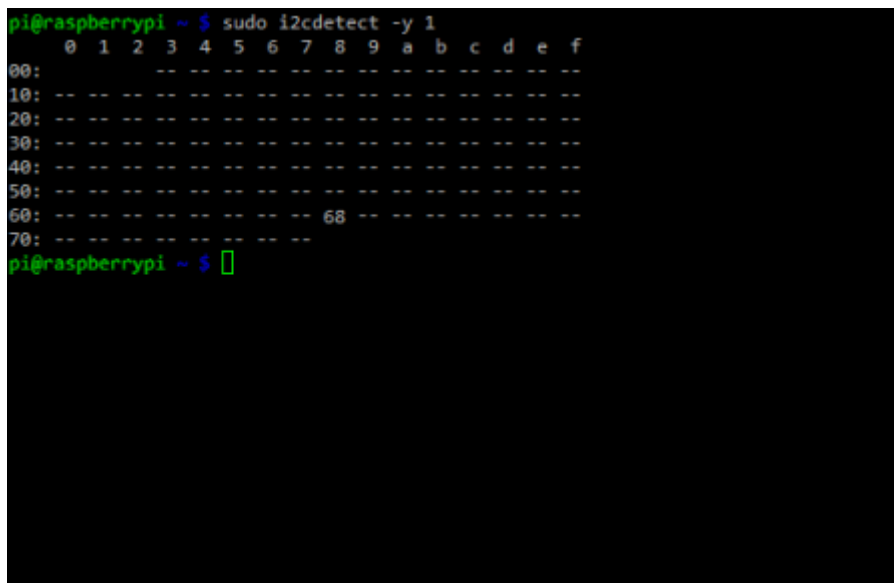
```
sudo i2cdetect -y 1
```

Si en retour vous avez *i2cdetect command not found*, pas de panic il suffit d'installer le paquet manquant :

```
sudo apt-get update
sudo apt-get install i2c-tools
```

Editez le fichier **nano /etc/modules** et rajouter cette ligne : `i2c-dev` Et faire un petit reboot.

La commande **sudo i2cdetect -y 1** doit nous retourner ceci :

A terminal window showing the output of the command 'sudo i2cdetect -y 1'. The output is a grid of dashes representing I2C addresses. The columns are labeled 0 through f. The row 60 shows a '68' at the address 0x68, indicating the presence of the DS1307 RTC. The prompt 'pi@raspberrypi ~ \$' is visible at the bottom.

```
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:  --  --  --  --  --  --  --  68  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
pi@raspberrypi ~ $
```

68 correspond à l'adresse i2c du circuit rtc DS1307.

On charge le module :

```
sudo modprobe rtc-ds1307
```

On avons besoin d'être super utilisateur :

```
sudo bash
```

Pour une Rev.2 Pi ou supérieure :

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

Rev.1 Pi :

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

On tape exit pour ne plus être root

```
exit
```

On vérifie la date, l'heure :

```
hwclock -r
```

On peut se connecter à internet par exemple, il nous donnera la bonne date:

```
date
```

Si la date est bien réglée, il faut la sauvegarder dans le module rtc :

```
sudo hwclock -w
```

On vérifie :

```
sudo hwclock -r
```

Vous l'avez sûrement deviné r : read et w : write.

```
root@raspberrypi:/home/pi# hwclock -r
ven. 26 juin 2015 14:56:07 CEST -0.272248 seconds
root@raspberrypi:/home/pi#
```

Utiliser le module RTC avec la rpi

Nous devons charger le module `rtc-ds1307`. Editons le fichier `sudo nano /etc/modules` et ajoutons la ligne `rtc-ds1307`.

```
GNU nano 2.2.6 Fichier : /etc/modules
# /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
rtc-ds1307
```

Activation au démarrage. Nous éditons le fichier : `sudo nano /etc/rc.local` et on rajoutes les lignes suivantes :

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

dat



Si vous avez une Rev 1 Pi, vous remplacez i2c-1 par i2c-0.

Testons

Vous débranchez votre raspberry du réseau, et vous l'éteignez. Lors du prochain démarrage, la commande date doit lire les données sauvegardées dans votre module RTC :)

From:

<https://wiki.openpathview.fr/> - **Open Path View**

Permanent link:

<https://wiki.openpathview.fr/doku.php?id=rtc>

Last update: **2017/03/12 16:04**

