

Mise en œuvre DS3231MZ+ sur Raspberry Pi2 en I²C



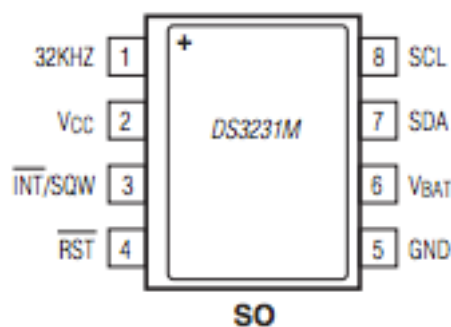
Tutoriel réalisé dans le cadre du projet Surveillance de Niveau de Cuves Cryogéniques, de deuxième année de BTS Systèmes Numériques (SN) option Électronique et Communication (EC).

Sources :

- [Utilisation du Bus I2C sur Raspberry – inovelectronique.fr](http://inovelectronique.fr)
- TP deuxième année SN EC DS1621 : Mesure de température sur BUS I2C
- Manuel I2C
- [Documentation technique DS3231M](#)

Horloge temps réel, DS3231MZ+, composant CMS

Avant toute chose, attardons nous sur la configuration des broches de l'horloge temps réel DS3231MZ+ :



Utilisation de l'I²C sur Raspberry Pi :

Dans un premier temps, vous devez activer l'I²C sur votre carte RPi, pour cela, loggez vous en « root » afin de taper les commandes qui vont suivre.

- Utilisez votre RPi en mode Super User :

```
pi@raspberrypi:~$ sudo su
```

- Éditez le fichier /etc/module avec l'éditeur nano :

```
root@raspberrypi:/home/pi# nano /etc/module
```

- Ajoutez les lignes :

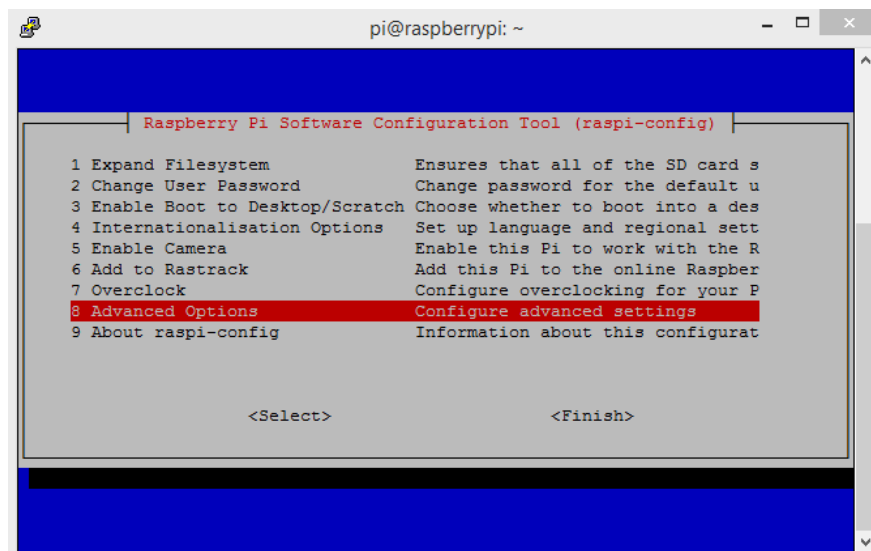
```
i2c-bcm2708
```

```
i2c-dev
```

- Sauvegardez, quittez l'édition de fichier et lancez l'interface de configuration :

```
root@raspberrypi:/home/pi# raspi-config
```

- Entrez maintenant dans le menu « Advanced Options » :



Validez dans ce menu tout ce qui concerne l'I²C.

- Quittez puis redémarrez la carte :

```
root@raspberrypi:/home/pi# reboot
```

- Pour vérifier le bon fonctionnement du BUS I²C sur votre carte, utilisez les deux commandes suivantes :

```
root@raspberrypi:/home/pi# dmesg|grep i2c
```

```
root@raspberrypi:/home/pi# ls /dev/i2c-
```

Le BUS I²C sur votre carte RPi est à présent fonctionnel comme le montre le résultat de la commande précédente.

Installation du DS3231MZ+ et exploitation :

Maintenant que le BUS I2C est exploitable sur votre carte...

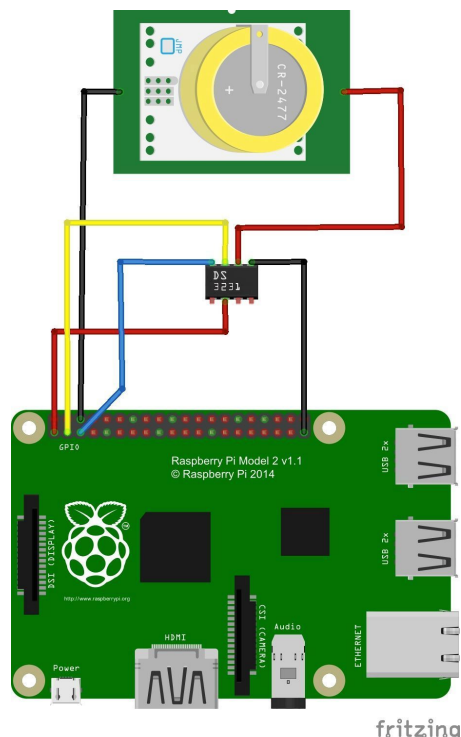
EXPLOITEZ-LE !

Attardons nous toutefois sur le port GPIO de la Raspberry Pi2 :

P1			P5			
<50mA	3V3		1	2	5V	
BCM GPIO00/02	SDA0/1	8	3	4	5V	
BCM GPIO01/03	SCL0/1	9	5	6	GND	
BCM GPIO04		7	7	8	15	TX
	GND		9	10	16	RX
BCM GPIO17		0	11	12	1	PWM0
BCM GPIO21/27		2	13	14		GND
BCM GPIO22		3	15	16	4	
<50mA	3v3		17	18	5	
BCM GPIO10	SPIMOSI	12	19	20		GND
BCM GPIO9	SPIMOSO	13	21	22	6	
BCM GPIO11	SPI SCLK	14	23	24	10	SPI CE0 N
	GND		25	26	11	SPI CE1 N
<50mA	3V3		2	1	5V	
BCM GPIO29	SCL0	18	4	3	17	SDA0
BCM GPIO31		20	6	5	19	
	GND		8	7		GND

Seules les broches entourées en rouge  sont utilisées pour la suite.

Pour avoir un idée de ce à quoi doit ressembler le montage utilisant la RTC, voici le schéma Fritzing qui vous aidera à câbler :



Il est important de noter que sur le logiciel Fritzing la pile seule n'existe pas et est indépendante de la carte sur laquelle elle se trouve en haut du schéma. Sur votre montage, pluggez donc simplement la pile sur une breadboard.

En ce qui concerne le DS3231, ne disposant de ce composant qu'en version CMS, un typon de test a été créé afin de l'exploiter. À vous de voir si vous faites de même ou si vous utilisez le composant dans sa version non-CMS.

Exploitation du DS3231MZ+ :

Nous arrivons à la phase de mise en œuvre de la RTC.

• Une fois le montage réalisé, vous devrez vous assurer que le circuit est détecté par la carte (toujours en mode de fonctionnement Super User) :

```
root@raspberrypi:/home/pi# apt-get install i2c-tools
```

Le résultat obtenu est les installations faites sur votre RPi.

```
root@raspberrypi:/home/pi# i2cdetect -y 1
```

Si la commande aboutit sur ce résultat c'est que le DS3231MZ+ est détecté et que le BUS I2C est exploitable sur votre carte :

	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:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Le caractère ASCII **68** correspond à l'adresse du composant, c'est donc bien l'horloge temps réel DS3231MZ+ qui est détectée.

On trouve dans la documentation de la RTC l'adresse 0b1101000 ce qui correspond en hexadécimal au caractère 0x68 que nous obtenons.

Afin de vérifier maintenant le bon fonctionnement du DS3231MZ+, suivez les instructions suivantes qui vont interroger le capteur de température interne de la RTC qui permet de retourner sa propre température.

• Le résultat de la conversion est accessible avec l'adresse 0x11 sur deux octets, on a donc besoin de renseigner dans la commande le numéro du BUS (ici **1**), le numéro et l'esclave (ici **0x68**) et le registre à lire ou écrire (ici **w** pour écrire) :

```
root@raspberrypi:/home/pi# i2cget -y 1 0x68 0x11 w
```

Le résultat de cette commande est donc une valeur hexadécimale composée de deux octets.

Vous pouvez essayer de faire évoluer la température du composant et relancer la commande pour vérifier que le résultat diffère.

Il est temps maintenant d'initialiser correctement la RTC, pour cela, il faut écrire la valeur 0x04 dans le registre « control » qui est le 0x0E ainsi que la valeur 0x08 dans le registre « status » qui est le 0x0F :

```
root@raspberrypi:/home/pi# i2cset -y 1 0x68 0x0E 0x04 b
root@raspberrypi:/home/pi# i2cget -y 1 0x68 0x0F b
0x88
root@raspberrypi:/home/pi# i2cset -y 1 0x68 0x0F 0x08 b
root@raspberrypi:/home/pi# i2cget -y 1 0x68 0x0F b
0x08
```

À présent vous pouvez lire les secondes, minutes, heures, jour, mois et année en questionnant le bon registre, registres allant de 0x01 à 0x06 et correspondant à chacune des valeurs ci-dessus dans l'ordre.

Par exemple la commande...

```
root@raspberrypi:/home/pi# i2cget -y 1 0x68 0x06 b
```

...va permettre d'obtenir la valeur hexadécimale correspondant à l'année.

Enfin, vous pouvez à présent installer l'horloge temps réel DS3231MZ+ sur votre carte Raspberry Pi. Les quelques commandes qui suivent vont vous permettre de récupérer la date et l'heure sur la RPi qui par défaut dispose de ces données au démarrage à condition d'avoir une connexion internet. Ce système permet donc de se passer éventuellement de cette connexion ou de garantir l'accès à ces données même en cas de panne :

```
root@raspberrypi:/home/pi# echo ds1307 0x68 >
/sys/class/i2c-adapter/i2c-1/new_device
```

(Notez que dans la commande figure la référence du composant DS1307 qui est une autre horloge temps réel mais qui est compatible avec notre DS3231MZ+)

```
root@raspberrypi:/home/pi# hwclock -r
root@raspberrypi:/home/pi# date
root@raspberrypi:/home/pi# hwclock -w
root@raspberrypi:/home/pi# hwclock -r
```

Vous venez d'écrire sur votre RPi les valeurs temporelles.

Détaillons les lignes de commande :

- Déclaration d'un nouvel esclave I2C
- Lecture des données temporelles de la RTC
- Vérification des données de temps
- Écriture de l'heure réelle dans l'horloge
- Vérification du bon fonctionnement de l'écriture.

Afin de tester si le système fonctionne bien, redémarrez votre carte sans que celle-ci soit connectée au réseau et assurez vous que les données de temps et de date correspondent bien.