Projet : Système de Base d'un Power Saving via Module RTC DS3231

Introduction :

Le but du projet est d'économiser le plus possible l'énergie d'un système basé sur un microcontrôleur (MCU) fonctionnant sur batterie.

Beaucoup de projets n'ont pas besoin d'être en mode actif en permanence (Ex : Horodateur, capteur etc.).
Les microcontrôleurs possèdent pour cela plusieurs modes de mise en veille.

Un évènement interne ou externe réveillera le MCU périodiquement, il effectuera les actions nécessaires en fonction du projet puis une instruction permettra de le reconfigurer dans son état de veille etc.

Il existe plusieurs façons de sortir un MCU d'un mode veille, pour ce système de base de Power Saving nous utiliserons un signal externe.
Notre microcontrôleur utilisera le mode veille profonde (**Power-down**), une impulsion externe déclenchera une **interruption** qui réveillera notre MCU.

J'essaye toujours de faire un premier prototype avec le matériel dont je dispose.
Nous utiliserons donc une carte **Arduino Pro Mini 5v/16 MHz** modifiée : la LED témoin de l'alimentation et le régulateur de tension 5v seront retirés du circuit.
La carte sera directement alimentée par une batterie dont la tension sera proche de 5v sur la broche Vcc par exemple : 3 x 1.5v.

Sur les cartes Arduino Pro Mini les broches **D2** et **D3** sont les entrées pour déclencher une interruption.

J'ai fait et testé beaucoup de systèmes de Power Saving (Voir ma page FB ou YT), je pense qu'il n'y a pas de solution universelle, tout dépend de l'utilité des projets fonctionnant sur batterie.

Cette fois j'avais en tête d'utiliser **un module RTC (Real Time Clock)** pour réveiller le MCU périodiquement ou à une certaine heure de la journée, de la semaine ou du mois.

Certains modules RTC possèdent 2 "**alarmes**" qui peuvent être programmées.
Quand les heures et/ou les minutes et/ou les secondes et/ou la date courantes correspondent aux valeurs programmées dans le module RTC, celui-ci peut entre autre générer une impulsion sur une sortie généralement appelée **SQW**.
C'est cette impulsion qui réveillera notre MCU.

Problème :

Toujours dans le but de faire un premier prototype avec le matériel dont je dispose, j'ai donc utilisé un **module RTC DS3231** disposant de 2 alarmes et d'une sortie SQW pouvant être configurée en mode simple impulsion.

La première chose à faire était bien sûr de mesurer la consommation du module.
Et bien sûr pas de chance j'ai mesuré un courant de **4.36mA sous 5v** : évidement c'est beaucoup trop !

J'ai dans un premier temps pensé à me résigner et utiliser un modèle RTC basse consommation, beaucoup de membre des groupes m'ont proposé des références, mais je pars toujours du principe (et j'en ai pas beaucoup !) que tout est possible, j'ai donc cherché sur interne des solutions pour réduire un maximum sa consommation.

J'ai d'abord vu dans des forums que je n'étais pas le seul à chercher des solutions pour ce module RTC, puis après des heures de lecture, je me suis aperçu qu'en réalité je connaissais mal ses composants et les fonctions dont ils disposent.

A force de lecture j'ai comblé une partie de mes lacunes, puis sur un forum je suis tombé sur ce lien :
https://thecavepearlproject.org/2014/05/21/using-a-cheap-3-ds3231-rtc-at24c32-eeprom-from-ebay/

Solutions :

Ce site est une mine d'informations, n'hésitez pas à cliquer sur tous les liens de la page.
Au travers d'un projet nommé "The Cave Pearl", qui est un data logger submersible, l'auteur de l'article propose une multitudes de solutions pour économiser l'énergie et notamment à cet égard des modifications du module RTC DS3231.

Donc la consommation de mon module RTC d'origine était de **4.36mA**.

- La première solution classique est de supprimer la résistance **R1 de la LED** témoin de la présence de l'alimentation : **3.3mA**.

- La deuxième solution est de supprimer la résistance **R5 faisant partie du système de charge de la pile bouton** au lithium rechargeable. Dans ce cas remplacez la pile bouton par une **CR2032** de bonne qualité : **120µA (consommation avec la carte Arduino incluse)**.

- La troisième solution est de **débrancher la broche Vcc** du module pour qu'il fonctionne avec sa tension **Vbat** :
Pour cela il faudra supprimer **le réseau de 4 résistances de tirage RP1**.
Ce sont les résistances de tirage pour les signaux : **SQW, SDA, SCL, 32K**.
La broche **32K** ne sera pas utilisée.

Les résistances de tirage internes du MCU pour le bus **I2C** (SDA et SCL) sont déjà activées dans la librairie **wire.h** dans le fichier **twi.c** :

```
// activate internal pullups for twi.
digitalWrite(SDA, 1);
digitalWrite(SCL, 1);
```

Dans le programme il faudra ajouter :

#define INTERRUPT_PIN 2
digitalWrite(INTERRUPT_PIN, HIGH); // Active la résistance de tirage de la broche de déclenchement de l'interruption externe (D2) ou (D3).

Il faudra déclencher l'interruption sur front descendant : **FALLING**.

Consommation avec la carte Arduino incluse : 18µA.

Remarque :

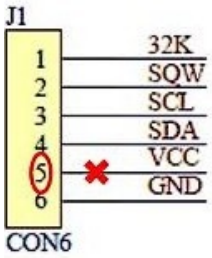
Sur certains modules il faudra modifier le **bit 6 (BBSQW)** du registre interne du module **RTC Control Register** à l'adresse **0x0E** :

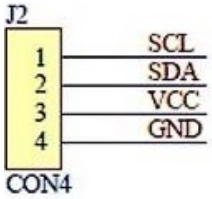
Extrait de la documentation :

Bit 6: Battery-Backed Square-Wave Enable (BBSQW).
Lorsqu'il est réglé sur un 1 logique avec INTCN = 0 et VCC < VPF, ce bit active le signal carré.
Lorsque **BBSQW est sur un 0** logique, la broche INT/SQW passe à une impédance élevée lorsque VCC < VPF.
Ce bit est désactivé (logique 0) lors de la première mise sous tension.

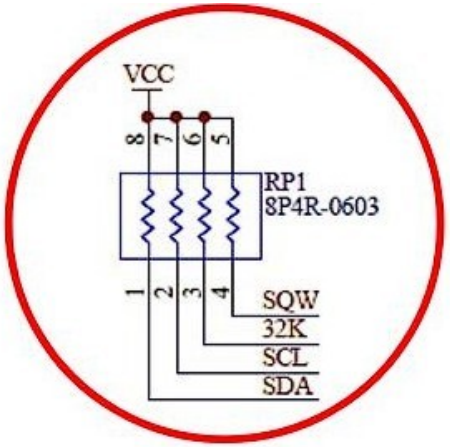
Sur certains modèles si vous débrancher la broche **Vcc** du module, vous devez définir le **bit 6** du registre interne **RTC Control Register** sur un 1 logique pour activer sinon les alarmes ne pourront pas s'activer :
DS3231M BBSQW bit **(if = 1)** enables **SQW** and RTC alarm interrupt when operating on **VBAT** supply.
"(si = 1) active l'interruption d'alarme SQW et RTC lors du fonctionnement sur l'alimentation VBAT)".

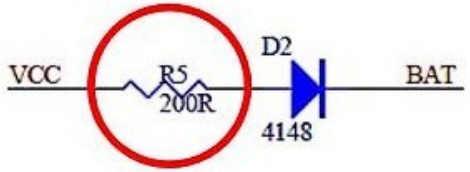
#define RTC_CONTROL 0x0E // Adresse du registre
RTC.writeRTC(RTC_CONTROL, (RTC.readRTC(RTC_CONTROL) | _BV(6))); // Positionne le bit 6 à 1

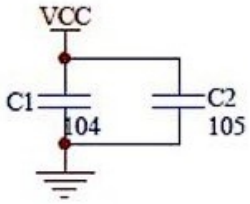


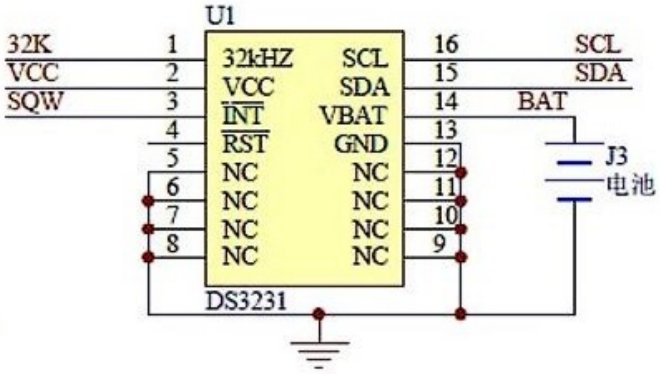


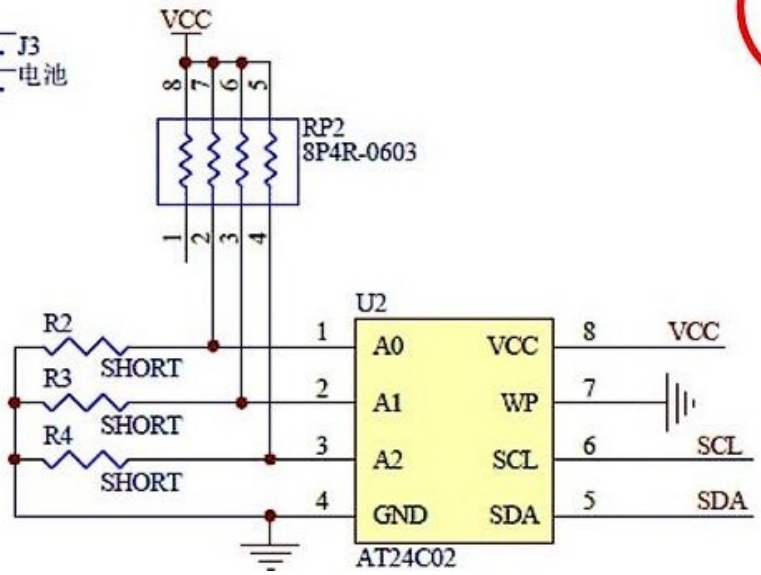
cascade port

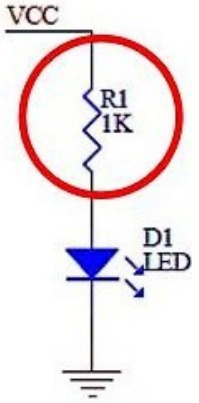












Size	Number
A4	
Date:	31-May-2013
File:	D:\原理图\DS3231\DS3231.ddb