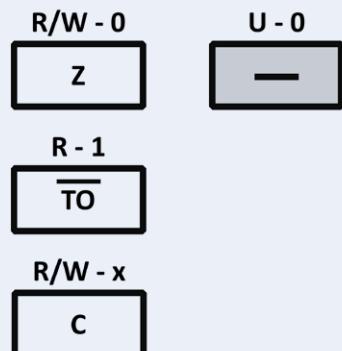


la documentation des microcontrôleurs sans peine (2)

registres et schémas de principe



R – readable
W – writable
U – unimplemented
1 – one after reset
0 – zero after reset
x – unknown value after reset
X – bit value inverted

Figure 1 : La description des bits des registres peut sembler quelque peu mystérieuse. (Source : Microchip Technology)

Stuart Cording (Elektor)

Que vous aimiez ou non la documentation des microcontrôleurs, vous aurez à négocier pour l'utiliser. Dans la première partie de cette série d'articles [1], nous avons examiné le contenu d'une fiche technique de microcontrôleur. Ici, nous allons plus loin en examinant comment les utilisateurs sont informés des fonctionnalités d'un appareil, en abordant tous les aspects, depuis les registres jusqu'aux schémas de principe des périphériques.

Fiche technique : premières pages

Maintenant que vous savez ce que contient une fiche technique, il est temps de commencer à examiner certains détails. Les fiches techniques semblent décrire les choses de manière quelque peu surprenante, et chaque fournisseur crée ses descriptions et ses schémas de différentes manières. Nous allons nous focaliser spécifiquement sur le microcontrôleur 8 bits PIC16F18877 de Microchip Technology [2]. Nous vous conseillons donc de télécharger la fiche technique si vous souhaitez suivre notre démarche.

Après avoir téléchargé et ouvert la fiche technique, nous avons sous les yeux les deux premières pages, qui contiennent une description succincte de ce microcontrôleur ainsi qu'une liste des principales caractéristiques et des périphériques inclus. Les fonctionnalités principales (*core features*) ne sont pas des périphériques autonomes ; il s'agit de capacités étroitement liées au noyau de traitement. On y trouve des fonctionnalités comme un circuit d'arrêt qui détecte le moment où la tension d'alimentation tombe au-dessous du niveau requis, mais aussi les temporiseurs « chien de garde »,

chargés de réinitialiser le microcontrôleur, activés par les systèmes de sécurité critiques.

La troisième page approfondit un peu plus le sujet en expliquant le nombre de chaque périphérique obtenu, et la quantité de mémoire RAM, de mémoire flash de programmation et d'EEPROM mise en œuvre. Viennent s'y ajouter les pages quatre à sept, qui contiennent des informations de base du conditionnement (boîtier) et expliquent les fonctionnalités affectées à chaque broche.

Les tableaux d'attribution des broches (Pin Allocation Tables) constituent un élément essentiel d'une fiche technique de microcontrôleur. Aujourd'hui, les microcontrôleurs ont davantage de fonctionnalités que de broches. Il est donc possible d'affecter une capacité (parfois même plusieurs) à chaque broche. Sur le composant abordé ici, la broche RA2 peut servir d'entrée de CA/N, de référence de tension analogique, d'entrée de comparateur, de sortie de CN/A ou de broche d'interruption sur changement. Pour obtenir une certaine flexibilité, vous pouvez également commuter une capacité spécifique entre une ou plusieurs broches. L'affectation des fonctionnalités aux broches peut donc s'avérer très complexe.

Comprendre les descriptions des registres

Les descriptions des registres se composent de deux parties : le nom du registre et les noms des bits (ou des groupes de bits) de ce

registre. La page 38 de la fiche technique décrit la fonctionnalité du registre STATUS, qui fait partie du processeur. Ce registre est composé de bits qui sont mis en œuvre et d'autres qui ne le sont pas.

Les bits mis en œuvre sont marqués d'une combinaison de mentions R et/ou W pour indiquer s'ils peuvent être lus ou écrits. Le trait d'union est généralement suivi d'un chiffre : « 0 » ou « 1 ». Ces chiffres indiquent la valeur de ce bit après réinitialisation de l'appareil. Certains bits peuvent être marqués d'un x, ce qui indique que leur valeur est inconnue. La valeur de certains bits peut dépendre de la raison pour laquelle le microcontrôleur est sorti de la réinitialisation. Si le microcontrôleur a simplement été mis sous tension, il s'est réactivé par le biais d'une réinitialisation lors de la mise sous tension (POR). Si l'alimentation électrique a provoqué une réinitialisation à la suite d'une panne de courant (BOR), cela peut également se traduire par une valeur différente dans un bit spécifique. Dans cet exemple, ces bits « dépendants de la réinitialisation » sont indiqués par la lettre q.

Les bits non mis en œuvre peuvent être un peu dangereux. Selon cette fiche technique, la lecture des trois bits supérieurs doit aboutir à un « 0 ». Cependant, il n'est pas évident de savoir ce qui survient si vous écrivez dans ces bits. La fiche technique peut comporter une recommandation générale. S'il n'y en a pas, vous devrez probablement vous assurer qu'une opération d'écriture dans ces registres

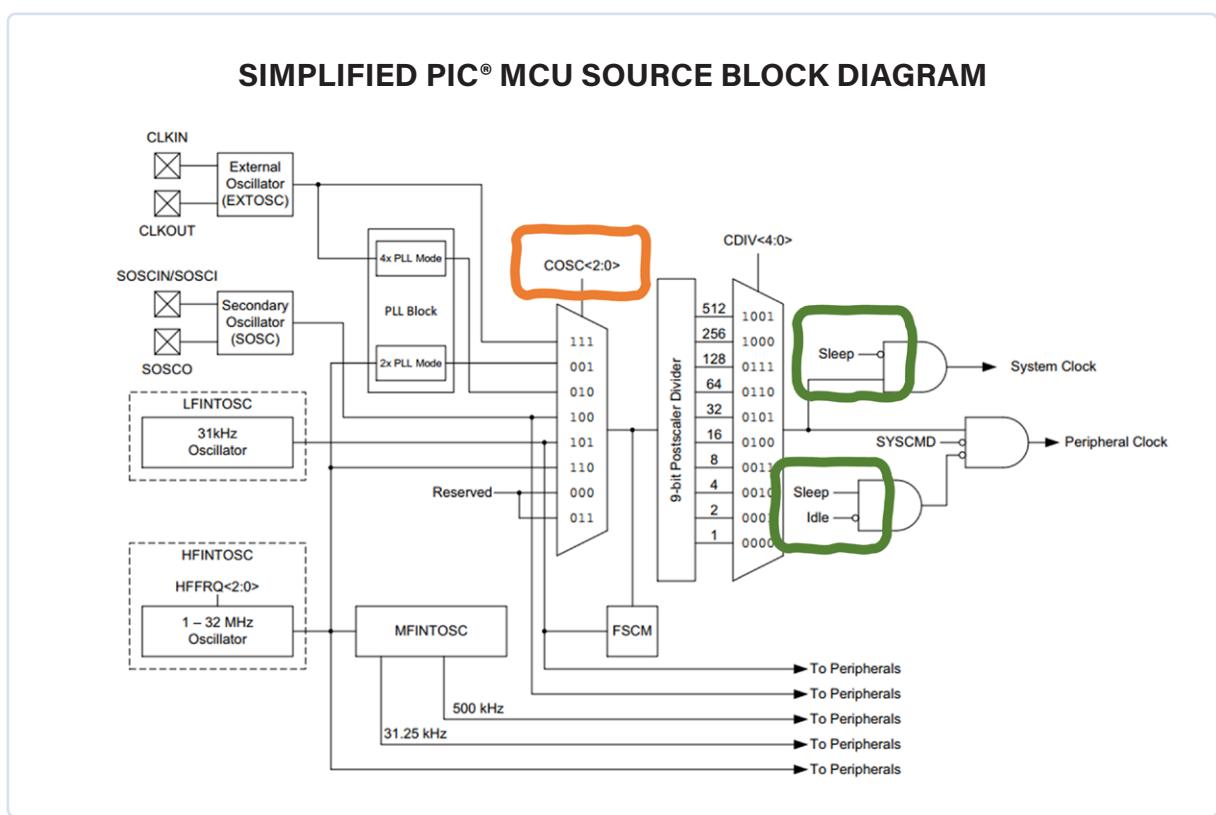


Figure 2. Schéma de principe d'un périphérique d'horloge, qui met en évidence le multiplexeur de sélection d'horloge et les commandes servant à désactiver la source d'horloge. (Source : Microchip Technology)



Bits non implémentés remis à zéro

```
myValue &= 0b00011111;      // clear 3 most-significant bits
                           // (unimplemented bits)
STATUS = myValue;          // write 'myValue' into STATUS register
```

garantit que les bits non mis en œuvre sont remis à zéro. En langage C/C++, cela peut se faire comme suit :

```
myValue &= 0b00011111; // clear 3 most-significant bits
                           // = (unimplemented bits)
STATUS = myValue;        // write 'myValue' into
                           // STATUS register
```

Remarque : il est judicieux d'insérer un commentaire expliquant la raison de cette démarche. Ainsi, personne n'essaiera ultérieurement d'« optimiser » la précaution prise !

La barre au-dessus du nom d'un bit indique que sa valeur est inversée. C'est le cas du bit « Time Out », TO. En cas de dépassement de délai, que l'on pourrait s'attendre à voir signalé par une réponse positive « 1 », la réponse sera en fait « 0 ». Souvent, ces bits inversés n'ont que peu de sens pour le programmeur ; vous devez simplement vous en accommoder et les traiter de manière appropriée dans votre code.

Parfois, un groupe de bits est nécessaire pour configurer quelque chose, par exemple un prédiviseur (ou prescaler) d'une source d'horloge ou la vitesse de transmission d'une interface série. Ces groupes de bits sont alors appelés **BIT_GROUP<X:Y>**, où **X** désigne le bit le plus significatif et **Y** le bit le moins significatif du groupe. Attention également : il arrive qu'un ou plusieurs bits appartenant à un même groupe soient répartis dans différentes zones d'un même registre, voire dans plusieurs registres !

Priorité à l'horloge

La partie la plus importante du microcontrôleur est son périphérique d'horloge. Il s'agit ici de définir la source d'horloge pour le

processeur et les périphériques intégrés. Il est donc judicieux de prendre le temps de revoir la structure du système, comme indiqué page 110. Normalement, ce périphérique est configuré au démarrage du code de l'application et reste inchangé ultérieurement. S'il est modifié par la suite, tous les périphériques peuvent être affectés, depuis la fréquence d'échantillonnage du CA/N jusqu'à la vitesse de transmission des interfaces série UART et CAN.

Dans cet exemple, nous avons des sources externes (oscillateurs à quartz ou à résonateur céramique) et des sources internes. Il est possible que ces sources internes ne soient pas suffisamment précises (en particulier avec les changements de température) pour servir de source fiable pour certaines interfaces, l'UART par exemple. La fiche technique le mentionne probablement, faute de quoi, vous devrez vérifier la précision dans la section relative aux caractéristiques électriques.

Dans le diagramme, nous pouvons voir qu'un groupe de trois bits, **COSC<2:0>**, surligné en orange et positionné dans un registre très probablement lié à l'oscillateur, peut servir à sélectionner différentes sources d'horloge disponibles à l'aide d'un multiplexeur. Pour des raisons indéterminées, seul le groupe de bits est nommé dans ces diagrammes et non le registre auquel ils appartiennent. Le meilleur moyen de savoir à quel registre appartient le groupe est de consulter la fiche technique au format PDF – il peut en effet se trouver dans un registre d'un autre périphérique ! Les diagrammes multiplexeurs sont fréquemment utilisés dans les schémas fonctionnels où les signaux peuvent être commutés.

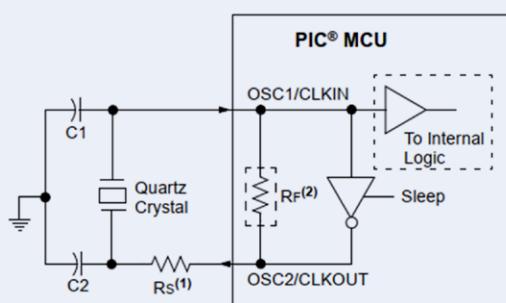
La sortie de ce bloc fournit l'horloge système (*System Clock*), probablement au processeur, et l'horloge périphérique (*Peripheral Clock*) pour les périphériques. La fiche technique peut faire référence à ces deux termes dans le reste du document, en particulier lorsqu'elle aborde les modes Basse consommation (*Low Power*) ou Veille (*Sleep*). D'après les sections marquées en vert, nous pouvons déjà déduire qu'il existe un mode Inactif (*Idle*) qui supprime l'horloge des périphériques uniquement, et un mode Veille (*Sleep*) qui supprime l'horloge des périphériques et du processeur.

Si un quartz ou un résonateur en céramique doit être utilisé, il est probable que des indications soient fournies sur la façon de construire le circuit et de réaliser le circuit imprimé. C'est également le cas ici (page 112), où deux condensateurs sont nécessaires, ainsi que, facultativement, une résistance en série. Quatre notes d'application relatives à la conception d'oscillateurs accompagnent ce schéma fonctionnel, ce qui souligne l'attention que nécessite la mise en œuvre des oscillateurs à quartz.

Au-delà des registres et des schémas de principe

Nous avons abordé les bases de la description des registres et la compréhension du schéma de principe du périphérique d'horloge.

Figure 3. Circuit recommandé pour un oscillateur à base de cristal de quartz. (Source : Microchip Technology)



Nous avons aussi jeté un coup d'œil à l'un des oscillateurs et à l'exemple de schéma de circuit pour son utilisation avec un cristal de quartz. Dans la prochaine partie de cette série, nous allons examiner un autre bloc périphérique clé qu'il faut maîtriser : le bloc de réinitialisation. Nous établirons également tout ce que la fiche technique n'aborde pas. Si vous recherchez des ressources pour accéder plus facilement à l'univers des microcontrôleurs, n'hésitez pas à consulter nos articles d'introduction [3] et nos livres. ↵

VF : Pascal Godart — 230101-04



Produits

- Livre en anglais : T. Hanna, *Microcontroller Basics with PIC*, Elektor 2020 (SKU 19188)
www.elektor.fr/19188
- Livre en anglais : A. Pratt, *Programming the Finite State Machine*, Elektor 2020 (SKU 19327)
www.elektor.fr/19327

Des questions, des commentaires ?

Envoyez un courrier à l'auteur (stuart.cording@elektor.com).

LIENS

- [1] Stuart Cording, « la documentation des microcontrôleurs sans peine (1) », [elektormagazine.fr](https://www.elektormagazine.fr/news/la-documentation-des-microcontroleurs-sans-peine-1ere-partie-la-structure-dune-fiche-technique) : <https://www.elektormagazine.fr/news/la-documentation-des-microcontroleurs-sans-peine-1ere-partie-la-structure-dune-fiche-technique>
- [2] microcontrôleur 8 bits PIC16F18877 : <https://microchip.com/wwwproducts/en/PIC16F18877>
- [3] Tam Hanna, « programmer les PIC – à petits pas », [elektormagazine.fr](https://www.elektormagazine.fr/articles/programmer-les-pic---petits-pas) : <https://www.elektormagazine.fr/articles/programmer-les-pic---petits-pas>

MagPi, le magazine officiel du Raspberry Pi



6 x MagPi :
Édition
imprimée



Accès aux
archives en
ligne du MagPi

12 mois
Plus de
100 projets
Le prix
54,95 €



COMMANDÉZ DÈS MAINTENANT AU
WWW.MAGPI.FR/ABO

MagPi
www.magpi.fr Magazine