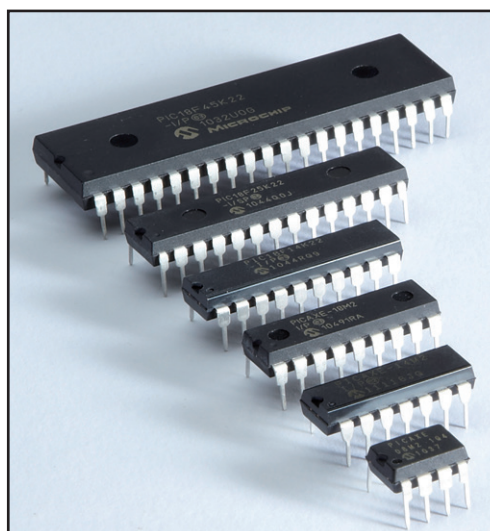


# PIC atout AXE

## Jouez la simplicité PICAXE

Si le micrologiciel d'un microcontrôleur est une page blanche ouverte à l'imagination de l'utilisateur, encore faut-il que cette liberté ne soit pas entravée par un manque de temps ou de connaissances techniques. Le système PICAXE a justement été créé pour que tout un chacun puisse programmer un microcontrôleur et l'interfacer avec des périphériques de façon simple, rapide, et à moindre coût.



**Wouter Spruit**  
(Pays-Bas)

Figure 1.  
La famille PICAXE

### La famille PICAXE

Les PICAXE sont des microcontrôleurs de la famille des PIC de *Microchip* qui ont été programmés avec un micrologiciel spécial développé par *Revolution Education* [1]. D'après les auteurs du système PICAXE, la meilleure façon d'apprendre à programmer un microcontrôleur est d'utiliser un système bon marché, qui ne requiert pas une grande expérience, mais offre néanmoins suffisamment d'interfaces complexes pour pouvoir être incorporé dans un projet avancé. La gamme des PICAXE comprend des microcontrôleurs de différentes dimensions, qui vont de la simple puce à 8 broches à capacités limitées, au modèle avancé à 40 broches (**fig. 1**). Depuis l'apparition des PICAXE il y a plus de 15 ans, les PIC qui formaient le socle historique du système PICAXE ont été remplacés par des modèles plus riches en mémoire et plus puissants en calcul, et même, dans de nombreux cas, dotés de capacités de traitement en parallèle [2].

Le magasin en ligne PICAXE [3] se veut une supérette de l'électronique : à côté des µC sont vendus des plaques d'expérimentation, des composants, des câbles, des outils, des afficheurs LCD... L'ambition du site est en

effet de réunir en un même lieu tout ce qu'il faut pour démarrer immédiatement un projet PICAXE. On y trouve même des kits de montage pour robots à base de PICAXE. Tous les périphériques sont vendus avec une documentation expliquant comment les interfacer avec un microcontrôleur (PICAXE). Se fournir en composants sur ce site peut donc être intéressant, même si l'on n'utilise pas de PICAXE.

L'originalité du système PICAXE vient de son micrologiciel. Une puce PICAXE est un PIC préprogrammé avec un micrologiciel PICAXE, doté d'un interpréteur BASIC et de plusieurs fonctions pré-chargées. Le code PICAXE permet à l'utilisateur de charger un programme BASIC PICAXE à n'importe quel moment via une liaison série (si tant est qu'elle ne soit pas utilisée par le programme en cours d'exécution), le nouveau code étant alors immédiatement exécuté. Puisqu'un programme PICAXE est écrit en BASIC qui est un langage interprété, le nombre d'instructions exécutées par seconde est modeste. Si pour cette même raison la communication série purement logicielle (le *bit banging*) semble impraticable, quelques fonctions d'E/S pré-programmées

permettent tout de même à la puce PICAXE de dialoguer de façon logicielle avec des périphériques I<sup>2</sup>C, ou d'envoyer et d'interpréter des signaux IR. Le véritable avantage du code BASIC interprété réside dans la façon dont les programmes sont chargés dans la puce. L'interface de programmation des PIC ordinaires est en effet complexe et difficile à utiliser, alors que côté PICAXE une simple fiche jack reliée aux broches d'interface série de la puce permet une reprogrammation *in situ* simple et rapide : mettez la puce sous tension, connectez le câble série au PC, et votre programme est prêt à être chargé dans le PICAXE.

Si les programmes peuvent être écrits et chargés rapidement, il n'en reste pas moins que le code est interprété en temps réel. Il est donc préférable de ne pas implanter des opérations mathématiques exigeant des milliers de cycles d'horloge par ligne de calcul, le code serait trop lent pour être utilisable dans une application en temps réel.

### PICAXE vs Arduino

Un programmeur PIC est nécessaire pour charger un programme dans un PIC classique. Même si quelques projets, *el cheapo* [4] p. ex., expliquent comment construire un programmeur PIC à moindre coût, le novice qui souhaite s'initier à la programmation des microcontrôleurs est souvent intimidé par le temps et l'argent qu'il lui faudra apparemment dépenser pour simplement faire clignoter une LED. Même refrain avec les AVR d'Atmel, même si l'apprentissage avec ces AVR peut être facilité par les cartes de prototypage rapide et l'EDI Arduino [5]. Cela dit ces cartes ne battent pas les PICAXE sur le terrain du prix et de la facilité d'implantation dans des projets « réels ».

### Documentation

PICAXE a pour origine un projet pédagogique, et il n'est donc pas étonnant qu'il soit si bien documenté. Le manuel PICAXE est disponible sur le site [6] et accompagne aussi l'éditeur de programmation et l'outil AXEpad. Le manuel est divisé en trois parties. La première liste les capacités de divers PICAXE et leur brochage, contient des instructions pour l'alimentation et la connexion de la puce au PC, ainsi que de nombreux tutoriels de prise en main. La deuxième par-

tie décrit l'ensemble des commandes BASIC et leur syntaxe ; les modèles auxquels s'applique chaque commande sont précisés, et un exemple de code commenté est donné. La troisième partie présente une collection de circuits d'interface servant à relier les E/S d'un PICAXE à différents périphériques, comme des LED, des interrupteurs, ou encore des moteurs. On y trouve également des exemples qui expliquent comment interfacier une puce PICAXE avec divers CI et composants du commerce, des instructions pour relier un afficheur LCD via I<sup>2</sup>C, et également la façon de relier un PICAXE à un PC via une liaison série. Ce qu'il y a de formidable avec ces exemples, c'est que les principes qu'ils mettent en œuvre sont assez généraux pour être applicables à d'autres puces que les PICAXE (en veillant toutefois ce que la tension de service et le courant maximum que peut absorber ou délivrer le microcontrôleur utilisé soient compatibles avec les exemples donnés).

Les racines éducatives du projet PICAXE n'enlèvent rien au potentiel de ces microcontrôleurs. Les PICAXE se montrent en particulier bien adaptés aux projets amateurs. Les utilisateurs sont d'ailleurs encouragés à présenter leurs projets dans la « galerie des projets » du site PICAXE. Et si vous avez des questions sur le système PICAXE, ou sur la façon de l'interfacier avec votre projet, vous pouvez poser vos questions sur le forum de ce même site (il existe un fil francophone).

### Environnement de programmation

Un programme BASIC peut être transféré d'un PC à un microcontrôleur PICAXE via une connexion série. PICAXE propose pour Windows un environnement de développement complet appelé *Programming Editor*. Cet éditeur permet de se connecter rapidement à une puce PICAXE, et aide à l'écriture et au débogage des programmes. L'équivalent pour Linux et Mac s'appelle *AXEpad*. Il offre la plupart des fonctions de l'éditeur pour Windows, dont la modification et le chargement d'un code dans une puce PICAXE, une fenêtre de terminal, et plusieurs assistants pour la production du code. La fenêtre de simulation ligne par ligne de l'exécution d'un programme PICAXE fait partie des fonctions manquantes.



Figure 2.  
Le câble USB et sa fiche jack.

Pour utiliser l'éditeur, servez-vous d'un câble PICAXE USB (**fig. 2**) ou PICAXE *Serial* pour relier votre puce au PC, et dans l'éditeur choisissez le port série utilisé pour la connexion. Sélectionnez ensuite le type de votre puce pour que l'éditeur s'adapte au modèle à programmer. Il existe également une fonction pour déterminer la version du micrologiciel de la puce connectée. Un microcontrôleur connecté se programme par chargement d'un programme BASIC dans la puce directement depuis l'éditeur. Le nouveau programme est exécuté aussitôt après avoir été chargé.

### Hello la LED

Prenons l'exemple classique du clignotement de LED pour illustrer la programmation d'un PICAXE. Nous utilisons un modèle 08M, mais l'exemple vaut pour n'importe quelle puce. Référez-vous au schéma de brochage de votre modèle [7] pour trouver une broche de sortie à relier à une LED.

Nous avons choisi la broche 3 (C.4) (**fig. 3**). Le site *Revolution Education* propose des kits de démarrage qui comprennent des cartes pré-assemblées, mais il est plus intéressant de monter soi-même le circuit sur une plaque d'expérimentation. Les broches dont nous avons besoin sont : +V (1), 0 V (8), entrée série (2, *serial in*), sortie série (7, *serial out*), et la broche de sortie (3) que nous relierons à la LED.

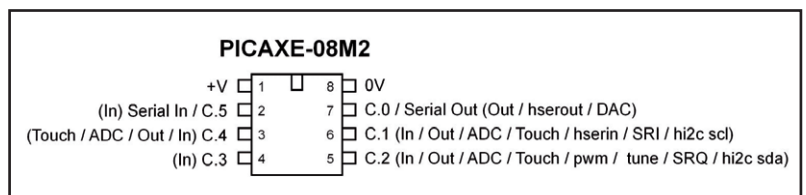


Figure 3.  
Brochage du PICAXE 08M2.

Le schéma de principe est représenté sur la **figure 4**. Les broches d'alimentation +V et 0 V sont à relier à une source de tension comprise entre 3,0 V et 5,0 V. Si votre alimentation est un bloc de piles (acheté p. ex. sur le site de *Revolution Education*), veillez à ne pas utiliser plus de trois piles AA de 1,5 V, ou plus de quatre AA rechargeables de 1,2 V. La **figure 5** montre que le circuit de téléchargement est relié aux broches entrée série, sortie série et 0 V du PICAXE. La LED est reliée à la broche 3 par l'intermédiaire d'une

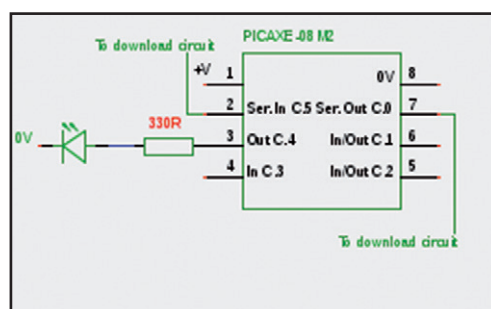


Figure 4.  
Schéma de principe du clignotement de LED.

résistance-talon de 330 Ω. Lorsque le câble de téléchargement n'est pas connecté à un circuit maison, la broche (2) d'entrée série doit être forcée au niveau bas à l'aide d'une résistance de rappel de 33 kΩ reliée au 0 V.

### Programmation du PICAXE

Installez l'éditeur *Programming Editor* si vous êtes sous *Windows*, *AXEpad* si vous êtes sous GNU/Linux ou *Mac* [8]. Les utilisateurs de *Windows* et de *Mac* qui utilisent le câble USB doivent d'abord télécharger et installer le pilote USB selon les instructions de la page [9]. Pour cet exemple, j'ai connecté la fiche jack de 3,5 mm de mon câble RS-232 à un PICAXE 08M monté sur une plaque sans soudure, et j'ai travaillé avec GNU/Linux. Connectez votre câble à votre puce, et attendez quelques secondes avant de lancer l'éditeur. Il faut ensuite indiquer le port COM utilisé. Sous *AXEpad*, allez dans *View/Options/onglet Ports* (même opération si votre câble est USB). Testez la connexion en cliquant sur le bouton *Firmware* de l'onglet *Mode*.

Si tout a été correctement configuré, le bouton *Firmware* retourne le modèle du microcontrôleur connecté. Sélectionnez votre modèle maintenant si ce n'est pas celui qui est sélectionné dans l'onglet *Mode*. Nous sommes prêts à écrire notre premier programme PICAXE. Recopiez le code du **listage 1** dans l'éditeur. Un mot toutefois sur la numérotation des broches et leurs noms internes avant de poursuivre : certaines puces peuvent avoir plusieurs blocs d'E/S, et dans ce cas le bloc que l'on veut sélectionner doit être préfixé par une lettre. Mon ancien modèle 08M n'avait pas besoin de préfixe, le code fonctionne donc avec un simple « 4 » pour désigner la sortie 4 sur la broche 3 : `high 4`. S'il ne fonctionne pas pour vous, référez-vous au brochage de votre modèle, et ajoutez un préfixe s'il y a lieu (par exemple, pour la broche 3, `high C.4` pour désigner la sortie 4 d'un bloc C). Chargez maintenant le programme dans votre PICAXE en cliquant sur le bouton *Upload*, en sélectionnant le menu *PICAXE/Program*, ou en appuyant sur F5. L'interface PC indique la progression du chargement ainsi que la mémoire disponible sur la puce PICAXE. Le message *programming successful* s'affiche à la fin de l'opération si tout s'est bien déroulé. Le nouveau programme démarre instantanément et la LED clignote.

#### Listage 1 : votre premier programme PICAXE

```
do ;repeat forever
  high 4 ;set output 4 high
  pause 1000 ;wait 1 second
  low 4 ;set output 4 low
  pause 1000 ;wait 1 second
loop
```

#### Listage 2 : commande de LED par bouton-poussoir

```
do ;repeat forever
  if pin3=0 then ;if the button is pressed
    high 4 ;light LED
  else ;if the button is NOT pressed
    low 4 ;turn off LED
  endif ;closes if statement
loop
```

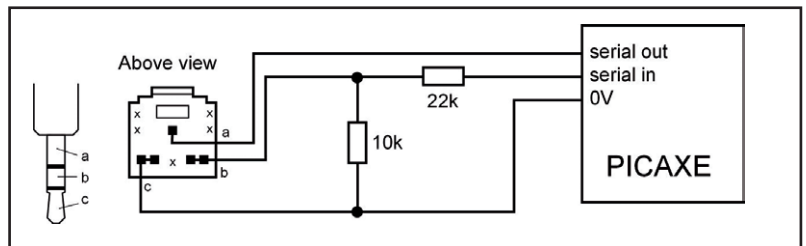


Figure 5. Le circuit de téléchargement.

#### Ajout d'un bouton-poussoir

Voyons sur un autre exemple comment allumer la LED avec un bouton-poussoir. Reliez un poussoir à la broche 4 du PICAXE (appelée input 3 dans le programme) (**fig. 6**). Copiez ensuite le code du **listage 2** dans l'éditeur. Après son chargement dans la puce, la LED ne s'allume que lorsqu'on appuie sur le bouton : l'instruction `if` allume la LED si la sortie 3 (sur la broche 4) est au niveau bas, sinon (`else`) elle l'éteint. Le **figure 7** montre le montage correspondant avec un PICAXE

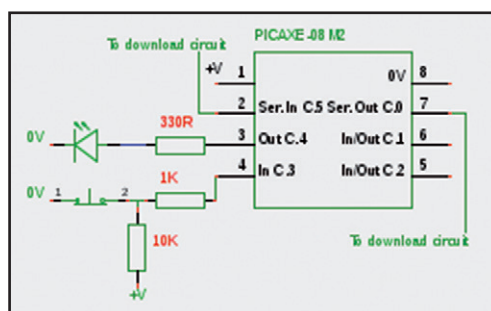


Figure 6. Schéma du circuit avec LED et bouton.

08M programmé et déconnecté du circuit de téléchargement.

Comme vous le voyez, vous convaincre de la simplicité du système PICAXE n'était pas difficile ! Nombreux sont ceux qui adoptent PICAXE pour se lancer dans la programmation de leurs projets. L'étendue des modèles disponibles et la qualité de la documentation fournie par *Revolution Education* ne sont sans doute pas non plus étrangers à ce succès.

(130137 – version française : Hervé Moreau)

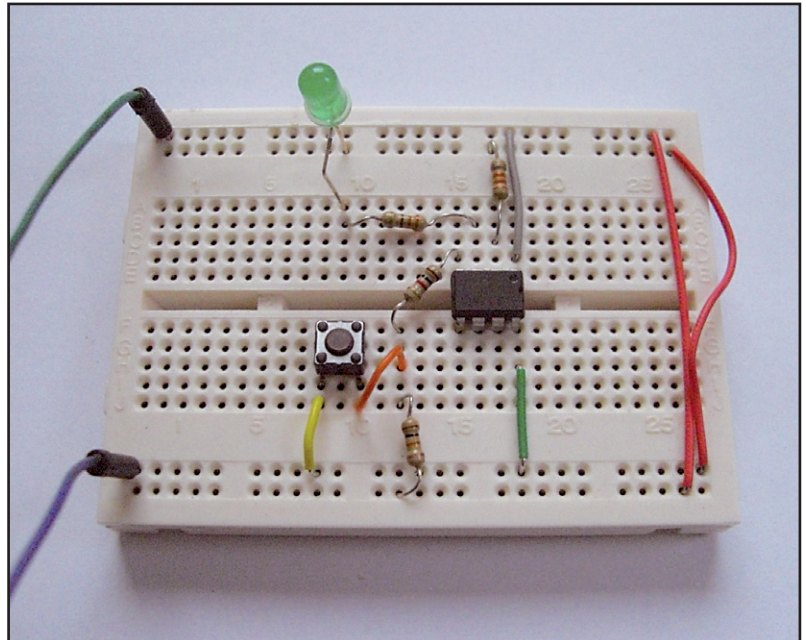


Figure 7. PICAXE 08M, LED et poussoir prêts à fonctionner.

## Liens

- [1] [www.picaxe.com](http://www.picaxe.com)
- [2] [www.picaxe.com/What-is-PICAXE/PICAXE-Chip-Sizes](http://www.picaxe.com/What-is-PICAXE/PICAXE-Chip-Sizes)
- [3] [www.techsupplies.co.uk/PICAXE](http://www.techsupplies.co.uk/PICAXE)
- [4] [www.rentron.com/myke4.htm](http://www.rentron.com/myke4.htm)
- [5] [www.arduino.cc](http://www.arduino.cc)
- [6] [www.picaxe.com/Getting-Started/PICAXE-Manuals](http://www.picaxe.com/Getting-Started/PICAXE-Manuals)
- [7] [www.picaxe.com/What-is-PICAXE/PICAXE-Pinouts](http://www.picaxe.com/What-is-PICAXE/PICAXE-Pinouts)
- [8] [www.picaxe.com/Software](http://www.picaxe.com/Software)
- [9] [www.picaxe.com/Getting-Started/Driver-Installation](http://www.picaxe.com/Getting-Started/Driver-Installation)