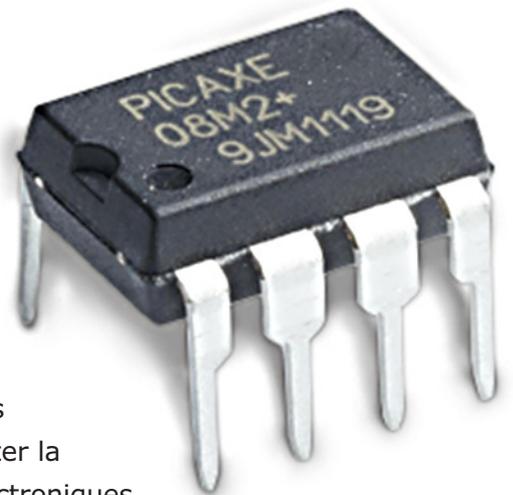


# PIC atout AXE (2)

## des E/S pour PICAXE



Le premier article de cette série, que vous avez reçu avec *elektor.POST* n°8, constituait une introduction au système PICAXE [1], avec des exemples sur la construction d'un circuit de programmation et sur la manière de programmer une puce PICAXE pour lui faire allumer une LED. Cette série couvrira des projets de circuits électroniques à base des microcontrôleurs de la gamme PICAXE. Commençons par doter la puce d'entrées et de sorties rien qu'avec des composants électroniques ordinaires. Il sera aussi question de vous montrer les facteurs déterminants pour calculer et choisir les composants des circuits d'interface. Les articles à venir vous feront découvrir de quelles capacités spécifiques ces puces disposent pour communiquer avec l'extérieur par PICAXE. On peut étendre ces possibilités par l'addition de puces spéciales et de périphériques (mémoire, clavier, écran LCD) ou même un PC avec une liaison série.

**Wouter Spruit**  
(Pays-Bas)

### La vérité est ailleurs, non ?

Internet est une mine d'informations pour qui se lance dans l'électronique des microcontrôleurs. Mais ce qui y manque le plus, ce sont celles qui expliquent les bases théoriques et des exemples particuliers pour les circuits d'interface. On trouve facilement où placer les composants dans les circuits d'exemple, mais comment savoir précisément pourquoi ils ont été choisis ? Pour les exemples d'interface présentés ici, vous trouverez aussi la théorie nécessaire pour comprendre ce que font les composants et comment sélectionner les bons dans vos projets, qu'ils soient similaires ou différents.

### Préparons le dispositif expérimental

Les expériences que nous allons décrire maintenant se font sur un PICAXE 08M2 dont le

brochage est représenté à la **figure 1** et qui fonctionne sous les 5 V délivrés par une alimentation ATX. Les circuits sont à construire sur des plaques d'expérimentation sans soudure. Pour programmer la puce PICAXE, nous passerons par un câble USB-série en suivant la méthode décrite dans l'article précédent [2], avec le logiciel LinAXEpad version 1.5.0 pour Linux (Arch). Pour éviter toute confusion, tous les numéros de broche se réfèrent à la numérotation physique sur la puce, p. ex. la broche 3 est une sortie et la broche 4, une entrée. Dans les listages du code, les numéros de broche font référence aux noms internes des broches (fig. 1). Pour les exemples sur la programmation d'une puce PICAXE, reportez-vous à l'article précédent [2] ou au manuel PICAXE [3]. On peut se procurer les puces PICAXE et différents périphériques p. ex. chez *Revolution Education* [4].

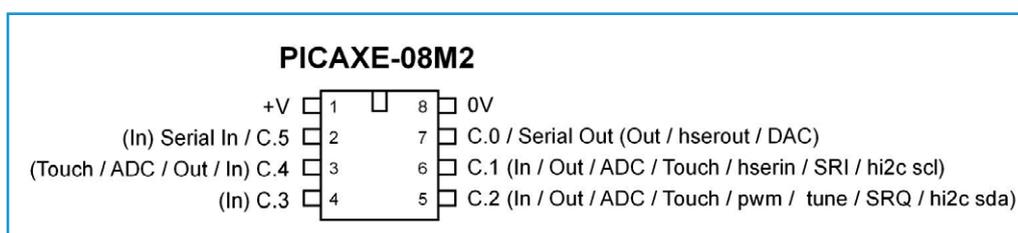


Figure 1.  
Le brochage du PICAXE 08M2.

**Prendre soin de ses composants**

Les composants se caractérisent par des valeurs limites pour la tension et le courant. Tout dépassement de ces maxima risque de les endommager définitivement. Le courant, la tension et la résistance sont liés entre eux par la loi d’Ohm :

tension (V) = courant (I) × résistance (R).  
Autrement dit, le courant qui traverse un composant est le résultat de la différence de potentiel (*voltage*) entre deux points et la résistance présente entre ces points. Avec une source de tension continue fixe de 5 V p. ex. comme pour une puce PICAXE, on peut limiter le courant en ajoutant une résistance en série. La valeur des résistances en série s’additionne pour chacune d’elles :  $R = R1+R2+R\dots+Rn$ , tandis qu’avec des résistances placées en parallèle, la résistance résultante diminue chaque fois :  $1/R = 1/R1 + 1/R2 + 1/R\dots + 1/Rn$ . Le courant qui traverse des résistances en série est le même partout, même si la résistance varie d’un composant à l’autre. On déduit de la loi d’Ohm que la différence de potentiel est divisée aux bornes des composants proportionnellement à leur résistance. En principe, tous ceux qui s’intéressent à l’électronique savent déjà tout cela et les informations complémentaires sur la loi d’Ohm sont aisément disponibles sur internet, par exemple sur ce site [5]. Il n’est jamais trop tard pour en apprendre davantage !

Quand on a compris comment ça marche, le calcul des résistances dans un circuit réel est plus facile. Prenons le diviseur de tension de la **figure 2**. Les résistances R1 et R2 sont branchées en série, le courant est donc constant partout depuis la source de tension jusqu’à la masse (0 V). En revanche, les chutes de tension sur R1 et R2 diffèrent en rapport avec leur résistance. Comme la loi d’Ohm s’applique à des différences de potentiel, la corrélation entre tension, courant et résistance s’applique aussi à une partie du circuit. Comme on le voit à la figure 2, la tension sur R1+R2 vaut +5 V et la tension sur R1 devient  $+5 V \times (R1/(R1+R2)) = 5 \times ((5 k\Omega/15 k\Omega) = 1,67 V$ , simple application de  $V = R \times I$ . Logiquement, la tension résiduelle sur R2 est de  $5 - 1,67 = 3,3 V$ , résultat obtenu aussi par  $5 \times (R2/(R1+R2))$ . Ceci est vrai pour des composants résistifs, mais pas nécessairement pour d’autres. Par exemple, les diodes, dont

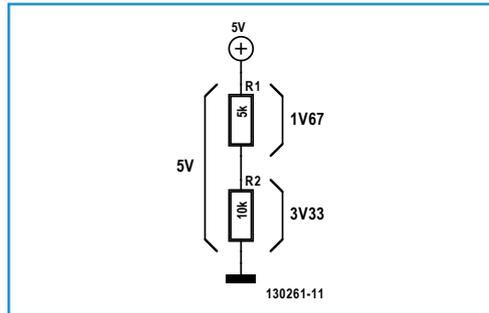


Figure 2. Exemple de diviseur de tension.

les LED ou les entrées de photocoupleurs (qui sont aussi des LED) ou les transistors. Pour calculer la résistance qui doit limiter le courant dans une LED, on utilise la chute de tension sur la LED au lieu d’essayer de mesurer sa résistance avec un multimètre ! Supposons une LED rouge à alimenter sur du 5 V avec un courant de 10 mA. On part de sa tension de fonctionnement qui est de 1,8 V. La chute de tension aux bornes de la résistance doit être de  $5 V - 1,8 V = 3,2 V$ . Utilisons  $V/I=R$  pour trouver  $3,2 V/0,01 A = 320 \Omega$ . La valeur habituelle de résistance existante (dans la série E6) la plus proche est  $330 \Omega$

Un diviseur de tension, c’est bien pratique quand on le combine avec l’entrée d’un microcontrôleur, par exemple. Imaginons de remplacer R1 ou R2, dans le circuit de la figure 2, par la résistance variable d’un capteur, pour mesurer la température ou l’intensité lumineuse. On relierait l’entrée du convertisseur A/N d’un microcontrôleur entre R1 et R2 pour mesurer une tension proportionnelle à la grandeur à évaluer.

Comme il y a une limite au courant qu’une puce peut drainer, si elle agit comme masse, ou fournir, si elle agit comme source, il est important de limiter le courant depuis ou vers la puce dans tout circuit d’interface. Cependant, dans la plupart des cas, les périphériques demandent plus de courant que la puce ne pourrait en donner sans partir en fumée.

**Le transistor**

Nous n’allons pas envisager ici les caractéristiques particulières de tous les transistors existants, ce n’est pas notre but. Utilisons simplement un transistor NPN disponible partout, le BC547B ou un équivalent, pour montrer comment éviter de dépasser la limite de courant sur les broches d’un microcontrôleur.

L'idée est d'utiliser un transistor de commutation capable de soutenir le courant nécessaire au fonctionnement d'un périphérique, quand la puce ne peut pas le faire.

Un transistor NPN régule le courant qui s'écoule de son collecteur à sa broche d'émetteur en fonction de la tension appliquée à sa base. Un petit courant qui passe par sa base entraîne un plus grand courant à travers ses broches de collecteur et d'émetteur, de sorte que le transistor travaille en amplificateur jusqu'à ce qu'il ne puisse plus limiter le courant, on dit alors qu'il est saturé. Le transistor PNP commute fondamentalement dans l'autre sens, mais ce n'est pas notre affaire aujourd'hui.

Forts de ce savoir, nous allons construire un circuit pour essayer le transistor dans son rôle de commutateur pour commander une LED. La broche de sortie du PICAXE ne doit rien faire d'autre que de fournir assez de courant pour faire commuter le transistor, le courant pour allumer la LED vient directement de l'alimentation dans le cas présent.

L'exemple du transistor reprend le même bouton-poussoir que dans l'article d'introduction de la série PICAXE, la **figure 3** vous montre le circuit à transistor que j'ai utilisé. On y ajoute le montage avec le simple commutateur de la **figure 4**. Le code nécessaire à la puce PICAXE se trouve dans le **listage 1**. Vous pouvez le télécharger comme fichier texte sur la page de cette série d'articles dans *elektor.LABS* [6]. La **figure 5** montre le circuit entier assemblé sur une plaque d'expérimentation, avec le bouton-poussoir. On allume la LED en poussant sur le bouton.

Quand vous utilisez le transistor pour commuter une charge dans votre projet, assurez-vous au préalable que la tension maximum sur les broches concernées et le courant maximum de collecteur ( $I_c$ ) ne dépassent pas les spécifications. Des caractéristiques comme la vitesse de commutation ou le gain ( $hFE$ ) sont moins critiques pour un fonctionnement « lent » en tout ou rien qu'à haute vitesse ou pour une commande d'amplification précise. En calculant la valeur de la résistance qui limite le courant venant d'une broche de sortie vers la base du transistor, assurez-vous

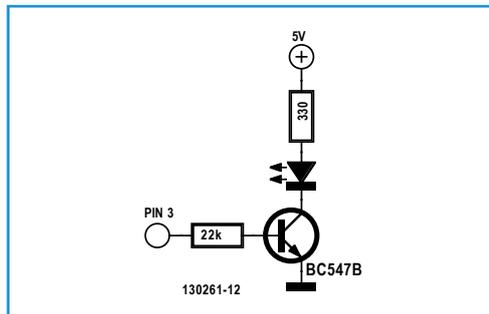


Figure 3. Un circuit à transistor.

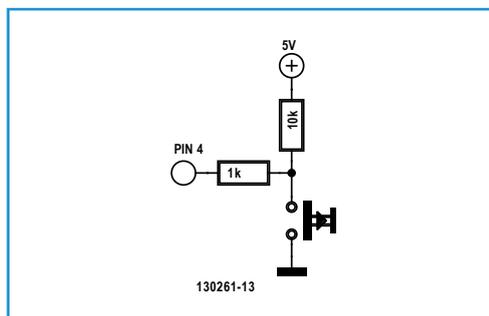


Figure 4. Le circuit à bouton-poussoir de l'article d'introduction.

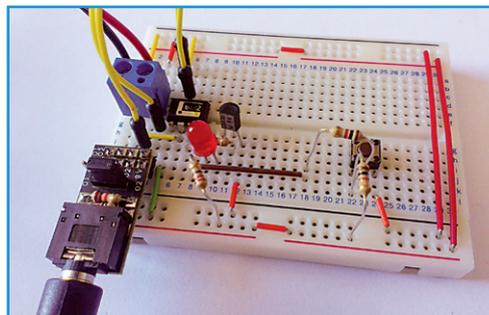


Figure 5. Un circuit expérimental à transistor avec un bouton-poussoir sur plaque d'expérimentation

que ce courant n'excède pas le maximum que la broche peut fournir, mais qu'il sera suffisant pour saturer le transistor : c'est ici que le gain intervient. Pour calculer le courant minimum dans la base, divisez le courant de sortie du microcontrôleur par le gain du transistor ( $hFE = 200$ ) et par sécurité, vous pouvez encore le doubler (il faut parfois le

### Listage 1: Switch code

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

multiplier par 5 !) pour être sûr de faire entrer le transistor en saturation complète.

Pour une LED qui consomme 10 mA, le courant de base doit valoir  $10/200 = 0,05$  mA, en doublant, on a 0,1 mA, soit 100 fois moins que ce que le microcontrôleur peut fournir. Calculons la résistance comme dans l'exemple du diviseur de tension. On part de 5 V, le transistor en prend 0,7 V comme une diode [7], ce qui laisse 4,3 V aux bornes de la résistance de limitation de courant :  $R = V/U$  donc  $R = 4,3/0,1 \times 10^{-3} = 43 \times 10^3 = 43$  k $\Omega$ . Mais pour une charge plus élevée que 10 mA, la résistance devra être de moindre valeur et pour atteindre le courant maximum que le transistor peut tolérer, 200 mA, il faut que le microcontrôleur puisse donner  $2 \times 200/200 = 2$  mA, ce qu'il est capable de fournir. Pour la résistance :  $R = 4,3/2 = 2,2$  k $\Omega$ . Une considération pratique : cette valeur existe bien, mais je n'ai sous la main que des 1 k $\Omega$ . Je peux choisir d'en mettre deux en série ou, si je n'ai pas à me soucier du courant parce que le circuit n'est pas alimenté par des piles, je pourrais simplement utiliser une 1 k $\Omega$  pour autant que le microcontrôleur puisse délivrer le courant nécessaire. 1 k $\Omega$  comme résistance de base est une valeur fréquente pour de faibles charges. D'ailleurs, des transistors ordinaires ont souvent un gain moindre !

### La paire Darlington

En 1952, Sidney Darlington a eu l'idée d'augmenter le gain du transistor en l'associant à un second dans une configuration particulière représentée à la **figure 6** et pour laquelle il obtint un brevet l'année suivante [8]. Le gain en courant résultant étant le produit des deux, la saturation est facilement assurée. Pas étonnant donc que plusieurs cartes de projet PICAXE, comme AXE002U, PICAXE-18M2 Starter Pack, emploient des paires Darlington comme interrupteur pour commander des appareils de grande puissance.

Construisez le circuit à paire Darlington de la figure 6 et ajoutez-y l'interrupteur de la figure 4. Programmez-le avec le code du listing 1 que vous connaissez déjà. Le montage d'essai complet du circuit est à la **figure 7**.

Le choix des transistors pour un Darling-

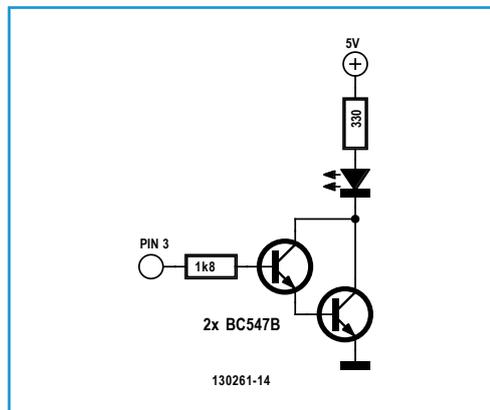


Figure 6. Circuit d'une paire Darlington.

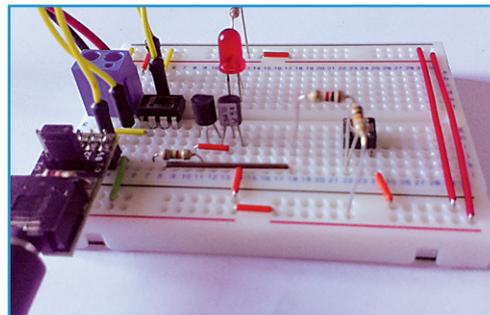


Figure 7. Le même sur plaque d'expérimentation.

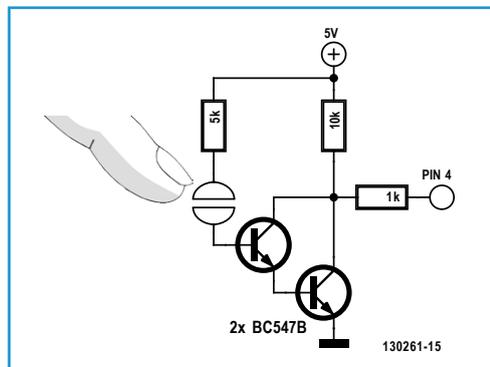


Figure 8. Une paire Darlington utilisée en interrupteur à touche sensitive

ton ne diffère pas de celui d'un transistor simple, sauf à se rappeler que les deux jonctions BE en série occasionnent une chute de tension double et que le temps de commutation est allongé, si la vitesse est un critère pour votre application. Le calcul de la résistance de limitation du courant à 2 mA est semblable au précédent,  $5-0,7-0,7 = 3,6$  V entraîne  $3,6/2 = 1,8$  k $\Omega$ , cependant, le gain en courant est beaucoup plus grand ( $200 \times 200 = 40\,000$ ), donc le courant de base est divisé par 200 et la valeur de la résistance devient  $200 \times 1,8$  k $\Omega = 360$  k $\Omega$ .

Comme pareille valeur dépasse la résistance superficielle de la peau de nos doigts [9],

nous pouvons utiliser un Darlington comme interrupteur sensible selon le schéma de la **figure 8**. Le code nécessaire reste le même. On peut s'interroger sur la sécurité de l'opération et éviter les inconvénients d'un choc électrique en cas de court-circuit accidentel par contact avec un autre conducteur. Un courant inférieur à 4 mA (information complémentaire sur [10]) ne peut pas entraîner de douleur. Il est fort peu probable qu'on puisse le dépasser avec une source de 5 V dans des conditions normales [11] et avec une résistance de 5 k $\Omega$ , le courant ne peut pas dépasser 1 mA.

Montez la paire Darlington en interrupteur sensible comme à la figure 8 et la sortie comme à la figure 6. Utilisez le même code que dans le listage 1. La **figure 9** montre le circuit terminé sur la plaque d'expérimentation et vous le voyez en action à la **figure 10**.

### Le photocoupleur

Il y a des cas où il faut éviter d'alimenter à la même source le microcontrôleur et l'appareil à commander. On se sert alors d'un photocoupleur pour commuter d'un circuit à l'autre, chacun disposant de sa propre alimentation et de masses séparées. C'est l'isolation galvanique, il n'y a plus de conducteur commun, le signal est transmis par rayonnement IR. Le photocoupleur se compose d'une LED et d'une photodiode ou d'un phototransistor dans le même boîtier. Quand la LED est éteinte, le composant photoélectrique dans l'autre circuit est bloquant ; dès qu'elle s'allume, elle le débloquent, il y circule donc un courant. Certains photocoupleurs n'agissent qu'en commutation, d'autres peuvent transmettre un signal analogique. Le choix d'un photocoupleur parfaitement adapté à vos besoins peut se révéler difficile, j'en sais quelque chose. Il y a sur internet et dans les magasins une large variété de modèles. Si vous ne connaissez pas quelqu'un d'expérimenté dans le domaine, vous pouvez passer des heures à parcourir des feuilles de caractéristiques.

Pour cet exemple, j'utilise le 4N25. Ce photocoupleur peut servir dans la plupart des applications à faible tension, mais vérifiez toujours la feuille de caractéristiques pour connaître la tension et le courant de la LED, la gamme de tensions et de courants pour le

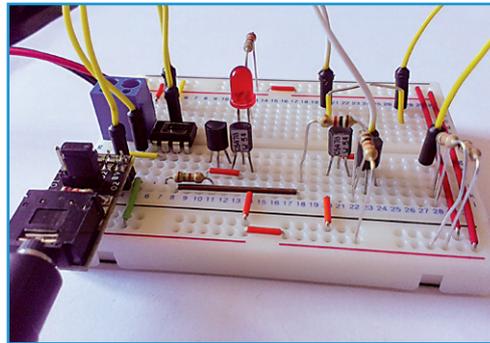


Figure 9. Le circuit de la touche sensible sur plaque d'expérimentation.

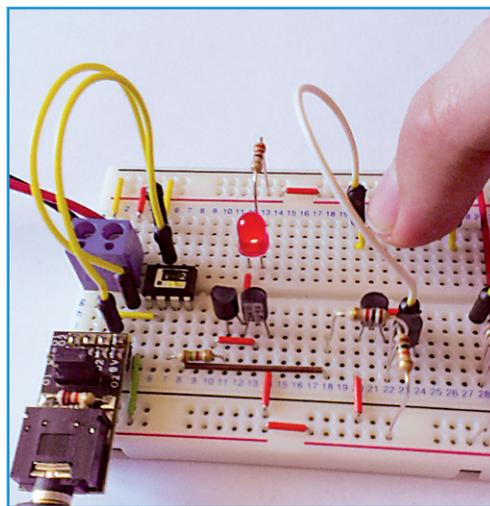


Figure 10. L'interrupteur à touche sensible en action.

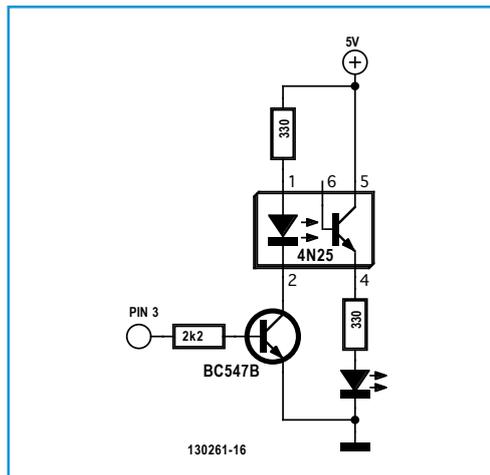


Figure 11. Schéma de montage d'un photocoupleur.

phototransistor et sa vitesse de commutation, selon votre application.

Construisez le circuit de la **figure 11** pour la broche de sortie et pour l'entrée, choisissez l'interrupteur rudimentaire de la figure 4 ou la version à touche sensible de la figure 8, en utilisant le code du listage 1 pour les deux. Le circuit complet est à la **figure 12**.

Dans cet exemple, il y a un transistor pour faire commuter le photocoupleur, mais dans bien des cas, le photocoupleur se connecte au microcontrôleur à travers une résistance pour limiter le courant. Pour calculer cette résistance, j'ai pris une tension sur la LED de 1,3 V et pour y faire circuler un courant d'à peu près 10 mA [12], il faut  $R = (5-1,3)/0,01 = 370 \Omega$ , valeur la plus proche : 330  $\Omega$ .

## Les relais

Quand il s'agit d'utiliser votre PICAXE pour allumer et éteindre des appareils alimentés sur secteur, il doit impérativement en être isolé. Les relais sont des interrupteurs animés par un solénoïde : appliquer la tension voulue à la bobine change l'état du commutateur, retirer son alimentation le renvoie dans sa position primitive. La position de départ peut être interrupteur fermé ou ouvert. Il existe aussi des relais bistables qui « mémorisent » leur état, ils ne consomment rien tant qu'ils restent dans le même état, seulement le temps d'en changer. Les relais sont des appareils mécaniques sujets à l'usure, ils ne peuvent commuter qu'un certain nombre de fois avant de tomber en panne, ils ne conviennent donc pas aux commutations rapides. Les relais à semi-conducteurs n'ont pas de pièces mobiles, comme les photocoupleurs, ils peuvent remplacer les relais traditionnels pour résoudre les limitations mécaniques, mais ils coûtent plus cher.

Quand un relais change d'état, il provoque une pointe de tension qui peut endommager l'électronique, il faut les doter d'une protection par diode polarisée en inverse, en parallèle sur la bobine. Cette diode de roue libre dévie l'extra-courant de rupture et empêche une surtension dangereuse, pour autant que ses caractéristiques de tension et courant soient respectées. Tout bien considéré, il est peut-être préférable de commander un relais à travers un photocoupleur plutôt qu'un simple transistor pour séparer l'alimentation du relais et ses pics de tension du microcontrôleur.

Dans tous les exemples précédents, nous avons utilisé un interrupteur ou un bouton qui donne une sortie uniquement quand on y touche, mais avec l'exemple du relais, il nous faudra un inverseur : quand on l'ac-

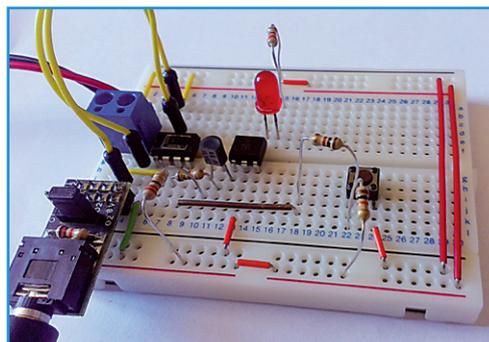


Figure 12. Le circuit à photocoupleur sur plaque d'expérimentation avec un bouton-poussoir élémentaire.

### Listage 2: On/Off code

```

b0 = 0           ;set state 0 (OFF)
b1 = 0           ;reset last pin state
low 4            ;reset output
main:
do               ;repeat forever
  if pin3=0 then ;button is pressed
    gosub btnpress ;jump to this label
  else
    if b1=1 then ;check button release
      b1=0       ;reset pin state variable
    endif
  endif
loop

btnpress:        ;btnpress label (for gosub)
  if b1=0 then   ;is this the first visit since
                 ;button release?
    b1=1         ;mark as visited
    gosub setstate ;jump to label
    pause 100    ;prevent switch bounce
  endif
return           ;return to gosub statement

setstate:       ;setstate label
  if b0=0 then   ;if current state is OFF
    b0=1         ;remember new state
    high 4       ;set pin 4 high
  else           ;current state is ON
    b0=0         ;set state OFF
    low 4        ;set output pin low
  endif
return
    
```

tionne, il change d'état entre marche et arrêt. Pour ce faire, nous devons modifier le code du bouton-poussoir du listage 1 pour enregistrer cet état après avoir relâché le bouton. Il y a plusieurs façons de stocker des données sur la puce PICAXE, nous les passerons en revue dans le prochain article. Pour l'instant, utilisons les variables d'usage général. Chacune de ces puces dispose d'un certain nombre de ces variables, au moins 14, appelées b0 à b13, chacune de 8 bits de long. Pour stocker une valeur de 16 bits, on utilise deux variables contiguës, w0 couvre alors b1 et b0, w1 sur b3 et b2, etc. Évidemment, écrire dans w1 revient à remplacer le contenu de b3 et b2. Dans certaines des nouvelles puces PICAXE, il est aussi possible de combiner quatre variables à 8 bits pour faire un mot de 32 bits. Certaines variables sont toujours utilisées pour stocker des données importantes pour des fonctions de PICAXE programmées d'origine.

Le code du **listage 1** montre comment utiliser la variable b0 pour enregistrer un 0 (arrêt) ou un 1 (marche) et basculer d'un état à l'autre. La sortie reflète ces états. Remarquez le retard inclus après l'appui sur le bouton. Il sert à éliminer l'effet de rebond, c'est une caractéristique des interrupteurs mécaniques après une action, ils produisent pendant un moment de rapides oscillations entre les états ouvert et fermé. La variable b1 sert à mémoriser l'état après un changement, sinon le basculement d'un état à l'autre continuerait tant que le bouton est enfoncé. Le code utilise l'instruction gosub par souci de clarté. Elle provoque un saut jusqu'à une étiquette d'où le code est exécuté jusqu'à la rencontre d'une instruction return, après quoi le programme se poursuit au point atteint avant le saut.

Le circuit de cet exemple se compose de celui à relais représenté à la **figure 13** et de l'interrupteur de la figure 4, mais à présent, le code est celui du **listage 2**. La **figure 14** montre le circuit complet assemblé sur plaque d'expérimentation.

Cette fois encore, le photocoupleur n'a pas besoin d'un transistor intermédiaire, puisque le microcontrôleur peut délivrer le courant nécessaire. Le relais dispose ici de sa propre alimentation derrière le photocoupleur et

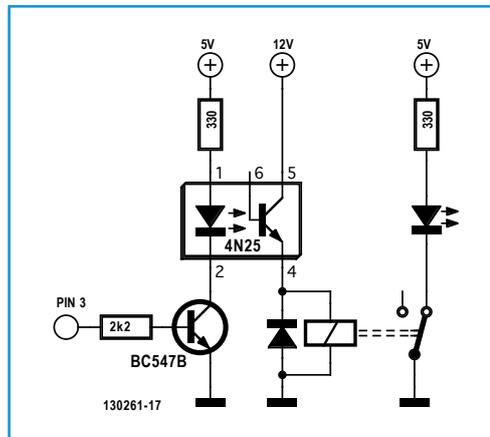


Figure 13. Le schéma du circuit à relais. Il est aussi commandé à travers un photocoupleur

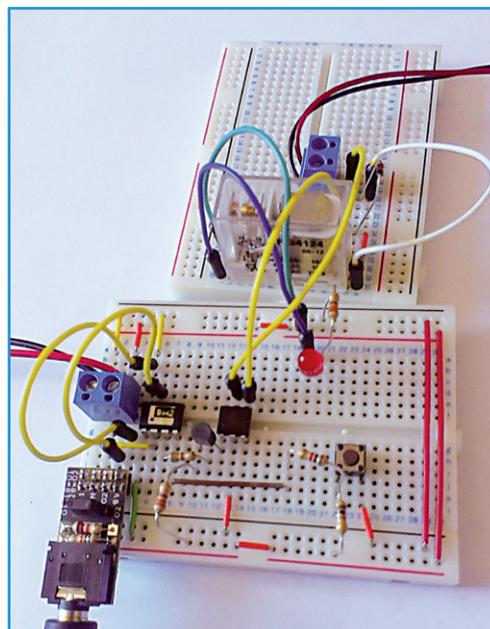


Figure 14. Le circuit à relais en action, il a été câblé sur une seconde plaque d'expérimentation.

d'une masse différente de celles du microcontrôleur. L'exemple utilise le +5 V et GND (donc GND et GND3 sont interconnectés dans ce cas-ci) du circuit du microcontrôleur pour le circuit à LED derrière le relais, mais selon le type de relais, on pourrait commuter des appareils alimentés sur secteur avec le même circuit.

Si vous voulez l'essayer, soyez prudent ! Si vous ne tenez pas à jouer avec la tension du secteur, abstenez-vous !

Enlevez la charge à LED et l'alimentation de la sortie du relais (et avant tout, assurez-vous que le relais peut réellement commuter la tension du secteur !). Pour créer un interrupteur, vous pourriez sacrifier une rallonge électrique (débranchée), couper les fils et les brancher

aux sorties du relais comme pour reformer les connexions d'origine quand les sorties du relais sont fermées. Après avoir testé l'action de commutation du relais sans avoir connecté la rallonge, connectez le câble à un appareil qui consomme un courant compatible avec les limites du relais (comme une lampe, pour un effet manifeste) et à une prise de courant. Maintenant, vous devriez être capable de commuter un appareil sur secteur avec un bouton PICAXE ! On est à deux pas d'un système complet de domotique !

Lors du choix d'un relais, il est important de vérifier la tension et le courant requis pour la commutation ainsi que la tension maximum et le courant qu'il peut commuter. Si vous actionnez un relais avec un photocoupleur, assurez-vous toujours que le photocoupleur est en mesure de fournir au relais le cou-

rant requis ainsi que de supporter la tension voulue !

### À chaque jour suffit sa peine

Au point où nous en sommes, nous avons vu comment câbler et programmer une puce PICAXE, ainsi que différents circuits d'entrée et de sortie avec le code. Nous savons à quoi faire attention dans le choix des composants pour nos propres projets et nous savons calculer les valeurs de résistance pour limiter le courant pour protéger ces composants. Le prochain article concernera des entrées et sorties élaborées, couvrant aussi des asservissements, p. ex. pour construire des robots. Puis nous explorerons les possibilités d'ajouter au projet PICAXE des circuits intégrés afin d'augmenter ses capacités et même d'interfacer un PICAXE avec un PC. Qu'en pensez-vous ?

(130261 - version française : Robert Grignard)

#### Liens & littérature

- [1] [www.picaxe.com](http://www.picaxe.com)
- [2] [Projet Elektor.POST n° 8 : PIC atout AXE](http://Projet%20Elektor.POST%20n%C3%B0%208%20%3A%20PIC%20atout%20AXE)  
[www.elektor-magazine.com/extra/post](http://www.elektor-magazine.com/extra/post)
- [3] [www.picaxe.com/Getting-Started/PICAXE-Manuals](http://www.picaxe.com/Getting-Started/PICAXE-Manuals)
- [4] [www.techsupplies.co.uk/PICAXE](http://www.techsupplies.co.uk/PICAXE)
- [5] <http://hyperphysics.phy-astr.gsu.edu/hbase/electric/ohmlaw.html>
- [6] [www.elektor-labs.com/picaxe](http://www.elektor-labs.com/picaxe)
- [7] [www.farnell.com/datasheets/410427.pdf](http://www.farnell.com/datasheets/410427.pdf)
- [8] [http://en.wikipedia.org/wiki/Darlington\\_transistor](http://en.wikipedia.org/wiki/Darlington_transistor)
- [9] C.J. Poletto et C.L. van Doren,  
« A high voltage, constant current stimulator for electrocutaneous stimulation through small electrodes »,  
IEEE Trans biomed. Eng., vol 46, n° 8, pp. 929 à 936, 1999
- [10] F.A. Saunders, « Electrocutaneous displays »  
in Proc. Conf. Cutaneous Commun. Syst. Devices, pp 20 à 26, 1973
- [11] K.A. Kaczmarek, « Optimal electrotactile stimulation waveforms for human information display »,  
PhD thesis, Dep. Elev. Eng. Univ. Wisconsin-Madison, 1991
- [12] <http://www.vishay.com/docs/83725/4n25.pdf>