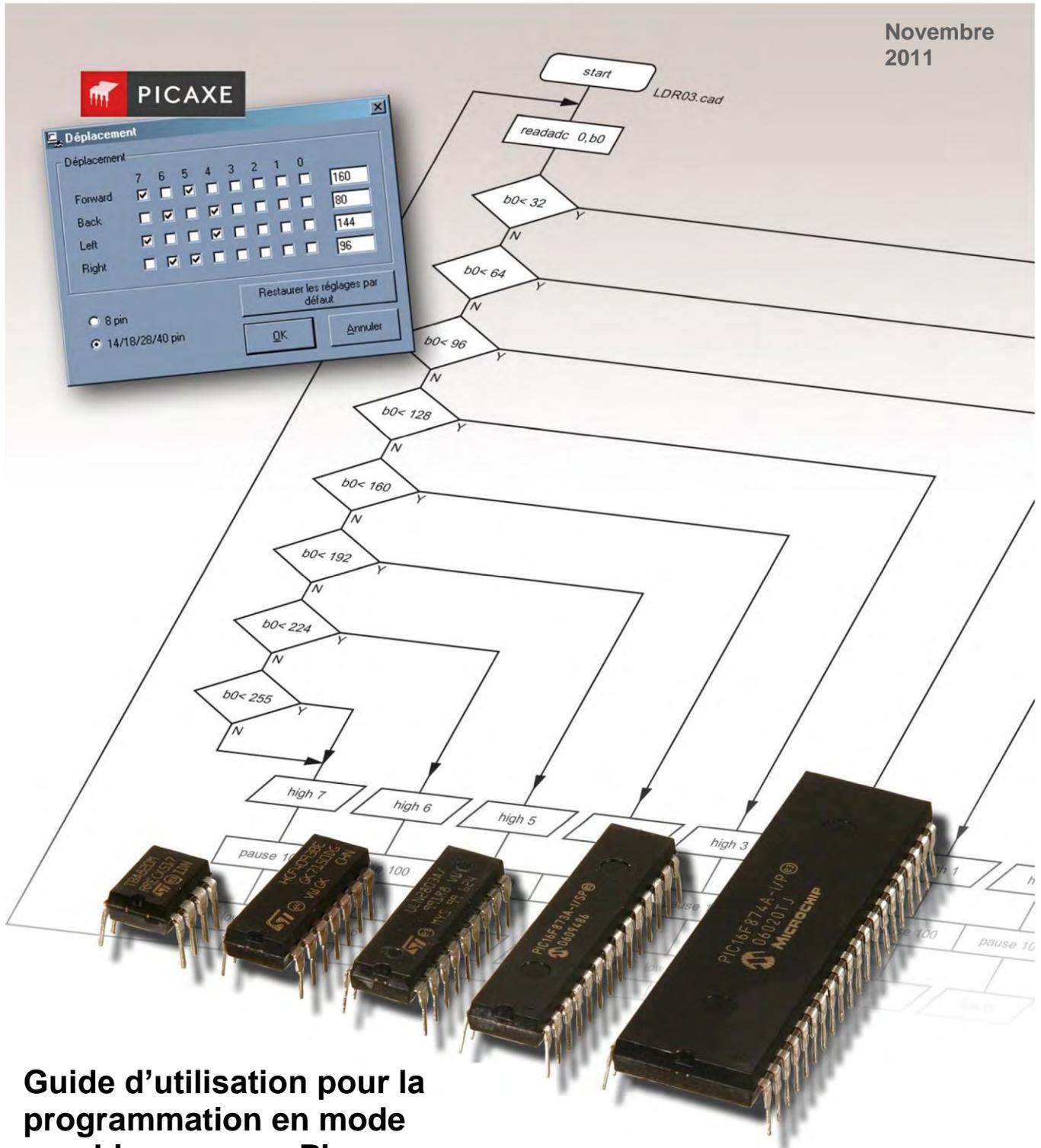


PICAXE PROGRAMMING EDITOR

Novembre
2011



**Guide d'utilisation pour la
programmation en mode
graphique avec « Picaxe
Programming Editor »**

V 1.0



Édité par la Société a4 Technologie

5 Avenue de l'Atlantique - Z.A. de Courtaboeuf - 91940 Les Ulis
Tél. : 01 64 86 41 00 - Fax. : 01 64 46 31 19 - www.a4.fr

André Bernot
Thierry Lancelot
Denis Hoffschir
Dominique Sauzeau

Le logiciel « **Picaxe Programming Editor** » permet d'installer l'environnement de programmation Picaxe. Il est disponible sur cédérom (Réf. : **CD-FPPE**) ou en téléchargement gratuit à partir du site www.a4.fr dans la rubrique « Automatismes et robotique / Picaxe ».

Le système **Picaxe®** et le logiciel graphique de programmation « **Picaxe Programming Editor** », sont des produits de la société « **Revolution Education** ». Ce guide a été conçu et publié par la société « **a4 Technologie** ».

Le logiciel « **Picaxe Programming Editor** » et la gamme des microcontrôleurs ou cartes d'applications Picaxe sont susceptibles d'évoluer. Il convient de vérifier l'existence de mises à jour de ces éléments et de ce guide sur www.a4.fr

Droit de reproduction

- Ce dossier et les fichiers de programmation associés sont libres d'utilisation pour les établissements scolaires.
- La copie ou la diffusion par quelque moyen que ce soit en dehors d'un usage interne à un établissement scolaire de tout ou partie du dossier ou du cédérom n'est pas autorisée sans l'accord écrit de la société A4.
- Le logiciel Picaxe Programming Editor © est protégé par les lois du copyright international. Tout ou partie de ce logiciel ne peut être reproduit, copié, vendu, revendu ou exploité dans un but commercial qui n'ait été expressément autorisé par la société Revolution Education.

SOMMAIRE

| | |
|--|----|
| Présentation générale | 03 |
| 1. Présentation du système Picaxe | 04 |
| 1.1 Présentation du système Picaxe | 04 |
| 1.2 Débuter avec le système Picaxe | 05 |
| 1.3 La gamme de microcontrôleurs Picaxe | 06 |
| 1.4 Les cartes d'applications Picaxe | 07 |
| 1.5 Les cartes de prototypage Picaxe | 08 |
| 1.6 Les cartes du système AutoProg | 10 |
| 2. Présentation du logiciel « Picaxe Programming Editor » | 11 |
| 2.1 Installation du logiciel « Picaxe Programming Editor » | 12 |
| 2.2 Premier lancement du logiciel | 13 |
| 2.3 Mise en service du câble de programmation | 14 |
| 2.4 Transfert d'un programme | 15 |
| 2.5 La table des symboles pour les diagrammes | 15 |
| 2.6 Le paramétrage des options | 16 |
| 3. Guide d'utilisation du logiciel « Picaxe Programming Editor » | 21 |
| 3.1 La barre d'outils de Picaxe Programming Editor | 22 |
| 3.2 Les outils pour la programmation en mode graphique | 23 |
| 3.3 Les commandes de sorties « Out » | 24 |
| 3.4 Les commandes de mouvements | 37 |
| 3.5 Les commandes de tests | 42 |
| 3.6 Les commandes de temporisation | 45 |
| 3.7 Les commandes de sous-programme | 48 |
| 3.8 Les autres commandes | 53 |
| 4. Connexion des composants à un microcontrôleur Picaxe | 63 |
| 4.1 Caractéristiques principales des microcontrôleurs Picaxe | 64 |
| 4.2 Brochage des microcontrôleurs Picaxe | 65 |
| 4.3 Connecter un capteur tout ou rien (TOR) | 67 |
| 4.4 Connecter un capteur numérique de température DS18B20 | 68 |
| 4.5 Connecter un capteur analogique (LDR, CTN, potentiomètre) | 69 |
| 4.6 Connecter un capteur de distance SRF05 | 70 |
| 4.7 Connecter un capteur infrarouge | 70 |
| 4.8 Connecter un afficheur LCD | 71 |
| 4.9 Connecter un circuit de puissance (ULN2003) | 71 |
| 4.10 Connecter un circuit de commande moteur (L293D) | 72 |
| 5. Annexes | 73 |
| 5.1 Installation de « Picaxe Programming Editor » | 74 |
| 5.2 Installation et mise en service du câble de programmation Picaxe | 77 |
| 5.3 Vérification du Firmware | 86 |
| 5.4 Utilisation et syntaxe de la table des symboles | 87 |
| 5.5 Utilisation du mode simulation | 88 |
| 5.6 Représentation des microcontrôleurs sous « Picaxe Programming Editor » | 90 |
| 5.7 Lexique | 92 |
| 5.8 Forum Picaxe francophone | 92 |

Présentation générale

Ce guide aide l'utilisateur à programmer en mode graphique le système **Picaxe** avec l'application « **Picaxe Programming Editor** » (la version 5.3.2 a été utilisée pour développer ce guide).

Chapitre 1. Présentation du système Picaxe

Ce chapitre décrit le système Picaxe et les cartes associées.

Chapitre 2. Présentation du logiciel « Picaxe Programming Editor »

Ce chapitre décrit le fonctionnement du logiciel de programmation graphique « Picaxe Programming Editor ».

Chapitre 3. Guide d'utilisation de « Picaxe Programming Editor »

Ce chapitre présente pour chaque commande du logiciel les caractéristiques de programmation (icône, compatibilité, fonction, option(s) sélectionnable(s), exemple de programmation, exemple de câblage).

Chapitre 4. Connexion des composants à un microcontrôleur Picaxe

Ce chapitre regroupe les schémas de câblage des capteurs et actionneurs (schémas d'application).

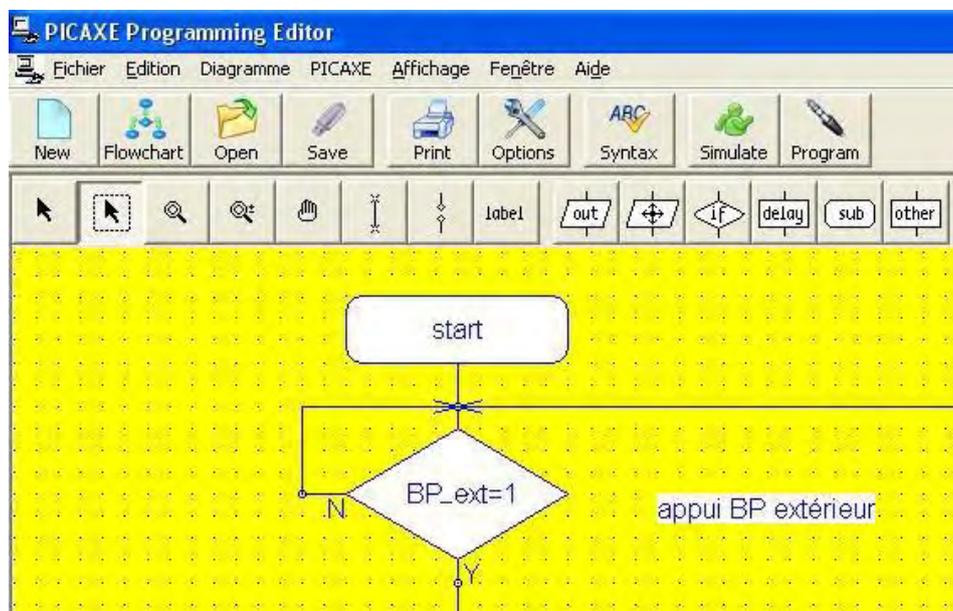
Chapitre 5. Annexes

Ce chapitre regroupe des fiches récapitulatives qui peuvent être utilisées à tout moment pour la mise en œuvre et l'utilisation de l'environnement de programmation « Picaxe Programming Editor ».



Le forum **PICAXE francophone** est visible à l'adresse suivante :
<http://www.picaxeforum.co.uk/forumdisplay.php?44-Le-forum-officiel-PICAXE-francophone>

Le forum Picaxe francophone permet aux utilisateurs de communiquer autour du système Picaxe et de trouver ou obtenir des réponses à leurs problèmes.



Les programmes décrits dans ce dossier sont disponibles en **téléchargement gratuit** à partir du site www.a4.fr
Le logiciel « **Picaxe Programming Editor** » évolue régulièrement. La dernière version de l'application est systématiquement disponible sur le site www.a4.fr



1. Présentation du système Picaxe

| | |
|---|----|
| 1.1 Le système Picaxe | 04 |
| 1.2 Débuter avec le système Picaxe | 05 |
| 1.3 La gamme de microcontrôleurs Picaxe | 06 |
| 1.4 Les cartes d'application Picaxe | 07 |
| 1.5 Les cartes de prototypage Picaxe | 08 |
| 1.6 Les modules AutoProg..... | 10 |

1.1 Le système Picaxe

Un **microcontrôleur** est un circuit intégré qui rassemble dans le même boîtier un microprocesseur, une **mémoire** (dans laquelle, il y aura le programme nécessaire à son fonctionnement) ainsi que les périphériques d'entrée et de sortie permettant de communiquer avec l'extérieur.

PIC est l'acronyme de « **Programmable Intelligent Computer** » ou « **Programmable Integrated Circuit** » et désigne les microcontrôleurs du constructeur « **Microchip** ».

Picaxe

Un Picaxe est un microcontrôleur **Pic** auquel est ajouté un logiciel d'amorçage développé par la société **Révolution Education**. Le logiciel d'amorçage (ou bootstrap) est un programme installé dans une zone spécifique du microcontrôleur ce qui permet de ne pas utiliser de programmeur spécifique. Ce mode particulier utilise deux broches du Pic (Serial Out et Serial In) et le dialogue s'effectue via une connexion **série RS232** ou **USB**.

Initialement commercialisés pour un usage pédagogique, ces circuits ont séduit de nombreux utilisateurs par leur simplicité de mise en œuvre, leurs performances et aussi par un coût très abordable.

La programmation des microcontrôleurs Picaxe s'effectue à l'aide d'un logiciel simple et gratuit : « **Picaxe Programming Editor** ». Le programme transféré dans le microcontrôleur Picaxe est conservé lorsque l'alimentation est arrêtée.

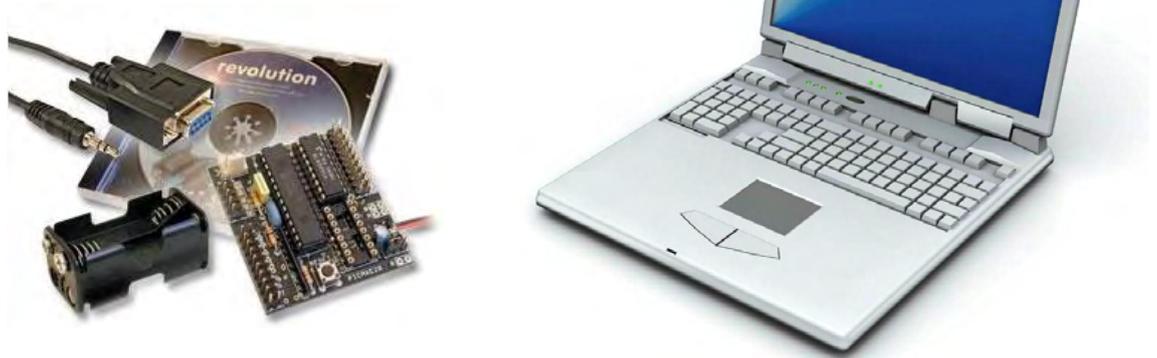
La gamme Picaxe est constituée de microcontrôleurs nus, de cartes d'application ou de prototypage et des produits dédiés.



1.2 Débuter avec le système Picaxe

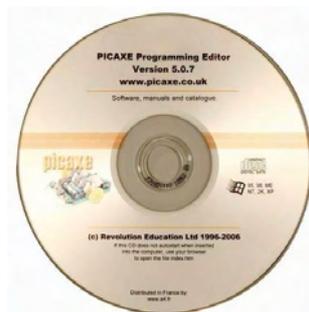
La mise en œuvre s'effectue en utilisant :

- un ordinateur type PC ou Mac ;
- une carte de prototypage ou d'application Picaxe ou le boîtier de commande « **AutoProg** » et ses modules ;
- le logiciel « Picaxe Programming Editor » ;
- un câble de programmation (USB ou RS232) ;
- des câbles de liaison.



Le logiciel « **Picaxe Programming Editor** » est en libre téléchargement sur le site Picaxe (<http://www.picaxe.com/>) ou sur le site de a4 Technologie (<http://www.a4.fr/>).

Un cédérom du logiciel « **Picaxe Programming Editor** » est disponible aussi chez a4 - Réf. : **CD-FPPE**. Selon la date du CD, il peut être nécessaire de télécharger la dernière version sur le site www.a4.fr.

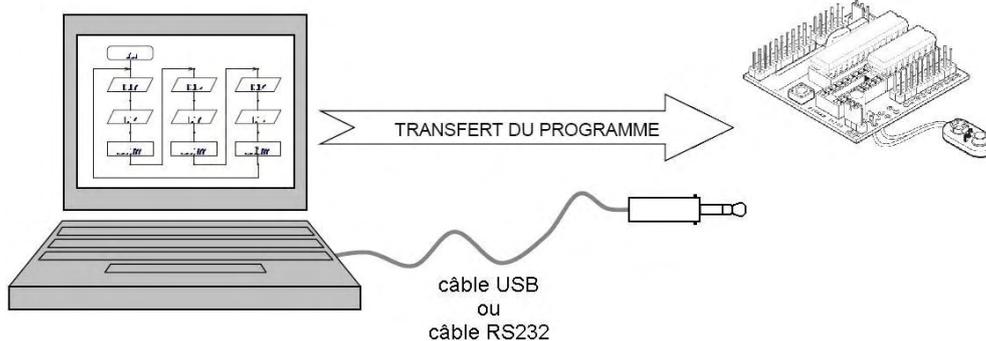


NOTE IMPORTANTE

*L'installation du logiciel « Picaxe Programming Editor » ainsi que l'installation et la mise en service du câble de programmation sont décrites dans les **sous-chapitres 5.1 et 5.2** des annexes de ce guide. Il est nécessaire de procéder à ces installations avant de poursuivre la lecture de ce guide.*

PC avec le logiciel Picaxe Programming Editor installé

Carte Picaxe



1.3 La gamme de microcontrôleurs Picaxe

Pour pouvoir s'adapter à différents projets, les microcontrôleurs Picaxe existent sous de nombreuses références, chaque Picaxe est caractérisé par :

- le nombre d'entrées et de sorties ;
- sa tension d'alimentation (5v ou 3,6V) ;
- le boîtier, facilement reconnaissable par le type (étroit, large, etc.) et le nombre de pattes ;
- la capacité mémoire, c'est-à-dire le nombre de lignes que l'on peut programmer ;
- les capacités spécifiques : bus i2c, gestion PWM, gestion IR, gestion infraout, etc.

Le tableau suivant récapitule les caractéristiques principales des microcontrôleurs Picaxe.

Nota : dans certains cas les E/S sont paramétrables (entrées numériques, entrées analogiques, sorties, etc.).

Ces configurations ne sont pas toutes exploitables dans le mode graphique de Picaxe Programming Editor. Vous trouverez des explications et des plans détaillés de chaque microcontrôleur Picaxe en fin de guide.

| Microcontrôleur Type Picaxe | Boîtier | Nombre d'entrées/sorties | Capacité mémoire (lignes) |
|--------------------------------|---------|-----------------------------|------------------------------|
| Picaxe 08M2 | DIL8 | 6 | 80 à 220 |
| Picaxe 14M2 | DIL14 | 12 | 80 à 220 |
| Picaxe 18M2 | DIL18 | 16 | 600 à 1800 |
| Picaxe 20M2 | DIL20 | 18 | 600 à 1800 |
| Picaxe 20X2 | DIL20 | 18 | 2000 à 3200 |
| Picaxe 28X1 | DIL28 | 22 | 2000 à 3200 |
| Picaxe 28X2 | DIL18 | 22 | 2000 à 3200 |
| Picaxe 40X1 | DIL40 | 33 | 2000 à 3200 |
| Picaxe 40X2 | DIL40 | 33 | 2000 à 3200 |



1.4 Les cartes d'applications Picaxe

Les cartes d'applications permettent de mettre en œuvre des applications variées. Elles disposent de capteurs et d'actionneurs constituant un système autonome.



Motopilot

Carte équipée d'un Picaxe 08 et d'un L293D.
Permet la mise en œuvre d'un robot mobile.



Kit capteur de température

Carte équipée d'un Picaxe 08M, d'un capteur numérique de température DS18B20, d'un bouton poussoir, d'une DEL et d'un buzzer. Une sortie de puissance permet de commander un moteur ou un électroaimant.



Kit capteur de lumière

Carte équipée d'un Picaxe 08 M, d'un capteur de lumière LDR, d'un bouton poussoir, de 3 DEL (vert, jaune, rouge) et d'un buzzer.



Servopilot

Carte équipée d'un Picaxe 08 M. Les entrées/sorties sont configurables. Cependant, cette carte est adaptée à la commande de 3 servomoteurs.



Kit Pic Lock kit

Carte équipée d'un Picaxe 18, d'un clavier 12 touches, d'un buzzer, d'une DEL bicolore et d'une sortie de puissance permettant de commander un électroaimant.

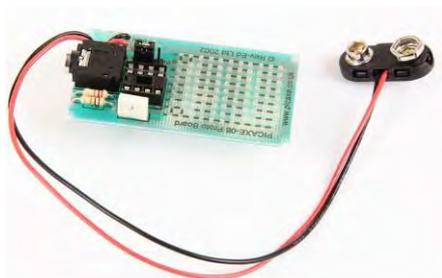


Picaxe-18 Tutorial Board

Carte non équipée de Picaxe (nécessite un Picaxe 18M ou 18X). Cette carte dispose d'un circuit de puissance ULN, d'un afficheur 7 segments, d'une LDR et de 2 boutons-poussoirs.

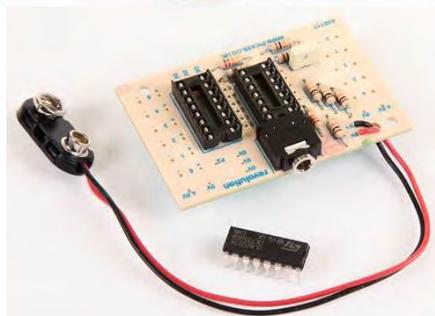
1.5 Les cartes de prototypage Picaxe

Les cartes de prototypage sont des montages de base pour la mise en œuvre des systèmes Picaxe. Elles doivent être complétées par le système Picaxe adapté, des composants (capteurs, actionneurs) et/ou des cartes annexes.



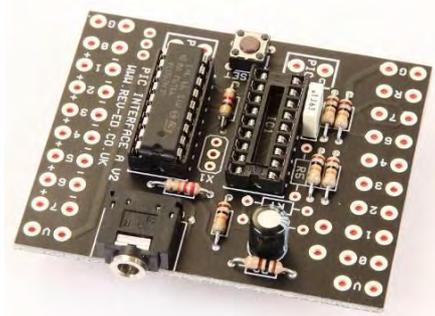
Picaxe 08 Proto Board

Prototypage rapide d'un Picaxe 8 pattes de type 08 ou 08M.



Picaxe 14M Project Board

Prototypage rapide d'un Picaxe type 14M.
6 sorties via ULN2803 permettent de commander jusqu'à 0,5 A.
5 entrées



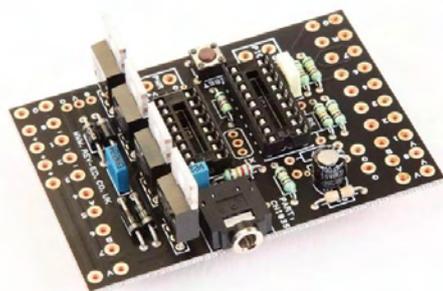
Picaxe 18 Project Board

Prototypage rapide d'un Picaxe type 18M ou 18X.
8 sorties via ULN2803 permettent de commander jusqu'à 0,5 A.
5 entrées



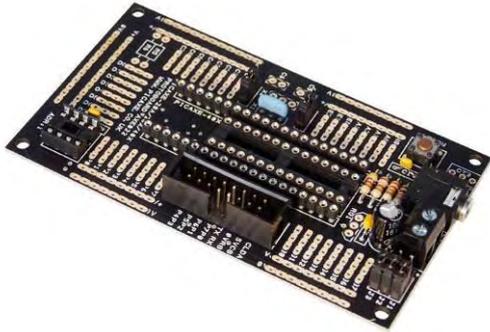
Picaxe 20M Project Board

Prototypage rapide d'un Picaxe type 20M.
8 sorties via ULN2803 permettent de commander jusqu'à 0,5 A.
8 entrées



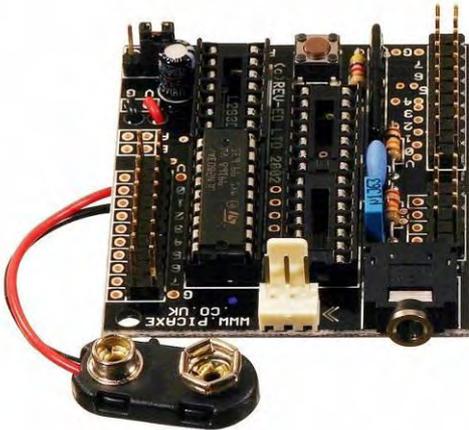
Picaxe 18 High Power

Prototypage rapide d'un Picaxe type 18M ou 18X.
4 sorties via Fet de puissance permettent de commander jusqu'à 4 A (à vérifier).
8 entrées
Possibilité d'installer un circuit de commande moteur L293d pour contrôler 2 moteurs à courant continu.



Picaxe 28X/40X Proto Board

Prototypage rapide d'un Picaxe type 28X ou 40X.
8 sorties directes Picaxe
8 entrées
4 entrées analogiques
Possibilité d'installer un circuit de mémoire type EEPROM 24LC.



Carte PowerProg (AXE020)

Prototypage rapide d'un Picaxe type 28X.
8 sorties via ULN2803 permettent de commander jusqu'à 0,5 A.
8 entrées
4 entrées analogiques
Possibilité d'installer un circuit de commande moteur L293D pour contrôler 2 moteurs à courant continu.

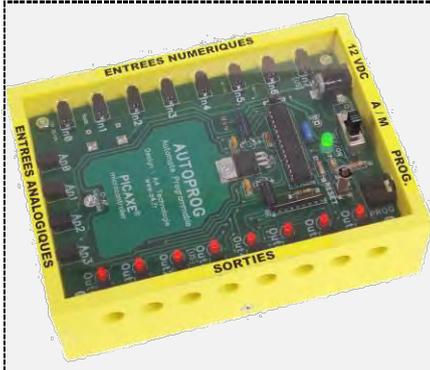


Pack Development Board

Carte de prototypage universelle pour Picaxe de types 8, 14, 18, 20, 28 ou 40.
Cette carte est équipée de capteurs, d'actionneurs, de supports de circuits intégrés et d'une plaque d'essai permettant d'expérimenter et de développer ses propres Travaux Pratiques.



1.6 Les modules AutoProg



Boîtier de commande « AutoProg »

Automate programmable du système AutoProg fonctionnant avec la technologie Picaxe.

Constitué d'un boîtier de commande programmable relié par des cordons de type jack à des modules capteurs ou actionneurs. Il dispose de 8 entrées numériques, 4 entrées analogiques et 8 sorties.

Module Microrupteur

Module Microrupteur de la gamme AutoProg permettant de se connecter facilement au boîtier automate avec un cordon de type jack.



Module Relais

Module relais de la gamme AutoProg permettant de piloter un circuit de puissance par 2 contacts RT.



Module Détecteur de marquage au sol

Module de la gamme AutoProg équipé de 3 capteurs infrarouge détectant des variations de couleur sur un sol.



Module Capteur de lumière

Module analogique de la gamme AutoProg dont le capteur résistif varie en fonction de la lumière.



Module Afficheur

Module afficheur LCD de la gamme AutoProg permettant un affichage de 2 lignes de 16 caractères.



Module Porte logique

Module de la gamme AutoProg permettant d'exploiter les fonctions logique de base : ET-OU-NON ET-NON OU



Le boîtier de commande ou automate « **AutoProg** » est système modulaire programmable fonctionnant avec la technologie Picaxe. Il est constitué d'un boîtier de commande programmable que l'on relie par des cordons type jack à des modules capteurs et actionneurs.

Tous ces modules sont visibles sur le site www.a4.fr

2. Présentation du logiciel « Picaxe Programming Editor »

| | |
|--|----|
| 2.1 Présentation du logiciel «Picaxe Programming Editor..... | 11 |
| 2.2 Premier lancement du logiciel..... | 12 |
| 2.3 Mise en service du câble de programmation..... | 14 |
| 2.4 Transfert d'un programme..... | 15 |
| 2.5 Table des symboles..... | 15 |
| 2.6 Paramétrage des options..... | 16 |

2.1 Présentation du logiciel Picaxe Programming Editor

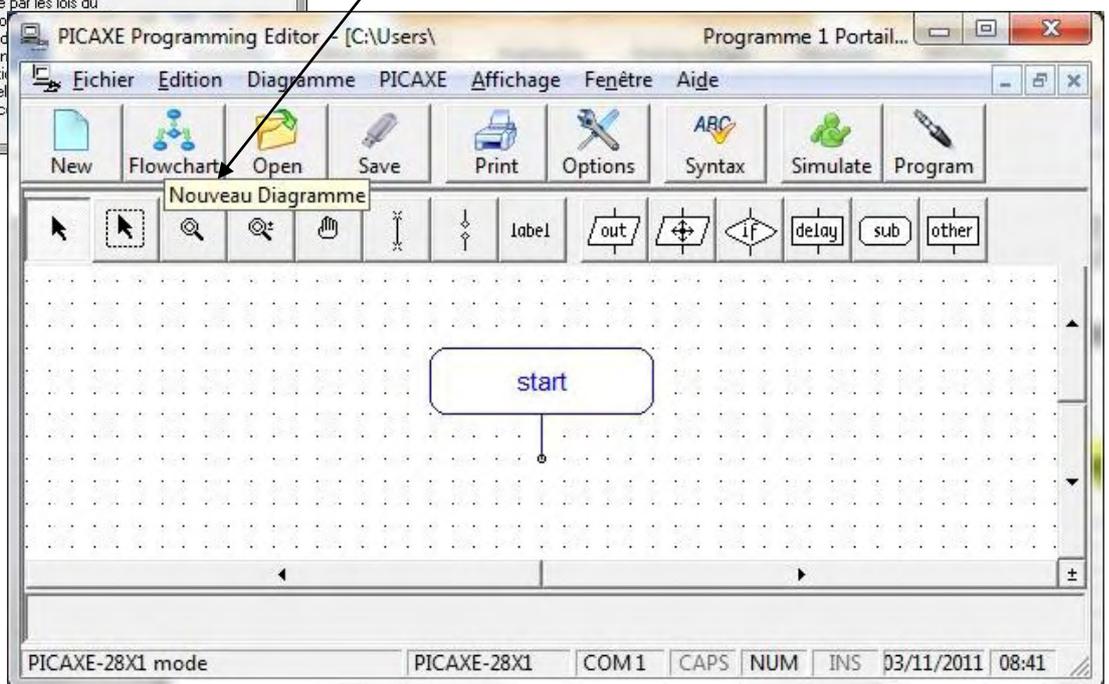


« **Picaxe Programming Editor** » est un logiciel permettant d'écrire des programmes en BASIC ou sous forme graphique (organigramme de programmation) puis de les transférer dans un microcontrôleur Picaxe.

A la mise en route, le logiciel « Picaxe Programming Editor » est en mode BASIC, ce mode de programmation n'est pas traité dans ce document.

La version utilisée dans ce guide est la « **5.3.2** ».

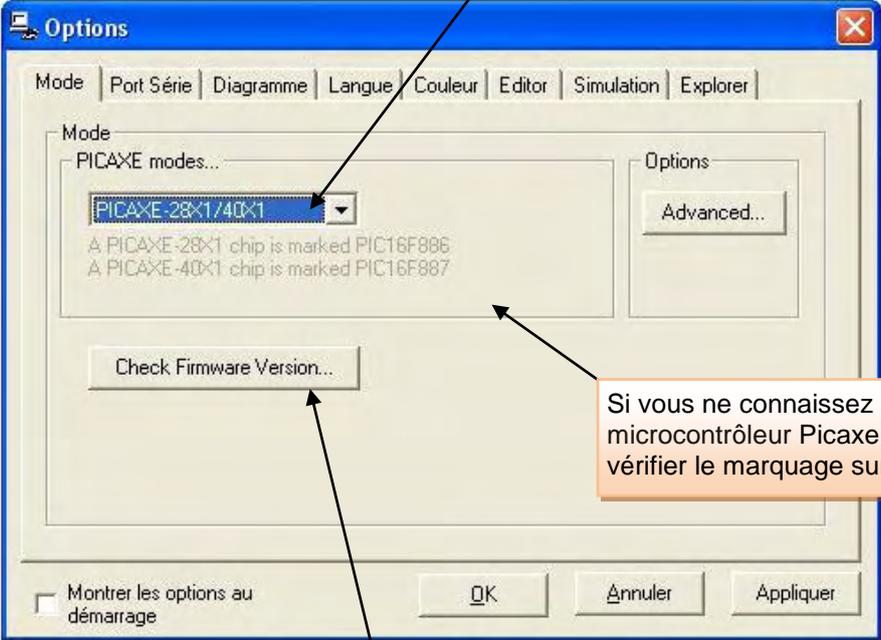
Pour se positionner en mode graphique, il faut cliquer sur l'icône « **Flowchart** ».



2.2 Premier lancement du logiciel

Au premier lancement du logiciel la fenêtre suivante s'affiche :
Il s'agit de la fenêtre « **Options** » qui permet de choisir le microcontrôleur Picaxe que l'on va programmer.

La liste suivante permet de sélectionner le type de microcontrôleur Picaxe de la carte d'application



Si vous ne connaissez pas le type de microcontrôleur Picaxe, vous pouvez vérifier le marquage sur le circuit intégré.

Permet de vérifier si l'ordinateur peut dialoguer avec le microcontrôleur Picaxe et affiche en retour son type.

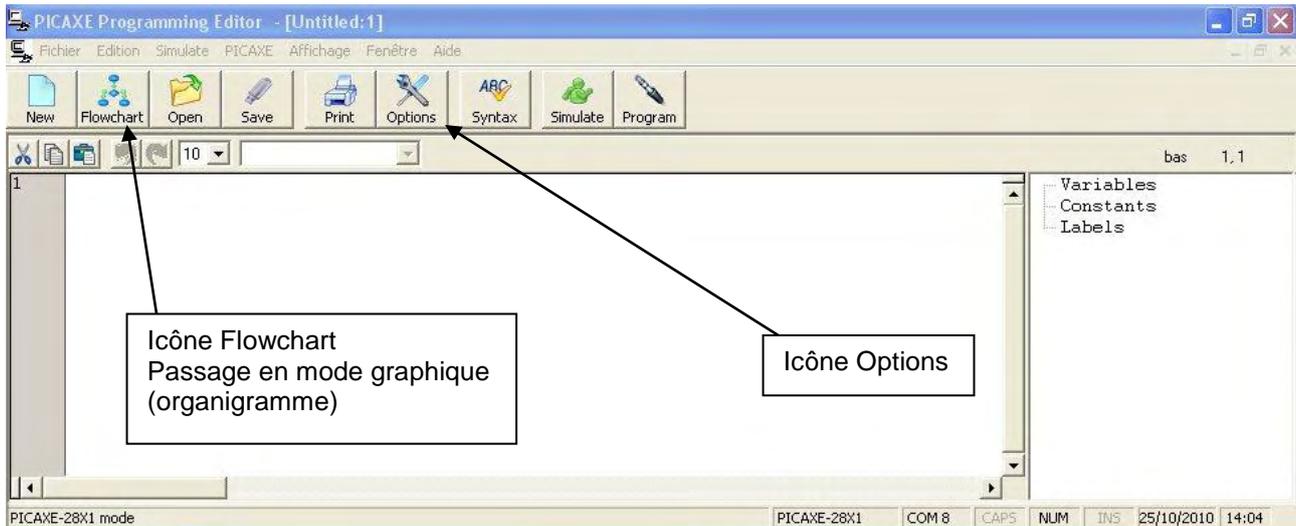
Si le dialogue est possible, la fenêtre suivante s'affiche :



En cas de problème, la fenêtre suivante s'affiche :

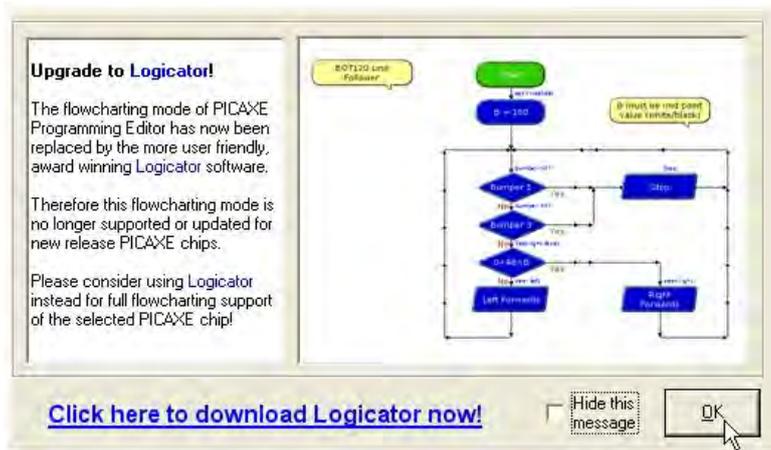


Lors du lancement du logiciel la première fenêtre affiche une feuille pour programmer en basic. Si vous souhaitez passer en mode graphique cliquer sur l'icône « **Flowchart** » (organigramme).

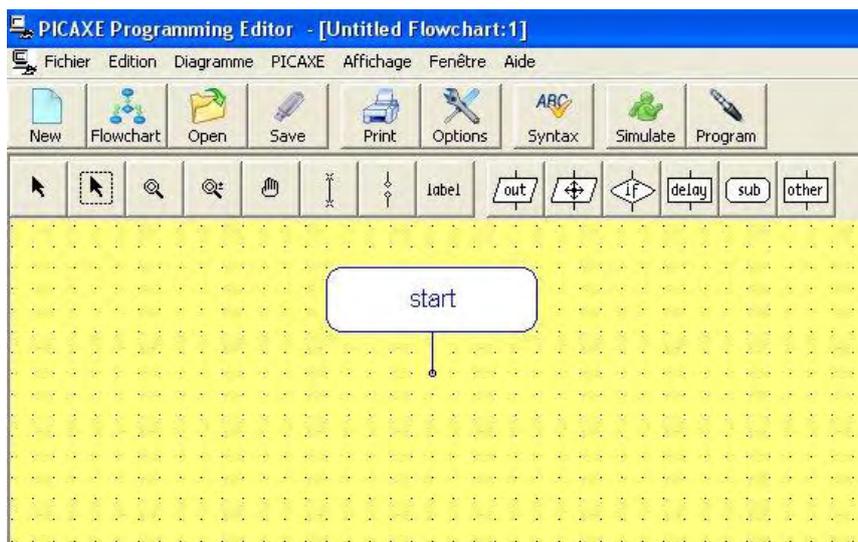


Icône Flowchart
Passage en mode graphique
(organigramme)

Icône Options



Validez avec OK



Mode graphique « Flowchart » (organigramme)

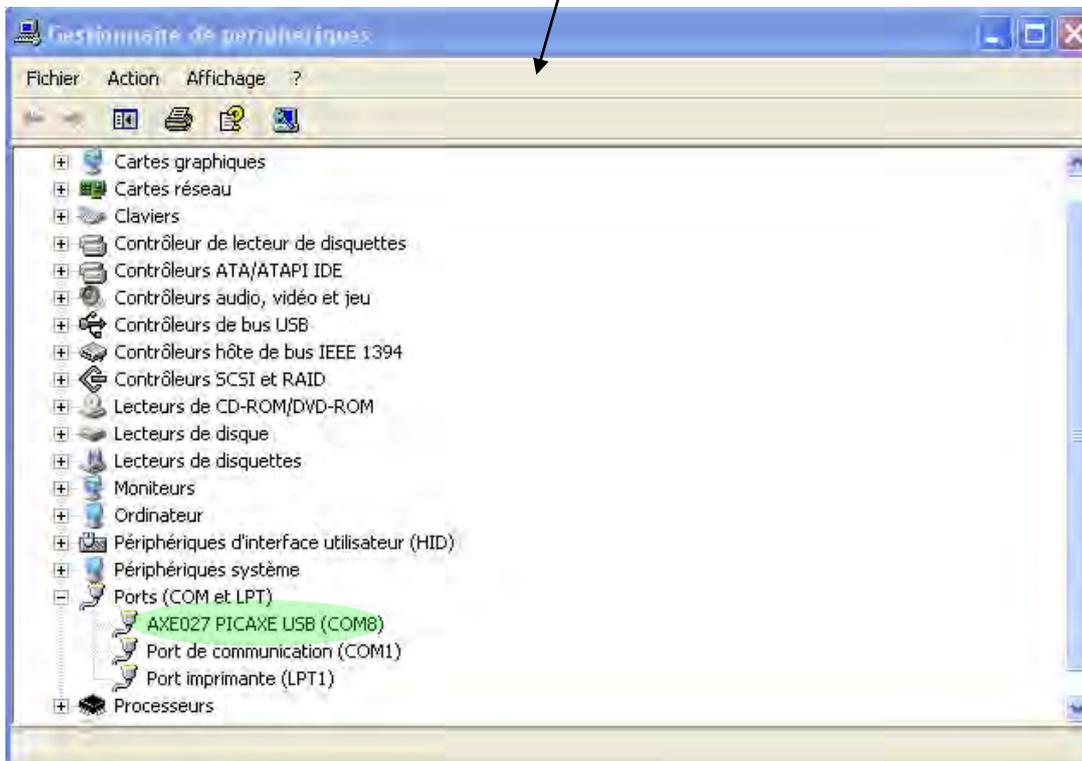
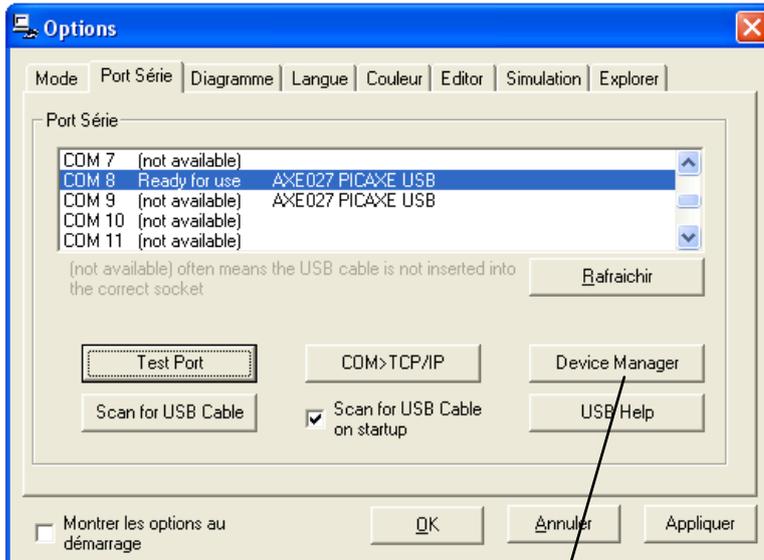
2.3 Mise en service du câble de programmation

La plupart des ordinateurs disposent d'un connecteur série 9 points auquel on peut connecter le câble de programmation pour port série (Réf. : **CABLE-FP**).

Pour les ordinateurs ne disposant pas de connecteur série, il est indispensable d'utiliser le port USB de l'ordinateur et le câble de programmation pour port USB (Réf. : **CABLE-USBPICAXE**).

La mise en œuvre du câble USB s'effectue avec les systèmes d'exploitation Windows 98SE, ME, 2000, 2003, XP, Vista et W7. Pour que le câble USB soit reconnu par le système d'exploitation, il est indispensable d'installer les pilotes correspondants.

La procédure d'installation est indiquée dans le dossier : « **Procédure installation câble réf CABLE-USBPICAXE.pdf** » sur le site www.a4.fr

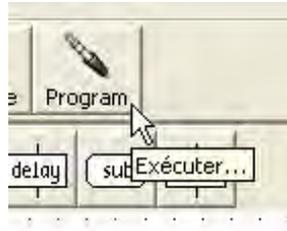


Le gestionnaire de périphérique indique la bonne installation du driver.

2.4 Transfert d'un programme

Mettre sous tension la platine d'essai ou de prototypage, sélectionner le port du câble de programmation (série RS232 ou USB) et raccorder l'ordinateur à la carte.

Cliquer sur l'icône « **Program** » ou appuyer sur la touche de fonction **F5**.



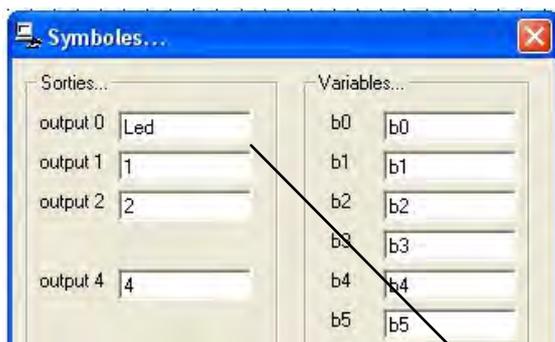
Si tout est en ordre, le programme est envoyé au microcontrôleur Picaxe et la fenêtre de chargement suivante s'affiche :



2.5 Table des symboles

Afin d'améliorer la lisibilité d'un programme, il est possible de changer le nom des symboles des entrées et des sorties dans la table des symboles.

Procédure : Aller dans le menu « **Diagramme** » puis activer la commande « **Table des symboles** ». Donner un nom à chaque entrée ou sortie.



Cela permet d'afficher

plutôt que

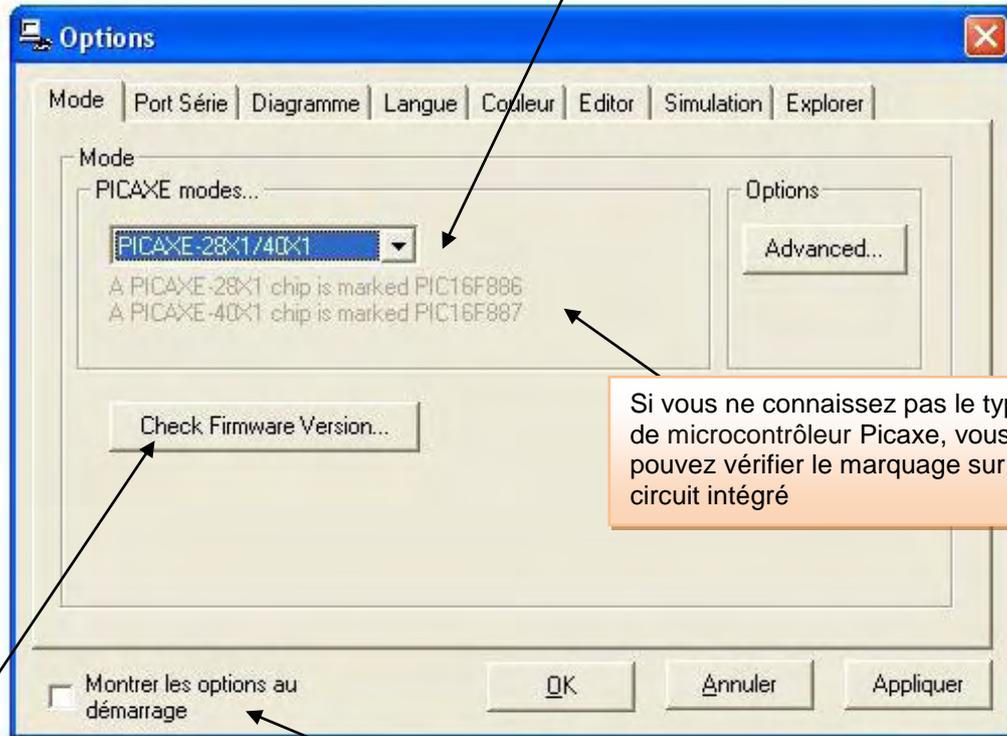
2.6 Paramétrage des options

Onglet Mode

Permet de choisir le microcontrôleur Picaxe que l'on va programmer.

Lorsque la sélection est faite, « **Picaxe Programming Editor** » initialise l'ensemble des instructions compatibles avec le système Picaxe choisi.

La liste suivante permet de sélectionner le type de microcontrôleur Picaxe



Si vous ne connaissez pas le type de microcontrôleur Picaxe, vous pouvez vérifier le marquage sur le circuit intégré

Permet d'afficher le menu Options au démarrage du logiciel

Permet de vérifier si l'ordinateur peut dialoguer avec le microcontrôleur Picaxe

Si le dialogue est possible, la fenêtre suivante s'affiche :

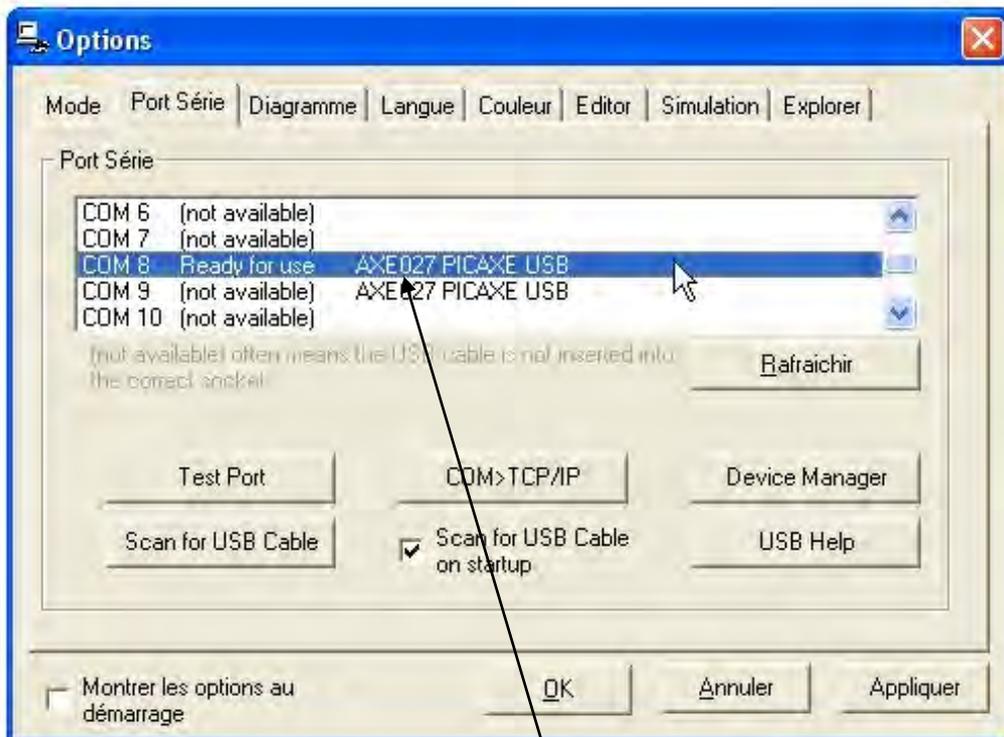
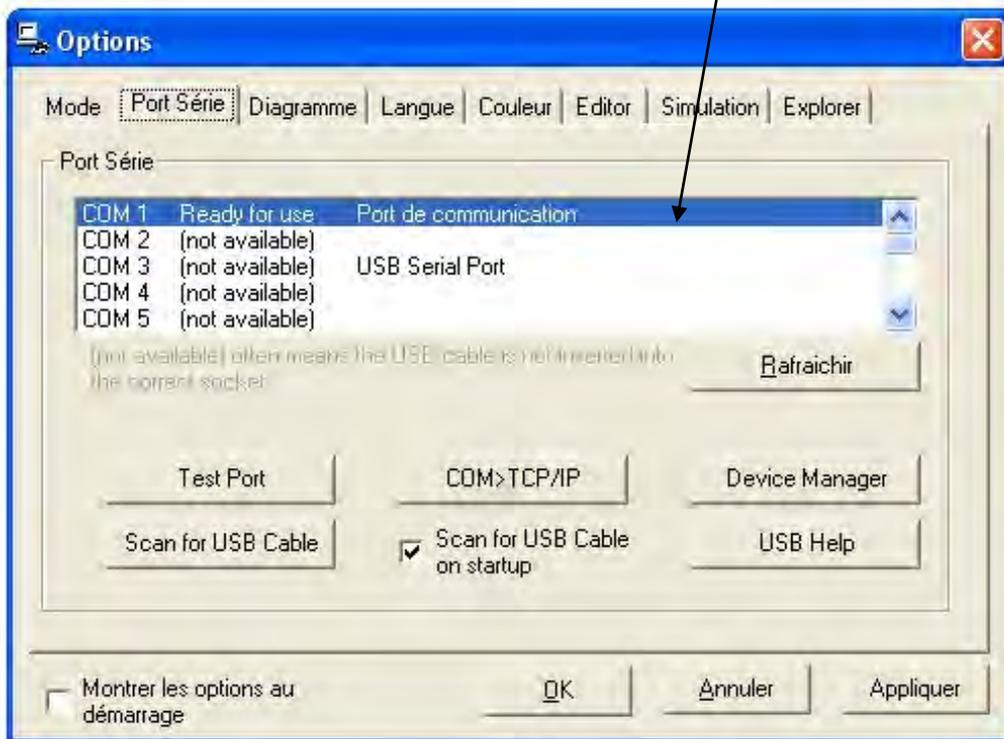


Cette information est différente en fonction du type de Picaxe connecté...

Onglet Port série

Permet de choisir sur quel port est relié le câble de liaison Ordinateur ↔ Picaxe

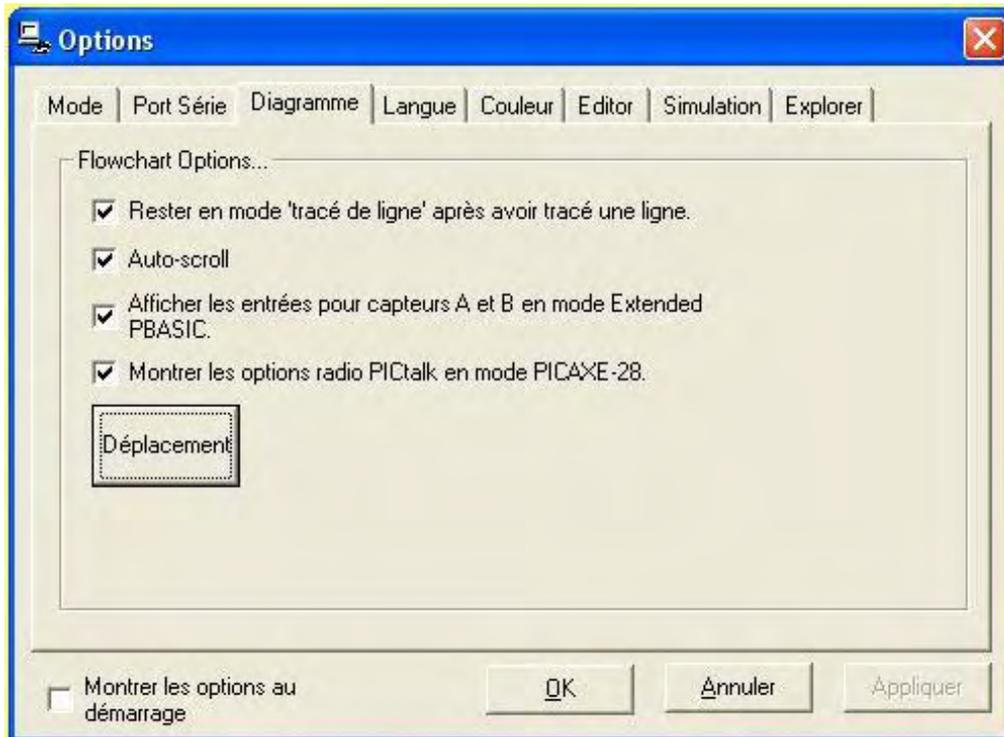
Le COM1 est en général le port série RS232



Si le câble USB AXE027 est relié et que le pilote est correctement installé, l'affichage devient « Ready for use »

Onglet Diagramme

Permet de choisir les options d'affichage et de fonctionnement du mode graphique « **Flowchart** » (organigramme). Vous pouvez cocher toutes les options indiquées.



La touche « **Déplacement** » est très importante. Elle permet de rendre compatible les blocs de déplacement (mouvement) avec n'importe quelle carte Picaxe.



Pour un circuit 14,18, 20, 40 broches

Le réglage par défaut indique la configuration :

- Forward correspond à la mise à 1 des sorties 7 et 5
- Back correspond à la mise à 1 des sorties 6 et 4
- Left correspond à la mise à 1 des sorties 7 et 4
- Right correspond à la mise à 1 des sorties 6 et 5



Pour un circuit 8 broches

Le réglage par défaut indique la configuration :

- Forward correspond à la mise à 1 des sorties 2 et 0
- Back correspond à la mise à 1 des sorties 4 et 1
- Left correspond à la mise à 1 des sorties 4 et 0
- Right correspond à la mise à 1 des sorties 2 et 1

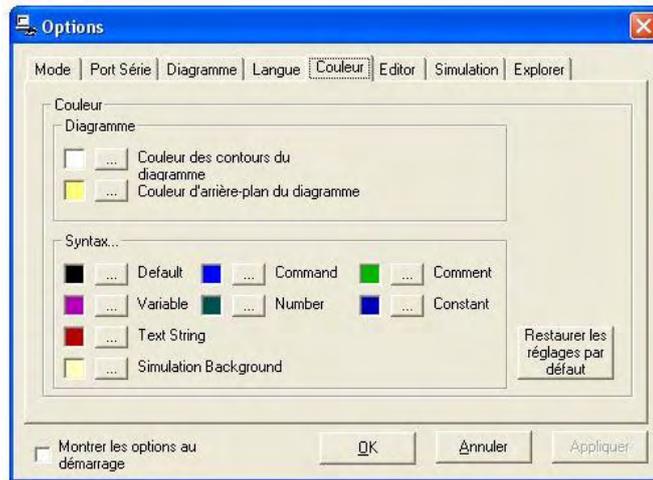
Onglet Langue

Permet de choisir la langue des menus de Picaxe Programming Editor. Cela ne concerne pas les commandes BASIC qui seront toujours écrites en anglais.



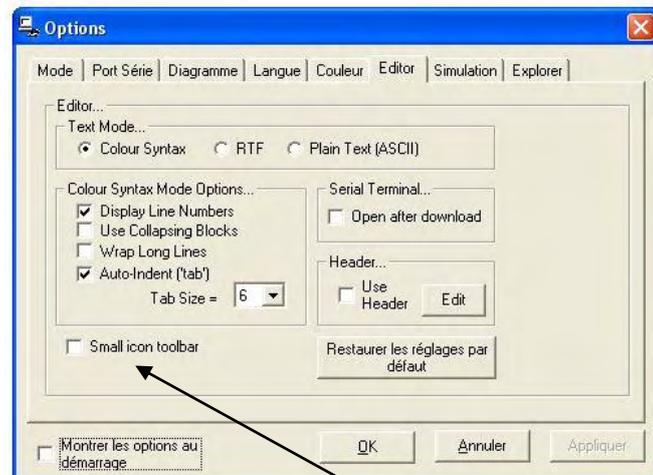
Onglet Couleur

Permet de choisir les couleurs de fonctionnement de Picaxe Programming Editor.



Onglet Editor

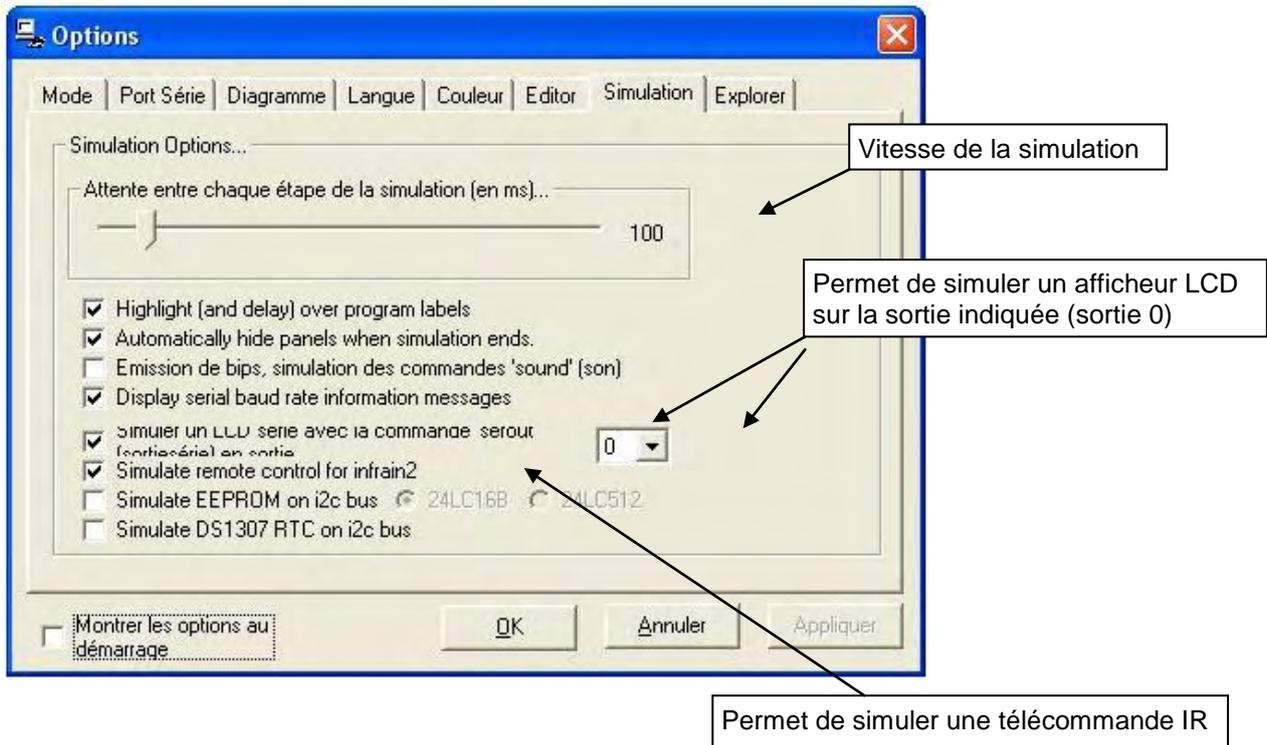
Il s'agit de modes avancés sur l'écriture en BASIC. Le mode par défaut est sans effet sur le mode **Flowchart**.



Permet d'afficher des icônes réduites

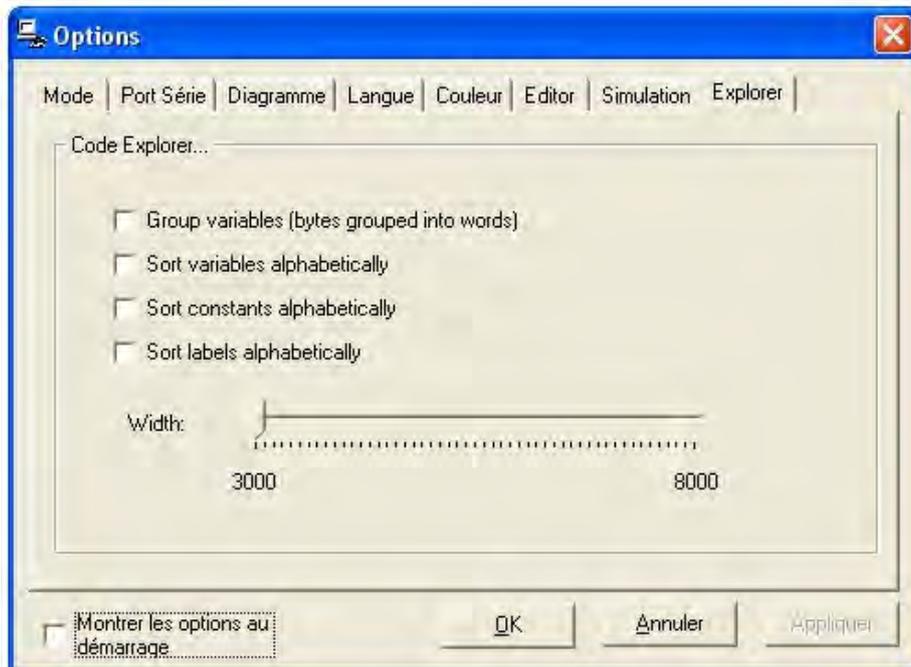
Onglet Simulation

Il s'agit d'adapter le logiciel avec le mode simulation. Pour simuler un afficheur LCD ainsi qu'une télécommande IR, n'oubliez pas de cocher les cases correspondantes.



Onglet Explorer

Il s'agit de modes avancés sur l'écriture en BASIC.



3. Guide d'utilisation du logiciel Picaxe Programming Editor

| | |
|--|----|
| 3.1 La barre d'outils du logiciel « Picaxe Programming Editor » | 22 |
| 3.2 Les outils pour la programmation en mode graphique | 23 |
| 3.3 Les commandes de sorties | 24 |
| Exemple d'utilisation de la commande high..... | 25 |
| Exemple d'utilisation de la commande pins..... | 26 |
| Exemple d'utilisation de la commande toggle..... | 27 |
| Exemple d'utilisation de la commande sound..... | 28 |
| Exemple d'utilisation de la commande serout | 29 |
| Exemple d'utilisation de la commande servo..... | 30 |
| Exemple d'utilisation de la commande play avec un circuit 08M..... | 31 |
| Exemple d'utilisation de la commande play..... | 32 |
| Exemple d'utilisation de la commande tune avec un circuit 08M | 33 |
| Exemple d'utilisation de la commande tune | 34 |
| Exemple d'utilisation de la commande irout | 35 |
| Exemple d'utilisation de la commande BASIC..... | 36 |
| 3.4 Les commandes de mouvements | 37 |
| Exemple d'utilisation des commandes forward, back, right, left, halt | 38 |
| Exemple d'utilisation de la commande speed..... | 39 |
| Exemple d'utilisation de la commande ultra | 40 |
| Exemple d'utilisation de la commande infra | 41 |
| 3.5 Les commandes de tests | 42 |
| Exemple d'utilisation de la commande de test pin | 43 |
| Exemple d'utilisation de la commande de test var | 44 |
| 3.6 Les commandes de temporisation | 45 |
| Exemple d'utilisation de la commande pause et wait | 46 |
| Exemple d'utilisation de la commande sleep et nap..... | 47 |
| 3.7 Les commandes de sous-programme | 48 |
| Exemple d'utilisation de la commande stop..... | 49 |
| Exemple d'utilisation des commandes de sous-programme | 50 |
| Exemple d'utilisation des commandes d'interruptions..... | 51 |
| 3.8 Les autres commandes | 53 |
| Exemple d'utilisation de la commande let..... | 54 |
| Exemple d'utilisation de la commande readadc | 55 |
| Exemple d'utilisation de la commande debug | 56 |
| Exemple d'utilisation de la commande random | 57 |
| Exemple d'utilisation de la commande write..... | 58 |
| Exemple d'utilisation de la commande read | 59 |
| Exemple d'utilisation de la commande poke..... | 60 |
| Exemple d'utilisation de la commande peek | 61 |
| Exemple d'utilisation de la commande temp | 62 |

3.1 La barre d'outils du logiciel « Picaxe Programming Editor »

Les boutons standards



Permet d'écrire une nouvelle feuille BASIC
Il s'agit du mode par défaut à la mise en route du logiciel.



Permet d'écrire une nouvelle feuille Diagramme
Ce mode permet d'effectuer une représentation graphique de l'enchaînement d'opérations et de décisions effectuées par un programme (diagramme – organigramme de programmation). Il s'agit du mode d'utilisation de ce guide.



Permet d'ouvrir un fichier réalisé avec Picaxe Programming Editor.
*La fenêtre de sélection du fichier à ouvrir affiche les fichiers de type BASIC (extension *.bas). Pour ouvrir un fichier de type diagramme, il faudra demander le type Diagramme (extension *.cad)*



Permet de sauvegarder votre feuille de travail.



Permet d'imprimer votre feuille de travail.



Permet d'ouvrir le menu options.
Les détails de ce menu se retrouvent en page 18.



Permet de vérifier la syntaxe de l'écriture en mode BASIC.
Cet outil ne permet pas de vérifier un diagramme. En mode BASIC, cela permet de vérifier la mémoire utilisée sur le Picaxe. Le succès de la vérification précise uniquement si l'écriture du programme est correcte, cela ne vérifie pas le bon fonctionnement du programme.



Permet d'activer le mode Simulation.
Ce mode permet de vérifier l'état des variables et des sorties du Picaxe. Il permet ainsi de tester le bon déroulement d'un programme.



Permet de télécharger dans le microcontrôleur Picaxe le programme.
Le téléchargement s'effectue uniquement si la liaison est correcte (câble USB installé, Picaxe sélectionné et alimenté). Un seul programme par Picaxe.

3.2 Les outils pour la programmation en mode graphique

Les outils d'édition



Sélection d'un bloc.
Cliquer sur le bloc et le déplacer en maintenant le bouton gauche de la souris enfoncé.



Permet de sélectionner un ensemble de blocs.
*Maintenir le bouton gauche de la souris enfoncé et déplacer la souris pour sélectionner la zone.
Cliquer sur un bloc et déplacer l'ensemble en maintenant le bouton gauche de la souris enfoncé.*



Permet d'effectuer un Zoom sur une fenêtre de sélection.
Maintenir le bouton gauche de la souris enfoncé et déplacer la souris pour sélectionner la zone où zoomer.



Permet d'effectuer un Zoom + ou -
Maintenir le bouton gauche de la souris enfoncé et déplacer la souris.
Nota : le menu *Diagramme / Ajuster* permet d'ajuster le diagramme à la taille de l'écran.



Permet de déplacer l'ensemble de l'organigramme.
Maintenir le bouton gauche de la souris enfoncé et déplacer la souris.



Permet de dessiner des lignes pour relier les blocs.
*Approcher la souris du bloc de départ, un cercle rouge apparaît, cliquer avec le bouton gauche, s'approcher du bloc de destination, un cercle rouge apparaît, cliquer de nouveau pour établir la liaison.
Cliquer sur une liaison pour l'effacer.*
Note : le menu *Options / Diagramme* permet de sélectionner l'option « Rester en mode tracé de ligne » afin de maintenir l'activation de cet outil.



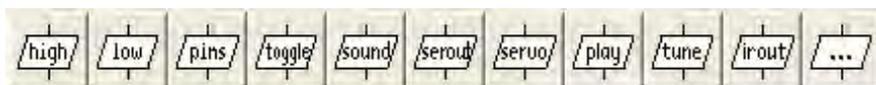
Permet de positionner des points de connexions entre des blocs.
Cliquer sur le bloc de départ, 2 points de connexion numérotés apparaissent. Cliquer avec le bouton droit de la souris puis sélectionner le deuxième point de connexion avec le bouton gauche de la souris et déplacer sur le bloc de destination.



Permet d'écrire des commentaires.
Cliquer à l'endroit où l'on souhaite écrire un commentaire puis écrire le commentaire à l'aide du clavier.

3.3 Les commandes de sorties

Ce groupe de commandes permet d'agir sur les sorties d'un microcontrôleur Picaxe.



Permet d'activer une sortie (mise à l'état haut de la sortie = présence de tension).

Le champ de saisie situé en bas à gauche de l'écran permet de sélectionner la sortie à activer.

Nota : lorsque cette commande est exécutée, la sortie concernée reste active en permanence jusqu'à temps qu'une autre commande ne modifie son état.



Permet de désactiver une sortie (mise à l'état bas de la sortie = absence de tension).

Le champ de saisie situé en bas à gauche de l'écran permet de sélectionner la sortie à désactiver.

Nota : lorsque cette commande est exécutée, la sortie concernée reste inactive en permanence jusqu'à ce que 'une autre commande ne modifie son état.



Permet d'activer et de désactiver l'ensemble des sorties.

Nota : cette commande remplace avantageusement une succession de high et low.



Permet d'effectuer le complément à 1 de l'état d'une sortie. Si la sortie sélectionnée est à 0, elle passe à 1 ou alors si la sortie est à 1, elle passe à 0.



Permet de produire un son sur la sortie sélectionnée. Il est possible d'agir sur la fréquence émise et la durée.



Permet de transmettre des données sur la sortie sélectionnée.

Nota : Son utilisation la plus courante est la commande de l'afficheur LCD.



Permet de positionner un servomoteur.

Nota : la commande servo utilise des ressources internes du Picaxe et interfère avec des commandes complexes comme pause, debug ou serout.



Permet de jouer un air de musique. Le choix s'effectue sur 4 mélodies qui sont préprogrammés dans le Picaxe. (mélodie : Joyeux anniversaire, Jingle Bells, Douce nuit, Rudolph the Red Nosed Reindeer

Nota : les options de cette commande sont différentes en fonction du circuit Picaxe utilisé.



Permet de jouer un air de musique de sa composition.

Nota : les options de cette commande sont différentes en fonction du circuit Picaxe utilisé.



Permet de transmettre un signal infrarouge en utilisant une diode infrarouge. La transmission de donnée utilise le protocole SIRC (Sony Infra Red Control).

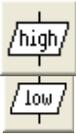
Nota : cette commande est disponible sur les circuits disposants d'une sortie infraout : 08M, 14M et 20M



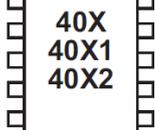
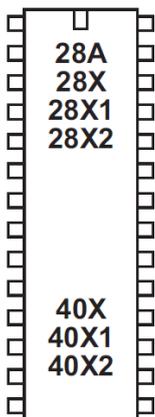
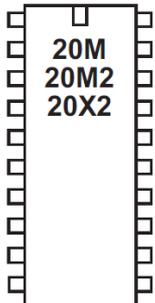
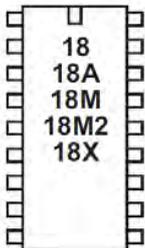
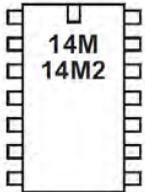
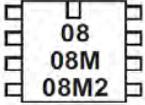
Permet d'écrire une commande en BASIC.

Nota : cela permet d'avoir accès à des fonctions non accessibles avec le mode Flowchart (organigramme).

Exemple d'utilisation des commandes high et low



Compatibilité



Fonction :

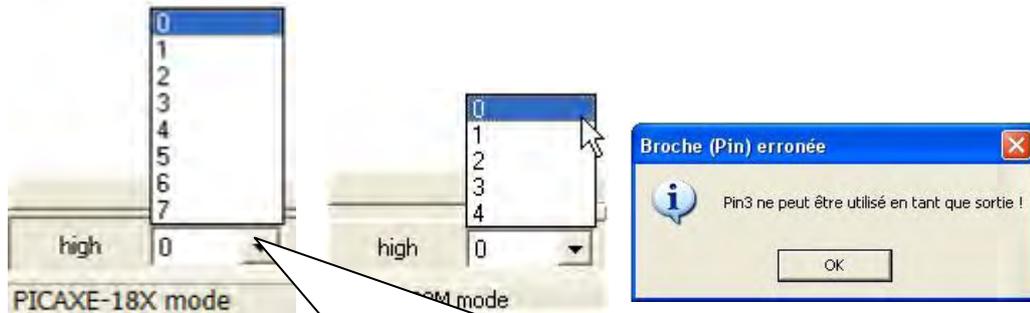
Positionne une sortie à l'état haut (high).

Fonction :

Positionne une sortie à l'état bas (low).

Cette commande fonctionne à l'opposé de la commande high.

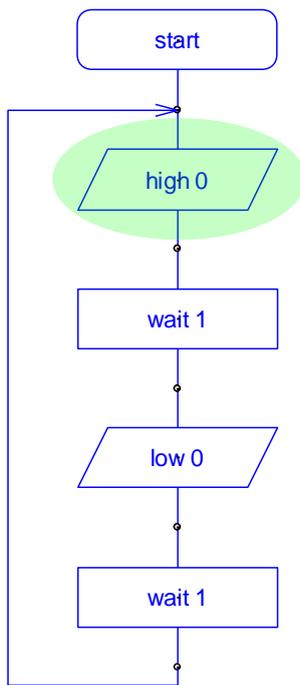
Option sélectionnable :



Le numéro sélectionné correspond à la broche de sortie à activer

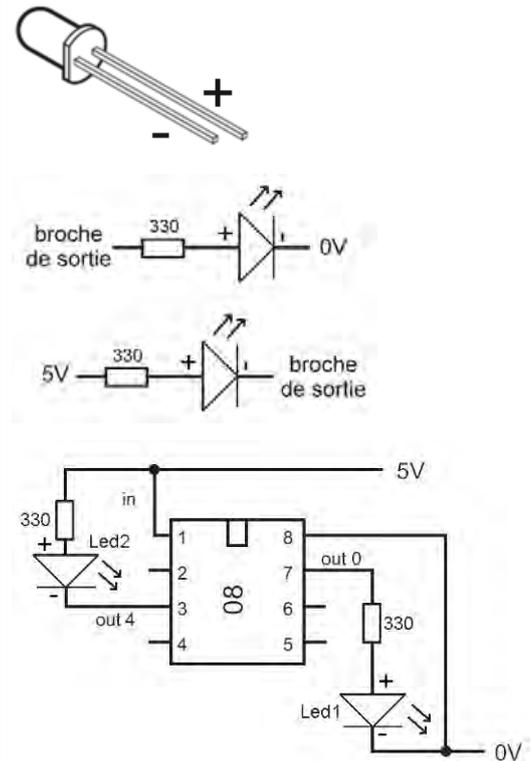
Nota : certains microcontrôleurs qui disposent d'entrées/sorties configurables (ex. Picaxe 08) configurent la broche sélectionnée en temps que sortie. Si une sortie n'est pas disponible, une fenêtre indique l'erreur.

Exemple de programme :



La sortie 0 est configurée en temps que sortie. Elle est activée à l'état haut pendant 1 seconde puis est désactivée pendant 1 seconde et ainsi de suite.

Exemple de câblage LED :

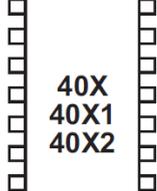
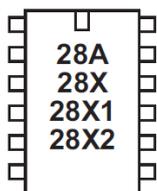
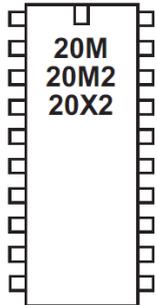
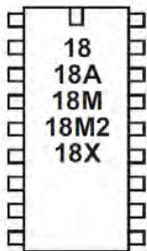
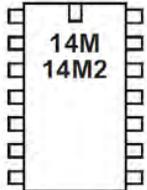
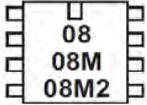


Une sortie peut commandée une Led par rapport au 0v ou par rapport au 5V. La Led1 s'allume quand la sortie Out 0 est high La Led2 s'allume quand la sortie Out 4 est low

Exemple d'utilisation de la commande pins



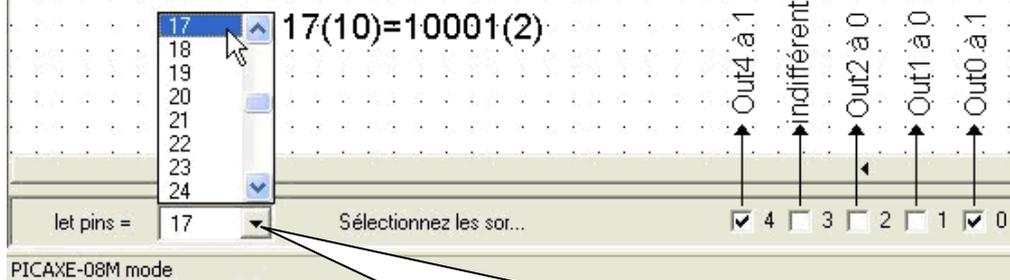
Compatibilité



Fonction :

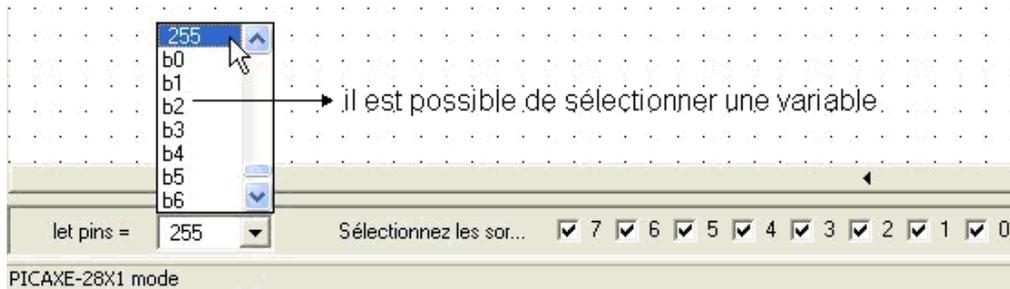
Permet d'activer ou de désactiver l'ensemble des sorties. Cette commande remplace avantageusement une succession de high et de low.

Option sélectionnable :



La valeur décimale correspond à la valeur binaire des sorties.

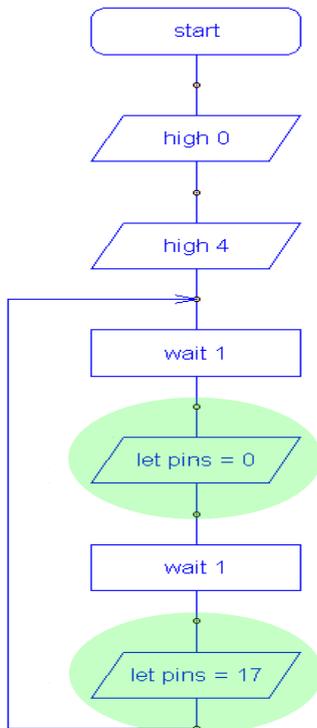
Nota : les microcontrôleurs qui disposent d'entrées/sorties configurables (ex. Picaxe 08) ne configurent pas les broches sélectionnées en temps que sortie. Il faut utiliser high ou low. Si une sortie n'est pas disponible, l'état 0 ou 1 est indifférent (exemple pour Out3).



il est possible de sélectionner une variable.

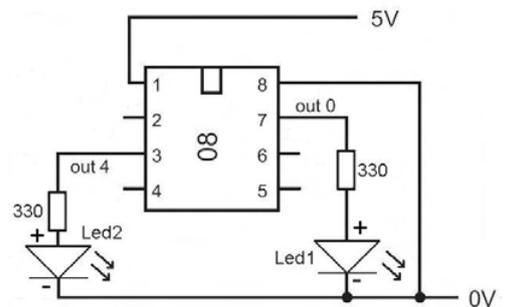
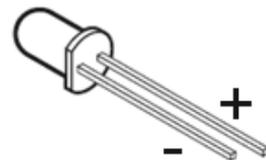
Pour un Picaxe 28X1, la valeur décimale évolue de 0 à 255.

Exemple de programme :



Les sorties 0 et 4 sont configurées en temps que sortie. Pins = 0 positionne les sorties à 0. pins = 17 positionne les sorties 0 et 4 à 1.

Exemple de câblage LED :



Led1 et Led2 s'allument quand les sorties Out 0 et Out 4 sont à high.

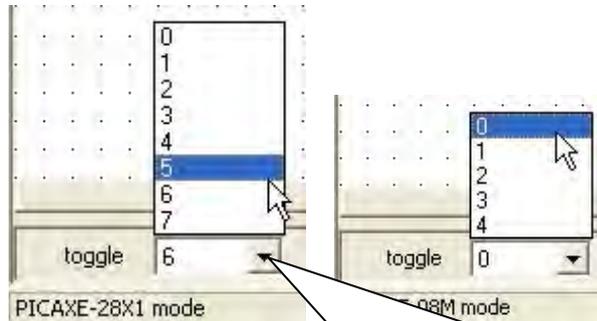
Led1 est sur la sortie Out 0 et Led2 est sur la sortie Out 4.



Fonction :

Permet d'inverser l'état d'une sortie. Met une sortie à 0 si elle est à 1 ou à 1 si elle est à 0. On utilise aussi le terme de « complément à 1 ».

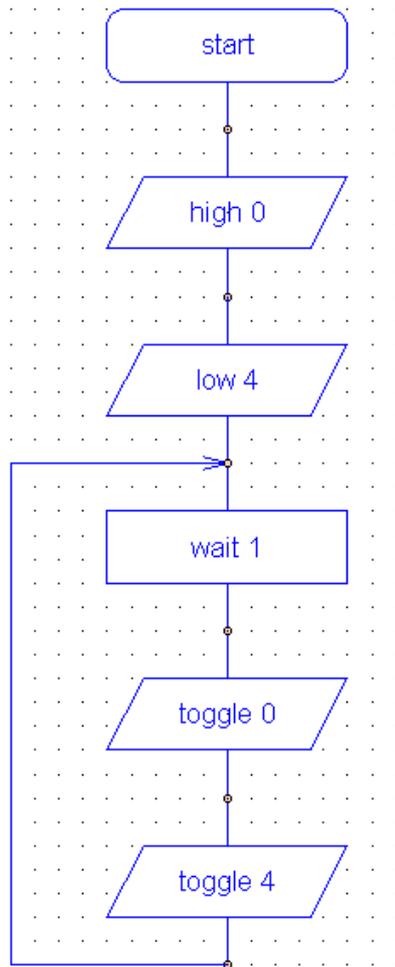
Option sélectionnable :



Le numéro sélectionné correspond à la broche de sortie à inverser

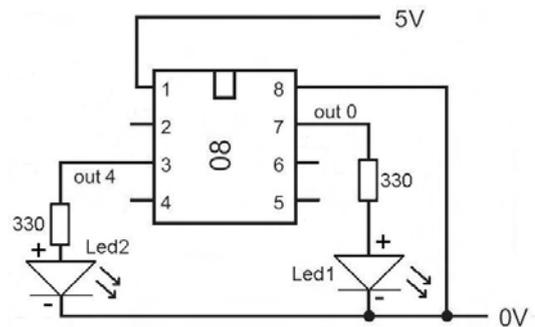
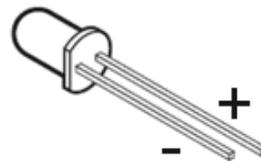
Nota : si une sortie n'est pas disponible, une fenêtre indique l'erreur.
Les microcontrôleurs qui disposent d'entrées/sorties configurables (ex. Picaxe 08) ne configurent pas les broches sélectionnées en temps que sortie. Il faut utiliser high ou low.

Exemple de programme :



Les sorties 0 et 4 sont inversées chaque fois qu'elle rencontre la commande toggle.

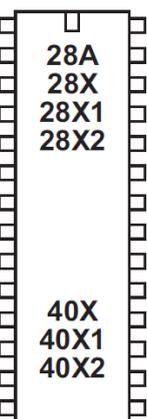
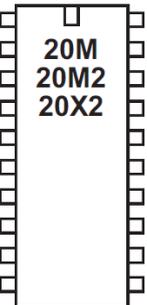
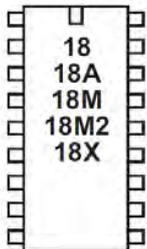
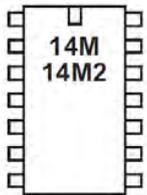
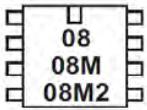
Exemple de câblage LED :



Led1 et Led2 s'allument quand les sorties Out 0 et Out 4 sont à high.

Led1 est sur la sortie Out 0 et Led2 est sur la sortie Out 4.

Compatibilité



Exemple d'utilisation de la commande sound

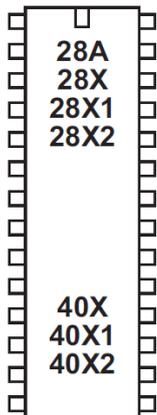
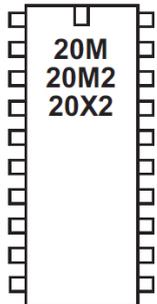
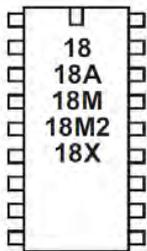
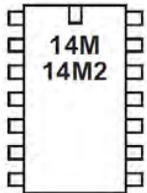
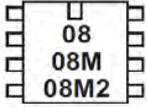


Fonction :

Permet de produire un son sur la sortie sélectionnée. Il est possible d'agir sur la fréquence émise et sur la durée de la note.

Option sélectionnable :

Compatibilité



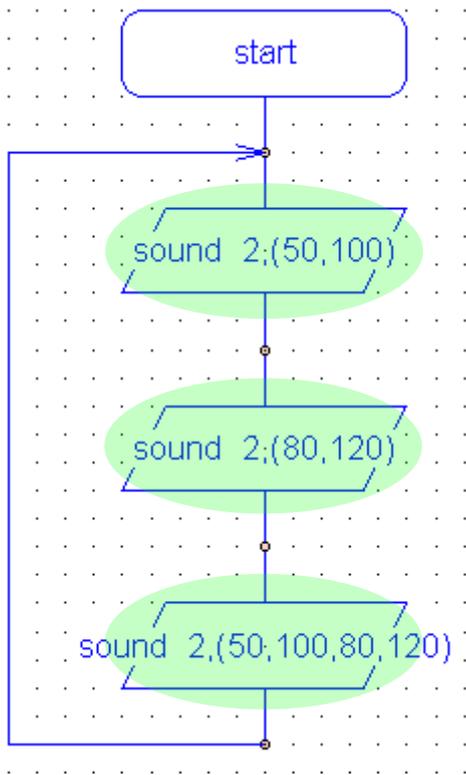
Choix de la sortie

Note : fréquence produite
Valeur de 0 à 255,
Variable possible

Durée de la note
Valeur de 0 à 255
Durée ≈ var x 10ms
Variable possible

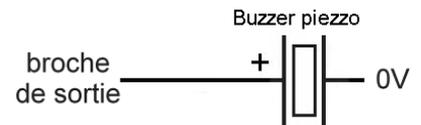
Nota : le meilleur rendement du buzzer est obtenu avec une valeur de fréquence d'environ 120.

Exemple de programme :

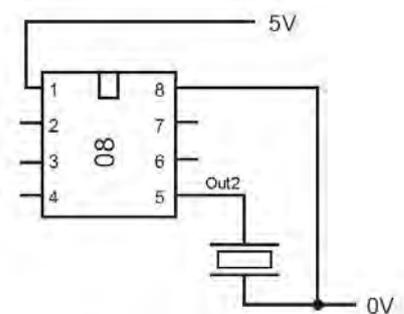
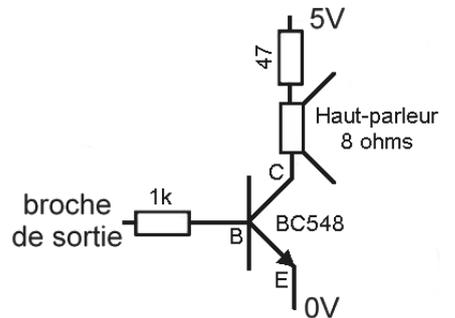


Le buzzer est sur la sortie 2. émet des notes de différentes fréquences et de durée différentes. Il est possible d'écrire plusieurs notes et durées sur la même commande.

Exemple de câblage d'un buzzer piezo :



Exemple de câblage d'un haut-parleur :



Le buzzer piezo est sur Out2.

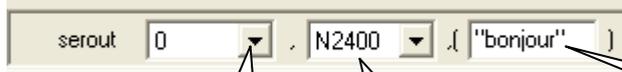
Exemple d'utilisation de la commande serout



Fonction :

Permet de transmettre des données sur la sortie sélectionnée.

Option sélectionnable :

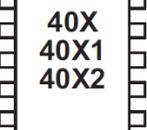
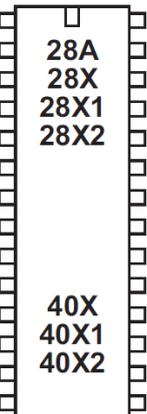
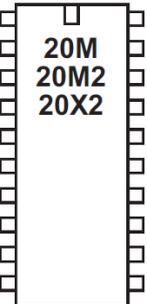
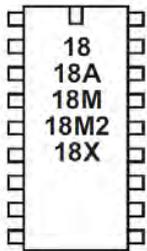
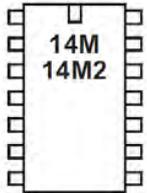
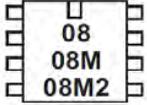


Choix de la sortie

Mode de transmission : N
Vitesse de transmission : 2400

Donnée à envoyer
Les " " indiquent qu'il s'agit d'un message.

Compatibilité

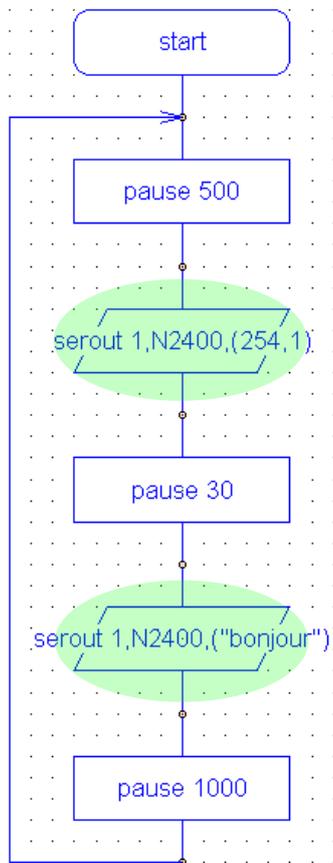


Nota : son utilisation la plus courante est la commande de l'afficheur LCD. La note d'application AXE033 donne les informations nécessaires à l'utilisation de l'afficheur LCD.

Exemple de programme :

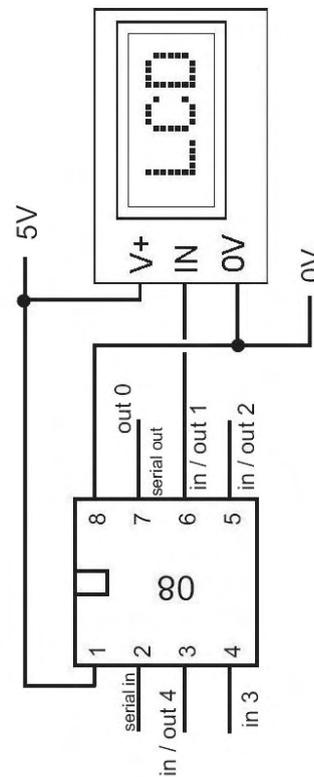
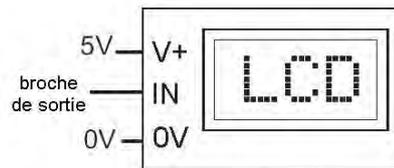
L'instruction serout1,N2400,(254,1) permet d'effacer l'afficheur.

L'instruction serout1,N2400,("bonjour") permet d'afficher le mot "bonjour".



"bonjour" clignote sur l'afficheur.

Exemple de câblage d'un afficheur LCD :



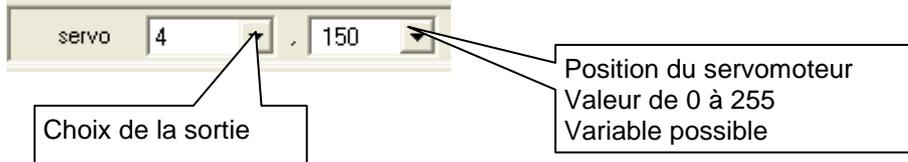
L'afficheur LCD est sur la sortie Out 1.

Exemple d'utilisation de la commande servo



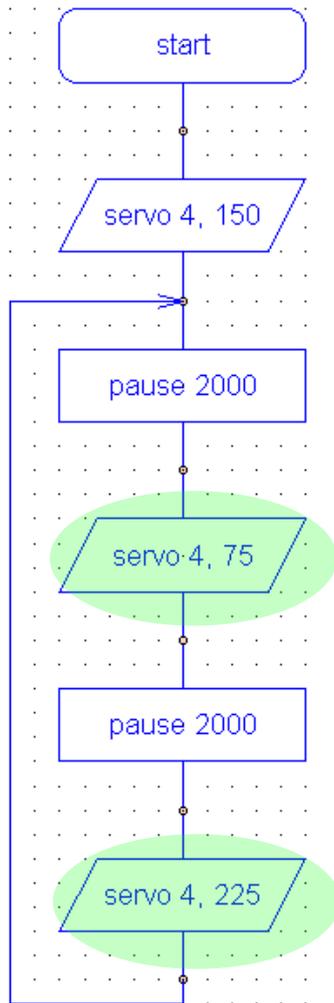
Fonction :
Permet de commander un servomoteur.

Option sélectionnable :



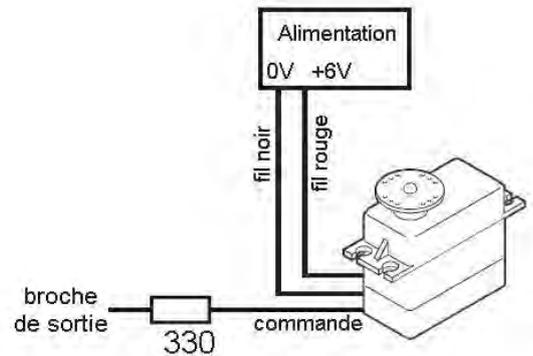
Nota : il est prudent de ne pas utiliser une valeur de position inférieure à 75 ou supérieure à 225 susceptible d'endommager certain servomoteur : il faudra tester les valeurs maximales et minimales « en situation ». La commande servo utilise des ressources internes du Picaxe et interfère avec des commandes complexes comme pause, debug ou serout.

Exemple de programme :



Le servomoteur est connecté à Out 4. La commande servo 4,150 positionne le servomoteur en position neutre. La commande servo 4,75 positionne le servomoteur en position max droite. La commande servo 4,225 positionne le servomoteur en position max gauche.

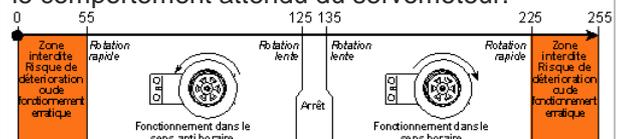
Exemple de câblage d'un servomoteur :



Un **servomoteur** est la combinaison d'un motoréducteur avec un système de position ; l'ensemble permet ainsi de disposer d'une position angulaire dépendant d'une valeur numérique. Le servomoteur permet ainsi de réaliser un système de direction ou de commandes d'une porte de portail.

Il existe aussi des servomoteurs à rotation continue permettant de réaliser des robots mobiles. Il s'agit de servomoteurs non limités en rotation qui permettent d'actionner une roue. L'intérêt de ces servomoteurs est de gérer la vitesse et le sens de rotation des moteurs avec un unique fil.

Ci-dessous tableau de correspondance du nombre à paramétrer dans la case "servo" selon le comportement attendu du servomoteur.





Fonction :

Permet de jouer un air de musique et de faire clignoter des leds.
Le choix s'effectue sur 4 mélodies et sur le mode de fonctionnement du clignotement.

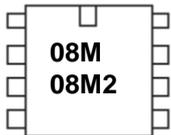
Option sélectionnable :



Options de sortie du clignotement
Valeur de 0 à 3

Choix de la mélodie
Valeur de 0 à 3

Compatibilité



Les options de sortie du clignotement sont :

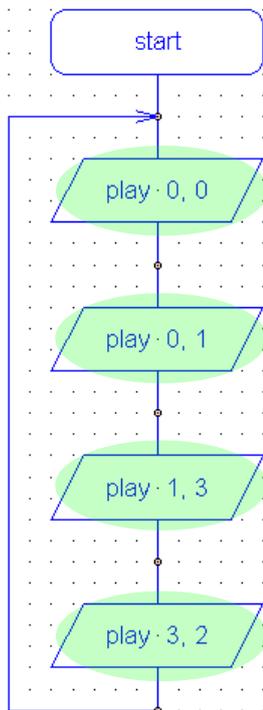
- 0 - pas de sorties utilisées
- 1 - clignotement de la sortie 0, Led1
- 2 - clignotement de la sortie 4, Led2
- 3 - clignotement alternatif de la sortie 0 et 4, Led1 et Led2

Les mélodies disponibles sont :

- 0 - Joyeux anniversaire
- 1 - Jingle Bells
- 2 - Douce nuit
- 3 - Rudolph the red nosed reinder

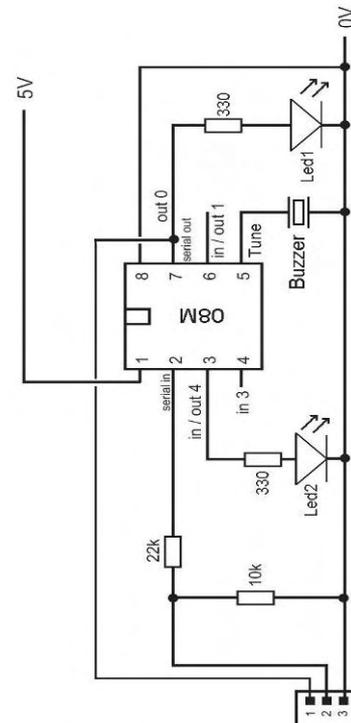
Nota : cette commande est disponible sur la sortie Tune du Picaxe 08M (pin 5 ou Out2), il est indispensable d'utiliser un buzzer piezzo ou un haut-parleur sur cette sortie.

Exemple de programme :



play 0, 0.joue "Joyeux anniversaire",
play 0, 1.joue "Jingle Bells" et fait clignoter Led1 et Led2,
play 3, 2.joue "Rudolph the red nose reinder" et fait clignoter Led2.

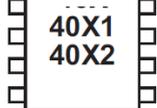
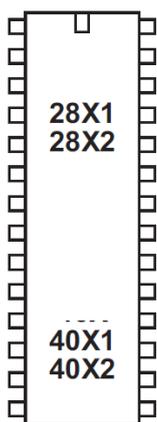
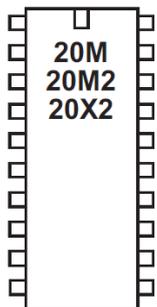
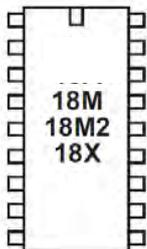
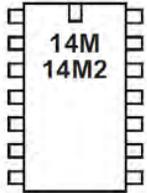
Exemple de câblage pour un Picaxe 08M :



Le buzzer piezzo est sur Tune (Out2), Led1 est sur Out 0 et Led2 est sur Out 4.



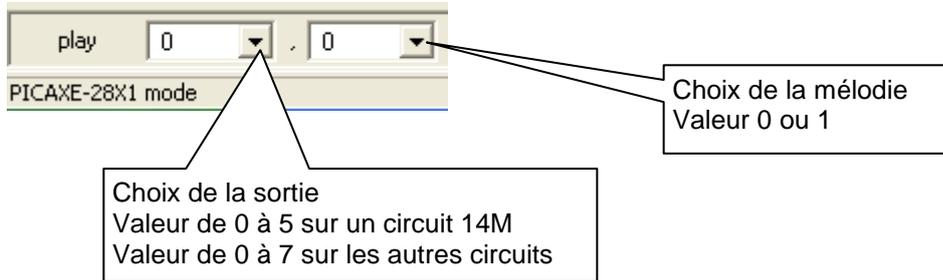
Compatibilité



Fonction :

Permet de jouer un air de musique sur la sortie sélectionnée.
Le choix s'effectue sur 2 mélodies.

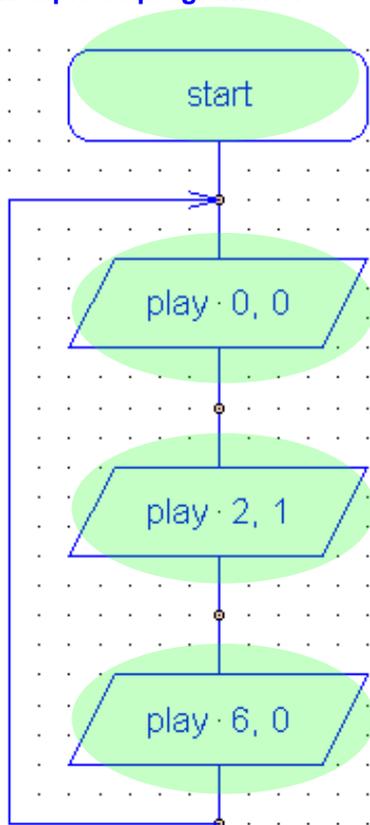
Option sélectionnable :



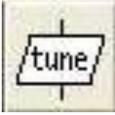
Les mélodies disponibles sont :
0 - Joyeux anniversaire
1 - Jingle Bells

Nota : il est indispensable d'utiliser un buzzer piezzo ou un haut-parleur sur la sortie sélectionnée.

Exemple de programme :



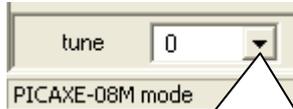
play 0, 0.joue "Joyeux anniversaire" sur la sortie 0,
play 2, 1.joue "Jingle Bells" sur la sortie 1,
play 6, 0.joue "Joyeux anniversaire" sur la sortie 6.



Fonction :

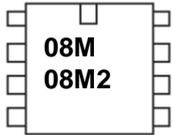
Permet de jouer un air de musique et de faire clignoter des leds.
Le choix s'effectue sur le mode de fonctionnement du clignotement.

Option sélectionnable :



Options de sortie du clignotement
Valeur de 0 à 3

Compatibilité

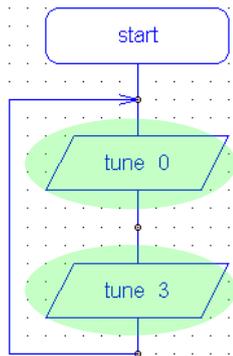


Les options de sortie du clignotement sont :

- 0 - pas de sorties utilisées
- 1 - clignotement de la sortie 0, Led1
- 2 - clignotement de la sortie 4, Led2
- 3 - clignotement alternatif de la sortie 0 et 4, Led1 et Led2

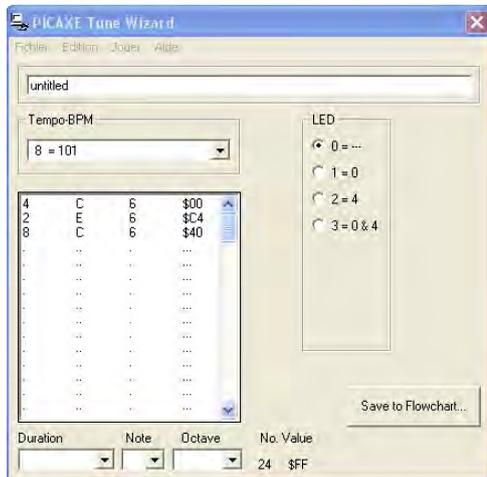
Nota : cette commande est disponible sur la sortie Tune du Picaxe 08M (pin 5 ou Out2), il est indispensable d'utiliser un buzzer piezzo ou un haut-parleur sur cette sortie.

Exemple de programme :

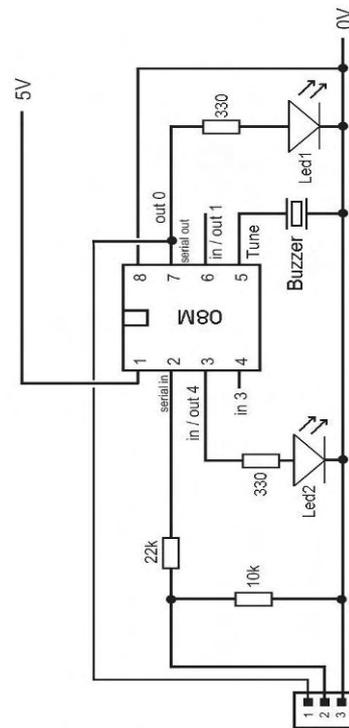


tune 0.joue votre composition,
tune 3.joue votre composition et fait clignoter Led1 et Led2.

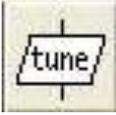
La réalisation de votre composition s'effectue avec le PICAXE Tune Wizard :



Exemple de câblage pour un Picaxe 08M :



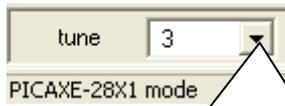
Le buzzer piezzo est sur Tune (Out2),
Led1 est sur Out 0 et Led2 est sur Out 4.



Fonction :

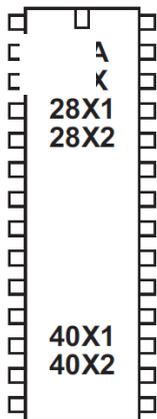
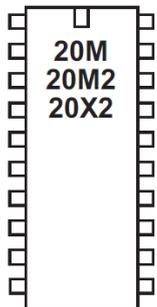
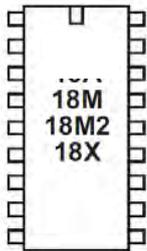
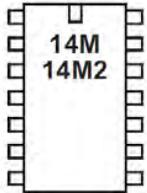
Permet de jouer un air de musique sur la sortie sélectionnée.

Option sélectionnable :



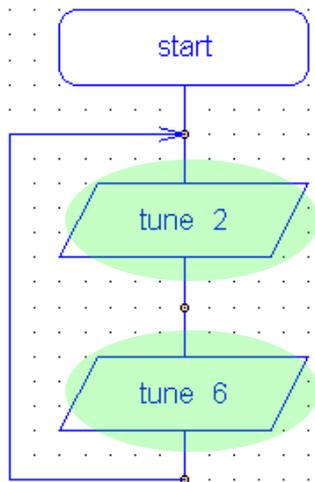
Choix de la sortie
Valeur de 0 à 5 sur un circuit 14M
Valeur de 0 à 7 sur les autres circuits

Compatibilité



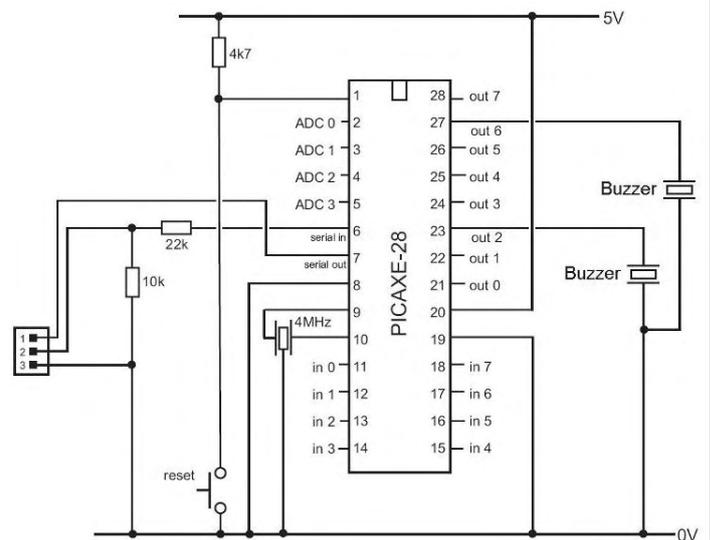
Nota : il est indispensable d'utiliser un buzzer piezzo ou un haut-parleur sur cette sortie.

Exemple de programme :

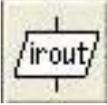


tune 2.joue votre composition sur Out 2,
tune 6.joue votre composition sur Out 6.

Exemple de câblage pour un Picaxe 28X1 :



Un buzzer piezzo est sur la sortie Out 2, un autre buzzer piezzo est sur sortie Out 6.



Fonction :

Permet de transmettre un signal infrarouge en utilisant une diode infrarouge.

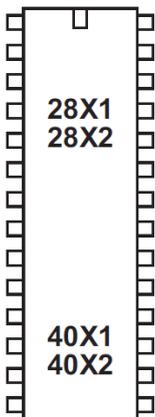
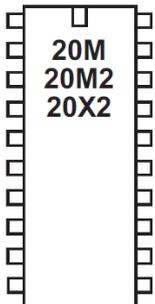
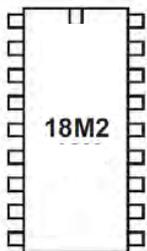
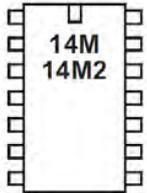
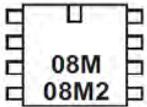
Option sélectionnable :



Compatibilité avec le Picaxe
Mettre à 1

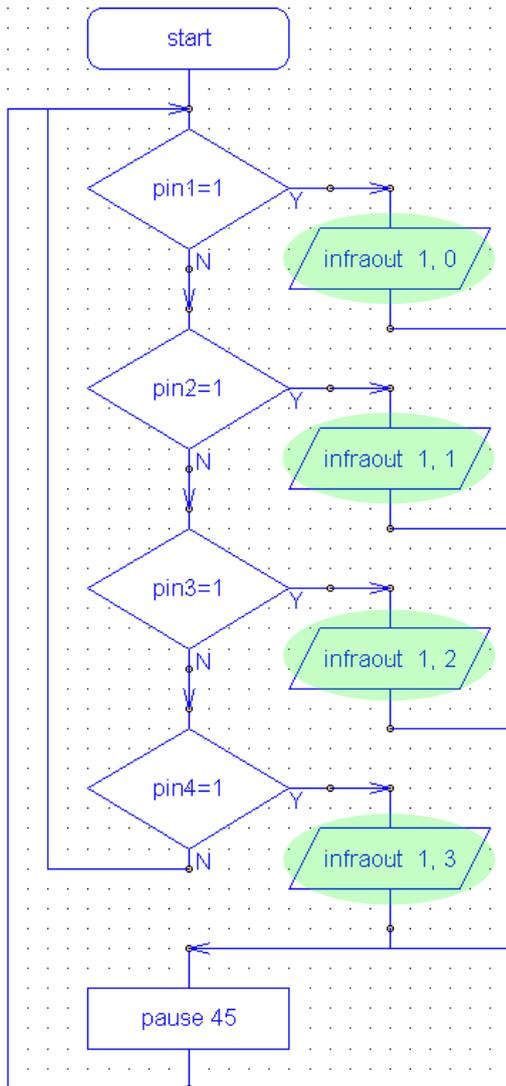
Donnée à envoyer
Valeur de 0 à 127

Compatibilité



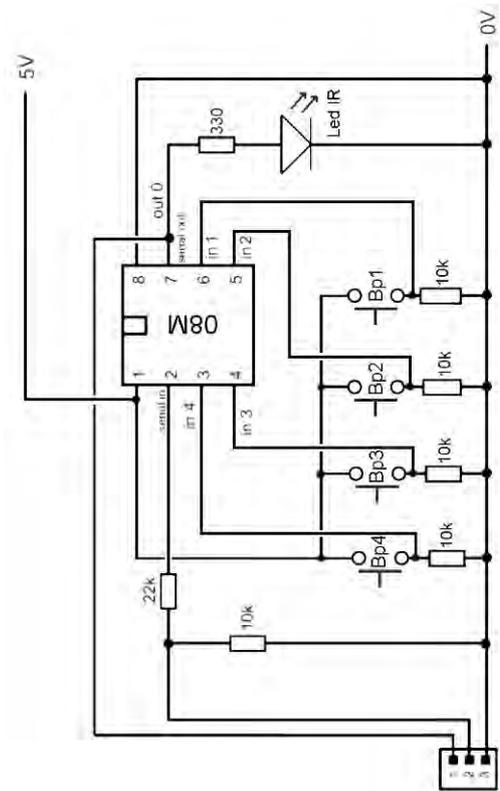
Nota : la transmission de données utilise le protocole SIRC (Sony Infra Red Control).
Cette commande est disponible sur les circuits disposant d'une sortie Infraout.

Exemple de programme :



En fonction de la touche appuyée, la donnée 0, 1, 2 ou 3 est envoyée.

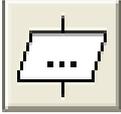
Exemple de câblage d'une LED infrarouge :



La diode IR est sur la sortie Infraout (Out 0), les boutons-poussoirs sont sur les entrées In 0 à In 3.



Guide d'utilisation "Picaxe Programming Editor" Exemple d'utilisation de la commande BASIC

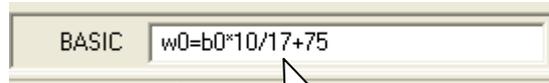
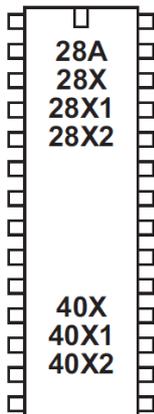
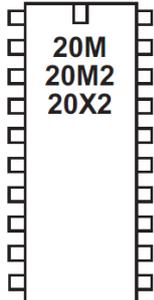
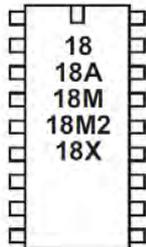
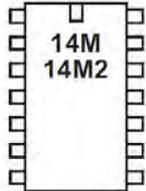
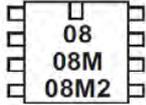


Fonction :

Permet d'écrire une commande basic non disponible ou d'un calcul dans le mode « flowchart » (organigramme).

Option sélectionnable :

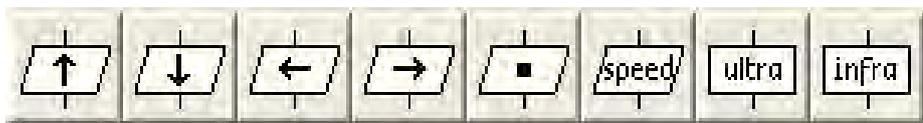
Compatibilité



Exemple d'écriture d'un calcul

3.4 Les commandes de mouvements

Ce groupe de commandes permet d'agir sur les mouvements d'un robot mobile basé sur un microcontrôleur Picaxe.



Permet d'activer la marche avant sur un robot mobile. Indication forward.

Note : lorsque cette commande est exécutée, le mouvement reste actif en permanence jusqu'à ce qu'une autre commande de mouvement ne modifie son état.



Permet d'activer la marche arrière sur un robot mobile. Indication back.

Note : lorsque cette commande est exécutée, le mouvement reste actif en permanence jusqu'à ce qu'une autre commande de mouvement ne modifie son état.



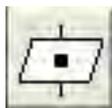
Permet d'activer la rotation à gauche sur un robot mobile. Indication left.

Note : lorsque cette commande est exécutée, le mouvement reste actif en permanence jusqu'à ce qu'une autre commande de mouvement ne modifie son état.



Permet d'activer la rotation à droite sur un robot mobile. Indication right.

Note : lorsque cette commande est exécutée, le mouvement reste actif en permanence jusqu'à ce qu'une commande de mouvement ne modifie son état.



Permet d'arrêter un robot mobile. Indication halt.



Permet de contrôler la vitesse de rotation du moteur gauche et/ou du moteur droit d'un robot mobile.

Note : cette commande nécessite l'utilisation d'un circuit spécifique PWM. C'est le cas du véhicule robot programmable minirobot.



Permet de mesurer la distance entre un capteur à ultrason et un éventuel obstacle.

Note : cette commande nécessite l'utilisation du capteur SRF05.



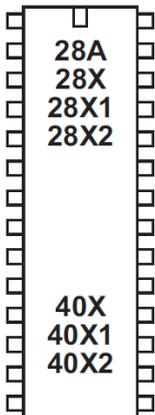
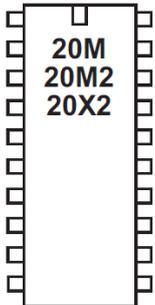
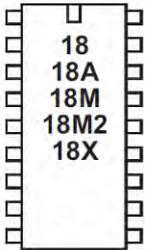
Permet de recevoir une information en provenance d'un capteur infrarouge.

Note : cette commande nécessite l'utilisation d'un capteur infrarouge, il est nécessaire d'utiliser une télécommande pour produire l'information infrarouge.

Nota : Sur les circuits 08 et 08 M, lors de la première utilisation, les commandes de mouvements (forward, back, left et right) n'activent pas les moteurs. En effet, les circuits 8 broches nécessitent de positionner les sorties Out 1, Out2 et Out4 en utilisant une commande comme high ou low.



Compatibilité



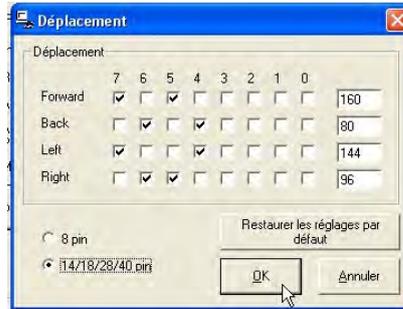
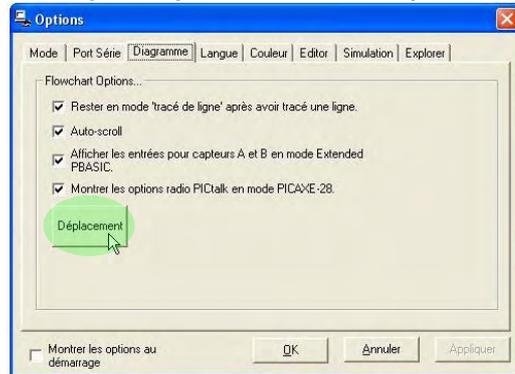
Fonction :

Toutes ces commandes permettent de contrôler les mouvements sur un robot mobile.

Option sélectionnable :

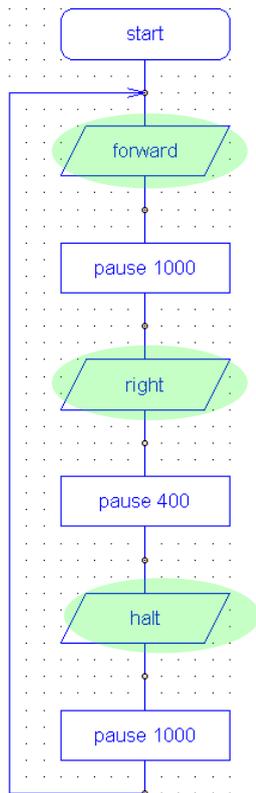
Aucune option.

Nota : les commandes de mouvements fonctionnent avec la configuration défini par Déplacement de l'onglet diagramme du menu Options :



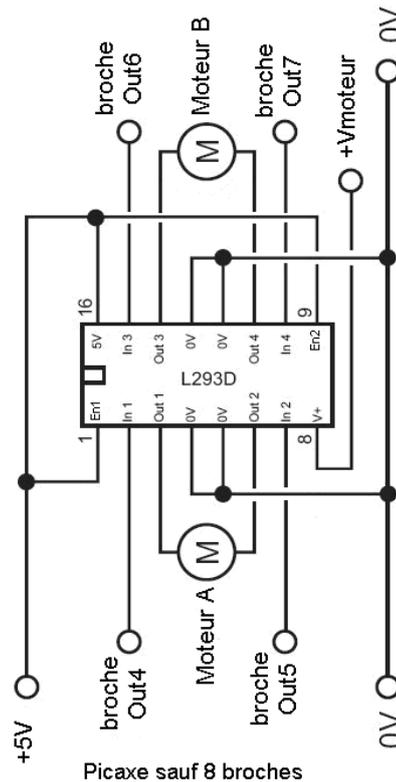
Le « schéma type » d'une commande moteur est réalisé avec un circuit spécifique : le L293D.

Exemple de programme :



Il est indispensable de positionner des pauses pour que les commandes soient effectives.

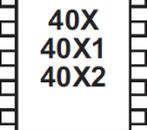
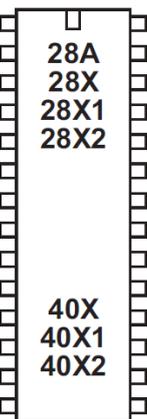
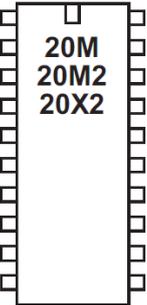
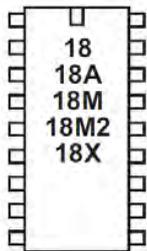
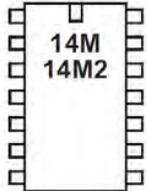
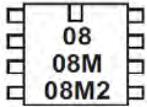
Exemple de câblage de 2 moteurs :



Le moteur A est commandé par les lignes Out4 et Out5. Le moteur B est commandé par les lignes Out6 et Out7. Ces lignes peuvent être différentes mais dans ce cas là, il sera nécessaire de modifier les valeurs de déplacement dans le menu options. Le circuit L293D peut commander des moteurs de petite puissance (maximum 600mA).



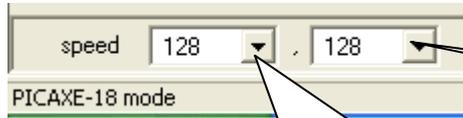
Compatibilité



Fonction :

Permet de contrôler la vitesse de rotation des moteurs.
Cette commande est disponible sur les systèmes intégrant un circuit spécifique PWM. C'est le cas du véhicule robot programmable minirobot.

Option sélectionnable :



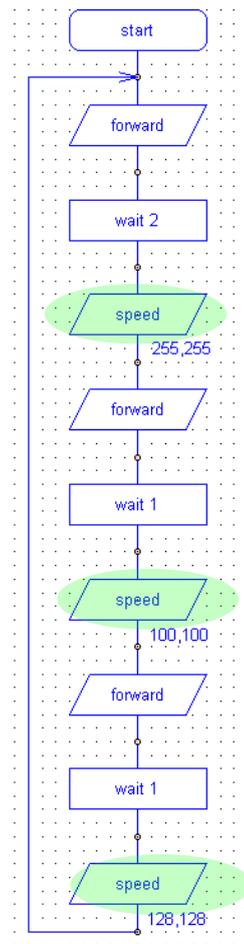
Contrôle vitesse du moteur gauche
De 0 à 255 Variable possible

Contrôle vitesse du moteur droit
De 0 à 255 Variable possible

0 : moteur à l'arrêt 255 : vitesse maximum

Nota : Si l'on introduit plusieurs changements de vitesse dans un programme, il conviendra de faire systématiquement précéder la commande speed par un temps d'attente de 100 ms, temps nécessaire à l'initialisation du composant PWM qui gère la vitesse des moteurs. La mise en rotation d'un moteur s'effectue vers une valeur d'environ 50.

Exemple de programme :



Au démarrage (et par défaut), speed est à (128, 128). La prise en compte de speed doit être validée par une commande (par exemple forward)

Exemple de câblage de 2 moteurs avec PWM :

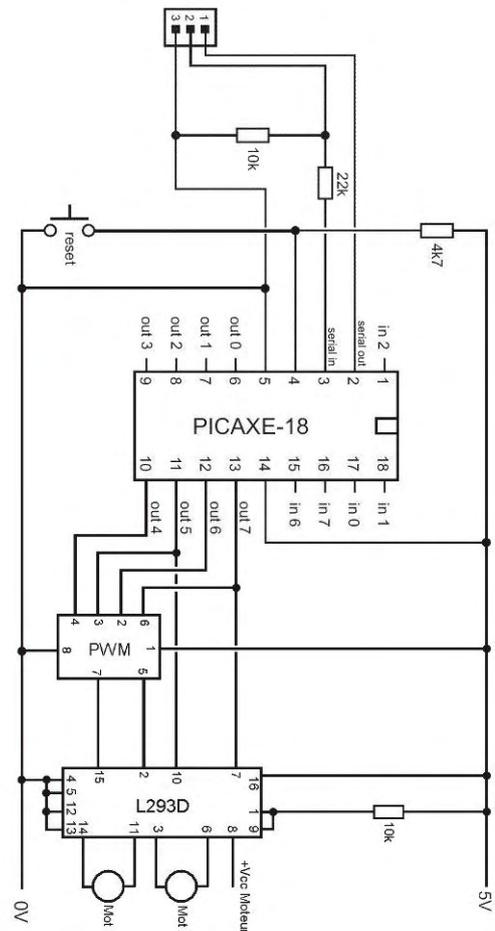


Schéma simplifié du minirobot.



Exemple d'utilisation de la commande ultra



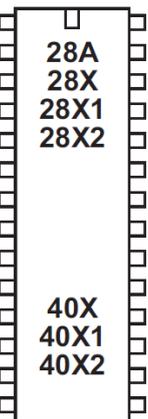
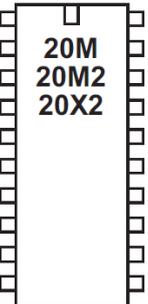
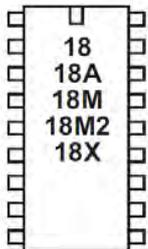
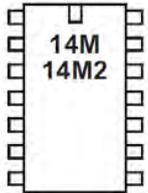
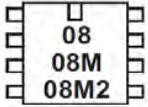
Fonction :

Permet de mesurer la distance entre le capteur SRF05 et un éventuel obstacle.

Option sélectionnable :



Compatibilité

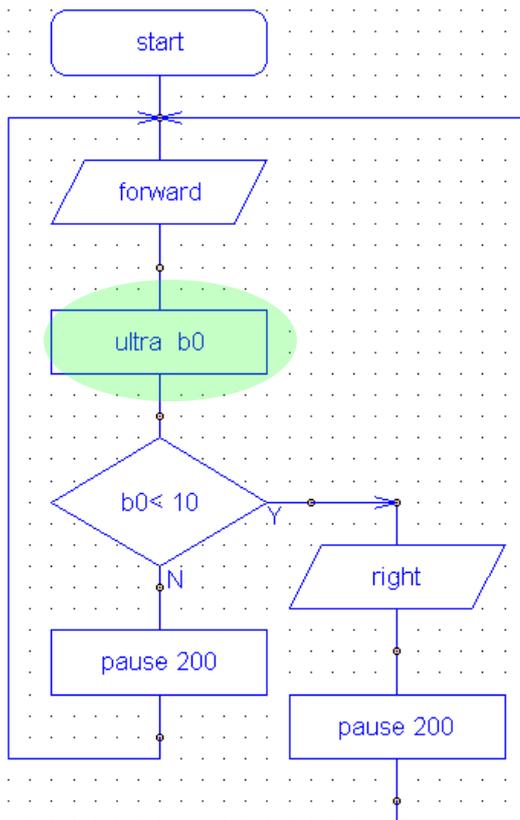


La variable (sur un octet) peut prendre la valeur de 3 à 255.

Nota : la sensibilité de détection permet la détection d'un plot de 3 cm de diamètre et de quelques cm de haut à une distance comprise entre 3 cm et 255 cm (soit 2,55m...).

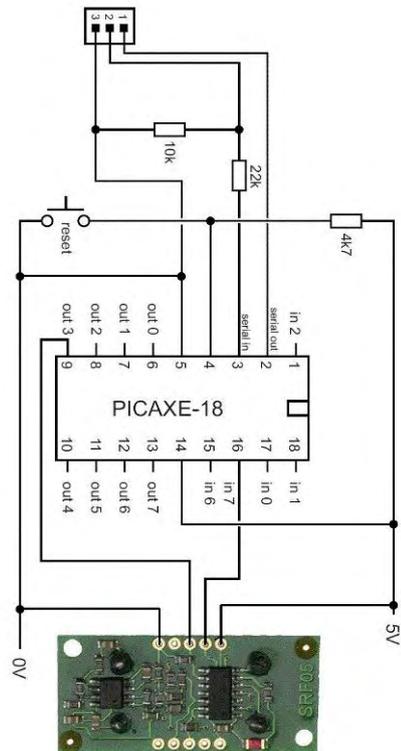
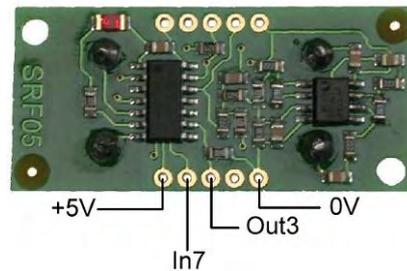
Cette commande nécessite un câblage spécifique avec l'entrée In7 et la sortie Out3

Exemple de programme :

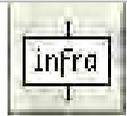


ultra b0 permet de mesurer une distance. Si la distance b0 est inférieure à 10 cm, le robot exécute une rotation à droite puis continue sa route.

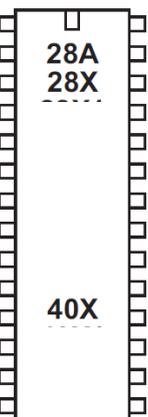
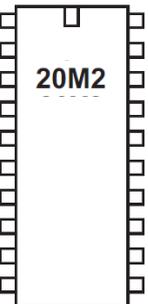
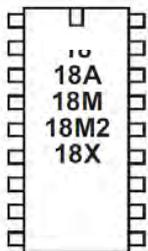
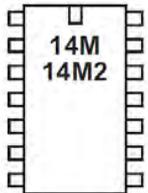
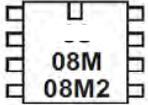
Exemple de câblage du moteur SRF05 :



Module SRF05 vue de dos



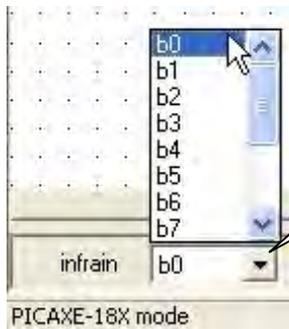
Compatibilité



Fonction :

Permet de recevoir une information infrarouge.

Option sélectionnable :



Le résultat de la réception infrarouge est positionné dans une variable

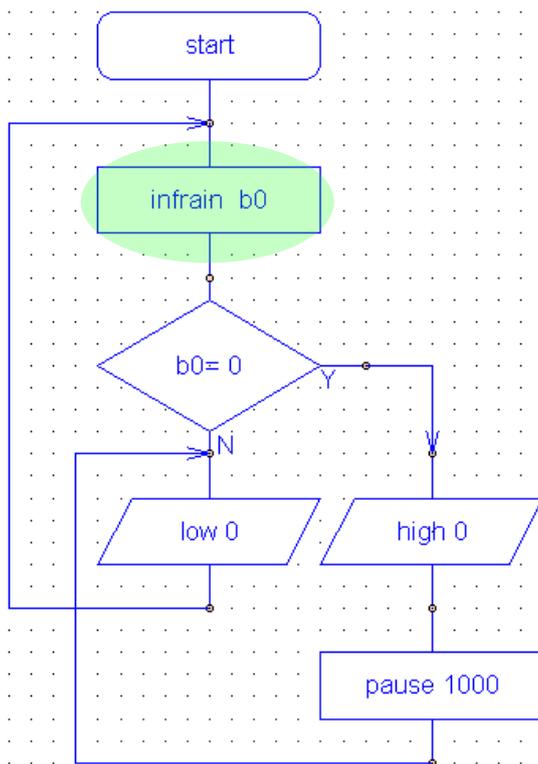
La variable (sur un octet) peut prendre la valeur de 0 à 255.

Nota : cette commande nécessite un capteur infrarouge et un câblage spécifique avec l'entrée In3 pour le circuit 08M ou In0 pour les autres Picaxes.

Avec cette commande, le programme est en attente d'une commande infrarouge.

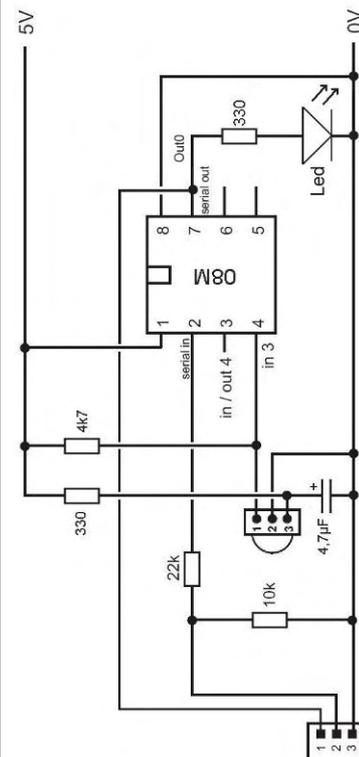
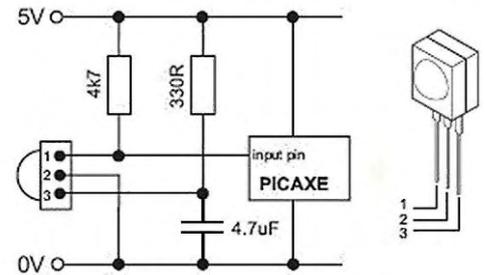
Cette commande doit être remplacée sur les autres Picaxes par la commande BASIC irin.

Exemple de programme :



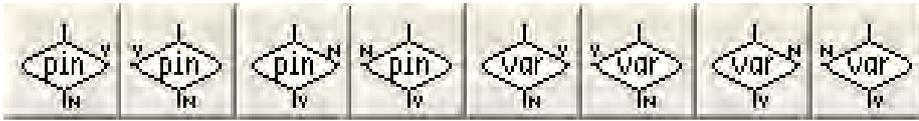
infrain b0 permet de recevoir l'information IR. Si cette information est égale à b0, la diode Del sur Out0 s'allume pendant 1 seconde.

Exemple de câblage d'un capteur IR:



3.5 Les commandes de tests

Ce groupe de commandes permet d'accéder aux instructions de tests. Les tests peuvent s'effectuer sur une entrée (pin) ou sur une variable (var).



Permet d'effectuer un test sur une seule broche du Picaxe, sortie Yes à droite.



Permet d'effectuer un test sur une seule broche du Picaxe, sortie Yes à gauche.



Permet d'effectuer un test sur une seule broche du Picaxe, sortie No à droite.



Permet d'effectuer un test sur une seule broche du Picaxe, sortie No à gauche.



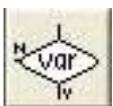
Permet d'effectuer un test sur une variable, sortie Yes à droite.



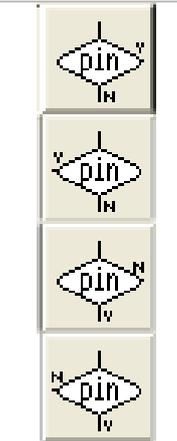
Permet d'effectuer un test sur une variable, sortie Yes à gauche.



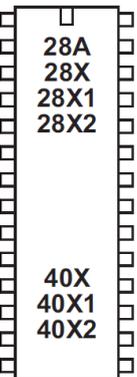
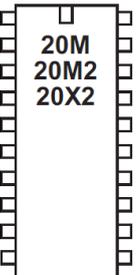
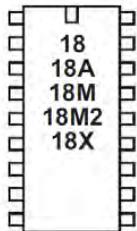
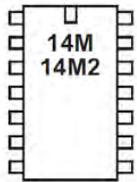
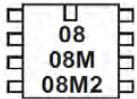
Permet d'effectuer un test sur une variable, sortie No à droite.



Permet d'effectuer un test sur une variable, sortie No à gauche.



Compatibilité

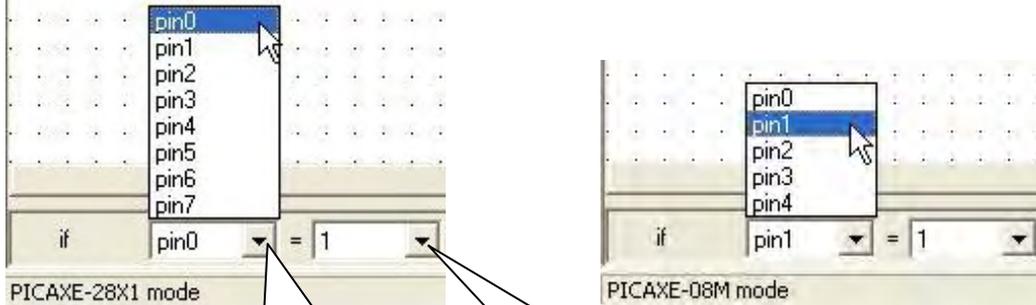


Fonction :

Permet de tester l'état logique d'une seule entrée numérique.

Option sélectionnable :

En fonction du circuit, le nombre d'entrées se modifie



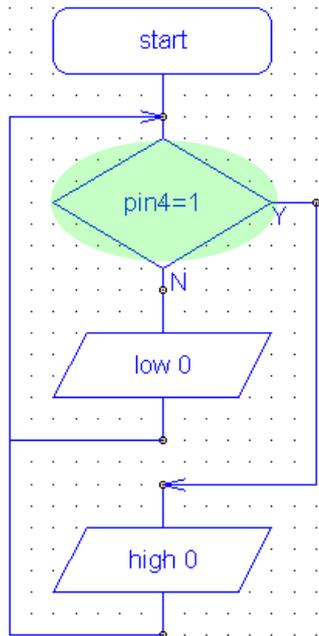
Sélection de l'entrée In

Valeur de test : 1 ou 0

Nota : les 4 tests permettent de choisir le sens de sortie de la condition (Y pour Yes et N pour No). Cela permet de simplifier les organigrammes en évitant de nombreux croisements des connecteurs.

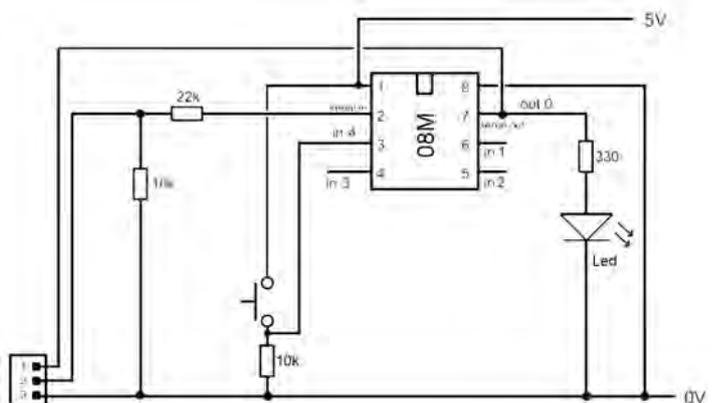
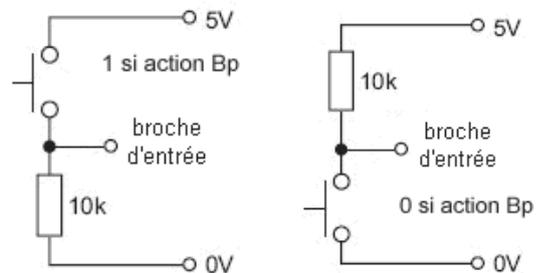
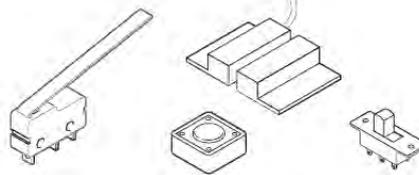
Exemple de programme :

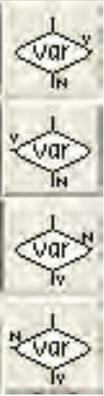
L'action sur le bouton- poussoir provoque l'allumage de la diode LED.



Quand le Bp n'est pas appuyé, la valeur sur In4 est à 0, le programme se dirige vers low 0. Lors de l'appui sur Bp In4 passe à 0 et provoque l'allumage de la Led.

Exemple de câblage d'un capteur TOR : (capteur Tout Ou Rien)

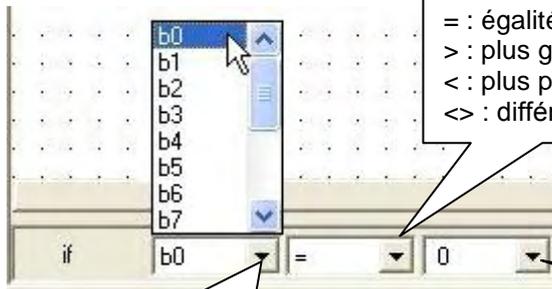




Fonction :

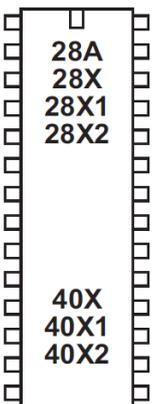
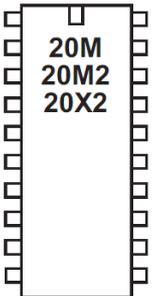
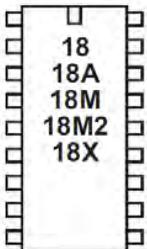
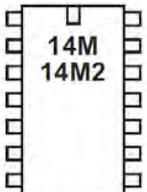
Permet d'effectuer un test sur une variable ou sur l'ensemble des broches d'entrée (pins).

Option sélectionnable :



Test à réaliser :
= : égalité
> : plus grand que
< : plus petit que
<> : différent de (≠)

Compatibilité

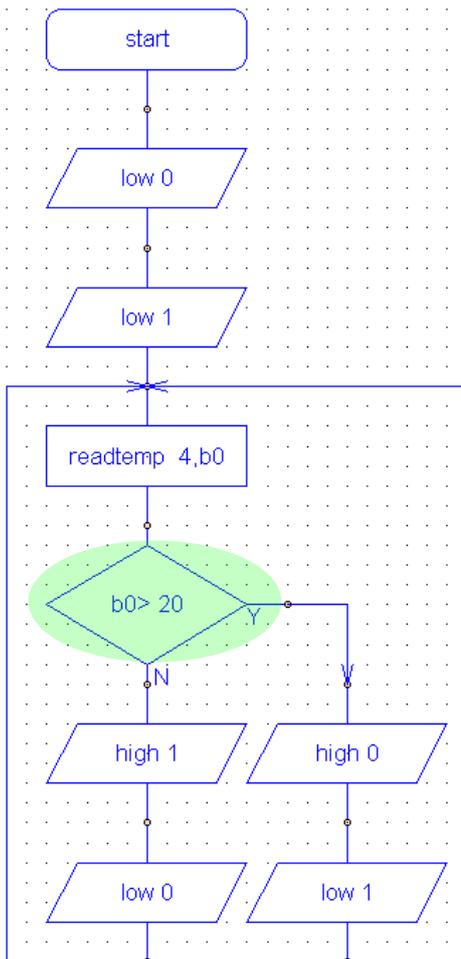


Choix de la variable à tester
pins correspond à l'ensemble des entrées.

Valeur sur laquelle s'effectue le test
pins correspond à la valeur sur toutes
les broches d'entrée.

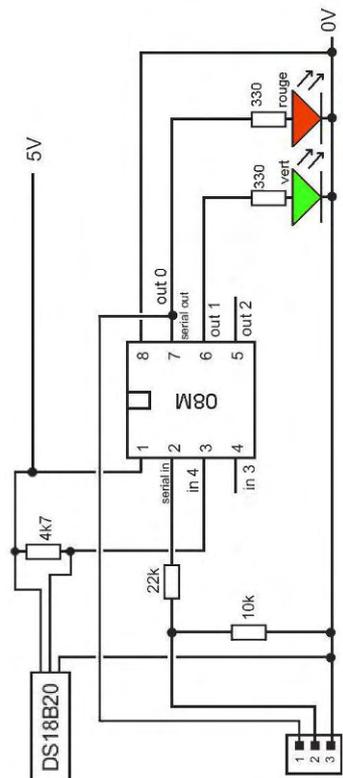
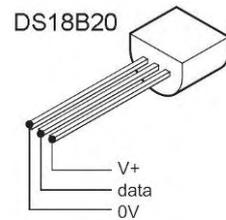
Nota : les 4 tests permettent de choisir le sens de sortie de la condition (Y pour Yes et N pour No).

Exemple de programme :



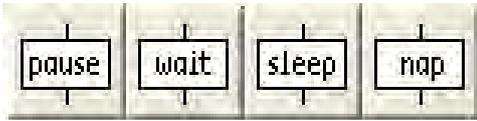
La commande readtemp permet de positionner la température dans la variable b0. Si b0 (température) est supérieure à 20°, la Led rouge s'allume, sinon, la led verte est allumée.

Exemple de câblage d'un capteur DS18B20 :



3.6 Les commandes de temporisation

Ce groupe de commandes permet d'accéder aux instructions d'attente. En effet, le fonctionnement d'un Picaxe est très rapide (à l'échelle humaine) et il est souvent indispensable de ralentir le programme pour visualiser les évènements.



Permet de positionner un temps d'attente en milliseconde.



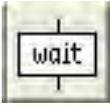
Permet de positionner un temps d'attente en seconde.



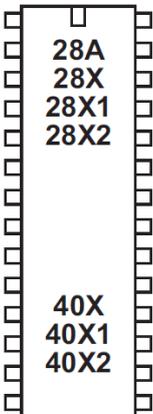
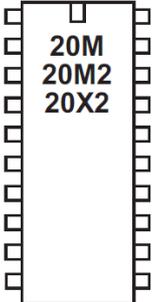
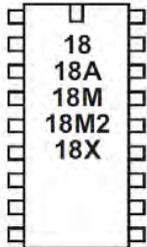
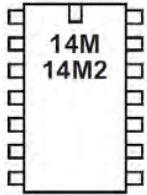
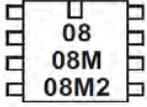
Permet de positionner le système Picaxe en mode sommeil. Ce mode particulier permet d'économiser l'énergie consommée par le Picaxe.



Permet de positionner le système Picaxe en mode sommeil pour une période courte.



Compatibilité



Fonction :

pause permet de positionner un temps d'attente en milliseconde.
wait permet de positionner un temps d'attente en seconde.

Option sélectionnable :



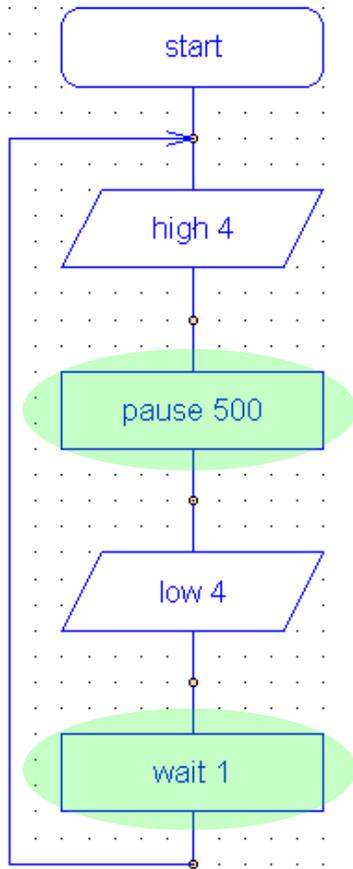
Attente en milliseconde
De 0 à 65535 ms
Variable possible



Attente en seconde
De 0 à 65 s

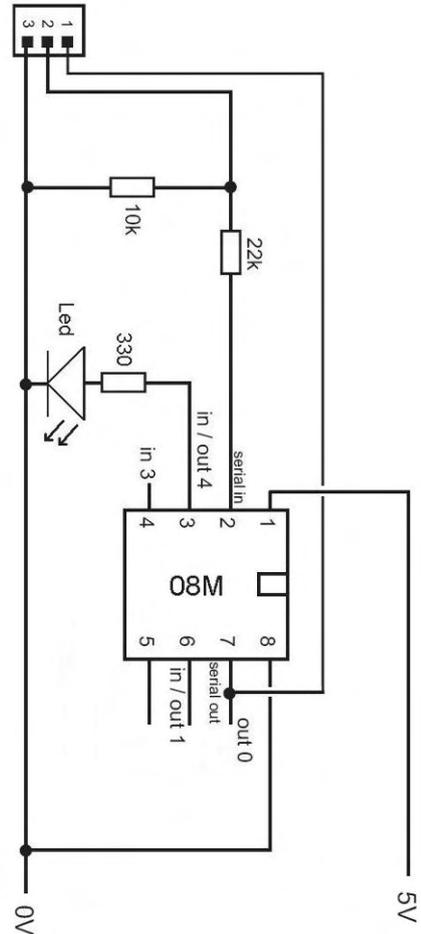
*Nota : pause 1000 correspond à une attente d'1 seconde.
pause 0 ou wait 0 sont possibles mais n'ont pas d'intérêt...*

Exemple de programme :



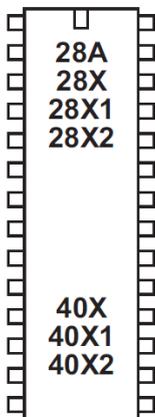
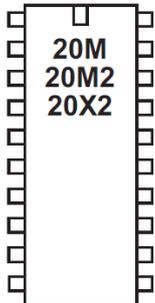
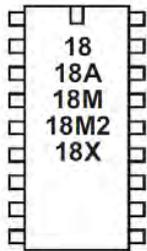
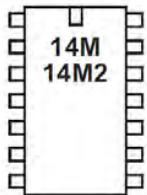
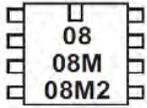
Clignotement de la led sur la sortie 4. La Led est allumée pendant 0,5 s et s'éteint pendant 1 s.

Exemple de câblage d'une led :





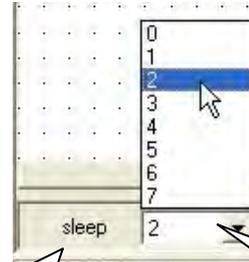
Compatibilité



Fonction :

Permet de positionner le Picaxe en mode sommeil pour une période de temps (multiple d'environ 2,3 secondes de la valeur indiquée). Cette commande ressemble à pause ou wait, cependant, elle met le Picaxe en mode faible puissance. Ce mode est particulièrement utile pour les applications alimentées par pile. nap fonctionne comme sleep mais elle est utilisée pour les courtes périodes.

Option sélectionnable :



Mise en sommeil
De 1 à 65535
Variable possible

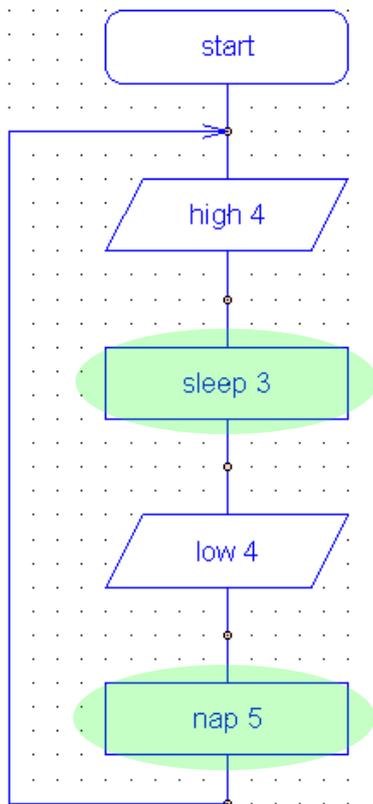
nap : problème d'écriture
au niveau du logiciel voir
le nota !

Mise en sommeil
De 0 à 7
Temps (voir tableau)

Nota : en mode sleep ou nap, les compteurs sont mis hors tension, les commandes pwmout ou servo cesseront de fonctionner. Les temps de sommeil sont susceptibles de varier énormément (tolérance - 50% à +100%).

Nota : La commande **nap** pose quelques problèmes d'affichage sur le logiciel en effet l'inscription à coté de la zone de sélection de temps n'indique pas « nap » mais indique la fonction précédente (sleep dans le cas de la copie d'écran ci-dessus !)

Exemple de programme :



Clignotement de la led sur la sortie 4. La Led est allumée pendant 6,9 s et s'éteint pendant 576 ms.

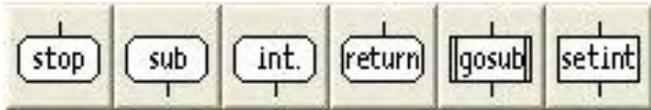
Tableau de Temps :

| Choix valeur sleep | |
|---|--|
| Valeur x 2,3 s | |
| Par exemple sleep 5 dure environ 5x2,3 = 11,5 s | |

| Choix valeur nap | Temps |
|------------------|-------|
| 0 | 18ms |
| 1 | 32ms |
| 2 | 72ms |
| 3 | 144ms |
| 4 | 288ms |
| 5 | 576ms |
| 6 | 1,1s |
| 7 | 2,3s |

3.7 Les commandes de sous-programmes

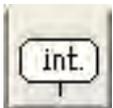
Ce groupe de commandes permet d'accéder aux instructions nécessaires à la mise en œuvre de sous-programme et des interruptions.



Permet d'indiquer la fin d'un programme.



Permet d'indiquer le point de départ d'un sous-programme.



Permet d'indiquer le point de départ d'un sous-programme géré en mode interruption.



Permet d'indiquer la fin d'un sous-programme. Cette commande est indispensable pour le retour au programme principal.



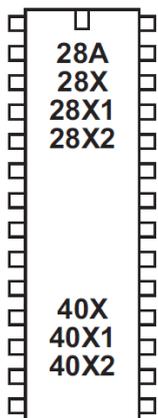
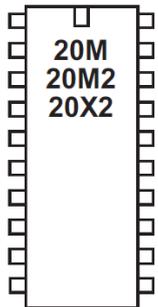
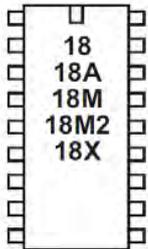
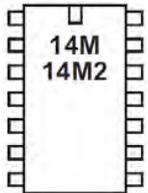
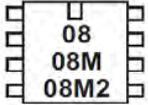
Permet d'effectuer un appel à un sous-programme.



Permet de gérer la mise en œuvre des interruptions.



Compatibilité



Fonction :

Permet d'indiquer la fin d'un programme. Cette commande n'est pas indispensable dans un programme.

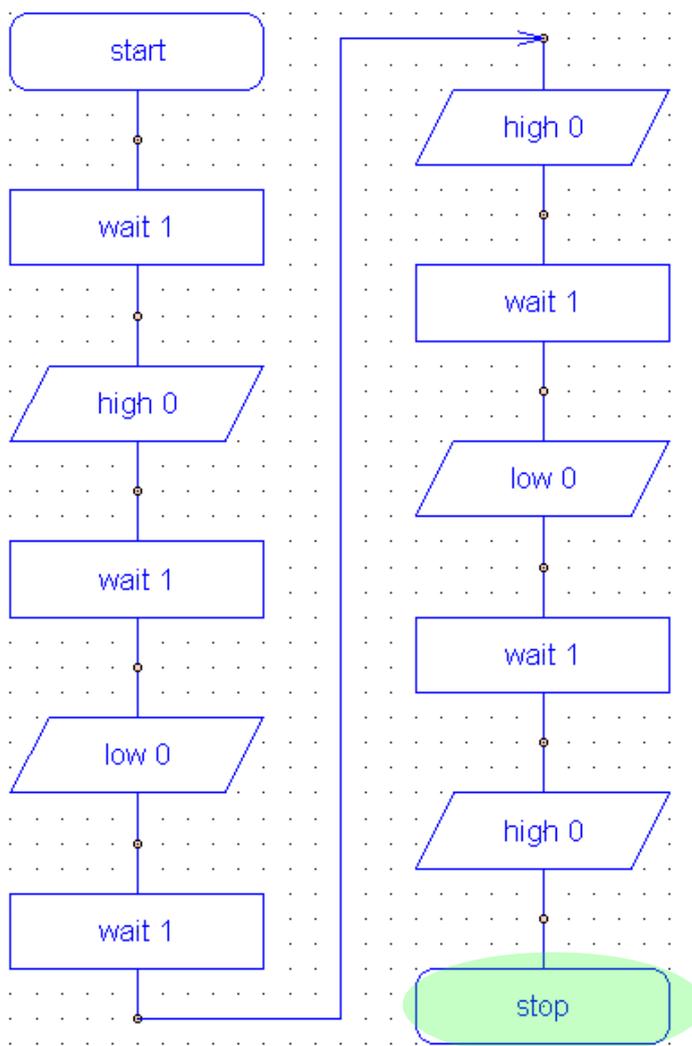
Option sélectionnable :

Aucune

Nota : la commande stop arrête le programme mais pas le Picaxe, il s'agit d'une boucle permanente. Les compteurs restent en fonctionnement, les commandes pwmout ou servo continuent de fonctionner.

Pour sortir de cette commande, un reset est nécessaire.

Exemple de programme :



Utilisation d'une led sur la sortie 0. Le programme permet d'effectuer 3 allumages de la led avant de s'arrêter avec la led allumée.



Les sous-programmes :

Un sous-programme est un "mini-programme" qui est exécuté lors d'un appel gosub. Le retour dans le programme principal s'effectue dès la présence de return.
Quand le programme principal fait un appel gosub, il exécute le sous-programme correspondant puis retourne dans le programme dès return.

Option sélectionnable :

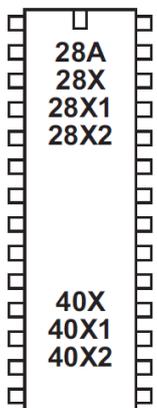
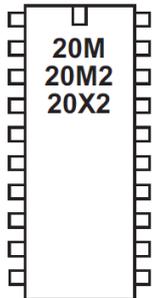
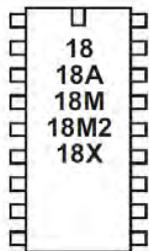
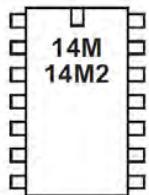
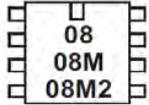


Ecriture du nom de sous-programme



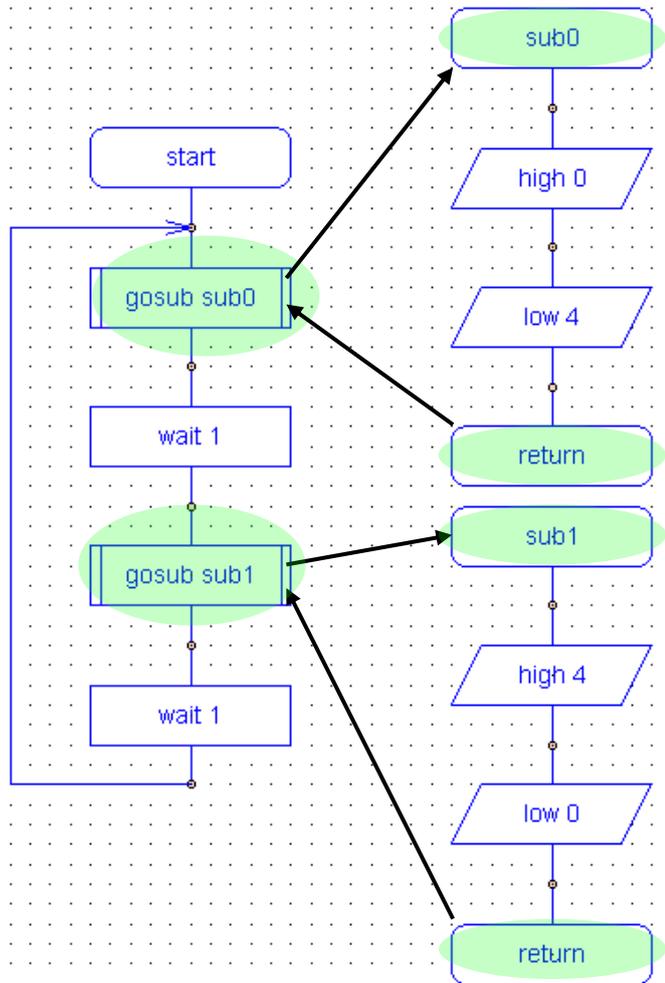
Sélection du sous-programme

Compatibilité



Nota : Les sous-programmes simplifient considérablement le programme principal en évitant la réécriture de blocs appelés régulièrement dans un programme.
La commande gosub doit obligatoirement être accompagné de la commande return.

Exemple de programme :

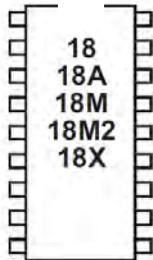
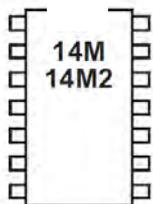
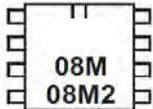


Le nombre maximum de gosub dans un programme est de 15 pour les microcontrôleurs Picaxe série 'M' et de 255 pour les microcontrôleurs Picaxe de la série 'X'.

La commande gosub peut-être imbriquée dans une autre commande gosub. L'imbrication est de 4 pour les microcontrôleurs Picaxe série 'M' et 'X' et de 8 pour les microcontrôleurs Picaxe de la série 'X1' et 'X2'.



Compatibilité



Les interruptions :

Une interruption est un sous-programme autorisant une prise en compte immédiate d'un évènement afin d'exécuter un traitement spécifique appelé routine d'interruption. Ainsi, une interruption permet de prendre en compte immédiatement un évènement survenant sur une entrée numérique. La commande setint permet de mettre en œuvre le mode interruption.

Option sélectionnable :

Permet de choisir la ou les entrées devant être présent en compte. (mask)
Variable possible

Permet de choisir la condition d'interruption (input)
L'interruption permet ainsi de réagir sur une condition 0 ou 1.
Variable possible

| | | | | | | | | |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|
| 8(2)= | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ |
| | In7 | In6 | In5 | In4 | In3 | In2 | In1 | In0 |

La valeur doit être convertie en valeur binaire : exemple $8_{(10)} = 00001000_{(2)}$
Donc mask = 8 correspond à 00001000 permet de prendre en compte l'entrée In3
input = 8 correspond à 00001000 permet de faire réagir l'entrée In3 pour un niveau 1

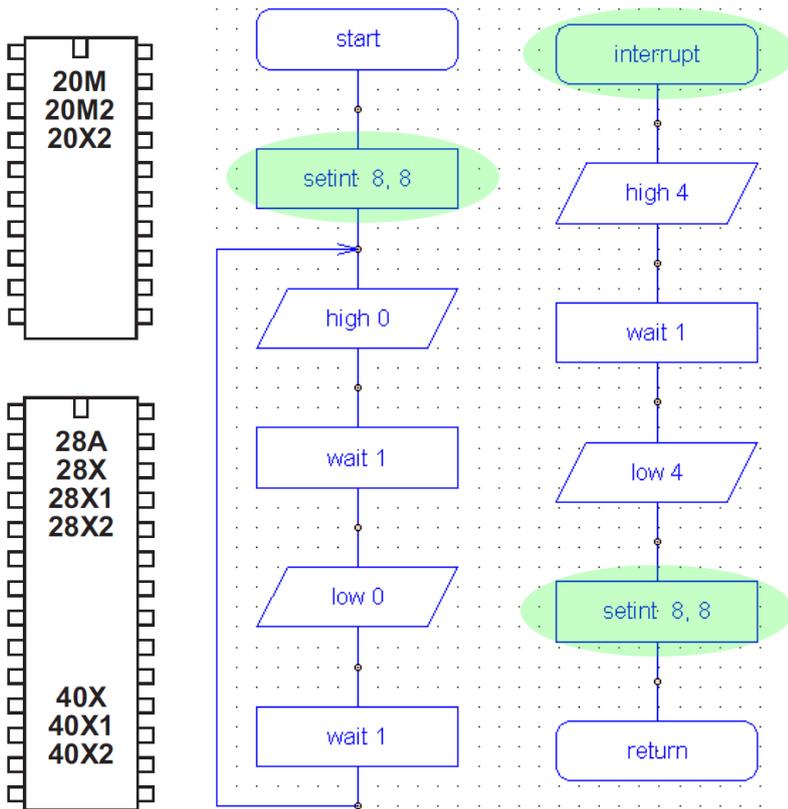
Tableau de paramétrage de la commande d'interruption.
Dans l'exemple cité, l'interruption doit ce faire avec l'entrée « in3 » à l'état « 1 » (entrée activée).

| Valeur décimale du masque d'entrée | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | Commande : Setint |
|--------------------------------------|-----|----|----|----|----------|---|---|---|-------------------|
| N° d'entrée | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| Valeur binaire du masque | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 8 |
| Valeur binaire de la condition « 1 » | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 8 |
| Valeur binaire de la condition « 0 » | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

Le paramétrage en valeur décimale permet de prendre en compte l'état de plusieurs entrées au niveau « 1 » ou « 0 ». Pour de plus amples informations veuillez vous référer au manuel Picaxe en Anglais accessible depuis le menu d'aide du logiciel Picaxe Programming Editor.

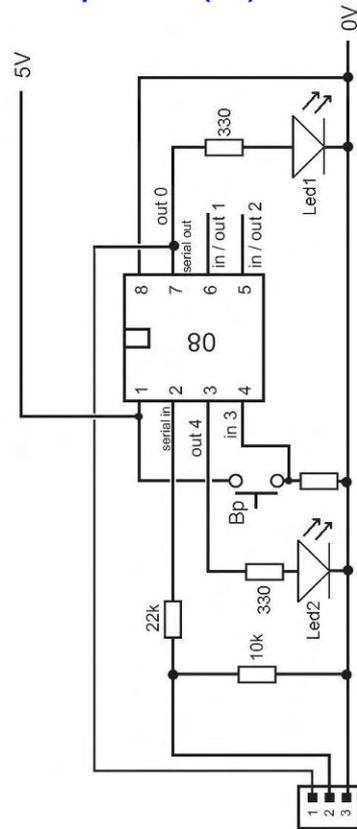
Note : une interruption est gérée comme un sous-programme et nécessite toujours un return. Après une interruption, la prise en compte d'une nouvelle interruption nécessite un réarmement. Une interruption maximum par programme.

Exemple de programme :



Clignotement de la led 1 sur la sortie 0. Dès l'appui du Bp, la led 2 s'allume pendant 1 s et s'éteint. Le réarmement s'effectue à la fin de l'interruption.

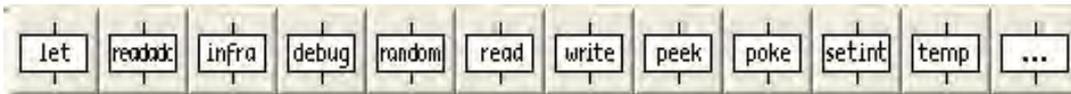
Exemple de câblage de 2 leds et d'un Bouton pousseur (BP) :



Le BP contrôle une interruption sur l'entrée In3.

3.8 Les autres commandes

Ce groupe de commandes permet de mettre en œuvre des applications courantes du Picaxe.



Permet d'effectuer des manipulations sur les variables.
Note : cette commande permet d'initialiser une variable ou de réaliser des calculs.



Permet de lire une valeur analogique sur la broche d'entrée sélectionnée du Picaxe.
Note : cette commande fonctionne sur 8 bits, la valeur pourra ainsi prendre une valeur de 0 à 255.



Permet de recevoir une information en provenance d'un capteur infrarouge.
Note : cette commande se retrouve dans les commandes de mouvement.



Permet d'envoyer au PC la valeur d'une variable afin de vérifier et de contrôler le bon fonctionnement d'un programme.
Note : cette commande nécessite de relier le Picaxe via un câble de liaison et ralentit le programme.



Permet de générer une valeur entre 0 et 255.
Note : la génération est effectuée par calcul, c'est pourquoi, cette valeur est pseudo aléatoire.



Permet de lire une donnée stockée dans l'EEPROM du Picaxe.
Note : la mémoire EEPROM n'est pas perdue lorsque l'alimentation est arrêtée.



Permet d'écrire une donnée dans l'EEPROM du Picaxe.
Note : la mémoire EEPROM n'est pas perdue lorsque l'alimentation est arrêtée.



Permet de lire des données (variables de 8 bits) de la mémoire du Picaxe.
Note : au-delà des variables normales b0 à b13, les Picaxes disposent de plus de mémoire qu'il est possible d'utiliser avec la commande peek. Il est possible de lire la valeur des registres du Picaxe. Pour expert uniquement...



Permet d'écrire des données (variables de 8 bits) dans la mémoire du Picaxe.
Note : cette commande permet d'accéder à des fonctions spéciales du Picaxe. Attention, cela peut aussi provoquer des blocages irréversibles du Picaxe. Pour expert uniquement...



Permet de gérer la mise en œuvre des interruptions.
Note : cette commande se retrouve dans les commandes de sous-programme. Voir page 52 et 53



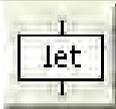
Permet de lire la température en degré Celsius à partir d'un capteur numérique DS18B20.
Note : la température est codée sur 7 bits (b0 à b6), le bit 7 de la variable est utilisé pour signaler une température négative. Si b7 est à 1, la température est négative.



Permet d'écrire une commande en BASIC.
Note : cela permet d'avoir accès à des fonctions non accessibles avec le mode Flowchart. Cette commande se retrouve dans les commandes de sorties. Voir page



Exemple d'utilisation de la commande let



Fonction :

Permet d'effectuer des manipulations sur les variables.

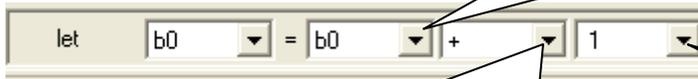
Option sélectionnable :

Variable dans laquelle s'effectue le stockage de l'initialisation ou le résultat du calcul

Valeur d'initialisation de la variable
Variable possible



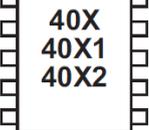
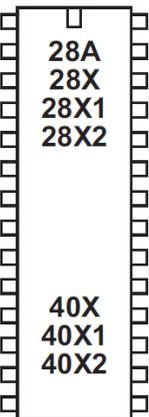
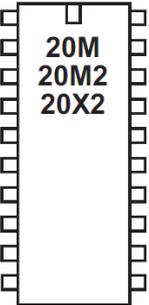
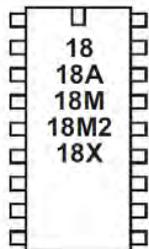
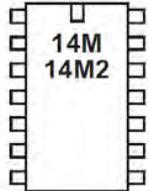
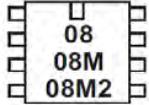
Valeur ou variable de calcul



Choix de l'opération mathématique (voir calculs)

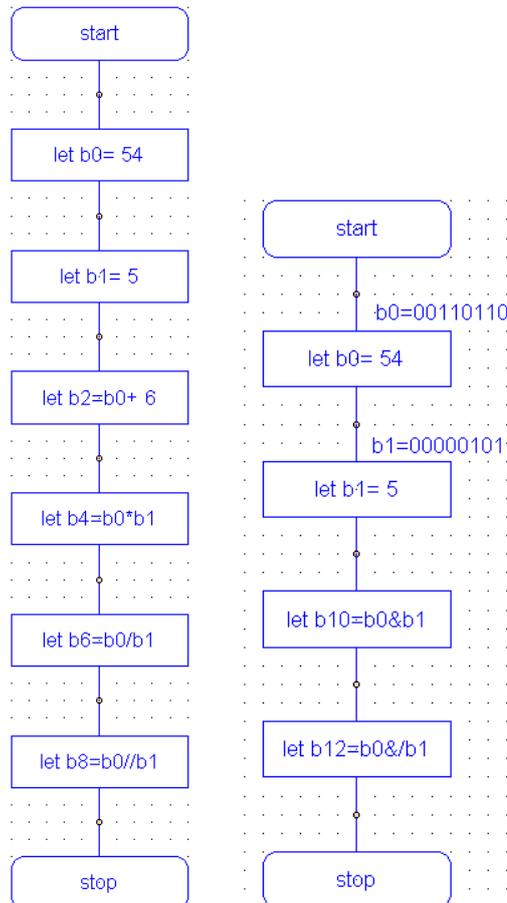
Valeur ou variable de calcul

Compatibilité



Nota : en mode « flowchart » (organigramme), les opérations s'effectuent sur des variables de 8 bits. Les opérations s'effectuent sur des nombres entiers positifs. La variable de stockage ne conserve que la partie entière d'un calcul (division). Si le calcul est supérieur à 255, la variable de stockage est tronquée.

Exemple de programme :



Après calcul : b2=60
b4=14 en effet $54 \times 5 = 270_{(10)} = 1'00001110'$
b6=11
b8=4 en effet $54/5 = 5 \times 10 + 4$

Calculs mathématiques ou logiques possibles :

- + : addition
- : soustraction
- * : multiplication
- / : division
- // : reste d'une division
- & : ET logique
- | : OU logique
- ^ : OU exclusif logique
- &/ : NON ET logique
- |/ : NON OU logique
- ^/ : OU exclusif logique

Concernant les calculs logiques :
 $b0 = 54_{(10)} = 00110110_{(2)}$
 $b1 = 5_{(10)} = 00000101_{(2)}$
donc
 $b0 \& b1 = 00000100_{(2)} = 4_{(10)}$
 $b0 \& / b1 = 11111011_{(2)} = 251_{(10)}$

Après calcul : b10=4
b12=251

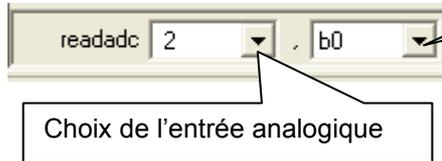


Fonction :

Permet de lire une valeur analogique sur la broche d'entrée sélectionnée du Picaxe.

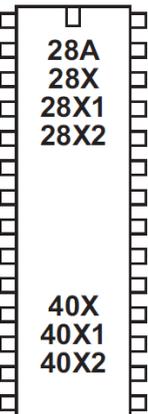
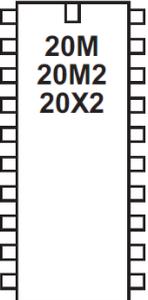
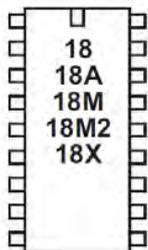
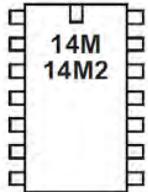
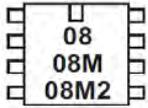
Option sélectionnable :

Variable dans laquelle s'effectue le stockage de la valeur analogique



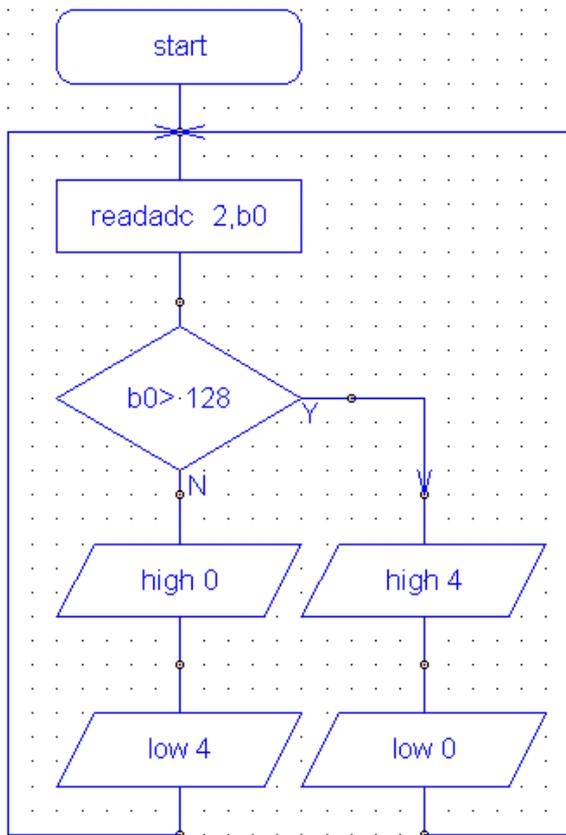
Choix de l'entrée analogique

Compatibilité



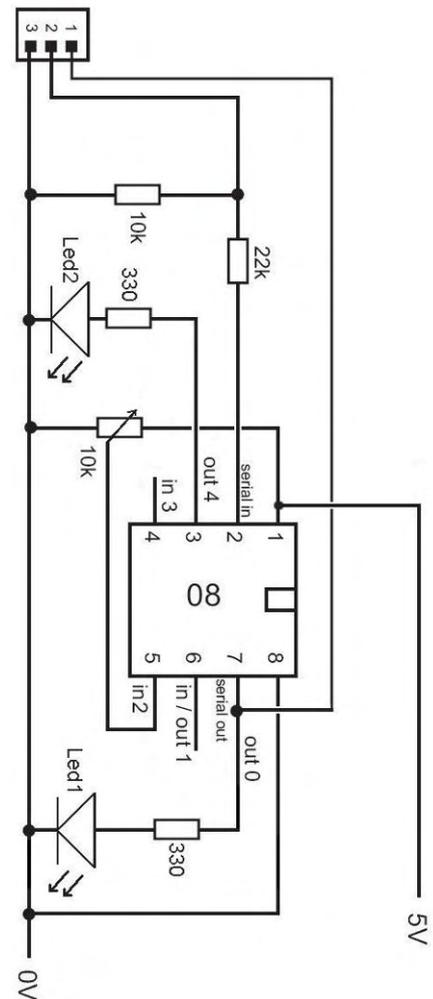
Nota : cette commande fonctionne sur 8 bits, la variable pourra ainsi prendre une valeur de 0 à 255.

Exemple de programme :



Le convertisseur analogique/numérique (CAN) du Picaxe délivre une valeur de 0 à 255 en fonction de l'entrée évoluant de 0 à 5V.
La position médiane du potentiomètre délivre une valeur de tension d'environ la moitié de l'alimentation soit $5/2=2,5V$. La valeur obtenue par le CAN sera donc de $255/2 = 128$.
En fonction de cette valeur l'allumage s'effectue sur la led1 ou sur la led2.

Exemple de câblage de 2 leds et d'un potentiomètre :



Le potentiomètre permet d'envoyer une information analogique représentative de sa position sur l'entrée analogique In2.



Guide d'utilisation "Picaxe Programming Editor"

Exemple d'utilisation de la commande debug



Fonction :

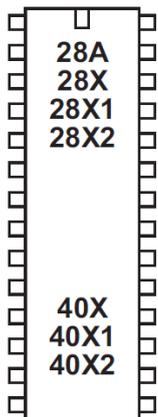
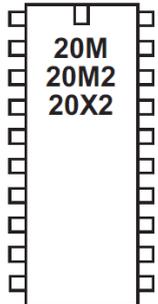
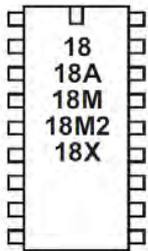
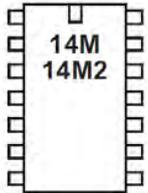
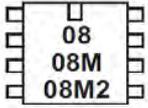
Permet d'envoyer au PC la valeur d'une variable afin de vérifier et de contrôler le bon fonctionnement d'un programme.

Option sélectionnable :



Variable affichée dans la fenêtre Debug

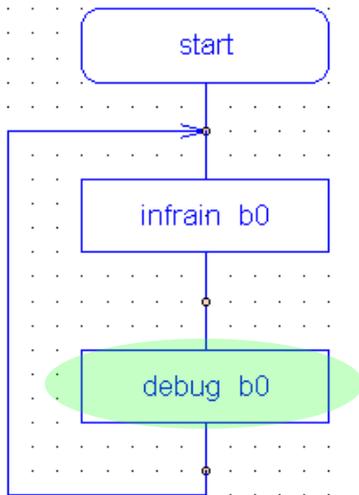
Compatibilité



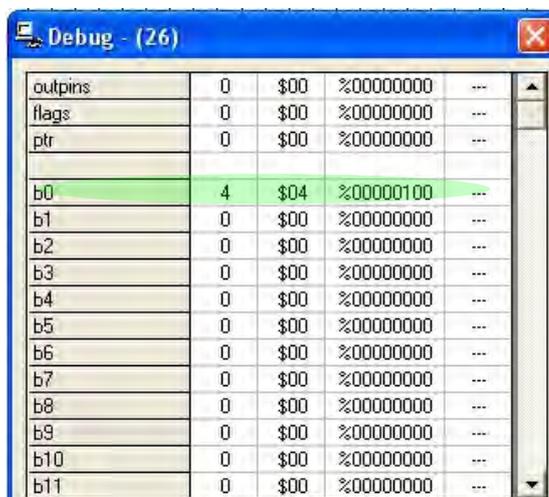
40X
40X1
40X2

Nota : cette commande nécessite de relier le Picaxe via un câble de liaison et ralentir considérablement le programme. Cette commande affiche aussi l'ensemble des variables du Picaxe. L'affichage s'effectue sous forme décimale, hexadécimale et binaire.

Exemple de programme :

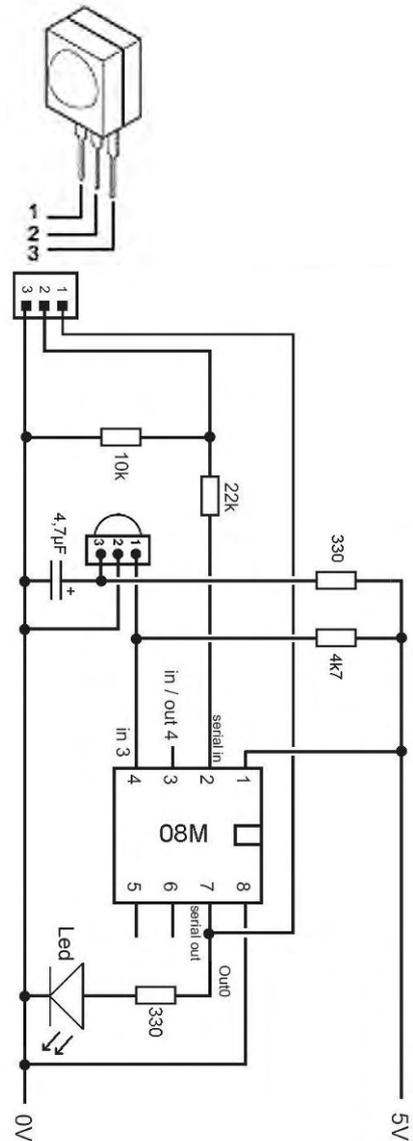


La lecture de l'information IR s'effectue très simplement dans la fenêtre Debug :



La fenêtre Debug affiche l'ensemble des variables du Picaxe.

Exemple de câblage d'un capteur IR :



Le capteur IR envoie l'information IR sur In3.



Guide d'utilisation "Picaxe Programming Editor"

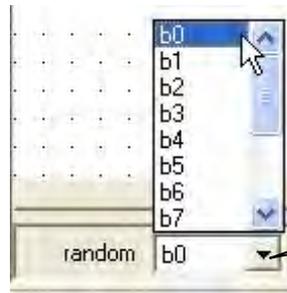
Exemple d'utilisation de la commande random



Fonction :

Permet de générer une valeur quelconque entre 0 et 255.

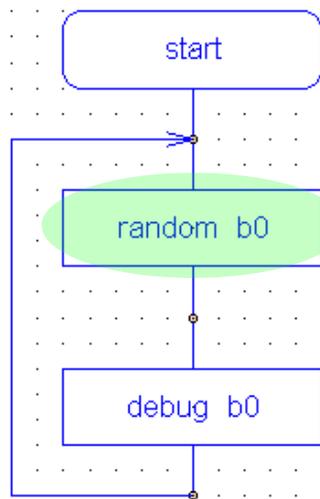
Option sélectionnable :



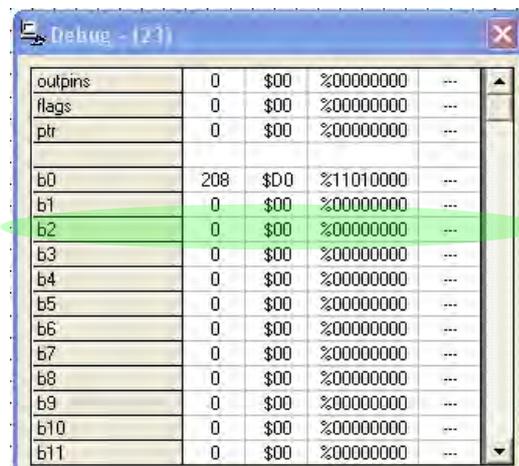
Variable dans laquelle s'effectue le stockage de la valeur pseudo aléatoire générée

Nota : la création de cette variable s'effectue par calcul, on parle donc de valeur pseudo aléatoire. En mode BASIC, la valeur générée est entre 0 et 65535.

Exemple de programme :

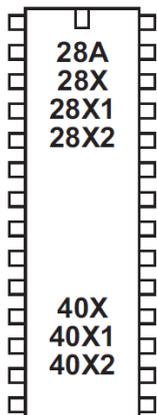
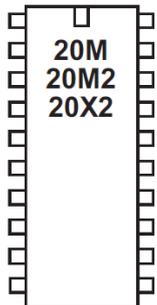
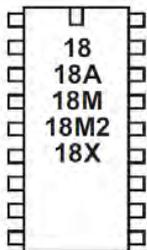
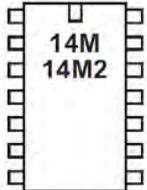
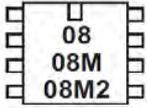


Visualisation de la variable pseudo aléatoire dans une fenêtre Debug :



Nota : cette fonction peut être utile pour les jeux de hasard (dés, roulettes, etc.)

Compatibilité



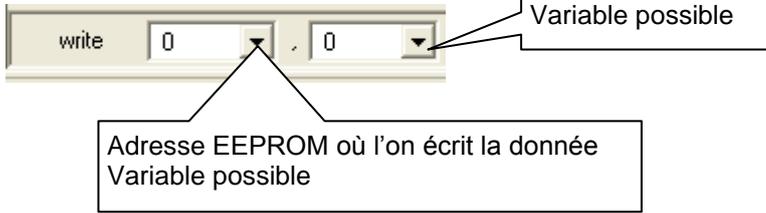
Exemple d'utilisation de la commande write



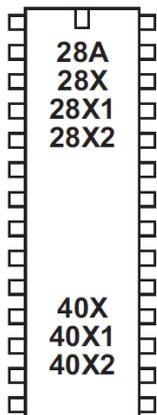
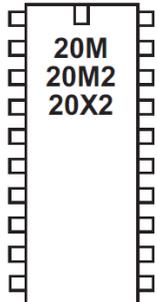
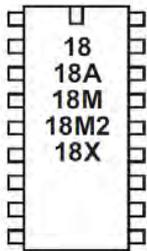
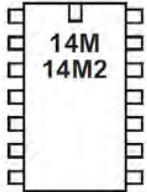
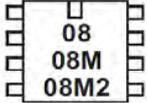
Fonction :

Permet d'écrire une donnée dans la mémoire EEPROM du Picaxe.

Option sélectionnable :



Compatibilité



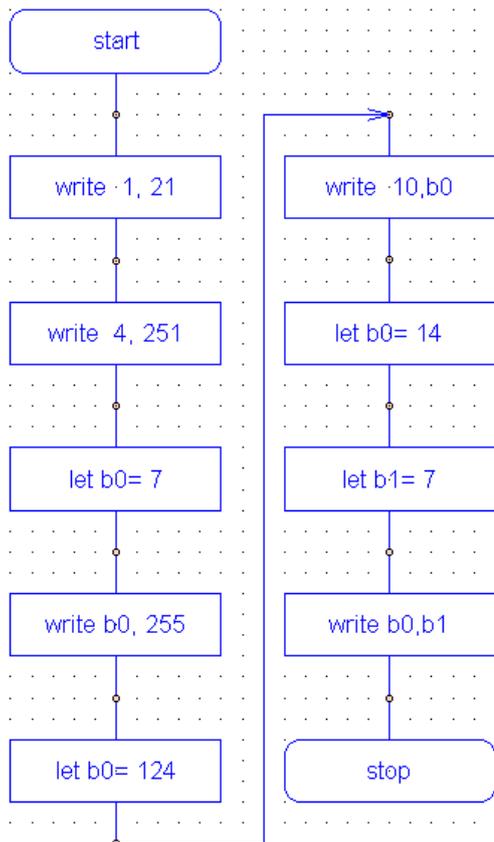
Nota : la donnée est une variable de 8 bits (1 octet).

Le contenu de la mémoire EEPROM n'est pas perdu lorsque l'alimentation est arrêtée.

Les microcontrôleurs Picaxe disposent de 16 emplacements EEPROM, utilisable en écriture, les adresses disponibles vont de l'adresse 0 à l'adresse 15.

En fonction du microcontrôleur Picaxe utilisé, la mémoire de donnée est partagée (voir datasheet).

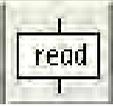
Exemple de programme :



A l'issue du programme, l'EEPROM du Picaxe se charge comme l'exemple à droite :

Organisation de la mémoire EEPROM du Picaxe 08 :

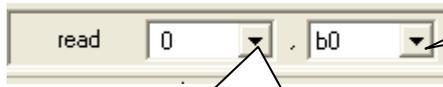
| Adresse | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---------|---|---|----|-----|---|-----|---|---|-----|---|----|----|----|----|----|----|
| 0 | 0 | 0 | 21 | 0 | 0 | 251 | 0 | 0 | 255 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 124 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 24 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 32 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 40 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 48 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 56 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 64 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 72 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 80 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 88 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | P | P | P | P | P | P | P | P | P |
| 96 | P | P | P | P | P | P | P | P | P | P | P | P | P | P | P | P |
| 104 | P | P | P | P | P | P | P | P | P | P | P | P | P | P | P | P |



Fonction :

Permet de lire une donnée stockée dans la mémoire EEPROM du Picaxe.

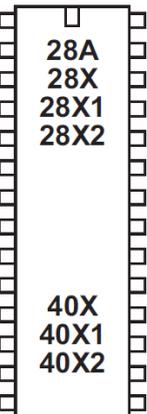
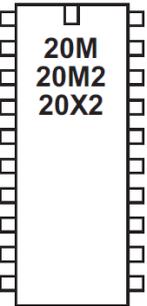
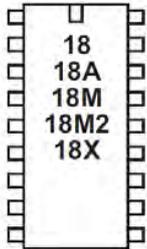
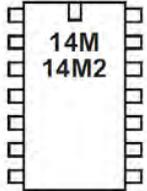
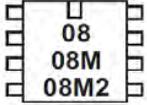
Option sélectionnable :



Variable dans laquelle s'effectue le stockage de la valeur lue dans l'EEPROM

Adresse EEPROM que l'on veut lire
Variable possible

Compatibilité



Nota : la donnée est une variable de 8 bits (1 octet).

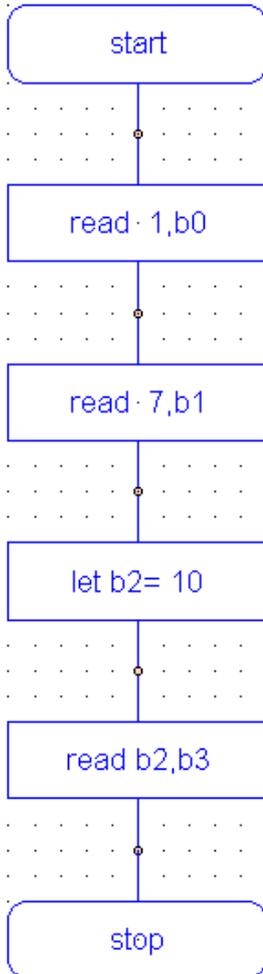
Le contenu de la mémoire EEPROM n'est pas perdu lorsque l'alimentation est arrêtée.

La mémoire EEPROM est effacée à chaque nouveau téléchargement de programme sur le Picaxe.

En fonction du Picaxe utilisée, la taille et le partage de la mémoire EEPROM est différente.

La taille disponible est de 92 adresses exploitables pour le Picaxe 08, 219 adresses exploitables pour le Picaxe 08M et jusqu'à 255 adresses exploitables pour les Picaxes série 'X' (voir datasheet).

Exemple de programme :



A l'issue du programme, si le Picaxe est chargée avec l'EEPROM de l'exemple à droite :

b0 = 21
b1 = 255
b3 = 124

Organisation de la mémoire EEPROM du Picaxe 08 :

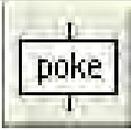
La mémoire EEPROM du Picaxe 08 dispose d'une capacité de 255 adresses. Cependant, seule les adresses 00 à 92 sont exploitables. Les adresses 93 à 255 sont partagées.

Adresse 0 Adresse 1 Adresse 10 Adresse 7

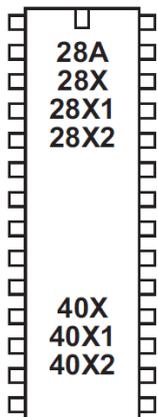
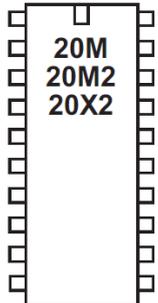
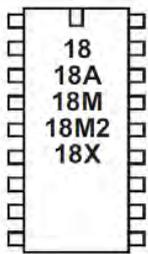
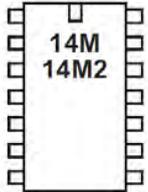
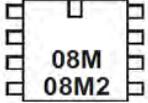
| | | | | | | | | |
|-----|---|----|-----|-----|---|---|---|-----|
| 0 | 0 | 21 | 0 | 124 | 0 | 0 | 0 | 255 |
| 8 | 0 | 0 | 124 | 0 | 0 | 0 | 7 | 0 |
| 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 24 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 32 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 40 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 48 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 56 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 64 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 72 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 80 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 88 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 96 | P | P | P | P | P | P | P | P |
| 104 | P | P | P | P | P | P | P | P |

Data (eeprom/read/write) Hex
 SFR (peek/poke)
 Scratchpad (put/get)
 Table (table/readtable)

Adresse 92
Adresse 93 partagée (indication P)



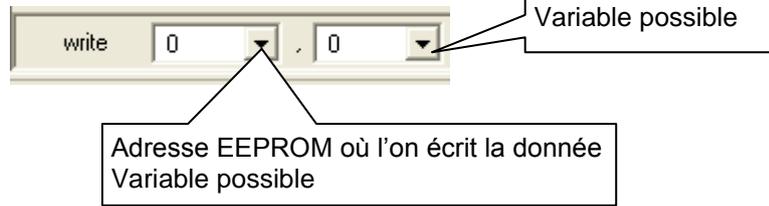
Compatibilité



Fonction :

Permet d'écrire une donnée (variables de 8 bits) dans la mémoire interne du Picaxe.

Option sélectionnable :



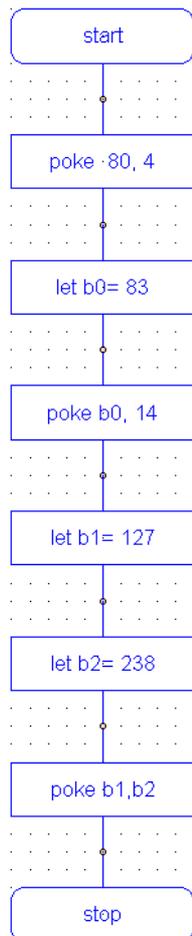
Nota : au-delà des variables normales b0 à b13, les Picaxes disposent d'emplacement mémoire qu'il est possible d'utiliser avec la commande peek. Il devient possible de lire la valeur des registres du Picaxe.

Cette commande demande de bien connaître l'organisation mémoire du Picaxe utilisé (voir datasheet).

Cette commande permet d'accéder à des fonctions spécifiques du Picaxe.

Attention, cela peut aussi provoquer des blocages irréversibles du Picaxe. Pour expert uniquement.

Exemple de programme :



A l'issue du programme, la mémoire interne du Picaxe se charge comme l'exemple à droite :

Ecriture dans la mémoire interne du Picaxe 08 :

La mémoire interne du Picaxe 08 se limite aux adresses 80 à 127.

| Adresse | 48 | 56 | 64 | 72 | 80 | 88 | 96 | 104 | 112 | 120 | 128 | 136 | 144 | 152 |
|---------|----|----|----|----|-----|----|----|-----|-----|-----|-----|-----|-----|-----|
| | -- | -- | -- | -- | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | -- | -- | -- | -- | 14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | -- | -- | -- | -- | 238 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- |
| | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- |
| | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- |
| | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- | -- |



Fonction :

Permet de lire des données (variables de 8 bits) de la mémoire du Picaxe.

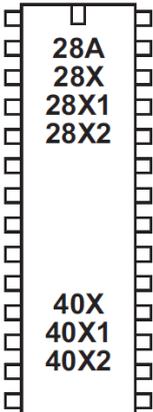
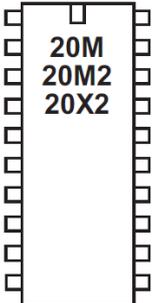
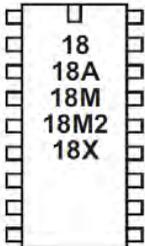
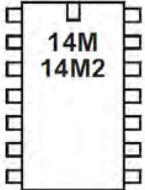
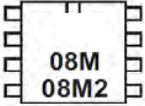
Option sélectionnable :



Variable dans laquelle s'effectue le stockage de la valeur lue

Adresse que l'on veut lire
Variable possible

Compatibilité

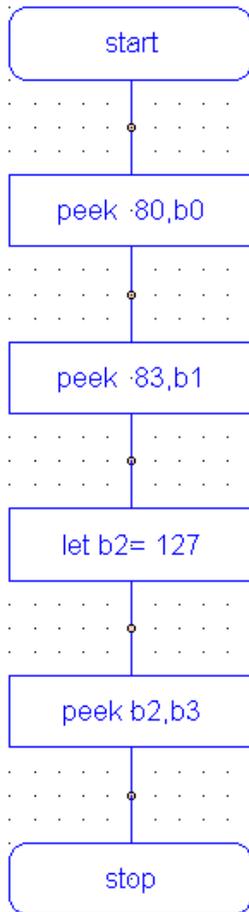


Nota : au-delà des variables normales b0 à b13, les Picaxes disposent d'emplacement mémoire qu'il est possible d'utiliser avec la commande peek. Il devient possible de lire la valeur des registres du Picaxe.

Cette commande demande de bien connaître l'organisation mémoire du Picaxe utilisé (voir datasheet).

Pour expert uniquement.

Exemple de programme :



A l'issue du programme, si le Picaxe est chargée avec l'exemple à droite :

- b0 = 4
- b1 = 14
- b3 = 238

Organisation de la mémoire interne du Picaxe 08 :

La mémoire interne du Picaxe 08 se limite aux adresses 80 à 127.

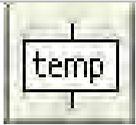
Adresse 80 Adresse 83 Adresse

127

| Adresse | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---------|----|----|----|----|----|----|----|-----|
| 48 | -- | -- | -- | -- | -- | -- | -- | -- |
| 56 | -- | -- | -- | -- | -- | -- | -- | -- |
| 64 | -- | -- | -- | -- | -- | -- | -- | -- |
| 72 | -- | -- | -- | -- | -- | -- | -- | -- |
| 80 | 4 | 0 | 0 | 14 | 0 | 0 | 0 | 0 |
| 88 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 96 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 104 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 112 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 120 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 238 |
| 128 | -- | -- | -- | -- | -- | -- | -- | -- |
| 136 | -- | -- | -- | -- | -- | -- | -- | -- |
| 144 | -- | -- | -- | -- | -- | -- | -- | -- |
| 152 | -- | -- | -- | -- | -- | -- | -- | -- |

Data (eeprom/read/write) Hex
 SFR (peek/poke)
 Scratchpad (put/get)
 Table (table/readtable)

-- : adresse inutilisable



Fonction :

Permet de lire la température en degré Celsius à partir d'un capteur numérique DS18B20.
Exp : capteur de température calibré AutoProg K-AP-MTEMP

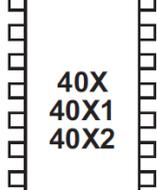
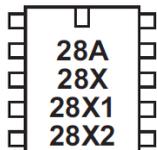
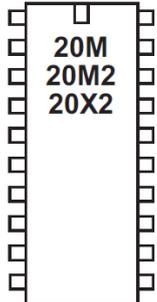
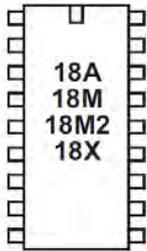
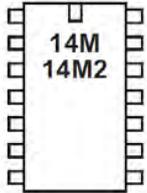
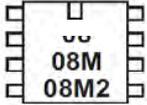
Option sélectionnable :



Adresse EEPROM où l'on écrit la donnée
Variable possible

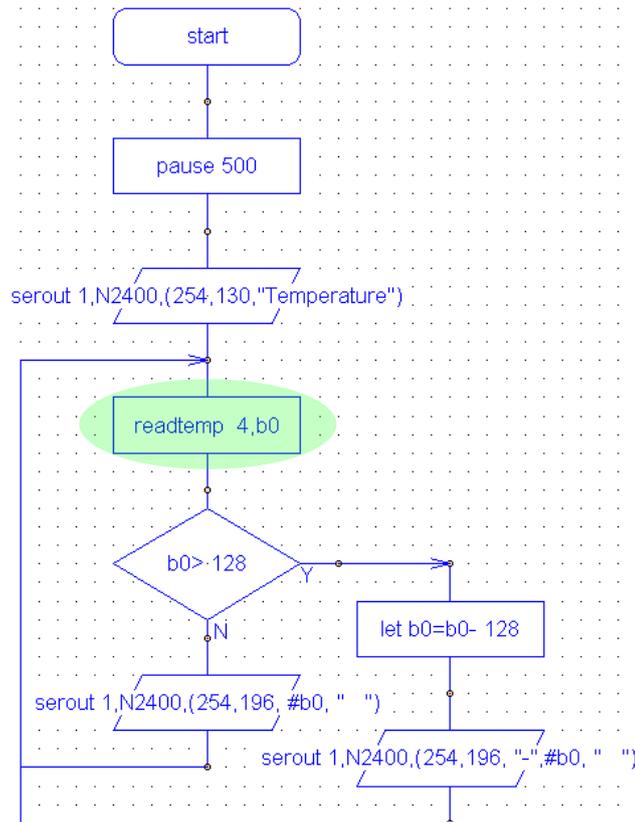
Variable recevant la température détectée
par le capteur numérique DS18B20

Compatibilité



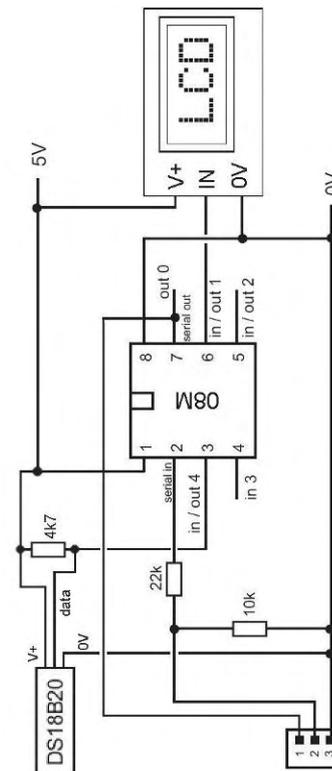
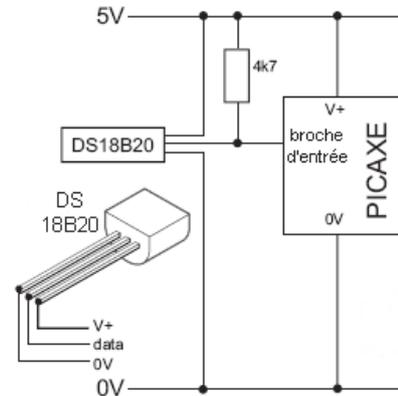
Nota : la température est codée sur 7 bits (b0 à b6), le bit 7 de la variable est utilisé pour signaler une température négative. Si b7 est à 1, la température est négative.
Le capteur DS18B20 fonctionne de -55°C à 125°C

Exemple de programme :



Ce programme permet de gérer le bit7 et autorise l'affichage de la température entre -55°C et 128°C.

Exemple de câblage d'un capteur DS18B20 :





4. Connexion des composants à un microcontrôleur Picaxe

Ce chapitre propose une liste non exhaustive de schémas de câblage qui illustre la mise en service de composant(s) couramment utilisés avec le système Picaxe.

| | |
|---|----|
| 4.1 Caractéristiques principales des microcontrôleurs Picaxe | 64 |
| 4.2 Brochage des microcontrôleurs Picaxe | 65 |
| 4.3 Connecter un capteur tout ou rien (TOR) | 67 |
| 4.4 Connecter un capteur numérique de température DS18B20 | 68 |
| 4.5 Connecter un capteur analogique (LDR, CTN, potentiomètre) | 69 |
| 4.6 Connecter un capteur de distance SRF05..... | 70 |
| 4.7 Connecter un capteur infrarouge | 70 |
| 4.8 Connecter un afficheur LCD | 71 |
| 4.9 Connecter un circuit de puissance (ULN2003) | 71 |
| 4.10 Connecter un circuit de commande moteur (L293D)..... | 72 |

4.1 Caractéristiques principales des microcontrôleurs Picaxe

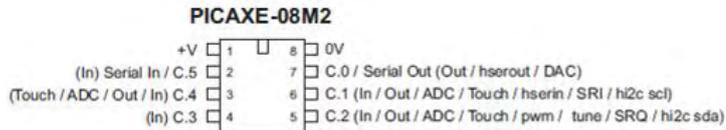
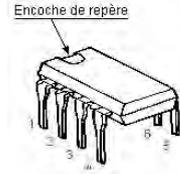
| | 08M2 | 14M2 | 18M2 | 20M2 | 20X2 | 28X2 | 40X2 |
|--|------|------|------|------|------|------|------|
| Capacité de mémoire RAM | 2048 | 2048 | 2048 | 2048 | 4096 | 4096 | 4096 |
| RAM (octets) | 128 | 512 | 256 | 512 | 256 | 1280 | 1280 |
| Variables (octets) | 28 | 28 | 28 | 28 | 56 | 56 | 56 |
| Broches E/S | 6 | 12 | 16 | 18 | 18 | 22 | 33 |
| Convertisseurs analogique / digital et entrées « Touch » | 3 | 7 | 10 | 11 | 11 | 16 | 27 |
| Fréquence maxi | 32 | 32 | 32 | 32 | 64 | 64 | 64 |
| E/S série | oui |
| E/S infrarouge | oui |
| I2C | oui |
| Fonction « Tune » | oui |
| Gestion de tâches en parallèle | 4 | 8 | 4 | 8 | 1 | 1 | 1 |

Nota : Certaines ressources décrites dans ce tableau ne sont pas exploitables avec le mode de programmation graphique « Flowchart » du logiciel Picaxe Programming Editor.

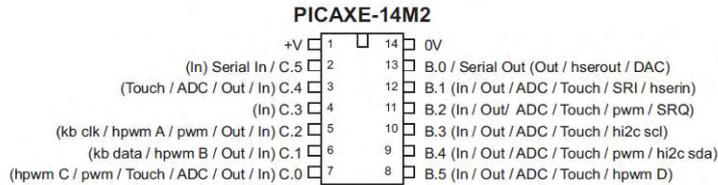
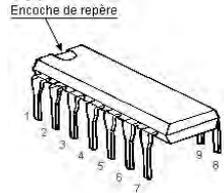


4.2 Brochage des microcontrôleurs Picaxe

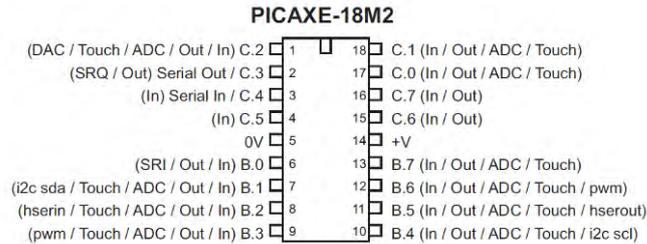
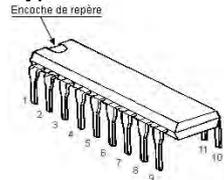
Type 08 :



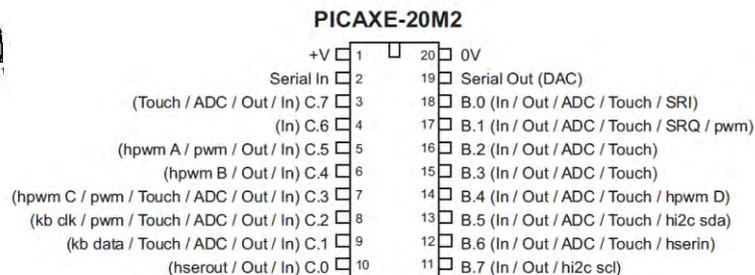
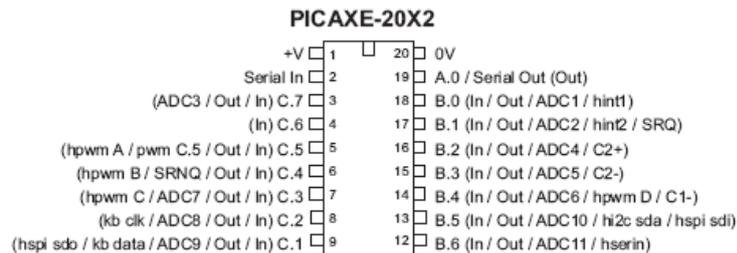
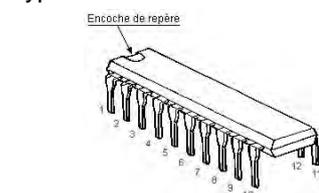
Type 14



Type 18 :

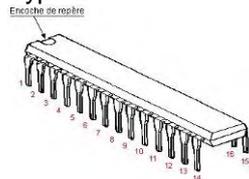


Type 20 :





Type 28 :



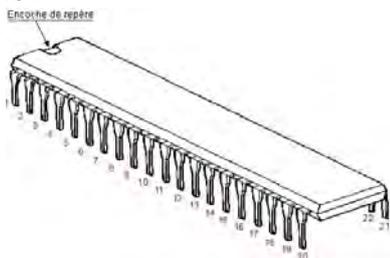
PICAXE-28X1

| | | | |
|-----------------------------------|----|----|-----------------------------------|
| Reset | 1 | 28 | Output 7 |
| ULPWU / ADC 0 / In a0 | 2 | 27 | Output 6 |
| ADC 1 / In a1 | 3 | 26 | Output 5 |
| ADC 2 / In a2 | 4 | 25 | Output 4 / hpwm D |
| ADC 3 / In a3 | 5 | 24 | Output 3 |
| Serial In | 6 | 23 | Output 2 / hpwm B |
| Serial Out | 7 | 22 | Output 1 / hpwm C |
| 0V | 8 | 21 | Output 0 |
| Resonator | 9 | 20 | +V |
| Resonator | 10 | 19 | 0V |
| timer clk / Out c0 / In 0 | 11 | 18 | In 7 / Out c7 / hserin / kb data |
| pwm 1 / Out c1 / In 1 | 12 | 17 | In 6 / Out c6 / hserout / kb clk |
| hpwm A / pwm 2 / Out c2 / In 2 | 13 | 16 | In 5 / Out c5 / spi sdo |
| spi sck / i2c scl / Out c3 / In 3 | 14 | 15 | In 4 / Out c4 / i2c sda / spi sdi |

PICAXE-28X2

| | | | |
|---------------------------|----|----|--------------------------------|
| Reset | 1 | 28 | B.7 |
| C1- / ADC0 / A.0 | 2 | 27 | B.6 |
| C2- / ADC1 / A.1 | 3 | 26 | B.5 |
| C2+ / ADC2 / A.2 | 4 | 25 | B.4 / ADC11 / (hpwm D) |
| C1+ / ADC3 / A.3 | 5 | 24 | B.3 / ADC9 |
| Serial In | 6 | 23 | B.2 / ADC8 / hint2 / (hpwm B) |
| Serial Out / A.4 | 7 | 22 | B.1 / ADC10 / hint1 / (hpwm C) |
| 0V | 8 | 21 | B.0 / ADC12 / hint0 |
| Resonator | 9 | 20 | +V |
| Resonator | 10 | 19 | 0V |
| timer clk / C.0 | 11 | 18 | C.7 / hserin / kb data |
| pwm C.1 / C.1 | 12 | 17 | C.6 / hserout / kb clk |
| (hpwm A) / pwm C.2 / C.2 | 13 | 16 | C.5 / hspi sdo |
| hi2c scl / hspi sck / C.3 | 14 | 15 | C.4 / hi2c sda / hspi sdi |

Type 40 :



PICAXE-40X1

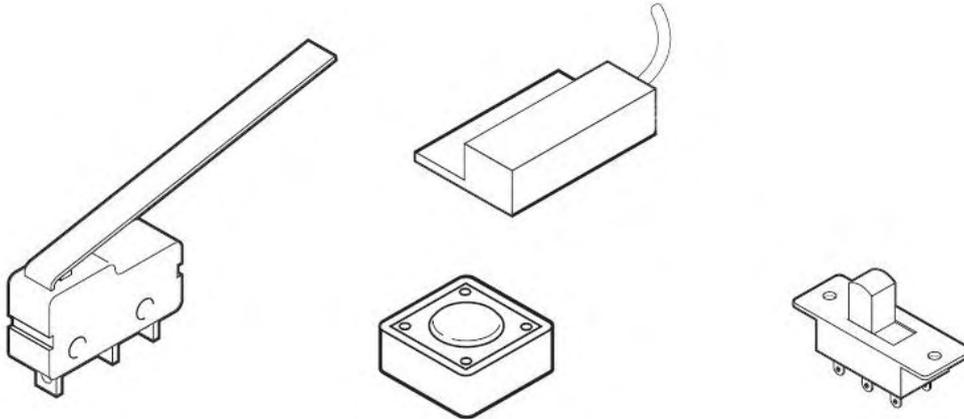
| | | | |
|------------------------------------|----|----|------------------------------------|
| Reset | 1 | 40 | Output 7 |
| ULPWU / ADC 0 / In a0 | 2 | 39 | Output 6 |
| ADC 1 / In a1 | 3 | 38 | Output 5 |
| ADC 2 / In a2 | 4 | 37 | Output 4 |
| ADC 3 / In a3 | 5 | 36 | Output 3 |
| Serial In | 6 | 35 | Output 2 |
| Serial Out | 7 | 34 | Output 1 |
| ADC 5 | 8 | 33 | Output 0 |
| ADC 6 | 9 | 32 | +V |
| ADC 7 | 10 | 31 | 0V |
| +V | 11 | 30 | Input 7 / kb data |
| 0V | 12 | 29 | Input 6 / kb clk |
| Resonator | 13 | 28 | Input 5 |
| Resonator | 14 | 27 | Input 4 |
| timer clk / Out c0 / In c0 | 15 | 26 | In c7 / Out c7 / hserin |
| pwm 1 / Out c1 / In c1 | 16 | 25 | In c6 / Out c6 / hserout |
| pwm 2 / Out c2 / In c2 | 17 | 24 | In c5 / Out c5 / spi sdo |
| i2c scl / spi sck / Out c3 / In c3 | 18 | 23 | In c4 / Out c4 / i2c sda / spi sdi |
| Input 0 | 19 | 22 | Input 3 |
| Input 1 | 20 | 21 | Input 2 |

PICAXE-40X2

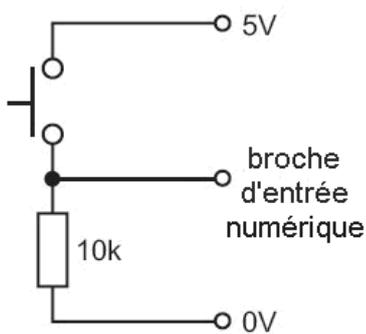
| | | | |
|---------------------------|----|----|---------------------------|
| Reset | 1 | 40 | B.7 |
| C1- / ADC0 / A.0 | 2 | 39 | B.6 |
| C2- / ADC1 / A.1 | 3 | 38 | B.5 |
| C2+ / ADC2 / A.2 | 4 | 37 | B.4 / ADC11 |
| C1+ / ADC3 / A.3 | 5 | 36 | B.3 / ADC9 |
| Serial In | 6 | 35 | B.2 / ADC8 / hint2 |
| Serial Out / A.4 | 7 | 34 | B.1 / ADC10 / hint1 |
| ADC5 / A.5 | 8 | 33 | B.0 / ADC12 / hint0 |
| ADC6 / A.6 | 9 | 32 | +V |
| ADC7 / A.7 | 10 | 31 | 0V |
| +V | 11 | 30 | D.7 / hpwm D / kb data |
| 0V | 12 | 29 | D.6 / hpwm C / kb clk |
| Resonator | 13 | 28 | D.5 / hpwm B |
| Resonator | 14 | 27 | D.4 |
| timer clk / C.0 | 15 | 26 | C.7 / hserin |
| pwm C.1 / C.1 | 16 | 25 | C.6 / hserout |
| hpwm A / pwm C.2 / C.2 | 17 | 24 | C.5 / hspi sdo |
| hi2c scl / hspi sck / C.3 | 18 | 23 | C.4 / hi2c sda / hspi sdi |
| D.0 | 19 | 22 | D.3 |
| D.1 | 20 | 21 | D.2 |

4.3 Connecter un capteur tout ou rien (TOR)

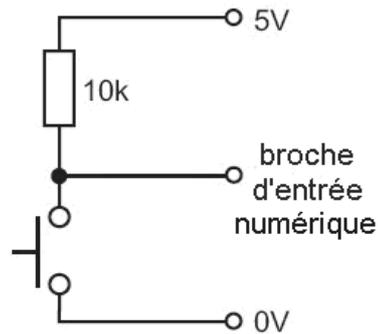
Ces capteurs disposent généralement de 2 contacts pouvant être ouverts ou fermés : microswitch, interrupteur ou ILS.



Ces capteurs permettent d'obtenir 2 valeurs de potentiel : 0V ou 5V représentatif des 2 états logiques : 0 ou 1. La broche d'entrée correspond à une entrée numérique du Picaxe. Selon le circuit, il est possible d'obtenir un niveau 0 ou 1 en position normalement ouvert.

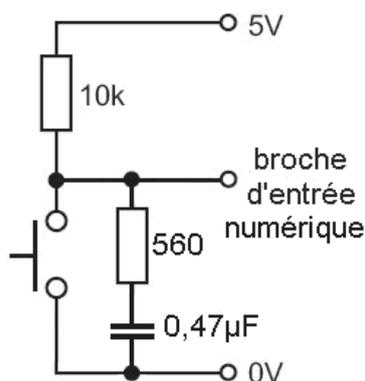


Présence d'un 0 quand le capteur TOR ouvert



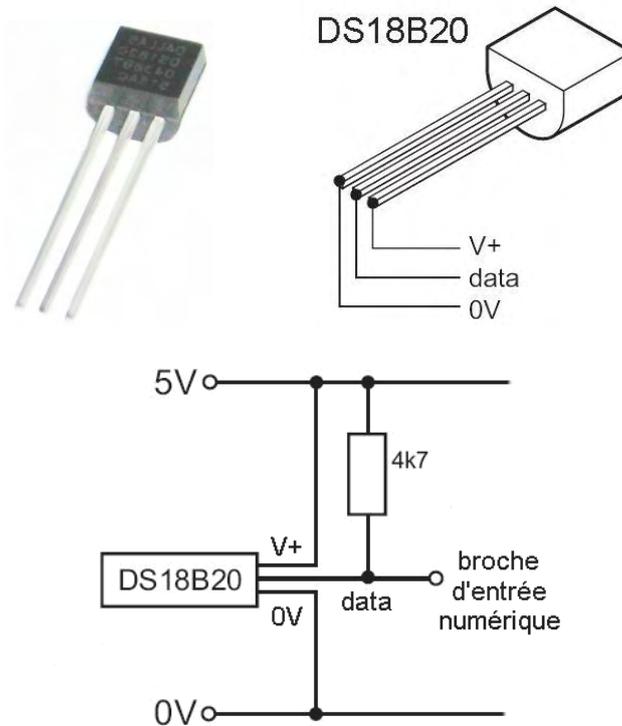
Présence d'un 1 quand le capteur est ouvert

Système anti-rebonds afin d'éviter des commutations parasites indésirables.



4.4 Connecter un capteur numérique de température DS18B20

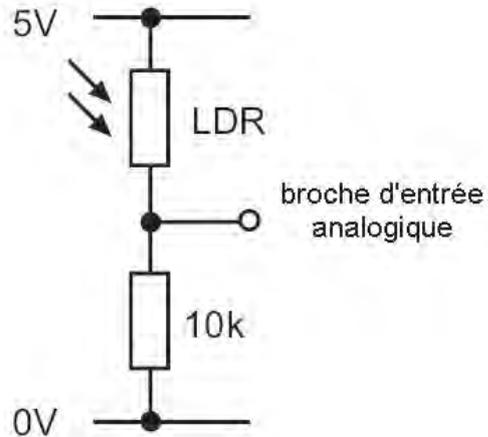
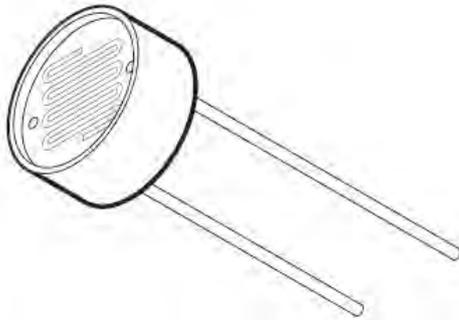
Ce capteur de température permet de fournir une information de -55°C à 125°C sur une seule ligne en utilisant la technique « 1-Wire Maxim Dallas ». Le codage de la valeur est sur 8 bits. Le bit de poids fort (valeur >127) indique le signe (positif ou négatif) de la température.



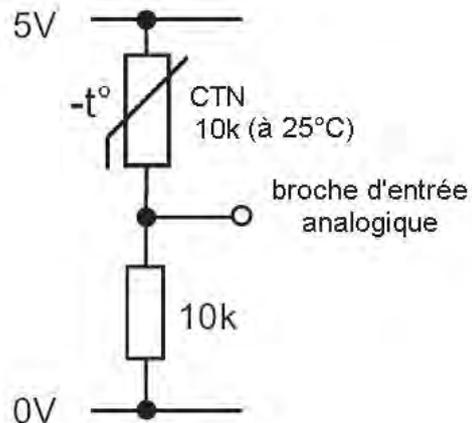
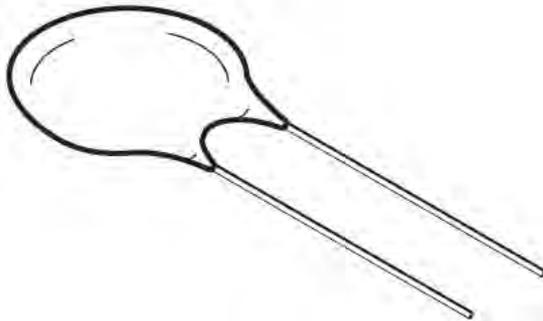
La broche d'entrée correspond à une entrée numérique du microcontrôleur Picaxe.

4.5 Connecter un capteur analogique (LDR, CTN, potentiomètre)

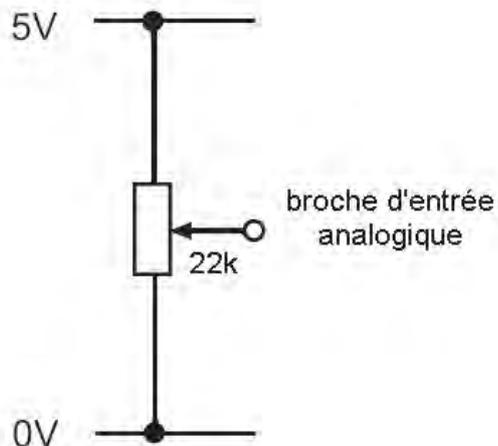
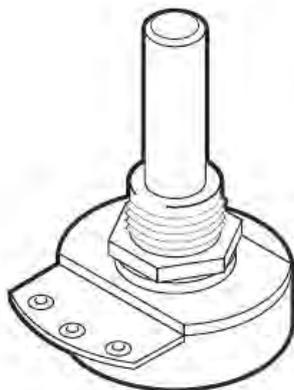
Le capteur LDR est un capteur résistif sensible à la lumière. La résistance diminue lorsque le niveau de lumière augmente. Le schéma potentiométrique permet d'obtenir une tension analogique évoluant entre $\approx 0V$ (LDR dans le noir) et $\approx 5V$ (LDR en pleine lumière).



Le capteur CTN est un capteur résistif sensible à la température. La résistance diminue lorsque la température augmente. Le schéma potentiométrique permet d'obtenir une tension analogique évoluant autour de 2,5V (valeur à 25°C). La tension augmente avec la température.

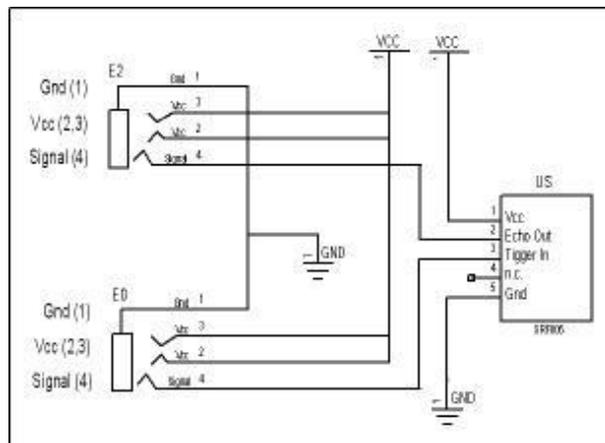
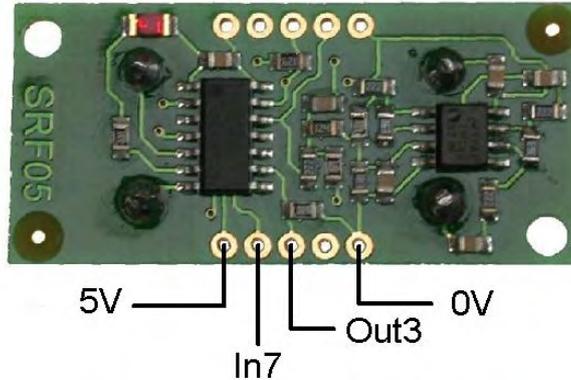


Le potentiomètre est un capteur de position.



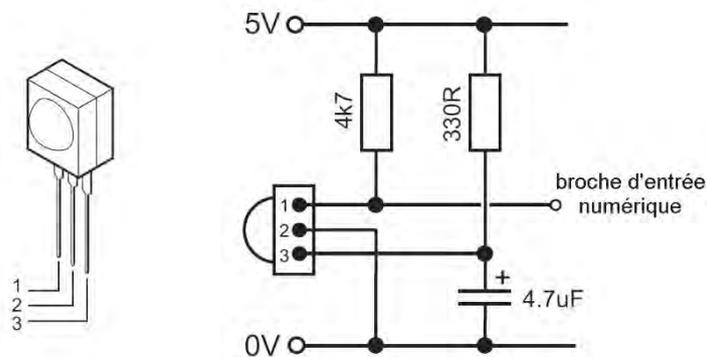
4.6 Connecter un capteur de distance SRF05

Le capteur de distance SRF05 est constitué d'un émetteur et d'un récepteur à ultrasons. Pour fonctionner avec la commande « ultra », le Picaxe envoie un signal en utilisant la sortie Out3 et vérifie le temps de retour de ce signal sur l'entrée In7.



4.7 Connecter un capteur infrarouge

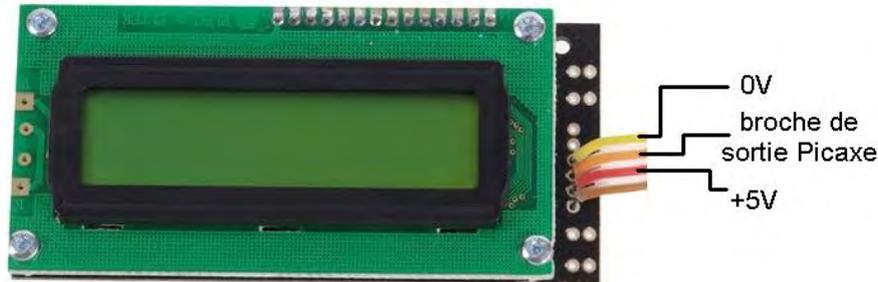
Le capteur infrarouge est en réalité un circuit intégré constitué d'une diode de réception infrarouge et de circuits de traitement permettant de s'affranchir des perturbations dues à l'éclairage ambiant. Ce circuit combine à la fois sensibilité et facilité de mise en œuvre.



4.8 Connecter un afficheur LCD

L'afficheur LCD permet d'afficher un message provenant d'un microcontrôleur Picaxe. Il est possible d'afficher jusqu'à 2 lignes de 16 caractères.

Le module afficheur décode une information d'une sortie du Picaxe en utilisant l'instruction « serout ».

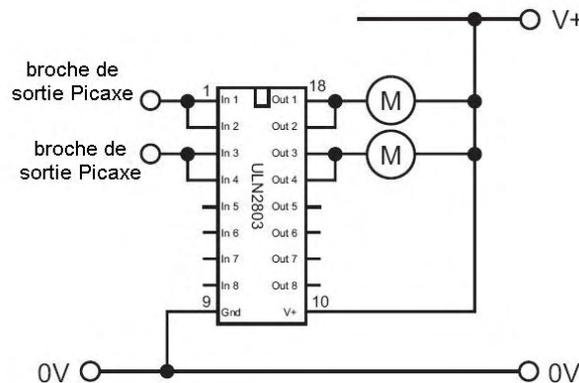


La liaison avec le fil marron est utilisée (avec une broche d'entrée Picaxe) si vous utilisez le circuit horloge optionnel.

4.9 Connecter un circuit de puissance

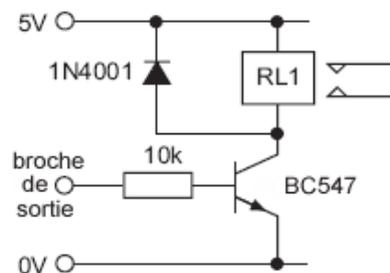
Le courant de sortie d'un Picaxe est limité à 20mA, pour commander des périphériques nécessitant un courant de sortie supérieur, l'utilisation d'un circuit intégré ULN 2803 (8 sorties) est particulièrement simple il permet de piloter jusqu'à 8 moteurs ou éléments de puissance simultanément. Exemple pour la commande de 2 moteurs. Pour augmenter le courant de sortie, il peut être utile de « coupler » les étages de puissance des circuits comme indiquer sur le circuit.

Exemple pour la commande de 2 moteurs



Note : ce type de montage ne permet pas de changer le sens de rotation. Pour la commande d'un seul élément de puissance on peut aussi réaliser le montage soit même, il suffit simplement d'utiliser un transistor.

Exemple, pour la commande d'un relais :

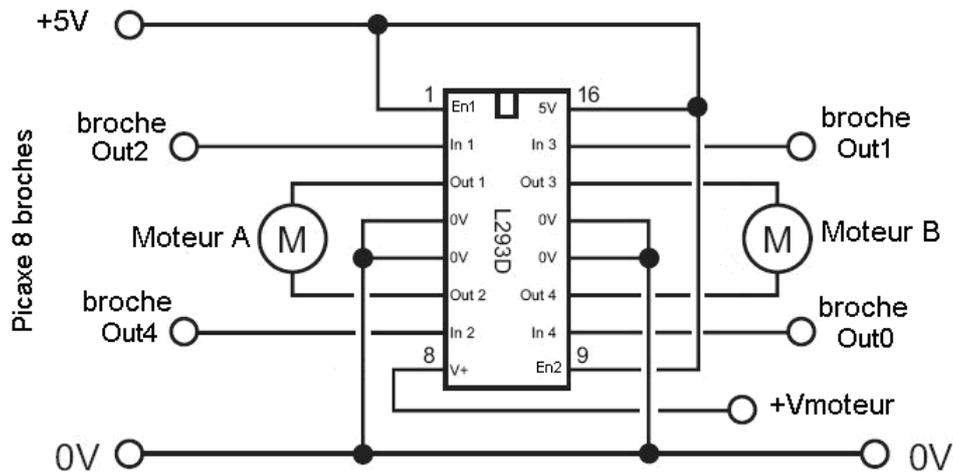


4.10 Connecter un circuit de commande moteur

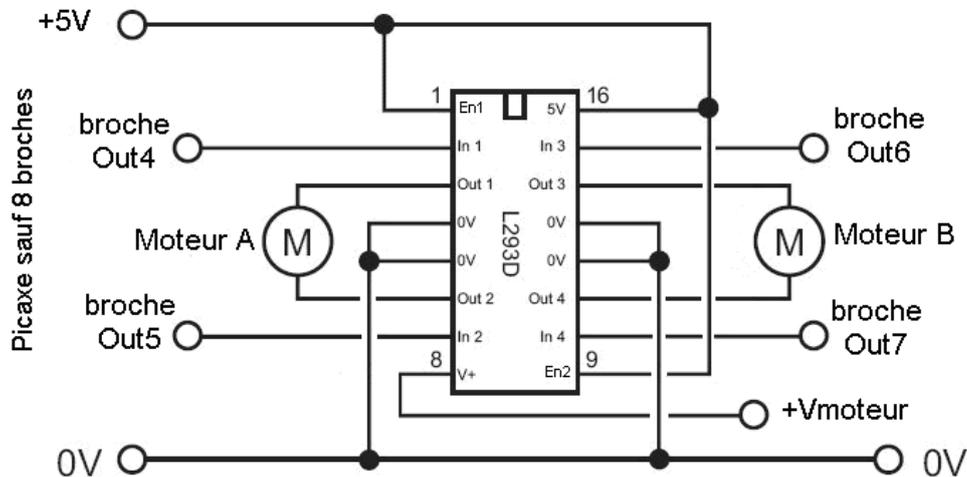
Le circuit de commande moteur de type L293D (commande de 2 moteurs jusqu'à 600mA) permet de contrôler 2 moteurs à courant continu en utilisant 4 lignes de données du Picaxe.

Si un seul moteur doit être contrôlé, il suffira d'utiliser uniquement 2 lignes de sorties du Picaxe.

Avec un Picaxe 8 broches (08 ou 08M), le moteur A est commandé par les lignes Out2 et Out4. Le moteur B est commandé par les lignes Out0 et Out1. Ces lignes peuvent être différentes mais dans ce cas là, il sera nécessaire de modifier les valeurs de déplacement dans le menu options.



Avec les autres Picaxes (sauf 14M), le moteur A est commandé par les lignes Out4 et Out5. Le moteur B est commandé par les lignes Out6 et Out7. Ces lignes peuvent être différentes mais dans ce cas là, il sera nécessaire de modifier les valeurs de déplacement dans le menu options.





5. Annexes

| | |
|---|----|
| 5.1 Installation de « Picaxe Programming Editor »..... | 74 |
| 5.2 Installation et mise en service du câble de programmation « Picaxe Programming Editor » | 77 |
| 5.3 Vérification du Firmware | 86 |
| 5.4. Utilisation et syntaxe de la table des symboles | 87 |
| 5.5 Utilisation du mode simulation | 88 |
| 5.6 Représentation des microcontrôleurs sous « Picaxe Programming Editor » | 90 |
| 5.7 Lexique..... | 92 |
| 5.8 Forum Picaxe francophone..... | 92 |

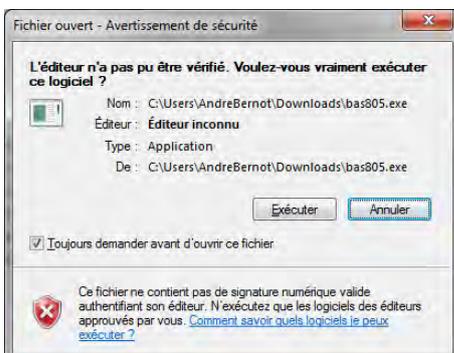
L'installation du logiciel « Picaxe Programming Editor » ainsi que l'installation et la mise en service du câble de programmation sont décrites dans les sous-chapitres 5.1 et 5.2 des annexes de ce guide. Il est nécessaire de procéder à ces installations avant de poursuivre la lecture de ce guide.

5.1 Installation de « Picaxe Programming Editor »

L'application évolue régulièrement, il est recommandé d'installer la dernière version de Picaxe Programming Editor à partir du site www.a4.fr.

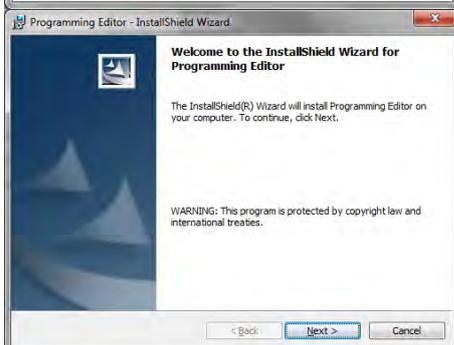
Double cliquer « **bas805** » pour lancer le processus d'installation.

1



Cliquer sur exécuter

2



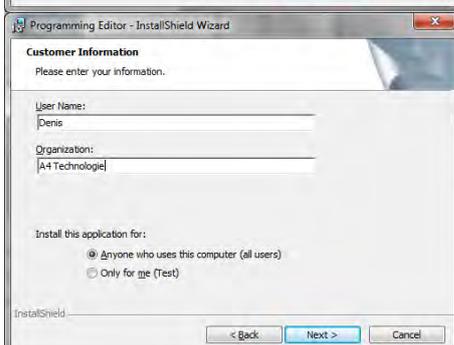
Cliquer sur Next

3



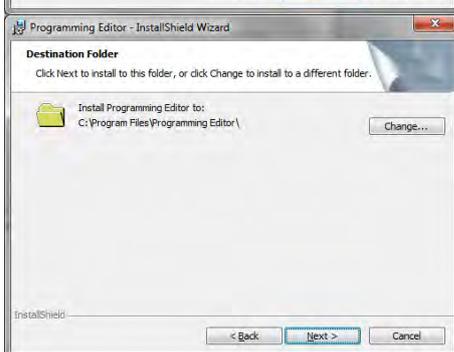
Sélectionner « I accept... » puis cliquer sur Next

4



Renseigner les champs puis cliquer sur Next.

5



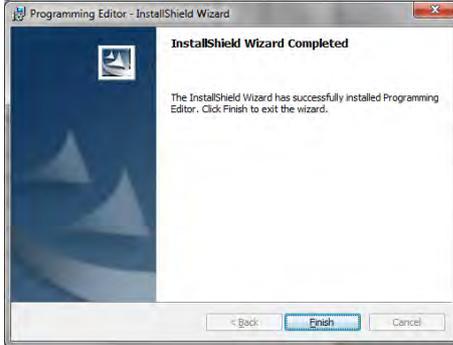
Cliquer sur Next

6



Cliquer sur Install

7



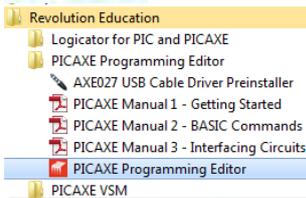
Attendre la fin du processus d'installation puis cliquer sur Finish.

8



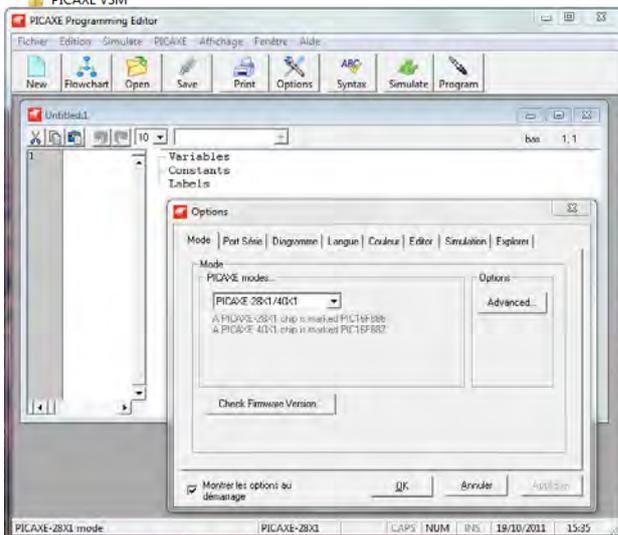
L'icône de lancement est sur le bureau. Cliquer sur cet icône pour lancer Programming Editor.

9



L'application peut aussi être lancée à partir du menu Démarrer \ Tous les programmes \ Revolution Education \ PICAXE Programming Editor \ PICAXE Programming Editor

10



Après le lancement, la fenêtre ci-dessus s'affiche.

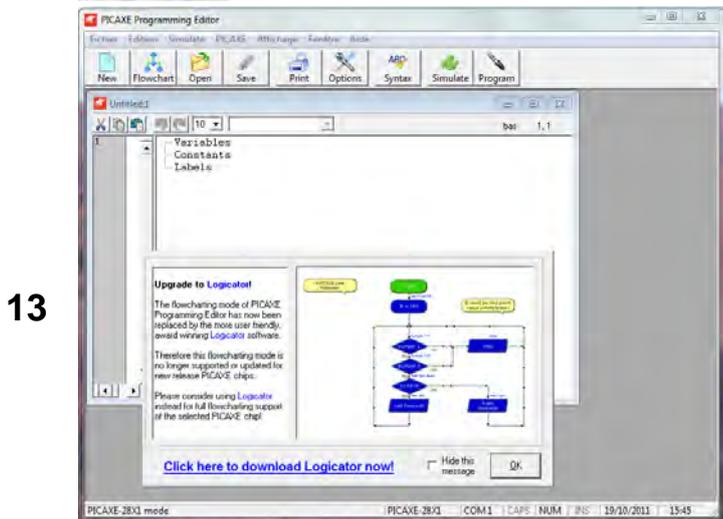
11



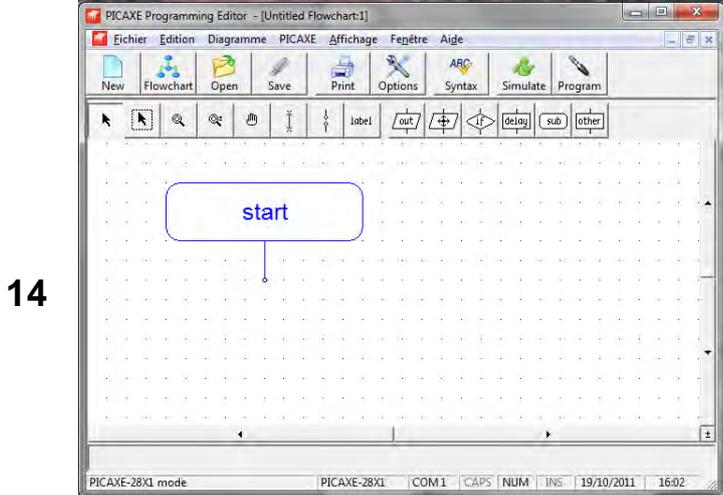
Dans la fenêtre Options, sélectionner l'onglet « Language » et cliquer sur le drapeau souhaité pour la langue d'affichage des menus, puis sur le bouton « Apply ».



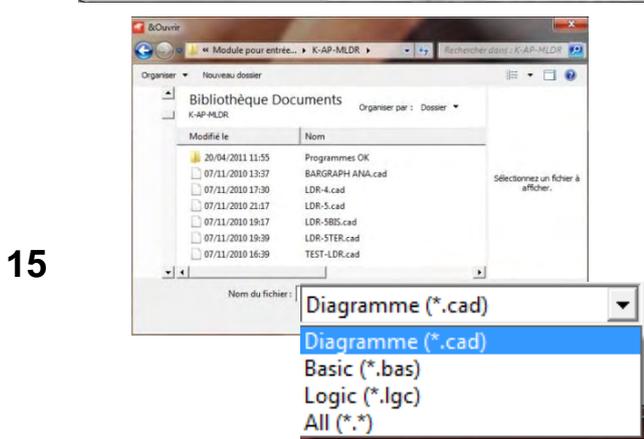
Cliquer sur l'icone « **Flowchart** » (organigramme) pour passer en mode de programmation graphique.



Valider l'écran par OK pour afficher la fenêtre de programmation en mode graphique



A présent vous pouvez ouvrir un modèle existant ou créer votre propre programme.



Si vous ouvrez un modèle existant, vérifier que le type de fichier à ouvrir est « Diagramme (*.cad) ».

5.2 Procédure de mise en service du câble de programmation Picaxe

5.2.1 Introduction

Le câble de programmation Picaxe permet d'assurer la liaison entre l'ordinateur et le système Picaxe destiné à être programmé. Il est équipé à une extrémité d'un connecteur pour port série ou bien pour port USB et à l'autre extrémité d'une fiche jack stéréo d 3,5mm compatible des embases jack qui équipent toutes les carte Picaxe. Les applications « Programming Editor » ou « Pic Logicator » possèdent un menu Option qui donne accès à la configuration du câble de programmation.



Câble de programmation Picaxe pour port USB
(Réf. **CABLE- USB-PICAXE**)



Câble de programmation Picaxe pour port série
(Réf. **CABLE- FP**)

La mise en service du câble de programmation Picaxe nécessite au préalable d'avoir installé la dernière version de l'application Programming Editor. La dernière version de cette application est téléchargeable sur www.a4.fr.

Avant d'utiliser le câble de programmation Picaxe en vue de transférer un programme dans un microcontrôleur Picaxe, il est nécessaire de procéder à sa mise en service.

5.2.2 Mise en service du câble de programmation pour port série

Connecter le câble sur un port série (9 points) de l'ordinateur (aucun pilote n'est nécessaire pour mettre en service ce câble).

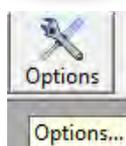
Nota : la mise en service du câble de programmation pour port série est identique sous Windows XP et W7.



Câble de programmation Picaxe pour port série
réf. **CABLE- FP**

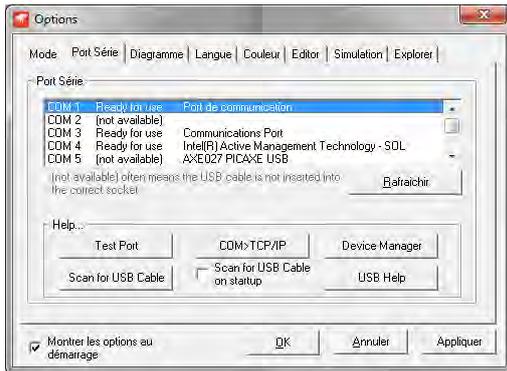


Connecteur port série côté PC



Lancer l'application Programming Editor puis ouvrir le menu « Options »

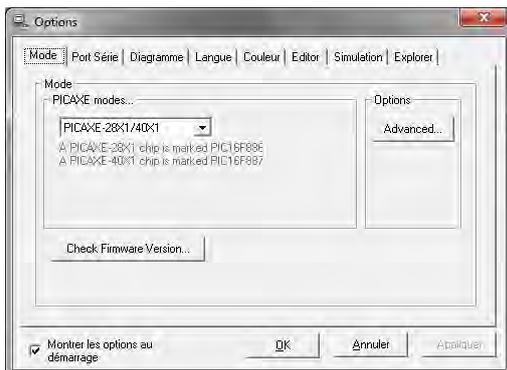
1



Sélectionner le port COM sur lequel est connecté le câble de programmation série puis cliquer sur OK. Il s'agit en général du port COM1 ou COM2, l'information « Ready for use » s'affiche en face du port concerné.

Menu Option / Port Série de Programming Editor

2



Pour vérifier que le câble série est opérationnel, sélectionnez l'onglet Mode de la fenêtre Option. Connecter la fiche jack dans l'embase d'une carte Picaxe, mettre sous tension la carte, et cliquer sur le bouton « Check Firwmare Version... »

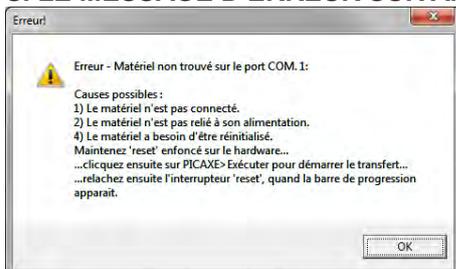
3



Si la communication est correcte la fenêtre ci-dessus apparaît indiquant le type de microcontrôleur connecté. A ce stade, le câble série est en ordre de marche.

SI LE MESSAGE D'ERREUR SUIVANT APPARAÎT

4



Après avoir vérifié les causes indiquées dans le message, vérifiez que le câble de programmation connecté est bien de type Série et non USB. Si le dysfonctionnement persiste, il se peut que votre carte Picaxe soit défectueuse ou que la procédure d'installation n'a pas été entièrement suivie.



5.2.3 Mise en service du câble de programmation pour port USB pour Windows XP et W7

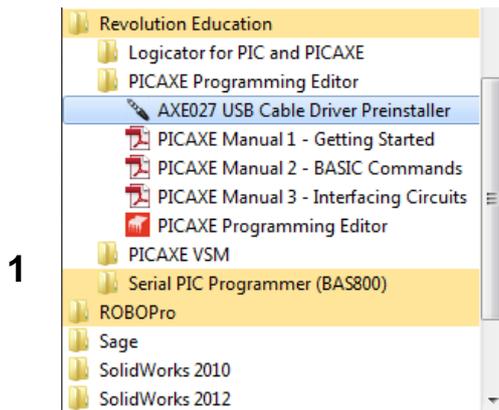


Câble de programmation Picaxe pour port USB
(Réf. **CABLE-USBPICAXE**)



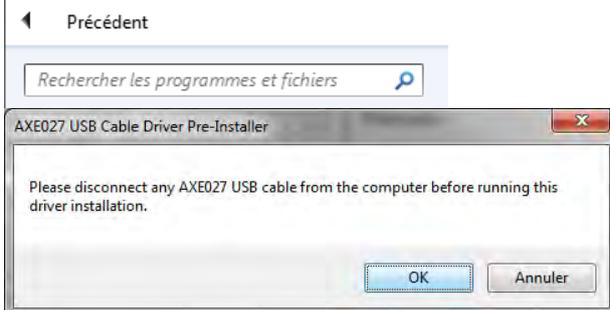
Connecteur port USB

La mise en service du câble de programmation de type USB nécessite d'installer un pilote approprié.



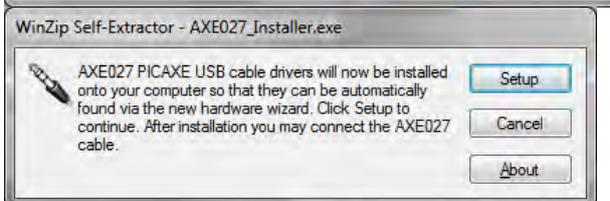
1

Dans le menu démarrer sélectionner l'assistant d'installation des pilotes du câble de programmation dans :
Démarrer \ Tous les programmes \ Revolution Education \ PICAXE Programming Editor \ AXE027 USB Cable Driver Preinstaller



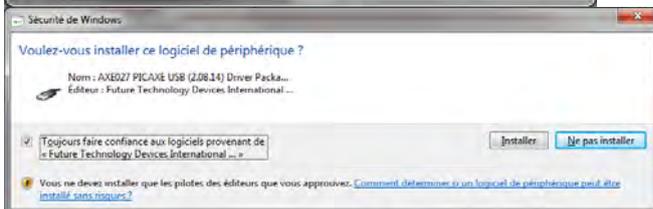
2

Déconnecter le câble si il est connecté, valider avec OK



3

Cliquer sur « Setup ».



4

Cocher la case « Toujours faire confiance aux logiciels provenant de « Future Technology Device International... » puis cliquer sur installer.



5

Il est important d'attendre la fin du processus d'installation avant de connecter le câble à l'ordinateur.
Le processus peut prendre quelques minutes. La disparition de la fenêtre ci-contre indique la fin du processus.

A ce stade si votre ordinateur est installé avec Windows 7, passer directement à l'étape 14.

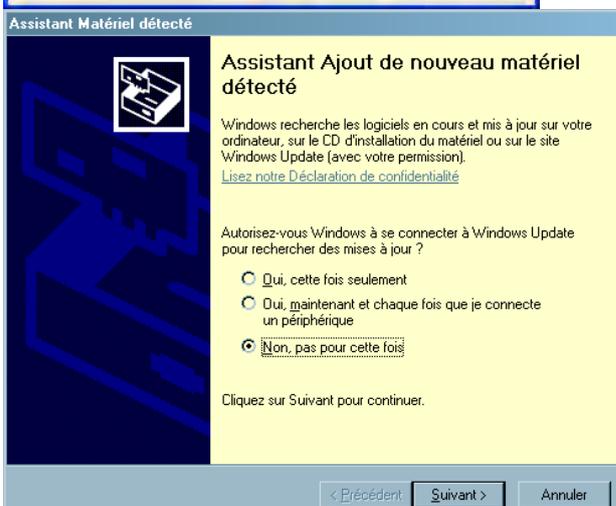
Installation sous XP après connexion du câble de programmation USB

6



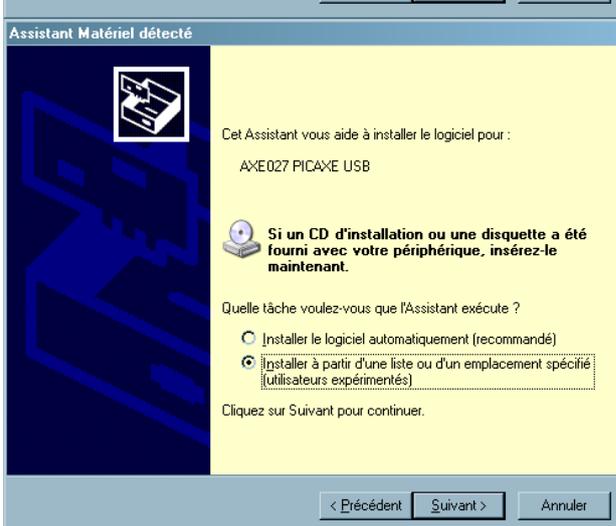
Si ce type de fenêtre apparaît à la connexion du câble, il est nécessaire de s'identifier comme administrateur local de la machine.

7



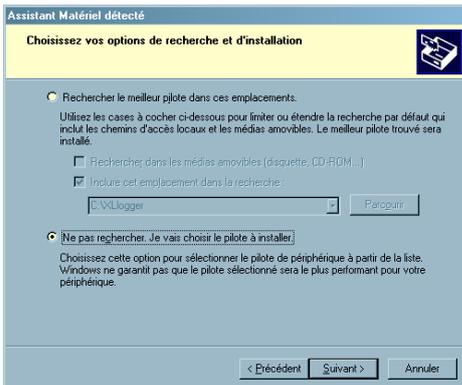
Sélectionner « Non, pas pour cette fois » puis cliquer sur le bouton « Suivant ».

8



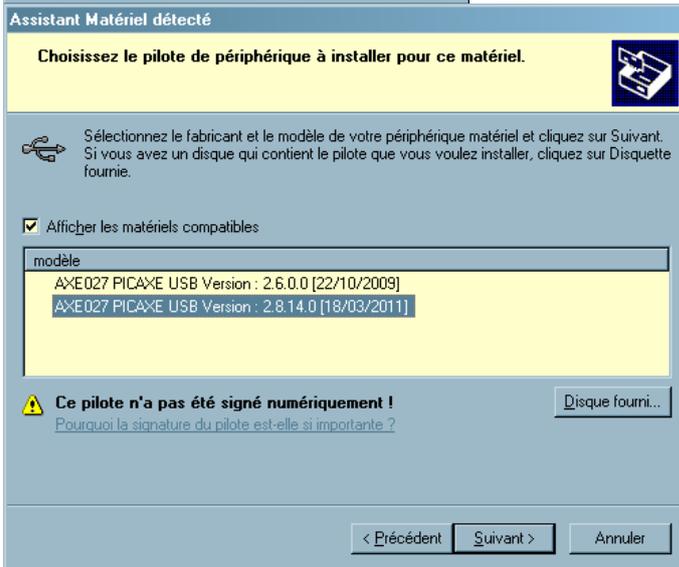
Sélectionner « Installer à partir d'une liste ou d'un emplacement spécifié » puis cliquer sur le bouton « Suivant ».

9



Sélectionner « Ne pas rechercher. Je vais choisir le pilote à installer » puis cliquer sur Suivant.

10



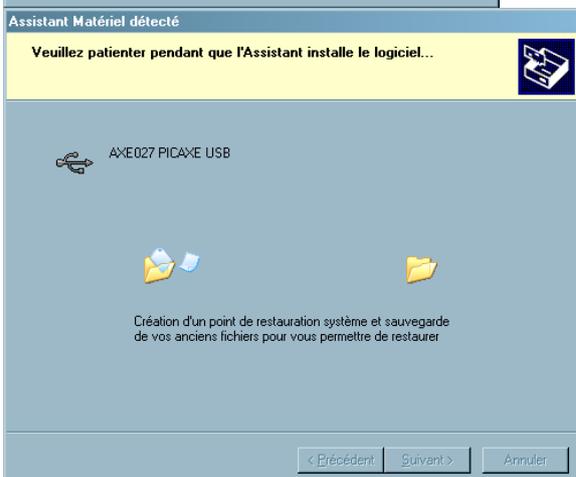
Sélectionner le pilote le plus récent (le dernier de la liste), puis cliquer sur Suivant.

11



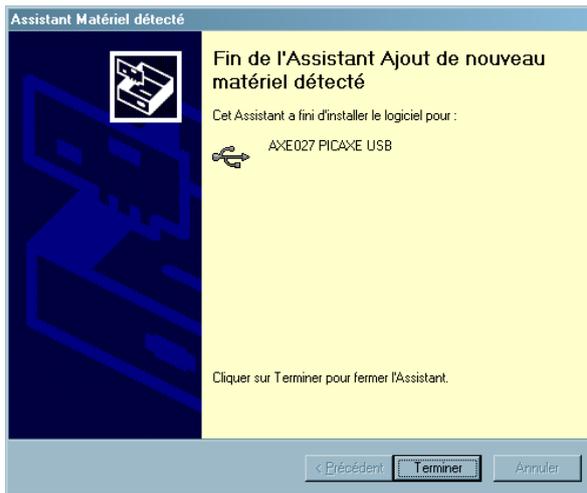
Cliquer sur « Continuer ».

12



Déroulement de l'installation (pas d'action)

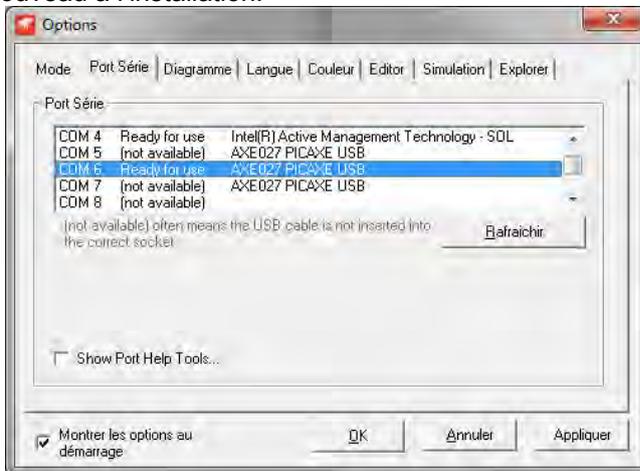
13



Cliquer sur « Terminer ».

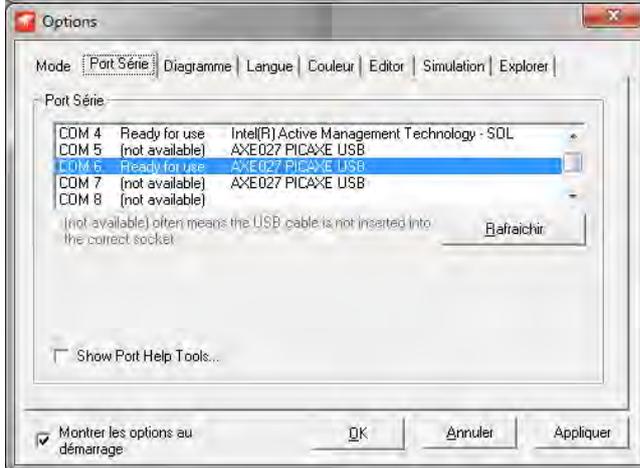
Il se peut qu'à l'issue de l'installation la première fenêtre réapparaisse ; dans ce cas, il est nécessaire de procéder de nouveau à l'installation.

14



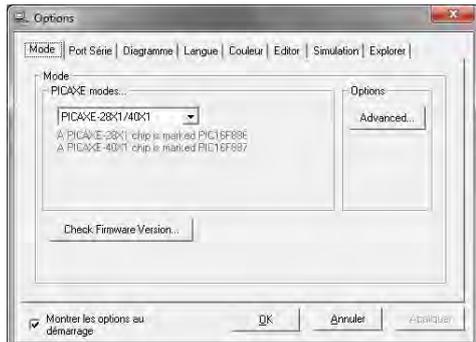
Connecter le câble de programmation USB à l'ordinateur, puis lancer Programming Editor.

15



Dans le menu Options, l'onglet Port Série affiche la liste des ports COM de l'ordinateur. L'inscription « AXE027 PICAXE USB » avec le statut « Ready for use » doit apparaître en face d'un port COM. Sélectionner ce port COM et cliquer sur appliquer.

16



Pour vérifier que le câble USB est opérationnel, sélectionnez l'onglet Mode de la fenêtre Option. Connecter la fiche jack dans l'embase d'une carte Picaxe, mettre sous tension la carte, et cliquer sur le bouton « Check Firwmare Version... ».

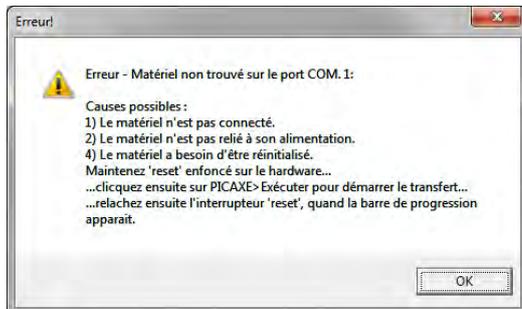
17



Si la communication est correcte la fenêtre ci-dessus apparaît indiquant le type de microcontrôleur connecté. A ce stade, le câble série est en ordre de marche.

SI LE MESSAGE D'ERREUR SUIVANT APPARAÎT

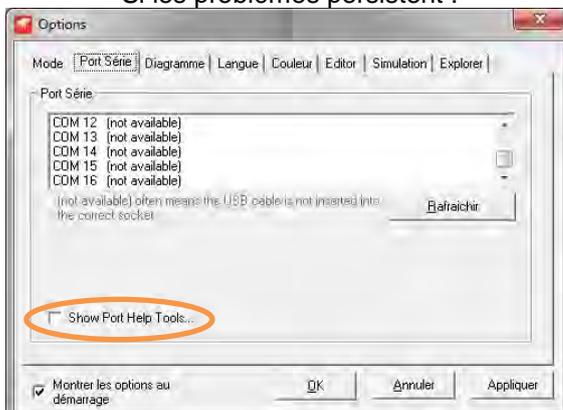
18



Si le dysfonctionnement persiste, il se peut que votre carte Picaxe soit défectueuse ou que la procédure d'installation n'a pas été entièrement suivie. Réappliquer la procédure si nécessaire.

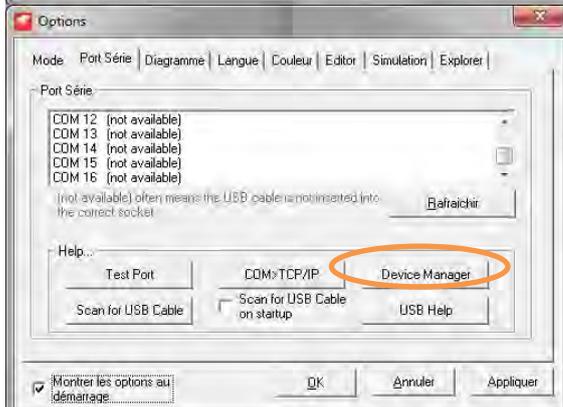
Si les problèmes persistent :

19



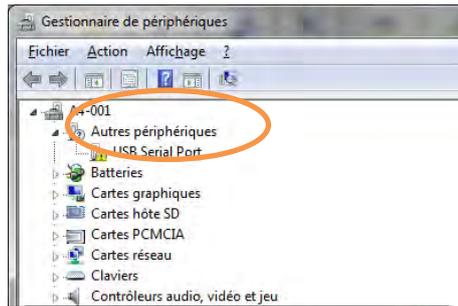
Sélectionner l'onglet Port Série et cocher la case « Show Port Help Tools ».

20



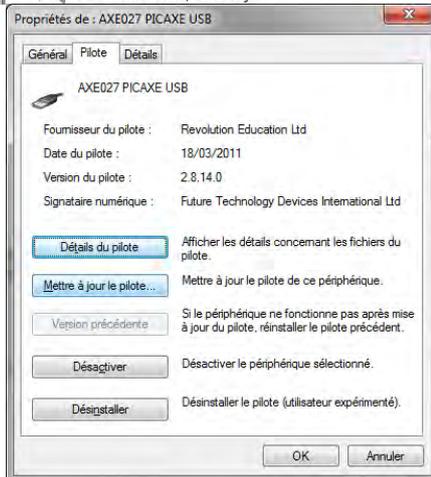
Cliquer sur Device Manager pour accéder directement au gestionnaire de périphériques Windows.

21



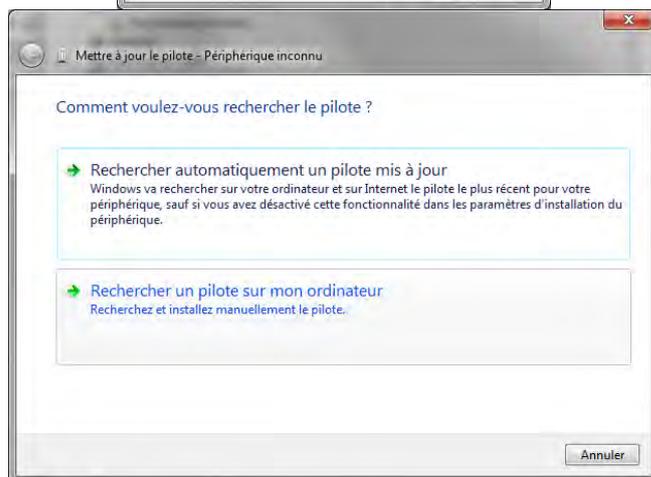
Si le pilote n'est pas correctement installé, un avertissement (triangle jaune) apparaît indiquant que le périphérique USB qui connecté n'est pas opérationnel.

22



Cliquer sur l'avertissement avec le bouton droit de la souris, sélectionner « Propriétés ». Dans l'onglet Pilote cliquer sur « Mettre à jour le pilote... ».

23



À ce stade votre machine peut éventuellement démarrer un assistant d'installation de périphérique USB ; dans ce cas veuillez poursuivre la procédure en vous reportant au point 7 de cette même procédure.

Si l'assistant de mise à jour de pilote n'apparaît pas automatiquement poursuivre la procédure au point suivant.

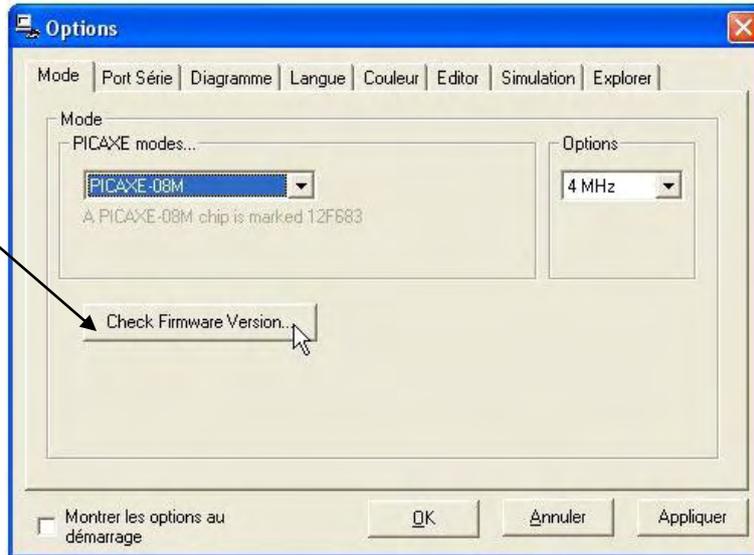
Deux points importants à noter :

- toujours connecter le câble de programmation avant l'application « Picaxe Programming Editor » ;
- toujours connecter le câble de programmation sur le même port USB de l'ordinateur.

5.3 Vérification du Firmware

Il est possible de vérifier l'établissement du dialogue entre le logiciel « **Picaxe Programming Editor** » et la cible Picaxe. Connecter un câble de liaison entre le PC et la cible Picaxe et vérifier la sélection du port COM (Options Port Série). Cliquer sur « **Check Firmware Version** », en fonction de votre microcontrôleur Picaxe, la boîte de dialogue Firmware indique le matériel connecté ainsi que la version du microprogramme :

Cette commande permet aussi de tester le bon fonctionnement du câble de communication.

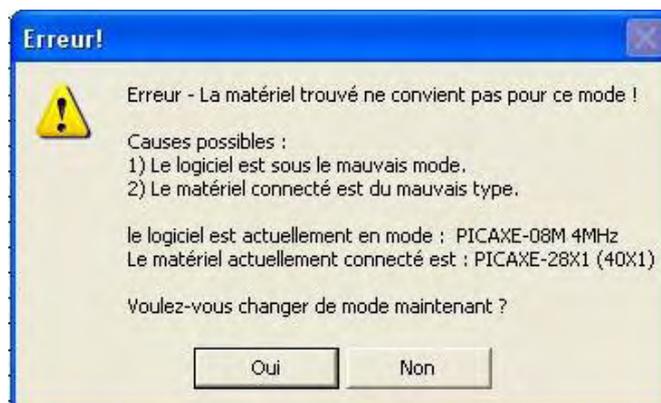


Il est possible de connaître le type de microcontrôleur Picaxe en fonction de la référence indiquée sur le composant.



Exemple :

Si le matériel connecté ne correspond pas au microcontrôleur Picaxe sélectionné, le téléchargement ne peut pas s'effectuer. Une boîte de dialogue « **Erreur !** » indique qu'il y a un problème et propose de changer de mode :



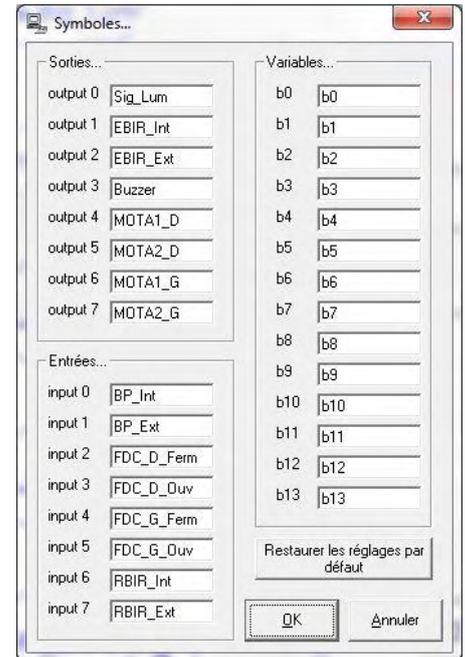
5.4 Utilisation et syntaxe de la table des symboles

Table des symboles

La table des symboles permet d'éclaircir un programme (voir exemple ci-contre).

Son utilisation est soumise aux règles suivantes :

- le nom des entrées ou sorties peut utiliser au maximum 10 caractères à partir du moment où ce n'est pas un mot réservé (par exemple low, high, return, input, etc.) ;
- il n'y a pas de différence entre les majuscules et les minuscules ;
- les caractères spéciaux (&, #, -, etc.) sont refusés et les noms ne doivent pas commencer par un chiffre.



Vérification de la syntaxe

En mode diagramme, la vérification ne peut pas être effectuée. Cependant, lors du transfert du programme une erreur de syntaxe peut se déclarer (par exemple la commande infrain est refusée sur un Picaxe 28X1), l'indication suivante apparaît :

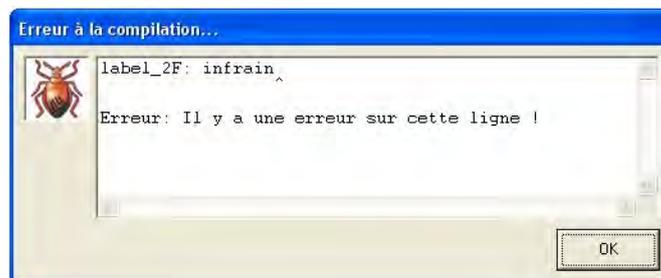


Malheureusement cette indication « Erreur à la compilation » ne nous indique pas où se situe le problème.



Dans ce cas, il est judicieux de transformer le programme en BASIC :

Même sans connaissance du langage BASIC, l'indication plus complète du mode BASIC nous renseigne :



Cette indication indique un problème sur la commande « infrain ».

Avec un peu d'expérience et cette méthode, la recherche d'un problème se résout assez rapidement.

5.5 Utilisation du mode simulation

Le mode simulation permet de tester le fonctionnement d'un programme sans pour autant télécharger le programme dans un Picaxe. Ce mode permet de visualiser l'état des sorties et les différentes variables du programme, il permet aussi de faire évoluer les entrées numériques et analogiques du microcontrôleur Picaxe.

L'affichage LCD, la télécommande Infrarouge et la mesure de distance peuvent aussi être simulés (voir page 22, onglet Simulation, la simulation de la télécommande fonctionne avec la commande « infra », elle n'est pas active avec la commande « irin »). Ce mode ralentit considérablement le programme ce qui permet de voir évoluer sous ses yeux le programme réalisé. L'avancement du programme est visualisable, la commande en cours devient rouge.



Son mode d'utilisation est très simple, après l'écriture du programme, cliquer sur l'icône **Simulate** les fenêtres de simulation s'adaptent au type de Picaxe sélectionné :

Exemple de simulation sur microcontrôleur un Picaxe 08 pour tester le changement d'état d'une sortie :

The diagram illustrates the simulation process. On the left is a flowchart: a 'start' oval leads to a 'high 0' parallelogram, followed by a 'wait 1' rectangle, then a 'low 0' parallelogram, and finally another 'wait 1' rectangle. In the center is a 'Simulation' window showing a virtual PICAXE chip with RXD at 0, A1 at 0, and a 'Generic' input at 0. A callout bubble points to a '>>' button. On the right is another 'Simulation' window showing the same chip but with RXD at 1. Below the chip is a table of variables:

| | | |
|---------------|---|-----------|
| b0 | 0 | %00000000 |
| b1 | 0 | %00000000 |
| b2 | 0 | %00000000 |
| b3 | 0 | %00000000 |
| b4 | 0 | %00000000 |
| b5 | 0 | %00000000 |
| b6 | 0 | %00000000 |
| b7 | 0 | %00000000 |
| b8 | 0 | %00000000 |
| b9 | 0 | %00000000 |
| b10 | 0 | %00000000 |
| b11 | 0 | %00000000 |
| b12 | 0 | %00000000 |
| b13 | 0 | %00000000 |
| Byte Word Hex | | |
| pins | 0 | %00000000 |
| dirs | 1 | %00000001 |
| infra | 0 | %00000000 |

Visualisation des variables

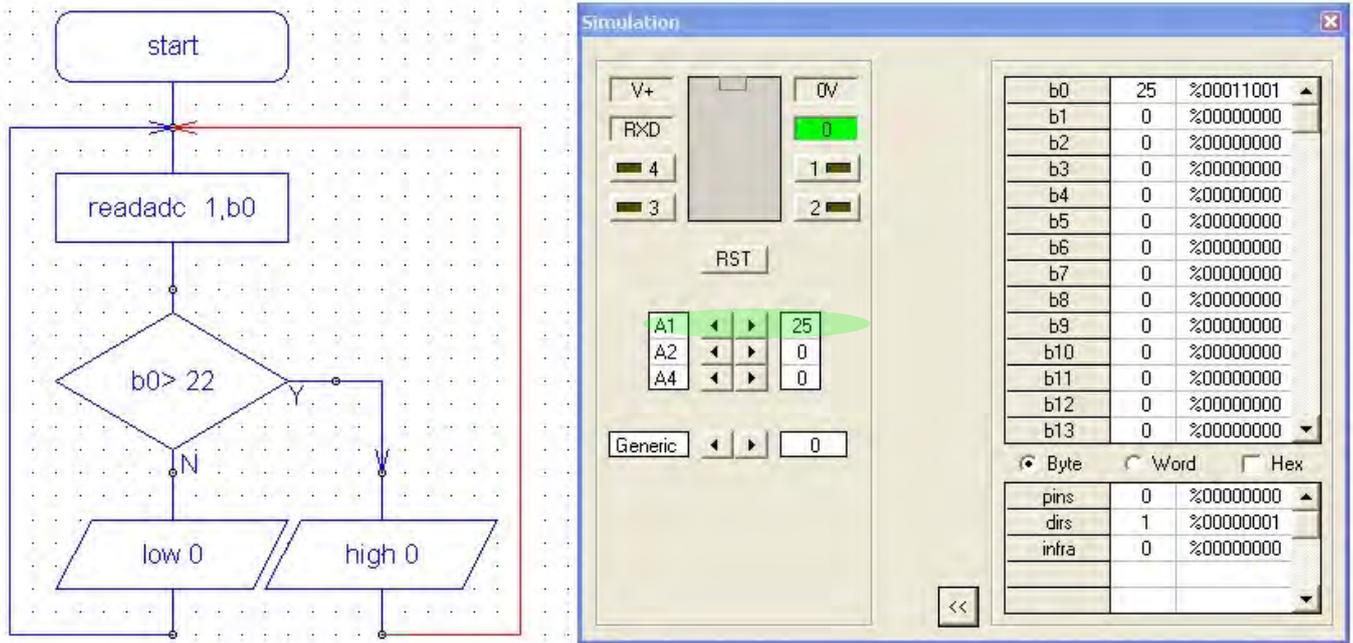
L'état haut d'une sortie (niveau 1) est indiqué par l'allumage de la sortie correspondante (exemple pour la sortie 0).

Exemple de simulation sur un microcontrôleur Picaxe 08 pour tester l'évolution d'une variable :

The diagram illustrates the simulation of a variable. On the left is a flowchart: a 'start' oval leads to a 'let b0=0' rectangle, then a decision diamond 'b0=60'. The 'Y' path loops back to 'let b0=0', and the 'N' path leads to 'let b0=b0+1', which loops back to the decision diamond. On the right is a 'Simulation' window showing the virtual PICAXE chip with RXD at 0, A1 at 0, and a 'Generic' input at 0. The table of variables shows:

| | | |
|---------------|----|-----------|
| b0 | 14 | %00001110 |
| b1 | 0 | %00000000 |
| b2 | 0 | %00000000 |
| b3 | 0 | %00000000 |
| b4 | 0 | %00000000 |
| b5 | 0 | %00000000 |
| b6 | 0 | %00000000 |
| b7 | 0 | %00000000 |
| b8 | 0 | %00000000 |
| b9 | 0 | %00000000 |
| b10 | 0 | %00000000 |
| b11 | 0 | %00000000 |
| b12 | 0 | %00000000 |
| b13 | 0 | %00000000 |
| Byte Word Hex | | |
| pins | 0 | %00000000 |
| dirs | 0 | %00000000 |
| infra | 0 | %00000000 |

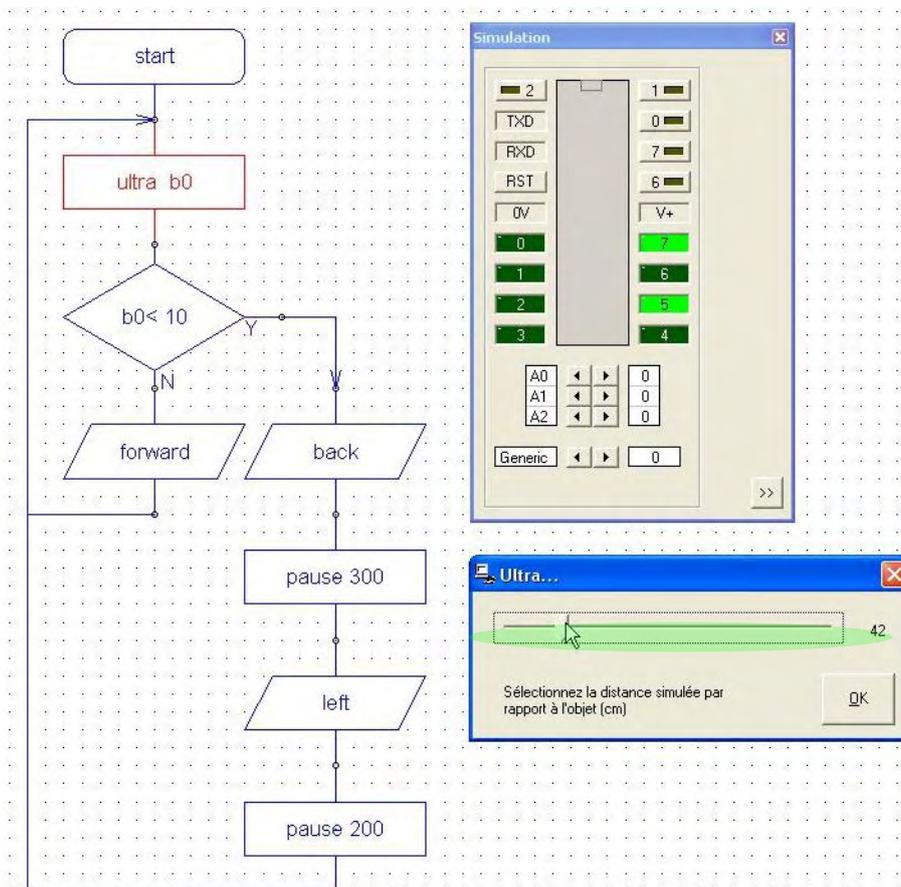
Exemple de simulation sur un microcontrôleur Picaxe 08M pour tester le changement d'état d'une sortie à l'aide d'une entrée analogique :



Evolution de l'entrée analogique A1, visible sur b0

En fonction de la valeur b0 de l'entrée analogique

Exemple de simulation sur un microcontrôleur Picaxe 18X pour tester le mouvement du robot en fonction de la distance :

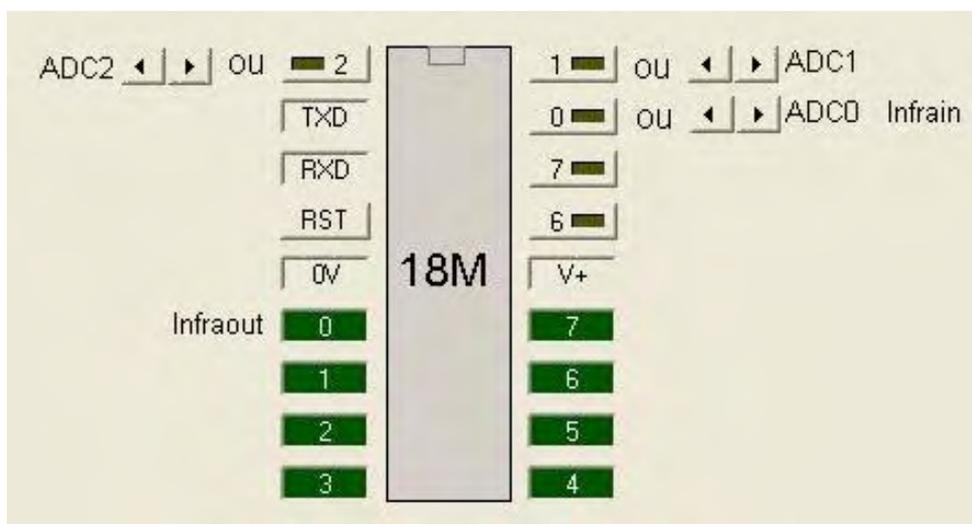
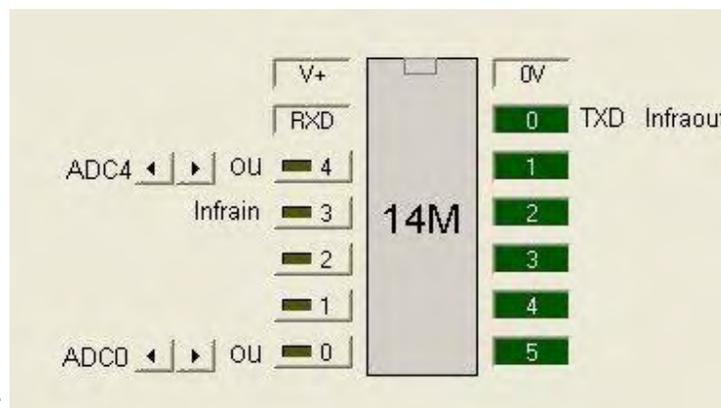
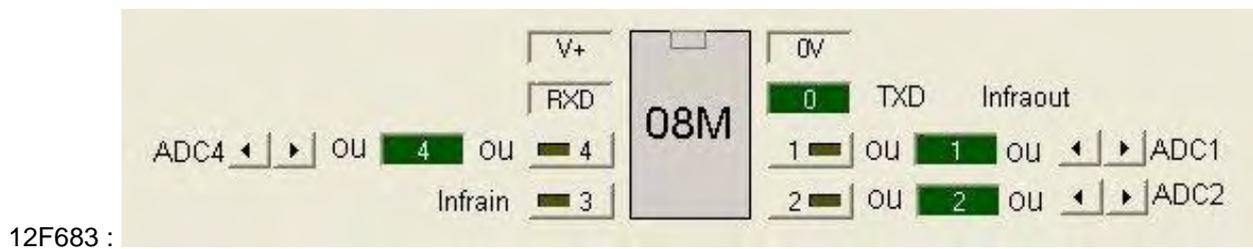
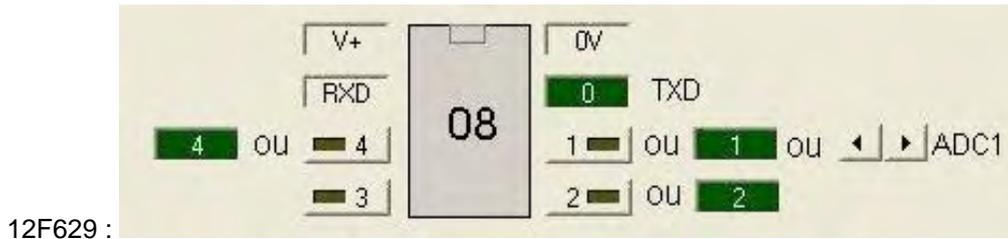


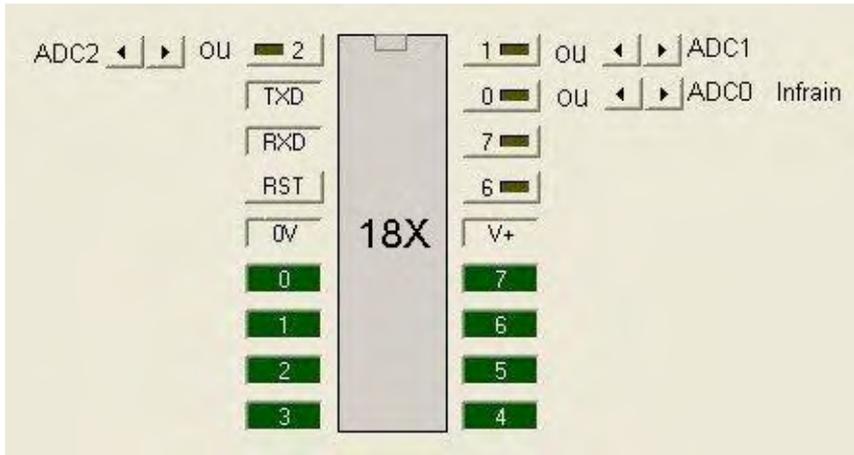
La sortie du mode simulation s'effectue en appuyant sur la touche de fonction du clavier **F4**.

5.6 Représentation des microcontrôleurs sous Picaxe Programming Editor

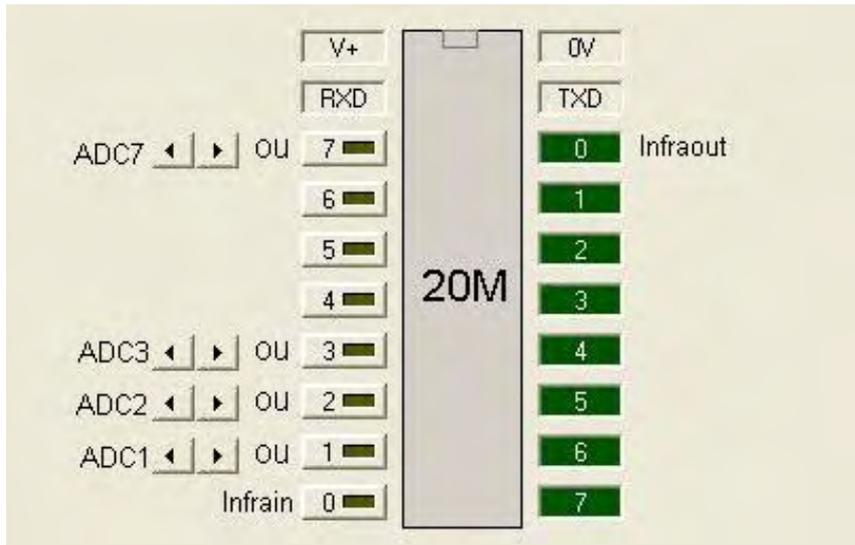


Légende :

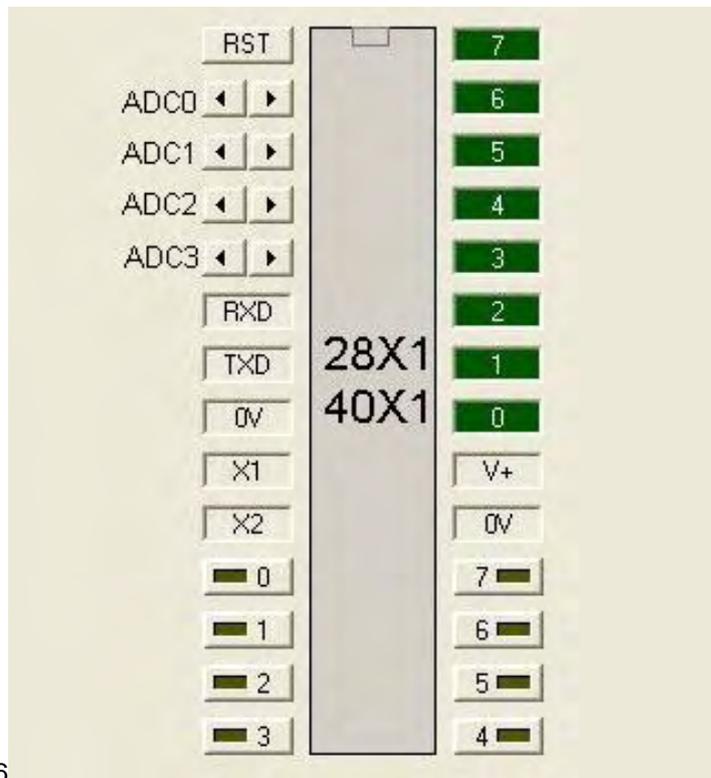




16F88 :



16F677 :



28X1 : 16F886

40X1 : 16F887

5.7 Lexique

| | |
|--------------|---|
| high | Etat haut |
| low | |
| forward | Avancer |
| back | Reculer |
| left | Aller à gauche |
| right | Aller à droite |
| halt | Arrêter |
| pin | Broche (ou patte) d'un circuit intégré |
| toggle | Inverser |
| ADC | Indique une entrée analogique, la valeur analogique évolue de 0 à 5V, la variable obtenue évolue entre 0 et 255 |
| Reset | Remise à zéro du circuit |
| PWM | Modulation de la largeur d'impulsion, permet de contrôler la vitesse de rotation d'un moteur à courant continu |
| Interruption | Prise en compte immédiate d'un évènement afin d'exécuter un traitement spécifique |

5.8 Forum Picaxe francophone



Le forum officiel PICAXE francophone permet aux utilisateurs de communiquer autour du système Picaxe et de trouver ou obtenir des réponses à leurs problèmes (questions/réponses).

Le forum PICAXE francophone est visible à l'adresse suivante :

<http://www.picaxeforum.co.uk/forumdisplay.php?44-Le-forum-officiel-PICAXE-francophone>

QUESTIONS/REPONSES - Voir forum Picaxe à l'adresse suivante :

<http://www.picaxeforum.co.uk/forumdisplay.php?44-Le-forum-officiel-PICAXE-francophone>

Exemple de questions-réponses courantes :

Quelle est l'intensité maxi délivrée par la sortie d'un microcontrôleur Picaxe ?

- Le courant maxi délivré par sortie est d'environ 20 mA, une led en série avec une résistance de 330 Ω consomme environ 15mA peut donc être utilisée. Un courant trop important provoque la destruction de l'étage de sortie du microcontrôleur Picaxe.

Comment augmenter l'intensité maxi délivrée par la sortie d'un microcontrôleur Picaxe ?

- Il est nécessaire de disposer d'un étage tampon. L'utilisation d'un transistor ou d'un circuit intégré est indispensable.

Comment expliquer qu'il n'y a aucune tension en sortie du ULN2803 ?

- Le circuit ULN2803 permet de fermer le circuit à la masse, il n'y a donc aucune tension. Le test d'une sortie d'un circuit ULN 2803 (ou 2003) doit s'effectuer en reliant une borne de l'appareil de mesure sur le +5V.

Le récepteur infrarouge ne fonctionne pas avec la télécommande infrarouge ?

- Il faut programmer le mode de fonctionnement de la télécommande.

Afin de faire évoluer ce dossier nous vous invitons à nous faire part de vos remarques éventuelles sur www.a4.fr à l'aide du formulaire de contact (rubrique « Contact »).

Ce dossier est susceptible d'évoluer ; nous vous invitons à consulter les mises à jour éventuelles disponibles sur www.a4.fr

