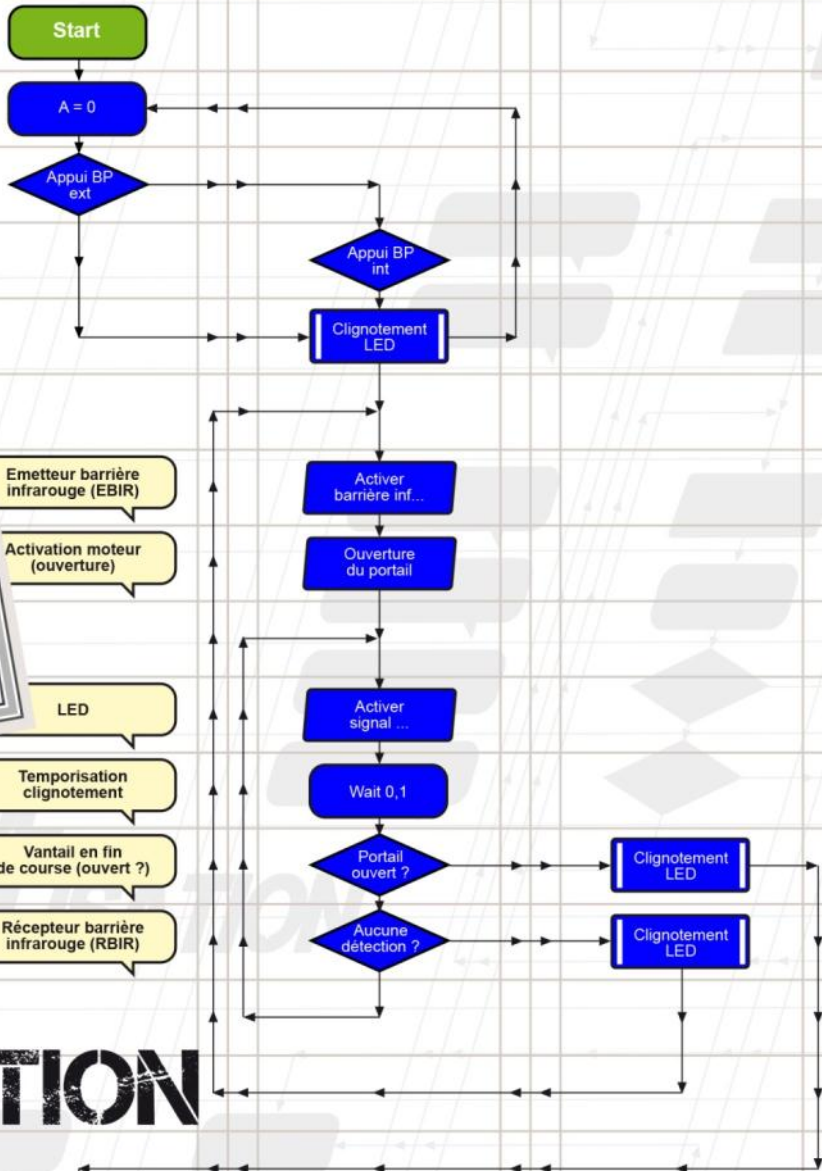
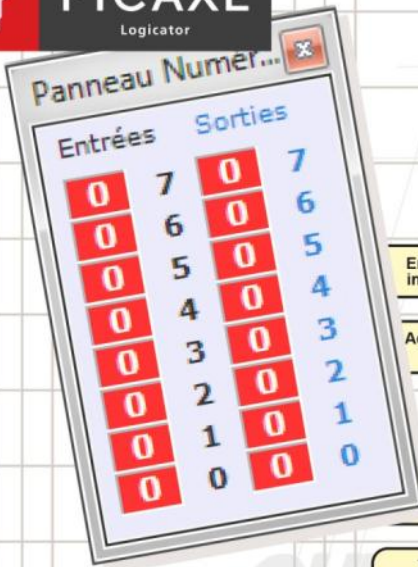
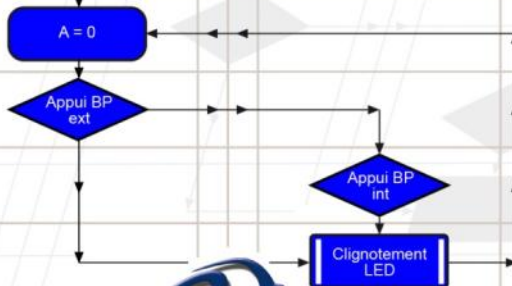


# PICAXE LOGICATOR



## GUIDE D'UTILISATION

OUVERTURE DU PORTAIL







## Guide d'utilisation

Thierry Lancelot - André Bernot - Denis Hoffschir



### Édité par la Société a4 Technologie

5 Avenue de l'Atlantique - Z.A. de Courtabœuf - 91940 Les Ulis  
Tél. : 01 64 86 41 00 - Fax. : 01 64 46 31 19 - [www.a4.fr](http://www.a4.fr)

#### **PICAXE® et Logicator**

PICAXE® et Logicator sont des marques déposées de la Société **Revolution Education** basée au Royaume Uni.

Le logiciel Logicator est protégé par les lois du copyright international.

#### **Ressources numériques**

L'ensemble des ressources numériques relatives à ce guide est téléchargeable librement et gratuitement sur notre site [www.a4.fr](http://www.a4.fr).

Elles sont également disponibles sur CD-ROM (réf. **CD-LG**).

Elles comprennent :

- le guide d'utilisation Logicator en français ;
- les programmes (au format .PLF) ;
- les schémas électroniques Picaxe VSM (au format .DSN) ;
- des photos et des dessins.

#### **Droits de reproduction**

Ce guide et toutes les ressources numériques peuvent être dupliqués pour les élèves, en usage interne à l'établissement scolaire.

La duplication est autorisée sans limite de quantité au sein des établissements scolaires, à seules fins pédagogiques, à la condition que soit cité le nom de l'éditeur : A4 Technologie.

La copie ou la diffusion de tout ou partie du guide ou des ressources numériques par quelque moyen que ce soit, à des fins commerciales n'est pas autorisée sans l'accord de la Société A4 Technologie.

La Société A4 Technologie demeure seule propriétaire de ce document et des ressources numériques.

#### **Crédits photographiques**

Revolution Education et A4 Technologie.

**Remerciements** à *Clive Seager* de la société Revolution Education pour son étroite collaboration dans l'adaptation du logiciel Logicator en version française.



# Table des matières

<b>1. Introduction.....</b>	<b>5</b>
<b>2. Mise en service de Logicator .....</b>	<b>8</b>
<b>2.1. Installer le logiciel et les pilotes .....</b>	<b>8</b>
<b>2.2. Lancer le logiciel .....</b>	<b>10</b>
2.2.1. Sélectionner le port manuellement.....	10
2.2.2. Sélectionner la version Logicator .....	11
2.2.3. Sélectionner la langue .....	11
<b>2.3. Naviguer dans le logiciel .....</b>	<b>12</b>
2.3.1. L'espace de travail .....	12
2.3.2. La barre d'outils .....	12
2.3.3. Les cellules.....	13
2.3.4. Les tracés de liaison.....	13
2.3.5. L'écran de paramétrage d'une instruction .....	14
2.3.6. La barre des commandes.....	14
2.3.7. Le menu contextuel et les raccourcis clavier .....	14
<b>2.4. Elaborer un programme .....</b>	<b>15</b>
2.4.1. Méthodologie pour constituer un programme .....	15
2.4.2. Sélection du type de microcontrôleur PICAXE.....	16
2.4.3. Répartition des entrées et sorties.....	16
2.4.4. Transfert d'un programme.....	17
<b>3. Les instructions.....</b>	<b>18</b>
<b>3.1. Général.....</b>	<b>20</b>
3.1.1. Général > Début .....	21
3.1.2. Général > Fin.....	21
3.1.3. Général > Sorties .....	22
3.1.4. Général > Activer et Désactiver .....	23
3.1.5. Général > Moteurs.....	24
3.1.6. Général > Attendre .....	26
3.1.7. Général > Décision.....	27
3.1.8. Général > Analogue .....	28
3.1.9. Général > Comment.....	29
3.1.10. Général > Code en BASIC .....	30
<b>3.2. Variables .....</b>	<b>31</b>
3.2.1. Variables > Comparer .....	32
3.2.2. Variables > Expression.....	33
3.2.3. Variables > Incrément et Décrément.....	34
3.2.4. Variables > Aléatoire .....	35
3.2.5. Variables > Entrées .....	36
3.2.6. Variables > Sortie .....	37
3.2.7. Variables > Lancer Timer et Arrêter Timer.....	38
3.2.8. Variables > Lire EEPROM.....	39
3.2.9. Variables > Ecrire EEPROM .....	40
<b>3.3. Procédures .....</b>	<b>41</b>
3.3.1. Procédures > Procédure .....	42
3.3.2. Procédure > Aller à Procédure.....	43
3.3.3. Procédure > Appeler Procédure.....	44
3.3.4. Procédures > Config Interrupt et Interrupt.....	45
3.3.5. Procédures > Répéter et Boucler .....	46
<b>3.4. Sorties son.....</b>	<b>48</b>
3.4.1. Sorties son > Son .....	49
3.4.2. Sorties son > Lire l'air.....	50
3.4.3. Sorties son > Lire air utilisateur .....	51
3.4.4. Sorties son > Lire MP3 ou lire iPod.....	52

<b>3.5. E/S séries</b> .....	<b>53</b>
3.5.1. E/S Séries > Lire .....	54
3.5.2. E/S Séries > Entrée Série .....	55
3.5.3. E/S Séries > Envoyer .....	56
3.5.4. E/S Séries > Sortie Série.....	57
3.5.5. Complément d'information sur le code ASCII .....	58
3.5.6. E/S Séries > Débug.....	59
3.5.7. E/S Séries > LCD .....	60
<b>3.6. Autres E/S</b> .....	<b>61</b>
3.6.1. Autres E/S > Tactile.....	62
3.6.2. Autres E/S > Mesurer impulsion.....	63
3.6.3. Autres E/S > Générer impulsion.....	63
3.6.4. Autres E/S > Basculer .....	64
3.6.5. Autres E/S > Servo.....	65
3.6.6. Autres E/S > Moteur Servo .....	66
3.6.7. Autres E/S > Compter .....	67
3.6.8. Autres E/S > Entrée Infrarouge .....	68
3.6.9. Autres E/S > Sortie Infrarouge .....	69
3.6.10. Autres E/S > Lire temp .....	70
3.6.11. Autres E/S > C.A.N.....	71
3.6.12. Autres E/S > Ultrason.....	72
3.6.13. Autres E/S > M.L.I. ....	74
3.6.14. Autres E/S > Sortie M.L.I.....	75
<b>3.7. Avancé</b> .....	<b>76</b>
3.7.1. Avancé > R.A.Z .....	77
3.7.2. Avancé > Sommeil .....	77
3.7.3. Avancé > Suspend et Reprendre .....	78
3.7.4. Avancé > Lire registre .....	79
3.7.5. Avancé > Ecrire registre.....	79
3.7.6. Avancé > Déconnect et Reconnect.....	80
<b>4. Simulation</b> .....	<b>82</b>
<b>4.1. Simulation d'un diagramme</b> .....	<b>83</b>
4.1.1. L'écran de simulation .....	83
4.1.2. Les panneaux de visualiation .....	83
4.1.3. Exemples de simulation .....	84
<b>4.2. Simulation d'un modèle virtuel</b> .....	<b>86</b>
4.2.1. Utiliser un modèle de la bibliothèque de simulation.....	86
4.2.2. Créer un modèle de simulation .....	88
<b>5. Autres fonctionnalités</b> .....	<b>89</b>
<b>5.1. Le langage BASIC</b> .....	<b>89</b>
5.1.1. Conversion en langage BASIC.....	89
5.1.2. Export d'un fichier en BASIC.....	89
5.1.3. Jeu d'instructions BASIC .....	90
<b>5.2. Calibrer un capteur analogique</b> .....	<b>91</b>
<b>6. Dépannage</b> .....	<b>92</b>
<b>6.1. Le câble de programmation AXE027</b> .....	<b>92</b>
6.1.1. Réinstaller le pilote pour câble de programmation USB AXE027 .....	93
<b>6.2. Vérifier le dialogue entre le PC et le microcontrôleur PICAXE</b> .....	<b>97</b>

# 1. Introduction

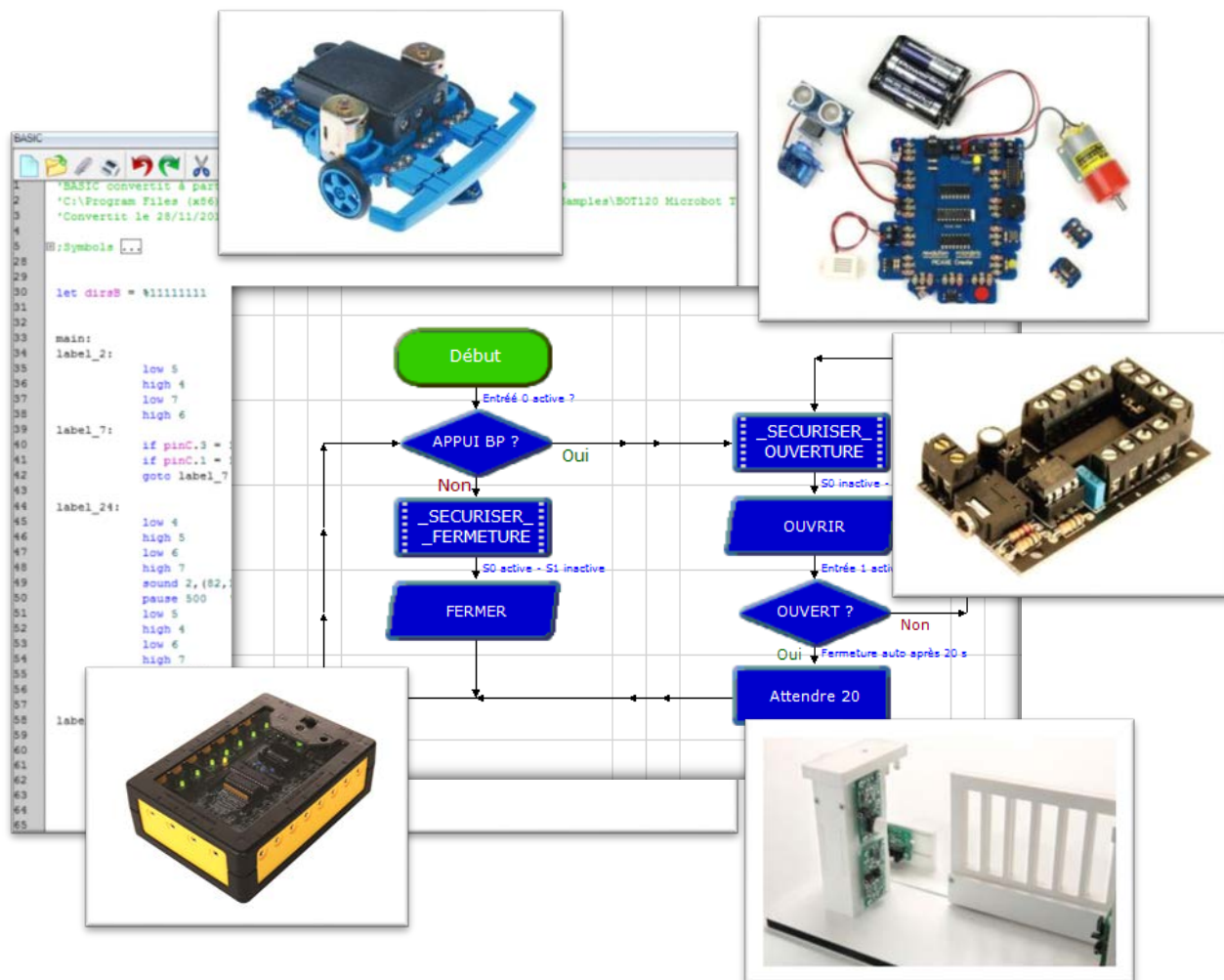
Ce guide d'utilisation est destiné aux utilisateurs de la version française du logiciel Logicator.

Il se divise en plusieurs parties :

- présentation du système Picaxe et de Logicator ;
- prise en main du logiciel ;
- description détaillée de toutes les instructions présentes dans les bibliothèques de Logicator ;
- simulation et dépannage.

## • La technologie PICAXE

Repose sur une famille de microcontrôleurs préconfigurés et programmables directement avec un logiciel de programmation graphique Logicator ou en langage évolué BASIC avec Programming Editor.



La gamme Picaxe est constituée de microcontrôleurs nus, de cartes d'applications et de produits dédiés prêts à l'emploi.

Pour pouvoir s'adapter à différents projets, les microcontrôleurs PICAXE existent sous plusieurs références et se caractérisent par :

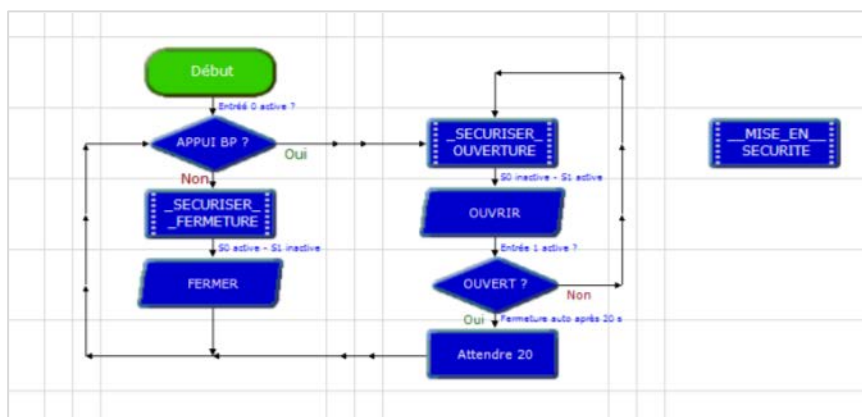
- le nombre d'entrées et de sorties ;
- la tension d'alimentation (5 V ou 3,6 V) ;
- le boîtier, facilement reconnaissable par le type (étroit, large, CMS, etc.) et le nombre de pattes ;
- la capacité mémoire (le nombre de lignes que l'on peut programmer) ;
- les fonctionnalités spécifiques (bus i2c, PWM, infrarouge, conversion analogique/numérique, etc.).

Les microcontrôleurs PICAXE font l'objet d'une documentation détaillée « *Getting started* » accessible à partir de l'aide de Logicator.

## • Logiciel de programmation graphique

Logicator est un logiciel de programmation graphique destiné à programmer des microcontrôleurs PICAXE. Il est développé et mis à jour depuis plus de 20 ans et est particulièrement adapté à l'enseignement technologique en France.

L'intégralité des menus et des instructions est disponible en français ; il est possible à tout moment de basculer dans le mode original en anglais.



L'application est téléchargeable à partir du site de A4 Technologie [www.a4.fr](http://www.a4.fr) ou à partir du site de l'éditeur [www.PICAXE.com](http://www.PICAXE.com).

• **Compatibilité Windows** : XP, Vista, 7 et 8 (32 and 64 bit).

• **Les 3 versions de Logicator**

Une fois installé, trois versions sont proposées :

– **LOGICATOR INITIAL**

Version **gratuite** de Logicator®.

Donne accès à une sélection d'instructions de base suffisantes pour réaliser des programmes simples. Logicator INITIAL peut ouvrir et transférer dans un microcontrôleur PICAXE tout fichier programme réalisé avec la version complète.

– **LOGICATOR VERSION COMPLÈTE**

Donne accès à l'intégralité des fonctions de programmation.

Elle est proposée en version établissement pour une utilisation sur un nombre illimité de postes.

– **LOGICATOR ÉTUDIANT**

Donne accès à l'intégralité des fonctions de programmation.

Elle est réservée à l'usage des étudiants dans le cadre de leurs études et n'est pas autorisée en installation sur les postes d'un établissement.

Le mode ÉTUDIANT est activé à partir de la version Logicator INITIAL en cochant la case « Version shareware non enregistrée » à partir du menu **Aide > Enregistrer**.

**Note** : une clé d'activation est nécessaire pour passer du mode INITIAL au mode VERSION COMPLÈTE et donner l'accès à des instructions évoluées.

• **Développer un diagramme de programmation**

Une approche intuitive de la programmation graphique.

Un simple cliquer/glisser à partir d'une bibliothèque d'instructions permet de bâtir facilement un programme.

Il est possible d'insérer des traitements évolués écrits en langage BASIC.

• **Programmation des microcontrôleurs PICAXE**

– Liaison par câble de programmation AXE027

Logicator détecte automatiquement le câble de programmation AXE027 connecté à l'ordinateur et configure le port COM utilisé pour établir la communication avec le microcontrôleur PICAXE.

– Liaison par modules de transmission sans fil URF/ERF

Réservée à des utilisateurs avertis qui souhaitent mettre en œuvre un mode de programmation sans fil.

Un assistant de configuration est disponible.



Le programme transféré dans le microcontrôleur PICAXE est conservé lorsque l'alimentation est arrêtée.

• **Exécuter des tâches en parallèle**

Logicator permet de gérer l'exécution simultanée de quatre programmes sur la nouvelle génération de microcontrôleurs PICAXE de la série M2.

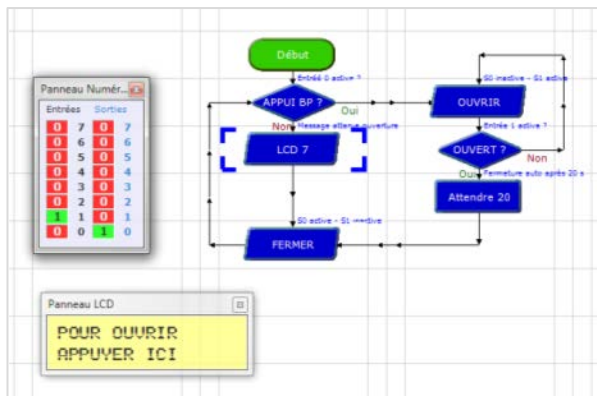


### • Simulation d'un diagramme

Permet de tester le fonctionnement d'un programme sans pour autant télécharger le programme dans un microcontrôleur PICAXE.

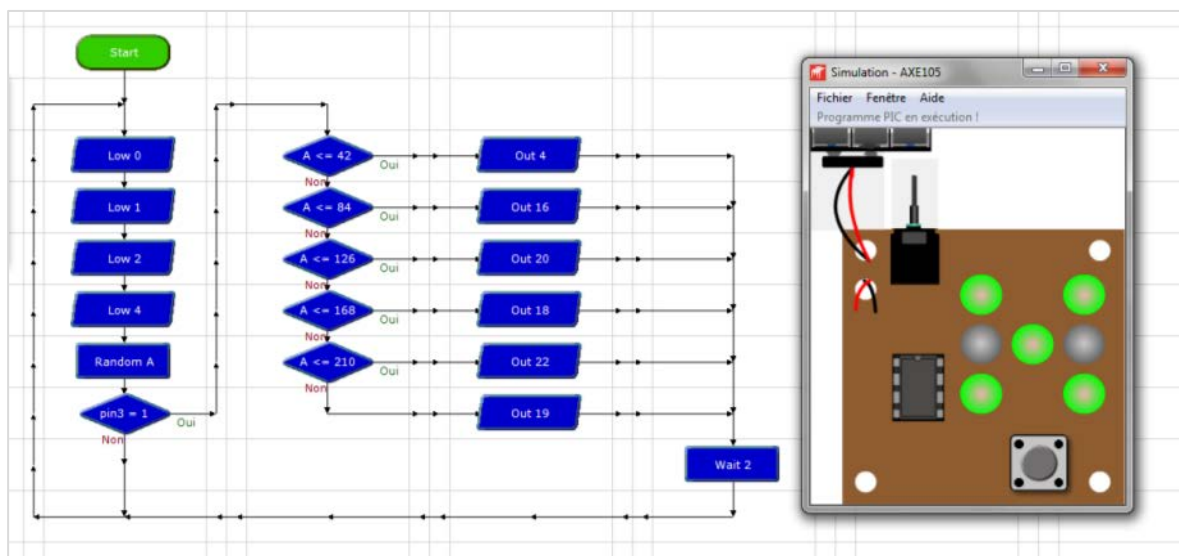
Ce mode permet de visualiser l'état des sorties et les différentes variables du programme, il permet aussi de faire évoluer les entrées numériques et analogiques du microcontrôleur PICAXE.

L'affichage LCD, la télécommande Infrarouge PICAXE et la mesure de distance peuvent aussi être simulés.



### • Simulation sur un modèle virtuel

Logicator donne accès à *Simulation Studio*, logiciel permettant de créer des modèles virtuels de cartes électroniques équipées d'un microcontrôleur PICAXE et de les animer en chargeant les programmes créés avec Logicator.



### • Instructions spécifiques

Logicator propose un certain nombre d'instructions spécifiques dédiées à l'utilisation de modules particuliers (afficheur LCD ou OLED, capteur infrarouge, capteur ultrason, etc.).

### • Autres ressources disponibles

Vous pouvez accéder à la documentation anglaise du logiciel à partir du menu **Aide > Manuels > Picaxe Manuals & Tutorials**.

Trois manuels sont disponibles : « *Getting started* », « *Basic commands* » et « *Interfacing circuits* ».

### • Forum PICAXE francophone

Réelle mine d'informations autour du système PICAXE régulièrement enrichie par un grand nombre d'utilisateurs.

Il est accessible à partir du site [www.a4.fr](http://www.a4.fr) ou sur [www.PICAXEforum.co.uk/forumdisplay.php?44-Le-forum-officiel-PICAXE-francophone](http://www.PICAXEforum.co.uk/forumdisplay.php?44-Le-forum-officiel-PICAXE-francophone)

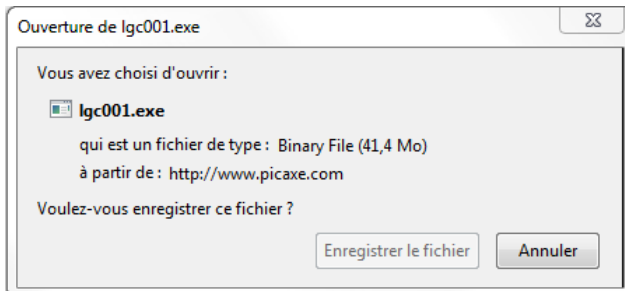
## 2. Mise en service de Logicator

Pour permettre une utilisation optimale du logiciel Logicator, vous devez procéder par étapes :

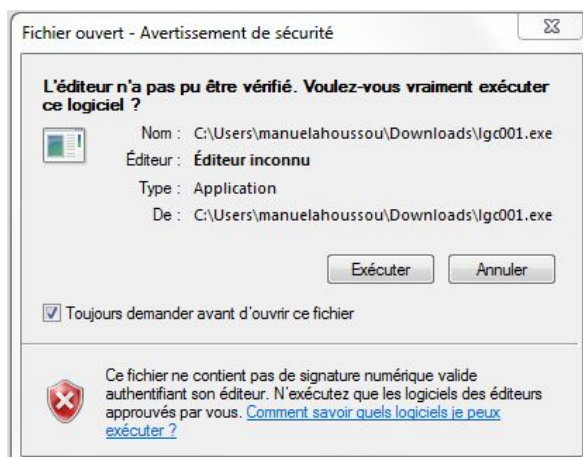
- 1) Installer le logiciel et les différents pilotes.
- 2) Connecter le câble de programmation AXE027 et lancer Logicator.
- 3) Sélectionner le port.
- 4) Vérifier le dialogue avec le microcontrôleur PICAXE.
- 5) Sélectionner le microcontrôleur PICAXE.

### 2.1. Installer le logiciel et les pilotes

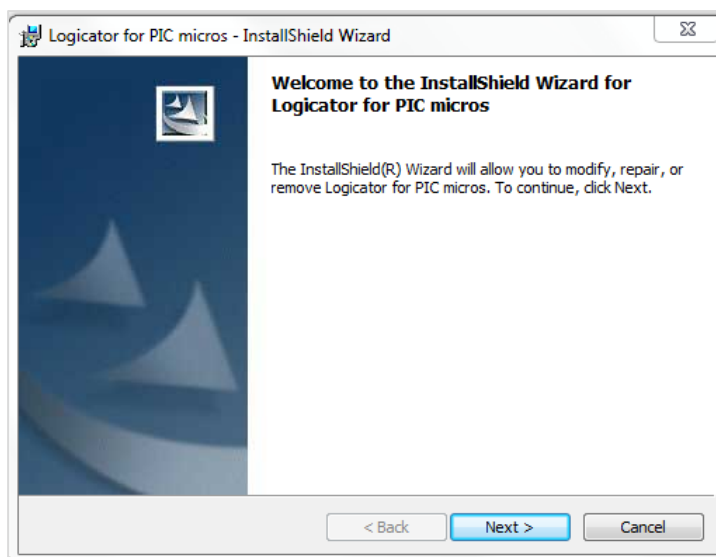
Télécharger l'application à partir du site [www.a4.fr](http://www.a4.fr) (entrer « Logicator » dans le moteur de recherche du site pour localiser le lien de téléchargement).



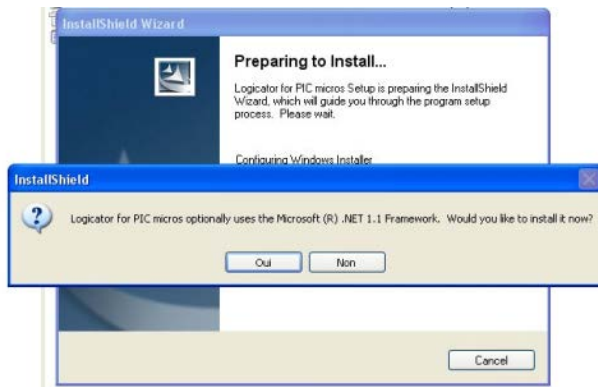
A l'issue du téléchargement, double-cliquer sur le fichier **lgc001.exe** pour lancer l'installation.



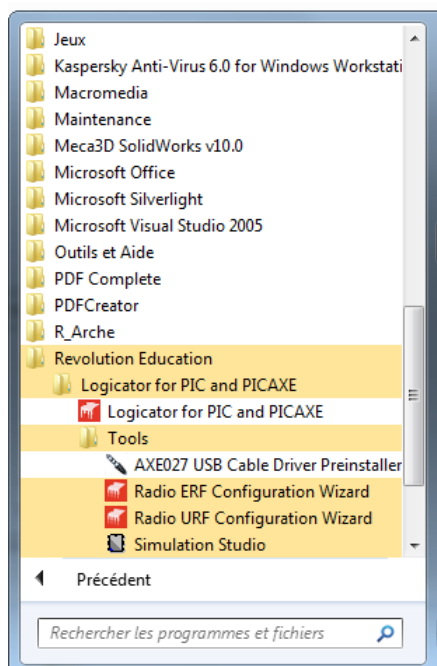
Cliquer sur **Exécuter** pour débiter l'installation du logiciel.



L'installation du logiciel s'effectue en utilisant l'assistant. Vous devez suivre les informations d'installation.



Le système est susceptible de demander l'installation du NET 1.1 Framework, comme l'indique l'écran ci-dessus. Elle n'est utile que pour certaines fonctions de Logicator®, vous n'êtes donc pas obligé de l'installer.



Par défaut, l'installation s'effectue dans le répertoire :

**C:\Program Files\Revolution Education\Logicator for PIC and PICAXE.**

Vous retrouvez les éléments suivants :

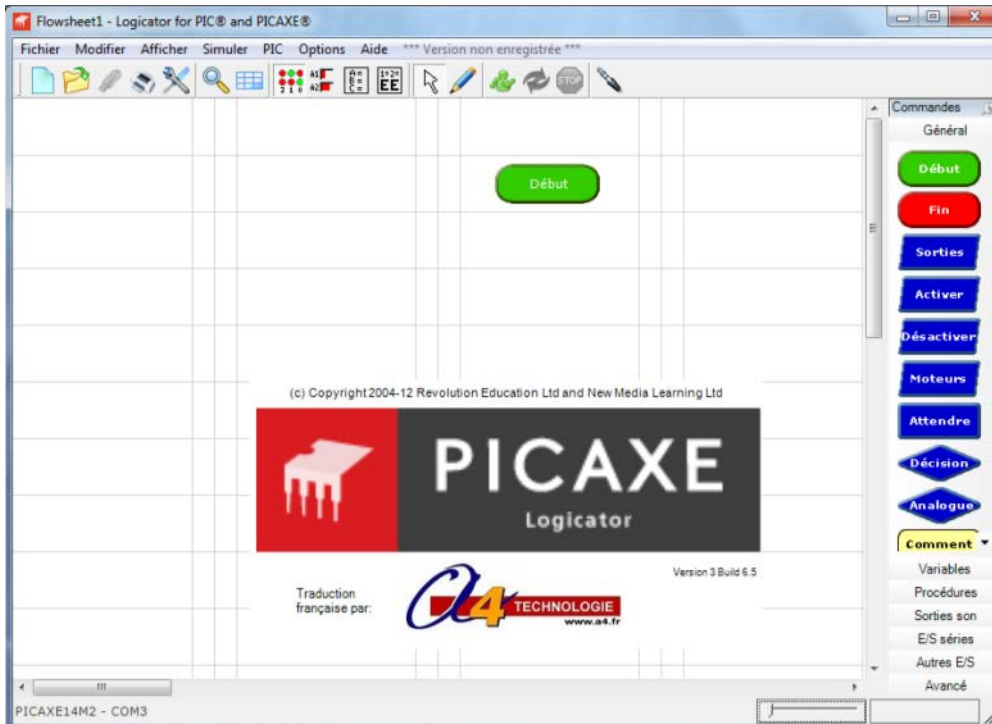
- le logiciel Logicator ;
- le pilote nécessaire au câble de programmation USB AXE027 ;
- les assistants de configuration des modules de transmission sans fil ERF et URF.
- le logiciel *Simulation Studio* pour créer des maquettes virtuelles de cartes PICAXE.

**IMPORTANT !** Avant de lancer Logicator, connecter le câble de programmation AXE027 à l'ordinateur.

## 2.2. Lancer le logiciel

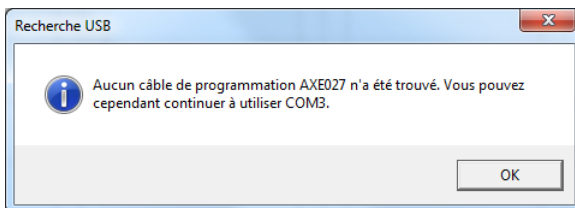


Lancer Logicator à partir du menu **Démarrer > Tous les programmes > Revolution Education > Logicator for PIC and PICAXE** ou en double-cliquant sur l'icône de raccourci sur le bureau.



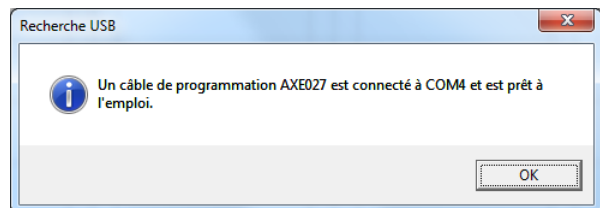
Le système vérifie la connexion du câble de programmation USB AXE027.

### Il ne détecte pas de câble



Vérifier la connexion du câble à l'ordinateur puis cliquer sur **OK**.

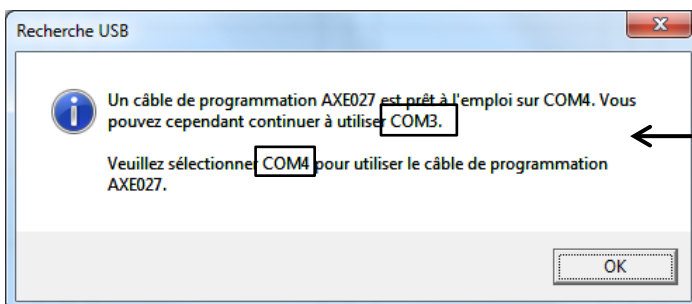
### Il détecte un câble



Le système indique le port COM sur lequel il est connecté. Cliquer sur **OK** pour continuer.

**Note :** *il est recommandé de toujours connecter le câble de programmation AXE027 sur le même port USB, sans quoi il est nécessaire de le sélectionner manuellement à chaque utilisation.*

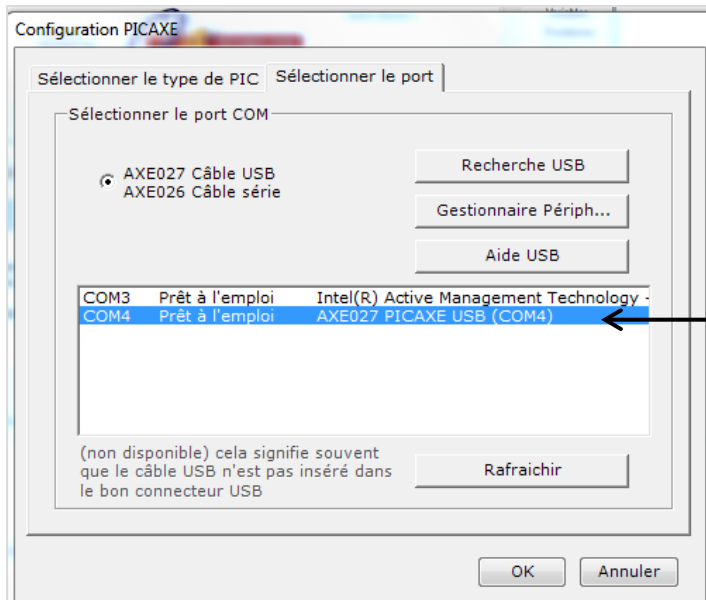
### 2.2.1. Sélectionner le port manuellement



A l'utilisation précédente, le câble était connecté sur le port **COM3**.  
A présent, il est connecté sur le port **COM4**.  
Il est nécessaire de sélectionner le port manuellement.



A partir de la barre des menus, cliquer sur **Options > Sélectionner le type de PIC...** ou utiliser l'icône de raccourci. Choisir ensuite l'onglet **Sélectionner le port**.



Cliquer sur le port COM4 qui mentionne « **Prêt à l'emploi – AXE027 PICAXE USB (COM4)** ».

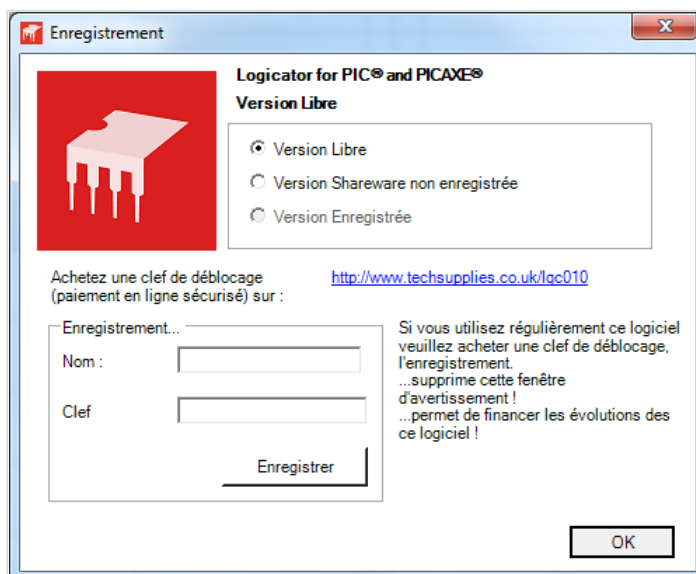
**Recherche USB** Permet d'effectuer une recherche USB.

**Aide USB** Permet d'ouvrir un dossier d'aide (version anglaise).

- Si le système n'attribue aucun port COM au câble de programmation AXE027, cliquer sur **Rafraichir**.
- Si l'absence de détection persiste, consulter en annexe le chapitre « Installation du pilote pour câble de programmation AXE027 ».

**Note** : si vous utilisez un câble de programmation pour port série 9 points, il faut utiliser le port **COM1**.

## 2.2.2. Sélectionner la version Logicator



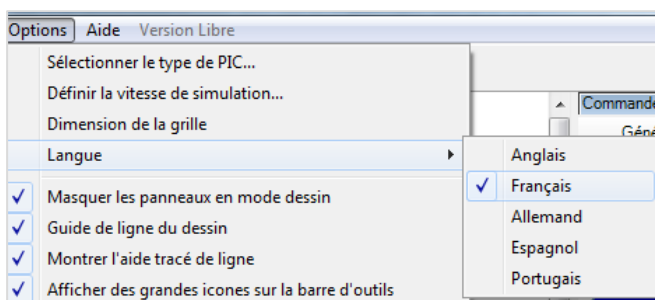
Il existe 3 versions :

- Version Libre = Logicator INITIAL.
- Version Shareware non enregistrée = Logicator ETUDIANT.
- Version Enregistrée = Logicator VERSION COMPLÈTE.

Par défaut, le système propose la version libre.

Cet écran apparaît systématiquement à l'ouverture de Logicator pour les versions gratuites.

## 2.2.3. Sélectionner la langue



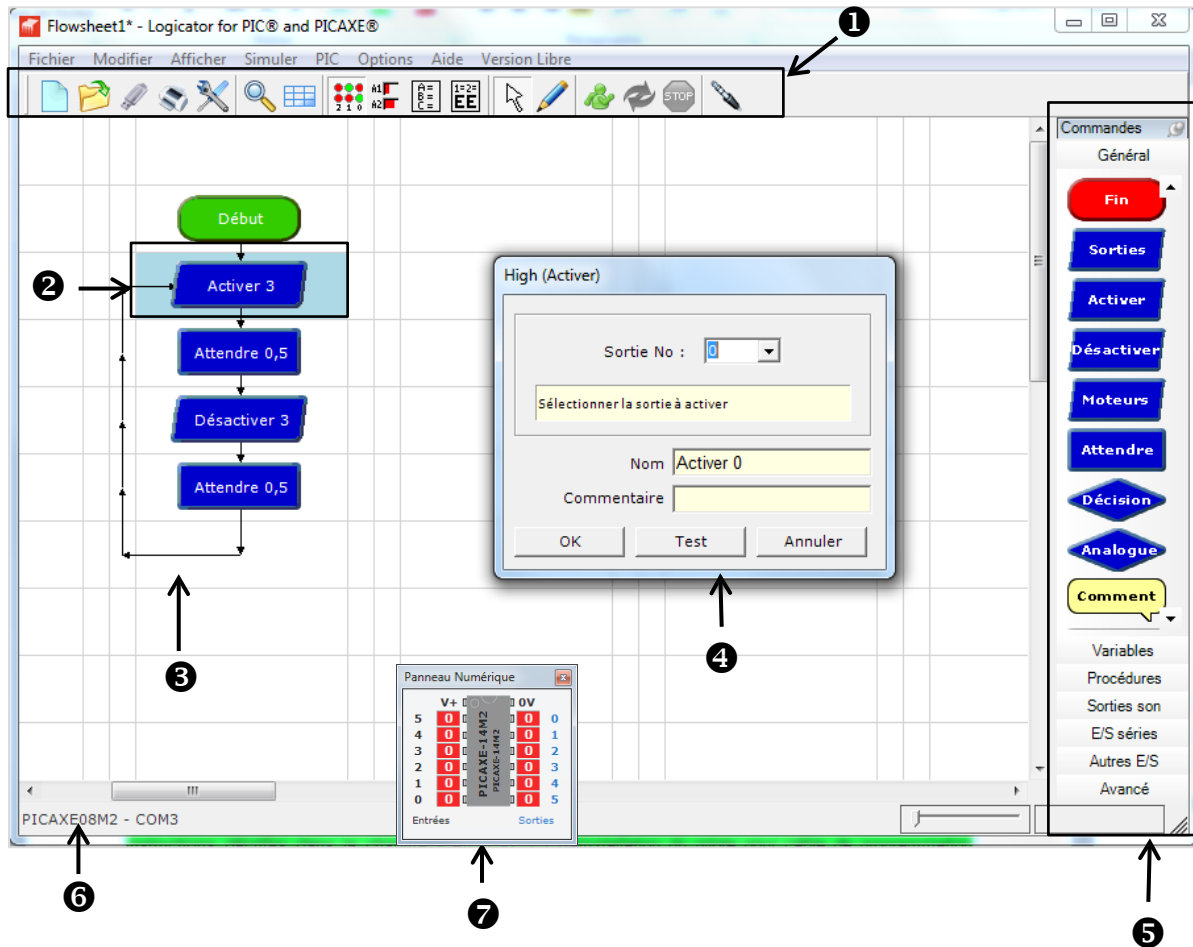
Par défaut, Logicator se lance en version anglaise.

Pour pouvoir travailler sur une version française, à partir de la barre des menus, cliquer sur **Options > Langue > Français**.

## 2.3. Naviguer dans le logiciel

Voici une description détaillée des différents éléments de votre espace de travail : barre d'outils, barre des commandes, cellule, etc.

### 2.3.1. L'espace de travail

















- ❶ Barre d'outils
- ❷ Cellule
- ❸ Flèches de liaison
- ❹ Ecran de paramétrage de l'instruction
- ❺ Barre des commandes
- ❻ Microcontrôleur sélectionné et port utilisé
- ❼ Ecran d'information utilisé pour la simulation.

**Note :** Par défaut, le système affiche le panneau numérique à l'ouverture de Logicator.

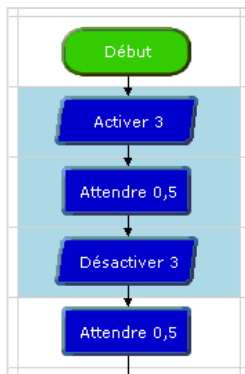
### 2.3.2. La barre d'outils

Regroupent les fonctionnalités principales de Logicator® accessibles par des touches de raccourcis. Voici le détail de toutes les icônes de la barre d'outils.

	<b>Créer un nouveau diagramme</b> (Ctrl + N) Permet d'écrire une nouvelle feuille.
	<b>Ouvrir un diagramme</b> (Ctrl + O) Permet d'ouvrir un fichier réalisé avec Logicator ( <i>extension *.plf</i> ).
	<b>Enregistrer ce diagramme</b> (Ctrl + S) Permet de sauvegarder votre travail.
	<b>Imprimer le diagramme</b> (Ctrl + P) Permet d'imprimer votre zone de travail.

	<b>Sélectionner le type de PIC...</b> Permet d'ouvrir la configuration du PICAXE.
	<b>Zoom</b> Permet de zoomer ou de dé-zoomer la zone de travail.
	<b>Plan de la fenêtre</b> Permet de visualiser le plan de la fenêtre de travail.
	Permet d'afficher ou de masquer : <ul style="list-style-type: none"> <li>• le <b>panneau Numérique</b>, pour visualiser les sorties et agir sur les entrées,</li> <li>• le <b>panneau Analogique</b>, pour simuler une valeur analogique,</li> <li>• le <b>panneau des Variables</b>, pour connaître les valeurs attribuées aux variables,</li> <li>• le <b>panneau EEPROM</b>, pour connaître les valeurs attribuées aux variables EEPROM.</li> </ul> Il est ainsi possible de tester le bon déroulement d'un programme.
	<b>Mode sélection</b> Cliquer sur le bloc et le déplacer en maintenant le bouton gauche de la souris enfoncé.
	<b>Mode tracé de liaison</b> Permet de dessiner des liaisons pour relier les blocs. Cliquer avec le bouton gauche sur le premier bloc puis cliquer sur le dernier bloc pour établir la liaison.
	<b>Exécuter le diagramme (F10)</b> Permet de tester son programme en effectuant la simulation bloc après bloc.
	<b>Réinitialiser le diagramme</b> Permet de réinitialiser la simulation du programme. Cette fonction s'utilise avec le mode simulation.
	<b>Arrêter la simulation du Diagramme (F11)</b> Cette fonction s'utilise avec le mode simulation.
	<b>Programmer le PIC</b> Permet de télécharger dans le microcontrôleur PICAXE le programme. La programmation s'effectue uniquement si la liaison est correcte (câble USB installé, microcontrôleur PICAXE sélectionné et alimenté). Un seul programme par PICAXE.

### 2.3.3. Les cellules



Il existe deux types de cellules : les grandes sont réservées au positionnement des instructions et les plus petites, permettent d'intégrer des flèches de liaison.

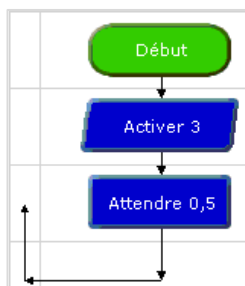
La sélection d'une cellule s'effectue par un simple clic.

Pour sélectionner plusieurs cellules : sélectionner la première cellule par un clic, appuyer sur la touche **Maj**. puis sélectionner la dernière cellule.

Toutes les cellules sélectionnées s'entourent en bleu.

Il est également possible d'appuyer sur la touche **Maj** et d'utiliser les flèches directionnelles.

### 2.3.4. Les tracés de liaison



Pour réaliser des tracés de liaison, cliquer sur l'icône 

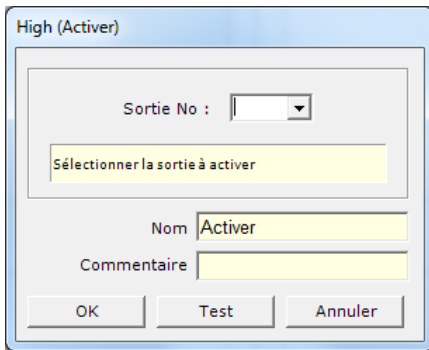
Sélectionner une cellule par un clic, appuyer sur la touche **Ctrl** puis sur les flèches directionnelles. Les liaisons s'effectuent très facilement.

Vous pouvez également faire un clic droit sur la commande et cliquer sur **Départ d'une liaison**.

Pour supprimer un tracé sans supprimer l'instruction, sélectionner la cellule puis cliquer sur **Ctrl + Suppr**.

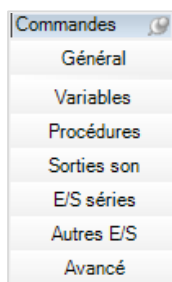
### 2.3.5. L'écran de paramétrage d'une instruction

Il est accessible en double-cliquant sur une instruction.



Les options et les zones accessibles varient en fonction du microcontrôleur sélectionné.

### 2.3.6. La barre des commandes

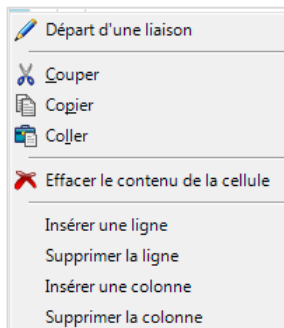


Elle regroupe toutes les instructions de Logicator regroupées en 7 bibliothèques. Les instructions se positionnent dans une cellule de la zone de travail par un simple cliquer/glisser.

**Note :** en fonction de la version, certaines fonctions (grisées) ne sont pas accessibles.  
La version Libre « INITIAL » permet d'ouvrir et d'exécuter des programmes conçus à partir d'une version Enregistrée, qui utilise des instructions grisées.

### 2.3.7. Le menu contextuel et les raccourcis clavier

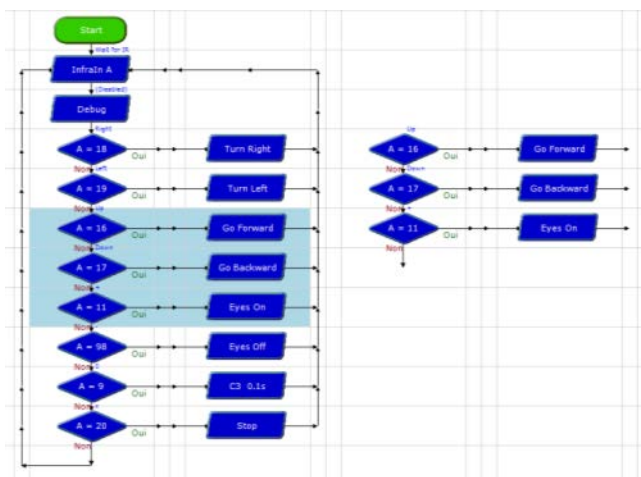
Un clic droit permet d'accéder au menu contextuel ce qui permet de réaliser la plupart des manipulations.



- Comme indiqué précédemment, il est possible de passer en mode tracé de liaison.
- Vous pouvez agir sur une commande : couper, copier, coller.
- Vous pouvez agir sur une cellule : supprimer son contenu.
- Vous pouvez ajouter ou supprimer des lignes et des colonnes sur votre espace de travail.

Par défaut, l'espace de travail est constitué de 22 colonnes et de 25 lignes.

Il existe de nombreux autres raccourcis clavier, la plupart sont rappelés dans les menus du logiciel. Pour gagner du temps il est possible d'utiliser les raccourcis clavier suivants :



- **Ctrl + X** : couper ;
- **Ctrl + C** : copier ;
- **Ctrl + V** : coller.
- **Ctrl + Glisser** : copier/coller
- **Ctrl+N** : nouveau diagramme.

← Exemple d'un copier/coller d'un bloc de cellules.



## 2.4. Elaborer un programme

Pour élaborer un programme, il y a plusieurs prérequis :

- Logicator est correctement installé ;
- le câble de programmation AXE027 est opérationnel ;
- l'utilisateur connaît la carte et le microcontrôleur PICAXE sur lequel il va travailler ;
- il a établi un cahier des charges de son programme compatible avec les ressources offertes par la carte et le microcontrôleur PICAXE ;
- il a identifié les entrées et les sorties utilisées dans son programme.

Il faut ensuite procéder par étapes.

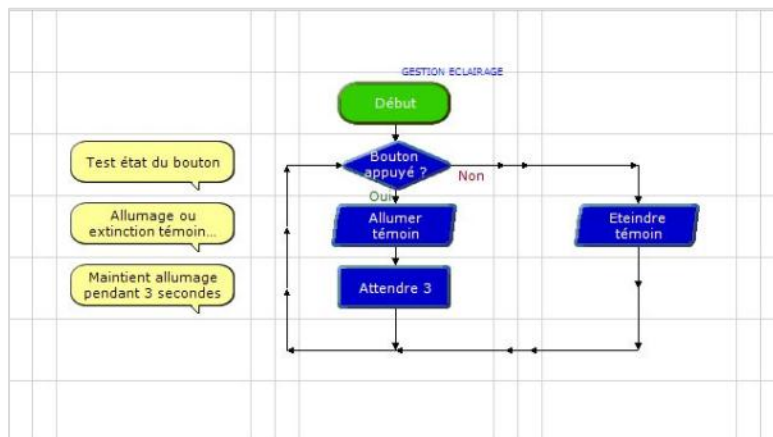
- 1) Sélectionner le microcontrôleur PICAXE (pour la série PICAXE 08 modifier éventuellement la répartition des entrées/sorties).
- 2) Constituer un programme (choix et paramétrage des instructions, flèches de liaison, etc.). Voir chapitre « *Les instructions* ».
- 3) Vérifier la syntaxe et simuler le bon fonctionnement du programme (voir chapitre « *Simulation* »).
- 4) Transférer le programme dans le microcontrôleur PICAXE.

### 2.4.1. Méthodologie pour constituer un programme

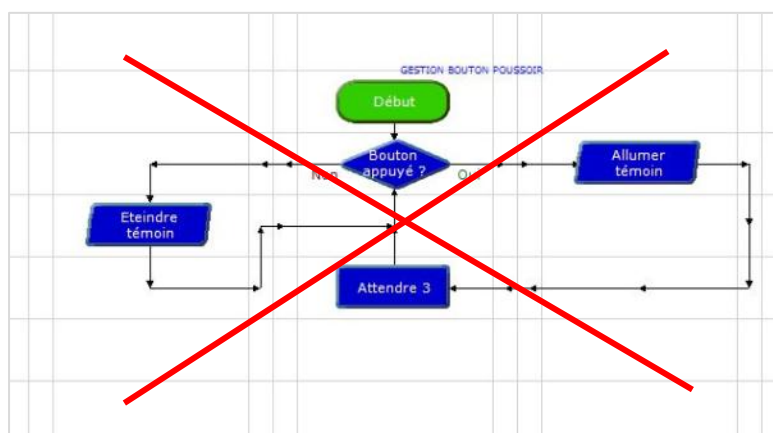
Il est important de respecter certaines règles de présentation qui facilitent la lecture et la compréhension des programmes :

- les instructions sont organisées dans un ordre logique ;
- les tracés de liaison partent d'une instruction vers une autre, de haut en bas ;
- les commentaires apportent des informations complémentaires.

Voici ci-après, deux exemples d'élaboration pour un même programme, qui fonctionne correctement :



- le premier respecte ces règles, il vous permet une vision claire et synthétique du programme.



- le deuxième est plus confus, les tracés de liaison partent dans tous les sens, dans certains cas, de bas en haut.

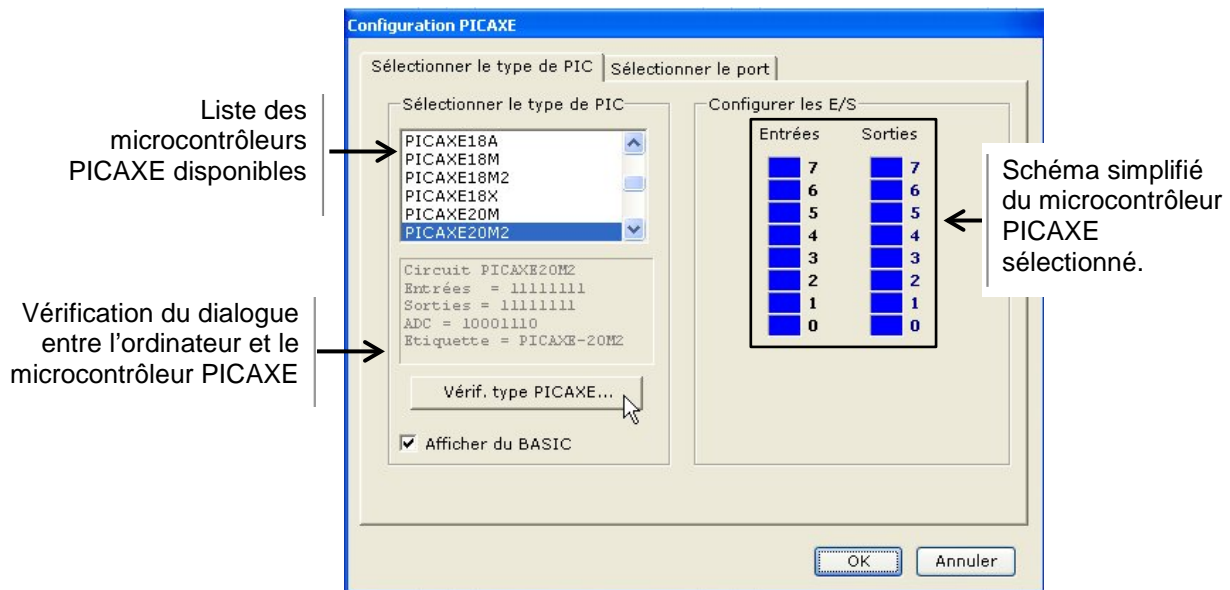
Ce type de présentation est à proscrire.

## 2.4.2. Sélection du type de microcontrôleur PICAXE

Vous devez choisir le microcontrôleur Picaxe que vous allez programmer.



A partir de la barre des menus, cliquer sur **Options > Sélectionner le type de PIC...** ou utiliser l'icône de raccourci.



Le bouton **Vérif. Type PICAXE** permet de :

- déterminer le type de microcontrôleur PICAXE utilisé sur la carte.
- vérifier que le dialogue est possible entre l'ordinateur et le microcontrôleur PICAXE.

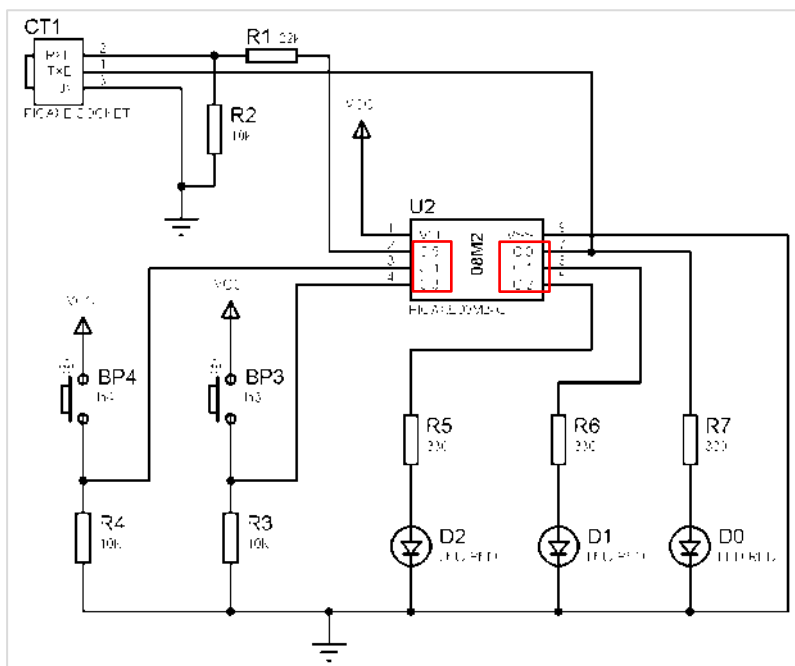
Si le dialogue ne s'établit pas, consulter le chapitre « Dépannage ».

Le schéma simplifié du microcontrôleur PICAXE permet de connaître le nombre d'entrées et de sorties du PICAXE sélectionné.

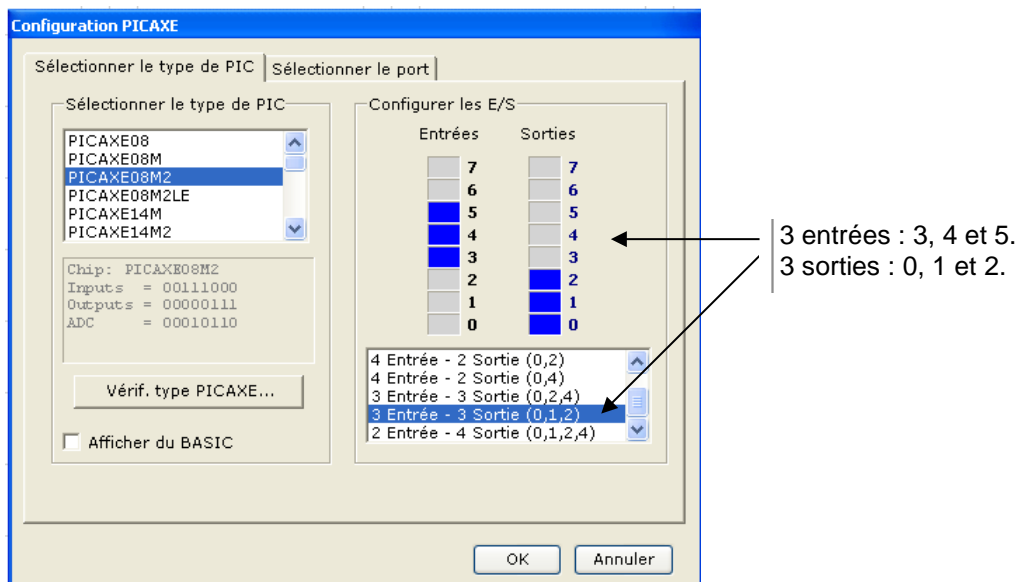
## 2.4.3. Répartition des entrées et sorties

Pour les microcontrôleurs PICAXE de la série 08, il est possible de modifier la répartition des entrées et sorties proposées par défaut par Logicator. Cela permet d'adapter la configuration du microcontrôleur PICAXE au cahier des charges préalablement défini.

Dans le schéma suivant, 2 boutons-poussoirs sont connectés sur les entrées 3 et 4 et 3 LED sur les sorties 0, 1 et 2 d'un microcontrôleur PICAXE 08M2.



Dans Logicator, il faut sélectionner une configuration d'entrées/sorties cohérente avec ce schéma :

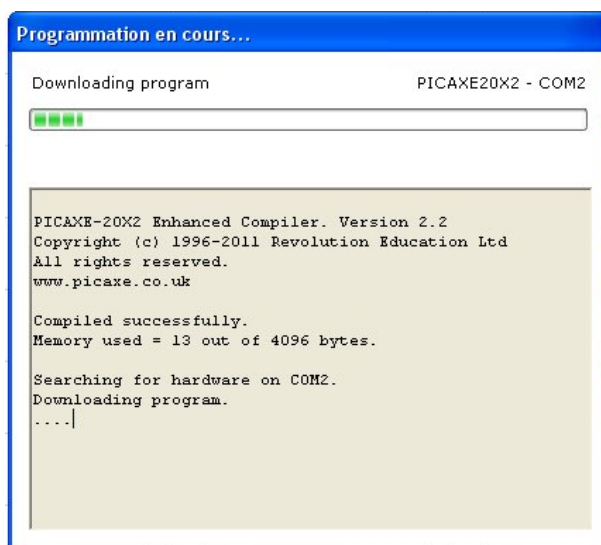


#### 2.4.4. Transfert d'un programme

Une fois le programme élaboré et les vérifications effectuées (dialogue, connexion du câble de programmation AXE027, etc.), vous pouvez transférer le programme dans le microcontrôleur PICAXE.



A partir de la barre des menus, cliquer sur **PIC > Programmer le PIC** ou sur l'icône de raccourci. Si tout est OK, la programmation s'effectue.



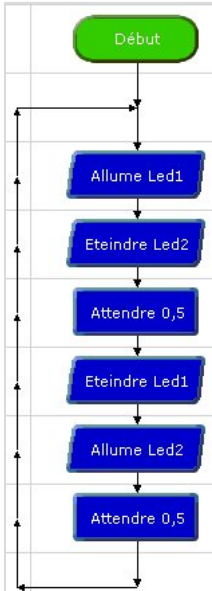
Il arrive parfois que le transfert ne puisse pas aboutir. Dans ce cas, consulter le chapitre « Dépannage » de cette documentation.

### 3. Les instructions

Ce chapitre décrit toutes les instructions proposées dans les différentes bibliothèques de Logicator :

- descriptif de la fonction et des options utilisables ;
- exemple de programme illustrant l'utilisation de l'instruction au format .PLF (téléchargeable sur [www.a4.fr](http://www.a4.fr));
- modèle virtuel de schéma électronique PICAXE VSM correspondant, au format .DSN.

#### Programmation graphique

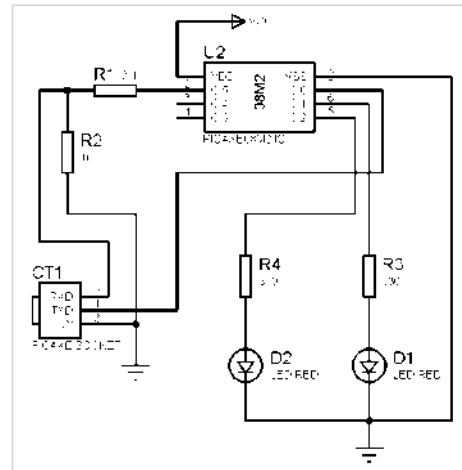


#### Programmation en langage évolué

```

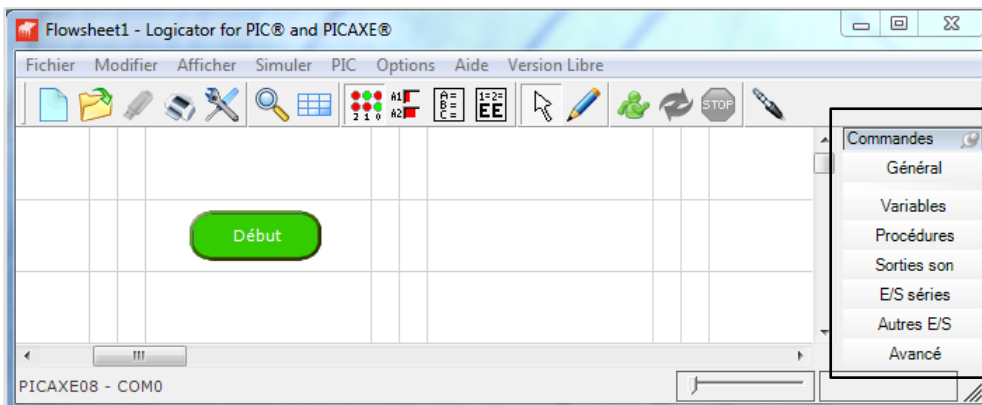
1000
1010 let dirout = %11111111
1020 let adobsetup = %
1030
1040
1050 main:
1060 let var0 = 0 'Expression command
1070 servo 6, var0
1080 servo 5, var0
1090 do until var0 = 1 ' repeat
1100 goto prc_SECTION_CAPTURE 'Do Procedure
1110
1120 prc_SECTION_CAPTURE:
1130 if var0 = 2 then label_172 'Compare command
1140 if var0 = 1 then label_172 'Compare command
1150
1160 label_240:
1170 if var0 < 35 then label_241 'Compare command
1180 goto prc_NOTE08 'Do Procedure
1190 loop 'End of repeat loop
1200
1210 loop0:
1220 goto loop0
1230
1240 prc_NOTE08:
1250 servopos 5,A
1260 pause 30
1270 servopos 6,B
1280 pause 30
1290 return 'Return
1300
1310 label_241: low 4
1320 low 5
1330 goto label_240
1340
1350 label_172: goto prc_COMMANDER_SERPOS 'Do Procedure
1360 goto label_240
1370
1380 prc_COMMANDER_SERPOS:
1390 servos C,0,%2400,5,(10,11,12,13,14,15),var0,var0
1400 servos 2,%2400,5,(184,125,%var0)
1410 servos 2,%2400,5,(194,192,%var0)
1420 return 'Return
1430
1440
  
```

#### Schéma électronique VSM



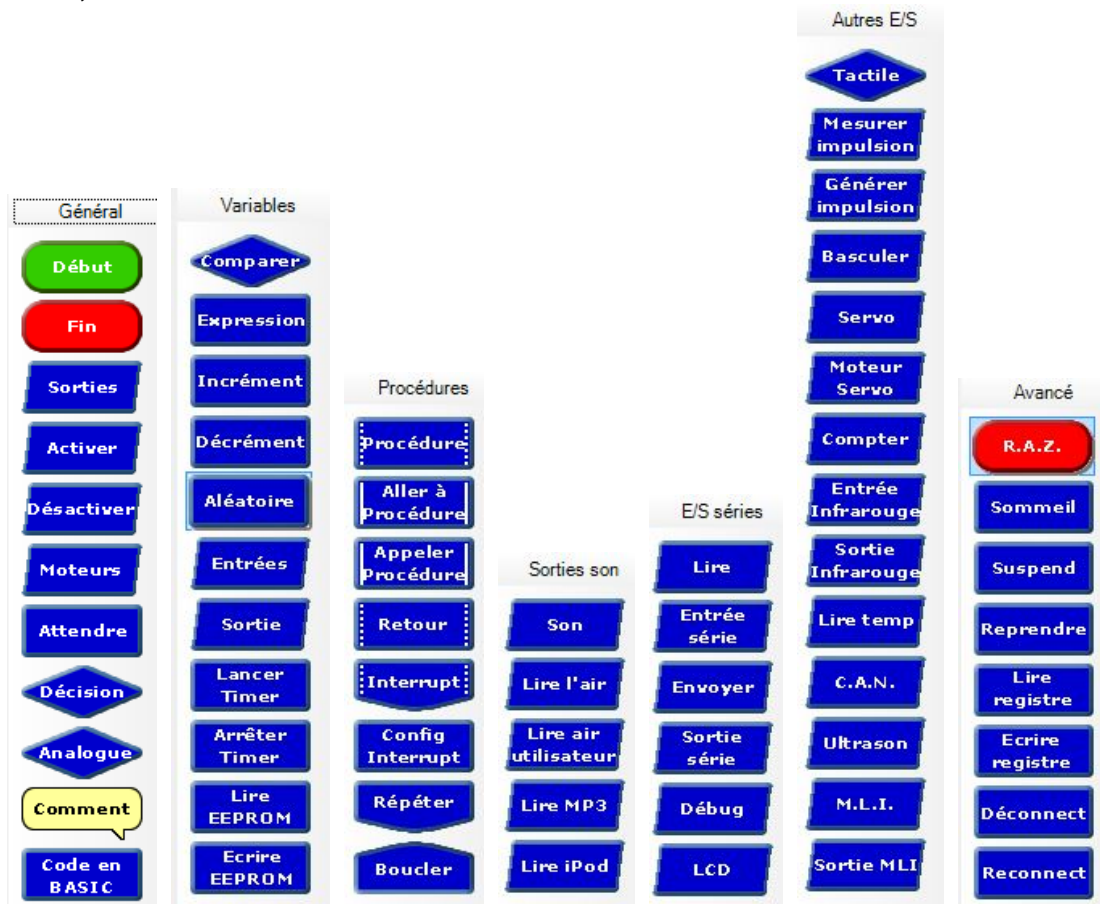
**Note :** *PICAXE VSM (Virtual System Modelling) est un logiciel de simulation de schémas électroniques qui combine les modèles virtuels des microcontrôleurs PICAXE et l'environnement de simulation et d'analyse de circuit SPICE. Il est téléchargeable sur [www.a4.fr](http://www.a4.fr) Il permet de simuler en temps réel le fonctionnement de projets mettant en œuvre le système PICAXE.*

Les instructions sont accessibles à partir de la barre des commandes.



Pour insérer une instruction dans votre programme, la sélectionner avec un clic et la faire glisser dans une cellule.

Les instructions sont regroupées en bibliothèques : **Général, Variables, Procédures, Sorties son, E/S séries, Autres E/S et Avancé.**



**ATTENTION !**

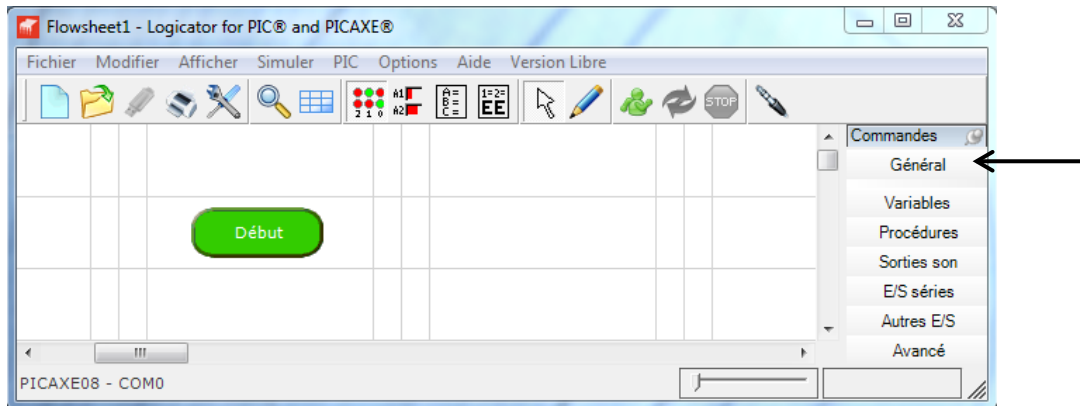
En fonction de la version dont vous disposez (Initial, Etudiant ou Complète) vous n'avez pas accès à toutes les instructions. Certaines sont grisées.

Une fois l'instruction intégrée à votre diagramme, double-cliquer dessus pour appeler l'écran de paramétrage.


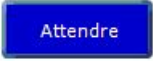

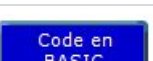
En fonction du microcontrôleur PICAXE sélectionné au départ, les zones affichées et les options sélectionnables peuvent varier.

### 3.1. Général

A partir de la barre des commandes, cliquer sur **Général**.



Ce groupe donne accès aux instructions les plus courantes.

* 	Tout programme doit commencer par une instruction <b>Début</b> .
* 	Indique la fin d'exécution d'un programme.
* 	Permet d'activer ou de désactiver simultanément plusieurs sorties d'un microcontrôleur PICAXE. Cette instruction remplace avantageusement une succession de <b>Activer</b> et de <b>Désactiver</b> .
* 	Permet de positionner une sortie à l'état haut.
* 	Permet de positionner une sortie à l'état bas.
* 	Permet d'agir simultanément sur plusieurs sorties du microcontrôleur PICAXE en vue de piloter deux moteurs à courant continu destinés à animer un robot.
* 	Permet de positionner un temps d'attente exprimé en secondes. (entre <i>0,001et 65 secondes</i> ).
* 	Permet de déclencher une action en fonction de l'état d'une entrée unique ou d'une combinaison de plusieurs entrées numériques du microcontrôleur PICAXE.
* 	Permet de déclencher une action en fonction de la valeur d'une entrée analogique.
* 	Permet d'écrire un commentaire, de donner, par exemple, des indications sur le fonctionnement du programme.
* 	Permet d'introduire un traitement évolué écrit en langage BASIC.

\* Instructions disponibles dans la version Logicator INITIAL.

### 3.1.1. Général > Début

Début 1

**Fonction :** Tout programme doit commencer par une instruction **Début**.



Certains microcontrôleurs PICAXE permettent l'exécution en parallèle de plusieurs programmes.

Il est possible de positionner plusieurs instructions **Début** dans un même programme.

### 3.1.2. Général > Fin

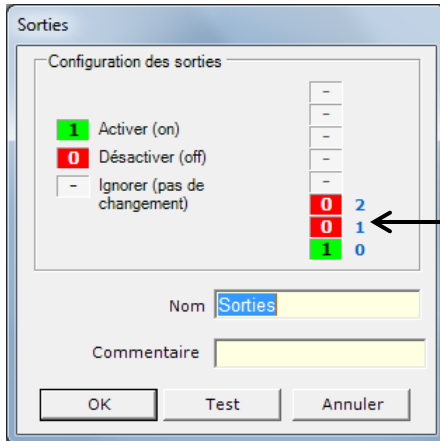
Fin

**Fonction :** Indique la fin d'exécution d'un programme.

### 3.1.3. Général > Sorties



**Fonction :** Permet d'activer ou de désactiver simultanément plusieurs sorties d'un microcontrôleur PICAXE.  
Cette instruction remplace avantageusement une succession de **Activer** et de **Désactiver**.



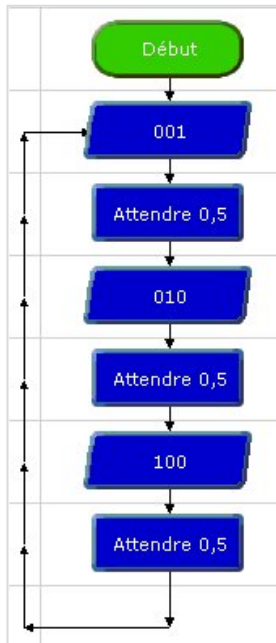
Dans cet exemple, l'instruction **Sorties** permet simultanément d'activer la sortie 0 et de désactiver les sorties 1 et 2.

**Nom** Un nom facile à retenir permet d'expliquer le rôle de la fonction. Attention, cela n'a aucun effet sur le programme.

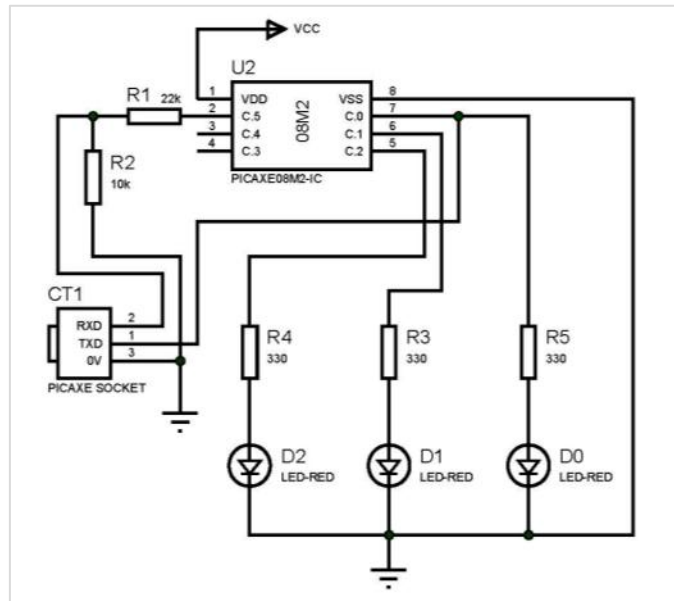
**Note :** en fonction du microcontrôleur PICAXE, les sorties accessibles sont affichées.

Le signe « - » **Ignorer** ne modifie pas l'état de la sortie.

**Programme :** 02Sorties.plf



**Exemple de câblage :** 02\_08M2.DSN



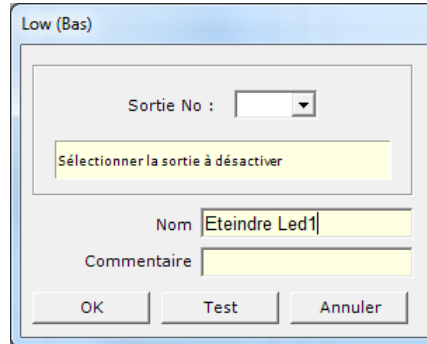
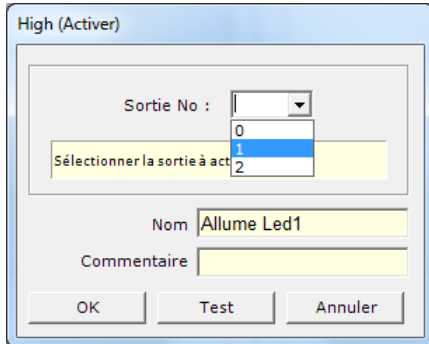
L'utilisation de l'instruction **Sorties** permet de réaliser très facilement un chenillard (animation lumineuse avec 3 LED).



### 3.1.4. Général > Activer et Désactiver



**Fonction :** Permettent de positionner une sortie à l'état haut (**activer**) ou à l'état bas (**désactiver**).



**Sortie No** Le numéro sélectionné correspond à la broche de sortie à activer/désactiver.

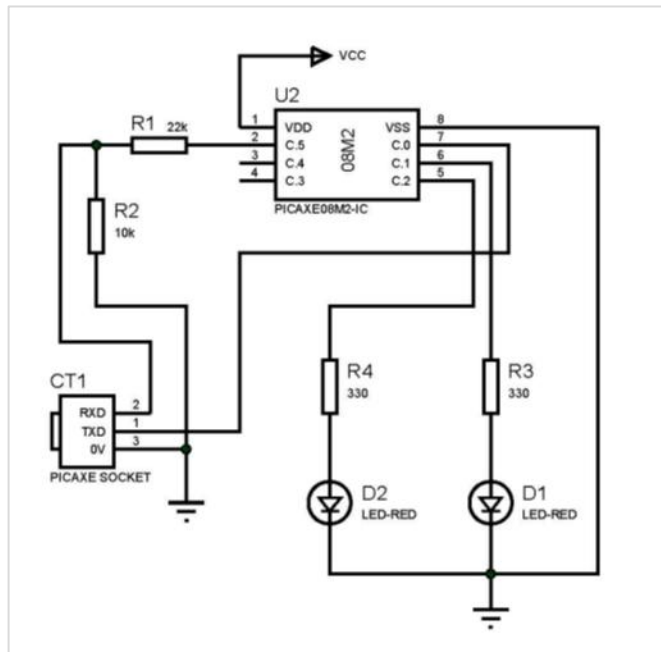
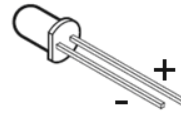
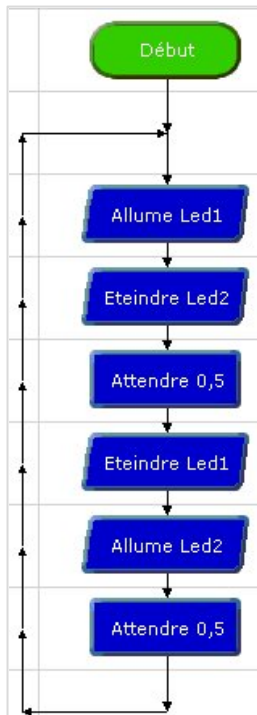
**Nom** Permet d'expliquer le rôle de la fonction. Attention, cela n'a aucun effet sur le programme. Un commentaire peut être ajouté.

Le bouton **Test** Permet de vérifier la sortie sur le panneau numérique.

**Note :** en fonction du microcontrôleur PICAXE, les sorties accessibles sont affichées.

**Programme :** 01Activer.plf

**Exemple de câblage :** 01\_08M2.DSN



#### Clignotant 2 LED :

- pendant 0,5 seconde, la sortie 1 est activée et la sortie 2 est désactivée ;
- puis, pendant 0,5 seconde, la sortie 1 est désactivée et la sortie 2 est activée ;
- et ainsi de suite.

Cela permet un clignotement alterné.

### 3.1.5. Général > Moteurs



**Fonction** : L'instruction **Moteurs** permet d'agir simultanément sur plusieurs sorties du microcontrôleur PICAXE en vue de piloter deux moteurs à courant continu destinés à animer un robot.

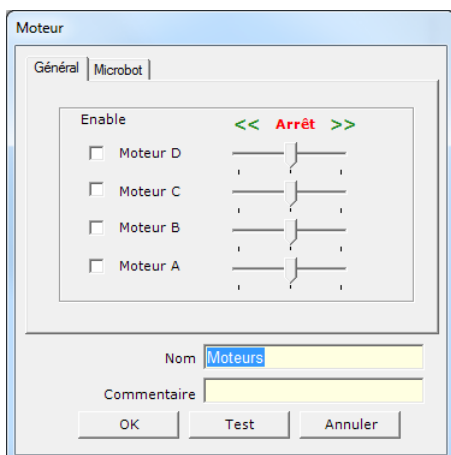
Le nombre de moteurs commandés dépend du microcontrôleur PICAXE utilisé.

**Note** : les moteurs à courant continu ne sont pas branchés en liaison directe avec les sorties du microcontrôleur PICAXE mais au travers d'un circuit de puissance, comme le L293.

Un circuit spécifique équipe éventuellement la carte de pilotage du robot afin de gérer sa vitesse (circuit PWM / MLI).

Cette instruction dispose de 2 onglets : **Général** et **Microbot**.

L'onglet **Général** donne accès à la sélection des moteurs utilisés.



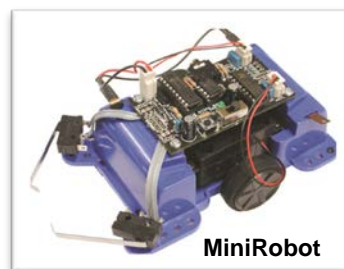
L'onglet **Microbot** est utilisé pour les robots Microbot et Minirobot, équipés respectivement d'un microcontrôleur PICAXE 20X2 et 18M.

Ecran spécifique à Microbot (20X2)

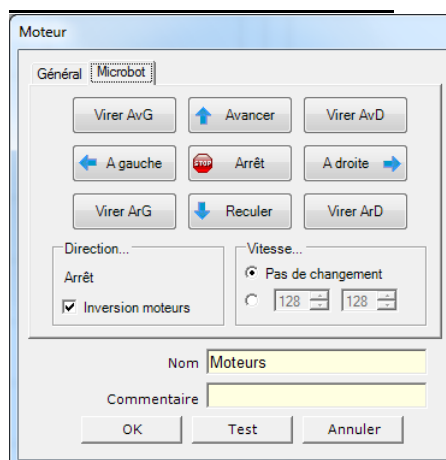
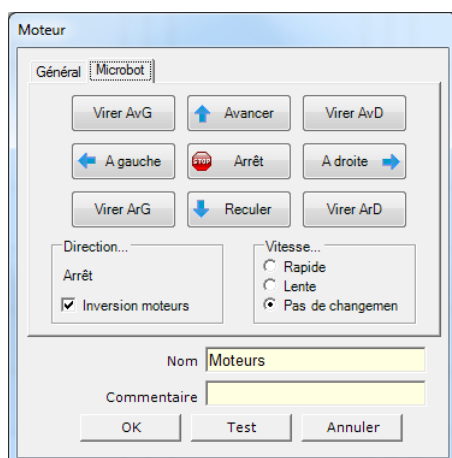


**Microbot**

Ecran spécifique à MiniRobot (18M)



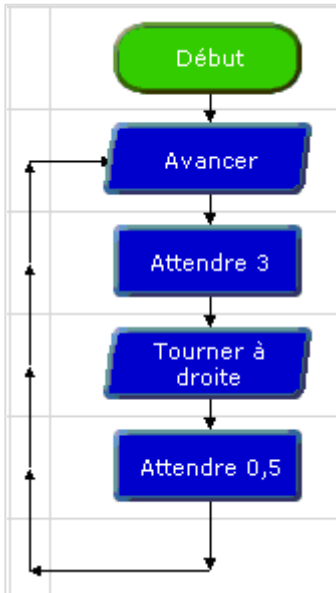
**MiniRobot**



Les boutons **Avancer**, **Reculer**, **A gauche** et **A droite** commandent 2 moteurs simultanément.

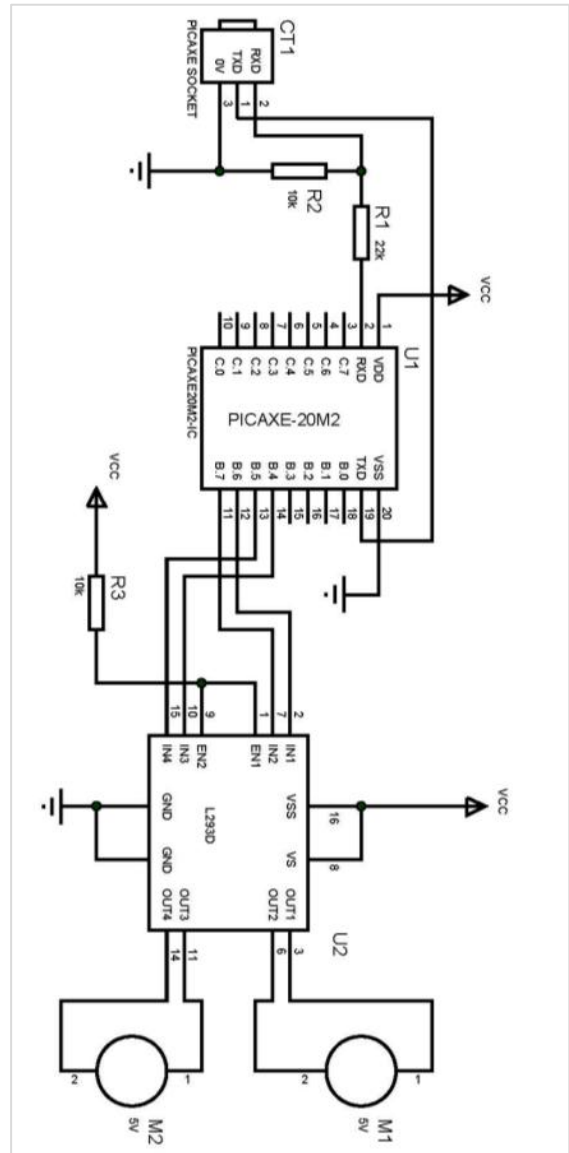
Les boutons **Virer AvG**, **Virer AvD**, **Virer ArG**, **Virer ArD** commandent 1 seul moteur à la fois.

**Vitesse** permet d'agir sur la vitesse de rotation du moteur. En fonction du microcontrôleur PICAXE sélectionné, vous avez accès ou non à cette option.

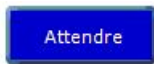


**Déplacement du robot :**

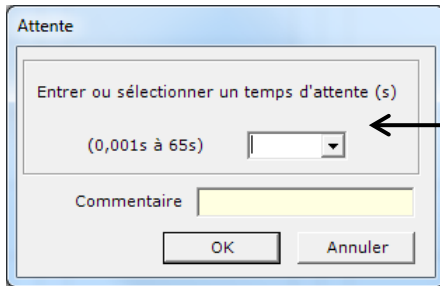
- le robot avance pendant 3 secondes ;
- il effectue une rotation à droite pendant 0,5 seconde ;
- et ainsi de suite.



### 3.1.6. Général > Attendre



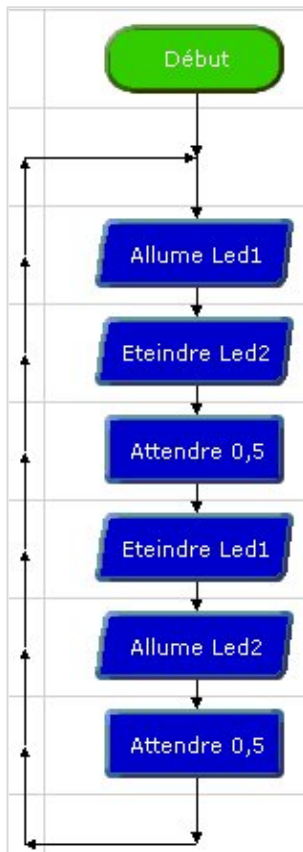
**Fonction :** Permet de positionner un temps d'attente exprimé en secondes.



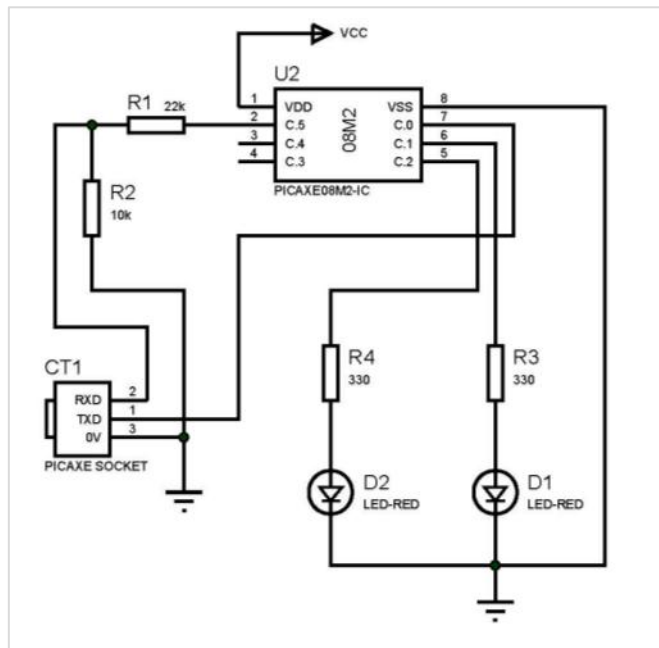
Sélection ou saisie du temps d'attente (de 0,001 à 65 secondes). Le temps peut également être défini par une variable.

**Note :** la saisie manuelle du temps nécessite l'utilisation du symbole « , ». Si vous utilisez le symbole « . » un message d'erreur apparaît à l'écran.

**Programme :** 04Attendre.plf

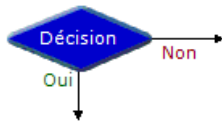


**Exemple de câblage :** 04\_08M2.DSN



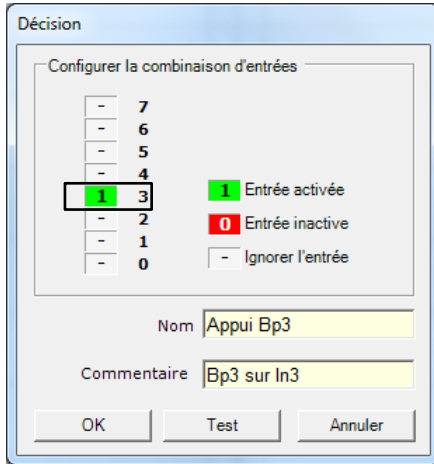
Il est indispensable de positionner des pauses pour que les actions soient visibles.

### 3.1.7. Général > Décision



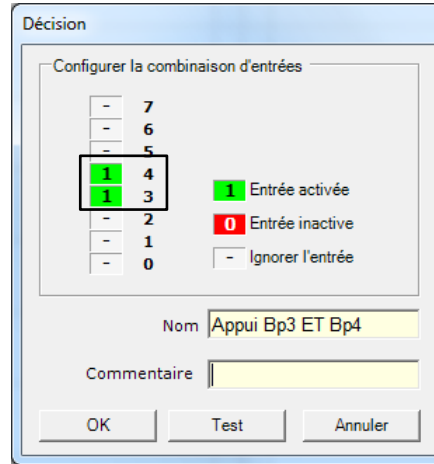
**Fonction :** Permet de déclencher une action en fonction de l'état d'une entrée unique ou d'une combinaison de plusieurs entrées numériques du microcontrôleur PICAXE.

Test d'une entrée unique



Ici, on teste l'état de l'entrée 3. Le résultat du test est positif si l'entrée 3 est active.

Test d'une combinaison d'entrées

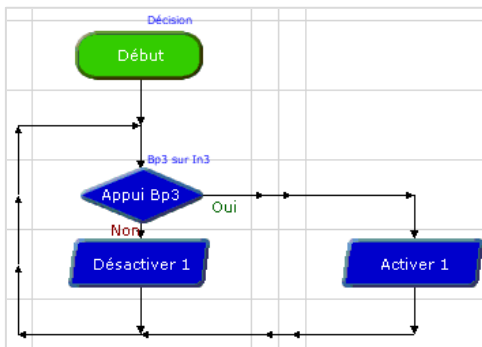


Ici, on teste l'état des entrées 3 et 4. Le résultat du test est positif si les 2 entrées sont actives simultanément (ET logique).

**Nom** Un nom facile à retenir permet d'expliquer le rôle de la fonction. Attention, cela n'a aucun effet sur le programme.

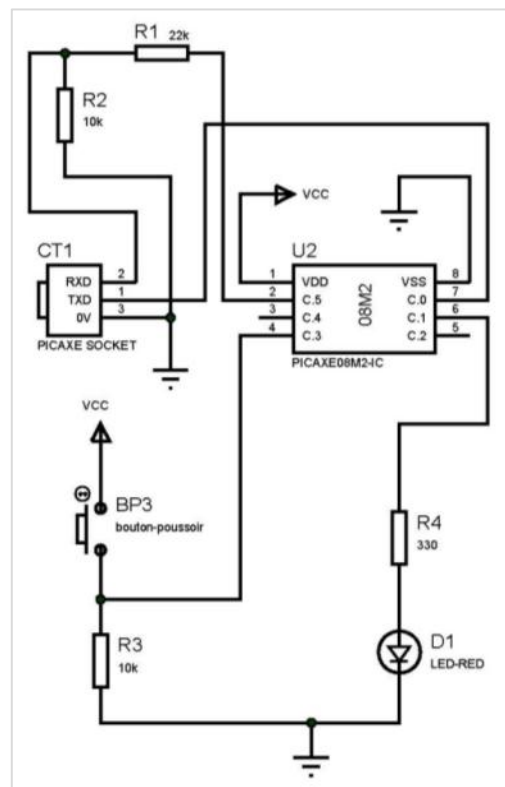
**Note :** Pour intervertir Oui et Non, effectuer un clic droit.

**Programme :** 05Decision.plf

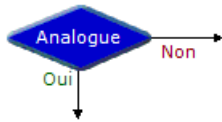


- Si on appuie sur le Bp3, le résultat du test est positif (Oui) et la sortie 1 est activée. = La LED D1 s'allume.
- Si on n'appuie pas sur le Bp3, le résultat du test est négatif (Non) et la sortie 1 est désactivée. = La LED D1 est éteinte.

**Exemple de câblage pour tester l'entrée In3 :** 05\_08M2.DSN

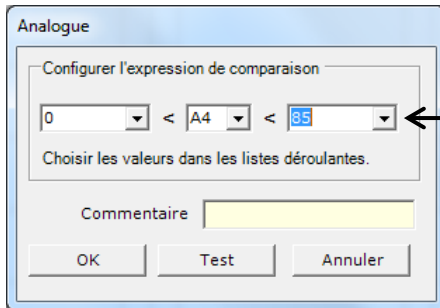


### 3.1.8. Général > Analogue



**Fonction :** Permet de déclencher une action en fonction de la valeur d'une entrée analogique.

**Note :** la fonction « Calibrer un capteur analogique » permet de visualiser en direct la valeur retournée par un capteur connecté à une entrée analogique du microcontrôleur PICAXE. Voir le chapitre correspondant à la fin de ce document.

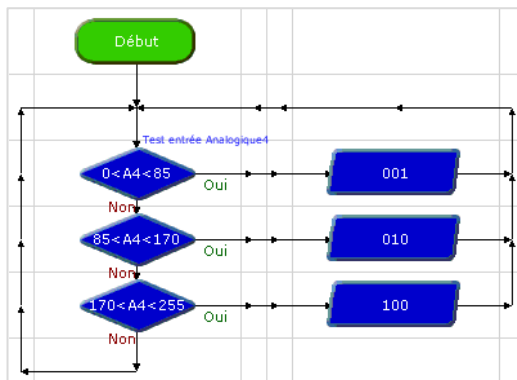


Il faut définir la fourchette des valeurs et choisir l'entrée analogique utilisée (dépend du microcontrôleur PICAXE).

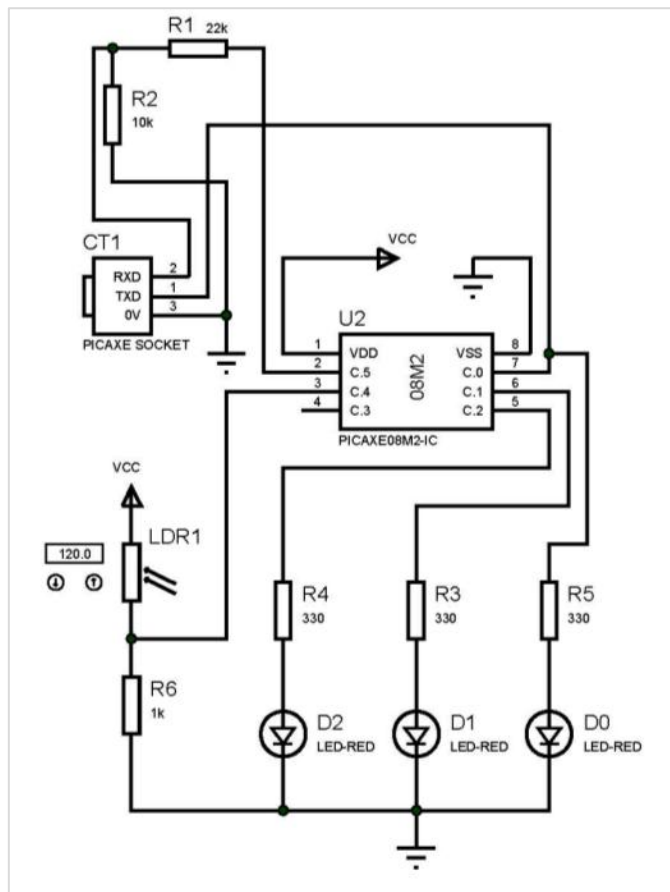
**Note :** Pour intervertir Oui et Non, effectuer un clic droit.

**Programme :** 06Analogue.plf

**Exemple de câblage :** 06\_08M2.DSN



Selon la valeur de l'entrée A4, une des sorties 0, 1 ou 2 est activée.



Allumage de 3 LED (D0, D1 et D2) en fonction de la valeur de l'entrée analogique C4.

### 3.1.9. Général > Comment



**Fonction :** Permet d'écrire un commentaire, de donner, par exemple, des indications sur le fonctionnement du programme.  
Cette instruction n'a aucun rôle sur le déroulement du programme.

Commentaire

Saisir le texte de ce commentaire et cliquer sur OK (50 caractères maxi.)

Texte du commentaire :

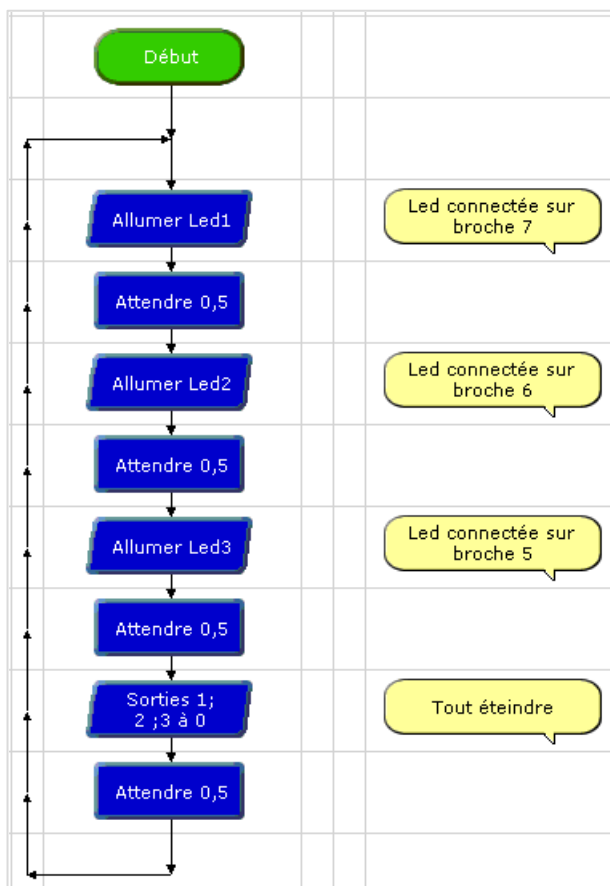
Commentaire

OK Annuler

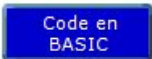
**Aucune option sélectionnable.**

**Note :** il est possible d'écrire un texte de 50 caractères.

**Programme commenté :** 07Comment.plf

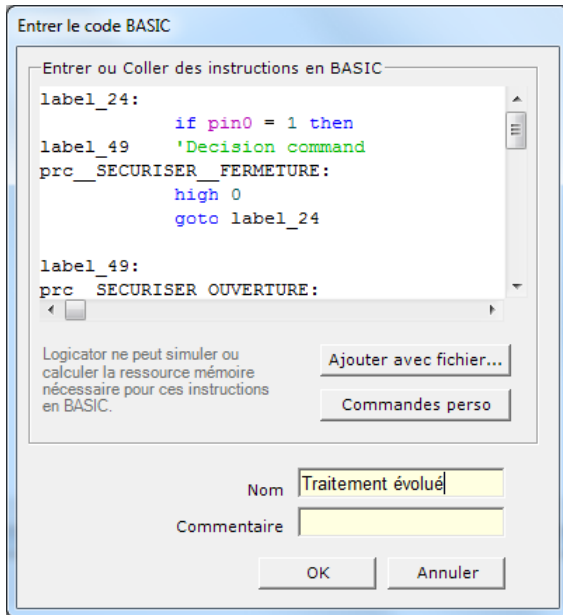


### 3.1.10. Général > Code en BASIC



**Fonction :** Permet d'introduire un traitement évolué écrit en langage BASIC.

Pour plus d'informations, se reporter au chapitre « Le langage BASIC » à la fin de ce document.

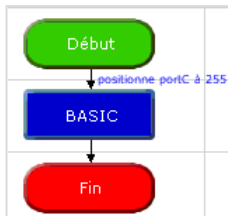


**Ajouter avec fichier...** Intègre des commandes BASIC écrites avec le bloc-notes. Les commandes BASIC peuvent être modifiées.

**Commandes perso** Intègre des commandes BASIC écrites avec le bloc-notes. Les commandes BASIC ne sont pas modifiables. Le nom de fichier s'affiche.

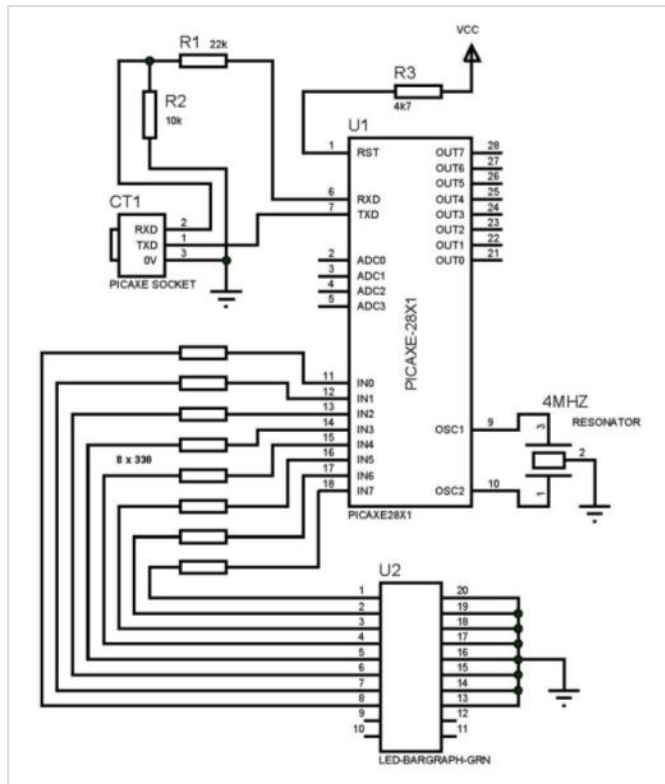
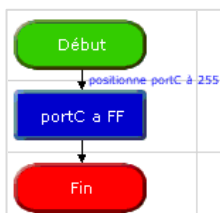
**Programme :** 08Basic.plf

**Exemple de câblage :** 08\_28X1.DSN



L'instruction BASIC permet de positionner le port C en sortie et de le positionner à FF donc toutes les sorties sont activées.

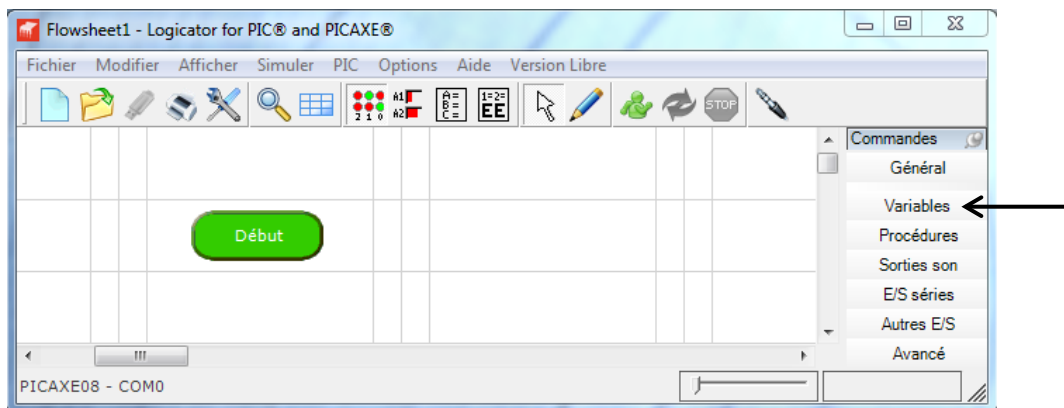
Si des instructions sont écrites dans le bloc-notes (portC à FF), la demande d'instructions perso indique le nom de fichier :





## 3.2. Variables

A partir de la barre des commandes, cliquer sur **Variables**.


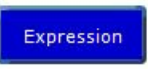

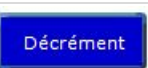
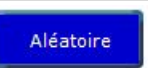


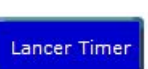


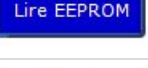


Ce groupe donne accès aux instructions manipulant les variables.

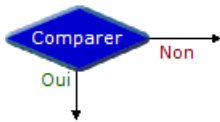
Les variables utilisées par Logicator sont notées de A à T, codées sur 8 bits, elles évoluent donc de 0 à 255.

Il existe aussi une variable particulière : Time qui permet de gérer le temps.

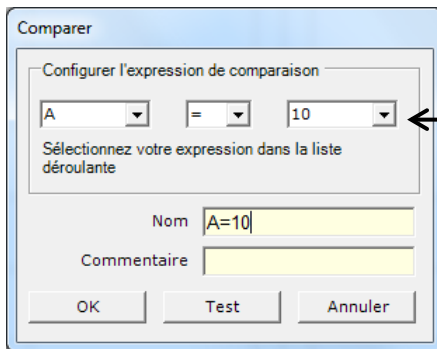
A la mise en route, ou après une initialisation, toutes les variables sont positionnées à 0.

	Permet de déclencher une action en fonction de la valeur d'une variable.
	Permet de positionner une variable ou d'effectuer un calcul.
	Permet d'incrémenter une variable.
	Permet de décrémenter une variable.
	Permet de générer une valeur pseudo-aléatoire (entre 0 et 255) et de la stocker dans une variable.
	Permet de lire simultanément l'état des entrées numériques du microcontrôleur PICAXE.
	Permet d'activer ou de désactiver simultanément plusieurs sorties en fonction de la valeur d'une variable.
	Permet de déclencher le comptage d'un temps écoulé (en secondes) jusqu'à l'utilisation d'une instruction <b>Arrêter Timer</b> .
	Permet d'arrêter le comptage d'un temps écoulé (en secondes).
	Permet de lire le contenu de la mémoire EEPROM
	Permet d'écrire une valeur dans la mémoire EEPROM.

### 3.2.1. Variables > Comparer



**Fonction :** Permet de déclencher une action en fonction de la valeur d'une variable.



Il faut :  
 – définir la variable à tester ;  
 – choisir l'opérateur de comparaison ;  
 – et définir la valeur de comparaison.

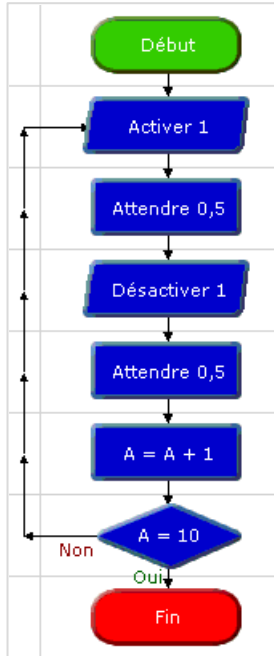
Liste des opérateurs de comparaison disponibles :

- = : égal à
- > : strictement supérieur à
- < : strictement inférieur à
- >= : supérieur ou égal à
- <= : inférieur ou égal à
- ≠ : différent de (≠)

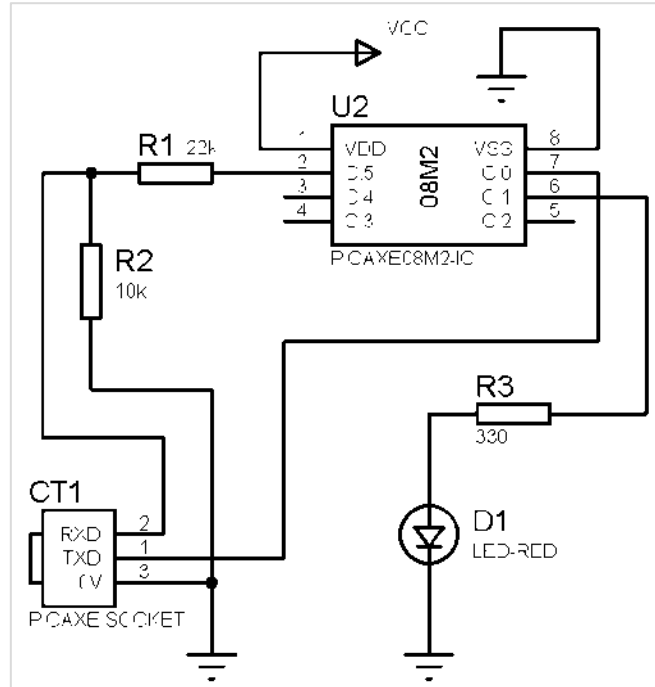
Dans l'exemple ci-dessus, on teste si la variable A est égale à 10 ou non.

**Note :** Pour intervertir Oui et Non, effectuer un clic droit.

**Programme :** 09Comparer.plf

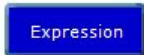


**Exemple de câblage :** 09\_08M2.DSN

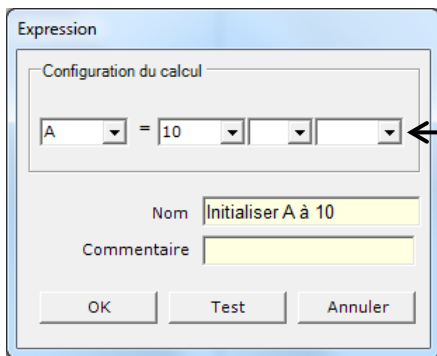


La LED D1 clignote 10 fois puis le programme s'arrête.

### 3.2.2. Variables > Expression



**Fonction :** Permet de positionner une variable ou d'effectuer un calcul.



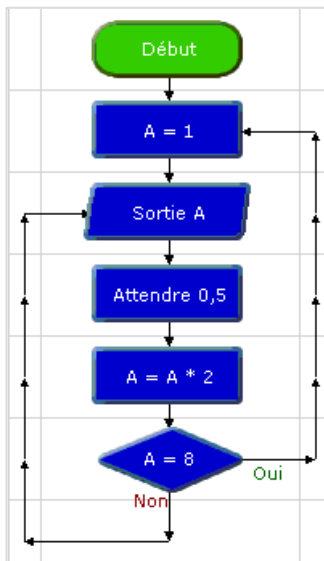
Il faut définir la variable à tester et lui affecter la valeur ou la formule souhaitée.

Liste des opérateurs disponibles :

- + : addition
- : soustraction
- \* : multiplication
- / : division
- % : modulo (reste d'une division)
- & : ET logique
- | : OU logique
- ^ : OU exclusif logique

Dans l'exemple ci-dessus, on initialise la variable A avec la valeur 10.

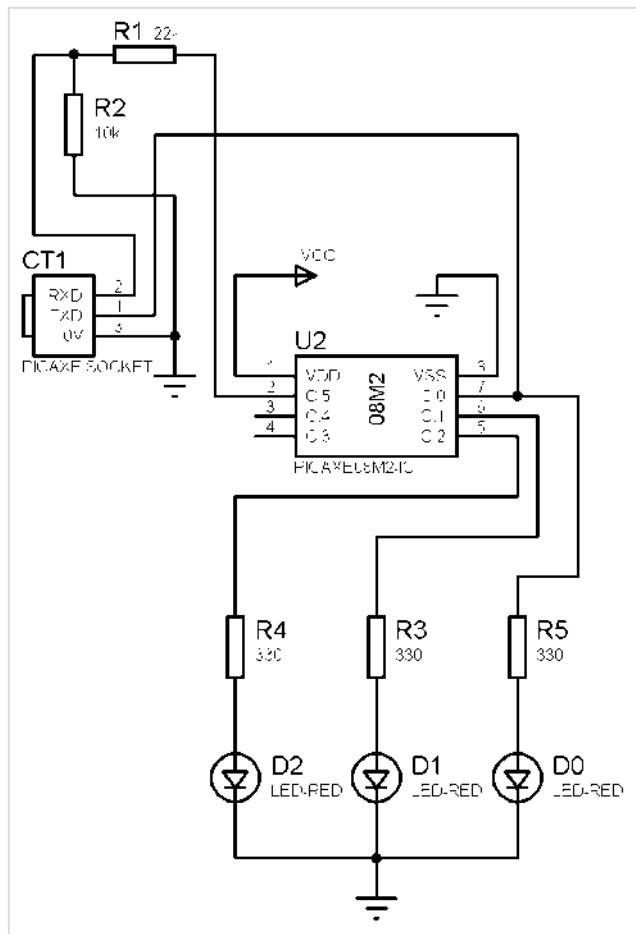
**Programme :** 10Expression.plf



On initialise la variable A à 1 puis on affiche cette variable ce qui permet d'allumer les LED.

L'opération  $A \times 2$  assure un décalage sur les sorties 0, 1 et 2.

**Exemple de câblage :** 10\_08M2.DSN

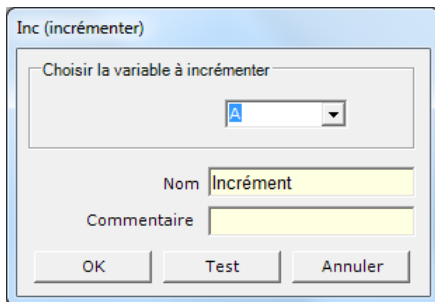


### 3.2.3. Variables > Incrément et Décrément

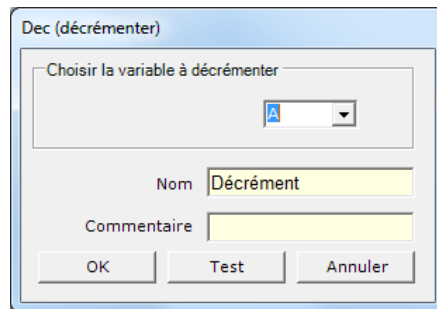
Incrément

**Fonction :** Permet d'incrémenter ou de décrémenter une variable.

Décrément

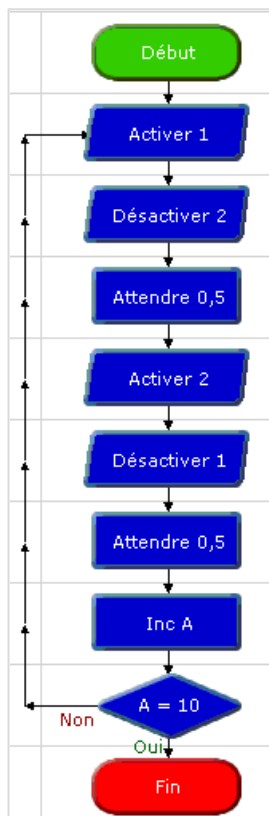


Il faut choisir la variable à incrémenter.  
Incrémenter A est équivalent à l'expression  $A = A + 1$

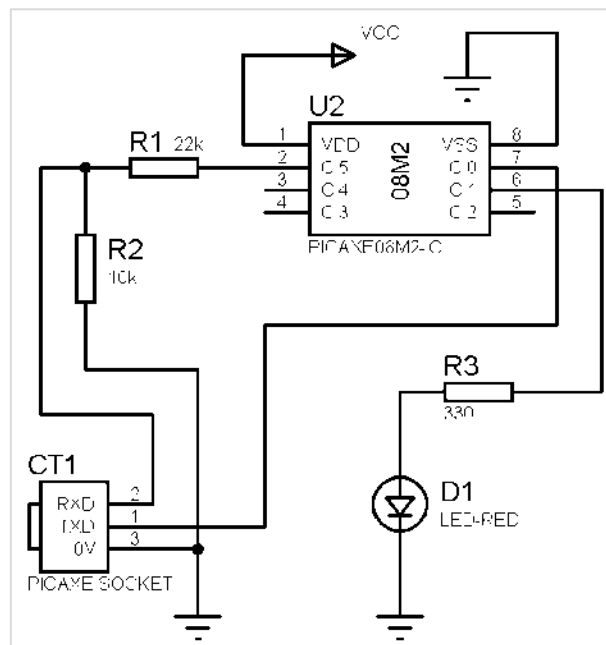


Il faut choisir la variable à décrémenter.  
Décrémenter A est équivalent à l'expression  $A = A - 1$

**Programme :** 11Increment.plf



**Exemple de câblage :** 11\_08M2.DSN

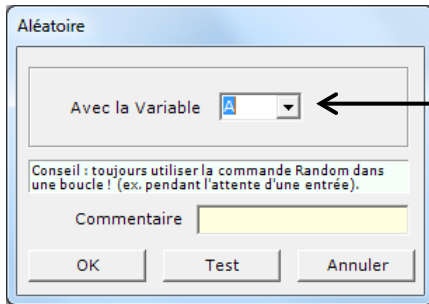


La LED D1 clignote 10 fois puis le programme s'arrête.

### 3.2.4. Variables > Aléatoire

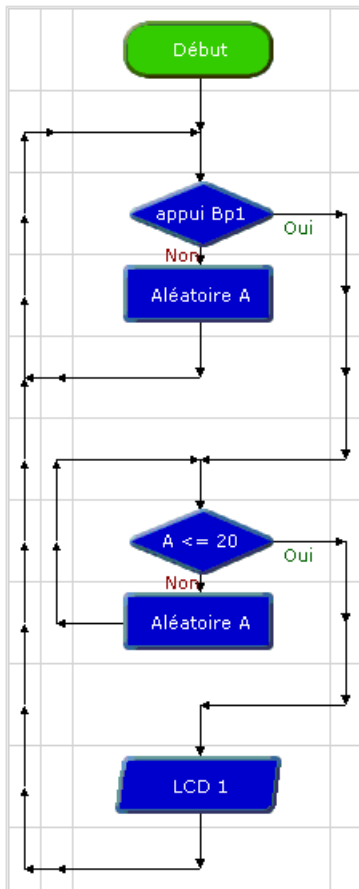


**Fonction :** Permet de générer une valeur pseudo-aléatoire (entre 0 et 255) et de la stocker dans une variable.

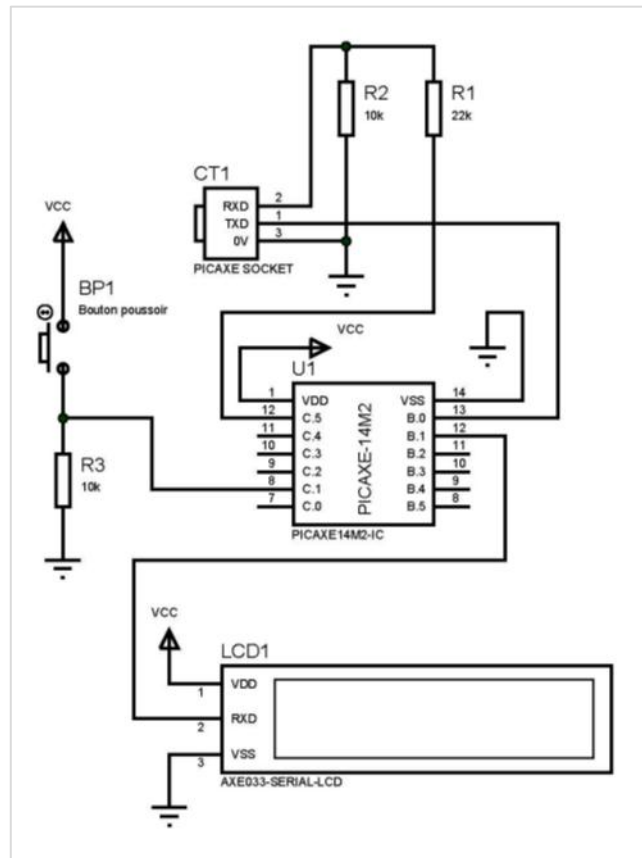


Il faut choisir la variable qui accueille la valeur pseudo-aléatoire.

Programme : 12Aleatoire.plf



Exemple de câblage : 12\_14M2.DSN



#### Génération et affichage d'une valeur aléatoire entre 0 et 20 :

La première instruction **Aléatoire** est incluse dans une boucle et génère un nombre aléatoire en permanence.

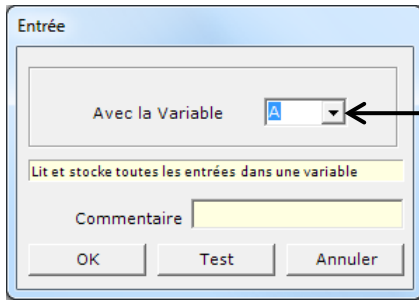
Dès l'appui de Bp1, la variable A est testée pour savoir si elle est inférieure ou égale à 20.

En fonction de ce test, le nombre apparaît sur l'afficheur LCD.

### 3.2.5. Variables > Entrées



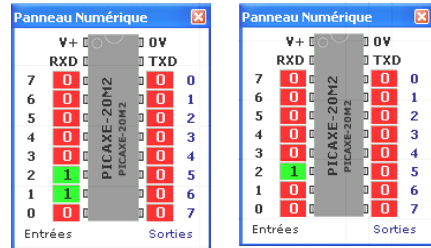
**Fonction :** Permet de lire simultanément l'état des entrées numériques du microcontrôleur PICAXE.  
Il s'agit d'une information binaire qui est stockée dans une variable.



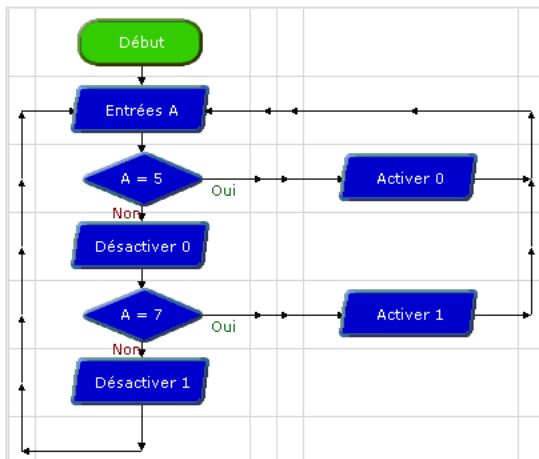
Il faut choisir la variable dans laquelle on stocke l'état des entrées numériques.

#### Exemples :

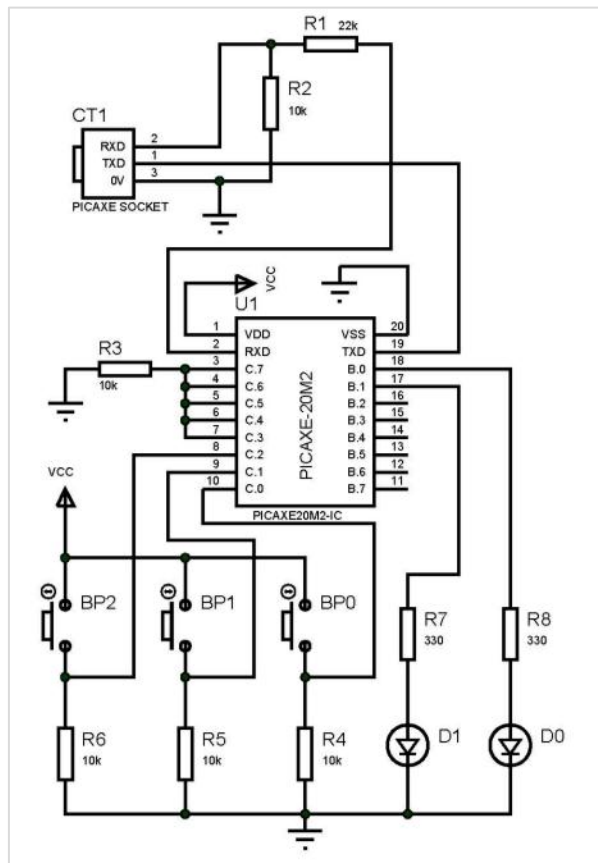
- si l'état des entrées numériques correspond au code binaire 0000110 (6 en décimal), la valeur stockée dans la variable est 6 ;
- si l'état des entrées numériques correspond au code binaire 0000100 (4 en décimal), la valeur stockée dans la variable est 4.



#### Programme : 13Entrees.DSN



#### Exemple de câblage : 13\_20M2.DSN



**Activation des LED D0 et D1 en fonction de l'appui des boutons poussoirs BP0, BP1 et BP2.**

La table de vérité ci-dessous représente l'état des sorties en fonction de toutes les combinaisons possibles sur les 3 entrées.

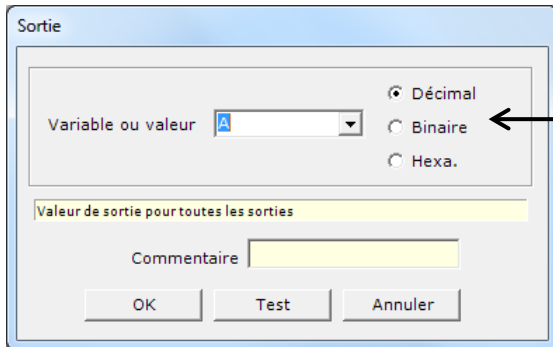
ENTREES (code binaire)			VAR. A (déc.)	SORTIES	
BP2	BP1	BP0		D1	D0
0	0	0	0	OFF	OFF
0	0	1	1	OFF	OFF
0	1	0	2	OFF	OFF
0	1	1	3	OFF	OFF
1	0	0	4	OFF	OFF
1	0	1	5	OFF	ON
1	1	0	6	OFF	OFF
1	1	1	7	ON	OFF

La variable A représente l'état des entrées numériques du microcontrôleur PICAXE (codées en décimal). Elle peut prendre une valeur de 0 à 7.

### 3.2.6. Variables > Sortie

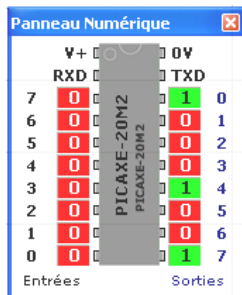


**Fonction :** Permet d'activer ou de désactiver simultanément plusieurs sorties en fonction de la valeur d'une variable.  
 Cette valeur peut être binaire, décimale ou hexadécimale.

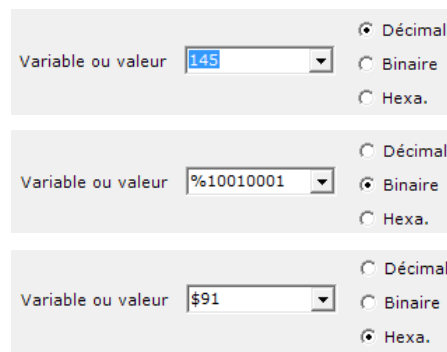


Les données présentées dans la liste déroulante varient en fonction de l'option choisie.

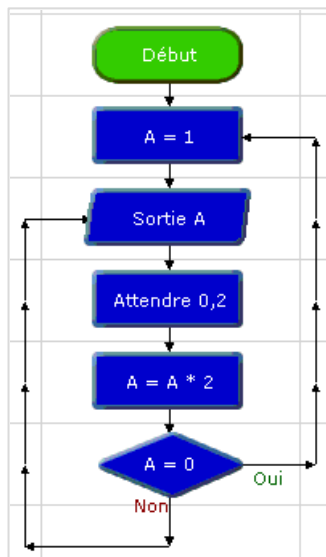
Exemple d'activation des sorties avec le code binaire %10010001



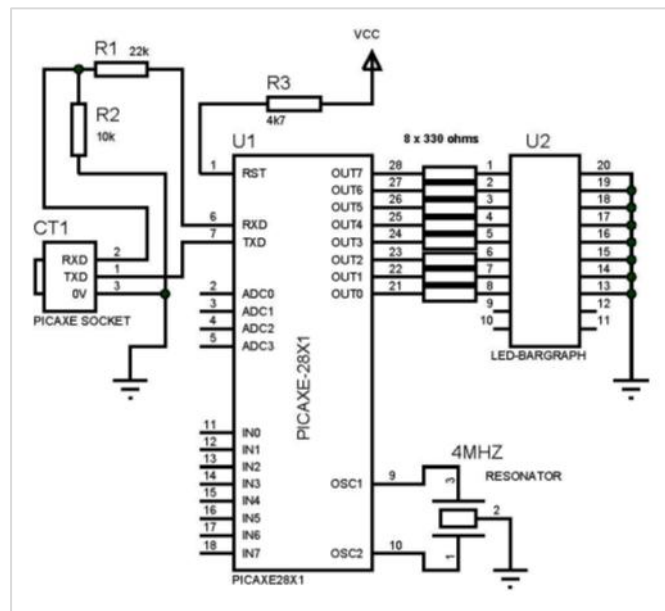
**Note :** les valeurs suivantes sont équivalentes d'une base à l'autre.



Programme : 14Sorties.DSN



Exemple de câblage : 14\_28X1.DSN



**Chenillard 8 sorties :**

L'expression  $A \times 2$  permet de décaler l'affichage des LED.

Le test  $A = 0$  permet de reprendre le comptage quand on arrive à 256 soit 0.

### 3.2.7. Variables > Lancer Timer et Arrêter Timer



**Fonction :** permet une mesure précise du temps écoulé entre l'exécution des deux instructions **Lancer Timer** et **Arrêter Timer**.

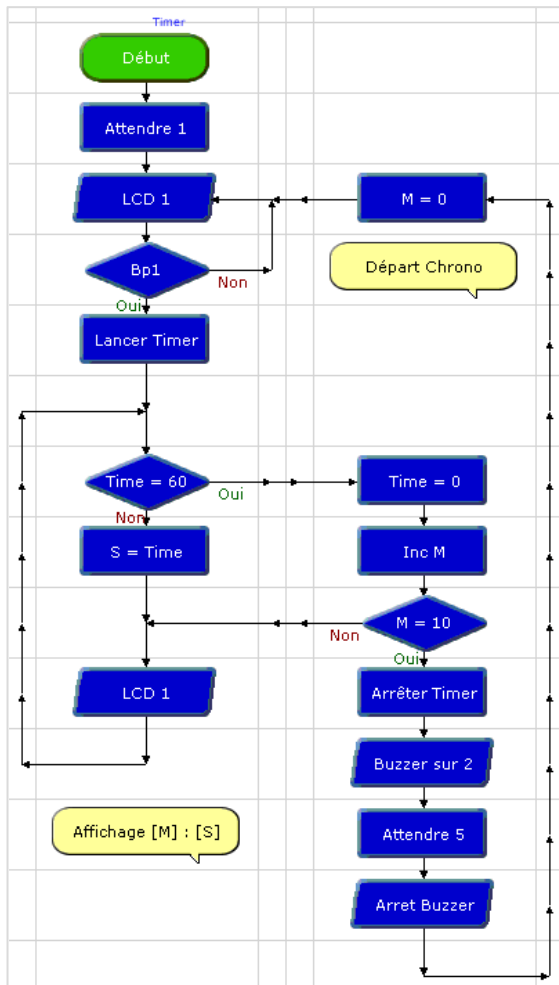
Le temps écoulé (exprimé en secondes) entre le lancement et l'arrêt du Timer est stocké dans la variable « Time ».

**Note :** Le Timer est une base de temps interne du microcontrôleur PICAXE qui s'incrémente en tâche de fond toutes les secondes.

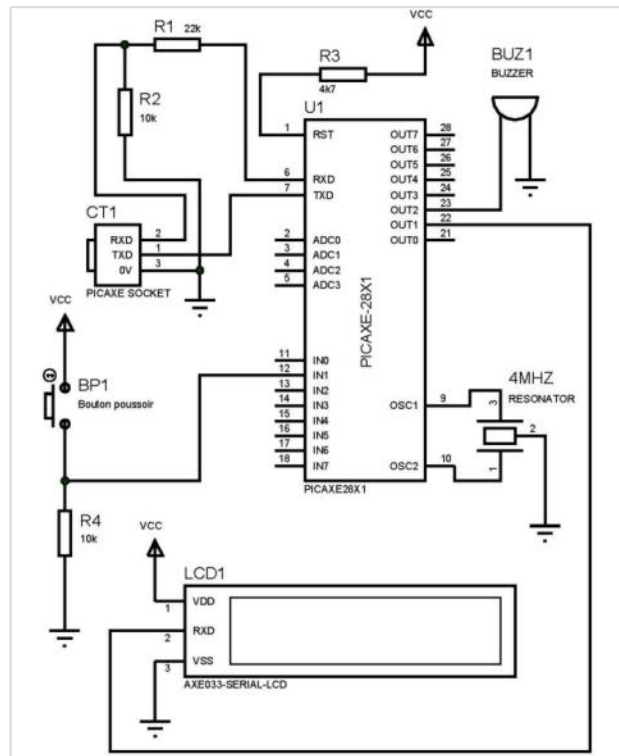
Aucune option sélectionnable.

**Note :** Cette instruction fonctionnant en tâche de fond n'est pas influencée par le déroulement d'un programme. Elle doit être préférée à l'instruction **Attendre**.

Programme : 15Timer.DSN



Exemple de câblage : 15\_28X1.DSN



#### Chronomètre 10 minutes :

La temporisation d'une seconde permet au LCD de se configurer correctement.

Dès l'appui de Bp1, le Timer est lancé et compte les secondes.

Le test à 60 secondes permet d'incrémenter les minutes.

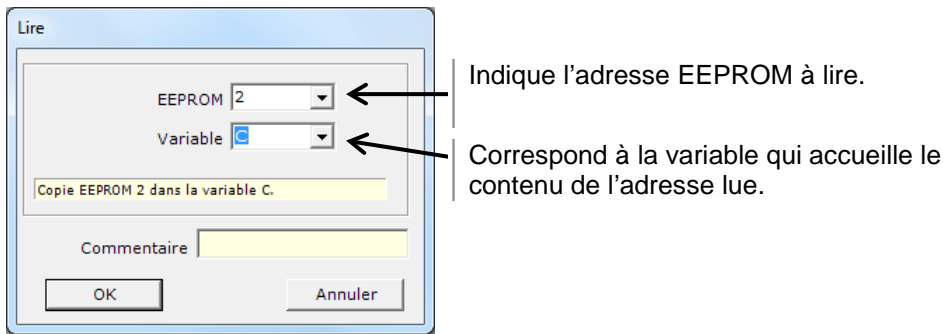
Au bout de 10 minutes, le Timer est arrêté, la sortie Out2 est activée, le buzzer résonne pendant 5 secondes.



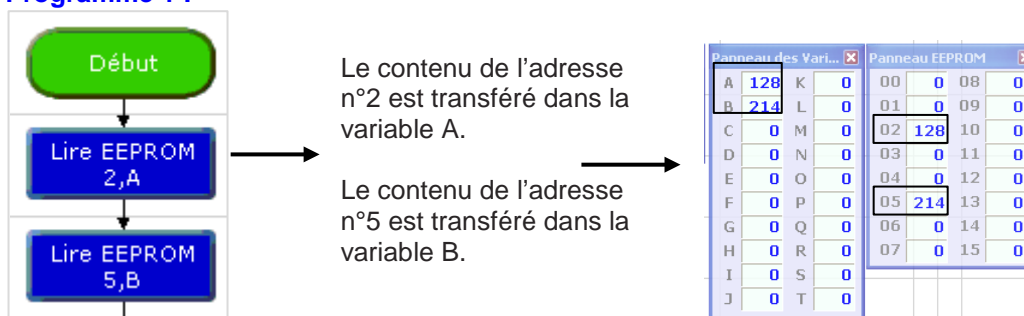
### 3.2.8. Variables > Lire EEPROM



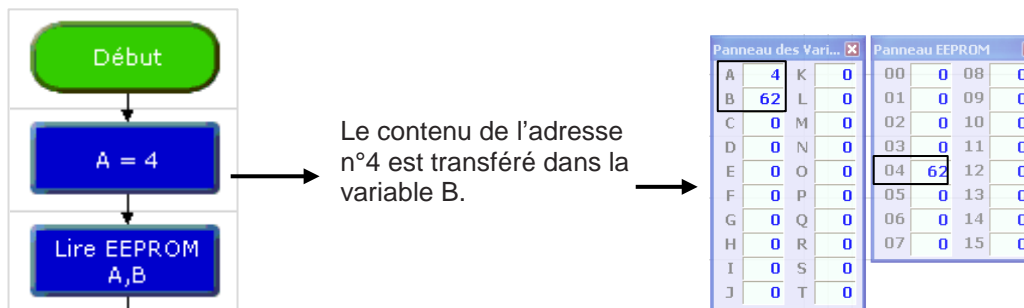
**Fonction :** Permet de lire le contenu de la mémoire EEPROM. La mémoire EEPROM conserve les données même lorsque le microcontrôleur PICAXE est hors tension. Elle permet de stocker jusqu'à 16 valeurs distinctes codées sur 8 bits (16 octets).



#### Programme 1 :



#### Programme 2 :

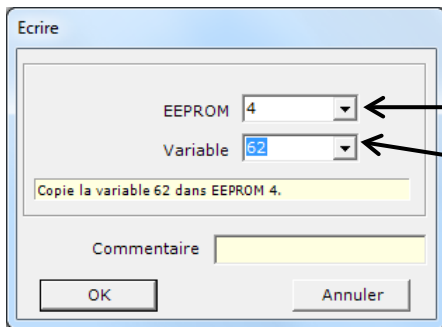


### 3.2.9. Variables > Ecrire EEPROM



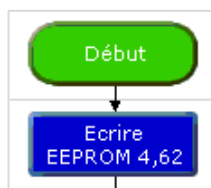
**Fonction :** Permet d'écrire une valeur dans l'EEPROM.

La mémoire EEPROM conserve les données même lorsque le microcontrôleur PICAXE est hors tension. Elle permet de stocker jusqu'à 16 valeurs distinctes codées sur 8 bits (16 octets).



Indique l'adresse EEPROM à écrire.  
Sélection de la variable ou de la valeur à écrire dans l'EEPROM.

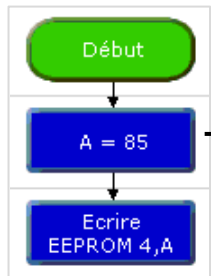
#### Programme 1 :



La valeur 62 est écrite à l'adresse n°4.

00	0	08	0
01	0	09	0
02	0	10	0
03	0	11	0
04	62	12	0
05	0	13	0
06	0	14	0
07	0	15	0

#### Programme 2 :

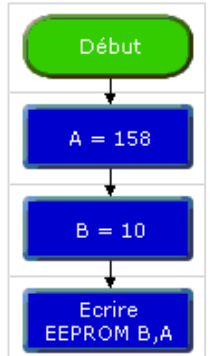


Le contenu de la variable A (85) est écrit à l'adresse n°4.

A	85	K	0
B	0	L	0
C	0	M	0
D	0	N	0
E	0	O	0
F	0	P	0
G	0	Q	0
H	0	R	0
I	0	S	0
J	0	T	0

00	0	08	0
01	0	09	0
02	0	10	0
03	0	11	0
04	85	12	0
05	0	13	0
06	0	14	0
07	0	15	0

#### Programme 3 :



Le contenu de la variable A (158) est écrit à l'adresse n°10 (B).

A	158	K	0
B	10	L	0
C	0	M	0
D	0	N	0
E	0	O	0
F	0	P	0
G	0	Q	0
H	0	R	0
I	0	S	0
J	0	T	0

00	0	08	0
01	0	09	0
02	0	10	158
03	0	11	0
04	0	12	0
05	0	13	0
06	0	14	0
07	0	15	0

### Valeurs initiales EEPROM

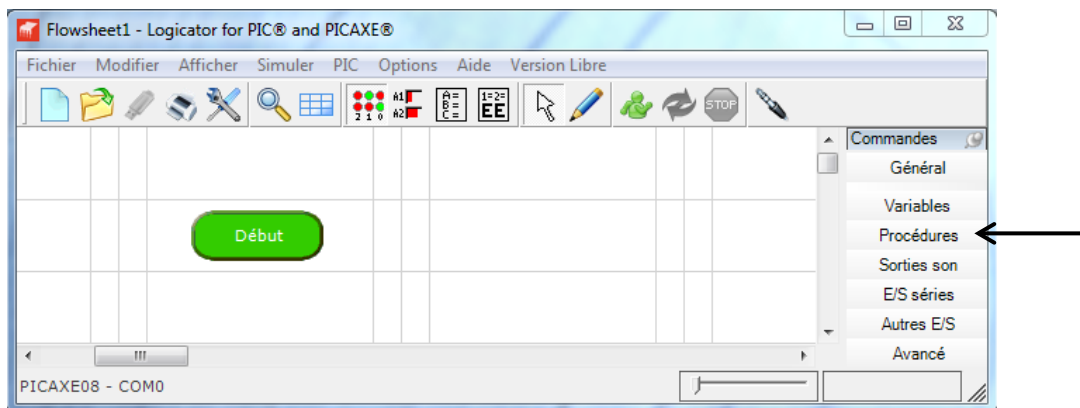
00	25	08	0
01	3	09	0
02	245	10	0
03	0	11	0
04	0	12	0
05	0	13	0
06	0	14	0
07	0	15	0

Logicator permet d'écrire dans la mémoire EEPROM uniquement à la programmation du microcontrôleur PICAXE.


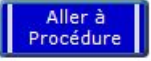

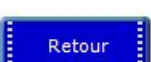
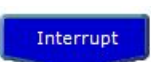
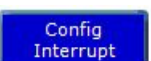
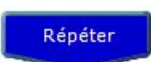

A partir de la barre des menus, cliquer sur **PIC > Valeurs initiales EEPROM**.

### 3.3. Procédures

A partir de la barre des commandes, cliquer sur **Procédures**.



Ce groupe donne accès aux instructions nécessaires à la mise en œuvre de sous-programmes, des interruptions et des répétitions.

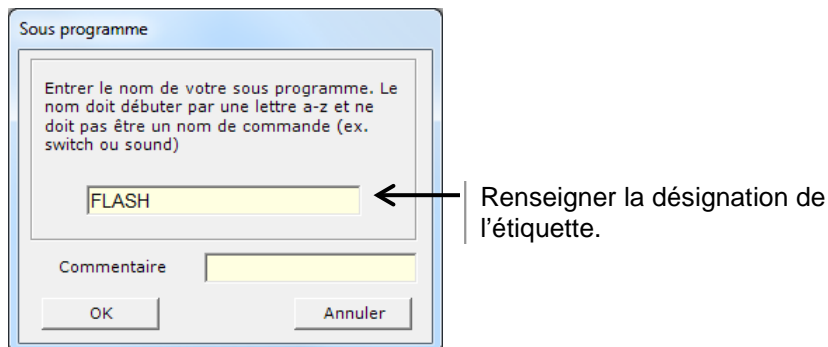
	Permet d'attribuer une étiquette au point de départ d'un groupe d'instructions ou d'un sous-programme à exécuter.
	Permet d'effectuer un branchement dans un même programme ( <i>GoTo</i> ).
	Permet d'effectuer un saut vers un sous-programme ( <i>GoSub</i> ) qui doit se clôturer par une instruction <b>Retour</b> .
	Permet d'indiquer la fin du sous-programme et le retour dans le programme principal.
	Permet d'indiquer le début d'un sous-programme de traitement d'interruption.
	Permet d'autoriser ou d'interdire la prise en compte d'une interruption.
	Permet de positionner une boucle dans un programme et gérer ainsi très facilement une répétition d'instructions.
	Permet de positionner une boucle dans un programme et gérer ainsi très facilement une répétition d'instructions.

### 3.3.1. Procédures > Procédure



Il est possible de réunir l'ensemble des instructions dans un même programme ou bien de les organiser dans un programme principal qui fait appel à des sous-programmes.

**Fonction :** permet d'attribuer une étiquette au point de départ d'un groupe d'instructions ou d'un sous-programme à exécuter.



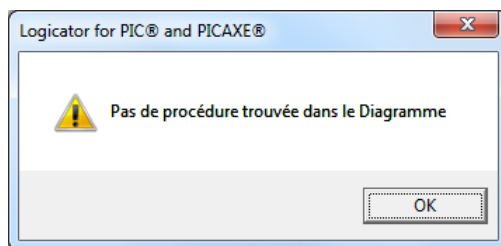
Il est possible de l'utiliser de deux manières :

- Avec l'instruction **Aller à Procédure**, pour effectuer un branchement dans un même programme (*GoTo*).
- Avec l'instruction **Appeler Procédure**, pour effectuer un saut vers un sous-programme (*GoSub*).

---

**Note :** il faut au préalable définir une procédure pour pouvoir utiliser les instructions **Aller à** et **Appeler**. Dans le cas contraire, le message suivant apparaît.

---



---

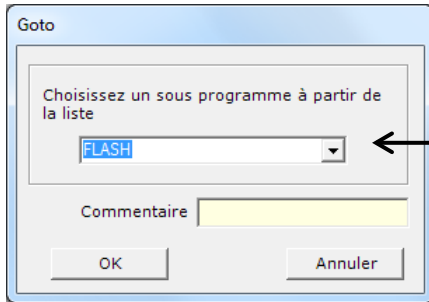
**Note :** le nombre maximum d'instructions **Procédure** dans un programme est de 15 pour les microcontrôleurs PICAXE série M et de 255 pour les microcontrôleurs PICAXE de la série X.

---

### 3.3.2. Procédure > Aller à Procédure



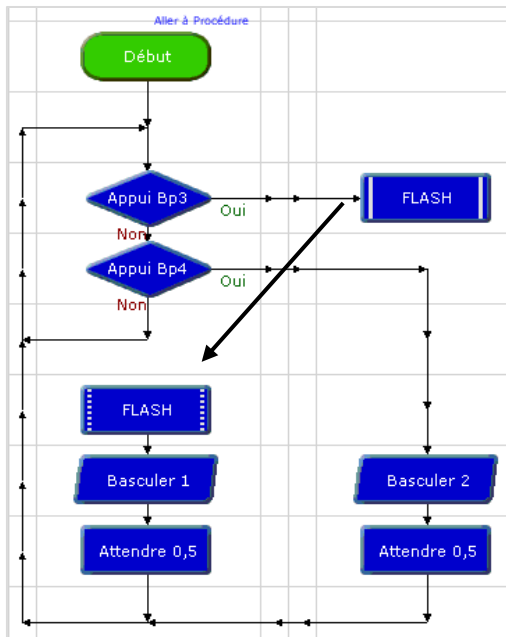
**Fonction :** permet d'effectuer un branchement dans un même programme (*GoTo*).



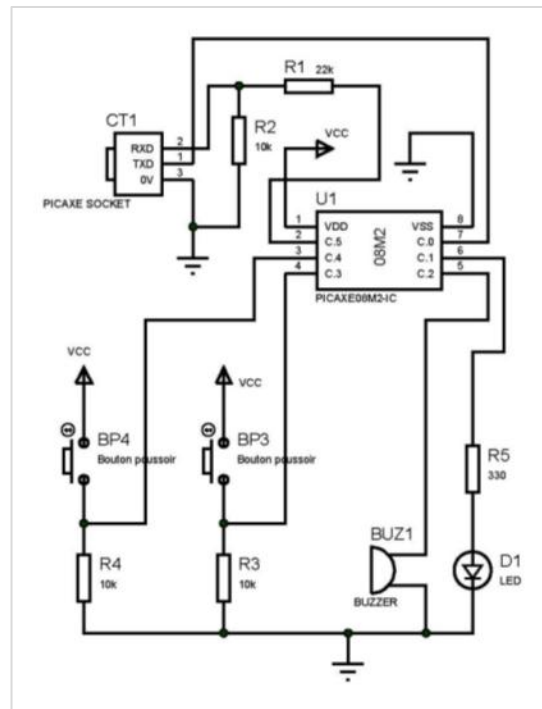
← Sélection de la procédure préalablement définie.

**Note :** cette instruction est à utiliser avec précaution, il faut veiller à ce que le programme ne rentre pas dans une boucle infinie.

**Programme :** 16 Goto.plf



**Exemple de câblage :** 16\_08M2.DSN



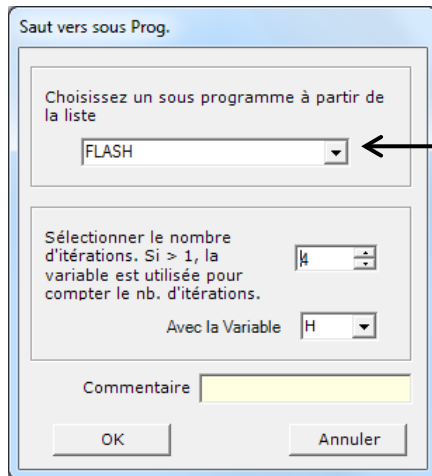
L'instruction **FLASH** effectue un branchement vers l'étiquette FLASH positionnée dans le même programme.

**Note :** cette instruction permet de poursuivre un programme tout en évitant des croisements de liaison.

### 3.3.3. Procédure > Appeler Procédure

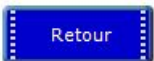


**Fonction :** permet d'effectuer un saut vers un sous-programme (*GoSub*) qui doit se clôturer par une instruction **Retour** qui indique la fin du sous-programme et le retour au programme principal.



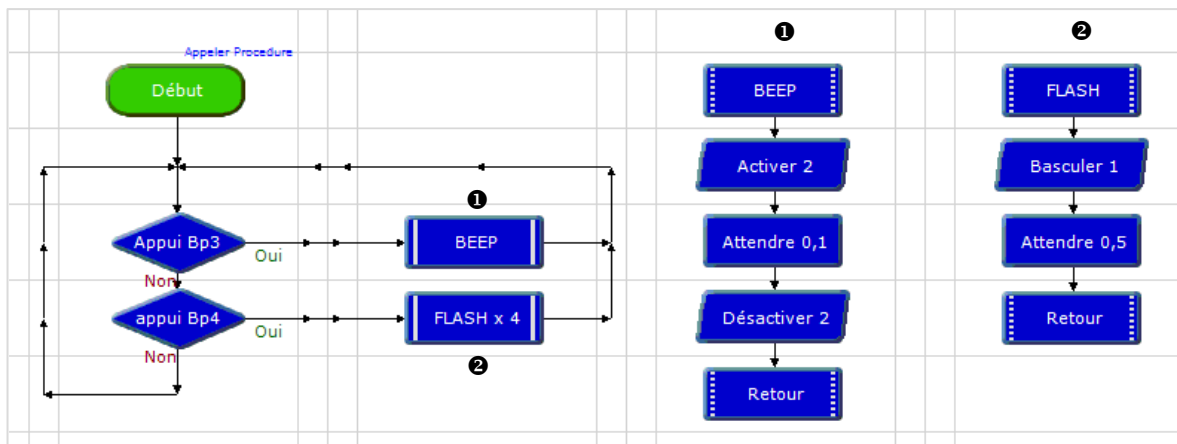
← Sélection du sous-programme préalablement défini.

**Note :** Il est possible d'appeler un sous-programme à partir d'un autre sous-programme. L'imbrication maximum est de 4 instructions **Appeler procédure** pour les microcontrôleurs PICAXE série M et X et de 8 pour les microcontrôleurs PICAXE de la série X1 et X2. Cette instruction est à utiliser avec précaution, il faut veiller à ce que le programme ne rentre pas dans une boucle infinie.



Indique la fin du sous-programme et le retour dans le programme principal.

**Exemples de programme :** 16Procedure.plf

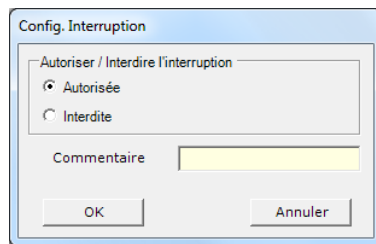
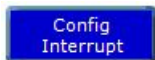


L'instruction BEEP ① déclenche le saut vers le sous-programme du même nom BEEP.  
L'instruction FLASH x 4 ② déclenche le saut vers le sous-programme FLASH qui doit être exécuté 4 fois avant de revenir au programme principal.

### 3.3.4. Procédures > Config Interrupt et Interrupt

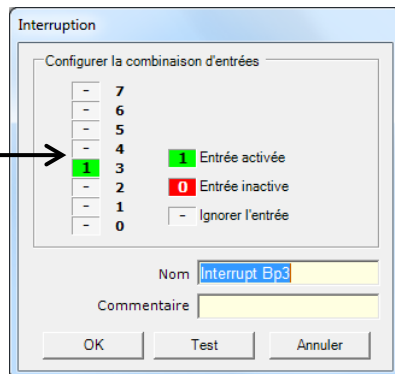
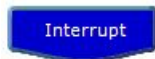
**Fonction :** permet d'interrompre instantanément le déroulement du programme en cours et de donner la priorité à un traitement spécifique.

Exemple : la détection d'une personne à la fermeture d'un portail doit entraîner son arrêt immédiat.



**Config Interrupt** permet d'autoriser ou d'interdire la prise en compte d'une interruption.

Après le traitement d'une interruption, il est nécessaire de réarmer l'interruption en l'autorisant une nouvelle fois avec l'instruction **Config. Interrupt**.



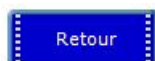
Ici, l'interruption se déclenche si l'entrée 3 est activée.

**Interrupt** permet d'indiquer le début d'un sous-programme de traitement d'interruption.

C'est le changement d'état d'une ou plusieurs entrées numériques qui déclenche l'interruption.

Il faut configurer l'entrée ou la combinaison des entrées numériques à prendre en compte.

**Note :** dans le cas de plusieurs entrées, elles doivent être actives simultanément (ET logique entre les différentes entrées).

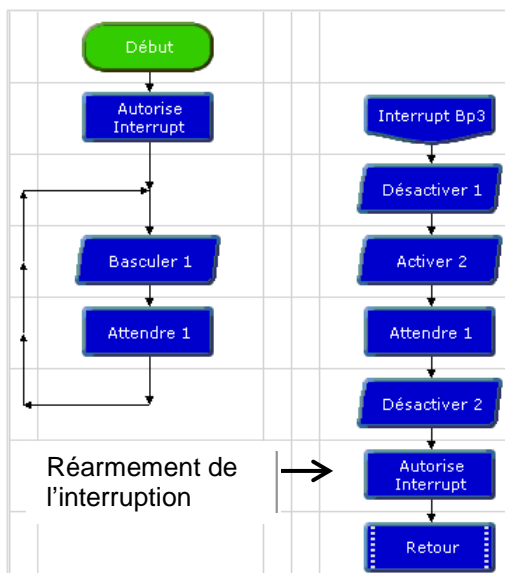


Indique la fin du sous-programme de traitement de l'interruption et le retour dans le programme principal.

L'interruption pouvant intervenir à tout moment, il est difficile de maîtriser son point de retour dans le programme principal.

L'instruction **Fin** peut être utilisée à la place de l'instruction **Retour** pour obliger l'utilisateur à redémarrer le système et relancer le programme à son début.

Programme : 17Interrupt.plf

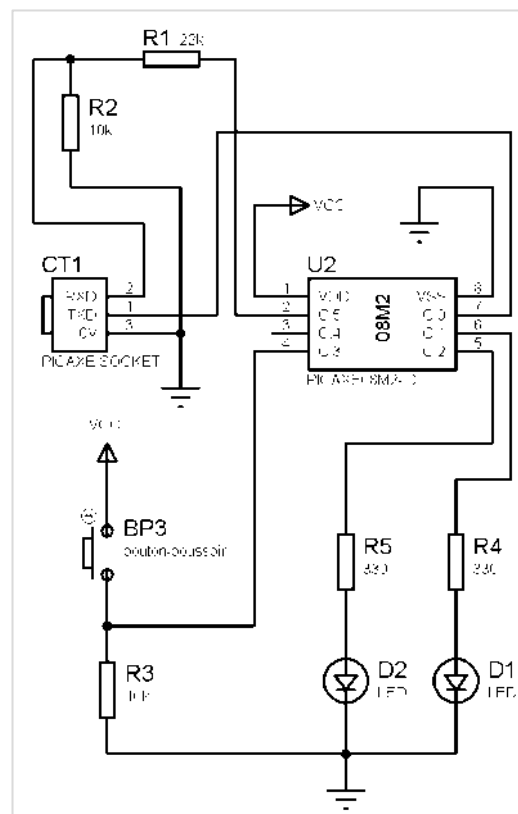


Le programme principal exécute une boucle permanente.

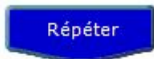
Le traitement de l'interruption est déclenché par l'appui du bouton-poussoir BP3, connecté à l'entrée 3 du microcontrôleur PICAXE.

Il interrompt immédiatement le programme principal pour exécuter le traitement d'interruption (Interrupt Bp3).

Exemple de câblage : 17\_08M2.DSN



### 3.3.5. Procédures > Répéter et Boucler

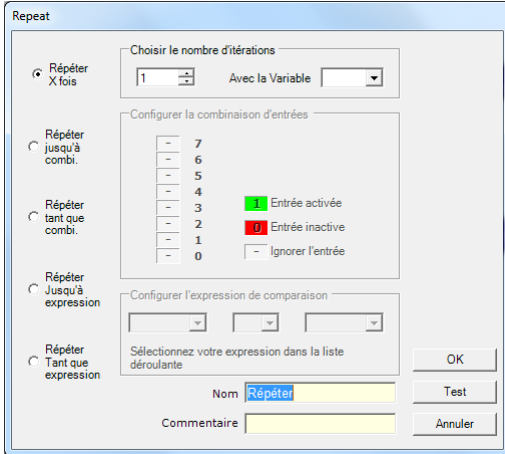


**Fonction :** permet de positionner une boucle dans un programme et gérer ainsi très facilement une répétition d'instructions.



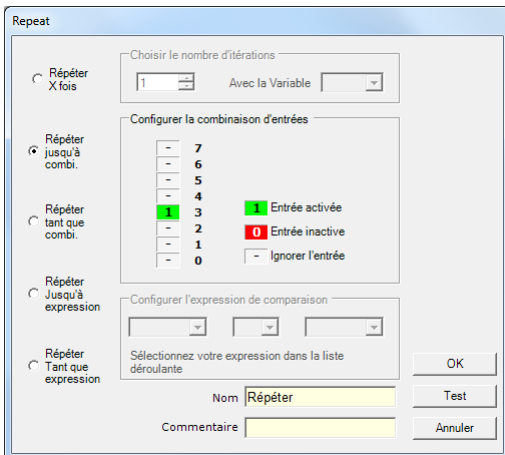
Plusieurs options sont disponibles :

- **Répéter X fois** pour répéter une instruction un certain nombre de fois.



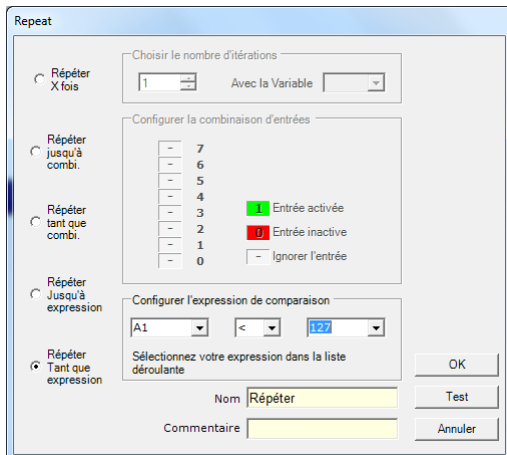
**Attention :** Ne pas utiliser plusieurs fois une même variable.

- **Répéter tant que** ou **jusqu'à** ce qu'une expression soit vérifiée.



Dans cet exemple, **répétition jusqu'à** ce que l'entrée 3 soit active. Il est possible de répéter la boucle avec l'option **tant que**.

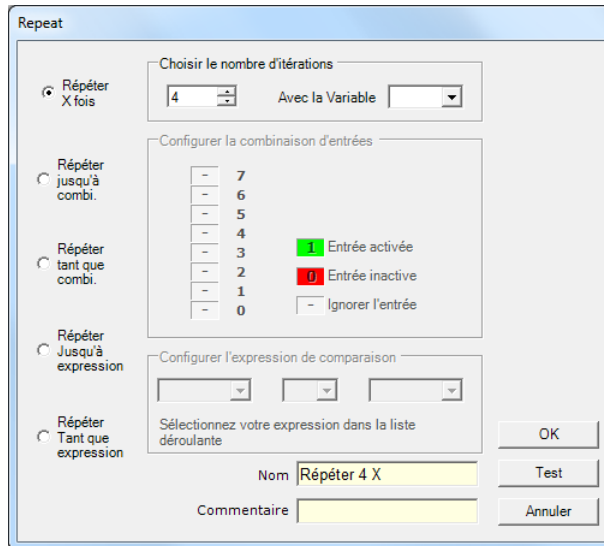
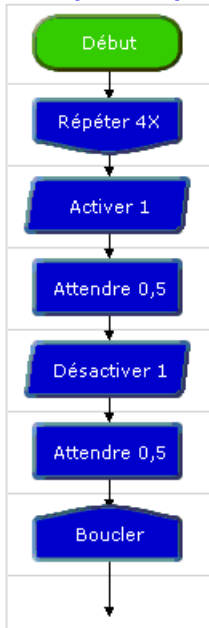
- **Répéter tant que** ou **jusqu'à ce que** la ou les entrées numériques prennent l'état souhaité (actif ou inactif).



Dans cet exemple, **répétition jusqu'à** ce que l'entrée A1 soit inférieure à 127. Il est possible de répéter la boucle avec l'option **tant que**.

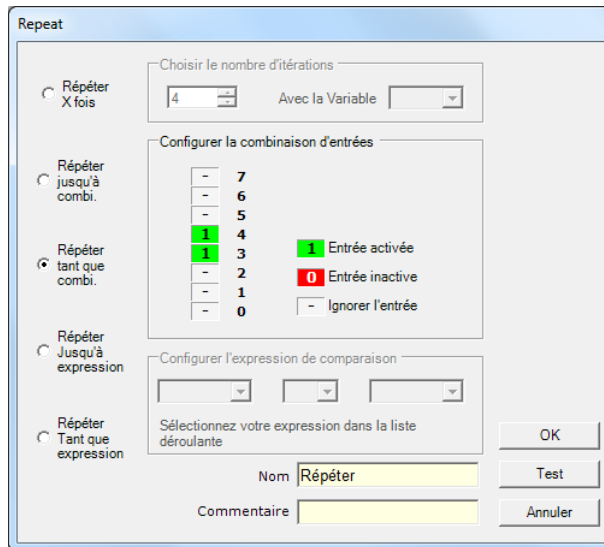
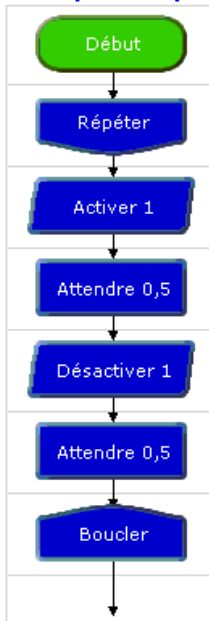


### Exemples de programme 1 :



Les instructions situées entre **Répéter** et **Boucler** sont réalisées 4 fois. Ensuite, le programme continue.

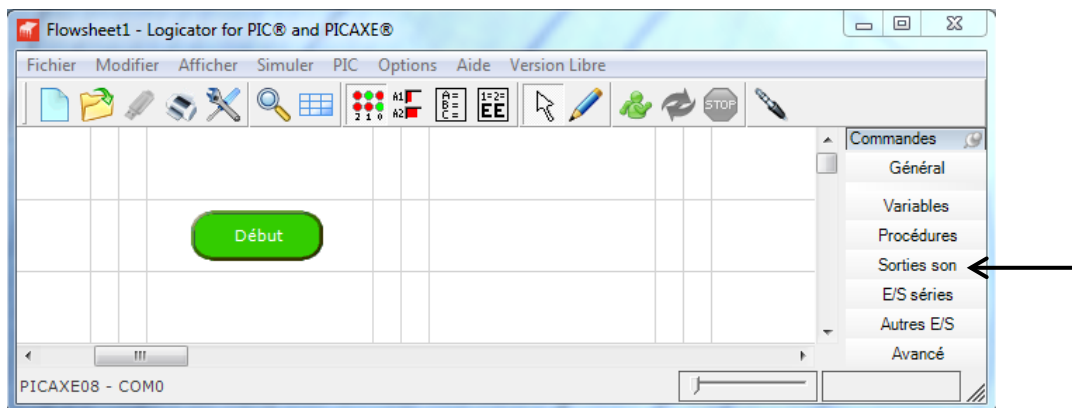
### Exemples de programme 2 :



Les instructions situées entre **Répéter** et **Boucler** sont réalisées indéfiniment tant que l'entrée 3 ET l'entrée 4 sont à 1.



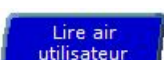


### 3.4. Sorties son

A partir de la barre des commandes, cliquer sur **Sorties son**.



Ce groupe donne accès aux instructions dédiées aux sons.

L'utilisation de la plupart de ces fonctions nécessite la présence d'un buzzer piézoélectrique ou d'un amplificateur sur la sortie sélectionnée.

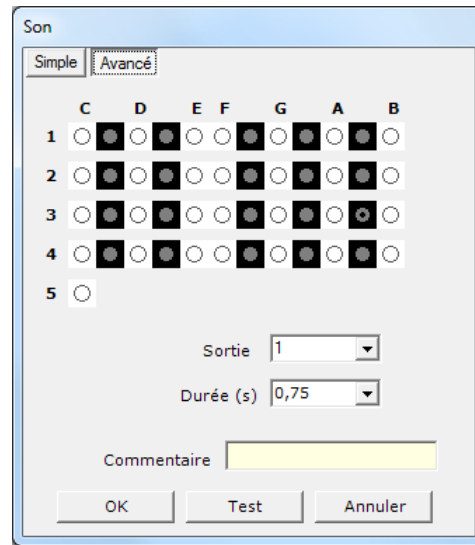
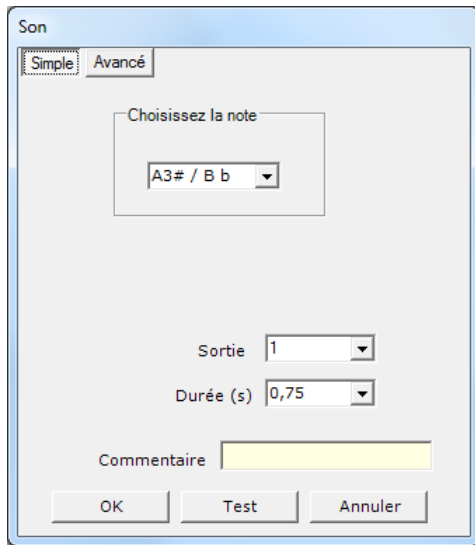
* 	Permet de produire une note de musique sur la sortie sélectionnée. Il est possible d'agir sur la fréquence et sur la durée de la note.
* 	Permet de jouer un air de musique parmi 4 mélodies et de faire clignoter des LED.
	Permet de jouer des airs de musique. Le choix de l'air s'effectue à partir d'une banque de données ou est réalisé par l'écriture dans un fichier texte par une suite de notes.
	S'utilise avec le module VMusic2 et permet de contrôler le module VMusic pour déclencher la lecture, agir sur le volume, changer de plage...
	S'utilise avec un module iPod et permet d'envoyer des instructions pour déclencher la lecture, agir sur le volume, changer de plage...

\* Instructions disponibles dans la version Logicator INITIAL.

### 3.4.1. Sorties son > Son

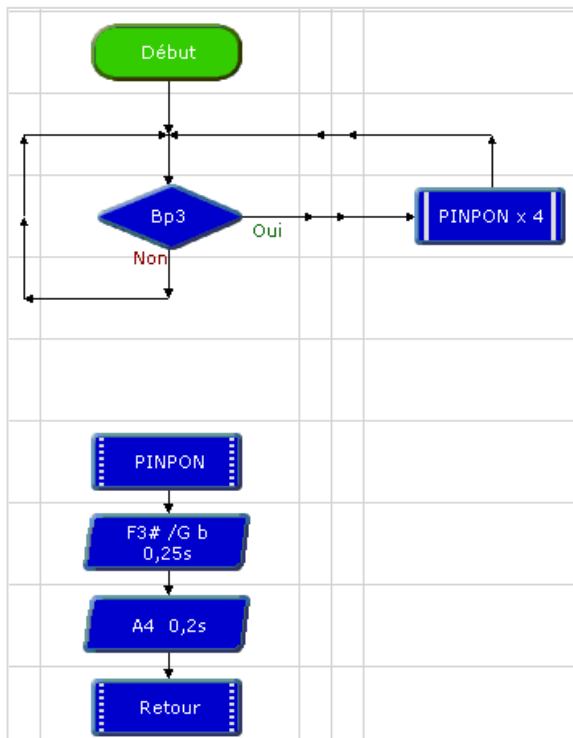


**Fonction :** Permet de produire une note de musique sur la sortie sélectionnée. Il est possible d'agir sur la fréquence et sur la durée de la note. Il est possible d'utiliser le mode **Simple** ou le mode **Avancé**, le résultat est identique.



Il faut choisir la fréquence, la sortie et la durée de la note. Le bouton **Test** permet de tester la note émise.

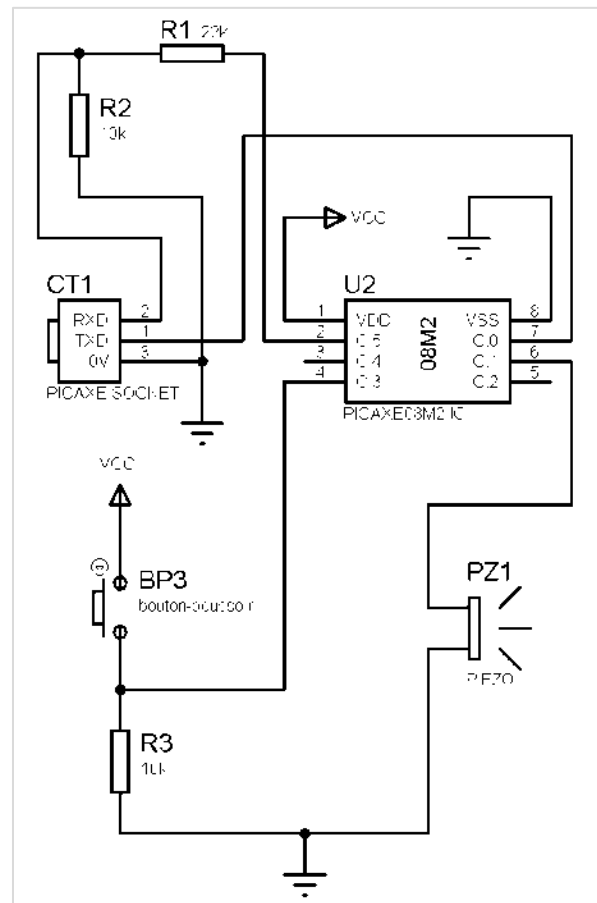
**Programme :** 18Son.plf



**Sirène :**

Dès l'appui de Bp3, la procédure PINPON est appelée 4 fois pour émettre un son de sirène.

**Exemple de câblage :** 18\_08M2.DSN

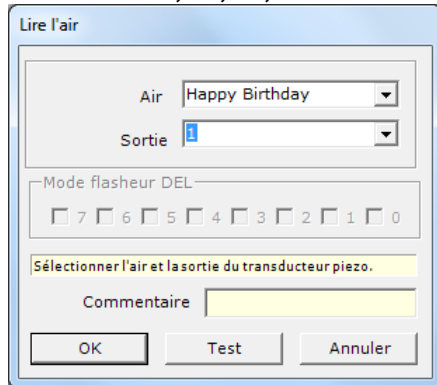


### 3.4.2. Sorties son > Lire l'air



**Fonction :** Permet de jouer un air de musique parmi 4 mélodies et de faire clignoter des LED.  
En fonction du microcontrôleur PICAXE sélectionné les options disponibles varient.

**Avec un microcontrôleur PICAXE des séries 14, 18, 20, 28 et 40.**

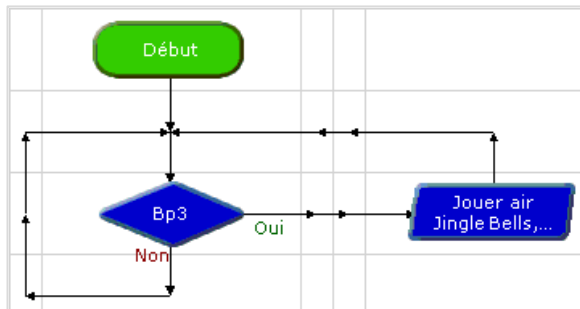


**Air** Choix de l'air de musique.

**Sortie** Choix de la sortie sur laquelle est reliée le buzzer piézoélectrique.

*Cette zone est affichée pour les microcontrôleurs PICAXE de la série 14, 18, 20, 28 et 40.*

**Programme :** 19LireAir.plf



L'appui de Bp3 déclenche la lecture de l'air de musique « Jingle Bells » et provoque le clignotement de la LED D0.

**Avec un microcontrôleur PICAXE 08M et 08M2.**

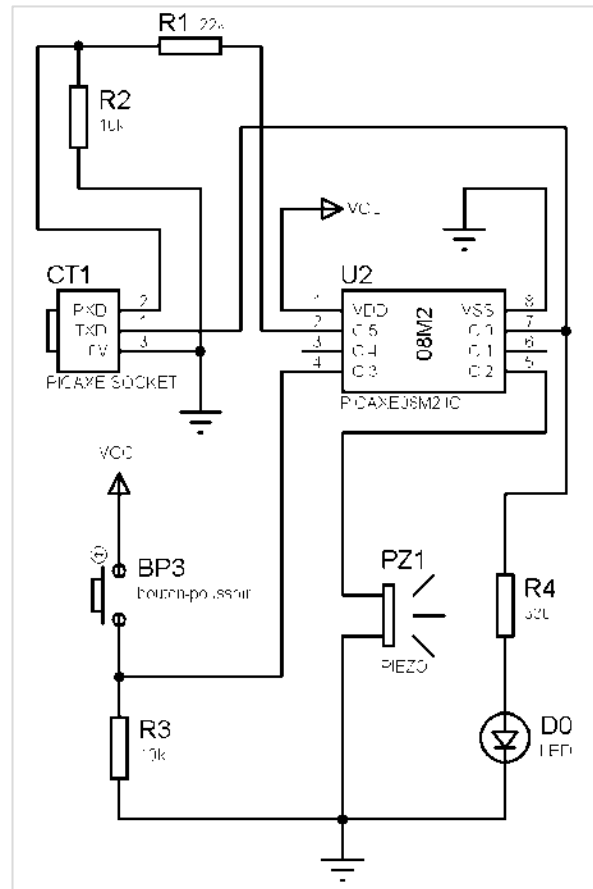


**Air** Choix de l'air de musique.

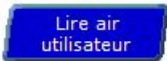
**Mode flasheur DEL** Permet de faire alterner l'état de la sortie 0 en vue de faire clignoter une LED.

*Cette zone est affichée pour les microcontrôleurs PICAXE 08M et 08M2.*

**Exemple de câblage :** 19\_08M2.plf



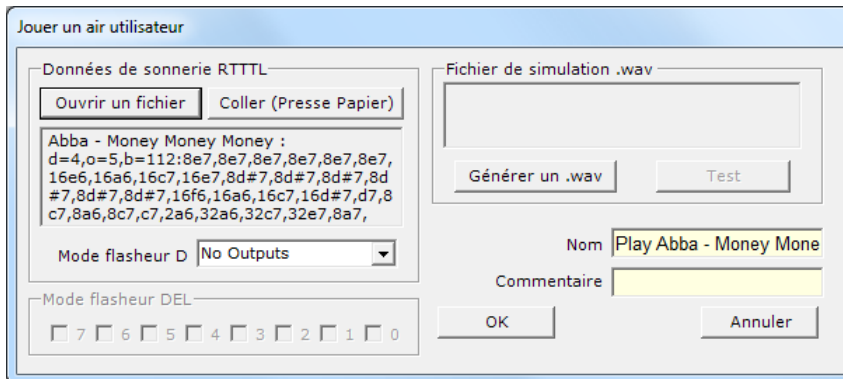
### 3.4.3. Sorties son > Lire air utilisateur



**Fonction :** Permet de jouer des airs de musique.

Le choix de l'air s'effectue à partir d'une banque de données ou est réalisé par l'écriture dans un fichier texte par une suite de notes.

De nombreux exemples existent dans le dossier RTTTL dans l'installation de Logicator.

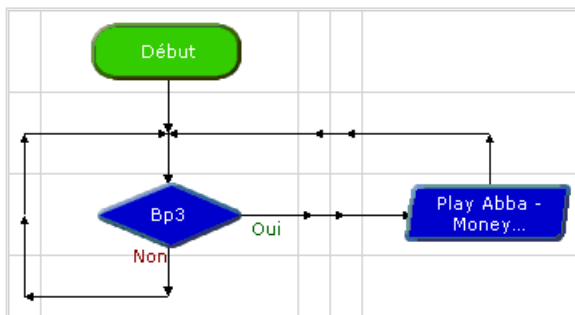


**Ouvrir un fichier** permet de sélectionner l'air de musique.

**Générer un .wav** permet de créer un fichier son pour écouter l'air sélectionné. la génération permet de tester facilement l'air sélectionné.

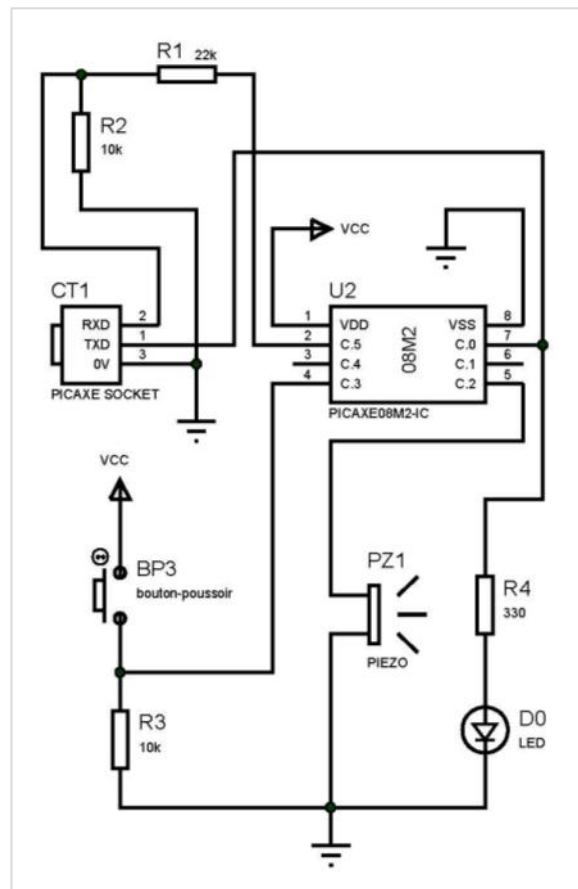
**Mode flasheur D** Cette zone est uniquement affichée pour les microcontrôleurs PICAXE 08M et 08M2.

**Programme :** 20LireAirUtilisateur.plf



Dès l'appui de Bp3, l'air « Money » de Abba est joué.

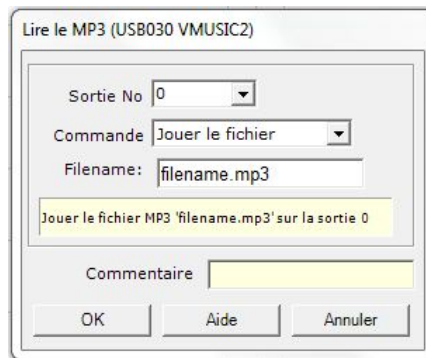
**Exemple de câblage :** 20\_08M2.plf



### 3.4.4. Sorties son > Lire MP3 ou lire iPod

**Fonction :** Ces deux instructions nécessitent des accessoires particuliers : le module Vmusic2 ou un iPod. Nous ne détaillerons pas ces instructions qui font l'objet de documents spécifiques.

Lire MP3

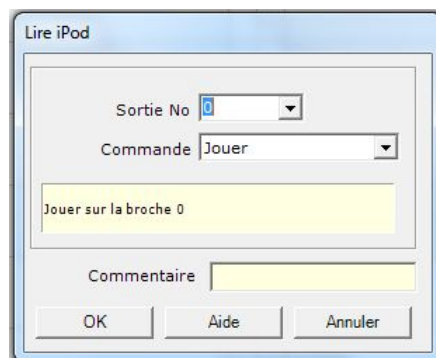


Module VMusic2. Réf : RAX-USB030

L'instruction **Lire MP3** s'utilise avec le module VMusic2 et permet de lire des fichiers musique au format MP3 situés sur une clé USB.

L'instruction **Lire MP3** permet de contrôler le module Vmusic2 pour déclencher la lecture, agir sur le volume, changer de page...

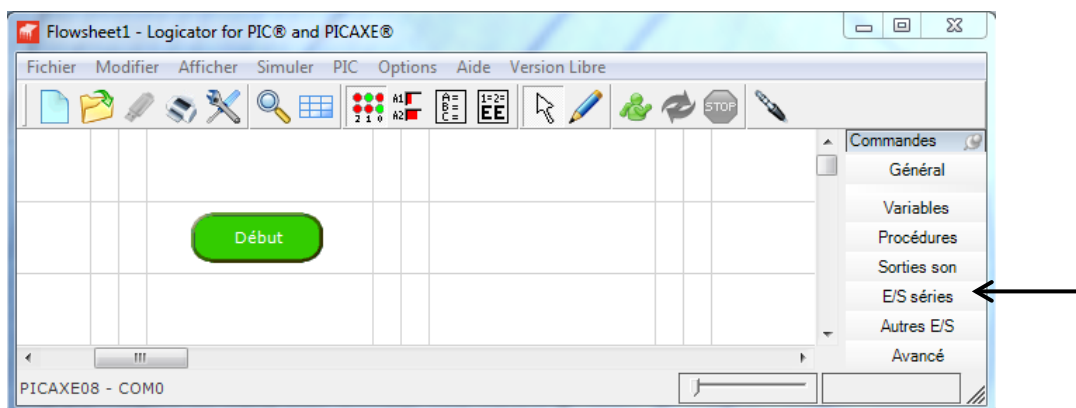
Lire iPod



L'instruction **Lire iPod** s'utilise avec un module iPod et permet d'envoyer des instructions pour déclencher la lecture, agir sur le volume, changer de page...


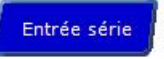



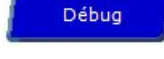
### 3.5. E/S séries

A partir de la barre des commandes, cliquer sur **E/S Séries**.



Ce groupe donne accès aux instructions utilisant les entrées et sorties du microcontrôleur PICAXE.

**Note** : ces instructions dépendent de la vitesse de fonctionnement du microcontrôleur PICAXE. Par défaut, les microcontrôleurs PICAXE de la série X2 sont cadencés à 8 MHz, les autres séries sont cadencées à 4 MHz.

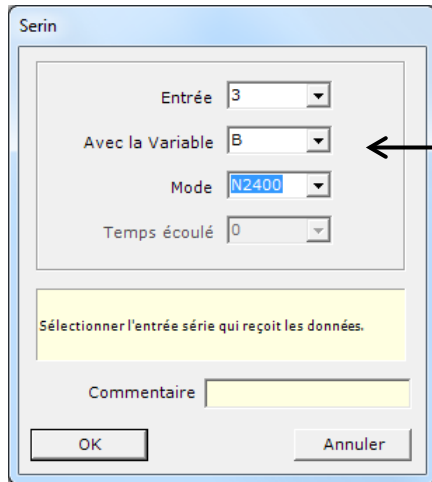
	Permet de lire une donnée série sur une entrée numérique du microcontrôleur PICAXE.
	Permet de lire une donnée série sur l'entrée Rxd du microcontrôleur PICAXE. <b>Note</b> : l'entrée série Rxd correspond à la liaison jack de programmation.
* 	Permet d'envoyer une donnée série sur une sortie numérique du microcontrôleur PICAXE.
* 	Permet d'envoyer une donnée série sur la sortie Txd du microcontrôleur PICAXE. <b>Note</b> : la sortie Txd correspond à la liaison jack de programmation.
* 	Permet d'afficher en direct les variables du microcontrôleur PICAXE sur une fenêtre du PC. <b>Note</b> : cette instruction nécessite la liaison du module PICAXE avec le PC via le câble de programmation AXE027.
* 	Permet d'envoyer un message alphanumérique vers un afficheur PICAXE LCD ou OLED.

\* Instructions disponibles dans la version Logicator INITIAL.

### 3.5.1. E/S Séries > Lire



**Fonction :** Permet de lire une donnée série sur une entrée numérique du microcontrôleur PICAXE.

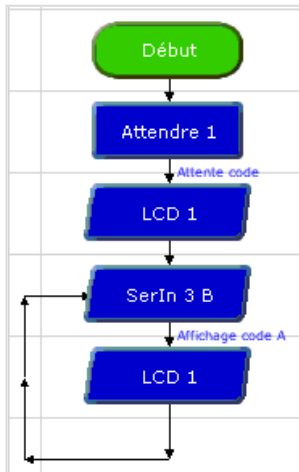


Il faut choisir l'entrée, la variable dans laquelle sera stockée la donnée reçue et définir le mode et la vitesse de transmission (T ou N).

**Note :** la vitesse de transmission (par exemple 2400 ou 4800 bauds) est conditionnée par la cadence de fonctionnement du microcontrôleur PICAXE.

**ATTENTION !** L'instruction **Lire** bloque l'exécution du programme en attente de la donnée à lire.

**Programme :** 21Lire.pf

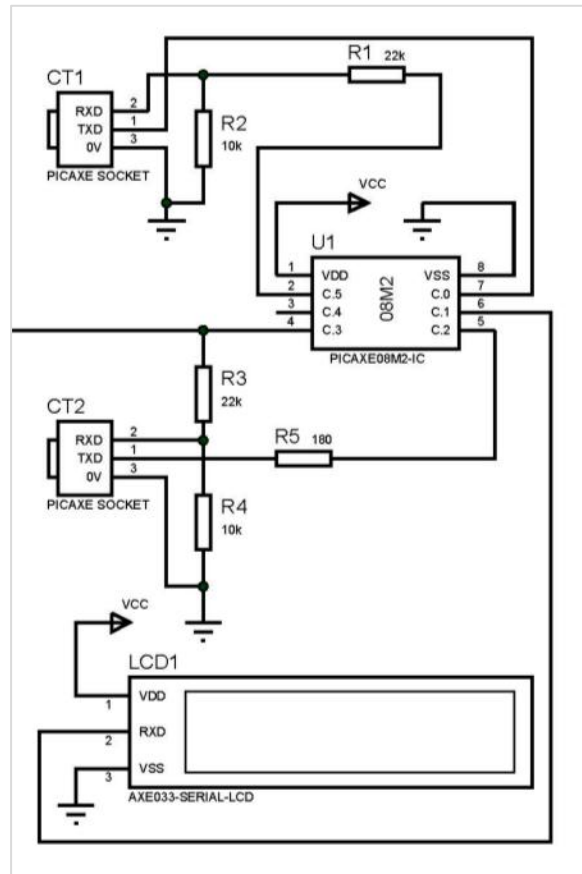


La temporisation d'une seconde permet au LCD de se configurer correctement.

Le jack CT2 permet de brancher un câble de programmation AXE027 sur le PC.

La donnée envoyée par le PC arrive sur l'entrée 3 et s'affiche sur le LCD.

**Exemple de câblage :** 21\_08M2.pf





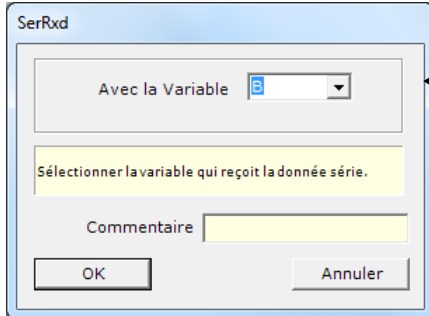
### 3.5.2. E/S Séries > Entrée Série



**Fonction :** Permet de lire une donnée série sur l'entrée Rxd du microcontrôleur PICAXE.

**ATTENTION !** L'instruction **Entrée série** bloque l'exécution du programme en attente de la donnée à lire.

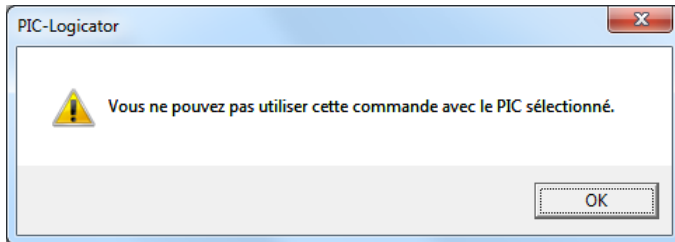
**Note :** l'entrée série Rxd correspond à la liaison jack de programmation.



La donnée reçue est stockée dans la variable B.

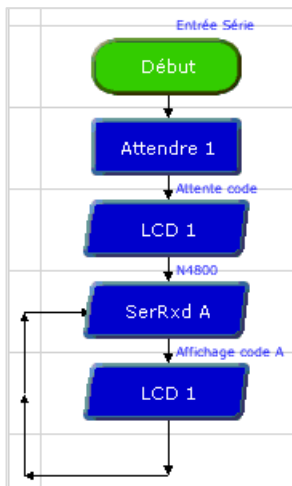
**Note :** la vitesse de transmission (par exemple 2400 ou 4800 bauds) est conditionnée par la cadence de fonctionnement du microcontrôleur PICAXE.

Certains microcontrôleurs PICAXE n'acceptent pas l'utilisation de l'instruction **Entrée Série**. Le message suivant apparaît.



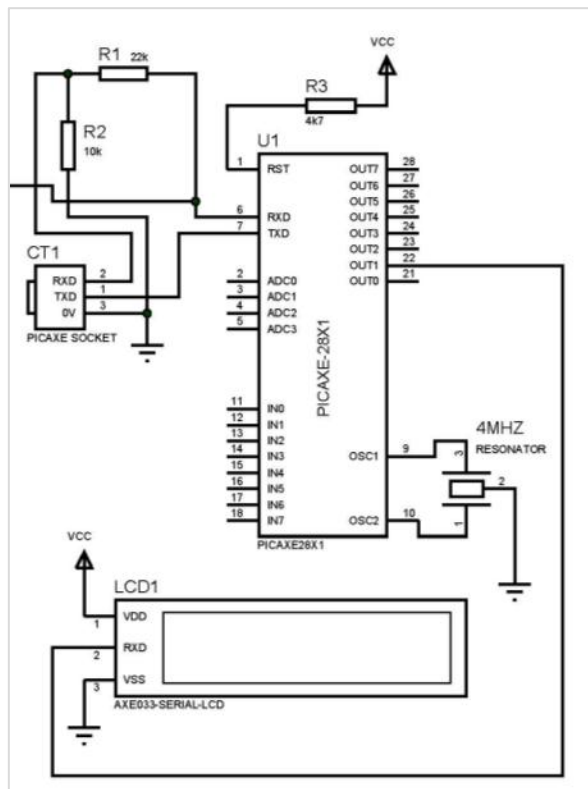
**Programme :** 22EntreeSerie.plf

**Exemple de câblage :** 22\_28X1.DSN

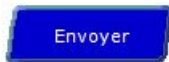


La temporisation d'une seconde permet au LCD de se configurer correctement.

La donnée envoyée par le PC arrive sur l'entrée Rxd du PICAXE et s'affiche sur le LCD.



### 3.5.3. E/S Série > Envoyer



**Fonction :** Permet d'envoyer une donnée série sur une sortie numérique du microcontrôleur PICAXE.

The SerOut dialog box contains the following fields and controls:

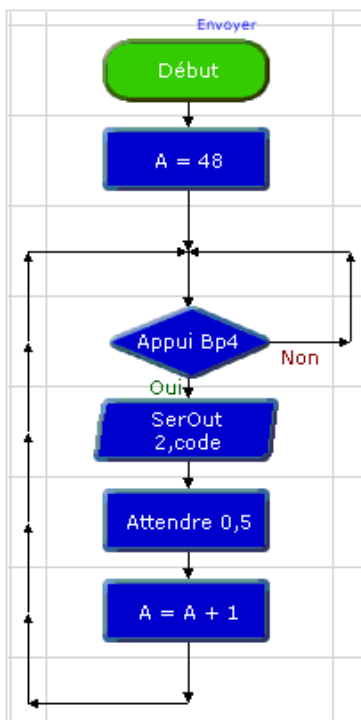
- Sortie No: 2 (dropdown)
- Donnée: code (text input), with a checked ASCII checkbox
- Donnée: [A] (text input), with a checked ASCII checkbox
- Donnée: (empty text input), with a checked ASCII checkbox
- Mode: N2400 (dropdown)
- Commentaire: (text input)
- Buttons: OK and Annuler

Below the input fields, there is a note: "Pour les variables, utiliser des crochets en mode ASCII (ex. [A])".

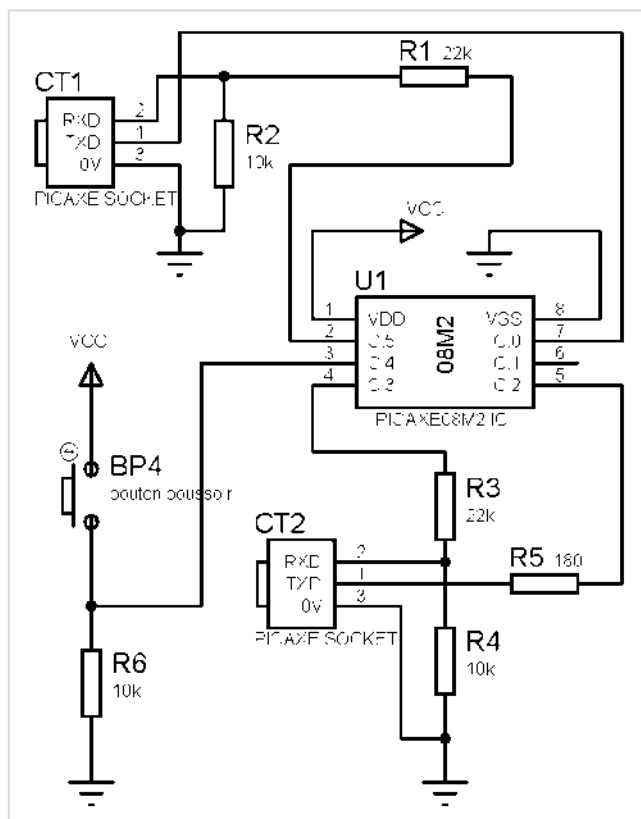
Il faut choisir la sortie, renseigner la donnée (code) et la variable (A) à envoyer, puis choisir le mode (T ou N) et la vitesse de transmission.

**Note :** la vitesse de transmission (par exemple 2400 ou 4800 bauds) est conditionnée par la cadence de fonctionnement du microcontrôleur PICAXE.

**Programme :** 23Envoyer.plf



**Exemple de câblage :** 23\_08M2.DSN



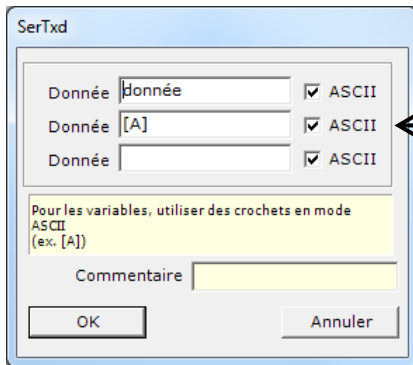
Le jack CT2 permet de brancher un câble de programmation AXE027 sur le PC. La donnée est envoyée par le microcontrôleur PICAXE sur la sortie 2 à chaque appui de Bp4. Dans cet exemple, le microcontrôleur PICAXE envoie le message « code » suivi de la valeur de la variable A.

### 3.5.4. E/S Séries > Sortie Série

Sortie série

**Fonction :** Permet d'envoyer une donnée série sur la sortie Txd du microcontrôleur PICAXE.

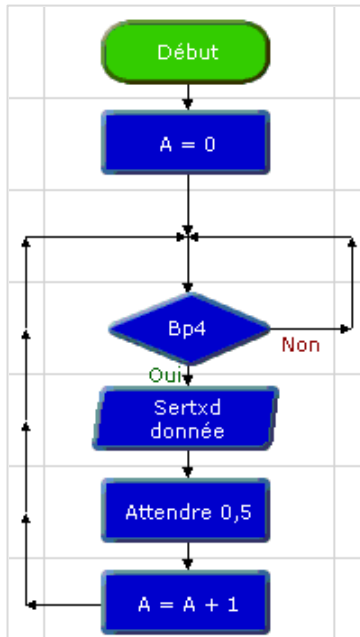
**Note :** la sortie Txd correspond à la liaison jack de programmation.



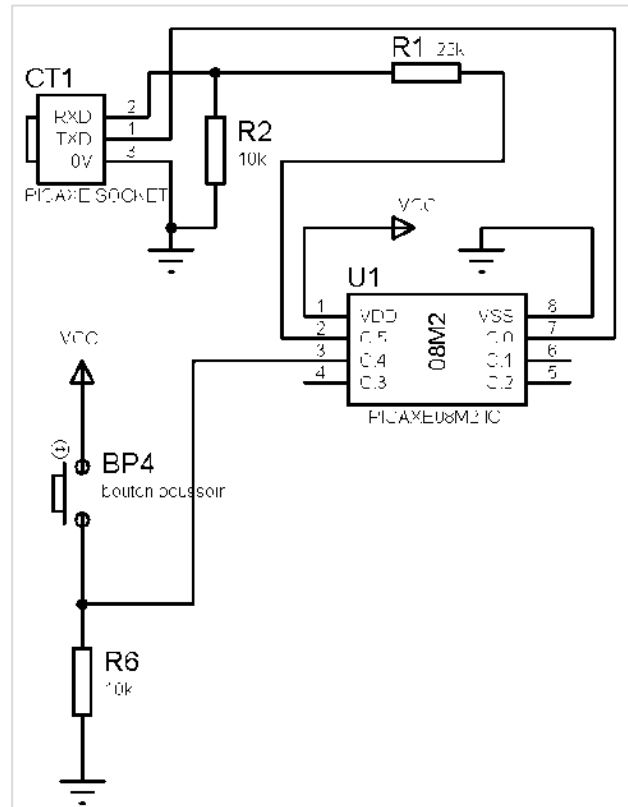
Il faut renseigner la donnée à envoyer (donnée) et la variable.

**Note :** la vitesse de transmission (par exemple 2400 ou 4800 bauds) est conditionnée par la cadence de fonctionnement du microcontrôleur PICAXE.

**Programme :** 24SortieSerie.plf



**Exemple de câblage :** 24\_08M2.DSN



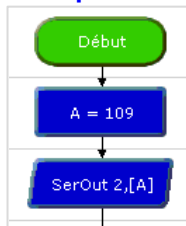
La donnée est envoyée par le microcontrôleur PICAXE sur la sortie Txd à chaque appui de Bp4. Dans cet exemple, le microcontrôleur PICAXE envoie le message « donnée » suivi de la valeur de la variable A qui est incrémentée à chaque appui.

### 3.5.5. Complément d'information sur le code ASCII

Le code ASCII définit une correspondance entre la valeur décimale (ou hexadécimale) et un caractère ou un contrôle (un retour chariot par exemple).

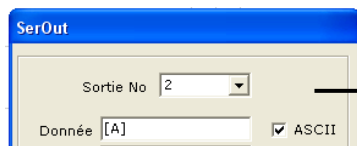
Décimal	Hexa.	ASCII	Décimal	Hexa.	ASCII	Décimal	Hexa.	ASCII	Décimal	Hexa.	ASCII
0	0	NUL	32	20	SP	64	40	@	96	60	`
1	1	SOH	33	21	!	65	41	A	97	61	a
2	2	STX	34	22	"	66	42	B	98	62	b
3	3	ETX	35	23	#	67	43	C	99	63	c
4	4	EOT	36	24	\$	68	44	D	100	64	d
5	5	ENQ	37	25	%	69	45	E	101	65	e
6	6	ACK	38	26	&	70	46	F	102	66	f
7	7	BEL	39	27	'	71	47	G	103	67	g
8	8	BS	40	28	(	72	48	H	104	68	h
9	9	HT	41	29	)	73	49	I	105	69	i
10	0A	LF	42	2A	*	74	4A	J	106	6A	j
11	0B	VT	43	2B	+	75	4B	K	107	6B	k
12	0C	FF	44	2C	,	76	4C	L	108	6C	l
13	0D	CR	45	2D	-	77	4D	M	109	6D	m
14	0E	SO	46	2E	.	78	4E	N	110	6E	n
15	0F	SI	47	2F	/	79	4F	O	111	6F	o
16	10	DLE	48	30	0	80	50	P	112	70	p
17	11	DC1	49	31	1	81	51	Q	113	71	q
18	12	DC2	50	32	2	82	52	R	114	72	r
19	13	DC3	51	33	3	83	53	S	115	73	s
20	14	DC4	52	34	4	84	54	T	116	74	t
21	15	NAK	53	35	5	85	55	U	117	75	u
22	16	SYN	54	36	6	86	56	V	118	76	v
23	17	ETB	55	37	7	87	57	W	119	77	w
24	18	CAN	56	38	8	88	58	X	120	78	x
25	19	EM	57	39	9	89	59	Y	121	79	y
26	1A	SUB	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC	59	3B	;	91	5B	[	123	7B	{
28	1C	FS	60	3C	<	92	5C	\	124	7C	
29	1D	GS	61	3D	=	93	5D	]	125	7D	}
30	1E	RS	62	3E	>	94	5E	^	126	7E	~
31	1F	US	63	3F	?	95	5F	_	127	7F	DEL

#### Exemple d'instruction :

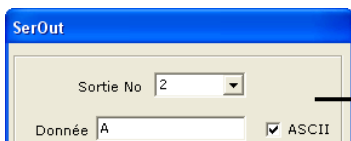
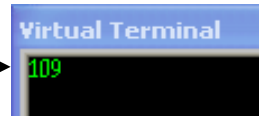


#### Information reçue :

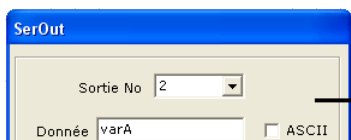
En positionnant A = 109



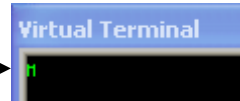
Cette instruction envoie la valeur 109 :



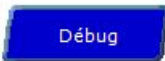
Cette instruction envoie A :



Cette instruction envoie M :

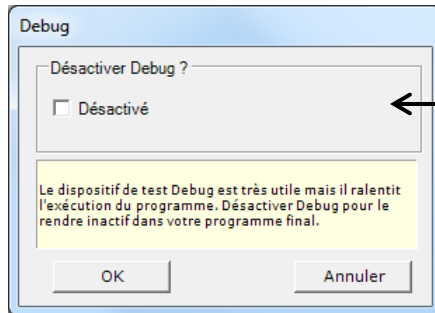


### 3.5.6. E/S Séries > Débug



**Fonction :** Permet d'afficher en direct les variables du microcontrôleur PICAXE sur une fenêtre du PC.

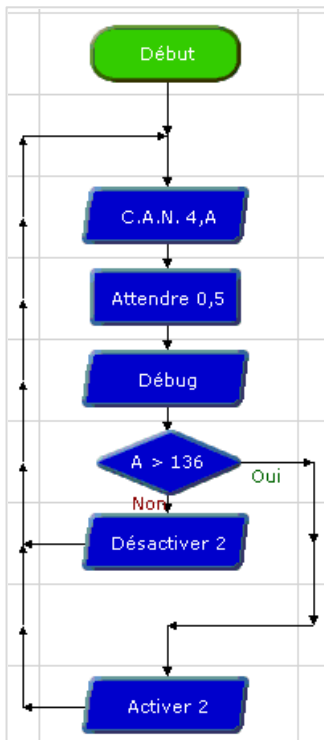
**Note :** cette instruction nécessite la liaison du module PICAXE avec le PC via le câble de programmation AXE027.



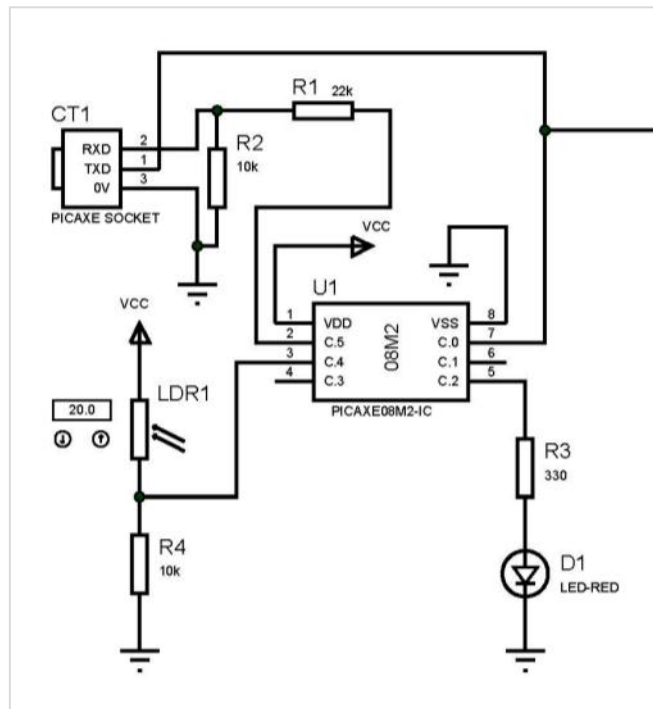
Désactivation possible.

L'utilisation de l'instruction **Débug** ralentit considérablement le programme et ne doit être utilisée que pour la mise au point.

**Programme :** 25Debug.plf



**Exemple de câblage :** 25\_08M2.DSN



Le couple LDR1 et R4 fournit sur l'entrée 4 du microcontrôleur PICAXE une tension proportionnelle à l'intensité lumineuse.

Cette valeur analogique est convertie en une donnée numérique par l'instruction CAN4.

L'instruction **Débug** permet d'afficher le résultat de la conversion stocké dans la variable A.

Ceci permet de suivre en direct l'évolution de l'intensité et de vérifier si elle dépasse ou non la valeur 136.

(CAN : convertisseur analogique numérique)

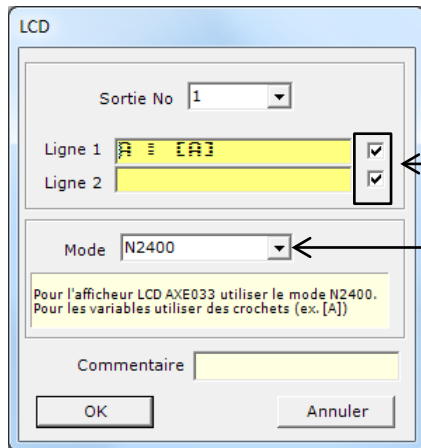


La fenêtre de débogage permet d'afficher les variables A à T.

### 3.5.7. E/S Séries > LCD



**Fonction :** Permet d'envoyer un message alphanumérique vers un afficheur PICAXE LCD ou OLED.



Permet de sélectionner les lignes à afficher.

Le mode de transmission N2400 est impératif pour l'afficheur LCD.

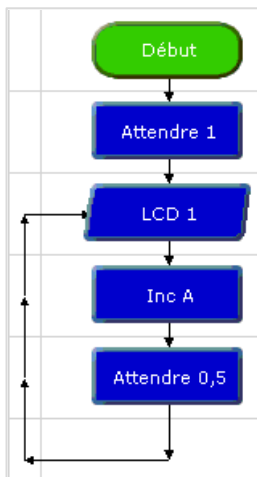
**Sortie No** Sortie sur laquelle est connectée l'entrée de l'afficheur PICAXE LCD ou OLED.

**Ligne 1 - 2** Texte alphanumérique à envoyer vers l'afficheur.

Possibilité d'afficher la valeur d'une variable en utilisant des crochets [A].

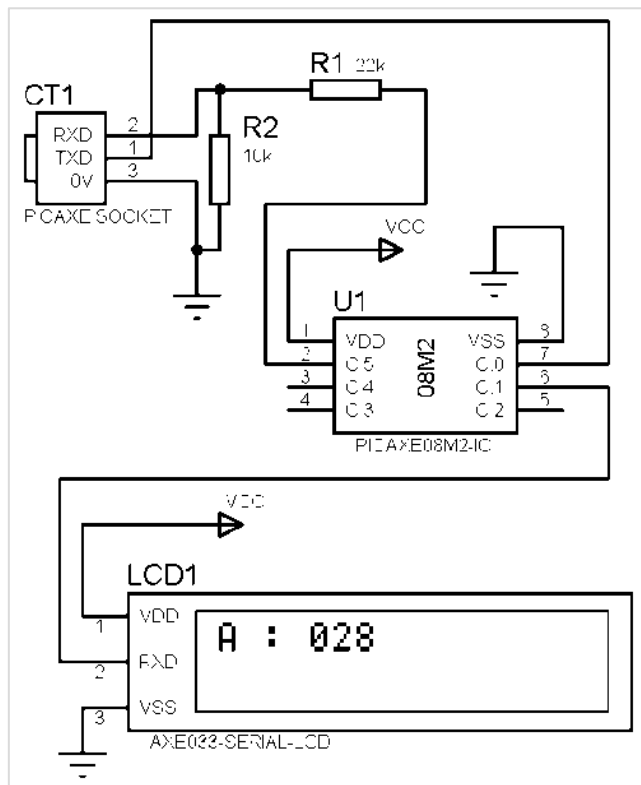
**Programme :** 26LCD.pf

**Exemple de câblage :** 26\_08M2.DSN



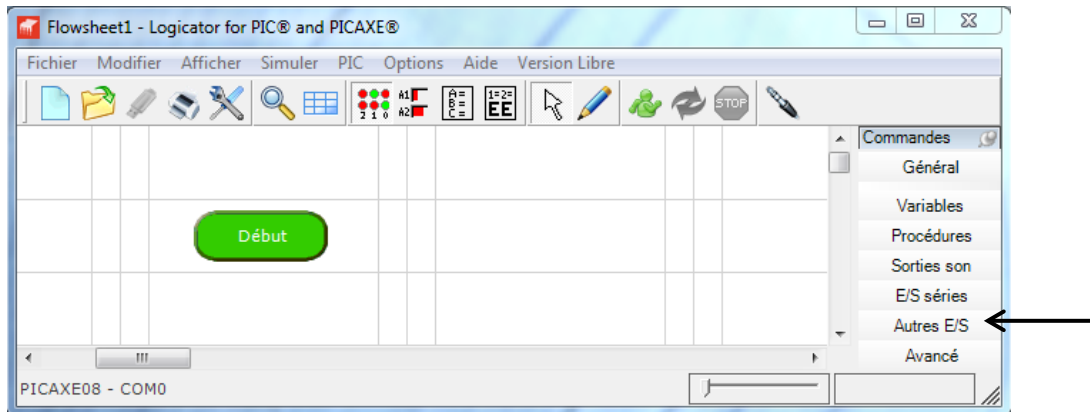
La variable A s'affiche et s'incrémente toutes les 0,5 secondes.

**Remarque :** pour permettre à l'afficheur de se configurer correctement, il est conseillé de positionner une attente avant d'effectuer une instruction **LCD**.

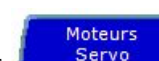
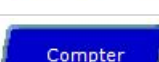


### 3.6. Autres E/S

A partir de la barre des commandes, cliquer sur **Autres E/S**.

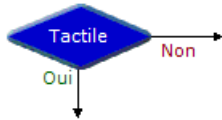


Ce groupe donne accès aux instructions avancées utilisant les entrées et sorties du PICAXE.

	Permet de lire un capteur tactile sur une entrée spécifique (de type <i>Touch</i> ) du microcontrôleur PICAXE et de comparer la valeur lue par rapport à une plage préalablement définie.
	Permet de mesurer la durée d'une impulsion et de stocker la valeur dans une variable par pas de 10µs.
	Permet de générer sur une sortie du microcontrôleur PICAXE une impulsion d'une durée prédéfinie entre 10 µs et 655350 µs par pas de 10 µs.
* 	Permet d'inverser l'état d'une sortie. Si la sortie sélectionnée est à l'état haut, elle passe à l'état bas, et inversement.
* 	Permet de piloter un servomoteur.
* 	Permet d'agir simultanément sur deux sorties du microcontrôleur PICAXE en vue de piloter deux servomoteurs à rotation continue destinés à animer un robot.
	Permet de compter le nombre d'impulsions pendant un temps donné.
* 	Permet de stocker dans une variable une donnée acquise par un capteur infrarouge (IR).
	Permet d'émettre une information véhiculée par un signal infrarouge (IR) selon le protocole Sony avec une modulation à 38 KHz.
	Permet de stocker dans une variable la température acquise par un capteur numérique DS18B20.
	Permet d'effectuer la conversion d'une donnée analogique (niveau de tension compris entre 0 et 5 V) en une valeur numérique sur une échelle décimale allant de 0 à 255.
	Permet de stocker dans une variable la distance qui sépare un capteur à ultrason de type SRF05 et un obstacle.
	Permet de générer un signal MLI (Modulation en Largeur d'Impulsion) pendant un nombre de cycles donné, afin de produire un signal sur la sortie 1,2 ou 4 d'un microcontrôleur PICAXE de la série 08 ou 08M.
	Permet de générer un signal MLI (Modulation en Largeur d'Impulsion) continu sur une sortie PWM d'un microcontrôleur PICAXE.

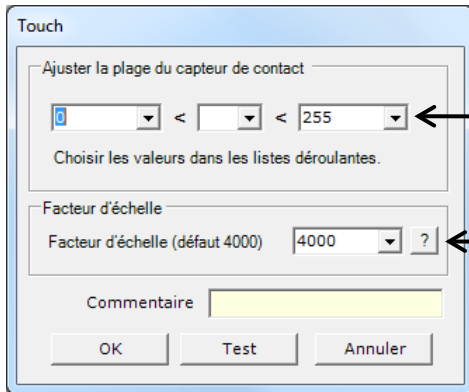
\* Instructions disponibles dans la version Logicator INITIAL.

### 3.6.1. Autres E/S > Tactile



**Fonction :** Permet de lire un capteur tactile sur une entrée spécifique (de type *Touch*) du microcontrôleur PICAXE et de comparer la valeur lue par rapport à une plage préalablement définie.

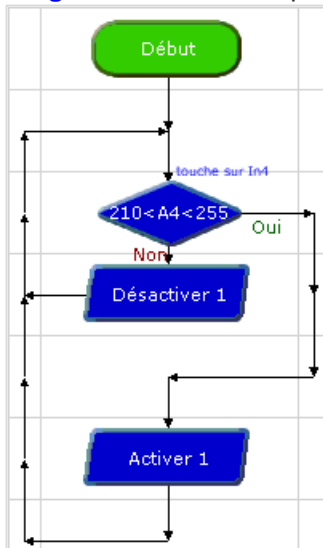
**Note :** la sensibilité du capteur tactile peut varier. La fonction « Calibrer un capteur analogique » permet de visualiser en direct la valeur retournée par le capteur afin d'ajuster la plage de détection souhaitée. Voir le chapitre correspondant à la fin de ce document.



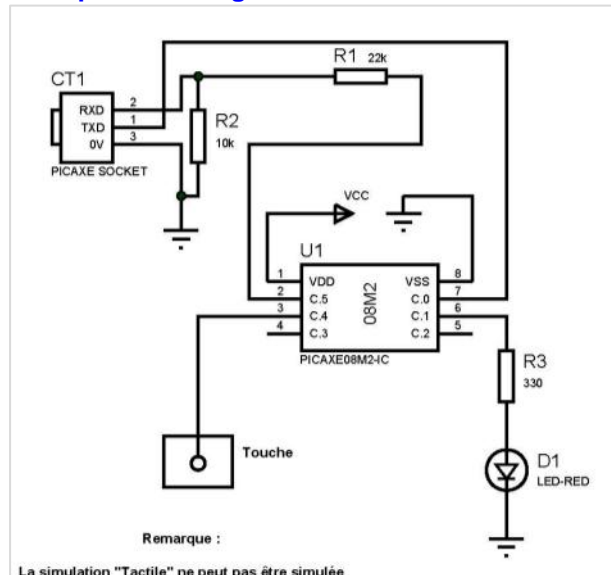
Choix de l'entrée et de la plage de détection.

Permet de s'adapter à la sensibilité du capteur pour les microcontrôleurs PICAXE de la série M2.

Programme : 27Tactile.plf



Exemple de câblage : 27\_08M2.DSN



**Utilisation d'une touche sensitive :**

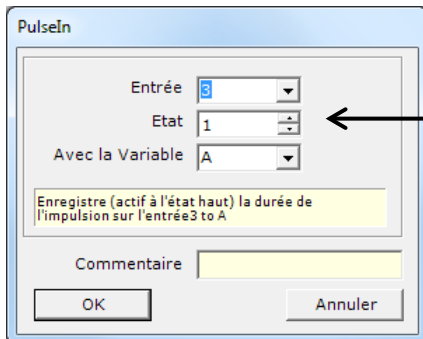
Dès le contact avec la touche sensitive, la variable A4 déclenche l'allumage de la LED D1.



### 3.6.2. Autres E/S > Mesurer impulsion

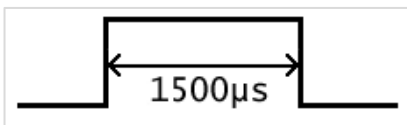
Mesurer impulsion

**Fonction :** Permet de mesurer la durée d'une impulsion et de stocker la valeur dans une variable par pas de 10  $\mu\text{s}$ .  
Plage de mesure : de 10  $\mu\text{s}$  à 2550  $\mu\text{s}$ .  
Au-delà de 2550  $\mu\text{s}$ , la valeur stockée est 0.

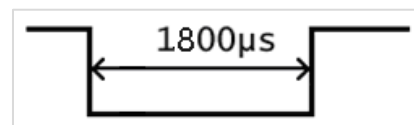


← **Etat 1 :** mesure impulsion à l'état haut.  
**Etat 0 :** mesure impulsion à l'état bas.

#### Exemples de mesure :



Impulsion positive  
A = 150

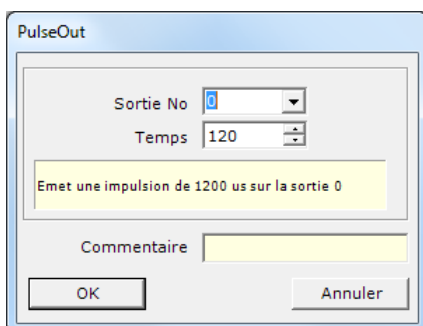


Impulsion négative  
A=180

### 3.6.3. Autres E/S > Générer impulsion

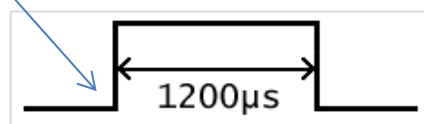
Générer impulsion

**Fonction :** Permet de générer sur une sortie du microcontrôleur PICAXE une impulsion d'une durée prédéfinie entre 10  $\mu\text{s}$  et 655350  $\mu\text{s}$  par pas de 10  $\mu\text{s}$ .  
Si la sortie est initialement à l'état bas, l'impulsion générée est à l'état haut, et inversement.



#### Exemple d'impulsion :

Générer impulsion



= Transition de l'état bas vers l'état haut pendant 1200  $\mu\text{s}$ .

### 3.6.4. Autres E/S > Basculer



**Fonction :** Permet d'invertir l'état d'une sortie.  
Si la sortie sélectionnée est à l'état haut, elle passe à l'état bas, et inversement.

Toggle (Basculer)

Sortie No :

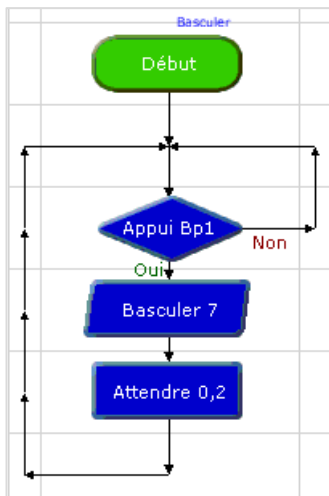
Sélectionner la sortie à faire basculer

Nom

Commentaire

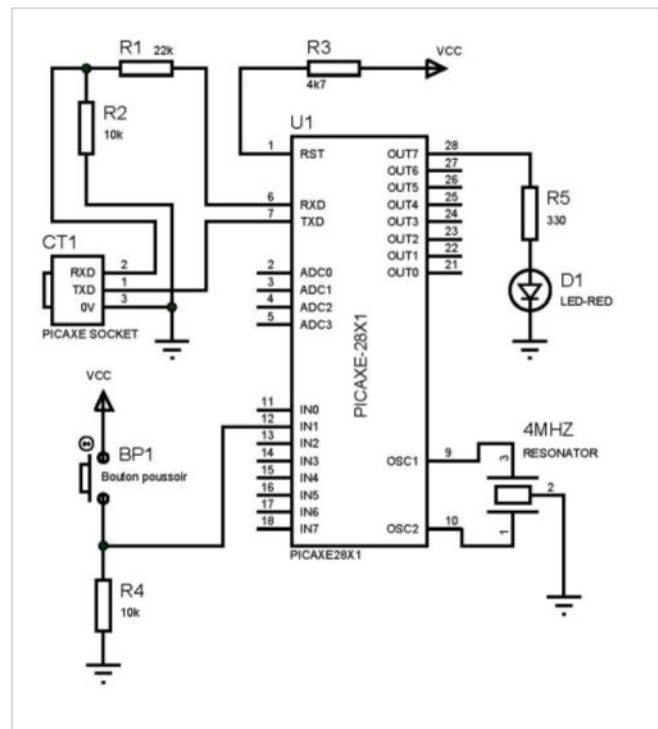
OK Test Annuler

**Programme :** 28Basculer.plf



**Clignotement :**  
Si on maintient le Bp1 appuyé, la sortie 7 change d'état toutes les 0,2 secondes.

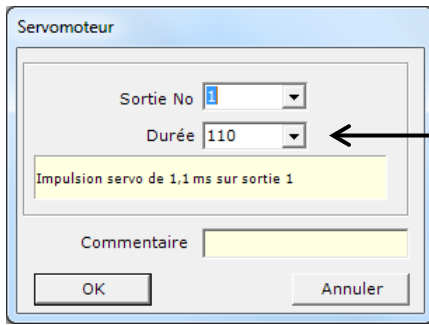
**Exemple de câblage :** 28\_08M2.DSN



### 3.6.5. Autres E/S > Servo



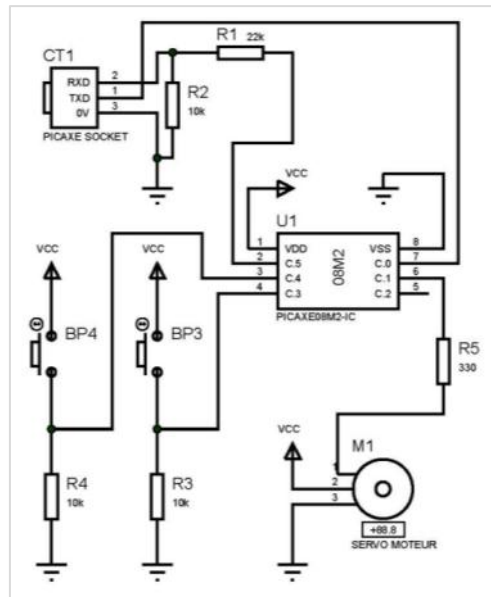
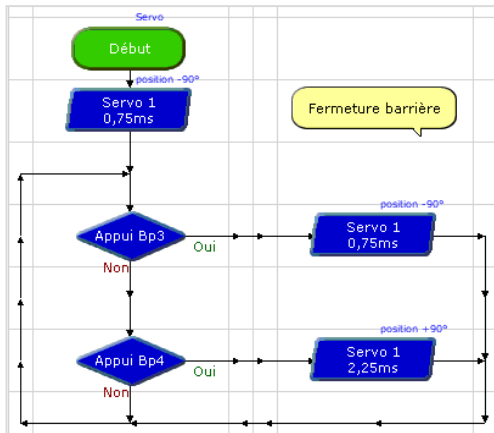
**Fonction :** Permet de piloter un servomoteur.



← Consigne de position angulaire (ou sens de rotation pour un servomoteur à rotation continue). La consigne peut également être définie par une variable.

**Programme :** 29Servo.plf

**Exemple de câblage :** 29\_08M2.DSN



**Contrôle d'une barrière :**  
Utilisation d'un servomoteur 180°.  
Appui du Bp3 ouvre la barrière.  
Appui du Bp4 ferme la barrière.

**Note :** L'utilisation de l'instruction **Servo** génère en continu des impulsions. La valeur de la consigne doit être comprise entre 75 et 225. Pour une même consigne, la position angulaire peut varier en fonction du servomoteur.

#### Servomoteurs simples, course limitée à 180° :

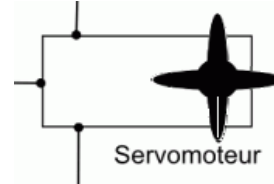
La consigne conditionne la position angulaire du servomoteur :



Durée = 75  
(Position ≈ 90°)

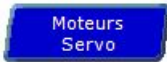


Durée = 150  
(Position ≈ 0°)

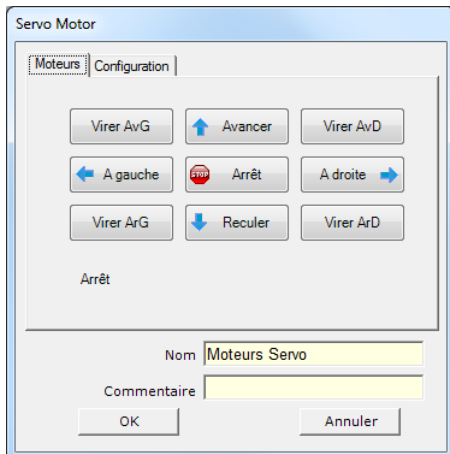


Durée = 225  
(Position ≈ -90°)

### 3.6.6. Autres E/S > Moteur Servo



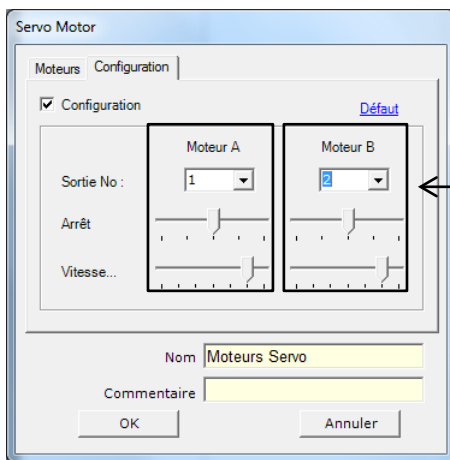
**Fonction :** Permet d'agir simultanément sur deux sorties du microcontrôleur PICAXE en vue de piloter deux servomoteurs à rotation continue destinés à animer un robot.



Les boutons **Avancer**, **Reculer**, **A gauche** et **A droite** commandent simultanément les 2 servomoteurs à rotation continue.

Les boutons **Virer AvG**, **Virer AvD**, **Virer ArG**, **Virer ArD** commandent 1 servomoteur à rotation continue à la fois.

**Note :** des tests doivent être effectués pour que le programme s'adapte à vos servomoteurs à rotation continue. Il est nécessaire de les configurer.



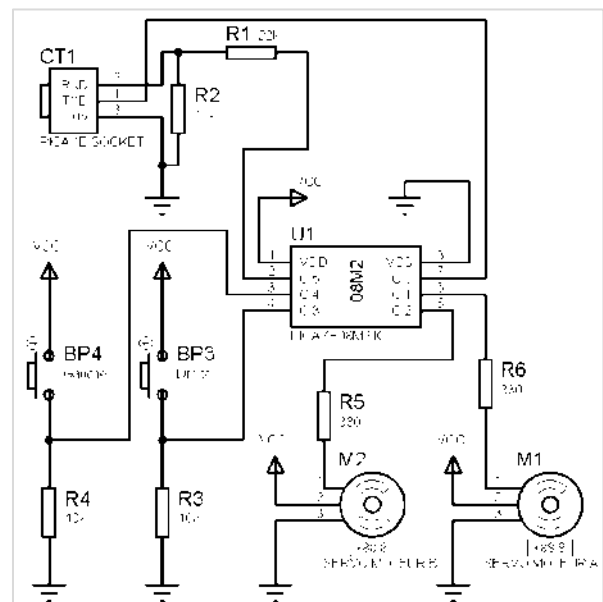
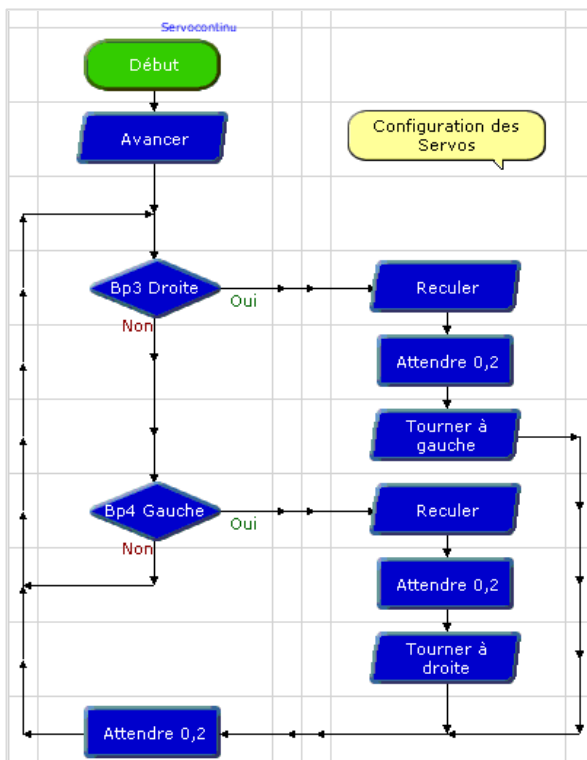
L'onglet **Configuration** permet de choisir les sorties utilisées et d'effectuer le calage des servomoteurs à rotation continue.

Paramétrage des servomoteurs A et B : choix de la sortie, réglage de l'arrêt et de la vitesse.

**Note :** le réglage **Arrêt** permet de s'adapter à chaque servomoteur à rotation continue. Le réglage est correct quand une demande d'arrêt moteur provoque un réel arrêt du Servomoteur.

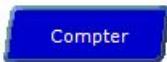
**Programme :** 30Servocontinu.pf

**Exemple de câblage :** 30\_08M2.DSN

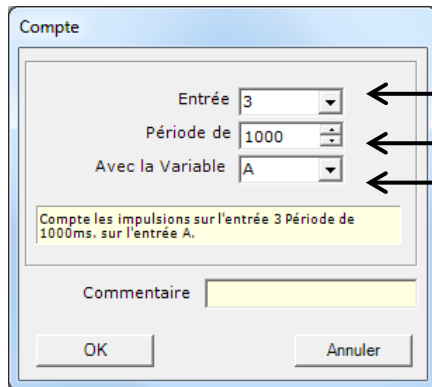


**Robot mobile avec détection droite et gauche :** Les détections des moustaches droite ou gauche permettent au robot mobile d'effectuer une manœuvre pour éviter l'obstacle.

### 3.6.7. Autres E/S > Compter



**Fonction :** Permet de compter le nombre d'impulsions pendant un temps donné. Le comptage d'impulsions pendant 1 seconde (1000 ms) permet d'obtenir facilement la fréquence d'un signal.

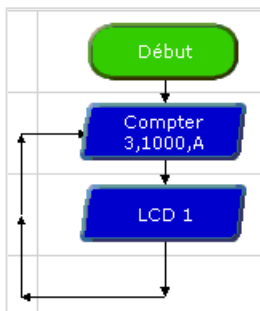


← Choix de l'entrée

← Période de mesure en ms.

← Choix de la variable dans laquelle on stocke l'information

**Programme :** 31compter.plf



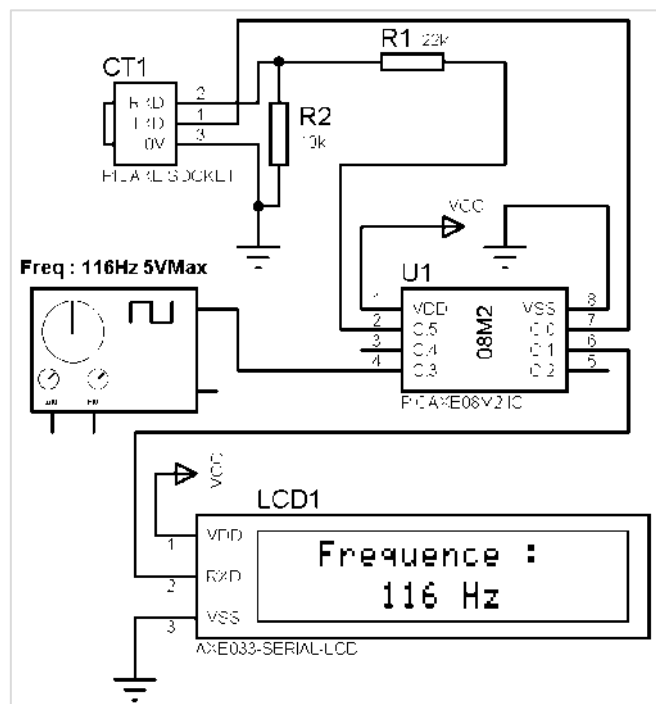
#### Mesure de fréquence :

Dans cet exemple, on compte le nombre d'impulsions générées par un signal périodique pendant 1 seconde.

La valeur stockée dans la variable A correspond à la fréquence de ce signal (exprimée en Hz).

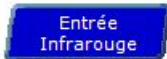
**Attention !** La variable A ne peut stocker des valeurs supérieures à 255, pour mesurer des fréquences supérieures à 255 Hz, il est nécessaire de modifier la période de mesure.

**Exemple de câblage :** 31\_08M2.DSN

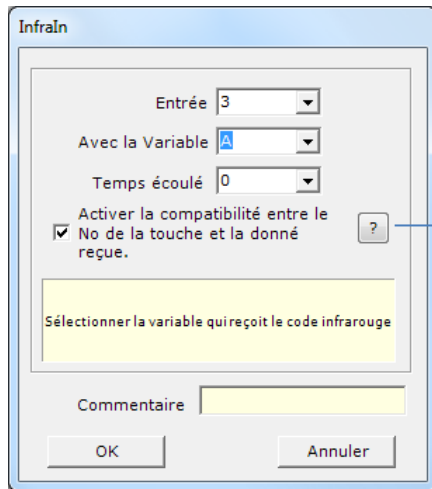


**Remarque :** Un fréquencesmètre fonctionnel doit être réalisé avec un étage d'entrée mettant en forme (carré) et limitant le signal à 5 V max sous peine de destruction du microcontrôleur PICAXE.

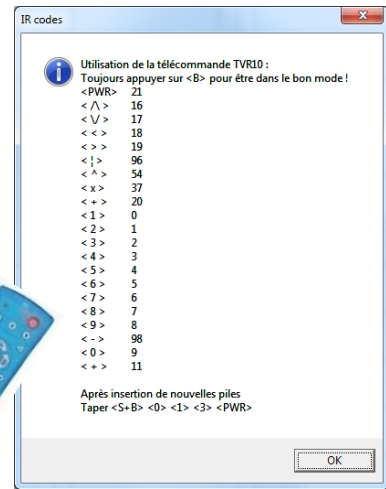
### 3.6.8. Autres E/S > Entrée Infrarouge



**Fonction :** Permet de stocker dans une variable une donnée acquise par un capteur infrarouge (IR).



Assure la cohérence entre la valeur stockée dans la variable et le numéro de la touche appuyée sur la télécommande PICAXE \* (Réf. RAX-TV010).



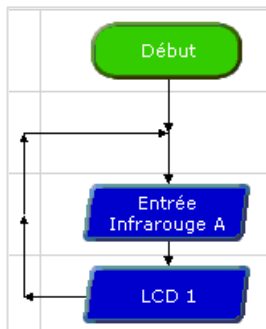
**Temps écoulé** Si un signal IR est détecté, l'instruction suivante est immédiatement exécutée.

Sinon, elle est exécutée une fois le temps écoulé (de 0,1 à 2,5 secondes).

Avec temps = 0, l'instruction n'est pas exécutée tant qu'un signal infrarouge n'est pas détecté par le capteur.

**Note :** l'accès à cette fonction dépend du microcontrôleur PICAXE sélectionné.

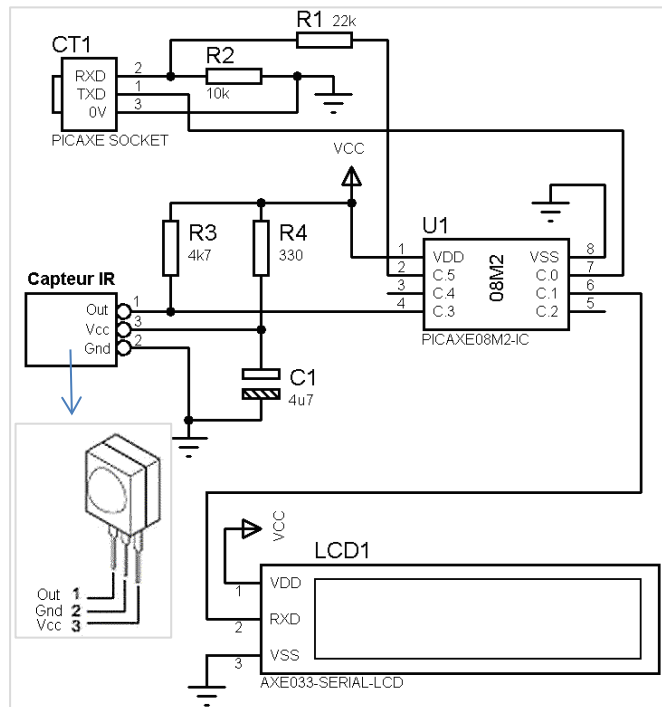
**Programme :** 32EntreeIR.plf



Ce programme permet de vérifier la valeur du capteur IR en fonction de la touche appuyée.

La donnée en provenance du capteur IR connecté à l'entrée 3, apparaît sur l'afficheur LCD ou OLED.

**Exemple de câblage :** 32\_08M2.DSN

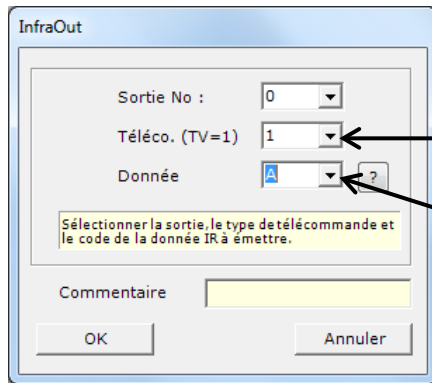


\* La télécommande PICAXE RAX-TV010 fonctionne selon le protocole Sony avec une modulation à 38 KHz.

### 3.6.9. Autres E/S > Sortie Infrarouge

**Sortie Infrarouge**

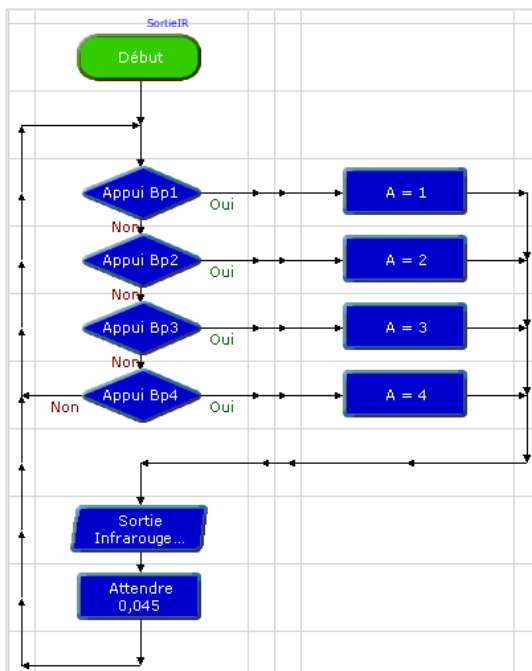
**Fonction :** Permet d'émettre une information véhiculée par un signal infrarouge (IR) selon le protocole Sony avec une modulation à 38 KHz.



Choisir 1 pour assurer la compatibilité pour la réception IR par un microcontrôleur PICAXE.

Sélection de l'information à transmettre (valeur possible de 0 à 127 + variable).

**Programme :** 33SortieIR.plf

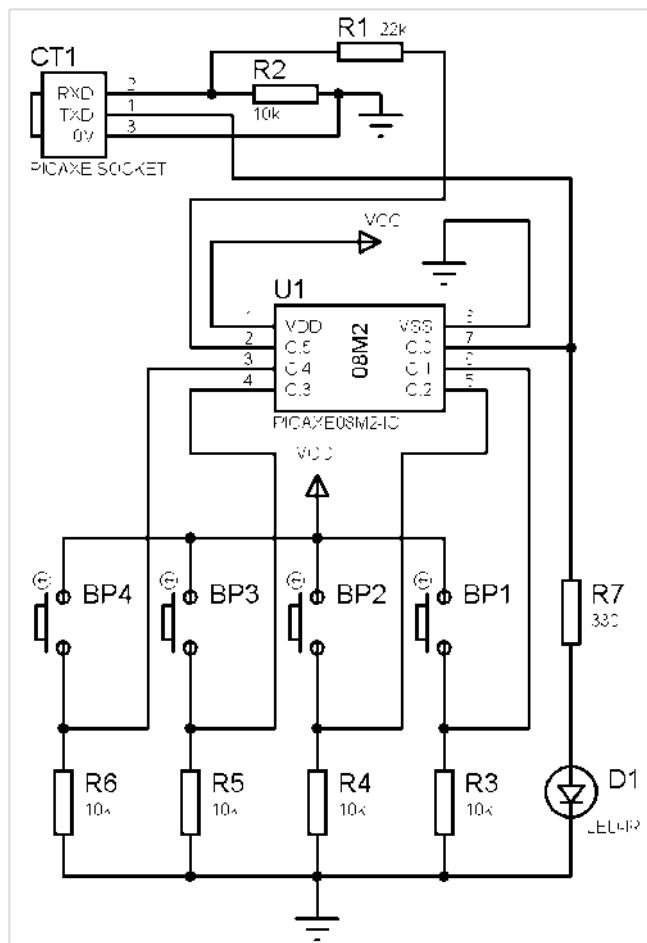


**Télécommande 4 choix :**

Chaque appui d'un bouton-poussoir charge la variable avec une donnée correspondante.

L'émission s'effectue au travers d'une LED infrarouge et se répète toutes les 45 ms tant que l'on appuie sur une touche.

**Exemple de câblage :** 33\_08M2.DSN



Ce montage permet de remplacer une télécommande TVR10.

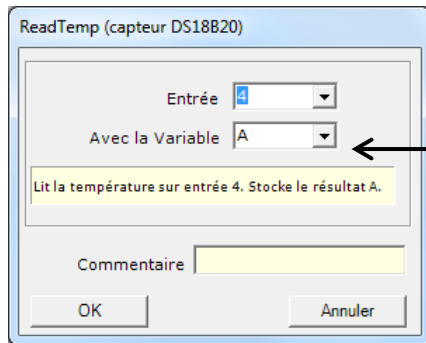
### 3.6.10. Autres E/S > Lire temp



**Fonction :** Permet de stocker dans une variable la température acquise par un capteur numérique DS18B20.

Le contenu de la variable correspond directement à la température exprimée en degrés Celsius. La plage de fonctionnement du capteur DS18B20 est comprise entre - 55° et 127°C.

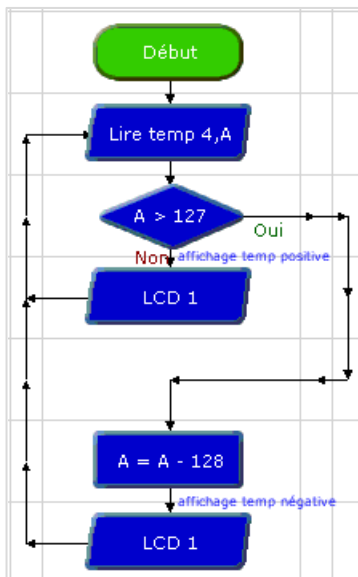
**Note :** la fonction « Calibrer un capteur analogique » permet de visualiser en direct la valeur retournée par le capteur de température connecté à une entrée numérique du microcontrôleur PICAXE. Voir chapitre correspondant à la fin de ce document.



Choix de la variable dans laquelle on stocke l'information.

**Note :**

**Programme :** 34Liretemp.plf

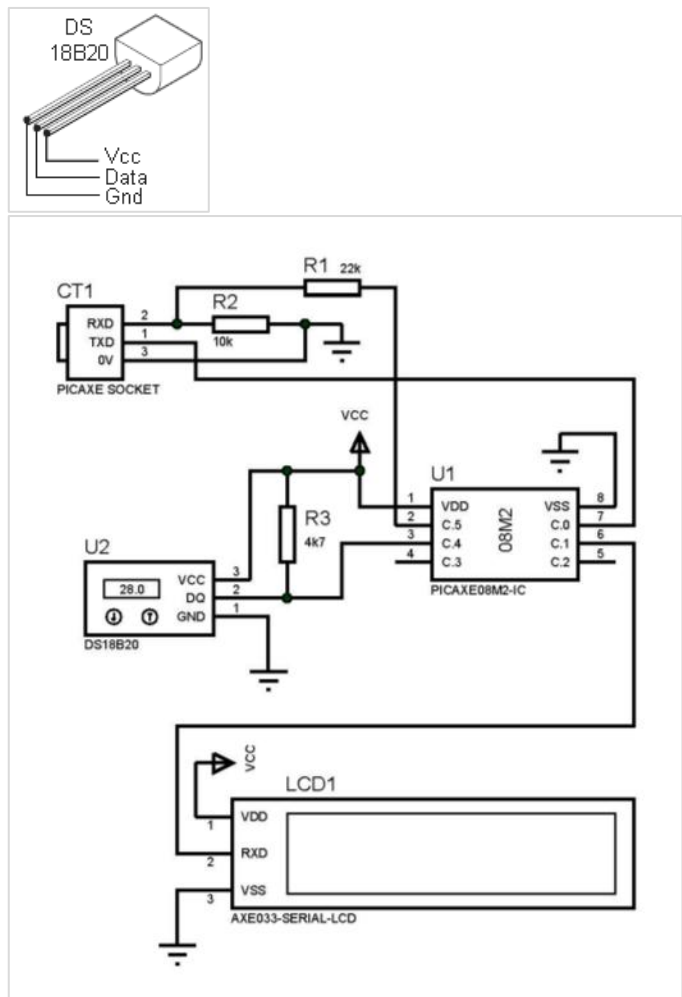


#### Affichage de la température :

La donnée en provenance du capteur de température connecté à l'entrée 3, apparaît sur l'afficheur LCD ou OLED.

Les valeurs supérieures à 127 correspondent à des températures négatives.

**Exemple de câblage :** 34\_08M2.DSN



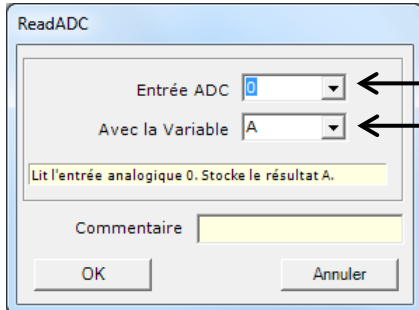


### 3.6.11. Autres E/S > C.A.N.

C.A.N.

**Fonction :** Permet d'effectuer la conversion d'une donnée analogique (niveau de tension compris entre 0 et 5 V) en une valeur numérique sur une échelle décimale allant de 0 à 255. Le résultat de la conversion est stocké dans une variable.

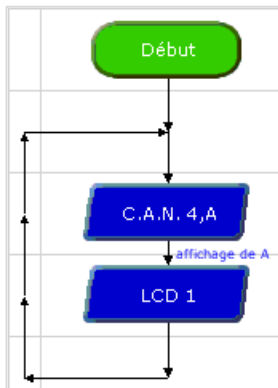
**Note :** la fonction « Calibrer un capteur analogique » permet de visualiser en direct la valeur retournée par un capteur connecté à une entrée analogique du microcontrôleur PICAXE. Voir chapitre correspondant à la fin de ce document.



← Choix de l'entrée analogique.

← Choix de la variable dans laquelle on stocke l'information.

Programme : 35CAN.pif

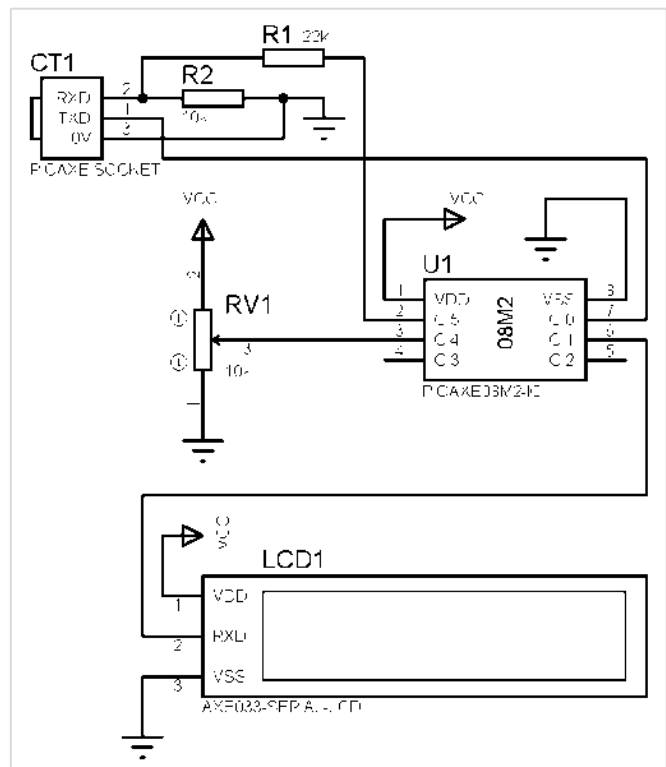


#### Affichage de la valeur analogique :

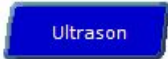
Une image de la tension en provenance de RV1 connectée à l'entrée 4, apparaît sur l'afficheur LCD ou OLED.

Valeur analogique	Valeur numérique
0 V	0
2,5 V	127
5 V	255

Exemple de câblage : 35\_08M2.DSN



### 3.6.12. Autres E/S > Ultrason

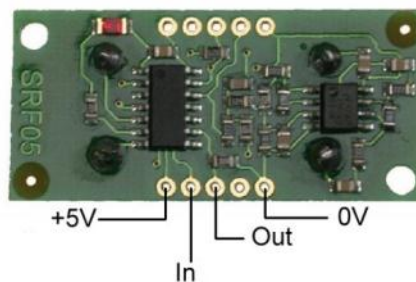
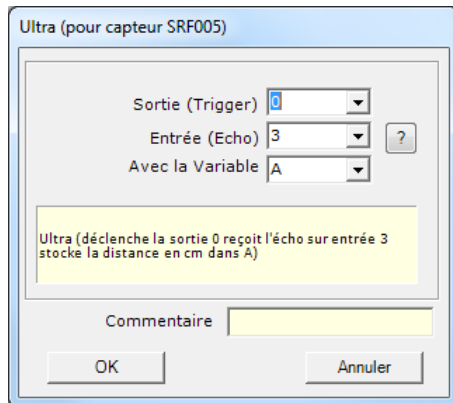


**Fonction :** Permet de stocker dans une variable la distance qui sépare un capteur à ultrason de type SRF05 et un obstacle. Le contenu de la variable correspond directement à la distance exprimée en centimètres.  
Les options et le câblage du module SRF05 dépendent du microcontrôleur PICAXE utilisé.

**Note :** la plage de fonctionnement du capteur à ultrason est comprise entre 0 et 255 centimètres. Au-delà, le capteur renvoie une valeur plafonnée à 255 cm.

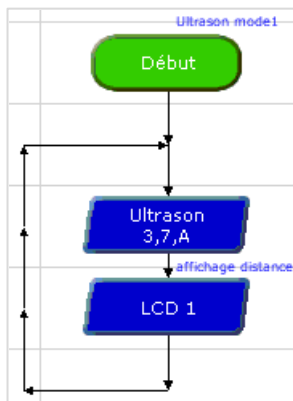
#### Mode 1 - Utilisation d'une broche d'entrée et d'une broche de sortie

Les microcontrôleurs de la série X1 fonctionnent en mode 1 : ils utilisent une sortie pour envoyer l'ordre d'acquisition de distance (Trigger) et une entrée pour collecter la valeur mesurée (Echo).



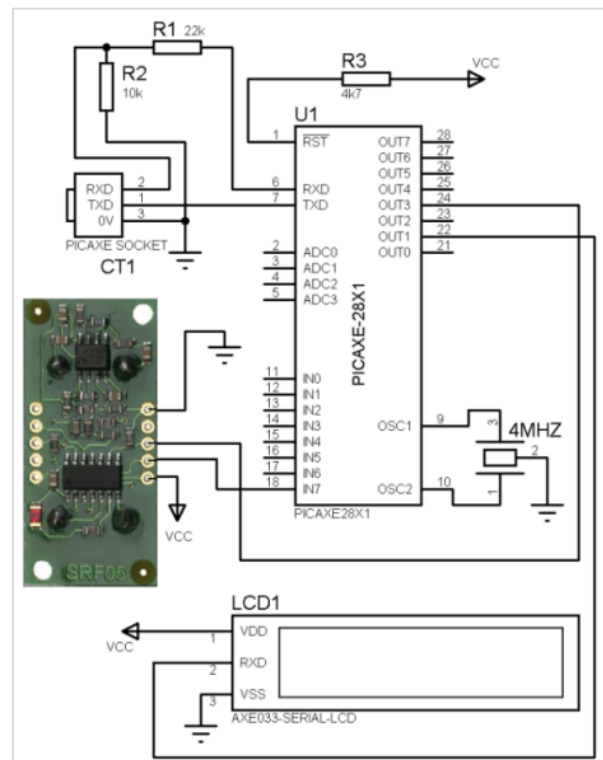
Programme : 36Ultrason1.plf (Mode 1)

Exemple de câblage : 36\_28X1.DSN (Mode 1)



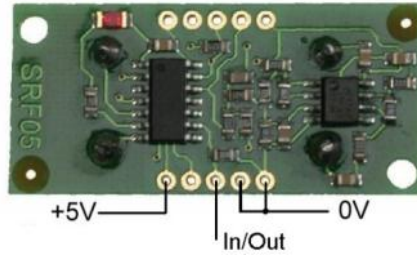
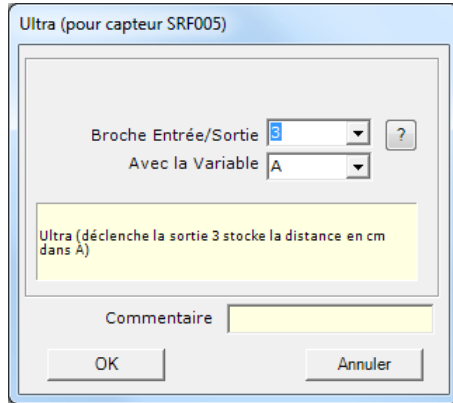
#### Affichage de la distance :

La donnée en provenance du capteur à ultrason SRF05 connecté à sortie 3 et à l'entrée 7, apparaît sur l'afficheur LCD ou OLED.

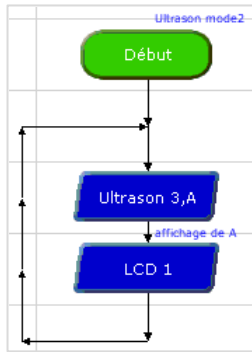


## Mode 2 - Utilisation d'une seule broche fonctionnant en entrée/sortie

Les microcontrôleurs de la série M2 et X2 fonctionnent en mode 2 : ils utilisent la même entrée/sortie pour envoyer l'ordre d'acquisition de distance (Trigger) et collecter la valeur mesurée (Echo).



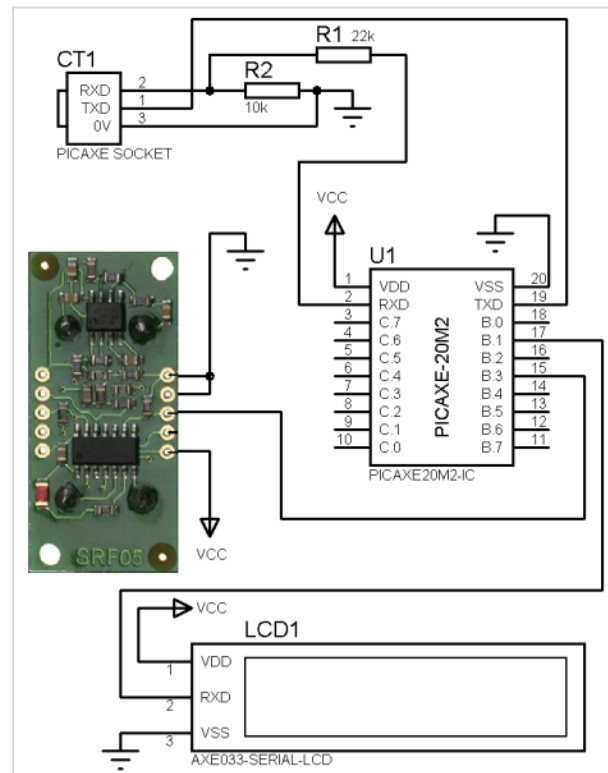
**Programme :** 36Ultrason2.plf (Mode 2)



### Affichage de la distance :

La donnée en provenance du capteur à ultrason SRF05 connecté à l'entrée/sortie 3, apparaît sur l'afficheur LCD ou OLED.

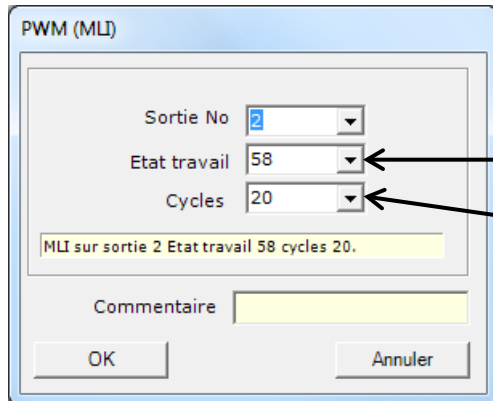
**Exemple de câblage :** 36\_20M2.DSN (Mode 2)



### 3.6.13. Autres E/S > M.L.I.

M.L.I.

**Fonction :** Permet de générer un signal MLI (Modulation en Largeur d'Impulsion) pendant un nombre de cycles donné, afin de produire un signal sur la sortie 1,2 ou 4 d'un microcontrôleur PICAXE de la série 08 ou 08M.

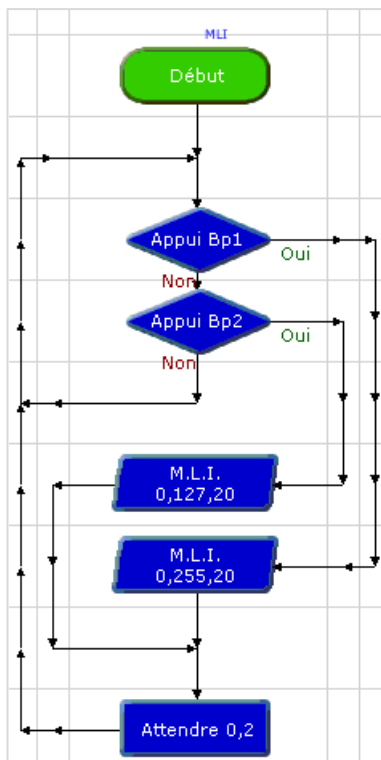


← Correspond au niveau analogique.

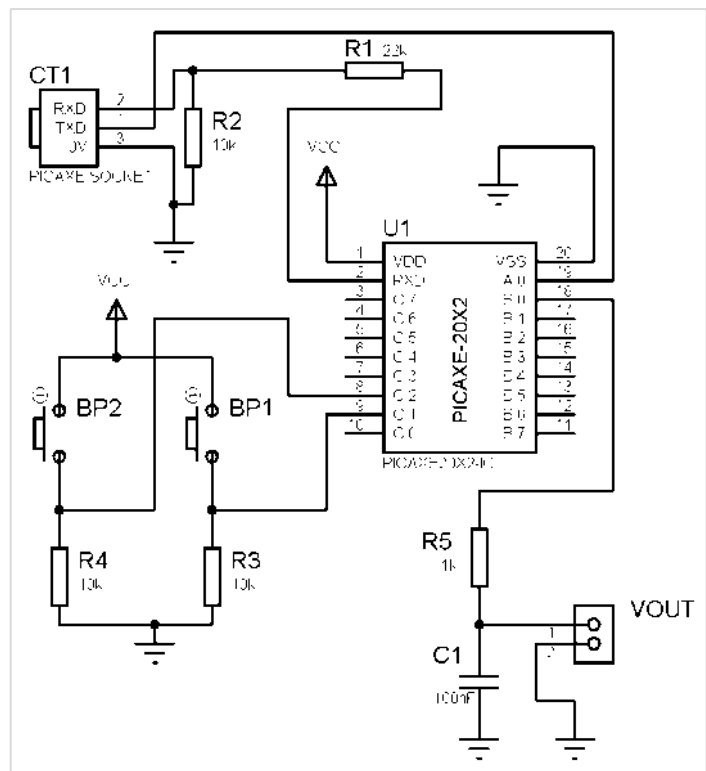
← Indique le nombre de cycles. 1 cycle = 5 ms.

**Note :** l'utilisation de cette instruction requiert la connexion d'un résistor et d'une capacité sur la sortie concernée qui devient alors une sortie analogique.  
L'instruction **MLI** doit être exécutée périodiquement afin de rafraîchir le niveau de tension présent sur la sortie.

**Programme :** 37MLI.pf



**Exemple de câblage :** 37\_20X2.DSN



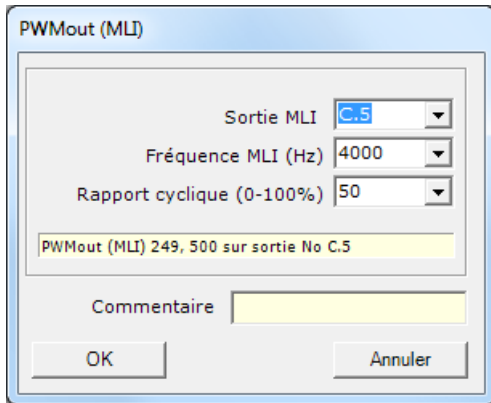
#### Génération d'une tension analogique :

L'appui du bouton-poussoir génère un niveau de tension sur la sortie 0.

### 3.6.14. Autres E/S > Sortie M.L.I

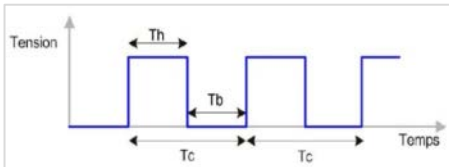
#### Sortie MLI

**Fonction :** Permet de générer un signal MLI (Modulation en Largeur d'Impulsion) continu sur une sortie PWM d'un microcontrôleur PICAXE.



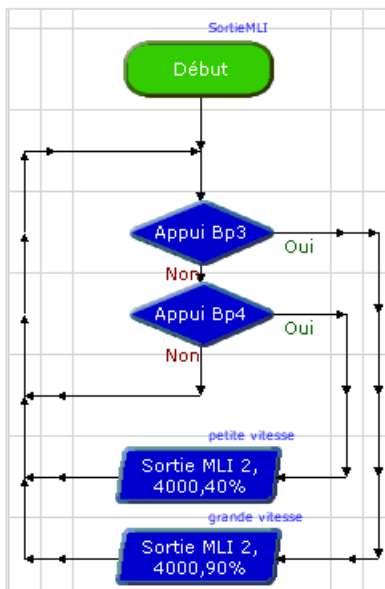
**Note :** l'arrêt du signal MLI s'effectue avec des paramètres de fréquence MLI à 0 Hz et de rapport cyclique à 0%.

#### Rappel :



Rapport cyclique =  $\text{Th}/\text{Tc}$

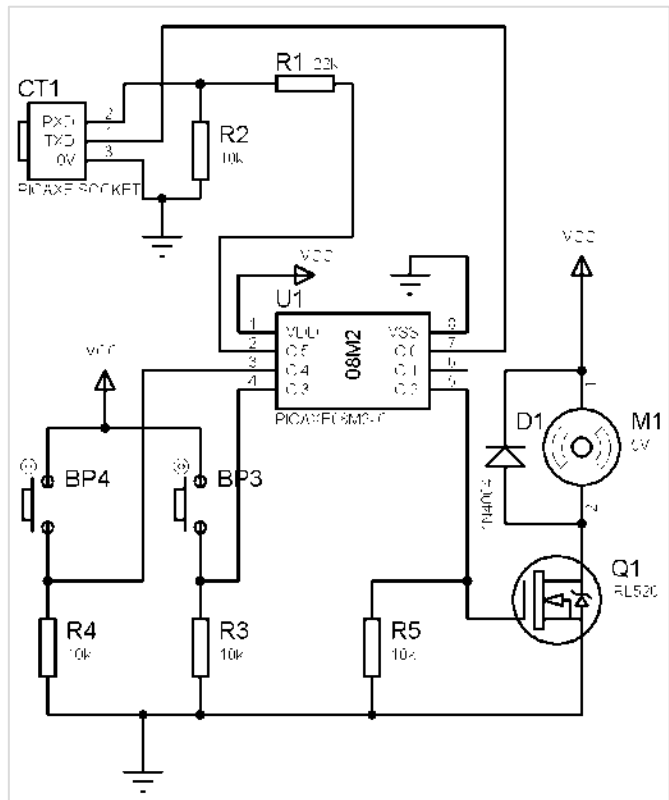
#### Programme : 38PWM.plf



#### Contrôle de la vitesse de rotation :

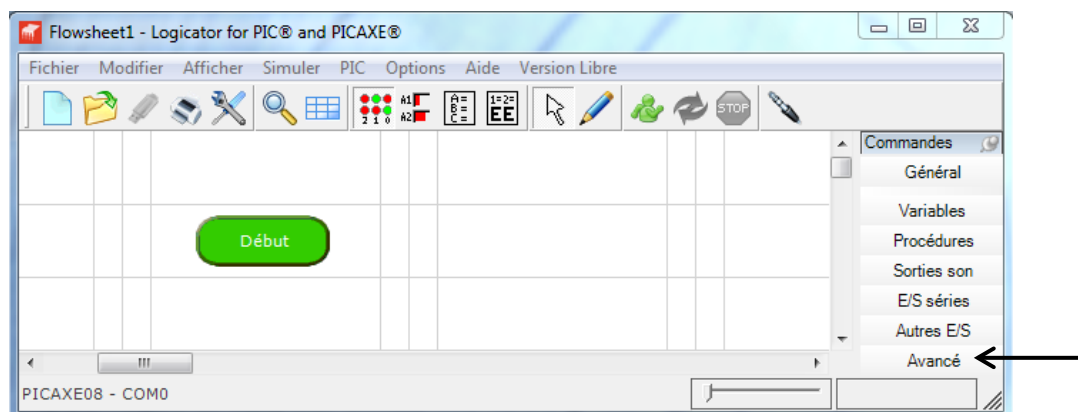
En fonction du bouton-poussoir actionné, le signal MLI varie, ce qui entraîne la vitesse du moteur par exemple.

#### Exemple de câblage : 38\_08M2.DSN



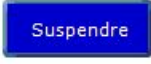
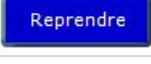
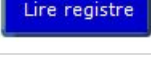
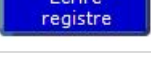
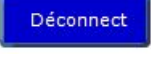
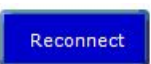


### 3.7. Avancé

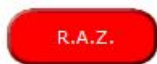
A partir de la barre des commandes, cliquer sur **Avancé**.



Ce groupe donne accès aux instructions avancées du PICAXE.

	Permet de réinitialiser le microcontrôleur PICAXE. C'est l'équivalent logiciel de la réinitialisation externe « RESET ».
	Permet de positionner le microcontrôleur PICAXE en mode d'économie d'énergie (mode sommeil) pour une période donnée.
	Permet de suspendre l'exécution d'un diagramme.
	Permet de reprendre l'exécution d'un diagramme.
	Permet de lire une donnée du registre SFR ( <i>Special Function Register</i> ) du microcontrôleur PICAXE et de la stocker dans une variable.
	Permet d'écrire la valeur d'une variable à une adresse du registre SFR ( <i>Special Function Register</i> ) du microcontrôleur PICAXE.
	Pour déconnecter la liaison de programmation avec le microcontrôleur PICAXE et interrompre la vérification cyclique de demandes de chargement d'un nouveau diagramme de programmation.
	Pour autoriser de nouveau le chargement d'un diagramme et rétablir la liaison de programmation après l'utilisation d'une instruction <b>Déconnect</b> .

### 3.7.1. Avancé > R.A.Z



**Fonction :** Permet de réinitialiser le microcontrôleur PICAXE.  
C'est l'équivalent logiciel de la réinitialisation externe « RESET ».  
Après cette instruction, le programme recommence au début et toutes les variables sont réinitialisées à zéro.

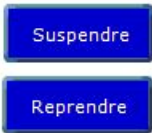
### 3.7.2. Avancé > Sommeil



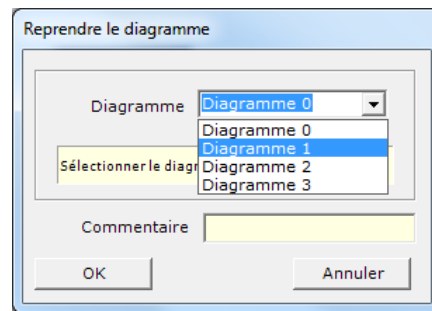
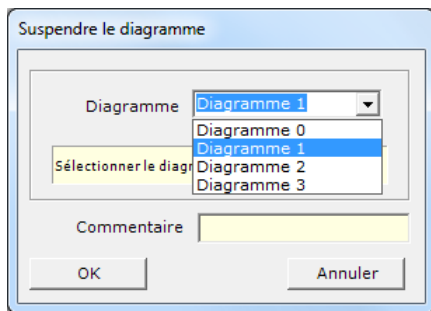
**Fonction :** Permet de positionner le microcontrôleur PICAXE en mode d'économie d'énergie (mode sommeil) pour une période donnée.  
Dans ce mode, les compteurs internes du microcontrôleur PICAXE sont arrêtés, ce qui a pour effet d'interrompre les instructions **Sortie M.L.I** et **Servo**.

← Détermine le temps de mise en sommeil exprimé par multiples de 2,3 secondes.

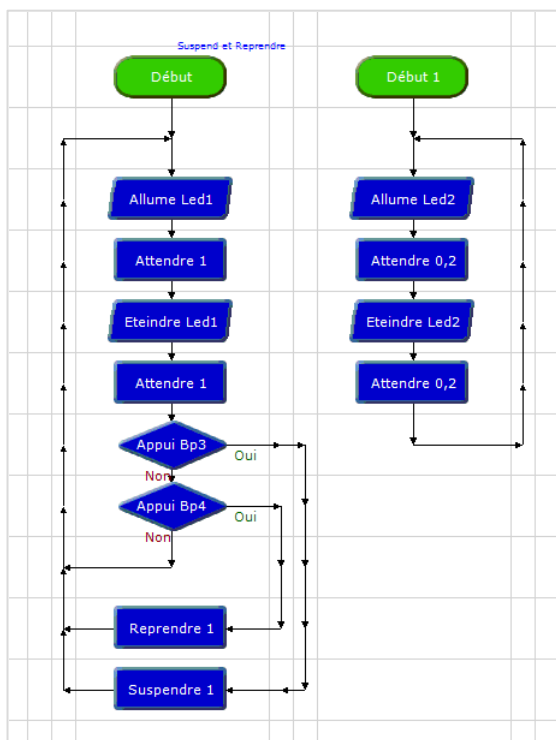
### 3.7.3. Avancé > Suspend et Reprendre



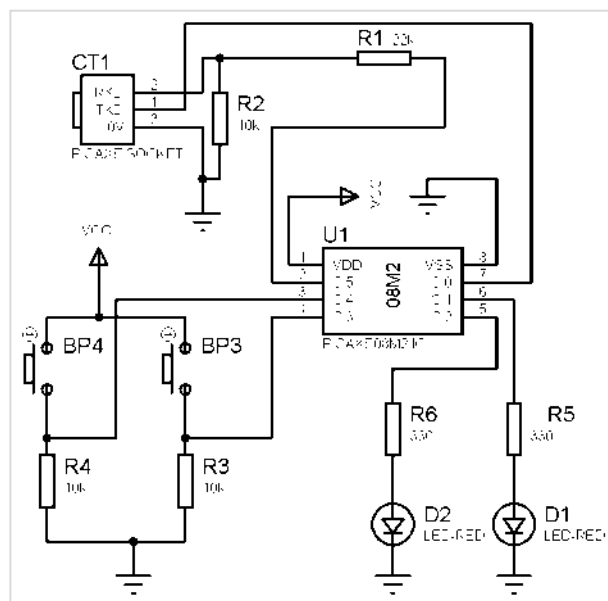
**Fonction :** **Suspend** et **Reprendre** permettent respectivement de suspendre l'exécution d'un diagramme et de la reprendre.  
Ces instructions s'utilisent avec les microcontrôleurs PICAXE de la série M2 qui permettent d'exécuter simultanément jusqu'à quatre diagrammes.



Programme : 39Suspend.pf



Exemple de câblage : 39\_08M2.DSN



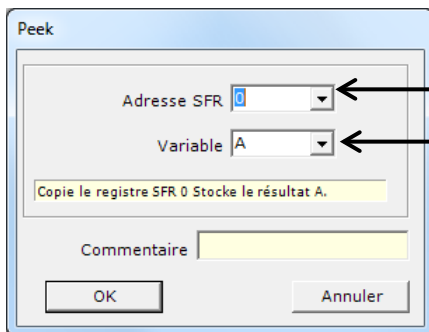
**Clignotements indépendants :** La LED D1 clignote en permanence.  
L'appui sur Bp3 suspend le clignotement de la LED D2 et l'appui de Bp4 reprend le clignotement.



### 3.7.4. Avancé > Lire registre

Lire registre

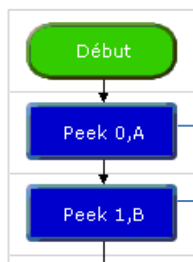
**Fonction :** Permet de lire une donnée du registre SFR (*Special Function Register*) du microcontrôleur PICAXE et de la stocker dans une variable.



← Choix de l'adresse de lecture.

← Choix de la variable dans laquelle on stocke l'information.

**Programme :**



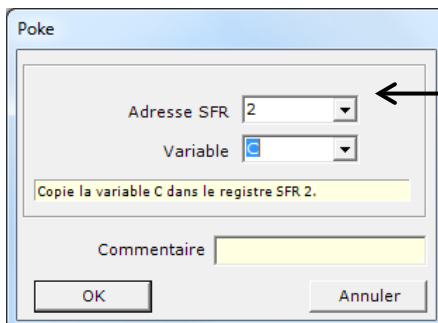
→ Le contenu de la mémoire SFR 0 est stocké dans la variable A.

→ Le contenu de la mémoire SFR 1 est stocké dans la variable B.

### 3.7.5. Avancé > Ecrire registre

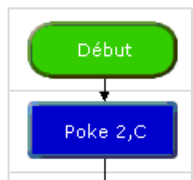
Ecrire registre

**Fonction :** Permet d'écrire la valeur d'une variable à une adresse du registre SFR (*Special Function Register*) du microcontrôleur PICAXE.



← Choix de l'adresse SFR où l'on écrit la donnée.

**Programme :**



→ Le contenu de la variable C est écrit dans la mémoire SFR 2.

### 3.7.6. Avancé > Déconnect et Reconnect



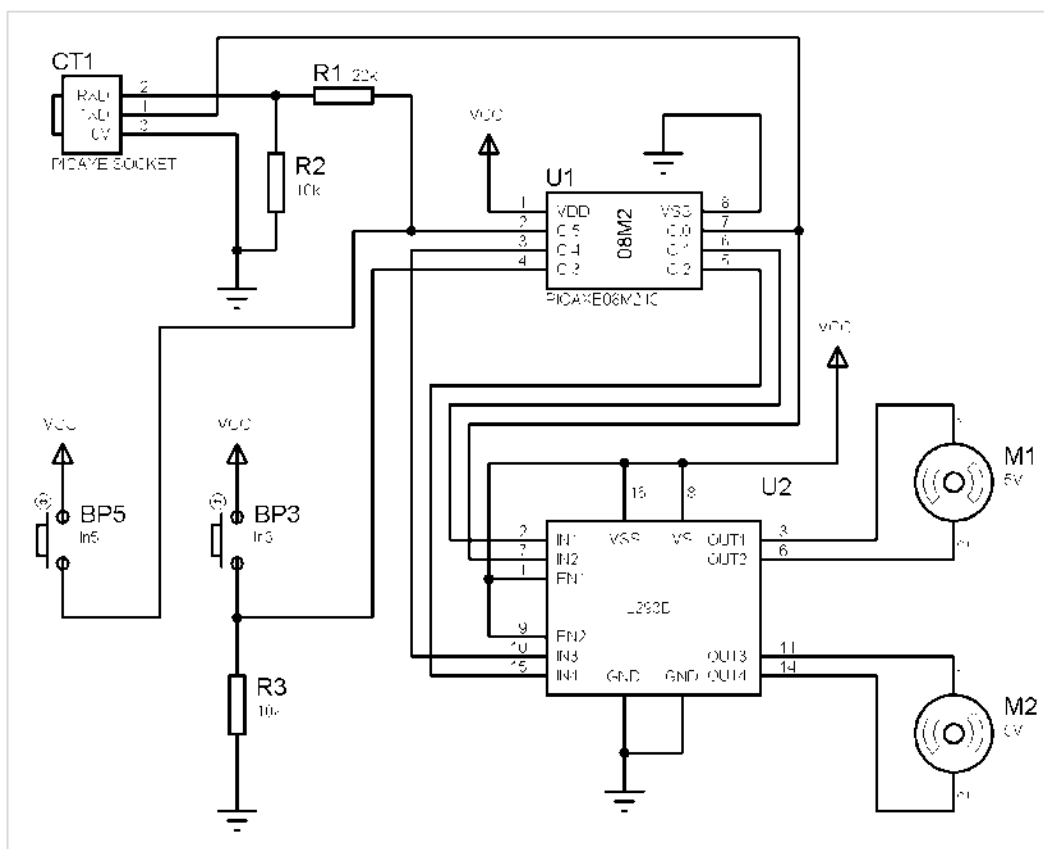
**Fonction :** Déconnecte la liaison de programmation avec le microcontrôleur PICAXE et interrompt la vérification cyclique de demandes de chargement d'un nouveau diagramme de programmation. Pour autoriser de nouveau le chargement d'un diagramme, il faut rétablir la liaison avec l'instruction **Reconnect**.

**Ces instructions ne nécessitent pas de boîte de dialogue.**

**Note 1 :** il est possible de forcer le chargement d'un programme en effectuant une réinitialisation du microcontrôleur PICAXE (RESET logiciel ou matériel).

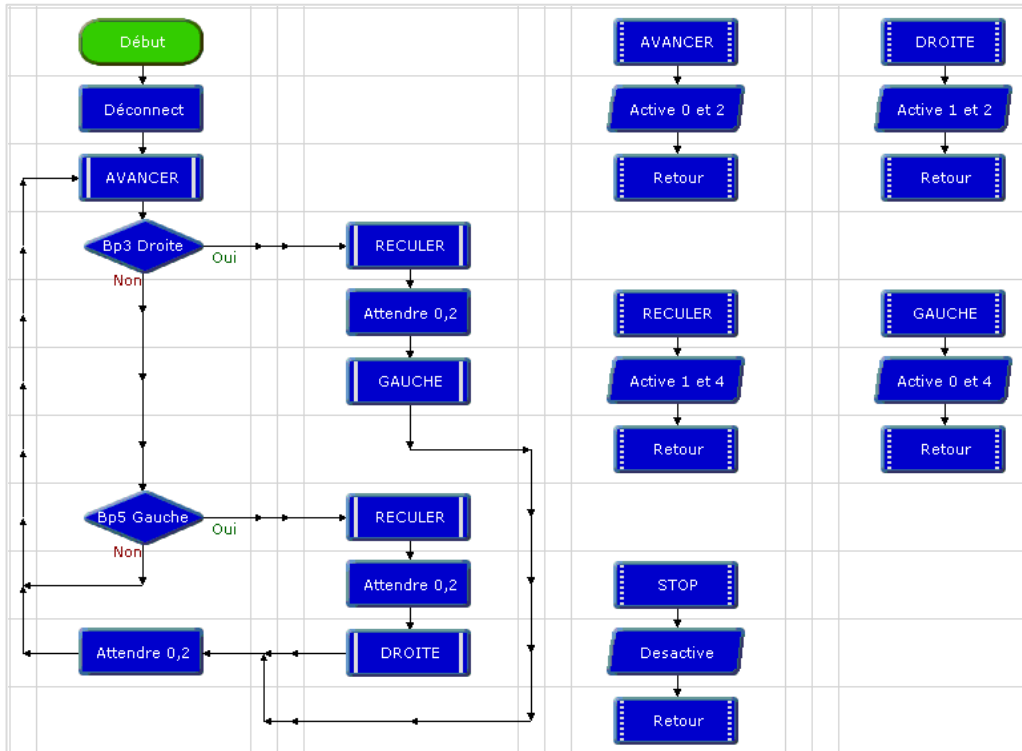
**Note 2 :** les instructions **Entrée série** et **Sortie série** déconnectent automatiquement et temporairement la liaison de programmation. Il n'est pas nécessaire d'utiliser l'instruction **Reconnect** pour rétablir la liaison.

**Exemple de câblage :** 40\_08M2.DSN



Dans certains cas, il peut être utile de pouvoir exploiter l'entrée réservée à la liaison de programmation. Il faut alors utiliser l'instruction **Déconnect**.

L'exemple ci-dessus illustre l'exploitation d'une information provenant d'un bouton-poussoir (BP5) en le connectant sur cette entrée (5).



**Robot mobile avec détection droite et gauche :**

L'instruction **Déconnect** en début de programme permet de prendre en compte l'entrée 5.

Les instructions de mouvement sont écrites en effectuant des sous-programmes.

En effet, l'instruction **Moteurs** est mal adaptée avec les microcontrôleurs de la série 08.

Les détections des moustaches droite ou gauche permettent au robot mobile d'effectuer un évitement de l'obstacle.

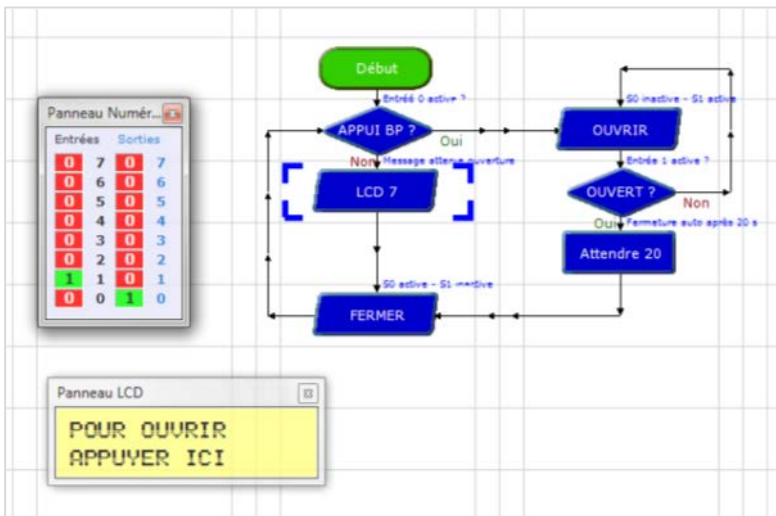
## 4. Simulation

Logicator offre deux modes de simulation très utiles pour mettre au point et corriger des programmes.

### • Simulation d'un diagramme

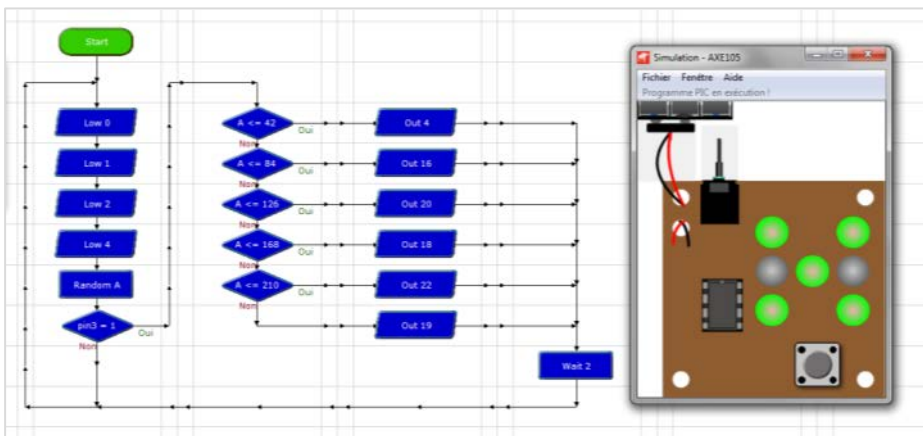
Permet de vérifier le bon enchaînement des instructions d'un diagramme de programmation.

Un curseur se déplace sur les instructions pour matérialiser leur exécution. L'utilisateur dispose de panneaux de visualisation pour agir sur les entrées et visualiser l'état des sorties.



### • Simulation sur un modèle virtuel

Le diagramme de programmation est transféré dans une maquette virtuelle créée avec le logiciel *Simulation Studio*. L'utilisateur agit sur la maquette virtuelle (ex. clic sur le bouton poussoir) et visualise le résultat (ex. allumage des LED).



## 4.1. Simulation d'un diagramme

Le mode simulation permet de tester le fonctionnement d'un programme sans pour autant le télécharger dans un microcontrôleur PICAXE.

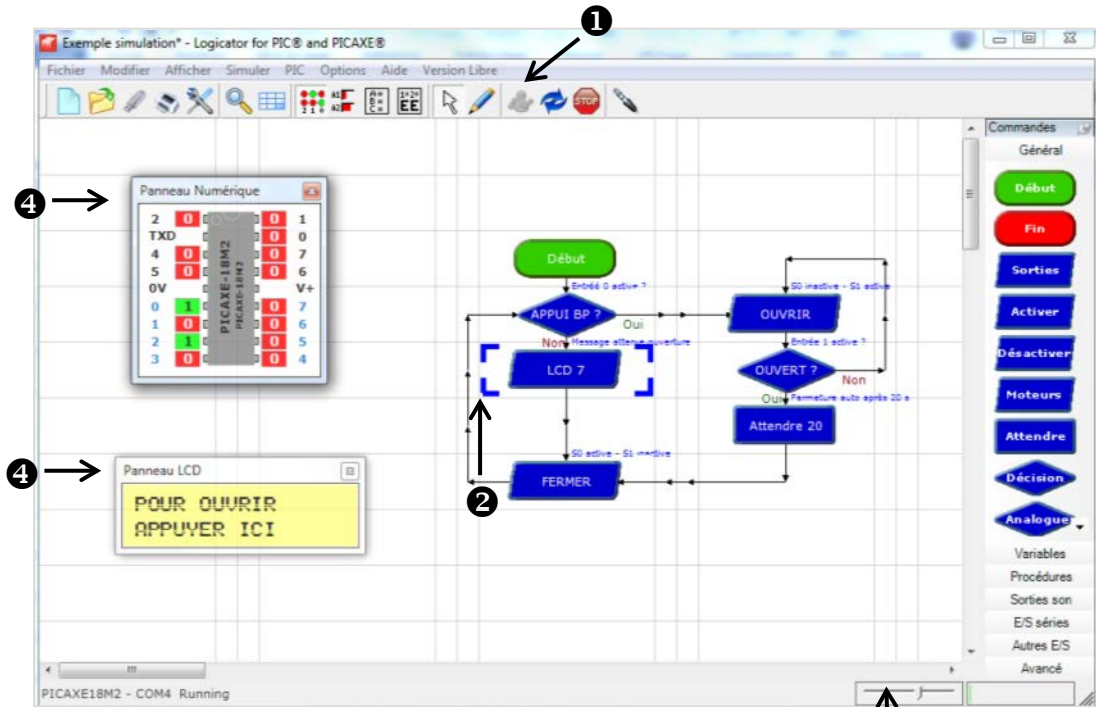
Ce mode permet de visualiser l'état des sorties et les différentes variables du programme, il permet aussi de faire évoluer les entrées numériques et analogiques du microcontrôleur PICAXE.

L'affichage LCD, la télécommande Infrarouge PICAXE et la mesure de distance peuvent aussi être simulés.

### 4.1.1. L'écran de simulation

Le mode simulation est accessible :

- à partir de la barre des menus, en cliquant sur **Simuler > Exécuter** ;
- en cliquant sur l'icône **Exécuter le Diagramme** ;
- en appuyant sur la touche **F10**.



- 1 Lancement de la simulation.
- 2 Identification de l'instruction en cours d'exécution [ ].
- 3 Modification de la vitesse de simulation avec le curseur.
- 4 Panneaux de visualisation. (Voir paragraphe suivant).

**Note** : L'arrêt du mode simulation s'effectue en cliquant sur ou en appuyant sur la touche **F11**.

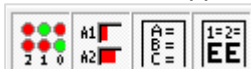
### 4.1.2. Les panneaux de visualisation

Il existe différents panneaux d'information et de paramétrage pour la simulation des programmes.

L'activité des entrées/sorties est matérialisée par la couleur rouge (inactif) ou verte (actif).

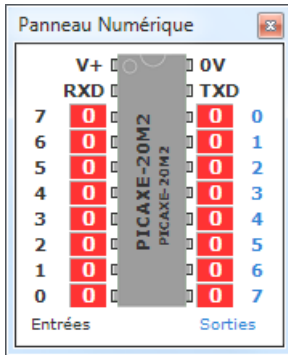
Le contenu des panneaux s'adapte au microcontrôleur PICAXE sélectionné.

A partir de la barre des menus, cliquer sur **Afficher**, puis cocher le ou les panneaux que vous souhaitez voir apparaître à l'écran. Vous disposez également des touches de raccourcis.



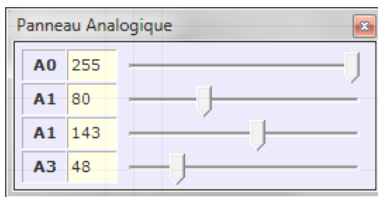
### Le panneau Numérique

Pour agir manuellement (clic gauche ou touches de fonction) sur les entrées et visualiser les sorties.



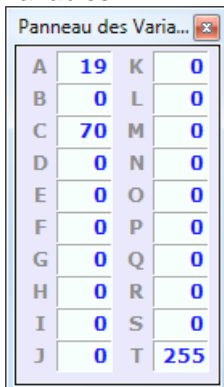
### Le panneau Analogique

Pour positionner la valeur des entrées analogiques.



### Le panneau des variables

Pour connaître les valeurs attribuées aux variables.

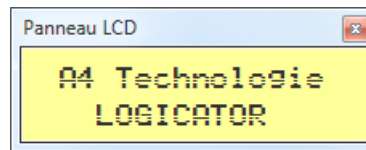


**Note :** les touches de fonction du clavier permettent d'agir simultanément sur plusieurs entrées numériques.



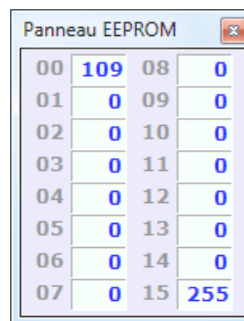
### Le panneau LCD

Pour simuler le fonctionnement d'un afficheur LCD PICAXE (2 lignes x 16 caractères).



### Le panneau EEPROM

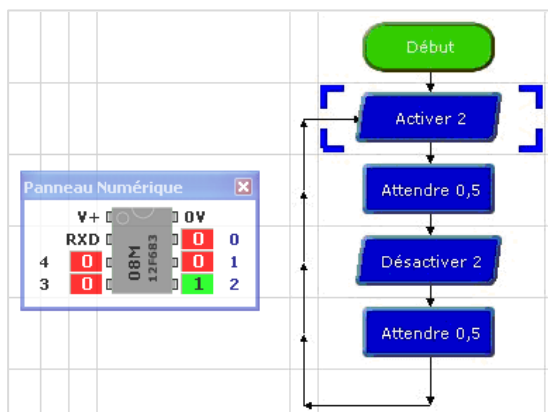
Pour connaître les valeurs attribuées aux variables EEPROM.



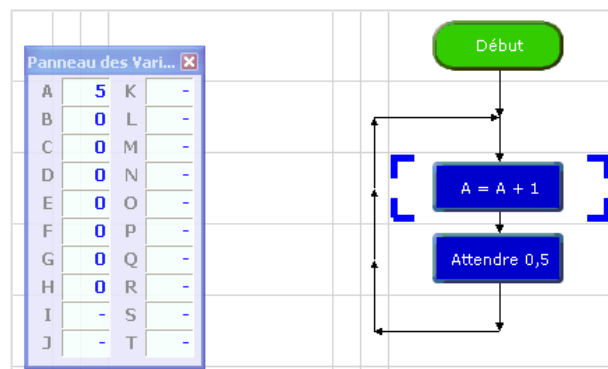
### 4.1.3. Exemples de simulation

Voici plusieurs exemples de simulations sur des microcontrôleurs PICAXE 08M.

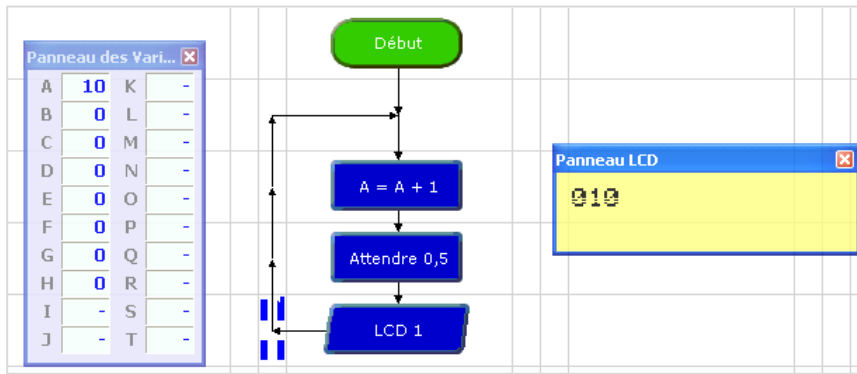
#### Changement d'état d'une sortie



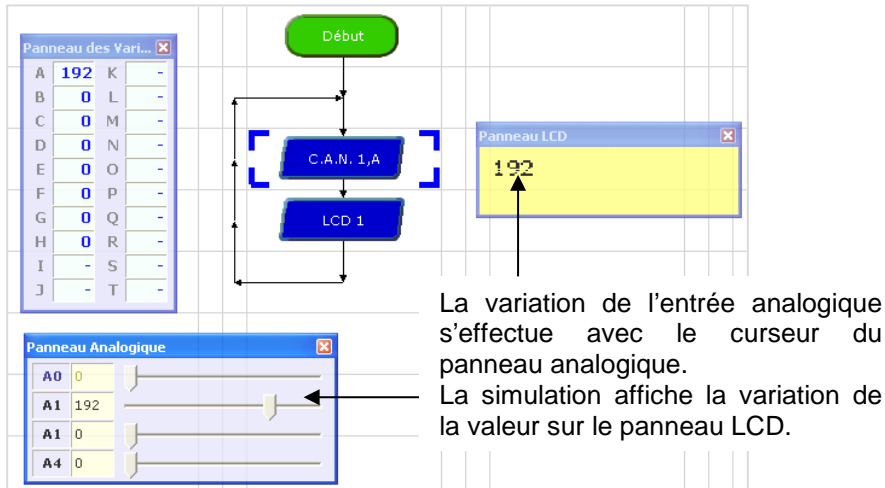
#### Evolution d'une variable



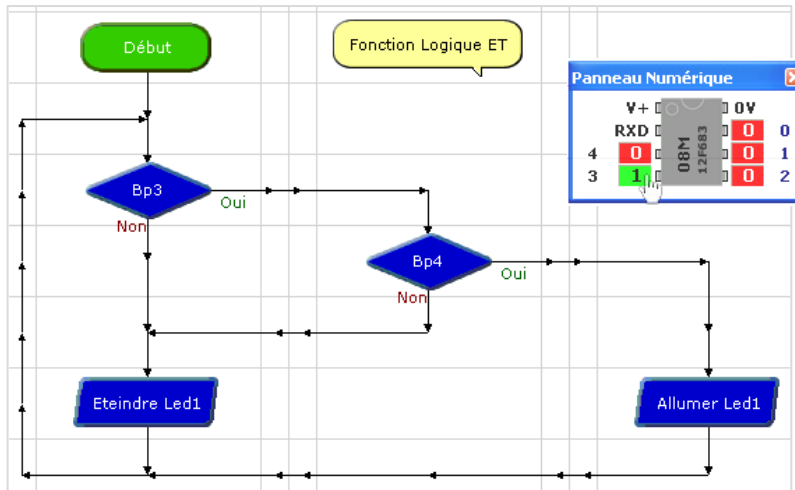
## Affichage LCD



## Changement d'état d'une sortie à l'aide d'une entrée analogique



## Fonction logique ET



## 4.2. Simulation d'un modèle virtuel

L'installation de l'application Logicator donne accès à l'environnement *Simulation Studio*.

Il permet de créer des modèles virtuels de cartes électroniques équipées d'un microcontrôleur PICAXE et de les animer en chargeant des programmes créés avec Logicator.

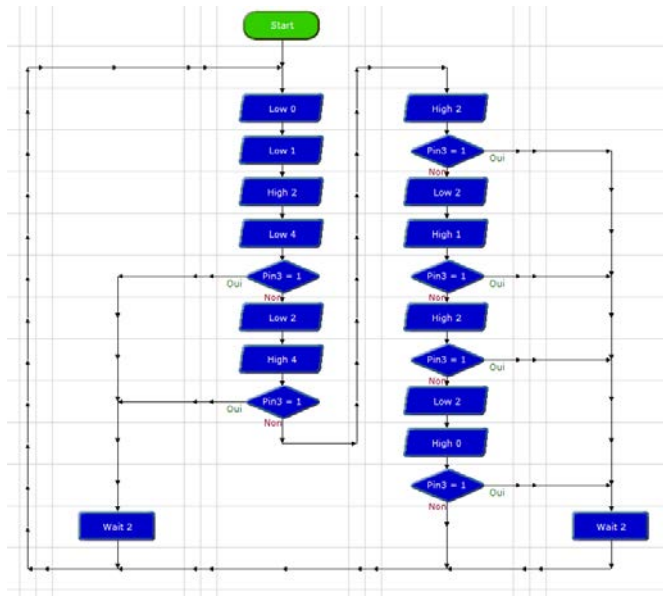
Cette simulation est très réaliste puisqu'elle permet à l'utilisateur d'effectuer les manipulations d'une maquette virtuelle comme il le ferait avec une maquette réelle.

Vous pouvez créer votre propre modèle de simulation ou travailler à partir d'un modèle de la banque de simulation proposée par Logicator.

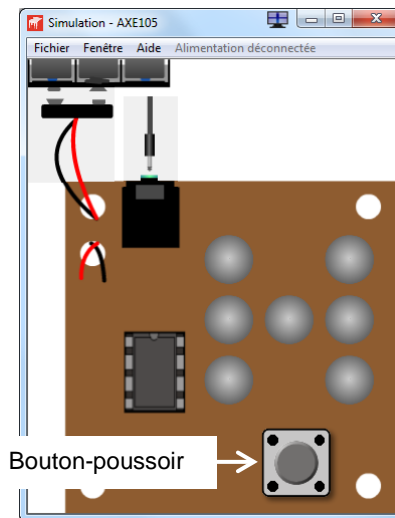
### 4.2.1. Utiliser un modèle de la bibliothèque de simulation

Logicator dispose d'une banque de simulation accessible à partir du menu **Simuler > Ouvrir la simulation > Kits PICAXE** et de programmes exemples associés accessibles à partir de **Fichier > Ouvrir des exemples...**

**Programme :** AXE105 Dice Program1.plf



**Exemple de fichier de simulation :** AXE105 Dice.pss

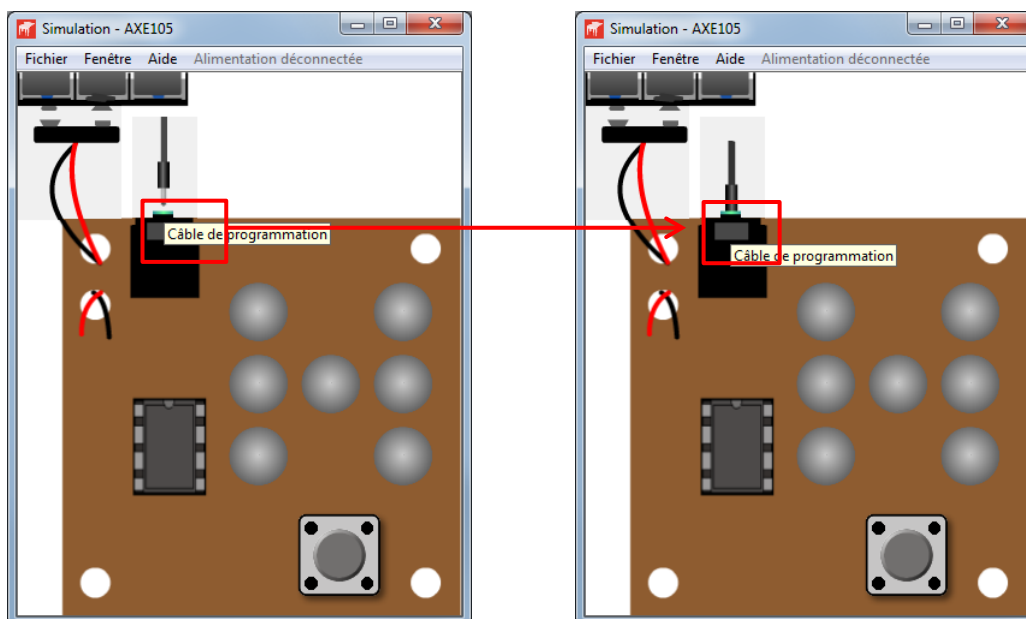


Les LED symbolisent les points d'un dé à jouer. L'appui du bouton-poussoir (Pin3) déclenche le lancer du dé.

**ATTENTION !** Il est important de sélectionner le microcontrôleur PICAXE correspondant à celui utilisé sur la maquette virtuelle (Menu **Options > Sélectionner le type de PIC...**).

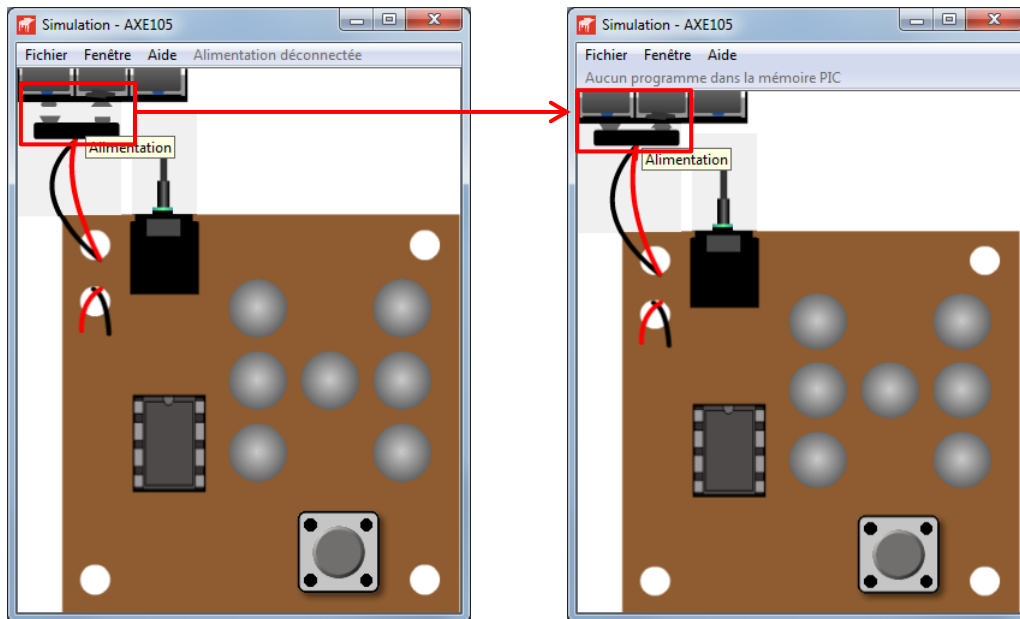
Ensuite, il faut connecter virtuellement le câble de programmation, mettre sous tension la maquette virtuelle et transférer le programme pour vérifier son bon fonctionnement.

**1. Brancher le câble de programmation**, cliquer dessus.



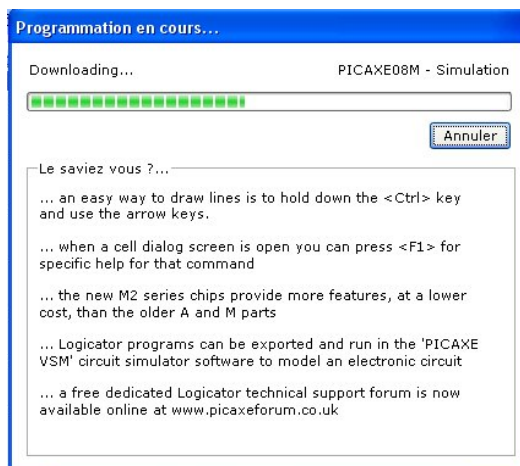


## 2. Connecter l'alimentation, cliquer dessus.

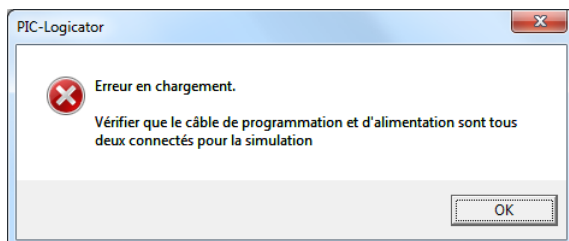


## 3. Télécharger le programme, cliquer sur l'icône Programmer le PIC.

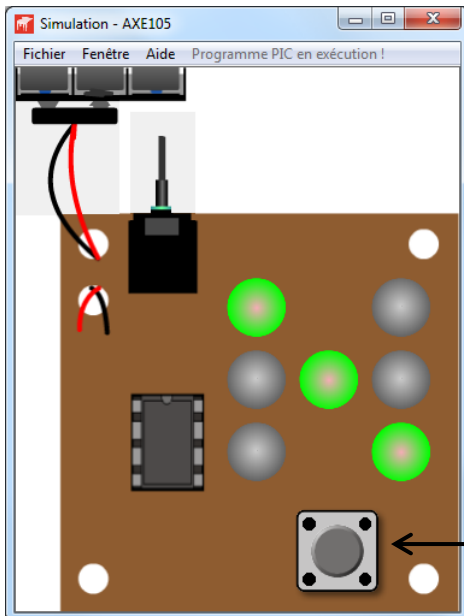
Le programme est transféré dans la maquette virtuelle et l'écran suivant apparaît.



**Note :** toute erreur de manipulation comme l'oubli de branchement du câble d'alimentation ou du câble de programmation est signalé par une fenêtre très explicite.



Si aucune erreur n'est détectée, vous pouvez alors tester le bon fonctionnement du programme.

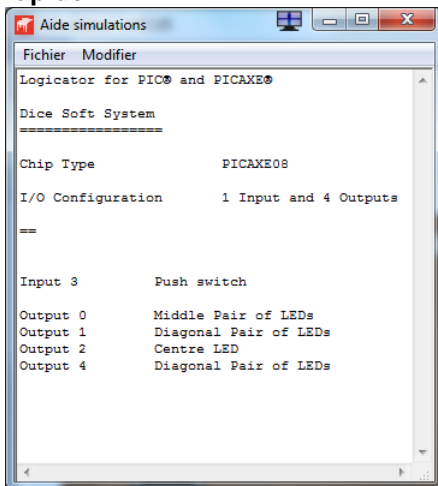


La maquette de simulation s'anime comme le ferait la maquette réelle.

Ce mode de simulation est particulièrement pédagogique pour prendre en main le système PICAXE.

← Cliquer avec la souris sur le bouton-poussoir pour lancer le dé.

Pour connaître les caractéristiques du circuit, à partir de la maquette virtuelle, cliquer sur **Aide > Aide rapide**.



#### 4.2.2. Créer un modèle de simulation

Il est possible de créer sa propre simulation en utilisant le logiciel *Simulation Studio*. A partir du menu **Démarrer**, cliquer sur **Tous les programmes > Revolution Education > Logicator for PIC and PICAXE > Tools > Simulation Studio**.



L'environnement de création de modèles virtuels *Simulation Studio* est en anglais.

Son utilisation n'est pas expliquée dans ce guide ; cependant elle est très intuitive et consiste essentiellement à manipuler des fichiers images.

L'utilisateur peut se familiariser très facilement avec cet environnement en manipulant les modèles virtuels évoqués précédemment.

Le bouton **Tutorial** appelle une aide en anglais pour l'utilisation de *Simulation Studio*.

**IMPORTANT :** les modèles personnels créés avec *Simulation Studio* doivent impérativement être placés dans le dossier **C:\Program Files (x86)\Revolution Education\Logicator for PIC micros\PIC-Syms** afin de garantir leur bon fonctionnement avec *Logicator*.

## 5. Autres fonctionnalités

Logicator propose des fonctionnalités pour une utilisation avancée de la programmation graphique : le langage BASIC et le calibrage de capteurs.

### 5.1. Le langage BASIC

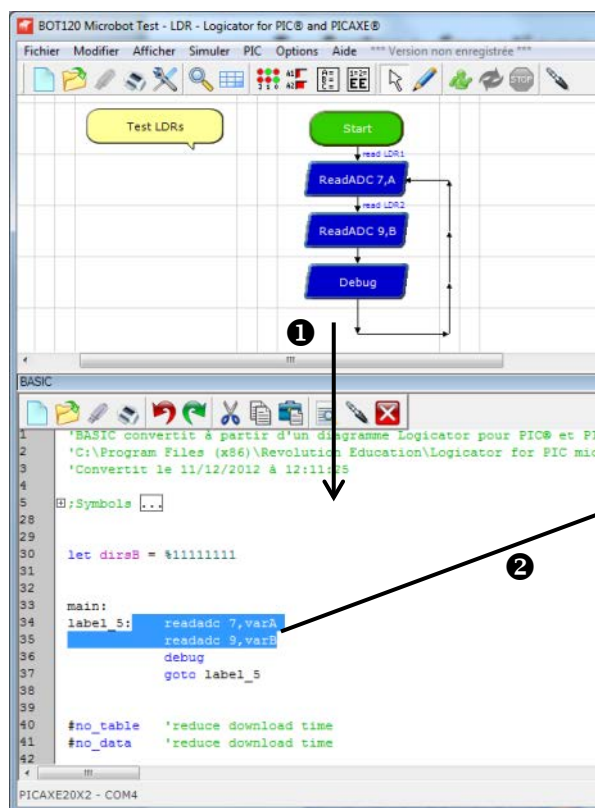
Le langage BASIC donne accès à un jeu d'instructions beaucoup plus étendu que celui proposé par le langage de programmation graphique et permet par conséquent de concevoir des programmes plus évolués.

#### 5.1.1. Conversion en langage BASIC

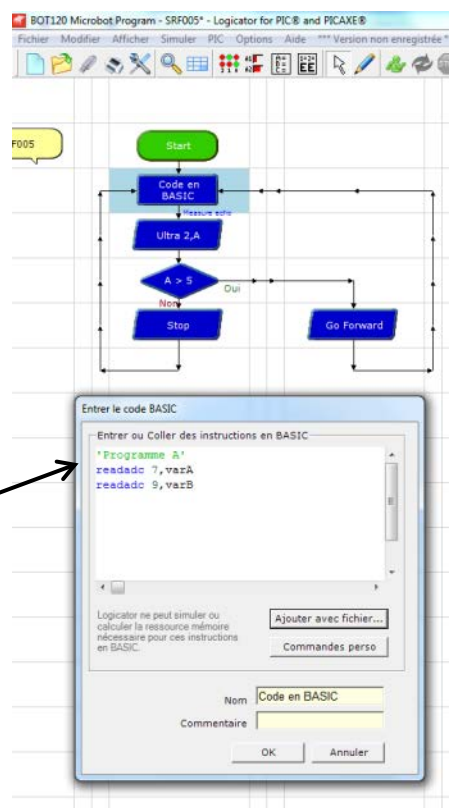
Logicator dispose d'un outil de conversion des diagrammes en langage évolué BASIC.

Cette fonctionnalité permet, sans la maîtrise complète du langage BASIC, de pouvoir récupérer un programme complexe, en le convertissant et en réintégrant les codes générés dans un autre programme.

##### Programme A (converti en BASIC)



##### Programme B (avec une partie du prog. A)



- 1 Convertir le programme A en BASIC (menu **PIC > Convertir le programme en BASIC**).
- 2 Sélectionner et copier la partie à réintégrer (prog. A) dans le bloc **Code en BASIC** (prog. B).

#### 5.1.2. Export d'un fichier en BASIC

Il est possible d'exporter un diagramme en BASIC (fichier .BAS) en vue de l'importer ultérieurement dans VSM ou de Programming Editor.

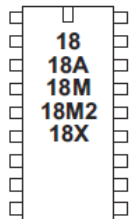
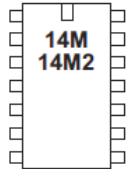
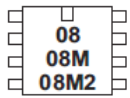
A partir de la barre des menus, cliquer sur **Fichier > Exporter > Exporter un fichier BASIC...**

### 5.1.3. Jeu d'instructions BASIC

Le document « *BASIC commands* », accessible à partir de l'aide, récapitule de manière détaillée l'intégralité du jeu d'instructions en BASIC.

Pour chaque instruction, vous trouvez la description de l'instruction, sa syntaxe, un exemple de programme et les microcontrôleurs PICAXE avec lesquels elle est compatible.

**Exemple** : instruction High



#### high

Activer

*Syntax:*

**HIGH pin {,pin,pin...}**

- Pin is a variable/constant which specifies the i/o pin to use.

*Function:*

Make pin an output and switch it high.

*Information:*

The high command switches an output on (high).

On microcontrollers with configurable input/output pins (e.g. PICAXE-08) this command also automatically configures the pin as an output.

*Example:*

```
main: high B.1      ; switch on output B.1
      pause 5000    ; wait 5 seconds
      low B.1       ; switch off output B.1
      pause 5000    ; wait 5 seconds
      goto main     ; loop back to start
```

## 5.2. Calibrer un capteur

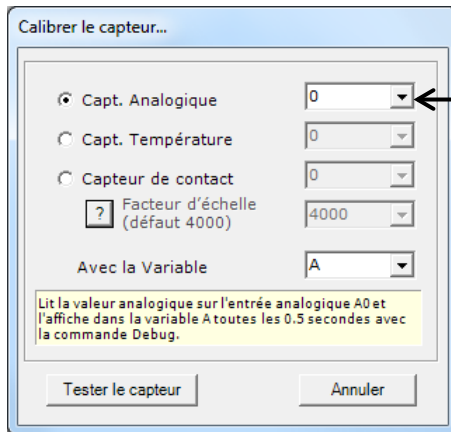
Il s'agit d'un assistant permettant de visualiser en direct la valeur retournée par un capteur connecté au microcontrôleur PICAXE.

Il est prévu pour 3 types de capteurs :

- capteur analogique connecté à une entrée analogique ;
- capteur de température type DS18B20 connecté à une entrée numérique ;
- capteur de contact « Tactile » connecté à une entrée type Touch.

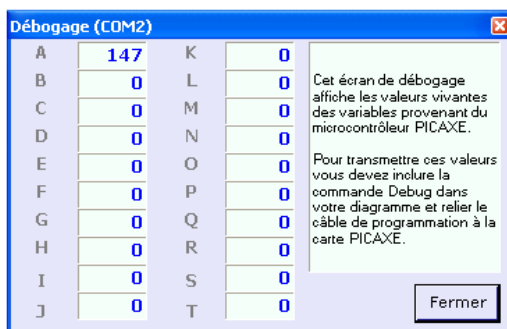
A partir de la barre des menus, cliquer sur **PIC > Calibrer un capteur Analogique...**

Voici un exemple avec un capteur analogique.



Le capteur est connecté sur l'entrée n°0.  
La valeur retournée est stockée dans la variable A.

Le bouton **Tester le capteur** génère et transfère automatiquement un programme qui permet d'afficher la valeur retournée par le capteur en direct à l'écran, dans la fenêtre de débogage ci-dessous.



A	147	K	0
B	0	L	0
C	0	M	0
D	0	N	0
E	0	O	0
F	0	P	0
G	0	Q	0
H	0	R	0
I	0	S	0
J	0	T	0

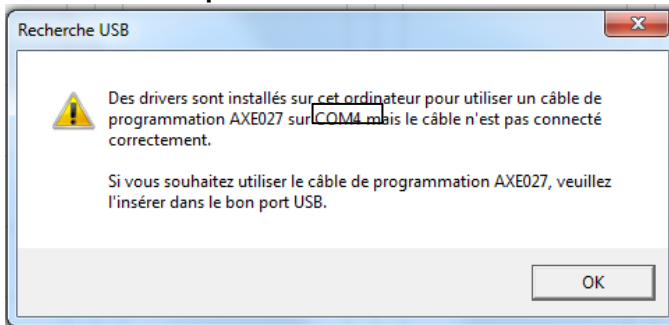
**Attention** : le programme transféré écrase le programme en cours. Si le microcontrôleur PICAXE est mis hors tension, il est possible de rappeler la fenêtre de débogage en cliquant sur **PIC > Débugger...**

## 6. Dépannage

### 6.1. Le câble de programmation AXE027

Lors du lancement de Logicator, il se peut que vous rencontriez des problèmes relatifs au câble de programmation AXE027. Voici différents cas de figure.

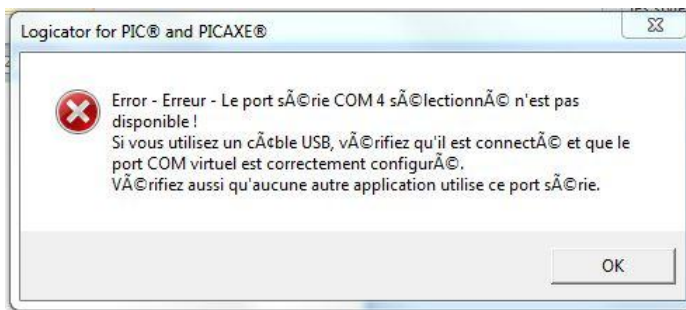
#### • Le câble n'est pas connecté



Brancher le câble sur le port USB indiqué.

Il est préférable de toujours brancher le câble de programmation AXE027 sur le même port.

#### • Le câble n'est pas connecté sur le bon port USB

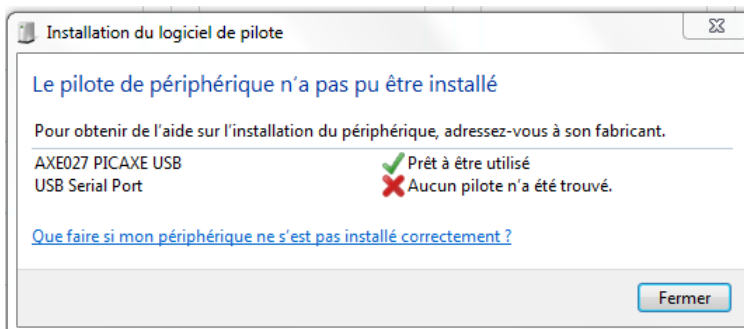
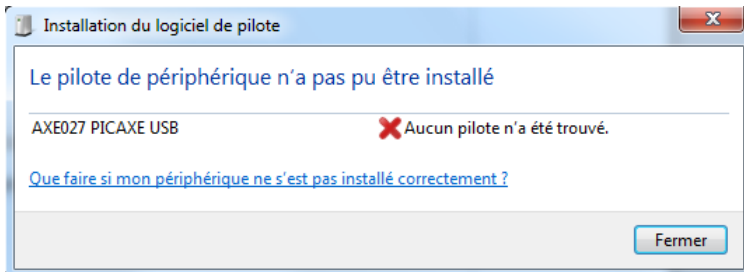


Brancher le câble sur un autre port.

Il peut arriver que le système ne reconnaisse pas certains ports USB.

Utiliser de préférence les ports situés à l'arrière de l'unité centrale. Eviter ceux qui sont sur l'écran directement ou sur la façade.

#### • Le pilote du câble de programmation AXE027 n'est pas installé correctement



Vous devez réinstaller le pilote pour le câble de programmation AXE027 :

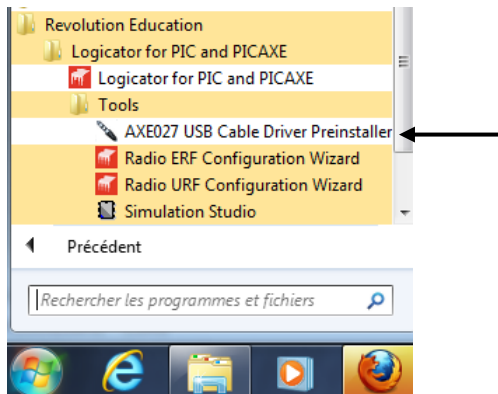
- avec l'assistant d'installation ;
- manuellement.

### 6.1.1. Réinstaller le pilote pour câble de programmation USB AXE027

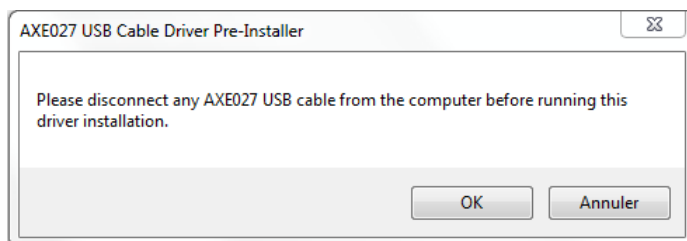
Si vous rencontrez des problèmes de reconnaissance du câble USB AXE027, il est possible de réinstaller le pilote USB.

#### Lancer l'assistant d'installation du pilote

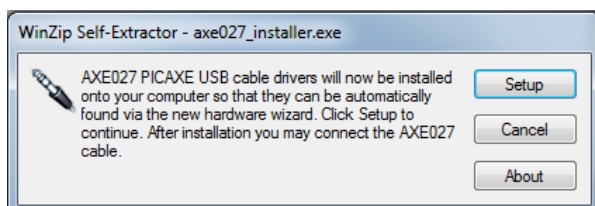
A partir du menu **Démarrer\Tous les programmes\Revolution Education\Logicator for PIC and PICAXE\Tools\AXE027 USB Cable Driver Preinstaller.**



L'écran suivant s'affiche :



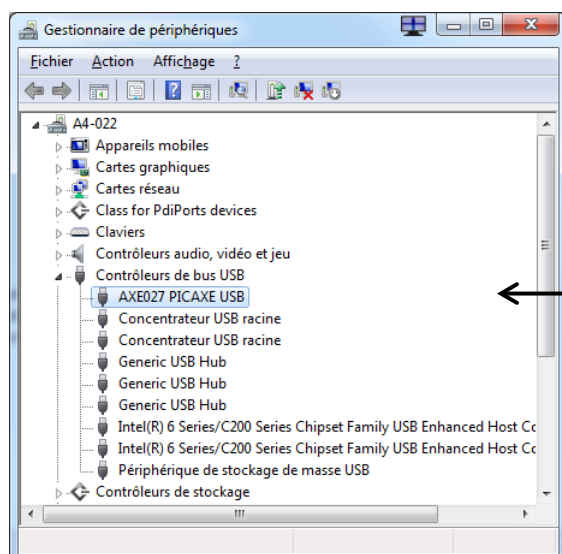
Déconnecter le câble de programmation et cliquer sur **OK**.



Cliquer sur **Setup** pour lancer l'installation du pilote.

#### Mettre à jour le pilote manuellement

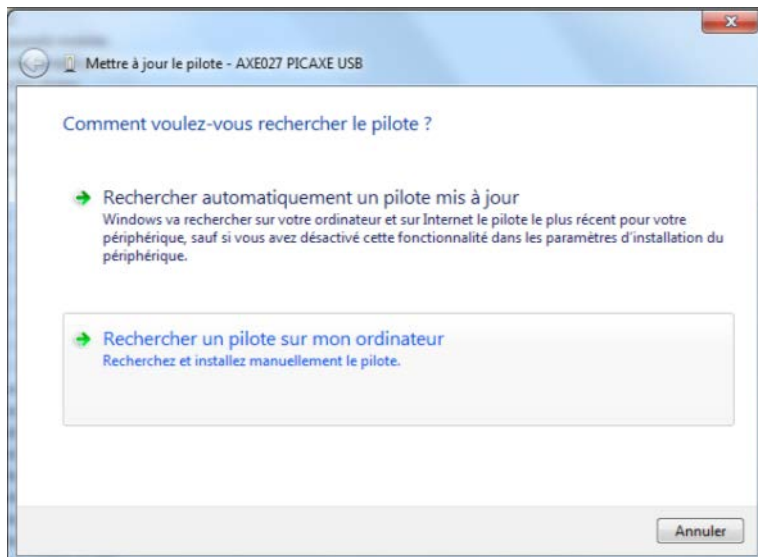
L'installation correcte du pilote est visible dans le **Gestionnaire de périphérique** dans la rubrique **Contrôleurs de bus USB**.



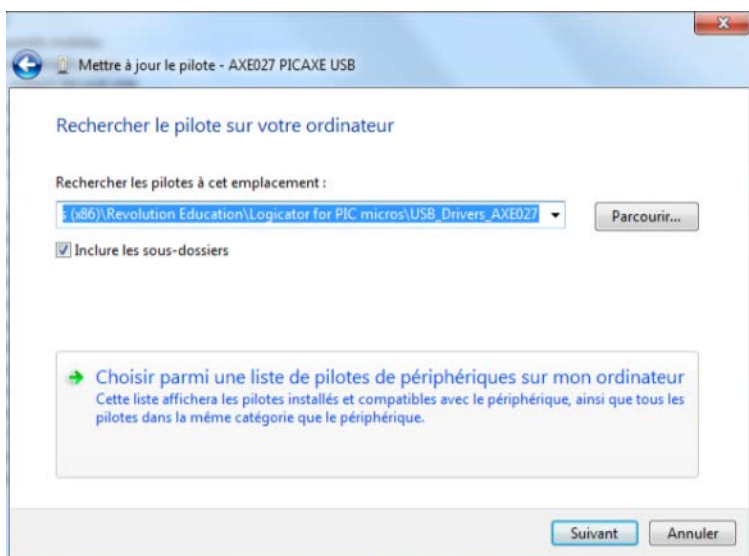
En cas de problème, le Gestionnaire de périphérique indique un point d'exclamation



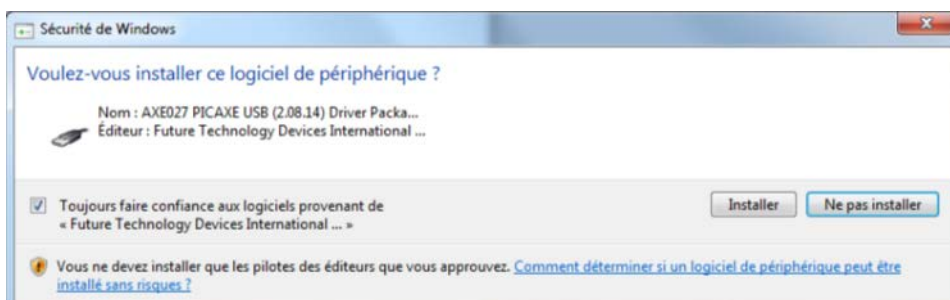
Sélectionner le câble de programmation puis faire un clic droit et cliquer sur **Mettre à jour le pilote...**



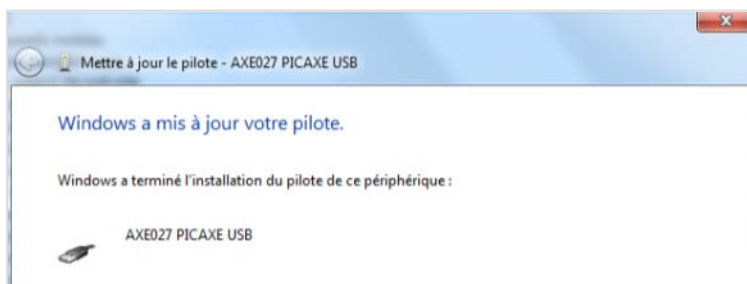
Cliquer sur **Rechercher un pilote sur mon ordinateur**.



Cliquer sur le bouton **Suivant**.



Confirmer la mise à jour du pilote en cliquant sur **Installer**.



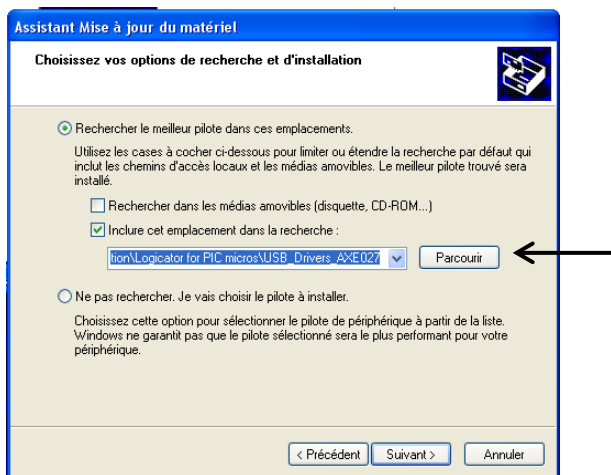


## Détection automatique du périphérique par Windows

Il arrive que Windows détecte automatiquement la présence d'un nouveau périphérique USB et propose l'assistant d'installation. Suivre la procédure ci-dessous :

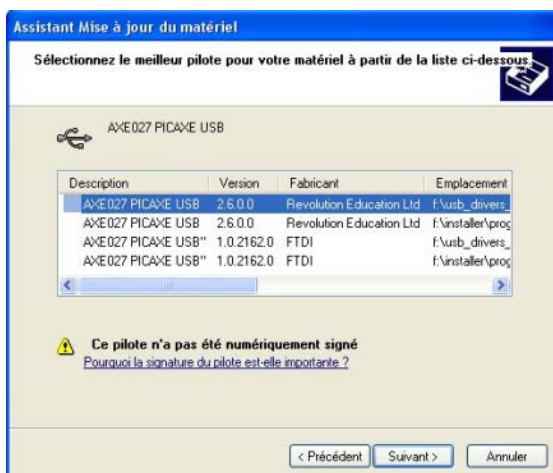


Sélectionner l'option **Installer à partir d'une liste ou d'un emplacement spécifié** puis cliquer sur **Suivant**.



Cocher l'option « **Inclure cet emplacement dans la recherche** ».

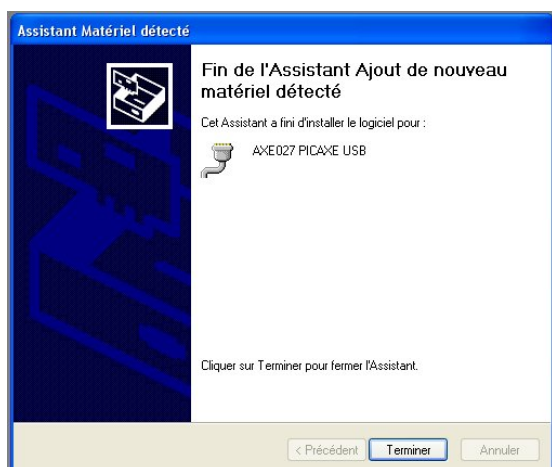
Cliquer sur **Parcourir**, aller chercher le pilote sur : C:\Program Files\Revolution Education\Logicator for PIC micros\USB\_Drivers\_AXE027 (si installation par défaut dans cet emplacement). Cliquer sur **Suivant**.



Après quelques secondes, Windows propose d'installer le câble AXE027. Cliquer sur **Suivant**.



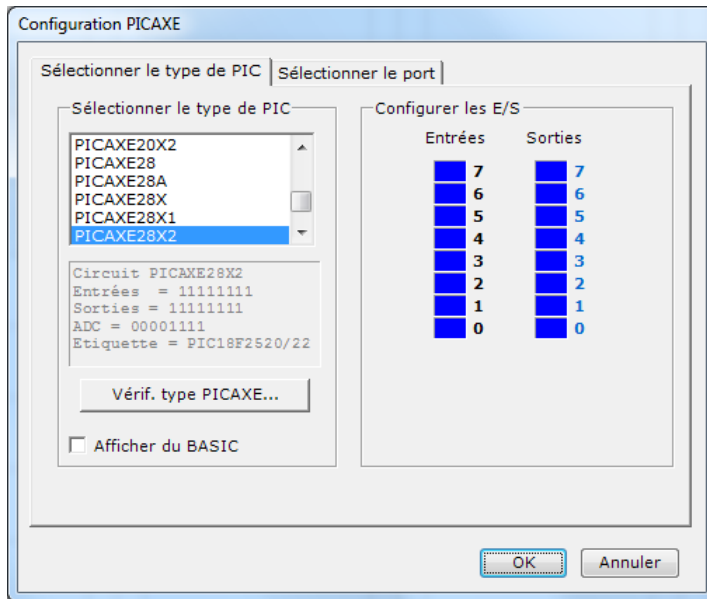
Windows vérifie la compatibilité du pilote. Cliquer sur **Continuer**.  
Si Windows redemande l'installation d'un pilote, continuer jusqu'à l'apparition de l'écran suivant :



Cliquer sur **Terminer**.

## 6.2. Vérifier le dialogue entre le PC et le microcontrôleur PICAXE

Logicator permet de vérifier si le PC peut dialoguer avec le PICAXE. A partir du menu **Options**, cliquer sur **Sélectionner le type de PIC...**



Cliquer ensuite sur le bouton **Vérif. Type PICAXE...**

Le dialogue est possible



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



En cas de problème, vérifier que :

- le câble de programmation AXE027 est correctement connecté ;
- le microcontrôleur PICAXE est sous tension ;

**Note** : si le problème persiste après ces vérifications, il se peut qu'un programme déjà transféré monopolise les ressources du microcontrôleur PICAXE et empêche le dialogue avec l'ordinateur. Mettre hors tension le microcontrôleur PICAXE, relancer le transfert du programme et remettre le microcontrôleur PICAXE sous tension dans les 3 secondes.

### Fiche d'évolution du guide utilisateur

Le logiciel 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](http://www.a4.fr).

Version	Date	Description
1.0	30/11/2012	Version initiale du guide d'utilisation en français basé sur la version V3.6.2.