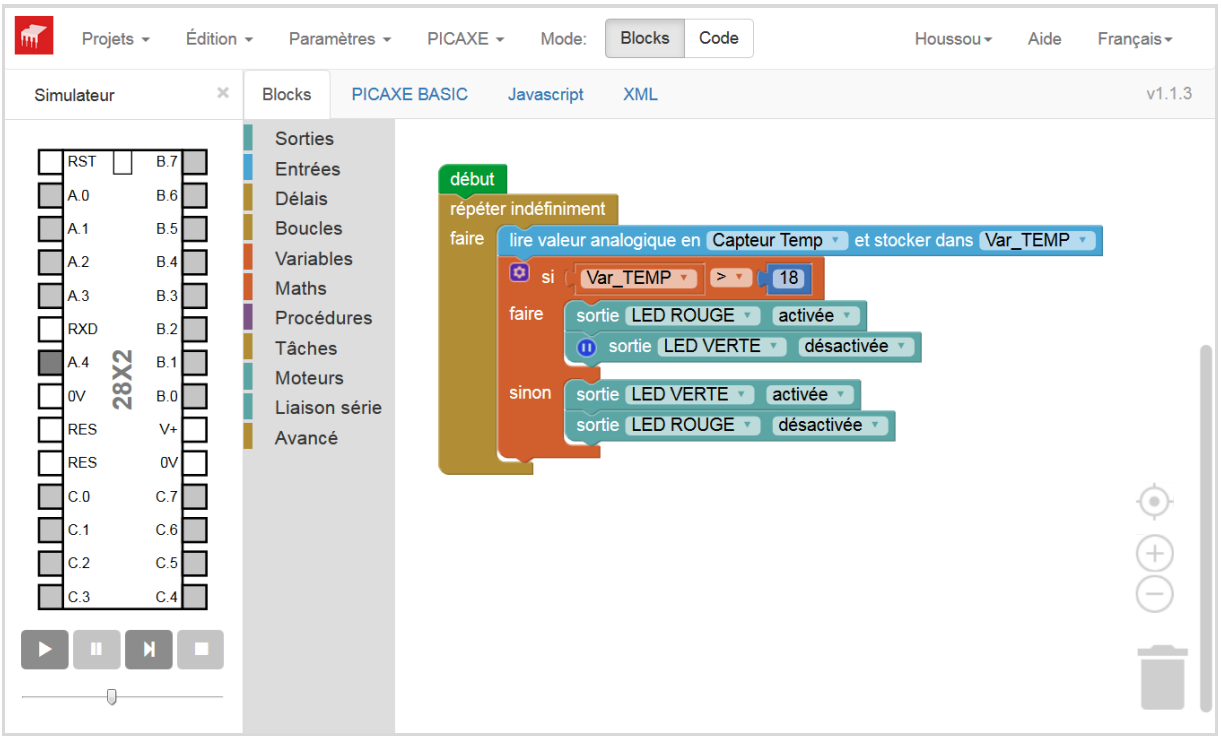
Blockly pour PICAXE
Guide utilisateur



Simuler et programmer un microcontrôleur PICAXE

Traduction : Hervé BLOREC
Relecture et adaptation : A4 Technologie.

TECHNOLOGIE
www.a4.fr

Introduction

Blockly pour PICAXE est un outil puissant de programmation graphique permettant de générer des programmes pour les microcontrôleurs PICAXE. En empilant des blocs de couleur, une commande peut être générée rapidement. Cette méthode simple de programmation par glisser/déposer permet aux étudiants de développer rapidement des séquences de commandes.

Blockly peut être exécuté sur un navigateur web sur n'importe quel équipement muni d'une connexion internet. Pour un usage hors ligne, Blockly est incorporé dans PICAXE Editor, il est également disponible comme une application Chrome autonome.

Blockly peut être utilisé de 3 manières :

1) *Au sein de PICAXE Editor 6 (PE6) qui est l'environnement principal de programmation de PICAXE (Windows). Téléchargeable gratuitement www.picaxe.com*

2) *Comme une application autonome de Chrome « PICAXE Blockly » (Windows/Mac/Linux/Chromebook).*

3) *En ligne, à partir de n'importe quel navigateur internet, sur : www.picaxecloud.com.*

Blockly fonctionne de la même façon sur les 3 plateformes. Le moteur de simulation de PE6 est plus performant.

La large gamme de blocs spécifiques PICAXE permet à l'utilisateur de commander des équipements extérieurs, tels que moteurs et LED qui sont connectés au microcontrôleur PICAXE.

Cette section du manuel explique comment sont utilisés les blocs les plus courants, donnant des exemples et des techniques dans le cadre de projets scolaires possibles.

Démarrage Rapide

Si vous n'êtes pas familier avec l'approche de programme pour la construction de systèmes de commande, il est préférable de commencer par vous familiariser avec les blocs les plus communément utilisés : *Sorties, Délai, Moteur et Entrée.*

1. Comment construire, modifier et tester un programme

2. Sorties

Cette section montre comment :

- Activer ou désactiver les sorties d'un microcontrôleur PICAXE, en utilisant : les blocs *Sorties, Moteur, Son, Jouer* ;
- utiliser le bloc *Serout* pour sortir des informations série depuis le microcontrôleur PICAXE.

3. Entrées

Cette section montre comment :

- vérifier l'état de capteurs digitaux connectés au microcontrôleur PICAXE par l'utilisation du bloc *entrée* ;
- utiliser le bloc *Interruption* pour une réponse instantanée à des capteurs numériques *;*
- utiliser le bloc *variable de décision* pour faire usage de lectures provenant de capteurs analogiques connectés à un microcontrôleur PICAXE, dans un système de commande.

4. Délais

Cette section montre comment créer des retards par l'utilisation de *pause* et *veille.*

5. Procédures

Cette section montre la technique importante de construction d'un système de commande comme une suite de sous-systèmes liés.

6. Maths & Variables

Cette section montre comment :

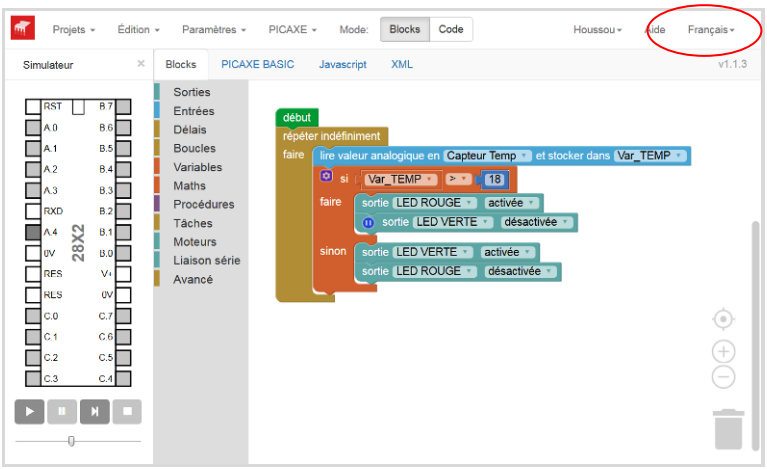
- créer un système de comptage avec les blocs *Incrémenter* et *Décrémenter ;*
- intégrer un calendrier dans un système de commande ;
- utiliser les blocs *Expression* et *Aléatoire* pour donner une valeur à une variable ;
- utiliser les blocs *Lire* et *Écrire* pour stocker et accéder à une valeur de variable utilisant la mémoire EEPROM du microcontrôleur PICAXE.

7. Avancés

Cette section montre comment utiliser les blocs de commande PICAXE les plus avancés.

Section 1. Comment construire, modifier et simuler dans Blockly

Avant de commencer la navigation, choisissez la langue française en haut à droite de l'écran.

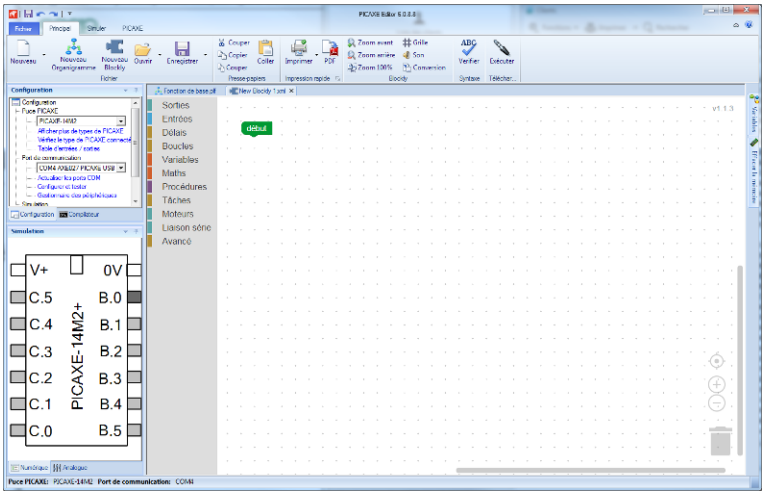


Simulateur - Exécute le programme à l'écran en mode pas à pas en mettant en surbrillance les blocs exécutés. Il est possible d'agir directement sur les entrées du microcontrôleur PICAXE et de vérifier l'effet de l'exécution du programme sur les sorties.

Blocks - Collection de blocs de commandes disponibles pour constituer le programme.

Espace de Travail - Zone à droite où vous créez votre programme.

Note : dans PE6, le Panneau Simulation est légèrement différent, mais réalise la même tâche.



Créer un projet

- Dans PICAXE Blockly, à partir de la barre des menus, cliquer sur **Fichier / Nouveau Projet**.
- Dans PE6, à partir de la barre des menus de l'onglet **Principal**, cliquer sur **Nouveau Blockly.**
- Dans l'application PICAXE Blockly en ligne, à partir de la barre des menus, cliquer sur **Projets/Nouveau projet**.

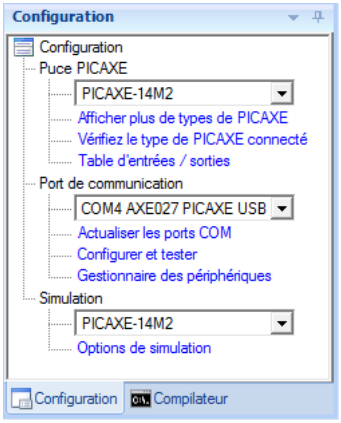
Sélectionner le type de PICAXE

Avant de créer un programme, il faut définir le microcontrôleur PICAXE.

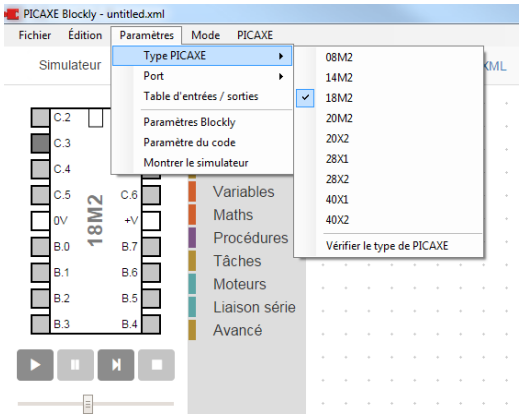
Si vous utilisez Blockly au sein de PE6 ou PICAXE Blockly en tant qu'application autonome, vous devez également sélectionner le port COM du câble de programmation.

Attention ! Les navigateurs de web ne vous permettent pas d'accéder au port USB de votre ordinateur. C'est une restriction de sécurité très sensible. Par conséquent, la version de Blockly en ligne sur le Cloud ne permet pas la programmation directe de votre microcontrôleur.

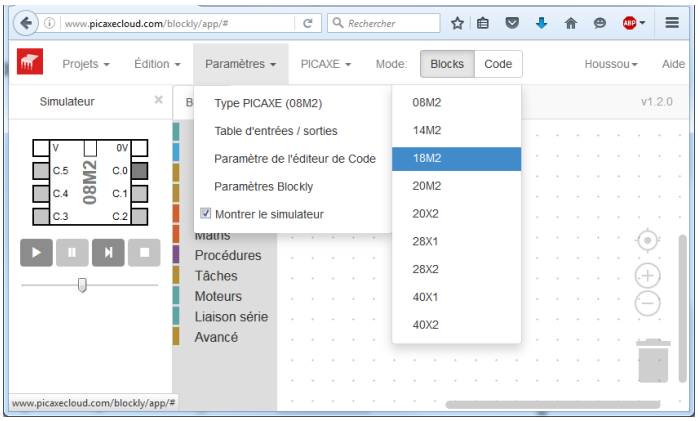
PE6 – *à partir de la fenêtre* ***Configuration Paramètres.***



PICAXE Blockly – *à partir du menu*



Blockly Cloud - *à partir du menu* ***Paramètres.***



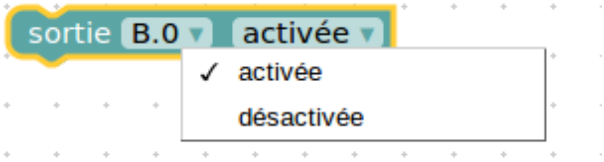
Note : Les blocs de commandes proposent des listes déroulantes qui permettent de sélectionner les entrées/sorties du PICAXE que l'on souhaite exploiter. Attention, les listes dépendent du PICAXE sélectionné.

Remarque: cette section ne traite que du programme. L''utilisation des blocs individuels est détaillée plus loin.

Ajouter un nouveau bloc

Glisser un bloc depuis l'onglet **Blocks** vers l'espace de travail.

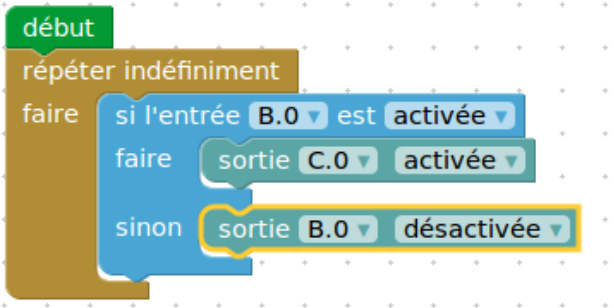
La plupart des blocs a une liste déroulante des options utilisables pour définir le mode de fonctionnement du bloc.



D'autres blocs ont une entrée «pièce du puzzle" où un autre bloc peut être imbriqué. Par exemple, vous pouvez déposer une constante (Nombre) ou une variable dans ce bloc.

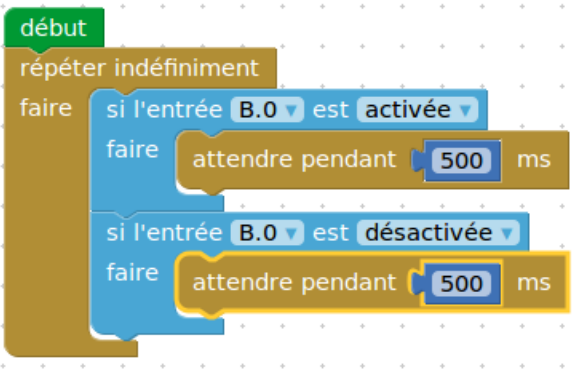
Si un bloc est manquant le symbole **Attention** apparaît. Cet avertissement disparaît automatiquement lorsque le bloc est inséré.

Les boucles et les blocs de décision permettent également d'empiler d'autres blocs à l'intérieur d'eux-mêmes, par exemple :



Bloc 

Un bloc **début** marque le point de départ du programme.



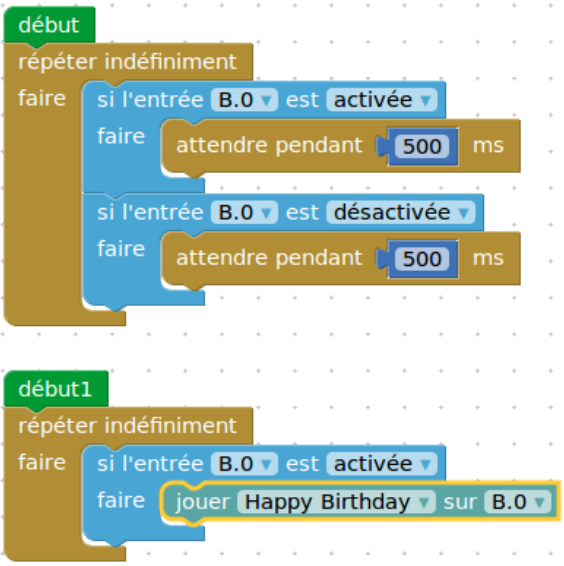
Lorsque le microcontrôleur PICAXE est réinitialisé ou mis sous tension, le programme commence au premier bloc **début**.

Chaque programme doit avoir au moins un bloc **début**.

L'exécution du programme cesse lorsqu'un bloc est atteint.

Pour les références PICAXE-M2, vous pouvez avoir jusqu'à 8 blocs **début** sur chaque programme.

Pour ajouter des nouveaux blocs **début**, aller dans l'onglet **Blocks**, rubrique **Tâches**.



Déplacer des blocs

Pour déplacer un bloc simple ou une pile de blocs, sélectionner le bloc supérieur et le déplacer à sa nouvelle position.

Zoom & Suppression

En bas à droite de l'écran Blockly, il y a 4 icônes.

Zoom 100 % au centre

Zoom avant
(Ctrl + ou Ctrl molette)

Zoom arrière
(Ctrl – ou Ctrl molette)

Suppression

Pour supprimer un bloc, vous pouvez le sélectionner et :

1) le glisser dans la poubelle ;
2) cliquer sur la touche **Suppression** du clavier ;
3) cliquer droit et cliquer sur « Supprimer le bloc ».

Note : le premier bloc **début** ne peut être supprimé.

Couper, copier et coller

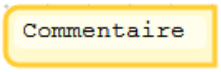
Utiliser les options Couper, Copier, Coller depuis le menu **Édition** pour couper ou coller les blocs sélectionnés ou la pile de blocs et les coller soit à un autre emplacement du même programme ou dans un autre programme.

Autrement vous pouvez cliquer-droit et **Dupliquer**.

Grille

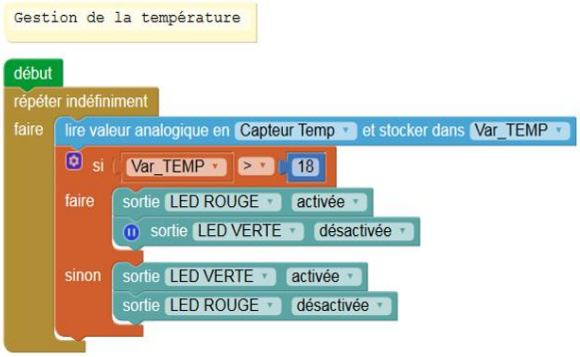
La grille peut être masquée ou affichée via les **Paramètres Blockly**. Lorsque la grille est affichée les blocs sont automatiquement « aimantés » au point de grille le plus proche.

Bloc



Il peut être utile d'insérer des commentaires dans votre programme.

Les blocs **Commentaire** se trouvent dans l'onglet **Blocks**, rubrique **Avancé**.

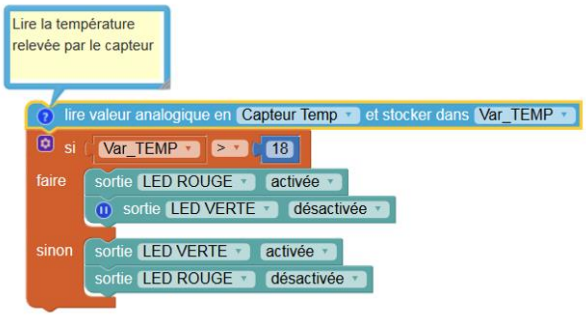


Commenter un bloc

Il peut être utile d'ajouter un commentaire à un bloc.

Cliquer droit sur le bloc puis cliquer sur « Ajouter un commentaire ».

Un point d'interrogation apparaît. Cliquez dessus et saisissez le commentaire.

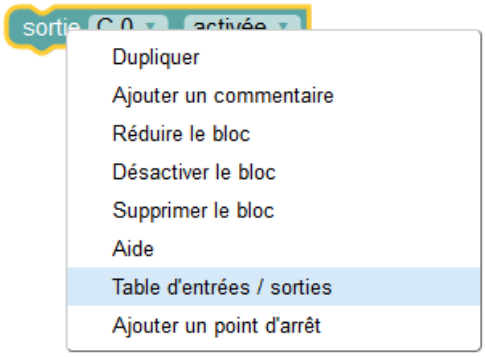


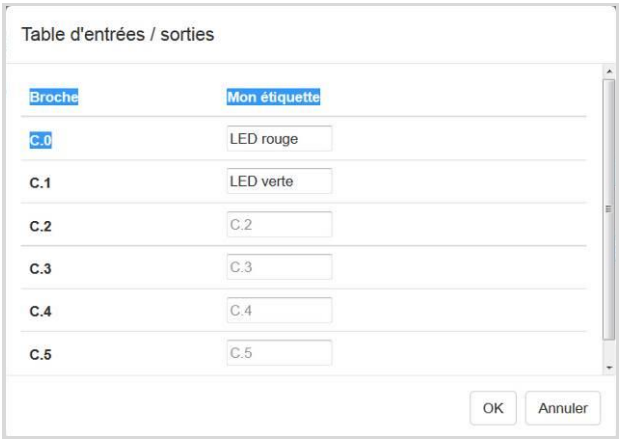
Le commentaire n'affecte pas le fonctionnement du bloc.

Nommer les entrées/sorties

Pour faciliter la lecture d'un programme, il est utile d'associer un nom « parlant » à chaque entrée/sortie.

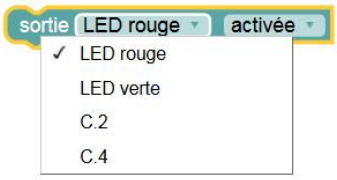
Cliquer droit sur un bloc puis sélectionner « Table d'entrées/sorties » ou sélectionner cette option à partir du menu **Paramètres**.





Vous devez renseigner le nom de chaque entrée/sortie dans la colonne « Mon étiquette ».

Les noms renseignés sont mémorisés et apparaissent dans toutes les listes déroulantes.



Comment tester l'exécution d'un programme

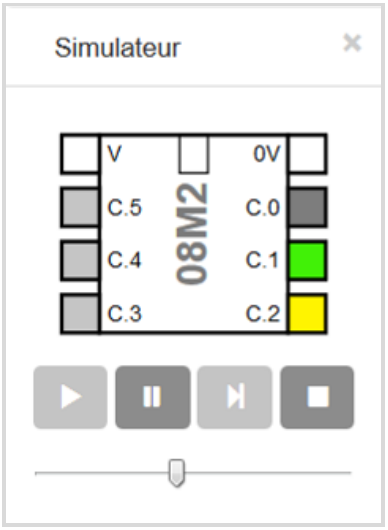
Le simulateur dispose d'un certain nombre de fonctionnalités qui vous permettent de tester l'exécution d'un programme.

1. Le Panneau Simulateur

Quand un programme s'exécute, le panneau Simulateur permet d'agir sur les entrées et de visualiser le changement d'état des sorties.

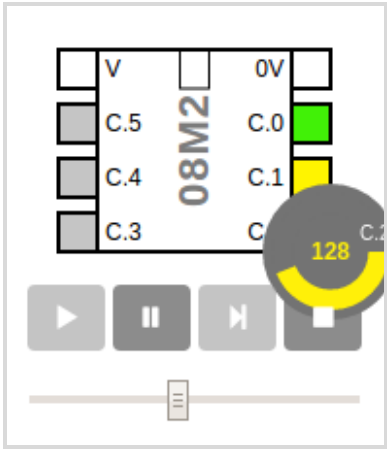
2. Simulation d'entrées numériques

Pour changer l'état d'une entrée, cliquer directement sur l'entrée souhaitée dans le simulateur. Elle change de couleur : gris (désactivée) à jaune (activée).



3. Simulation d'entrées analogiques

Pour changer la valeur d'une entrée analogique, cliquer droit sur l'entrée pour afficher le curseur circulaire. Tourner le curseur jusqu'à la valeur souhaitée (0 à 255).



4. Exécution et Arrêt

Pour tester l'exécution d'un programme, vous disposez des boutons ci-dessous.

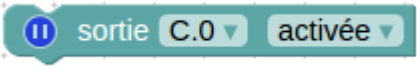
Quand la simulation est lancée, les instructions exécutées sont mises en surbrillance.

Le curseur permet d'agir sur la vitesse d'exécution du programme.

5. Points d'Arrêts

On peut forcer l'arrêt de la simulation sur une instruction donnée. Cliquer droit sur le bloc puis « Ajouter un point d'arrêt ».

Un symbole apparaît.

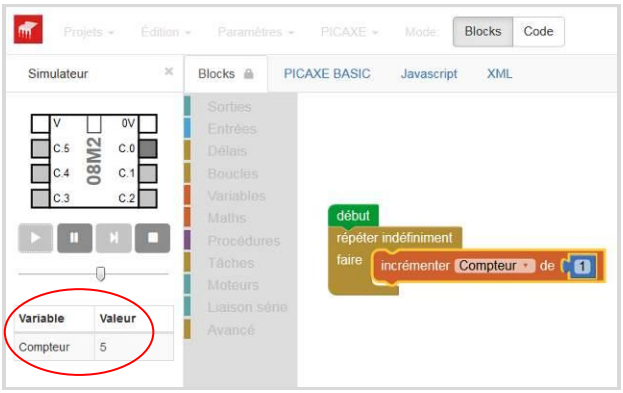


Quand la simulation atteint ce point, elle se met en pause. Pour réactiver la simulation, cliquer sur **Lecture**.

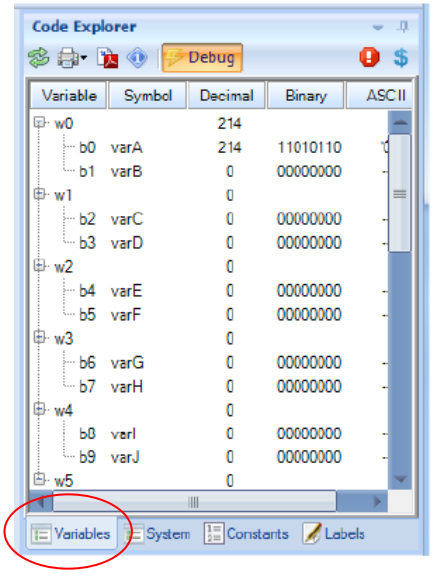
6. Affichage des variables

Pendant l'exécution de votre programme, il peut être utile de visualiser les valeurs des variables utilisées.

- Dans PICAXE Blockly, les variables apparaissent dans le panneau **Simulateur**.



- Dans PE6, les variables sont visualisées dans l'explorateur de code.



Télécharger un programme dans un microcontrôleur PICAXE

PE6 et Application (non Cloud)

1. Connecter votre carte PICAXE à l'ordinateur avec le câble de programmation AXE027.
2. Alimenter votre carte.

   Note : le téléchargement d'un nouveau programme écrase un programme précédemment chargé.

3. Cliquer le bouton **Exécuter** sur la barre d'outils ou presser <F5>.
4. La fenêtre de progression de chargement apparaît.
5. Le temps de téléchargement dépend du type de microcontrôleur et du code à programmer.
6. La barre de progression disparaît lorsque la programmation est terminée.

Si vous rencontrez des difficultés lors du téléchargement, consulter le manuel PICAXE 1 chapitre « Procédure Réinitialisation matériel ». www.picaxe.com/docs/picaxe_manual1_fr.pdf

PICAXE Blockly sur le cloud

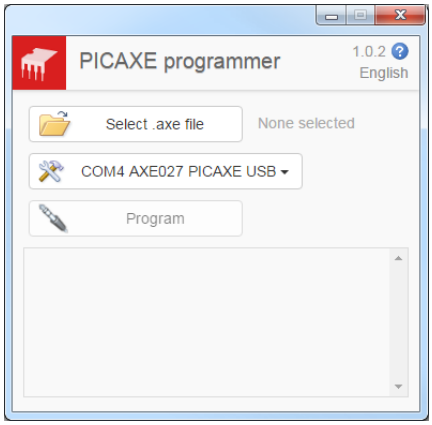
Les navigateurs internet ne vous permettent pas d'accéder au port USB de votre ordinateur. C'est une restriction de sécurité très sensible.

Par conséquent, la version de Blockly en ligne sur le Cloud ne permet pas la programmation directe de votre microcontrôleur (contrairement à PE6 et à l'application PICAXE Blockly).

Vous devez ensuite utiliser l'application **Chrome Programmer App** pour télécharger le fichier .axe dans votre microcontrôleur PICAXE.

www.picaxe.com/Software/Drivers/PICAXE-Programmer-App/

Utilisation de Cloud Programmer App



1. Connecter votre carte PICAXE à l'ordinateur avec le câble de programmation AXE027.
2. Alimenter votre carte.

   Note : le téléchargement d'un nouveau programme écrase un programme précédemment chargé.

3. Sélectionner le fichier .axe et le port COM affecté au câble de programmation AXE027.
4. Cliquer sur le bouton **Program.**
5. La fenêtre de progression de chargement apparaît.
6. Le temps de téléchargement dépend du type de microcontrôleur et du code à programmer.
7. La barre de progression disparaît lorsque la programmation est terminée.

Si vous rencontrez des difficultés lors du téléchargement, consulter le manuel PICAXE 1 chapitre « Procédure Réinitialisation matériel ». www.picaxe.com/docs/picaxe_manual1_fr.pdf

Affichage et utilisation du BASIC

Blockly convertit tout programme complet en BASIC ou Javascript.
Le BASIC est un langage de programmation textuelle utilisé pour tout programmer tout microcontrôleur PICAXE ou PC.
Le Javascript est un langage de programmation utilisé pour le développement de sites Web.

Pourquoi convertir ?

Le langage de programmation par blocs est facile à comprendre et rapide à mettre en œuvre.

Le langage de programmation en BASIC permet d'élaborer des programmes plus complexes pour des utilisateurs de niveau avancé.

La conversion permet de se familiariser avec le BASIC.

Conversion d'un programme en BASIC

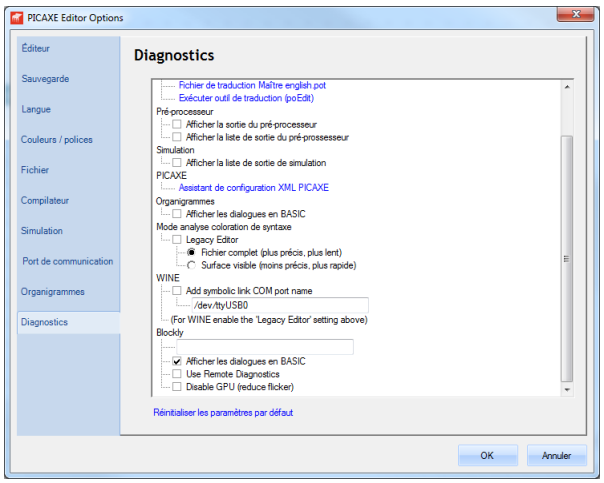
Concevoir un programme et le tester à l'aide des outils de simulation.

Pour le convertir en BASIC PICAXE :

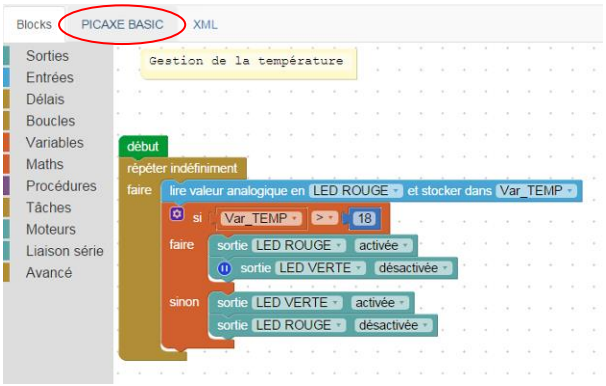
- Dans PICAXE Blockly, cliquer sur l'onglet **PICAXE BASIC**.
- Dans PE6, cliquer sur le bouton **Conversion** à partir de l'onglet **Principal**.

La fenêtre de texte BASIC est alors affichée contenant la conversion de votre programme.

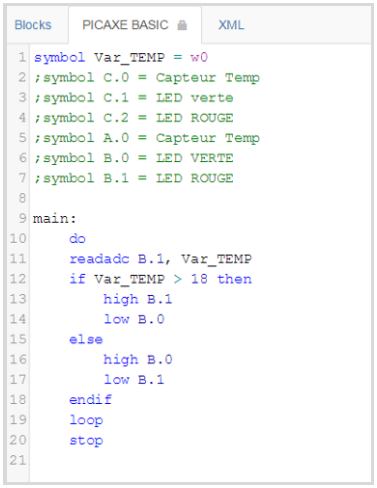
Il est aussi possible de faire apparaître l'onglet BASIC dans PE6. A partir du menu **Fichier, sélectionner Options / Diagnostics**.



A partir de la rubrique Blockly, cocher « Afficher les dialogues en BASIC ».



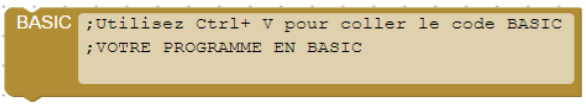
En cliquant sur l'onglet **PICAXE BASIC**, vous visualisez en direct la conversion de votre programme en BASIC.



Remarque : seuls les blocs connectés au bloc **début** dans votre programme sont convertis.

Il n'est pas possible de convertir un programme BASIC en programme Blockly.

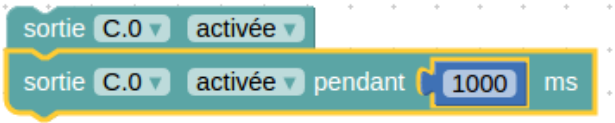
L'utilisation de bloc **BASIC** dans la rubrique **Avancé** vous permet d'ajouter des sections de code BASIC dans votre programme.



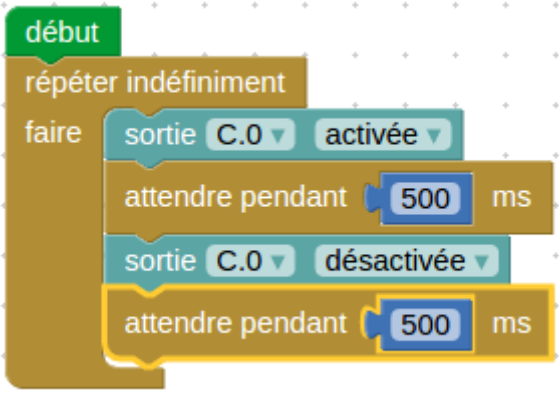
Pour plus d'informations sur l'utilisation de BASIC pour programmer les microcontrôleurs PICAXE, se reporter au manuel PICAXE 2 « Commandes du BASIC ».
www.picaxe.com/docs/picaxe_manual2_fr.pdf

Section 2. Sorties

Blocs activation/désactivation des sorties



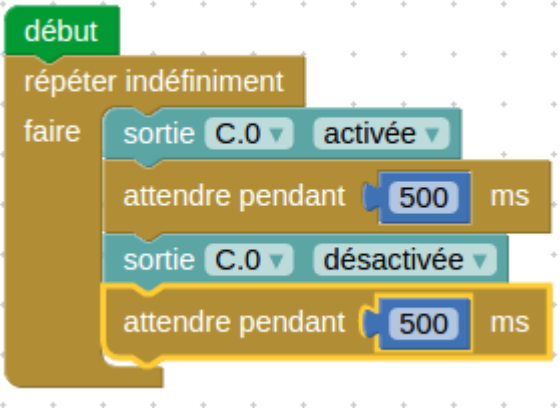
Ces blocs sont utilisés pour activer ou désactiver une sortie.



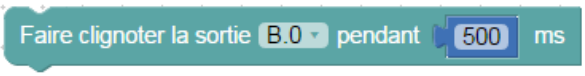
Note : vous pouvez attribuer un nom à une sortie à partir de la table d'entrées/sorties.
Se reporter au chapitre « Nommer les entrées/sorties » de ce document.

Bloc

Un bloc **attendre pendant** permet d'introduire un temps d'attente exprimé en millisecondes avant l'exécution du bloc suivant.



Bloc Faire clignoter la sortie

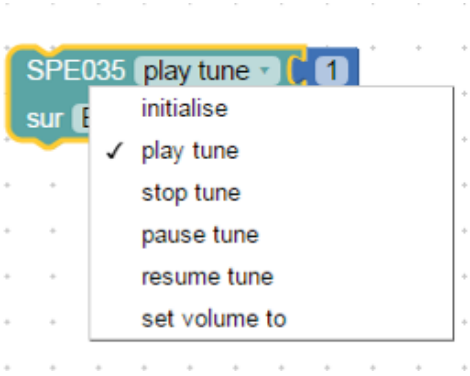


Un bloc **Faire clignoter la sortie** permet de commuter la sortie pendant un laps de temps donné.

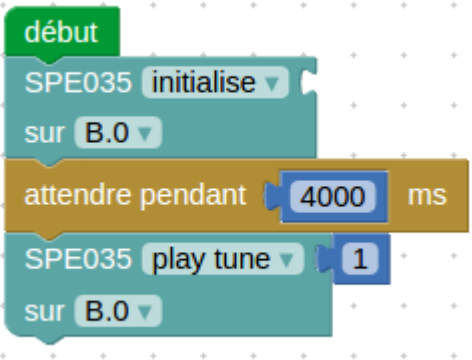
Bloc SPE035

Le lecteur MP3 SPE035 est un module économique permettant d'écouter des fichiers MP3 stockés sur une carte micro SD (non fournie avec le module).

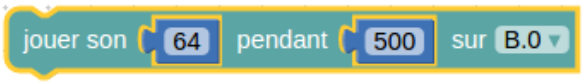
Le bloc SPE035 permet de piloter ce module. Vous disposez de plusieurs options :



Dans l'exemple ci-dessous, l'instruction « play tune 1 » déclenche la lecture du fichier « 1.mp3 » stocké sur la carte micro SD.

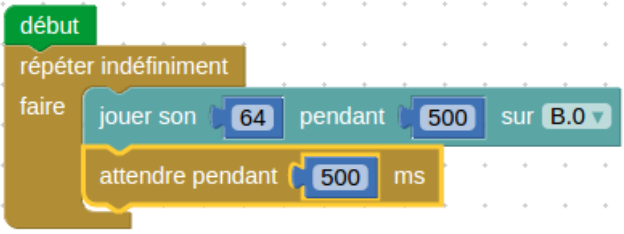


Bloc jouer son



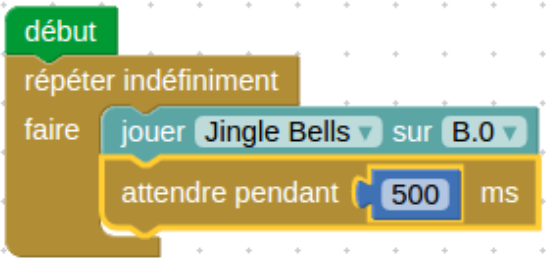
Le bloc **jouer son** permet d'émettre un son avec un buzzer piézo électrique connecté à une sortie du microcontrôleur PICAXE.

La fréquence et la durée du son émis dépendent des valeurs renseignées.



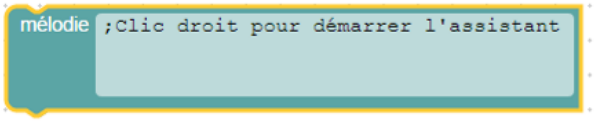
Bloc jouer

Le bloc **jouer** permet de déclencher la lecture de musiques préprogrammées dans la plupart des microcontrôleurs PICAXE.



Note : ces morceaux préprogrammés occupent une place très restreinte dans la mémoire du microcontrôleur PICAXE.

Bloc mélodie



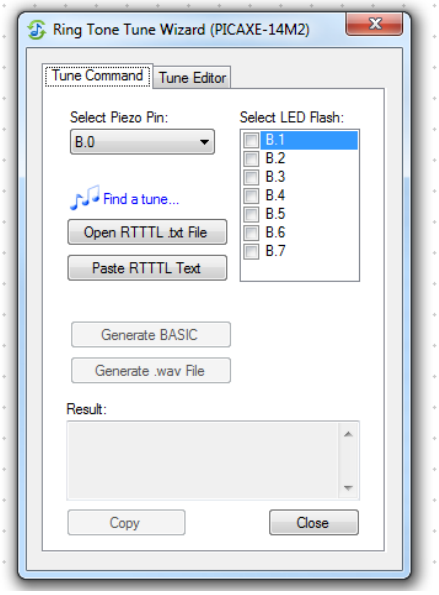
Le bloc **mélodie** permet de jouer des airs musicaux spéciaux provenant de fichiers RTTTL (fichiers de sonneries pour téléphones portables).

Vous trouvez une multitude de fichiers RTTTL sur internet, téléchargeables librement. Ils se présentent sous forme de fichiers texte qui contiennent une succession de notes et de durées qui constituent une mélodie.

Vous disposez d'un assistant pour convertir ces sonneries en blocs **mélodie** PICAXE.

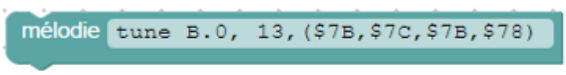
Dans PE6, cliquer droit sur le bloc **mélodie** pour démarrer l'assistant.

Dans l'application PICAXE Blockly, vous devez démarrer séparément l'Assistant.



Cliquer sur **Open RTTTL.txt File** pour sélectionner le fichier.

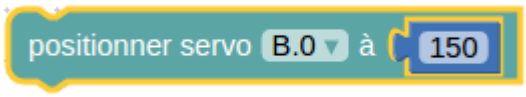
Une fois que vous avez généré la mélodie BASIC, cliquer sur **Copy** et coller le code dans le bloc **mélodie**.



L'assistant vous permet de sélectionner une sortie destinée à animer le clignotement d'une LED.

Note : si vous souhaitez jouer plusieurs fois une mélodie, utiliser le bloc **mélodie** dans une procédure pour économiser la mémoire.

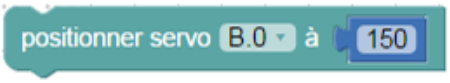
Blocs servo



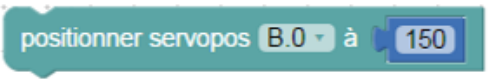
Un servomoteur est un actionneur équipé d'un moteur miniature et d'un motoréducteur. Un module électronique interne permet de gérer sa position de manière précise.
Il reçoit une consigne de position standardisée sous la forme d'impulsions de 0,75 à 2,25 répétées toutes les 20 ms.

Il existe deux instructions pour piloter un servomoteur :

- Le bloc **positionner servo** permet d'activer le servomoteur pour définir sa position initiale.



- Le bloc **positionner servopos** permet ensuite de modifier cette position.

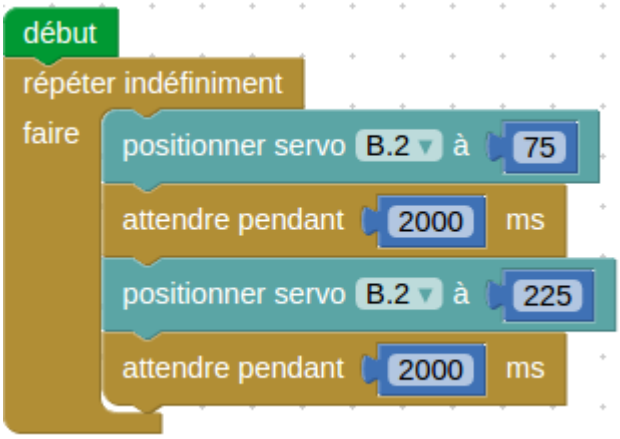


Les blocs **servo** ont deux paramètres : la broche de sortie à laquelle le servo est connecté et la durée de l'impulsion. La valeur de l'impulsion doit être comprise entre 75 et 225.

Note : si vous souhaitez modifier fréquemment la position d'un servomoteur, il est préférable d'utiliser le bloc **positionner servo** une seule fois pour procéder à son initialisation, puis d'utiliser des blocs **positionner servopos**.

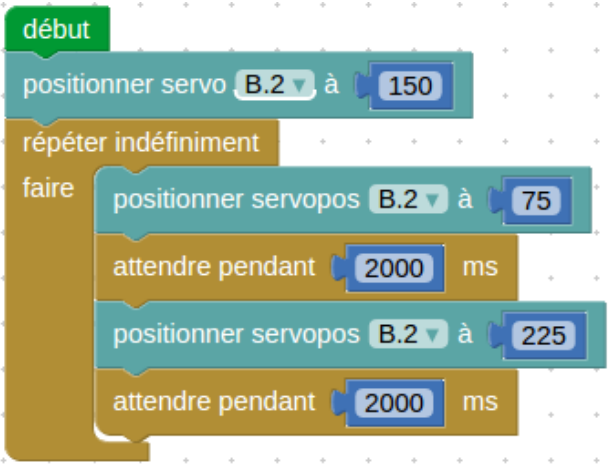
Exemple

Le programme ci-dessous active un servomoteur connecté à la broche B.2, d'une butée à l'autre, en continu.

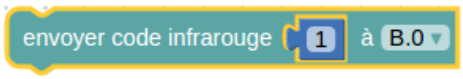


*Utilisation du bloc **Servo***

Note : le bloc **Servo** est requis pour initialiser la pulsation servo. Une fois la pulsation initialisée, il faut utiliser l'instruction **Servopos** pour les mouvements suivants.



Bloc envoyer code infrarouge



Le bloc **envoyer code infrarouge** est utilisé pour transmettre des données infrarouges à un équipement au protocole Sony®.

Il peut aussi être utilisé pour transmettre des données à un autre récepteur infrarouge qui utilise le bloc **lire valeur infrarouge**. La donnée est transmise via une LED infrarouge (connectée à la sortie 0) utilisant le protocole SIRC (Sony Infrared Control).

Le bloc **envoyer code infrarouge** peut être utilisé pour transmettre n'importe quel code de données TV valides (0-127).

Note : le protocole Sony utilise seulement 7 bits pour la donnée, ainsi les codes données de valeurs 128 à 255 ne sont pas valides.

Bloc signal pwm

Le bloc **signal pwm** est utilisé pour fournir sur une sortie un signal périodique avec un rapport variable. C'est souvent utilisé pour contrôler la vitesse des moteurs.

Pour plus d'informations sur l'utilisation de BASIC pour programmer les microcontrôleurs PICAXE, se reporter au manuel PICAXE 2 « Commandes du BASIC » - chapitre « Pwmout ».
www.picaxe.com/docs/picaxe_manual2_fr.pdf

Bloc initialiser les broches

Quand le déroulement du programme arrive à un bloc **initialiser les broches**, le port de sortie est initialisé à la valeur binaire du nombre entré dans le bloc.

Si vous êtes familier avec le système binaire alors le bloc initialisation broche est un moyen aisé de combiner les commutations de sorties On ou Off.

bit	7	6	5	4	3	2	1	0
value	128	64	32	16	8	4	2	1

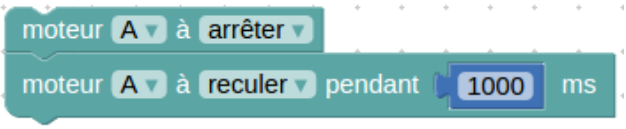
Exemple : Mettre le port B à 4 correspond à allumer la LED sur B.2 (et coupera toutes les autres sorties).

Bloc initialiser le type des broches

Le bloc **initialiser le type des broches** convertit les broches en entrée ou en sortie. Un bit de valeur binaire 1 signifie sortie, une valeur de 0 signifie une entrée

Exemple : Les commandes telles que **Activer sortie** initialiseront automatiquement le bit dirs en une sortie.3.

Blocs Moteur



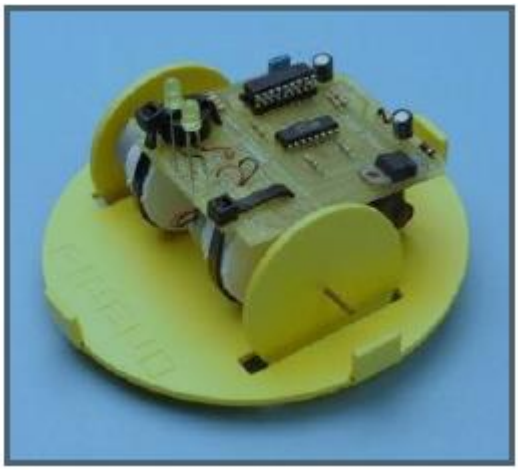
Le **bloc Moteur** permet de gérer deux sorties sur un microcontrôleur PICAXE pour commander un moteur en avant, en arrière, arrêt.

Les moteurs reçoivent une lettre de A à D, qui commandent les sorties comme suit :

	Non-8 pin	08M2
A	(B.0, B.1)	(C.0, C.1)
B	(B.2, B.3)	(C.2, C.4)
C	(B.4, B.5)	
D	(B.6, B.7)	

Rappelez-vous que le sens dans lequel le moteur tourne dépend du sens du courant qui le traverse, et donc de la manière dont il est connecté à l'alimentation. Donc si le moteur tourne dans le mauvais sens, il peut être nécessaire d'inverser les deux fils.

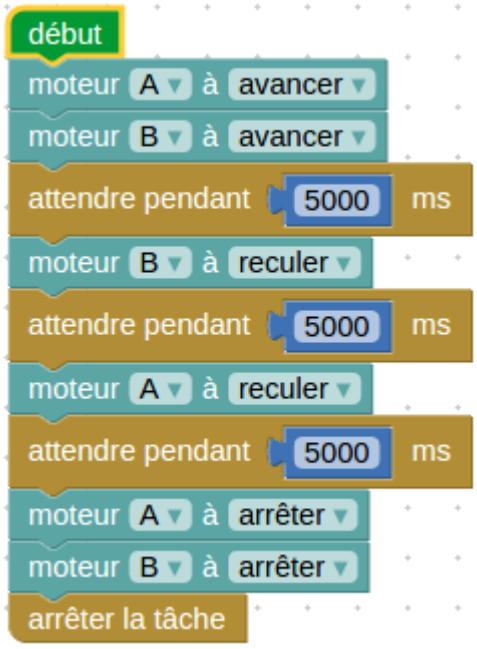
Note : **Output** et **bloc Moteur** utilisent les mêmes broches de sortie pour commuter les sorties d'un microcontrôleur PICAXE.



Exemple :

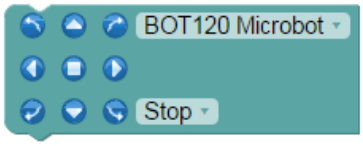
Un buggy orientable est généralement entraîné par deux moteurs, l'un alimentant chaque roue motrice avec une roue jockey libre pour le garder stable.

Le programme ci-dessous montre comment une séquence de blocs moteur peut être utilisée pour conduire un buggy qui a un moteur connecté aux sorties 0 et 1 (moteur A) et l'autre Moteur connecté aux sorties 2 et 3 (moteur B).

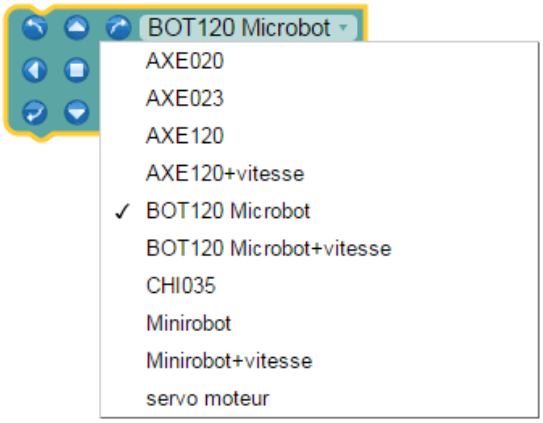


Blocs Moteur Robot (variante)

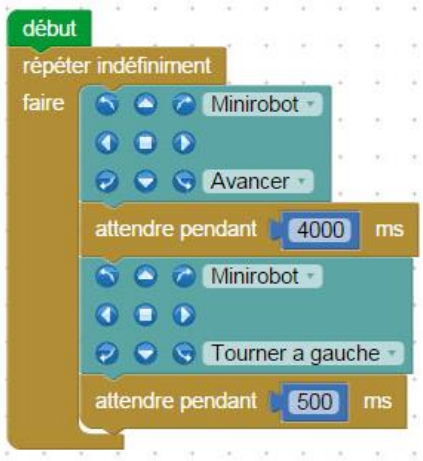
Certains robots tels que le Microbot120, utilisent deux moteurs (C et D) pour contrôler leur mouvement.



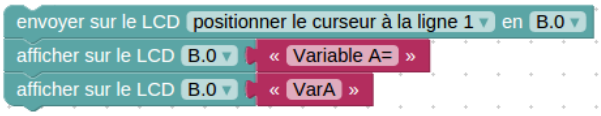
Il existe pour ces robots, un bloc moteur dédié.



Par exemple, pour que le robot aille vers l'avant, sélectionnez la case centrale dans la rangée du haut. Cela commute à la fois les moteurs C et D dans la direction vers l'avant.



Blocs envoyer/afficher sur le LCD

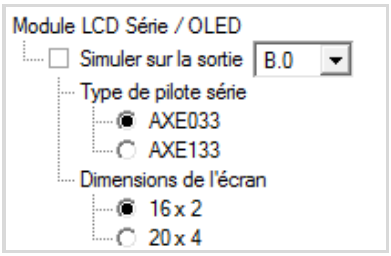


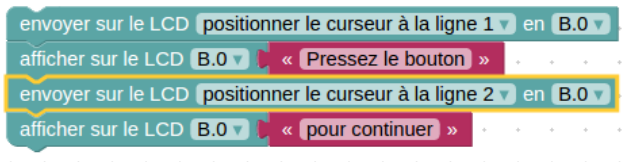
Ce bloc peut être utilisé pour afficher un message sur un écran LCD PICAXE.

www.picaxe.com/products/axe133y

La simulation est active uniquement sur PE6 (pas par l'appli).

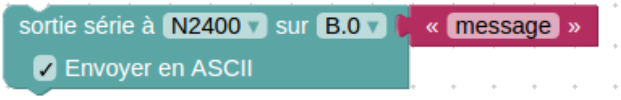
Dans PE6, une petite fenêtre de l'écran LCD apparaît pendant l'exécution pour afficher le message de l'écran LCD.
Assurez-vous que la broche de simulation pour l'écran LCD est correctement paramétrée à partir du menu **Fichier / Options / Simulation**.





Ecran LCD simulé

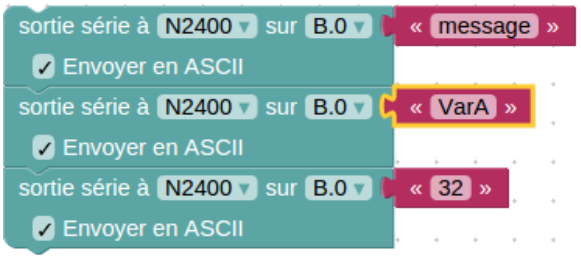
Bloc sortie série (Serout)



Ce bloc permet d'envoyer une information du microcontrôleur PICAXE vers un écran série LCD, ou un autre PICAXE qui est connecté à une sortie du microcontrôleur.

La première donnée définit le baud rate de la liaison série (ici N2400).

La seconde est utilisée pour sélectionner la broche de sortie du microcontrôleur PICAXE qui enverra la donnée.



La donnée à envoyer est liée à l'entrée sur la droite. Ça peut être un texte, une variable ou une constante.

Si utilisation d'un texte, assurez-vous d'avoir coché « envoyer en ASCII ».

Si utilisation d'une variable ou constante, vous pouvez soit envoyer la valeur de la ligne binaire ex : 32 (ASCII non cochée) ou le caractère ASCII équivalent ex : « 3 » puis « 2 » (case ASCII cochée).

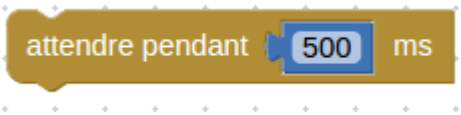
Bloc Sertxd

Le bloc **sertxd** est similaire au bloc **serout**, mais agit par l'intermédiaire de la broche de sortie série plutôt qu'une broche de sortie générale. Ceci permet à la donnée d'être renvoyée à l'ordinateur par le câble de programmation. C'est très utile pour le débogage.

Pour plus d'informations, se reporter au manuel PICAXE 2 « Commandes du BASIC ».
www.picaxe.com/docs/picaxe_manual2_fr.pdf

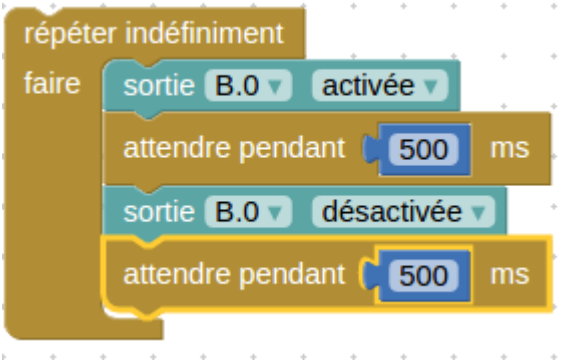
Section 3. Délais

Bloc attendre

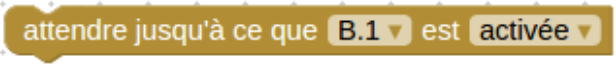


Un bloc **attendre** permet de définir un délai d'attente (en millisecondes) avant l'exécution de l'instruction suivante.

Vous pouvez l'utiliser pour garder les périphériques de sortie activés ou désactivés pour un temps donné.

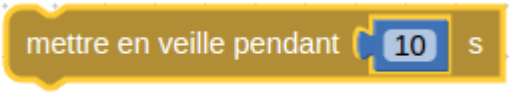


Bloc attendre jusqu'à ce que



Ce bloc **attend jusqu'à ce qu'une condition soit vraie sur** une broche avant de passer à l'instruction suivante.

Bloc mettre en veille



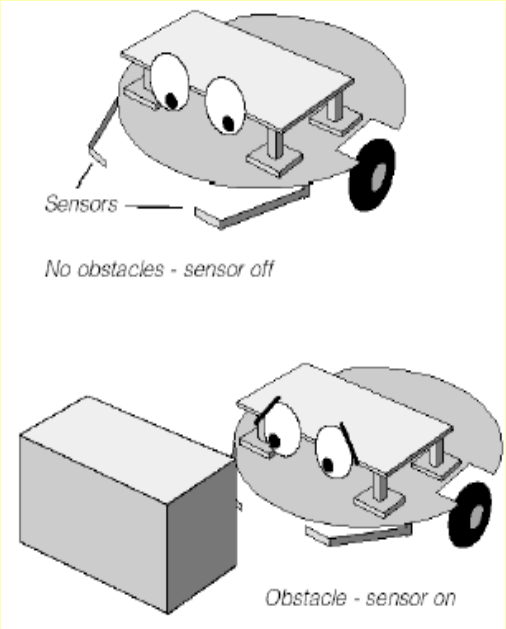
Ce bloc met le microcontrôleur PICAXE dans un mode basse consommation pour un nombre de secondes spécifié. Ce mode peut être utilisé pour économiser de la batterie.

Tous les périphériques de sortie seront laissés dans leur état actuel, mais les signaux des périphériques d'entrée ne seront plus pris en compte tant que le microcontrôleur est en mode veille.

Section 4. Entrées

Les équipements d'entrée tels qu'interrupteurs ou capteurs envoient des informations depuis le monde extérieur au système de commande. Les équipements de sortie sont activés en fonction des informations fournies par les équipements d'entrée.

Exemple : Un buggy équipé de microrupteurs.



Le contact d'un microrupteur peut être utilisé comme condition pour arrêter les moteurs et commencer une séquence de mouvements pour se déplacer autour de l'obstacle.

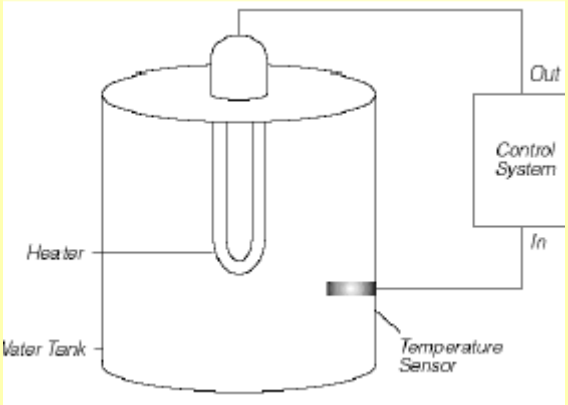
Un microcontact est un capteur numérique. Il n'a que deux états - **ON ("fermé") et OFF (ouvert).**

L'état d'un capteur numérique est souvent représenté par 0 (désactivé) ou 1 (activé).

Exemple :

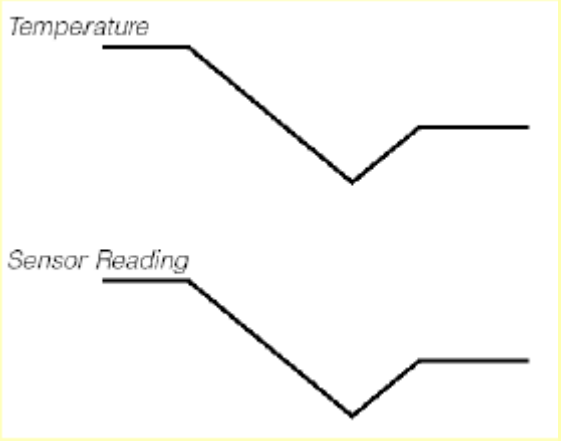
Un système d'eau chaude commandé comprend un capteur de température qui surveille en permanence la température de l'eau.

Le chauffe-eau est sous tension et hors tension en réponse aux informations fournies par le capteur. Si la température de l'eau descend en dessous d'un niveau défini, le chauffage est allumé jusqu'à ce qu'il atteigne à nouveau ce niveau. Ensuite, l'appareil est éteint.

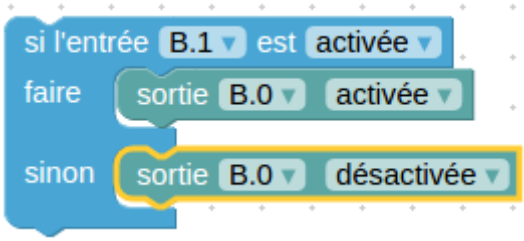


Un capteur de température est un capteur analogique.

Il fournit une lecture qui change en fonction du niveau de température



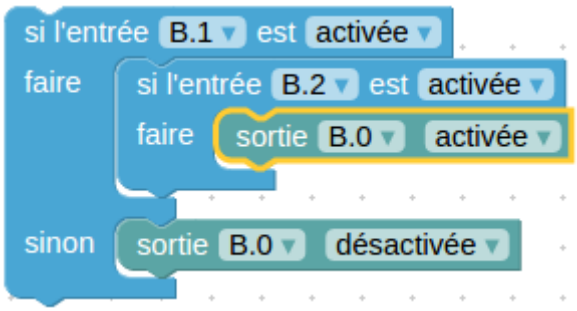
Bloc entrée



Utiliser ce bloc pour tester l'état d'un capteur digital connecté à une entrée digitale du microcontrôleur PICAXE.

Lorsque le programme atteint le bloc d'entrée, il ne traitera que l'une des deux sections, selon que la broche d'entrée est activée (haute) ou désactivée (basse).

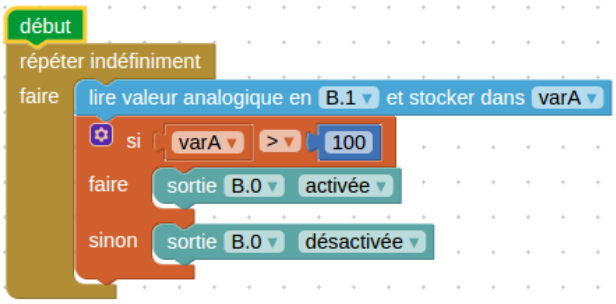
Plusieurs blocs d'entrée peuvent également être imbriqués pour tester, par exemple, si deux broches sont toutes les deux activées.



Bloc lire valeur analogique / température / distance ultrason

Ces trois blocs opèrent de la même façon : enregistrement d'une valeur analogique dans une variable.

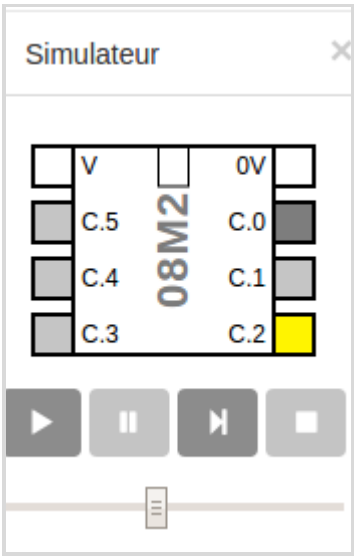
Un bloc de condition sur une variable est utilisé pour effectuer une action en fonction de la valeur renvoyée par le capteur.



- Analogique – tous les capteurs génériques comme une LDR.
- Température – DS18B20 capteur Température
- Ultrason SFR005 capteur de distance.

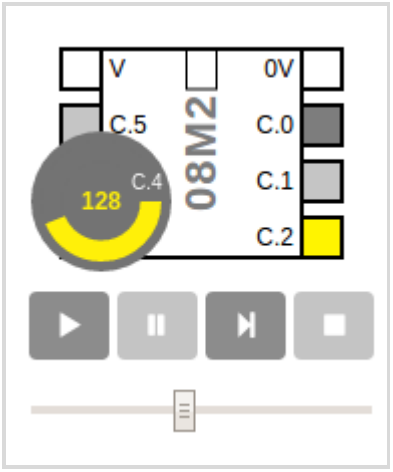
Simulation d'une entrée digitale

Pour changer l'état d'une entrée, cliquer simplement sur l'entrée pendant la simulation. Elle changera d'état de gris (off) à jaune (on).



Simulation d'une Entrée Analogique

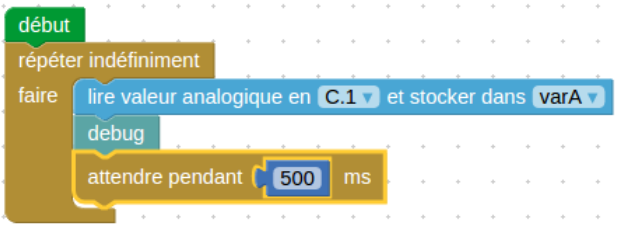
Pour simuler un changement de valeur pour des blocs analogiques, clic droit sur la broche correspondante dans le panneau de simulation et utiliser le curseur circulaire.



Calibrer un Capteur par l'utilisation de Debug

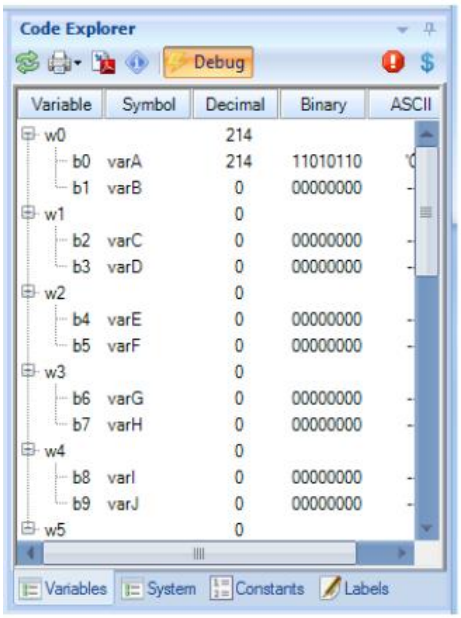
Souvent, lors de l'utilisation de capteurs analogiques, il est utile de tester la valeur renvoyée par le capteur afin de trouver le point de seuil correct.

Pour lire des valeurs analogiques en temps réel à partir d'un microcontrôleur PICAXE, nous pouvons aussi utiliser le bloc de débogage dans une boucle comme dans le programme ci-dessous.

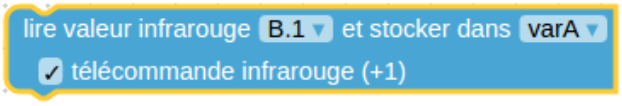


- Dans l'application PICAXE Blockly, à partir du menu PICAXE, cliquer sur **Debug**.
- Dans PE6, cliquer sur **Débogage** dans le panneau Explorateur de code.

La valeur de la Var A sera alors affichée sur l'écran de l'ordinateur et sera remise à jour toutes les 500 ms.



Bloc lire valeur infrarouge



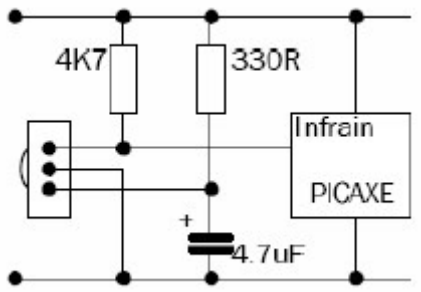
Le bloc **lire valeur infrarouge** est utilisé pour recevoir une information depuis un émetteur infrarouge.

Le bloc va attendre un nouveau signal infrarouge provenant d'un émetteur de type Télécommande infrarouge. Il peut également être utilisé pour recevoir un bloc « envoyer infrarouge » envoyé à partir d'un microcontrôleur PICAXE séparé.

Tous les processus s'arrêtent jusqu'à ce que le nouveau bloc soit reçu. La valeur du code reçu est placée dans la variable choisie.

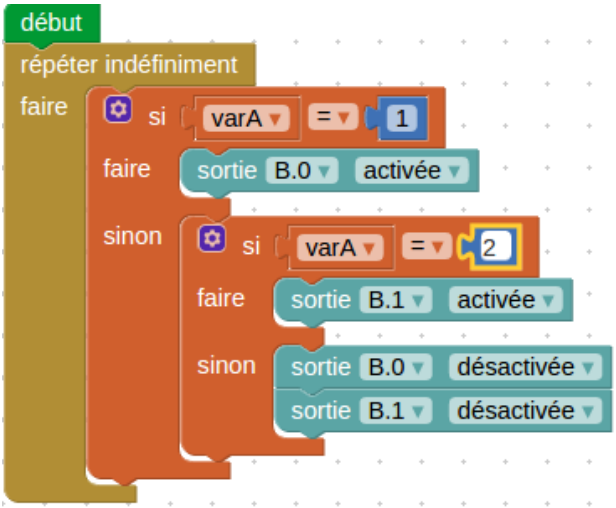
Comme les télécommandes de style TV (par exemple une VR 010) envoient une valeur de décalage (par exemple la valeur 0 pour le canal 1, la valeur 1 pour le canal 2) la case à cocher vous permet d'ajouter automatiquement 1 à la valeur reçue, de sorte que la valeur de la variable

Le circuit de base requis pour ce bloc est le suivant. Le dispositif sur le côté gauche du circuit est une LED réceptrice IR, réf de la pièce : LED020.



Exemple :

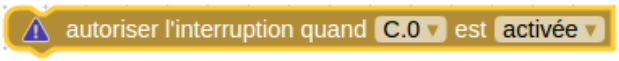
Dans le programme suivant un signal est reçu d'une télécommande infrarouge de téléviseur. Les lumières sont allumées si la touche 1 ou 2 est pressée.



Le bloc **lire valeur infrarouge** attend qu'un signal soit reçu, et sauvegarde celui-ci comme un nombre dans la variable A.

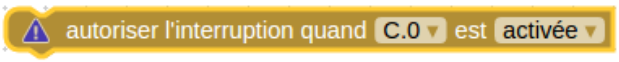
Si la valeur est 1 ou 2 les sorties sont basculées à On, Autrement les sorties sont Off.

Bloc autoriser l'interruption quand



Lorsque l'interruption est déclenchée, le programme saute immédiatement vers un bloc de sous procédure (qui doit être appelé « interruption ») puis parcourt tous les blocs qui suivent jusqu'à ce qu'il atteigne la fin de cette sous-procédure. Il revient alors au point dans le programme principal où il était au moment de l'interruption.

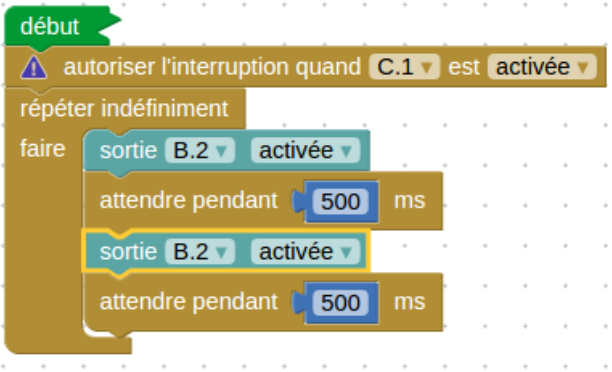
L'instruction **Interruption** configurée afin de déterminer la condition et la sortie correspondant à l'interruption.

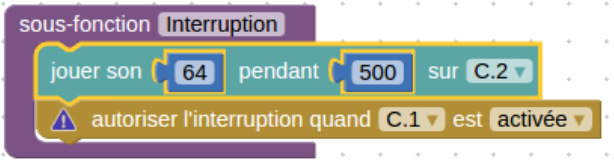


Pour éviter qu'une interruption se boucle sur elle-même, l'interruption est automatiquement désactivée une fois déclenchée. Pour réactiver une autre, l'activation d'un nouveau bloc d'interruption est requise.

Exemple :

Un microcontrôleur PICAXE qui exécute en continu une boucle de lumières clignotantes doit être en mesure de réagir à une pression sur le bouton et jouer un son d'avertissement.



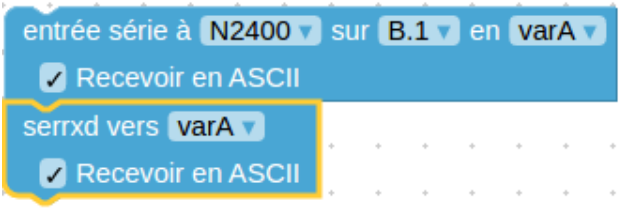


L'Interruption est utilisée pour capturer les infos et jouer un son. L'interruption est ensuite autorisée une fois encore avant de revenir à l'endroit où elle a quitté le programme principal.

Il n'y a pas de limite aux blocs dans l'Interruption. Il y a une technique courante pour ajouter un bloc « Ajouter Interruption » avant la fin de la procédure, de sorte que, lorsque la sous-procédure d'interruption retourne au programme l'interruption est réactivée.

Il ne peut être utilisé qu'une seule sous procédure Interruption par programme

Bloc entrée série



Le bloc **entrée série** est utilisé pour recevoir des données séries sur une broche entrée du microcontrôleur.

La broche d'entrée est l'entrée du PICAXE qui va recevoir la donnée.

L'option Variable est un emplacement variable où la donnée est stockée une fois que celle-ci est reçue.

La première option spécifie le débit et la polarité du signal.

Lors de l'utilisation d'une simple interface résistive, utiliser signaux N (inversé).

Lors de l'utilisation d'une interface de type MAX232 utiliser signaux T (vrai).

Le protocole est fixé à N, 8, 1 (pas de parité, 8 bits de donnée, 1 bit stop)

Le bloc **entrée série** force le microcontrôleur PICAXE à attendre que la donnée série soit reçue par l'entrée choisie. Cette donnée est stockée dans la variable choisie.

Pour stocker une ligne de données binaires (en opposition à une chaîne ASCII), assurez-vous que la case ASCII ne soit pas cochée

Exemple :

La donnée série est reçue d'un autre microcontrôleur PICAXE et doit être stockée dans l'EEPROM.

Dans le programme ci-dessous, la donnée série est lue dans la variable B par la broche d'entrée C.2.

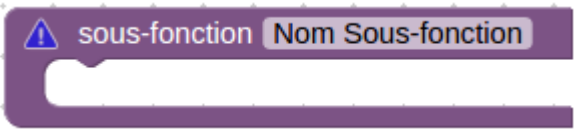
Le bloc **écrire** est utilisé pour stocker la valeur dans la variable B dans l'EEPROM. Ce processus est répété 16 fois pour remplir les emplacements de mémoire EEPROM.

*Utilisation du bloc **entrée série** pour recevoir une donnée série.*

Section 5.Procédures

Blockly fournit une méthode claire et pas-à-pas pour construire un système de commande complexe, par la création d'un nombre de sous-systèmes liés appelés « sous-procédures ».

Comment construire une sous-procédure

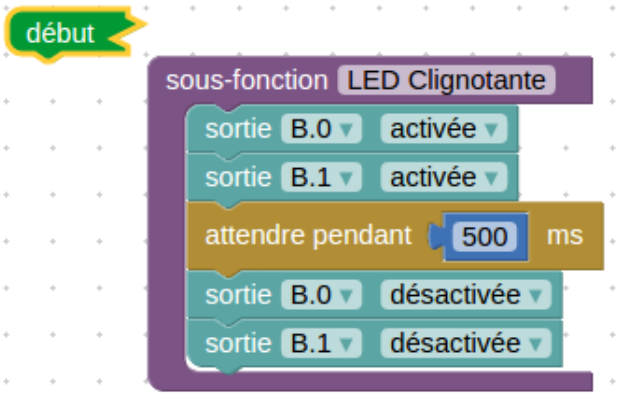


Glisser un bloc **sous fonction** pour commencer une procédure.

Glisser le bloc sur le programme et le placer séparément du bloc **début** comme montré ci-dessous.

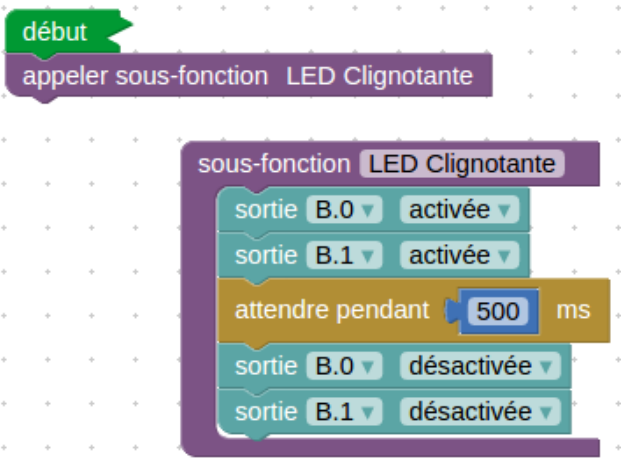
Choisir un nom en rapport avec la tâche à effectuer.

Utiliser d'autres blocs habituels pour créer la procédure.



Comment utiliser une sous-procédure

Une fois que vous avez construit une procédure, vous pouvez l'appeler chaque fois que vous le voulez dans le programme en cours en utilisant le bloc d'appel, comme montré ci-dessous.



*Le bloc **appeler** appelle la procédure pour l'utiliser.*

Glisser le bloc **appeler sous-fonction** dans le programme. Placez-le à l'endroit où vous voulez appeler la procédure pour l'utiliser.

Noter que toutes les procédures qui ont été construites dans un programme sont automatiquement listées dans la boite à outils. Quand le programme atteint un bloc d'appel d'une sous-fonction, il saute à la procédure ayant le même nom.

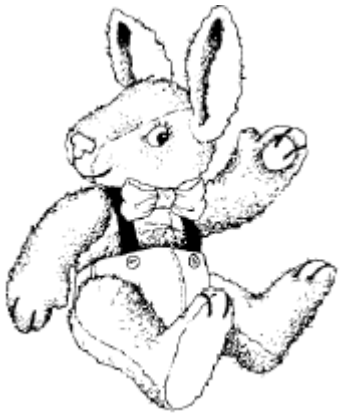
Quand le déroulement du programme atteint la fin de la procédure, il revient là où le bloc a appelé la procédure.

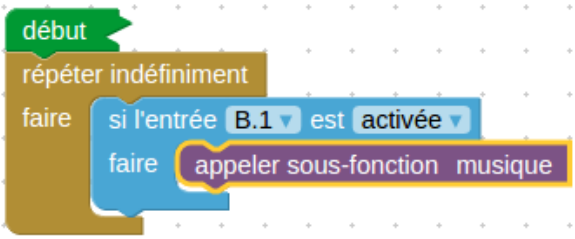
Exemple :

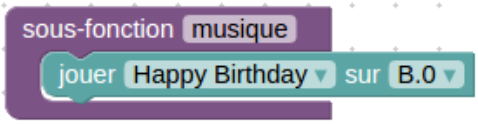
Un microcontrôleur PICAXE est utilisé pour commander un système dans un jouet d'enfant qui joue une musique quand il est étreint.

Un Transducteur Piézo est connecté à une broche de sortie, et un bouton-poussoir est utilisé comme capteur quand le jouet est étreint.

Le programme pour ce système est montré ci-dessous. Le son est créé par une procédure qui peut être testée et modifiée séparément du programme principal.



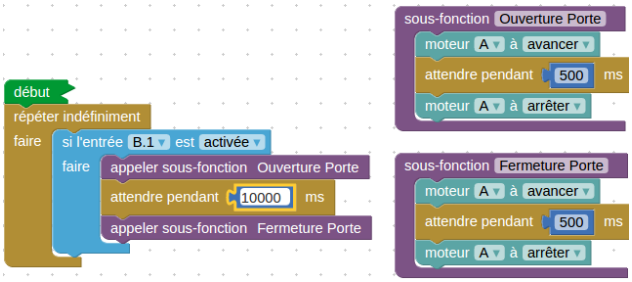




Exemple :

Le programme montré ci-dessous est un système de commande pour une porte coulissante.

Quand un contact est activé, la porte s'ouvre. Elle reste ouverte dix secondes et ensuite se referme.



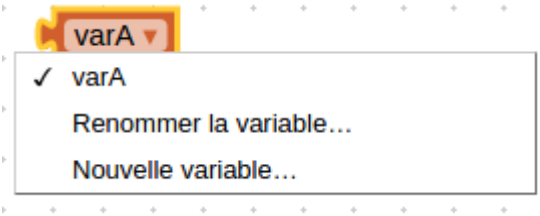
Système de commande de porte coulissante utilisant les procédures.

Section 6. Maths et Variables

Dans Blockly, une variable est un « numéro de case » qui peut contenir une valeur donnée entre 0 et 65535.

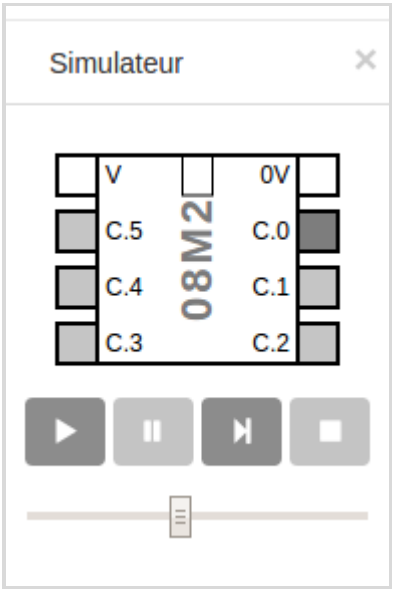
Les variables sont appelées var A à var Z par défaut, mais peuvent être renommées avec n'importe quel nom que vous choisissez.

Pour renommer une variable, cliquer simplement sur la flèche à droite du nom de la variable et sélectionner « Renommer la variable... ».



Cette section explique comment elles peuvent être utilisées dans une variété de comptage et à des fins de synchronisation principalement.

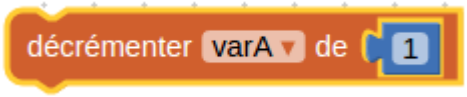
La valeur courante de la variable peut être vue pendant une simulation dans le panneau de simulation.



Bloc incrémenter Variable

Chaque fois que le programme passe à travers un bloc **incrémenter**, la valeur désirée est ajoutée à la valeur de la variable sélectionnée.

Bloc décrémenter Variable



Le bloc **décrémenter** travaille de façon similaire au bloc Incrémenter.

La différence est qu'au lieu d'ajouter la valeur à la valeur de la variable, il le soustrait à la valeur de la variable.

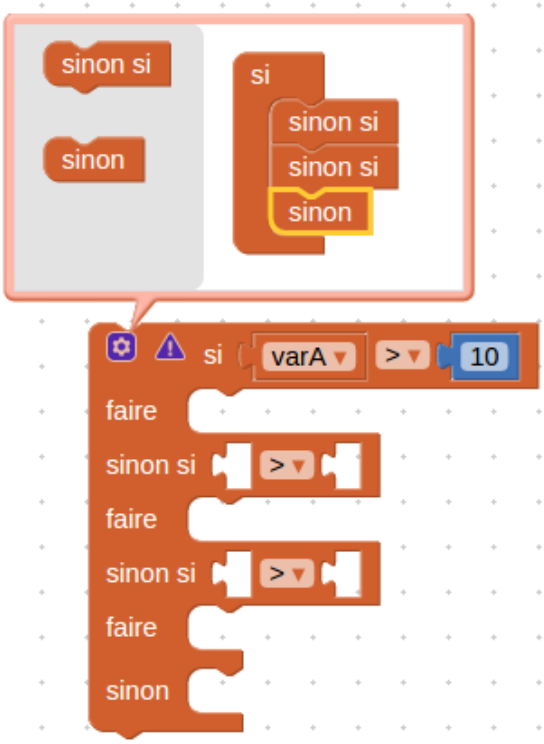
Note : les microcontrôleurs PICAXE ne supportent pas les nombres négatifs. Tous les nombres en dessous de 0 seront en dépassement, ex -1 devient 65535.

Bloc Variable Si

Pour tester la variable, le bloc **variable si** est utilisé. La valeur peut être testée pour voir si elle est plus grande, moins grande ou égale à une autre variable ou une valeur fixe.

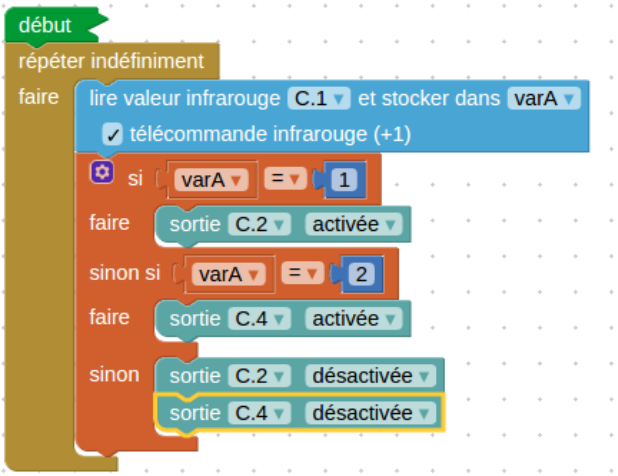
Ce bloc est unique en ce que sa forme peut être modifiée pour ajouter plus de tests.

Cliquer sur le symbole **Paramètres** pour compléter le travail au sein de la petite fenêtre, configurer le bloc en glissant le nombre de commandes requises.



Exemple 1 :

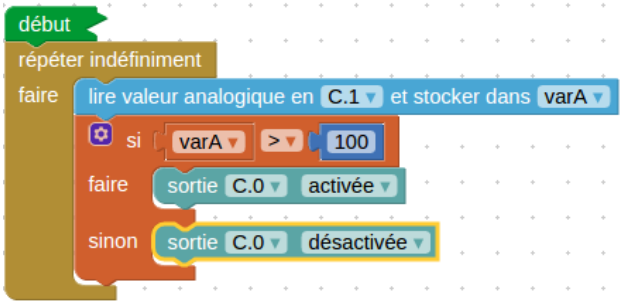
En utilisant ce système, vous pouvez configurer les blocs pour un nombre de tests consécutifs, comme montré dans cet exemple infrarouge.



Exemple 2 :

Un microcontrôleur PICAXE est utilisé pour contrôler une lampe. Un capteur lumière est connecté à l'entrée analogique 0.

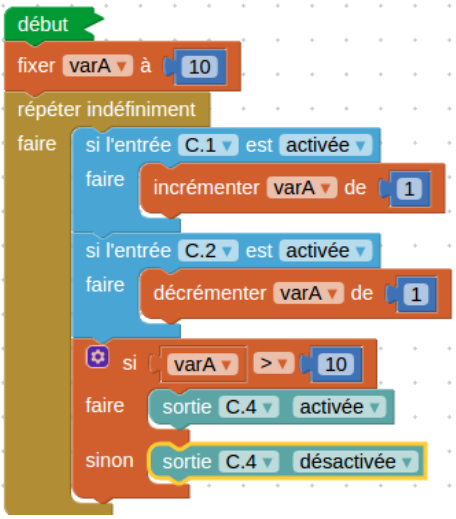
Le système allumera la lampe automatiquement dans les périodes sombres.



Exemple 3 :

Un microcontrôleur PICAXE est utilisé pour commander un système comptant les véhicules entrants et sortants d'un parking en utilisant deux capteurs digitaux.

Quand 10 véhicules sont dans le parking, la lampe « Complet » est allumée.



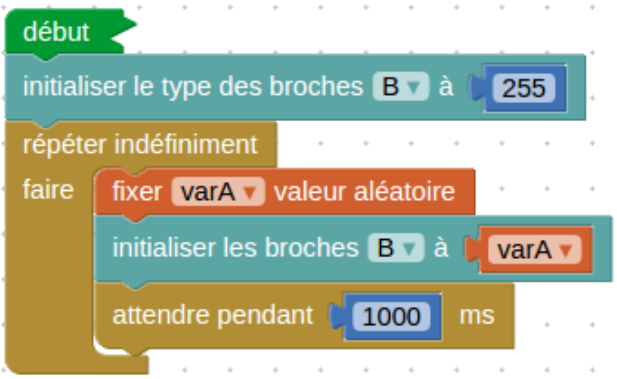
Bloc fixer Variable

Une valeur peut également être donnée à une variable sous forme d'expression mathématiques comme montré dans l'exemple ci-dessus (varA = varB + varC).

Note : les expressions multiples peuvent aussi être imbriquées à l'intérieur d'autres pour faire une équation plus longue (varA= vabB + varC + 10).

Bloc fixer valeur aléatoire

En utilisant le bloc **fixer valeur aléatoire**, une variable peut recevoir une valeur aléatoire entre 0 et q. Dans l'exemple présenté ci-dessous, un jeu de lampes pour un arbre de Noël est connecté à 8 entrées d'un microcontrôleur PICAXE. Chaque seconde l'affichage changera aléatoirement.



Note : comme avec tous les microcontrôleurs et ordinateurs, la génération des nombres aléatoires est basée sur une séquence.

Section 7. Blocs Avancés

Blocs Lire et Écrire

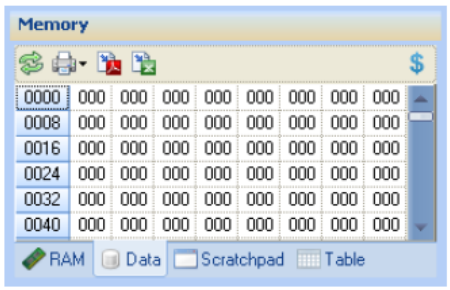
Quand l'exécution d'un programme est lancée, toutes les variables sont réinitialisées à 0. Ainsi, quand le microcontrôleur PICAXE est réinitialisé ou réalimenté, toutes les valeurs de variables sont réinitialisées à 0.

Si vous voulez conserver les valeurs de variables quand le microcontrôleur PICAXE est soit réinitialisé ou réalimenté, vous pouvez utiliser le bloc écrire pour stocker des valeurs dans la mémoire de données EEPROM. Le bloc lecture est utilisé pour retrouver les valeurs depuis la mémoire du microcontrôleur.

Le bloc **lire** prend la valeur qui est actuellement stockée dans l'adresse sélectionnée (dans ce cas adresse 0) et met cette valeur dans la variable sélectionnée (dans ce cas la variable A).

La mémoire EEPROM de données du microcontrôleur PICAXE a 255 adresses séparées. Chaque adresse peut stocker un nombre entre 0 et 255.

La fenêtre EEPROM (PE6 seulement) affiche le contenu de la mémoire quand vous exécuté un test du programme.



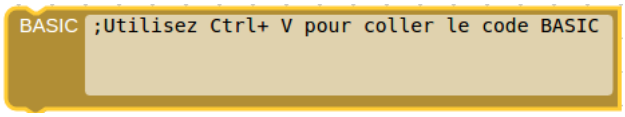
Fenêtre données EEPROM

Bloc Écrire I2C et Lire I2C

Les commandes I2C permettent des communications avec des équipements tiers tels que mémoires EEPROM ou des capteurs.

Pour plus de détails, se reporter au manuel 2 PICAXE.
www.picaxe.com/docs/picaxe_manual2_fr.pdf

BASIC



Le bloc **BASIC** est utilisé comme une extension d'un programme. Tout code PICAXE BASIC valide peut être tapé dans la fenêtre de la cellule du bloc. Quand le déroulement du programme arrive à ce bloc, le code BASIC dans celui-ci est exécuté comme si c'était une procédure.

Note : seul PE6 peut simuler le code BASIC. Les versions app/cloud de Blockly sauteront ce bloc.
En effet, PE6 dispose d'un simulateur plus puissant que les versions web.

Pour plus d'informations sur les autres commandes avancées (ex : suspend, reconnect, etc.), se reporter au manuel PICAXE 2 Commandes BASIC.
www.picaxe.com/docs/picaxe_manual2_fr.pdf

Section 8. Licences

Blockly pour PICAXE utilise les projets open source suivants et remercie les développeurs de ces projets.

Blockly https://github.com/google/blockly/blob/master/COPYING

Bootstrap https://github.com/twbs/bootstrap/blob/master/LICENSE

FileSaver https://github.com/eligrey/FileSaver.js/blob/master/LICENSE.md

Blob https://github.com/eligrey/Blob.js/blob/master/LICENSE.md

CodeMirror https://github.com/codemirror/CodeMirror/blob/master/LICENSE

JS-Interpreter https://github.com/NeilFraser/JS-Interpreter/blob/master/LICENSE

Acorn https://github.com/ternjs/acorn/blob/master/LICENSE

jQuery https://github.com/jquery/jquery/blob/master/LICENSE.txt

Mustache.js https://github.com/janl/mustache.js/blob/master/LICENSE

RequireJS https://github.com/jrburke/requirejs/blob/master/LICENSE

Signals https://github.com/millermedeiros/js-signals