

# BLOCKLY pour PICAXE

## Le boîtier interface Autoprogrammable



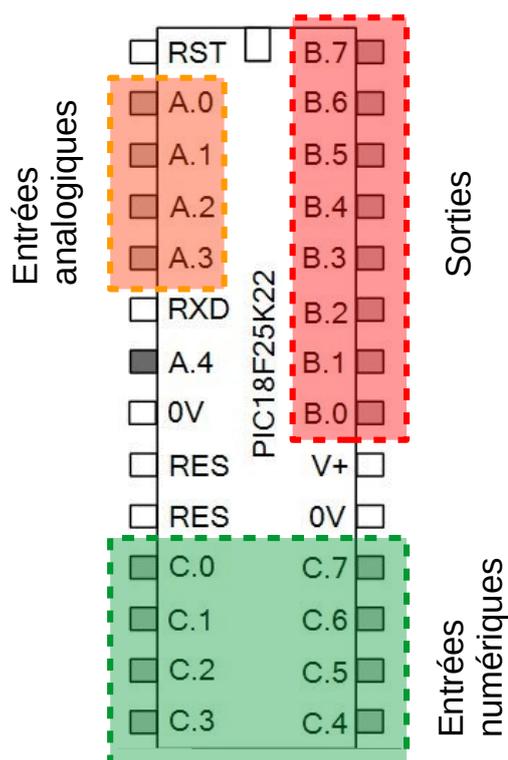
**AutoProg®** est un système modulaire d'automatisme, constitué d'un **automate programmable** autour duquel se connectent une large gamme de modules **capteurs** et **actionneurs**.

Le boîtier de commande **AutoProg®** est équipé d'un **microcontrôleur PICAXE®** de type 28X2

Le boîtier **AutoProg®** comporte des connecteurs pour :

- **8 sorties** repérées **S0 à S7**, sur lesquelles peuvent se connecter les modules **actionneurs** (moteur, DEL, gyrophare, etc) ;
- **8 entrées numériques** repérées **EN0 à EN7**, sur lesquelles peuvent se connecter des **capteurs** de type « tout-ou-rien » (bouton poussoir, fin de course, phototransistor, etc) ;
- **4 entrées analogiques** repérées **A0 à A3**, sur lesquelles peuvent se connecter des capteurs de type thermorésistance, LDR, etc.

## Le microcontrôleur Picaxe



## Correspondance des entrées / sorties

	Autoprogrammable	Picaxe
Entrées	En0	C0
	En1	C1
	En2	C2
	En3	C3
	En4	C4
	En5	C5
	En6	C6
	En7	C7
Sorties	S0	B0
	S1	B1
	S2	B2
	S3	B3
	S4	B4
	S5	B5
	S6	B6
	S7	B7

# BLOCKLY pour PICAXE

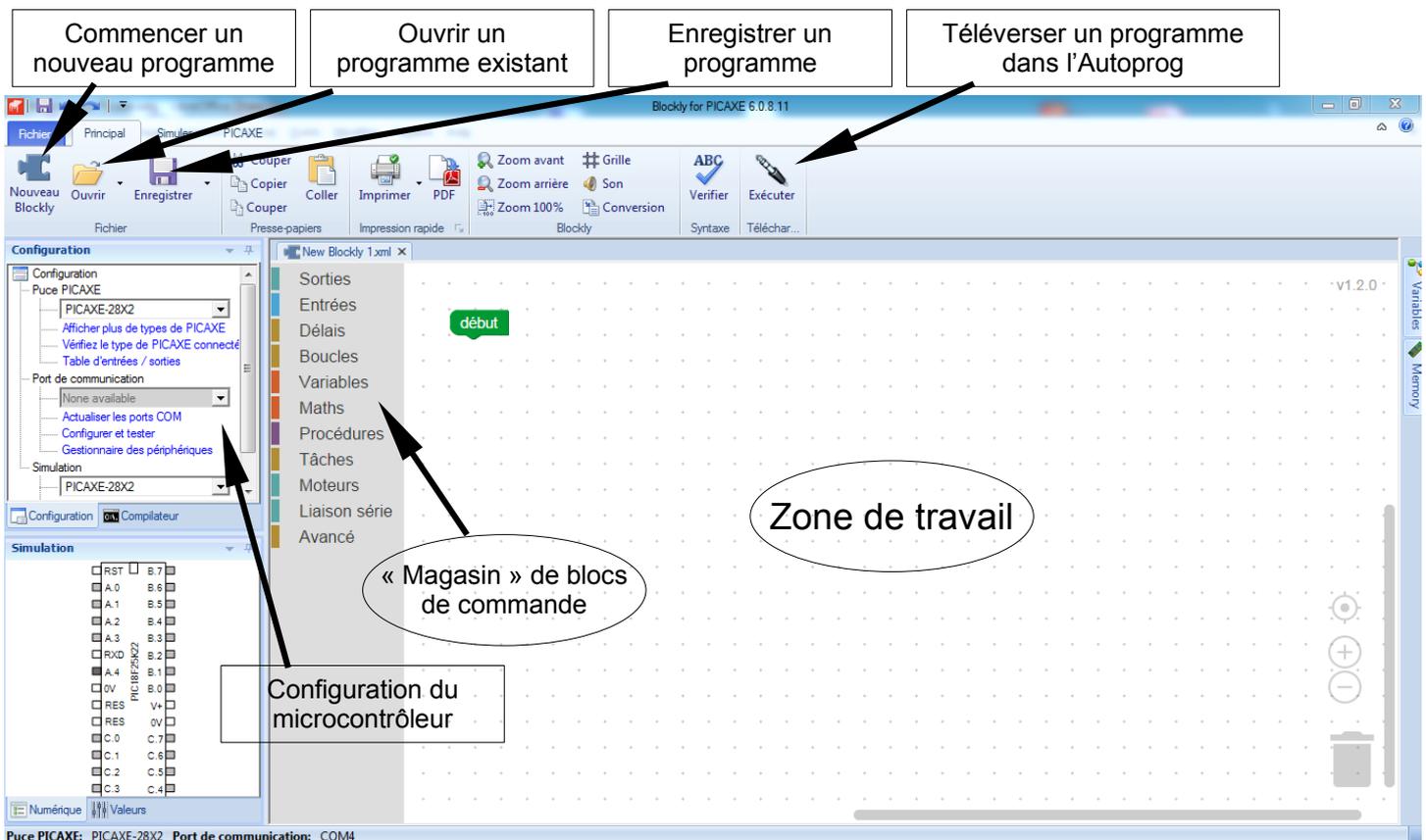
## Le logiciel Blockly pour Picaxe



**Blockly** est un outil puissant de programmation graphique permettant de générer des programmes pour les microcontrôleurs Picaxe en empilant des **blocs de couleur**.

Blockly utilise une méthode simple de programmation par **glisser/déposer** qui permet de développer rapidement des séquences de commandes, à la manière de **Scratch**.

## L'interface graphique de Blockly

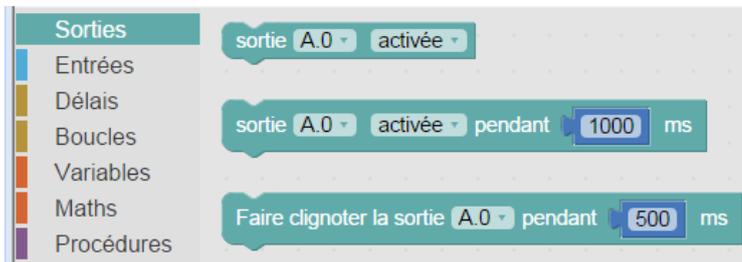


## Les catégories de blocs de commande

Sorties	→	Agir sur les sorties
Entrées	→	Tester les entrées
Délais	→	Programmer des temporisations
Boucles	→	Programmer des boucles de programmation
Variables	→	Utiliser des variables
Maths	→	Utiliser des formules mathématiques ou logiques
Procédures	→	Paramétrer des sous-programmes
Tâches	→	
Moteurs	→	Commander les moteurs ou servo-moteurs
Liaison série	→	
Avancé	→	

# BLOCKLY pour PICAXE

## Les sorties (B0, B1, ..., B7)



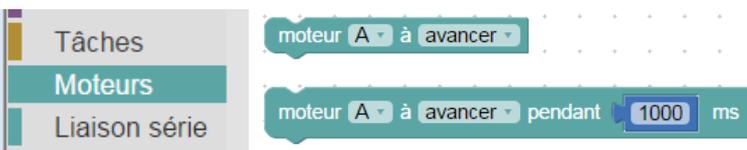
Ces blocs sont utilisés pour **activer** ou **désactiver** une sortie.

« **Activer** » une sortie, c'est lui fournir de l'énergie électrique.

« **Désactiver** » une sortie, c'est lui supprimer cette énergie.

Ex : Si on branche une DEL sur une sortie, si cette sortie est activée, la DEL va s'allumer. Si la sortie est désactivée, la DEL s'éteindra.

## Les commandes de moteurs



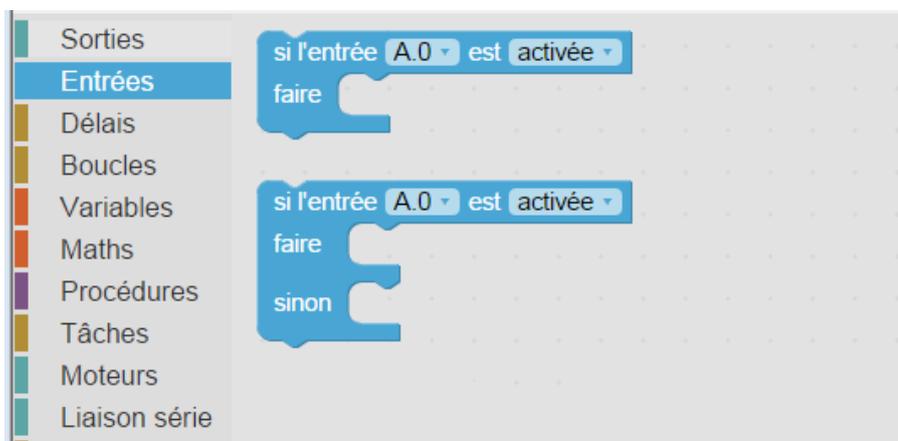
Ces blocs permettent de piloter des moteurs.

Le moteur est nommé selon ses connexions au boîtier Autoprogram :

A → B0 & B1      B → B2 & B3  
C → B4 & B5      D → B6 & B7

Le moteur peut être programmé selon 3 états :  
Avancer / Reculer / Arrêter

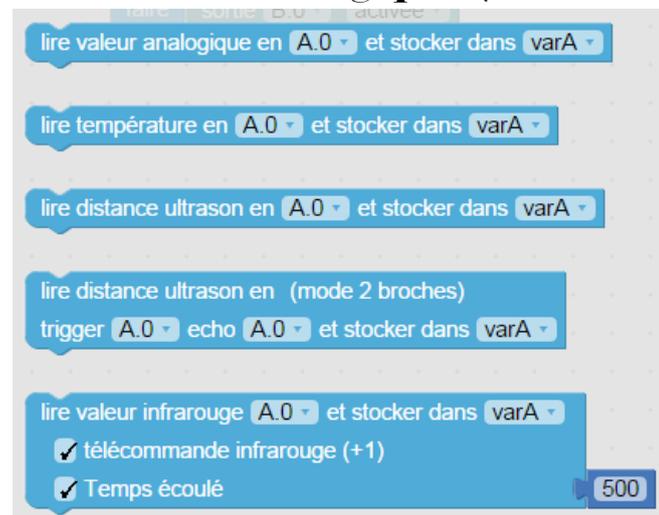
## Les entrées numériques (C0, C1, ..., C7)



Les **entrées numériques** (« tout-ou-rien » Ex : bouton-poussoir, contact fin de course, etc) sont soit « **Activées** » (= contact fermé), soit « **Désactivées** » (= contact ouvert).

Les **tests** sur ces entrées sont de type « **Si [Test] Alors [Action]** » ou « **Si [Test] Alors [Action1] Sinon [Action2]** »

## Les entrées analogiques (A0, A1, ..., A4)



Les **entrées analogiques** (dont l'état varie en fonction d'une grandeur physique Ex : capteur de température, détecteur infra-rouge, capteur d'ultrasons, etc) doivent être traitées au moyen de **variables** dans lesquelles on stocke leur **valeur instantanée**.

Les **tests** se font alors sur ces **variables**.

## Conseils d'utilisation

Ce ne sont que quelques conseils, non des obligations, qui peuvent simplifier la programmation sous Blockly.

1. Bien **étudier** le système à programmer (= la **partie opérative**).  
Repérer les **entrées** (capteurs) et les **sorties** (actionneurs).
2. Réaliser l'**algorithme** et/ou l'**algorithme** du système à programmer.
3. Établir un **tableau des entrées / sorties** et y noter la correspondance avec les entrées & sorties sous Blockly (En0 ↔ C0 ; En1 ↔ C1 ; etc ; S0 ↔ B0 ; S1 ↔ B1 ; etc) (*Voir Page 1*)
4. Affecter une « **étiquette** » à chaque entrée / sortie et compléter le tableau.
5. Recopier ce tableau dans la « **table d'entrées / sorties** » de Blockly. (*Voir Page 10*)
6. Réaliser le **programme** sous Blockly en essayant de le rendre le plus **clair** et le plus **simple** possible. Penser à utiliser éventuellement des **sous-programmes** (*voir page 11*) et à **renommer** les variables s'il y en a (*voir page 8*)
7. Pensez à **enregistrer régulièrement** votre programme pour pouvoir le récupérer en cas de problème.  
Ne pas essayer d'**ouvrir un programme** déjà enregistré à partir de l'explorateur de Windows (double-clic sur son nom dans « Mes documents » par exemple), cela ne fonctionnera pas.  
**Charger les programmes** à partir de la commande « Ouvrir » de Blockly.
8. Quand la rédaction du programme est **terminée**, brancher l'interface Autoprogram au PC, puis cliquer sur la commande « Exécuter » pour y **téléverser** le programme.

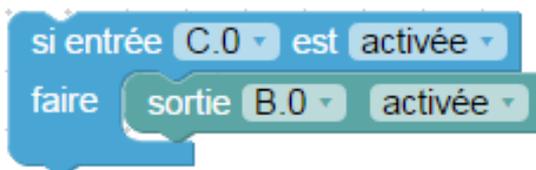
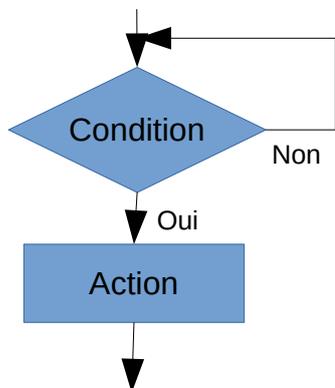
	Autoprogram	Picaxe	Étiquette
Entrées	En0	C0	Bp Appel bas
	En1	C1	Bp Appel haut
	En2	C2	Fin course bas
	En3	C3	Fin course haut
	En4	C4	
	En5	C5	
	En6	C6	
	En7	C7	
Sorties	S0	B0	Voyant bas
	S1	B1	Voyant haut
	S2	B2	Moteur sens 1
	S3	B3	Moteur sens 2
	S4	B4	
	S5	B5	
	S6	B6	
	S7	B7	

# BLOCKLY pour PICAXE

## Correspondance algorithme ↔ Blockly

**Boucle « Si ... alors ... »** Si l'entrée C0 est activée, on active la sortie B0.

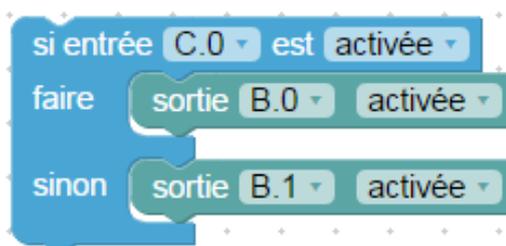
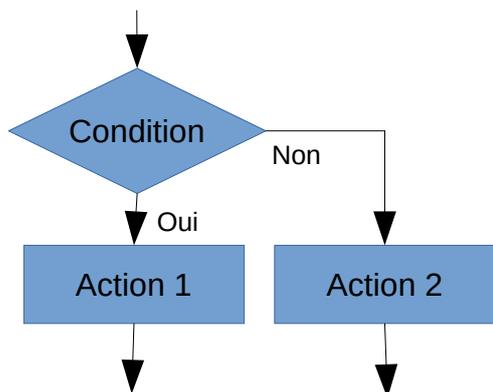
*Ex : Si un bouton-poussoir est connecté à l'entrée 0 du boîtier Autoprogram, qu'une DEL est connectée à sa sortie 0, un appui sur le bouton-poussoir allumera la DEL .*



**Boucle « Si ... alors ... sinon ... »**

Si l'entrée C0 est activée, on active la sortie B0, sinon on active la sortie B1.

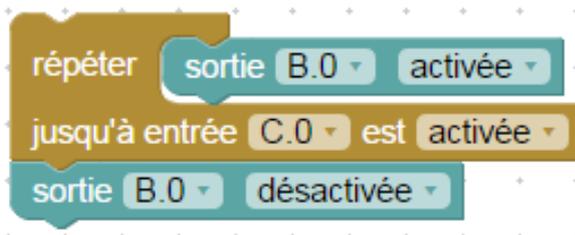
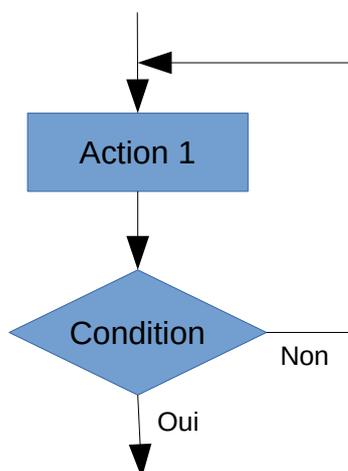
*Ex : Si un bouton-poussoir est connecté à l'entrée 0 du boîtier Autoprogram, qu'une DEL verte est connectée à sa sortie 0 et une DEL rouge à la sortie 1, un appui sur le bouton-poussoir allumera la DEL verte, aucun appui sur le bouton-poussoir allumera la DEL rouge.*



**Boucle « Répéter jusqu'à ... »**

La sortie B0 est activée jusqu'à ce que l'entrée C0 soit activée

*Ex : Si un bouton-poussoir est connecté à l'entrée 0 du boîtier Autoprogram et qu'une DEL est connectée à sa sortie 0, la DEL sera allumée jusqu'à ce qu'on appuie sur le bouton-poussoir .*

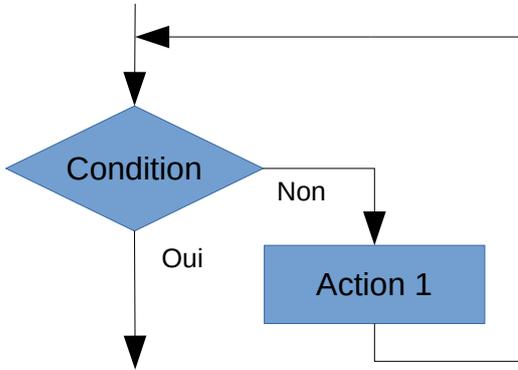


*Remarque : Avec un microcontrôleur Picaxe, une sortie activée reste activée jusqu'à ce qu'on la désactive, donc ne pas oublier l'instruction « sortie ... désactivée ».*

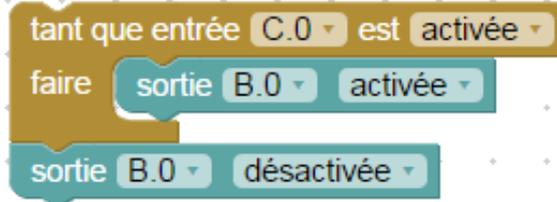
# BLOCKLY pour PICAXE

## Correspondance algorithme ↔ Blockly

**Boucle « Tant que ... faire »** Tant que l'entrée C0 est activée, la sortie B0 est activée

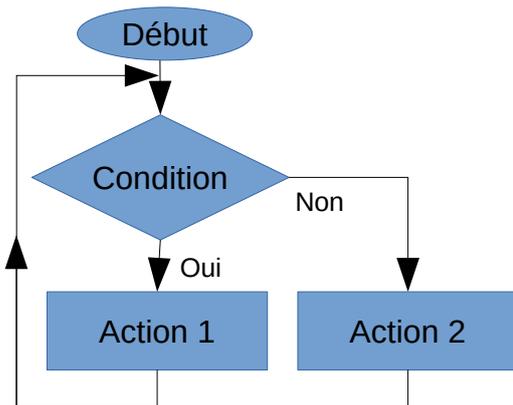


Ex : Si un bouton-poussoir est connecté à l'entrée 0 du boîtier Autoprogrammable et qu'une DEL est connectée à sa sortie 0, la DEL sera allumée jusqu'à ce qu'on appuie sur le bouton-poussoir .

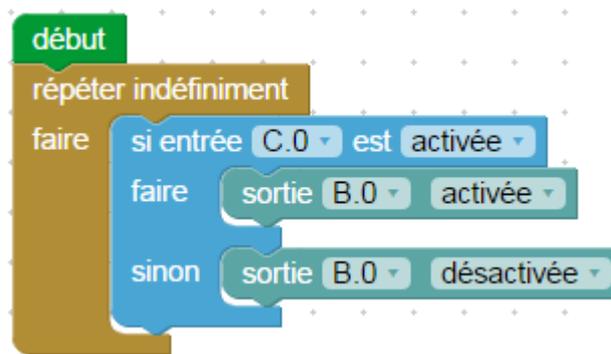


Remarque : Avec un microcontrôleur Picaxe, une **sortie activée reste activée jusqu'à ce qu'on la désactive**, donc ne pas oublier l'instruction « sortie ... désactivée ».

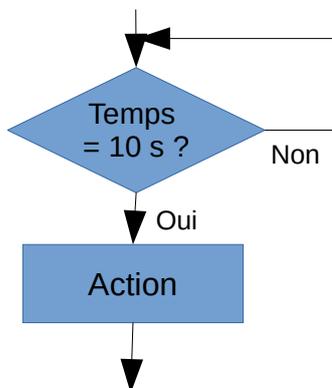
**Boucle « Répéter indéfiniment »**



Permet de recommencer un cycle indéfiniment



**Les temporisations**

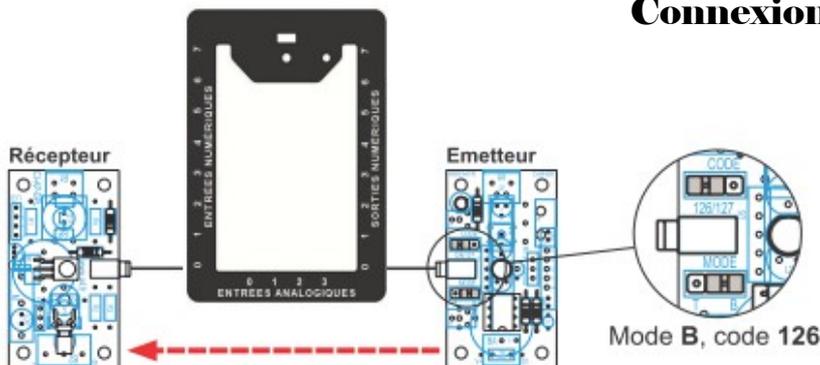


Les temps sont exprimés en millisecondes (ms)  
1 s = 1000 ms

# BLOCKLY pour PICAXE

## Programmer une barrière infrarouge (Ir)

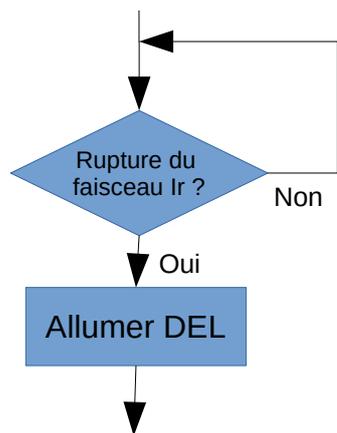
### Connexion des modules



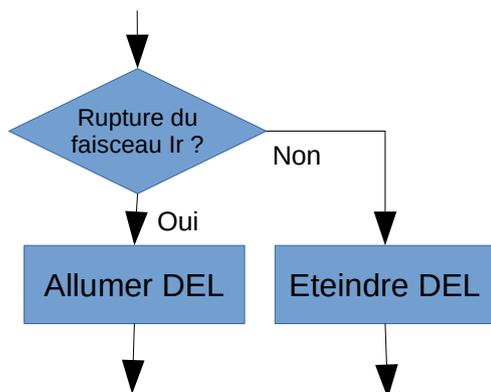
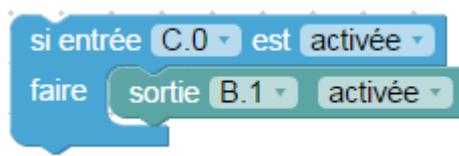
- 1) Vérifier la position des **cavaliers** situés autour du connecteur du module **émetteur infrarouge** qui doit être en **Mode B** et **Code 126**. (Dans ce mode, l'émetteur émet une information IR en permanence).
- 2) Connecter le module **récepteur infrarouge** sur une entrée du boîtier Autoprogram (EN0 à EN7) et le module **émetteur infrarouge** sur une sortie (S0 à S7).

En mode barrière infrarouge, le récepteur Ir se comporte comme un contact « Tout ou rien » :

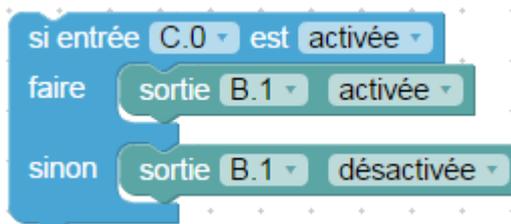
- Faisceau non coupé → Contact ouvert → Entrée à 0 → Entrée désactivée
- Faisceau coupé → Contact fermé → Entrée à 1 → Entrée activée



**Exemple** : Si le faisceau de la barrière infrarouge est coupé, une DEL est allumée.

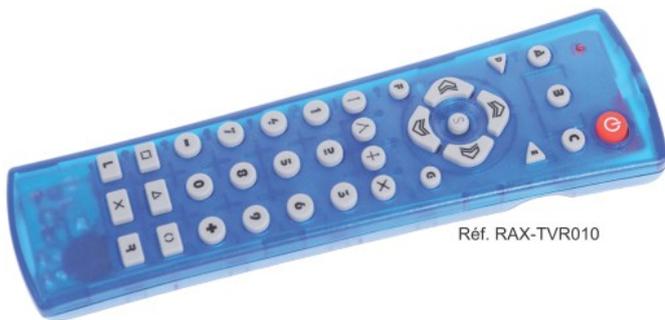


**Exemple** : Si le faisceau de la barrière infrarouge est coupé, une DEL est allumée, sinon cette DEL est éteinte.



# BLOCKLY pour PICAXE

## Recevoir un signal de la télécommande infrarouge



Réf. RAX-TVRO10

Fonctionne avec le module récepteur infrarouge réf. K-AP-MRIR. Voir page 2.11.1.



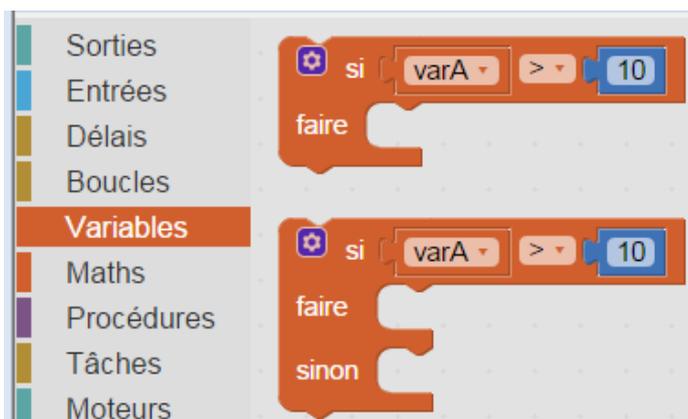
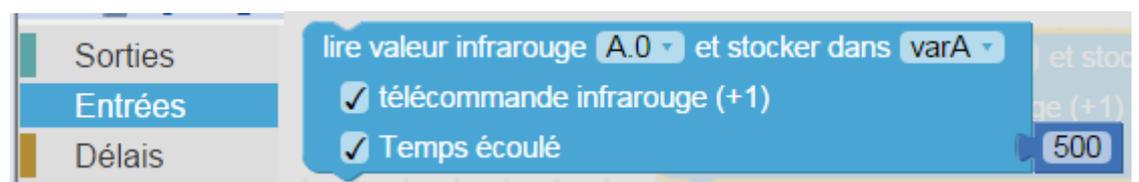
La télécommande PICAXE émet un signal infrarouge qui véhicule un code propre à chaque touche appuyée.

Ce code est reçu par le module récepteur infrarouge réf. K-AP-MRIR ; celui-ci est connecté sur une entrée numérique du boîtier de commande AutoProg.

Touche	Code	Touche	Code	Touche	Code
1	0		21		96
2	1		16		54
3	2		17		37
4	3		19		20
5	4		18		98
6	5				11
7	6				
8	7				
9	8				
0	9				

Le code capté par le récepteur infrarouge est stocké dans une variable.

Les tests se font alors sur cette variable.



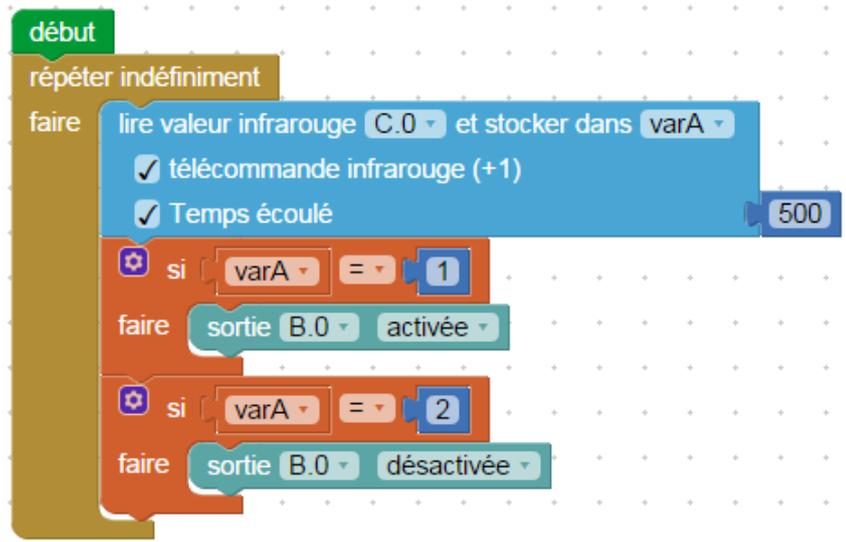
**Remarque** : Pour clarifier, il est possible de donner un nom significatif à la variable.

Pour cela, cliquer sur « varA » et choisir « Renommer la variable ».

# BLOCKLY pour PICAXE

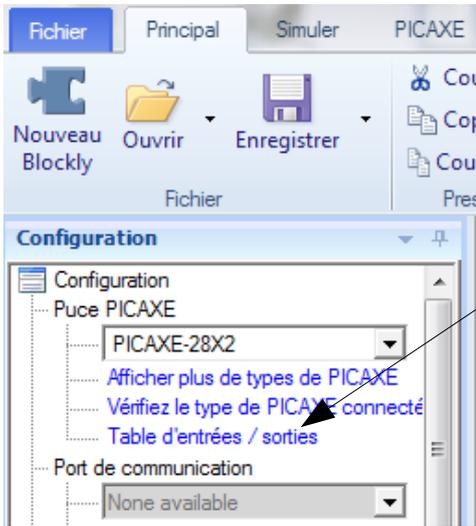
## Recevoir un signal de la télécommande infrarouge

**Exemple** : Un appui sur la touche 1 de la télécommande provoque l'allumage d'une DEL branchée sur la sortie S0.  
Un appui sur la touche 2 provoque l'extinction de cette LED.



# BLOCKLY pour PICAXE

## Nommer les entrées et les sorties



Afin de simplifier la compréhension du programme mis au point avec Blockly, il est possible de nommer les entrées et les sorties.

Pour nommer clairement les entrées et les sorties, on utilise la **table d'entrées / sorties**

Broche	Mon étiquette
A.0	<input type="text" value="A.0"/>
A.1	<input type="text" value="A.1"/>
A.2	<input type="text" value="A.2"/>
A.3	<input type="text" value="A.3"/>
B.0	<input type="text" value="B.0"/>
B.1	<input type="text" value="B.1"/>

OK Annuler

Nommer une entrée ou une sortie consiste à lui attribuer une « **étiquette** » dans la table d'entrées / sorties.

**Attention** : les étiquettes ne peuvent pas comporter plus de 22 caractères.

## Exemple avec le monte-charge

```
début
répéter indéfiniment
faire
  si entrée C.1 est activée
  faire
    sortie B.4 désactivée
    répéter
      moteur D à avancer
    jusqu'à entrée C.3 est activée
  moteur D à arrêter
  sortie B.5 activée

début
répéter indéfiniment
faire
  si entrée BP Appel Haut est activée
  faire
    sortie Voyant Bas désactivée
    répéter
      moteur D à avancer
    jusqu'à entrée Cabine en Haut est activée
  moteur D à arrêter
  sortie Voyant Haut activée
```

Broche	Mon étiquette
B.0	<input type="text" value="Voyant bas"/>
B.1	<input type="text" value="Voyant haut"/>
B.2	<input type="text" value="B.2"/>
B.3	<input type="text" value="B.3"/>
B.4	<input type="text" value="B.4"/>
B.5	<input type="text" value="B.5"/>
B.6	<input type="text" value="Mont. cabine"/>
B.7	<input type="text" value="Desc. cabine"/>
C.0	<input type="text" value="Appel bas"/>
C.1	<input type="text" value="Appel haut"/>
C.2	<input type="text" value="Cabine bas"/>
C.3	<input type="text" value="Cabine haut"/>
C.4	<input type="text" value="C.4"/>

OK Annuler

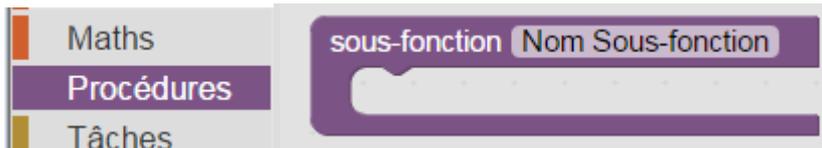
# BLOCKLY pour PICAXE

## Utiliser des sous-programmes

Les sous-programmes permettent :

- de simplifier la conception des programmes ;
- de clarifier la lecture de programmes imposants.

Dans Blockly, un sous-programme est nommé une « sous-fonction ». L'instruction « sous-fonction » se trouve dans le menu « Procédures »



### Exemple : Programmation du monte-charge avec des sous-programmes

