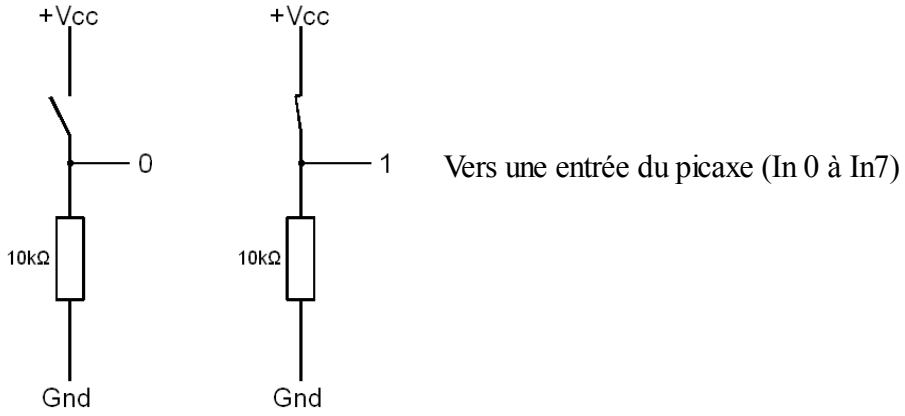
COMPRENDRE LES INTERRUPTIONS

1 - Les interruptions sur un Picaxe

Une interruption est une prise en compte immédiate d'un évènement afin d'exécuter un traitement spécifique appelé routine d'interruption.
En effet, la programmation (avec Programming Editor) est de type séquentielle. Les enchainements d'opérations s'effectuent dans un ordre bien défini, seuls les embranchements des tests permettent de modifier ce bel assemblage.
Il peut être nécessaire de traiter rapidement un événement sans attendre de passer par un test, c'est le rôle d'une interruption. L'évènement le plus simple sur un Pic est un changement d'état d'une entrée.

Pour exploiter le mode interruption, nul besoin d'un schéma électrique spécifique, un interrupteur sur une des entrées du picaxe suffit :



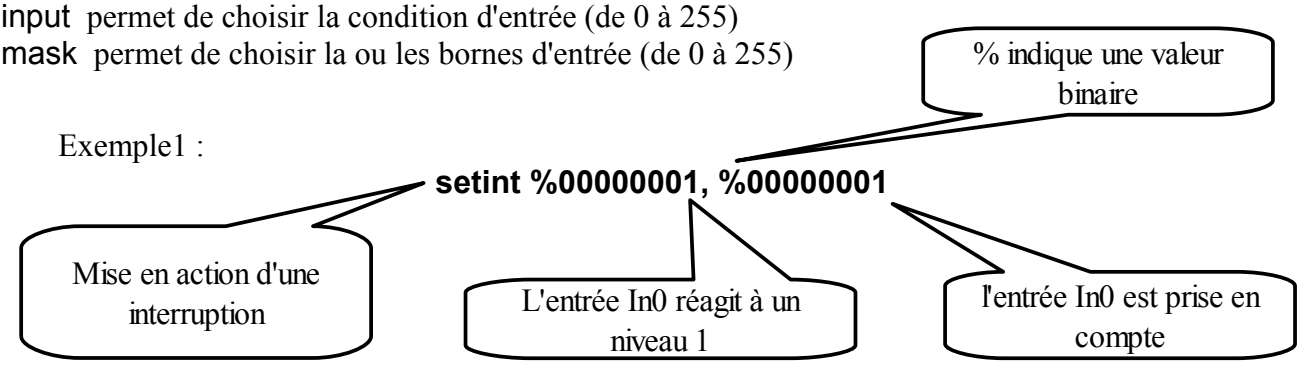
2 - Exploiter une entrée en mode interruption

La mise en œuvre d'une interruption est disponible grâce à la commande setint.
Cette commande peut être configurée pour de nombreux modes de fonctionnement et des informations sont disponibles dans le manuel Picaxe2 - BASIC commands. La syntaxe la plus simple est :

setint input, mask

- input permet de choisir la condition d'entrée (de 0 à 255)
- mask permet de choisir la ou les bornes d'entrée (de 0 à 255)

Exemple1 :



Il serait possible d'écrire **setint 1, 1**

Exemple2 :

setint %00000000, %00000010

L'entrée In1 réagit à un niveau 0

l'entrée In1 est prise en compte

Il serait possible d'écrire **setint 0, 2** (10base2 = 2base10)

Exemple3 :

setint %00000011, %00000111

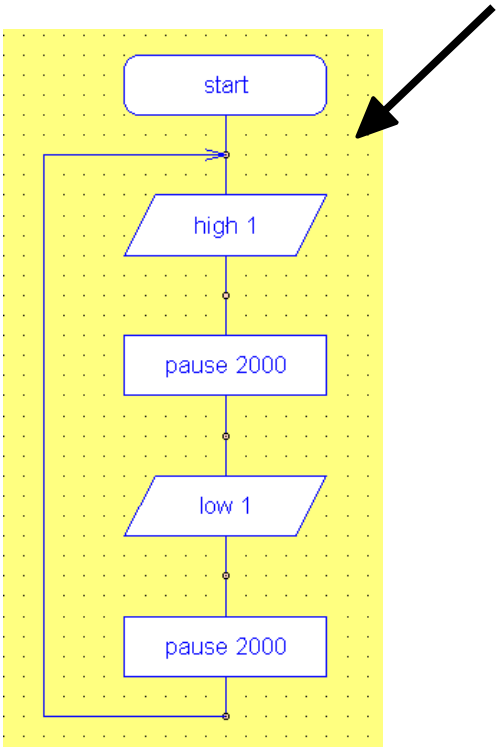
L'entrée In0 réagit à 1 et In1 à 1 et In2 à 0

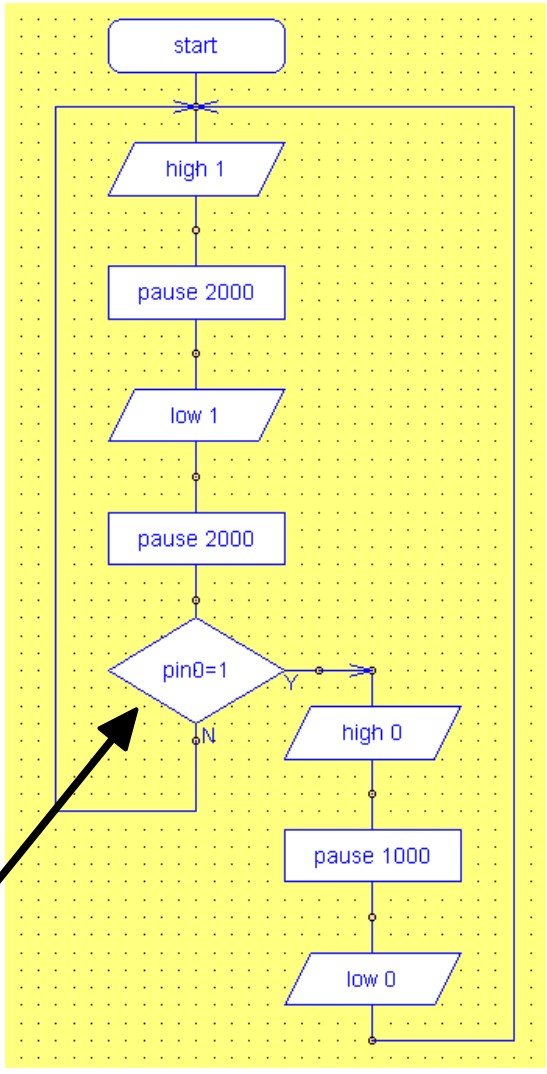
l'entrée In0, In1et In2 sont prises en compte

Il serait possible d'écrire **setint 3, 7** (11base2 = 3base10 et 111base2 = 7base10)

3 - Exemples de programme

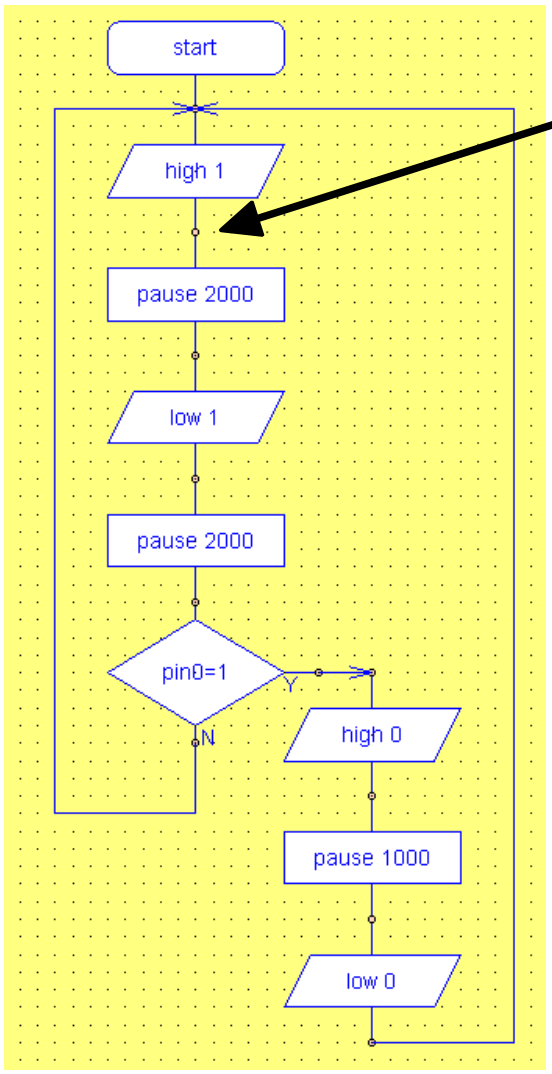
Le programme suivant est très simple, il permet de faire clignoter une diode Led branché sur la sortie 1 :





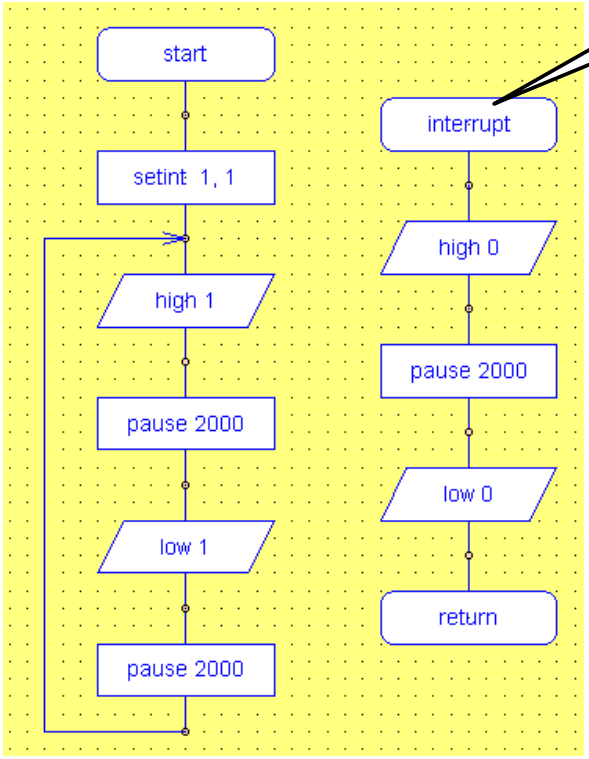
Si l'on veut détecter l'appui d'un Bouton poussoir sur l'entrée In0 permettant l'allumage d'une Led sur la sortie 0, l'idée est de faire un test.

Bien sur, cette solution fonctionne, cependant, si l'appuie du bouton poussoir s'effectue à cet instant



Il va falloir patienter pendant 4 secondes.
Pire si l'appui du Bp est fugitif, la prise en compte n'est pas effectuée...

L'interruption permet de nous sortir de ce mauvais pas....
Le programme suivant fonctionne en interruption :



Pour écrire cette commande, aller dans le menu **sub**

Une interruption est gérée comme une routine.
Elle doit toujours terminée par return.

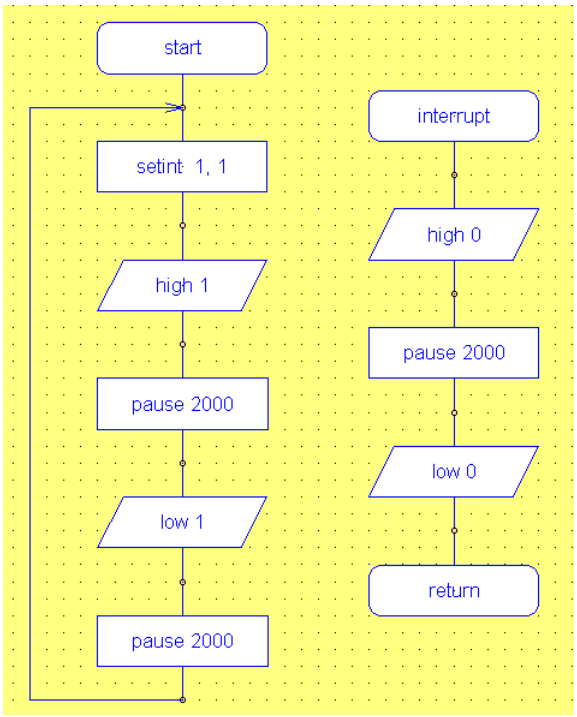
Maintenant, la prise en compte en compte est immédiate.

Cependant, seul le premier appui du Bp est pris en compte...

Cette interruption doit être réarmée.

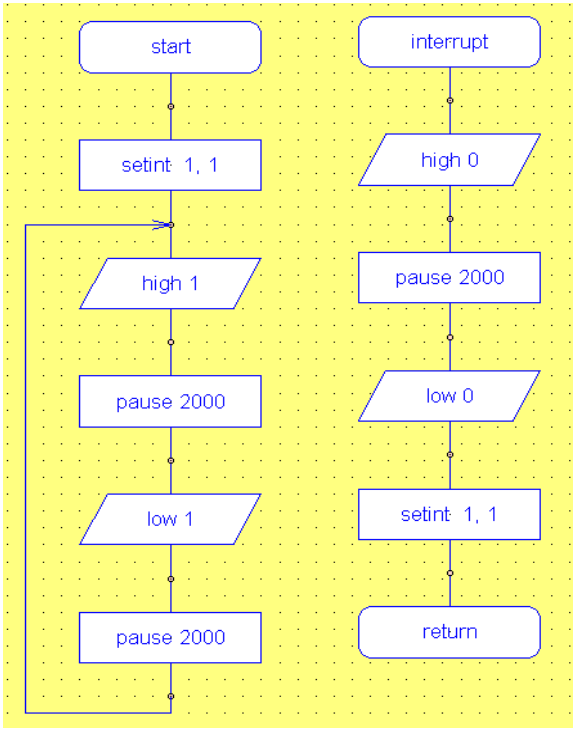
Différentes solutions pour réarmer l'interruption :

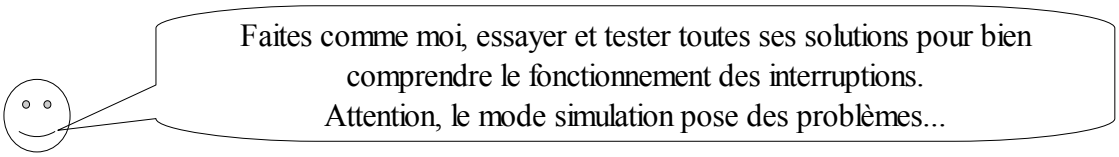
Rebouclage du programme sur l'interruption. Possible mais risque d'attente...



Réarmement à la fin de la routine

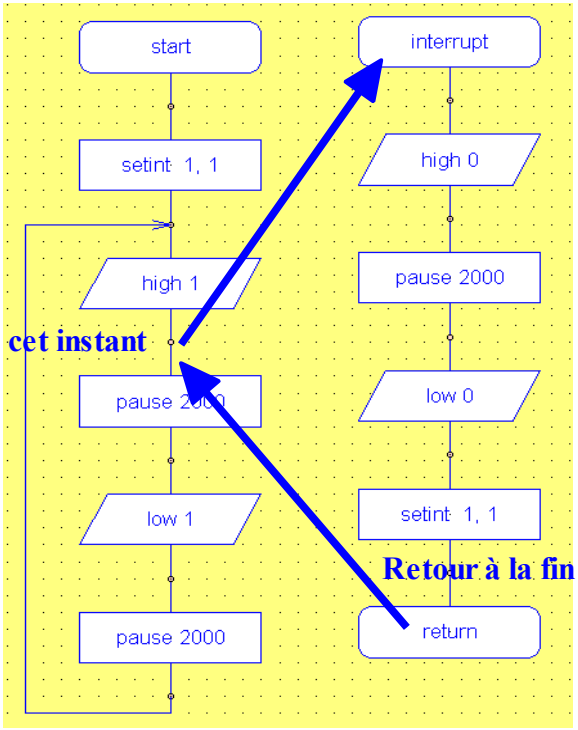
Solution la plus efficace...





Les interruptions fonctionnent comme les sous-programmes. Ainsi, dès le « return » le programme retourne à la séquence au moment de l'appel de l'interruption :

Demande d'interruption à cet instant

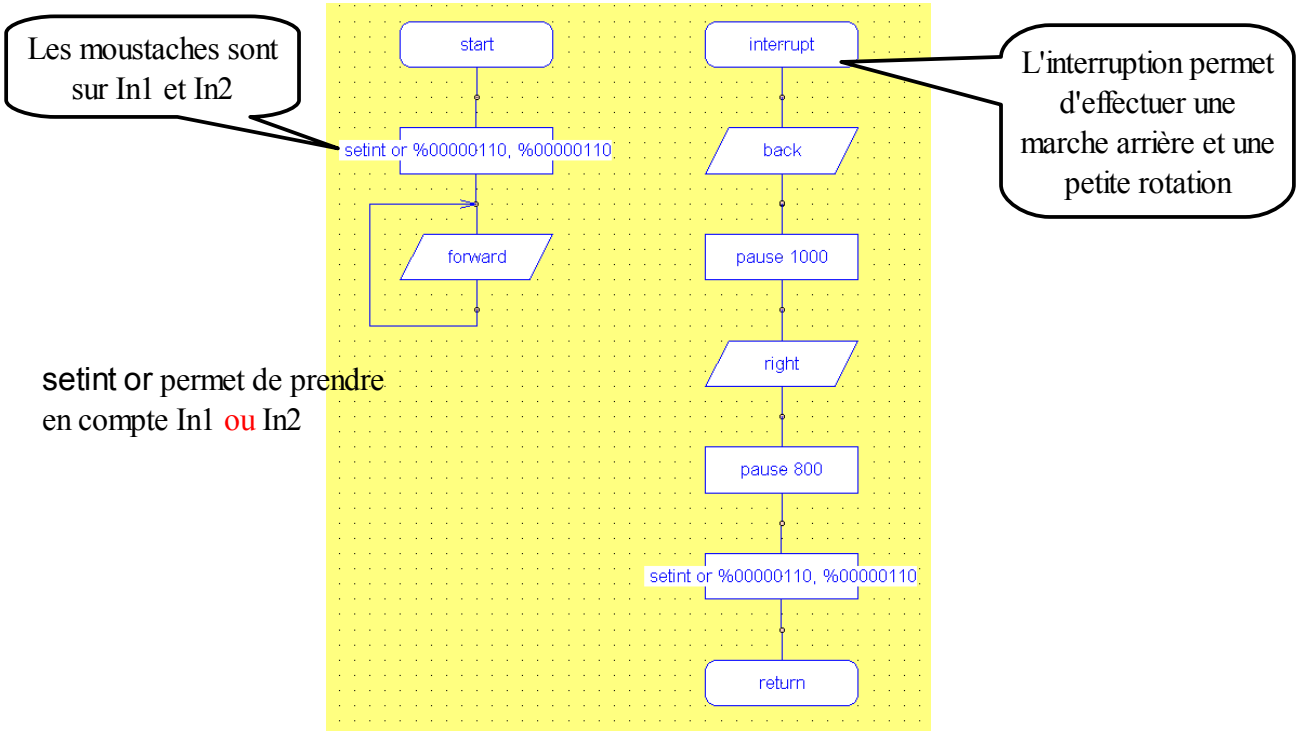


Retour à la fin de l'interruption

Une seule interruption par programme.

4 - Contrôler un robot mobile sous interruption

La gestion d'un mobile sous interruption est particulièrement efficace pour pouvoir prendre en compte instantanément les détections des obstacles par les moustaches.
Exemple de programme d'un robot avançant en ligne droite et réagissant à des obstacles :



Programme complet permettant de réagir différemment en fonction de la détection :