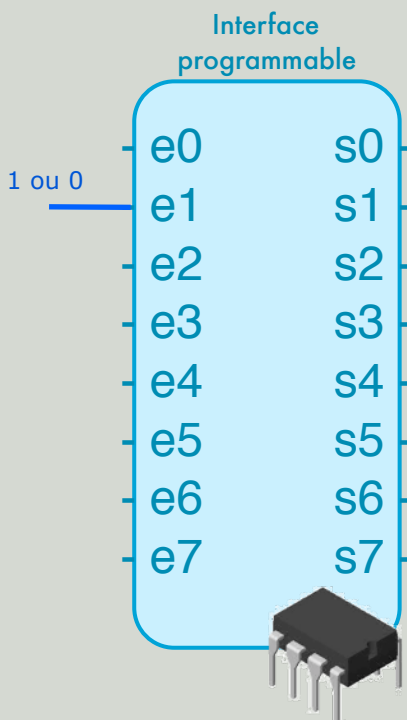


## EXPLICATIONS



Une interruption est une prise en compte **immédiate** du changement d'état d'une entrée (de 0 à 1 ou inversement).

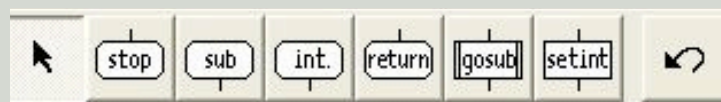
En effet, la programmation sur PICAXE est de type séquentielle : Les enchainements d'opérations s'effectuent dans un ordre bien défini, seuls les embranchements des tests (questions) permettent de modifier ce bel assemblage.

Il peut être nécessaire de traiter rapidement un événement sans attendre de passer par un test, c'est le rôle d'une interruption.



**Sur PICAXE : 1 seule interruption par programme !**

## ENTRÉE EN MODE INTERRUPTION



Sous programme INTERRUPTION

Retour

INTERRUPTION

# PARAMÉTRER L'INTERRUPTION

In7 - In6 ... In3 - In2 - In1 - In0  
 In7 - In6 ... In3 - In2 - In1 - In0  
 1/0 - 1/0 ... 1/0 - 1/0 - 1/0 - 1/0

**setint %0000000**1**, %0000000**1** = setint 1, 1**

En décimal

Mise en action d'une interruption

L'entrée In0 réagit à un niveau logique 1

Uniquement l'entrée In0 est prise en compte

% indique une valeur binaire

**setint %0000000**0**, %00000**1**00 = setint 0, 4**

En décimal

Mise en action d'une interruption

L'entrée In2 réagit à un niveau logique 0

Uniquement l'entrée In2 est prise en compte

Conversion : Binaire / Décimal

Binaire	2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
Décimal = 4	0	0	0	0	0	1	0	0
Décimal = 0	0	0	0	0	0	0	0	0

$$(2^7 \times 0) + (2^6 \times 0) + (2^5 \times 0) + \dots + (2^2 \times \mathbf{1}) + (2^1 \times 0) + (2^0 \times 0) = \mathbf{4}$$

$$(2^7 \times 0) + (2^6 \times 0) + (2^5 \times 0) + \dots + (2^2 \times \mathbf{0}) + (2^1 \times 0) + (2^0 \times 0) = \mathbf{0}$$

**setint %000000**011**, %00000**111** = setint 3, 7**

En décimal

Mise en action d'une interruption

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

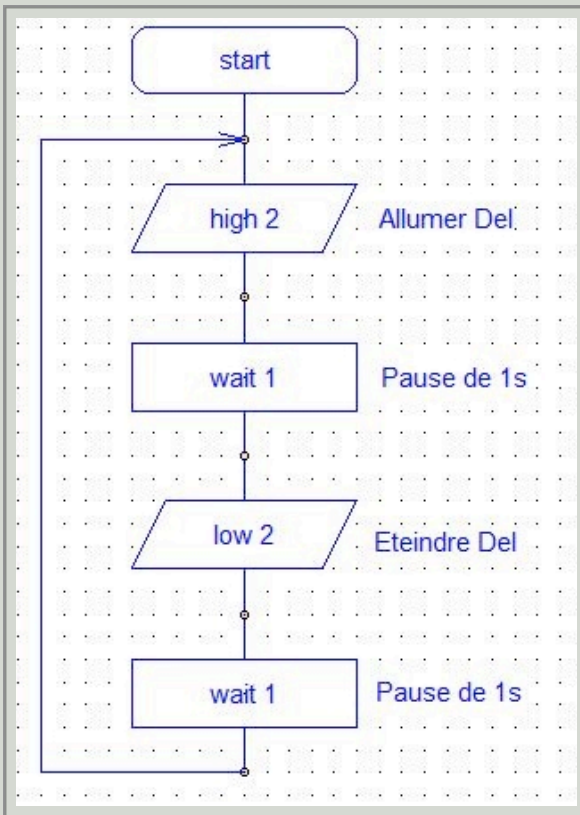
Uniquement les entrée In0, In1 et In2 sont prises en compte

$$(2^7 \times 0) + (2^6 \times 0) + (2^5 \times 0) + \dots + (2^2 \times \mathbf{1}) + (2^1 \times \mathbf{1}) + (2^0 \times \mathbf{1}) = \mathbf{7}$$

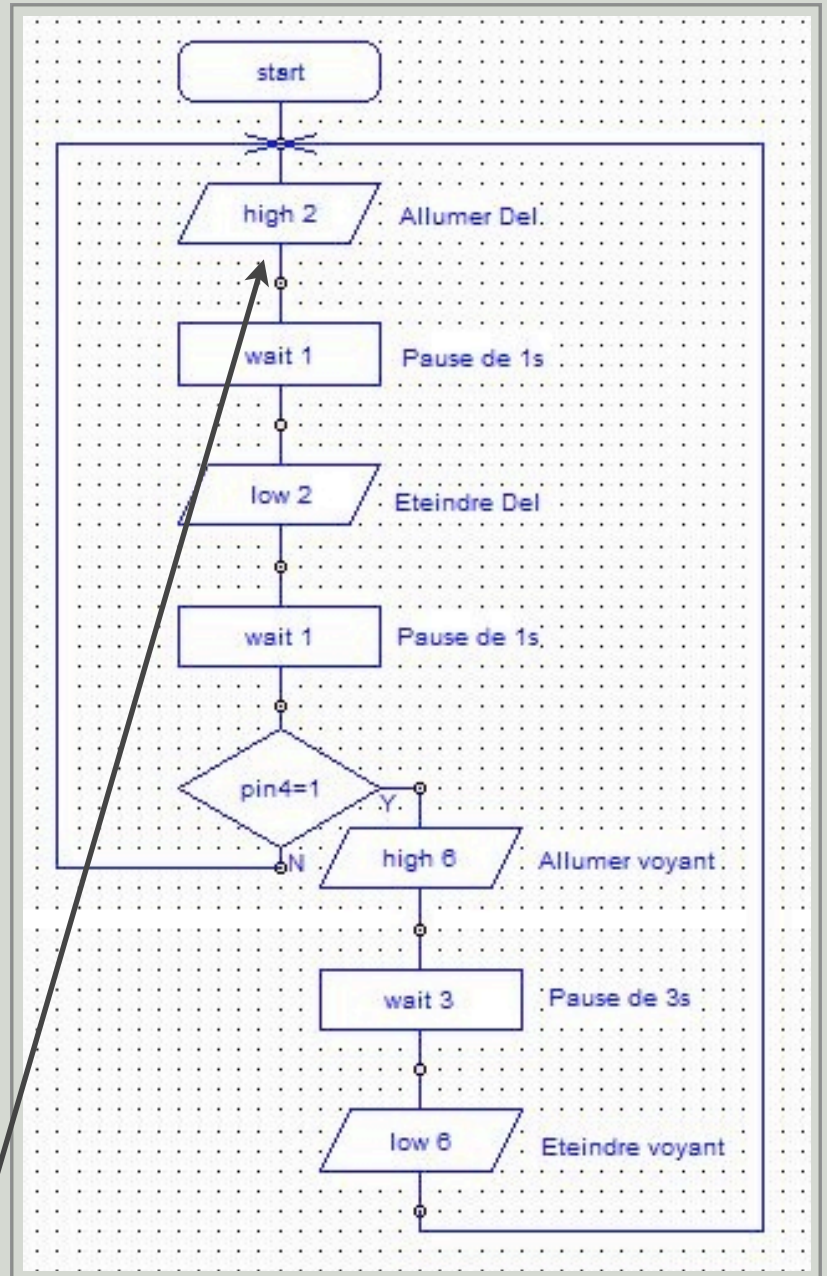
$$(2^7 \times 0) + (2^6 \times 0) + (2^5 \times 0) + \dots + (2^2 \times \mathbf{0}) + (2^1 \times \mathbf{1}) + (2^0 \times \mathbf{1}) = \mathbf{3}$$

# PROGRAMME SANS INTERRUPTION

Le programme suivant est très simple, il permet de faire clignoter une DEL sur la sortie 2 toutes les secondes :



Si l'on veut détecter l'appui d'un Bouton poussoir sur l'entrée In4 permettant d'allumer une Del sur la sortie 6, l'idée est de faire un test.



Bien sur, cette solution fonctionne, cependant, si l'appuie du bouton poussoir s'effectue à **cet instant précis**, il va falloir patienter pendant 2 secondes.

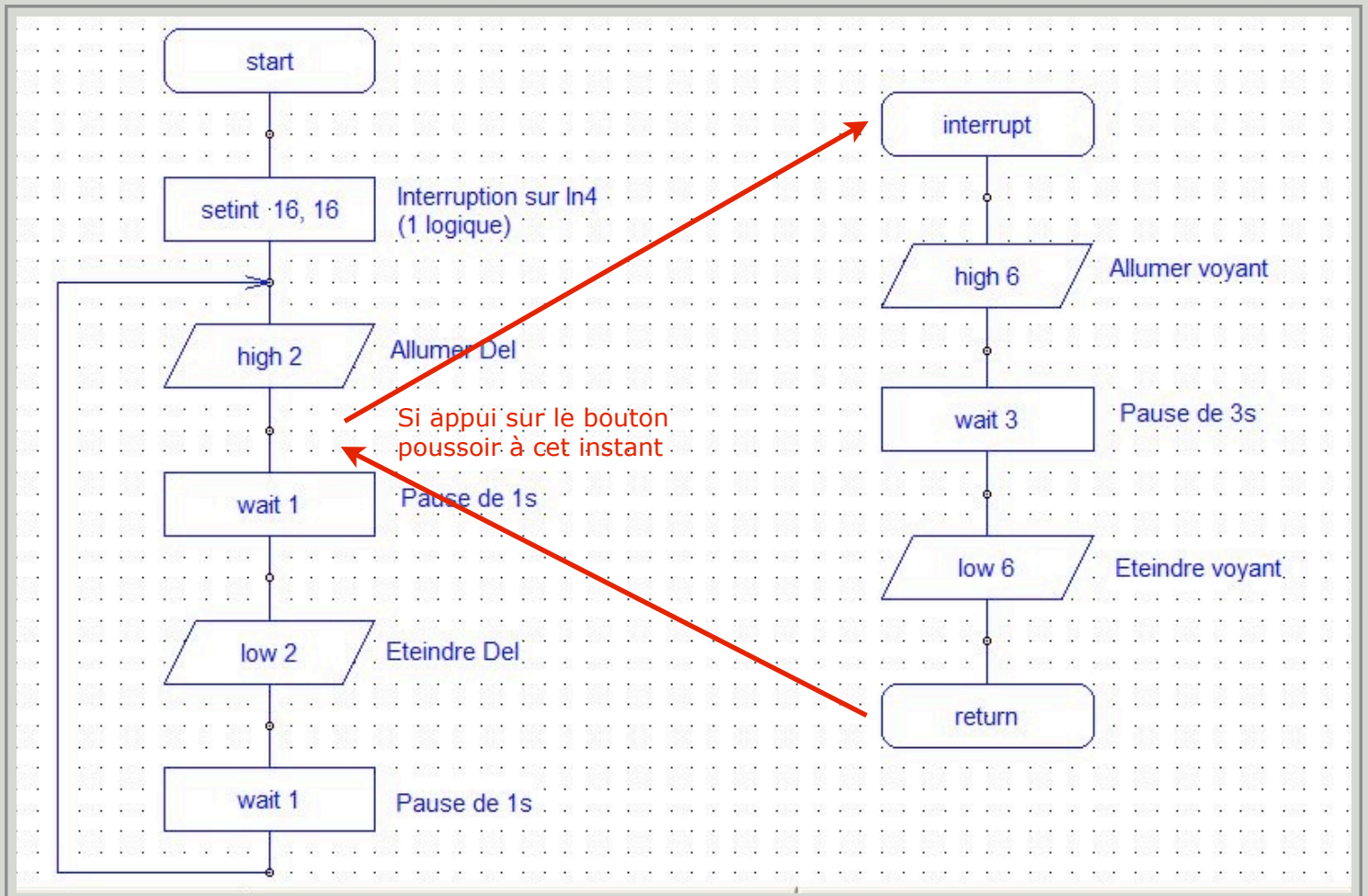
Pire si l'appui du bouton poussoir est fugitif, la prise en compte n'est pas effectuée...

L'interruption permet de nous sortir de ce mauvais pas....

# PROGRAMME AVEC INTERRUPTION

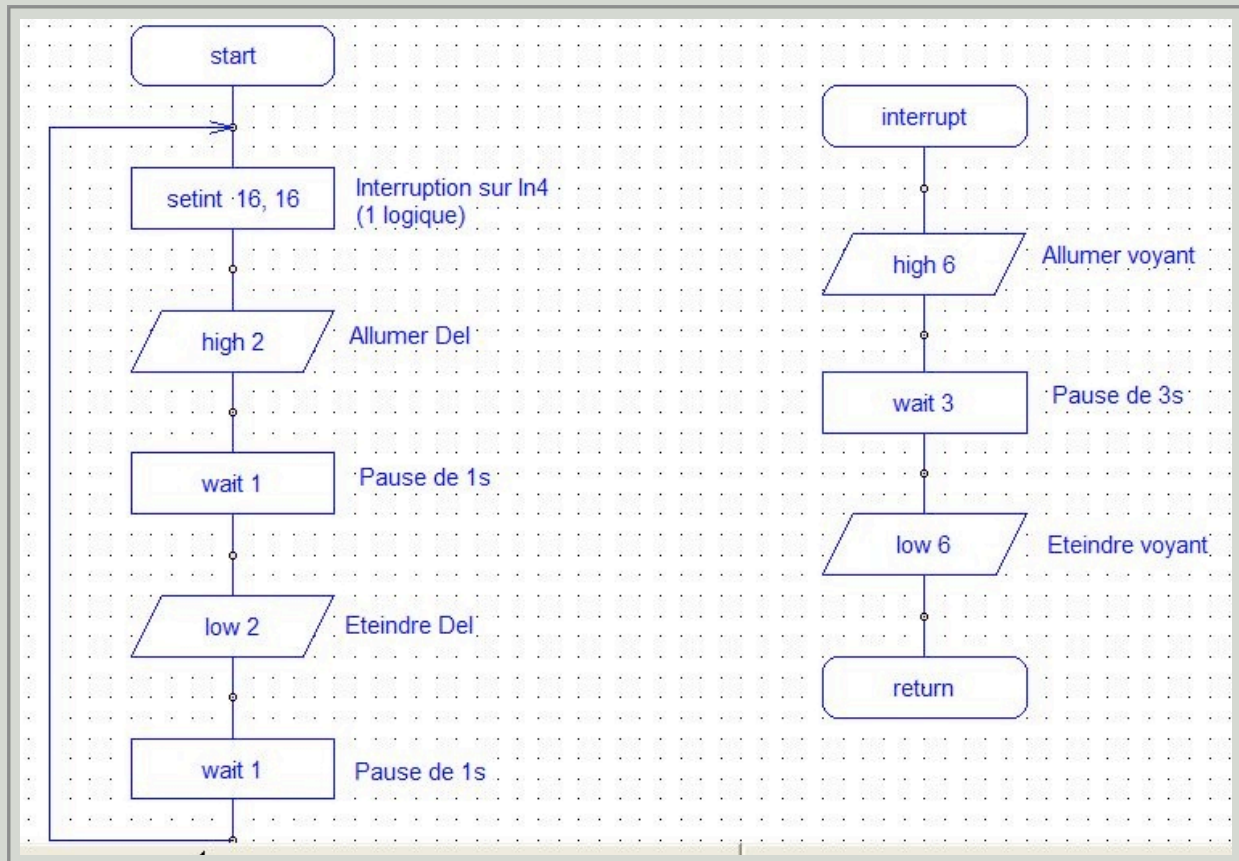
Une interruption est gérée comme une routine (un sous-programme). Elle doit toujours terminée par «Return».

Maintenant, la prise en compte du bouton poussoir est immédiate.



Cependant, seul le premier appui sur le bouton poussoir est pris en compte ... Effectivement, il faudra redémarrer le programme pour qu'une autre interruption soit pris en compte.

Une des solutions pour palier à ce problème consiste à reboucler sur l'interruption. Possible mais il y a des risques d'attente...



La solution la plus efficace consiste à «réarmer» l'interruption en fin de routine de l'interruption, comme ici :

