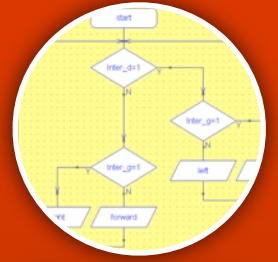


LOGICIEL PICAXE

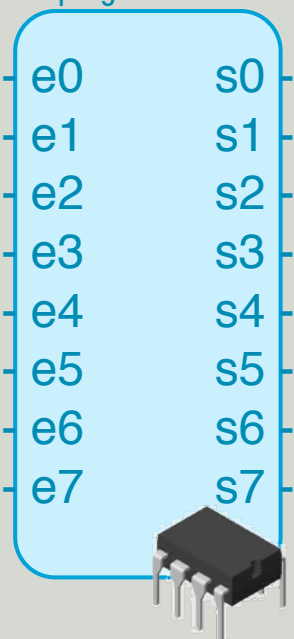
LE TIMER



Programming
Editor

EXPLICATIONS

Interface
programmable



Le timer est un compteur permettant de compter :

- ★ des cycles d'horloge du picaxe lui-même. Comme l'horloge du picaxe est fixe, nous pourrons ainsi disposer d'une base de temps précise.
- ★ de compter des impulsions sur une pin spécifique du pic : L'entrée In0.

Le timer travaille sur 16 bits, ce qui limite le comptage à 65536.

Conversion : Binaire / Décimal

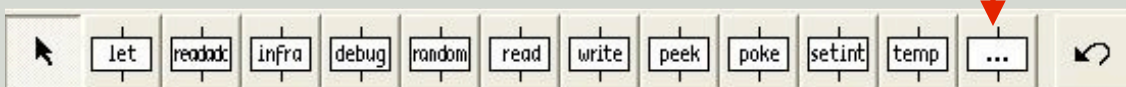
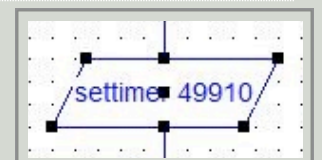
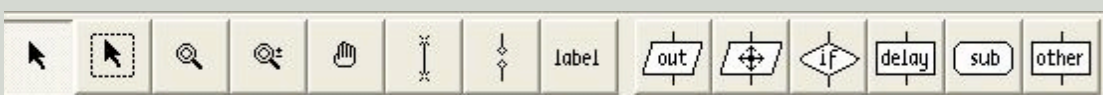
Binaire	2 ¹⁵	2 ¹⁴	2 ¹³	2 ^{...}	2 ³	2 ²	2 ¹	2 ⁰
Décimal = 65536	1	1	1	1	1	1	1	1

$$(2^{15} \times 1) + (2^{14} \times 1) + \dots + (2^1 \times 1) + (2^0 \times 1) = 65536$$



**Sur PICAXE : 1 seul timer
par programme !**

LANCER ET ARRÊTER LE TIMER



«settimer off»
permet d'arrêter le timer

«settimer + preload»
permet de configurer le timer

BASIC settimer 49910

PARAMÉTRER LA TEMPORISATION DU TIMER

L'appel d'incréméntation du timer (tempo) dépend de la fréquence de fonctionnement du picaxe et de la valeur du preload que vous configurez.

Généralement les interfaces picaxes utilise comme horloge fixe un quartz de fréquence 4 Mhz.

Petit rappel : 4 Mhz = 4 x 10⁶ Hz

$$\text{preload} = 65535 - \frac{(\text{Tempo désirée} \times \text{Fréquence du Picaxe})}{256}$$

Pour une tempo de 1 s :

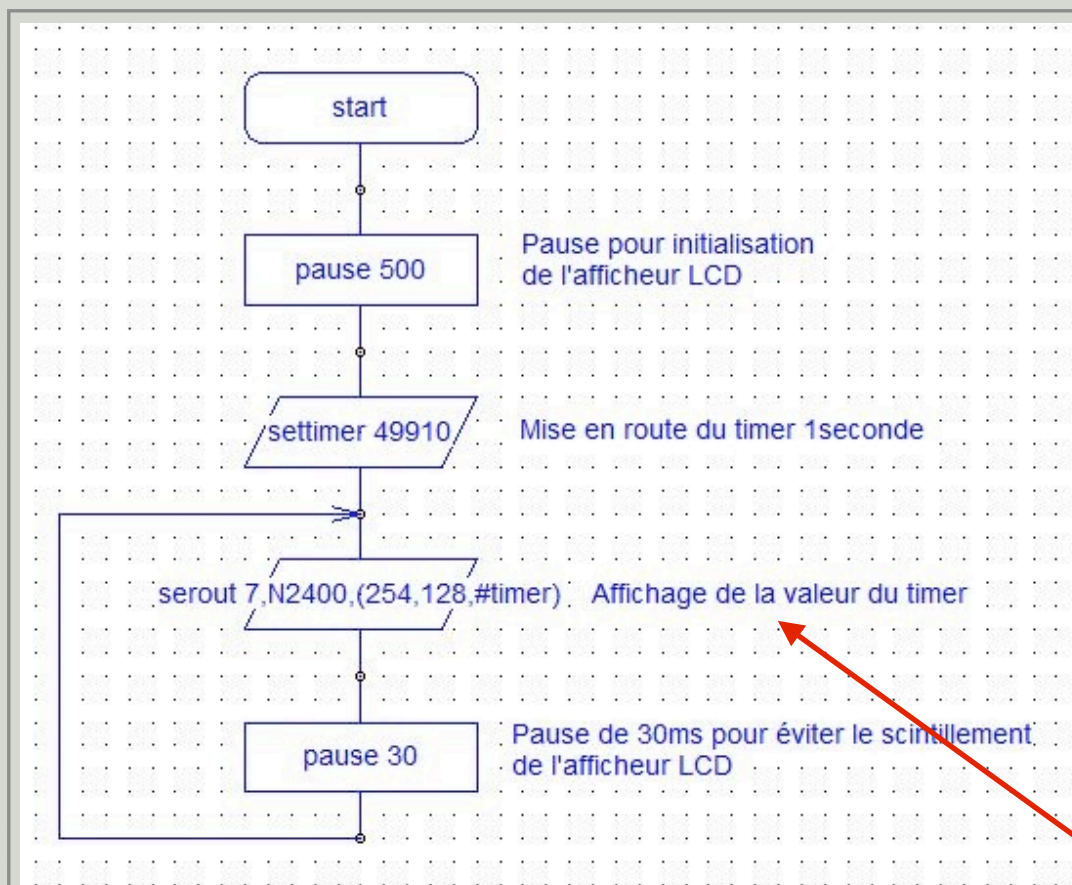
$$\text{preload} = 65535 - (1/256 \times 4 \times 10^6) = 49910$$

Pour une tempo de 0,1 s :

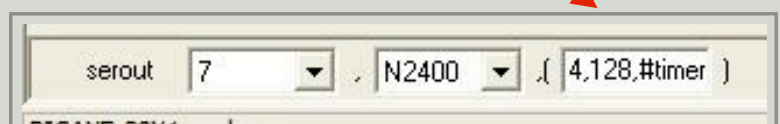
$$\text{preload} = 65535 - (0,1/256 \times 4 \times 10^6) = 63972,5 \text{ soit } 63972 \text{ ou } 63973$$

(bien sur dans ce cas, il va y avoir une petite erreur de temps de comptage)

EXEMPLE SIMPLE : TIMER + AFFICHEUR



Ici l'organigramme permet d'afficher la valeur du «timer» sur l'afficheur. Le timer étant calé sur 1 seconde (49910), l'afficheur affiche les secondes écoulées.

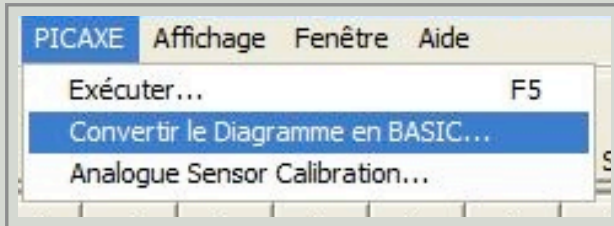


COMMENT SIMULER LE TIMER ?



Attention il n'est pas possible de tester le Timer en mode Floatchart.

Il n'est pas possible de simuler le fonctionnement du Timer en mode Floatchart. Il faut obligatoirement le transférer en mode Basic (langage de programmation).



Pour cela :
Menu PICAXE > Convertir le Diagramme en BASIC

Le langage BASIC reste simple de lecture, goto label_14 indique «aller au» titre label_14, le reste des instructions sont identique à l'organigramme.

```
1 | BASIC converted from flowchart:
2 | 'C:\Documents and Settings\pascal\Mes documents\test
3 | 'Converted on 12/08/2011 at 16:18:46
4 |
5 |
6 |
7 | main:
8 |     pause 500
9 |     settimer 49910
10 | label_14: serout 7,N2400,(254,128,#timer)
11 |     pause 30
12 |     goto label_14
```



Pensez à enregistrer l'organigramme avant de le convertir en Basic !

```
'BASIC converted from flowchart:
'C:\Documents and Settings\pascal\Mes documents\test-picaxe\timer-et-afficheur
'Converted on 12/08/2011 at 16:21:12

main:
    pause 500
    settimer 49910
label_14: serout 7,N2400,(254,128,#timer)
    pause 30
    goto label_14
```

Il est maintenant possible de lancer la simulation.
Pensez à activer ici la sortie 7 en tant que afficheur LCD !

Menu Option > Onglet Simulation > «Simuler un LCD avec la commande Serout

Simulation Options...

Attente entre chaque étape de la simulation (en ms)... 102

- Highlight (and delay) over program labels
- Automatically hide panels when simulation ends.
- Emission de bips, simulation des commandes 'sound' (son)
- Display serial baud rate information messages
- Simuler un LLD serie avec la commande serout (sortie série) en sortie 7
- Simulate remote control for infrain2
- Simulate EEPROM on i2c bus 24LC16B 24LC512
- Simulate DS1307 RTC on i2c bus

Montrer les options au démarrage

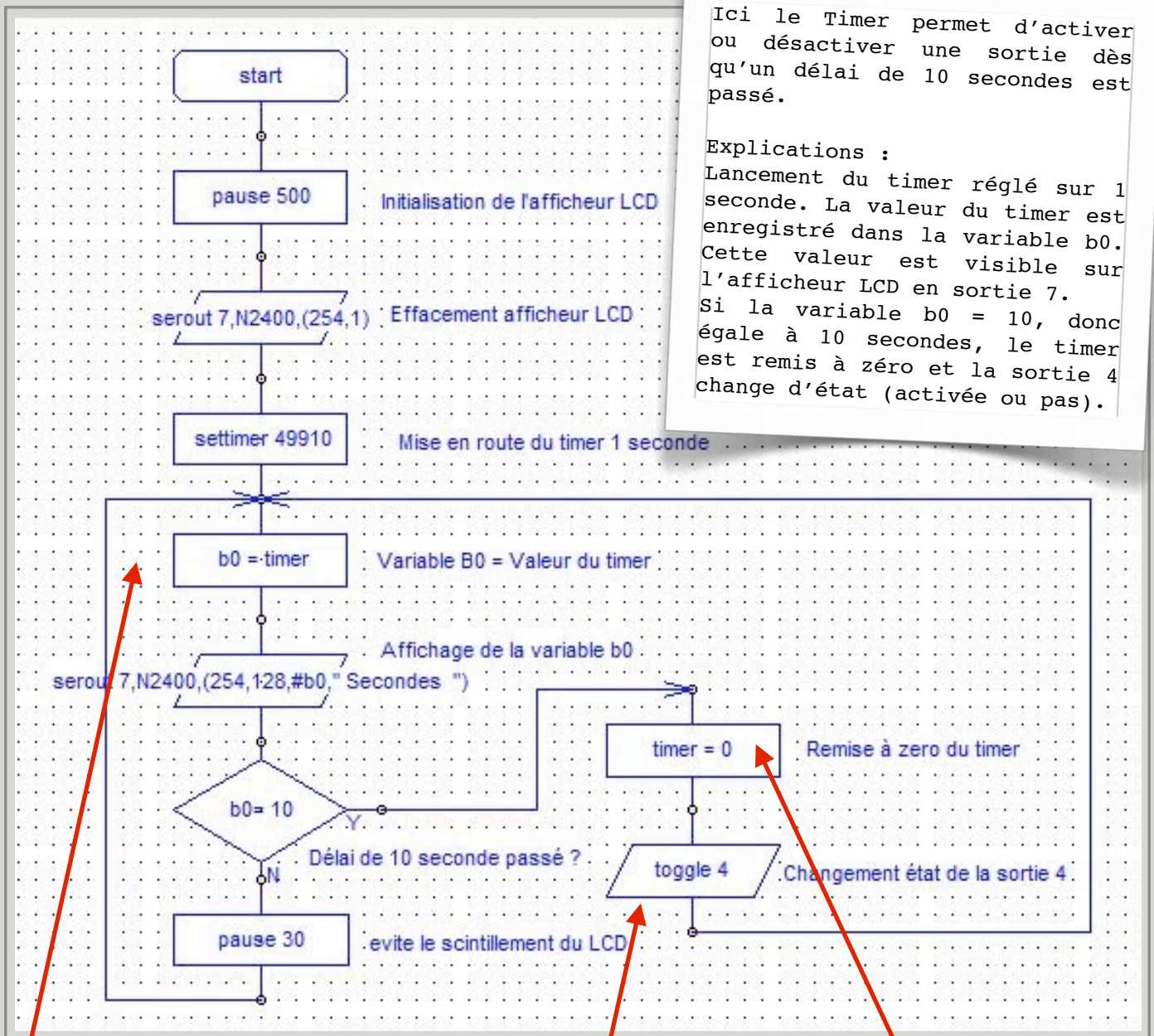
OK Annuler Appliquer

AUTRE EXEMPLE SIMPLE : TIMER + VARIABLE

Ici le Timer permet d'activer ou désactiver une sortie dès qu'un délai de 10 secondes est passé.

Explications :

Lancement du timer réglé sur 1 seconde. La valeur du timer est enregistré dans la variable b0. Cette valeur est visible sur l'afficheur LCD en sortie 7. Si la variable b0 = 10, donc égale à 10 secondes, le timer est remis à zéro et la sortie 4 change d'état (activée ou pas).



BASIC `b0 = timer`

BASIC `timer = 0`

Navigation icons: arrow, magnifying glass, hand, vertical line, label, out, if, delay, sub, other.

Function icons: high, low, pins, toggle, sound, serout, servo, play, tune, irout, ...

toggle 4

```

1 'BASIC converted from flowchart:
2 'C:\Documents and Settings\pascal\Bureau\timer-img\test_timer1sec.c
3 'Converted on 01/01/2012 at 19:33:36
4
5
6 main:
7     pause 500
8     serout 7,N2400,(254,1)
9     settimer 49910
10
11 label_22:
12     b0 = timer
13     serout 7,N2400,(254,128,#b0," Secondes ")
14     if b0= 10 then label_44
15     pause 30
16     goto label_22
17
18 label_44:
19     timer = 0
20     toggle 4
21     goto label_22

```

Comme il n'est pas possible de simuler le fonctionnement du Timer en mode Floatchart.
 La simulation de cet exemple se fait obligatoirement en mode Basic :
 Menu PICAXE > Convertir le Diagramme en BASIC

The screenshot shows the PICAXE software interface. The main window displays the BASIC code. A 'Simulation' window is open, showing a digital display with pins 0-7 and a table of variables. A 'Serial LCD...' window is also open, showing '7 Secondes'. Red arrows point from the 'Serial LCD' window to the 'Simulation' window, and from the 'Simulation' window to the text below.

b0	7	%00000111
b1	0	%00000000
b2	0	%00000000
b3	0	%00000000
b4	0	%00000000
b5	0	%00000000
b6	0	%00000000
b7	0	%00000000
b8	0	%00000000
b9	0	%00000000
b10	0	%00000000
b11	0	%00000000
b12	0	%00000000
b13	0	%00000000

Vérification :
 Effectivement, toutes les 10 secondes
 la sortie 4 change d'état.