

# Temporisation en langage machine

par André C. avec l'aide de Christian L.

Cet article a été corrigé grâce aux indications précieuses de Christian L. "Assinie" concernant l'interférence des interruptions dans la routine TEMPOB. Qu'il en soit cordialement remercié.

La commande Basic WAIT est une petite merveille de simplicité d'utilisation. Il suffit d'indiquer une valeur de temporisation en centièmes de secondes et pourvu que les IRQ ne soient pas inhibées, l'exécution du programme se fige pendant le délai indiqué. La valeur du délai indiqué doit être comprise entre 0 et 65535 (en pratique entre 1 à 65535), ce qui permet des temporisations de 0,1 à 655 secondes (presque 11 min donc). C'est plus qu'il n'est utile. La précision de WAIT est étonnamment bonne. Mais il n'est pas possible d'obtenir des délais très brefs,

car la valeur minimale est de 100 µs, ce qui est beaucoup comparé aux 2 µs qu'il est possible d'obtenir en langage machine, en insérant un NOP dans un programme. En effet NOP (No OPeration) ne fait rien, mais ça prend 2 cycles de microprocesseur, soit 2 µs dans le cas de l'Oric. Moi quand je ne fais rien, ça prend quand même plus de temps, heureusement ça n'arrive pas souvent !

## Temporisation à boucle

Dans la plupart des programmes LM ainsi que dans les routines de la Rom, les temporisations sont basées sur des boucles. Mais vous verrez qu'il y a peut-être mieux à faire.

Voici le principe d'une temporisation à boucle :

```
DEBUT   LDX #5       ; Initialise le compteur avec la valeur 5
BOUCL   DEX          ; Décrémente le compteur
        BNE BOUCL   ; Reboucle tant que le compteur n'a pas atteint 0
SUITE   ; Exécute la suite quand le délai est écoulé
```

C'est simple non ? Mais si on veut calculer le délai correspondant à cette valeur 5, ce n'est pas si facile. En effet, il faut tenir compte du nombre de cycles de microprocesseur que prend chaque instruction.

De plus, très souvent ce nombre de cycle n'est pas une valeur fixe, mais dépend du contexte. Ainsi si LDX et DEX sont des cas simples car nécessitant toujours 2 cycles chacun, c'est plus compliqué avec BNE où trois cas peuvent se présenter :

- 2 cycles si le branchement n'a pas lieu.
- 3 cycles si le branchement a lieu (cas normal pendant la boucle).
- 4 cycles si le branchement a lieu avec un changement de page mémoire.

Si on fait un bilan pour l'exemple ci-dessus, on a 2 cycles pour LDX, plus 4 fois 2+3 cycles (2 pour DEY +

3 pour BNE avec branchement) plus 1 fois 2+2 cycles (2 pour DEY + 2 pour BNE sans branchement) on arrive à un total de 26 cycles, soit 26 µs pour l'Oric, qui tourne à 1 MHz. Le calcul précédent peut se formuler ainsi : Le délai vaut  $D = 5Y - 1 + 2$  ( 1 pour le dernier BNE sans branchement et +2 pour le LDX initial).

Le délai maximum pour une simple boucle est obtenu avec  $Y=\#FF$  et se limite à environ 1,3 ms. ( $D = 256 \times 5 - 1 + 2 = 1281 \mu s$ ). C'est exactement ce code que l'on trouve dans la Rom en #FAE1 pour la durée de ZAP.

## Temporisation à 2 boucles

Si on veut obtenir un délai plus long, il faut imbriquer deux boucles. C'est ce qu'a fait Fabrice Broche pour OUPS, le nouveau son préprogrammé du Telestrat :

```
DEBUT   LDY #$60     ; 2 cycles (Y=96 en décimal)
        LDX #$00     ; 2 cycles
BOUCL   DEX          ; 2 cycles, 1e boucle
        BNE BOUCL   ; 3 cycles, 1e boucle
        DEY         ; 2 cycles, 2e boucle
        BNE BOUCL   ; 3 cycles, 2e boucle
SUITE
```

Essayons de calculer le délai produit par l'exécution de cette routine :

- Pour les LDY et LDX d'initialisation : 4 cycles.
- Pour la 1e boucle :  $D = 256 \times 5 - 1 = 1279$  cycles (-1 pour le dernier BNE de la 1e boucle). Notez que cette 1e boucle se termine avec  $X=0$  donc X est régénéré à sa valeur initiale pour la suite.
- Pour la 2e boucle :  $D = 96(1279+5) - 1 = 123263$  cycles (avec 1279 pour l'exécution de la 1e boucle et -1 pour le dernier BNE sans branchement de la 2e boucle). Notez que le temps d'exécution d'un tour de la 2e boucle inclus celui de l'exécution totale de la 1e boucle.

Au total on a donc 123267 cycles, soit 123267  $\mu$ s pour l'Oric. La durée théorique finale de OUPS est donc de 0,123 s (sauf si je me suis planté). On voit aussi que l'on peut **simplifier ce calcul** car il y a quelques cycles négligeables, par exemple les LDY et LDX initiaux et la réduction de 1 cycle pour BNE sans branchement. On arrive alors à un délai  $D = 96(1280+5) = 123360 \mu$ s, soit une approximation de +0,08%.

Le délai maximum que l'on puisse obtenir avec 2

boucles avec  $X=\#FF$  et  $Y=\#FF$  est d'environ 0,33 ms.

Le **calcul simplifié** donne  $D = 256(1280+5)=328960 \mu$ s. Ça risque de n'être pas assez dans certains cas. On pourrait ajouter quelques NOPs (2 cycles à chaque fois) dans la 1e boucle qui est exécutée  $256 \times 256=65536$  fois. L'exécution s'allongerait alors de  $65536 \times 2=131072$  cycles par NOP ajouté, c'est-à-dire 0,13 s, ce qui n'est pas négligeable. Mais, par exemple, pour atteindre un délai total de 1 s, il faudrait 5 NOP ( $131072 \times 5 = 655360$  cycles supplémentaires plus 328960 cycles initiaux = 984320 cycles, soit environ 0,98 s). Mais il ne serait pas raisonnable d'aller plus loin en accumulant les NOP.

### Temporisation à 3 boucles

Si 1 s ne vous suffit pas, il faut ajouter une 3e boucle en utilisant le 3e registre du 6502 : L'accumulateur A (il n'existe pas de DEA pour décrémenter A, mais c'est facilement contournable). La temporisation maximale possible est tellement grande que l'on peut ajuster finement les valeurs respectives de A, Y et X pour obtenir un délai en chiffre rond. En voici un exemple pour obtenir un délai de 40 s. J'ai choisi cette durée afin de chronométrer la durée obtenue.

```
; Source TEMPOB.ASM de temporisation à 3 boucles
      JSR $FB2A ; Bruit touche contrôle pour démarrage du chrono
      SEI      ; Interdit les interruptions
DEBUT LDA #$C8 ; 2cy Initialise le 3e compteur avec valeur 200
BOUCL3 LDY #$9C ; 2cy Initialise le 2e compteur avec valeur 156
      LDX #$00 ; 2cy Initialise le 1e compteur avec valeur 0
BOUCL1 DEX      ; 2cy Ici X prendra 256 valeurs #FF à #00
      BNE BOUCL1 ; 3cy (2cy pour le dernier BNE)
      DEY      ; 2cy Ici Y prendra 156 valeurs de #9B à #00
      BNE BOUCL1 ; 3cy (2cy pour le dernier BNE)
      TAX      ; 2cy Ecrase X=#00 de sortie de la boucle X
      DEX      ; 2cy Décrémente le 3e compteur
      TXA      ; 2cy Positionne Z lorsque A tombe à zéro
      BNE BOUCL3 ; 3cy (2cy pour le dernier BNE)
      CLI      ; Autorise les interruptions
      JMP $FB2A ; Si A=0 bruit touche contrôle pour arrêt du chrono

FIN
```

Cette structure des boucles est nécessitée par la réinitialisation des compteurs au début de chaque boucle. A l'issue de la 1e boucle, X est automatiquement réinitialisé à zéro. Toutefois, cette remise à zéro est perdue dans le cas de l'exécution de la 3e boucle et doit être repositionnée. D'autre part, l'exécution de la 2e boucle se termine avec  $Y = \text{zéro}$ .

Si l'on veut utiliser une valeur initiale de Y différente de zéro, il faut également repositionner Y. C'est pourquoi la 3e boucle ne se termine pas par un BNE BOUCL1, mais avec un BNE BOUCL3.

En faisant abstraction de la durée réduite des BNE de fin de boucle et de 6 cycles pour les LDA, LDY et LDX initiaux, on peut faire le **calcul simplifié** suivant :

- L'exécution de la boucle X dure  $D = 256 \times 5 = 1280 \mu\text{s}$ .
- L'exécution de la boucle Y dure  $D = 156(1280+5) = 200460 \mu\text{s}$  soit environ 0,20 s
- L'exécution de la boucle A dure  $D = 200(200460+9) = 40093800 \mu\text{s}$  soit environ 40 s.

En ajustant les valeurs de Y et A il est possible d'obtenir la durée dont on a besoin. Il suffit de faire quelques calculs avec les formules ci-dessus.

Durée maximale possible avec une temporisation à 3 boucles (hors ajout de NOPs) : 84 s

- L'exécution de la boucle X dure  $D = 256 \times 5 = 1280 \mu\text{s}$ .
- L'exécution de la boucle Y dure  $D = 256(1280+5) = 328960 \mu\text{s}$  soit environ 0,32 s
- L'exécution de la boucle A dure  $D = 256(328960+9) = 84216064 \mu\text{s}$  soit environ 84 s.

## Temporisation avec le Timer 2

Vous vous rappelez la simplicité de mise en œuvre de la commande Basic WAIT ? Si on consulte "L'Oric à nu" page 186, on peut examiner ce que fait cette routine. Le paramètre qui suit WAIT (nombre de centièmes de secondes de délai) est analysé, puis placé en page zéro aux adresses #33 (LL de la valeur) et #34 (HH de la valeur). Le registre A est initialisé avec #02 (n° du timer à utiliser), puis le registre Y reçoit le contenu de la mémoire #33 et X celui de #34. La routine #EEAB écrit la valeur YX dans le timer 2, lequel est décrémenté tous les centièmes de secondes. On lit l'état du timer 2 avec la routine #EE9D et on sort quand il tombe à zéro.

Voici un exemple qui permet d'obtenir un délai de 40 s :

```
; Source TEMPOT.ASM de temporisation avec le timer 2
DEBUT JSR $FB2A ; Bruit touche contrôle pour démarrage du chrono
      LDA #$02 ; Pour initialiser le timer 2
      LDY #$A0 ; Y=LL et X=HH de valeur sur 2 octets
      LDX #$0F ; YX=#0FA0 soit 4000 centièmes de s = 40s
      JSR $EEAB ; Ecrit 4000 dans le timer 2
BOUCL JSR $EE9D ; Lit timer 2 qui décroît tous les 1/100e de s
      CPY #$00 ; Teste LL octet de poids faible du timer 2
      BNE BOUCL ; Reboucle jusqu'à ce que LL soit nul
      CPX #$00 ; Teste HH octet de poids fort du timer 2
      BNE BOUCL ; Reboucle jusqu'à ce que HH soit nul
      JMP $FB2A ; Bruit touche contrôle pour arrêt du chrono
FIN
```

### Quelques mesures

Pour vérifier les exemples, j'ai comparé ce que donnent WAIT 4000, TEMPOB et TEMPOT avec un Atmos réel, un Atmos sous Euphoric+DosBox et un Atmos sous Oricutron 1.2. Voici les résultats :

	Atmos	Euphoric	Oricutron
WAIT 4000	40s	42s	41s
TEMPOB	40s	42s	41s
TEMPOT	40s	42s	41s

L'Atmos donne les valeurs escomptées pour WAIT 4000, TEMPOB et TEMPOT. Par contre Oricutron

donne des valeurs légèrement plus élevées, mais qui restent dans la limite de la précision de mon chronométrage. Pour Euphoric, c'est encore plus marqué, mais peut-être est-ce dû à DosBox. J'ai déjà remarqué ce défaut de réactivité avec DosBox dans d'autres situations.

### Conclusion

La temporisation avec le timer 2 est plus facile à mettre en œuvre que les temporisations à boucles et devrait être utilisée préférentiellement pour les délais de 0,1 à 655 s (comme WAIT). Il n'en reste pas moins que pour les petits délais une simple boucle permet de temporiser jusqu'à 1281  $\mu\text{s}$  (et éventuellement deux boucles pour aller jusqu'à 0,33 s). Il n'y a donc que l'embarras du choix quant à la méthode à utiliser !