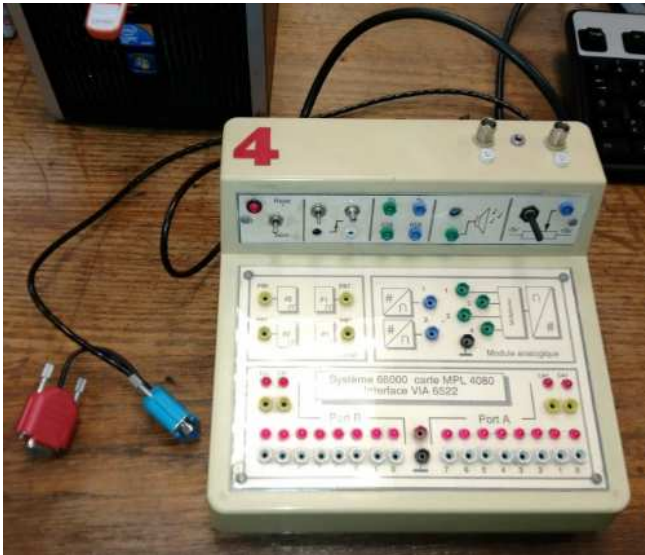
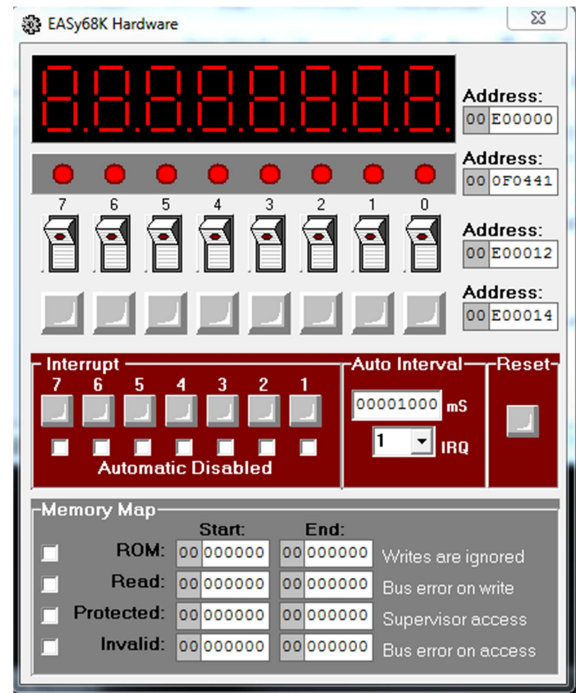


Une carte MPL 4080 peut accueillir 2 circuits VIA dont les adresses de base ainsi que les adresses de leurs registres sont définies dans le fichier equvia.X68 qu'il faut inclure dès la première ligne du programme. **'include c:\easy68K/equvia.X68**

La platine était de ce type :



Son simulateur ou émulateur dans Easy68K :



Les fichiers .X68 et un compte rendu de TP contenant les organigrammes des programmes seront récupérés à la fin de la prochaine séance.

### Exercice 1 : Utilisation d'un port et programmation modulaire.

- 1.1. Réaliser un chenillard classique (une diode allumée, un décalage à la fois) sur le port A : écrire l'organigramme, programmer et exécuter pas à pas.
- 1.2. Modifier l'organigramme pour introduire une boucle de temporisation simple pour pouvoir l'exécuter en temps réel.
- 1.3. Introduire 2 boucles de temporisation dont l'une est emboîtée dans l'autre. Ajuster les paramètres des boucles (nb de fois que les boucles sont exécutées) pour voir le chenillard se décaler toutes les 0,5 secondes environ.
- 1.4. Modifier l'organigramme pour rendre le programme modulaire de manière à le découper en 3 fonctions séparées décrites ci-dessous, puis écrire les 3 sous-programmes correspondants : « decal » : fonction de décalage ; « affich » : fonction d'affichage ; « tempo » : temporisation.

## **Exercice 2 : Utilisation de 2 ports, programmation modulaire, modification par lecture des entrées.**

- 2.1** Réaliser un chenillard sur le port A dont la séquence est : 1 4 7 2 5 0 3 6 1 ....  
Pour cela modifier le sous-programme « decal ».
- 2.2** Prévoir une fonction « lect » de lecture (sur le port B) de la vitesse de défilement.  
Cette lecture permettra de modifier la durée d'une des 2 boucles de « tempo ».
- 2.3** Prévoir une fonction « sens » pour choisir le sens de défilement.  
Pour cela restreindre la lecture de la vitesse de défilement sur les 7 bits de poids les plus faibles du port B et lire le sens sur le bit de poids le plus fort.

## **Exercice 3 : Programmation par interruption**

- 3.1.** Réaliser un programme modulaire d'un chenillard sur le port A dont la séquence est 2 diodes allumées, 2 décalage à la fois.
- 3.2** Modifier l'organigramme et le programme pour que le sens de rotation soit inversé si un front montant est détecté sur l'entrée d'interruption.
- 3.3** Modifier l'organigramme et le programme pour obtenir une temporisation plus précise en remplaçant la fonction temporisation de base par une temporisation matérielle faite par T1 = 100 ms.  
La période de défilement sera d'une seconde.

## Exemples de programme

```
Fichier equvia.X68
via1    equ    $f0441
via2    equ    $f0461
orb     equ    $0
irb     equ    $0
ora     equ    $2
ira     equ    $2
ddrb    equ    $4
ddra    equ    $6
t1lr    equ    $8
t1hr    equ    $a
t1l     equ    $c
t1h     equ    $e
t2l     equ    $10
t2h     equ    $12
shr     equ    $14
acr     equ    $16
rpc     equ    $18
ifr     equ    $1a
ier     equ    $1c
ora_h   equ    $1e
orb_h   equ    $1e

*-----
* Title      :testvia
* Written by :FB
*-----
    include c:\easy68K/equvia.X68
    ORG     $1000
START
*****initialisation du vial*****
    lea vial,a0
*****mise en sortie des ports A et B*****
    move.b #$FF,ddra(a0)
*****chargement des valeurs a afficher au debut*****
    move.b #$01,d5

debut    bsr affich    ****appel du ssprog d'affichage
         bsr tempo     ****appel du ssprog de temporisation

         rol.b #1,d5    ****decalage dun bit a gauche

         bra debut     ****boucle de retour

         move.b #9,d0
         trap #15
*****sous prog affichage sur ports*****
affich   move.b d5,ora(a0)    on envoie le contenu de d5 sur le port A
         rts
*****sous prog tempo double boucle modif*****
tempo    move.b #10,d2        debut boucle1 (valeur suivant vitesse voulue)
cycleee  move.l #6000,d1      debut boucle2 (base de temps compatible)
cycleei  sub #1,d1
         cmp.l #0,d1
         bgt cycleei
         sub.b #1,d2
         cmp.b #0,d2
         bgt cycleee
         rts
        END      START      ; last line of source*
```

```

*-----
* Program      :inter
* Written by   :DN
*-----
include c:\easy68K/equvia.X68

START      ORG      $1000
           lea vial, a0
           move.b #$00, ddra(a0)
           move.b #$ff, ddrb(a0)
           lea textel, a1

init_inter
           move.l #inter, $70
           move.b #$1, pcr(a0)          %interrup front montant CA1
           move.b #%10000010, ier(A0)  %interrup CA1 valide
           move.w #$2300, SR

           move.b #$0F, d0
           bsr affich

Cycle      move.b ira(a0), d4
           cmp #00,d4
           beq gauche

droite     ror.b #2, d0
           bra suite

gauche     rol.b #2, d0

Suite      bsr affich
           bra cycle
           bra arret_sim

textel    dc.w 'inter'

affich     move.b d0, orb(a0)
Tempo      move.l #$1FF, d2
           move.l #$FFF, d1
cycle1     move.l d1,d3
cycle2     dbf d3, cycle2
           dbf d2, cycle1
           rts

inter      bchg #0, d4
           move.w #14, d0      * pour afficher un texte
           trap #15           * affichage du texte pointé par a1

           bset #1, ifr(a0)
           rte

arret_sim  move.b #9,d0
           trap #15
           END      START

MOVE.B     #9,D0
TRAP       #15      Halt Simulator

END      START

```