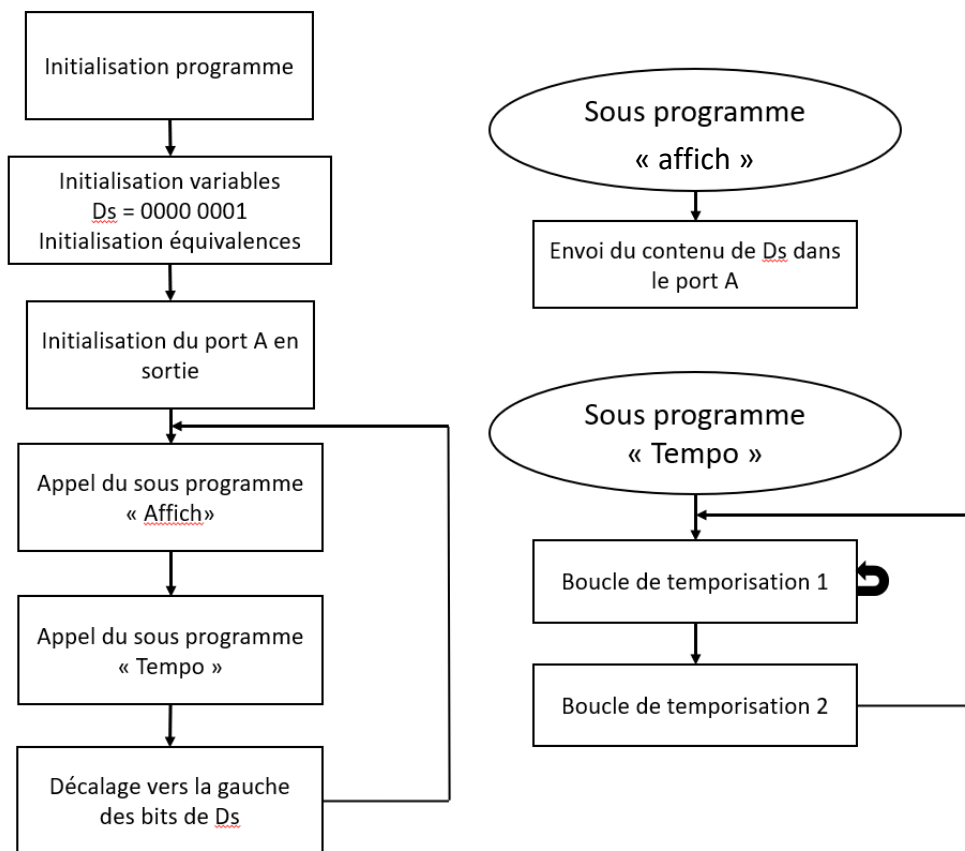


## II601 Informatique Industrielle TP3

### Exercice 1

Organigramme :

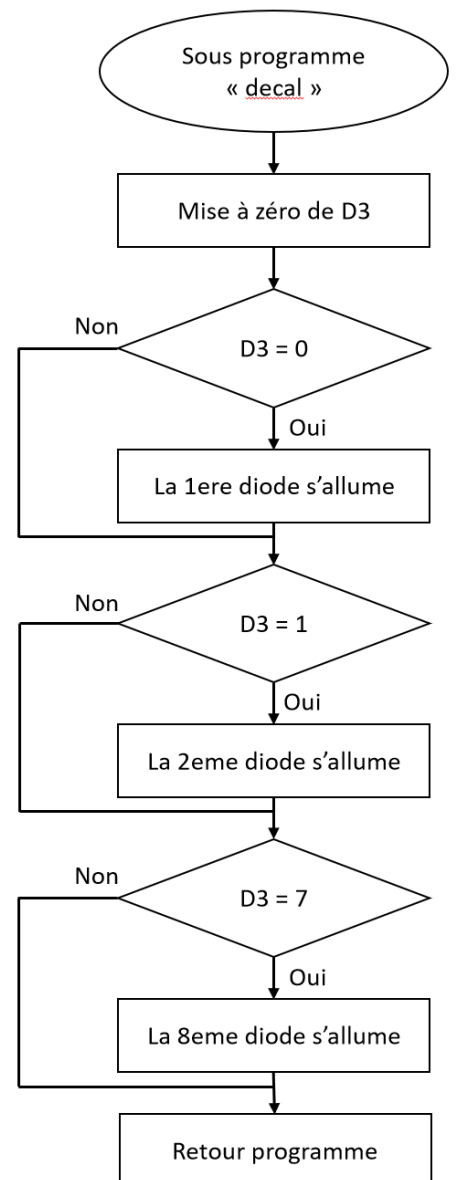
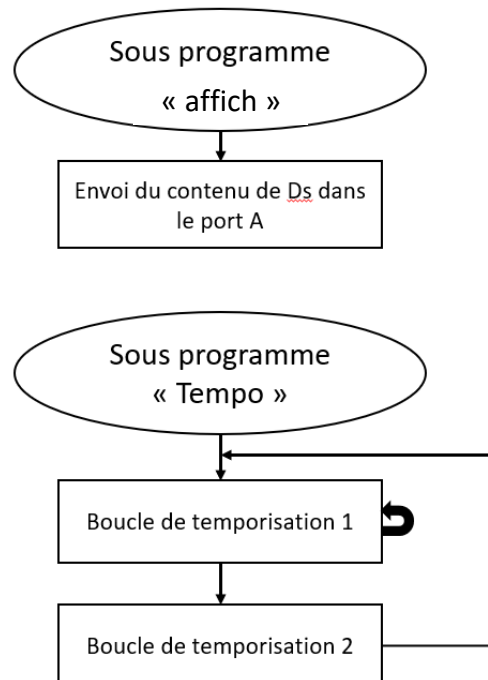
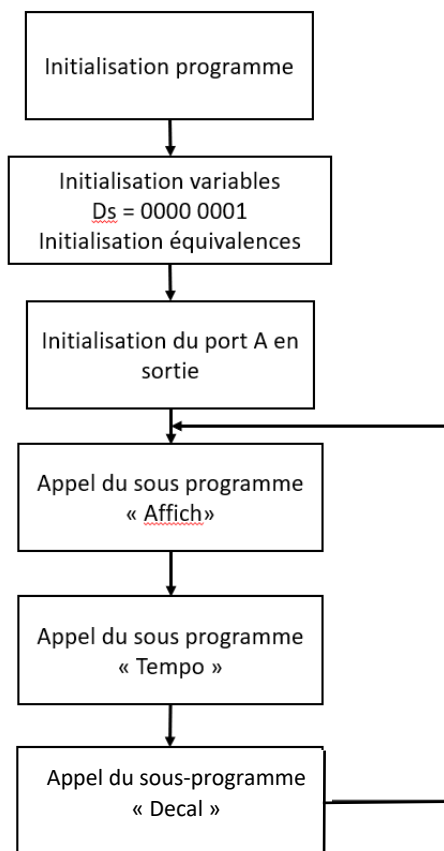


## Programme :

```
* Title      : EXERCICE1
* Written by : EL ALAMI et VILELA
* Date       : 13/03/2023
* Description: TP3
vial equ $f0441
via2 equ $f0461
orb equ $0
irb equ $0
ora equ $2
ira equ $2
ddrb equ $4
ddra equ $6
tllr equ $8
tlhr equ $a
tll equ $c
tlh 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/equivia.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 #50,d2 debut boucle1 (valeur suivant vitesse voulue) **On a
chagé le temps de 0.1 a 0.5s
cyclee move.l #6000,d1 debut boucle2 (base de temps compatible)
cyclei sub #1,d1
        cmp.l #0,d1
        bgt cyclei
        sub.b #1,d2
        cmp.b #0,d2
        bgt cyclee
        rts
END START ; last line of source*
```

Exercice 2 :

Organigramme :



Programme :

2.1 Pour obtenir la sequence 1 4 7 2 5 0 3 6, il suffit de modifier la valeur du roll on la passe de 1 à 3.

On a donc :

```
via1 equ $f0441
via2 equ $f0461
orb equ $0
irb equ $0
ora equ $2
ira equ $2
ddrb equ $4
ddra equ $6
tllr equ $8
tlhr equ $a
tll equ $c
tlh 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 $1000
START
*****initialisation du via1*****
    lea via1,a0
*****mise en sortie des ports A et B*****
    move.b #$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 #3,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)
cyclee move.l #6000,d1 debut boucle2 (base de temps compatible)
cyclei sub #1,d1
    cmp.l #0,d1
    bgt cyclei
    sub.b #1,d2
    cmp.b #0,d2
    bgt cyclee
    rts
END START ; last line of source*
```

2.2 Pour la fonction « lect » qui agira sur la vitesse de défilement, il faut commencer par initialiser les ports A en entrée et B en sortie.

On a donc :

```
*****initialisation des ports A et B*****
    move.b #$FF,ddra(a0)
    move.b #$00,ddrb(a0)
```

Puis on ajoute la fonction « lect »

```
lect move.b irb(a0),d4
    rts
```

2.3 En ce qui concerne la fonction « sens » qui agira sur le sens de défilement, il faut récupérer la valeur du port B puis la placer dans 2 registres distinct (D4 et D6).

D4 agira sur la durée du Timer alors que D6 agira sur le décalage.

On a donc :

```
lect
    move.b irb(a1),d4
    ANDI #$7F,D4
    rts

sens
    move.b irb(a1),d6
    ANDI #$80,D6
    rts

decal *La diode va s'allumer en séquence, D3 va etre incrémenté de 0 a 8
    CMP.B #$0,D6
    BEQ EGAL
    BNE PASEGAL
EGAL:
    rol.b #1,d5
    rts
PASEGAL:
    ror.b #1,d5
    rts
```

Exercice 3  
Organigramme :

