

TP Assembleur - E2I3

Léo Dubus - tp4-1.asm

```
1  CODE    $1000
2  DATA    $2000
3  PTH      $0260
4  DDRH     $0262
5  PERH     $0264
6  PPSH     $0265
7
8
9          ORG     DATA
10 VAR      DS.B   1
11
12          ORG     CODE
13
14          LDAA    #$FF    ;On charge les valeurs 255 et 0 dans les
15          LDAB    #$00    ;registres A et B
16
17          STAA    PERH    ;On active tout le port H
18          STAB    PPSH    ;On met le port H en pull up et fronts descendants
19          STAB    DDRH    ;On met tout le port H en entrée
20
21          LDAA    PTH      ;On récupère la position des switches levés
22                      ;sous un octet
23          STAA    VAR      ;On stocke cet octet dans la variable définie
24
25
26          SWI
```

```

1  CODE    $1000
2  DATA   $2000
3  PTH     $0260
4  DDRH    $0262
5  PERH    $0264
6  PPSH    $0265
7  PORTB   $0001
8  DDRB    $0003
9
10         ORG     DATA
11
12         ORG     CODE
13
14         LDAA    #$FF    ;On charge les valeurs 255 et 0 dans les
15         LDAB    #$00    ;registres A et B
16
17         STAA    PERH    ;On active tout le port H
18         STAB    PPSH    ;On met le port H en pull up et fronts descendants
19         STAB    DDRH    ;On met tout le port H en entrée
20
21         STAA    DDRB    ;On initialise les LEDs en sortie
22
23
24         LDAA    PTH     ;On récupère les switchs baissés
25         STAA    PORTB   ;Et on affiche sur les LEDs
26
27
28         SWI

```

```

1  CODE    $1000
2  DATA   $2000
3  PTH     $0260
4  DDRH    $0262
5  PERH    $0264
6  PPSH    $0265
7  PIFH    $0267
8  PORTB   $0001
9  DDRB    $0003
10
11         ORG     DATA
12  VAR     DS.B   1
13
14         ORG     CODE
15
16         LDAA    #$FF    ;On charge les valeurs 255 et 0 dans les
17         LDAB    #$00    ;registres A et B
18
19         STAB    VAR      ;On initialise au passage la variable qui va compter
20                        ;les fronts descendants
21
22         STAA    PERH     ;On active tout le port H
23         STAA    PIFH     ;On réinitialise tous les flags du port H
24         STAB    PPSH     ;On met le port H en pull up et fronts descendants
25         STAB    DDRH     ;On met tout le port H en entrée
26
27         STAA    DDRB     ;On initialise les LEDs en sortie
28
29  BOUCLE  LDAA    PIFH     ;On récupère les flags du port H
30         CMPA    #$01     ;Si le premier switch est activé, alors
31         BEQ     INCR      ;On passe à la fonction d'incrémentatation
32         BRA     BOUCLE    ;Sinon on reboucle indéfiniment.
33         SWI
34
35  INCR    INC     VAR      ;On incrémente la variable
36         STAA    PIFH     ;On réinitialise les flags
37         LDAB    VAR      ;On récupère la valeur de la variable
38         COMB     ;On l'inverse
39         STAB    PORTB    ;Pour l'affichage sur les LEDs.
40         BRA     BOUCLE    ;Et on recommence.

```

```

1  CODE    $1000
2  DATA   $2000
3  PORTB   $0001
4  DDRB    $0003
5
6          ORG     DATA
7  COMPT32 DS.B    4      ;On créé un compteur 32 bits sur 4 octets
8                          ;On pourrait également faire ça sur 2 mots binaires
9                          ;de 16 bits
10
11         ORG     CODE
12  INIT    LDX     #COMPT32;On charge le compteur 32 bits dans X
13         LDD     #0000 ;On initialise l'accumulateur D
14         STD     0,X    ;On met à zéro la totalité du compteur
15         STD     2,X    ;
16         LDAA    #0xFF ;On charge 255 dans A
17         STAA    DDRB   ;On initialise les LEDs en sortie
18         STAA    PORTB  ;On éteint les LEDs
19
20
21         SEI                      ;On arrête temporairement l'interruption pour
22                                   ;l'installer
23         LDD     #IT          ;On charge l'adresse de la fonction interruption
24                                   ;dans D
25         STD     $3E72        ;On charge l'adresse dans le vecteur d'interruption
26                                   ;sur IRQ
27
28         CLI                      ;On enlève le masque sur l'interruption
29
30
31  BOUCLE  BRA     BOUCLE ;On réalise une boucle infinie vide
32
33         SWI
34
35  IT      BSR     INCR32 ;L'interruption consiste en l'incrémentation du
36                                   ;compteur 32 bits et l'affichage du poids le plus
37                                   ;faible sur les LEDs.
38         LDAB    3,X    ;On récupère le poids le plus faible
39         COMB                    ;On l'inverse
40         STAB    PORTB  ;On l'affiche sur les LEDs
41         RTI                      ;Retour d'interruption
42
43 ;*-- Incrémentation compteur 32 bits -----
44 ;On va incrémenter du poids le plus faible vers
45 ;le poids le plus fort. Le changement se fait si
46 ;la carry est levée. Sinon on sort de la fonction.
47
48  INCR32  INC     3,X
49         BCC     FIN
50         INC     2,X
51         BCC     FIN

```

```
52      INC      1,X
53      BCC      FIN
54      INC      0,X
55  FIN      RTS      ;Retour de sous-routine
56  ;*-----
```

```

1  CODE    $1000
2  DATA   $2000
3  PTH     $0260
4  DDRH    $0262
5  PERH    $0264
6  PPSH    $0265
7  PIEH    $0266
8  PIFH    $0267
9  PORTB   $0001
10 DDRB    $0003
11
12         ORG     DATA
13 COMPT    DS.B   1
14
15         ORG     CODE
16 INIT     SEI                     ;On masque l'interruption
17                                     ;Initialisation des paramètres
18         LDD     #IT              ;On charge l'adresse de l'interruption
19         STD     $3E4C            ;Pour la stocker dans le vecteur d'interruption
20                                     ;du port H
21
22         LDAA    #$FF            ;On charge £FF
23         LDAB    #$00            ;On charge £00
24
25         STAA    PERH            ;On active tout le port H
26         STAB    PPSH            ;On met le port H en pull up et fronts descendants
27         STAB    DDRH            ;On met tout le port H en entrée
28
29         STAA    PIEH            ;On valide les interruptions sur tous les pins du
30                                     ;port H.
31
32         CLI                     ;On enlève le masque de l'interruption.
33
34         STAA    DDRB            ;On active les LEDs en sortie (debug)
35         LDAA    #$00            ;
36
37         STAA    COMPT           ;On met le compteur à 0.
38
39 BOUCLE   LDAB    COMPT           ;On récupère la valeur du compteur
40         COMB                     ;On l'inverse
41         STAB    PORTB          ;On l'affiche avec les LEDs
42         COMB                     ;On le réinverse
43         CMPB    #$05            ;Si on est à 5, on arrête (debug)
44         BNE     BOUCLE          ;On reboucle.
45
46         SWI
47
48 IT       INC     COMPT           ;On incrémente le compteur
49         LDAA    #$FF
50         STAA    PIFH            ;On remet les flags d'interruption à 0
51         BRA     TEMPO           ;On temporise

```

```

52 FIN_IT RTI ;On retourne dans le programme principal.
53
54 ;*-- TEMPORISATION -----
55 TEMPO LDAA #$FA ;Temporisation logicielle de durée grossière.
56 REC2 LDX #$03E7 ;Fait "à l'oeil"
57 REC1 DEX
58 BNE REC1
59 DECA
60 BNE REC2
61 NOP
62 BRA FIN_IT
63 ;*-----

```