

```

*****MOIRAGE*****
* Programme de moirage de la memoire
* Point
BC00 000C BC
    ORG ENDMEM=$400 1K Reserve
    DIRECT EQU *(-8) Page 0
    SETDP DIRECT
    TITLE Balayage Ecran
    INCLUD EQUATES Fichier contenant

*
* POINTS D'ENTREE DU MONITEUR T07
*
E800 INIT EQU *E800 Initialisation
E803 PUTCH EQU *E803 Affichage
E806 GETCH EQU *E806 clavier
E809 KTST EQU *E809 Test clavier
E80C DRAW EQU *E80C Ligne
E80F PLOT EQU *E80F Point
E812 RSCONT EQU *E812 RS-232
E815 K7CONT EQU *E815 Lecteur de k7
E818 GETLP EQU *E818 Crayon optique
E81B LPINT EQU *E81B Interrupteur.
E81E NOTE EQU *E81E Musique
E821 GETPT EQU *E821 Lecteur Point
E824 GETSC EQU *E824 lecteur caractere
E827 JOYSTK EQU *E827 Manche a balai
E82A DKCONT EQU *E82A Controleur disque

*
* Adresses Physiques
*
E7C3 PORTC EQU *E7C3 Le bit 0 du Port C
* controle l'acces a la memoire ecran
* 1=Points , 0=couleur
4000 STADR EQU $4000 Debut de l'ecran
5F40 ENDADR EQU $5F40 Fin de l'ecran +1

*
* CODES ASCII
*
001F US EQU $1F UNIT SEPARATOR
000C FF EQU $C FORM FEED
0004 EOT EQU 4 END OF TRANSM
001B ESC EQU $1B ESCAPE
000D CR EQU $D CARRIAGE RETURN
000A LF EQU $A LINE FEED

* les Principales adresses d'entree du
* Moniteur T07
    PAGE
BC00 34 7E START PSHS A,B,X,Y,U,DP Sauvegarde
BC02 BD E800 JSR INIT Initialisation
BC05 B6 E7C3 LDA PORTC Mise en memoire

* Points Par mise a 1 du bit 0 du Port C
    ORA #1
    STA PORTC
    LDX #STADR Adresse de debut

* de l'ecran
M2 LDY #20 Compteur colonne
* on affiche 20 fois 2 octets, soit
* 40 octets par ligne
    LDD #5555 Moirage
BC14 CC 5555

```

```

BC17 ED 01 * alternance de 1 et 0 sur la ligne
BC19 31 3F M0 STD ,X++ Charger l'ecran
BC1B 26 FA LEAY -1,Y
BC1D 100E 0014 BNE M0 Repete 20 fois
BC21 CC AAAA LDY #20 Ligne suivante
    LDD #AAAA Le motif est inver

* se Pour decaler les 1 et les 0 d'une
* ligne a l'autre et obtenir un moirage
M1 STD ,X++
    LEAY -1,Y
    BNE M1 Toujours 20 fois
    CMPX #ENDADR Fin d'ecran
    BLS M2 Sinon on recom
    * mence 2 lignes a motifs alternes
    SWI Retour au moniteur
    END START
BC2F 3F BC00

```

00000 Total Errors

PROGRAMME OBJET

COMPTEUR D'ERREURS

PROGRAMME SOURCE

```

CR 000D
DIRECT 000C
DKCONT E82A
DRAW E80C
ENDADR 5F40
EOT 0004
ESC 001B
FF 000C
GETCH E806
GETLP E818
GETPT E821
GETSC E824
INIT E800
JOYSTK E827
K7CONT E815
KTST E809
LF 000A
LPINT E81B
M0 BC17
M1 BC24
M2 BC1D
NOTE E81E
PLOT E80F
PORTC E7C3
PUTCH E803
RSCONT E812
STADR 4000
START BC00
US 001F

```

TABLE DES SYMBOLES

3.1 Options d'assemblage

/WE : Arrête l'assemblage si une erreur est détectée

Commande : Tapez A <descripteur de fichier> /WE **ENTREE**
ou A/WE **ENTREE**

Effet : À chaque erreur détectée, l'option d'assemblage /WE affiche le message correspondant à l'erreur et arrête l'assemblage en cours. À ce niveau l'utilisateur dispose de trois possibilités pour reprendre la main. En tapant :

- CNT/C, le système abandonne l'assemblage et retourne sous contrôle de l'éditeur
- C, (minuscule ou majuscule) le système continue l'assemblage mais ne s'arrêtera pas sur les erreurs potentielles dans la suite du programme.
- sur une touche quelconque le système reprend l'assemblage et s'arrêtera sur la prochaine erreur détectée.

Exemple : Écrivons une erreur dans le programme « MOIRAGE », par exemple :

```
SWI          Retour au moniteur  
N          START
```

```
Ø : MOIRAGE .ASM  > A/WE
```

ENTREE

```
BC2F 3F          SWI          Retour au moniteur  
Bad Opcode  
NED          START  
Missing END Statement  
00002 Total Errors
```

/NL : Supprime l'affichage du listing

Commande : Tapez A <descripteur de fichier> /NL **ENTREE**
ou A/NL **ENTREE**

Effet : L'option d'assemblage /NL supprime l'affichage du listing du programme.

- Restent seulement visualisés :
 - Les messages d'erreur
 - Le compteur d'erreurs
 - La table des symboles

Exemple :

```
Ø : MOIRAGE .ASM  > A/NL
```

ENTREE

```
00000 Total Errors  
CR          0000  
DIRECT      00BC  
DKCONT      E92A  
DRAW        E90C  
ENDADR      5F40  
EOT          0004  
ESC         001E  
FF          000C
```

```
PUTCH       E803  
RSCONT      E812  
STADR       4000  
START       BC00  
US          001F
```

/LP : Commutation de l'impression sur l'imprimante.

Commande : Tapez A <descripteur de fichier> /LP **ENTREE**
ou A/LP **ENTREE**

Effet : L'option d'assemblage /LP commute l'impression des listings de sortie d'assemblage sur l'imprimante.
Après l'impression, la fin de la page est éjectée et l'imprimante se positionne en haut de la page suivante.

Exemple :

Ø : MOIRAGE .ASM > A/LP

ENTREE

L'imprimante est activée.

/NO : Suppression du programme objet généré en mémoire

Commande : Tapez A/NO **ENTREE**

Effet : L'option d'assemblage /NO supprime le programme objet généré en mémoire. Si un descripteur de fichier et l'option /NO sont spécifiés en même temps dans la commande d'assemblage, l'option /NO sera ignorée, car de toutes façons le descripteur de fichier dirige l'assemblage vers un périphérique. Le programme objet sera sauvegardé dans le fichier spécifié.

Exemple :

Ø : MOIRAGE .ASM > A/NO

ENTREE

L'assemblage est effectué, mais le programme objet n'est pas implanté en mémoire. Cela sert principalement à vérifier l'absence d'erreur de syntaxe.

/NS : Suppression de la table des symboles

Commande : Tapez A <descripteur de fichier> /NS **ENTREE**
ou A/NS **ENTREE**

Effet : L'option d'assemblage /NS supprime la table des symboles dans le listing de sortie d'assemblage.

Exemple :

Ø : MOIRAGE .ASM > A/NS

ENTREE

Le listing est visualisé à l'écran sans la table des symboles.

/SS : Édition sur des lignes séparées des programmes source et objet

Commande : Tapez A <descripteur de fichier> /SS **ENTREE**
ou A/SS **ENTREE**

Effet : L'option d'assemblage /SS édite les programmes source et objet sur des lignes différentes.

Exemple :

Ø : MOIRAGE .ASM > A/SS

ENTREE

```
* mence 2 lignes a motifs alternes .
SWI                               Retour au moniteur
END      START
00000 Total Errors
```

3.2 Directives d'assemblage.

Directive EQU : EQUATE (affectation d'une valeur à un symbole)

Syntaxe : <symbole> EQU <expression> <commentaire>

But : La directive d'assemblage EQU assigne au symbole placé dans le champ étiquette la valeur de l'expression (8 ou 16 bits) placée dans le champ opérande. Ces équivalences sont consignées dans la table des symboles.

Remarques : EQU est une directive qui donne au symbole une valeur qui n'est pas liée au compteur de programme (PC).

Les symboles définis avec EQU ne peuvent pas être redéfinis dans la suite du programme.

Les symboles utilisés dans la définition d'autres symboles ou le calcul d'une expression doivent être définis au préalable.

Pour faciliter la compréhension du programme, il est pratique de placer les directives EQU au début du programme.

Exemple :

Affectation d'une valeur à un symbole.

Prog. objet	Prog. source			
	INIT	EQU	\$E800	initialisation
	US	EQU	\$1F	UNIT SEPARATOR

Affectation d'un symbole à un autre symbole.

	INIT	EQU	\$E800	initialisation
	RAZ	EQU	INIT	RAZ=INIT=E800

Affectation d'un symbole à une expression.

	INIT	EQU	\$E800	initialisation
	PUTCH	EQU	INIT+03	PUTCH=E800+03

Directive SET : Set symbol to value (affectation temporaire d'une valeur à un symbole)

Syntaxe : <symbole> SET <expression> <commentaire>

But : La directive d'assemblage SET assigne au symbole placé dans le champ étiquette la valeur de l'expression (8 ou 16 bits) placée dans le champ opérande. Ces équivalences sont consignées dans la table des symboles et repérés par le caractère S (comme SET).

Remarques : SET est une directive d'assignation semblable à EQU mais contrairement à EQU, les symboles définis avec SET peuvent être redéfinis dans la suite du programme, avec une directive SET.

Les symboles utilisés dans la définition d'autres symboles ou le calcul d'une expression doivent être définis au préalable.

La directive SET est très utile pour définir temporairement des symboles ou des étiquettes réutilisables dans la suite du programme.

Note : La table des symboles indique la dernière assignation des symboles définie par la directive SET.

Exemple : La séquence ci-dessous montre l'emploi de la directive SET.

Prog. objet	Prog. source		
	CHIFF	SET	\$4532
	CHIFF	SET	\$3678
	VAL	SET	\$04
	VAL	SET	VAL+VAL
	VAL	SET	VAL*VAL
		END	
00000 Total Errors			
	CHIFF	3678	S
	VAL	0040	S

Directive RMB : Reserve Memory Bytes (réservation d'octets en mémoire)

Syntaxe : <symbole> RMB <expression> <commentaire>

But : La directive d'assemblage RMB réserve une zone en mémoire définie par l'expression contenue dans le champ opérande. Pour cela RMB ajoute au compteur de programme (PC) la valeur de l'expression.

Remarques : RMB réserve un espace mémoire non initialisé à une valeur particulière. Les symboles utilisés dans la définition ou le calcul de l'expression du champ opérande doivent être définis avant la rencontre de la directive RMB.

Avec la directive RMB, le symbole présent dans le champ étiquette est assigné à la valeur courante du compteur de programme.

Dans le cas d'un assemblage avec un <descripteur de fichier> aucun code n'est créé dans le fichier binaire par la directive RMB. Ainsi, le fichier .BIN ne pourra pas se comparer avec l'image mémoire chargée par VERIFY dans le moniteur.

Exemple : La séquence ci-dessous montre l'emploi de la directive RMB.

Prog. objet	Prog. source
	RMB 16
	RES1 RMB 16
	RES2 RMB 16*2
	FIN EQU *
	END
00000 Total Errors	6900
FIN 6940	16 octets
RES1 6910	16 octets
RES2 6920	32 octets
	6940
	DEBUT OBJET
	RES1
	RES2
	FIN

Directive ORG : Origine (initialisation du compteur de programme)

Syntaxe : ORG <expression> <commentaire>

But : La directive d'assemblage ORG charge le compteur de programme (PC) à la valeur spécifiée par l'expression contenue dans le champ opérande. Les instructions qui suivent ORG sont assemblées en incrémentant PC. ORG définit ainsi le début du programme objet qui résulte de l'assemblage.

Remarques : Il peut y avoir plusieurs ORG dans un programme source.

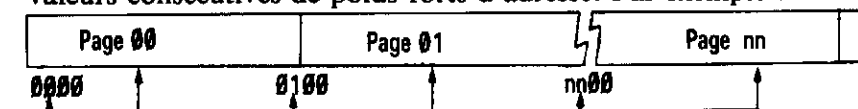
La directive ORG ne doit pas être écrite avec une étiquette.

Si ORG est omise dans le programme source, le compteur de programme est chargé à :

— 0000 si l'assemblage est demandé avec un fichier destination. Attention : un programme assemblé en 0000 ne se charge pas sur TO7 ou TO7-70 (sauf si un offset est précisé).

— la valeur de la première page qui suit la table des symboles, si l'assemblage est demandé en mémoire.

Une page de mémoire correspond aux 256 octets définis entre deux valeurs consécutives de poids forts d'adresse. Par exemple :



L'étiquette définissant la fin de la mémoire disponible (ENDMEM) peut être utilisée dans l'expression pour calculer le début du programme objet.

Exemple : Les séquences ci-dessous montrent les conséquences de la présence ou de l'absence de la directive ORG dans le programme source.

A- ORG est présent et l'expression utilise ENDMEM

0 : MOIRAGE .ASM > A

* point	ORG	ENDMEM-\$400 1K Reserve
DIRECT EQU	*-<8	Page 0
SETOP	DIRECT	
START	PSMS	A,B,X,Y,U,OP Sauvegarde
	JSR	INIT Initialisation

B- ORG est présent et l'expression contient l'adresse de début du programme objet.

Ø : MOIRAGE.ASM > A

ENTREE

```
* point
ORG      $9000
DIRECT EQU *K-8    Page 0
SETDP DIRECT

START PSHS A,B,X,Y,U,DP Sauvegarde
      JSR  INIT Initialisation
```

C- ORG est absent et la commande ne contient pas de fichier destinataire.

Ø : MOIRAGE.ASM > A

ENTREE

```
* point
DIRECT EQU *K-8    Page 0
SETDP DIRECT

START PSHS A,B,X,Y,U,DP Sauvegarde
      JSR  INIT Initialisation
```

D- ORG est absent et la commande précise le fichier destinataire.

Ø : MOIRAGE.ASM > A0 :

ENTREE

```
* point
DIRECT EQU *K-8    Page 0
SETDP DIRECT

START PSHS A,B,X,Y,U,DP Sauvegarde
      JSR  INIT Initialisation
```

Directive FCB : Form Constant Byte (définition d'une constante d'un octet)

Syntaxe : <symbole> FCB <expr>, <expr>, <expr> <commentaire>

But : La directive d'assemblage FCB stocke des constantes de 8 bits, dans des positions mémoire définies par la valeur courante du compteur de programme (PC). Pour chaque octet chargé, le PC est incrémenté de 1.

Remarques : FCB crée un espace mémoire initialisé à une valeur particulière.

FCB peut définir plusieurs constantes séparées par une virgule.

Les constantes peuvent être des valeurs numériques, des caractères, des symboles, des expressions.

Les constantes supérieures à huit bits seront tronquées à huit bits et signalées par le message « Operand Too Large » (opérande trop grand).

Les symboles utilisés dans la définition ou le calcul de l'expression contenue dans le champ opérande peuvent être définis après la rencontre de la directive FCB.

Avec la directive FCB, le symbole présent dans le champ étiquette est assigné à la valeur courante du compteur de programme.

Les constantes sont restituées sur le listing du programme objet à raison de quatre octets par ligne.

Exemple : La séquence ci-dessous montre l'emploi de la directive FCB.

Prog.objet

Prog. source

VAL	FCB	\$04,\$08,'A,\$20+\$40
VAL4	FCB	255,255
VAL5	FCB	TRE
TRE	EQU	\$34
	END	

Directive FDB : Form Double Byte Constant (définition d'une constante de deux octets ou un mot)

Syntaxe : <symbole> FDB <expr>, <expr>, <expr> <commentaire>

But : La directive d'assemblage FDB stocke des constantes de seize bits, dans des positions mémoire définies par la valeur courante du compteur de programme (PC). Pour chaque mot chargé, le PC est incrémenté de 2.

Remarques : FDB crée un espace mémoire initialisé à une valeur particulière.

FDB peut définir plusieurs constantes séparées par une virgule. Les constantes peuvent être des valeurs numériques, des caractères, des symboles, des expressions.

Les constantes supérieures à seize bits seront tronquées à seize bits. Les symboles utilisés dans la définition ou le calcul de l'expression contenue dans le champ opérande peuvent être définis après la rencontre de la directive FDB.

Avec la directive FDB, le symbole présent dans le champ étiquette est assigné à la valeur courante du compteur de programme.

Les constantes sont restituées sur le listing du programme objet à raison de deux mots par ligne.

Exemple : La séquence ci-dessous montre l'emploi de la directive FDB.

Prog.objet	Prog. source		
5A00 040B 0041	MOT	FDB	\$040B, 'A, \$2000+\$4000
5A04 6000			
5A06 FFFF 0001	MOT6	FDB	65535, 65537
5A0A 3678	MOT10	FDB	TRU
3678	TRU	EQU	\$3678
0000		END	

Directive FCC : Form Constant Character String (définition d'une constante chaîne de caractères)

Syntaxe : <symbole> FCC <déli><chaîne><déli><commentaire>

But : La directive d'assemblage FCC stocke des chaînes de caractères dans des positions mémoires définies par la valeur courante du compteur de programme (PC). À chaque caractère le PC est incrémenté de 1.

Remarques : FCC crée un espace mémoire initialisé à une valeur particulière.

FCC définit la chaîne de caractères à stocker entre deux délimiteurs identiques.

Ces délimiteurs peuvent être n'importe lequel des caractères imprimables.

L'espace utilisé dans le champ opérande provoque une tabulation automatique. Pour cette raison il est préférable d'utiliser CNT/S (espace en vidéo inversée) pour générer un espace dans une chaîne de caractères.

Avec la directive FCC, le symbole présent dans le champ étiquette est assigné à la valeur courante du compteur de programme.

Les chaînes sont restituées sur le listing du programme objet à raison de quatre caractères par ligne.

Notez que FCC /A/ est équivalent à FCB 'A

Exemple : La séquence ci-dessous montre l'emploi de la directive FCC.

Prog.objet	Prog. source		
0000 3A 32 4F 50	TG	FCC	!TROP GRAND!
0004 30 47 5E 41			
0008 1E 3A			
000C 21 3D 20 44	TP	FCC	#AU DESSOUS DE VALEUR#
0010 43 3B 38 2F			
0014 00 35 30 44			
0018 1E 30 2E 21			
001C 30 2E 21 2C			
0020 30 2E 21 2C			
0024 30 2E 21 2C			
0028 30 2E 21 2C			
002C 30 2E 21 2C			
0030 30 2E 21 2C			
0034 30 2E 21 2C			
0038 30 2E 21 2C			
003C 30 2E 21 2C			
0040 30 2E 21 2C			
0044 30 2E 21 2C			
0048 30 2E 21 2C			
004C 30 2E 21 2C			
0050 30 2E 21 2C			
0054 30 2E 21 2C			
0058 30 2E 21 2C			
005C 30 2E 21 2C			
0060 30 2E 21 2C			
0064 30 2E 21 2C			
0068 30 2E 21 2C			
006C 30 2E 21 2C			
0070 30 2E 21 2C			
0074 30 2E 21 2C			
0078 30 2E 21 2C			
007C 30 2E 21 2C			
0080 30 2E 21 2C			
0084 30 2E 21 2C			
0088 30 2E 21 2C			
008C 30 2E 21 2C			
0090 30 2E 21 2C			
0094 30 2E 21 2C			
0098 30 2E 21 2C			
009C 30 2E 21 2C			
00A0 30 2E 21 2C			
00A4 30 2E 21 2C			
00A8 30 2E 21 2C			
00AC 30 2E 21 2C			
00B0 30 2E 21 2C			
00B4 30 2E 21 2C			
00B8 30 2E 21 2C			
00BC 30 2E 21 2C			
00C0 30 2E 21 2C			
00C4 30 2E 21 2C			
00C8 30 2E 21 2C			
00CC 30 2E 21 2C			
00D0 30 2E 21 2C			
00D4 30 2E 21 2C			
00D8 30 2E 21 2C			
00DC 30 2E 21 2C			
00E0 30 2E 21 2C			
00E4 30 2E 21 2C			
00E8 30 2E 21 2C			
00EC 30 2E 21 2C			
00F0 30 2E 21 2C			
00F4 30 2E 21 2C			
00F8 30 2E 21 2C			
00FC 30 2E 21 2C			
0100 30 2E 21 2C			
0104 30 2E 21 2C			
0108 30 2E 21 2C			
010C 30 2E 21 2C			
0110 30 2E 21 2C			
0114 30 2E 21 2C			
0118 30 2E 21 2C			
011C 30 2E 21 2C			
0120 30 2E 21 2C			
0124 30 2E 21 2C			
0128 30 2E 21 2C			
012C 30 2E 21 2C			
0130 30 2E 21 2C			
0134 30 2E 21 2C			
0138 30 2E 21 2C			
013C 30 2E 21 2C			
0140 30 2E 21 2C			
0144 30 2E 21 2C			
0148 30 2E 21 2C			
014C 30 2E 21 2C			
0150 30 2E 21 2C			
0154 30 2E 21 2C			
0158 30 2E 21 2C			
015C 30 2E 21 2C			
0160 30 2E 21 2C			
0164 30 2E 21 2C			
0168 30 2E 21 2C			
016C 30 2E 21 2C			
0170 30 2E 21 2C			
0174 30 2E 21 2C			
0178 30 2E 21 2C			
017C 30 2E 21 2C			
0180 30 2E 21 2C			
0184 30 2E 21 2C			
0188 30 2E 21 2C			
018C 30 2E 21 2C			
0190 30 2E 21 2C			
0194 30 2E 21 2C			
0198 30 2E 21 2C			
019C 30 2E 21 2C			
01A0 30 2E 21 2C			
01A4 30 2E 21 2C			
01A8 30 2E 21 2C			
01AC 30 2E 21 2C			
01B0 30 2E 21 2C			
01B4 30 2E 21 2C			
01B8 30 2E 21 2C			
01BC 30 2E 21 2C			
01C0 30 2E 21 2C			
01C4 30 2E 21 2C			
01C8 30 2E 21 2C			
01CC 30 2E 21 2C			
01D0 30 2E 21 2C			
01D4 30 2E 21 2C			
01D8 30 2E 21 2C			
01DC 30 2E 21 2C			
01E0 30 2E 21 2C			
01E4 30 2E 21 2C			
01E8 30 2E 21 2C			
01EC 30 2E 21 2C			
01F0 30 2E 21 2C			
01F4 30 2E 21 2C			
01F8 30 2E 21 2C			
01FC 30 2E 21 2C			
0200 30 2E 21 2C			
0204 30 2E 21 2C			
0208 30 2E 21 2C			
020C 30 2E 21 2C			
0210 30 2E 21 2C			
0214 30 2E 21 2C			
0218 30 2E 21 2C			
021C 30 2E 21 2C			
0220 30 2E 21 2C			
0224 30 2E 21 2C			
0228 30 2E 21 2C			
022C 30 2E 21 2C			
0230 30 2E 21 2C			
0234 30 2E 21 2C			
0238 30 2E 21 2C			
023C 30 2E 21 2C			
0240 30 2E 21 2C			
0244 30 2E 21 2C			
0248 30 2E 21 2C			
024C 30 2E 21 2C			
0250 30 2E 21 2C			
0254 30 2E 21 2C			
0258 30 2E 21 2C			
025C 30 2E 21 2C			
0260 30 2E 21 2C			
0264 30 2E 21 2C			
0268 30 2E 21 2C			
026C 30 2E 21 2C			
0270 30 2E 21 2C			
0274 30 2E 21 2C			
0278 30 2E 21 2C			
027C 30 2E 21 2C			
0280 30 2E 21 2C			
0284 30 2E 21 2C			
0288 30 2E 21 2C			
028C 30 2E 21 2C			
0290 30 2E 21 2C			
0294 30 2E 21 2C			
0298 30 2E 21 2C			
029C 30 2E 21 2C			
02A0 30 2E 21 2C			
02A4 30 2E 21 2C			
02A8 30 2E 21 2C			
02AC 30 2E 21 2C			
02B0 30 2E 21 2C			
02B4 30 2E 21 2C			
02B8 30 2E 21 2C			
02BC 30 2E 21 2C			
02C0 30 2E 21 2C			
02C4 30 2E 21 2C			
02C8 30 2E 21 2C			
02CC 30 2E 21 2C			
02D0 30 2E 21 2C			
02D4 30 2E 21 2C			
02D8 30 2E 21 2C			
02DC 30 2E 21 2C			
02E0 30 2E 21 2C			
02E4 30 2E 21 2C			
02E8 30 2E 21 2C			
02EC 30 2E 21 2C			
02F0 30 2E 21 2C			
02F4 30 2E 21 2C			
02F8 30 2E 21 2C			
02FC 30 2E 21 2C			
0300 30 2E 21 2C			
0304 30 2E 21 2C			
0308 30 2E 21 2C			
030C 30 2E 21 2C			
0310 30 2E 21 2C			
0314 30 2E 21 2C			
0318 30 2E 21 2C			
031C 30 2E 21 2C			
0320 30 2E 21 2C			
0324 30 2E 21 2C			
0328 30 2E 21 2C			
032C 30 2E 21 2C			
0330 30 2E 21 2C			
0334 30 2E 21 2C			
0338 30 2E 21 2C			
033C 30 2E 21 2C			
0340 30 2E 21 2C			
0344 30 2E 21 2C			
0348 30 2E 21 2C			
034C 30 2E 21 2C			
0350 30 2E 21 2C			
0354 30 2E 21 2C			
0358 30 2E 21 2C			
035C 30 2E 21 2C			
0360 30 2E 21 2C			
0364 30 2E 21 2C			
0368 30 2E 21 2C			
036C 30 2E 21 2C			
0370 30 2E 21 2C			
0374 30 2E 21 2C			
0378 30 2E 21 2C			
037C 30 2E 21 2C			
0380 30 2E 21 2C			
0384 30 2E 21 2C			
0388 30 2E 21 2C			
038C 30 2E 21 2C			
0390 30 2E 21 2C			
0394 30 2E 21 2C			
0398 30 2E 21 2C			
039C 30 2E 21 2C			
03A0 30 2E 21 2C			
03A4 30 2E 21 2C			
03A8 30 2E 21 2C			
03AC 30 2E 21 2C			
03B0 30 2E 21 2C			
03B4 30 2E 21 2C			
03B8 30 2E 21 2C			
03BC 30 2E 21 2C			
03C0 30 2E 21 2C			
03C4 30 2E 21 2C			
03C8 30 2E 21 2C			
03CC 30 2E 21 2C			
03D0 30 2E 21 2C			
03D4 30 2E 21 2C			
03D8 30 2E 21 2C			
03DC 30 2E 21 2C			
03E0 30 2E 21 2C			
03E4 30 2E 21 2C			
03E8 30 2E 21 2C			
03EC 30 2E 21 2C			
03F0 30 2E 21 2C			
03F4 30 2E 21 2C			
03F8 30 2E 21 2C			
03FC 30 2E 21 2C			
0400 30 2E 21 2C			
0404 30 2E 21 2C			
0408 30 2E 21 2C			
040C 30 2E 21 2C			
0410 30 2E 21 2C			
0414 30 2E 21 2C			
0418 30 2E 21 2C			
041C 30 2E 21 2C			
0420 30 2E 21 2C			
0424 30 2E 21 2C			
0428 30 2E 21 2C			
042C 30 2E 21 2C			
0430 30 2E 21 2C			
0434 30 2E 21 2C			
0438 30 2E 21 2C			
043C 30 2E 21 2C			
0440 30 2E 21 2C			
0444 30 2E 21 2C			
0448 30 2E 21 2C			
044C 30 2E 21 2C			
0450 30 2E 21 2C			
0454 30 2E 21 2C			
0458 30 2E 21 2C			
045C 30 2E 21 2C			
0460 30 2E 21 2C			
0464 30 2E 21 2C			
0468 30 2E 21 2C			
046C 30 2E 21 2C			
0470 30 2E 21 2C			
0474 30 2E 21 2C			
0478 30 2E 21 2C			
047C 30 2E 21 2C			
0480 30 2E 21 2C			
0484 30 2E 21 2C			
0488 30 2E 21 2C			
048C 30 2E 21 2C			</

Directive SETDP : Set Direct Page (positionnement de la page directe)

Syntaxe : SETDP <expression> <commentaire>

But : La directive d'assemblage SETDP définit une « page 0 », dans l'espace mémoire, où l'adressage étendu sera modifié en adressage direct. Cela pour accélérer la vitesse d'exécution et diminuer la longueur du programme objet (deux ou trois octets au lieu de trois ou quatre octets).

Remarques : SETDP n'admet pas d'étiquette.

La directive SETDP ne modifie pas le registre de page 0 (DP) du micro-processeur 6809.

L'utilisateur devra prévoir son chargement en conformité avec la dernière valeur de la « page 0 ». L'omission de chargement du registre (DP) conduirait à une erreur à l'exécution du programme mais pas à l'assemblage. Le système ne peut connaître les intentions du programmeur !! La valeur du registre de page (DP) est appelée « page de base ».

A l'initialisation du système, la « page 0 » est positionnée à 00.

La directive SETDP peut être employée plusieurs fois dans une séquence de programme.

Le champ opérande d'une directive SETDP peut être occupé par : une constante, un symbole, une expression.

Les symboles utilisés dans la définition de SETDP doivent être définis en amont.

Si le calcul de l'expression de SETDP conduit à une valeur supérieure à huit bits, la « page 0 » sera chargée avec les huit bits de poids faible de l'expression et le message « Operand Too Large » (opérande trop grand) sera généré par l'assembleur.

Notez que l'utilisateur a toujours la possibilité de soustraire une instruction à l'adressage direct dans la « page 0 » en forçant l'assembleur à utiliser l'adressage étendu par (>).

Exemple : Les séquences ci-après montrent les diverses possibilités d'emploi de la directive SETDP.

Prog.objet

Prog. source

0000			ORG #0000

			***SETDP N'EST PAS ENCORE PROGRAMME ***

			"PAGE 0":00
0001 0A	03		DEC #00A3
0002 0C	44		INC #0044
	0000		DIRECT EQU *(-8)

			***SETDP UTILISE AVEC UNE CONSTANCE ***

			SETDP #00 "PAGE 0":00
0004 0A	22		DEC #0022
0005 7C	9020		INC #9020
	90		SETDP #90 "PAGE 0":90
0009 7A	0022		DEC #0022
000C 0C	20		INC #9020

			***SETDP UTILISE AVEC UNE EXPRESSION **

			SETDP #04+096 "PAGE 0":XX1A
000E 0A	56		DEC #1A56
0010 0C	70		INC #1A70

			FORCAGE DE L'ADRESSAGE ETENDU *

			DEC >#1A56
			INC >#1A70

			SETDP UTILISE AVEC UN SYMBOLE *

			SETDP DIRECT "PAGE 0":B0
0014 0A	00		DEC #B0A0
0016 0C	77		INC #B077
	0000		END
00001	Total Errors		
			DIRECT 0000

Avertissement : SETDP est inactif quand l'adressage étendu est défini par un symbole spécifié en aval.

Directive PAGE : Saut de Page

Syntaxe : PAGE

But : La directive d'assemblage PAGE fait avancer le papier de l'imprimante au début de la page suivante.

Si l'impression n'est pas demandée (par /LP) la directive PAGE est sans effet.

Note : L'insertion de lignes blanches dans le listing se fait simplement en mettant des lignes blanches dans le programme source.

Directive TITLE : Titre (définition d'un titre)

Syntaxe : TITLE <chaîne>

But : La directive d'assemblage TITLE permet de spécifier un en-tête de page par la chaîne de caractères contenue dans le champ opérande. TITLE est active avec l'option d'assemblage /LP. Elle est sans effet dans les autres cas.

Remarques : La chaîne est délimitée par la fin de la ligne et peut contenir 26 caractères. Ce nombre peut être porté à 28, si le programmeur utilise le reste du champ code opération.

La chaîne spécifiée est imprimée en haut de chaque page tant qu'une autre directive TITLE n'a pas été rencontrée.

TITLE n'admet pas de symbole dans le champ étiquette, ni de commentaire.

Exemple :

```
TITLE  CECI EST UN TITRE*****
```

```
TITLEceci est encore un titre++++
```

Directive INCLUD : Inclusion d'un programme source

Syntaxe : INCLUD <descripteur> <commentaire>

But : La directive d'assemblage INCLUD fusionne, à l'assemblage, le fichier spécifié dans le champ opérande avec le programme source résidant en mémoire. C'est l'équivalent d'une commande M (MERGE) qui ne se réalise qu'au moment de l'assemblage.

Remarques : La directive INCLUD n'est pas utilisable avec un fichier résidant sur le lecteur de cassettes, pour les raisons suivantes :

— L'assembleur multipasse nécessite plusieurs passages sur le programme source. Il serait donc nécessaire de repositionner le lecteur plusieurs fois.

— Avec un lecteur de cassettes, il n'est pas possible, en même temps, de lire le programme source et de sauvegarder le programme objet.

INCLUD n'admet pas d'inclusion imbriquée : le programme à inclure ne doit donc pas contenir la directive INCLUD.

Si INCLUD est rencontrée pendant l'inclusion, le message d'erreur « BAD INCLUD » (erreur d'inclusion) est généré. Il en est de même si une tentative d'inclusion est faite à partir du LEP.

Exemple : La séquence ci-dessous montre l'emploi de la directive INCLUD.

Prog. objet

Prog. source

	TITLE Balayage Ecran			
	INCLUD EQUATES Fichier contenant			
	* * POINTS D'ENTREE DU MONITEUR T07 *			
0000	INIT	EQU	#E000	Initialisation
0010	ESC	EQU	#1B	ESCAPE
0000	CR	EQU	#D	CARRIAGE RETURN
	* les Principales adresses d'entree du			
	* Moniteur T07			
	PAGE			

Directive END : Fin du programme source

Syntaxe : **END** <expression> <commentaire>

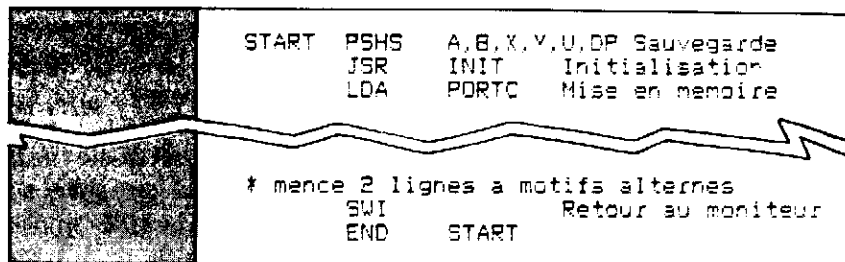
But : La directive d'assemblage **END** marque la limite du programme source. Les lignes de programmes situées derrière **END** seront ignorées par l'assembleur.

Remarques : Si la directive d'assemblage **END** est omise à la fin d'un programme, le système générera le message d'erreur : « Missing **END** statement » (la directive **END** est manquante).

END n'admet pas de symbole dans le champ étiquette.

L'expression facultative contenue dans le champ opérande précise l'adresse d'exécution du programme. Cette valeur sera utilisée pour spécifier l'adresse d'exécution, dans le fichier, à la création du programme objet.

Exemple : la séquence ci-dessous montre l'emploi de la directive **END**.

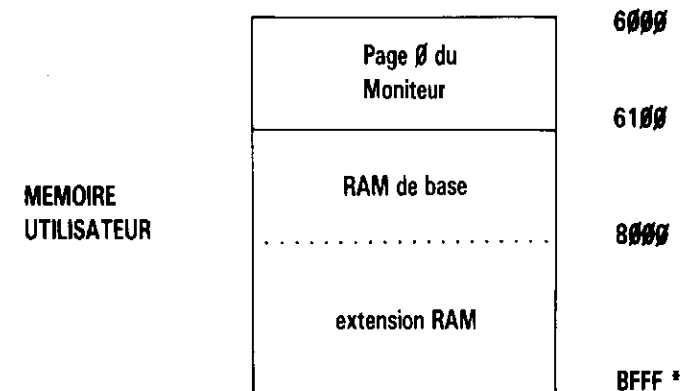


Où est situé le programme objet ?

Quelle est la fin du programme source ?

Voilà les questions que se posent bien des programmeurs, au début de l'utilisation d'un nouvel assembleur.

Avant de répondre à ces questions, il convient de rappeler l'emplacement de la mémoire utilisateur dans le champ adressable.



C'est dans cette zone de mémoire que doivent être logés :

- Le programme source.
- La table des symboles.
- Le programme objet.

La position du programme source est choisie par le système.

Il se situera au début de l'espace utilisateur.

Le programme source sera suivi de la table des symboles.

Le reste de la mémoire, que nous appellerons « mémoire libre », pourra être occupé par le programme objet.

La position du programme objet en « mémoire libre » est définie par la directive d'assemblage **ORG** (origine).

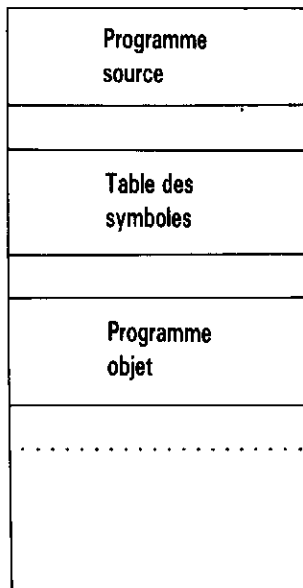
Sans directive d'assemblage **ORG** dans le programme source, le programme objet sera résidant à partir de la première page suivant la table des symboles.

Une étiquette spéciale **ENDMEM** définit la fin de la mémoire RAM.

* Le TO7-70 possède une RAM utilisateur continue de \$6000 à \$DFFF (voir plan ci-après).

ENDMEM
(version de base)
(T07)

ENDMEM
(avec extension)



BFFF
(DFFF sur T07-70)

ENDMEM peut être utilisée pour définir une position dans la mémoire.

Exemple : ORG ENDMEM-#3000

... est acceptable par l'assembleur

Note : « * » ou « . » sont des symboles spéciaux qui représentent la valeur courante du compteur de programme. Ces symboles peuvent être utilisés dans le calcul d'expressions.

Exemple : DIRECT EQV */256 (division entière)

Une tentative d'implantation du programme objet (par ORG) dans la zone occupée par le programme source n'est pas autorisée et génère le message : « Bad memory » (mauvaise mémoire).

Il n'est donc pas possible d'écraser le programme source avec le programme objet.

Un programme source peut être suffisamment long pour ne pas laisser assez de place en mémoire pour l'implantation du programme objet.

Devant ce problème deux solutions sont envisageables :

1 - Demander l'assemblage du programme objet sur un fichier.

Dans ce cas toute la mémoire disponible est utilisée par le programme source.

2 - Sauvegarder le programme source sur un fichier et l'appeler seulement dans la phase d'assemblage par directive **INCLUDE**.

Par exemple :

```
DEBUT  ORG      *
        INCLUDE PROGSOU  (fichier source)
END     DEBUT
```

Dans ce cas toute la mémoire (ou presque) est disponible pour implanter le programme objet.

Les commandes ci-dessous exposent, l'implantation du programme objet en fonction des paramètres d'entrée.

La directive d'assemblage ORG est présente dans le programme source.

Commande

Demande d'assemblage avec un descripteur de fichier.

> A <descripteur>

Résultat

Le programme objet est implanté sur le périphérique assigné. ORG spécifie l'adresse d'origine du programme objet.

Commande

Demande d'assemblage sans descripteur de fichier. Un fichier est présent dans la partie gauche.

<descripteur> > AC

Résultat

Le programme objet est implanté sur le périphérique du fichier en cours. ORG spécifie l'adresse d'origine du programme objet.

Commande

Demande d'assemblage sans descripteur de fichier.

> A

Résultat

Le programme objet est implanté en mémoire à l'adresse spécifiée par ORG.

La directive d'assemblage ORG est absente dans le programme source.

Commande

Demande d'assemblage avec un descripteur de fichier.

> A <descripteur>

Résultat

Le programme objet est implanté sur le périphérique assigné avec l'adresse 0000 comme origine.*

Commande

Demande d'assemblage sans descripteur de fichier. Un fichier est présent dans la partie gauche.

<descripteur> > AC

Résultat

Le programme objet est implanté sur le périphérique assigné et portera le nom du fichier en cours avec l'adresse 0000 comme origine.*

Commande

Demande d'assemblage sans descripteur de fichier.

> A

Résultat

Le programme objet est implanté en mémoire à partir de la première page qui suit la table des symboles.

Note : Pour que les programmes * puissent être exécutés, il faut qu'ils soient relogeables sur TO7 ou TO7-70.

CHAPITRE 4 LE MONITEUR

Le moniteur est une partie du programme résidant dans la mémoire de la cartouche ROM.

Le moniteur est à la fois un outil de vérification et de modification du contenu des mémoires et des registres internes du microprocesseur 6809, et un outil d'exécution et de mise au point des programmes objet issus de l'assembleur.

Le moniteur offre la possibilité d'exécuter en totalité ou partiellement un programme, de connaître les valeurs intermédiaires des registres du 6809 et le contenu des mémoires pour le modifier éventuellement.

Le moniteur utilise l'écran dans sa totalité et affiche en permanence, sur la ligne 0, les noms des registres du microprocesseur.

PC	A	B	DP	CC	X	Y	U	S
#								

La ligne 25 (rouge) est destinée à recevoir les commentaires et messages d'erreur du moniteur.

Exemple :

Cassette Ready Y/N?

(cassette prête O/N ?)

Les entrées et sorties du moniteur apparaissent à l'écran :

- Les entrées en bleu sur fond noir
- Les sorties en jaune sur fond noir (bleu tramé dans le manuel)

L'état « Attente de commande moniteur » est matérialisé par « # » à gauche du curseur.

Accès au moniteur

Pour accéder au moniteur, l'utilisateur dispose de deux entrées :

- A partir de MENU.
- A partir de l'ÉDITEUR.

À partir de MENU :

La commande de MENU **2** active le bloc moniteur.

Sans descripteur de fichier :

2 MONITOR **ENTREE**

L'écran du moniteur est généré et positionné dans l'état d'attente commande moniteur

PC	A	B	DP	CC	X	Y	U	S
#								

Avec descripteur de fichier :

2 MONITOR **ENTREE**

ou **2** MONITOR **ENTREE**

ou **2** MONITOR **ENTREE**

ou **2** MONITOR **ENTREE**

Note : Ø : C : et .BIN sont facultatifs et sont pris par défaut.

L'écran du moniteur est généré comme ci-dessus.

Le fichier objet spécifié dans la commande est chargé en mémoire aux adresses qui ont été définies au moment de la sauvegarde du programme objet.

À partir de l'ÉDITEUR :

La commande **X** en mode COMMANDE ÉDITEUR permet l'accès au moniteur (avec chargement éventuel d'un fichier objet).

> X <fichier>

ENTREE

L'écran du moniteur est généré comme ci-dessus. Le passage de l'éditeur au moniteur ne modifie pas le contenu de la mémoire. Seuls les fichiers objet peuvent être chargés par le moniteur. Un appel de fichier source sous moniteur provoque l'erreur : « File format error » (erreur de format de fichier).

4.1 Commandes moniteur

Les commandes moniteur sont des commandes actives à partir du clavier, le moniteur étant en état « attente de commande ».

INPUT : entrée

Commande : Le moniteur en état « attente de commande » « # »

Tapez : ID **ENTREE**

ou : IH **ENTREE**

Effet : La commande $\{H \text{ ou } D\}$ (INPUT) déclare le mode numérique d'entrée dans le moniteur.

Les codes admis sont :

DÉCIMAL : D

HEXADÉCIMAL : H

Le code non déclaré devra être précisé dans la commande par son préfixe ou son suffixe.

Une nouvelle déclaration annule la déclaration précédente.

Si la déclaration d'entrée est omise, le MONITEUR interprétera les entrées en hexadécimal par défaut.

Les valeurs d'entrée sont visualisées en bleu sur l'écran.

Exemple :

#ID

ENTREE (code courant d'entrée : décimal)

#8000 (8000 en decimal)

#8000H ou #8000 (8000 en hexadecimal)

#IH

ENTREE (code courant d'entrée : hexadécimal)

#8000 (8000 en hexadecimal)

#18000 ou #8000T (8000 en decimal)

(ADRESSE)/ : Affichage et/ou modification du contenu d'une adresse mémoire ou des registres internes du microprocesseur 6809

Commande : Le moniteur en état « attente de commande » « # »

Tapez : <expression> / <contenu> <expression> **1** ou **2** ou **ENTREE**

ou : = <contenu> <expression> **1** ou **2** ou **ENTREE**

Effet : La commande (ADRESSE)/ permet d'afficher et/ou de modifier le contenu de la case mémoire dont l'adresse est spécifiée dans la commande.

Si l'adresse est omise, le moniteur pointera sur la dernière adresse spécifiée.

Si aucune adresse n'a été spécifiée auparavant, le moniteur pointera sur l'adresse 0000.

L'adresse peut être définie par une étiquette contenue dans la table des symboles issue du dernier assemblage, ou par les symboles représentant les registres internes du 6809 (PC, A, B, D, DP, CC, X, Y, U, S).

Le nombre d'octets examinés à partir de l'adresse pointée dépend du mode de sortie.

En mode mnémonique, le nombre d'octets examinés peut être de un à cinq. En mode numérique le nombre d'octets dépend de la valeur du paramètre de la commande N (voir plus loin). Le caractère « = » seul visualise le contenu de la dernière adresse pointée.

Après le caractère « / » qui indique la fin de l'adresse, le moniteur affiche le contenu de la mémoire ou du registre.

Remarquez qu'un 0 précède tous les nombres hexadécimaux commençant par un caractère alphabétique, ceci pour éviter la confusion entre les nombres hexadécimaux et les registres du microprocesseur.

Après l'affichage du contenu de la mémoire, le système attend l'entrée d'une nouvelle valeur destinée à remplacer la précédente.

Cette entrée est facultative.

La touche **1** permet d'examiner les octets suivants sans revenir au point « attente de commande ».

La touche **2** permet d'examiner les octets précédents, mais par décrétement de un octet uniquement.

La touche **ENTREE** termine la commande et retourne au point « attente de commande ».

Exemple :

ENTREE #8C00/ PSHS U,Y,X,DP,B,A -