
FORTH 83-STANDARD
V2.4
pour ORIC TELESTRAT
T. BESTEL

(additif au mini-manuel utilisateur ATMOS F83 V2.0 de M. ZUPAN)

MODIFICATIONS DE LA V2.0 A LA V2.4 - VERSION KERNEL- VOCABULAIRE FORTH :

Mots supprimés : FLD, CAP-IN, RAM/ROM.

Mots nouveaux : BIOS, OPCH, CLCH, (KBD), 2!, 2@, 2CONSTANT, 2VARIABLE, BNK, BANK, STATLINE, CLEAR, (COPY), COPY, (.VERSION), .VERSION, OK, PROMPT.

- VOCABULAIRE DOS (V2.4) - SEDORIC (V2.0) :

Mots supprimés : FCB!, (SAVE-FILE).

Mots nouveaux : BDOS, XSAVE, XLOAD, SEARCH-, DRV, .DRV, .NAME.

Mots équivalents :

?SED (V2.0) = ?DOS (V2.4)
 SED-FILE (V2.0) = DOS-FILE (V2.4)

N.B. : ?RANGE se trouve maintenant dans le vocabulaire FORTH.

Tous ces mots sont décompilés et commentés en annexe.

- AUTRES MODIFICATIONS :

- U*D est désormais corrigé.
- Tous les messages sont en français et sont directement inspirés du FIG-FORTH TELE-FORTH V1.1 (à noter le message (redefini), (isn't unique en V2.0) qui est suivi d'un CR pour une meilleure lisibilité des mots redéfinis).

- Les attributs par défaut (encre cyan, papier noir) sont supprimés à l'initialisation. De ce fait, l'utilisateur peut profiter pleinement des 40 colonnes du terminal.
- Pour les mêmes raisons, les sigles de la ligne de STATUS ont été supprimés (CAPS, H, D, P).
- La pile passe de 61 à 22 cellules, ceci pour rendre la version compatible avec la télématique (FORTH n'empiète plus sur les zones systèmes en page zéro). De toutes façons, il devient toujours lourd de manipuler plus de 4 ou 5 valeurs à la fois sur la pile...
- Le prompt est identique à celui de TELE-FORTH : il est vectorisé. De plus, il indique la base de travail et le nombre de valeurs dans la pile.

Exemple :

Ok.2 indique une base décimale (.), ainsi que 2 valeurs dans la pile.

Ok%11 indique une base binaire (%), ainsi que 3 valeurs (11 en binaire) dans la pile.

Ok#A indique une base hexadécimale (#), ainsi que 10 valeurs (A en hexadécimal) dans la pile.

Ok*J : * indique une autre base (50, 60...).

Thierry BESTEL, le 17 Août 1989.

P.S. : Vous disposez du vrai standard 83 : les 100 mots standards y figurent, mais la version minimale (F83KER.COM) vous en propose déjà plus de 450!

- ANNEXE : DECOMPILATION ET COMMENTAIRE SUR LES NOUVEAUX MOTS

CC CONSTANT AYX

CF CONSTANT PIO

CODE BIOS (fun --)

LL LDA, BRK 1+ STA, AYX LDA, AYX 1+ LDY, X!, AYX 2+ LDX,
 LABEL BRK fun BRK, AYX STA, AYX 1+ STY, AYX 2+ STX, PHP, PLA,
 PIO STA, X@, POP, ;C

BIOS prend le n° de fonction sur la pile et prend les valeurs des registres dans la variable AYX (adresse CC hexa, registre accumulateur en CC, registre Y en CD, registre X en CE) et l'indicateur de retenue C dans le bit0 de la variable PIO (adresse CF); après exécution de BIOS, les valeurs des registres sont retournées aux mêmes adresses, et l'indicateur d'état est retourné à l'adresse PIO (bit7à0 : NV-BDIZC), pour exemples, voir les différentes interfaces avec le moniteur (OPCH, CLCH, (KBD)).

: OPCH (ch io --)

080 OR AYX C! 3 AND BIOS. ;

OPCH ouvre sur le canal ch le driver de l'entrée/sortie io.

: CLCH (ch io --)

080 OR AYX C! 3 AND 4 + BIOS ;

CLCH ferme sur le canal ch le driver de l'entrée/sortie io.

: (KBD) (n --)

AYX C! XGOKBD BIOS ;

(KBD) permet de changer le type de clavier.

n = 0 QWERTY \

n = 1 AZERTY > existent dans le vocabulaire FORTH

n = 4 ACCENT /

n = 2 FRENCH

: 2! (d adr --)

DUP -ROT ! 2+ ! ;

2! écrit à adr le double-mot d (32 bits); le mot de poids fort, au sommet de la pile, est stocké à adr, le mot de poids faible à adr+2.

: 2@ (adr -- d)

DUP @ SWAP 2+ @ SWAP ;

2@ lit le double-mot d (32 bits) stocké à adr (voir 2!).

: 2CONSTANT (d -- | -- d)

CREATE , , DOES> 2@ ;

2CONSTANT crée une constante-double (32 bits) de valeur d; une constante-double retourne sa valeur sur la pile.

: 2VARIABLE (-- | -- adr)

0 0 2CONSTANT DOES> ;

2VARIABLE crée une variable-double (32 bits) initialisée à 0.; une variable-double retourne sur la pile l'adresse à laquelle se trouve sa valeur (pour 2! et 2@ par exemple).

USER BNK (-- ad)

BNK retourne l'adresse à laquelle se trouve la valeur de la banque courante. Ex. : BNK @ . 6 Ok.0.

CODE BANK (n --)

PHP, SEI, 1L LDA, 7 AND, 2A LDY, UP)Y STA, V2DRA LDA,
F8 AND, UP)Y ORA, V2DRA STA, PLP, POP, ;C

BANK permet de passer sur la banque n.

BB80 CONSTANT STATLINE

STATLINE retourne l'adresse de début d'écran du terminal.

```
: CLEAR ( scr -- )
    DUP SCR ! BUFFER B/BUF BLANK UPDATE ;
CLEAR efface l'écran scr.
```

```
: (COPY) ( srce dest -- )
    SWAP BLOCK DROP 1 BUFFER# ! UPDATE ;
(COPY) copie l'écran srce vers l'écran dest.
```

```
DEFER COPY ( srce dest -- )
    ' (COPY) IS COPY
Le vecteur standard de COPY est (COPY).
```

```
: (.VERSION) ( -- )
    CR VERSION TYPE ;
(.VERSION) affiche la version en cours.
```

```
DEFER .VERSION
    ' (.VERSION) IS .VERSION
Le vecteur standard de .VERSION est (.VERSION).
```

```
: OK ( -- )
    BASE @ DUP 0A =
    IF ASCII .
    ELSE DUP 10 =
        IF ASCII #
        ELSE DUP 2 =
            IF ASCII %
            ELSE ASCII *
            THEN
        THEN
    THEN EMIT DROP DEPTH . ;
```

OK est le message d'invite initial qui affiche la base utilisée et le nombre de valeurs dans la pile.

```
DEFER PROMPT
    ' OK IS PROMPT
PROMPT a pour vecteur standard OK; c'est le message d'invite utilisé par QUIT et est redéfinissable par l'utilisateur.
```

CODE BDOS (fun -- f)

```

X!, ERRNB STY, PHP, 1L LDA, fun 1+ STA, TSX, DEX, DEX,
SAVES STX, SEI, V2DRA LDA, 7 AND, N 1- STA, V2DRA LDA,
F8 AND; V2DRA STA, PLP, PHP, AYX LDA, AYX 1+ LDY, AYX 2+ LDX,
LABEL fun XTRVNM JSR, AYX STA, AYX 1+ STY, AYX 2+ STX, PHP, PLA,
PIO STA, SEI, V2DRA LDA, N 1- ORA, V2DRA STA, PLP, X@,
ERRNB LDA, 1L STA, NEXT, ;C

```

BDOS est la primitive d'appel dans la banque mémoire du DOS de l'adresse de la fonction à exécuter, dont la forme est FFxx hexa, avec xx=fun (Y est toujours nul à l'entrée d'une primitive); un code différent de zéro est retourné dans ERRNB s'il y a eu erreur; comme pour BIOS, les registres sont passés et retournés à travers les variables AYX et PIO.

: XSAVE (org fin ftype --)

```

FTYPE C! FISALO ! DESALO ! C0 VSALO0 ! <XSAVE BDOS DISK-ERR?

```

```

;

```

XSAVE sauve (SAVEU) le contenu de la mémoire des adresses org à fin (incluses), dans un fichier de type ftype dont le nom doit se trouver dans BUFNOM (après utilisation de NAME! par exemple).

: XLOAD (vsalo0 --)

```

VSALO0 ! <XLOAD BDOS DISK-ERR? ;

```

XLOAD charge le contenu d'un fichier en mémoire; le nom du fichier doit être dans BUFNOM, comme pour XSAVE.

```

vsalo0 = 00 : rien de spécial

```

```

vsalo0 = 40 : afficher status

```

```

vsalo0 = 80 : forcer ne pas exécuter

```

: SEARCH- (-- f)

```

<XTRVNX BDOS DISK-ERR? PIO C@ 2 AND 0= ;

```

SEARCH- recherche, comme SEARCH, un fichier sur disque, mais à partir de la position courante.

: DRV (drv --)

```
3 AND DUP TABDRV + C@ 0= ABORT~ lecteur absent~ DUP DRIVE
C! DRVDEF C! ;
```

DRV change le lecteur par défaut; si le lecteur demandé n'est pas connecté, un message "lecteur absent" est généré.

drv = 0 : lecteur A

drv = 1 : lecteur B

drv = 2 : lecteur C

drv = 3 : lecteur D

: .DRV (--)

```
BNK @ DRIVE C@ ASCII A + EMIT ." -: ~ 0 BANK C202 ?
." secteurs libres~ BANK ;
```

.DRV affiche le lecteur courant ainsi que le nombre de secteurs libres. Ex. : .DRV A-: 210 secteurs libres Ok.0

: .NAME (--)

```
BNK @ 0 BANK OR C400 POSNMX C@ + DUP 9 TYPE ." ." DUP 9 +
3 TYPE 0E + C@ 1- 4 / 4 .R ." .K~ BANK ;
```

.NAME affiche le nom du fichier courant, son extension ainsi que sa longueur en Ko. Ex. : .NAME F83KER .COM 13.K Ok.0