

FAST FORTH V2.0 EN RÉSUMÉ

Les mots entre accolades {} sont des mots **MARKER** qui servent de balise à la fonction ramasse-miettes de FastForth.

FORTH vocabulary

Les mots avec hyperlien sont conformes aux mots du standard ANS94. Les autres sont détaillés en fin de page.

COLD	WARM	WIPE	RST_HERE	PWR_HERE	RST_STATE	PWR_STATE	MOVE
LEAVE	+LOOP	LOOP	DO	REPEAT	WHILE	AGAIN	UNTIL
BEGIN	THEN	ELSE	IF	>BODY	DEFER	DOES>	CREATE
CONSTANT	VARIABLE	↑	↑	POSTPONE	RECURSE	IMMEDIATE	IS
[']	↓	↓	↓	EXECUTE	ABORT'	ABORT	QUIT
EVALUATE	COUNT	LITERAL	↓	SPACE	>NUMBER	FIND	WORD
"	S"	TYPE	SPACES	ALLOT	CR	NOECHO	ECHO
EMIT	ACCEPT	KEY	C,		HERE	.	D.
U.							
SIGN	HOLD	#>	#S	#	UM/MOD	<#	BL
STATE	BASE	CIB	J	I	UNLOOP	U<	>
<	=	0>	0<	0=	DABS	1-	1+
ABS	NEGATE	-	+	C!	C@	!	@
DEPTH	R@	R>	>R	ROT	OVER	SWAP	NIP
DROP	?DROP	DUP	LIT	EXIT			

Les mots KEY, EMIT, CR, WARM sont des mots DEFERÉS primaires, ils sont immédiatement suivis par leur code exécuté par défaut, c'est-à-dire avant toute redirection effectuée par IS.

exemple, après cette entrée :

```
' DROP IS KEY
  KEY                               exécute DROP (exécute l'indirection faite par IS),
  ' KEY >BODY EXECUTE               exécute KEY (exécute l'action par défaut).
```

Et pour revenir à l'action par défaut:

```
' KEY >BODY IS KEY
```

with the START and STOP words in MSP430-FORTH\RC5toLCD.f
you see the assembly version with the primary DEFERred word SLEEP.

COLD	Software reset
WARM	mot DEFERÉ primaire, affiche le message de départ et lance l'interpréteur FORTH.
WIPE	remet FASTFORTH dans son état initial.
RST_HERE	définit la limite du programme protégé contre COLD ou un reset hardware.
PWR_HERE	définit la limite du programme protégé contre ON/OFF et la survenue d'une erreur.
RST_STATE	Supprime tous les ajouts de mots effectués après la commande RST_HERE
PWR_STATE	Supprime tous les ajouts de mots effectués après la commande PWR_HERE
NOECHO	Supprime l'affichage
ECHO	Remet en service l'affichage
CIB -- addr	fournit l'adresse de départ du Current Input Buffer
LIT	Ce mot est la partie exécution du mot LITERAL

ASSEMBLER vocabulary

?GOTO	GOTO	FW3	FW2	FW1	BW3	BW2	BW1
?JMP	JMP	REPEAT	WHILE	AGAIN	UNTIL	ELSE	THEN
IF	0=	0<	U>=	U<	0<	0>=	S<
S>=	RRUM	RLAM	RRAM	RRCM	POPM	PUSHM	CALL
PUSH.B	PUSH	SXT	RRA.B	RRA	SWPB	RRC.B	RRC
AND.B	AND	XOR.B	XOR	BIS.B	BIS	BIC.B	BIC
BIT.B	BIT	DADD.B	DADD	CMP.B	CMP	SUB.B	SUB
SUBC.B	SUBC	ADDC.B	ADDC	ADD.B	ADD	MOV.B	MOV
RETI	LO2HI	COLON	ENDASM	ENDCODE	SLEEP		

ASM **CODE** HI2LO <-- ajoutés au vocabulaire forth

ASM <word> Crée un mot ASSEMBLEUR comme le mot CODE sauf qu'il ne peut pas être interprété par FORTH (le retour se fait par RET). Un tel mot doit se finir par la balise ENDASM.

CODE <word> Crée un mot FORTH prêt à être écrit en assembleur. Ce mot ainsi défini doit être terminé avec la balise ENDCODE sauf si utilisation de COLON ou LO2HI.

DFRCODE <word> Crée un mot FORTH DEFERÉ dont la partie exécution par défaut est prête à être écrite en assembleur. Ce mot ainsi défini doit être terminé avec ENDCODE sauf si utilisation de COLON ou LO2HI.

HI2LO dans la définition d'un mot, sert à passer du FORTH à l'assembleur. La définition doit alors être terminée avec ENDCODE

?GOTO GOTO Après un test (0=,0<>,U>=,U<,0<,S<,S>=) réalise un branchement conditionnel au label FWx ou BWx
branchement inconditionnel au label FWx ou BWx

FW3 Label de branchement en avant n°3
FW2 Label de branchement en avant n°2
FW1 Label de branchement en avant n°1

BW3 Label de branchement en arrière n°3
BW2 Label de branchement en arrière n°2
BW1 Label de branchement en arrière n°1

?JMP JMP Après un test ((0=,0<>,U>=,U<,0<,S<,S>=) réalise un branchement conditionnel à un mot existant
branche inconditionnellement à un mot existant

REPEAT Version assembleur du mot FORTH REPEAT (Fin d'une structure boucle BEGIN ... WHILE ... REPAT)
WHILE Version assembleur du mot FORTH WHILE (branchement conditionnel après test 0=,0<>,U>=,U<,0>=,S<,S>=)
AGAIN Version assembleur du mot FORTH AGAIN (Fin d'une structure boucle BEGIN ... AGAIN)
UNTIL Version assembleur du mot FORTH UNTIL (branchement conditionnel après test 0=,0<>,U>=,U<,0>=,S<,S>=)
ELSE Version assembleur du mot FORTH ELSE (branchement inconditionnel)
THEN Version assembleur du mot FORTH THEN (fin d'une structure IF ... THEN ou IF ... ELSE ... THEN)
IF Version assembleur du mot FORTH IF (branchement conditionnel après test 0=,0<>,U>=,U<,0>=,S<,S>=)

LO2HI dans une définition sert à passer de l'assembleur au FORTH, sans sauvegarder IP. La définition doit alors se terminer par ;

COLON sauvegarde IP puis réalise LO2HI, exemple: CODE <word> ... assembly code ... COLON ... FORTH words ... ;
 ENDASM finit la définition d'un mot ASM
 ENDCODE finit la définition d'un mot CODE
 SLEEP mot DEFERÉ qui permet de créer une tâche de fond ; la tâche de fond par défaut: MOV #GIE+LPM0,SR

Le fichier \MSP430-FORTH\ANS_COMP.f est un bon exemple d'utilisation de l'assembleur, et le fichier \MSP430-FORTH\RC5toLCD.f est un bon modèle de programme multitâches temps réel faisant appel aux interruptions.

CONDCOMP

[DEFINED] [UNDEFINED] [IF] [ELSE] [THEN] COMPARE MARKER

VOCABULARY

DEFINITIONS ONLY PREVIOUS ALSO ASSEMBLER FORTH VOCABULARY

FORTH remplace le premier vocabulaire dans CONTEXT par le vocabulaire FORTH
 ASSEMBLER remplace le premier vocabulaire dans CONTEXT par le vocabulaire ASSEMBLER
 VOCABULARY VOCABULARY TRUC crée dans le vocabulaire CURRENT le nouveau vocabulaire TRUC

ANS_COMPLEMENT

PAD	>IN	SOURCE	⌈	⌈	DECIMAL	HEX	FILL
[CHAR]	CHAR	+!	2/	2*	MIN	MAX	RSHIFT
LSHIFT	INVERT	2OVER	ZSWAP	2DROP	2DUP	2!	2@
S>D	CELL+	CELLS	CHAR+	CHARS	ALIGN	ALIGNED	*/
*/MOD	MOD	/	/MOD	*	FM/MOD	SM/REM	M*
UM*	XOR	OR	AND	{ANS_COMP}			

SD_CARD_LOADER

LOAD" CIB
 LOAD" LOAD" SD_TEST.4TH" charge le fichier source SD_TEST.4TH dans FASTFORTH.
 CIB met sur la pile l'adresse du Current Input Terminal, par défaut TIB.

BOOTLOADER

BOOT
 QUIT becomes a primary DEFERED word
 BOOT la séquence: ' BOOT IS QUIT permet au prochain redémarrage de charger en mémoire le bootstrap BOOT.4th installé à la racine de la SD CARD.
 Pour annuler le bootstrap: ' QUIT >BODY IS QUIT

SD_CARD_READ_WRITE

TERM2SD"	SD_EMIT	WRITE	READ	CLOSE	DEL"	WRITE"	READ"
----------	---------	-------	------	-------	------	--------	-------

TERM2SD" TERM2SD" SD_TEST.4TH" copie le fichier texte sur la SD_CARD (utilisé par CopySourceFileToTarget_SD_Card.bat)
 SD_EMIT envoie la sortie des caractères à la fin du dernier fichier ouvert en écriture.
 WRITE écrit séquentiellement le contenu de BUFFER dans la SD_CARD.
 READ charge séquentiellement le contenu d'un secteur dans le BUFFER
 CLOSE ferme le fichier ouvert en dernier.
 DEL" DEL" SD_TEST.4TH" supprime ce fichier de la SD_CARD.
 WRITE" WRITE" TRUC" ouvre ou crée le fichier TRUC prêt pour écrire à partir de la fin
 READ" READ" TRUC" ouvre le fichier TRUC et charge son premier secteur dans le BUFFER

Consulter le fichier SD_TEST.f

FIXPOINT

2CONSTANT	D>F	S>F	F.	F*	F#S	F/	F-
F+	HOLDS	{FIXPOINT}					

UTILITY

DUMP	U.R	WORDS	?	.RS	.S	{UTILITY}
------	-----	-------	---	-----	----	-----------

U.R u z -- affiche le nombre u avec z digits
 .RS affiche le contenu de la pile de retour
 {UTILITY} si vous tapez ce mot, tous les mots définis après cette balise seront poubellisés.

SD_TOOLS

DIR	FAT	CLUSTER	SECTOR	{SD_TOOLS}
-----	-----	---------	--------	------------

DIR affiche le dump du premier secteur du répertoire courant de la SD_CARD
 FAT affiche le dump du premier secteur de la FAT n°1
 CLUSTER .123 affiche le dump du premier secteur du cluster 123
 SECTOR .123456789 affiche le dump du secteur 123456789
 {SD_TOOLS} si vous tapez ce mot, tous les mots définis après cette balise seront poubellisés.

Comment créer votre copie locale de FASTFORTH

télécharger <https://github.com/jean-michel/FAST-FORTH/archive/master.zip>

une fois que vous l'avez dézippé dans un répertoire créé dans "mes documents", partagez ce répertoire (avec vous-même) et noter son chemin réseau.
Puis clic droit sur la racine de votre PC pour connecter un lecteur réseau, recopiez le chemin réseau que vous avez noté (en prenant soin de remplacer les backslash \ par des slash /), puis affectez-lui une lettre de lecteur, par exemple Z.

Si tout se passe bien, vous devriez obtenir ceci dans l'explorateur (drive = la lettre de votre lecteur):

```
drive:\prog\                TERATERM.ini
drive:\prog\gema\
drive:\prog\MacroAssemblerAS\bin\
drive:\prog\MSP430Flasher\
drive:\prog\Srecord\
drive:\prog\wscite\        SciteGlobal.properties

drive:\
drive:\ADD-ON\
drive:\MSP430-FORTH\
drive:\config\gema\
drive:\config\scite\
drive:\config\scite\AS_MSP430\
                        fichiers programme du FASTFORTH, y compris ceux pour les ADD-ON du noyau
                        fichiers programme du FASTFORTH pour les ADD-ON optionnels (leur version non effaçable)
                        fichiers source FORTH
                        fichiers modèles GEMA
                        others.properties
                        hex.properties
                        fichiers de configuration SCITE pour AS
```

fichiers programme du FASTFORTH, y compris ceux pour les ADD-ON du noyau:

```
drive:\ForthMSP430FRxxxx.asm      programme principal FASTFORTH
  ForthMSP430FRxxxx_ASM.asm      assembleur
  ForthMSP430FRxxxx_SD_ACCEPT.asm ACCEPT version SD_CARD
  ForthMSP430FRxxxx_SD_INIT.asm  initialisation SD_CARD (FAT16/32)
  ForthMSP430FRxxxx_SD_LOAD.asm  pour charger les fichiers source à partir de la SD_CARD
  ForthMSP430FRxxxx_SD_LowLevel.asm routines SPI + Read / write d'un secteur
  ForthMSP430FRxxxx_SD_Rw.asm    pour lire, écrire, créer, supprimer des fichier sur SD_CARD + copie directe
  prog.bat                       bat 'drag and drop' pour flasher (hard link)
  *.inc files                    fichiers de configuration
  *.asm files                    fichiers d'initialisation
  *.mac files                    macros pour l'assembleur
  *.txt files                    fichiers programme prêts à être flashés via un 'drag and drop' sur load.bat
  SciteDirectories.properties    copie de \config\scite\AS_MSP430\SciteDirectories.properties
```

fichiers programme FASTFORTH pour les ADD-ON optionnels (non effaçable version):

```
drive:\ADD_ON\ALIGNMENT.asm
  ANS_COMPLEMENT.asm
  ARITHMETIC.asm
  CONDCOMP.asm
  DOUBLE.asm
  PORTABILITY.asm
  SD_TOOLS.asm
  UTILITY.asm
```

fichiers source FORTH:

```
drive:\MSP430-FORTH\*.4th      fichiers sources prêt à être téléchargé dans FAST FORTH
  *.f                          fichiers source nécessitant le préprocesseur gema avant d'être téléchargés
  *.pat                        décalation assembleur pour une cible particulière
  *.bat                        pour télécharger les fichiers sources dans FASTFORTH, dans la SD_CARD ou pour analyse
  ANS_COMP.f                  identique à ANS_COMP.asm, version effaçable.
  SD_TOOLS.f                  identique à SD_TOOLS.asm, version effaçable.
  UTILITY.f                   identique à UTILITY.asm, version effaçable.
  RTC.f                       pour mettre à l'heure l'horloge temps réel embarquée RTC
  BOOT.f                      exécute un bootstrap
  RC5toLCD.f                  exemple de multitâche
  SD_test.f                   tests pour le driver de SD_CARD, avec explications
```

drive:\MSP430-FORTH\MISC\ répertoire vide. Voir SD_TEST.f

fichiers modèles GEMA

```
drive:\config\gema\FastForthREGtoTI.pat  remplace le nom FORTH des registres par son nom TI (Rx)
  \config\gema\MSP430FR2x4x.pat          déclarations pour les familles MSP430FR2xxx MSP430FR4xxx
  \config\gema\MSP430FR5x6x.pat         déclarations pour les familles MSP430FR5xxx MSP430FR6xxx
  \config\gema\MSP430FR57xx.pat         déclarations pour la famille MSP430FR57xx
  \config\gema\MSP430FRxxxx.pat         déclarations pour les MSP430FRxxxx
  \config\gema\RemoveComments.pat
  \config\gema\SciteDirectory.properties copie de \config\scite\AS_MSP430\SciteDirectories.properties
  \config\gema\tiRegtoFastForth.pat     remplace le nom TI (Rx) des registres par son nom FORTH
```

fichiers de configuration SCITE:

```
drive:\config\scite\AS_MSP430\SciteDirectories.properties  fichier de configuration "répertoire" pour l'éditeur scite
  asm.properties      configuration des fichiers *.inc,*.mac,*.asm
  forth.properties   configuration des fichiers *.f,*.4th
  fortran.properties configuration des fichiers *.pat
```

fichiers bat:

```
drive:\config\MSP430\SendFile.tt1      macro TERATERM pour envoyer un fichier source à FASTFORTH
  SendToSD.tt1                         macro TERATERM pour envoyer un fichier source à la SD_CARD
  build.bat                             utilisé par scite pour faire le fichier programme target.txt
  prog.bat                              pour flasher la cible avec le fichier target.txt
  CopyTo_SD_Card(.bat)
  SendSource(.bat)
  Preprocess(.bat)
  CopySourceFileToTarget_SD_Card.bat    à recopier dans votre répertoire pour le drag'n drop
  SendSourceFileToTarget.bat           à recopier dans votre répertoire pour le drag'n drop
  PreprocessSourceFile.bat             à recopier dans votre répertoire pour le drag'n drop
  SelectTarget.bat                     appelé par les 3 précédents pour sélectionner la cible
```

Note: toutes les commandes réalisées par l'éditeur scite se font au moyen de fichiers bat. On peut donc facilement réutiliser ceux-ci avec n'importe quel éditeur de texte.

Note: les commandes pour flasher ou pour télécharger les fichiers peuvent être réalisées indifféremment depuis l'éditeur scite ou bien directement avec les fichiers bat.

Maintenant il s'agit de préparer l'interface de programmation pour FAST FORTH:

Tout d'abord, il faut récupérer le "flasher" de Texas Instrument et pour cela il faut s'enregistrer ici:

<http://www.ti.com/>, en haut à droite de la page.

Rendez-vous après sur :<http://www.ti.com/tool/msp430-flasher?DCMP=MSP430&HQS=Other+OT+msp430flasher>
On vous demandera si vous voulez vous en servir pour un usage militaire... Bref, installez le logiciel tel quel, En fin de programme d'installation on vous demandera de télécharger "download USB FET drivers, faites-le !

http://software-dl.ti.com/msp430/msp430_public_sw/mcu/msp430/MSPDS/latest/exports/MSP430_DLL_Developer_Package_Rev_3_13_0_001.zip

enfin puis recopiez MSP430Flasher.exe and MSP430.dll dans **drive:\prog\MSP430Flasher**

installation par défaut de teraterm: <http://logmett.com/tera-term-the-latest-version>

Au tour de gema: <https://sourceforge.net/projects/gema/files/latest/download>
dézippez dans **drive:\prog\gema**

Pareil pour scite: <http://www.scintilla.org/Sc400.exe> à dézipper dans **drive:\prog\wscite**
En plus il faut renommer Sc400.exe en scite.exe.

L'assembleur pour compiler FASTFORTH, à dézipper dans **drive:\prog\MacroAssembler**:
<http://john.ccac.rwth-aachen.de:8000/ftp/as/precompiled/i386-unknown-win32/aswcurr.zip>

Et enfin Srecord: <https://sourceforge.net/projects/srecord/files/latest/download>
à dézipper dans **drive:\prog\Srecord**

Au final, voici le minimum requis pour faire tourner l'environnement de programmation (IDE):

drive:\prog\gema\	gema.exe	syntactic preprocessor
drive:\prog\MacroAssembler\bin\	asw.exe P2hex.exe as.msg cmdarg.msg ioerrs.msg P2hex.msg tools.msg	macro assembler
drive:\prog\MSP430Flasher\	MSP430Flasher.exe MSP430.dll	flasher
drive:\prog\Srecord\	srec_cat.exe	TI.hex to TI.txt files converter
drive:\prog\wscite\	scITE.exe ScITEGlobal.properties	text editor

Mais il faut encore modifier ces 4 fichiers liens:

drive:\prog.bat

drive:\MSP430-FORTH\SendSourceFileToTarget.bat
CopySourceFileToTarget_SD_Card.bat
PreprocessSourceFile.f.bat

à partir de l'explorateur, on fait un clic droit sur le premier fichier:
sélectionner "**propriétés**"
changer la lettre du lecteur dans la rubrique "**cible**"

répétez la manip pour les 3 autres fichiers.

Et enfin on crée les associations de fichiers pour l'éditeur scite:

clic droit sur un fichier **.asm**,
sélectionnez "ouvrir avec...",
"plus d'applications"
"rechercher une autre application sur ce PC" : **drive:\prog\wscite\scite.exe**

répéter avec des fichiers **.inc, .mac, .lst, .f, .4th, .pat, .properties, .TTL** files.

voilà c'est fait, rendez-vous dans **forthMSP430FRxxxx.asm** où vous trouverez comment configurer TERATERM !