

## FAST FORTH V2.0 EN RÉSUMÉ

Les mots entre parenthèses () exécutent par défaut les mêmes mots sans parenthèses qui sont des mots DEFER, c'est-à-dire des mots dont l'exécution peut être reroutée. c.f. les mots START et STOP dans le fichier \MSP430-FORTH\RC5toLCD.f

les mots entre accolades {} sont des mots MARKER qui servent de balise.

### FORTH vocabulary

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

ASM	<a href="#">CODE</a>	<a href="#">HI2LO</a>	<a href="#">COLD</a>	<a href="#">WARM</a>	<a href="#">(WARM)</a>	<a href="#">WIPE</a>	<a href="#">RST_HERE</a>
<a href="#">PWR_HERE</a>	<a href="#">RST_STATE</a>	<a href="#">PWR_STATE</a>	<a href="#">MOVE</a>	<a href="#">LEAVE</a>	<a href="#">+LOOP</a>	<a href="#">LOOP</a>	<a href="#">DO</a>
<a href="#">REPEAT</a>	<a href="#">WHILE</a>	<a href="#">AGAIN</a>	<a href="#">UNTIL</a>	<a href="#">BEGIN</a>	<a href="#">THEN</a>	<a href="#">ELSE</a>	<a href="#">IF</a>
<a href="#">DEFER</a>	<a href="#">DOES&gt;</a>	<a href="#">CREATE</a>	<a href="#">CONSTANT</a>	<a href="#">VARIABLE</a>	<a href="#">]</a>	<a href="#">]</a>	<a href="#">POSTPONE</a>
<a href="#">RECURSE</a>	<a href="#">IMMEDIATE</a>	<a href="#">IS</a>	<a href="#">[']</a>	<a href="#">]</a>	<a href="#">]</a>	<a href="#">]</a>	<a href="#">]</a>
<a href="#">ABORT"</a>	<a href="#">ABORT</a>	<a href="#">QUIT</a>	<a href="#">EVALUATE</a>	<a href="#">COUNT</a>	<a href="#">LITERAL</a>	<a href="#">]</a>	<a href="#">EXECUTE</a>
<a href="#">&gt;NUMBER</a>	<a href="#">FIND</a>	<a href="#">WORD</a>	<a href="#">"</a>	<a href="#">S"</a>	<a href="#">TYPE</a>	<a href="#">+ SPACES</a>	<a href="#">SPACE</a>
<a href="#">CR</a>	<a href="#">(CR)</a>	<a href="#">NOECHO</a>	<a href="#">ECHO</a>	<a href="#">EMIT</a>	<a href="#">(EMIT)</a>	<a href="#">(ACCEPT)</a>	<a href="#">ACCEPT</a>
<a href="#">KEY</a>	<a href="#">(KEY)</a>	<a href="#">C.</a>	<a href="#">ALLOT</a>	<a href="#">HERE</a>	<a href="#">]</a>	<a href="#">D.</a>	<a href="#">U.</a>
<a href="#">SIGN</a>	<a href="#">HOLD</a>	<a href="#">#&gt;</a>	<a href="#">#S</a>	<a href="#">#</a>	<a href="#">UM/MOD</a>	<a href="#">&lt;#</a>	<a href="#">BL</a>
<a href="#">STATE</a>	<a href="#">BASE</a>	<a href="#">CIB</a>	<a href="#">J</a>	<a href="#">I</a>	<a href="#">UNLOOP</a>	<a href="#">U&lt;</a>	<a href="#">&gt;</a>
<a href="#">&lt;</a>	<a href="#">=</a>	<a href="#">0&lt;</a>	<a href="#">0&lt;</a>	<a href="#">0=</a>	<a href="#">DABS</a>	<a href="#">1&lt;</a>	<a href="#">1+</a>
<a href="#">ABS</a>	<a href="#">NEGATE</a>	<a href="#">=</a>	<a href="#">+</a>	<a href="#">C!</a>	<a href="#">C@</a>	<a href="#">]</a>	<a href="#">@</a>
<a href="#">DEPTH</a>	<a href="#">R@</a>	<a href="#">R&gt;</a>	<a href="#">&gt;R</a>	<a href="#">ROT</a>	<a href="#">OVER</a>	<a href="#">SWAP</a>	<a href="#">NIP</a>
<a href="#">DROP</a>	<a href="#">?DUP</a>	<a href="#">DUP</a>	<a href="#">LIT</a>	<a href="#">EXIT</a>			

ASM <word>	Crée un mot ASSEMBLER comme le mot CODE quand il ne peut pas être interprété par FORTH (le retour se fait par RET). Un mot ainsi créé doit se finir par ENDASM.
HI2LO	dans une définition sert à passer du FORTH à l'assembleur.
COLD	Software reset
WARM	mot DEFERÉ, exécute (WARM) par défaut.
(WARM)	Réalise un démarrage à chaud.
WIPE	Remet FASTFORTH dans son état initial, en supprimant tous les ajouts effectués..
RST_HERE	définit la limite du programme protégé contre COLD ou bien un reset hardware.
PWR_HERE	définit la limite du programme protégé contre ON/OFF et quand une erreur survient.
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
(CR)	exécute par défaut le mot CR.
(EMIT)	exécute par défaut le mot EMIT
(ACCEPT)	exécute par défaut le mot ACCEPT
(KEY)	exécute par défaut le mot KEY
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

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

<a href="#">?GOTO</a>	Après un test (0=,0<,U>=, U<,0<,S<,S>=) réalise un branchement conditionnel au label FWx ou BWx
<a href="#">GOTO</a>	branche inconditionnellement au label FWx ou BWx
<a href="#">FW3</a>	Label de branchement en avant n°3
<a href="#">FW2</a>	Label de branchement en avant n°2
<a href="#">FW1</a>	Label de branchement en avant n°1
<a href="#">BW3</a>	Label de branchement en arrière n°3
<a href="#">BW2</a>	Label de branchement en arrière n°2
<a href="#">BW1</a>	Label de branchement en arrière n°1
<a href="#">?JMP</a>	Après un test ((0=,0<,U>=, U<,0<,S<,S>=) réalise un branchement conditionnel à un mot existant
<a href="#">JMP</a>	branche inconditionnellement à un mot existant
<a href="#">REPEAT</a>	Version assembleur du mot FORTH REPEAT (Fin d'une structure boucle BEGIN ... WHILE ... REPEAT)
<a href="#">WHILE</a>	Version assembleur du mot FORTH WHILE (branchement conditionnel après test 0=,0<,U>=, U<,0>=, S<, S>=)
<a href="#">AGAIN</a>	Version assembleur du mot FORTH AGAIN (Fin d'une structure boucle BEGIN ... AGAIN)
<a href="#">UNTIL</a>	Version assembleur du mot FORTH UNTIL (branchement conditionnel après test 0=,0<,U>=, U<,0>=, S<, S>=)
<a href="#">ELSE</a>	Version assembleur du mot FORTH ELSE (branchement inconditionnel)
<a href="#">THEN</a>	Version assembleur du mot FORTH THEN (fin d'une structure IF ... THEN ou IF ... ELSE ... THEN)
<a href="#">IF</a>	Version assembleur du mot FORTH IF (branchement conditionnel après test 0=,0<,U>=, U<,0>=, S<, S>=)
<a href="#">LO2HI</a>	dans une définition sert à passer de l'assembleur au FORTH, sans sauvegarder IP.
<a href="#">COLON</a>	sauvegarde IP puis réalise LO2HI, exemple: CODE <word> ... assembly code ... COLON ... FORTH words ... ;
<a href="#">ENDASM</a>	fini la définition d'un mot ASM
<a href="#">ENDCODE</a>	fini la définition d'un mot CODE
<a href="#">(SLEEP)</a>	tâche de fond par défaut. Voir la définition (ACCEPT) dans ForthMSP430FRxxxx.asm
<a href="#">SLEEP</a>	Mot DEFERÉ, exécute (SLEEP) par défaut, permet de créer une tâche de fond spécifique.

Le fichier \MSP430-FORTH\ANS\_COMP.f est un bon exemple d'utilisation de l'assembleur.

### CONDCOMP ADD-ON

[\[DEFINED\]](#)      [\[UNDEFINED\]](#)      [\[IF\]](#)      [\[ELSE\]](#)      [\[THEN\]](#)      [COMPARE](#)      [MARKER](#)

### VOCABULARY ADD-ON

[DEFINITIONS](#)      [ONLY](#)      [PREVIOUS](#)      [ALSO](#)      [ASSEMBLER](#)      [FORTH](#)      [VOCABULARY](#)

FORTH remplace la première librairie dans CONTEXT par la librairie FORTH  
 ASSEMBLER remplace la première librairie dans CONTEXT par la librairie ASSEMBLER  
 VOCABULARY VOCABULARY crée la nouvelle librairie TRUC

## ANS\_COMPLEMENT\_ADD-ON

PAD	>IN	>BODY	SOURCE	+C	C	DECIMAL	HEX
FILL	[CHAR]	CHAR	+!	2/	2*	MIN	MAX
RSHIFT	LSHIFT	INVERT	2OVER	2SWAP	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\_ADD-ON

LOAD"

LOAD"           LOAD" SD\_TEST.4TH" charge le fichier source SD\_TEST.4TH dans FASTFORTH.

## SD\_CARD\_READ\_WRITE\_ADD-ON

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 un secteur sur la SD\_CARD.

READ          écrit 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

## UTILITY\_ADD-ON

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\_ADD-ON

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

<code>drive:\prog\gema\</code>	<code>gema.exe</code>	syntactic preprocessor
<code>drive:\prog\MacroAssemblerAS\bin\</code>	<code>asw.exe</code> <code>P2hex.exe</code> <code>as.msg</code> <code>cmdarg.msg</code> <code>ioerrs.msg</code> <code>P2hex.msg</code> <code>tools.msg</code>	macro assembler
<code>drive:\prog\MSP430Flasher\</code>	<code>MSP430Flasher.exe</code> <code>MSP430.dll</code>	flasher
<code>drive:\prog\Srecord\</code>	<code>srec_cat.exe</code>	TI.hex to TI.txt files converter
<code>drive:\prog\wscite\</code>	<code>scITE.exe</code> <code>SciTEGlobal.properties</code>	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 !