

# FAST FORTH V2.0 RESUMED

Words in parentheses () are the default execution of their paired word without parentheses that are DEFERred words. Example of use: see words START and STOP in \MSP430-FORTH\RC5toLCD.f

words in braces {} are MARKER words.

## FORTH vocabulary

words with hyperlink are ANSI compliant. The others are detailed below.

ASM	<a href="#">CODE</a>	HI2LO	COLD	WARM	(WARM)	WIPE	RST_HERE
PWR_HERE	<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>
REPEAT	<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="#">IMMEDIATE</a>	<a href="#">DEFER</a>	<a href="#">DOES&gt;</a>	<a href="#">CREATE</a>	<a href="#">CONSTANT</a>	<a href="#">VARIABLE</a>	<a href="#">POSTPONE</a>
RECURSE	<a href="#">ABORT</a>	<a href="#">IS</a>	<a href="#">[']</a>	<a href="#">COUNT</a>	<a href="#">[</a>	<a href="#">\</a>	<a href="#">EXECUTE</a>
ABORT"	<a href="#">FIND</a>	<a href="#">QUIT</a>	<a href="#">EVALUATE</a>	<a href="#">S"</a>	<a href="#">LITERAL</a>	<a href="#">+ SPACES</a>	<a href="#">SPACE</a>
>NUMBER	<a href="#">(CR)</a>	<a href="#">WORD</a>	<a href="#">."</a>	<a href="#">EMIT</a>	<a href="#">TYPE</a>	<a href="#">(ACCEPT)</a>	<a href="#">ACCEPT</a>
CR	<a href="#">(KEY)</a>	<a href="#">NOECHO</a>	<a href="#">ECHO</a>	<a href="#">HERE</a>	<a href="#">(EMIT)</a>	<a href="#">D.</a>	<a href="#">U.</a>
KEY	<a href="#">HOLD</a>	<a href="#">C.</a>	<a href="#">ALLLOT</a>	<a href="#">#</a>	<a href="#">&lt;#</a>	<a href="#">BL</a>	<a href="#">STATE</a>
SIGN	<a href="#">&gt;IN</a>	<a href="#">#&gt;</a>	<a href="#">CIB</a>	<a href="#">PAD</a>	<a href="#">J</a>	<a href="#">I</a>	<a href="#">UNLOOP</a>
BASE	<a href="#">&gt;</a>	<a href="#">CPL</a>	<a href="#">=</a>	<a href="#">O&gt;</a>	<a href="#">Q&lt;</a>	<a href="#">O=</a>	<a href="#">DABS</a>
U<	<a href="#">NEGATE</a>	<a href="#">XOR</a>	<a href="#">OR</a>	<a href="#">AND</a>	<a href="#">=</a>	<a href="#">+</a>	<a href="#">C!</a>
ABS	<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>
C@	<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>
OVER							

## ASSEMBLER vocabulary

words without hyperlink are detailed below.

?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>=	<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>	

Here are adds-on to be compiled

### CONDCOMP

<a href="#">[DEFINED]</a>	<a href="#">[UNDEFINED]</a>	<a href="#">[IF]</a>	<a href="#">[ELSE]</a>	<a href="#">[THEN]</a>	<a href="#">COMPARE</a>	<a href="#">MARKER</a>
---------------------------	-----------------------------	----------------------	------------------------	------------------------	-------------------------	------------------------

### VOCABULARY

<a href="#">DEFINITIONS</a>	<a href="#">ONLY</a>	<a href="#">PREVIOUS</a>	<a href="#">ALSO</a>	<a href="#">ASSEMBLER</a>	<a href="#">FORTH</a>	<a href="#">VOCABULARY</a>
-----------------------------	----------------------	--------------------------	----------------------	---------------------------	-----------------------	----------------------------

### SD\_CARD\_LOADER

LOAD"

### SD\_CARD\_READ\_WRITE

TERM2SD"	<a href="#">SD_EMIT</a>	WRITE	READ	CLOSE	DEL"	WRITE"	READ"
----------	-------------------------	-------	------	-------	------	--------	-------

Below, adds-on that can be compiled in kernel or loaded later

### FIXPOINT

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

### ANS\_COMPLEMENT

<a href="#">&gt;BODY</a>	<a href="#">SOURCE</a>	<a href="#">.(</a>	<a href="#">(</a>	<a href="#">DECIMAL</a>	<a href="#">HEX</a>	<a href="#">FILL</a>	<a href="#">[CHAR]</a>
<a href="#">CHAR</a>	<a href="#">+!</a>	<a href="#">2/</a>	<a href="#">2*</a>	<a href="#">MIN</a>	<a href="#">MAX</a>	<a href="#">1-</a>	<a href="#">1+</a>
<a href="#">RSHIFT</a>	<a href="#">LSHIFT</a>	<a href="#">INVERT</a>	<a href="#">2OVER</a>	<a href="#">2SWAP</a>	<a href="#">2DROP</a>	<a href="#">2DUP</a>	<a href="#">2!</a>
<a href="#">2@</a>	<a href="#">S&gt;D</a>	<a href="#">CELL+</a>	<a href="#">CELLS</a>	<a href="#">CHAR+</a>	<a href="#">CHARS</a>	<a href="#">ALIGN</a>	<a href="#">ALIGNED</a>
<a href="#">*/</a>	<a href="#">*/MOD</a>	<a href="#">MOD</a>	<a href="#">/</a>	<a href="#">/MOD</a>	<a href="#">*</a>	<a href="#">FM/MOD</a>	<a href="#">SM/REM</a>
<a href="#">UM/MOD</a>	<a href="#">M*</a>	<a href="#">UM*</a>	<a href="#">{ANS_COMP}</a>				

### UTILITY

<a href="#">DUMP</a>	<a href="#">U.R</a>	<a href="#">WORDS</a>	<a href="#">?</a>	<a href="#">.RS</a>	<a href="#">.S</a>	<a href="#">{UTILITY}</a>
----------------------	---------------------	-----------------------	-------------------	---------------------	--------------------	---------------------------

### SD\_TOOLS

DIR	FAT	CLUSTER	SECTOR	<a href="#">{SD_TOOLS}</a>
-----	-----	---------	--------	----------------------------

### OTHER FASTFORTH WORDS (not ANSI)

ASM <word>	creates an assembler word as CODE but which is not interpretable by FORTH (because use of CALL ... RET). this defined <word> must be ended with ENDASM.
HI2LO	used to switch from a high level (FORTH) to low level (assembler) modes.
COLD	software reset
WARM	DEFERred word, initially executes (WARM)
(WARM)	performs a hot start

WIPE resets the program memory to its original state.

RST\_HERE defines the boundary of the program memory protected against COLD or hardware reset.

PWR\_HERE defines the boundary of the program memory protected against ON/OFF and against any error occurring.

RST\_STATE remove all words defined after RST\_HERE

PWR\_STATE remove all words defined after PWR\_HERE

(CR) executes ANS definition CR

(EMIT) executes ANS definition EMIT

(ACCEPT) executes ANS definition ACCEPT

(KEY) executes ANS definition KEY

NOECHO stop display on output

ECHO start display on output

CPL -- size of Current Input Buffer (Chars Per Line)

CIB -- addr of Current Input Buffer

PAD -- addr of PAD

LIT execution part of LITERAL

#### FASTFORTH ASSEMBLER words

?GOTO used after a conditional (0=,0<>,U>=,U<,0<,S<,S>=) to branch to a label FWx or BWx  
GOTO used as unconditional branch to a label FWx or BWx

FW3 FORWARD branch destination n°3  
FW2 FORWARD branch destination n°2  
FW1 FORWARD branch destination n°1

BW3 BACKWARD branch destination n°3  
BW2 BACKWARD branch destination n°2  
BW1 BACKWARD branch destination n°1

?JMP used after a conditional (0=,0<>,U>=,U<,0<,S<,S>=) to jump to a defined word  
JMP unconditional jump to a defined word

REPEAT assembler version of the FORTH word REPEAT (unconditional branch)  
WHILE assembler version of the FORTH word WHILE (conditional branch preceded by 0=,0<>,U>=,U<,0>=,S<,S>=)  
AGAIN assembler version of the FORTH word AGAIN (unconditional branch)  
UNTIL assembler version of the FORTH word UNTIL (conditional branch preceded by 0=,0<>,U>=,U<,0>=,S<,S>=)  
ELSE assembler version of the FORTH word ELSE (unconditional branch)  
THEN assembler version of the FORTH word THEN ends IF or IF ELSE statements  
IF assembler version of the FORTH word IF (conditional branch preceded by 0=,0<>,U>=,U<,0>=,S<,S>=)

LO2HI switches between low level and high level interpretation mode (counterpart of HI2LO), without saving IP.

COLON pushes IP then performs LO2HI, used as: CODE <word> ... assembly code ... COLON ... FORTH words ... ;

ENDASM to end an ASM definition

ENDCODE to end a CODE definition

(SLEEP) performs the default background task. See (ACCEPT) in ForthMSP430FRxxxx.asm

SLEEP DEFERred word, initially executes (SLEEP), and which enables you to create your own background task.

To better understand the use of the assembler I refer you to \MSP430-FORTH\ANS\_COMP.f.

#### SD\_CARD

LOAD" LOAD" SD\_TEST.4TH" loads file SD\_TEST.4TH to FASTFORTH.  
TERM2SD" TERM2SD" SD\_TEST.4TH" copy input file to SD\_CARD (use CopySourceFileToTarget\_SD\_Card.bat to do)  
SD\_EMIT sends output stream to the end of last opened as write file.  
WRITE write sequentially BUFFER content to a sector  
READ read sequentially a sector to BUFFER  
CLOSE close last opened file.  
DEL" DEL" SD\_TEST.4TH" remove this file from SD\_CARD.  
WRITE" WRITE" TRUC" open or create TRUC file ready to write to the end of this file  
READ" READ" TRUC" open TRUC and load its first sector in BUFFER

see SD\_TEST.f

#### VOCABULARY

FORTH replace first words set in CONTEXT by the words set FORTH  
ASSEMBLER replace first words set in CONTEXT by the words set ASSEMBLER  
VOCABULARY VOCABULARY TRUC creates a new words set called TRUC

#### UTILITY

U.R u z -- display unsigned number u with size z  
.RS display Return Stack content  
{UTILITY} if you type {UTILITY} all subsequent loaded words are removed

#### SD\_TOOLS

DIR dump first sector of current directory  
FAT dump first sector of FAT1  
CLUSTER .123 display first sector of cluster 123  
SECTOR .123456789 display sector 123456789  
{SD\_TOOLS} if you type {SD\_TOOLS} all subsequent loaded words are removed

# build your FASTFORTH local copy

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

once you have unzipped it into your folder, share it (with you) and notice its network path. Then right clic on the root of your notepad to create a network drive by recopying this network path (change backslashes \ to slashes /); then set drive letter as you want.

In explorer you should obtain that:

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

drive:\ source files to build FASTFORTH, including files for KERNEL ADD-ON switches
drive:\ADD-ON\ FASTFORTH build ADD-ON files for OPTIONAL KERNEL ADD-ON switches (not erasable version)
drive:\MSP430-FORTH\ FORTH source files
drive:\config\gema\ GEMA pattern files
drive:\config\scite\ others.properties
hex.properties
drive:\config\scite\AS_MSP430\ SCITE configuration files
```

source files to build FASTFORTH, including files for KERNEL ADD-ON switches:

```
drive:\ForthMSP430FRxxxx.asm main FASTFORTH program
ForthMSP430FRxxxx_ASM.asm assembler
ForthMSP430FRxxxx_SD_ACCEPT.asm ACCEPT from SD_CARD
ForthMSP430FRxxxx_SD_INIT.asm to init SD_CARD (FAT16/32)
ForthMSP430FRxxxx_SD_LOAD.asm to load source files from SD_CARD
ForthMSP430FRxxxx_SD_LowLevel.asm SPI routines + Read / write sector
ForthMSP430FRxxxx_SD_Rw.asm to read create write del SD_CARD files + file copy from terminal to SD_CARD
prog.bat 'drag and drop' programming bat file (hard link)
*.inc files targets configuration files
*.asm files targets (minimalist) init files
*.mac files macros files for AS assembler
*.txt files program files ready to 'drag and drop' onto prog.bat
SciTEDirectories.properties copy of \config\scite\AS_MSP430\SciTEDirectories.properties
```

FASTFORTH build ADD-ON files for OPTIONAL KERNEL ADD-ON switches (not erasable option version):

```
drive:\ADD-ON\ALIGNMENT.asm
ANS_COMPLEMENT.asm
ARITHMETIC.asm
CONDCOMP.asm
DOUBLE.asm
PORTABILITY.asm
SD_TOOLS.asm
UTILITY.asm
```

FORTH source files:

```
drive:\MSP430-FORTH\*.4th pure FORTH generic source files ready to download without preprocessing
*.f source files with use of assembler, must be preprocessed before downloading
*.pat assembly declarations for specific target
*.bat to download source file to target, to SD_CARD target, and to debug (hard links)
ANS_COMP.f same as ANS_COMP.asm, (erasable)
SD_TOOLS.f same as SD_TOOLS.asm, (erasable)
UTILITY.f same as UTILITY.asm, (erasable)
RTC.f to set time and date with embedded RTC
BOOT.f performs bootstrap
RC5toLCD.f multitasking example:
SD_test.f tests for SD_CARD option: contains the explanations
```

drive:\MSP430-FORTH\MISC\ empty directory. See use in SD\_TEST.f

GEMA pattern files

```
drive:\config\gema\FastForthREGtoTI.pat converts FORTH symbolic registers names to TI Rx registers
\config\gema\MSP430FR2x4X.pat declarations for MSP430FR2 MSP430FR4 families, assembly part
\config\gema\MSP430FR2x4X_FastForth.pat declarations for MSP430FR2 MSP430FR4 families, FORTH part
\config\gema\MSP430FR5x6X.pat declarations for MSP430FR6 MSP430FR6 families, assembly part
\config\gema\MSP430FR5x6X_FastForth.pat declarations for MSP430FR5 MSP430FR6 families, FORTH part
\config\gema\MSP430FR57xx.pat declarations for MSP430FR57 family, assembly part
\config\gema\MSP430FR57xx_FastForth.pat declarations for MSP430FR57 family, FORTH part
\config\gema\MSP430FRxxxx.pat assembly declarations for device MSP430FRxxxx
\config\gema\RemoveComments.pat
\config\gema\SciTEDirectory.properties copy of \config\scite\AS_MSP430\SciTEDirectories.properties
\config\gema\tiREGtoFastForth.pat converts TI Rx registers to FORTH symbolic registers names
```

SCITE configuration files:

```
drive:\config\scite\AS_MSP430\SciTEDirectories.properties scite directory config file
asm.properties configuration for *.inc,*.mac,*.asm files
forth.properties configuration for *.f,*.4th files
fortran.properties configuration for *.pat files
SendFile.ttl TERATERM macro file to send source file to FASTFORTH
SendToSD.ttl TERATERM macro file to send source file to embedded SD_CARD
error.bat
build.bat called by scite to build target.txt program
prog.bat to flash target with target.txt file
CopySourceFileToTarget_SD_Card.bat to send a file to target SD_CARD
SendSourceFileToTarget.bat to send file to FASTFORTH
PreprocessSourceFile.f.bat to convert generic .f file to specific .4th file
```

Note: all actions made from SciTE editor are processed via bat files. So you can easily use your preferred editor by reuse them.

Note: all actions (flashing target, downloading files) can be made by using bat files directly, i.e. without use of SciTE editor.

The next is to download IDE:

First get TI's programs

go here: <http://www.ti.com/> and registers you to enable MSP430Flasher downloading:

<http://www.ti.com/tool/msp430-flasher?DCMP=MSP430&HQS=Other+OT+msp430flasher>  
and  
[http://software-dl.ti.com/msp430/msp430\\_public\\_sw/mcu/msp430/MSP430\\_FET\\_Drivers/latest/index\\_FDS.html](http://software-dl.ti.com/msp430/msp430_public_sw/mcu/msp430/MSP430_FET_Drivers/latest/index_FDS.html)

install in the suggested directory,  
then copy MSP430Flasher.exe and MSP430.dll to **drive:\prog\MSP430Flasher\**

download and install teraterm: <http://logmett.com/tera-term-the-latest-version>

<https://sourceforge.net/projects/gema/files/latest/download>  
unzip in **drive:\prog\gema\**

download <http://www.scintilla.org/Sc400.exe> to **drive:\prog\wscite\**  
then rename Sc400.exe to scite.exe

<http://john.ccac.rwth-aachen.de:8000/ftp/as/precompiled/i386-unknown-win32/aswcurr.zip>  
unzip in **drive:\prog\MacroAssemblerAS\**

<https://sourceforge.net/projects/srecord/files/latest/download>  
unzip in **drive:\prog\Srecord\**

In explorer you should obtain that (minimum requested programs):

<b>drive:\prog\</b>	<b>TERATERM.ini</b>	
<b>drive:\prog\gema\</b>	<b>gema.exe</b>	syntactic preprocessor
<b>drive:\prog\MacroAssemblerAS\bin\</b>	<b>asw.exe</b> <b>P2hex.exe</b> <b>as.msg</b> <b>cmdarg.msg</b> <b>ioerrs.msg</b> <b>P2hex.msg</b> <b>tools.msg</b>	macro assembler linker
<b>drive:\prog\MSP430Flasher\</b>	<b>MSP430Flasher.exe</b> <b>MSP430.dll</b>	flasher
<b>drive:\prog\Srecord\</b>	<b>srec_cat.exe</b>	TI.hex to TI.txt files converter
<b>drive:\prog\wscite\</b>	<b>scITE.exe</b> <b>scITEGlobal.properties</b>	text editor

Next we need to change the drive letter in hard links below:

**drive:\prog.bat**

**drive:\MSP430-FORTH\SendSourceFileToTarget.bat**  
**CopySourceFileToTarget\_SD\_Card.bat**  
**PreprocessSourceFile.f.bat**

to do, right clic on them  
select "properties"  
set your drive letter in "target"

The last step is ask windows to associate scite editor with file types:

right clic on a **.asm** file,  
select "open with",  
select "other application" then select: **drive:\prog\wscite\scite.exe**

repeat for **.inc**, **.mac**, **.lst**, **.f**, **.4th**, **.pat**, **.properties**, **.TTL** files.

IT's done ! See **forthMSP430FRxxxx.asm** to configure TeraTerm