FAST FORTH V3.6 RESUMED

https://framagit.org/Jean-Mi/FAST-FORTH

Words in braces {} are MARKER words.

FORTH vocabulary

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

COLD	WARM	WIPE	RST_HERE	PWR_HERE	RST_STATE	PWR_STATE	BEGIN
CREATE	:	;	IMMEDIATE	POSTPONE	]	[	\
'	[']	?ID	ABORT"	ABORT	QUIT	TREAT	COUNT
LITERAL	,	>NUMBER	FIND	WORD	."	S"	.
U.	SIGN	HOLD	#>	#S	#	<#	KEY
CR	TYPE	NOECHO	ECHO	EMIT	ACCEPT		

COLD — Software reset

WARM — primary DEFERed word, 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

?ID — %deviceID% ?ID --
String sent by Teraterm to the target before downloading a file.4TH issued from a source file.f.
Resets terminal; If %deviceID% = 0, do nothing else
?ID substracts %deviceID% from the target's value then performs ABORT" DeviceID mismatch!"

TREAT — text interpreter, common part of EVALUATE and QUIT

NOECHO — stop display on output, set LINE = 1

ECHO — start display on output, set LINE = 0

ASSEMBLER vocabulary

?GOTO	GOTO	FW3	FW2	FW1	BW3	BW2	BW1
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					
ASM	CODE	HI2LO	<-- added to FORTH vocabulary				

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.

CODE <word> — creates a FORTH words,ready to be written in assembly.
This word must be terminated with ENDCODE unless using COLON or LO2HI.

HI2LO — used to switch from a high level (FORTH) to low level (assembler) modes.

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

FW3 — FORWARD branch destination n°3 (single use), must be written in first 7 columns of the line
FW2 — FORWARD branch destination n°2 (single use), -
FW1 — FORWARD branch destination n°1 (single use), -

BW3 — BACKWARD branch destination n°3, must be written in first 7 columns of the line
BW2 — BACKWARD branch destination n°2, -
BW1 — BACKWARD branch destination n°1, -

REPEAT — assembler version of the FORTH word REPEAT (unconditionnal backward branch)
WHILE — assembler version of the FORTH word WHILE (used with 0=,0<>,U>=,U<,0>=,S<,S>=)
AGAIN — assembler version of the FORTH word AGAIN (unconditionnal loop)
UNTIL — assembler version of the FORTH word UNTIL (used with 0=,0<>,U>=,U<,0>=,S<,S>=)
ELSE — assembler version of the FORTH word ELSE (unconditionnal forward branch)
THEN — assembler version of the FORTH word THEN ends IF or IF ELSE statements
IF — assembler version of the FORTH word IF (used with 0=,0<>,U>=,U<,0>=,S<,S>=)

LO2HI — switches from low level to high level interpretation mode (counterpart of HI2LO), without saving IP.
COLON — pushes IP then performs LO2HI, used as: CODE <name> assembly instr. ... COLON word ... ;
ENDASM — to end an ASM definition
ENDCODE — to end a CODE definition

To better understand the use of the assembler I refer you to \MSP430-FORTH\ANS_COMP.f and \MSP430-FORTH\RC5toLCD.f

Extended ASSEMBLER words

RPT	PUSHX.B	PUSHX.A	PUSHX	SXTX.A	SXTX	RRAX.B	RRAX.A
RRAX	SWPBX.A	SWPBX	RRUX.B	RRUX.A	RRUX	RRCX.B	RRCX.A
RRCX	ANDX.B	ANDX.A	ANDX	XORX.B	XORX.A	XORX	BISX.B
BISX.A	BISX	BICX.B	BICX.A	BICX	BITX.B	BITX.A	BITX
DADDX.B	DADDX.A	DADDX	CMPX.B	CMPX.A	CMPX	SUBX.B	SUBX.A
SUBX	SUBCX.B	SUBCX.A	SUBCX	ADDCX.B	ADDCX.A	ADDCX	ADDX.B
ADDX.A	ADDX	MOVX.B	MOVX.A	MOVX	CALLA	SUBA	ADDA
CMPA	MOVA						

RPT #n|RPT Rn — used with Reg and Reg,Reg eXtended instructions, to repeat them 1 to 16 times.
Example: RPT #12 ADDX R1,R1 will shift left 12 times R1

Here are adds-on that can be compiled in kernel only

CONDCOMP

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

VOCABULARY

DEFINITIONS ONLY PREVIOUS ALSO ASSEMBLER FORTH 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
```

SD_CARD_LOADER

```
LOAD"       LOAD" SD_TEST.4TH" compiles/executes file SD_TEST.4TH from current_directory.
            LOAD" \MISC\TEST_ASM.4TH" compiles/executes file TEST_ASM.4TH from current_directory\MISC\.
            LOAD" \MISC" changes to directory \MISC
            LOAD" ..\" changes to parent directory
            LOAD" \" changes to root directory
```

SD_CARD_READ_WRITE

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

```
TERM2SD"    TERM2SD" SD_TEST.4TH" copy input file to SD_CARD (use CopySourceFileToTarget_SD_Card.bat to do)
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" quiet 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

DEFERRED_ADD-ON

:NONAME DEFER IS CODENNM

```
CODENNM     assembly counterpart of :NONAME
```

BOOTLOADER

BOOT

QUIT becomes a primary DEFERed word

```
BOOT        the input: ' BOOT IS QUIT    allow downloading BOOT.4th from SD CARD during the process RESET.
            to cancel the bootstrap: ' QUIT >BODY IS QUIT
```

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

CORE_ANS

VALUE	TO	PAD	>IN	BASE	STATE	SOURCE	EXECUTE
EVALUATE	HERE	ALLOT	RECURSE	+LOOP	LOOP	I	DO
REPEAT	WHILE	AGAIN	UNTIL	THEN	ELSE	IF	>BODY
LEAVE	UNLOOP	SPACES	SPACE	BL	J	.(	(
DECIMAL	HEX	FILL	[CHAR]	CHAR	+!	MIN	MAX
2/	2*	RSHIFT	LSHIFT	>	<	INVERT	XOR
OR	AND	C,	C!	C@	NIP	2OVER	2SWAP
2DROP	2DUP	2VALUE	2!	2@	R@	ROT	OVER
CELL+	CELLS	CHAR+	CHARS	ALIGN	ALIGNED	*/	*/MOD
MOD	/	/MOD	*	1+	+	ABS	NEGATE
FM/MOD	SM/REM	UM/MOD	M*	UM*	S>D	TO	VALUE
DOES>	CONSTANT	VARIABLE	U<	=	0<	0=	!
@	1-	-	DEPTH	R>	>R	SWAP	DROP
?DUP	DUP	EXIT	MOVE	{CORE_ANS}			

```
{CORE_ANS}      do nothing if compiled in core, else remove all from {CORE_ANS}.
```

UTILITY

DUMP U.R WORDS ? .RS .S {TOOLS}

```
U.R   u z --    display unsigned number u with size z
.RS             display Return Stack content
{TOOLS}       do nothing if compiled in core, else remove all from {TOOLS}
```

SD_TOOLS

DIR FAT CLUSTER SECTOR {SD_TOOLS}

```
DIR             dump first sector of current directory
FAT             dump first sector of FAT1
CLUSTER         .123 CLUSTER displays first sector of cluster 123
SECTOR          .123456789 SECTOR displays sector 123456789
{SD_TOOLS}      do nothing if compiled in core, else remove all from {SD_TOOLS}.
```

DOUBLE

you must uncomment the DOUBLE_INPUT compilation switch before use this word set.

D.R	2LITERAL	2VALUE	2CONSTANT	2VARIABLE	M*/	DMIN
DMAX	D2*	D2/	DABS	DNEGATE	D-	M+
D+	DU<	D<	D=	D0<	D0=	D>S
2ROT	D.	2R>	2R@	2>R	{DOUBLE}	

FIXPOINT you must uncomment the FIXPOINT_INPUT compilation switch before use this add-on.

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

S>F	u/n -- Qlo Qhi convert u/n in a Q15.16 value
F.	display a Q15.16 value
F*	Q15.16 multiplication
F#S	Qlo Qhi u -- Qhi 0 convert fractionnal part of a Q15.16 value displaying u digits
F/	Q15.16 division
F-	Q15.16 soustraction
F+	Q15.16 addition
{FIXPOINT}	do nothing if compiled in core, else remove all from {FIXPOINT}.

build your FastForth local copy

download https://framagit.org/Jean-Mi/FAST-FORTH/tree/master
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 this:

```
drive:\                                 forthMSP430FR.asm files ready to build
      \ForthMSP430FR.asm                main FASTFORTH program
      \ForthMSP430FR_ASM.asm            assembler
      \ForthMSP430FR_EXTD_ASM.asm       extended assembler
      \ForthMSP430FR_CONDCOMP.asm       conditionnal compilation
      \ForthMSP430FR_SD_ACCEPT.asm      ACCEPT for SD_Card
      \ForthMSP430FR_SD_INIT.asm        init SD_CARD (FAT16/32)
      \ForthMSP430FR_SD_LOAD.asm        load source files from SD_CARD
      \ForthMSP430FR_SD_LowLevel.asm    SPI routines + Read / write sector
      \ForthMSP430FR_SD_RW.asm          read create write del SD_CARD files + file copy from terminal to SD_CARD
      \ForthMSP430FR_TERM_I2C.asm       I2C terminal
      \ForthMSP430FR_TERM_UART.asm      full duplex UART terminal
      \ForthMSP430FR_TERM_UART_HALF.asm half duplex UART terminal
      \SciTEDirectories.properties      copy of \config\scite\AS_MSP430\SciTEDirectories.properties

drive:\ADD-ON\                          FASTFORTH OPTIONAL KERNEL ADD-ON switches (not erasable version)
            \CORE_ANS.asm               set of complementary words to pass CORETEST.4TH
            \FIXPOINT.asm               adds HOLDS F+ F- F* F/ F#S F. S>F
            \SD_TOOLS.asm               adds some trivial words to display sectors content
            \UTILITY.asm                adds WORDS, DUMP, ? .S

drive:\binaries\                files.txt|files.HEX ready for drag'n drop to prog.bat
              \prog(.bat)       used to program targets.

drive:\config\          some files.bat
      \config\          Teraterm macros files.ttl
      \config\          SCITE configuration files.properties

drive:\inc\             MACRO ASsembler files.inc, files.asm, GEMA preprocessor files.pat
         \MSP430FRxxxx.inc      device configuration for AS assembler
         \MSP430FRxxxx.asm      device init code for AS assembler
         \MSP_EXP430FRxxxx.asm  target configuration for AS assembler
         \FastForthREGtoTI.pat  converts FORTH symbolic registers names to TI Rx registers
         \tiREGtoFastForth.pat  converts TI Rx registers to FORTH symbolic registers names
         \MSP430FRxxxx.pat      device configuration for gema preprocessor
         \MSP_EXP430FRxxxx.pat  target configuration for gema preprocessor
         \ThingsInFirst.inc     general pre configuration for AS assembler
         \ThingsInLast.inc      general post configuration for AS assembler

drive:\MSP430-FORTH\    FORTH generic_source_files.f and targeted_source_files.4th
                  \PreprocessSourceFile.bat             (link)
                  \SendSourceFileToTarget.bat           (link)
                  \CopySourceFileToTarget_SD_Card.bat   (link)
                  \*.f          source files which must be preprocessed before downloading
                  \*.4th        source files ready to download to any target
                  \LAST.4TH     last source file issued by preprocessor and downloaded to your target
                  \BOOT.f       performs bootstrap
                  \CHNGBAUD.f   allows you to change terminal baudrate
                  \CORE_ANS.f   same as CORE_ANS.asm, (but erasable)
                  \CORETEST.4TH ANS core tests
                  \CORDIC.f     for afficionados
                  \DOUBLE.f     adds DOUBLE word set
                  \FIXPOINT.f   same as FIXPOINT.asm, (but erasable)
                  \FF_SPECS.f   shows all specificities of FAST-FORTH compiled on your target
                  \RTC.f        set date and time (MSP430FR5xxx, FR6xxx)
                  \RC5toLCD.f   multitasking example
                  \SD_test.f    tests for SD_CARD driver
                  \SD_TOOLS.f   same as SD_TOOLS.asm, (but erasable)
                  \TESTASM.f    some tests for embedded assembler
                  \TESTXASM.f   some tests for embedded extended assembler
                  \UARTI2CS.f   I2C_Master driver to link TERMINAL UART with any I2CSlave target
                  \UTILITY.f    same as UTILITY.asm, (but erasable)


drive:\prog\            SciTEGlobal.properties, TERATERM.INI + programs.url


SCITE configuration files:
drive:\config\asm.properties            configuration for *.inc,*.asm files
            \forth.properties           configuration for *.f,*.4th files
            \fortran.properties         configuration for *.pat files


drive:\config\SendFile.ttl                      TERATERM macro file to send source file to FASTFORTH
            \SendToSD.ttl                       TERATERM macro file to send source file to embedded SD_CARD
            \build(.bat)                        called by scite to build target.txt program
            \BSL_prog(.bat)                     to flash target with target.txt file with BSL_Scripter
            \FET_prog(.bat)                     to flash target with target.txt file with MSP430Flasher
            \CopyTo_SD_Card(.bat)               to copy in your MSP430-FORTH
            \SendSource(.bat)                   to send file to FASTFORTH
            \Preprocess(.bat)                   to convert generic .f file to specific .4th file
            \CopySourceFileToTarget_SD_Card.bat copy it in any user folder for drag'n drop use
            \SendSourceFileToTarget.bat         copy it in any user folder for drag'n drop use
            \PreprocessSourceFile.bat           copy it in any user folder for drag'n drop use
            \SelectTarget.bat                   called to select target, device and deviceID
```

Note: all actions made from SciTE editor are also processed via bat/bash files.
So you can easily use your prefered 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 (WINDOWS):

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

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

download BSL-Scripter-v3.4.2.zip and unzip as **drive:\prog\BSL-Scriper.exe**

download and install teraterm: https://osdn.net/projects/ttssh2/releases/

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

download http://www.scintilla.org/Sc41x.exe to **drive:\prog**
then rename Sc41x.exe to scite.exe

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

https://sourceforge.net/projects/srecord/files/srecord-win32/1.64/
unzip in **drive:\prog**

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

drive:\prog
- **as.msg**
- **asw.exe**
- **BSL-Scripter.exe**
- **cmdarg.msg**
- **gema.exe**
- **ioerrs.msg**
- **MSP430.dll**
- **MSP430Flasher.exe**
- **P2hex.exe**
- **P2hex.msg**
- **srec_cat.exe**
- **sCiTE.exe**
- **SciTEGlobal.properties**
- **tools.msg**

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

drive:\binaries\prog.bat

drive:\MSP430-FORTH\SendSourceFileToTarget.bat
CopySourceFileToTarget_SD_Card.bat
PreprocessSourceFile.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\scite.exe**

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

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

IDE for linux UBUNTU / MINT

First search from ti.com:

http://software-dl.ti.com/msp430/msp430_public_sw/mcu/msp430/MSP430Flasher/latest/index_FDS.html

untar in a home folder then:
open MSPFlasher-1.3.16-linux-x64-installer.run
    install in MSP430Flasher (under home)

open a terminal in MSP430Flasher/Drivers:
    sudo ./msp430uif_install.sh

copy MSP430Flasher/MSP430Flasher to /usr/local/bin/MSP430Flasher
copy MSP430Flasher/libmsp430.so to /usr/local/lib/MSP430Flasher/libmsp430.so

open an editor as superuser in /etc/ld.so.conf.d/
     write on first line (of new file): /usr/local/lib/msp430flasher/
     save this new file as libmsp430.conf
then in a terminal: sudo /sbin/ldconfig

install the package srecord

install the package scite
as super user, edit /etc/scite/SciTEGlobal.properties
uncomment (line 18): position.maximize=1
uncomment (line 257): properties.directory.enable=1
add line 7: PLAT_WIN=0
add line 8: PLAT_GTK=1
save file

at the end of your ~.profile file, add these two lines:
FF="/the_root_of_your_FastForth_local_copy"
export FF

https://sourceforge.net/projects/gema/files/gema/gema-1.4-RC/gema-1.4RC-src.tgz/download
untar in a home folder then:
make (ignore warnings)
sudo make install (ignore warnings)
make clean
result in: /usr/local/bin/gema

http://john.ccac.rwth-aachen.de:8000/ftp/as/source/c_version/asl-current.tar.gz
untar in a home folder then:
copy /Makefile.def-samples/Makefile.def-i386-unknown-linux2.x,x to ../Makefile.def
edit this Makefile.def to remove "-march=i586" option from line 7 (if any)
make
make test
sudo make install
make clean
result: asl files are in /usr/local

install minicom package

sudo gpasswd --add ${USER} dialout

copy /config/msp430/.minirc.dfl in your home directory.

In /inc/RemoveComments.pat, deselect windows part, select linux part.

With scite editor you can
   - assemble FastForth then download it to eZFET target,
   - edit your source files
   - preprocess file.f to file.4th

With minicom you can send a file.4th to your target via dev/ttyUSB0, up to 4Mbauds:
CTRL_A + Y to send a file