FAST FORTH V2.0 RESUMED

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

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
NOECHO	stop display on output
ECHO	start display on output
CIB	leave addr of Current Input Buffer
LIT	execution part of 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 <-- 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 writtent 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
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 conditionnal (0=,0<>,U>=,U<,0<,S<,S>=) to jump to a defined word
JMP	unconditionnal jump to a defined word
REPEAT	assembler version of the FORTH word REPEAT (unconditionnal branch)
WHILE	assembler version of the FORTH word WHILE (conditionnal branch preceded by 0=,0<>,U>=,U<,0>=,S<,S>=)
AGAIN	assembler version of the FORTH word AGAIN (unconditionnal branch)
UNTIL	assembler version of the FORTH word UNTIL (conditionnal branch preceded by 0=,0<>,U>=,U<,0>=,S<,S>=)
ELSE	assembler version of the FORTH word ELSE (unconditionnal branch)
THEN	assembler version of the FORTH word THEN ends IF or IF ELSE statements
IF	assembler version of the FORTH word IF (conditionnal 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	DEFERed word, initially executes the default background task, It 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 and \MSP430-FORTH\RC5toLCD.f

Here are adds-on to be compiled

CONDCOMP

[DEFINED] [UNDEFINED] [IF] [ELSE] [THEN] COMPARE 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"          CIB

LOAD"          LOAD" SD_TEST.4TH" loads file SD_TEST.4TH to FASTFORTH.
CIB            leave on stack address of CIB (Current Input Terminal), by default: TIB.
```

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
```

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)
SD_EMIT        sends output stream at 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
```

NONAME_ADD-ON

```
:NONAME        CODENNM

CODENNM        assembly counterpart of :NONAME
```

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

FIXPOINT

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

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

S>F            u/n -- Qlo Qhi       convert u/n in a s15.16 value
F.             display a s15.16 value
F*             s15.16 multiplication
F#S            Qlo Qhi u -- Qhi 0   convert fractionnal part of a s15.16 value displaying u digits
F/             s15.16 division
F-             s15.16 soustraction
F+             s15.16 addition
{FIXPOINT}     do nothing if compiled in core, else remove all FIXPOINT add-on.
```

ANS_COMPLEMENT

```
PAD            >IN            SOURCE         .(             (              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}
```

UTILITY

```
DUMP           U.R            WORDS          ?              .RS            .S             {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            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}     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\msp430\               bat 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' programing 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
      \ADD-ON


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
                    *.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\SciTEDirectory.properties    copy of \config\scite\AS_MSP430\SciTEDirectories.properties
      \config\gema\tiREGtoFastForth.pat         converts TI Rx registers to FORTH symbolic registers names
      \config\gema\RemoveComments.pat
      \config\gema\device.pat                   declarations for device
      \config\gema\launchpad.pat                declarations for target


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

drive:\config\msp430\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
                     prog(.bat)                            to flash target with target.txt file
                     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    to copy in any user folder for drag'n drop use
                     SendSourceFileToTarget.bat            to copy in any user folder for drag'n drop use
                     PreprocessSourceFile.bat              to copy in any user folder for drag'n drop use
                     SelectTarget.bat                      called by them three to select target


Note: all actions made from SciTE editor are 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\MSP430Flasher**

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

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

download http://www.scintilla.org/Sc411.exe to **drive:\prog\wscite**
then rename Sc411.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):

drive:\prog	**TERATERM.ini**	
drive:\prog\gema	**gema.exe**	syntactic preprocessor
drive:\prog\MacroAssemblerAS\bin	**asw.exe** **P2hex.exe** **as.msg** **cmdarg.msg** **ioerrs.msg** **P2hex.msg** **tools.msg**	macro assembler linker
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

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.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

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
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 /config/gema/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