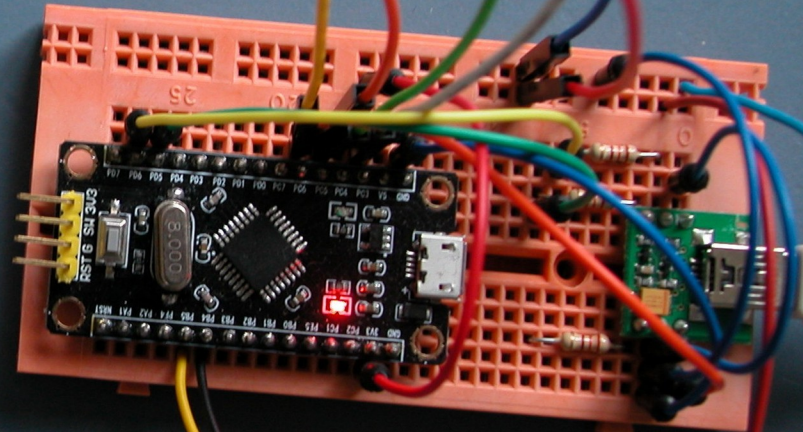
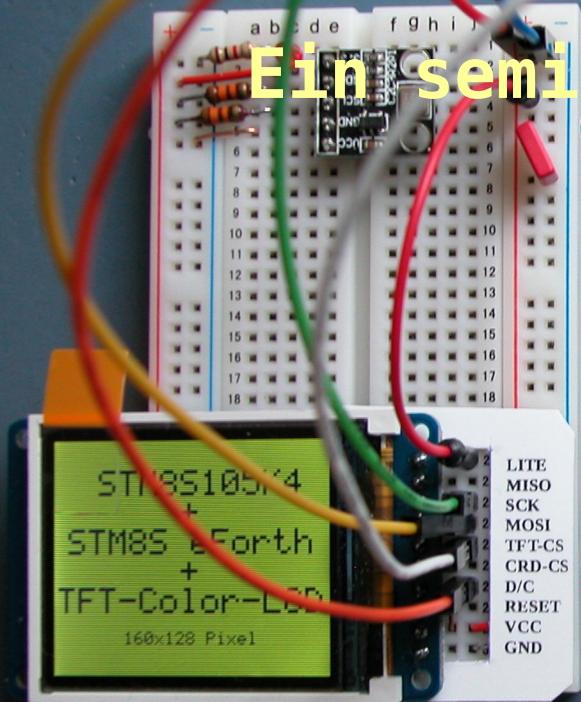


STM8 eForth

Ein semi-tethered Forth ?



Manfred Mahlow Forth-Tagung 2018

(Kontakt: Vorname PUNKT Nachname AT forth-ev.de)

STM8 eForth (stm8ef)

Ein Software-Entwicklungsprojekt auf GitHub

- <https://github.com/TG9541/stm8ef>

„TG9541/STM8EF extends [Dr. C.H. Ting's eForth for the STM8S Discovery](#) into a lightweight self-hosted ("untethered") Forth system for STM8 μ Cs with a maximum "feature-to-binary-size" ratio. It provides a plug-in system for board support, base dictionary configuration, a library infrastructure, and uCsim based binary level simulation. With the kind permission of the original author TG9541/STM8EF has a permissive [FOSS license](#).“

Umfangreiche Dokumentation im Wiki

- <https://github.com/TG9541/stm8ef/wiki>

STM8 eForth (stm8ef)

„Ready to run“ Images unter

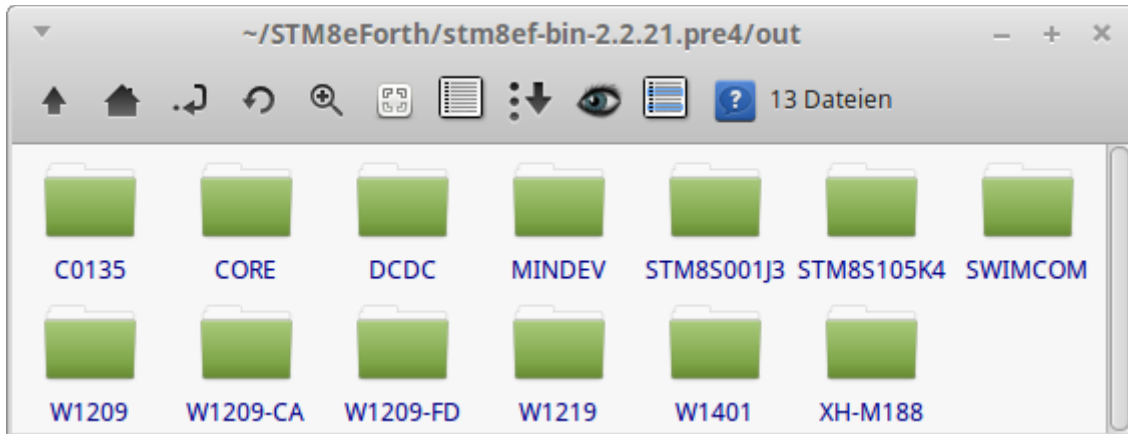
- <https://github.com/TG9541/stm8ef/releases>



- Native Code (STC) Forth
- Dictionary im RAM und Flash/NVM
 - gesucht wird erst im RAM, dann im Flash
 - kompiliert wird ins RAM oder ins Flash/NVM

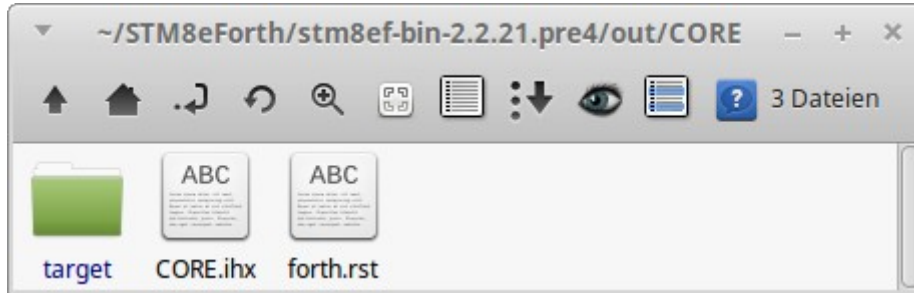
STM8 eForth (stm8ef)

Unterstützt div. Breakout Boards und „Chinese Gadgets“



STM8 eForth (stm8ef)

CORE.ihx : 3971 Bytes, 113 Worte mit und 84 Worte ohne „Header“



WORDS

```
IRET SAVEC RESET RAM NVM WORDS .S DUMP IMMEDIATE ALLOT VARIABLE CONSTANT
CREATE ] : ; OVERT ." AFT REPEAT WHILE ELSE THEN IF AGAIN UNTIL BEGIN NEXT
FOR LITERAL C, , ' CR [ \ ( . ( ? . U. TYPE SPACE KEY DECIMAL HEX FILL CMOVE
HERE +! PICK 0= ABS NEGATE NOT 1+ 1- 2+ 2- 2* 2/ */ */MOD M* * UM* / MOD
/MOD M/MOD UM/MOD WITHIN MIN MAX < U< = DNEGATE 2DUP ROT ?DUP BL BASE - 0<
OR AND XOR + UM+ I OVER SWAP DUP 2DROP DROP NIP >R R@ R> C! C@ ! @ 2@ 2!
EXIT EXECUTE EMIT ?KEY 'BOOT COLD ok
```

STM8 eForth (stm8ef)

Ein klassisches „untethered“ eForth ?

Nicht ganz!

- „Header“ für die „header-less words“ können „on-demand“ aus dem Verzeichnis **target** ins RAM (temporär) oder ins Flash/NVM (permanent) nachgeladen werden.

```
ls target
<#      ABORT"      $COMPILE  doVAR      $INTERPRET  NVMCP      SAME?      UPP
$, "    ACCEPT      CONTEXT   DOXCODE    '?KEY       PACK$      SIGN       U.R
$"      AHEAD        COUNT     'EMIT     kTAP        PAD        ?STACK    VAR
$"|     BASE        CP        EVAL      LAST        parse     str        WIPE
#       branch     CTOP      'EVAL     LOCKF       PARSE     TAP        WORD
#>     ?branch     DEPTH    EXTRACT   $,n         PRESET    #TIB      YTEMP
0       CALL,       DIGIT    find      NAME>       'PROMPT   TOKEN
1       >CHAR     DIGIT?   ^H        NAME?       QUERY     TX!
-1      COMPILE   dm+      HLD       next        QUIT      _TYPE
abort"  COMPILE?   do$      HOLD     NUMBER?     ?RX       ULOCKF
ABORT   [COMPILE]  docon    >IN     NVMCONTEXT  #S        ?UNIQUE
```

- Erforderlich ist dafür ein Terminal, das - wie e4thcom - bedingtes Hochladen von Quelltextdateien unterstützt.

```

STM8 eForth CORE.ihx im e4thcom Terminal
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
#ls target
<#      ABORT"      $COMPILE doVAR      $INTERPRET  NVMCP      SAME?      UPP
$, "    ACCEPT      CONTEXT DOXCODE  '?KEY      PACK$      SIGN       U.R
$"      AHEAD        COUNT   'EMIT     KTAP       PAD        ?STACK    VAR
$"|     BASE        CP      EVAL      LAST       parse     str       WIPE
#       branch     CTOP    'EVAL     LOCKF      PARSE     TAP       WORD
#>     ?branch     DEPTH   EXTRACT  $,n        PRESET    #TIB     YTEMP
0      CALL,       DIGIT   find     NAME>      'PROMPT   TOKEN
1      >CHAR      DIGIT?  ^H       NAME?      QUERY     TX!
-1     COMPILER   dm+     HLD      next       QUIT      _TYPE
abort" COMPILER?   do$     HOLD     NUMBER?    ?RX       _UNLOCKF
ABORT  [COMPILE] docon   >IN     NVMCONTEXT #S        ?UNIQUE

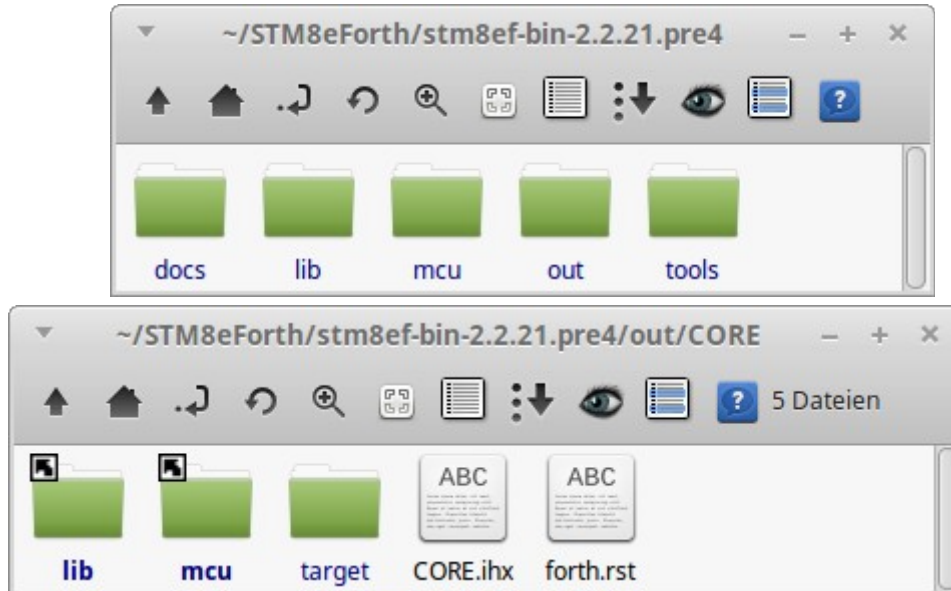
#r DEPTH Uploading: ./target/DEPTH
: DEPTH [ $CC C, $85A9 , OVERT ok
Closing: ./target/DEPTH ok
WORDS
DEPTH IRET SAVEC RESET RAM NVM WORDS .S DUMP IMMEDIATE ALLOT VARIABLE CONSTANT
CREATE ] : ; OVERT ." AFT REPEAT WHILE ELSE THEN IF AGAIN UNTIL BEGIN NEXT FOR
LITERAL C, , ' CR [ \ ( . ( ? . U. TYPE SPACE KEY DECIMAL HEX FILL CMOVE HERE +!

```

STM8 eForth (stm8ef)

unterstützt e4thcom

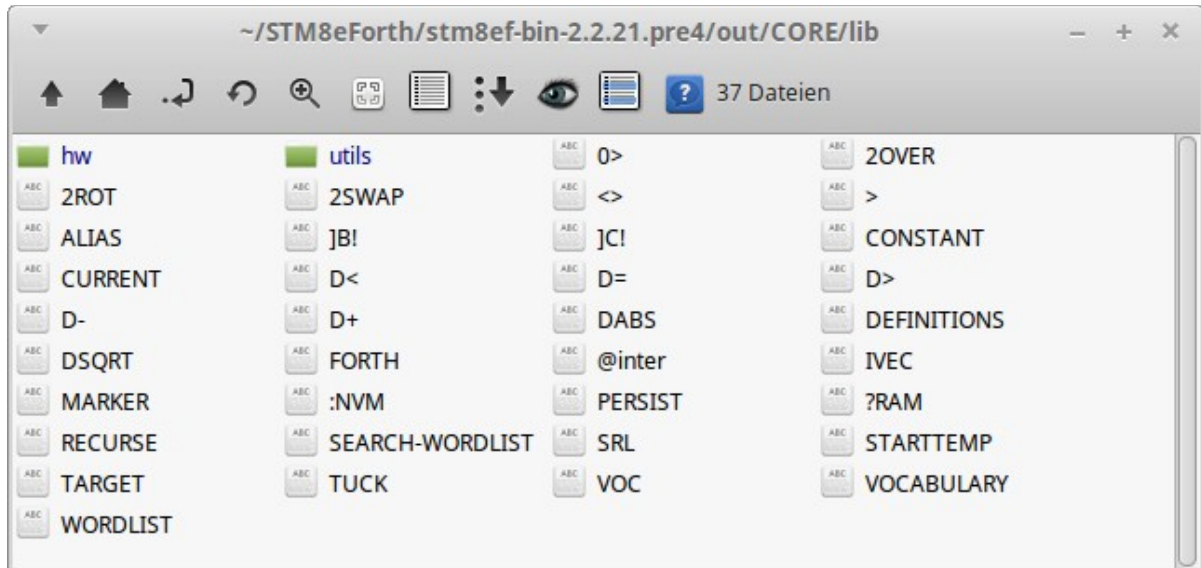
- Quelltext-Dateien für weitere Worte und Bibliotheken werden e4thcom-konform in den Verzeichnissen lib und mcu bereitgestellt.



STM8 eForth (stm8ef)

und e4thcom

- Quelltext-Dateien können mit `#require <name>` aus den Verzeichnissen **target**, **mcu** und **lib** ins RAM (temporär) oder ins Flash/NVM (permanent) geladen werden.



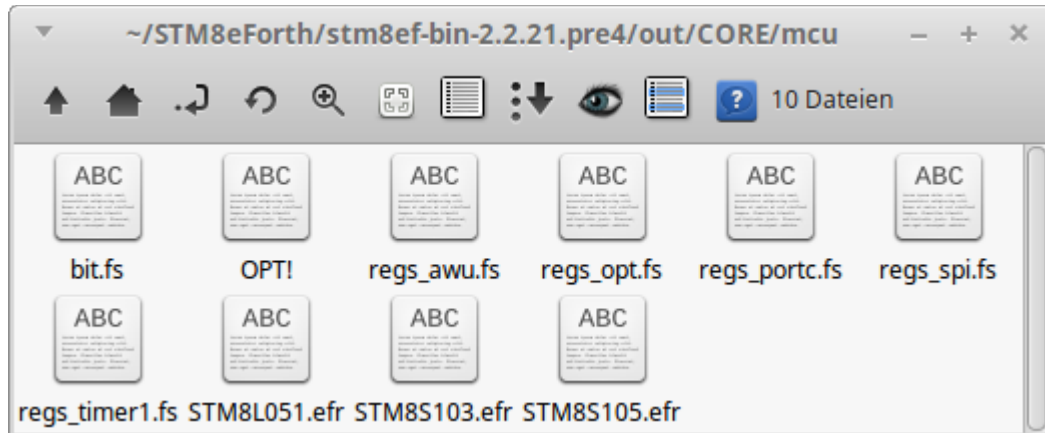
STM8 eForth (stm8ef)

und e4thcom

- MCU-spezifische Register- und Bit-Bezeichner sind in <mcu>.efr Dateien im Verzeichnis `mcu` definiert und können mit

```
\res MCU: <mcu> export <name1> <name2> ... <nameN>
```

ins RAM (temporär) oder ins Flash/NVM (permanent) kompiliert werden.



STM8 eForth (stm8ef)

RAM als Werkbank

Konstanten

- werden wie Zahlen als Literal kompiliert (3 Bytes)
- können im RAM nach Verwendung wieder gelöscht werden

```
RAM
  $5200 CONSTANT SPI_CR1   $5201 CONSTANT SPI_CR2
NVM
\ Init and enable SPI
: SPIon ( -- )
  $14 SPI_CR1 C! \ master, CLK/8, CPOL=CPHA=0, MSB first
  $01 SPI_CR2 C! \ no NSS, FD, no CRC
  $54 SPI_CR1 C! \ enable SPI
;

RAM WIPE
```

STM8 eForth (stm8ef)

RAM als Werkbank

Code im Flash und „Header“ im RAM

- „Header“ im RAM können nach Verwendung wieder gelöscht werden

```
RAM
  #require :NVM
  #require ALIAS
  #require LAST
  #require COMPILE
  #require [COMPILE]

:NVM R> @ CONTEXT ! ;RAM ALIAS dovoc ( "name" -- )

NVM
  : VOCABULARY ( "name" -- )
    : COMPILE dovoc LAST @ , [COMPILE] [ OVERT IMMEDIATE
  ;
RAM WIPE
```

STM8 eForth (stm8ef)

Background Task

Einfache INPUT-PROCESS-OUTPUT Steuerungsaufgaben

- können interaktiv als Background Task entwickelt und getestet werden
- die Background Task wird alle 5 ms ausgeführt

Beispiel:

```
\ blinky using the background task timer to flash all board outputs  
  
VARIABLE timemask  
  
$40 timemask !  
  
: blinky timemask TIM OVER AND = OUT! ;  
  
' blinky BG ! \ set background word, start flashing  
  
$80 timemask ! \ interactive: slower flashing
```

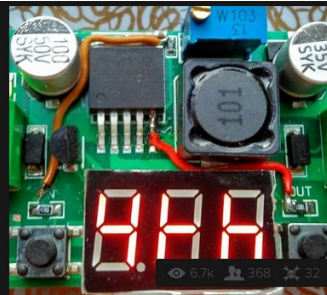
STM8 eForth (stm8ef)

auf hackaday.com



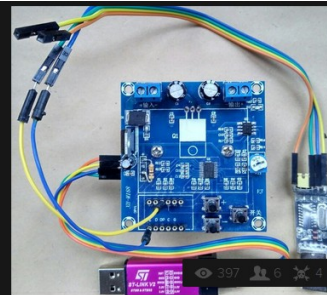
eForth for cheap STM8S gadgets

Thomas



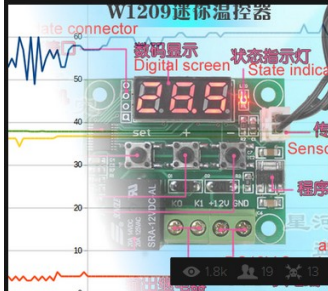
Low-Cost Programmable Power Sup...

Thomas



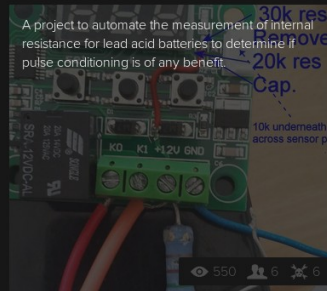
XH-M188 Programmable Linear Po...

Thomas



W1209 Data Logging Thermostat

Thomas



Internal Resistance of Lead Acid Bat...

richard