

430eForth: Tips, Tools & Tests

430eForth in der Linux-Umgebung

430eForth Reset-Falle

Etwa mehr Komfort beim Flashen

Entschärfen/Vermeiden der Reset-Falle

430eForth & Co. - Ein Vergleich

430eForth v4.3

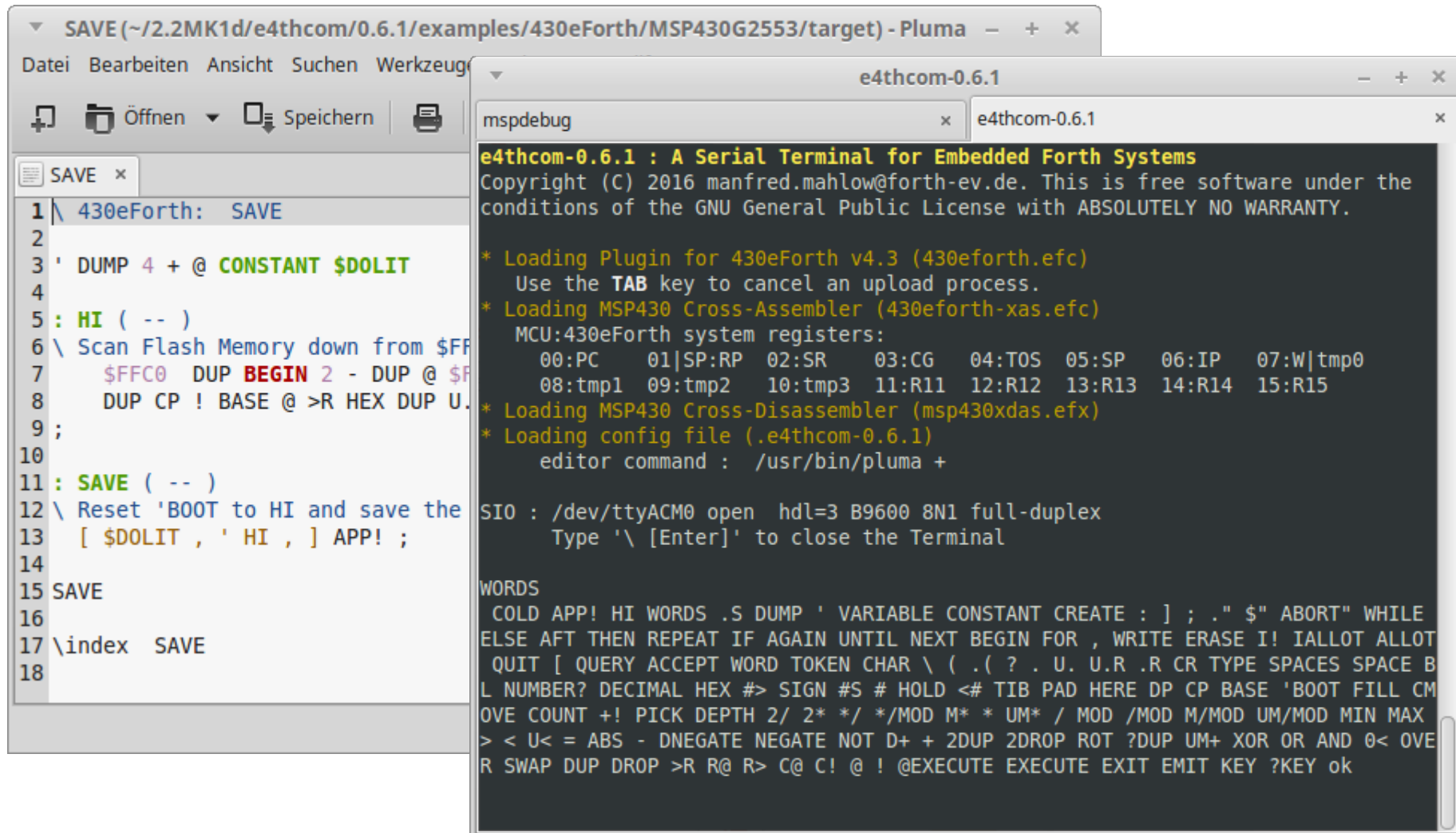
Chen-Hanson Ting: *The memory is reduced to 4136 bytes in the direct thread model. This is the smallest working Forth system with a text interpreter, a command compiler, and some debugging tools for any microcontroller, ever!*

Manfred Mahlow Forth-Tagung 2017

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

430eForth in der Linux-Umgebung

MSPDebug + Terminal(e4thcom-0.6.x) + Editor



The screenshot displays a Linux desktop environment with two windows. The background window is a code editor titled 'SAVE (~/.2MK1d/e4thcom/0.6.1/examples/430eForth/MSP430G2553/target) - Pluma'. The foreground window is a terminal titled 'e4thcom-0.6.1' with two tabs: 'mspdebug' and 'e4thcom-0.6.1'. The terminal shows the startup sequence of the 430eForth system, including loading plugins and displaying system registers.

```
SAVE (~/.2MK1d/e4thcom/0.6.1/examples/430eForth/MSP430G2553/target) - Pluma
Datei Bearbeiten Ansicht Suchen Werkzeuge
Öffnen Speichern
SAVE x
1 \ 430eForth: SAVE
2
3 ' DUMP 4 + @ CONSTANT $DOLIT
4
5 : HI ( -- )
6 \ Scan Flash Memory down from $FF
7   $FFC0 DUP BEGIN 2 - DUP @ $F
8   DUP CP ! BASE @ >R HEX DUP U.
9 ;
10
11 : SAVE ( -- )
12 \ Reset 'BOOT to HI and save the
13 [ $DOLIT , ' HI , ] APP! ;
14
15 SAVE
16
17 \index SAVE
18
```

```
e4thcom-0.6.1
mspdebug x e4thcom-0.6.1 x
e4thcom-0.6.1 : A Serial Terminal for Embedded Forth Systems
Copyright (C) 2016 manfred.mahlow@forth-ev.de. This is free software under the
conditions of the GNU General Public License with ABSOLUTELY NO WARRANTY.

* Loading Plugin for 430eForth v4.3 (430eforth.efc)
  Use the TAB key to cancel an upload process.
* Loading MSP430 Cross-Assembler (430eforth-xas.efc)
  MCU:430eForth system registers:
    00:PC  01:SP:RP  02:SR  03:CG  04:TOS  05:SP  06:IP  07:W|tmp0
    08:tmp1 09:tmp2 10:tmp3 11:R11 12:R12 13:R13 14:R14 15:R15
* Loading MSP430 Cross-Disassembler (msp430xdas.efx)
* Loading config file (.e4thcom-0.6.1)
  editor command : /usr/bin/pluma +

SIO : /dev/ttyACM0 open hdl=3 B9600 8N1 full-duplex
Type '\ [Enter]' to close the Terminal

WORDS
COLD APP! HI WORDS .S DUMP ' VARIABLE CONSTANT CREATE : ] ; ." $" ABORT" WHILE
ELSE AFT THEN REPEAT IF AGAIN UNTIL NEXT BEGIN FOR , WRITE ERASE I! IALLOT ALLOT
QUIT [ QUERY ACCEPT WORD TOKEN CHAR \ ( . ( ? . U. U.R .R CR TYPE SPACES SPACE B
L NUMBER? DECIMAL HEX #> SIGN #S # HOLD <# TIB PAD HERE DP CP BASE 'BOOT FILL CM
OVE COUNT +! PICK DEPTH 2/ 2* */ */MOD M* * UM* / MOD /MOD M/MOD UM/MOD MIN MAX
> < U< = ABS - DNEGATE NEGATE NOT D+ + 2DUP 2DROP ROT ?DUP UM+ XOR OR AND 0< OVE
R SWAP DUP DROP >R R@ R> C@ C! @ ! @EXECUTE EXECUTE EXIT EMIT KEY ?KEY ok
```

Die 430eForth Reset-Falle

Durch **COLD** oder **Reset** wird der Code-Pointer ungültig, sofern der Systemzustand vorher nicht mit ' **HI APP!** gespeichert wurde.

Die Folge:

- Nachfolgendes Compilieren führt ohne Fehlermeldung zu unerwarteten Ergebnissen, z.B. Namensfragmente im Wörterbuch, Crash bei deren Aufruf.
- Irritierend für Anfänger und für alle, die Zen of LaunchPad nicht gelesen haben.
- Fehlerbehebung ist nur durch erneutes Flashen der MCU möglich.
- Das Flashen dauert zwar nur ca. 8 Sekunden, wird aber schnell lästig.

Etwas mehr Komfort beim Flashen

- Schließt man eine bestehende Terminalverbindung vor dem Flashen nicht, ist diese danach weiter funktionsfähig.
(Getestet mit Linux Mint, mspdebug und verschiedenen Terminalprogrammen.)
- Das Flashen lässt sich mit einem Skript vereinfachen, das unter Linux z. B. wie folgt aussehen kann:

```
#!/bin/bash
#
# flash-2553lp : Script to program a MSP430G2553 Launchpad          MM-161230
IMAGE="430eforth.a43"
echo -ne "\033[1mMSPDebug: Programing MSP430 LaunchPad with $IMAGE ?\033[0m"
read -n 1 -p " [N/y] " REPLY ; echo
if [ ! "$REPLY" == "y" ] ; then exit ; fi
mspdebug rf2500 'erase all' "prog $IMAGE" "verify $IMAGE"
```

Entschärfen der Reset-Falle

Ein TOP-DOWN Flash-Speicher-Scan beim Booten setzt den Code-Pointer CP auf die niedrigste noch freie Speicherzelle:

Quelle: e4thcom-0.6.1/templates/430eForth/MSP430G2553/target/SAVE

```
1 | 430eForth: SAVE MM-170206
2
3 | ' DUMP 4 + @ CONSTANT $DOLIT
4
5 | : HI ( -- )
6 | \ Scan Flash Memory down from $FFC0 and set CP to the lowest free memory cell.
7 |   $FFC0 DUP BEGIN 2 - DUP @ $FFFF - UNTIL 2 + ( a0 a1 )
8 |   DUP CP ! BASE @ >R HEX DUP U. ." : " - DECIMAL U. R> BASE ! HI
9 | ;
10
11 | : SAVE ( -- )
12 | \ Reset 'BOOT to HI and save the user variables to the cold start data area.
13 | [ $DOLIT , ' HI , ] APP! ;
14
15 | SAVE
16
17 | \index SAVE
18
```

- Ein Tool mit minimaler Codegröße (74 Bytes).
- Beschriebener Flash-Speicher wird nicht gelöscht sondern ausgeblendet.
- Inhalte die mit SAVE gesichert wurden, werden nicht ausgeblendet.
- Ist der Flash-Speicher voll, muss 430eForth neu geflashed werden.

Vermeiden der Reset-Falle

Ein Flash-Management-Tool löscht beim Booten die Speicherzellen, deren Inhalt nicht durch SAVE gesichert wurde:

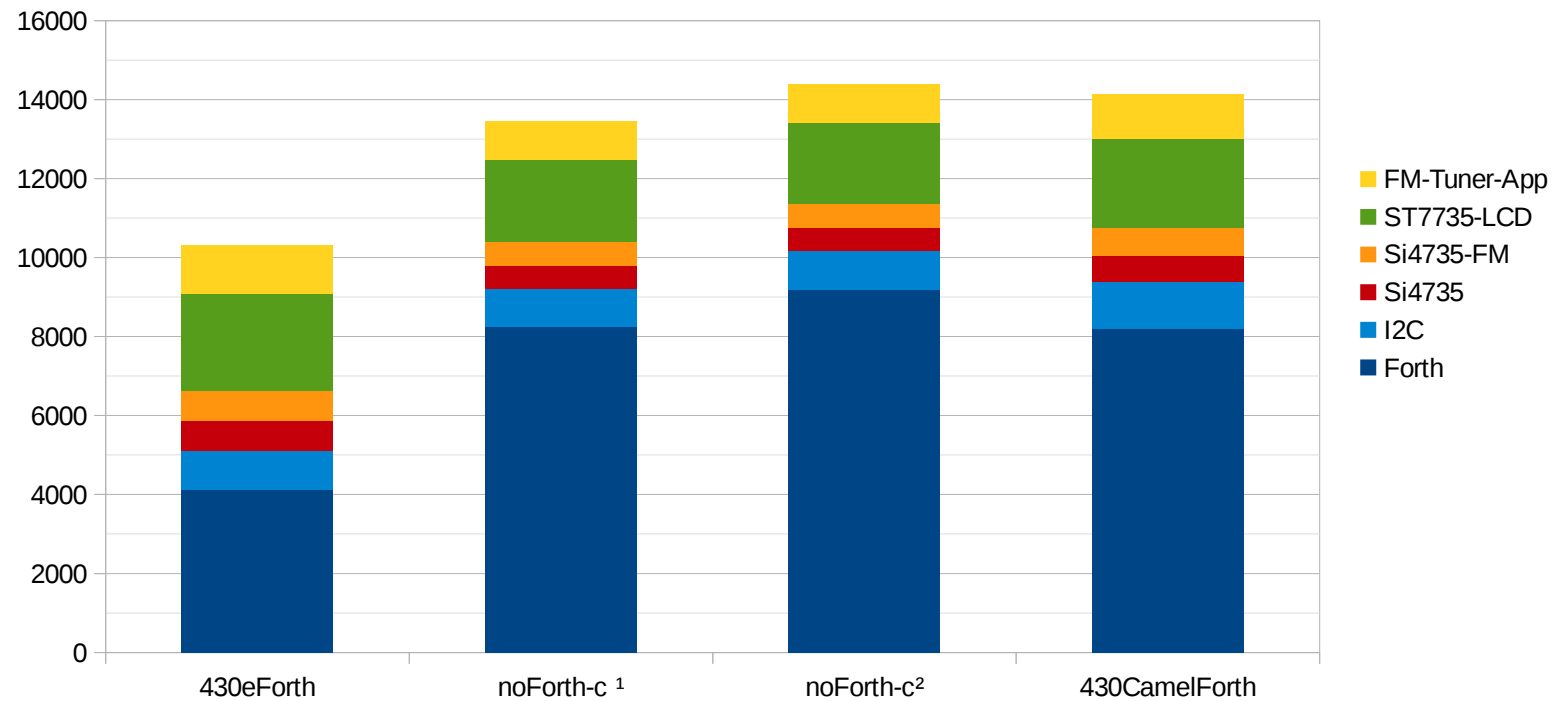
Quelle: e4thcom-0.6.1/templates/430eForth/MSP430G2553/target/eForth

```
\ 430eForth : eForth SAVE COLD SHIELD MM-170127
\
\ A Flash Memory Management Tool for 430eForth that makes program development
\ much more comfortable. Adds the words eForth, SAVE and SHIELD to the eForth
\ kernel and extends COLD.
\
\ Requires:  * 430eForth v4.3 on a MSP430G2553 MCU
\            * 442/560 Bytes of Flash Memory (without/with) SHIELD
\
\ Recommends: * e4thcom-Terminal 0.6.x or later with 430eforth plug-in
\
\ eForth ( -- ) Reset DP , CONTEXT , LAST , CP and 'BOOT to initial values that
\ were defined when loading this extension. Erase the Flash Memory
\ from CP to $FP and execute SAVE.
\
\ SAVE ( -- ) Save the current eForth state ( DP , CONTEXT , LAST , CP and
\ 'BOOT ... ) to the Flash Information Memory.
\ Reset 'BOOT if it is pointing to an invalid address ( >= CP ).
\
\ COLD ( -- ) Restore the last SAVED eForth state ( DP , CONTEXT , LAST , CP ,
\ 'BOOT ... ) from Flash Information Memory and erase the Flash
\ Memory from CP to $FP.
\
\ SHIELD ( "name" -- ) Create a word "name" and execute SAVE. "name" restores
\ the saved state when executed.
\
\ After loading this code, the free Flash Memory starts at the next page-
\ aligned address ( $D200 with a newly flashed MCU ) and ends at $FP=$FC00.
\
```

430eForth & Co. - Ein Vergleich

Si4735 FM-Tuner mit ST7735 Color Graphic Display

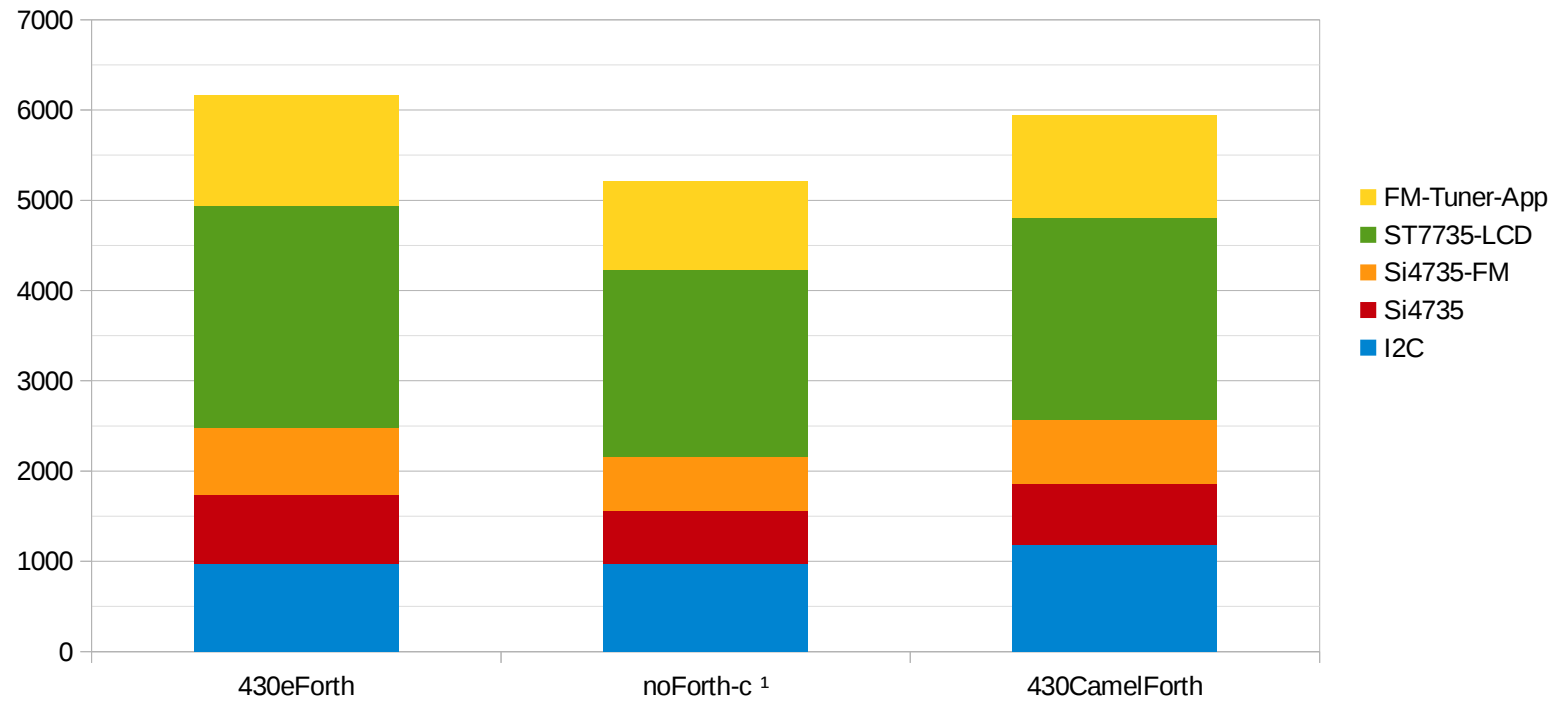
MSP430G2 Lauchpad : Applikationsgröße einschließlich Forth-Kern



¹⁾ ohne Tools , ²⁾ mit Tools

Si4735 FM-Tuner mit ST7735 Color Graphic Display

MSP430G2 LaunchPad : Applikationsgröße ohne Forth-Kern



¹⁾ ohne Tools