



De bijeenkomst van zaterdag 14 augustus 2021

Deze is na lange tijd weer in Bilthoven van 11 tot 15 uur

Het programma

Wegens vakanties, is er een ander programma dan gewoonlijk.

Er is gelegenheid om te laten zien waar je aan werkt met forth en elektronica of om andere forthers hier over te spreken. Het gaat dus om onderlinge kennis-uitwisseling.

Internationale Zoom sessie Forth2020

Er is een internationale Forth bijeenkomst via Zoom op de tweede zaterdag van elke maand. Een vorige keer was het onderwerp libraries een verhit punt van discussie. Er zijn meerdere presentaties van elk een half uur in het Engels. De Zoom sessie wordt gehost door Peter Forth in Brazilië en Ulrich Hoffmann in Duitsland. Opdat de sessie over de hele wereld op een handige tijd gevolgd kan worden, is voor Nederland 16:00 de begintijd. De oude sessies zijn ook nog op youtube te bekijken. Zie het trefwoord **FORTH2020** .

Vernieuwing van de Egelpagina

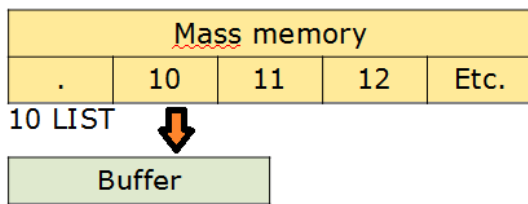
De laatste toevoeging gaat over het gebruik van de aloude blokken waarmee Forth vanaf 1979 werkte. Forth79, FigForth en Forth83 hadden blokken van 1 kB in plaats van de files die we nu gebruiken.

Willem Ouwerkerk en Jeroen Hoekstra hebben hieraan gewerkt.

Als de beamer aanwezig is (Albert is met vakantie) kan het gedemonstreerd worden op een MSP430FR5994 launchpad met 256 kB FRAM.

xx Blocks

The classic Forth blocks, with editor, using extended memory words



0	\ Screen 10 with INDEX show first line of screens
1	v: also forth
2	: INDEX (n1 n2 --)
3	1+ swap ?do
4	cr i 2 .r space
5	i block #line -trailing type
6	loop ;
7	v: previous
8	
9	
10	
11	
12	
13	
14	
15	

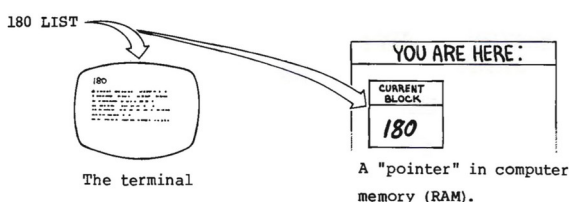
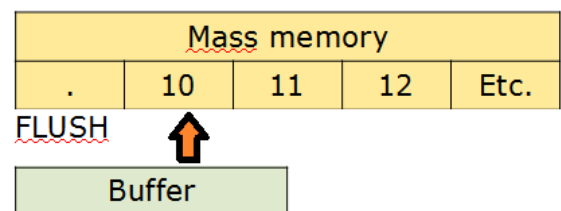
In the early days of Forth, source-code was almost always stored and edited in blocks. Of course, this has now all been succeeded by file based tools. But if you ever longed for the feeling of using a block-based system, this is your chance. Using blocks connects you directly to the great Forth-coders of the past. They all worked with blocks and developed great software with them.

Blocks are also a very good way of showing string-handling. It is almost unbelievable how little code is needed to get a full block management system, including a block-editor,

running. It takes a bit of time and effort to get used to it. But what is a bit of time if you get transported all the way back to the seventies!!

Overview:

In this example the FRAM memory of the MSP430 is used for the storage of the blocks. The advantages of FRAM are very evident here: very fast, permanent and can be erased and rewritten as often as you want.



A MSP430FR5994 has 256 KB FRAM, from 0x4000 till 0x43FFF. A MSP430FR5969 has 63 KB FRAM, from 0x4400 till 0x13FFF. The memory above 0xFFFF can be reached with the noForth words:

Forth commands:

```
X!    ( n d -- ) store n @ address d (double)
X@    ( d -- n ) fetch data from address d (double)
XC!   ( c d -- ) store char @ address d (double)
XC@   ( d -- c ) fetch char from address d (double)
```

The memory above 0xFFFF is used for the storage of the blocks. On the MSP430FR5994 there is space for 208 blocks, which is ample for some very serious programming. On a MSP43FR5969 there is space for 16 blocks, enough to experiment.

Use of the block-system and editor

Load the program (not yet published) to get the block file system and the editor.

Activate the editor by typing "n EDIT".

very short quick-start on using the editor:

- activate the editor by typing "n EDIT".
- When it is unused, first "CLEAR" the selected block.
- Than add some text:
0 P (Hello there)
2 P : Hello ." World" ;
- Use "FLUSH" to save the block
- Use "n LOAD" to load the source-code on your block/block

The Forth commands:

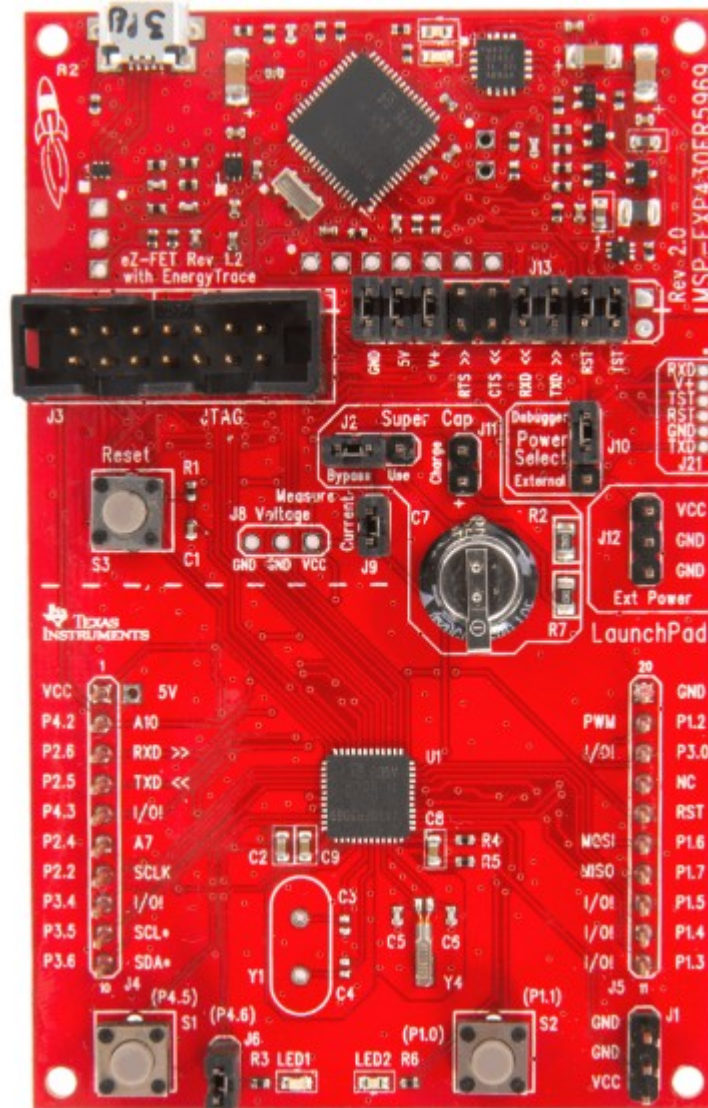
\ Read & write blocks

```
UPDATE      ( -- )   Mark current screen as changed
FLUSH       ( -- )   Write UPDATED buffers to (RAM)disk & erase them
EMPTY-BUFFERS ( -- )   Erase all buffers
SAVE-BUFFERS ( -- )   Write changed buffer(s) to (RAM)disk
BUFFER      ( +n -- a ) Write changed buffer(s) to (RAM)disk
BLOCK       ( +n -- a ) Perform buffer & read block +n to the buffer
                    when it was not already present
```

\ Editor words overview

```
EDITOR ( -- )           Activate editor vocabulary
```

EDIT	(+n -)	Start editing screen +n
LOAD	(+n --)	Load screen +n
THRU	(+n1 +n2 --)	Load screen +n1 to screen +n2
LIST	(+n --)	List and select screen +n for editing
CLEAR	(--)	Fill active screen with blanks
COPY	(+n1 +n2 --)	Copy screen +n1 to screen +n2
LL	(+n --)	Clear screen & perform LIST
L	(--)	Clear screen & (re)display active screen
N	(--)	Display & select next screen for editing
B	(--)	Display & select previous screen for editing
P	(+n "ccc" --)	Place text "ccc" at line +n
H	(+n --)	Hold a copy of line +n
E	(+n --)	Erase line +n
S	(+n --)	Insert (spread) empty line at +n
R	(+n --)	Replace line +n with copy
I	(+n --)	Insert copy at line +n
D	(+n --)	Copy line +n & delete it too
Q	(--)	Quit (close) editor



MSP430FR5969 launchpad met 63 kB FRAM for 16 blocks

Example for calculation of the greatest common divisor (GCD) of two numbers.

This screen appeared in Forth Dimensions volume II number 6 page 167.

```
0 ( Greatest common divisor, a demo      WFR-79DEC09 )
1 : G-C-D
2   OVER OVER <
3   IF SWAP THEN          ( use larger as quotient )
4   BEGIN SWAP OVER      ( save divisor third )
5     /MOD OVER          ( test remainder zero )
6     WHILE ( not zero ) DROP ( this dividend )
7   REPEAT
8   DROP DROP CR ." The 'G-C-D' is " . ;
9 CR ." Input two numbers, then execute 'G-C-D'. The greatest"
10  ." common divisor of these numbers will be displayed."
11 CR
12
13
14 ;S
15
```

This program can also easily be used for Forths with normal files.

Ook iets te melden?

Stuur uw ideeën, tips, programma's of projecten naar de redactie, zodat anderen daar kennis van kunnen nemen.

Bijdragen liefst per E-mail, Uiterlijk 1 week voor de bijeenkomst, naar f.l.van.der.markt@kader.hcc.nl

Onze website is veranderd, maar de URL is hetzelfde gebleven.
Website van HCC!Forth:

<https://forth.hcc.nl>

