Bibliotheek voor noForth (t)

Wat is een bibliotheek voor Forth. In ons geval een grote verzameling kleine en grotere programma's. Deze zijn allen samengevoegd in een groot bestand dat op een toegankelijke plek wordt opgeslagen.

Door de tegen stribbelende USB-driver, wilde ik van mijn *noForth t* voorbeelden een bibliotheek maken. Geïnspireerd door *WANT* kwam ik tot onderstaande:

- Eenvoudige structuur (idee A.N.)
- Snelle zoek functie
- Uitbreidbaar: `OPEN-LIB LIB-SECTION CLOSE-LIB WIPE-LIB`
- Eenvoudig te gebruiken: `NEED NEEDED .LIB`

Hoe zit het bibliotheek bestand in elkaar? Elke cel in het plaatje is één byte, in het geheugen gebied waar de bibliotheek opgeslagen is.

0D	09	\	BL	N	A	A	M	1	BL
E	T	C	.	0D	F	O	R	T	H
BL	S	O	U	R	C	E	BL	C	O
D	E	BL	0D	09	\	BL	N	A	A
M	2	BL	E	T	C	.	0D	M	E
E	R	BL	C	O	D	E	0D	09	FF
FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

De interne functies

SKIP	(a2 a1 ch -- a2 a3)	Sla char. ch over
SCAN	(a2 a1 ch -- a2 a3)	Zoek char. ch
LIB-FIND	(a +n -- sa)	Zoek string a +n
LIB-REFILL	(src-id -- f)	Lees code regel in
LIB-LOAD	(sa --)	Laad code sectie

Het werkpaard

```
  1008,1000 constant LIB     \ Library start address
  LIB value LIBHERE          \ Current end address of library

: LIB-FIND  ( a +n -- sa )   \ a,+n=name, sa=begin-source-section
    >fhere  count upper      \ Place as counted string a,b at FP
    libhere  lib             \ Library address range
    begin
        09 scan  1+          \ Find library section, skip 09
        2dup = ?abort 2dup   \ Nothing found?
        0D scan nip over 1+ \ Get line with keywords, skip \
        begin
        2dup > while         \ Line not done?
            bl skip dup >r   \ Skip leading spaces
            bl scan          \ After keyword
            r@  over r> -    \ Keyword length
            FP count s<> 0= \ Keyword found, leave source address
            if  2drop nip exit
            then
        repeat
        2drop
    again ;
```

Bibliotheek source code

```
wipe-lib

open-lib

lib-section ROLL
v: forth definitions
: ROLL  ( i*x u -- j*x )    \ Roll item u to TOS
    dup 1 <
    if      drop
    else    swap >r  1- RECURSE  r> swap
    then ;
%%

lib-section VERSION VERSIE VSN
cr .(    NEED version 0.31     )
cr .(   Library version 0.20   )
cr .( Size is about 168 kBytes )
%%

lib-section  PIN  SQUERY
( GPIO -- ) \ Change GPIO pin for S?
need [IF]
dup dm 29 > [if]  drop  dm 24  [then]
0  swap  b+b   1 cfg ! \ Use UART-0 & GPIO-xx for S?
4 cfg @ abs 4 cfg !    \ Make sure to (re)start
config  cr .( Test S? ) s? .
%%

close-lib
```

Voorbeelden

.LIB (--)	List all library headers in 2 columns
NEED (“name” --)	Add “name” to noForth t ...
NEEDED (a +n --)	Add word name a +n to noForth t

Met .LIB krijgen we een overzicht van de opgenomen programma stukjes. We laden eerst de functie VIEW erbij, om de bibliotheek source te kunnen bestuderen.

```
.LIB

NEED view
VIEW view

Willen we de assembler gebruiken:
NEED asm\

Om b.v. commacode woorden te genereren:
NEED commacode

NEED variant met een string op de stack:
S” .CFG” NEEDED

NEED allwords
ch d chwords
4 lenwords
```

Etc.

Toetje voor AvdH

```
: WANT  ( "ccc" "etc." -- ) \ Add all the words
    begin
    bl parse ?dup while  \ Next string not zero?
       needed            \ Yes, add it
    repeat  drop ;       \ No, clear stack
```