



## Programma HCC!Forth komende zaterdag

**Zaterdag 14 juni 2014 op de bekende locatie:  
gebouwtje naast de Zuiderkapel aan de Boslaan 1 in Bilthoven ( zie foto )**

- 10:30 Zaal open en koffie voor vroege vogels
- 11:00 Betere aansturing Märklin motor met Forth?, door Cees Nobel
- 11:15 Voortgang werkgroepen, door Albert van der Horst
- 11:25 ciforth als filter, de unix-methode om programma's aaneen te rijgen,  
door Albert van der Horst
- 12:30 Pauze
- 13:00 De experimentele software van de HEXAPOD uitgelegd,  
door Willem Ouwerkerk
- 15:00 Sluiting.

**Tot ziens in Bilthoven!**



Zuiderkapel Bilthoven

## **Betere aansturing Märklin motor met forth?**

Cees Nobel

Op de komende Forth dag wil ik iets laten zien en voorleggen aan de leden en hun commentaar er over meenemen voor de verdere ontwikkeling.  
Het zal maar 10 minuutjes inbeslag nemen, het gaat om:

Een wisselstroom motor "MARKLIN" door de launchpad laten accelereren/decelereren en CW/CCW draaien.

Op een houtje-touwtje methode, wat met Forth beter moet gaan??

## **Filters met ciForth**

Albert van der Horst

Het model dat een Forther heeft om een computer wat te laten doen is een terminal waarin opdrachten getikt worden. Dit werkt met woorden, die aaneengeregen worden tot zinnen. Hiermee kunnen vrij moeilijke opdrachten aan de computer worden gegeven. De computer snapt hele zinnen! De Forther respecteert zijn computer en probeert er op een menselijke manier mee te praten, en veelal begrijpt de computer dat.

Hoe anders behandelt de Windowser zijn computer! Hij kan alleen voorgebakken opdrachten uitvoeren die als zg. iconen op zijn computerscherm zijn weergegeven. Dit is als het roepen van commando's "apport", "zit", "doodliggen" naar een hond. Denk niet dat de hond de commando's begrijpt. Het zijn geconditioneerde reflexen, die er met een langdurige training in gedruild zijn. (Ik bedoel niet denigrerend te doen over de hond. Het apport commando is erg moeilijk voor een robot.)

Filters komen in zicht wanneer we Forth willen gebruiken in een opdrachtzin als Forth gecombineerd moet worden met andere programma's.  
Bijvoorbeeld om te weten hoeveel woorden er in een bestand staan, gebruikt men het programma 'wc'. Als je nu wil weten hoeveel woorden een Forth heeft kan je zeggen:

```
echo WORDS | wina | wc -w
```

Dit is een voorbeeld waarin wina (een Forth) als filter gebruikt wordt. De uitvoer van "echo WORDS" (de string "WORDS") wordt door de FORTH geïnterpreteerd als commando en er komt een stroom woorden uit: "ONLY ENVIRONMENT NOOP LIT EXECUTE RECURSE ... ". Het programma wc telt dan deze woorden en het totaalresultaat is:  
351

Een essentieel aspect van filters is dat zij een standalone programma zijn, dus een binaire executable naast de Forth interpreter.  
In het kader van de overgang van chforth naar ciforth hebben Coos en ik de voorbeeldprogramma's van chforth bekeken.

Vele hiervan zijn filters, zoals het programma dat hoofdletters in kleine letters omzet. Beide Forthen hebben een TURNKEY commando, die zulke programma's genereert, maar in de praktijk komen filters er heel anders uit te zien.

Het unix model voor het aaneenrijgen van programma's wordt inmiddels universeel gebruikt. Aan de hand van het voorbeeld

```
echo WORDS | wina | wc -w
```

zien we een aantal aspecten hiervan:

- het begrip standaard invoer en standaard uitvoer. Voor een Forth is standaard invoer gewoon het toetsenbord en standaard uitvoer het scherm.
- het pipe symbool dat twee programma's koppelt. Dit betekent dat de standaard uitvoer (wat het eerste programma op het scherm wil zetten) doorgegeven wordt als invoer voor het tweede programma.
- je wilt argumenten. Dit zijn bijvoorbeeld filenamen of de -w die aan wc wordt doorgegeven.

<http://www.forth.hcc.nl/w/Producten/Ciforth>

Zaterdag 14 juni 2014 ga ik verder in op de hulpwoorden die nodig zijn om filters goed te laten werken, en op het inlezen van datastromen die via een pijp binnen komen.

## **De experimentele software van de HEXAPOD uitgelegd**

Willem Ouwerkerk

De experimentele HEXAPOD software bestaat uit de vijf hieronder genoemde lagen:

- a) I<sup>2</sup>C EEPROM routines met eigen geheugen management
- b) Genereren van pulsen voor de servomotoren
- c) Random generator voor het testen van de niveaus (b en d)
- d) PiliPlop routine van Albert Nijhof voor het genereren van testpatronen om de servomotoren te besturen.
- e) Poot besturingspatronen om met verschillende bewegingen te kunnen experimenteren, gebruikt alle software lagen.

Ik zal in detail laten zien hoe de interrupt routine en zijn datastructuur opgebouwd is.

```
\ First version of double interrupt routine to control 2 x 10 servos by using  
\ two timer interrupts. It trades RAM and free registers for processor cycles,  
\ that should be no problem because we have 2 kByte RAM on a MSP430F149.
```

```
hex
```

```
: VARIABLES      create cells allot  does> @ swap cells + ;
```

```
\ I/O-adres and bit-nr for 20 outputs
```

```
\ Separated here over P4, P5 and the first part of P3
```

```
create #PORTDATA      ( -- adr )
```

```
011D , 021D , 041D , 081D , 101D , 201D , 401D , 801D , ( P4 )
```

```
0131 , 0231 , 0431 , 0831 , 1031 , 2031 , 4031 , 8031 , ( P5 )
```

```
0119 , 0219 , 0419 , 0819 , align ( P3 )
```

```

( Tuned numbers for every MG90 servo, may be corrected )
ecreate #begin
  029E e, 029E e, 02D0 e,
  02EE e, 029E e, 029E e,
  029E e, 02E4 e, 029E e,
  029E e, 02BC e, 029E e,
  029E e, 0320 e, 029E e,
  029E e, 02BC e, 02F8 e,
  029E e, 029E e,

ecreate #end
  0988 e, 0988 e, 0988 e,
  0988 e, 094C e, 092E e,
  0988 e, 0988 e, 0988 e,
  0988 e, 0988 e, 094C e,
  0988 e, 0988 e, 0924 e,
  0898 e, 0988 e, 0988 e,
  08CA e, 08FC e,

028 variables SERVOS          \ Space for 20 times port-data & servo positions

: PULSE!      ( pulse s -- ) 2* servos cell+ ! ;
: SERVO       ( u s -- )     range >r r@ scale r> pulse! ;

\ The error margin in the generated pulses is maximal 4.5 microseconds
\ Exclusive registers: XX=addr pointer and ZZ=workregister
\ Code needs 36 cycles and uses 32 bytes code.
\ The CPU overhead is about 0.8% to 0.25% using pulses from .6 to 2 ms.
\ The servo repeat frequency changes from 45 Hz to 170 Hz, average 70 Hz.
\ This routine handles the pulses for the first ten servomotors.
code PULSEGEN1 ( -- )          \ 6 for timer-interrupt
  xx )+ zz .b mov              \ 2 - 1 Load previous port-addr
  xx )+ zz ) .b bic            \ 5 - 2 Clear previous servo bit
  #2 xx add                    \ 1 - 1 Skip pulse time
  014 servos # xx cmp          \ 2 - 2 Yes, halfway table check
  =? if,                       \ 2 - 1 Pointer after halfway SERVOS?
    0 servos # xx mov         \ 2 - 2 Yes, pointer to start of SERVOS
  then,
  xx )+ zz .b mov              \ 2 - 1 Load current port-addr
  xx )+ zz ) .b bis            \ 5 - 2 Set current servo bit
  xx )+ 0172 & mov             \ 5 - 2 Set current pulse time
  #4 xx sub                    \ 1 - 1 Correct pointer
  reti                         \ 5 - 1
end-code

\ End

```

Bit1	Port	Time	Bit2	Port	Time	Etc.														

Data in de WORKPLACE tabel waar de interrupt uit leest

## In memoriam Arie Kattenberg



Geheel onverwachts is ons ontvallen een Forther van het eerste uur, Arie Kattenberg. Hij leidde een project om meteorologische data van Indonesië te digitaliseren. Tijdens een afrondend bezoek aan Djakarta, na een lezing te hebben gegeven, is hij overleden aan een hartstilstand.

Hij heeft de de laatste paar bijeenkomsten bezocht, nu hij zich na zijn pensioen weer meer met Forth ging bezighouden. Daarnaast had hij het plan opgevat om ciforth te porten naar de ARM, en ik heb enige genoeglijke sessies met hem gehouden. Mijn beeld van Arie is dat van de meester, die mij met Forth in aanraking heeft gebracht. Nu had ik hem wat te leren, een vreemde ervaring.

Binnen de Forth gebruikersgroep was Arie vooral de man van de Werkgroep Metaforth in het begin van de tachtiger jaren. Enkele mensen hebben hun eigen Forth daarop gebaseerd, maar voor iedereen die deze werkgroep volgde was het een bron van inspiratie. Voor mij persoonlijk was het een studiegenoot van de lichting 1967 in Utrecht. Wij werkten o.a. samen aan de 68000 figForth en Forth op de NASCOM, een van de eerste Z80 computers.

Hoe kortstondig de hernieuwing van het contact ook was, het voelt als een groot gemis.

Albert van der Horst

## **Mailings**

Het HCC-mechanisme om mailings te verzenden is gewijzigd. In plaats van het eenvoudige POP-Bull is er nu een Content Management Systeem bij Tripolis, met flinke beperkingen in de samenstelling van de nieuwsbrief.

Daardoor is het niet langer mogelijk om deze PDF als bijlage te versturen naar alle leden van HCC!Forth. Als tussenoplossing zal deze mailing tevens op onze website te vinden zijn als link bij de agenda <http://www.forth.hcc.nl/w/Agenda/Agenda>.

Voor de leden die zich destijds met hun E-mailadres hebben aangemeld verandert er niets.

## **Iets te melden?**

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

Bijdragen liefst per E-mail, Uiterlijk 2 weken voor de bijeenkomst, naar [f.l.van.der.markt@kader.hcc.nl](mailto:f.l.van.der.markt@kader.hcc.nl)



**HCC!Forth**

website van HCC!Forth: <http://forth.hcc.nl>