

## Beste leden van hcc!forth interessegroep en in forth geïnteresseerde computeraars,



Er is een bijeenkomst van hcc!forth i.g.  
Op zaterdag 13 juni 2026  
Van 11.00 uur tot 15.00 uur  
Locatie Boslaan 1 Bilthoven  
Naast de Zuiderkapel

Zie ook de hcc!forth agenda-pagina op [forth.hcc.nl/agenda](https://forth.hcc.nl/agenda)

## Programma

- 10.30** – Zaal open en koffie voor vroege vogels
- 11.00** – Werkgroepen verslag (AvdH/WO)
- 11.15** – Opdracht: Tegels (AN)
- 12.15** – Pauze
- 13.00** – PIO en DMA (WO)
- 13.45** – CREATE DOES> (AvdH)
- 14.30** – O.K.U.
- 15.00** – Afsluiting

Omdat het programma goed gevuld is wordt er wat scherper op de tijd gelet



<https://forth-ev.de/wiki/en:pfw:welcome>

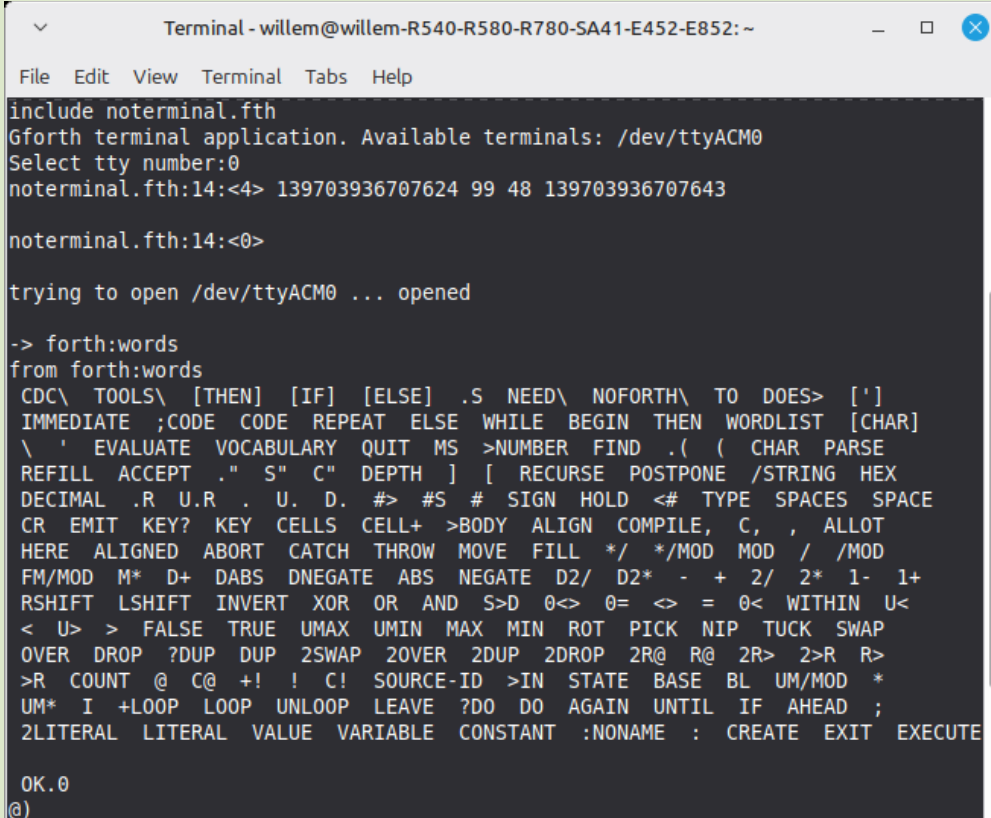
## Werkgroepen verslag

### Draadloos maas netwerk

We gaan nu stap voor stap door de hele source heen. Daarmee hopen we de code nog wat degelijker en waar mogelijk compacter te maken. De focus ligt nu op de zend routine >NODE

### Terminal emulator

We hebben twee schetsen voor een terminal emulator onder Linux en Windows. Beide geschreven in Gforth. Het doet al wat maar we zijn er nog lang niet. Een vraag aan AI leverde op dat 4TH van Hans Bezemer mogelijk een bruikbaar systeem ervoor is.



```

Terminal - willem@willem-R540-R580-R780-SA41-E452-E852: ~
File Edit View Terminal Tabs Help
include noterminal.fth
Gforth terminal application. Available terminals: /dev/ttyACM0
Select tty number:0
noterminal.fth:14:<4> 139703936707624 99 48 139703936707643

noterminal.fth:14:<0>

trying to open /dev/ttyACM0 ... opened

-> forth:words
from forth:words
CDC\ TOOLS\ [THEN] [IF] [ELSE] .S NEED\ NOFORTH\ TO DOES> ['
IMMEDIATE ;CODE CODE REPEAT ELSE WHILE BEGIN THEN WORDLIST [CHAR]
\ ' EVALUATE VOCABULARY QUIT MS >NUMBER FIND .( ( CHAR PARSE
REFILL ACCEPT ." S" C" DEPTH ] [ RECURSE POSTPONE /STRING HEX
DECIMAL .R U.R . U. D. #> #S # SIGN HOLD <# TYPE SPACES SPACE
CR EMIT KEY? KEY CELLS CELL+ >BODY ALIGN COMPILE, C, , ALLOT
HERE ALIGNED ABORT CATCH THROW MOVE FILL */ */MOD MOD / /MOD
FM/MOD M* D+ DABS DNEGATE ABS NEGATE D2/ D2* - + 2/ 2* 1- 1+
RSHIFT LSHIFT INVERT XOR OR AND S>D 0<> 0= <> = 0< WITHIN U<
< U> > FALSE TRUE UMAX UMIN MAX MIN ROT PICK NIP TUCK SWAP
OVER DROP ?DUP DUP 2SWAP 2OVER 2DUP 2DROP 2R@ R@ 2R> 2>R R>
>R COUNT @ C@ +! ! C! SOURCE-ID >IN STATE BASE BL UM/MOD *
UM* I +LOOP LOOP UNLOOP LEAVE ?DO DO AGAIN UNTIL IF AHEAD ;
2LITERAL LITERAL VALUE VARIABLE CONSTANT :NONAME : CREATE EXIT EXECUTE

OK.0
@)

```

## noForth voor RP2350

We maken kleine stapjes omdat we het hele maakproces en de wijzigingen in de processor structuur moeten doorgronden. Jeroen is bezig met een zelf startend programmaatje die de LED op het Pico2 bordje moet laten knipperen. Een leeg programma laadt al succesvol, d.w.z. dat we een voor de RP2350 werkende UF2 kunnen genereren met *picobin blok!* Ook de flitser laadt en de LED flitst.

### Test programma:

```
\ test THUMB flasher for rp2350 BOOT

forget fllcd : fllcd ;
\ asm:
meta: decimal
: bit6 1 6 lshift ; : bit7 1 7 lshift ; : bit8 1 8 lshift ;
: bit9 1 9 lshift ; : bit25 1 25 lshift ;

hex      10000000 constant startFLASH
          20000000 constant startSRAM
          20078000 constant stacklim
          20080000 constant stacktop
          40020000 constant resets_base
          resets_base constant resets_reset
resets_base 3000 + constant resets_reset_clear
resets_base 8 + constant reset_done
          40028000 constant io_bank0_base
          40038000 constant padsbase
          40048000 constant xosc_base
          40048000 constant xosc_ctrl
          xosc_base 4 + constant xosc_status
          xosc_base C + constant xosc_startup
          40010000 constant clock_base
clock_base 48 + constant clk_peri_ctrl
          400E8000 constant rosc_base
          rosc_base 4 + constant rosc_freqa
          rosc_base 8 + constant rosc_freqb
          rosc_base 14 + constant rosc_div
          D0000000 constant sio_base
          E0000000 constant ppb_base \ ppb_base=M33 related registers
          ppb_base ED88 + constant cpacr \ checked

lbl[
  newlbl loop:          newlbl main:          newlbl gpioresetwait:
  newlbl xoscstable?:  newlbl wait:

\ thumb: fstblink [      \ for wabiForth
: fstblink [            \ for Forth-systems with the Dumb-assembler running
\ the minimal reset-vectors -----
  stacktop WORD,        \ initial stack pointer on vector 0x0
  startFLASH 8 + 1+ WORD, \ start address on vector 0x1 - Thumb bit set
\ catch any misaddressing with nop's -----
  nop, nop,
  b, main:
\ PICO.BIN for IMAGE DEF -----
  0 8B, 0 8B,           \ for alignment
  fffffded3 word,      \ magic word
  10210142 word,       \ family
  000001ff word,       \ << de 1 duidt op 1 PICO.BIN block
  00000000 word,       \ 0x0 = loop back to first block
```

```

ab123579 word,          \ magic word

main: \ -- gpio and pad reset: -----
    r0, resets_reset ldv32, \ ldr  r0, =RESETS_RESET
    r1, r0, 0  i5ldr,      \ ldr  r1, [r0]
    r2, bit6   ldv16,
    r1, r2,    bics,       \ bic  r1, r1, #(1<<6)=gpio
    r2, bit9   ldv16,
    r1, r2,    bics,       \ bic  r1, r1, #(1<<9)=pad-system
    r1, r0, 0  i5str,      \ str  r1, [r0]
gpioresetwait:
    r0, reset_done ldv32,  \ ldr  r0, =RESET_DONE
    r1, r0, 0  i5ldr,      \ ldr  r1, [r0]
    r2, bit6   ldv16,      \ 0x40=bit6
    r1, r2,    tst,        \ tst  r1, #(1<<6)
    beq, gpioresetwait:    \ beq  .GPIO_Reset_Wait
\ -- enable coprocessor P0 -----
    r0, cpacr  ldv32,      \ ldr  r0, =CPACR
    r1, r0, 0  i5ldr,      \ ldr  r1, [r0]
    r2, 3      ldv16,      \ jh: set bit [0:1] to give full access
    r1, r2,    orrs,       \ jh: iets anders gedaan als via: orr r1, r1, #(1<<0)
    r1, r0, 0  i5str,      \ str  r1, [r0]
    dsb,
    isb,
    \ isb - ditto
\ .GPIO_Config_Modify_CTRL: gpio25=0xCC
    r0, IO_BANK0_BASE_CC + ldv32, \ <<< exBUG: was C4!!!
    r1, 5      ldv16,      \ alt-func 5 = GPIO
    r1, r0, 0  i5str,      \ str  r1, [r4] - put modified value back
\ jh: pad isolation off <- Raspberry: do this after all stuff for gpio is set and done
\ gpio25=0x68
    r0, 40038068 ldv32,     \ gpio25 control address
    r1, r0, 0  i5ldr,      \ get value from gpio25
    r2, bit7   ldv16,
    r1, r2,    bics,       \ clear bit 7 -> enable output
    r2, bit6   ldv16,
    r1, r2,    orrs,       \ set bit 6 -> enable input
    r2, bit8   ldv16,
    r1, r2,    bics,       \ clear bit 8 -> switch isolation of
    r1, r0, 0  i5str,      \ store masked value at 0x40038068 to stop isolating pad_gpio
hex \ -- set GPIO25 as output -----
    r0, sio_base 30 + ldv32, \ 30=GPIO_OUT_SET
    r2, bit25   ldv32,      \ bit25 = gpio25
    r2, r0, 0  i5str,      \ do the actual switch of the value of GPIO25
\ -- than loop on/off  sio_base + 0x028 GPIO_OUT-XOR
loop:
    r0, sio_base 28 + ldv32, \ =D0000000 + 28 = GPIO_OUT_XOR
    r2, bit25   ldv32,      \ bit25 = gpio25
    r2, r0, 0  i5str,      \ do the actual switch of the value of GPIO25
decimal \ loop around 0.5s
    r1, 150000  ldv32,      \ zou ~0,3-0,7s bij een 3 cycle loop and 100-300 Khz
label0:
    r1, r1, 1  i3subs,
    bne,      label0:
    b,        loop:        \ loop back forever
] ;

```

## noForth vragenuurtje

Om de week op maandagavond om 20.00 uur is er een noForth vragenuurtje. De eerst volgende is op 15 juni, 29 juni, etc.

Daar kun je terecht met alle noForth vragen. Als ze erg technisch zijn, kun je ze vooraf insturen ([w.ouwerkerk@kader.hcc.nl](mailto:w.ouwerkerk@kader.hcc.nl)) of krijg je tijdens het volgende vragenuurtje antwoord.

### noForth vragenuurtje

20:00 - 21:00 (WET)

Vindt elke 14 dagen plaats

Koppeling naar vragenuurtje: [noForth vragenuurtje](#) | [Microsoft Teams](#) | [Meetup-Deelnemen](#)

## De noForth bibliotheek

Dit maal een stukje code die bruikbaar is bij de opdracht van deze maand. Deze code maakt dat je de tegel met een compacte bitmap kunt representeren, slechts 256 bits (32 bytes) zijn nodig.

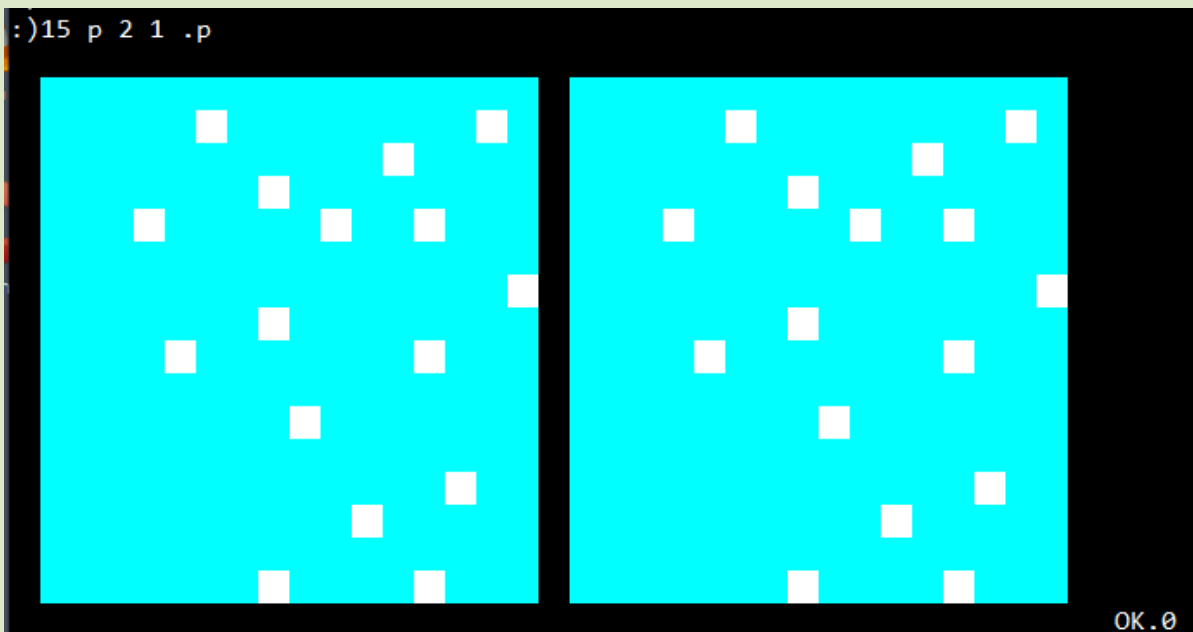
```
hex \ Bit array for compact noting of precence or on/off state
v: extra definitions
\ : LOC      ( bit-nr a -- bit byte-addr ) \ Bit location in word-addr
\   cell+ over 5 rshift + >r \ Convert to byte addresses
\   1F and bitmask r> ;    \ Convert low nibble to bit mask
code LOC      ( bit-nr a -- bit byte-addr ) \ Calc. bit location in word-addr
    680C3304 , 00AD0965 , 251F195B ,
    2501402C , 600D40A5 , CA10C804 , FFFF46A7 ,
end-code
                                \ Leave bit & word-adr
: BITARRAY
  create      ( bits "name"-- ) ( exec: -- a )
    20 /mod          \ Calculate length in cells & remainder
    swap 0<> - dup , \ Round & remember size
    cells allot ;   \ Reserve array

: *SET      ( nr a -- )      loc **bis ; \ Set bit nr in array a
: *CLR      ( nr a -- )      loc **bic ; \ Erase bit nr from array a
: GET*      ( nr a -- 0|msk ) loc bit** ; \ Bit nr set in array a?
: *ZERO     ( a -- )         @+ cells 0 fill ; \ Erase bit-map a
v: fresh
```

Een extra hulpmiddel uit de bibliotheek is COUNT\* die telt het aantal gezette bits in een bitmap. Ook handig bij de *tegel* opdracht.

```
v: extra definitions
\ Leave the number of bits set in bit array a
: COUNT* ( a -- +n ) \ Counted noted high bits
  0 over @ hx 20 * 0 ?do \ a +n=0
    over i swap get* \ a +n bit Bits present?
    if 1+ then \ a n+1 Add 1 when found
  loop nip ; \ +n
v: fresh
```

De *tegel* opdracht op het scherm met noForth t:



## HCC 50 jaar



Uw redacteur had tijdens het opstellen van ForthWords van april veel zin in een feestje maar was te vroeg.

In April **2027** is het 50 jaar geleden dat de HCC is opgericht.  
Hieronder een sfeer opname van de beginjaren van HCC.



Als er forth liefhebbers zijn die al langer lid zijn kunnen ze een artikel maken over de beginjaren van de Forth interessegroep. Mogelijk wordt dit op de HCC site worden gepubliceerd.

## Het Forth klasje

*/	123.45	.F	2DROP	procenten	D+	Celsius
D.	dubbels	DNEGATE	Fahrenheit	$\pi$	inhoud	12345.
	tweecellig	2>R	schaling	coördinaten	breuken	S>D
2DUP	vaste komma		vorm	eencellig	gemengd	

### Onderwerp tijdens deze bijeenkomst

Zie de 'Woorden raden' opdracht in het volgende artikel.

### Nog te behandelen onderwerpen:

- Getal uitvoer bouwen met <# # #S #>
- +LOOP voor string of COUNT
- CREATE DOES>
- Control flow stack
- OK lus
- KEY
- S"
- BEGIN WHILE WHILE REPEAT THEN
- COMPILE,

### Reeds behandelde onderwerpen. Zie de [tijdlijn](#) voor meer gegevens.

- \*/ gebruik voor schaling en vaste komma
- [Dubbele getallen](#) (Februari 2023)
- [CATCH](#) gebruiken (Augustus 2023)
- [WORDLIST](#) (VOCABULARY) (Oktober 2023)
- [Parsing \[IF\]](#) etc. (December 2023)
- [EVALUATE](#) (Februari 2024)
- [EXECUTE](#) (Februari 2024)
- [INVERT vs NOT](#) (April 2024)
- [MIN vs UMIN](#) (April 2024)
- [\[\] Arrays](#) (Juni 2024)
- Turnkey in (no)Forth (Augustus 2024)

Zijn er mogelijk nog andere onderwerpen naast die uit bovenstaande lijsten, vertel het Albert Nijhof!

Denk vast na over welk onderwerp tijdens de bijeenkomst van December wordt behandeld.

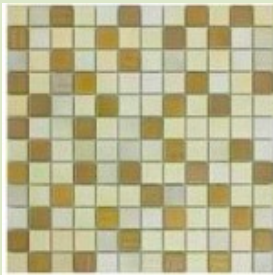
Vandaag behandelen we de  
programmeer opdracht: Mozaïektegels

Albert heeft onderstaande opdracht voor ons verzonnen om de kennis van noForth te vergroten.

Opgave voor juni 2026

## Mozaïektegels

Mozaïektegels zijn tegels die bestaan uit kleine tegeltjes, vierkante steentjes. Je vindt ze bijvoorbeeld op muren, in de badkamer of de keuken. Als ze vakkundig op de muur zijn aangebracht kun je de grenzen tussen de mozaïektegels niet meer zien.



Dit is één mozaïektegel.

Het is een prefab ding, de eindgebruiker hoeft al die steentjes dus niet afzonderlijk tegen de muur te plakken. Als de steentjes van kleur verschillen is hun plaatsing gewoonlijk random. Omdat mozaïektegels onderling ook verschillen is het totaaleffect van een muurvlak met mozaïektegels ook weer random. Dat vind ik nogal goedkoop en saai.

In mijn schuur staan 15 dozen met blauwe steentjes en 1 doos met witte. Daarmee ga ik mozaïektegels maken van 16 bij 16 steentjes met per mozaïektegel random verspreid 16 witte steentjes tegen een achtergrond van blauwe steentjes, met de volgende beperking:

- De witte steentjes mogen elkaar niet raken, ook niet aan hun hoekpunten.
- Als identieke mozaïektegels zowel horizontaal als verticaal tegen elkaar gelegd worden raken de witte steentjes elkaar nog steeds niet.

### Wat is hier nieuw aan?

Ik ga ervan uit dat we **identieke** mozaïektegels tegen een muur plakken. Daardoor ontstaan zich herhalende patronen, zoals vroeger bij behang. Dat is toch leuker dan die karakterloze random verdeling en hoogstwaarschijnlijk een gat in de markt!

## Opgave

1. Bedenk een programma `P ( nr -- )` dat patronen voor de witte steentjes ontwerpt. Bij verschillende `nr` horen verschillende patronen, een bepaald ontwerp krijgt dus een vast nummer. `nr` mag elke waarde hebben.
2. Bedenk een programma `.P ( hor ver -- )` dat een aantal mozaïektegels met het laatst ontworpen patroon naast en onder elkaar afbeeldt zodat je het effect op een groter vlak kunt bekijken



P.S.

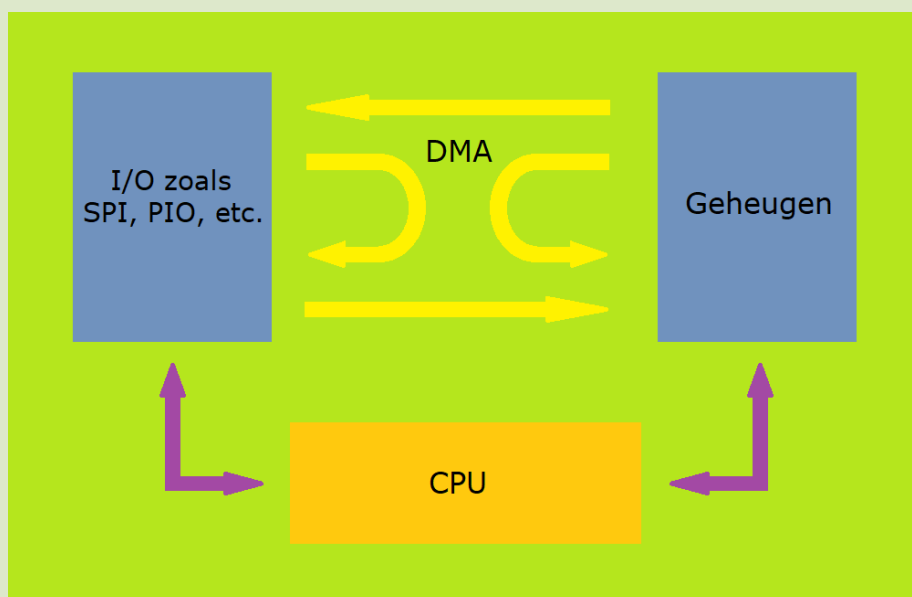
Nu Willem nog zover zien te krijgen dat hij een robot bouwt die mozaïektegels aan de hand van hun nummer in elkaar kan zetten

## PIO en DMA

Willem Ouwerkerk

De lezing behandelt stap voor stap het gebruik van DMA, dit is de structuur:

- Wat is PIO en wat is DMA
- DMA eigenaardigheden
- Het DMA CTRL register
- DMA van geheugen naar geheugen
- DMA van geheugen naar PWM zonder koppeling (chaining)
- De structuur van PIO opcode genaamd SET
- DMA van geheugen naar PIO met koppeling (chaining)
- Onafhankelijk DMA/PIO systeem met vier gekoppelde (chained) DMA's



## Object georiënteerd forth

Albert van der Horst

Forth was een van de eerste talen die objecten invoerde via CREATE DOES>. Het bindt een actie (methode) met data.

Bijvoorbeeld je kan een array definiëren die voor een index een waarde teruggeeft. Dat is beperkt tot slechts een actie. Maar als je het adres van het derde element nodig hebt om door te geven om te veranderen?

Ik keer CREATE DOES> als het ware binnenste buiten, zodat meer acties op dezelfde data structuur mogelijk zijn. Het definierende woord definieert nu een classe en een structuur tegelijkertijd. Een handjevol woorden vervangt >BODY STRUCT FIELD en wat dies meer zij.

Dit zijn class endclass M: M; .

## Volgende bijeenkomst zaterdag 8 augustus 2026

### Datums van de bijeenkomsten in 2026

In 2026 zijn de bijeenkomsten op de 2<sup>e</sup> zaterdag van de even maand.

- 8 augustus
- 10 oktober
- 12 december

### Datums van de bijeenkomsten in 2027

De volgende datums voor 2027 zijn vastgelegd.

- 13 februari
- 10 april
- 12 juni
- 14 augustus

Meld je aan voor de nieuwsbrief van hcc!forth

Abonneer je nu op de nieuwsbrief en blijf op de hoogte van onze activiteiten!

[Hier aanmelden](#)

## Maak jij al gebruik van de HCC!helpdesk en HCC!vraagbaak?

Voor al jouw vragen over Windows, Apple, Android, Linux en/of het gebruik van software kun je gebruik maken van de telefonische helpdesk.

Op werkdagen is deze bereikbaar onder nummer: 085-0441808

Maar je kunt ook altijd jouw vraag online aan ons stellen. Je ontvangt dan zo snel mogelijk een reactie per e-mail.

## Contact met HCC

[verenigingszaken@hcc.nl](mailto:verenigingszaken@hcc.nl) voor nagenoeg alle vragen van kader- en kernleden over de vereniging.

[ledenadministratie@hcc.nl](mailto:ledenadministratie@hcc.nl) voor alle vragen over individuele HCC-leden

[agenda@hcc.nl](mailto:agenda@hcc.nl) voor (spoed)vragen over de HCC-agenda

[pc-active@hcc.nl](mailto:pc-active@hcc.nl) voor vragen over de PC-Active

[hoofdbestuur@hcc.nl](mailto:hoofdbestuur@hcc.nl) voor vragen aan het Hoofdbestuur