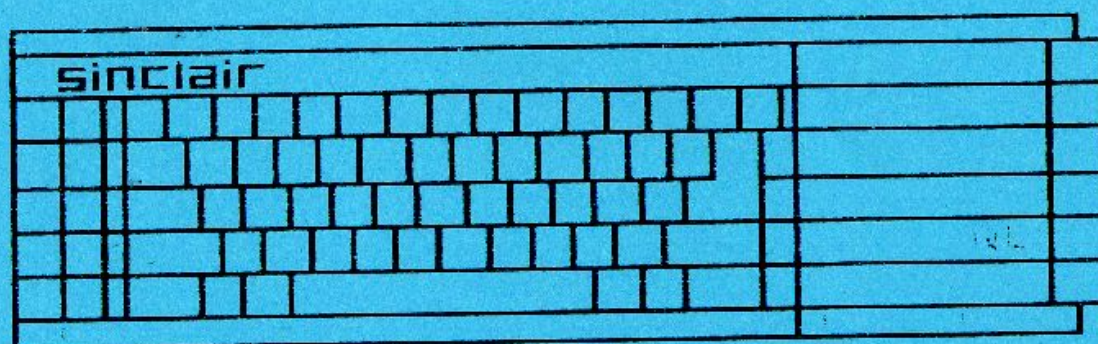


# QLONE

QL GROEP NOORD NEDERLAND



mei 1988  
Jaargang 3 nummer 5



Een uitgave van : QL GROEP NOORD-NEDERLAND

VOORZITTER : J. Doesburg 2e : B. Johnson  
Kalkwijk 7 Froukemaheerd 177  
9603 BB Hoogezand 9736 RR Groningen  
tel. 05980-27633 050-418699

SECRETARIS : R. Russchen  
Westermaad 55  
8431 TB Oosterwolde  
tel. 05160-5922

PENNINGMEESTER : B. Scheidema  
Rijksstraatweg 313  
9752 CE Haren  
tel. 050-345123

REDAKTIE QLONE : J. Doesburg - A. Stam - J. van Bruggen  
Bosweg 27  
9414 BD Hooghalen  
tel. 05939-501

---

BANKREKENING : 48.09.90.468 t.n.v. B. Scheidema Haren  
GIROREKENING : 3801095 t.n.v. B. Scheidema Haren

Contributie fl. 75,--. per jaar Abonnement QLONE fl. 20,--.  
Jeugdleden fl. 37,50. per jaar

---

De volgende clubavond is op 3 mei 1988 in het Denksport  
Centrum, Oliemuldersweg 43 te Groningen. Aanvang 19.30 uur.

---

Copy voor de 15e van de maand inleveren bij, of toezenden aan  
de redactie. Inleveren tijdens de clubavond kan ook. Listings  
tekeningen e.d. (brieven als Quill document) uitsluitend op een  
microdrive cartridge of 3 1/2 inch floppy disk.

---

Qlone wordt gemaakt op een QL met gebruik van Quill, Archive,  
SuperBASIC programma's en een QL printer.

---



## INHOUD

Van de redactie .....	3
XINPUT II .....	4
Vroom .....	7
Z88 .....	7
Waar haalt hij het vandaan? .....	8
QUILL header .....	9
Speuren in SuperBASIC .....	10
Sir Clive als chip-bakker .....	12
Tips .....	14
Algemene ledenvergadering .....	14
Oproep .....	15
Buitenlands nieuws .....	15

## Van de redactie

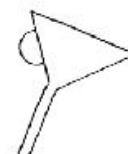
Toen ik mijn diskdrive aanschafte, dacht ik dat ik mijn programmatuur op 5 schijfjes (720K) wel zou kunnen houden. Nou tien dan misschien.



Dat pakte dus wel even anders uit! Op het ogenblik heb ik er zo'n 25 in gebruik en het dreigen er nog steeds meer te worden. Nog niet zo lang geleden kwam programmatuur maar mondjesmaat binnendruppelen en dan ook vaak nog van bedenkelijke kwaliteit. Tegenwoordig komt er zoveel goede programmatuur binnen dat een doorsnee QL'er het vrijwel niet meer bij kan houden. (zomaar een greep: de nieuwe EDITOR, verbeterde versie van Turbo, een database genaamd FLASHBACK enz).

Met andere woorden, als je dacht dat je op softwaregebied behoorlijk bij de tijd was, dan zal het softwareoverzicht dat de heren flop & micro nog voor de vakantie hopen uit te brengen je wel eens uit de droom helpen.

En wat ik met al die nieuwe programmatuur doe? Rekken en d'r bij blijven, zoals een bekend cabaretier placht te zeggen. En neem van mij aan dat dat niet meevalt!



# XINPUT (deel II)

Deze maand bekijken we het algoritme en de BASIC-versie van onze routine. Aan het eind van deze tekst staat het algoritme in mijn eigen pseudo-pseudo-code. De algemene structuur van het algoritme is als volgt:

```
Initialisatie
Inlezen toets
Default wissen ja/nee?
LUS
  Toetsafhankelijke instructies (delete, cursor verplaatsen etc.)
  Als de toets (een variant van) het ENTER commando was dan stoppen
  Nieuwe cursorpositie berekenen en inlezen toets
EINDE LUS
```

Tot het ENTER commando wordt gerekend alle toetsen met een code tussen 0 en 31 en tussen 192 en 254 (m.u.v. de cursor control commando's). De cursor control commando's zijn: cursor naar links/rechts en wissen aan de linker-/rechterkant.

Meer wilde ik niet zeggen over het algoritme, bestudeer het maar en bekijk voor de verduidelijking ook de BASIC-versie.

Albert Stam

## -----ALGORITHM-----

```
PROCEDURE:  XINPUT      syntax: XINPUT [#n,] var,default
                                var:= string variable
                                default:= string expression
```

```
.....
opm.: ==> betekent "spring naar"
```

```
      := betekent "is gedefinieerd als"
```

```
.....
c      : character-positie van de cursor in de input string
```

```
x,y    : screen-positie van de cursor
```

```
x0,y0: start screen-positie van de cursor
```

```
l      : lengte string
```

```
a$     : input string
```

```
line_len: maximaal aantal mogelijke characters op één regel
```

START

LABEL \*INIT\*

-c=1 l=LEN(a\$) (a\$ bevat nu de default waarde)

-welke window (of default)?

-waar staat de cursor nu? -> x,y gezet

-hoeveel characters kunnen er op één regel

(bij deze CSIZE)? -> line\_len

-x0=x y0=y

```

==> subroutine *A*
    := zet cursor op y0,x0 print a$
    := lees een toets uit -> key

-was key een printable character? (d.w.z. 31 < key < 192)
    ...ja -> wis a$

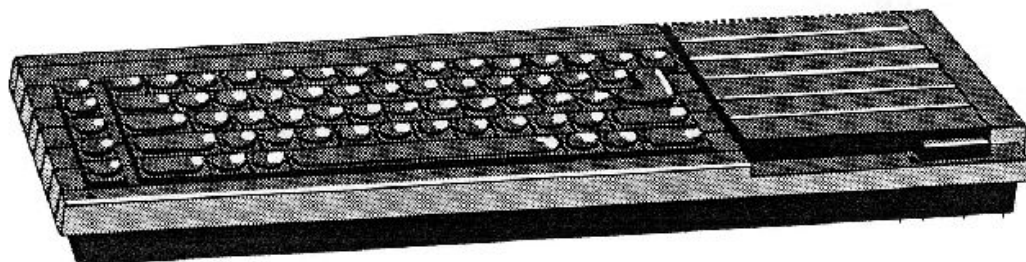
LABEL 1    *test key*
-cursor left: c=c-1 [ is c=0? ja: c=1 ]      ==>*UIT*
-cursor right: c=c+1 [ is c=1+2? ja: c=1+1]  ==>*UIT*
-CTRL cursor left: c=c-1 [is c=0? ja: c=1    ==>*UIT* ]
                                ==>subroutine *WIPE*
                                := wis character op positie c in a$
                                ==>*UIT*
-CTRL cursor right: c=c+1 ==>subroutine *WIPE* ==>*UIT*
-niet printable (en geen cursor control)?
    ja: key toevoegen en ==> *EXIT*
-(dus wel printable):
    key toevoegen op positie c
    1 met 1 verhogen      c met 1 verhogen

LABEL *UIT*
-bereken nieuwe cursor positie:
    y= INT(x0 + (c-1)/line_len )
    x= x0 + (c-1) - (y*line_len)
    y= y + y0
-wis a$ van het scherm af
==> subroutine *A*
==> LABEL 1 *test key*

LABEL *EXIT*
:= END

```

In het basicprogrammaatje op blz. 6 is dit algoritme verder uitgewerkt.



```

100 line_len=32:WINDOW line_len*6+5,205,0,0:BORDER 1,7
110 CLS:xinput 1,0,0,'default'
120 :
130 REMark EXTENDED INPUT (BASIC VERSION)
140 REMark by Albert Stam
150 :
160 DEFine PROCedure xinput(ch,y,x,default$)
170 time_out=10
180 AT#ch;y,x:a$=default$:c=1:l=LEN(a$):y0=y:x0=x
190 read_key
200 IF i>31 AND i<192:a$=''
210 :
220 REPEAT main_loop
230 REPEAT loop
240 IF i=192:c=c-1:IF c=0:c=1:EXIT loop:ELSE EXIT loop
250 IF i=200:c=c+1:IF c=1+2:c=1+1:EXIT loop:ELSE EXIT loop
260 IF i=194
270 c=c-1:IF c=0:c=1:EXIT loop
280 wipe:a$=t$:l=LEN(a$):EXIT loop
290 END IF
300 IF i=202:wipe:a$=t$:l=LEN(a$):EXIT loop
310 IF i<32 OR (i>192 AND i<>194 AND i<>200 AND
i<>202):a$=a$&i$:RETURN
320 t$=''
330 FOR n=0 TO LEN(a$)+1
340 IF n=c:t$=t$&i$
350 IF n<=LEN(a$) AND n>0:t$=t$&a$(n)
360 END FOR n
370 a$=t$:l=LEN(a$):c=c+1:EXIT loop
380 END REPEAT loop
390 CLS#ch
400 y=INT((c-1)/line_len)
410 x=x0+(c-1)-(y*line_len)
420 y=y+y0:read_key
430 END REPEAT main_loop
440 END DEFine
450 :
460 DEFine PROCedure read_key
470 AT#ch;y0,x0:PRINT#ch;a$;:OVER#ch;-1
480 REPEAT get_key
490 AT#ch;y,x:INK#ch;7:PRINT#ch;'_';:INK#ch;4
500 i$=INKEY$(#ch,time_out):IF i$<>'':EXIT get_key
510 END REPEAT get_key
520 i=CODE(i$):OVER#ch,0
530 END DEFine
540 :
550 DEFine PROCedure wipe
560 t$=''
570 FOR n=1 TO LEN(a$):IF n=c:NEXT n:EXIT n:ELSE t$=t$&a$(n)
580 END DEFine

```

VROOM, VROOM, VROOM...

Wat zou dit nu kunnen zijn ? Een stofzuigerspel ? Nee, natuurlijk een racespel. Dit spel is geschreven door Pyramide. Je moet op elk circuit op zijn minst 10 auto's achter je laten, en dat is best moeilijk. Ik heb het spel nu een tijdje maar verder dan de G.P. van Amerika, de derde ben ik nog niet gekomen.

Je gaat van start trekt op tot de maximum snelheid van 360 Km per uur, twee vervelende details de banden draaien niet realistisch en je hebt een digitale kilometer teller. Je moet flink remmen, tenminste ik (ik stuif er met 360 in) in de bochten om te voorkomen dat je eruit vliegt. Dat vertraagt niet alleen je kan ook zo tegen een bord op knallen.

Een fantastisch spel, realistische geluiden alleen je kan niet afsnijden. Bv. in een haarspeldbocht kan je niet recht door de bocht uit rijden en zo een stuk afsnijden. Maar ja, je kan niet alles hebben en is dat dan wel helemaal eerlijk...

Peter Venema

PS> Als iemand een bepaald onderwerp over Basic uitgediept wil hebben, schrijf dan even !!  
Als je belangstelling hebt voor een srt. QL CONNEXIONS in QLONE schrijf dan even !!  
Duidelijk formuleren wat de bedoeling is !!

Alleen schrijven naar:

J.P. Venema  
Westereems 1  
9642 KP Veendam

---

## Z88

Op de paasinstuif van de SGGE was hij al te bewonderen, de nieuwe machine van Sir Clive. Op zaterdagmiddag 14 mei zal er in de school de Wendakker een demonstratie van deze machine worden gegeven door iemand van Anglo-Dutch Data, de importeur van de Z88 en QLGNN leden zijn van harte welkom deze demonstratie bij te wonen.

# WAAR HAALT HIJ HET VANDAAN?

Wat komt er uit de volgende som:

$$(\text{SQRT}(10) * \text{SQRT}(10)) - 10$$

Natuurlijk is het antwoord nul, maar de meeste computers, calculators en andere rekenwonders zullen fout antwoorden (de QL inclusief). De reden is dat een wortel, een sinus en alle andere bijzondere functies door een digitale getallenkraker niet berekend worden maar benaderd worden. Als de BASIC functie of de QDOS trap voor zo'n bijzondere wiskundige functie benaderd moet worden gebruikt de computer daar een ingebouwd programma voor. De benadering van de wortel uit het getal 10 gaat als volgt:

Begin met een ruwe benadering; neem  $\text{SQRT}(10) = 3$ .  
 Neem als volgende benadering  $((10/3) + 3)/2 = 3.1667$   
 Dit is een betere benadering want  $3*3=9$  en  $3.1667*3.1667=10.03$   
 Doe hetzelfde nog een keer maar neem nu 3.1667 i.p.v. 3  
 uitkomst 3.1623; dit is nog nauwkeuriger want  
 $3.1623*3.1623=10.0001$   
 Enzovoort.

Op deze manier kan de computer een lange rij benaderingen maken waarbij de volgende benadering steeds nauwkeuriger is dan de vorige. Dit gebeurt met de volgende formule:

$$\text{nieuwe benadering} = (\text{oude benadering} + a/\text{oude benadering})/2$$

Voor de computer maakt een paar berekeningen meer of minder niet veel uit. Daarom wordt als eerste benadering bij worteltrekken meestal hetzelfde getal genomen, namelijk 1.

Het onderstaande programma berekent de wortel met de bovenstaande methode. Met benaderen wordt doorgegaan totdat de benadering nauwkeurig genoeg is. Er wordt gekeken of de breuk  $a/(b*b)$  dicht genoeg bij 1 ligt (a is het getal waaruit de wortel genomen moet worden en b is de berekende wortel). Bijvoorbeeld als die breuk niet meer dan  $1E-08$  van 1 verschilt. Probeer het programma maar eens uit en bekijk hoeveel benaderingen de computer nodig heeft om tot een bevredigend antwoord te komen.

```
100 INPUT "WELKE WORTEL? ";a
110 b=1
120 REPEAT loop
130   b=(b+a/b)/2:PRINT b
140   verschil=1-(a/(b*b))
150   IF ABS(verschil)<1E-08:EXIT loop
160 END REPEAT loop
170 PRINT:PRINT SQRT(a):PRINT
```



# De QUILL header

Zoals bekend verondersteld mag worden kan het programma Qspell alleen doc\_files van Quill inlezen. Als je nu een andere file (bijvoorbeeld van de Editor) toch wilt controleren op spelling zou je deze dus eerst in quill moeten importeren en vervolgens als doc wegsaven. Aangezien Quill niet bepaald uitblinkt door zijn importeertijd, zocht ik naar een andere oplossing. De oplossing die ik nu toepas is heel simpel en gaat veel sneller dan de toestand via Quill.

Met onderstaande regels maak je een file aan die een doc\_file (gedeeltelijk) nabootst.

```
100 OPEN_NEW #3,flp1_quill_simulator
110 a$=CHR$(0)&CHR$(20)&'vrmlqdf0'
120 a$=a$&CHR$(0)&CHR$(0)&CHR$(13)&CHR$(231)
130 a$=a$&CHR$(1)&CHR$(214)&CHR$(0)&CHR$(38)
140 a$=a$&CHR$(0)&CHR$(88)&CHR$(0)&'page nn'&CHR$(0)
150 PRINT #3,a$
160 CLOSE #3
```

Als ik nu een editor\_file via Qspell op spelling wil controleren, merge ik via het commando "af.flp1\_quill\_simulator" de quill\_naboots\_tekst boven de aanwezige tekst. Vervolgens schrijf je dit op de normale manier via de Editor weg, met dien verstande dat je je aan de quill\_naamgeving moet houden, dus maximaal acht karakters en het achtervoegsel \_doc.

Als je nu vanuit Qspell deze file inleest zie je dat deze inderdaad als Quill\_file herkend wordt.

Er is een klein nadeel (ik vond het de moeite niet waard om er een oplossing voor te zoeken) en dat is dat de schermuitvoer niet helemaal klopt. Editor en Quill gebruiken namelijk niet dezelfde methode om het eind van een regel aan te geven. Je ziet nu daarom de woorden aan elkaar verschijnen. Het programma herkent ze echter wel als afzonderlijke woorden, dus er is niets aan de hand.

Als Qspell de hele tekst heeft gehad, geeft hij de foutmelding "NOT VALID QUILL FILE". Aangezien de tekst nu helemaal gecontroleerd is, zal ons dat een vreugde zijn en drukken we op een toets om verder te gaan en kunnen we het woordenboek wegschrijven met de nieuw toegevoegde woorden.

Rein Dijk

## Speuren in SuperBASIC

In verband met de komende cursus SuperBASIC een programma met uitleg, en wel eentje van de master himself. Ik heb deze genomen omdat er allerlei technieken in zitten verwerkt die eenvoudig zijn toe te passen in eigen programma's. Het heet fback en is de snelste manier om een copie te maken van wat dan ook. Het is echter geen kraakprogramma, het copiëert alleen bliksemsnel. Benodigdheden : Toolkit 1 of toolkit 2 (van Tony Tebby)



```
100 REMark fast non_selective backup      @ 1984 Tony Tebby  QJUMP
110 WTV 4:CLS:CLEAR
    WTV 4  Toolkit commando die de tv screen genereert de 4 houd in
    dat er naar mode 4 geschakeld wordt (alleen WTV is ook afdoende)
    Een 8 maakt ook de default tv screen maar dan mode 8.

120 PRINT "Backup"!!FREE_MEM:"free memory"
    FREE_MEM Functie die aangeeft hoeveel vrije bytes er in je QL
    aanwezig zijn. Hier wordt voor de veiligheid 512 bytes
    afgetrokken (ruimte om een file te closen)

130 INPUT 'Source drive > ';source$
    INPUT spreekt voor zich.

140 IF source$ INSTR '12345678': source$='flp'&source$&'_'
    Als het ingetikte woord voorkomt in de string "12345678" dan is
    er alleen een nummer ingetikt. Er wordt in dit geval 'flp' voor
    "geplakt" zodat we als er een "1" was getikt we dan "flp1" krijgen.
    Is de "IF" voorwaarde niet, waar dan gebeurt er niets.
    Een kleine bug hier is als er iemand "123" of "456" intypt maar
    dit is iets wat nooit of zelden zou gebeuren.
    Uiteraard zou de "flp" door "mdv" of "ram" enz kunnen worden
    vervangen. Merk op dat in een één regelig IF-END IF statement
    END IF kan vervallen. Nota Bene als men hier een END IF zou
    plaatsen op regel b.v. 141 dan stop het programma (BUG) waarbij
    eigenlijk de foutmelding END IF without IF zou moeten verschijnen

150 IF source$(LEN(source$))<>'_': source$=source$&'_'
    Als de laatste karakter van het ingetikte woord iets anders is
    dan een "_" (=underline), LEN(string) geeft de lengte van een
    string B.V :
```

10 PRINT LEN("123") geeft 3. Door de slice die word uitgevoerd met source\$, in ons geval :

```
10 PRINT "123"(LEN("123"))
```

zou een "3" opleveren. Was er een "1" in regel 130 ingetikt dan zorgde regel 140 voor "flp" ervoor en deze regel, 150 dus, voor een "\_" zodat we dan "flp1\_" zouden hebben.

Merk ik nog even op dat als er in regel 130 de volle device-naam b.v. "mdv1\_" gegeven was, was er aan de hele string source\$ niets veranderd, omdat dat geen van de voorwaarden waar was



160 STAT source\$

STAT is een soort "DIR" maar dan geeft deze alleen naam en sector specificaties b.v:  
10 STAT "mdv1\_"

170 INPUT 'Destination drive > ';dest\$

Weer een input met dit verschil dat de eerste ervoor was om aan te geven waar de files weg moesten komen (source op z'n Engels) en deze is om aan te geven waar het naar toe moet (destination).

180 IF dest\$ INSTR '12345678': dest\$='flp'&dest\$&'\_'

Net als in regel 140 kijken of er één cijfer is ingetikt.  
zoja "plak" er "flp" voor

190 IF dest\$(LEN(dest\$))<>'\_': dest\$=dest\$&'\_'

Zie regel 150 maar nu met dest\$

200 IF NOT FOP\_DIR(#3,dest\$): CLOSE #3

FOP\_DIR is een heel handig commando. Iedereen weet dat de magnetische opslag media voor de QL (behalve cassettes) in sectoren zijn opgedeeld (met FORMAT "drivenummer\_naam"). Welnu de Directory (van DIR mdv1\_, DIR flp1\_ enz) staat op de nulde sectoren. Dit commando OP\_DIR "opent" die sectoren, en verbindt deze aan ; raampje / hok / channel / "#" nummer in dit geval drie. de drive wordt aangegeven met de tweede parameter, en wel één die ik eerder heb aangegeven n.l. dest\$. Het leuke is dat er ook gecontroleerd wordt of er wel een floppy of een cartridge in de drive aanwezig is b.v.  
10 PRINT FOP\_DIR(#3,'mdv1\_')  
geeft als er geen cartridge is -7 (NOT FOUND)  
als er een niet leesbare cartridge in zit -16 (BAD OR CH MEDIUM)  
als er een goeie in zit een 0.  
Merk ook op dat het kanaal direct weer wordt gesloten.  
Op deze plaats in het programma is de bedoeling van FOP\_DIR dan ook niet om de directory te lezen maar om ruimte te maken in het geheugen en wel de COMMON HEAP. Waarom en wat dat is later in de listing. Als laatste nog, als er een floppy of cartridge wordt gevonden (IF NOT FOP\_DIR(#3,dest\$)) moet #3 weer worden gesloten bij een foutmelding hoeft dit niet het wordt dan door QDOS gelijk weer gesloten, wel wordt dan de nodige geheugen ruimte vrijgemaakt, vandaar dat het programma hier niet wordt onderbroken voor een foutmelding.

René (KOMT volgende maand) Russchen

In de volgende QLONE ontrafelt René weer een stuk van dit programma.



### Sir Clive als chip-bakker

Chip-fabricage lijkt op het drukken van een kleurenfoto in een krant of tijdschrift. Een pagina wordt opgebouwd uit vier losse films voor de kleuren en het zwart; net zo bestaat het ontwerp van een geïntegreerde schakeling uit afzonderlijke maskers die over elkaar worden belicht. De "afdruk" verschijnt op silicium, in de vorm van een ronde wafel met een doorsnee van meer dan 10 centimeter, waarop tientallen of zelfs honderden chips tegelijk ontstaan. Na controle wordt de wafel in stukken gesneden en krijgen de chips elk hun eigen huis van keramiek of kunststof. Soms is één chip een compleet elektronisch apparaat. Maar in de meeste gevallen ontmoeten vele chips elkaar op de print, een patroon van koperbanen op een plak isolatiemateriaal.

In principe is het mogelijk om heel wat stappen in de fabricage over te slaan door de chips direct op de wafel elektrisch te verbinden. Het geheel past dan in één behuizing, wat gunstig is voor de afmetingen en de prijs. Vooral in computers zou gebruik van de hele wafel interessant zijn, omdat ze zoveel chips bevatten en weinig niet integreerbare onderdelen (grote condensators, spoelen en dergelijke).

Helaas steekt er een spaak in het wiel. Wafels zijn nooit volmaakt; een aantal chips doen het niet, hoe zorgvuldig de fabrikant ook te werk gaat. Eind jaren zeventig zochten verschillende fabrikanten de oplossing in wafelsystemen waarop alle functies in drievoud waren uitgevoerd. Onderlinge controle moest een goede werking verzekeren. Het bleek een aanpak die alleen maar faillissementen opleverde.

De industrie likte haar financiële wonden en ging op de oude voet door met losse chips, tot in 1984 Sir Clive Sinclair zich ermee bemoeide. Hij was op het hoogtepunt van zijn succes met de kleine homecomputer en had geld genoeg voor de oprichting van Metalab, een "denktank" waaruit nieuwe wereldschokkende projecten moesten rollen.

Een van de eerste mensen die hij in dienst nam was Ivor Catt. Hoewel buiten de vakkringen minder beroemd, is Catt net zo'n kleurrijke Engelsman als Sir Clive zelf. In 1953 won hij (17 jaar oud) een staatsbeurs voor wiskunde. Daarna werkte hij voor Ferranti in Manchester, toen een van de belangrijkste centra van computerontwikkeling. In de jaren zestig vertrok hij naar de VS, waar hij voor Motorola supersnelle verbindingen tussen logische schakelingen onderzocht. Na zijn terugkeer gaf hij les aan Watford College in Engeland. Tijdens zijn lange carrière roeide Catt meer dan eens tegen de hoofdstroom in. Hij is het bijvoorbeeld absoluut niet eens met de gangbare verklaring van elektromagnetische verschijnselen. Ook heeft hij forse kritiek op Einsteins relativiteitstheorie. De laatste jaren schrijft hij vrij regelmatig in Wireless World, waar zijn artikelen goed zijn voor boeiende veldslagen in de correspondentiekolommen.



Wat computers betreft geeft Catt de voorkeur aan seriële boven parallelle communicatie. Als tussen chips op een print informatie parallel heen en weer gaat - bijvoorbeeld in "woorden" van 16 bits tegelijk, dus over 16 lijnen - kost dat extra printoppervlak, de bedrading wordt ingewikkelder en de chips hebben meer aansluitpinnen nodig.

Het doel van dit alles is extra snelheid, maar die wordt lang niet altijd bereikt. Een ander Catt-standpunt is dat computers zichzelf moeten genezen als een chip het begeeft, door functies automatisch opnieuw te verdelen over de resterende hardware. De twee gedachten gaan samen in Catts methode voor gebruik van de hele wafel. Uitgangspunt is een wafel vol identieke chips, bijvoorbeeld geheugens. Hiertussen ligt een netwerk van seriële verbindingen, met per geheugen één knooppunt. Als de wafel voor het eerst wordt ingeschakeld, draait een programma dat knooppunt nummer 1 opdracht geeft contact te zoeken met zijn vier naaste burens, die op hun goede werking worden getest. Het programma noteert de resultaten en gaat verder volgens een systeem dat het vinden van alle goede knooppunten garandeert. Dan maakt het programma een kaart met daarop een lange, slingerende route die de werkende knooppunten met elkaar verbindt, en controleert het alle geheugenschakelingen die via de route toegankelijk zijn. De complete kaart wordt vastgelegd in een EPROM. Elke volgende keer dat de wafel wordt ingeschakeld dient de kaart voor opbouw van gevonden route - binnen een fractie van een seconde - waarbij knooppunten opdracht krijgen defecte delen uit te sluiten. De route is daarna het seriële pad waarlangs informatie de wafel binnenkomt en verlaat. Mocht later nog een geheugen of een deel van de verbinding stuk gaan, dan is opnieuw draaien van het testprogramma voldoende om de kaart aan te passen. De wafel verliest in zo'n geval een klein deel van zijn capaciteit, maar hoeft niet meteen in de vuilnisbak te verdwijnen.

De ontwikkeling van Catts principe verliep zonder ernstige problemen. In 1985 kondigde Sinclair een "solid state Winchester drive" voor zijn QL-computer aan. Helaas verscheen het apparaat niet op de markt. Vrouwe Fortuna liet Sir Clive in de steek en hij moest zijn homecomputer-imperium aan concurrent Amstrad verkopen. Ook voor Metalab viel het doek, maar uit de as verrees Anamartic Ltd.

Het prototype van een wafel-geheugen is de Wisper-1; de capaciteit is 500 kilobytes. Het eerste commerciële produkt van Anamartic wordt de Wisper-2, die 7 megabytes kan onthouden en eind 1987 op de markt moet komen. In vergelijking met de klassieke Winchester hard disk is de Wisper een ware revolutie. Om te beginnen zijn de afmetingen veel kleiner, net als het stroomverbruik. Verder is de toegangstijd 1000 maal korter (minder dan 50 microseconden in plaats van 30 à 90 miliseconden) en verloopt het transport van data tien maal zo snel (2.5 megabytes per seconde tegen 150 kilobytes voor een Winchester).

Wispers kunnen heel goed dienen als buffer tussen het snelle werkgeheugen van een computer en de nogal trage optische schijf. Ook toepassing in draagbare computers is ongetwijfeld mogelijk, hoewel de wafel stroom moet verbruiken om informatie vast te houden.

Catts methode is niet beperkt tot geheugens. Even gemakkelijk kunnen andere chips op een wafel met elkaar worden verbonden, zoals microprocessors die gezamenlijk aan hetzelfde probleem werken, als aan een lopende band. Dat opent de weg naar snelle en goedkope verwerking van radar- en videosignalen. Anamartic wil zich hiermee bezighouden zodra de Wisper-2 een commercieel succes is.

---

### Tip(s)

Als je in Quill met zéér grote documenten wilt werken verdient het aanbeveling om twee cartridges (schijven) te gebruiken. Eén voor de \_doc waarmee je wilt werken en één voor de def\_tmp. De cartridge (schijf) voor de def\_tmp doe je in de drive waarop jouw Quill zijn \_doc verwacht. De cartridge voor de \_doc doe je dan in de drive waarvan Quill wordt geladen.

Om ASCII te importeren in Quill, hoeft zo'n file geen achtervoegsel te hebben in de vorm van \_exp! Elk willekeurig drieletterig achtervoegsel is voldoende.

Heb je wel eens last van crashes in je QL? Treden ze ze gerekend vanaf het moment van opstarten van je QL steeds op hetzelfde moment op? Dan heb je te maken met warmte(afvoer) problemen. probeer eens:

- een 2 ampère spanningsregelaar met koelpasta
- druk de IC's eens aan
- vervang het (waardeloze) voetje van de processor

---

### Algemene ledenvergadering

Op dinsdag 7 juni a.s. wordt voorafgaand aan de gebruikelijke activiteiten een algemene ledenvergadering gehouden.

Alvast even in de agenda noteren!



**OPROEP:**

Gezien de lijst software moeten de leden elk vele spelen/utilities hebben. Nou zijn de handleidingen lang niet allemaal even duidelijk (Ik snap geen barst van Archive, TABS in QUILL is een raadsel om over multi-line assembling met Assembler Workbench maar te zwijgen. En ik best wel zeer goed overweg met de QL. Schrijf zelf kleine eenvoudige machinecode programmaatjes, heb o.a. uit wanhoop een nieuwe database geschreven etc.), dus als je een programma fantastisch vindt of nieuwe mogelijkheden weet : schrijven.

En zou iemand zich kunnen opwerpen als vraagbaak voor leden. Bv. over machinetaal/Basic (ik zelf, alleen schrijven) Pascal. Ook een cursus machinetaal (zou ik wel willen doen, maar mijn kennis hierover is te beperkt). Pascal zou ook niet te versmaden zijn. En net zo als in QL WORLD een DIY TOOLKIT met op verzoek handige basic uitbreidingen.

J. P. Venema  
Westereems 1  
9642 KP Veendam



---

**BASIC Toolbox for the Sinclair QL**

Mega Toolbox for the QL contains 170 new commands and is compatible with QL BASIC compilers. In addition to improved command-line editing and job-control/memory-allocation commands, the Toolbox features tune-playing jobs; alarm clocks; copying, mirroring, zooming, saving and restoring of windows; multitasking print commands for creating rolling demos and slide shows; three-dimensional text printing; screen-swapping; handling of pies; keyboard control; memory copying; and fill.

Price £29.95

Contact: Compware, 75 Repton Dr.,  
Haslington, Crewe CW1 1SA, U.K.  
0270 - 5822301

bron: BYTE 2-1988

Wie kent dit pakket? Reacties graag !!!



DRUKWERK

Afz. Q.L.G.N.M.  
Redaktieadres  
Bosweg 27  
9414 BD HOOCHALEN

