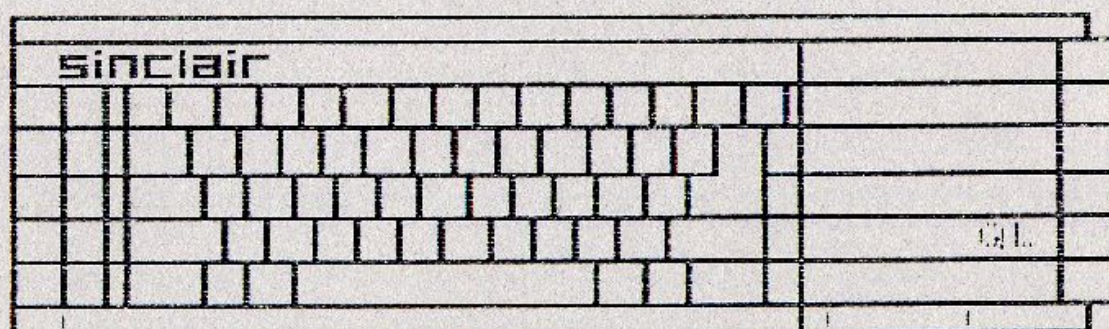


QLONE

QL GROEP NOORD NEDERLAND



oktober 1988
Jaargang 3 nummer 8

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 : vacature

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 4 oktober 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

Infodag SGGE	4
When	5
XINPUT	6
ANAMARTIC	8
Tip(s)	9
Infodag Hasselt	9
Speuren in SuperBASIC.....	10



vergeten ze het belangrijkste.... de QL!!!

***** INFODAG ** INFODAG *****

Op zaterdag 8 oktober aanstaande organiseert de Sinclair Gebruikers Groep Eemsmond van 10.00 tot 18.00 uur voor de vijfde keer haar jaarlijkse Noordelijke Computer Informatie Dag in de grote zaal van de Molenberg te Delfzijl.

Aanwezig zullen zijn:

Sinclair Gebruikers Groep Eemsmond
QL Groep Noord Nederland
Atari Gebruikers Groep Groningen
MSX Computer CLUB Eemsmond
Commodore Groep Appingedam
Amiga Groep Groningen
Algemene Personal Computer CLUB

Tevens zullen de volgende bedrijven uit de computer- en aanverwante branches zich in de Molenberg presenteren.

Dataskip, Gouda	Sinclair produkten
Byte, Groningen	Diverse computers
Therminal Software Production, Baarle Nassau	Sinclair Gids
Engelsman Kant. autom., Delfzijl	Kantoormachines
Venema & Veendorp, Delfzijl	PC's, MSX
Marketing Etiketting, Lelystad	Computer behoeftigheden

Zoals je ziet is ook de QLGN in Delfzijl van de partij.
Voor het bemannen van onze hoek in de Molenberg zoeken wij nog enkele QL enthousiastelingen. Ik heb gehoord dat onze penningmeester ons tussen de middag een warme hap aanbiedt. Maar dat moet je maar niet verder vertellen, want anders hebben we natuurlijk zo de hele club d'r zitten!

WHEN ... END WHEN

Deze keywords dienen voor de controle van variabelen.
De syntax is als volgt:

```
WHEN <voorwaarde>
  <statement>
END WHEN
```

<voorwaarde> is een expressie. De te controleren variabele moet links van het "=" teken staan.

<statement> kunnen één of meer SuperBASIC-regels zijn.

Een WHEN-constructie wordt geopend met WHEN en afgesloten met END WHEN.

In tegenstelling tot SuperBASIC procedures en functies, die direct aanroepbaar zijn, wordt een WHEN-constructie pas uitgevoerd als aan de voorwaarde wordt voldaan.

In een programma mag een onbegrensd aantal WHEN-constructies staan. Er mogen voor elke te controleren variabele meerdere WHEN-constructies worden toegepast.

Alle variabelen, die worden gecontroleerd, zijn in een tabel opgenomen. Zodra een variabele een nieuwe waarde toegewezen krijgt test het systeem de WHEN-voorwaarde op "waar" of "niet waar".

Een INPUT geldt niet als toewijzing van een waarde. Dit is met de volgende truc te omzeilen: INPUT a: a=a. Alleen globale variabelen zijn te controleren.

Na afhandeling springt het programma weer terug naar de plaats waar aan de voorwaarde van de WHEN-constructie werd voldaan en voert vervolgens het volgende statement uit.

Onderstaande listing is een eenvoudig voorbeeld voor de toepassing van WHEN.

100 WHEN nieuw=rechts	280 mark bereik
110 nieuw=links+1	290 AT 10,oud: PRINT "*"
120 bereik=bereik+1	300 :
130 mark bereik	310 REPEAT main
140 END WHEN	320 nieuw=oud+RND(-1 TO 1)
150 :	330 AT 10,oud : PRINT " "
160 WHEN nieuw=links	340 AT 10,nieuw: PRINT "*"
170 nieuw=rechts-1	350 oud=nieuw
180 bereik=bereik-1	360 END REPEAT main
190 mark bereik	370 :
200 END WHEN	380 DEFINE PROCEDURE mark(bereik)
210 :	390 AT 11,links : PRINT bereik,
220 links=10	400 AT 11,rechts: PRINT bereik+1,
230 rechts=30	410 END DEFINE
240 bereik=0	420 :
250 oud=20: CLS	430 DEFINE PROCEDURE update
260 AT 10,links : PRINT "I"	440 SAVE_O when_demo
270 AT 10,rechts: PRINT "I"	450 END DEFINE

XINPUT 3

Deze maand eindelijk het werkelijke assembly programma. Dat wil zeggen het eerste deel, want het zal in drie delen geplaatst worden. Het eerste deel bevat de variabelen, de installatie en het inlezen van de parameters.

Allereerst de variabelen, er zijn zes voor de gebruiker interessante variabelen: TIME_OUT, CURSOR, SPACE, FLASH, DEF_CHAN, MAXL. Ze staan op volgens mij de handigste waarden. (TIME_OUT bepaalt de knippersnelheid van de cursor, de rest wijst zichzelf).

De initialisatie is een standaardprocedure in assembly dus daar heb ik niets aan toe te voegen.

Het inlezen van de parameters is min of meer ook een standaard procedure. Dit programma is nog niet compleet maar het kan al wel zelfstandig werken. Dan moeten alleen nog twee labels gedefinieerd worden: INPUT en CHANID. Ze worden in het laatste deel als volgt gedefinieerd:

CHANID	DS.L	1	
INPUT	DS.W	1	length of input\$
	DS.B	MAXL	space for input\$
	DS.W	1	space for entercommand%

Met deze toevoeging werkt het eerste deel al zelfstandig. Volgende maand kan het tweede deel gewoon na dit deel gezet worden beginnend bij het label BEGIN.

Verder nog iets over de werking van de machinetaal versie van XINPUT omdat het een tikje anders werkt dan de SUPERBASIC versie van de vorige keer. Er is een extra parameter toegevoegd: entercommand%, in deze integer variabele wordt de ASCII code van de toets die gebruikt werd als entercommando gezet. Verder kunnen alleen stringvariabelen als parameters gebruikt worden en zijn die variabelen zowel als de in te voeren waarde aan een maximum gebonden. Deze beperkingen hebben programmeer-technisch gezien natuurlijk een reden maar degene die zich interesseert voor de opzet van het programma kan dat zelf wel ontdekken.

QL 68000 Assembly Language

1988 Jul 06 20:28:52

```

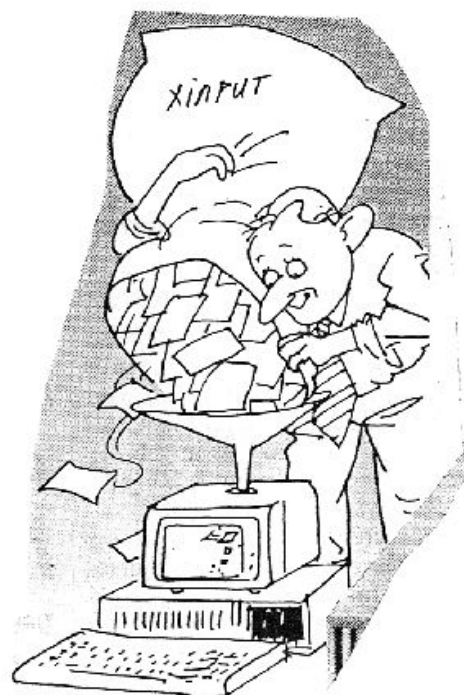
1 * EXTENDED INPUT      a SuperBasic extension by Albert A. Stam
2 *                      code: EXT0 1988
3 *
4 *                      extended input
5 *                      Format: XINPUT [#n],input$,entercommand%,default$
6 *
7 BV_CHBAS EQU          $30          base of channel name list
8 BV_CHF EQU            $34          end of channel name list
9 BV_RIP EQU            $58          runningpointer to arithmetic stack
10 TIME_OUT EQU          10          time_out for IO.FBYTE
11 CURSOR EQU            " "          cursor character
12 SPACE EQU             " "          space character
13 FLASH EQU            -1          -1: flash cursor; 0: flash off

```

14	DEF_CHAN	EQU	1	default channel
15	MAXL	EQU	64	max len default\$ and input\$
16	BP.INIT	EQU	\$110	install procedure
17	BP.LET	EQU	\$120	return a parameter value to BASIC
18	BV.CHRIX	EQU	\$11A	allocate space on arithmetic stack
19	CA.GTINT	EQU	\$112	get integer para's from nametable
20	CA.GTSTR	EQU	\$116	get string para's from nametable
21	SD.CHENQ	EQU	\$0B	get window size&curs.pos.(char.)
22	SD.SETMD	EQU	\$2C	set OVER mode
23	SD.POS	EQU	\$10	position cursor (char.coord.)
24	IO.SSTRG	EQU	\$07	send a string of bytes
25	IO.SBYTE	EQU	\$05	send a byte
26	IO.FBYTE	EQU	\$01	fetch a byte
27				
28	* first create new keyword			
29		LEA	PROC(PC),A1	def.-list
30		MOVEA.W	BP.INIT,A2	
31		JSR	(A2)	
32		RTS		
33	PROC	DC.W	1	no. of procedures
34		DC.W	BEGIN-*	start of procedure
35		DC.B	6,'XINPUT',0	procedure name
36		DC.W	0,0,0	end of list
37				
38	ERR_BP	MOVEQ	#-15,D0	error BAD PARAMETER
39		RTS		
40	ERR_NO	MOVEQ	#-6,D0	error CHANNEL NOT OPEN
41		RTS		
42	ERR_OV	MOVEQ	#-18,D0	error OVERFLOW
43		RTS		
44				
45	READ_PARA	MOVEQ	#DEF_CHAN,D1	default channel
46		CMPA.L	A3,A5	are there parameters?
47		BEQ.S	ERR_BP	...no
48		BTST	#7,1(A6,A3.L)	has 1st para a #?
49		BEQ.S	GET_CHAN	...no: get channelID of default
50	*			...yes: get 1st para
51		MOVEA.L	A5,-(A7)	save top of parameters and
52		MOVEA.L	A3,A5	set new top of parameters to
53		ADDQ.L	#8,A5	top of first parameter
54		MOVEA.L	A5,-(A7)	(it's the new base)
55		MOVEA.W	CA.GTINT,A2	get an integer
56		JSR	(A2)	
57		MOVEA.L	(A7)+,A3	restore nametable pointers
58		MOVEA.L	(A7)+,A5	(doesn't influence SR)
59		BNE.S	ERR_BP	something went wrong (BP or XP)
60		MOVE.W	0(A6,A1.L),D1	this is the channelnumber
61	GET_CHAN	MULU	#\$28,D1	find channeltable of
62		ADD.L	BV_CHBAS(A6),D1	this channelnumber
63		CMP.L	BV_CHP(A6),D1	does it exist?
64		BHI.S	ERR_NO	...no
65		MOVEA.L	0(A6,D1.L),A0	get channel-ID
66		LEA	CHANID,A2	save it
67		MOVEA.L	A0,(A2)+	and get VAR-address

68				
69	CMPL	A3,A5	are there still para's?	
70	BEQ.S	ERR_BP	...no	
71	CLR.L	D2		
72	MOVE.B	1(A6,A3.L),D2	fetch nametable entry code byte	
73	AND.B	##00001110,D2	is next para a string? (bit 0 set)	
74	BNE.S	ERR_BP	...no	
75	MOVE.B	9(A6,A3.L),D2	code byte next para	
76	AND.B	##00000011,D2	is next para a integer?	
77	CMPL.B	#3,D2	bit 0 & 1 set?	
78	BNE.S	ERR_BP	...no	
79	MOVE.L	A3,(A2)	save tablepointer-to-inputs	
80 *			for later use with BP.LET	
81	ADDA.L	#16,A3	base of last para	
82				
83 * read default\$				
84	CMPL	A3,A5	are there still para's?	
85	BEQ.S	ERR_BP	...no	
86	MOVEA.W	CA.GTSTR,A2	get string	
87	JSR	(A2)		
88	TST.L	D0		
89	BNE	ERR_BP		
90	MOVE.W	#MAXL,D2	max length	
91	LEA	INPUT(PC),A2		
92 READ_STR	MOVE.W	0(A6,A1.L),D3	length	
93	MOVE.W	D3,(A2)+	store it	
94	SUB.W	D3,D2	too long?	
95	BMI	ERR_BP	...yes	
96	SUBQ.W	#1,D3	counter	
97	BMI.S	DONE	len=0, nothing to store	
98 NEXT	MOVE.B	2(A6,A1.L),(A2)+	get character, and store it	
99	ADDQ.L	#1,A1		
100	DBF	D3,NEXT		
101 DONE	CLR.L	D0		
102	RTS		done	
103				
104 BEGIN				

volgende maand deel 2 van dit
super-luxe zelfbouw commando.



Internationale QL Gebruikersdag

De Sinclair QL Gebruikersgroep Limburg in België organiseert op 23 oktober in Hotel De Borggraaf te Hasselt een internationale QL-dag. Het is de bedoeling om fans van de QL uit heel Europa bij elkaar te brengen. Deze QL-dag moet de eerste zijn van een reeks van QL-dagen in de komende jaren in andere Europese landen. Een andere doelstelling is het opzetten van een internationale QL gebruikers database en uiteindelijk de opzet van een internationale QL/Thor gebruikersgroep.

De Gebruikersgroep Limburg is druk doende steun te verzamelen van software- en hardware leveranciers en vraagt ook de QL gebruikersgroepen om op alle mogelijke manieren deze internationale QL-dag onder de aandacht van het publiek te brengen.

En dus.....

Is de QLGNN van plan om samen met de SGGE af te reizen naar Hasselt. Als je belangstelling hebt om mee te gaan, neem dan even contact op met René Russchen.

tel. 05160 - 5922

TIP(s)

Hallo, ben jij misschien nog zo'n ploeteraar die nog niet met TOOLKIT II werkt? Dan wordt het toch wel de hoogste tijd. Vraag maar eens een clubgenoot die het ding wel heeft. Gegarandeerd dat hij compleet lyrisch wordt en je ook wel kan vertellen hoe ook jij de gelukkige bezitter van dit stukje programmeursgeluk kunt worden.

Last van storingen in je QL bij het in- en uitschakelen van elektrische apparaten? Probeer eens een ontstøringscondensator (b.v. 0,1 uF 250V~) over de primaire kant van je voedingstrafo.

* Onderstaande functie geeft -1 als v negatief is;

1 als v positief is en 0 als v nul is

```
100 DEFine FuNction SGN(v)
```

```
110   RETurn v/( ABS(v) + NOT(v) )
```

```
120 END DEFine
```

* 100 FOR n=1 TO 5,1 TO 5

```
↑↓
```

```
100 FOR m=1,2
```

```
110   FOR n=1 TO 5
```

```
120 END FOR m
```

Speuren in SuperBasic IV

Dit is dan al weer het vierde en laatste deel van onze speurtocht door SuperBasic.



```

540 :
550 REMark reserve space
560 :
570 base=ALCHP(tbytes)
580  IF base=0: PRINT 'Out of memory': STOP
    ALCHP is te vergelijken met RESPR met het verschil:
    met CLCHP heb je je geheugen terug
    ALCHP reserveert geheugen aan de andere kant van de memory
    map.
    RESPR = Residente procedure space
    ALCHP = Allocate memory comon heap
    CLCHP = Clear comon heap (geef geheugen terug)
    Omdat er ALCHP gebruikt wordt is de schijnbaar nodeloze
    FOP_DIR op regel 200
    Wat is het geval. De directory blocks staan ook in de
    comon-heap. Dus als men eerst all files ging laden hier,
    en daarna een directory ging opvragen van een nog niet
    gespecificeerde "flpl_" of "mdvl_" dan ging het dus behoorlijk
    mis. Ik hoop dat men dit kan volgen. Eenvoudig gezegd: je kan
    maar één ding op één plaats tegelijk in het geheugen houden.
    ALCHP(aantal bytes) geeft start adres van het gereserveerde
    geheugen dit word gegeven aan de variable base.
590 :
600 REMark read all files
610 :
620  next_base=base
    volgende base is hier dus de eerste

630  FOR file=1 TO files
    files was het aantal files

640  PRINT 'Loading ';source$&name$(file);'...'
    Boodschap wat ben ik aan het doen.

650  LBYTES source$&name$(file),next_base
    Laad de file vanaf next_base

660  next_base=next_base+wbytes(file)
    Volgende next_base word aangegeven met memory pointer

670  END FOR file
    Doe dit tot alle gelezen files zijn geladen.

680 :
```



```

690 IF group
700 IF source$=dest$: PRINT 'Replace backup in'!dest$!'...!'
710 ELSE
720 PRINT 'Format'!dest$!'...!'
730 END IF
    group zie regel 250 is niet 0 als er al iets is gecopieerd
    anders moet er nog geformatteerd worden.
    Als de destination dezelfde is als de source
    moet hier de copy weer worden geïnstalleerd.

740 IF NOT group OR source$=dest$
750 PRINT '!space to continue'
760 REPEAT wait: IF INKEY$(-1)=" " THEN EXIT wait
770 END IF
780 IF NOT group: FORMAT dest$&medium$
790 next_base=base
    volgende base

800 FOR file=1 TO files
810 PRINT 'Copying ';name$(file);'...'
820 IF dbytes(file)
830 SEEXECdest$&name$(file),next_base,fbytes(file),dbytes(file)
840 ELSE
850 SBYTES dest$&name$(file),next_base,fbytes(file)
860 END IF
    Hier hetzelfde maar dan nu wegschrijven.
870 next_base=next_base+wbytes(file)
880 END FOR file
890 PRINT 'Verifying...'
900 RECHP base
910 REPEAT wait: IF NOT PEEK(163840+238): EXIT wait: REMark SV_MDRUN
920 IF eof_dir: EXIT group
930 IF source$=dest$
940 PRINT 'Replace source cartridge in'!source$!'and press enter'
950 REPEAT wait: IF (CODE(INKEY$(-1)))=10: EXIT wait
960 END IF
970 END FOR group
980 STOP

```

Ik hoop dat er nu een hele hoop duidelijk is geworden. Ik moet echter wel bekennen dat ik het ook niet zo kan programmeren. Ja nu ik het een keer heb gezien. Maar bedenk maar eens wat nieuws en maak er dan zo'n programma van. dat valt niet mee dus. Verder wegens tijdgebrek aan het eind van de listing wat minder commentaar, ik denk toch dat de meeste mensen nu bepaalde dingen van de QL beter snappen. Mochten er nog vragen zijn dan hoor ik het wel.

Rene Russchen
(de man die sneller programmeert dan zij schaduw)

DRUK WIEKK

Afz. Q.L.G.-M.M.
Redaktieadres
Postweg 27
9414 BD HOOCHALEN

