

# BULLETIN

SINCLAIR  
GEBRUIKERSGROEP  
GRONINGEN/ASSEN



met dank  
aan  
Johan  
Koelman  
en  
Ronald  
Raaijen.

15e jaargang nr **6 + 7**  
juni - juli 1988



## IMPRESSUM



VOORZITTER/  
PENNINGMEESTER/  
VERHUUR:  
Jan Arends  
Heiligelaan 66  
9636 CP Zuidbroek  
tel: 0598 - 452247

Giro 5965342 t.n.v.  
rekening SGG,  
ZUIDBROEK.

SECRETARIS:  
Martin den Hollander  
Numero Dertien 8  
9644 TV Veendam  
tel: 0597 - 645474

VICE VOORZITTER/  
VICE SECRETARIS:  
Roelof Koning  
Selwerderstraat 26  
9717 GK Groningen  
tel: 050 - 3124298

VICE PENNINGMEESTER/  
MATERIAAL COMMISSARIS:  
Coen Ballintijn  
B. Boermalaan 7  
9765 AP Paterswolde  
tel: 050 - 3091482

BESTUURSLEDEN:  
mevr. F. Elstrodt  
Rudy Biesma  
Tonny Stap

Het lidmaatschap van onze gebruikersgroep bedraagt per kalenderjaar (jan-dec):  
f 20,00 voor personen tot en met 17 jaar  
en f 30,00 voor personen van 18 jaar en ouder.  
De data van de SGG-bijeenkomsten (10 per jaar), staan in elk SGG-BULLETIN op de  
'2 na laatste' pagina vermeld.

De SGG-club-diskettes zijn in de volgende formaten verkrijgbaar:  
3.5 inch 40 tracks SS (OPUS) en 3.5 inch 80 tracks DS (DISCIPLE).  
Ze kosten: f 5,00, uitgezonderd "TW3 DTP", die is f 10,00.  
en "RWMS-DOS", die is f 7,50.  
Verzendkosten f 2,50 per stuk, f 3,00 bij meer.

*U kunt tijd worden van de SGG  
DOOR U OP TE GEVEN BIJ DE PENNINGMEESTER.*

Het SGG-BULLETIN is een uitgave van de Sinclair gebruikersgroep Groningen/Assen.  
Het geeft naast verenigingsnieuws, veel informatie over en voor de SINCLAIR en  
SAM computers (Hardware, software, programma's, aanpassingen, uitbreidingen enz.)  
Het SGG-Bulletin verschijnt 10 keer per jaar. (elke maand, met in juni en sept  
gecombineerde juni/juli en aug/sept nummers).

De redactie is er trots op dat het blad nog steeds bijna volledig op de Spectrum  
gemaakt wordt! Dit met behulp van Tasword III en een eigen DTP-programma.  
Artikelen, listings, illustraties en andere inzendingen zijn voor de  
verantwoordelijkheid van de inzender. De 'Type in' programma's zijn soms getest  
maar ook dan niet gegarandeerd zonder fouten.

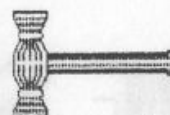
De gepubliceerde programma's worden meestal ook op clubdiskettes uitgebracht.  
Overname van artikelen, illustraties en andere publicaties zijn uitsluitend  
toegestaan met toestemming van de redactie en/of auteur.  
De leden krijgen het SGG-BULLETIN gratis. Nabestelde oudere nummers tegen  
kostprijs (copy+porto).

Niet-leden: Losse nummers f 4,00 . (nabestellen van oudere nummers á f 4,00).  
Advertentiekosten (Sgg-tjes) voor niet-leden f 5,00.

*N.B. Als u een diskette/tape met kopij/software instuurt, vermeldt hierop dan  
a.u.b. duidelijk uw eigen adres, zodat bij beschadiging van de enveloppe, de  
diskette/tape weer bij de afzender terugkomt.*



## VOORZITTER / INHOUD



Hallo allemaal,

In tegenstelling met de hierbovenstaande kop vindt u hier niet het 'praatje' van de voorzitter. Wegens ziekte is Jan Arends daar niet aan toe gekomen.

Vanaf deze plaats willen we hem dan ook namens ons allemaal van harte beterschap wensen.

Om dat Jan naast voorzitter ook penningmeester is zult u op bv. het financiële overzicht en de ledenlijst nog even moeten wachten.

Gelukkig zijn buiten het bestuur en de redactie nog enkele leden die regelmatig (en soms zelfs veelvuldig) kopij inleveren. Vandaar dat we die nu eens (op de cover) wat hulde willen brengen.

Met vr. gr.

Johan.

In dit nummer:	auteur	blz.
- Coverscreen: Hulde voor de Schrijvers --	Impressum :	2
- SAM: VM, virtualer Z80 processor?	: SPC: Stefan Haller	4
- Fou(t/d)meldingen op de PC	: Johan Koelman	8
- Machinecode Onderwijzer (1)	: JwK	9
- Hidden Extras: More in Z80 then known	: Your Sinclair	13
- Oproep: Hulp bij Hardware projekten	: Johan Koelman	15
- Sintech Produkt News, oa: de Kompakt !!	: Sintech	16
- Programmeer wedstrijd	: Johan Koelman	20
- Kleine Machinecode routines Part 1	: JwK.	21
- In Memoriam: A.J. Schoneveld	:	23
- Zomaar even een artikeltje	: Stefan Postema	24
- Multiveel Multiface Pokes part 3	: Ronald Raaijen	25
- Alch News: a small lesson in Time	: Justin Thorpe	27
- Oproep voor de 6e Hol. Sincl.en Sam dag:	HCC-sgg + SGG-G/A	29
- Bijeenkomsten: SGG / HCC / Rijswijk	:	30
- Reparatie adres	: Austermuhle / Rennefeld	30
- Esgeeggetjes , Handel, Z80-emulator adressen :		31

Sluitingsdatum copy:

Sept/Aug nummer: 15 Augustus

Oktober nummer: 12 September

REDACTIE ADRES: Johan Koning, Mieden 6, 9866 TM LUTJEGAST. tel: 0594 - 612807  
email: JwKoning@fremail.nl

## DIE SEITEN FÜR DEN SAM!!

### VM - ein virtueller Z80 Prozessor?

Hallo SAM-Freunde!

Da es in letzter Zeit ziemlich ruhig um den SAM im Info geworden ist, und wir auf die 100. Ausgabe zugehen (ich hoffe, die 101. tut's auch. Wo), dachte ich mir, daß es mal wieder Zeit ist, etwas für den SAM zu schreiben.

Also habe ich mich hingesetzt und mir ein Thema ausgedacht. Das ist gar nicht so leicht, zumal mein SAM leider kaputt ist (Soundchip und eine Tastaturreihe) und der Emulator auf meinen Mac (zu alt!) nicht läuft. Daher ist alles was ich hier sage bzw. behaupte rein theoretisch. Wenn etwas nicht 100% stimmt oder ich Syntaxfehler mache, bitte entschuldigt.

Als der noch SAM noch lief und ich in Assembler programmierte, war es grausam sich um das Speichermanagement zu kümmern. Es waren immer nur 64Kb aufgrund des internen Aufbaus des Z80 des Speicher zugreifbar. Das Paging half zwar, nur mußte man beim Programmieren immer aufpassen, daß man die richtige Speicherbank einschaltet, und sein Programm nicht zu lang wurde, das es in den verrückbaren Speicherbereich hineinfiel. Kurzum, wurde das Programm länger oder der Speicherzugriff komplizierter, mußten richtig komplizierte Techniken entwickelt werden.

Eine Idee wurde geboren: Der Z80VM. Das soll ein neuer virtueller Z80-Prozessor werden. VM steht für "Virtual Machine" und bedeutet übersetzt nichts anderes als "Virtuelle Maschine". Das momentan berühmteste Beispiel für VM ist Java. Der Z80VM sieht auf den ersten Blick aus wie der handelsübliche Z80. Es ist nur ein erweiterter Z80, der 3 neue 21bit-Register und ca. 100 neue Befehle hat, die softwaremäßig emuliert werden. Daher ist dieser Z80 nur eine VM. Nun wie soll dieser Z80VM meine Probleme lösen? Ganz einfach. Der Z80VM besitzt 3 neue Register: VX, VY und VO ("V" steht für virtuell), und sind 21bit-Register. Die neuen Befehle bewirken den Zugriff und Verwendung der neuen Register. Die Tabelle am Ende verdeutlicht die neuen Befehle mit Bedeutung und Parameter.

Warum sind die neuen Register 21bit? Verwirrend, nicht wahr? Nun, logisch mußten es nun 24bit sein, aber diese Register sollen mir das Speichermanagement erleichtern. Nun wie? Um 21bit im Speicher darzustellen sind 3 Bytes notwendig. Das erste Byte ist die Bank, und die

letzten beiden Bytes zeigen auf die Adresse in der Bank. Eine Bank ist nur 16Kb groß, also hat einen

Adressbereich von 0 bis 16383. Und die Adresse 16383 benötigt nur 13 Bits. Das macht zusammen 21 Bit. Es stehen damit die vollen 4 Mb (mit 4x1Mb Speichererweiterung) in 255 Banken à 16Kb zur Verfügung.

Außerdem bin ich fasziniert von der DLL-Technik (Windows) bzw. der SHARED LIBRARY-Technik (MacOS). Das brachte mich auf die Idee, daß es doch eigentlich egal sein sollte an welche Speicheradresse ich mein Programm lade. Es sollte immer laufen, ob ich es nach 32768 lade oder 237453. Dieses Problem will ich mit Hilfe des VO-Registers lösen. VO gibt sozusagen den Nullpunkt im Speicher an. Beispiel? Okay.

Nehmen wir an VO hat den Wert 36000. Nun sagen wir dem Z80VM: Springe relativ nach Adresse 49152 (JR 49152). Der Z80VM errechnet nun die absolute Speicheradresse (49152 + VO), die lautet 85152, und springt dorthin. Hätte VO nun den Wert 30000, würde der Z80VM zur absoluten Speicheradresse 79152 springen. Damit wäre unser Programm IMMER frei beweglich, sprich REALLOKIERBAR. Natürlich kann man auch weiterhin mit JP <pnn> absolut springen, in dem Falle direkt nach Speicheradresse 49152.

Ich weiß, daß das alles hier ziemlich kompliziert ist, aber ich denke mir doch, daß mich die Programmierer unter uns doch ein klein wenig verstanden haben. Wenn nicht dann emailt mir einfach. Darüber würde ich mich sehr freuen, auch wenn es nicht um den Z80VM geht. Genuß geredet, jetzt lasse ich Beispiele für mich sprechen.

Beispiel: Eine Routine zum Addieren von 10000 und 20000, und dann zu der Adresse springen, die das Ergebnis anzeigt, also 30000.

```
LD  VX,10000 ; VX bekommt den Wert 10000
LD  VY,20000 ; VY bekommt den Wert 20000
ADD VX,VY    ; VX und VY addieren, das
              ; Ergebnis in VX speichern
JP  [VX]     ; Zur Adresse springen, das VX
              ; anzeigt -> 30000
```

Ich hoffe, daß ich bald den Sourcecode für die VM fertig habe, so daß, falls sich jemand dafür interessiert, diesen bei mir anfordern kann.



# SGG BULLETIN

So nun die Tabelle mit allen neuen Opcodes, die über RST 8 eingeleitet werden. Beispiel:

```
LD VX, 32767 -> RST 8 ; Aktiviere VM
DB 13          ; Opcode für LD VX, <pnn>
DB 1           ; Erste Byte ist Page <p>
               ((=32767/16384)-1)
DW 10000       ; danach Adresse <nn>
               (=32767-P*16384)
```

Opcode	Mnemonic	Bedeutung
0	RET	Ersatz für RET, damit VM richtig zurückspringen kann. Normaler RET kann nur innerhalb 0 und 65535 zurückspringen. Der RET von der VM kann zwischen 0 und 4194304.
1	JP <pnn>	Absoluter Sprung nach Page <p>, Adresse <nn>
2	JR <pnn>	Relativer Sprung von Offset VO um Page <p>, Adresse <nn>
3	JP [VX]	Absoluter Sprung zur Adresse, die VX beinhaltet
4	JR [VX]	Relativer Sprung von Offset VO um Wert, den VX beinhaltet
5	JP [VY]	Absoluter Sprung zur Adresse, die VY beinhaltet
6	JR [VY]	Relativer Sprung von Offset VO um Wert, den VY beinhaltet
7	CALL <pnn>	Absoluter Sprung nach Page <p>, Adresse <nn> mit Rücksprung durch RET (Opcode 0)
8	CALL [VX]	Wie JP (VX) nur mit Rücksprung RET-Möglichkeit
9	CALL [VY]	Wie JP (VY) nur mit Rücksprung RET-Möglichkeit
10	CALR <pnn>	Relativer Sprung von Offset VO um Page <p>, Adresse <nn> mit Rücksprung durch RET (Opcode 0)
11	CALR [VX]	Wie JR (VX) nur mit Rücksprung RET-Möglichkeit
12	CALR [VY]	Wie JR (VY) nur mit Rücksprung RET-Möglichkeit
13	LD VX, <pnn>	Lade Register VX mit 21bit Wert <p>=Page, <nn>=Adr.
14	LD VY, <pnn>	Lade Register VY mit 21bit Wert <p>=Page, <nn>=Adr.
15	LD VO, <pnn>	Lade Offset-Register VO mit 21bit Wert
16	LD VXA, <p>	Lade Pagebyte des VX-Registers mit Page <p>
17	LD VYA, <p>	Lade Pagebyte des VY-Registers mit Page <p>
18	LD VOA, <p>	Lade Pagebyte des VO-Offset-Registers mit Page <p>
19	LD VXHL, <nn>	Lade Adressbytes des VX-Registers mit Adresse <nn>
20	LD VYHL, <nn>	Lade Adressbytes des VY-Registers mit Adresse <nn>
21	LD VOHL, <nn>	Lade Adressbytes des VO-Offset-Registers mit <nn>
22	LD VXH, <n>	Lade Highbyte der Adressbytes von VX mit <n>
23	LD VYH, <n>	Lade Highbyte der Adressbytes von VY mit <n>
24	LD VOH, <n>	Lade Highbyte der Adressbytes von VO mit <n>
25	LD VXL, <n>	Lade Lowbyte der Adressbytes von VX mit <n>
26	LD VYL, <n>	Lade Lowbyte der Adressbytes von VY mit <n>
27	LD VOL, <n>	Lade Lowbyte der Adressbytes von VO mit <n>
28	LD VX, [<pnn>]	Lade VX mit Wert von absoluter Speicheradr. <pnn>
29	LD VY, [<pnn>]	Lade VY mit Wert von absoluter Speicheradr. <pnn>
30	LD VO, [<pnn>]	Lade VO mit Wert von absoluter Speicheradr. <pnn>
31	LR VX, [<pnn>]	Lade VX mit Wert von relativer Speicheradr. <pnn>
32	LR VY, [<pnn>]	Lade VY mit Wert von relativer Speicheradr. <pnn>
33	NOP	- Reserved -
34	LD VXA, [<pnn>]	Lade Pagebyte von VX mit Wert von absoluter Speicheradr. <pnn>
35	LD VYA, [<pnn>]	Lade Pagebyte von VY mit Wert von absoluter Speicheradr. <pnn>
36	LD VOA, [<pnn>]	Lade Pagebyte von VO mit Wert von absoluter Speicheradr. <pnn>
37	LR VXA, [<pnn>]	Lade Pagebyte von VX mit Wert von relativer Speicheradr. <pnn>
38	LR VYA, [<pnn>]	Lade Pagebyte von VY mit Wert von relativer Speicheradr. <pnn>
39	NOP	- Reserved -
40	LD VXHL, [<pnn>]	Lade Adressbytes von VX mit Wert von absoluter Speicheradr. <pnn>
41	LD VYHL, [<pnn>]	Lade Adressbytes von VY mit Wert von absoluter absoluter Speicheradr. <pnn>

Opcode	Mnemonic	Bedeutung
42	LD VOHL,(<pnn>)	Lade Adressbytes von VO mit Wert von absoluter Speicheradr. <pnn>
43	LR VXHL,(<pnn>)	Lade Adressbytes von VX mit Wert von relativer Speicheradr. <pnn>
44	LR VYHL,(<pnn>)	Lade Adressbytes von VY mit Wert von relativer Speicheradr. <pnn>
45	NOP	- Reserved -
46	LD VXH,(<pnn>)	Lade Highbyte der Adressbytes von VX mit Wert von absoluter Speicheradr. <pnn>
47	LD VYH,(<pnn>)	Lade Highbyte der Adressbytes von VY mit Wert von absoluter Speicheradr. <pnn>
48	LD VOH,(<pnn>)	Lade Highbyte der Adressbytes von VO mit Wert von absoluter Speicheradr. <pnn>
49	LR VXH,(<pnn>)	Lade Highbyte der Adressbytes von VX mit Wert von relativer Speicheradr. <pnn>
50	LR VYH,(<pnn>)	Lade Highbyte der Adressbytes von VY mit Wert von relativer Speicheradr. <pnn>
51	NOP	- Reserved -
52	LD VXL,(<pnn>)	Lade Lowbyte der Adressbytes von VX mit Wert von absoluter Speicheradr. <pnn>
53	LD VYL,(<pnn>)	Lade Lowbyte der Adressbytes von VY mit Wert von absoluter Speicheradr. <pnn>
54	LD VOL,(<pnn>)	Lade Lowbyte der Adressbytes von VO mit Wert von absoluter Speicheradr. <pnn>
55	LR VXL,(<pnn>)	Lade Lowbyte der Adressbytes von VX mit Wert von relativer Speicheradr. <pnn>
56	LR VYL,(<pnn>)	Lade Lowbyte der Adressbytes von VY mit Wert von relativer Speicheradr. <pnn>
57	NOP	- Reserved -
58	LD A,VXA	Lade Register A mit Pagebyte von VX
59	LD A,VYA	Lade Register A mit Pagebyte von VY
60	LD A,VOA	Lade Register A mit Pagebyte von VO
61	LD A,VXH	Lade Register A mit Highbyte der Adressbytes von VX
62	LD A,VYH	Lade Register A mit Highbyte der Adressbytes von VY
63	LD A,VOH	Lade Register A mit Highbyte der Adressbytes von VO
64	LD A,VXL	Lade Register A mit Lowbyte der Adressbytes von VX
65	LD A,VYL	Lade Register A mit Lowbyte der Adressbytes von VY
66	LD A,VOL	Lade Register A mit Lowbyte der Adressbytes von VO
67	LD VXA,A	Lade Pagebyte von VX mit Register A
68	LD VYA,A	Lade Pagebyte von VY mit Register A
69	LD VOA,A	Lade Pagebyte von VO mit Register A
70	LD VXH,A	Lade Highbyte der Adressbytes von VX mit Register A
71	LD VYH,A	Lade Highbyte der Adressbytes von VY mit Register A
72	LD VOH,A	Lade Highbyte der Adressbytes von VO mit Register A
73	LD VXL,A	Lade Lowbyte der Adressbytes von VX mit Register A
74	LD VYL,A	Lade Lowbyte der Adressbytes von VY mit Register A
75	LD VOL,A	Lade Lowbyte der Adressbytes von VO mit Register A
76	LD HL,VXHL	Lade Register HL mit Adressbyte von VX
77	LD HL,VYHL	Lade Register HL mit Adressbyte von VY
78	LD HL,VOHL	Lade Register HL mit Adressbyte von VO
79	LD VXHL,HL	Lade Adressbyte von VX mit Register HL
80	LD VYHL,HL	Lade Adressbyte von VY mit Register HL
81	LD VOHL,HL	Lade Adressbyte von VO mit Register HL
82	LD A,(<pnn>)	Lade Register A mit Wert in absoluter Speicheradresse <pnn>
83	LD A,(VX)	Lade Register A mit Wert in absoluter Speicheradresse, die Register-VX anzeigt
84	LD A,(VY)	Lade Register A mit Wert in absoluter Speicheradresse, die Register-VY anzeigt



# SGG BULLETIN

Opcode	Mnemonic	Bedeutung
85	LD A, (VO)	Lade Register A mit Wert in absoluter Speicheradresse, die Register VO anzeigt
86	LR A, (<pnn>)	Lade Register A mit Wert in relativer Speicheradresse <pnn>
87	LR A, (VX)	Lade Register A mit Wert in relativer Speicheradresse, die Register VX anzeigt
88	LR A, (VY)	Lade Register A mit Wert in relativer Speicheradresse, die Register VY anzeigt
89	NOP	- Reserved -
90	LD HL, (<pnn>)	Lade Register HL mit Wert in absoluter Speicheradresse <pnn>
91	LD HL, (VX)	Lade Register HL mit Wert in absoluter Speicheradresse, die Register VX anzeigt
92	LD HL, (VY)	Lade Register HL mit Wert in absoluter Speicheradresse, die Register VY anzeigt
93	LD HL, (VO)	Lade Register HL mit Wert in absoluter Speicheradresse, die Register VO anzeigt
94	LR HL, (<pnn>)	Lade Register HL mit Wert in relativer Speicheradresse <pnn>
95	LR HL, (VX)	Lade Reg. HL mit Wert in relativer Speicheradresse, die Reg. VX anzeigt
96	LR HL, (VY)	Lade Reg. HL mit Wert in relativer Speicheradresse, die Reg. VY anzeigt
97	NOP	- Reserved -
98	ADD A, VXA	Addiere Register A mit Pagebyte von VX
99	ADD A, VYA	Addiere Register A mit Pagebyte von VY
100	ADD A, VOA	Addiere Register A mit Pagebyte von VO
101	ADD A, VXH	Addiere Reg. A mit Highbyte der Adressbytes von VX
102	ADD A, VYH	Addiere Reg. A mit Highbyte der Adressbytes von VY
103	ADD A, VOH	Addiere Reg. A mit Highbyte der Adressbytes von VO
104	ADD A, VXL	Addiere Reg. A mit Highbyte der Adressbytes von VX
105	ADD A, VYL	Addiere Reg. A mit Highbyte der Adressbytes von VY
106	ADD A, VOL	Addiere Reg. A mit Highbyte der Adressbytes von VO
107	SUB VXA	Subtrahiere Pagebyte von VX von Register A
108	SUB VYA	Subtrahiere Pagebyte von VY von Register A
109	SUB VOA	Subtrahiere Pagebyte von VO von Register A
110	SUB VXH	Subtrahiere Highbyte der Adr. bytes von VX von Reg. A
111	SUB VYH	Subtrahiere Highbyte der Adr. bytes von VY von Reg. A
112	SUB VOH	Subtrahiere Highbyte der Adr. bytes von VO von Reg. A
113	SUB VXL	Subtrahiere Lowbyte der Adr. bytes von VX von Reg. A
114	SUB VYL	Subtrahiere Lowbyte der Adr. bytes von VY von Reg. A
115	SUB VOL	Subtrahiere Lowbyte der Adr. bytes von VO von Reg. A
116	ADD HL, VXHL	Addiere HL mit Adressbytes von VX
117	ADD HL, VYHL	Addiere HL mit Adressbytes von VY
118	ADD HL, VOHL	Addiere HL mit Adressbytes von VO
119	ADD VXHL, HL	Addiere Adressbytes von VX mit HL
120	ADD VYHL, HL	Addiere Adressbytes von VY mit HL
121	ADD VOHL, HL	Addiere Adressbytes von VO mit HL
122	SBC HL, VXHL	Subtrahiere Adressbytes von VX von HL
123	SBC HL, VYHL	Subtrahiere Adressbytes von VY von HL
124	SBC HL, VOHL	Subtrahiere Adressbytes von VO von HL
125	SBC VXHL, HL	Subtrahiere HL von Adressbytes von VX
126	SBC VYHL, HL	Subtrahiere HL von Adressbytes von VY
127	SBC VOHL, HL	Subtrahiere HL von Adressbytes von VO



Stephan Haller, Schützheider Weg 32, 51465 Bergisch Gladbach, Telefon: 02202/31052.  
Email: Nomad@change.eun.de -oder- cadofitzba@aol.com

## FOU(T/D)MELDINGEN OP DE PC

Door de duitse SPECTRUM PROFI CLUB,  
werden wij ingelicht over de niet beschreven foutcodes die op de  
PC kunnen verschijnen. We hebben ze voor U even naar het  
Nederlands vertaald.

Fout      Betekenis

- 001      Operating System (WINDOWS ?) wordt geladen, Uw systeem  
         verkeerd in gevaar!
- 002      Geen fouten, tenminste tot op heden!
- 004      Posing tot multitasking, programma's zijn met elkaar  
         verward geraakt.
- 005      Kwaadaardige fout, onbekend Operating Systeem  
         aangetroffen.
- 006      Ontoereikende schijfruimte, maak minstens 200Mb vrij.
- 007      Onverklaarbare fout, informeer ons hoe dit ontstaan is.
- 008      Toetsenbord geblokkeerd, probeer of het u bevalt.
- 009      Onoplosbare fout, het systeem wordt vernietigd.  
         Koop een nieuw syteem, de garantie van de oude is bij  
         deze komen te vervallen.
- 010      Gebruikersfout, is niet onze schuld. Erewoord!
- 011      Operating System overschreven, installeer opnieuw.  
         Onze excuses voor het ongemak!
- 014      Ontoelaatbare fout, u heeft het recht niet om deze fout  
         te krijgen. Bij herhaling kunnen gerechtelijke stappen  
         ondernomen worden.
- 015      Muisfout, Muisdriver is niet geïnstalleerd.  
         Druk op de linker muistoets om verder te gaan!
- 016      Out of error-error! Er zijn te veel fouten opgetreden.  
         De foutmeldingen zijn op.

Er schijnen nog meer niet beschreven fouten te zijn. Indien deze  
bekend worden, worden wij hierover ingelicht.

Mocht u thuis ook nog enkele fouten zijn tegen gekomen, laat het  
ons dan weten.

Johan Koelman.

- 017      External Failure 1:  
         De gebruiker weet of kan (en wil!) meer dan het  
         operating systeem aankan.
- 018      External Failure 2:  
         De gebruiker is te slim om het programma te kunnen  
         volgen en/of te gebruiken.
- 019      System Failure:  
         Het door u gewenste progamma is in één keer geladen en  
         vertoont geen problemen. U bent vergeten Windows te  
         laden!

JWK.



## MACHINECODE ONDERWIJZER (1)

Uit de reactie op de enquête's blijkt telkens weer de behoefte naar eenvoudige Machinecode uitleg, eventueel gecombineerd met korte routines.

Met het brengen van het schriftelijk deel van de programma-serie "Machinecode Tutor" beogen we, naast het voldoen aan uw wensen, uw aandacht te vragen voor deze op cassettes verschenen cursus. Helaas weten we niet waar deze cursus nu nog commercieel verkrijgbaar is, maar even rondvragen bij andere Sinclair liefhebbers zou best iets op kunnen leveren. Hoe dan ook, Veel plezier, en steek er wat van op.

JWK.

De lessen van de Machine code tutor zijn verdeeld in 4 groepen. Bij het laden van een deel verschijnt een introductie op het scherm. Het menu voor de betreffende groep lessen kunt u oproepen door indrukken van SPACE. Door in een les of een voorbeeld BREAK in te drukken kunt u terugkeren naar het menu.

Hier volgt een volledige lijst van alle toetsen die in de verschillende fasen worden gebruikt met een volledige beschrijving van hun functie.

### MENU

ENTER : Start de in het menu aangegeven les of voorbeeld

SPACE : Indrukken van deze toets laat u kiezen welke les wordt gestart.

### LESSEN EN VOORBEELD TEKSTEN

SPACE : Intoetsen van deze toets beeld de de volgende pagina af. Aan het einde van een les brengt dit u terug naar het menu.

BREAK : Brengt u naar het menu terug.

SIMULATOR : geeft "auto run" op alle toetsen.

RUN : Wist alle registers en start het programma.

Elke toets: Voert tijdens het runnen de aangegeven functie uit.

STOP : Dit stopt het runnen van het programma.  
(A + Symbol Shif )

---

# SGG BULLETIN

---

EDIT : Hierdoor komt u in de editor, waarna u uw programma kunt wijzigen of aanpassen. (l + Symbol Shift)

g : Overschakeling van decimaal op HEX notatie van het display. Deze functie werkt alleen wanneer het programma niet runt.

BREAK : Terug naar het menu.

Wanneer u een oefening wilt herhalen, nadat hij beëindigd is drukt u RUN in.

EDITOR : Geeft "auto run" op alle toetsen.

Cursor toetsen:

Indrukken hiervan verplaatst de cursor in de richting van de pijlen.

& : Dit wist het editor scherm en geheugen en brengt de cursor links-boven.

SPACE : Tabuleert naar het begin van de instructie wanneer de cursor in de label kolom is, in andere gevallen wordt een spatie ingevoerd.

ENTER : Brengt de cursor naar het begin van de volgende regel.

DELETE : Wist het karakter links van de cursor en verplaatst de cursor 1 positie naar links.

STOP : Start het assembleren van het programma op het scherm. Wanneer het assembleren zonder fouten verloopt gaat het programma over in de simulator. Wanneer een fout wordt gevonden wordt een foutmelding gegeven bij de betreffende regel van het programma en wacht de editor op correctie van de fout.

BREAK : Brengt u terug bij het menu.

De editor beeld letters altijd als hoofdletters af, maar gebruik van de CAPS SHIFT is niet nodig.



## INTRODUCTIE VAN DE LESSEN

De Complete Machine Code Tutor omvat 35 lessen die tezamen alle instructies behandelen van de Z80, de processor van de ZX Spectrum.

Alle lessen worden op het scherm uitgebreid behandeld en worden veelal gevolgd door voorbeeld programma's, die u als oefeningen kunt gebruiken door ze zelf te wijzigen. Er bestaat geen gevaar dat u hierbij het systeem laat "crashen".

Er volgt nu een lijst van de titels van alle lessen met daarbij aangegeven welke lessen gevolgd worden door voorbeelden. Onder iedere titel volgt een opsomming van instructies die voor u duidelijk zullen worden naarmate u vordert met de MC Tutor. Deze opsommingen zijn bedoeld als een register van instructies, dat u kunt gebruiken voor herhalingen, zonder dat u daarvoor de scherm teksten hoeft te raadplegen.

Les 1 : Registers and memory (Registers en geheugen)  
Hierin worden de A, B, C, D, E, H en L registers behandeld.

Les 2 : Simple load instructions (Eenvoudige laad instructies)

De volgende instructies worden behandeld:

LD r,r' : Waarin r en r' kunnen zijn: A, B, C, D, E, H en L.

LD r,n : Waarin n een getal is van 0 tot 255.

LD A,(nn) : Waarin nn een geheugen locatie is van 0 tot 65535.

LD (nn),A

Deze les wordt gevolgd door voorbeeld programma's.

Les 3 : Register pairs (Register paren)  
Een getal in een register paar is 256 maal de meest significante byte + de minst significante byte.

---

# SGG BULLETIN

---

De volgende instructies worden behandeld:

LD dd,nn : Waarin dd een register paar BC, DE of HL is, nn is een getal van 0 tot 65535.

LD dd,(nn) : Waarin nn een geheugen adres is van 0 tot 65535.

LD (nn),da

EX DE,HL : Verwisselt de inhoud van de registers.

Deze les wordt gevolgd door voorbeeld programma's.

Les 4 : Indirect addressing (Indirecte adressering)

De volgende instructies worden behandeld:

LD r,(HL) : Waarin r een enkel register A, B, C, D, E, H of L is.

LD (HL),r

LD A,(BC)

LD A,(DE)

LD (BC),A

LD (DE),A

Deze les wordt gevolgd door voorbeeld programma's.

Les 5 : Additions and the carry flag  
(Optellingen de carry vlag)

Optellingen met de accumulator en het HL register paar worden behandeld, zowel als ADD met carry.

De volgende instructies worden behandeld:

ADD A,n : waarin n een getal is van 0 tot 255.

ADD A,r : waarin r een willekeurig register is.

ADD A,(HL)

ADD HL,BC

ADD HL,DE

ADC A,n : optellen met carry.

ADC A,r

ADC A,(HL)

ADC HL,BC

ADC HL,DE

Deze les wordt gevolgd door 2 voorbeeld programma's.



If you reckon you know all there is to know about Z80 machine code, prepare for a shock. As well as the documented instructions, there are over a hundred that have been hidden away. David Jones uncovers them and adds the missing info. Now you really can have your chip with everything!

## HIDDEN EXTRAS

Inside every Spectrum lives a chip — the Z80 created by Zilog — and with it there are nearly seven hundred well-documented instructions for the machine code programmer to play about with. Much more interesting though, are the instructions that Zilog left in the Z80 but never bothered to document. Why? Because they couldn't be guaranteed to work in every chip. In all, there are 102 of them and you'll find that they're being used more and more in many commercial games. You may even have hacked into one such program and just not been able to make any sense of it.

There are two reasons for this increasing popularity with professional programmers. Firstly, these instructions are a very neat way of fooling the hacker. As the only assembler on the market that can cope with any of these instructions is Picturesque's, hackers can have a rough time of it trying to decipher what's going on if they're only armed with a standard assembler. Secondly and even more important, is that these instructions perform operations that can't be

carried out anywhere near as easily using the standard instruction set.

There has to be a catch, you're saying. And yes, there is — sort of. As Zilog hasn't provided any official info, a new version of the Z80 minus some or all these extra commands can come out at any time. It may even be inside your Speccy!

### BUG BYTES

The first ten of these hidden instructions are the result of a Z80 'bug'. In the original specifications for the Z80, Zilog included SLA (Shift Left Arithmetic), SLL (Shift Left Logical) SRA (Shift Right Arithmetic) and SRL (Shift Right Logical). Unfortunately, the SLL instruction didn't work, so rather than re-design the Z80 or admit that there was a non-working instruction, Zilog simply pretended that it just didn't exist. Sneaky, eh? The way that SLL fails to work is pretty simple but it can be put to some use in certain circumstances. It should shift a byte left and set bit zero to 0. In reality, it shifts the byte left and then sets bit zero to 1.

In all there are ten SLL instructions since it can operate on data supplied by various methods. It's doubtful that you'll find any earth-shattering uses for these instructions as the Z80's already supplied with a pretty



comprehensive set of 8-bit shifts and rotates. Still, you can now include these 'bug' instructions in the SLS (Shift Left Set) mnemonic.

If you're into scribbling in the margins of your technical manuals, turn to page 136 of the *Spectrum Basic Programming Manual*. You can now fill in the space from code 48 with the new mnemonic SLS eight times, followed by the appropriate registers in the same order as they appear for SLA, SRA and SRL. That takes care of eight out of the first ten. The remaining two are the same instructions using data supplied by the index registers. Now, using our new mnemonics, they are SLS(IX+0) and SLS(IY+0).

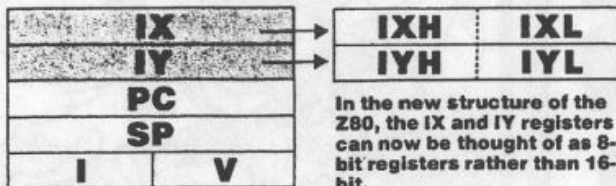
Speccy speed-freaks are going to find the next chunk of 92 instructions very useful for adding that extra zip to their machine code. If you take a look at the layout of the Z80's internal structure, you'll see that the IX and IY registers are split into four 8-bit registers instead of the normal two 16-bit registers. This split is the basis for the remaining instructions.

Now, the Z80 was designed in such a way that it includes all of the instructions available on the 8080 processor plus its own additional instructions and extra registers. So, amongst the extra features are bit-handling, use of the IX and IY registers and a whole series of 'special' functions, including LDIR. To make use of these new features, the processor has some one-byte 'flag instructions' that perform the following:

## BANK NOTES

The old and the new. The internal structure of the Z80 performs a metamorphosis without the aid of props.

A	F	A'	F'
B	C	B'	C'
D	E	D'	E'
H	L	H'	L'



## Hex Byte Function

- CB** Tells the Z80 to expect one of the following instructions in the next three bytes: RLC, RRC, RL, SLA, SLS (see above), SRL, BIT, RES or SET.
- DD** Tells the Z80 to use the IX register instead of the HL for the next instruction.
- ED** Tells the Z80 to expect a special instruction next. This way it has some quite useful block move and compare instructions.
- FD** Tells the Z80 to use the IY register instead of HL for the next instruction.

You'll find the secret of the vast majority of these extra instructions is tucked away in the DD and FD bytes. The IX

## HIDDEN EXTRAS

and IY register instructions are simply HL register instructions prefixed with DD or FD and with the ability to include an offset byte. If DD or FD is placed in front of an instruction that uses either the H or L register, then you'll

find that in a large percentage of cases the appropriate portion (IXH, IXL, IYH or IYL) of the IX or IY register will be used instead. And in all cases the split index register instructions change the flags in the same way as the equivalent

H or L instruction.

As I said before, I can't guarantee that all these instructions will work with every Spectrum but I've yet to find one that won't. If it turns out to be yours, I want to be the first to know.

## THE EXTRA SET

This is your new found Z80 instruction set. The first ten are the SLL instruction set.

MNEMONIC	HEXADECIMAL	DECIMAL	MACHINE CYCLES
SLS A	CB 37 ...	203 55 ...	8
SLS B	CB 30 ...	203 48 ...	8
SLS C	CB 31 ...	203 49 ...	8
SLS D	CB 32 ...	203 50 ...	8
SLS E	CB 33 ...	203 51 ...	8
SLS H	CB 34 ...	203 52 ...	8
SLS L	CB 35 ...	203 53 ...	8
SLS (HL)	CB 36 ...	203 54 ...	15
SLS (IX+DIS)	DD CB XX 36	221 203 DIS 54	23
SLS (IY+DIS)	FD CB XX 36	253 203 DIS 54	23

If you want to use the IY register in your own machine code programs, start the instruction with FD instead of DD. You'll also have to avoid most ROM calls and disable the interrupts with DI or you could have a few problems.

MNEMONIC	HEXADECIMAL	DECIMAL	MACHINE CYCLES
ADC A, IXH	DD 8C ..	221 140 ...	8
ADC A, IXL	DD 8D ..	221 141 ...	8
ADD A, IXH	DD 84 ..	221 132 ...	8
ADD A, IXL	DD 85 ..	221 164 ...	8
AND IXH	DD A4 ..	221 165 ...	8
CP IXH	DD BC ..	221 188 ...	8
CP IXL	DD BD ..	221 189 ...	8
DEC IXH	DD 25 ..	221 37 ...	8
DEC IXL	DD 2D ..	221 45 ...	8
INC IXH	DD 24 ..	221 36 ...	8
INC IXL	DD 2C ..	221 44 ...	8
LD A, IXH	DD 7C ..	221 124 ...	8

LD A, IXL	DD 7D ..	221 125 ...	8
LD B, IXH	DD 44 ..	221 68 ...	8
LD B, IXL	DD 45 ..	221 69 ...	8
LD C, IXH	DD 4C ..	221 76 ...	8
LD C, IXL	DD 4D ..	221 77 ...	8
LD D, IXH	DD 54 ..	221 84 ...	8
LD D, IXL	DD 55 ..	221 85 ...	8
LD E, IXH	DD 5C ..	221 92 ...	8
LD E, IXL	DD 5D ..	221 93 ...	8
LD IXH, A	DD 67 ..	221 103 ...	8
LD IXL, A	DD 6F ..	221 111 ...	8
LD IXH, B	DD 60 ..	221 96 ...	8
LD IXL, B	DD 68 ..	221 104 ...	8
LD IXH, C	DD 61 ..	221 97 ...	8
LD IXL, C	DD 69 ..	221 105 ...	8
LD IXH, D	DD 62 ..	221 98 ...	8
LD IXL, D	DD 6A ..	221 106 ...	8
LD IXH, E	DD 63 ..	221 99 ...	8
LD IXL, E	DD 6B ..	221 107 ...	8
LD IXH, IXH	DD 64 ..	221 100 ...	8
LD IXH, IXL	DD 65 ..	221 101 ...	8
LD IXL, IXH	DD 6C ..	221 108 ...	8
LD IXL, IXL	DD 6D ..	221 109 ...	8
LD IXH, XX	DD 26 XX	221 38 XXX	11
LD IXL, XX	DD 2E XX	221 46 XXX	11
OR IXH	DD B4 ..	221 180 ...	8
OR IXL	DD B5 ..	221 181 ...	8
SBC A, IXH	DD 9C ..	221 156 ...	8
SBC A, IXL	DD 9D ..	221 157 ...	8
SUB IXH	DD 94 ..	221 148 ...	8
SUB IXL	DD 95 ..	221 149 ...	8
XOR IXH	DD AC ..	221 172 ...	8
XOR IXL	DD AD ..	221 173 ...	8



## OPROEP: WIE HELPT, WEET, HEEFT

Nu of nooit, een zelfbouwproject.

Al jaren wil ik een zelfbouwprojectje doen, maar ontbreekt mij de kennis van de connector om zomaar hieraan te gaan knutselen. Zo wil ik, heel simpel, door middel van een OUT-opdracht LEDjes laten branden. Anderzijds lijkt het mij ook wel leuk om zelf een interrupt te maken door middel van handshaking. De softwarekennis heb ik in huis. Op school heb ik e.e.a al eens op een PDP11 geprogrammeerd. Op de ZX Spectrum is het er nooit van gekomen. Wie kan mij een beginopzet geven? Ik zelf dacht al aan de volgende schakeling, maar durf deze niet zonder meer aan de Spectrum te koppelen.

```

+-----+-----+-----+-----+-----+-----+-----+-----+
!       !       !       !       !       !       !       !       !
!   LED   LED   LED   LED   LED   LED   LED   LED   LED
!       !       !       !       !       !       !       !       !
GND  D0    D1    D2    D3    D4    D5    D6    D7

```

En met handshaking intrupts de GND door +5V en de LEDjes door schakelaars te vervangen.

Wie kan deze startende hardwaregebruiker helpen?

Johan Koelman  
W. v. d. Veldestraat 1  
5831 BW BOXMEER



## Produktnews

Ginge es nach der Fachpresse, ist der Spectrum schon seit 10 Jahren tot, ginge es nach den heutigen Jugendlichen, hat er nie existiert.

Ist der Spectrum wirklich am Ende? Mitnichten, die Realität sieht anders aus. Noch nie hatten Spectrum-User es so einfach, an eine große Zahl von Hard- und Software zu kommen und noch nie gab es so viele tolle und neue Produkte für diesen Rechner.

In den letzten 2 Jahren wurden wir geradezu mit neuer Software aus der GUS oder Osteuropa überschwemmt. Nun zieht auch SINTECH mit der Hardware nach. Die neuesten Produkte wollen wir hier vorstellen:

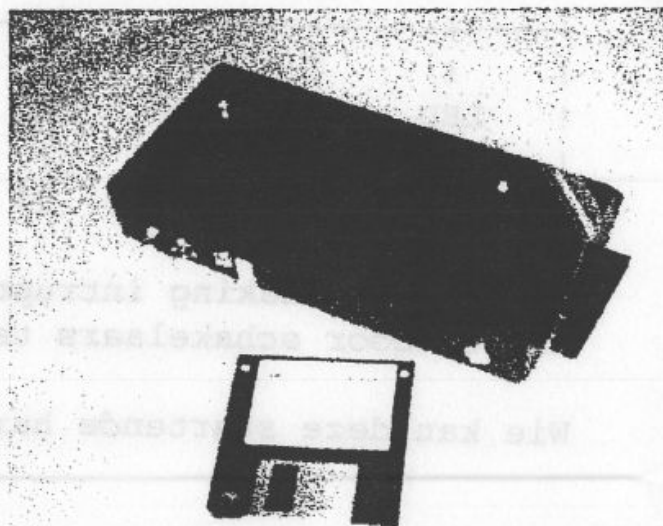
Produkte der **8BitCompany** im Vertrieb von SINTECH



### Das MB02+

Lange hat es gedauert, bis die Entwicklung vollendet war, aber seit etwa 1 Jahr befindet sich das MB02+ im Verkauf. Noch nie machte ein Interface solche Schlagzeilen, aber auch noch nie gab es ein Interface wie dieses. Da ja bereits so vieles darüber gesagt wurde, hier nur noch einmal die wichtigsten Features:

- Diskettenkapazität: bei HD 1.86 MB, bei DD 840 KB
- Geschwindigkeit: HD: 40-50 K/Sek., DD 25 K/Sek.
- Interner Speicher: EPROM 2K (max. 64K)  
SRAM 128K (max. 512K)
- Erstklassiges DOS
- Kassetten-kompatibles System
- Bis zu 256 Directories auf einer Diskette
- Eingebauter DMA-Chip
- Lieferung mit Echtzeit-Uhr möglich
- Eingebauter Joystick-Port und Drucker-Schnittstelle



Als einer der größten Vorteile des MB02+ ist das Betriebssystem zu sehen. Es wurde von Busysoft in der Slowakei entwickelt und ist sehr anwenderfreundlich. In der Tat muß man kaum neue Befehle lernen, den durch das Kassetten-kompatible System kann man die Diskette wie eine Kassette benutzen. Mehr noch: Man kann Software ohne sie zu ändern von Kassette auf Diskette kopieren und dann von Diskette laden.

Auch der DMA-Chip kann erstaunliches leisten. Lernt man ihn zu programmieren, stehen einem Welten in schnellere Dimensionen offen. Es wurden auch schon Programme speziell für diesen Chip geschrieben und einige Hacker haben sich auch schon an Spiele gemacht, die durch den DMA bis zu 50% schneller laufen.



## Data-Gear

Der Data-Gear ist ein neues Produkt der 8BC für alle, die sich aus verschiedenen Gründen nicht für ein MB02+ entscheiden können. Dieses Interface enthält den DMA-Chip des MB02+ und ermöglicht damit die Möglichkeiten, auch ohne das Diskinterface die Geschwindigkeit, die normalerweise nur das MB02+ bietet, zu erleben.

Man kann damit die DMA-Demos oder Spiele auf jedem Spectrum laufen lassen und auch die wundervollen "Full-Screen-Multicolour" Bilder anschauen, die der neue "Multitech" Standard bietet, anzuschauen. Der Data-Gear steht momentan kurz vor Vollendung und wird in Kürze erhältlich sein. Hier einige Bilder vom neuen MULTITECH-Multicolour Farbmodus:



## Produkte von Kompakt-Servis im Vertrieb von SINTECH:

### Didaktik Kompakt

Der Didaktik Kompakt ist der letzte Spectrum der noch professionell hergestellt wird. Er ist zwar nicht von Sinclair, dafür aber in der Tat sehr kompakt. In einem formschönen Gehäuse befinden sich 48K RAM, 32K ROM, ein 3,5" Diskettenlaufwerk und das dazugehörige Betriebssystem. Da dieser Computer aus der Slowakei stammt, ist keine deutsche Dokumentation vorhanden. Das Disketten-Betriebssystem besteht aber nur aus sehr wenigen Befehlen und ansonsten ist der Didaktik genauso wie ein echter Sinclair Spectrum. Der Didaktik hat kein externes Netzteil, sondern wird direkt an die Steckdose angeschlossen. Auch Monitor-Anschluß oder Fernseheranschluß sind kein Problem, den der Didaktik verfügt über TV, Audio, Video und sogar Scart-Anschluß. Durch seine Kompaktheit ist dieser Rechner ideal auf Reisen oder auch um flexibler zu sein.



## 128k Aufrüstung

Man kennt es: Den 48k hat man schon seit Jahren lieb gewonnen, aber die Möglichkeiten des 128k reizen schon. Nun, man kann zwei Rechner haben, alternativ bieten wir aber auch etwas neues: Den 128k Umbau. Bei Kompakt-Servis wurde eine kleine Aufrüst-Platine entwickelt, die sogar in einem Gummi-48k Platz hat und die vollen Speicher-Möglichkeiten enthält. Die Platine kann aber nur von absoluten Profis angeschlossen werden, da zum Anschluß der Platine auch noch andere Veränderungen vorgenommen werden müssen, z.B. sämtliche Speicherchips müssen ausgetauscht werden. Wir bieten jedem geduligen User an, diesen Umbau zu übernehmen.

## Proface AT

Die Tastatur, zumindest vor Einführung des Spectrum +2 ein Leidthema des Spectrums. Spätestens mit dieser Erfindung muß kein Spectrum-User mehr neidisch auf die Tastaturen anderer Rechner schauen, den das Proface macht es möglich, jedes AT-Keyboard an den Spectrum anzuschließen. Das Handling ist sehr einfach, die externe Version (mit durchgeführtem Bus) wird einfach auf den Spectrum-Bus aufgesteckt und die Tastatur dann in das Interface. Egal ob Infrarot-Tastatur oder ein ganz normales PC-Keyboard, die Tastatur muß von nun an nicht mehr direkt auf dem Rechner sitzen. Zusatztasten wie Zehnerblock, F-Tasten und auch der Warmstart wurden eingebaut.

## Melodik

Melodik ist ein AY-Soundinterface. Mit diesem Interface kann auch ein 48k Spectrum AY-Sounds (AY ist der Soundchip des 128k) abspielen. Das Interface hat einen durchgeführten Bus und einen kleinen Lautsprecher. Alternativ kann statt des Lautsprechers auch ein Verstärker angeschlossen werden. ACB-Stereo ist bei dem Verstärkeranschluß ebenfalls bereits eingebaut.

## Mice Maus

Diese Maus kann an jeden Kempston-Port angeschlossen werden, der über eine 5V-Leitung verfügt (wie benötigt für Dauerfeuer und sonstige Zusatzfunktionen von Joysticks). Jedes Spiel oder Programm, das bisher über Kempston-Joystick gesteuert werden konnte, kann von nun an ohne weitere Änderungen mit dieser Maus benutzt werden.

## Produkte von Romantic Robot

### Multiface One

Das Hack-Modul für den 48k Spectrum. Man kann damit Programme unterbrechen und abspeichern, um sie später an exakt der gleichen Stelle weiterzuspielen. Auch nur der aktuelle Bildschirm kann gespeichert werden. Außerdem ist ein Toolkit zur Eingabe von Pokes ebenfalls eingebaut.

### Multiface 128

Wir konnten einen letzten Posten der Hack-Module für die 128k Rechner (nicht +3) erwerben. Es handelt sich um die letzte Version mit voller +D Kompatibilität.



**Die Preisliste****Neu:**

MB 02+ Standardversion, 2K Eprom, 128k SRAM	290,00 DM
+ Extra 128k SRAM	55,00 DM
+ Extra 256k SRAM	97,00 DM
+ Extra 384k SRAM (maximum)	135,00 DM
Echtzeit-Uhr (RTC-Chip)	39,00 DM

**Data Gear**

noch nicht verfügbar

Didaktik Kompakt 48k	250,00 DM
Didaktik Kompakt mit 128k Aufrüstung	350,00 DM
Proface AT Extern	100,00 DM
Proface AT intern	90,00 DM
Melodik ohne Gehäuse	40,00 DM
Melodik mit Gehäuse	50,00 DM
128k Aufrüstung (nur für Hardware-Profis)	35,00 DM
128k Aufrüstung inkl. Einbau (Platine einschicken)	100,00 DM
Mice Maus	40,00 DM
Spectrum +2A/+2B Platine voll bestückt	65,00 DM
+2 Kassettenrekorder	50,00 DM
SPECDROM - Soundinterface mit Drumkit	50,00 DM
Programmierbares Joystick-Interface	20,00 DM
Floppylaufwerk (1,86 bei MB02, 720k bei Opus, 780k f. +D) gew. Typ ang.	60,00 DM
Powerkit (wandelt +3 Netzteil um in Stromvers. f. 2 Floppy's + MB02)	15,00 DM
Netzteil für +2A/B oder +3	30,00 DM
Netzteil für +2 (auch 48k und 128k)	40,00 DM
Floppykabel für 2 Laufwerke	7,00 DM
Microdrive	25,00 DM
Interface I	45,00 DM
Multiface 128 (läuft auch am 48k)	50,00 DM
Abdeckhaube 48k+/128k	12,50 DM
Abdeckhaube +2/+3	14,00 DM
48k Tastaturfolie	20,00 DM
Farbband original STAR LC 10 im Doppelpack, 2 Stück	10,00 DM

Preise zuzüglich Porto. Alle Angebote solange Vorrat reicht. Mitglieder des SUC bekommen 10% Rabatt auf Hardware (ausgenommen Produkte von 8BC und Kompakt Servis).

**SINTECH**

Gastäckerstr. 23

70794 Filderstadt

Tel./Fax: 0711/775033

e-mail: [sintech@online.de](mailto:sintech@online.de)<http://www.online.de/home/sintech>

**PROGRAMMEER WEDSTRIJD !!**

---

In navolging van het blad Impuls heb ik de volgende programmeerwedstrijd bedacht.

in BASIC is er een 8-cijferig willekeurig getal X dat de eigenschap heeft dat de cijfers 1 t/m 8 er elk 1 keer in voorkomen. De bedoeling van de wedstrijd is om een zo SNEL mogelijk (dus niet zo kort mogelijk) programma te schrijven dat dit getal vind.

De enige voorwaarde die gesteld wordt is dat de variabele X ALLEEN gebruikt mag worden in een IF-statement (bv. IF A=X THEN STOP). Anders zou het snelste programma zijn:

```
10 PRINT X
```

Een voorbeeldprogramma geef ik alvast (dit is zeker niet de snelste)

```
10 FOR F=10000000 TO 99999999: IF F<>X THEN NEXT F
20 PRINT F
```

Aan jullie de taak een sneller programma te maken. Elk programma zal met 10 verschillende waarden van X getest worden. De gemiddelde tijd van deze zal genomen worden voor de eindklassering.

Stuur je programma's naar

Johan Koelman  
W v.d. Veldestraat 1  
5831 BW BOXMEER



## KLEINE MACHINECODE ROUTINES

Deze scroll methode is een ideale hulp in een programma om het scherm te wissen zonder CLS te gebruiken, bovendien print de computer elke nieuwe regel onderaan het scherm en schrijft op deze wijze vanaf onderen het scherm vol.

Dit scroll effect is bijvoorbeeld ideaal te gebruiken voor een rol-lende krant. Met bepaalde PAUSES in het programma kunt U zelfs nog de leessnelheid regelen.

U heeft vast wel de nodige fantasie om er iets leuk van te maken.

```

10 REM gebruik van speciale sc
rol-techniek
20 LET a$="@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@
30 LET b$="xxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxx"
40 FOR n=1 TO 4
50 PRINT a$;a$;b$;b$;a$
60 NEXT n
70 REM scroll
80 POKE 23692,42
90 PRINT .....;b$
;a$
100 PRINT "scroll"
110 PRINT "scroll"
120 PRINT "scroll"
130 REM hu scherm wissen
140 REM met deze methode.
150 FOR n=1 TO 4
160 PRINT a$;a$;a$;a$
170 NEXT n
180 FOR n=1 TO 100
190 NEXT n
200 POKE 23692,42
210 PRINT .....
220 PRINT "scherm gewist van on
der > boven "
```

## Renumber MCode

Deze MCode routine zorgt ervoor dat Uw BASIC programma instant wordt hernummerd. Echter de GOSUB's, GOTO's en RESTORE's moet U wel zelf aanpassen.

De hele routine is 40 bytes lang.

In de listing is zowel de 16K als 48 K versie opgenomen.

Het gebruik van de routine is vrij eenvoudig: U POKE't eerst de aangegeven adressen zoals in de listing aangegeven en gebruikt daarna de RANDOMIZE USR opdracht.

```

10 CLEAR 65326: REM (32558)
20 FOR N=65327 TO 65366:
   REM (32559 TO 32598)
30 READ A: POKE N,A
40 NEXT N
50 DATA 0,237,107,83,92,1,10,0
,17,100,0,213,229,237,91,75,92,1
75,237,82,225,209,200,197,114,35
,115,35,78,35,70,35,9,235,193,9,
235,24,228,201
60 REM DEZE ROUTINE HERNUMMERT
   UIT ZICHZELF HET GEHELE
   PROGRAMMA EN START DE
   NIEUWE LIJNNUMMERS OP LIJN
   100 IN STAPPEN VAN 10
70 REM OM DE NIEUWE PROGRAMMA-
   LIJNEN TE LAIEN BEGINNEN OP
   LIJN 10;
POKE 65336,10 (POKE 32568,10);
   OM IN STAPPEN VAN 5 TE
   HERNUMMEREN;
POKE 65333,5 (POKE 32565,5)
80 REM OM TE HERNUMMEREN NAAR
   LIJNNUMMERS GROTER DAN 127
   GEBRUIKT U;
   L=LIJNN0.
(PKE 65336,(L-INT(L/256))
   (16K=32568,.....)
   EN DAN
POKE 65337,(INT(L/256))
   (16K=32569,.....)
90 REM OM TE HERNUMMEREN ;
   RAND USR 65328
   (RAND USR 32560)

```



## Vrij geheugen MCode

Deze vrij korte MCode routine geeft U direct het aantal vrije bytes dat U nog kunt gebruiken.

De routine is slechts 15 bytes lang.

Voor de 16K spectrum staat alles tussen haakjes.

```
10 CLEAR 65350 : REM (32583)
20 FOR N=65351 TO 65364: REM (
32584 TO 32598)
30 READ A: POKE N,A
40 NEXT N
50 DATA 33,0,0,57,237,91,101,9
2,167,237,82,229,193,201
60 REM ROUTINE ACTIVEREN ;
PRINT USR 65351 (32584)
```

## **IM MEMORIAM: A. J. SCHONEVELD**

Volkomen onverwacht bereikte ons het bericht van het overlijden van een van onze leden,

A.J. Schoneveld

En hoewel het gemis aan zo'n humorvol en actief lid, nog vol ideeën rond zijn en onze hobby, een zware klap betekent voor onze vereniging, valt zij volledig in het niet bij het verlies dat zijn nabestaanden nu ondervinden.

Wij willen die dan ook via deze weg ons medeleven betuigen, en ze de kracht op dit verlies te kunnen dragen toewensen.

Bestuur SGG-Groningen/Assen

**ZOMAAR EVEN EEN ARTIKELTJE**

---

Men heeft mij ook weer gevraagd, om wat te schrijven voor het clubblad, en uiteraard heb ik daar mee ingestemd. Want waarom zou je het niet doen.

Je kan schrijven wat je wilt, want de redactie zit toch meestal om kopij verlegen, zodat het ze eigenlijk niets uitmaakt, wat er binnen komt, als er maar iets binnenkomt. En commentaar vanuit de leden komt er ook nauwelijks (ten minste, naar mijn kant uit), dus daar hoef je ook niet bang voor te zijn.

Nou, ik drijf af. Toen ik dus werd gevraagd, een stukje te schrijven voor het clubblad, ben ik meteen maar aan de slag gegaan (na 2 weken).

Maar ik heb dus nog geen flauw idee waar het over moet gaan. Maar meestal kol ik daar  
Hllll, de letter 'l' is  
gebruik

ik nu laar even de  
leest lisschien een  
haar went u wel  
nu gaat he 'h' er  
Oke, voor 'h'  
gebruik ik wel een

worht wel een beetje bont zo. Laar  
we laten ons niet visten. O gut, Laar gaat he 'v'. Laar haar he 'v' voor  
gebruiven.



vanzelf wel op.  
kapot. Daarol

letter 'l'. Het  
beetje raar, laar  
aan. O nee he,  
ook wel aan.

'h' voor. Het

Oppassen nu, lat iv geen schunnige woorden lrijg. Het is wel  
lastig tivven zo, als he letters niet werven. Het is net, of iv een vunst  
gebit in heb. Ach, hit worht nivs zo. Als iv zo hoorga, is er stravs  
nivs leer over van lijn tevst. hit was hus een zeer vort artivel. Volgenhe  
veer beter.

**Stefan Postela**



# SGG BULLETIN

## MULTIVEEL MULTIFACE POKES (3)

ATF (Advanced Tactical Fighter)	Set number of lives	32815	num
	Infinite lives	32990	0
	No weight limit	35717	0
		35718	0
	Always full speed	36451	62
		36452	100
		36453	0
Athena 48	Mega jumps	50267	0
	Infinite lives	51212	0
	Infinite time	55268	61
Atic Atac	Infinite lives and energy	36519	0
		35353	0
		39092	92
Attack of the Killer Tomatoes	Infinite time	25323	0
	Infinite lives	49433	81
ATV (All Terrain Vehice) Simulator	Infinite lives	60250	0
	Infinite time	53718	201
Auto Mania	Set number of lives	53801	num
	Infinite lives	64968	167
Avalon	???	58316	201
Avenger	Infinite kwon-calls	41619	24
	Infinite lives	58294	0
	Infinite time	51919	201
	200 keys	41052	201
Ball Breaker	Infinite lives	37888	0
		37952	0
	Infinite missiles	39844	0
	Set starting screen	35685	0
		32880	num
Ball Crazy	Set number of lives	28086	num
	Infinite lives	28488	0
	Immortality	32995	0
Barbarian	Infinite lives	50914	0
	Set number of lives	51005	num
Barbarian 1	Infinite lives	37480	12
Barbarian 2 48	Infinite lives	38508	0
	Infinite energy	40159	0
Basil the Great Detective	Infinite lives	41296	0
		41968	201

# SGG BULLETIN

Batman	Infinite lives	36798	0
	Infinite big jumps	37512	166
Batty	Set number of lives	47633	num
	Infinite lives	48440	0
Bazooka Bill	Invincibility	29489	201
	Infinite lives	41931	0
BC Bill	Infinite lives	47589	201
Beach Buggy	Infinite fuel	45878	0
Beach Head 1	Set number of lives	32963	num
Beam Rider	Infinite lives	51009	0
Bearbovver	Set number of lives	38401	num
	Set number of bombs	38400	num
Bedlam	Infinite lives	58691	0
		59430	0
		59783	0
		59972	0
		60069	0
	Stop balls	30037	0
Bestial Warrior	Infinite lives	32417	0
		32418	0
		32419	0
		32610	0
		32611	0
		32612	0
Beyond the Ice Palace 48	Infinite lives	38279	0
Big Foot	Infinite lives	55189	0
Big Trouble in Little China	Infinite lives	30236	183
Bionic Commando 128	Infinite lives	34690	0
	Set number of lives	34741	num
	Infinite time	34200	0
		34201	0
Birds and the Bees 1, The	Infinite lives	37852	255
	Set number of flowers * 5	37088	num
	Set start level (0-8)	37664	num
	Infinite time	31981	0



## A SMALL LESSON IN TIME

We'll begin this article with a look at time. The 24 hour day, seven day week, 52 week year has not been in existence for, well, all of time.

Most early calendars were based on the moon, which takes roughly 29.5 days to circle the earth. The ancient Babylonian calendar, adopted by the Hebrews, was entirely lunar based, with 12 months of alternately 29 and 30 days and thus a 354 day year.

But, this calendar rapidly got ahead of the seasons, so it was adjusted to keep it roughly in step with the solar year. The Babylonians observed that 235 lunar months coincided with a cycle of 19 solar years. So, at around 380BC they decided that 7 out of every 19 years should have 13 months instead of 12.

The Jewish calendar still follows this lunar-solar arrangement, but the Muslim calendar remains lunar based, with regressive months.

### WEAK WEEKS

Weeks have always been an odd time out in mankind's measurement of the passing of the seasons. Days are related to the spin of the earth, months to the phases of the moon and years to the earth's journey around the sun. But weeks have been as varied as the people who used them. The ancient Greeks and Egyptians split their month into three ten-day weeks, while some African tribes used a four-day week, and the ancient Assyrians arranged their lives around a six-day week.

The seven-day week, which is now standard throughout the world, seems to have originated with the Babylonians about 4000 years ago. (The Jews already had a 7-days Ritme ! JWK)

It's days were names for the seven planets known at that time. Various countries have tried to change the habit, but have always been defeated by the force of custom. In 1792 the French introduced a decimal ten-day week to go with their new metric system, but it was soon abolished by Napoleon. Russian efforts to start a five-day week in 1929 and a six-day week in 1932 failed because people ignored them.

Dividing a day into 24 hours is another man made notion. The Sumerians divided their day into six watches (three day and three night), and the Babylonians divided it into 12 equal periods. It was probably the Egyptians who first picked a 24 hour day to mark their priestly ceremonies. They chose 24 because stargazers noticed that the night was marked off by the consecutive rising of 12 bright stars. Bent on nothing more than symmetry, they divided the day to match.

Then about 1400 BC the Egyptian pharaoh Amenophis decided that he needed a better way to tell the time at night.

---

## SGG BULLETIN

---

He had his chief engineer build a water clock. Shadow clocks primitive sundials - give the approximate time during daylight. The water clock worked by allowing water to trickle slowly out of a carefully graded bowl. By dipping his fingers in and feeling how many of the bowl's ridged markings were above the water-line, Amenophis could then for the first time tell roughly how long it was until sunrise.

The idea was developed by Greeks and Romans, whose water clocks (called CLEPSYDRAS) were able to measure fractions of hours.

There were many attempts to devise a calendar that approximated the actual time it takes the earth to circle the sun - 365 days, 5 hours, 48 minutes and 46 seconds - without losing the all important 12 month lunar year.

But, we must also add the factor of the earth slowing down - approximately one second per year. Just about every New Year's Eve you will get a news report about it. Ho hum, how original.

The ancient Egyptians had nearly succeeded, with a 365 day civil calendar. But it was apparently overlooked by the rest of the world until Julius Caesar adopted and modified it as the official calendar of Rome. His first reformed calendar began on January 1, 46 BC.

Caesar was also the first to introduce the idea of a four year leap year (with a 30 day February) to make up for the extra fraction of a day over 365 that the earth takes to complete its cycle.

Caesar's Julian calendar lasted well. It was much more accurate than its predecessors. But it was still 0.0078 days too long. And 1500 years later the accumulated error had added up to ten full days.

At this time, Pope Gregory XIII established the Gregorian calendar used today. In 1582 he solved the problem of then ten extra days - by simply omitting them. He decreed that in that year only, October 4 would be followed by October 15.

In Britain and the American colonies, September 3rd 1752 and the following ten days never happened. For in that year, Britain adopted the Gregorian calendar, and to get back in step meant losing 11 days. This decision caused riots at the time because people thought that the government had stolen 11 days of their lives. (Makes a change from stealing our money in 'taxes')

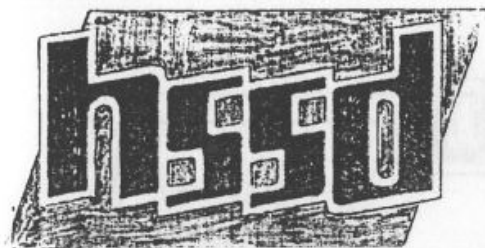
The reason why England and the colonies had to lose 11 days rather than ten was they did not adopt the Gregorian calendar until 170 years later, by which time it was one full day ahead of the 'real' year.

The Pope's Gregorian calendar was a reformed version of the Julian one. Gregory rearranged the lengths of the months into their modern versions and, more importantly, devised a way of overcoming the slight inaccuracy in the Julian calendar.

He decreed that in order to qualify as a leap year, a century year (100, 1500 and so on) had to be divisible not just by four but by 400. That is why the years 1700, 1800 and 1900 were not leap years, but 2000 is.

Justin Thorpe





SINCLAIR  
GEBRUIKERSGROEP  
GRONINGEN/ASSEN

6<sup>e</sup> Hollandse Sinclair & Sam dag

3 oktober '98

Geref. Kerk

Camminghalaan 5

Bunnik

10.00 u. - 15.30 u.

entree: gratis.



 Hill Sinclair

## BIJeenKOMSTEN



In het: RPV gebouw  
RABENHAUPTSTRAAT 45  
GRONINGEN  
Telefoon: 050 - 5261379

DATA SGG-GRONINGEN/ASSEN :

zaterdag	20 juni	van 14.00 - 17.30 uur
zaterdag	12 of 19 sept	van 14.00 - 17.30 uur (zie N.B.)
zaterdag	3 okt	van 10.00 - 15.30 uur BUNNIK

N.B. I.v.m. ziekte staat de datum in september nog niet helemaal vast, hoogstwaarschijnlijk wordt het de 12e !!

HCC-sgg bijeenkomsten in 1998:

zaterdag	18 juli	van 10.00 - 15.30 uur	in HOUTEN.
** zaterdag	3 okt	van 10.00 - 15.30 uur	in BUNNIK.
zaterdag	7 nov	van 10.00 - 15.30 uur	in HOUTEN.

In het Kantoorgebouw van de HCC: de Molen 24, te HOUTEN.

RIJSWIJK: Elke 2e en 4e woensdag van de maand, om 19.00 uur.  
adres: "Bovenveen", Spinozalaan 310, VOORBURG.

BUITENLAND:

SUC-treffen in de buurt van Stuttgart, datum nog niet bekend.

## REPARATIE ADRES

Jean Austeruhle,  
Sternwartstrasse 69,  
D-40223 Dusseldorf,  
GERMANY.  
tel/fax: 02131-69733  
off tel: 0211-395460

Peter Rennefeld  
Kupper 32  
D-52525 Heinsberg  
GERMANY





## ESGEEGEETJES



Gezocht: Tractor-feed voor een Smith-Corona printer  
type Fasttext 80.

D. ten Cate, Oranjestraat 20, 9671 BB Winschoten  
tel: 0597 - 412918

\*\*\*\*\*

Adres:  
Sinclair-QL Gebruikers Groep  
J.J. van de Molengraaf  
Mullerweg 17  
5264 JC Eindhoven  
+31 - (0)40 - 244 23 09

Informatie van u naar ons toe is van harte welkom.

\*\*\*\*\*

### IN- en VERKOOP van:

2e. handse apparatuur, boeken, handleidingen en/of software.

F.P. Vink, Zuiderkruis 49, Veenendaal. tel: 0318 - 550708 .

## Z80 EMULATOR V3.05



Hebt u vragen over de Emulator,  
of wilt u hem bestellen, dan kunt u schrijven aan:  
Gerton Lunter, P.O. Box 2535, 9704 CM GRONINGEN. (Netherl.)

Hebt u echter haast dan gaat het sneller via:

Brian Gaff, address: BG Services  
62 Roebuck Road  
CHESSINGTON KT9 1JX  
Great Britain



D R U K W E R K

RECHTIE-ADRES:

J.W. KONING  
MIEDEN 3

9866 TM LUTJEGAST

tel: 0594 - 512887

