

# Practical Computing

85p April 1984  
Volume 7 Issue 4

Win a  
£2,500  
RML 480Z outfit

## HEALTH

**Can a micro  
replace your doctor?**

Special guide to  
operating systems –  
CP/M, MS-DOS, Unix

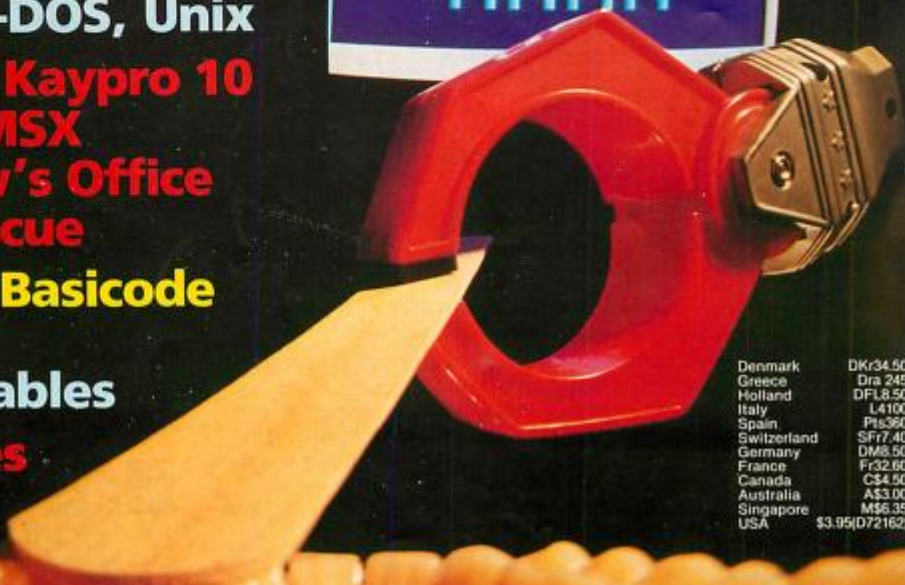
Reviews – Kaypro 10  
Yamaha MSX

Tomorrow's Office  
Delta, Rescue

**All about Basicode**

Top 20  
transportables

**BBC games**



Denmark	DKr34.50
Greece	Dra 245
Holland	DFL3.50
Italy	L4100
Spain	Pts360
Switzerland	SFr7.40
Germany	DM8.50
France	Fr32.60
Canada	C\$4.50
Australia	A\$3.00
Singapore	M\$6.35
USA	\$3.95(D72162)

# Programs

SAILORS listening to BBC Radio 4 just after the late shipping forecast recently may have been annoyed to discover an apparent fault in their radios which would suddenly burst into a wild banshee cry. This two-tone high-pitched whine will be familiar to anyone who has loaded programs from cassettes: it is Basicode, Basic's equivalent of Esperanto, and it is broadcast as part of BBC Radio 4's new computer magazine programme *The Chip Shop*.

The big advantage of Basicode programs is that not only can they be transmitted, recorded and later fed into home micros, but by using the system micro-to-micro communication is possible. There are two stages to the process of conversion into Basicode. First there is the problem of compatibility when transmitting data. Micros store programs on audio tapes by converting the bits — binary digits — into one of two frequencies, representing a binary 0 and 1 respectively. But the exact details and frequencies of audio signals generated vary widely.

## Frequencies used

Basicode first standardises the frequencies to be used at 1,200Hz for 0 and 2,400Hz for 1. Then Basic programs are converted character by character using the standard ASCII codes. For example A has an ASCII code of 65, which is 1000001 in binary form. ASCII uses only seven bits to represent the standard alphanumeric characters and Basicode sets a further eighth bit to be 1. After adding a logical 0 start bit to mark the beginning of the string and two logical 1s as stop bits to signal the end, the Basicode audio translation of the colon character appears as in the diagram below.

Using this Basicode translation technique, any Basic program can be transferred to another machine running Basicode. There remains, of course, the problem of dialects. Most machines have some quirk in their Basic implementation making them incompatible with other Basics.

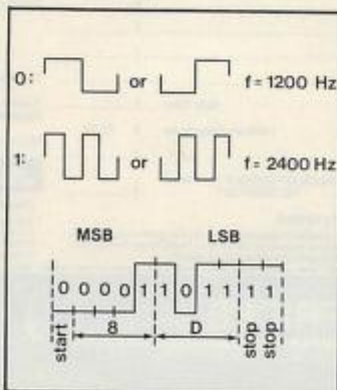
## Levelling down

Basicode's solution is drastic but it works. A subset from each machine's set of Basic commands like Goto, If-Then and Proc, which are common to most popular micros, has been chosen as standard. This is a levelling-down process, and what is gained in breadth of applicability is paid for in the reduced complexity of the programs. Machines for which Basicode translation programs can be run include the Apple, BBC Micro, the Commodore family, CP/M computers and the Tandy Models I and III. The permitted Basic commands and operations are listed in table 1.

A number of common operations are absent from this list. They include features such as clearing the screen and generating random numbers, the syntax for which varies from computer to computer. To



The original Dutch-English version of the Basicode 2 manual.



Transfer format built up in Basicode.

## Line-number scheme

The following is used to build up a Basicode 2 program:  
 Lines 0-999: standard routines. These routines are different for each computer and are therefore contained in the translation program.  
 Line 1000: first line of the Basicode 2 program. It must be in the form:  
 1000 A = (value): Goto 20; REM  
 program name  
 value is the maximum number of characters that can be used by all strings together. Line 20 is used to reserve memory space for the strings in those computers which need it.  
 Lines 1010-32767: the main program. There are no restrictions on this section, except that line numbers above 32767 are forbidden.

# on the air

Glyn Moody looks at how radio programmes for hobbyists have prompted the rise of Basicode.

provide them in a machine-independent way, special subroutines are constructed. So instead of writing CLS to clear the screen, Gosub 100 is used. The machine-dependent routine is then called at line 100.

There are a number of these special subroutines, all stored by Basicode convention in lines 0 to 999 of the Basic program. The main program itself then begins at line 1000, and can only use the permitted commands of table 1 plus calls to Gosub routines.

Typically the subroutines and translation programs are written in machine code. After the input program is translated, lines 0 to 999 are appended to provide the subroutines, which can then be called by the main body of the program.

Additional restrictions on the structure of a program are that the screen dimensions are fixed at 24 lines by 40 characters, that numeric variables are real and single precision, and that the name of a variable can have a maximum of two letters. There

ABS	LOG	TAB
AND	MID\$	TAN
ASC	NEXT	THEN
ATN	NOT	TO
CHRS	ON	VAL
COS	OR	+
DATA	PRINT	-
DIM	READ	*
END	REM	/
FOR	RESTORE	↑
GOSUB	RETURN	
GOTO	RIGHT\$	=
INT	RUN	<
IF	SGN	>
INPUT	SIN	<>
LEFT\$	SQR	<=
LEN	STEP	>=
LET	STOP	

Table 1. Basicode's permitted Basic commands and operations.

## Basicode

- Basicode is available in the U.K. for the Apple II and IIe, BBC Models A and B, Commodore 64, Pet and Vic-20, Sharp MZ-80A, Sinclair ZX-81, Tandy and the Video Genie.
- The Basicode kit, which includes subroutine programs on tape and a handbook, is available from Broadcasting Support Services, PO Box 7, London W3 6XJ for £3.95.
- *The Chip Shop* is broadcast at 5pm on BBC Radio 4 on Saturdays, and is repeated in an extended version at 11pm on Tuesdays. Basicode programs are broadcast after the 12.15am shipping forecast in the early hours of Sunday, Monday, Wednesday and Thursday.

are also a number of reserved variables used in the standard subroutines.

Basicode arose out of a desire by Teleac — the Netherlands' equivalent to the Open University — to supplement its computer course with software which could be distributed to students throughout the country. In 1979 Teleac approached the Dutch domestic service NOS and the first broadcasts of software, though only for specific machines, were given by the hi-fi programme *Hobbyscoop*.

In response to public pressure for software that could be used by a wider range of micro enthusiasts, an engineer at Philips developed a prototype of Basicode consisting of the translation program only. Basicode 2, which also includes the initial routines and standard command set, was first broadcast at the beginning of 1983.

The original Basicode system was also used for a series of experimental broadcasts by *Media Network*, a programme produced for Radio Netherlands, the external services arm of NOS. These short-wave transmissions proved highly successful, with signals reaching as far afield as Australia and the east coast of America.

Partly as a result of the interest generated by these broadcasts, *Media Network's* producer, Jonathan Marks, has become increasingly involved in promoting Basicode overseas. Today, a sister programme *Radioactivity* is supplied on a transcription basis to over 155 educational stations in countries such as Sweden, America, Italy and the U.K. *Media Network* is broadcast at 18.40 GMT on Sundays at 747kHz.

The attitude of the Dutch radio service has always been avowedly evangelical. Programs are free, and there is a conscious attempt to use this new means of transmitting software to get people to think about computers as more than just a black box. Programs have been very largely educational, and have been broadcast with the additional hope of encouraging micro owners to talk to each other and not just to their machines.

## Sole rights

To further these aims in this country, NOS has entered into an agreement with the BBC granting it sole rights in the U.K., with the proviso that Basicode must be offered on a cost-only basis.

Although the BBC now broadcasts Basicode as part of its new *Chip Shop* series on Radio 4, the first transmissions in the U.K. using this system were by BBC Radio Wales last autumn. More recently, Radio Wales' programme *The Micro Show* broadcast purely machine-specific pro-

grams catering for the Spectrum and the BBC Micro among others.

The distinction of the first software broadcast in Britain probably belongs to BBC Radio Leeds, whose programme *Abacus* transmitted software on an experimental basis back in October 1982. *Abacus* is still around, and goes out at 6.45pm on Tuesdays. Other local stations that have shows with software include Radio West's *Datarama* at 5pm on Sundays, and LBC's *Young London* programme, also on a Sunday afternoon, which broadcasts software at about 3.30pm.

## Narrow audience

The use of machine-specific Basics — and in the near future machine codes themselves — will enable more complex programs to be broadcast, though necessarily to a narrower audience. This exacerbates the problem that audio tones grate on the ear, and there is clearly a limit to the amount that people who are not interested in recording software will tolerate. *The Chip Shop* has solved this problem by tucking its broadcasts safely away after midnight.

The software that is broadcast — with or without Basicode — is offered free. BBC stations are particularly concerned that there should be no commercial tie-ins, and that programs are only broadcast when relevant, not merely as a gimmick. So it is not surprising that educational uses tend to predominate.

## Programs scarce

Some British producers of programmes on micros broadcasting free software have expressed concern that suitable computer programs might prove scarce, though the Dutch experience suggest that such fears are groundless. A competition launched there recently attracted 1,500 Basicode programs. Now a foundation is being set up in Holland to promote Basicode and publish a quarterly selection from the many programs that are sent into *Hobbyscoop*. All sales will be at cost price.

It is estimated that out of the 100,000 computer hobbyists in the Netherlands, some 60,000 have *Hobbyscoop's* Basicode. It is widely used to communicate between different machine user groups. In the U.K. *The Chip Shop* reported that 12,000 Basicode kits went out even before the first broadcast of the series, and the figure rose to 40,000 after the first broadcast. Basicode may be limited as a language, but it cannot be a bad thing if it helps computer users to communicate with each other. □

# BASICODE:

## Esperanto voor microcomputers

Wie heeft nog nooit van BASICODE gehoord? Niet? Dan is er iets mis met uw radio. Want elke woensdag van 20.03 tot 20.30 u op Hilversum 1 (FM) en elke vrijdag van 20.10 tot 20.15 u op Hilversum 5 (middengolf, 1008 KHz) zendt de NOS op Hilversum 2 in stereo het programma "Hobbyscoop" uit. De staf van dit programma ontwikkelde een soort Esperanto voor computers. Een vertaalprogramma dat door 17 verschillende computers wordt "verstaan". En waarmee dus programma's van verschillende computers kunnen worden ingelezen. Ook op uw eigen ZX 81 of Spectrum.

### Gratis programma's

BASICODE is enorm populair. Begrijpelijk, want door deze unieke vinding wordt er meteen een groot aantal mogelijkheden toegevoegd aan wat uw computer nu al kan. Programma's van andere computers kunnen vertaald in uw eigen apparaat worden ingelezen. Bovendien zendt de NOS elke zondagavond ook nog 'ns gratis programma's in BASICODE uit. En daarmee heeft u de kosten van BASICODE-boek en vertaalprogramma op cassette (f 27,50, over te maken op gironummer 1419 t.n.v. Alg. Secretariaat NOS, Hilversum, onder vermelding van BASICODE-2) er snel uit.

### Historie

BASICODE is een recente ontwikkeling. In 1978 werden voor het eerst computerprogramma's via de radio uitgezonden. Hobbyscoop deed dit bij wijze van experiment. Toen in 1979 bij Teleac de behoefte ontstond hun cursisten regelmatig van voorbeeldprogramma's e.d. te voorzien, had Hobbyscoop hier een passende oplossing voor. Vanaf dat moment werden er iedere week computerprogramma's uitgezonden. Dat waren overigens slechts programma's voor of de Apple, of de Exidy Sorcerer, of de PET/CBM, of de TRS 80. Net zoals dat bij de

### Een Commodoreprogramma in uw ZX 81? Een Spectrumprogramma in een BBC model B? Het kan. Met een BASICODE-vertaalprogramma.

ZX 81 en de Spectrum bijvoorbeeld het geval is, kan het programma van de ene computer niet in de computer van een ander type geladen worden. Iedere computer gebruikt namelijk bij het beschrijven van cassettes (het SAVE) andere tonen. Het gevolg was dat een luisteraar maar eens per maand een programma voor zijn computer kon ontvangen. En andere typen kwamen helemaal niet aan bod.

### Robers

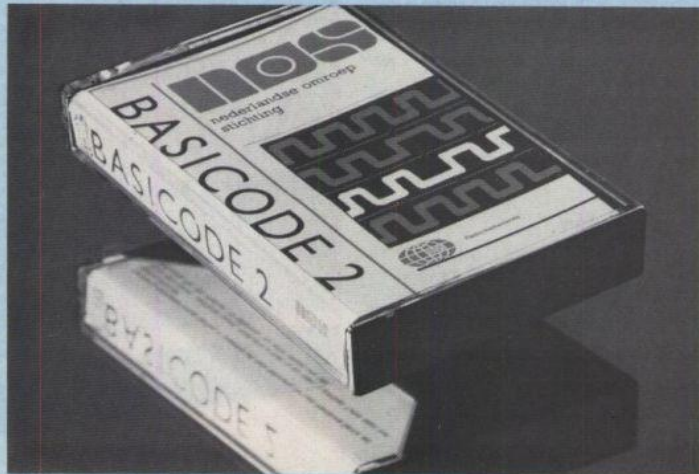
Totdat Klaas Robers voor Hobbyscoop een systeem bedacht waarbij het verschil in die cassettegeluiden werd op-

gelost. In het vervolg zou ieder merk dezelfde tonen moeten maken bij dezelfde letters. Tonen waarvan bovendien bewezen was dat ze het meest geschikt waren om per radio te worden uitgezonden. Klaas Robers noemde het BASICODE (de latere BASICODE-1).

Dat betekende overigens niet dat de programma's na het inlezen gelijk konden worden gebruikt. Vaak moesten met de hand nog tal van wijzigingen worden aangebracht om het voor de eigen computer geschikt te maken. En dat was dan weer nodig omdat de BASIC van de verschillende computers onderling zoveel verschillen vertoonden. Om een voorbeeld te noemen: om het scherm schoon te maken, moet in de ene computer "HOME" worden gebruikt, terwijl de andere weer "PRINT CHR\$18" of "CLS" verwacht.

### Subroutines

Van dit probleem was Robers zich terdege bewust. En samen met Jochem Herrmann uit Eindhoven werkte hij aan een oplossing. Deze werd gevonden in het gebruik van subroutines. De verbe-



terde BASICODE-versie met als doel programma's direct na het inlezen te laten lopen, werd BASICODE-2 genoemd. Een BASICODE-2-programma begint met regel 1000. Onder die regel 1000 staan allerlei subroutines die nooit uitgezonden worden. Zij worden met het vertaalprogramma geladen. Maar op het moment dat men in zo'n BASICODE-2-programma het scherm schoon wil maken, kan men geen "HOME", "PRINT CHR\$18" of "CLS" gebruiken, maar "GOSUB 100". Op regel 100 staat dan de sub-routine die als functie heeft het scherm schoon te maken. Voor Sinclair dus: 100 CLS

105 RETURN

Zo zijn er nog allerlei andere functies door middel van subroutines opgelost. In een van de volgende nummers van Sinclair Gebruiker kom ik daar op terug.

## Merkgericht

Het idee was er, maar de merkgerichte subroutines moesten nog worden gemaakt. Hiervoor werden mensen uitgenodigd die aan eerdere experimenten hadden meegewerkt. Via de Hobby Computer Club Nederland werd aan de gebruikersgroepen van de diverse merken gevraagd of er inlees/vertaalen schrijfroutines konden worden gemaakt.

De Sinclairs hebben er echter altijd een beetje achteraan gehobbeld. De eerste aanzet werd gegeven in 1981 door de toenmalige voorzitter van de Sinclair Gebruikers Groep (Erik Visser). Tijdens een bijeenkomst meldten zich drie kandidaten voor de uitdaging een BASICODE-programma te schrijven. Uiteindelijk kwam de taak op mijn schouders te liggen. En einde '82 kwam ik met de inlees(LOAD)-routine voor de ZX 81. Dat gaf in ieder geval de mogelijkheid om BASICODE-programma's in te lezen en te listen. Om ze te laten draaien, moesten ze zo goed als worden overgetypt, om van de verschillen in BASIC nog maar niet te spreken.

In januari '83 werd dit programma met subroutines via de radio uitgezonden. Juist op het moment dat BASICODE-2 werd geïntroduceerd.

## Spectrum

Toen in 1983 de Spectrum uitkwam was de massale verkoop er de reden van dat Hobbyscoop mij verzocht om ook voor deze nieuwe machine een BASICODE-programma te schrijven, nu gericht op de BASICODE-2-versie. Dit was klaar in februari '84 (versie 5.1). 90 procent van de op dat tijdstip uitgezonden programma's konden worden geladen, vertaald en GERUND. Ik had er bovendien een schrijfroutine bij

ontwikkeld (versie 6.0) die je in staat stelde zelf geschreven Spectrumprogramma's uit te schrijven (te SAVEN) als BASICODE-programma's. Inmiddels hadden naast de NOS de Engelse BBC, de Duitse t.v. en Radio Nederland Wereldomroep interesse voor BASICODE getoond. Via deze omroepen worden de kijkers/luisteraars regelmatig op computerprogramma's getraceerd, bijvoorbeeld via het ook hier veelal te ontvangen programma "Computerclub" dat iedere eerste zondagmiddag van de maand bij de WDR (Duitsland 3) te zien is.

## Het Hobbyscoop/ BASICODE-boek

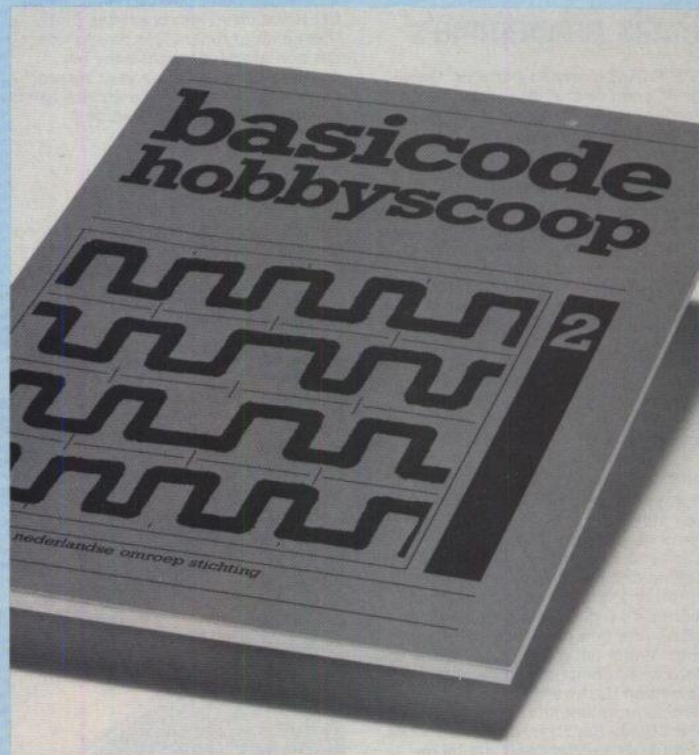
Voor de inlees/vertaal- en schrijfroutines van februari 1984 werd een nieuwe uitgave gemaakt van het Hobbyscoop/BASICODE-boek. Daarin de Sinclair-versie 5.1. Dit boek kwam helaas pas in juni/juli op de markt en toen bleek dat niet meer 90 %, maar slechts 50 % van de uitgezonden programma's met succes ingelezen, vertaald en gedraaid kon worden. Inmiddels waren de uitgezonden programma's namelijk groter, interessanter en moeilijker geworden.

Bovendien gaf de bij het boek geleverde cassette problemen bij het inlezen van de "voorbeeldprogramma's" (kant 2). Deze staan er nogal hard op, en als u voor uw Sinclair-programma's normaal volumestand 10 gebruikt (1-10), dan moet u voor de voorbeeldprogramma's stand 2-4 gebruiken.

Maar dat was niet het enige probleem. Een groot aantal van de voorbeeldprogramma's gebruikte nu ook subroutines boven regel 9999. En de oplossing die de versie 5.1 hiervoor bood, was niet meer toereikend. Verder zat er nog een foutje in de — RUN 400 — routine waarmee variabelen gezocht en vervangen konden worden, en werden de statements "LEFTS", "MIDS", "RIGHTS", "ON-GOTO" en "ON-GOSUB" nog niet vertaald.

## Nieuwste versie

Opnieuw aan de slag dus. De versie 5.1 werd uitgebreid met een hernummeringroutine, de vertaalroutine werd verbeterd op het punt van snelheid, maar ook werden de voor de Sinclair BASIC niet toegestane statements vertaald naar statements die de Sinclair wel kent. De zoek- en vervangroutine RUN 400 werd versneld en de gevoeligheid van de inleesroutine vertienvoudigd.





Op de trappen van de studio: het BASICODE-team van het eerste uur. Onderaan de trap, tweede van links, BASICODE-uitvinder Klaas Roberts. Derde van rechts, met baard, Hobbyscoop-programmamaker Hans G. Janssen.

Ook had ik me altijd vergist in het belang van de "laadstrepen" waar u als Sinclairgebruiker natuurlijk aan gewend bent. Uit vragen en reacties bleek overduidelijk dat er behoefte aan was om ook bij het inlezen van BASICODE-programma's laadstrepen te zien. Die kwamen er. Kortom versie 5.2 was daar.

## Bestellen

Voor de BASICODE-gebruikers houdt dat alles echter in dat BASICODE pas echt bruikbaar is als zij naast het BASICODE-boek ook de laatste versie 5.2 gebruiken. In het Hobbyscoop-boek staat nog de 5.1-versie. Vraag eens bij vrienden of kennissen als u deze versie nog niet bezit. Het is in augustus jl. uitgezonden. Er moeten dus kopieën van zijn. Mocht het niet lukken eraan te komen, dan kunt u de cassette met de laatste BASICODE-versie tegen kostprijs bij mij bestellen. Maak dan f 15,— over op giro 3 74 99 35 t.n.v. Rik Koevoets onder vermelding van SPECTRUM BASICODE 5.2. U ontvangt dan een nieuwe (gewijzigde) handleiding en een cassette met de laatste versie. Ook vragen en opmerkingen blijven bij mij van harte welkom.

Rik Koevoets

# Het ZX Spectrum handboek

## Nu ook in het Nederlands!

Het officiële Sinclair ZX Spectrum handboek van Steven Vickers en Robin Bradbeer is vanaf nu ook in het Nederlands verkrijgbaar. "Introduction" en "BASIC programming" zijn samengevoegd tot één geheel en vertaald door Wichert van Engelen. Het boek telt 240 pagina's en is compleet met karakterset, foutmeldingenlijst, een overzicht van de gebruikte BASIC-statements en een uitgebreide index.

Bestel meteen. Maak f 29,75 over op girorekening 4 75 39 t.n.v. Sinclair Gebruiker, Lezersservice, Leiderdorp. En vermeld het bestelnummer: 710-30. *Verzendkosten en BTW nemen wij voor onze rekening.*

**Vanaf 15 november ook in de boekhandel verkrijgbaar.**



(Advertentie)

# BASICODE

## Lees- & schrijfprogramma

Het uitgezonden BASICODE-vertaalprogramma bestaat uit vijf delen, t.w.:

1. BASICODE-2 standaardsubroutines + menuprogramma
2. BASICODE-2 inlees- en vertaalroutine
3. BASICODE-2 schrijfroutine
4. 42-koloms printroutine
5. BASIC-uitbreiding die de Spectrum-BASIC aanpast aan de BASICODE-eisen.

De BASIC-uitbreiding is in het vertaalprogramma opgenomen om u de moeite te besparen alle BASICODE-programma's te moeten aanpassen aan de Spectrum-BASIC. Deze uitbreiding verandert de werking van de Spectrum-BASIC zodanig, dat deze voldoet aan de BASICODE-norm.

### Handleiding

Laad het programma op de gebruikelijke wijze met "LOAD". U hebt nu alle delen van het programma in één keer in het geheugen geladen en nadat u eventueel deze handleiding hebt bekeken, ziet u een menu met zeven mogelijkheden:

J: Inlezen BASICODE-programma;  
T: Vertalen BASICODE-programma;  
K: List programma;  
R: Run programma;  
C: Omzetten Spectrum programma naar BASICODE;  
P: Print het omgezet programma;  
S: Save programma in BASICODE;  
<DELETE>: Verwijder menu-programma en Save BASICODE-programma.

U kunt u eventueel een BASICODE-programma inladen. Start daartoe de recorder en druk op "J" (de LOAD-toets). Op de schermrand moeten nu strepen zichtbaar zijn, zoals bij het laden van gewone Spectrumprogramma's, alleen iets dunner.

Gedurende de leader-toon zijn de strepen rood/cyaan, daarna worden ze blauw/geel. De eerste regel van het BASICODE-programma, waarin de naam staat, verschijnt tijdens het la-

Jan Bredenbeek uit Hilversum schreef voor de Spectrum een BASICODE-2 lees- en schrijfprogramma (versie 8.01) dat op 26 juni en 3 juli jl. door de NOS is uitgezonden. Voor de BASICODE-liefhebbers en met dank aan de redactie van "Hobbyscoop" drukken we hierbij de op dit programma betrekking hebbende handleiding integraal af.

den op het scherm. De overige regels worden niet afgebeeld omdat er te weinig tijd is om hiervoor te scrollen.

Het inlezen stopt overigens als men

- op de spatietoets drukt
- het geheugen vol is
- of het einde van het BASICODE-programma bereikt is.

Er verschijnt dan resp. de melding:

- "BREAK tijdens inlezen"
- "Geheugen vol"
- "Inlezen OK" of "Checksum ERROR" op het scherm. Dit laatste betekent dat er door een drop out o.i.d. één of meer karakters fout zijn ingelezen, het programma hoeft hierdoor echter nog niet onbruikbaar te zijn geworden!

Het programma staat nu in het geheugen in de vorm zoals het werd uitgezonden, en met een druk op de "K" (LIST) toets kunt u het desgewenst bekijken. Eventuele fouten zijn er nu al uit te halen, want de BASIC-uitbreiding test niet de juiste "syntax" van de regels. Wilt u nu verbeteringen aanbrengen toets dan BREAK in om uit het menu te komen. Nadat u klaar bent met het aanpassen van programmaregels is d.m.v. RUN het vertaalprogramma opnieuw te starten.

### Spectrum-vertaling

Om het programma nu in een voor de Spectrum bruikbare vorm te krijgen moet het nog "vertaald" worden. Druk daartoe op "T". Het vertaalprogramma zal nu de BASIC-woorden door tokens vervangen, LET- en GOTO-statements toevoegen waar dat nodig is en zo meer. Verder worden LEFT\$, MID\$, RIGHT\$ en VAL vervangen door FN-functies (de VAL-functie van de Spectrum blijkt in BASICODE soms problemen te geven). Ook worden ON-GOTO en ON-GOSUB vervangen door GOTO/GOSUB (r1 AND v=1) + (r2 AND v=2) + . . . waarin r1, r2, . . . de regelnummers zijn en v de uitdrukking tussen ON en GOTO of GOSUB. Dit alles kunt u op het scherm zien omdat de vertaalroutine elke vertaalde regel op het scherm uitprint. Indien u tijdens het vertalen op de spatietoets drukt, zal het vertalen stoppen, als u daarna weer op "T" drukt in het BASICODE-menu gaat de vertaalroutine weer verder waar hij gebleven was.

Als het programma vertaald is, kunt u het gaan runnen met "R". De meeste programma's zullen nu meteen lopen, maar soms kunt u wat problemen krijgen, wat blijkt uit een foutmelding of iets dergelijks. Na een foutmelding wordt de "programmacursor" op de foute regel gezet, met "ENTER" krijgt u hem dus direct in beeld. De melding "Nonsense in BASIC" wordt gegeven als er in het programma een syntax-fout zit. Meestal betreft het dan vergeten puntkomma's in PRINT-statements of ontbrekende aanhalingstekens-sluiten. Of er ontbreekt een variabele achter NEXT. Deze moet u dan zelf even verbeteren (na NEXT komt altijd de variabele van de laatste FOR). Moeilijker wordt het bij READ-statements die deze melding geven. Er zit dan namelijk een fout in de DATA-statements (vanaf regel 25000). Kijk hier dan ook naar vergeten aanhalingstekens-sluiten. Mocht u niets kunnen vinden dan werd de melding waarschijnlijk veroorzaakt door het lezen van een numerieke waarde ▶

in een string of omgekeerd. Probeer dan de plaats in de DATA te achterhalen en zet dan indien mogelijk een RESTORE-statement voor het desbetreffende READ-statement.

Verder komt de melding "Variable not found" nog al eens voor. De programmeur heeft dan vergeten de desbetreffende variabele op nul te zetten (of de lege string bij een stringvariabele). Soms krijgt u deze melding ook bij arrays. Maak voor in het programma (b.v. op regel 1005) een regel waarin u de desbetreffende variabele op 0 of "" zet. In het geval van een array moet u een DIM-statement zetten met een dimensiegrootte van 10. Bij programma's waarin een wachtluist zit in de vorm van b.v. FOR I=1 TO 1000:NEXT I zult u meestal merken dat deze te lang duurt. Dit komt door de nogal trage Spectrum-BASIC. Druk in dat geval tijdens het wachten op BREAK, tik ENTER en maak de eindwaarde van de FOR—NEXT-loop een factor 5 tot 10 kleiner. Opmerking: bij sommige programma's loopt deze vertraging via een variabele, pas dan alleen deze aan!

## Schrijven in BASICODE

Met dit vertaalprogramma kunt u ook BASIC-programma's saven in BASICODE-formaat. U kunt hiertoe zelf een programma schrijven of een bestaand programma inladen of mergen, als het maar op regel 1000 begint en voldoet aan het BASICODE-2 protocol (zie hoofdstuk 3 van het BASICODE-boek). Om een geschikt BASIC-programma om te zetten naar BASICODE toetst u de "C" van het menu in. Het omzetprogramma zal nu eerst gaan kijken of u geen verboden statements hebt gebruikt in uw programma. Indien dit wel zo is, wordt de melding "•ERROR•" gegeven, gevolgd door de regel waarin de fout zat met het bekende vraagteken direct na het foute woord. Dit zal ook gebeuren als u vergeten bent haakjes te gebruiken bij functies, b.v. SIN X i.p.v. SIN(X) of TAB 10 i.p.v. TAB(10). In de meeste BASIC's en dus ook in BASICODE moeten altijd haakjes gebruikt worden bij functies, ook als dat wiskundig gezien niet nodig is. OPMERKING: tussen de functienaam en het haakje mogen geen spaties of control-codes staan, anders krijgt u toch nog een foutmelding!

Nadat u de fout verbeterd hebt, moet u het vertaalprogramma opnieuw starten d.m.v. RUN en vervolgens weer "C" kiezen. Indien het omzetprogramma geen fouten vond krijgt u nu de melding "Syntax O.K. Omzetten?"

(j/n)". Wanneer u met "N" antwoordt, wordt het omzetten afgebroken en springt de computer terug naar het menu. Als u echter "J" kiest, zal het BASICODE-programma vervolgens omgezet worden in een z.g. ASCII-file in een string P\$. Daarbij wordt het oorspronkelijke BASIC-programma verwijderd.

Alle tokens van de Spectrum worden nu geëxpandeerd naar afzonderlijke letters. Verder zullen LET-statements onderdrukt worden, evenals GOTO's direct achter een THEN, wat een beetje ruimte bespaart en beter overeenkomt met het BASICODE-protocol. Ook worden spaties toegevoegd indien dat nodig blijkt, b.v. voor TO en THEN. Sommige computers hebben dit nodig.

## Extra's

Er zijn nog een paar extra's m.b.t. de niet-SPECTRUM-statements zoals: LEFT\$, MID\$, RIGHT\$, ON-GOTO en ON-GOSUB. Deze statements kunt u in uw programma als volgt nabootsen:

```
LEFT$ door FN I$ of FN L$:
MID$ door FN m$ of FN M$:
RIGHT$ door FN r$ of FN R$:
ON v GOTO 11,12,13 . . . door GOTO
(11 AND v=1) + (12 AND v=2) + . . .
(v = variabele, 11,12 enz. zijn regelnummers).
```

Op dezelfde manier vervangt u ON . . . GOSUB door GOSUB (11 AND v=1) + . . . OPMERKING: Tussen de GOTO/ GOSUB en het haakje mogen geen spaties of control-codes zitten!

Het omzetprogramma zorgt ervoor dat deze statements automatisch vervangen worden door BASICODE-statements. Indien er bij het omzetten een regel langer dan 60 karakters wordt gegenereerd, ontstaat de melding "Regel NNNN te lang" waarin NNNN het regelnummer is. Het omzetten zal hierdoor overigens niet stoppen, wel zult u om dit te verbeteren het originele programma opnieuw in moeten laden (ev. via BASICODE). Overigens zal een regel die in de oorspronkelijke listing op het scherm 60 karakters lang is meestal resulteren in een iets kortere BASICODE-regel vanwege het onderdrukken van spaties en LET-statements e.d.

Als het programma omgezet is, kan het in BASICODE-formaat worden gesaved met "S" uit het menu. Let op: het saven begint DIRECT na het indrukken van de S-toets: zet de recorder dus aan voor u op "S" drukt! Met de "P"-optie is het mogelijk om het omgezette programma in de string P\$ zien. Als u om een of andere reden uit het menu bent geraakt, moet u dit starten met GOTO 400. Doe het niet met RUN, aangezien deze handeling de laatste de string uitwist!

## Fouten

Ondanks het feit dat het omzetprogramma bepaalde fouten opspoor, is nog wel een kwestie van goed oppassen of een programma wel aan de BASICODE-norm voldoet. Let met name op de volgende punten:

- Variabelen mogen uit hoogstens 2 tekens bestaan en mogen geen kleine letters bevatten.
- Strings mogen hoogstens 255 karakters lang zijn.
- String slicing, b.v. A\$(6 TO 8) is niet toegestaan; gebruik hiervoor LEFT\$, MID\$ en RIGHT\$. Zie boven.
- Achter GOTO en GOSUB moet een regelnummer staan, dus niet b.v. GOTO 1000+100\*A. Gebruik hiervoor ON-GOTO (zie boven). Bovendien de expressie naar bestaande regels verwijzen.
- RESTORE en RUN mogen geen regelnummers hebben.
- Niet IF A>B and C>D THEN . . . maar IF (A>B) AND (C>D) THEN . . .
- U mag alleen strings met een zuiver numerieke inhoud loslaten op de VAL-functie, dus geen formules o.i.d. Gebruik desnoods FN V i.p.v. VAL in uw programma. CLS is wel bruikbaar in een programma; het omzetprogramma maakt daar namelijk automatisch GOSUB 100 van.

De eerste regel van een BASICODE-programma moet altijd luiden: 1000 LET A=N:GOTO 20:REM "programmanaam".

Hierbij moet voor N een getal ingevuld worden dat gelijk is aan de maximale lengte van alle strings te zamen. Bij sommige computers (niet de Spectrum) moet namelijk eerst ruimte voor de strings worden gereserveerd. Dat gebeurt door de sprong naar regel 20. Meestal kan men voor N een waarde 100 aanhouden, maar bij veel lange strings moet u een hogere waarde kiezen. Als regel 1000 niet aanwezig is, doet het omzetprogramma niets! Indien u het schrijfprogramma niet wenst te gebruiken, kunt u intikken CLEAR 56755; hierdoor wordt RAM-TOP verhoogd zodat er ruim 1 K extra geheugen vrijkomt, maar het schrijfprogramma vernietigd wordt zodat u het niet meer kunt gebruiken.

## De 42-koloms printroutine

Dit nieuwe Spectrum-vertaalprogramma bevat een 42-koloms printroutine die het scherm van de Spectrum aanpast aan de BASICODE-norm. Dit kunt u zien tijdens het draaien van het programma of het listen. Alleen bij het editen van BASIC-regels is het



scherm nog 32-koloms. Indien u de 42-koloms routine in uw eigen programma's wilt gebruiken, kan dat door hem te saven met:  
Save "42-koloms" CODE 57989,1715. In uw programma dient u dan na het inladen van de routine te zorgen voor de initialisatie. Dit geschiedt als volgt:

RANDOMIZE USR 57989: Print 42 kolommen op het scherm;  
RANDOMIZE USR 57993: Print 42 kolommen op de printer;  
RANDOMIZE USR 58048: Terug naar 32 kolommen op het scherm;  
RANDOMIZE USR 58058: Terug naar 32 kolommen op de printer.

Voorbeeld: indien u 42 kolommen op scherm en printer wilt (van de ZX printer of andere printer die via de normale Spectrum printerbuffer werkt) moet u na het laden zowel RANDOMIZE USR 57989 als RANDOMIZE USR 57993 intikken. Deze printroutine is zonder problemen te gebruiken met de Microdrive/Interface 1.

De routine kent afgezien van de normale control-codes nog drie extra codes:

- \* Print CHR\$ 24 + CHR\$ N: Hiermee stelt u het aantal schermregels in met N (1 t/m 24).
- \* Print CHR\$ 4: Zet scroll-prompt uit. Nu zal het scherm continu doorrollen. Overigens kan het printen te allen tijde worden afgebroken met BREAK.
- \* Print CHR\$ 5: Zet scroll-prompt aan. Er wordt nu weer met scrollen gewacht, maar de scroll-prompt verschijnt alleen in beeld als er minder dan 23 regels zijn ingesteld.

## De BASIC-uitbreiding

Om de vele problemen die bij het laten draaien van BASICODE-programma's op de Spectrum ontstaan het hoofd te bieden bevat dit nieuwe vertaalprogramma een BASIC-uitbreiding die de Spectrum-BASIC aanpast aan BASICODE, als volgt:

1. Regelnummers tot van 10000 tot 32767 zijn nu toegestaan.
2. String-, array- en FOR-NEXT-variabelen mogen een naam hebben die 255 karakters lang is i.p.v. 1 karakter, dus b.v. IN\$ en FOR HO = 1 TO 10 zijn nu toegestaan.
3. String-arrays en gewone strings met dezelfde naam worden onderling onderscheiden (b.v. A\$ en A\$(1)).
4. Arrays (numeriek of string) mogen een index nul hebben, dus A(0) is toegestaan. Met DIM A(10) declareert men nu een array met index 0 t/m 10, dus met 11 elementen.

5. Met een DIM-opdracht kan men nu meer arrays declareren, b.v. DIM A(10),B(12),C\$(5).
6. De elementen van een string-array hebben nu een variabele lengte i.p.v. een vaste lengte. Met DIM A\$(12,10) ontstaat een tweedimensionale string-array met 13 x 11 strings die ieder zich gedragen als normale strings. In normaal Sinclair-BASIC zou dan een array van 12 strings met een vaste lengte van 10 karakters ontstaan, maar dit strookt niet met de BASICODE-norm zodat hier mogelijkheden zouden ontstaan.

String-slicing zoals dat in Sinclair-BASIC gebruikelijk is (b.v. A\$(6 TO 8)) is nu echter niet meer mogelijk omdat er dan moeilijkheden zouden ontstaan bij string-arrays.

Verder werkt het INPUT-statement nu anders. De cursor verschijnt nu niet op de onderste regel van het scherm, maar direct achter het laatst geprinte karakter. Dan is de cursor bij INPUT een streepje: " \_ " i.p.v. een knipperende L of C. Ook is een prompt zoals in INPUT "Naam?";A\$ niet toegestaan, evenals meer variabelen achter INPUT. Wel toegestaan zijn file-nummers, zoals in INPUT £4:A\$. Indien u bij een numerieke INPUT een regel intypt waarin een fout zit (b.v. alleen ENTER) wordt de melding

"?REENTER" gegeven en moet u opnieuw een waarde intoetsen.

De BASIC-uitbreiding is compatible met de Microdrive/Interface 1, alhoewel voor het gebruik van de Interface 1-statements de volgende beperkingen geldt:

De statements SAVE . . DATA, LOAD . . DATA en VERIFY . . DATA zijn niet mogelijk met de microdrive. Wel met de cassetterecorder, maar daarbij geldt dan dat de getal- of string-array achter DATA een naam heeft van 1 letter lang.

U kunt programma's die met dit vertaalprogramma vertaald zijn niet draaien zonder BASIC-uitbreiding. Wel kunt u normale Spectrum-BASIC programma's laden of mergen. In dat geval worden echter alle variabelen van het programma gewist omdat deze niet uitwisselbaar zijn met die van de BASIC-uitbreiding. U kunt echter numerieke (niet string-)arrays inladen met LOAD . . DATA en deze gebruiken, maar u moet er dan wel rekening mee houden dat de index van alle elementen 1 omhoog schuift, dus A(1) wordt A(0), B(2,3) wordt B(1,2) enz.

Tenslotte: dit hele vertaalprogramma laat zich saven met RUN 700 (cassette) of RUN 800 (microdrive). Indien u deze handleiding niet mee wenst te saven, moet u in het menu de "J"-optie kiezen en vervolgens BREAK gebruiken. ■

## BASICODE-versie 8.01 en printerproblemen

In de 42-koloms printroutine is een fout gevonden die het printen op de ZX-printer beïnvloedt. Indien u namelijk een regel langer dan 74 karakters op deze printer uitprint zal de eerst geprinte regel 42 karakters lang zijn, maar alle opeenvolgende geprinte regels zullen slechts 32 karakters lang zijn. Om dit te verhelpen dienen de volgende POKES ingevoerd te worden:

POKE 58925,14:POKE 58926,43:  
POKE 58927,195  
POKE 58928,12:POKE 58929,230

Overigens zult u bij normaal BASICODE-gebruik van deze fout weinig last hebben.

Verder zal de ZX-printer van de eerstvolgende regel slechts 32 karakters uitprinten als u COPY hebt gebruikt of tijdens het printen op BREAK hebt gedrukt. Om dit te verhelpen: POKE 23679,43 voordat u opnieuw gaat printen.

## Gebruik met andere printers dan de ZX-Printer

Over het algemeen zult u de software voor de printerinterface eerst moeten laden alvorens u het BASICODE-vertaalprogramma laadt. Nu zorgt het laatste programma echter voor 42 kolommen op de ZX- of Seikosa-printer waardoor uw "echte" printer waarschijnlijk niet meer functioneert. We moeten dus voorkomen dat de 42-koloms printroutine voor de printer wordt ingeschakeld. Dit doet u door het statement "RANDOMIZE USR 57993" uit de regels 750, 870, 940 en 960 van het vertaalprogramma te verwijderen. Let op: alleen de RANDOMIZE USR 57993, dus niet de hele regel! Nadat u dit hebt gedaan, SAVE het vertaalprogramma met RUN 700 (cassette) of RUN 800 (Microdrive). U kunt ▶

nu normaal uw printer gebruiken terwijl het scherm gewoon 42-koloms blijft.  
Opmerking: de printersoftware mag geen adressen boven 55600 gebruiken!

## Diskinterfaces, Wafadrive etc.

Bij het gebruik van deze apparaten kunt u moeilijkheden verwachten, speciaal bij BASIC-statements die op het gebruik van de interface slaan. Bij de initialisatie van de Extended Basic wordt gekeken of u een Interface 1 van Sinclair aangesloten hebt. Zo ja, dan worden een aantal maatregelen getroffen die de goede werking van de Extended Basic met de Interface 1 garanderen. Het kan nu zijn dat een andere interface op deze acties verkeerd reageert. Dit kunt u uittesten door RANDOMIZE USR 6E4 in te tikken (uiteraard als u het vertaalprogramma geladen hebt!). Als er iets an-

ders gebeurt dan de melding OK gaat er iets mis en dat kunt u verhelpen door de volgende pokes in te voeren:

```
POKE 61100,24:POKE 61101,14:
POKE 61125,24
```

Daarnaast kunt u zoals gezegd problemen krijgen bij de disk-statements. Dit komt voornamelijk omdat de disk-interface voor het interpreteren van de regel routines uit de Spectrum-ROM aanroept. En zoals u weet kunnen deze geen BASICODE aan.

U kunt bij het werken met deze statements over het algemeen de volgende vuistregels hanteren:

a. Geen variabelen gebruiken, dus geen LOAD "\*"m";1,a\$. In plaats van variabelen moet u de waarde direct invullen.

b. Getallen intypen als VAL "getal", dus LOAD "\*"m";VAL "1","naam" bijvoorbeeld. Waarschijnlijk is dit laatste bij de BETA diskinterface niet nodig, vermoedelijk wel bij de Wafadrive en de OPUS Discovery 1.

Dan is er nog een probleem bij het optreden van een foutmelding die op een disk-statement slaat. Waarschijnlijk zal er dan teruggesprongen worden naar de oude BASIC-interpret. Om de Extended Basic weer in te schakelen moet u dan RANDOMIZE USR 6E4 invoeren.

Overigens stel ik het zeer op prijs als bezitters van deze interfaces een oplossing weten voor bovengenoemde problemen, of technische informatie kunnen geven, dit te vernemen via Sinclair Gebruiker.

Ten slotte geef ik nog de startadressen van de diverse onderdelen van het vertaalprogramma:

```
55600 Omzet en SAVE-routine
56756 Inlees- en vertaalroutine
57989 42-koloms printroutine
59711 Machinecode voor FN-functies
LEFT$, MID$, RIGHT$ en VAL
60000—65367 Extended BASIC. ■
```

Jan Bredenbeek.

## NIEUWS

### PCM Show

Op 11 t/m 13 oktober wordt in de Jaarbeurs te Utrecht de Personal Computer Magazine Show georganiseerd. Het initiatief van deze Show is genomen door het microcomputerblad



Personal Computer Magazine. Het wordt een grootscheepse manifestatie met een speciaal karakter. Zakelijk-beroepsmatige gebruikers, maar ook geïnteresseerden in toepassingen van de computer thuis, komen volop aan hun trekken, zo claimt PCM. Voor hen zijn er speciale demonstraties, lezingen, mini-seminars en film- en videopresentaties op touw gezet. Daarbij is er rekening gehouden met verschillende kennisniveaus.

Tijdens de PCM-Show '85 zullen zowel de exposanten als de beursorganisatoren veel aandacht besteden aan het actief betrekken van de bezoeker bij het gebruik van hard- en software. Ook de voorlichting is afgestemd op

verschillende gebruikersdoelgroepen. Doel van dit alles: het verlagen van de angstdrempels, het vergroten van kennis en kunde en het bewust maken van de grote waarde van de micro voor iedereen, stellen de organisatoren.

Er is een uitgebreid evenementenprogramma waaruit de bezoekers een keuze kunnen maken. Met ingang van 10 september a.s. zijn er op het adres:

VNU-Business Publications BV,  
Afd. PCM-Marketing, F. de Haas,  
Rijnsburgerstraat 11,  
1059 AT Amsterdam, (020)-5102911  
lijsten verkrijgbaar waar alle evenementen op vermeld staan.

Alvast een kleine, zeker niet volledige greep: lezingen voor ondernemers uit het Midden- en Kleinbedrijf, lezingen voor personen uit het taalonderwijs, lezingen voor LBO- en HBO-onderwijs, etc. Er is een grote stand waar men zelf achter de personal computer kan plaatsnemen en waar men als bij een soort talenpracticum op de computer wegwijs wordt gemaakt. Er is een film- en videocentrum met korte, interessante audiovisuele, een stand met noviteiten en een demonstratiecentrum voor het tonen van allerlei onderwijsprogramma's. Nieuwe Revue en Televisier sponsoren de beide weekenddagen en komen met speciale acties. PCM zelf organiseert een wedstrijd en Microcomputer Club Nederland (MCN) is met het Nationale Beursspel aanwezig.

Een groots opgezet gebeuren, dat zeker uw aandacht waard is. Noteer de datum dus in uw agenda! ■

## Congressen en tentoonstellingen

Op dinsdag 10 september wordt in het RAI-Congrescentrum in Amsterdam het Nationaal Informatica Congres gehouden. Het congres valt samen met de vakbeurs Kantoorinnovatie '85 die van 10 tot en met 13 september in het Europacomplex van de RAI wordt georganiseerd. Thema van het congres is "Communicatieaspecten van kantoorautomatisering". Vier inleiders zullen vanuit hun eigen discipline hun visie op verschillende facetten van de informatieverwerking geven. Ook tijdens Kantoorinnovatie '85 zullen verschillende lezingen worden gehouden. Beide activiteiten worden georganiseerd door de Vereniging van Importeurs en Fabrikanten van Kantoorapparaten (VIFKA) in samenwerking met de RAI.

Geïnteresseerden in CAD/CAM kunnen hun hart ophalen in de periode van woensdag 27 t/m vrijdag 29 november. Dan vindt in de RAI voor het eerst de manifestatie "CAD/CAM Nederland" plaats. Het initiatief tot de organisatie van deze vakbeurs voor computergestuurd ontwerpen, tekenen en fabriceren werd genomen door de sectie CAD/CAM van de Vifka. De belangrijkste leveranciers van CAD/CAM-systemen zijn bij deze branche-organisatie aangesloten, zodat u mag verwachten dat u tijdens deze manifestatie een zeer representatief overzicht zal kunnen krijgen van de producten en diensten die op dit terrein in ons land voorhanden zijn. ■

# BASICODE

## Het programma van Herman Ament

In het eerste nummer van Sinclair Gebruiker zijn we diep ingegaan op het BASICODE-programma van Rik Koevoets.

Direct nadat dit programma was uitgezonden via NOS-Hobbyscoop, werd nog een tweede vertaalprogramma voor de 48K Spectrum, geschreven door Herman Ament, via de ether bij de hobbyisten gebracht (30 september '84). Omdat velen ook dit programma op de cassette hebben staan, drukken we hierbij de toelichting bij dat programma integraal af.

Om een kopie van het geheel te krijgen, hoeft men alleen maar het programma te stoppen door BREAK in te toetsen nadat alles is geladen en daarna RUN 890 uit te laten voeren. Om in het menu te komen, kunt u elke keer als u het programma heeft onderbroken GOTO 810 intypen. Normaal komt u vanzelf in het menu. Dit menu bestaat uit drie gedeelten:

1. een *inleesroutine*
2. een *listroutine*
3. een *vertaalroutine*.

### De inleesroutine

De inleesroutine verwijdert eerst alle regels van 900 t/m 919 en alle regels boven de 1000. Daarna wordt het BASICODE-programma ingelezen en opgeslagen in een REM-regel en wel regel 900. De begintoon is bij deze inleesroutine verplicht!

Als alles goed is gegaan, verschijnt er op het scherm "O.K. checksum = 0". Anders komt één van de volgende "foutmeldingen":

- break gedurende laden: dat is wel duidelijk
- programma te groot: er is te weinig geheugen in de computer om het hele BASICODE-programma in te lezen, het is slechts gedeeltelijk ingelezen
- Checksum error: dit is het equivalent van een tape loading error. Er is niet foutloos ingelezen. Dit komt omdat óf het volume niet goed staat óf er een storing op de band is. Neem

Als u één van de velen bent die het BASICODE-programma voor de Spectrum via de radio heeft opgenomen, weet u waarschijnlijk hoe u ermee moet werken. U laadt het programma in, en vervolgens worden automatisch de machinecode voor het inlezen, vertalen en de codes om 42/51 kolommen op het scherm te krijgen ingelezen.

altijd op met dezelfde cassetterecorder als die u voor uw computer gebruikt!

### De listroutine

Het BASICODE-programma wordt tijdens het inlezen onveranderd gelaten. Daardoor is het mogelijk om het programma te listen zoals het origineel is ingetypt, met de oude regelnummers, variabele-namen en functies en opdrachten zoals LEFT\$ etc. Deze

listing kan naar de printer of naar het beeldscherm gestuurd worden. Hiermee kan men ook zien wat er allemaal fout is ingeladen bij een checksum error.

### De vertaalroutine

Deze routine bestaat uit drie gedeelten die automatisch vanuit BASIC worden aangeroepen:

- een *vertaalroutine*
- een *renumberroutine*
- een routine om *variabele-namen* aan te passen.

Om te beginnen de vertaalroutine. Deze routine vertaalt de eerste regel tussen 900 en 919. Elke regel wordt in Spectrum BASIC omgezet, op de juiste plaats in het programma gezet, en het stukje wat vertaald werd, wordt verwijderd. Er worden tijdens het vertalen enkele zaken aangepast:

- END wordt STOP:STOP:STOP:
- LEFT\$, RIGHT\$ en MID\$ worden omgezet naar Spectrum BASIC en wel op de volgende manier:  
LEFT\$(A\$,I) wordt (A\$(TO I)  
RIGHT\$(A\$,J) wordt (A\$(LEN  
(A\$) - (J) + 1 TO  
MID\$(A\$,I,J) wordt (A\$(I TO I + J - 1)
- ON A GOTO etc. deze opdracht wordt ook aangepast. In verband met het renumberen moet elke GO-

TO/GOSUB-opdracht een apart statement worden. Aan de hand van een voorbeeld laten we zien hoe het werkt:

BASICODE	Vertaald voor de Spectrum
1030 PRINT:ON A GOSUB 1100,1110,1300:PRINT "KLAAR";	1030 PRINT: 1030 IF INT(A + 5) = 001 THEN GOSUB 1100 1030 IF INT(A + 5) = 002 THEN GOSUB 1110 1030 IF INT(A + 5) = 003 THEN GOSUB 1300 1030 PRINT "KLAAR";

Door het renummeren worden de regels 1030 weer allemaal verschillend. N.B.: Als A gelijk is aan 1 en in sub-routine 1100 wordt A 2 tijdens het runnen van het programma, dan zal daarna subroutine 1110 óók aangeroepen worden, terwijl dat eigenlijk niet moest!

\* DIM achter DIM kan men in BASICODE meer arrays declareren. Bij de Spectrum kan dat niet. De vertaalaroutine voegt DIM toe waar dat nodig is. Omdat bij Basicode de indices van arrays bij nul beginnen, wordt bij elke index " + 1" toegevoegd. Hierdoor gaat FOR I=0 TO 12: READ A(I): NEXT I altijd goed.

### Een speciaal geval zijn de "gedimensioneerde string arrays".

## Speciaal geval

\* Een speciaal geval zijn de *gedimensioneerde string arrays*. Als in BASICODE de volgende uitdrukking voorkomt:  
DIM A\$(10)  
dan declareert men 11 strings waarbij elke string een variabele lengte van 0 tot 255 kan hebben. Bij de Spectrum declareert men dan één string van lengte 10.  
Een illustratie aan de hand van een voorbeeld:

BASICODE	Onaangepast	Verbeterd
DIM A\$(2); A\$(2) = "PIET"	DIM A\$(2 + 1); LET A\$(2 + 1) = "PIET"	DIM A\$(2 + 1,5); LET A\$(2 + 1,TO) = "PIET"
De lengte van de desbetreffende array-elementen:		
LEN A\$(0) = 0 LEN A\$(1) = 0 LEN A\$(2) = 4	LEN A\$(0 + 1) = 1 LEN A\$(1 + 1) = 1 LEN A\$(2 + 1) = 1	LEN A\$(0 + 1) = 5 LEN A\$(1 + 1) = 5 LEN A\$(2 + 1) = 5

Bestudeer het hiervoor staande goed, want dit is een essentieel verschil! Zo levert RIGHTS(A\$(2),2) in het hiervoor staande voorbeeld "ET" op, terwijl de verbeterde Spectrum-versie "T" oplevert, omdat men als extra dimensie 5 heeft ingevuld.  
Er moet dus nog een extra dimensie toegevoegd worden, waarvan men de grootte moet gokken. De vertaalaroutine voegt als extra grootte TO toe. Dit geeft een foutmelding bij het runnen van het programma en men zal

dan zelf een geschikte keuze moeten maken.  
(Achter de foutmelding staat het regel-

nummer. B.v. "Nonsense in BASIC, 1030:4" betekent een fout in regel 1030.) Bij normale statements wordt ook TO toegevoegd (b.v. LET A\$(I) = "PIET"). Als men echter bij elke statement waar een string array wordt gebruikt een foutmelding wil hebben, kan men daar voor zorgen door op adres 64622 een code te poken die daarvoor zorgt, bijvoorbeeld een sterretje (Zie schema links onder op pagina 58).

### De karakterroutine kan ook zonder BASICODE worden gebruikt.

Als er commando's met kleine letters voorkomen (b.v. if), worden die ook herkend. Door alle hiervoor genoemde aanpassingen wordt het programma over het algemeen langer dan het originele programma. Om het korter te maken, kan men met de hand alle regels aflopen en eventuele berekeningen zoals 1 + 3 samenvoegen. Maar meestal past een BASICODE-programma gemakkelijk in een Spectrum.

## Het renummeren

De *renumberroutine* renumbert alle regels boven de 1000 opnieuw met een stapgrootte van 10. Alle GOTO's, GOSUB's, SAVE . . . LINE's, RESTORE's, LIST, en LLIST worden indien nodig

aangepast. Als de foutmelding "Out of memory" verschijnt, was er niet genoeg ruimte om alles te renummeren. Er is dan nog niets gebeurd. Het beste kan men dan het vertaalprogramma verwijderen (optie DELETE) en daarna RANDOMIZEUSR 62459 intypen. Waarschijnlijk is er dan wel genoeg ruimte. Als de foutmelding "Subscript Wrong" verschijnt, is de stapgrootte te groot. Men kan op adres 62453 een kleinere stapgrootte poken.

Op regel 790 t/m 794 staat een aantal GOTO statements. Deze wijzen naar speciale regels. Na het renummeren staat achter GOTO het nieuwe regelnnummer. In het commentaar van een BASICODE-programma staat vaak wat bepaalde regels doen. Om ook die nieuwe regelnnummers te weten te komen, moet voor het renummeren op een bepaalde regel (b.v. 999 of 795, daar komt het programma toch nooit) een GOTO statement gemaakt worden met een verwijzing naar de gewenste regel. Na het renummeren staat achter GOTO het nieuwe nummer!

## Het aanpassen van de variable-namen

Bij BASICODE is het volgende toegestaan wat niet bij de Spectrum mag:  
\* Namen van FOR..NEXT-loops en namen van arrays en strings mogen 2 karakters lang zijn.  
\* De naam van een string en een string array mogen dezelfde zijn. Voorbeeld: A\$(1) en A\$ refereren naar verschillende variabelen! Met deze routine worden automatisch alle namen veranderd waarvoor dat moet.

### Als er commando's met kleine letters komen, b.v. "if", worden die ook herkend.

Men kan een lijst van de oude en nieuwe namen naar printer of beeldscherm sturen. Als er als nieuwe naam een sterretje verschijnt, betekent dat dat die naam niet aangepast kan worden omdat de namen op zijn. Men zal dan zelf iets moeten verzinnen om dat op te lossen. Overigens wordt het hele programma gecontroleerd, dus ook regel 1 t/m 999. Bij een aanpassing kunnen de volgende vijf veranderingen verschijnen:

- \* A\$ = n betekent: er komen in het programma zowel een string A als een string array A voor. Het string array behoudt zijn naam en alle strings A krijgen de naam n. Strings met slicers staan nl. tussen haakjes.
- \* F1 = x betekent de FOR..NEXT-variabele F1 krijgt de naam x.
- \* A2 (= y) betekent het numerieke array A2 krijgt de naam y.
- \* S1\$ = z betekent de string S1 krijgt de naam z.
- \* SAS(= t) betekent het string array SAS krijgt de naam t.

N.B.: De routine past de variabelen met kleine letters ook aan. De nieuwe namen zijn in hoofdletters.

## De DELETE-optie

Als men het programma heeft vertaald en gere-numbered en alle variabelen heeft aangepast, heeft men het menu-programma niet meer nodig. Door DELETE in te drukken worden alle regels tussen 800 en 919 verwijderd. Op regel 893 staat hoe men de codes van de 42-kolomsroutine kan saven en op regel 896 staat hoe men de routine moet opstarten. Kopieer deze regels eventueel naar een regel die niet verwijderd wordt. Op regel 861 staat nog wat extra informatie over die 42-kolomsroutine.

## De 42/51-kolomsroutine

Speciaal voor de Spectrum wordt een routine meegeleverd die 42 of 51 karakters op het beeldscherm afdrukt. Om deze routine te initialiseren, moet men het volgende doen:

### RANDOMIZE getal

Dit getal geeft het kanaal aan waarlangs men wil printen. Normaal is dat kanaal 2. Kanaal 3 gaat normaal naar de ZX-printer (LPRINT). RANDOMIZE 2 zorgt ervoor dat alles wat met PRINT wordt afgedrukt naar de 42-kolomsroutine gaat.

### RANDOMIZE USR 6E4

Nu wordt kanaal 2 naar de 42-kolomsroutine geopend. Normaal zijn er 3 kanalen, maar door deze routine komen er 5 kanalen en het begin van het BASIC-programma is daardoor niet op 23755, maar op adres 23765. (Zonder INTERFACE 1.) Op

het moment dat deze tekst wordt geschreven, is nog niet bekend of deze 42-kolomsroutine compatibel is met interface 1, al dan niet met een microdrive.

## Speciale karakters

Men kan speciale karakters naar deze routine sturen. Deze zijn behalve de karakters voor de kleuren en printposities op het scherm:

```
CHR(1)      print 42 karakters op 1 regel
CHR(2)      print 51 karakters op 1 regel
CHR(0) + CHR(I) print alleen op de eerste (I + 1) regels
CHR(30)     vraag indien nodig om scroll
CHR(31)     vraag niet om scroll
N.B.: INPUT werkt niet om scroll
counter te resetten.
```

regel 892 verwijderen  
regel 893 laatste VERIFY" "CODE verwijderen  
regel 895 eventueel CLEAR aanpassen (RAMTOP lager dan 61860) en laatste LOAD" "CODE verwijderen  
regel 896 geheel verwijderen  
regel 897 geheel verwijderen.

Als men deze routine gewoon wil gebruiken zonder BASICODE, moet men het volgende doen:

\* Zorg ervoor dat de RAMTOP lager is dan 60000 en dat er geen machine-

code wordt gebruikt op adres 60000 t/m 61610.

\* LOAD" "CODE de machinecode voor 42 karakters per regel.

\* RANDOMIZE 2:RANDOMIZE USR 6E4 klaar!

\* PRINT CHR\$(1);CHR\$(0);CHR\$(23):  
REM 42 karakters en 24 regels per scherm.

Om het te saven: SAVE naam CODE 6E4,1601.

Nog één opmerking. Op regel 860 t/m 863 staat een deel van de hiervoor vermelde informatie in REM-regels.

Ik hoop dat u hiermee met succes programma's die u via Hobbyscoop of vrienden of vriendinnen krijgt, om kunt zetten voor uw Spectrum.

Om de normale 32 karakters per regel weer terug te krijgen, hoeft men alleen maar CLOSE #2 in te typen.

## De inleesroutine verwijdert eerst alle regels van 900 t/m 919 en alle regels boven de 1000.

Men moet dan van regel 110 en 120 een REM-regel maken en van regel 111 en 121 de eerste REM weghalen. Op die regels staan de speciale BASICODE-subroutines voor de PRINT-posities.

Als men deze routine in het geheel niet wil, moet men ook regel 890 tot 897 aanpassen:

## Let ook op het volgende:

### De inleesroutine:

De BORDER-kleur is paars/groen als de routine zoekt naar de begintoon en wit/zwart als er daadwerkelijk geladen wordt.

### De listroutine:

Als er in de tekst een niet-toegestane karakter staat door een storing op de band of iets dergelijks, wordt een zwart blokje CHR(143) afgedrukt. Aan het eind van de tekst staat een getal. Dat is de checksum van dat programma.

### Het aanpassen van de variabele-namen:

bedenk dat door deze routine de variabelen IN\$, SR\$ in de subroutines 100 tot 400 ook veranderen, en dat men, als men een ander programma wil aanpassen, het BASIC-gedeelte opnieuw geladen moet worden.

### Lege FOR..NEXT-loops:

Dienen vaak als pauzefunctie. Deze ►

**SCHEMA**

Het verschil tussen string arrays in BASICODE en Spectrum BASIC.

<p><b>Programma:</b></p> <pre>10 DIM A\$(4) 20 LET A\$(0) = "PIET" 30 LET A\$(2) = "12345" 40 LET A\$(3) = "OPGESLOTEN"</pre> <p><b>indexvariabele lengte</b></p> <table style="border-collapse: collapse; margin: 5px 0;"> <tr><td style="border: 1px solid black; padding: 2px;">0</td><td style="border: 1px solid black; padding: 2px;">P</td><td style="border: 1px solid black; padding: 2px;">I</td><td style="border: 1px solid black; padding: 2px;">E</td><td style="border: 1px solid black; padding: 2px;">T</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">1</td><td colspan="4" style="border: 1px solid black; padding: 2px;">lege string</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">2</td><td style="border: 1px solid black; padding: 2px;">1</td><td style="border: 1px solid black; padding: 2px;">2</td><td style="border: 1px solid black; padding: 2px;">3</td><td style="border: 1px solid black; padding: 2px;">4</td><td style="border: 1px solid black; padding: 2px;">5</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">3</td><td style="border: 1px solid black; padding: 2px;">O</td><td style="border: 1px solid black; padding: 2px;">P</td><td style="border: 1px solid black; padding: 2px;">G</td><td style="border: 1px solid black; padding: 2px;">E</td><td style="border: 1px solid black; padding: 2px;">S</td><td style="border: 1px solid black; padding: 2px;">L</td><td style="border: 1px solid black; padding: 2px;">O</td><td style="border: 1px solid black; padding: 2px;">T</td><td style="border: 1px solid black; padding: 2px;">E</td><td style="border: 1px solid black; padding: 2px;">N</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">4</td><td colspan="10" style="border: 1px solid black; padding: 2px;">lege string</td></tr> </table> <p><b>Fig. 1. A\$ in BASICODE</b></p>	0	P	I	E	T	1	lege string				2	1	2	3	4	5	3	O	P	G	E	S	L	O	T	E	N	4	lege string										<p><b>Vertaald voor de Spectrum:</b></p> <pre>10 DIM A\$(4+1, getal) 20 LET A\$(0+1, TO ) = "PIET" 30 LET A\$(2+1, TO ) = "12345" 40 LET A\$(3+1, TO ) = "OPGESLOTEN"</pre> <p><b>index vaste lengte per array-element</b></p> <table style="border-collapse: collapse; margin: 5px 0;"> <tr><td style="border: 1px solid black; padding: 2px;">1</td><td style="border: 1px solid black; padding: 2px;">P</td><td style="border: 1px solid black; padding: 2px;">I</td><td style="border: 1px solid black; padding: 2px;">E</td><td style="border: 1px solid black; padding: 2px;">T</td><td style="border: 1px solid black; padding: 2px;"></td></tr> <tr><td style="border: 1px solid black; padding: 2px;">2</td><td colspan="5" style="border: 1px solid black; padding: 2px;"></td></tr> <tr><td style="border: 1px solid black; padding: 2px;">3</td><td style="border: 1px solid black; padding: 2px;">1</td><td style="border: 1px solid black; padding: 2px;">2</td><td style="border: 1px solid black; padding: 2px;">3</td><td style="border: 1px solid black; padding: 2px;">4</td><td style="border: 1px solid black; padding: 2px;">5</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">4</td><td style="border: 1px solid black; padding: 2px;">O</td><td style="border: 1px solid black; padding: 2px;">P</td><td style="border: 1px solid black; padding: 2px;">G</td><td style="border: 1px solid black; padding: 2px;">E</td><td style="border: 1px solid black; padding: 2px;">S</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">5</td><td colspan="5" style="border: 1px solid black; padding: 2px;"></td></tr> </table> <p><b>Fig. 3. A\$ bij de Spectrum na vertaling.</b> Voor getal is hier 5 gekozen. Men zal dat zelf moeten kiezen en invullen!</p>	1	P	I	E	T		2						3	1	2	3	4	5	4	O	P	G	E	S	5					
0	P	I	E	T																																																																	
1	lege string																																																																				
2	1	2	3	4	5																																																																
3	O	P	G	E	S	L	O	T	E	N																																																											
4	lege string																																																																				
1	P	I	E	T																																																																	
2																																																																					
3	1	2	3	4	5																																																																
4	O	P	G	E	S																																																																
5																																																																					

**Fig. 2. A\$ bij de Spectrum zonder vertaling.**

1	0	
---	---	--

vaste lengte van 4

**N.B. Lege hokjes zijn spaties.**

**Attentie:**

De uitzendtijden van het NOS-programma *Hobbyscoop* zijn sinds begin oktober gewijzigd. Het hoofdprogramma wordt uitgezonden in FM stereo op woensdagavond 20.00—20.30 uur op Hilversum 1 (en 5). De computerprogramma's alleen op Hilversum 5 (middengolf 1008 kHz) op vrijdagavond 20.10—20.15 uur.

Speciaal voor de Spectrum bevat het BASICODE-programma een routine die 42 of 51 karakters op het beeldscherm afdruckt.

duren op Spectrum vaak veel langer dan de bedoeling is. Als er in een programma een poosje niets gebeurt, BREAK dan om te zien of het een lege FOR...NEXT-loop was. En pas in dat geval de bovenlimiet aan. B.v. FOR I = 1 TO 5000:NEXT I wordt FOR I = 1 TO 400:NEXT I, de nieuwe bovenlimiet moet men uitproberen. ■



## RUBRIEKSADVERTENTIES

### Sinclairtjes

Als abonnee van Sinclair Gebruiker kunt u profiteren van een uniek aanbod: een gratis "Sinclairtje". U hebt het recht op één gratis *niet-commerciële* advertentie per maand. Dit moet u daarvoor doen: kijk op de wikkel die u — als abonnee — bij dit blad hebt ontvangen. Op de achterzijde daarvan staat een matrix van 7 x 25 afgedrukt. Schrijf daarin de tekst die u als "Sinclairtje" opgenomen wenst te zien (maximaal 175 tekens). Vergeet uiteraard niet uw naam, adres en/of telefoonnummer te vermelden! Stop vervolgens de wikkel compleet (dus inclusief uw adres aan de voorzijde) in een envelop, frankeer deze en stuur 'm naar: *Sinclair Gebruiker, Leidsestraat 2, 2352 BA Leiderdorp*. Als uw Sinclairtje voor de vijftiende van de maand bij ons binnen is, kan hij nog in het eerstvolgende nummer worden opgenomen.

ZX81, 16K machine code prm. 3 stuks op cassette voor f 15,—, o.a. formule 1, Bibods invasie bombardieren. B.W. Jurrin, Postbus 99, 7440 AB Nijverdal. Tel. 05486—1 69 34 na 14.00 uur voor inlicht.

Ruilen Spectrum software. Stuur uw lijst naar A. Leeuwesteijn, Vondellaan 142, 3351 HG Papendrecht. Ik heb o.a. Harrier attack-fred-fighter pilot-beach head e.a.

Gezocht: de handleiding van de 48K uitbreiding van b.v. microsource of Etra. Rob Macaré, Safierstraat 36B, 3051 XW Rotterdam. Tel. (na 18.00 uur): 18 11 71.

Te ruil gevraagd: ZX Spectrum software (16/48K). Alleen mach. taal. Schrijven naar: J.P. Enkelaar, Deurzerdiep 30, 8032 NA Zwolle. Tel. 038—54 45 77.

Te koop: instellison-spelcomputer met 7 spellen — golf — voetbal — baseball — tennis — bowling — roulette — black-jack + kleurentelevisie + zw-witte televisie — 66 cm — 53 cm f 900,— totaal. Tel. 071—76 62 34.

Wie kan mij helpen aan het juni/juli nummer van "ZX-COMPUTING"? (natuurlijk tegen vergoeding) P. Wouda, Bottemastr. 7, 8442 JV HEERENVEEN. Tel. (05130)—2 43 78 (na 6 u.).

Kontakt gezocht met spectrum-gebruikersgroep te Groningen. Tel. 050—71 34 27. Frans.

Wie weet een leverancier in Nederland, waar het Saga-Emperor toetsenbord voor de Spectrum te koop is. Na 20.00 uur. Tel. 01899—1 03 15.

Te koop 1 fruitgokautomaat geheel compleet met reserveonderdelen, traforetailslampjes, goed werkend zoals in amusementshallen e.d. Brons. Rembrandtlaan 96, Enschede. Tel. 053—30 10 49. Prijs: f 625,—.

Sinclair is mijn ene hobby. De andere is het verzamelen van oude schaatsen. Kijk eens op zolder. Kan je afstand doen van die oude schaatsen, bel mij dan even op. Tel. 072—12 88 56.

Gepensionneerd financieel computerenthousiast vecht met de Spectrum LOI-cursus en zoekt medestrijder, buurt Gouda/Reeuwijk. Tel. 01829—39 78.

ZX81-bezitters gezocht in IJmuiden Oost, Velsen-Zuid en Driehuis, om kennis en gegevens e.d. uit te wisselen. L.A. van Leeuwen, Groeneweg 82 (Sporhuis). IJmuiden. Tel. 02550—1 69 74.

Zaanstreek. Wie wil er met ons Spectrum programma's ruilen of uitwisselen? C. de Groen, Se-rooskerkestr. 64, 1561 TT Krommenie. Tel. 075—28 67 33.

Wie kan mij helpen aan een Nederlandse vertaling van Fighter pilot, zie Sinclair Gebruiker nr. 1, pag. 21. De handleiding. A. Schoumans, Schoolstr. 4, 1261 EV Blaricum.

Wie wil software, hardware en andere gegevens van de ZX-Spectrum uitwisselen? Schrijf naar: M. Winkelman, Postbus 13763, 2501 ET Den Haag.

LOI Microcomp. & BASIC index van 300 trefw. X, Y & Z-les. Toont trefw. drukt trefw. af, opzoeken trefw. Dit mag u niet missen! Slechts f 1,50. Spect. 46K. Bestellen via 020—17 15 32.

Te koop in originele verpakking: Alchemist, arrow of death 1, penetrator, pi-ball e.d., pedro, planet of death, invincible Island. f 15,— astron, Hobbit f 20,— ex porto. Bel na 7 uur 030—78 01 63.

Gezocht: mach. code progr. om screendump te maken met (brother HR5) RS 232-printer. Verder een DATABASE geschikt voor RS 232 printout. Tel. 020—62 03 88.

Wie wil ZX Spectrum software ruilen? Heb zelf ± 150 programma's. Bel of schrijf J. v. Santen, Walvisbaai 12, 2904 BM Capelle a/d IJssel. Tel. 010—51 11 47.

Spectrum software ruilen? Stuur mij een bandje en je krijgt een ander bandje terug. Hans Matzer, Asserlaan 4, 6006 NL Weert.

Te koop Tigg/4A ext. BASIC joystick, cassrec. spelmodules, spelcassettes en veel lectuur en lesboeken, prijs f 750,— of ruilen tegen printer voor Spectrum. Tel. 01140—1 05 37.

Wie kan mij aan een tweedehands kleurenmonitor en/of Kempson-joystick voor de ZX Spectrum helpen? B. de Koning. 080—44 17 01.

Te koop DK Tronics toetsenbord. ZX Spectrum met spaceball. Prijs f 125,—. Tel.: 02987—32 01. (Assendelft.)

Te koop: ZX81 + 16K + geluidsgenerator + I/O poort + keyboardje (rubber) + 4 boeken + datarecorder. Prijs f 350,—. Ook bereid ruilen v. Spectrum. Tel.: 05163—3 17.

Te koop aangeboden: een ZX-printer voor ZX 80-81 en Spectrum + 4 rollen à f 100,—. Nw.pr.: f 275,—. T.B. J.A.P. Stoop, Wilhelm-nastr. 35-C, 4921 KC Made. Tel. 01626—39 62.

Te koop voor ZX 80/81 memotech centronics interface incl. verb.kabel. Voor ieder redelijk bot. Tel. 030—73 24 00 na 17 uur.

Te koop gevraagd. ZX-Spectrum 48K-reacties via Vidibus 400014418 of Postbus 600—7300 AP te Apeldoorn (met prijsopgave) J.H. van Ee.

Te koop wegens overcompleet DK Tronics toetsenbord microdrive compatibel interface 1 kan worden ingebouwd. Tel. na 6 uur 055—55 96 67 (Apeldoorn). Prijs: f 100,—.

Te koop: 16K Rampack voor ZX81. Vraagprijs f 40,—. M. Beckmann, Noordzeestr. 21, 3522 PH Utrecht. Tel.: 030—89 39 33.

Te koop microline 80 printer 80 koloms Matrix met par. centronics aansluiting, vraagprijs ± f 600,—. Tel. 070—25 05 90 ('s avonds na 8 u.).

Te koop gevraagd kleurenmonitor voor Spectrum. Meffert, Postbus 9, 3956 ZR Leersum.

Te koop aangeboden: Rotronics wafadrive incl. tekstverwerkingprogramma, 2 extra 64K wafa's en Ientronics + RS 232 int. na kantooruren 071—72 02 56.

Te koop Philips videopac spelcomputer + 9 cassettes o.a. Pacman, Qbert, Nieuwwaardje ± f 900,—, nu f 450,—. Leen Zwartbol, Schenkeldijk 63, Strijen. Tel. 01854—18 20 na 18.30 uur.

Te koop: MPF-I comp. os-2KBTBASIC-2kB, Z80/CTC/PIO. Boards: printer/disassembl, eeprom (1—4k) programmer. Manuals, voedingen, eeprom, RAM, exp. board — Nieuw — f 700,—. Tel.: 01891—46 49. ■

# Hobbyscoopprogramma's aangepast

Met de bestaande BASICODE-inleesprogramma's voor de Spectrum of ZX81 kunnen helaas niet alle uitgezonden BASICODE-programma's vertaald worden naar lopende Sinclair-programma's. Om de programma's dan toch aan de gang te krijgen, zal het een en ander gewijzigd moeten worden. In het november- en decembernummer van vorig jaar liet Rik Koevoets zien hoe een aantal programma's van de demonstratiecassette van de NOS veranderd kunnen worden. In het maartnummer van dit jaar behandelde hij de programma's die op 5 december 1984 door NOS-Hobbyscoop werden uitgezonden. Jacobus van Doeselaar is bij zijn programma-aanpassingen van de na 1 april uitgezonden programma's uitgegaan van het gebruik van de BASICODE-inleesroutine van Herman Ament (versie 2.2.), uitgezonden op 12 december 1984. Degenen die dit programma niet hebben opgenomen, kunnen een kopie bestellen door f 8,— over te maken op gironummer 3874370 t.n.v. Herman Ament, Wagneningen. Voor het inlezen en vertalen van de BASICODE-programma's kunnen we verwijzen naar de toelichting bij Herman Aments vertaalprogramma in het mei- en juninummer van Sinclair Gebruiker. Voor de op deze bladzijden behandelde programma's is de volgende vertaalroutine van toepassing:

- lees het programma in met routine 1
- vertaal het ingelezen programma met routine 3, gebruikmakend van de originele subroutines, de data als "VAL" behandelend en met "FN"
- pas de variabelen aan met routine 4
- DELETE menu prog.

DELETE 111, 121, 790 t/m 794. Na het inlezen en vertalen kunnen de

Het wijzigen van BASICODE-programma's om ze op de Spectrum draaiende te krijgen gaat niet altijd zonder problemen. Jacobus van Doeselaar geeft deze maand zijn aanpassingen prijs van een aantal van de na 1 april door NOS-Hobbyscoop uitgezonden programma's.

programma's worden aangepast. In de listings op deze bladzijden staan de regels zoals ze er uit moeten gaan zien; de regels met hetzelfde regelnummer in het BASICODE-programma moeten door deze regels worden vervangen.

## Korfbal 1 en 2

Twee korfbal-enthousiastelingen schreven dit programma, dat uit twee delen bestaat. Het eerste deel (uitgezonden op 26 april) heeft betrekking op de theorie van zaalkorfbal, het tweede deel (uitgezonden op 3 mei) behandelt de spelregels. Pas het eerste deel met behulp van deze regels aan:

```
1010 DIM D$(1+3,1): DIM A(1+30)
11: DIM B(1+30): DIM R$(1+7,34)
3240 LET C=0: LET B=15: GO SUB 1
10
```

```
3245 IF B$="D" OR B$="d" THEN ST
OP
3250 PRINT "Doe de cassette met
het tweede programma in de reco
rder, start de recorder en het
programma wordt geladen..."
3252 LOAD ""
3254 STOP
4340 PAUSE U*10: RETURN
```

De gewijzigde regels voor het tweede deel:

```
5 RUN 1000
10 LET Q$=X$
11 FOR U=LEN X$ TO 1 STEP -1
12 LET Y$=X$(U TO U)
13 IF Y$(">") THEN GO TO 15
14 NEXT U
15 LET X$=Q$( TO U)
16 RETURN
1010 DIM F$(1+3,48): DIM A(1+30)
11: DIM B(1+30): DIM C$(1+7,39): D
IM B$(1+10,1): DIM C(1+3): DIM S
$(1+10,4)
1800 LET K$=F$(1+J, TO 32): GO S
UB 100: GO SUB 4520: GO SUB 4500
1820 PRINT "Dit is veld ";J
2335 FOR U=2 TO 14 STEP 2: PRINT
"AT U,2;" 18 spaties "; NEXT U
3505 FOR U=4 TO 14 STEP 2: PRINT
"AT U,2;" 17 spaties "; NEXT U
4060 GO SUB 4780: PRINT " Dit is
speler ";P
4155 FOR U=4 TO 14 STEP 2: PRINT
"AT U,2;" 17 spaties "; NEXT U:
PRINT "AT 16,0;" 17 spaties "
4380 PRINT "Je hebt ";S;" ";M$;"
goed."
4880 LET X$=F$(1+F): GO SUB 10:
PRINT X$: NEXT F: GO SUB 4990
4990 PAUSE U*5: RETURN
5040 LET X$=C$(1+B/2): GO SUB 1
0: PRINT X$;: NEXT B: RETURN
```

## Statistiek

Dit programma voor wiskundeliefhebbers werd uitgezonden op 3 mei. Het programma past zich automatisch aan de afmetingen van het scherm aan. Naar keuze kunnen permutaties, combinaties, faculteiten, gemiddelden en verzamelingen worden berekend. Als u niet weet wat dat allemaal zijn, kunt u daar in het programma meer over lezen. Aanpassen van het programma:

```
DELETE 20- 1820- 2310- 2320-
2330- 2350- 2370- 2550
20 FOR A=6 TO 22: PRINT AT A,0;"
32 spaties "; NEXT A
30 PRINT AT 6,0: LET tel=0
40 RETURN
```

## WORDT PROGRAMMEUR

Rob Baas en Carel van der Velden (auteurs van o.m. de LOI-cursussen: Microcomputers en BASIC, en Pascal) bieden een aantal van hun cursisten de mogelijkheid goedkoop een mondelinge opleiding tot PASCAL - PROGRAMMEUR te volgen.

(4 AMBI deelcertificaten!) Vraag folder aan bij:  
POSTBUS 69 6920 AB DUIVEN

(Advertentie)

```

1000 LET A=200: LET U$="" : LET F
#0: REM STATISTIEK
1160 LET UR=22
1830 GO SUB 2045
2045 PRINT AT 22,25; BRIGHT 1;"D
ruk op ENTER..."; BRIGHT 0
2050 IF CODE INKEY$<>13 THEN GO
TO 2050
2120 GO SUB 1900: INPUT X1: LET
X1=INT X1: PRINT ; BRIGHT 1;X1;
BRIGHT 0
2155 LET tel=0
2165 IF tel>=13 THEN GO SUB 20
2170 PRINT "Getal ";M; " = "; IN
PUT X(1+M): PRINT X(1+M)
2185 LET tel=tel+1
2340 PRINT "(Dus tussen de ";C-0
;" en de ";C+0
2350 PRINT " zitten ongeveer 70%
van de getallen)";
2360 GO SUB 2045
2450 GO SUB 1900: INPUT Y1: PRIN
T ; BRIGHT 1;Y1; BRIGHT 0
2560 GO SUB 2045
2700 GO SUB 1900: INPUT P1: PRIN
T BRIGHT 1;P1; BRIGHT 0
2890 LET VE=VE+3: LET HO=0: LET
S$=""
2900 GO SUB 1900
2970 GO SUB 1900: PRINT "(1-33)
"; INPUT U1: PRINT ; BRIGHT 1;U
1; BRIGHT 0
3010 PRINT "(1-";U1;") "; INPUT
U2: PRINT ; BRIGHT 1;U2; BRIGHT
0
3200 LET VE=VE+3: LET HO=0: LET
S$=""
3210 GO SUB 1900
3300 INPUT K: PRINT ; BRIGHT 1;K
; BRIGHT 0
3370 GO SUB 1900: INPUT L: PRINT
; BRIGHT 1;L; BRIGHT 0
3580 LET HO=4: LET VE=VE+2: GO S
UB 110
3590 LET H$=STR$ T: LET S$="Er i
s ijn "+H$+" combinaties."
3610 LET S$=""

```

### Flevo

Dit programma geeft een overzicht van de uitzendtijden van de Engelstalige programma's van de Wereldomroep. De plaats waar men zich bevindt is van belang, evenals het tijdstip waarop men naar de uitzending luistert. De bijbehorende frequenties worden naar keuze op het scherm of op de printer geleverd. Het programma houdt verband met de zenderwisseling van Radio Nederland, die onlangs de nieuwe zenders en antenne's in de Flevopolder in gebruik heeft genomen. Uitzenddatum: 24 mei.

```

1020 DIM E$(11,20): DIM F$(11,30)
: DIM G$(11,37): DIM H$(11,23):
DIM K$(10,32)
1045 PRINT AT 23,25;"Druk op ENT
ER..."
2030 PRINT 0;": ";K$(1+0-1)
2040 NEXT 0
2470 LET A$="" : FOR Q=1 TO LEN
(K$(1+R-1))-5: LET A$=A$+CHR$(4
5): NEXT Q

```

```

2500 LET A$="" : FOR Q=1 TO LEN
(K$(1+R-1))-5: LET A$=A$+CHR$(4
5): NEXT Q
2550 PRINT AT 23,25;"Druk op ENT
ER..."
2552 IF CODE INKEY$<>13 THEN GO
TO 2552
2654 GO TO 2270

```

### Twee keer patience

Het eerste Patience-programma werd uitgezonden op 5 juni. Breng de volgende wijzigingen aan:

```

1000 LET SC=0: LET B$="" : LET A=
1000: REM PATIENCE
1002 FOR I=USR "a" TO USR "e"-1:
READ data: POKE I,data: NEXT I
1004 DATA 0,0,20,20,62,62,42,0,0
,20,62,62,20,20,0,0,0,0,20,62,
20,0,0,0,20,20,0,62,62,42,0
1020 LET H$="B"
1030 LET K$="D"
1040 LET R$="C"
1050 LET S$="A"

```

(De regels 1002 en 1004 hebben tot gevolg dat de letters A, B, C en D wanneer zij in de G-mode worden ingetoetst, de kaartsymbolen voor harten, klaveren, ruiten en schoppen opleveren. In de vorenstaande regels zijn de letters A, B, C en D dan ook ingetoetst in de G-mode. Na één keer kunnen worden deze regels vanzelf aangepast.)

```

1020 LET H$="F"
1030 LET K$="S"
1040 LET R$="T"
1050 LET S$="A"

```

Verdere aanpassingen:

```

1070 DIM A$(52,3): DIM X(35): DI
M Y(35)
1150 RESTORE 3650
1160 READ X(P),Y(P)
1265 PRINT AT 19,13;"Momentje..."
1370 LET A$(P)=FN H$(B$,R,3)
1390 LET B$=B$( TO R-1)+B$(R+3 T
O)
1450 LET HO=X(P): LET E=Y(P): GO
SUB 110
1460 PRINT A$(P)
1610 LET G$(P)
2410 IF A$(C)=" " THEN IF C>7
THEN LET C=C-7: GO TO 2410
2420 IF A$(C)=" " THEN GO TO 2
540
2450 LET A=CODE (FN M$(A$(C),2,1
))
2490 LET HO=X(C): LET E=Y(C): GO
SUB 110

```

```

2510 LET G$=A$(C): LET A$(C,)=
2560 FOR J=1 TO 50: NEXT J: LET
UR=1: RETURN
2750 PRINT I$
2930 GO SUB 110: PRINT I$

```

Op 7 juni zond NOS-Hobbyscoop een tweede Patience-programma uit. Het aanpassen van dit programma gaat als volgt:

```

1000 POKE 23656,6: LET N=0: REM
PATIENCE
1040 DIM I$(52,3): DIM H$(52): DI
M J(52): DIM P(5,16): DIM M$(14,
2): DIM K$(5,1): DIM Q$(5,2)
1320 PRINT "BEGINCOORDINATEN: X1
"; INPUT A$: L
ET HO=23: GO SUB 110: PRINT A$
1410 LET VE=0: LET HO=28: GO SUB
110: PRINT " Y1=" : INPUT A$:
PRINT A$
1490 PRINT "EINDCOORDINATEN: X2
"; INPUT A$: L
ET HO=23: GO SUB 110: PRINT A$
1550 LET VE=0: LET HO=28: GO SUB
110: PRINT " Y2=" : INPUT A$:
PRINT A$
1600 PAUSE 200
2740 IF C$=" " AND E$="HR" THEN
GO SUB 5200: RETURN
2770 IF D$=" " AND E$="HR" AND Y
2=1 THEN RETURN
2780 IF D$=" " THEN GO TO 2800
3030 PAUSE 200
3210 IF Q$(11+21)= " " AND R$=" R
THEN GO TO 3350
3475 PRINT AT 20,3;"Momentje..."
4450 PAUSE 160
5010 PAUSE 200

```

Vorenstaande aanpassingen zijn noodzakelijk om het programma op de Spectrum te laten rennen. De aanpassing/uitbreiding van het programma met de symbolen van de speelkaarten geschiedt door het tussenvoegen en wijzigen van onderstaande regels:

```

1002 FOR I=USR "a" TO USR "e"-1:
READ data: POKE I,data: NEXT I
1004 DATA 0,20,62,62,20,20,0,0,0,
20,62,62,20,20,0,0,0,0,20,62,
20,0,0,0,20,20,0,62,62,42,0

```

Door de regels 1002 en 1004 worden de grafische A, B, C en D toetsen gekoppeld met de symbolen van de speelkaarten van achtereenvolgens harten, klaver, ruiten en schoppen. Onderstaande regels moeten worden aangepast, waarbij:  
— H vervangen wordt door A in G-mode  
— K vervangen wordt door B in G-mode  
— R vervangen wordt door C in G-mode  
— S vervangen wordt door D in G-mode



```

2650 IF A$="A" AND C$="D" THEN G
O TO 2770
2670 IF A$="C" AND C$="D" THEN G
O TO 2770
2680 IF A$="A" AND C$="B" THEN G
O TO 2770
2690 IF A$="C" AND C$="B" THEN G
O TO 2770
2700 IF A$="D" AND C$="A" THEN G
O TO 2770
2710 IF A$="B" AND C$="A" THEN G
O TO 2770
2720 IF A$="D" AND C$="C" THEN G
O TO 2770
2730 IF A$="B" AND C$="C" THEN G
O TO 2770
3140 IF Y2=1 AND A$="A" THEN GO
TO 3190
3150 IF Y2=2 AND A$="D" THEN GO
TO 3190
3160 IF Y2=3 AND A$="B" THEN GO
TO 3190
3170 IF Y2=4 AND A$="C" THEN GO
TO 3190
5320 DATA "A","B","C","D"

```

Na één keer kunnen worden deze regels vanzelf aangepast. Als alles goed is gedaan, zijn de regels als volgt veranderd:

```

2650 IF A$="V" AND C$="A" THEN G
O TO 2770
2670 IF A$="+" AND C$="A" THEN G
O TO 2770
2680 IF A$="V" AND C$="S" THEN G
O TO 2770
2690 IF A$="+" AND C$="S" THEN G
O TO 2770
2700 IF A$="A" AND C$="V" THEN G
O TO 2770
2710 IF A$="S" AND C$="V" THEN G
O TO 2770
2720 IF A$="+" AND C$="+" THEN G
O TO 2770
2730 IF A$="A" AND C$="+" THEN G
O TO 2770
3140 IF Y2=1 AND A$="V" THEN GO
TO 3190
3150 IF Y2=2 AND A$="A" THEN GO
TO 3190
3160 IF Y2=3 AND A$="S" THEN GO
TO 3190
3170 IF Y2=4 AND A$="+" THEN GO
TO 3190
5320 DATA "V","S","+", "A"

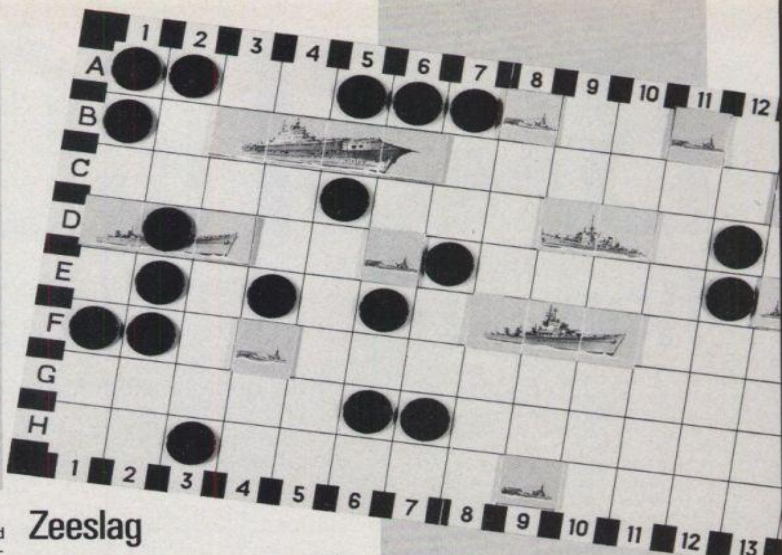
```

In het BASICODE-vertaalprogramma van Herman Ament worden de arrayvariabelen tijdens het vertalen met 1 opgehoogd. Als men hier geen rekening mee houdt, loopt het programma wel, maar ontbreekt er een kaart. Het spel bevat dan 51 kaarten en dat is natuurlijk niet de bedoeling. Daarom moet er nog een aantal regels worden gewijzigd:

```

1300 IF PP=1 AND D=54 THEN GO TO
4950
1830 IF D=54 THEN RETURN
2310 IF D=54 AND X2=9 THEN GO T
O 2340
2320 IF D=54 THEN GO SUB 1730:
GO TO 1250

```



## Zeeslag

Tenslotte nog een programma van de demonstratiecassette van de NOS. Het spel Zeeslag, het tweede programma van de demonstratiecassette, wil voor Spectrumbezitters nogal eens wat moeilijkheden opleveren. Het spel bevat een Nederlandse, een Engelse, een Franse en een Duitse versie. Er moeten tamelijk veel regels worden gewijzigd:

```

310 LET B$="00000000000000000000
000000000000" ( TO CT) : IF (ABS S
R=1E-5) OR (ABS SR=1E5) OR (CT
<CN*LEN STR$(INT SR)) THEN RETU
RN
1010 DIM D$(15,15,1) : DIM A(6) :
DIM B(6) : DIM C(6) : DIM D(6) : DI
M E(11)
1020 DIM F(4) : DIM E$(5,3) : DIM
C$(10,5) : DIM F$(4,8) : DIM S$(6,
23)
1030 DIM G(6) : DIM H(6) : DIM G$(
24,42)
1205 PRINT AT 9+NO,26; BRIGHT 1;
NO; BRIGHT 0
1300 PRINT G$(1, TO 16)
1305 PRINT AT 23,0;"Ogenblikje..
"
1300 PRINT G$(3, TO 41); GO SUB
210
1570 DELETE deze regel
1800 PRINT G$(24, TO 17)
1810 PAUSE 35
1830 PAUSE 35
1920 LET C=10: LET D=0: GO SUB 1
10: PRINT G$(1, TO 16)
2070 LET C=2: LET D=21: GO SUB 1
10: PRINT "VERT:  " : HORZ:  " : G$(
5, TO 18)
2115 PRINT AT 21,7; BRIGHT 1;" "
: BRIGHT 0; AT 21,15;
2190 IF (X<1) OR (X>14) THEN GO
TO 2115
2200 PRINT AT 21,15; BRIGHT 1;" "
: BRIGHT 0
2205 LET C=15: GO SUB 110
2500 LET D=13: GO SUB 110: PRINT
G$(6, TO 9)
2520 LET D=14: GO SUB 110: PRINT
G$(7, TO 9)

```

```

2540 LET D=16: GO SUB 110: PRINT
G$(8, TO 9)
2550 LET D=17: GO SUB 110: PRINT
G$(9, TO 9)
2620 LET C=20: LET D=3: GO SUB 1
10: PRINT G$(4, TO 10)
2875 PRINT AT 22,0;"Ogenblikje..
"
3015 PRINT AT 20,0;"Ogenblikje 9
edu(d...
"
3020 FOR I=1 TO 500: READ U: NEX
T
3150 DELETE deze regel
3150 GO SUB 100: PRINT G$(14, TO
25) : PRINT G$(15, TO 25)
3170 PRINT : PRINT : PRINT G$(1
0, TO 39)
3230 PRINT G$(11, TO 16);RP
3240 PRINT G$(12, TO 16);MP
3270 PRINT G$(16, TO 20) : PRINT
G$(17, TO 25) : PRINT : PRINT
3290 PRINT G$(1+16+I);
3350 PRINT G$(22, TO 39) : PRINT
G$(23) : PRINT : PRINT
3370 PRINT G$(2, TO 27); GO SUB
210
3940 DATA "LOGSSCHEPEN BESTAAT,
TOT ZINKEN TE BRENGEN"
4060 DATA "DO YOU WANT TO SEE TH
E RULES OF THE GAME?"
4310 DATA "VOULEZ VOUS VOIR LES
REGLES DE JEU (O/N)?"

```

De aanpassingen voor alle programma's van de BASICODE-demonstratiecassette van de NOS en alle sedert maart uitgezonden BASICODE-programma's kunt u bestellen bij Jacobus van Doeselaar door overmaking van f 5,— op girorekening 1675174 t.n.v. J.C. van Doeselaar, Oss. Een aangepast BASICODE-programma (uitgezonden na 1 maart 1985) op cassette kunt u tegen kostprijs bij hem bestellen door overmaking van f 10,—. Kortere programma's kunnen eventueel op één cassette worden gezet, langere programma's moeten apart worden besteld.

# BASICODE

## hoe krijg ik het lopende?

In het begin van dit jaar heeft NOS HOBBIYSCOOP een cassette geproduceerd met de titel "Best of basicode no: 1". Het bevat een reeks programma's (16) van de heer B.P.M. Rintjema, welke door Hobbyscoop wordt omschreven als "Een zo omvangrijk stuk programmatuur dat we het niet konden uitzenden vanwege zijn lengte". De functies van het pakket worden omschreven als "Een integrale huishoudrekening", en de programma's vervullen naast bestandsonderhoudsfuncties ook functies zoals tekstverwerking. De gebruiksmogelijkheden en handleiding vindt u in vier aparte programma's die u echt van alles vertellen.

Tijdens het vertalen van alle 16 programma's van de cassette bleek echter, nadat de toelichtingsprogramma's vertaald waren, dat de programma's waar het werkelijk om gaat, niet zonder meer op de Spectrum lopen. Het probleem zit hem in de wijze waarop de heer Rintjema met variabelen omgaat, hetgeen hem al gauw noodzaakt meer dan 26 string-variabelen te gebruiken, en dat levert onoverkomelijke problemen op.

Na enkele telefoontjes met de heer Rintjema hebben wij afgesproken samen te trachten variabelen die gecombineerd kunnen worden te onderscheiden, om zo doende toch alle programma's lopende te krijgen. Of dit lukt is niet geheel zeker, maar mocht dat het geval zijn, dan leest u het zeker in deze kolom.

Voor degene die de cassette al gekocht hebben, volgen hier de aanwijzingen die u kunnen helpen bij het vertalen van de eerste vijf programma's van de cassette. Alle hier opgegeven wijzigingen slaan op het gebruik met de BASICODE versie 5.2(a) of 5.3 van Rik Koevoets. Een aantal wijzigingen/oplossingen van problemen geldt ook voor het gebruik van de vertaalroutine van Herman Ament.

### Truc

In de toelichtingsprogramma's wordt op diverse plaatsen naar regelnum-

Een onlangs door de NOS uitgebrachte programmacassette wordt door Rik Koevoets van aantekeningen voorzien.

mers verwezen. Daar deze na het hernummeren niet meer terug te vinden zijn, en voor het hernummeren soms onzichtbaar, kunt u de volgende truc toepassen: Laadt het desbetreffende programma nogmaals, maar plaats voor het vertalen (RUN 700) bijvoorbeeld op regel 1001 de volgende regel:

```
1001 REM GOTO 25000
```

Het woordje "GOTO" tikt u in als losse letters, het regelnummer erachter is het regelnummer waarvan u wilt weten wat het na hernummering wordt. Nadat u het programma vertaald heeft (en hernummerd) zal regel 1001, die intussen 1010 is geworden, u precies aangeven wat er van regel 25000 gemaakt is.

### De toelichtingsprogramma's

Het volgende rijtje variabelen dient vervangen te worden bij alle toelichtingsprogramma's. Eigenlijk zijn dit vier dezelfde programma's met andere DATA-regels. Wellicht werkt het vervangen met de RUN 400-routine wat prettiger als u regel 495 vervangt door "495 GOTO 425". U hoeft dan niet steeds "RUN 400, 2 (voor vervangen)" in te toetsen.

woord → vervangen door:

```
"INS" → "IS" "WDS" → "JS"
"SR$" → "RS" "LL$" → "KS"
"N1$" → "B$" "RL$" → "OS"
"N2$" → "CS" "RS$" → "PS"
"RT$" → "DS" "NN$" → "MS"
"RW$" → "ES" "NA$" → "QS"
"LR$" → "GS" "NP$" → "SS"
"LS$" → "HS" "DT$" → "PS"
```

Na deze wijzigingen zullen de toelichtingen nog vastlopen met een foutmelding "integer out of range". Dit wordt voorkomen door de volgende wijzigingen:

```
2330 IF FN r$(L$,3) = "999" THEN
GO TO 2360
2330 IF LEN L$ >= 3 THEN IF
FN r$(L$,3) = "999" THEN
GO TO 2360
```

```
2470 IF FN r$(K$,3) = "..." THEN
LET L$ = G$: GO TO 2560
2470 IF LEN L$ >= 3 THEN IF
FN r$(K$,3) = "..." THEN
LET L$ = G$: . . .
```

```
2480 IF FN r$(K$,3) = "..." THEN
LET L$ = H$: GO TO 2560
2480 IF LEN L$ >= 3 THEN IF
FN r$(K$,3) = "..." THEN
LET L$ = H$: . . .
```

### Hoofdmenu

Het hoofdmenuprogramma kan een centrale plaats in het totale pakket innemen. Het grondidee is dat, door middel van dit programma, alle andere programma's worden opgeroepen. U moet dan wel op de diverse plaatsen, die duidelijk worden aangegeven zelf de statements voor de Spectrum toevoegen. Wellicht kan het heel aardig samen werken met microdrives.

woord → vervangen door:

```
"N1$" → "AS"
"N2$" → "BS"
"TT$" → "CS"
"INS" → "IS"
"SR$" → "RS"
```

```
1030 LET B$ = "HOOFD-MENU": GO
SUB 1330
1030 LET B$ = "HOOFD-MENU": LET
M = 0: GO SUB 1330
```

```
2410 PRINT "Als u 2 disk-drives hebt,
doe:"
2410 PRINT "Als u 2 disk-drives hebt,
doe:"
```



```
1510 DATA " " " " " X" " X
  XX
1520 DATA " " " " X" " X " "
  X
1530 DATA "X " " "X "X" "X" " X
  XXX"
```

Bij deze DATA-statements staan origineel 5 tekens tussen de quotes (""). Maak hier 4 tekens van door steeds de middelste te verwijderen.

## Programmeerwedstrijd

Op 9 september 1984 werd vanaf de Firato de uitslag van de BASICODE-2-programmeerwedstrijd bekendgemaakt. De uitgezonden programma's die avond stonden beide in het teken van deze wedstrijd. Het eerste programma gaf een lijst met de winnaars, het tweede was het winnende programma "Gitaar-accorden". Dit programma won de prijs vanwege zijn originele toepassing. Daar ben ik het roerend mee eens. Maar als er gekeken was naar de toegepaste programmeertechnieken, waren anderen met de eer gaan strijken. Het programma "Gitaar-accorden" gebruikt namelijk meer dan 26 verschillende stringnamen, M1\$, M2\$, M3\$ enz. Hierdoor kan het programma onmogelijk op de SINCLAIR-computers vertaald en gedraaid worden. Het gebruik van zoveel verschillende stringvariabelen kan ook nooit gerechtvaardigd worden door een behoefte aan zoveel verschillende namen. Veel eerder zou men een tweede DIMensie moeten gebruiken. T\$ dus voor de teksten. Bij meer teksten wordt dan gebruik gemaakt van T\$(1), T\$(2), T\$(3), en nooit van T1\$, T2\$, T3\$. Bij het verwerken van deze stringvariabelen moet in het tweede geval voor iedere variabele een stukje programma geschreven worden terwijl bij het gebruik van meer DIMensies door middel van een FOR—NEXT loop steeds hetzelfde stukje programma gebruikt kan worden.

Voorbeeld:

```
1100 LET T1$=B$+MP$
1110 LET H0=M1: LET VE=U1
1120 LET SR$=T1$: GOSUB 300

1130 LET T2$=B$+MP$
1140 LET H0=M1: LET VE=U1
1150 LET SR$=T2$: GOSUB 300

1160 LET T3$=B$+MP$
1170 LET H0=M1: LET VE=U1
1180 LET SR$=T3$: GOSUB 300

↓

1100 FOR N=1 TO ET
1110 LET T$(N)=B$+MP$
1120 LET H0=M1: LET VE=U1
1130 LET SR$=T$(N): GO SUB 300

1140 NEXT N
```

## Faculteit

Uitzenddatum: 16 september 1984. Dit programma berekent faculteiten. Dat gaat niet erg snel. Maar faculteit-berekeningen zijn altijd erg langdurig. Dat merk je wel aan je pocket-calculator, die er ook een beetje moeite mee lijkt te hebben. Pocket-calculators houden meestal op bij faculteit 69. Dit programma gaat dan nog even door. Verander de volgende variabele:

```
woord: --> vervangen door:
"SR$-->"R$"
```

Als u de volgende regel toevoegt, zal het programma niet steeds stoppen met de vraag "SCROLL?", maar door blijven rekenen tot faculteit 255.

```
1165 POKE 23692,255
```

## Rente-berekeningen

Uitzenddatum: 23 september 1984. "Rente-berekeningen" is een erg fraai en gebruikersvriendelijk programma. We noemen het menu-gestuurd omdat steeds gebruik wordt gemaakt van een aantal keuzeschermen die naar de diverse renteberekeningen leiden. Verander de volgende variabelen:

```
woord: --> vervangen door:
"INS"-->"J$"
"SR$"-->"B$"
"LT"-->"L1$"
"KT"-->"K1$"
"TT$"-->"T$"
```

Verwijder op de volgende 3 regelnummers de spaties zoals aangegeven. Zie bovendien onderstaande wijziging van de standaard routine GO SUB 310.

```
4110, 4210, 4310 PRINT "JAAR  SC
SCHULD  AFLÖSSING  TOTAL"
↓
4110, 4210, 4310 PRINT "JAAR  SC
HULD  AFLÖSSING  TOTAL"

4160, 4260, 4360 PRINT " "I$,"
"R$," "T$
↓
4160, 4260, 4360 PRINT I$," "R$
" "T$
```

Regel 1710 en 3080 bevatten als laatste teken een dubbele punt. Nu zit er in het vertaalprogramma een mechanisme dat waar nodig automatisch de benodigde "LET"- en "GO TO"-statements tussenvoegt, aan het begin van de regel, en na dubbele punten. Daar er na die laatste dubbele punt op regel 1710 en 3080 een getal volgt, het regelnummer van de volgende regel, wordt hier (ten onrechte) een "GO TO"-statement gegenereerd. U moet dit op beide gevallen verwijderen.

## Wijziging BASICODE standaard routine GO SUB 310

Een perfectionering in de standaard-routine van regel 310 is o.a. nodig voor het bovenstaande interestprogramma van 23 september. Breng deze wijzigingen als volgt aan: — Laad de BASICODE-2 inleesroutine versie 5.2a. — Tik de volgende regels in:

```
310 LET R$="" : LET CN=CN+1: LET
SR$=SR$+CN
312 LET R$=STR$(INT (SR$10^CN+
E1/20^CN) : LET CN=CN-1: LET R$=R
$( TO (LEN R$-1)) : IF R$(LEN R$-
1) THEN LET R$=R$ TO (LEN R$-
1)
313 IF LEN R$=CT THEN RETURN
314 IF LEN R$=CT THEN LET R$=""
315 GO TO 313
9999 CLOSE #9
```

— Verander de variabele R\$ nu in SR\$ door middel van de RUN 400-routine en verwijder daarna regel 9999. — SAVE het nu gewijzigde BASICODE-inleesprogramma door in te tikken: "POKE 23728,1: GO TO 9000" ENTER. — Werk in het vervolg met deze versie.

(Advertentie)

**NU VERKRIJGBAAR**  
VOOR COMODORE 64 EN SPECTRUM

**TOTO** PRIJS 49,50

- \* 40K MONSTERPROGRAMMA
- \* VOORSPELLING TOTO-UITSLAGEN
- \* MEER DAN 40 GEWOONEN FACTOREN
- \* WAARONDER HANDBOOKS
- \* HOUDT ALLE STANDEN BIJ
- \* OOK VOOR JE EIGEN KOMPETITIE
- \* EN VOOR ANDERE SPORTEN

**NEDERLANDSTALIG**  
NET ALS

**TASWORD TWEE**

- \* DE BEROEMDE TEKSTVERWERKER
- \* GEBEEL NEDERLANDSE VERTAALING
- \* PRINT 64 LETTERS PER REGEL
- \* IDEEAAL VOOR GROTE PRINTERS
- \* MAAK OOK VOOR ZX OF SIKOSHA
- \* DUIDELIJKE HANDLEIDING
- \* VOOR 48K SPECTRUM F 70

**EN OOK EDUCATIEF**  
BIJVOORBEELD

**SPELLINGSBAK** EN  
TT

- \* NEDERLANDSE SPELLINGSREGELS
- \* IN THEORIE EN OEFENINGEN
- \* DOOR LERAAR NEDERLANDS
- \* VOOR SCHOOL EN THUIS
- \* PRIJS PER DEEL F 49,50

VEEL MEER PROGRAMMA'S STAAN IN DE GRATIS CATALOGUS VRAAG 3 AAN BIJ FILOSOF POSTBUS 1353 3701 BIJ GRONINGEN BESTELLING TELEFONISCH OF DOOR OVERMAKING V-H VERSCHULDIGDE BEDRAG PRIJS F 3,50 VERZENDKOSTEN OP GIREN 20792 TNV FILOSOF GRONINGEN

**FILOSOF** DETAILHANDEL  
TEL. INFORMATIE  
BEL 650-137746

# BASICODE

## gjør oversetting av BASIC-versjoner til en lek!

Av RAGNAR FYRI

Basic og Basic, fru Blom. Utallige varianter har vi nå fått etter hvert, og de er faktisk enda forskjelligere enn de ser ut — selv om alle Basic-maskiner staver «skriv ut» PRINT, lagres kommandoen som Basicprogram i kodeform, med et bestemt tall for hver kommando; det er interpreteren som (tolken) som staver det for oss når vi sier LIST. Så selv om vi kunne kopiere innholdet av RAM på en maskin over i RAM på en annen type — noe som heller ikke lar seg gjøre, de lagrer jo data på forskjellig måte alle sammen — ville det ikke være mulig å kjøre det resulterende programmet (hvis det da i det hele tatt ble til noe som lignet et program!) hverken med eller uten videre.

Så hvordan får vi et VIC-program til å virke på en TRS? Vi skaffer oss en utskrift av programmet, taster nøysommelig inn alle linjene, retter feil, modifierer, retter feil, kompenserer for forskjellig skjermformat, retter feil, retter feil, osv. Langt på natt tar vi så en pause for å gruble over om det ikke kunne la seg gjøre å skrive ut mer programmer på noe en datamaskin kunne lese selv og «taste inn» uten feil; og tenk så fint det hadde vært med et magisk formular som kunne få et program til å kjøre på en hvilken som helst maskin.

Slutt å gruble kjære leser! Det finnes en slik metode. Det finnes en slik trylleformel. Det magiske ordet er BASICODE - et fra brukerens synspunkt ganske enkelt system for lagring av programmer på kassettbånd som i enkelte tilfelle faktisk er mer pålitelig enn enkelte maskiners «medfødte» SAVE- og LOAD-rutiner. Med Basicode kan man skrive et program på en hvilken som helst av for tiden 17 forskjellige maskiner, spille det inn på bånd, spille det av på en hvilken som helst av de 16 andre maskinene, og så tut og kjør! Den magiske formelen som gjør det siste mulig er en «pakke» med subrutiner som er forskjellig fra maskin til maskin, men gjør nøyaktig det samme — sett fra hovedpro-

grammet (som er det eneste som lagres på bånd — subrutinene lagres altså ikke) fungerer denne pakken som en «sort boks» — man sender et par parametere inn i et mørkt hull, og så skjer det ett eller annet der inne. Samme hva som skjer - et PRINT @, en VTAB og en HTAB, eller en streng kontrolltegn — resultatet blir det samme, i dette tilfelle at markøren dukker opp på et bestemt sted på skjermen.

Basicode er utviklet av nederlandsk radio (!), nærmere bestemt redaksjonen for hobbyprogrammet Hobbyscoop (—skoop), som er kjent (?) for sine eksperimenter med utsending av computerprogrammer pr. radio — i Basicode, selvfølgelig. Den som sender 25 gylden (ca. 60 kroner) til NOS-Hobbyscoop, Postbus 1200, 1200 BE Hilversum, Nederland, får en bok med utførlige forklaringer av Basicode på nederlandsk og et noe haltende (men leselig) engelsk, og en kassett med over setterprogrammer for forskjellige maskiner pluss 21 demonstrasjonsprogrammer i Basicode-format som faktisk nesten er verdt prisen alene. De som har Apple, BBc, CBM, Commodore 64, PEt, Vic 20 eller Sharp trenger ikke mer enn kassetten og boksen for å komme i gang. Brukere av Sorcerer, TRS-80 eller Videogenie trenger et mer komplisert interface.

Nederlandsk radio (nærmere bestemt Hilversum 2 og 4) når såvidt jeg vet ikke fram til Norge, men NOS (som for øvrig står for Nederlandsche Om roep Stichting — nederlandsk kringkastingsselskap (?)) produserer også et engelskspråklig program kalt «Radio Activity» som de sender ut (på bånd) til en rekke radiostasjoner rundt i verden, som så sender ut programmene lokalt. Den nærmeste radiostasjonen som er med på dette samarbeidet sies å ligge i Sverige, men jeg har ikke fått noen svar på spørsmål om hvilken stasjon det er eller når den sender i Basic-code. Hva om vi gikk for å overtale NRK?

### Basicode-News



Zu unserem Artikel über *Basicode* (64'er 5/90, Seite 17 ff.) erreichten uns folgende Hinweise:

- Unter der ISBN 3-473-44010-8 ist bereits 1984 ein deutschsprachiges Buch beim Otto Maier Verlag, Ravensburg, erschienen. Autoren waren Michael Wiegand sowie Manfred und Heike Fillingner in Zusammenarbeit mit NOS.

- «dt-64» hat sämtliche *Basicode*-Sendungen eingestellt.

- «DDR II» sendet 14tägig freitags von 23.05 bis 23.20 Uhr auf UKW (REM-Spezial).

- Die Sendung «*Basicode*-Software-Service» wird über 1008 kHz schon um 21 Uhr ausgestrahlt, nicht um 21.35 Uhr.

- Das Begleitmaterial steht bei «Radio DDR» nicht mehr zur Verfügung. Dort sind aber deutschsprachige Versionen der *Basicode*-Übersetzungsprogramme für C64, Sinclair Spectrum, Atari und die KC-Serie (DDR) auf Schallplatte erhältlich. (pd)

Radio DDR: Schulfunk, Herrn Dr. J. Baumann, Nalepastraße 10-50, DDR-1160 Berlin  
Stichting Basicode: Herr Haubrich, Postbus 1410, NL-5602 BK Eindhoven  
N.O.S. Hilversum: Hobbyscoop, Postbus 1200, NL-1200 BE Hilversum  
M. van Deelen: Rudolphlaan 28, NL-3794 MZ de Glind (speziell zum C64/C128)

The majority of modern hobby computers use the programming language BASIC. However, that does not mean that a BASIC program can be exchanged between two different types of computers, either directly or via a cassette. The BASIC commands may well be the same but the way in which the computer deals with them and how they are put on cassette are often completely different. Basicode was developed to solve this problem. It is a sort of universal communication standard to allow BASIC programmes to be interchangeable between different types of computers.

# basicode-2

a code to  
make BASIC  
programmes  
exchangeable

It is about two years since NOS, the Dutch broadcasting company, came up with the idea of developing a standardised code to make it possible to exchange BASIC programs between two different types of computers. As with most things that Murphy gets a hand in, this is not entirely straightforward. First of all there is the problem of storage on cassette. Most hobby computers use cassette recorders as a means of storing programmes. The method of registering data on tape and the frequencies used are different for each type of computer. A second difficulty is the BASIC language used. Even though a standard BASIC exists, each computer uses a different 'dialect' with its own peculiarities. There is also a problem as regards how programs are stored and processed within a computer, as there is no international agreement on this. Because of these factors BASIC, even though it is widespread, is not at all interchangeable between two computers that 'think' differently.

The Basicode standard is a fixed audio code by means of which BASIC programs can be stored on cassette. Through this standardisation, programs can be written onto cassette from any type of computer and read back to any type of computer. That is not to say that Basicode is simply a translation programme to store BASIC programs on tape in a specific manner. Just as important are the agreement on the BASIC commands used, the arrangement of line numbers, the names of variables and the screen format.

At present there is already a second version of Basicode available that uses a series of standard subroutines. At the same time a few other points have been changed from the original version with the aim of making Basicode even more universal. This Basicode-2 is the subject of this article.

## Basicode on tape

Basicode uses frequencies of 1200 and

2400 Hz. A logic '0' corresponds to one whole period of 1200 Hz while a logic '1' is two full periods of 2400 Hz. Each byte is transmitted serially at a rate of 1200 baud, and every byte is built up as follows (see also figure 1):

- 1 start bit (logic zero)
- 8 data bits, least significant bit first
- 2 stop bits (logic one)

The BASIC programme is coded character by character in the form seen when a LIST command is given. No internal computer notation is used. All letters and figures are simply represented in ASCII code. Every BASIC instruction must be followed by a space, and each BASIC line must be finished with CR (carriage return,  $8D_{hex}$ ). The most significant bit of every ASCII sign is made '1'.

A complete programme on tape consists of the following sections:

- a leader consisting of a 5 second tone of 2400 Hz
- the ASCII sign 'start text' ( $82_{hex}$ )
- the BASIC programme in ASCII code
- the ASCII sign 'end of text' ( $83_{hex}$ )
- a 'checksum'
- a trailer, consisting of a 5 second tone of 2400 Hz.

The checksum, which is used for error detection, consists of a bit-by-bit exclusive OR function of all previous bytes (including the 'start text' sign). This checksum is 8 bits long (1 byte).

## The Basicode-2 protocol

### General agreements

The only BASIC statements allowed are those which are known by all computers. These statements are listed in table 1, and we will return to this later. A number of line numbers are reserved for special defined subroutines. This ensures that certain operations are possible that cannot easily be achieved in standard BASIC. These routines are not transmitted with the pro-

gram, so they must either be a part of the Basicode translation programme or they must be written in separately before RUNNING a BASIC programme.

The screen dimensions are fixed at 24 lines of 40 characters. Because some computers have less than 24 lines on the screen or less than 40 characters per line, it is recommended that no more than 16 screen lines be used and that the lines should be kept as short as possible.

A program line, including line number, spaces and carriage return, can have a maximum of 60 signs.

#### How a program is built up

The following groups of line numbers are reserved in Basicode-2:

0-999: standard routines. These are specially developed for the computer in question and are supplied through the translation programme or are read in separately.

1000: the first line of the program. It must have this form:

1000 A = (value): GOTO 20: REM program name (value) is the maximum number of characters that are used together in all strings. By jumping to line 20, the computers that need it reserve some memory space for the strings.

1010: the first line that can be used for the program.

1010-32767: space for the program. There is no compulsory system within the programme, but the developers of Basicode recommend the following groupings:

1000-19999: main program

20000-24999: subroutines for the programme, in which statements exist that are not permitted in Basicode-2

25000-29999: data statements

30000-32767: REM statements. These can be a description of the program, references or the name and address of the programmer.

It is recommended that the line numbers are increased in steps of 10. As regards the subroutines at lines 20000-24999, these should be avoided as much as possible. If this is not possible, it should be made perfectly clear what each subroutine does.

#### Standard subroutines in Basicode-2

These subroutines are very much dependent upon each particular computer so this is just a general description of the function of the subroutines with no examples given.

**GOSUB 100:** This clears the screen and places the cursor at position 0.0 (upper left corner of the screen).

**GOSUB 110:** Set the cursor at a specific place on the screen. The desired location must be stored in variables HO and VE. HO is the position in a line (0 is completely left) and VE gives a line number (uppermost line is number 0). As the screen format in Basicode-2 is 40 characters on 24 lines, HO cannot be greater than 39 and VE no bigger than 23. The values of HO and VE do not change by calling this subroutine.

**GOSUB 120:** The position of the cursor on the screen is set in the variables HO and VE. With this system HO = 0 is the first position in a line and VE = 0 is the top line. This routine can be used with the previous one to, for example, move the cursor one or more lines higher or lower.

**GOSUB 200:** See if a button is pressed and store the value of this key in IN\$. If no key is pressed at that moment, IN\$ is empty. In principle, control characters could also be stored but this requires caution as these have different meanings for different computers. One exception is RETURN, which is ASCII code 13 in all computers.

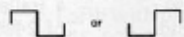
**GOSUB 210:** This routine waits until a key is pressed and stores the value in IN\$. This routine actually waits for a key to be pressed, whereas in the previous one a value was only stored if a key was pressed at the actual instant when the routine was running.


**GOSUB 250:** This subroutine gives a bleep in computers that have this facility. The frequency and duration of the bleep are not specified here.

**GOSUB 260:** An arbitrary number between 0 and 1 is generated and stored in variable RV.

**GOSUB 270:** The whole variable space is cleared up and the routine finds out how much memory space remains (the variables are not cleared!). The number of free bytes is stored in variable FR.

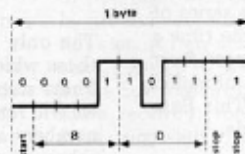
Figure 1. This is how the transfer format is built up in Basicode. Note that transfer begins with the least significant byte.

0:  f = 1200 Hz

1:  f = 2400 Hz

Byte build up:

Example: 08hex (ASCII 8 with "1" as the eighth bit)



83101-1-1

**GOSUB 300:** The value of variable SR is stored as a string in SR\$. The string cannot contain a space at the beginning or end of a number. This is in contrast to STR\$ which does this sometimes. STR\$ is not permitted as a Basicode-2 statement in any case.

**GOSUB 310:** This routine supplies a string SR\$ built up as follows. The value of SR\$ is equal to the contents of variable SR and is always in fixed-point notation. The total length of SR\$ contains CT characters and the number of characters after the decimal point is defined by CN. If the number does not fit in the stated format SR\$ consists of CT asterisks. CT, CN and SR are not changed by calling this routine. An example of this routine is: CT = 7, CN = 3 and SR = 0.6666, then SR\$ = ' 0.667'.

**GOSUB 350:** Prints SR\$ on the printer but does not finish the line yet. This makes it possible to print different strings one after another on the same line.

**GOSUB 360:** Closes a print line with a carriage return and new line command.

**Variables**

To ensure that the interchangeability of programs is maintained, there are some limitations as regards the variables used in any program:

- Numeric variables are always real and with single precision.
- The name of a variable can only have a maximum of two characters, and the first must be a letter. The second may, depending on use, be a letter or number. String variables have a \$ after the name. Lower case letters are not permitted in a variable.
- Logic variables can only be either true or not true. Any value that could be confused with something else by the computer may not be used, for example +1 for true and 0 for not true.
- It must not be assumed that all variables are reset to zero at the start of a programme.
- String variables can be no longer than 255 characters.
- Variables may not begin with the letter Q, as this is reserved for the standard subroutines.

- Variables AS, AT, FN, GR, IF, PI, ST, TI, TI\$, and TO may not be used.
- The variables HO, VE, FR, SR, CN, CT, RV, IN\$ and SR\$ are used for communication between the BASIC programme and the standard subroutines.

**BASIC limitations**

Table 1 gives a summary of all the permitted BASIC commands and operators. Here some basic agreements are necessary. There are some variants in the BASIC language but usually the meanings of commands are much the same as in the official BASIC, so we will not discuss the variations here.

There are, however, a few points about BASIC commands that do require clarification. A variable name may not be used directly after a GOSUB or GOTO; so A = 1000 : GOTO A is not permitted. The command IF must always be followed by THEN. For example: IF ... THEN A = 5, IF ... THEN 1000 and IF ... THEN GOSUB 20000. The form IF ... THEN ... ELSE is not allowed. Comments or multiple variables are not permitted after an INPUT; so INPUT 'The value is'; A\$ is forbidden. A line number may not be given after RUN. If using the TAB statement, remember that some computers start counting at zero and others begin at one.

**In practice**

Those are the most important points about Basicode-2. Apart from these, a translation program and the 'permitted' subroutines are needed but we will not give them here because they are different for each computer. The translation program is in machine code and sometimes has a BASIC part, depending on the type of computer. There are already programs available for various different types of computer, and generally a specialized computer club can help here. If everything went according to plan, the Basicode-2 book is already available, giving the complete Basicode-2 protocol and several different translation programmes for common types of computer. Further information can be obtained from Hans G. Janssen, Hobbyscoop, Postbus 1200, 1200 BE Hilversum, The Netherlands. The Basicode-2 book itself, which is printed with English and Dutch in the same book, is also available from Hobbyscoop. Basicode programs are also broadcast during the Hobbyscoop programme on Sundays from 17.10 ... 17.45 GMT (summer) or 18.10 ... 18.45 (winter) on 747 kHz. Finally, to return to our own Junior Computer. Elsewhere in this issue we have an article giving the translation programme and various subroutines for the BASIC Junior Computer. Translation programmes for both the expanded Junior and the DOS junior are available and this article has both of them!

**Table 1.** These are the permitted BASIC commands and operations.

ABS	DIM	INPUT	NOT	RETURN	STOP
AND	END	LEFT\$	ON	RIGHT\$	TAB
ASC	FOR	LEN	OR	RUN	TAN
ATN	GOSUB	LET	PRINT	SGN	THEN
CHR\$	GOTO	LOG	READ	SIN	TO
COS	INT	MID\$	REM	SQR	VAL
DATA	IF	NEXT	RESTORE	STEP	
+	↑	<>			
-	=	<=			
*	<	>=			
/	>				



Elsewhere in this issue we described the theory behind Basicode-2 so it is only natural that we should show how the Junior Computer can use it. Here we give the Basicode software and everything else that is necessary to allow the Junior Computer to use Basicode-2. This means that the Junior Computer can now easily exchange BASIC programs stored on cassette tape with other computers. Moreover, 'received' programs can run directly on the JC, so that BASIC in combination with Basicode is a universal, completely exchangeable computer language.

Basicode-2 interface for the Junior Computer  
elektor october 1983

# basicode-2

## interface for the Junior Computer

As we have already described all the various facets of Basicode, we will simply begin here by talking about the Junior with Basicode. The Basicode translation programs for the expanded Junior and the DOS Junior are not the same as they use different BASICs and handle their memory in different ways. To use Basicode, either an expanded Junior with KB-9-BASIC and Elekterminal or a DOS Junior and Elekterminal are needed.

### The translation programs

The translation programs for both Junior versions are written in machine code. The complete source listing is given in table 1, complete with explanatory text. This is for the expanded Junior with KB-9-BASIC. The source listing for the DOS Junior is not given as it is almost the same as this listing, only a few of the addresses are different. The hexdumps are shown in table 2 (Junior with KB-9-BASIC) and table 3 (DOS Junior).

In the expanded Junior with KB-9-BASIC the translation program is at addresses \$0200...\$059B, and in the DOS Junior it is at \$E000...\$E39B. These ranges are selected because there is generally RAM there, and the programs really have to be in RAM to work properly (so they cannot be placed in an EPROM). Once the program is typed in, it can simply be written to a cassette or floppy disk, so that the next time it is to be used it can easily be read in. The program consists of a write and a read section. We will concentrate on the expanded Junior in order to describe how the program is used, but at any point where the DOS Junior differs, this is indicated by the comments in brackets.

### Reading

First the Basicode translation program is typed in (or read in, if it is already stored on cassette). Both read and write programs can be stored in one file on cassette: SA = 0200, EA = 059C (DOS Junior: SA = E000, EA =

E39C). Next the KB-9-BASIC is read in from cassette (see Elektor April 1982), or from a floppy in the case of the DOS Junior. Then the BASIC can be started in the usual way. At this stage a Basicode program can be loaded. This requires the small interface described at the end of this article. A program is loaded as follows:

First type NEW to erase any old programs.

Then type:

POKE 8256,0 : POKE 8257,4 : X = USR(X)

(POKE 574,0 : POKE 575,226 : X =

USR(X))

followed by a (carriage) return.

The sign  $\equiv$  now appears on the hex display of the Junior and indicates that there is no synchronization. The cassette recorder can then be started. If the program receives any signals the  $\equiv$  sign jumps back and forth on the two right-hand displays. If the 2400 Hz header is now received, a slowly jumping sign appears on the right-hand displays. This shows that the program is working on synchronizing. This jumping display lasts about 2 seconds, then the sign is stationary on both displays for the rest of the leader. At the end of the leader when the actual program begins, both displays show  $\equiv$ , and as long as the data is properly received this sign lights evenly on both displays. When the complete program is received, the computer automatically gives a listing of it on the screen or printer. After doing this, the computer gives an 'OK'.

If an error has appeared while reading in the program, the message 'CHECKSUM ERROR' is given after the listing. In this case the program must be checked or it could be read in again in the hope of a better result. On no account must the listing be interrupted by pressing a key. If this is done, there is a chance that both BASIC and Basicode programs will have to be read again (or retyped!). Even if faults are seen in the listing, such as lines being written over one another (that can happen if there is sudden interference on the tape), you must still wait until the

Junior communicates with other computers

Basicode-2 interface  
in the J.C.  
elektor october 1983

Table 1

0010:	BASICODE WRITE PROGRAM FOR JUNIOR COMPUTER	1140:	0279 A9 09	LDAM	003
0020:	WITH KB-9 BASIC	1150:	0281 20 42 02	JGR	TABLE
0030:		1160:	0284 A0 F8 03	LDA	0000H
0040:		1170:	0287 20 40 02	JGR	CHECKSUM TO TABLE (WITHOUT SETTING BIT7)
0050:	DATS:14-6-83	1180:	028A A9 7F	LDAM	07F
0060:		1190:	028C 00 0E 10	STA	00R
0070:	ORG 00100	1200:	028F A9 C0	LDAM	00F
0080:		1210:	0291 00 00 10	STA	00R
0090:	POINTERS IN PAGE ZERO (BASIC)	1220:	0294 A9 01	LDAM	001
0100:		1230:	0296 00 05 10	STA	00R
0110:	0299	1240:	0299 20 20 03	JGR	HEADER
0120:	0299	1250:	029C 20 39 03	LDADTA	03R
0130:	0299	1260:	029F A0 5F 0F	LDTM	03FFH
0140:	0299	1270:	02A2 A8 06	LDAM	000
0150:	0299	1280:	02A4 4A	OUTDMA	03RA
0160:		1290:	02A6 40	RMA	
0170:	0299	1300:	02A8 08 05	RCS	HIGH
0180:	0299	1310:	02AB 20 39 03	JGR	END ONE PERIOD OF 1200HZ (#0)
0190:	0299	1320:	02AD 70 00	DVS	HEAT
0200:	0299	1330:	02B0 20 40 03	JGR	ONE TWO PERIODS OF 2400HZ (#1)
0210:	0299	1340:	02B2 08	HEXT	PLA
0220:	0299	1350:	02B5 08	HEXT	DEV
0230:	0299	1360:	02B8 08 00	DVS	OUTDMA
0240:	0299	1370:	02BA 00 40 03	JGR	ONE STOPBIT+L
0250:	0299	1380:	02BC 20 40 03	JGR	ONE STOPBIT+L
0260:	0299	1390:	02BE 00 00 00	DVS	OUTDMA
0270:	0299	1400:	02C0 08 03	RMA	LDT
0280:	0299	1410:	02C2 00 A1 02	RMC	03R0AL+02
0290:	0299	1420:	02C5 AC A3 02	LDC	03R
0300:	0299	1430:	02C8 AC A8 02	LDC	LDTM+01
0310:	0299	1440:	02CB C8 05	CPE	HIMDML
0320:	0299	1450:	02CE 08 00	DVS	OUTDMA
0330:	0299	1460:	02D0 08 03	RMA	HIMDML
0340:	0299	1470:	02D2 08 00	RMC	03R0AL
0350:	0299	1480:	02D5 08 00	RMC	03R0AL
0360:	0299	1490:	02D8 00 A0 02	STA	03R0AL+01
0370:	0299	1500:	02DB 00 A5 70	LDA	007F
0380:	0299	1510:	02DE 00 A1 02	LDA	007F
0390:	0299	1520:	02E0 AC 0A 02	CAWAVE	03V
0400:	0299	1530:	02E3 00 00 00	DVS	03R0AL+02
0410:	0299	1540:	02E6 00 00 00	DVS	03R0AL+02
0420:	0299	1550:	02E9 00 00 00	DVS	03R0AL+02
0430:	0299	1560:	02EB 00 00 00	DVS	03R0AL+02
0440:	0299	1570:	02EE 00 00 00	DVS	03R0AL+02
0450:	0299	1580:	02F0 00 00 00	DVS	03R0AL+02
0460:	0299	1590:	02F3 00 00 00	DVS	03R0AL+02
0470:	0299	1600:	02F6 00 00 00	DVS	03R0AL+02
0480:	0299	1610:	02F9 00 00 00	DVS	03R0AL+02
0490:	0299	1620:	0302 00 00 00	DVS	03R0AL+02
0500:	0299	1630:	0305 00 00 00	DVS	03R0AL+02
0510:	0299	1640:	0308 00 00 00	DVS	03R0AL+02
0520:	0299	1650:	0311 00 00 00	DVS	03R0AL+02
0530:	0299	1660:	0314 00 00 00	DVS	03R0AL+02
0540:	0299	1670:	0317 00 00 00	DVS	03R0AL+02
0550:	0299	1680:	0320 00 00 00	DVS	03R0AL+02
0560:	0299	1690:	0323 00 00 00	DVS	03R0AL+02
0570:	0299	1700:	0326 00 00 00	DVS	03R0AL+02
0580:	0299	1710:	0329 00 00 00	DVS	03R0AL+02
0590:	0299	1720:	0332 00 00 00	DVS	03R0AL+02
0600:	0299	1730:	0335 00 00 00	DVS	03R0AL+02
0610:	0299	1740:	0338 00 00 00	DVS	03R0AL+02
0620:	0299	1750:	0341 00 00 00	DVS	03R0AL+02
0630:	0299	1760:	0344 00 00 00	DVS	03R0AL+02
0640:	0299	1770:	0347 00 00 00	DVS	03R0AL+02
0650:	0299	1780:	0350 00 00 00	DVS	03R0AL+02
0660:	0299	1790:	0353 00 00 00	DVS	03R0AL+02
0670:	0299	1800:	0356 00 00 00	DVS	03R0AL+02
0680:	0299	1810:	0359 00 00 00	DVS	03R0AL+02
0690:	0299	1820:	0362 00 00 00	DVS	03R0AL+02
0700:	0299	1830:	0365 00 00 00	DVS	03R0AL+02
0710:	0299	1840:	0368 00 00 00	DVS	03R0AL+02
0720:	0299	1850:	0371 00 00 00	DVS	03R0AL+02
0730:	0299	1860:	0374 00 00 00	DVS	03R0AL+02
0740:	0299	1870:	0377 00 00 00	DVS	03R0AL+02
0750:	0299	1880:	0380 00 00 00	DVS	03R0AL+02
0760:	0299	1890:	0383 00 00 00	DVS	03R0AL+02
0770:	0299	1900:	0386 00 00 00	DVS	03R0AL+02
0780:	0299	1910:	0389 00 00 00	DVS	03R0AL+02
0790:	0299	1920:	0392 00 00 00	DVS	03R0AL+02
0800:	0299	1930:	0395 00 00 00	DVS	03R0AL+02
0810:	0299	1940:	0398 00 00 00	DVS	03R0AL+02
0820:	0299	1950:	0401 00 00 00	DVS	03R0AL+02
0830:	0299	1960:	0404 00 00 00	DVS	03R0AL+02
0840:	0299	1970:	0407 00 00 00	DVS	03R0AL+02
0850:	0299	1980:	0410 00 00 00	DVS	03R0AL+02
0860:	0299	1990:	0413 00 00 00	DVS	03R0AL+02
0870:	0299	2000:	0416 00 00 00	DVS	03R0AL+02
0880:	0299	2010:	0419 00 00 00	DVS	03R0AL+02
0890:	0299	2020:	0422 00 00 00	DVS	03R0AL+02
0900:	0299	2030:	0425 00 00 00	DVS	03R0AL+02
0910:	0299	2040:	0428 00 00 00	DVS	03R0AL+02
0920:	0299	2050:	0431 00 00 00	DVS	03R0AL+02
0930:	0299	2060:	0434 00 00 00	DVS	03R0AL+02
0940:	0299	2070:	0437 00 00 00	DVS	03R0AL+02
0950:	0299	2080:	0440 00 00 00	DVS	03R0AL+02
0960:	0299	2090:	0443 00 00 00	DVS	03R0AL+02
0970:	0299	2100:	0446 00 00 00	DVS	03R0AL+02
0980:	0299	2110:	0449 00 00 00	DVS	03R0AL+02
0990:	0299	2120:	0452 00 00 00	DVS	03R0AL+02
1000:	0299	2130:	0455 00 00 00	DVS	03R0AL+02
1010:	0299	2140:	0458 00 00 00	DVS	03R0AL+02
1020:	0299	2150:	0461 00 00 00	DVS	03R0AL+02
1030:	0299	2160:	0464 00 00 00	DVS	03R0AL+02
1040:	0299	2170:	0467 00 00 00	DVS	03R0AL+02
1050:	0299	2180:	0470 00 00 00	DVS	03R0AL+02
1060:	0299	2190:	0473 00 00 00	DVS	03R0AL+02
1070:	0299	2200:	0476 00 00 00	DVS	03R0AL+02
1080:	0299	2210:	0479 00 00 00	DVS	03R0AL+02
1090:	0299	2220:	0482 00 00 00	DVS	03R0AL+02
1100:	0299	2230:	0485 00 00 00	DVS	03R0AL+02
1110:	0299	2240:	0488 00 00 00	DVS	03R0AL+02
1120:	0299	2250:	0491 00 00 00	DVS	03R0AL+02
1130:	0299	2260:	0494 00 00 00	DVS	03R0AL+02
1140:	0299	2270:	0497 00 00 00	DVS	03R0AL+02
1150:	0299	2280:	0500 00 00 00	DVS	03R0AL+02
1160:	0299	2290:	0503 00 00 00	DVS	03R0AL+02
1170:	0299	2300:	0506 00 00 00	DVS	03R0AL+02
1180:	0299	2310:	0509 00 00 00	DVS	03R0AL+02
1190:	0299	2320:	0512 00 00 00	DVS	03R0AL+02
1200:	0299	2330:	0515 00 00 00	DVS	03R0AL+02
1210:	0299	2340:	0518 00 00 00	DVS	03R0AL+02
1220:	0299	2350:	0521 00 00 00	DVS	03R0AL+02
1230:	0299	2360:	0524 00 00 00	DVS	03R0AL+02
1240:	0299	2370:	0527 00 00 00	DVS	03R0AL+02
1250:	0299	2380:	0530 00 00 00	DVS	03R0AL+02
1260:	0299	2390:	0533 00 00 00	DVS	03R0AL+02
1270:	0299	2400:	0536 00 00 00	DVS	03R0AL+02
1280:	0299	2410:	0539 00 00 00	DVS	03R0AL+02
1290:	0299	2420:	0542 00 00 00	DVS	03R0AL+02
1300:	0299	2430:	0545 00 00 00	DVS	03R0AL+02
1310:	0299	2440:	0548 00 00 00	DVS	03R0AL+02
1320:	0299	2450:	0551 00 00 00	DVS	03R0AL+02
1330:	0299	2460:	0554 00 00 00	DVS	03R0AL+02
1340:	0299	2470:	0557 00 00 00	DVS	03R0AL+02
1350:	0299	2480:	0560 00 00 00	DVS	03R0AL+02
1360:	0299	2490:	0563 00 00 00	DVS	03R0AL+02
1370:	0299	2500:	0566 00 00 00	DVS	03R0AL+02
1380:	0299	2510:	0569 00 00 00	DVS	03R0AL+02
1390:	0299	2520:	0572 00 00 00	DVS	03R0AL+02
1400:	0299	2530:	0575 00 00 00	DVS	03R0AL+02
1410:	0299	2540:	0578 00 00 00	DVS	03R0AL+02
1420:	0299	2550:	0581 00 00 00	DVS	03R0AL+02
1430:	0299	2560:	0584 00 00 00	DVS	03R0AL+02
1440:	0299	2570:	0587 00 00 00	DVS	03R0AL+02
1450:	0299	2580:	0590 00 00 00	DVS	03R0AL+02
1460:	0299	2590:	0593 00 00 00	DVS	03R0AL+02
1470:	0299	2600:	0596 00 00 00	DVS	03R0AL+02
1480:	0299	2610:	0599 00 00 00	DVS	03R0AL+02
1490:	0299	2620:	0602 00 00 00	DVS	03R0AL+02
1500:	0299	2630:	0605 00 00 00	DVS	03R0AL+02
1510:	0299	2640:	0608 00 00 00	DVS	03R0AL+02
1520:	0299	2650:	0611 00 00 00	DVS	03R0AL+02
1530:	0299	2660:	0614 00 00 00	DVS	03R0AL+02
1540:	0299	2670:	0617 00 00 00	DVS	03R0AL+02
1550:	0299	2680:	0620 00 00 00	DVS	03R0AL+02
1560:	0299	2690:	0623 00 00 00	DVS	03R0AL+02
1570:	0299	2700:	0626 00 00 00	DVS	03R0AL+02
1580:	0299	2710:	0629 00 00 00	DVS	03R0AL+02
1590:	0299	2720:	0632 00 00 00	DVS	03R0AL+02
1600:	0299	2730:	0635 00 00 00	DVS	03R0AL+02
1610:	0299	2740:	0638 00 00 00	DVS	03R0AL+02
1620:	0299	2750:	0641 00 00 00	DVS	03R0AL+02
1630:	0299	2760:			

Basicode-2 interface  
for the J.C.  
elektor october 1983

```

0000: BASICODE READ PROGRAM FOR JUNIOR COMPUTER
0001: WITH KB-9 BASIC
0002: DATE(14-8-83)
0003: ORG 0000
0004: POINTERS IN PAGE ZERO (BASIC)
0005:
0006:
0007:
0008:
0009:
0010:
0011:
0012:
0013:
0014:
0015:
0016:
0017:
0018:
0019:
0020:
0021:
0022:
0023:
0024:
0025:
0026:
0027:
0028:
0029:
0030:
0031:
0032:
0033:
0034:
0035:
0036:
0037:
0038:
0039:
0040:
0041:
0042:
0043:
0044:
0045:
0046:
0047:
0048:
0049:
0050:
0051:
0052:
0053:
0054:
0055:
0056:
0057:
0058:
0059:
0060:
0061:
0062:
0063:
0064:
0065:
0066:
0067:
0068:
0069:
0070:
0071:
0072:
0073:
0074:
0075:
0076:
0077:
0078:
0079:
0080:
0081:
0082:
0083:
0084:
0085:
0086:
0087:
0088:
0089:
0090:
0091:
0092:
0093:
0094:
0095:
0096:
0097:
0098:
0099:
0100:
0101:
0102:
0103:
0104:
0105:
0106:
0107:
0108:
0109:
0110:
0111:
0112:
0113:
0114:
0115:
0116:
0117:
0118:
0119:
0120:
0121:
0122:
0123:
0124:
0125:
0126:
0127:
0128:
0129:
0130:
0131:
0132:
0133:
0134:
0135:
0136:
0137:
0138:
0139:
0140:
0141:
0142:
0143:
0144:
0145:
0146:
0147:
0148:
0149:
0150:
0151:
0152:
0153:
0154:
0155:
0156:
0157:
0158:
0159:
0160:
0161:
0162:
0163:
0164:
0165:
0166:
0167:
0168:
0169:
0170:
0171:
0172:
0173:
0174:
0175:
0176:
0177:
0178:
0179:
0180:
0181:
0182:
0183:
0184:
0185:
0186:
0187:
0188:
0189:
0190:
0191:
0192:
0193:
0194:
0195:
0196:
0197:
0198:
0199:
0200:
0201:
0202:
0203:
0204:
0205:
0206:
0207:
0208:
0209:
0210:
0211:
0212:
0213:
0214:
0215:
0216:
0217:
0218:
0219:
0220:
0221:
0222:
0223:
0224:
0225:
0226:
0227:
0228:
0229:
0230:
0231:
0232:
0233:
0234:
0235:
0236:
0237:
0238:
0239:
0240:
0241:
0242:
0243:
0244:
0245:
0246:
0247:
0248:
0249:
0250:
0251:
0252:
0253:
0254:
0255:
0256:
0257:
0258:
0259:
0260:
0261:
0262:
0263:
0264:
0265:
0266:
0267:
0268:
0269:
0270:
0271:
0272:
0273:
0274:
0275:
0276:
0277:
0278:
0279:
0280:
0281:
0282:
0283:
0284:
0285:
0286:
0287:
0288:
0289:
0290:
0291:
0292:
0293:
0294:
0295:
0296:
0297:
0298:
0299:
0300:
0301:
0302:
0303:
0304:
0305:
0306:
0307:
0308:
0309:
0310:
0311:
0312:
0313:
0314:
0315:
0316:
0317:
0318:
0319:
0320:
0321:
0322:
0323:
0324:
0325:
0326:
0327:
0328:
0329:
0330:
0331:
0332:
0333:
0334:
0335:
0336:
0337:
0338:
0339:
0340:
0341:
0342:
0343:
0344:
0345:
0346:
0347:
0348:
0349:
0350:
0351:
0352:
0353:
0354:
0355:
0356:
0357:
0358:
0359:
0360:
0361:
0362:
0363:
0364:
0365:
0366:
0367:
0368:
0369:
0370:
0371:
0372:
0373:
0374:
0375:
0376:
0377:
0378:
0379:
0380:
0381:
0382:
0383:
0384:
0385:
0386:
0387:
0388:
0389:
0390:
0391:
0392:
0393:
0394:
0395:
0396:
0397:
0398:
0399:
0400:
0401:
0402:
0403:
0404:
0405:
0406:
0407:
0408:
0409:
0410:
0411:
0412:
0413:
0414:
0415:
0416:
0417:
0418:
0419:
0420:
0421:
0422:
0423:
0424:
0425:
0426:
0427:
0428:
0429:
0430:
0431:
0432:
0433:
0434:
0435:
0436:
0437:
0438:
0439:
0440:
0441:
0442:
0443:
0444:
0445:
0446:
0447:
0448:
0449:
0450:
0451:
0452:
0453:
0454:
0455:
0456:
0457:
0458:
0459:
0460:
0461:
0462:
0463:
0464:
0465:
0466:
0467:
0468:
0469:
0470:
0471:
0472:
0473:
0474:
0475:
0476:
0477:
0478:
0479:
0480:
0481:
0482:
0483:
0484:
0485:
0486:
0487:
0488:
0489:
0490:
0491:
0492:
0493:
0494:
0495:
0496:
0497:
0498:
0499:
0500:
0501:
0502:
0503:
0504:
0505:
0506:
0507:
0508:
0509:
0510:
0511:
0512:
0513:
0514:
0515:
0516:
0517:
0518:
0519:
0520:
0521:
0522:
0523:
0524:
0525:
0526:
0527:
0528:
0529:
0530:
0531:
0532:
0533:
0534:
0535:
0536:
0537:
0538:
0539:
0540:
0541:
0542:
0543:
0544:
0545:
0546:
0547:
0548:
0549:
0550:
0551:
0552:
0553:
0554:
0555:
0556:
0557:
0558:
0559:
0560:
0561:
0562:
0563:
0564:
0565:
0566:
0567:
0568:
0569:
0570:
0571:
0572:
0573:
0574:
0575:
0576:
0577:
0578:
0579:
0580:
0581:
0582:
0583:
0584:
0585:
0586:
0587:
0588:
0589:
0590:
0591:
0592:
0593:
0594:
0595:
0596:
0597:
0598:
0599:
0600:
0601:
0602:
0603:
0604:
0605:
0606:
0607:
0608:
0609:
0610:
0611:
0612:
0613:
0614:
0615:
0616:
0617:
0618:
0619:
0620:
0621:
0622:
0623:
0624:
0625:
0626:
0627:
0628:
0629:
0630:
0631:
0632:
0633:
0634:
0635:
0636:
0637:
0638:
0639:
0640:
0641:
0642:
0643:
0644:
0645:
0646:
0647:
0648:
0649:
0650:
0651:
0652:
0653:
0654:
0655:
0656:
0657:
0658:
0659:
0660:
0661:
0662:
0663:
0664:
0665:
0666:
0667:
0668:
0669:
0670:
0671:
0672:
0673:
0674:
0675:
0676:
0677:
0678:
0679:
0680:
0681:
0682:
0683:
0684:
0685:
0686:
0687:
0688:
0689:
0690:
0691:
0692:
0693:
0694:
0695:
0696:
0697:
0698:
0699:
0700:
0701:
0702:
0703:
0704:
0705:
0706:
0707:
0708:
0709:
0710:
0711:
0712:
0713:
0714:
0715:
0716:
0717:
0718:
0719:
0720:
0721:
0722:
0723:
0724:
0725:
0726:
0727:
0728:
0729:
0730:
0731:
0732:
0733:
0734:
0735:
0736:
0737:
0738:
0739:
0740:
0741:
0742:
0743:
0744:
0745:
0746:
0747:
0748:
0749:
0750:
0751:
0752:
0753:
0754:
0755:
0756:
0757:
0758:
0759:
0760:
0761:
0762:
0763:
0764:
0765:
0766:
0767:
0768:
0769:
0770:
0771:
0772:
0773:
0774:
0775:
0776:
0777:
0778:
0779:
0780:
0781:
0782:
0783:
0784:
0785:
0786:
0787:
0788:
0789:
0790:
0791:
0792:
0793:
0794:
0795:
0796:
0797:
0798:
0799:
0800:
0801:
0802:
0803:
0804:
0805:
0806:
0807:
0808:
0809:
0810:
0811:
0812:
0813:
0814:
0815:
0816:
0817:
0818:
0819:
0820:
0821:
0822:
0823:
0824:
0825:
0826:
0827:
0828:
0829:
0830:
0831:
0832:
0833:
0834:
0835:
0836:
0837:
0838:
0839:
0840:
0841:
0842:
0843:
0844:
0845:
0846:
0847:
0848:
0849:
0850:
0851:
0852:
0853:
0854:
0855:
0856:
0857:
0858:
0859:
0860:
0861:
0862:
0863:
0864:
0865:
0866:
0867:
0868:
0869:
0870:
0871:
0872:
0873:
0874:
0875:
0876:
0877:
0878:
0879:
0880:
0881:
0882:
0883:
0884:
0885:
0886:
0887:
0888:
0889:
0890:
0891:
0892:
0893:
0894:
0895:
0896:
0897:
0898:
0899:
0900:
0901:
0902:
0903:
0904:
0905:
0906:
0907:
0908:
0909:
0910:
0911:
0912:
0913:
0914:
0915:
0916:
0917:
0918:
0919:
0920:
0921:
0922:
0923:
0924:
0925:
0926:
0927:
0928:
0929:
0930:
0931:
0932:
0933:
0934:
0935:
0936:
0937:
0938:
0939:
0940:
0941:
0942:
0943:
0944:
0945:
0946:
0947:
0948:
0949:
0950:
0951:
0952:
0953:
0954:
0955:
0956:
0957:
0958:
0959:
0960:
0961:
0962:
0963:
0964:
0965:
0966:
0967:
0968:
0969:
0970:
0971:
0972:
0973:
0974:
0975:
0976:
0977:
0978:
0979:
0980:
0981:
0982:
0983:
0984:
0985:
0986:
0987:
0988:
0989:
0990:
0991:
0992:
0993:
0994:
0995:
0996:
0997:
0998:
0999:

```

Table 1. This is the source listing for the complete translation program. This particular example is for the expanded Junior with KB-9-BASIC. The listing for the DOS Junior is almost identical to the one shown here, with the exception of a few address locations.

Basiccode-2 interface for the Junior Computer  
elektor october 1983

Table 2

```

M
HEXDUMP: 0200,0363
0 1 2 3 4 5 6 7 8 9 A B C D E F
0200: AD 52 2A AC 53 2A 8D F8 03 8C F1 03 A9 02 A8 02
0210: 8D 52 2A 8C 53 2A A5 04 A5 05 02 F2 03 02 F3 03
0220: A9 77 A8 02 05 04 04 05 A5 7E A4 7F 8D 49 02 8C
0230: 0A 02 8D A8 02 8C A1 02 A9 00 8D F4 03 8D F8 03
0240: A9 02 C9 0A F8 30 09 80 8D FF FF 4D F4 03 8D F4
0250: 03 EE 49 02 08 03 EE 4A 02 A5 85 CD 4A 02 08 14
0260: A5 84 CD 49 02 08 0F A9 FF 8D F8 03 A5 78 8D 49
0270: 02 A5 79 8D 4A 02 60 48 98 08 A9 8D 20 42 02 A5
0280: 03 20 42 02 AD F4 03 28 40 02 A9 7F 8D 0E 18 A9
0290: CF 8D 08 18 A9 01 8D 05 18 20 03 20 39 03 AD
02A0: FF FF AD 08 4A 48 08 05 20 39 03 70 03 20 46 03
02B0: 68 88 D8 F0 20 46 03 20 46 03 EE A9 02 D0 03 EE
02C0: A1 02 AC A1 02 AE A0 02 C4 05 D8 0E E4 84 D8 0A
02D0: A5 78 8D A8 02 A5 79 8D A1 02 CC 0A 02 D8 8D EC
02E0: 49 02 D8 88 20 2B 03 0C 00 18 AD F2 03 AC F3 03
02F0: 85 04 85 05 AD F8 03 AC F1 03 8D 52 2A 8C 53 2A
0300: AD F8 03 02 20 A5 79 85 7D 05 7D 05 7E A4 70 C8
0310: C8 C8 84 7A 84 7C 84 7E A9 00 A8 91 78 C8 91 78
0320: A8 2C 20 59 05 68 A8 68 4C 03 98 A2 70 A0 17 20
0330: 46 03 CA D8 FA 88 D8 F7 68 A9 9F 8D 96 18 A9 01
0340: 8D 07 18 4C 53 03 20 49 03 A9 CE 8D 96 18 A9 01
0350: 8D 07 18 AD 04 18 2C 8D 18 58 FB AD 04 18 2C 8D
0360: 18 58 FB 68

```

JUNIOR

```

M
HEXDUMP: 0400,0500
0 1 2 3 4 5 6 7 8 9 A B C D E F
0400: A9 7F 8D 8E 18 A9 8D 8C 18 8D F4 03 A9 73 8D
0410: 82 1A A5 7E 8D 05 04 A6 7F 8E 84 05 8D 62 8E 66
0420: 04 A3 10 8D 06 03 A9 36 28 4D 05 28 FD 04 C9 22
0430: 98 FF C9 4E 88 8B EE F5 03 D8 F7 83 28 48 05 98
0440: C8 F6 03 D8 06 8A 38 19 8C 8D F7 83 28 48 05 98
0450: 98 05 CD F7 03 98 F5 38 08 28 08 05 CD F7 83 28
0460: F8 28 24 05 8D FF C9 03 F0 21 EE 05 04 D0 03
0470: EE 04 04 A5 05 CD 66 04 D8 DF A5 84 CD 45 04 D8
0480: D8 A8 98 28 59 05 A9 67 8D 82 1A 68 28 48 05 CD
0490: F7 03 98 F8 28 24 05 A9 67 8D 82 1A A5 7E 8D 8F
04A0: 84 A6 7F 88 88 C8 04 AD 57 24 AC 58 24 8D F8 03
04B0: 8C FA 03 A9 8E A8 8D 57 24 8C 58 24 8D AD 66 04
04C0: FF A8 20 14 13 EE DF 24 D8 03 CD 8F 04 D8 1E AD F5 03
04D0: CD C8 04 D8 1E AD 65 04 CD 8F 04 D8 1E AD F5 03
04E0: 8D 57 24 AD FA 03 8D 58 24 AD F4 03 F8 05 A8 12
04F0: 20 59 85 A8 35 20 59 85 A9 8D 68 98 68 28 00 85
0500: A9 02 2C 8D 18 F8 F8 8D 8D 18 A9 01 4D 0C 18 8D
0510: 8C 18 A9 FF AA 4D F6 1A 8E F5 1A AA 18 6D F6 03
0520: 8E F8 03 68 A9 55 28 4D 05 A8 8D 48 28 FD 04 CD
0530: F7 02 88 06 28 FD 04 38 8D 01 18 6A 88 D8 8E
0540: 48 4D F4 03 8D 04 03 8D 29 7F 68 A9 69 8D 88 1A
0550: AD 82 1A 49 82 8D 82 1A 68 89 68 05 C9 03 F8 07
0560: 20 34 13 C8 4C 59 85 68 8D 0A 4F 55 24 28 4F 46
0570: 28 4D 45 4D 4F 52 59 8A 8D 03 8D 0A 43 48 45 43
0580: 4B 53 55 4D 28 45 52 52 8F 03 8A 8D 03 8D 0A 4F
0590: 4B 8A 8D 03 8D 8A 4C 45 57 8A 8D 03

```

JUNIOR

Table 2. The hexdump for the translation program for the Junior with KB-9-BASIC.

Table 3

```

M
HEXDUMP: E000,E163
0 1 2 3 4 5 6 7 8 9 A B C D E F
E000: AD 11 23 AC 12 23 8D F8 03 8C F1 03 A9 02 A8 02
E010: 8D 11 23 8C 12 23 8D F8 03 8C F1 03 A9 02 A8 02
E020: A9 77 A8 02 05 04 04 05 A5 7E A4 7F 8D 49 02 8C
E030: 0A 02 8D A8 02 8C A1 02 A9 00 8D F4 03 8D F8 03
E040: A9 02 C9 0A F8 30 09 80 8D FF FF 4D F4 03 8D F4
E050: 03 EE 49 02 08 03 EE 4A 02 A5 85 CD 4A 02 08 14
E060: A5 84 CD 49 02 08 0F A9 FF 8D F8 03 A5 78 8D 49
E070: 02 A5 79 8D 4A 02 60 48 98 08 A9 8D 20 42 02 A5
E080: 03 20 42 02 AD F4 03 28 40 02 A9 7F 8D 0E 18 A9
E090: CF 8D 08 18 A9 01 8D 05 18 20 03 20 39 03 AD
E0A0: FF FF AD 08 4A 48 08 05 20 39 03 70 03 20 46 03
E0B0: 68 88 D8 F0 20 46 03 20 46 03 EE A9 02 D0 03 EE
E0C0: A1 02 AC A1 02 AE A0 02 C4 05 D8 0E E4 84 D8 0A
E0D0: A5 78 8D A8 02 A5 79 8D A1 02 CC 0A 02 D8 8D EC
E0E0: 49 02 D8 88 20 2B 03 0C 00 18 AD F2 03 AC F3 03
E0F0: 85 04 85 05 AD F8 03 AC F1 03 8D 52 2A 8C 53 2A
E100: AD F8 03 02 20 A5 79 85 7D 05 7D 05 7E A4 70 C8
E110: C8 C8 84 7A 84 7C 84 7E A9 00 A8 91 78 C8 91 78
E120: A8 2C 20 59 05 68 A8 68 4C 03 98 A2 70 A0 17 20
E130: 46 03 CA D8 FA 88 D8 F7 68 A9 9F 8D 96 18 A9 01
E140: 8D 07 18 4C 53 03 20 49 03 A9 CE 8D 96 18 A9 01
E150: 8D 07 18 AD 04 18 2C 8D 18 58 FB AD 04 18 2C 8D
E160: 18 58 FB 68

```

JUNIOR

```

M
HEXDUMP: E200,E398
0 1 2 3 4 5 6 7 8 9 A B C D E F
E200: A9 7F 8D 8E 18 A9 8D 8C 18 8D F4 03 A9 73 8D
E210: 82 FA A5 7E 8D 05 04 A6 7F 8E 84 05 8D 62 8E 66
E220: 04 A3 10 8D 06 03 A9 36 28 4D 05 28 FD 04 C9 22
E230: 98 FF C9 4E 88 8B EE F5 03 D8 F7 83 28 48 05 98
E240: C8 F6 03 D8 06 8A 38 19 8C 8D F7 83 28 48 05 98
E250: 98 05 CD F7 03 98 F5 38 08 28 08 05 CD F7 83 28
E260: F8 28 24 05 8D FF C9 03 F0 21 EE 05 04 D0 03
E270: EE 04 04 A5 05 CD 66 04 D8 DF A5 84 CD 45 04 D8
E280: D8 A8 98 28 59 05 A9 67 8D 82 1A 68 28 48 05 CD
E290: F7 03 98 F8 28 24 05 A9 67 8D 82 1A A5 7E 8D 8F
E2A0: 84 A6 7F 88 88 C8 04 AD 57 24 AC 58 24 8D F8 03
E2B0: 8C FA 03 A9 8E A8 8D 57 24 8C 58 24 8D AD 66 04
E2C0: FF A8 20 14 13 EE DF 24 D8 03 CD 8F 04 D8 1E AD F5 03
E2D0: CD C8 04 D8 1E AD 65 04 CD 8F 04 D8 1E AD F5 03
E2E0: 8D 57 24 AD FA 03 8D 58 24 AD F4 03 F8 05 A8 12
E2F0: 20 59 85 A8 35 20 59 85 A9 8D 68 98 68 28 00 85
E300: A9 02 2C 8D 18 F8 F8 8D 8D 18 A9 01 4D 0C 18 8D
E310: 8C 18 A9 FF AA 4D F6 1A 8E F5 1A AA 18 6D F6 03
E320: 8E F8 03 68 A9 55 28 4D 05 A8 8D 48 28 FD 04 CD
E330: F7 02 88 06 28 FD 04 38 8D 01 18 6A 88 D8 8E
E340: 48 4D F4 03 8D 04 03 8D 29 7F 68 A9 69 8D 88 1A
E350: AD 82 1A 49 82 8D 82 1A 68 89 68 05 C9 03 F8 07
E360: 20 34 13 C8 4C 59 85 68 8D 0A 4F 55 24 28 4F 46
E370: 28 4D 45 4D 4F 52 59 8A 8D 03 8D 0A 43 48 45 43
E380: 4B 53 55 4D 28 45 52 52 8F 03 8A 8D 03 8D 0A 4F
E390: 4B 8A 8D 03 8D 8A 4C 45 57 8A 8D 03

```

JUNIOR

Table 3. Hexdump for the translation program for the DOS Junior.

POKE 8256,0 : POKE 8257,2 : X =  
USR(X) : LIST  
(POKE 574,0 : POKE 575,224 : X =  
USR(X) : LIST)

The recorder is then set to record and started. Only then is the (carriage) return given. The whole program is then saved on the tape in Basiccode form. After the computer gives the 'OK' signal the recorder can be stopped. It is also possible to save only a part of the program on tape (for example, lines 1000-1090):

POKE 8256,0 : POKE 8257,2 : X =  
USR(X) : LIST 1000-1090  
(POKE 574,0 : POKE 575,224 : X =  
USR(X) : LIST 1000-1090).

Before the BASIC program is stored on tape, the computer 'translates' the program first into 'LIST' format and places that in a table which appears above the BASIC program in the RAM range. With large programs, the RAM range may not be big enough to store both of these so after the program is stored on tape the computer returns the 'NEW' message. This means that the original BASIC program is erased from the memory. As it is in Basiccode form on the tape anyway, it can also be read in again.

#### Details of the translation program

This next section is a description of the write and read routines (more details are given in the listing of table 1).

#### The write program

When this routine is called by means of X = USR(X), the OUTPUT vector (of the BASIC Junior) is changed for the start address of a machine code routine (TABLE in the write program). This routine stores an ASCII character from ACCU into RAM. After giving a LIST command (with POKE . . . : POKE . . . : X = USR(X) : LIST), the computer will list the program on the screen (or on the printer). Because the OUTPUT vector is changed (it normally points to the 'print character' routine), the TABLE routine is used to store the listing in RAM above the original BASIC program. The program is then stored in this table in LIST format.

After the BASIC Junior notes the end of the program and is therefore finished listing, it jumps via the JMP command at addresses 0003 . . . 0005 to SVECAS. This routine sets the whole table onto cassette with 1200 and 2400 Hz tones. When that is done the OUTPUT vector and the JMP at address 0003 are reset and the computer returns to BASIC.

#### The read program

After this program is called by X = USR(X), the Basiccode program is read from cassette and stored in the form of a table in RAM. Again the program is in LIST format. When the 'end of text' character and the checksum are read in, the whole program is located in this table, the INPUT vector (in the BASIC Junior) is changed for the start address of the LDIND routine, and the computer returns to normal BASIC.

The computer should now really wait for an input from the terminal (the INPUT vector normally points to the receive character routine), but because the INPUT vector points to the LDIND routine the characters are called one by one from the table by the BASIC Junior (and printed at the same time). This makes it seem as if a program is being typed in at high speed. The program thus read out of the table is then processed and stored in the normal way. Finally, the INPUT vector is reset and the computer returns with 'OK'. The user can then work with the program as usual.

### BASIC subroutines

Apart from the translation program there is also a need for some subroutines, written in Basicode-2 protocol. These are dealt with in depth in the descriptive article, 'Basicode-2', in this issue.

Three of these subroutines are not usable with the Junior/Elekterminal combination. These are routines 120, 200 and 250.

Subroutine 120 relates to the position of the cursor on the screen and subroutine 200 checks whether at a specific moment a key is pressed. Neither is possible because of the arrangement of the Elekterminal. Subroutine 250 just gives a bleep, but the Elekterminal is mute.

If the main BASIC program calls subroutine 120 or 250 nothing happens because in the Junior these subroutines consist of the 'RETURN' command. For subroutine 200 IN\$ is an empty string so that it seems as if no key is pressed at that moment.

The standard subroutines for the expanded Junior and the DOS Junior, both with the Elekterminal, are given in tables 4 and 5 respectively. Subroutines 350 and 360 should really refer to a printer but in our case they refer to the terminal.

The subroutines can be read in either before or after the Basicode program. That makes no difference as long as they are present when the program is RUN. If, for example, the Basicode program has already been read in, the subroutines can simply be added by reading them in using POKE . . . : POKE . . . : X = USR(X).

Two program sections can be added to form one program by reading them both in separately. The only prerequisite is that the two parts have no identical line numbers.

### Practical points

After reading in a Basicode programme it is only common sense to check it through carefully. Often there are some details that have a different meaning on your computer to what they meant to the computer on which the program was developed. This is a common reason for programs not to work.

Consider this case, for example: we have a Basicode program that draws a maze, and it contains the necessary PRINT statements.

If part of the maze is now drawn on the screen and the program wants to PRINT something in the middle of the maze, a carriage return and line feed are automatically generated after the print statement. With the Elekterminal a carriage return

Table 4

```

LIST
10 GOTO 1000
20 GOTO 1010
100 PRINT
101 POKE6745,200:PRINT CHR$(12);
102 POKE6745,3
103 RETURN
110 IF HO>=3 THEN RETURN
111 IF VE>15 THEN RETURN
112 POKE6745,200:PRINT CHR$(20);
113 POKE6745,3
114 PRINT
115 IF HO=0 GOTO 117
116 FOR CO=1 TO HO:PRINT CHR$(9);:NEXT
117 FOR CP=-1 TO 15-VE:PRINT CHR$(11);:NEXT
118 RETURN
120 RETURN
200 IN$="" :RETURN
210 OS=PEEK(8256):OT=PEEK(8257)
211 POKE8256,(10*16+14):POKE8257,(1*16+2)
212 O=USR(O)
213 POKE8256,OS:POKE8257,OT
214 OX=PEEK(8754) AND 127)
215 IN$=CHR$(OX)
216 RETURN
250 RETURN
260 RV=RND(1):RETURN
270 PH=PRE(0):RETURN
300 IF SR<.01 AND SR>-.01 THEN SR=0
301 SR=USR(1) THEN SR=STR$(SR):RETURN
302 SR=MID$(STR$(SR),2):RETURN
310 OS=ABS(SR)+.5*10^(-CN:OI=INT(OS):OD=OS-OI+1
311 SR$=""
312 IF OD=1E9 THEN 321
313 IF CN=0 THEN ODS="" :GOTO 317
314 IF CO=1 THEN ODS="" :GOTO 316
315 ODS=MID$(STR$(OD),3,CN+1)
316 IF LEN(ODS)<CN+1 THEN ODS=ODS+"0":GOTO 316
317 SR$=MID$(STR$(OI),2)+ODS
318 IF SR<0 AND VAL$(SR)<0 THEN SR$="--"+SR$
319 IF LEN$(SR)<CT THEN SR$="" +SR$:GOTO 319
320 IF LEN$(SR)>CT THEN SR$="" :GOTO 319
321 IF LEN$(SR)<CT THEN SR$=SR$+"*":GOTO 321
322 RETURN
350 PRINT SR$:RETURN
360 PRINT :RETURN
OK

```

Table 5

```

LIST
10 GOTO 1000
20 GOTO 1010
100 PRINT
101 POKE64089,200:PRINT CHR$(12);
102 POKE64089,3
103 RETURN
110 IF HO>=3 THEN RETURN
111 IF VE>15 THEN RETURN
112 POKE64089,200:PRINT CHR$(20);
113 POKE64089,3
114 PRINT
115 IF HO=0 GOTO 117
116 FOR CO=1 TO HO:PRINT CHR$(9);:NEXT
117 FOR CP=-1 TO 15-VE:PRINT CHR$(11);:NEXT
118 RETURN
120 RETURN
200 IN$="" :RETURN
210 OS=PEEK(574):OT=PEEK(575)
211 POKE574,(1*16+11):POKE575,(15*16+14)
212 O=USR(O)
213 POKE574,OS:POKE575,OT
214 OX=PEEK(9859)
215 IN$=CHR$(OX)
216 RETURN
250 RETURN
260 RV=RND(1):RETURN
270 PH=PRE(0):RETURN
300 IF SR<.01 AND SR>-.01 THEN SR=0
301 SR=USR(1) THEN SR=STR$(SR):RETURN
302 SR=MID$(STR$(SR),2):RETURN
310 OS=ABS(SR)+.5*10^(-CN:OI=INT(OS):OD=OS-OI+1
311 SR$=""
312 IF OD=1E9 THEN 321
313 IF CN=0 THEN ODS="" :GOTO 317
314 IF CO=1 THEN ODS="" :GOTO 316
315 ODS=MID$(STR$(OD),3,CN+1)
316 IF LEN(ODS)<CN+1 THEN ODS=ODS+"0":GOTO 316
317 SR$=MID$(STR$(OI),2)+ODS
318 IF SR<0 AND VAL$(SR)<0 THEN SR$="--"+SR$
319 IF LEN$(SR)<CT THEN SR$="" +SR$:GOTO 319
320 IF LEN$(SR)>CT THEN SR$="" :GOTO 319
321 IF LEN$(SR)<CT THEN SR$=SR$+"*":GOTO 321
322 RETURN
350 PRINT SR$:RETURN
360 PRINT :RETURN
OK

```

Basicode-2 interface for the Junior Computer  
elektor october 1983

Table 4. The standard subroutines for the expanded Junior with KB-9-BASIC.

Table 5. The standard subroutines for the DOS Junior.

Basicode-2 interface for the Junior Computer  
elektor october 1983

Figure 1. The circuit diagram for the interface circuit that must be connected between a cassette recorder and the Junior Computer.

Parts list

Resistors:  
R1 = 4k7  
R2,R4,R7 = 1 k  
R3 = 10 k  
R5 = 1 M  
R6 = 56 k  
P1 = 25 k preset

Capacitors:  
C1 = 220 n  
C2 = 10 μ/10 V  
C3 = 56 n  
C4 = 100 n

Semiconductors:  
IC1 = 3140

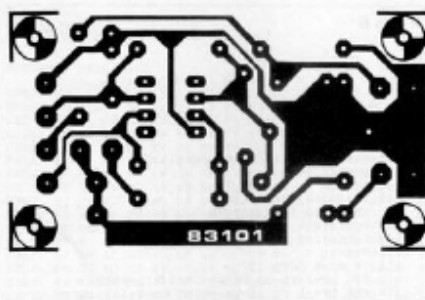
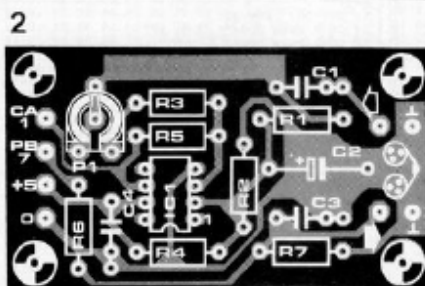
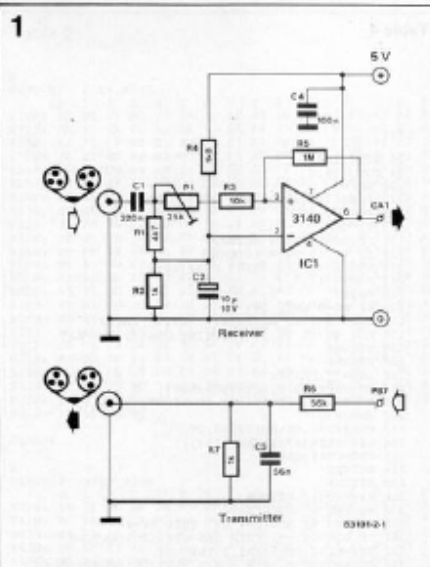
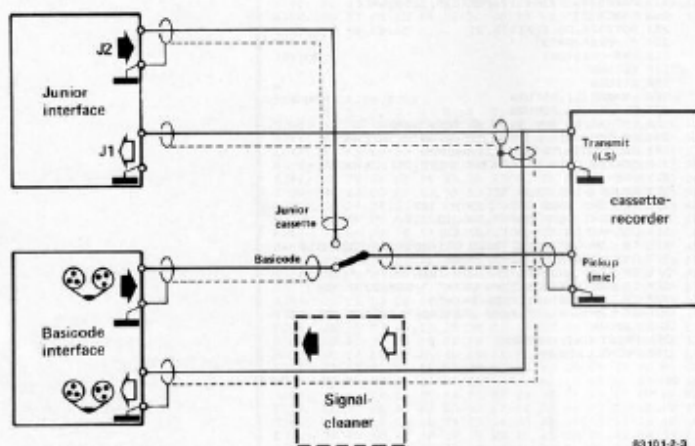


Figure 2. The printed circuit board layout for the interface circuit.

Figure 3. This shows the wiring layout that must be used if both the normal Junior cassette interface and the Basicode interface are to be used together.

3



means that everything after the print statement on this line is erased. In this example the program can easily be adapted by following the PRINTs in question with a ;. The CR and LF are not produced then and the program runs properly.

A program could, of course, also call a subroutine that the Junior/Elektterminal does not recognize (120, 200 and 250). Subroutines 200 and 250 are no real problem and can easily be avoided, but is sometimes more difficult to do without routine 120. If sub 120 is used, for example, in a game to define the position of the cursor on the screen, it can be very difficult to adapt the program. Subroutine 120 is also quite often used to define the screen size. This can also be done by leaving out the appropriate lines and stating on the free lines how large the screen is (16 lines of 64 characters on the Elektterminal). In the case where, for example, the screen format is defined for a section of a program, and after leaving this section, variables VV and HH must contain the height and width of the screen. In our case this program section is simply changed by VV = 15 : HH = 63 (remember that the first position has always number zero).

A final note about the @ sign in KB-9-BASIC. If the computer sees this sign the whole line is erased and CR and LF are given.

The hardware

The hardware for the Basicode interface consists of a small adapter circuit which is connected between the cassette recorder and the Junior Computer. The circuit diagram is shown in figure 1. It consists of a transmitter and a receiver section. The receiver contains only one IC (3140) which is connected as a schmitt trigger/level adapter. Using P1, the trigger level can be set between certain limits, but normally the circuit works correctly if the pot is roughly in mid position. The transmitter section simply reduces the output signal from the Junior and filters out the higher harmonics from the signal.

The printed circuit board for the interface (figure 2) is designed so that two phono plugs (for input and output) can be soldered directly onto the board using some wire links. Points CA1 and PB7 are connected to the corresponding points on the VIA connector on the interface board.

If the normal Junior cassette interface and the Basicode interface are to be connected at the same time (the former is needed to read in machine code programs), care must be taken when wiring the interfaces. The wiring diagram for connecting both interfaces is given in figure 3. Any deviation from this layout is likely to result in earth losses occurring and the possibility of oscillation is greater. This same diagram also shows a block called signal cleaner! This circuit, which is also described in this issue, is only needed if the signal from the recorder (or radio) is of very poor quality. It is easy to try without this interface first and if this does not work, the circuit could always be added.

Two of our recent projects, the VDU card and the Basicode-2 interface, can both be used individually with the Junior Computer. However, there are bound to be some JC users who are interested in using these two 'extra's' together. The program given here was designed to do just this and thus provide the best of both worlds. Two versions of the software have been developed, for the extended Junior and for the DOS Junior.

# Basicode-2 for Junior plus VDU card

Junior  
Computer +  
VDU card +  
Basicode-2 =  
the best of all  
worlds

The description of Basicode-2 and the adaption to use Basicode-2 with the Junior Computer in particular have already been dealt with in Elektor no. 102, October 1983. All details of the hardware and software needed are given there so we will not go into that again here. The only change needed to use Basicode-2 with the Junior Computer and VDU card is to modify the standard subroutines. Two tables of these subroutines are given in this article: one for the extended Junior with VDU card and the other for DOS Junior with VDU card.

## A few changes

Some modifications or additions to the 'old' subroutines are needed. Subroutine 110 is changed. We have written a small machine-code program to speed up the positioning of the cursor (to HQ, VE). Whenever a jump to line 20 is made in a Basicode-2 program (as always happens), then a piece of machine-code is first written into RAM. If the program then comes to subroutine 110 at any stage this machine-code program is called and the cursor is brought very quickly to the position defined by HQ and VE. Subroutine 120, requesting the position of the cursor on the screen, is possible with this combination even though it did not work with the Junior/Elektterminal combination.

The only routine that is still unworkable is subroutine 200. The Junior simply cannot determine if a key is pressed at a particular moment. A GOSUB 200 in a program must therefore be changed. Actually there are two routines that do not work, the second being subroutine 250. However, the bleep that should be generated by a GOSUB 250 can hardly be considered essential for the correct operation of a program.

One further important note. If the Basicode-2 translation program is used with the DOS Junior, great care should be exercised when using the DISK! command. If, for instance, a BASIC program is loaded from the floppy with the command DISK!"LO..." and this program is then to be saved on tape in Basicode-2 format, the 'save' may not work because DISK!"..." causes page zero to be 'swapped'. The result of this is that the pointers needed in the Basicode-2 translation program are not longer correct. There is a very simple solution for this. After removing anything from (or storing anything on) the floppy disk the number 1 is typed and a (CARRIAGE) RETURN given. A dummy line has then been included and the pointers are again correct. Everything will then operate correctly provided there is nothing on line 1, otherwise a different (blank) line number must be used.

Table 1. These are standard subroutines for the extended Junior with the VDU card.

Table 2. This is a list of the subroutines to use for the DOS Junior with the VDU card.

```

1 LIST
20 GOTO 1000
21 DATA 20,125,19,179,102,2,141,07,26,30,134,10
22 DATA 16,112,125,10,4,20,20,12,28,144,240,76
23 FOR D=1 TO 24:POKE D,POKEC(27)+D,0:ND=4:G
24 GOTO 1000
25 PRINT CHR$(27)+PRINT CHR$(10)
26 RETURN
27 IF HD=27 THEN RETURN
28 IF HD=28 THEN RETURN
29 POKE27,HD:POKE28,LV
30 POKE29,HD:POKE30,LV
31 POKE31,HD:POKE32,LV
32 PRINT CHR$(10)
33 GOTO 1000
34 POKE33,OS:POKE34,OT
35 RETURN
36 POKE35,OT:VIEWPE(167)
37 RETURN
38 POKE38,OT:VIEWPE(167)
39 RETURN
40 POKE40,OT:VIEWPE(167)
41 POKE41,OT:VIEWPE(167)
42 POKE42,OT:VIEWPE(167)
43 POKE43,OT:VIEWPE(167)
44 POKE44,OT:VIEWPE(167)
45 POKE45,OT:VIEWPE(167)
46 POKE46,OT:VIEWPE(167)
47 POKE47,OT:VIEWPE(167)
48 POKE48,OT:VIEWPE(167)
49 POKE49,OT:VIEWPE(167)
50 POKE50,OT:VIEWPE(167)
51 POKE51,OT:VIEWPE(167)
52 POKE52,OT:VIEWPE(167)
53 POKE53,OT:VIEWPE(167)
54 POKE54,OT:VIEWPE(167)
55 POKE55,OT:VIEWPE(167)
56 POKE56,OT:VIEWPE(167)
57 POKE57,OT:VIEWPE(167)
58 POKE58,OT:VIEWPE(167)
59 POKE59,OT:VIEWPE(167)
60 POKE60,OT:VIEWPE(167)
61 POKE61,OT:VIEWPE(167)
62 POKE62,OT:VIEWPE(167)
63 POKE63,OT:VIEWPE(167)
64 POKE64,OT:VIEWPE(167)
65 POKE65,OT:VIEWPE(167)
66 POKE66,OT:VIEWPE(167)
67 POKE67,OT:VIEWPE(167)
68 POKE68,OT:VIEWPE(167)
69 POKE69,OT:VIEWPE(167)
70 POKE70,OT:VIEWPE(167)
71 POKE71,OT:VIEWPE(167)
72 POKE72,OT:VIEWPE(167)
73 POKE73,OT:VIEWPE(167)
74 POKE74,OT:VIEWPE(167)
75 POKE75,OT:VIEWPE(167)
76 POKE76,OT:VIEWPE(167)
77 POKE77,OT:VIEWPE(167)
78 POKE78,OT:VIEWPE(167)
79 POKE79,OT:VIEWPE(167)
80 POKE80,OT:VIEWPE(167)
81 POKE81,OT:VIEWPE(167)
82 POKE82,OT:VIEWPE(167)
83 POKE83,OT:VIEWPE(167)
84 POKE84,OT:VIEWPE(167)
85 POKE85,OT:VIEWPE(167)
86 POKE86,OT:VIEWPE(167)
87 POKE87,OT:VIEWPE(167)
88 POKE88,OT:VIEWPE(167)
89 POKE89,OT:VIEWPE(167)
90 POKE90,OT:VIEWPE(167)
91 POKE91,OT:VIEWPE(167)
92 POKE92,OT:VIEWPE(167)
93 POKE93,OT:VIEWPE(167)
94 POKE94,OT:VIEWPE(167)
95 POKE95,OT:VIEWPE(167)
96 POKE96,OT:VIEWPE(167)
97 POKE97,OT:VIEWPE(167)
98 POKE98,OT:VIEWPE(167)
99 POKE99,OT:VIEWPE(167)
100 POKE100,OT:VIEWPE(167)
101 POKE101,OT:VIEWPE(167)
102 POKE102,OT:VIEWPE(167)
103 POKE103,OT:VIEWPE(167)
104 POKE104,OT:VIEWPE(167)
105 POKE105,OT:VIEWPE(167)
106 POKE106,OT:VIEWPE(167)
107 POKE107,OT:VIEWPE(167)
108 POKE108,OT:VIEWPE(167)
109 POKE109,OT:VIEWPE(167)
110 POKE110,OT:VIEWPE(167)
111 POKE111,OT:VIEWPE(167)
112 POKE112,OT:VIEWPE(167)
113 POKE113,OT:VIEWPE(167)
114 POKE114,OT:VIEWPE(167)
115 POKE115,OT:VIEWPE(167)
116 POKE116,OT:VIEWPE(167)
117 POKE117,OT:VIEWPE(167)
118 POKE118,OT:VIEWPE(167)
119 POKE119,OT:VIEWPE(167)
120 POKE120,OT:VIEWPE(167)
121 POKE121,OT:VIEWPE(167)
122 POKE122,OT:VIEWPE(167)
123 POKE123,OT:VIEWPE(167)
124 POKE124,OT:VIEWPE(167)
125 POKE125,OT:VIEWPE(167)
126 POKE126,OT:VIEWPE(167)
127 POKE127,OT:VIEWPE(167)
128 POKE128,OT:VIEWPE(167)
129 POKE129,OT:VIEWPE(167)
130 POKE130,OT:VIEWPE(167)
131 POKE131,OT:VIEWPE(167)
132 POKE132,OT:VIEWPE(167)
133 POKE133,OT:VIEWPE(167)
134 POKE134,OT:VIEWPE(167)
135 POKE135,OT:VIEWPE(167)
136 POKE136,OT:VIEWPE(167)
137 POKE137,OT:VIEWPE(167)
138 POKE138,OT:VIEWPE(167)
139 POKE139,OT:VIEWPE(167)
140 POKE140,OT:VIEWPE(167)
141 POKE141,OT:VIEWPE(167)
142 POKE142,OT:VIEWPE(167)
143 POKE143,OT:VIEWPE(167)
144 POKE144,OT:VIEWPE(167)
145 POKE145,OT:VIEWPE(167)
146 POKE146,OT:VIEWPE(167)
147 POKE147,OT:VIEWPE(167)
148 POKE148,OT:VIEWPE(167)
149 POKE149,OT:VIEWPE(167)
150 POKE150,OT:VIEWPE(167)
151 POKE151,OT:VIEWPE(167)
152 POKE152,OT:VIEWPE(167)
153 POKE153,OT:VIEWPE(167)
154 POKE154,OT:VIEWPE(167)
155 POKE155,OT:VIEWPE(167)
156 POKE156,OT:VIEWPE(167)
157 POKE157,OT:VIEWPE(167)
158 POKE158,OT:VIEWPE(167)
159 POKE159,OT:VIEWPE(167)
160 POKE160,OT:VIEWPE(167)
161 POKE161,OT:VIEWPE(167)
162 POKE162,OT:VIEWPE(167)
163 POKE163,OT:VIEWPE(167)
164 POKE164,OT:VIEWPE(167)
165 POKE165,OT:VIEWPE(167)
166 POKE166,OT:VIEWPE(167)
167 POKE167,OT:VIEWPE(167)
168 POKE168,OT:VIEWPE(167)
169 POKE169,OT:VIEWPE(167)
170 POKE170,OT:VIEWPE(167)
171 POKE171,OT:VIEWPE(167)
172 POKE172,OT:VIEWPE(167)
173 POKE173,OT:VIEWPE(167)
174 POKE174,OT:VIEWPE(167)
175 POKE175,OT:VIEWPE(167)
176 POKE176,OT:VIEWPE(167)
177 POKE177,OT:VIEWPE(167)
178 POKE178,OT:VIEWPE(167)
179 POKE179,OT:VIEWPE(167)
180 POKE180,OT:VIEWPE(167)
181 POKE181,OT:VIEWPE(167)
182 POKE182,OT:VIEWPE(167)
183 POKE183,OT:VIEWPE(167)
184 POKE184,OT:VIEWPE(167)
185 POKE185,OT:VIEWPE(167)
186 POKE186,OT:VIEWPE(167)
187 POKE187,OT:VIEWPE(167)
188 POKE188,OT:VIEWPE(167)
189 POKE189,OT:VIEWPE(167)
190 POKE190,OT:VIEWPE(167)
191 POKE191,OT:VIEWPE(167)
192 POKE192,OT:VIEWPE(167)
193 POKE193,OT:VIEWPE(167)
194 POKE194,OT:VIEWPE(167)
195 POKE195,OT:VIEWPE(167)
196 POKE196,OT:VIEWPE(167)
197 POKE197,OT:VIEWPE(167)
198 POKE198,OT:VIEWPE(167)
199 POKE199,OT:VIEWPE(167)
200 POKE200,OT:VIEWPE(167)
201 POKE201,OT:VIEWPE(167)
202 POKE202,OT:VIEWPE(167)
203 POKE203,OT:VIEWPE(167)
204 POKE204,OT:VIEWPE(167)
205 POKE205,OT:VIEWPE(167)
206 POKE206,OT:VIEWPE(167)
207 POKE207,OT:VIEWPE(167)
208 POKE208,OT:VIEWPE(167)
209 POKE209,OT:VIEWPE(167)
210 POKE210,OT:VIEWPE(167)
211 POKE211,OT:VIEWPE(167)
212 POKE212,OT:VIEWPE(167)
213 POKE213,OT:VIEWPE(167)
214 POKE214,OT:VIEWPE(167)
215 POKE215,OT:VIEWPE(167)
216 POKE216,OT:VIEWPE(167)
217 POKE217,OT:VIEWPE(167)
218 POKE218,OT:VIEWPE(167)
219 POKE219,OT:VIEWPE(167)
220 POKE220,OT:VIEWPE(167)
221 POKE221,OT:VIEWPE(167)
222 POKE222,OT:VIEWPE(167)
223 POKE223,OT:VIEWPE(167)
224 POKE224,OT:VIEWPE(167)
225 POKE225,OT:VIEWPE(167)
226 POKE226,OT:VIEWPE(167)
227 POKE227,OT:VIEWPE(167)
228 POKE228,OT:VIEWPE(167)
229 POKE229,OT:VIEWPE(167)
230 POKE230,OT:VIEWPE(167)
231 POKE231,OT:VIEWPE(167)
232 POKE232,OT:VIEWPE(167)
233 POKE233,OT:VIEWPE(167)
234 POKE234,OT:VIEWPE(167)
235 POKE235,OT:VIEWPE(167)
236 POKE236,OT:VIEWPE(167)
237 POKE237,OT:VIEWPE(167)
238 POKE238,OT:VIEWPE(167)
239 POKE239,OT:VIEWPE(167)
240 POKE240,OT:VIEWPE(167)
241 POKE241,OT:VIEWPE(167)
242 POKE242,OT:VIEWPE(167)
243 POKE243,OT:VIEWPE(167)
244 POKE244,OT:VIEWPE(167)
245 POKE245,OT:VIEWPE(167)
246 POKE246,OT:VIEWPE(167)
247 POKE247,OT:VIEWPE(167)
248 POKE248,OT:VIEWPE(167)
249 POKE249,OT:VIEWPE(167)
250 POKE250,OT:VIEWPE(167)
251 POKE251,OT:VIEWPE(167)
252 POKE252,OT:VIEWPE(167)
253 POKE253,OT:VIEWPE(167)
254 POKE254,OT:VIEWPE(167)
255 POKE255,OT:VIEWPE(167)
256 POKE256,OT:VIEWPE(167)
257 POKE257,OT:VIEWPE(167)
258 POKE258,OT:VIEWPE(167)
259 POKE259,OT:VIEWPE(167)
260 POKE260,OT:VIEWPE(167)
261 POKE261,OT:VIEWPE(167)
262 POKE262,OT:VIEWPE(167)
263 POKE263,OT:VIEWPE(167)
264 POKE264,OT:VIEWPE(167)
265 POKE265,OT:VIEWPE(167)
266 POKE266,OT:VIEWPE(167)
267 POKE267,OT:VIEWPE(167)
268 POKE268,OT:VIEWPE(167)
269 POKE269,OT:VIEWPE(167)
270 POKE270,OT:VIEWPE(167)
271 POKE271,OT:VIEWPE(167)
272 POKE272,OT:VIEWPE(167)
273 POKE273,OT:VIEWPE(167)
274 POKE274,OT:VIEWPE(167)
275 POKE275,OT:VIEWPE(167)
276 POKE276,OT:VIEWPE(167)
277 POKE277,OT:VIEWPE(167)
278 POKE278,OT:VIEWPE(167)
279 POKE279,OT:VIEWPE(167)
280 POKE280,OT:VIEWPE(167)
281 POKE281,OT:VIEWPE(167)
282 POKE282,OT:VIEWPE(167)
283 POKE283,OT:VIEWPE(167)
284 POKE284,OT:VIEWPE(167)
285 POKE285,OT:VIEWPE(167)
286 POKE286,OT:VIEWPE(167)
287 POKE287,OT:VIEWPE(167)
288 POKE288,OT:VIEWPE(167)
289 POKE289,OT:VIEWPE(167)
290 POKE290,OT:VIEWPE(167)
291 POKE291,OT:VIEWPE(167)
292 POKE292,OT:VIEWPE(167)
293 POKE293,OT:VIEWPE(167)
294 POKE294,OT:VIEWPE(167)
295 POKE295,OT:VIEWPE(167)
296 POKE296,OT:VIEWPE(167)
297 POKE297,OT:VIEWPE(167)
298 POKE298,OT:VIEWPE(167)
299 POKE299,OT:VIEWPE(167)
300 POKE300,OT:VIEWPE(167)
301 POKE301,OT:VIEWPE(167)
302 POKE302,OT:VIEWPE(167)
303 POKE303,OT:VIEWPE(167)
304 POKE304,OT:VIEWPE(167)
305 POKE305,OT:VIEWPE(167)
306 POKE306,OT:VIEWPE(167)
307 POKE307,OT:VIEWPE(167)
308 POKE308,OT:VIEWPE(167)
309 POKE309,OT:VIEWPE(167)
310 POKE310,OT:VIEWPE(167)
311 POKE311,OT:VIEWPE(167)
312 POKE312,OT:VIEWPE(167)
313 POKE313,OT:VIEWPE(167)
314 POKE314,OT:VIEWPE(167)
315 POKE315,OT:VIEWPE(167)
316 POKE316,OT:VIEWPE(167)
317 POKE317,OT:VIEWPE(167)
318 POKE318,OT:VIEWPE(167)
319 POKE319,OT:VIEWPE(167)
320 POKE320,OT:VIEWPE(167)
321 POKE321,OT:VIEWPE(167)
322 POKE322,OT:VIEWPE(167)
323 POKE323,OT:VIEWPE(167)
324 POKE324,OT:VIEWPE(167)
325 POKE325,OT:VIEWPE(167)
326 POKE326,OT:VIEWPE(167)
327 POKE327,OT:VIEWPE(167)
328 POKE328,OT:VIEWPE(167)
329 POKE329,OT:VIEWPE(167)
330 POKE330,OT:VIEWPE(167)
331 POKE331,OT:VIEWPE(167)
332 POKE332,OT:VIEWPE(167)
333 POKE333,OT:VIEWPE(167)
334 POKE334,OT:VIEWPE(167)
335 POKE335,OT:VIEWPE(167)
336 POKE336,OT:VIEWPE(167)
337 POKE337,OT:VIEWPE(167)
338 POKE338,OT:VIEWPE(167)
339 POKE339,OT:VIEWPE(167)
340 POKE340,OT:VIEWPE(167)
341 POKE341,OT:VIEWPE(167)
342 POKE342,OT:VIEWPE(167)
343 POKE343,OT:VIEWPE(167)
344 POKE344,OT:VIEWPE(167)
345 POKE345,OT:VIEWPE(167)
346 POKE346,OT:VIEWPE(167)
347 POKE347,OT:VIEWPE(167)
348 POKE348,OT:VIEWPE(167)
349 POKE349,OT:VIEWPE(167)
350 POKE350,OT:VIEWPE(167)
351 POKE351,OT:VIEWPE(167)
352 POKE352,OT:VIEWPE(167)
353 POKE353,OT:VIEWPE(167)
354 POKE354,OT:VIEWPE(167)
355 POKE355,OT:VIEWPE(167)
356 POKE356,OT:VIEWPE(167)
357 POKE357,OT:VIEWPE(167)
358 POKE358,OT:VIEWPE(167)
359 POKE359,OT:VIEWPE(167)
360 POKE360,OT:VIEWPE(167)
361 POKE361,OT:VIEWPE(167)
362 POKE362,OT:VIEWPE(167)
363 POKE363,OT:VIEWPE(167)
364 POKE364,OT:VIEWPE(167)
365 POKE365,OT:VIEWPE(167)
366 POKE366,OT:VIEWPE(167)
367 POKE367,OT:VIEWPE(167)
368 POKE368,OT:VIEWPE(167)
369 POKE369,OT:VIEWPE(167)
370 POKE370,OT:VIEWPE(167)
371 POKE371,OT:VIEWPE(167)
372 POKE372,OT:VIEWPE(167)
373 POKE373,OT:VIEWPE(167)
374 POKE374,OT:VIEWPE(167)
375 POKE375,OT:VIEWPE(167)
376 POKE376,OT:VIEWPE(167)
377 POKE377,OT:VIEWPE(167)
378 POKE378,OT:VIEWPE(167)
379 POKE379,OT:VIEWPE(167)
380 POKE380,OT:VIEWPE(167)
381 POKE381,OT:VIEWPE(167)
382 POKE382,OT:VIEWPE(167)
383 POKE383,OT:VIEWPE(167)
384 POKE384,OT:VIEWPE(167)
385 POKE385,OT:VIEWPE(167)
386 POKE386,OT:VIEWPE(167)
387 POKE387,OT:VIEWPE(167)
388 POKE388,OT:VIEWPE(167)
389 POKE389,OT:VIEWPE(167)
390 POKE390,OT:VIEWPE(167)
391 POKE391,OT:VIEWPE(167)
392 POKE392,OT:VIEWPE(167)
393 POKE393,OT:VIEWPE(167)
394 POKE394,OT:VIEWPE(167)
395 POKE395,OT:VIEWPE(167)
396 POKE396,OT:VIEWPE(167)
397 POKE397,OT:VIEWPE(167)
398 POKE398,OT:VIEWPE(167)
399 POKE399,OT:VIEWPE(167)
400 POKE400,OT:VIEWPE(167)
401 POKE401,OT:VIEWPE(167)
402 POKE402,OT:VIEWPE(167)
403 POKE403,OT:VIEWPE(167)
404 POKE404,OT:VIEWPE(167)
405 POKE405,OT:VIEWPE(167)
406 POKE406,OT:VIEWPE(167)
407 POKE407,OT:VIEWPE(167)
408 POKE408,OT:VIEWPE(167)
409 POKE409,OT:VIEWPE(167)
410 POKE410,OT:VIEWPE(167)
411 POKE411,OT:VIEWPE(167)
412 POKE412,OT:VIEWPE(167)
413 POKE413,OT:VIEWPE(167)
414 POKE414,OT:VIEWPE(167)
415 POKE415,OT:VIEWPE(167)
416 POKE416,OT:VIEWPE(167)
417 POKE417,OT:VIEWPE(167)
418 POKE418,OT:VIEWPE(167)
419 POKE419,OT:VIEWPE(167)
420 POKE420,OT:VIEWPE(167)
421 POKE421,OT:VIEWPE(167)
422 POKE422,OT:VIEWPE(167)
423 POKE423,OT:VIEWPE(167)
424 POKE424,OT:VIEWPE(167)
425 POKE425,OT:VIEWPE(167)
426 POKE426,OT:VIEWPE(167)
427 POKE427,OT:VIEWPE(167)
428 POKE428,OT:VIEWPE(167)
429 POKE429,OT:VIEWPE(167)
430 POKE430,OT:VIEWPE(167)
431 POKE431,OT:VIEWPE(167)
432 POKE432,OT:VIEWPE(167)
433 POKE433,OT:VIEWPE(167)
434 POKE434,OT:VIEWPE(167)
435 POKE435,OT:VIEWPE(167)
436 POKE436,OT:VIEWPE(167)
437 POKE437,OT:VIEWPE(167)
438 POKE438,OT:VIEWPE(167)
439 POKE439,OT:VIEWPE(167)
440 POKE440,OT:VIEWPE(167)
441 POKE441,OT:VIEWPE(167)
442 POKE442,OT:VIEWPE(167)
443 POKE443,OT:VIEWPE(167)
444 POKE444,OT:VIEWPE(167)
445 POKE445,OT:VIEWPE(167)
446 POKE446,OT:VIEWPE(167)
447 POKE447,OT:VIEWPE(167)
448 POKE448,OT:VIEWPE(167)
449 POKE449,OT:VIEWPE(167)
450 POKE450,OT:VIEWPE(167)
451 POKE451,OT:VIEWPE(167)
452 POKE452,OT:VIEWPE(167)
453 POKE453,OT:VIEWPE(167)
454 POKE454,OT:VIEWPE(167)
455 POKE455,OT:VIEWPE(167)
456 POKE456,OT:VIEWPE(167)
457 POKE457,OT:VIEWPE(167)
458 POKE458,OT:VIEWPE(167)
459 POKE459,OT:VIEWPE(167)
460 POKE460,OT:VIEWPE(167)
461 POKE461,OT:VIEWPE(167)
462 POKE462,OT:VIEWPE(167)
463 POKE463,OT:VIEWPE(167)
464 POKE464,OT:VIEWPE(167)
465 POKE465,OT:VIEWPE(167)
466 POKE466,OT:VIEWPE(167)
467 POKE467,OT:VIEWPE(167)
468 POKE468,OT:VIEWPE(167)
469 POKE469,OT:VIEWPE(167)
470 POKE470,OT:VIEWPE(167)
471 POKE471,OT:VIEWPE(167)
472 POKE472,OT:VIEWPE(167)
473 POKE473,OT:VIEWPE(167)
474 POKE474,OT:VIEWPE(167)
475 POKE475,OT:VIEWPE(167)
476 POKE476,OT:VIEWPE(167)
477 POKE477,OT:VIEWPE(167)
478 POKE478,OT:VIEWPE(167)
479 POKE479,OT:VIEWPE(167)
480 POKE480,OT:VIEWPE(167)
481 POKE481,OT:VIEWPE(167)
482 POKE482,OT:VIEWPE(167)
483 POKE483,OT:VIEWPE(167)
484 POKE484,OT:VIEWPE(167)
485 POKE485,OT:VIEWPE(167)
486 POKE486,OT:VIEWPE(167)
487 POKE487,OT:VIEWPE(167)
488 POKE488,OT:VIEWPE(167)
489 POKE489,OT:VIEWPE(167)
490 POKE490,OT:VIEWPE(167)
491 POKE491,OT:VIEWPE(167)
492 POKE492,OT:VIEWPE(167)
493 POKE493,OT:VIEWPE(167)
494 POKE494,OT:VIEWPE(167)
495 POKE495,OT:VIEWPE(167)
496 POKE496,OT:VIEWPE(167)
497 POKE497,OT:VIEWPE(167)
498 POKE498,OT:VIEWPE(167)
499 POKE499,OT:VIEWPE(167)
500 POKE500,OT:VIEWPE(167)
501 POKE501,OT:VIEWPE(167)
502 POKE502,OT:VIEWPE(167)
503 POKE503,OT:VIEWPE(167)
504 POKE504,OT:VIEWPE(167)
505 POKE505,OT:VIEWPE(167)
506 POKE506,OT:VIEWPE(167)
507 POKE507,OT:VIEWPE(167)
508 POKE508,OT:VIEWPE(167)
509 POKE509,OT:VIEWPE(167)
510 POKE510,OT:VIEWPE(167)
511 POKE511,OT:VIEWPE(167)
512 POKE512,OT:VIEWPE(167)
513 POKE513,OT:VIEWPE(167)
514 POKE514,OT:VIEWPE(167)
515 POKE515,OT:VIEWPE(167)
516 POKE516,OT:VIEWPE(167)
517 POKE517,OT:VIEWPE(167)
518 POKE518,OT:VIEWPE(167)
519 POKE519,OT:VIEWPE(167)
520 POKE520,OT:VIEWPE(167)
521 POKE521,OT:VIEWPE(167)
522 POKE522,OT:VIEWPE(167)
523 POKE523,OT:VIEWPE(167)
524 POKE524,OT:VIEWPE(167)
525 POKE525,OT:VIEWPE(167)
526 POKE526,OT:VIEWPE(167)
527 POKE527,OT:VIEWPE(167)
528 POKE528,OT:VIEWPE(167)
529 POKE529,OT:VIEWPE(167)
530 POKE530,OT:VIEWPE(167)
531 POKE531,OT:VIEWPE(167)
532 POKE532,OT:VIEWPE(167)
533 POKE533,OT:VIEWPE(167)
534 POKE534,OT:VIEWPE(167)
535 POKE535,OT:VIEWPE(167)
536 POKE536,OT:VIEWPE(167)
537 POKE537,OT:VIEWPE(167)
538 POKE538,OT:VIEWPE(167)
539 POKE539,OT:VIEWPE(167)
540 POKE540,OT:VIEWPE(167)
541 POKE541,OT:VIEWPE(167)
542 POKE542,OT:VIEWPE(167)
543 POKE543,OT:VIEWPE(167)
544 POKE544,OT:VIEWPE(167)
545 POKE545,OT:VIEWPE(167)
546 POKE546,OT:VIEWPE(167)
547 POKE547,OT:VIEWPE(167)
548 POKE548,OT:VIEWPE(167)
549 POKE549,OT:VIEWPE(167)
550 POKE550,OT:VIEWPE(167)
551 POKE551,OT:VIEWPE(167)
552 POKE552,OT:VIEWPE(167)
553 POKE553,OT:VIEWPE(167)
554 POKE554,OT:VIEWPE(167)
555 POKE555,OT:VIEWPE(167)
556 POKE556,OT:VIEWPE(167)
557 POKE557,OT:VIEWPE(167)
558 POKE558,OT:VIEWPE(167)
559 POKE559,OT:VIEWPE(167)
560 POKE560,OT:VIEWPE(167)
561 POKE561,OT:VIEWPE(167)
562 POKE562,OT:VIEWPE(167)
563 POKE563,OT:VIEWPE(167)
564 POKE564,OT:VIEWPE(167)
565 POKE565,OT:VIEWPE(167)
566 POKE566,OT:VIEWPE(167)
567 POKE567,OT:VIEWPE(167)
568 POKE568,OT:VIEWPE(167)
569 POKE569,OT:VIEWPE(167)
570 POKE570,OT:VIEWPE(167)
571 POKE571,OT:VIEWPE(167)
572 POKE572,OT:VIEWPE(167)
573 POKE573,OT:VIEWPE(167)
574 POKE574,OT:VIEWPE(167)
575 POKE575,OT:VIEWPE(167)
576 POKE576,OT:VIEWPE(167)
577 POKE577,OT:VIEWPE(167)
578 POKE578,OT:VIEWPE(167)
579 POKE579,OT:VIEWPE(167)
580 POKE580,OT:VIEWPE(167)
581 POKE581,OT:VIEWPE(167)
582 POKE582,OT:VIEWPE(167)
583 POKE583,OT:VIEWPE(167)
584 POKE584,OT:VIEWPE(167)
585 POKE585,OT:VIEWPE(167)
586 POKE586,OT:VIEWPE(167)
587 POKE587,OT:VIEWPE(167)
588 POKE588,OT:VIEWPE(167)
589 POKE589,OT:VIEWPE(167)
590 POKE590,OT:VIEWPE(167)
591 POKE591,OT:VIEWPE(167)
592 POKE592,OT:VIEWPE(167)
593 POKE593,OT:VIEWPE(167)
594 POKE594,OT:VIEWPE(167)
595 POKE595,OT:VIEWPE(167)
596 POKE596,OT:VIEWPE(167)
597 POKE597,OT:VIEWPE(167)
598 POKE598,OT:VIEWPE(167)
599 POKE599,OT:VIEWPE(167)
600 POKE600,OT:VIEWPE(167)
601 POKE601,OT:VIEWPE(167)
602 POKE602,OT:VIEWPE(167)
603 POKE603,OT:VIEWPE(167)
604 POKE604,OT:VIEWPE(167)
605 POKE605,OT:VIEWPE(167)
606 POKE606,OT:VIEWPE(167)
607 POKE607,OT:VIEWPE(167)
608 POKE608,OT:VIEWPE(167)
609 POKE609,OT:VIEWPE(167)
610 POKE610,OT:VIEWPE(167)
611 POKE611,OT:VIEWPE(167)
612 POKE612,OT:VIEWPE(167)
613 POKE613,OT:VIEWPE(167)
614 POKE614,OT:VIEWPE(167)
615 POKE615,OT:VIEWPE(167)
616 POKE616,OT:VIEWPE(167)
617 POKE617,OT:VIEW
```

# Basicode – Software für alle

von Nikolaus Heusler

Der Rundfunk der DDR bezeichnet es zu Recht als »Esperanto für alle Computer« – die Universal-Programmiersprache *Basicode*. Bei *Basicode* handelt es sich um eine Basic-ähnliche Sprache mit einem Umfang von etwa 80 Befehlen. Das Besondere daran: Sie können damit Programme schreiben, die jeder Computer versteht. Man kann ein Programm beispielsweise unter *Basicode* auf einem Atari XE schreiben – es funktioniert ohne Anpassung sofort auch auf dem C64.

## Codiertes Basic

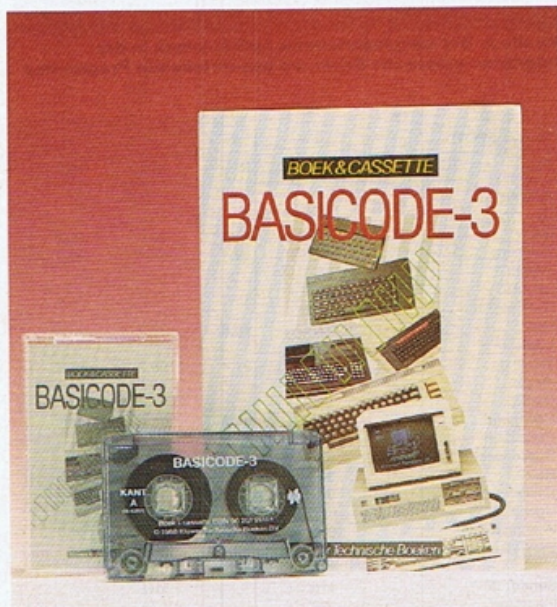
Damit das alles funktioniert, sind spezielle Übersetzerprogramme, sog. »Bascode«, notwendig. Ein einfaches Beispiel: In einem *Basicode*-Programm soll ein Pieps ertönen. Im Programmfile wird jetzt nicht der Befehl geschrieben, der den Ton hervorbringt (beim C64 wären dies sogar mehrere Befehle), sondern

GOSUB 400  
der Befehl für »springe in das Unterprogramm Tonerzeugung«. In den reservierten Variablen SV, SD und SP werden die Lautstärke, Dauer und Tonhöhe übergeben. Wird dieses Programm z. B. auf einem C64 eingelesen, sorgt der C64-Bascode dafür, daß sich ab Zeile 400 eine Routine mit den entsprechenden POKEs findet, die den Ton erklingen läßt. Wird das Programm auf einem PC eingelesen, löst dieser in Zeile 400 ein PRINT CHR\$(7) aus, der auf dem PC einen Ton hervorbringt. Der Bascode, der dieses Programm auf einem Atari ST umsetzt, hat seinerseits die Befehlsfolge ab Zeile 400, die einen ST zum Tönen bringt. Sie sehen, die *Basicode*-Programme »sagen« nur den Bascodern, die es mittlerweile für sehr viele Computer gibt, was sie zu tun haben. Die Bascode »wissen«, wie der jeweilige Computer programmiert werden muß, um die gewünschte Aktion – etwa die Ausgabe eines Tons – zu starten. Ein *Basicode*-Programm darf daher erst ab Zeile 1000 beginnen, da in den Zeilen davor die computerspezifischen Unterprogramme der jeweiligen Bascode stehen.

## Computer-Esperanto

Ein anderes Beispiel: Sie wollen in *Basicode* den Bildschirm löschen. Befehle wie  
PRINT CHR\$(147) bzw.  
PRINT <CLR HOME>  
für den C64 oder  
CLS bzw. CLEAR SCREEN  
für andere Computer sind verbo-

Dasselbe Programm läuft auf unterschiedlichen Computern – *Basicode* macht's möglich. Zur Zeit gibt es dieses als »Computeresperanto« gepriesene System nur in Holland und der DDR, aber es könnte schon bald bei uns auftauchen.



Der Einstieg in *Basicode* erfolgt leider auf holländisch: ausführliches Buch mit Programmkassette

ten, Sie müssen statt dessen  
GOSUB 100  
schreiben. Zeile 100 gehört zum Bascode-Bereich, sie enthält den Befehl, der auf dem jeweiligen Rechner den Bildschirm löscht. Ähnlich könnte man sich das z. B. mit dem LOCATE-Befehl vorstellen, der einen Text an einer bestimmten Stelle auf dem Bildschirm ausgibt. Da es diese Anweisung im C64-Basic gar nicht gibt, wird der Text vom Bascode mit Hilfe von POKEs in die System-Speicherzellen, die die Cursorposition bestimmen, hineingeschrieben. Tabelle 1 zeigt einige Beispiele für Befehle, die vom Bascode umgesetzt werden.

Befehl	Funktion
GOSUB 100	Bildschirm löschen
GOSUB 260	Zufallszahl erzeugen
GOSUB 350	Druckerausgabe
GOSUB 400	Ton ausgeben
GOSUB 500	Datei öffnen
GOSUB 630	Linie in Hires-Grafik zeichnen
GOTO 950	Programmende

Tabelle 1. Einige Beispiele für Subroutinen in *Basicode*

## Universal-Lader

Damit man die *Basicode*-Programme (also die Listings nach dem Bascode ab Zeile 1000) auch auf jedem Computer laden kann und nicht erst eintippen muß, wurde außerdem ein einheitliches Kassettenformat eingeführt, das alle Bascode beim Speichern anwenden und beim Laden verstehen. So kann ein Band, das z. B. auf einem Sinclair Spectrum unter *Basicode* aufgezeichnet wurde, mit jedem anderen Rechner wieder gelesen werden (sofern daran ein Kassettenlaufwerk angeschlossen ist). Dem Austausch von Software sind somit kaum Grenzen

gesetzt. *Basicode* ist daher auch für Mailboxen oder den Rundfunk interessant.

Ein weiterer Pluspunkt: Kommt ein neuer Computer auf den Markt, muß dafür nur ein weiterer Bascode verfaßt werden und schon läßt sich die gesamte Auswahl an *Basicode*-Programmen weiterverwenden.

## Die Schattenseiten

Natürlich hat dieses System auch einige Nachteile. Beispielsweise müssen *Basicode*-Programme ein ganz spezielles Format einhalten, damit wirklich jeder Bascode etwas damit anfangen kann und das Programm für alle verständlich ist. Das Programmieren in *Basicode* ist daher etwas gewöhnungsbedürftig (Tabelle 2). Außerdem muß sich der gesamte *Basicode*-Standard vom Leistungsumfang her am schlechtesten Computer orientieren. Beispiel: Wäre in *Basicode* eine Grafikauflösung von beispielsweise 640 x 400 Punkten vorhanden, so könnte man Programme, die diese Auflösung verwenden, zwar auf einem Atari ST anwenden, nicht jedoch auf dem C64, der ja nur 320 x 200 Punkte bietet. Da dies der *Basicode*-Philosophie widerspricht, gibt es keine absolute Vorgabe zur Bildschirmauflösung, sondern nur die Vorschrift, daß das Verhältnis von Zeilen zu Spalten etwa 3:4 sein soll.

Der Bascode zum C64 arbeitet z. B. mit einer Auflösung von 200 Zeilen mit 288 Punkten. Eine Grafik erscheint so auf allen Rechnern an der gleichen Stelle und in gleicher Form. Nur die Qualität hängt vom jeweiligen Computer ab.

Ein Nachteil ist auch, daß neben dem rechner-spezifischen Bascode für manche Computer (z. B. PCs) erst zusätzliche Hardware angeschafft werden muß, um ein Kassettenlaufwerk anschließen zu können. Eine kleine Auswahl von Computern, für die bereits ein Bascode zur Verfügung steht: C64, C128, C 16, Plus/4, verschiedene Modelle von Apple, Acorn, IBM-kompatible PCs, Atari 800, XL, XE, ST (in der Entwicklung) und Sinclair ZX-Spectrum.

## Basicode in der Praxis

Zum Betrieb von *Basicode* auf dem C64 braucht man neben einer normalen Datasette 1530 keine weitere Hardware. Zunächst sollte man sich einen passenden Bascode beschaffen (z. B. aus dem am Ende des Artikels genannten Buch) und diesen zur Sicherheit auf Diskette kopieren. Dann kön-



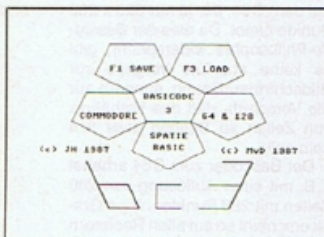
## BASICODE-3

voor commodore 64

(c) 1987 j. haubrich  
eindhoven  
(c) 1987 m.v. deelen  
de glind

**So meldet sich die C64-Version von Basicode-3 nach dem Laden von Kassette oder Diskette**

nen die *Basicode*-Programme ganz einfach auf Tastendruck über die Datasette eingelesen werden, seien es Aufzeichnungen aus dem Radio oder von Freunden bzw. selbst bespielte Kassetten. Wir haben es ausprobiert: Es klappt tatsächlich ohne Probleme. Die *Basicode*-Programme werden über das Menü des Bascoders bequem vom Band eingelesen, was mit der üblichen, nicht besonders hohen Geschwindigkeit vonstatten geht. LOAD ERRORS tauchten erstaunlicherweise nicht auf. Die Programme lassen sich wie gewohnt mit LIST ansehen und editieren oder mit RUN starten. Die Ablaufgeschwindigkeit ist, obwohl es sich doch eigentlich um Basicprogramme handelt, recht hoch, weil der Bascoder selbst ein Maschinenprogramm ist.



**Das Hauptmenü der C64-Version von Basicode-3 erscheint nach Druck auf die F1-Taste**

### Basicode im Radio

Die Idee zu *Basicode* kommt aus Holland, es wurde dort bereits 1979 entwickelt. Zur Zeit wird hauptsächlich die dritte Version der Sprache, *Basicode-3*, verwendet. Findige Programmierer sind bereits daran, *Basicode-4* zu entwickeln. Dank der holländischen Spezialisten stehen Bascoders heute für praktisch alle verbreiteten Rechner zur Verfügung, weitere werden gewiß noch entstehen. Da ein *Basicode*-Programm für alle Rechner gleich »aussieht« und sich über UKW oder Mittelwelle ausstrahlen und – bei Mittelwelle sogar über viele hundert Kilometer – empfangen läßt, bietet es sich an, die Programme auch im Rundfunk zu verbreiten. Solche Sendungen gibt es in Holland und auch in der DDR regelmäßig.

Zeilen	Inhalt
0-999	Kopf des Bascoders, Subroutinen
1000	Programmname des <i>Basicode</i> -Programms
1010-19999	Hauptprogramm in <i>Basicode</i>
20000-24999	evtl. Maschinensprache-Unterprogramme, verbotene Befehle
25000-29999	DATA-Zeilen
30000-31999	REM-Zeilen: kurze Programmbeschreibung, Literaturhinweise
32000-32767	REM-Zeilen: Programmautor, Adresse, andere formale Hinweise

**Tabelle 2. Der vorgeschriebene Aufbau eines in der Programmiersprache Basicode geschriebenen Programms**

Sender (UKW)	1	2	3	4
FM	MHz	MHz	MHz	MHz
Amsterdam	94,3			
Den Haag	88,4			
Goes	101,9	87,9	95,0	99,8
Hulsberg	95,3	92,1	103,9	98,7
Irnsrum	88,6			
Lalystad	102,1			
Loon op Zand	98,2			
Lopik	92,6	96,8	98,9	
Losser	89,4			
Markelo	95,6	91,4	96,2	98,4
Megen	89,1			
Mierlo	91,9			
Roermond	100,3	88,2	90,9	94,5
Rotterdam			93,4	
Roosendaal	95,4			
Ruurlo	90,4			
Smilde	90,8	88,0	91,8	94,8
Ugchelen	103,5			
Wieringermeer	93,9	87,7	89,8	92,2
Zwollerkerspel	99,4			
Sender (MW)	kHz		kHz	
Flevoland	747		675	
Hulsberg	1251			

**Tabelle 3. Sendefrequenzen von Basicode-Programmen des niederländischen Rundfunks**

Sender (UKW)	Radio DDR II	DT 64 (MHz)
Frankfurt/O.	87,6	101,5
Putbus	88,6	91,5
Helpterberg	90,5	103,8
Marlow	91,0	100,8
Sonneberg	91,7	102,7
Dresden	92,2	102,4
Inselsberg	92,5	102,2
Schwerin	92,8	101,3
Karl-Marx-Stadt	92,8	100,0
Suhl	93,7	—
Leipzig	93,9	102,9
Brocken	94,6	101,4
Dequede	94,9	101,0
Löbau	98,2	91,8
Cottbus	98,6	103,2
Berlin	99,7	102,6
Hoyerswerda	100,4	—
Marlow	102,8	—
Sender (MW)		
Burg	—	657 kHz
Neubrandenburg	—	657 kHz
Reichenbach	—	657 kHz

**Tabelle 4. Sendefrequenzen von Basicode-Programmen des DDR-Rundfunks**

Ein C64-Anwender, der ein im Radio gesendetes *Basicode*-Programm anwenden will, geht so vor: Er zeichnet die Radiosendung mit einem beliebigen Kassettenrecorder auf eine normale Kompaktkassette auf. Anschließend ist mit dem C64 der Bascoder von Diskette oder Kassette zu laden und zu starten. Nun wird die Kassette mit der Aufzeichnung vom normalen Kassettenrecorder in die Datasette gelegt, das aus dem Radio aufgezeichnete Programm läßt sich in den C64 einlesen. Anwender anderer Computer gehen genauso vor, immer vorhanden sein muß der systemspezifische Bascoder.

Eine ausführliche deutschsprachige Broschüre über die Programmierung in *Basicode* mit den Sendefrequenzen von *Radio DDR* (Tabelle 3) ist auf Anfrage kostenlos bei *Radio DDR* erhältlich. Leider kann man diese Sendungen in vielen Regionen Westdeutschlands ohne spezielle Antennenanlagen nicht empfangen, das gilt auch für die Sendungen des holländischen Rundfunks (Tabelle 4). Es ist jedoch nicht ausgeschlossen, daß auch unsere Rundfunkstationen irgendwann einmal die Möglichkeiten von *Basicode* entdecken und – beispielsweise im Rahmen einer regelmäßigen Sendereihe zum Thema *Computer* – Programme ausstrahlen. Wir haben auch gleich telefonisch bei einigen Rundfunkanstalten angefragt, doch kaum jemand hätte überhaupt etwas davon gehört. Andererseits fanden alle Befragten die Idee ganz toll, und das geht uns nicht anders. Gerade jetzt, da die DDR sich öffnet, wäre auch hier eine Zusammenarbeit wünschenswert. Je mehr *Basicode*-Anwender es gibt, desto mehr Software wird es für das »Esperanto der Computer« geben. (pd)

#### Kontaktadressen:

Stichting *Basicode*: Herr Haubrich, Postbus 1410, NL-5602 BK Eindhoven

N.O.S. Hilversum: Hobbyscoop, Postbus 1200, NL-1200 BE Hilversum

M. van Deelen: Rudolphlaan 28, NL-3794 MZ de Glind (speziell zum C64/C128)

Radio DDR: Schulfunk, Herr Dr. J. Baumann, Nalepastraße 10-50, DDR-1160 Berlin

#### Sendungen:

Holland: *Basicode Software-Service*: jeden Montag 21.35 bis 22.00 Uhr, Frequenz 1008 kHz (Mittelwelle)

Holland: *de TROS en BASICODE-3*: jeden Mittwoch 17.40 bis 17.55 Uhr, Frequenz 1008 kHz (Mittelwelle)

DDR: *REM - Das Computermagazin*: jeden ersten und dritten Mittwoch im Monat, 17.00 bis 17.30 Uhr auf *Radio DDR II* (UKW, diverse Frequenzen)

DDR: *Computer-Club*: jeden zweiten und vierten Donnerstag im Monat, 18.30 bis 19.00 Uhr auf *DDR-Jugendradio DT 64*, Frequenz 657 kHz (Mittelwelle)

#### Literatur:

*Het BASICODE-3 boek*, Buch und Kassette, ISBN 90-201-2111-1, NUGI 434/857, Kluwer Technische Boeken BV Verlag, Deventer-Antwerpen, Belgien (zwar auf holländisch, aber auch ohne Kenntnisse dieser Sprache einigermaßen verständlich. Dem Buch liegt eine Kassette mit einem Bascoder u.a. zum C64 und Demoprogrammen bei.)

## BASICODE-3 Programmatuur op verzamelcassette bijeen

Alle BASICODE-3 programma's die tussen juli 1986 en februari 1987 zijn uitgezonden via het wekelijkse radio-programma 'De TROG en BASICODE-3' (elke woensdag vanaf 17.41 op radio 5, 1007 MHz), zijn nu op een speciale verzamelcassette bijengebracht, samen met enkele programma's die om diverse redenen niet werden uitgezonden. In totaal bevat de cassette ruim 50 minuten computeprogramma's in BASICODE.

Deze verzamelcassette is samengesteld door de Stichting BASICODE en wordt tegen kostprijs beschikbaar gesteld. De cassette kan eenvoudig worden besteld door 19.50 over te maken op girorekening 5391330 van de Stichting BASICODE te Eindhoven onder vermelding van "verzamelcassette 1". De cassette wordt dan thuisgestuurd; de prijs is inclusief verzendkosten.

BASICODE-3 blijkt een hernieuwde belangstelling voor het systeem BASICODE te hebben opgeleverd. Dit is ongetwijfeld mede te danken aan de in BASICODE-3 geïntroduceerde grafische mogelijkheden en aan de mogelijkheid om gegevens in de vorm van een bestand op cassetteband op te slaan. Deze gegevens kunnen voor

worden ingelezen in elke computer waarvoor een BASICODE-3 vertaalprogramma bestaat.

De meeste thans beschikbare vertaalprogramma's staan op de cassette die meestal bij het BASICODE-3 boek wordt geleverd en dat gewoon in de boekhandel verkrijgbaar is. Dit betreft vertaalprogramma's voor de Acorn Electron en BBC computer, de Apple-II familie, de Commodore 64 en 128, de Exidy Sorcerer, alle MSX-computers, de Philips P2000T en P2000M, alle Sinclair Spectrum modellen en tenslotte de Spectrandio SV.318 en SV.328.

Vertaalprogramma's voor de Acorn Master, de IBM PC en compatibelen, de Schneider CPC familie en de Tandy Color Computer zijn nog niet op de cassette opgenomen, maar zijn wel apart verkrijgbaar.

Een exemplaar van de verzamelcassette wordt u op aanvraag toegezonden.

## Basicoding cassettes

READERS OF June's article on my Basicode 2 adventure writing system may like to know that cassettes of the program at £2 each post free are available from RTL, Westowan, Porthtowan, Truro, TR4 8AX. To use one of these cassettes readers will need a Basicode kit for their computer, and some machines may require the program to be modified before it will run. Details will be with their kits.

*John de Rivas*  
Porthtowan  
Truro

# Changing locations automatically using Basicode

*John de Rivaz introduces the Basicode language and puts it to work*

THE PROBLEM of incompatibility dogs the micro world, but a language called Basicode 2 promises a solution.

Basicode is a "kernel" language which runs on various micros, provided you've bought the appropriate translation program — available from the BBC for micros such as the Spectrum, BBC B and Commodore 64. The BBC also broadcasts Basicode listings on its Radio 4 Chip Shop program.

This article looks at using Basicode to provide automatic location changing — but the techniques described apply equally to other Basics, so don't be deterred if you haven't got a Basicode kit yet. As further encouragement, advice on translating Basicode to machine-dependent Basics is given towards the end of the article.

## Background

But first a bit of background. In the June issue of *Micro Adventurer* a system of writing adventure games in Basicode was described, introducing the Chrononauts series as an example game.

This system involves most of the data for the adventure being fixed as DATA statements. There is only one string array, and that is the object table. The object table consists of a single string for each object. It is introduced to the program as DATA statements, but this time these are loaded into an array and the array is manipulated by the player's commands.

Listing one shows the object table in the second Chrononauts adventure, the Graveyard. This is really the heart of the game.

The listing starts with a letter denoting the object, a text description of the object, and a further string of characters that are used as flags for various actions by the computer. One of these characters is the location of each object. The locations are denoted by letters of the alphabet, but there is no reason why two characters cannot be reserved for larger numbers of locations. The text message must be somewhat short, but a further DATA table of additional text is used. This starts with the letter of the object followed by the extra text. When the command look is issued, every such DATA

statement is printed, with the exception of the initial identifying letter.

For example, line 26160 is a cleared rockfall. It can be made if a rockfall (object O), a shovel (object E), and a pick axe (object D) are present. As it is not present initially, the last letter of the line is a blank. If it is climbed into, the player is moved to location E. Of course, it can't be moved, killed, destroyed or burnt.

And now a programming solution to a problem of logic. Adventure games frequently result in the player being killed, and he then re-starts the game and tries again, and eventually he solves the puzzle. This is, of course, totally illogical. In real life, once one is dead that is it. There is no second chance, and the possibility of a personal afterlife is now even questioned by some of those people who still have religious beliefs.

A better representation of the situation would be to stop a killed player ever attempting that adventure again, but this would hardly be popular. However, the concept of Immortalism, the use of science and technology to extend life and eventually eliminate death, does have a logical answer — which actually stems from computing.

## Clones

If a programmer is working on a program, he will save the program every so often. Should there be a power cut, he does not lose all of his work, but only that between the last saving and the cut. Indeed, the same would apply were the computer to catch fire and be totally consumed. All that would be required would be to load the program into a replacement computer.

In Immortalism, the concept exists of saving the program and data in the brain at periodic intervals, and these can be replayed into a clone should the individual concerned die. In the Graveyard this concept is used, except that the player is provided with a transmitter which continually sends program and data in his brain to his time ship. Listing 2 illustrates this. If the player is killed (by supposed vampires) he is re-constituted within his ship, and the objects he was carrying are

left at the location where he was killed. The "vampires" conveniently remove his remains.

Line 15400 checks to see if an object is worn or carried, and if so line 15410 re-sets its location to that at which the player is killed. Line 15430 re-sets the player's location to that of the ship, where his brain patterns are played into the clone.

## Sirens

In the third Chrononauts adventure, the Sirens of Space, an alternative to killing the player is used. Actually in this case he can be killed by outright stupidity in one instance, although this sequence would probably be avoided by most adventure enthusiasts as it is so obvious.

However, the most likely problem he will meet is being rendered unconscious, and whilst he is unconscious some of the objects are shuffled around the locations. In fact, he can perform actions to prevent this sequence happening, but clearly with a regular re-distribution of objects this can take some time to achieve.

Shuffling the objects is programmed quite simply, as in Listing 3 from the Sirens of Space. Since the object strings have a single character which denotes their location, a subroutine is called that inserts another character there. A DATA statement, line 15220, is used to contain the letters defining the objects that can be so moved, and each object is then called in turn at line 15230 and its location is randomised in line 15340. The subroutine 260 is the Basicode subroutine that puts a random number between 0 and 1 into variable RV.

I would like to stress the technique used of putting a slash (/) at the end of the data statement, and checking whether this is present before jumping out of the routine. I have used this technique extensively in this system, instead of FOR NEXT loops, because it makes for far easier alteration. All you have to do is to add another object. If a FOR NEXT loop is used, you also have to alter the FOR NEXT variable each time you add or remove an object.

There's various ways to use the ideas outlined in this article. The June issue of

```

26000 Rem OBJECT DATA. Object letter, description, start loc.
26001 Rem penultimate W indicates it can be worn
26002 Rem A b before this indicates it can be burned
26003 Rem A k before this, killed, and a d destroyed.
26004 Rem An f before this means it can be filled
26005 Rem any other letter means it can be emptied,
26006 Rem and the letter is the contents, unused for fill
26007 Rem The next five are the ingredients required
26008 Rem to make this item. If(S, fill in with S
26009 Rem If they are all 4, then it can't be made.
26010 Rem The next letter is an i if the object is not
26011 Rem used up when making something.
26012 Rem If instead of a W the penultimate letter is a
26013 Rem letter, it means the object can be entered, and
26014 Rem the location of its interior is the letter. If
26015 Rem the letter is an "i" it simply means the object
26016 Rem cannot be taken, it can only be taken if the
26017 Rem if a W or a -. The location W cannot be used for
26018 Rem for the inside of an object.
26026 Data "A) A coffin. -#####-d-E"
26028 Data "B) A body. -#####-d"
26030 Data "C) A locked gate. -#####-I?"
26040 Data "D) A pickaxe. #####-d-J"
26050 Data "E) A shovel. #####-d-J"
26060 Data "F) Holy water. #####-d"
26070 Data "G) A font. -#####-d-A"
26080 Data "H) A newsheet. -#####-d-N"
26090 Data "I) A tin hat. -#####-d-W"
26100 Data "J) A box of matches. #####-d-O"
26110 Data "K) A box. -#####-d-O"
26120 Data "L) A key. #####-d-"
26130 Data "M) A sack. -#####-d-P"
26140 Data "N) A flame gun. -#####-d-J"
26150 Data "O) A rockfall. -#####-d"
26160 Data "P) A cleared rockfall. -#####-E"
26170 Data "Q) An unlocked gate. -#####-K"
26180 Data "R) A locked gate. -#####-H"
26190 Data "S) A Lunatic Asylum. -#####-L"
26200 Data "T) Ruins of a lunatic asylum. -#####-I"
26210 Data "U) A Vampire. -#####-I"
26220 Data "V) An unlit lamp. -#####-d-O"
26230 Data "W) Tar oil. #####-d-I"
26240 Data "X) A lit lamp. -#####-d-"
26250 Data "Y) A red bottle -#####-d-Z"
26260 Data "a) A blue bottle -#####-d-Z"
26270 Data "b) A person. -#####-d-Z"
26280 Data "c) A coded message. -#####-d-Z"
26990 Data "/"
26998 Rem
Done.

```

Listing one — the object table

```

15250 Rem
15260 Rem Return player to ship
15270 Rem
15280 Print:Print"          However I am"
15290 Print"carrying an apparatus that sends"
15300 Print"continuous updating information on"
15310 Print"the program and data in my brain"
15320 Print"to the ship's computer. As soon as"
15330 Print"I am killed, the computer activates"
15340 Print"a brainless clone and plays the data"
15350 Print"into it. Thus I have invulnerability!"
15360 Print"However, I must now start again from"
15370 Print"from the ship, and all the objects"
15380 Print"are left where I was killed."
15390 For N=1 to TH: X$=Right$(TH$(N),1)
15400 If X$(">") And X$("<") Then 15420
15410 TH$(N)=Left$(TH$(N),Len(TH$(N))-1)+LO$
15420 Next N:Rem objects dropped.
15430 LO$="P":Print
15440 Print"Any key to continue":Gosub 210:Goto 2020
19000 Rem
Done.

```

Listing two — drop objects and return to ship

```

15190 Rem randomise objects,
15200 For N=2 to TH:Restore 15210
15210 Rem objects to be randomised
15220 Data "B" "C" "D" "G" "I" "J" "M" "/"
15230 Read X$:If X$="" Then 15360
15230 If Left$(TH$(N),1) < X$ Then 15230
15240 Gosub 260: X$=Chr$(RV$(14)+65)
15250 TH$(N)=Left$(TH$(N),Len(TH$(N))-1)+X$:Goto 15230
15360 Next N
Done.

```

Listing three — randomising objects

Micro Adventurer had a listing of the first Chrononauts adventure, Morphostasis, and you could start by getting that working. Alternatively, obtain a listing of Basicode cassette of the three adventures as detailed below, then modify their tables to suit your own plot. Either choose a plot that is similar to the type of plot in these adventures, or, if you are more adventurous, use additional Basicode statements to modify the special sections. See how the second and third games evolved from each other. Ideally, the original core of subroutines (take, fill, climb into, etc) described in the previous article can be used in conjunction with many different games. Writers will add new routines such as those described here to make new types of games.

As Basicode becomes more popular, broadcasting stations throughout the world will want to take this material. In theory, it should be possible to transmit Basicode by CB radio locally. Morphostasis has been sent to the BBC, although at the time of writing I do not know whether they will be broadcasting it.

### Translating

Listings of the three Chrononauts adventures are available from the address below. Send two 12½p stamps for each adventure required. You will receive the listings, which are very long, to type into your machine. Owners of Sinclair machines will have to change the string handling from MDS etc to the simpler Sinclair system. Unless they have Basicode kits, users of all machines will have to insert a line 10 "GOTO 1000" and compose routines to clear string space in line 20 according to the value A, and GOTO1010. Also subroutines will have to be composed as follows:

100 Clear screen and set cursor to left hand corner.

200 Check for keypress and put in IN\$. If none, IN\$=" ".

210 Call 200 until key pressed.

260 Give random number RV between 0 and 1.

If you type in one of the programs, then entering the other programs can be performed by merely changing lines, as a lot of the code is common.

Or the adventures are available priced £2 each on cassette post free. These are recorded to the Basicode standard, and should only be ordered by those who already have Basicode kits for their machines. Deduct 25p for each additional program after the first ordered at the same time.

A free fact sheet on the practice of Immortalism in the USA is also available. It gives details of further reading, including a recent best seller on life extension. Also listed are addresses from which monthly magazines can be obtained. There is, of course, no obligation to purchase any of these.

And, finally, the address to write to is RTL, Westowan, Porthowan, Truro TR4 8AX, Cornwall. © John de Rivaz. □

# Treat for Chip Shop fans

*John de Rivaz outlines a system that enables speedy adventure writing*

THIS SYSTEM will enable adventure programs to be produced very quickly on any BASICODE2 computer. They may not be as fast as machine code adventures, nor may they be able to make pretty pictures; however they can run on most popular computers, creating an enormous audience for these programs.

If you do not have a BASICODE kit for your computer, then you can usually get one from your local broadcaster (the BBC in the United Kingdom) for around £3.99.

The system is more flexible and quicker to use than The Quill. This is because the program can be edited visually, not a line at a time, and basic subroutines can be added, for example if a simple arcade type action sequence is required. This is impossible with The Quill.

The kernel of the program is the data statements starting from line 25000. The data is not read in a FOR NEXT loop, therefore there can be any number of entries. Each section is ended by a "/". At the start and end of the strings there are letters that control the program.

## Locations

At the start of the location text is the location character. This is the character that defines the location. This somewhat limits the number of locations, but a well written adventure needn't have thousands of locations. After the description, there is a space followed by a - if the location is light, or a \* if it requires a source of light.

Next are the characters N S E W U D, for North South . . . Down, if followed by a blank, one cannot go in this direction. If followed by a letter, one can go to that location. It is therefore essential to make a plan of your proposed adventure before filling in this table. (Yes, I know that is the kind of boring thing a computer should be able to help with, but at the moment there is no software to do it.)

The Object Data follows a similar pattern. The first character, which is printed this time, is followed by text and further

Start address	20020		Figure 1
Input variables	LOS	A single letter denoting location	
Output variables	A\$	Full location text	
Variables used	ERS	"ERROR"	
	TH	No of things	
Arrays used	TH\$(TH)	Things array	
Internal variables	F1 N		
Function		Prints location and holds the text in A\$. Also printed is things worn or carried, and the contents of things filled.	

Start address	20410		Figure 2
Input variables	CS	Single byte — identification of object	
	LOS	A single letter denoting location	
Output variables	N	Number in array of thing	
Variables used	TH	No of things	
Arrays used	TH\$(TH)	Things array	
Function		Given letter of thing required, and LOS, it returns with CS a null string if object not there, worn or carried.	

Start address	20310		Figure 3
Input variables	INS	BASICODE-2 single byte input	
	LOS	A single letter denoting location	
Internal variable	CS		
Output variables	N	Number in array of thing	
Variables used	TH	No of things	
Arrays used	TH\$(TH)	Things array	
Function		Requests letter of thing required and returns to main loop and prints "It is not here" or returns with N being the thing no.	

Start address	3220		Figure 4
Input variables	INS	BASICODE 2 input byte	
Output variables	none		
Variables used	none		
Arrays used	none		
Internal variables	A\$ F2 N		
Function		Returns with "I see nothing unusual" or a description of the thing being looked at.	

Start address	3210		Figure 5
Input variables	INS	BASICODE-2 single byte input	
	LOS	A single letter denoting location	
	B\$	^ for take or ~ for wear	
Output variables	N	Number in array of thing	
Variables used	TH	No of things	
Internal variables	D		
Arrays used	TH\$(TH)	Things array	
Function		Takes or wears, returns message if thing not there or cannot be worn.	

Start address	3310		Figure 6
Input variables	INS	BASICODE-2 single byte input	
	LOS	A single letter denoting location	
	B\$	^ for take or ~ for wear	
Output variables	N	Number in array of thing	
Variables used	TH	No of things	
Internal variables	D		
Arrays used	TH\$(TH)	Things array	
Function		Leaves or removes, returns message if thing not there or cannot be worn.	

Start address	3510		Figure 7
Input variables	INS	BASICODE-2 single byte input	
	LOS	A single letter denoting location	
Variable used	CS		
Internal variables	D	delay loop	
	I\$	string store	
Output variables	N	Number in array of thing	
Variables used	TH	No of things	
Arrays used	TH\$(TH)	Things array	
Function		Removes objects burned, killed or destroyed, or sends message if this is not sensible or possible.	

Start address	3710	for fill	Figure 8
Input variables		obtained by calls	
Internal variables	C\$ LOS N D		
Variables used	TH	No of things	
Arrays used	TH\$(TH)	Things array	
Function	Fills a requested object with second requested object, or send message if this is not sensible or possible.		

Start address	3900	for empty	Figure 9
Input variables		obtained by calls	
Internal variables	C\$ LOS N D		
Variables used	TH	No of things	
Arrays used	TH\$(TH)	Things array	
Function	Empties a requested object and sends message if this is not sensible or possible.		

Start address	4020		Figure 10
Input variables		obtained by calls	
Internal variables	C\$ LOS N D J J1 J2 J3		
Variables used	TH	No of things	
Arrays used	TH\$(TH)	Things array	
Function	Makes a requested object from a list. Prints message if this is not possible, or if required objects not present.		

Start address	4320		Figure 11
Input variables		obtained by calls	
Internal variables	B\$ C\$ LOS N D		
Arrays used	TH\$(TH)	Things array	
Function	Sets locations to that specified by the object entered or returns with message if this is not possible or sensible.		

```

1000 Let A=2000: Goto 20
1010 Rea LOAD UP ARRAY
1020 Restore 26000:N=1:ER$="ERROR - ":LD$="A":FF=0
1030 Read A$:If A$=""Then I050
1040 N=N+1:Goto 1030
1050 TH=N-1:Dim TH$(TH):Restore 26000
1060 For J=1 To TH:Read TH$(J): Next J
1070 Gosub 100
1072 Print "THE CHRONONAUTS":Print
1074 Print "This series of adventures features the"
1076 Print "the concept of immortalise, where people"
1078 Print "strive to reach a future age where"
1080 Print "where death is abolished."
1082 Print:Print"Any key to continue":Gosub 210:Gosub 100
1084 Print "Episode 01 - Morpheostasis":Print
1086 Print "I am a Victorian scientist looking for"
1088 Print "a way to survive. Can you help?":Print
1090 Print "(Hold keys down until soething happens)"
1092 Print:Print"Any key to continue":Gosub 210
1992 Rea
2000 Rea MAIN LOOP
2010 Rea MOVE AROUND
2012 Rea
2020 Gosub 100
2030 Gosub 20020:AA$=A$:If FS=1Then 2100
    
```

Continued on page 28

characters. The last of these defines the start location. If the object is not located, the last character must be a space. Use "" if it starts off being carried, and "-" if it starts off being worn.

The following single characters, which are "." if not present, define whether the thing can be the subject of the following activities: being used for filling, filled, burned, killed, destroyed or worn (ufbkdW). If the worn character is a - the object can be worn or climbed into (entered). If it is a letter other than W, then this is the location letter of the location that is reached if the object is entered. If it is \* then this object cannot be taken and carried. It may be necessary to fill another object with this one if it is to be moved. At the current development of the program, objects have to be emptied out before the contents can be manipulated ie if the red bottle contains chloroform and you want to use chloroform to make something, then the red bottle must first be emptied.

#### Ingredients

Preceding these is a group of five, which are "" if not used. These are ingredients if the object can be made up from other objects. Preceding this is another space that is either "." or "I". It is "I" if the object is not used up when it is used to make things. For example, if you use sand to make concrete, you may have some left on the heap.

The messages for "Looks" are printed upon the command L for "Look" (surprisingly). They are followed by the object to be described. If more than one line is used, then simply have more than one data line preceded by the same letter. The sub-routines start at line 20000. They are called from the main loop, which itself is quite short, but it branches into a number of command loops and calls a large subroutine which controls the computer's moves. The most important of these is the subroutine that prints out the current location and conditions (Fig 1).

It is often required to see if an object is at a particular location, and find its position in the object array. This is the function of the routine starting at 20410 (Fig 2).

An important variation on this is to get a key press and then see if the object is present. If it is not, return is to the main loop, or a return to the calling routine is effected (Fig 3).

The main loop starts at 2020, with a printing of the location and a menu of commands. Each command is entered by a single key press. It is not considered a valid pastime to get the user of an adventure game to guess at what words are in the menu, or to get him to bash away at his keyboard typing in long words like examine when pressing L would do just as well. The main loop controls the direction commands, and other commands are controlled by jumps to routines starting at 3220. The first routine is Look, and it is at 3220 (Fig 4).

Take and Wear are grouped in the next routine (Fig 5) - while Leave and Remove follow (Fig 6).

Burn, Kill and Destroy are the next routines. These only work on objects that >

< can be so treated, to be set up as previously described under object data. If an unsuitable object is chosen, then a message is printed and control passed back to the main loop. If not, then the object is taken out of circulation by having its right most character changed to a space (Fig 7).

Line 3610 is used to check that a means of burning is present. If object J is matches, the routine can be used without modification. If another object is used, object Z for example, then replace the J with Z.

Fill and empty are quite complicated, as they have to check up on two objects and check that a silly combination is not used, the object to be filled is not full, or the eg the object to be filled is not full (Figs 8 and 9).

Making things is an important part of these adventures. The puzzle is usually how to find the bits to make various objects. Also, one can make a hole in a wall and then go through it, which can be difficult if you first have to find something with which to make the hole (Fig 10).

The routine does say what things can be made, but just to make it a little more difficult some of the things listed will be destroyed objects that cannot be made. For example, in the specimen program if you kill the bantam it appears on the list of things to make.

Climbing into an object, mentioned above, is very useful if you want to block off part of the locations until preliminary tasks are completed. For example a hole in the wall can be made if the wall and a pick axe are present. The wall is destroyed, and the pick axe left. The wall has an "L" in the object data, where the pick axe has an "I" (Fig 11).

### Specialising

The advantage of a BASIC adventure game over a machine code one written with a system like The Quill is that you can easily add specialist routines such as drill a hole in the example program. This sets a flag HC or HD if objects C or D have a hole drilled in them. Actually it wasn't used in the final version of the game, but was left in as a red herring.

What happens now as a result of the player's manipulation of the environment within the computer, and (usually) various computer moves? In the case of the adventure Morphostasis the game is finished if a flag is set when the friend is found, and the morphostatic coffin is made. If the player goes into the garden without a coat he stands a greater chance of dying, and also his chances of dying increase after 90 moves.

There is a counter NM in line 2100 which records the number of moves. This flag is used by the routine at 15130 to set the probability of dying. Also, there is a routine at 15220 that detects if the player is outside with no coat. In order to save run-time, the actual number of the coat in the array is used (nine). The flag is set if the player passes through the first outside location without the coat, and reset when the player comes inside.

The torch may also get blown out. This

```

2040 Print "Press N S E W U D to move"
2042 Print "L - look: ";
2044 Print "T - take: ";
2046 Print "W - wear"
2048 Print "I - leave an item: ";
2050 Print "R - remove clothing"
2052 Print "B - burn: ";
2054 Print "K - kill: ";
2056 Print "D - destroy"
2058 Print "S - suppress: ";
2059 Print "R - re-instate instructions"
2061 Print "F - fill: ";
2062 Print "E - empty: ";
2064 Print "M - make: ";
2066 Print "C - climb"
2090 Rem Specialist commands
2092 Print "H - drill a hole"
2100 B$="":Gosub 210:NH=NH+1:Gosub 15000
2130 If IN$(L)Then 3020
2132 If IN$(T)Then B$="":Goto 3210
2134 If IN$(W)Then B$="":Goto 3210
2136 If IN$(I)Then B$="":Goto 3310
2138 If IN$(R)Then B$="":Goto 3310
2140 If IN$(B)Or IN$(K)Or IN$(D)Then 3510
2142 If IN$(C)Then FS=1:Goto 2020
2144 If IN$(R)Then FS=0:Goto 2020
2146 If IN$(F)Then 3710
2148 If IN$(E)Then 3910
2150 If IN$(M)Then 4040
2160 If IN$(C)Then 4320
2220 Rem Specialist commands
2222 If IN$(H)Then 10020
2226 Rem
2228 Rem THE MOVEMENT SECTION
2229 Rem
2230 A$=A$:For N=1To 6
2240 If Mid$(A$,Len(A$)-13+NH2,1)<>IN$Then 2260
2242 If Mid$(A$,Len(A$)-12+NH2,1)=""Then 2262
2250 L$=Mid$(A$,Len(A$)-12+NH2,1):Goto 2020
2260 Next N
2262 Print "I can't go there":Goto 2100
2990 Rem
3000 Rem Action routine section
3002 Rem
3010 Rem *** LOOK ***
3012 Rem
3020 Restore 27000
3040 Gosub 20310
3080 F2=0
3090 Read A$
3100 If A$<>"/"Then 3120
3110 If F2=0Then Print "I see nothing unusual.":Goto 2100
3112 Print:Print "Press any key to continue":Gosub 210
3118 Goto 2010
3120 If Left$(A$,1)<>IN$Then 3090
3130 If F2=0Then Gosub 100:F2=1
3140 Print Right$(A$,Len(A$)-2):Goto 3090
3190 Rem
3200 Rem *** TAKE AND WEAR ***
3202 Rem
3210 Gosub 20310
3220 If B$=""And Mid$(TH$(N),Len(TH$(N))-1,1)<>"W"Then 3280
3222 If Mid$(TH$(N),Len(TH$(N))-1,1)=""WThen 3240
3224 If Mid$(TH$(N),Len(TH$(N))-1,1)=""Then 3240
3226 Print "That is impossible!":Goto 2100
3240 TH$(N)=Left$(TH$(N),Len(TH$(N))-1)+B$
3250 Gosub 100:If B$=""Then Print "I have taken it"
3260 If B$=""Then Print "I am putting it on."
3270 For D=1 To 500:Next D:Goto 2020
3280 Print "I can't wear THAT!":Goto 2100
3290 Print "I can't take THAT!":Goto 2100
3298 Rem
3300 Rem *** LEAVE AND REMOVE CLOTHING ***
3301 Rem
3310 Gosub 20310
3312 If B$=""And Right$(TH$(N),1)<>"W"Then 3390
3314 If B$=""And Right$(TH$(N),1)<>"R"Then 3380
3320 If B$=""And Mid$(TH$(N),Len(TH$(N))-1,1)<>"M"Then 3382

```

occurs at 15050. The routine changes the lit torch to the unlit torch. A flag is set at 15180 if the player meets the friends and also looks at him. Finally, a routine 19030 detects whether the friend has been met and looked at and the morphostatic coffin made.

If you would like to use the system for your ideas you should first make a plan on paper of your environment. Place interconnecting lines between the boxes showing the directions between them. Then show at each location the objects to be found there. Then make a list of the objects to be made. There should be one object that when made provides the solution to the game, or can be used to find the final location. For example, you could make a space ship that is entered and then flown (a special instruction) to the final location, and the computer's move section merely detects that you get to that location. The components of this special object could be made from other objects.

### Grim

The Chrononauts is a series of adventures based on the idea of individuals' struggles to find the secret of immortality, or to prevent themselves falling victim to The Grim Reaper. There are several Chrononauts adventures, with graphics, written for the Spectrum with The Quill and distributed by Micronet 800, and these adventures are also available on the Porthowan Combo Tape for the Spectrum which costs £4. Morphostasis is the first Chrononauts adventure written in BASICODE2. Although the copyright of each Chrononauts adventure remains with the writer, others users are encouraged to add to the series without legal formality or costs.

The BASICODE2 adventure writing system has interesting possibilities for mass audience text adventures. It does not have all the features and the run-time advantages of machine code systems like The Quill but has the advantage of greater flexibility and the possibility of adding special functions, even arcade action sequences.

Adventures written with my BASICODE2 system can be typed into computers without using a BASICODE2 kit, if readers are willing to write some simple subroutines to suit their machines. These are as follows:

100: Clear screen and sets cursor at 0,0 (top left hand corner).

200: Checks for keypress. If so, puts character in IN\$. If no character found, then IN\$ is an empty string.

210: Calls 200 repeatedly until a character is found.

260: Gives a random number in variable RV between 0 and 1.

Some computers require string space to be cleared, and this action should be performed before the main program is run. Line 1000 of a BASICODE2 program sets a variable A, which is the string space to be cleared, and then goes to line 20 which clears it and this return to line 1010.

There are more features than this in BASICODE2 but it is just these that are used in present examples of my system.

© John de Rivaz ☐

```

3540 TH$(N)=Left$(TH$(N),Len(TH$(N))-1)+L0$
3550 Gosub 100:If D$=""ThenPrint "I have left it"
3560 If D$=""ThenPrint "I am taking it off."
3570 For D=1 To 500 : Next D: Goto 2020
3580 If Mid$(TH$(N),Len(TH$(N))-1,1)=""Then 3400
3582 Print "I couldn't possibly be wearing THAT!":Goto2100
3590 Print "I am not carrying it":Goto 2100
3600 Print "I am not wearing it":Goto2100
3610 Rem
3620 Rem *** BURN KILL DESTROY ***
3630 Rem
3640 If IN$="b" Then Let I$="burn":Gosub 3610
3650 If IN$="k" Then Let I$="kill"
3660 If IN$="d" Then Let I$="destroy"
3670 Gosub 20310:ForJ=2To4
3680 If Mid$(TH$(N),Len(TH$(N))-J,1)=Left$(I$,1)Then3580
3690 Next J
3700 Print "I cannot 'iI$' THAT!":Goto2100
3710 TH$(N)=Left$(TH$(N),Len(TH$(N))-1)+ "
3720 Gosub 100: Print "I am now 'iI$'ing it."
3730 For D=1 To 500: Next D:Goto 2020
3740 N1=N:C$="J":Gosub 20410:Rem *** Insert matches etc ***
3750 If C$="J" Then N=N1:Return
3760 Print "I can't yet": Goto 2100
3770 Rem
3780 Rem *** FILL ***
3790 Rem
3800 Print "What am I to fill":Gosub20310:N1=N:Gosub 3770
3810 Print "With what am I to fill it?":Gosub 20310
3820 If N=N1 Then Print "Don't be silly!":Goto2100
3830 D$=Left$(TH$(N),Len(TH$(N))-6,1)<>"u"ThenN1=N:Goto3722
3840 D$=Left$(TH$(N),Len(TH$(N))-6)+IN$
3850 TH$(N1)=D$+Right$(TH$(N),5)
3860 TH$(N)=Left$(TH$(N),Len(TH$(N))-1)+ "
3870 Gosub100:Print "I am filling it up":ForD=1To500:NextD
3880 Goto 2020
3890 If Mid$(TH$(N),Len(TH$(N))-5,1)<>" "Then3790
3900 Print "I can't fill THAT!": Goto 2100
3910 If Mid$(TH$(N),Len(TH$(N))-5,1)=""ThenReturn
3920 Print "It is full": Goto 2100
3930 Rem
3940 Rem *** EMPTY ***
3950 Rem
3960 Print "What am I to empty?":Gosub20310:Gosub 3970
3970 C$=Mid$(TH$(N),Len(TH$(N))-5,1):D$=L0$:L0$=" "
3980 D$=Left$(TH$(N),Len(TH$(N))-6)+ "f"
3990 TH$(N)=D$+Right$(TH$(N),5):Gosub20410:L0$=D$
4000 If C$=""ThenPrint "Error - contents do not exist":Stop
4010 TH$(N)=Left$(TH$(N),Len(TH$(N))-1)+L0$
4020 Gosub100:Print "I am tipping it out":ForD=1To500:NextD
4030 Goto 2020
4040 If Mid$(TH$(N),Len(TH$(N))-5,1)<>" "Then3990
4050 Print "I can't empty THAT!": Goto 2100
4060 If Mid$(TH$(N),Len(TH$(N))-5,1)=""ThenReturn
4070 Print "It is already empty": Goto 2100
4080 Rem
4090 Rem *** MAKE THINGS ***
4100 Rem
4110 F1=0:ForN=1ToTH
4120 If Right$(TH$(N),1)<>" "Then4080
4130 If F1=0ThenPrint "I might be able to make":F1=1
4140 Print Left$(TH$(N),Len(TH$(N))-PO)
4150 Next N
4160 Print "Press E if you don't want anything made."
4170 Print "Press letter preceding the object to be made."
4180 Gosub 210:If IN$="E"Then 2010
4190 C$=IN$:D$=L0$:L0$=" ":Gosub 20410:L0$=D$:J2=N
4200 Print "I'll see if I can."
4210 If C$=""ThenPrint "I can't.":Goto 2100
4220 J2=5:D$=Mid$(TH$(N),Len(TH$(N))-11,J2)
4230 J1=0:ForJ=1ToJ2:Rem see if possible
4240 C$=Mid$(D$,J2-J+1,1)
4250 If C$=" "ThenJ1=J1+1:Goto 4190
4260 Gosub 20410
4270 If C$=""ThenPrint "I haven't all I need":Goto 2100
4280 NextJ:IfJ1=J2ThenPrint "I can't make that.":Goto2100
4290 Gosub 100:Print "I am making it."

```

Continued on page 20



```

4210 For J=1 To J2: Rem delete enshau
stable supplies
4220 C$=Mid$(R$,J2-2+1,1)
4230 If C$="*"Then 4270
4240 Gosub 20410
4250 I$(M$(THEIN),Len(THEIN)-1
2,1)+C$Then 4270
4260 THEIN=Left$(THEIN),Len(THE
IN)-1+" "
4270 Next J
4280 THEJ3=Left$(THEJ3),Len(T
HEJ3)-1+L$
4290 Goto 2010
4300 Rem
4305 Rem *** CLIMB INTO AN OBJEC
T ***
4310 Rem
4320 Print "What shall I climb i
n?"
4330 Gosub 20310
4340 B$=Mid$(THEIN),Len(THEIN)-
1,1)
4350 If B$="N"Then Print "I can
NEAR that."
4360 Goto 2100
4370 If B$="E"Then Print "I can't e
nter that."
4380 Goto 2100
4390 If B$="S"Then Print "I can't e
nter that."
4400 Goto 2100
4410 If B$="W"Then Print "I can't e
nter that."
4420 Goto 2100
4430 If B$="U"Then Print "I can't e
nter that."
4440 Goto 2100
4450 If B$="D"Then Print "I can't e
nter that."
4460 Goto 2100
4470 If B$="R"Then Print "I can't e
nter that."
4480 Goto 2100
4490 If B$="O"Then Print "I can't e
nter that."
4500 Goto 2100
4510 If B$="P"Then Print "I can't e
nter that."
4520 Goto 2100
4530 If B$="A"Then Print "I can't e
nter that."
4540 Goto 2100
4550 If B$="I"Then Print "I can't e
nter that."
4560 Goto 2100
4570 If B$="L"Then Print "I can't e
nter that."
4580 Goto 2100
4590 If B$="K"Then Print "I can't e
nter that."
4600 Goto 2100
4610 If B$="H"Then Print "I can't e
nter that."
4620 Goto 2100
4630 If B$="G"Then Print "I can't e
nter that."
4640 Goto 2100
4650 If B$="F"Then Print "I can't e
nter that."
4660 Goto 2100
4670 If B$="C"Then Print "I can't e
nter that."
4680 Goto 2100
4690 If B$="V"Then Print "I can't e
nter that."
4700 Goto 2100
4710 If B$="Z"Then Print "I can't e
nter that."
4720 Goto 2100
4730 If B$="X"Then Print "I can't e
nter that."
4740 Goto 2100
4750 If B$="Y"Then Print "I can't e
nter that."
4760 Goto 2100
4770 If B$="Q"Then Print "I can't e
nter that."
4780 Goto 2100
4790 If B$="J"Then Print "I can't e
nter that."
4800 Goto 2100
4810 If B$="M"Then Print "I can't e
nter that."
4820 Goto 2100
4830 If B$="N"Then Print "I can't e
nter that."
4840 Goto 2100
4850 If B$="O"Then Print "I can't e
nter that."
4860 Goto 2100
4870 If B$="P"Then Print "I can't e
nter that."
4880 Goto 2100
4890 If B$="Q"Then Print "I can't e
nter that."
4900 Goto 2100
4910 If B$="R"Then Print "I can't e
nter that."
4920 Goto 2100
4930 If B$="S"Then Print "I can't e
nter that."
4940 Goto 2100
4950 If B$="T"Then Print "I can't e
nter that."
4960 Goto 2100
4970 If B$="U"Then Print "I can't e
nter that."
4980 Goto 2100
4990 If B$="V"Then Print "I can't e
nter that."
5000 Goto 2100
5010 If B$="W"Then Print "I can't e
nter that."
5020 Goto 2100
5030 If B$="X"Then Print "I can't e
nter that."
5040 Goto 2100
5050 If B$="Y"Then Print "I can't e
nter that."
5060 Goto 2100
5070 If B$="Z"Then Print "I can't e
nter that."
5080 Goto 2100
5090 If B$="A"Then Print "I can't e
nter that."
5100 Goto 2100
5110 If B$="B"Then Print "I can't e
nter that."
5120 Goto 2100
5130 If B$="C"Then Print "I can't e
nter that."
5140 Goto 2100
5150 If B$="D"Then Print "I can't e
nter that."
5160 Goto 2100
5170 If B$="E"Then Print "I can't e
nter that."
5180 Goto 2100
5190 If B$="F"Then Print "I can't e
nter that."
5200 Goto 2100
5210 If B$="G"Then Print "I can't e
nter that."
5220 Goto 2100
5230 If B$="H"Then Print "I can't e
nter that."
5240 Goto 2100
5250 If B$="I"Then Print "I can't e
nter that."
5260 Goto 2100
5270 If B$="J"Then Print "I can't e
nter that."
5280 Goto 2100
5290 If B$="K"Then Print "I can't e
nter that."
5300 Goto 2100
5310 If B$="L"Then Print "I can't e
nter that."
5320 Goto 2100
5330 If B$="M"Then Print "I can't e
nter that."
5340 Goto 2100
5350 If B$="N"Then Print "I can't e
nter that."
5360 Goto 2100
5370 If B$="O"Then Print "I can't e
nter that."
5380 Goto 2100
5390 If B$="P"Then Print "I can't e
nter that."
5400 Goto 2100
5410 If B$="Q"Then Print "I can't e
nter that."
5420 Goto 2100
5430 If B$="R"Then Print "I can't e
nter that."
5440 Goto 2100
5450 If B$="S"Then Print "I can't e
nter that."
5460 Goto 2100
5470 If B$="T"Then Print "I can't e
nter that."
5480 Goto 2100
5490 If B$="U"Then Print "I can't e
nter that."
5500 Goto 2100
5510 If B$="V"Then Print "I can't e
nter that."
5520 Goto 2100
5530 If B$="W"Then Print "I can't e
nter that."
5540 Goto 2100
5550 If B$="X"Then Print "I can't e
nter that."
5560 Goto 2100
5570 If B$="Y"Then Print "I can't e
nter that."
5580 Goto 2100
5590 If B$="Z"Then Print "I can't e
nter that."
5600 Goto 2100
5610 If B$="A"Then Print "I can't e
nter that."
5620 Goto 2100
5630 If B$="B"Then Print "I can't e
nter that."
5640 Goto 2100
5650 If B$="C"Then Print "I can't e
nter that."
5660 Goto 2100
5670 If B$="D"Then Print "I can't e
nter that."
5680 Goto 2100
5690 If B$="E"Then Print "I can't e
nter that."
5700 Goto 2100
5710 If B$="F"Then Print "I can't e
nter that."
5720 Goto 2100
5730 If B$="G"Then Print "I can't e
nter that."
5740 Goto 2100
5750 If B$="H"Then Print "I can't e
nter that."
5760 Goto 2100
5770 If B$="I"Then Print "I can't e
nter that."
5780 Goto 2100
5790 If B$="J"Then Print "I can't e
nter that."
5800 Goto 2100
5810 If B$="K"Then Print "I can't e
nter that."
5820 Goto 2100
5830 If B$="L"Then Print "I can't e
nter that."
5840 Goto 2100
5850 If B$="M"Then Print "I can't e
nter that."
5860 Goto 2100
5870 If B$="N"Then Print "I can't e
nter that."
5880 Goto 2100
5890 If B$="O"Then Print "I can't e
nter that."
5900 Goto 2100
5910 If B$="P"Then Print "I can't e
nter that."
5920 Goto 2100
5930 If B$="Q"Then Print "I can't e
nter that."
5940 Goto 2100
5950 If B$="R"Then Print "I can't e
nter that."
5960 Goto 2100
5970 If B$="S"Then Print "I can't e
nter that."
5980 Goto 2100
5990 If B$="T"Then Print "I can't e
nter that."
6000 Goto 2100
6010 If B$="U"Then Print "I can't e
nter that."
6020 Goto 2100
6030 If B$="V"Then Print "I can't e
nter that."
6040 Goto 2100
6050 If B$="W"Then Print "I can't e
nter that."
6060 Goto 2100
6070 If B$="X"Then Print "I can't e
nter that."
6080 Goto 2100
6090 If B$="Y"Then Print "I can't e
nter that."
6100 Goto 2100
6110 If B$="Z"Then Print "I can't e
nter that."
6120 Goto 2100
6130 If B$="A"Then Print "I can't e
nter that."
6140 Goto 2100
6150 If B$="B"Then Print "I can't e
nter that."
6160 Goto 2100
6170 If B$="C"Then Print "I can't e
nter that."
6180 Goto 2100
6190 If B$="D"Then Print "I can't e
nter that."
6200 Goto 2100
6210 If B$="E"Then Print "I can't e
nter that."
6220 Goto 2100
6230 If B$="F"Then Print "I can't e
nter that."
6240 Goto 2100
6250 If B$="G"Then Print "I can't e
nter that."
6260 Goto 2100
6270 If B$="H"Then Print "I can't e
nter that."
6280 Goto 2100
6290 If B$="I"Then Print "I can't e
nter that."
6300 Goto 2100
6310 If B$="J"Then Print "I can't e
nter that."
6320 Goto 2100
6330 If B$="K"Then Print "I can't e
nter that."
6340 Goto 2100
6350 If B$="L"Then Print "I can't e
nter that."
6360 Goto 2100
6370 If B$="M"Then Print "I can't e
nter that."
6380 Goto 2100
6390 If B$="N"Then Print "I can't e
nter that."
6400 Goto 2100
6410 If B$="O"Then Print "I can't e
nter that."
6420 Goto 2100
6430 If B$="P"Then Print "I can't e
nter that."
6440 Goto 2100
6450 If B$="Q"Then Print "I can't e
nter that."
6460 Goto 2100
6470 If B$="R"Then Print "I can't e
nter that."
6480 Goto 2100
6490 If B$="S"Then Print "I can't e
nter that."
6500 Goto 2100
6510 If B$="T"Then Print "I can't e
nter that."
6520 Goto 2100
6530 If B$="U"Then Print "I can't e
nter that."
6540 Goto 2100
6550 If B$="V"Then Print "I can't e
nter that."
6560 Goto 2100
6570 If B$="W"Then Print "I can't e
nter that."
6580 Goto 2100
6590 If B$="X"Then Print "I can't e
nter that."
6600 Goto 2100
6610 If B$="Y"Then Print "I can't e
nter that."
6620 Goto 2100
6630 If B$="Z"Then Print "I can't e
nter that."
6640 Goto 2100
6650 If B$="A"Then Print "I can't e
nter that."
6660 Goto 2100
6670 If B$="B"Then Print "I can't e
nter that."
6680 Goto 2100
6690 If B$="C"Then Print "I can't e
nter that."
6700 Goto 2100
6710 If B$="D"Then Print "I can't e
nter that."
6720 Goto 2100
6730 If B$="E"Then Print "I can't e
nter that."
6740 Goto 2100
6750 If B$="F"Then Print "I can't e
nter that."
6760 Goto 2100
6770 If B$="G"Then Print "I can't e
nter that."
6780 Goto 2100
6790 If B$="H"Then Print "I can't e
nter that."
6800 Goto 2100
6810 If B$="I"Then Print "I can't e
nter that."
6820 Goto 2100
6830 If B$="J"Then Print "I can't e
nter that."
6840 Goto 2100
6850 If B$="K"Then Print "I can't e
nter that."
6860 Goto 2100
6870 If B$="L"Then Print "I can't e
nter that."
6880 Goto 2100
6890 If B$="M"Then Print "I can't e
nter that."
6900 Goto 2100
6910 If B$="N"Then Print "I can't e
nter that."
6920 Goto 2100
6930 If B$="O"Then Print "I can't e
nter that."
6940 Goto 2100
6950 If B$="P"Then Print "I can't e
nter that."
6960 Goto 2100
6970 If B$="Q"Then Print "I can't e
nter that."
6980 Goto 2100
6990 If B$="R"Then Print "I can't e
nter that."
7000 Goto 2100
7010 If B$="S"Then Print "I can't e
nter that."
7020 Goto 2100
7030 If B$="T"Then Print "I can't e
nter that."
7040 Goto 2100
7050 If B$="U"Then Print "I can't e
nter that."
7060 Goto 2100
7070 If B$="V"Then Print "I can't e
nter that."
7080 Goto 2100
7090 If B$="W"Then Print "I can't e
nter that."
7100 Goto 2100
7110 If B$="X"Then Print "I can't e
nter that."
7120 Goto 2100
7130 If B$="Y"Then Print "I can't e
nter that."
7140 Goto 2100
7150 If B$="Z"Then Print "I can't e
nter that."
7160 Goto 2100
7170 If B$="A"Then Print "I can't e
nter that."
7180 Goto 2100
7190 If B$="B"Then Print "I can't e
nter that."
7200 Goto 2100
7210 If B$="C"Then Print "I can't e
nter that."
7220 Goto 2100
7230 If B$="D"Then Print "I can't e
nter that."
7240 Goto 2100
7250 If B$="E"Then Print "I can't e
nter that."
7260 Goto 2100
7270 If B$="F"Then Print "I can't e
nter that."
7280 Goto 2100
7290 If B$="G"Then Print "I can't e
nter that."
7300 Goto 2100
7310 If B$="H"Then Print "I can't e
nter that."
7320 Goto 2100
7330 If B$="I"Then Print "I can't e
nter that."
7340 Goto 2100
7350 If B$="J"Then Print "I can't e
nter that."
7360 Goto 2100
7370 If B$="K"Then Print "I can't e
nter that."
7380 Goto 2100
7390 If B$="L"Then Print "I can't e
nter that."
7400 Goto 2100
7410 If B$="M"Then Print "I can't e
nter that."
7420 Goto 2100
7430 If B$="N"Then Print "I can't e
nter that."
7440 Goto 2100
7450 If B$="O"Then Print "I can't e
nter that."
7460 Goto 2100
7470 If B$="P"Then Print "I can't e
nter that."
7480 Goto 2100
7490 If B$="Q"Then Print "I can't e
nter that."
7500 Goto 2100
7510 If B$="R"Then Print "I can't e
nter that."
7520 Goto 2100
7530 If B$="S"Then Print "I can't e
nter that."
7540 Goto 2100
7550 If B$="T"Then Print "I can't e
nter that."
7560 Goto 2100
7570 If B$="U"Then Print "I can't e
nter that."
7580 Goto 2100
7590 If B$="V"Then Print "I can't e
nter that."
7600 Goto 2100
7610 If B$="W"Then Print "I can't e
nter that."
7620 Goto 2100
7630 If B$="X"Then Print "I can't e
nter that."
7640 Goto 2100
7650 If B$="Y"Then Print "I can't e
nter that."
7660 Goto 2100
7670 If B$="Z"Then Print "I can't e
nter that."
7680 Goto 2100
7690 If B$="A"Then Print "I can't e
nter that."
7700 Goto 2100
7710 If B$="B"Then Print "I can't e
nter that."
7720 Goto 2100
7730 If B$="C"Then Print "I can't e
nter that."
7740 Goto 2100
7750 If B$="D"Then Print "I can't e
nter that."
7760 Goto 2100
7770 If B$="E"Then Print "I can't e
nter that."
7780 Goto 2100
7790 If B$="F"Then Print "I can't e
nter that."
7800 Goto 2100
7810 If B$="G"Then Print "I can't e
nter that."
7820 Goto 2100
7830 If B$="H"Then Print "I can't e
nter that."
7840 Goto 2100
7850 If B$="I"Then Print "I can't e
nter that."
7860 Goto 2100
7870 If B$="J"Then Print "I can't e
nter that."
7880 Goto 2100
7890 If B$="K"Then Print "I can't e
nter that."
7900 Goto 2100
7910 If B$="L"Then Print "I can't e
nter that."
7920 Goto 2100
7930 If B$="M"Then Print "I can't e
nter that."
7940 Goto 2100
7950 If B$="N"Then Print "I can't e
nter that."
7960 Goto 2100
7970 If B$="O"Then Print "I can't e
nter that."
7980 Goto 2100
7990 If B$="P"Then Print "I can't e
nter that."
8000 Goto 2100
8010 If B$="Q"Then Print "I can't e
nter that."
8020 Goto 2100
8030 If B$="R"Then Print "I can't e
nter that."
8040 Goto 2100
8050 If B$="S"Then Print "I can't e
nter that."
8060 Goto 2100
8070 If B$="T"Then Print "I can't e
nter that."
8080 Goto 2100
8090 If B$="U"Then Print "I can't e
nter that."
8100 Goto 2100
8110 If B$="V"Then Print "I can't e
nter that."
8120 Goto 2100
8130 If B$="W"Then Print "I can't e
nter that."
8140 Goto 2100
8150 If B$="X"Then Print "I can't e
nter that."
8160 Goto 2100
8170 If B$="Y"Then Print "I can't e
nter that."
8180 Goto 2100
8190 If B$="Z"Then Print "I can't e
nter that."
8200 Goto 2100
8210 If B$="A"Then Print "I can't e
nter that."
8220 Goto 2100
8230 If B$="B"Then Print "I can't e
nter that."
8240 Goto 2100
8250 If B$="C"Then Print "I can't e
nter that."
8260 Goto 2100
8270 If B$="D"Then Print "I can't e
nter that."
8280 Goto 2100
8290 If B$="E"Then Print "I can't e
nter that."
8300 Goto 2100
8310 If B$="F"Then Print "I can't e
nter that."
8320 Goto 2100
8330 If B$="G"Then Print "I can't e
nter that."
8340 Goto 2100
8350 If B$="H"Then Print "I can't e
nter that."
8360 Goto 2100
8370 If B$="I"Then Print "I can't e
nter that."
8380 Goto 2100
8390 If B$="J"Then Print "I can't e
nter that."
8400 Goto 2100
8410 If B$="K"Then Print "I can't e
nter that."
8420 Goto 2100
8430 If B$="L"Then Print "I can't e
nter that."
8440 Goto 2100
8450 If B$="M"Then Print "I can't e
nter that."
8460 Goto 2100
8470 If B$="N"Then Print "I can't e
nter that."
8480 Goto 2100
8490 If B$="O"Then Print "I can't e
nter that."
8500 Goto 2100
8510 If B$="P"Then Print "I can't e
nter that."
8520 Goto 2100
8530 If B$="Q"Then Print "I can't e
nter that."
8540 Goto 2100
8550 If B$="R"Then Print "I can't e
nter that."
8560 Goto 2100
8570 If B$="S"Then Print "I can't e
nter that."
8580 Goto 2100
8590 If B$="T"Then Print "I can't e
nter that."
8600 Goto 2100
8610 If B$="U"Then Print "I can't e
nter that."
8620 Goto 2100
8630 If B$="V"Then Print "I can't e
nter that."
8640 Goto 2100
8650 If B$="W"Then Print "I can't e
nter that."
8660 Goto 2100
8670 If B$="X"Then Print "I can't e
nter that."
8680 Goto 2100
8690 If B$="Y"Then Print "I can't e
nter that."
8700 Goto 2100
8710 If B$="Z"Then Print "I can't e
nter that."
8720 Goto 2100
8730 If B$="A"Then Print "I can't e
nter that."
8740 Goto 2100
8750 If B$="B"Then Print "I can't e
nter that."
8760 Goto 2100
8770 If B$="C"Then Print "I can't e
nter that."
8780 Goto 2100
8790 If B$="D"Then Print "I can't e
nter that."
8800 Goto 2100
8810 If B$="E"Then Print "I can't e
nter that."
8820 Goto 2100
8830 If B$="F"Then Print "I can't e
nter that."
8840 Goto 2100
8850 If B$="G"Then Print "I can't e
nter that."
8860 Goto 2100
8870 If B$="H"Then Print "I can't e
nter that."
8880 Goto 2100
8890 If B$="I"Then Print "I can't e
nter that."
8900 Goto 2100
8910 If B$="J"Then Print "I can't e
nter that."
8920 Goto 2100
8930 If B$="K"Then Print "I can't e
nter that."
8940 Goto 2100
8950 If B$="L"Then Print "I can't e
nter that."
8960 Goto 2100
8970 If B$="M"Then Print "I can't e
nter that."
8980 Goto 2100
8990 If B$="N"Then Print "I can't e
nter that."
9000 Goto 2100
9010 If B$="O"Then Print "I can't e
nter that."
9020 Goto 2100
9030 If B$="P"Then Print "I can't e
nter that."
9040 Goto 2100
9050 If B$="Q"Then Print "I can't e
nter that."
9060 Goto 2100
9070 If B$="R"Then Print "I can't e
nter that."
9080 Goto 2100
9090 If B$="S"Then Print "I can't e
nter that."
9100 Goto 2100
9110 If B$="T"Then Print "I can't e
nter that."
9120 Goto 2100
9130 If B$="U"Then Print "I can't e
nter that."
9140 Goto 2100
9150 If B$="V"Then Print "I can't e
nter that."
9160 Goto 2100
9170 If B$="W"Then Print "I can't e
nter that."
9180 Goto 2100
9190 If B$="X"Then Print "I can't e
nter that."
9200 Goto 2100
9210 If B$="Y"Then Print "I can't e
nter that."
9220 Goto 2100
9230 If B$="Z"Then Print "I can't e
nter that."
9240 Goto 2100
9250 If B$="A"Then Print "I can't e
nter that."
9260 Goto 2100
9270 If B$="B"Then Print "I can't e
nter that."
9280 Goto 2100
9290 If B$="C"Then Print "I can't e
nter that."
9300 Goto 2100
9310 If B$="D"Then Print "I can't e
nter that."
9320 Goto 2100
9330 If B$="E"Then Print "I can't e
nter that."
9340 Goto 2100
9350 If B$="F"Then Print "I can't e
nter that."
9360 Goto 2100
9370 If B$="G"Then Print "I can't e
nter that."
9380 Goto 2100
9390 If B$="H"Then Print "I can't e
nter that."
9400 Goto 2100
9410 If B$="I"Then Print "I can't e
nter that."
9420 Goto 2100
9430 If B$="J"Then Print "I can't e
nter that."
9440 Goto 2100
9450 If B$="K"Then Print "I can't e
nter that."
9460 Goto 2100
9470 If B$="L"Then Print "I can't e
nter that."
9480 Goto 2100
9490 If B$="M"Then Print "I can't e
nter that."
9500 Goto 2100
9510 If B$="N"Then Print "I can't e
nter that."
9520 Goto 2100
9530 If B$="O"Then Print "I can't e
nter that."
9540 Goto 2100
9550 If B$="P"Then Print "I can't e
nter that."
9560 Goto 2100
9570 If B$="Q"Then Print "I can't e
nter that."
9580 Goto 2100
9590 If B$="R"Then Print "I can't e
nter that."
9600 Goto 2100
9610 If B$="S"Then Print "I can't e
nter that."
9620 Goto 2100
9630 If B$="T"Then Print "I can't e
nter that."
9640 Goto 2100
9650 If B$="U"Then Print "I can't e
nter that."
9660 Goto 2100
9670 If B$="V"Then Print "I can't e
nter that."
9680 Goto 2100
9690 If B$="W"Then Print "I can't e
nter that."
9700 Goto 2100
9710 If B$="X"Then Print "I can't e
nter that."
9720 Goto 2100
9730 If B$="Y"Then Print "I can't e
nter that."
9740 Goto 2100
9750 If B$="Z"Then Print "I can't e
nter that."
9760 Goto 2100
9770 If B$="A"Then Print "I can't e
nter that."
9780 Goto 2100
9790 If B$="B"Then Print "I can't e
nter that."
9800 Goto 2100
9810 If B$="C"Then Print "I can't e
nter that."
9820 Goto 2100
9830 If B$="D"Then Print "I can't e
nter that."
9840 Goto 2100
9850 If B$="E"Then Print "I can't e
nter that."
9860 Goto 2100
9870 If B$="F"Then Print "I can't e
nter that."
9880 Goto 2100
9890 If B$="G"Then Print "I can't e
nter that."
9900 Goto 2100
9910 If B$="H"Then Print "I can't e
nter that."
9920 Goto 2100
9930 If B$="I"Then Print "I can't e
nter that."
9940 Goto 2100
9950 If B$="J"Then Print "I can't e
nter that."
9960 Goto 2100
9970 If B$="K"Then Print "I can't e
nter that."
9980 Goto 2100
9990 If B$="L"Then Print "I can't e
nter that."
10000 Goto 2100

```

```

C$="*"Then Return
19050 Gosub 1000:Print "I have made
a morphostatic coffin."
19060 Print "I also made other ar
rangements."
19070 Rem Can't say exactly what
, or LIST cheating easy
19080 Print "Possibly I will be
revived in the future."
19090 Print "You may follow what
happens in a future."
19100 Print "CHIRONOMYS adventu
re game"
19110 Print "Press E for another
go, or E to end."
19120 Print "You took "YM" sev
er."
19130 Gosub 2100:If IN$(C$)Then R
et
19140 If IN$(C$)Then IN$(C$)Then E
nd
19150 Goto 19120
19160 Rem
19170 Rem
19180 Rem
19190 Rem
19200 Rem ***DESCRIBE CURRENT LO
CATION***
19210 Rem
19220 Rem L$=location
19230 Restore 25000
19240 Read A$:If Left$(A$,1)=L$
Then 20000
19250 If Left$(A$,1)="/"Then Pri
nt ERR:"LOCATION" is not
19260 Goto 20030
19270 If Mid$(A$,Len(A$)-12,1)="/"
Then 20000
19280 Rem *** Insert in C$ the o
bject to see LIGHT ***
19290 C$="*"
19300 Gosub 20410:If C$="*"
Then 20000
19310 Print "I can see NOTHING!"
Return
19320 Print Mid$(A$,2,Len(A$)-1
1)
19330 Print "I can see exits "
19340 For N=1 To 8
19350 If Mid$(A$,Len(A$)-12+NE2,
1)="/"Then 20120
19360 Print Mid$(A$,Len(A$)-12+NE
2,1)
19370 Print Mid$(A$,Len(A$)-12+NE
2,1)
19380 Next N
19390 For N=1 To 8
19400 If Right$(THEIN),1)=L$ T
hen 20172
19410 Print "I can also see"
19420 Print Left$(THEIN),Len(THE
IN)-10)
19430 Print Left$(THEIN),Len(THEIN
)-5,1)
19440 If C$="*"Then 20172
19450 Print "It is filled with "
19460 For J=1 To 8
19470 If Left$(THEIN),1)=L$ Then
Next J:Goto 20172
19480 Print Mid$(THEIN),5,Len(THE
IN)-10)
19490 Next J
19500 For N=1 To 8
19510 If Right$(THEIN),1)=L$ Th
en 20220
19520 If E=0 Then Print "I have
with me"
19530 Print Left$(THEIN),Len(THE
IN)-10)
19540 Next N
19550 For N=1 To 8
19560 If Right$(THEIN),1)=L$ Th
en 20200
19570 If F=0 Then Print "I am we
aring"
19580 Print Left$(THEIN),Len(THE
IN)-10)
19590 Next N:Return
19600 Rem
19610 Rem *** IS OBJECT AT CURRE
NT LOCATION? ***
19620 Rem
19630 Print "Press letter precedi
ng the object"
19640 Gosub 2100
19650 C$=Mid$(THEIN),Len(THEIN)-

```

```

Then Return
20540 Print "It is not here."
20550 Goto 2100
20560 Return
20570 Rem *** Is another object
here? ***
20580 For N=1 To 8
20590 If Right$(THEIN),1)=L$ Th
en 20440
20600 If Right$(THEIN),1)=L$ T
hen 20450
20610 If Left$(THEIN),1)=C$ Then
Return
20620 Next N:Return
20630 Rem
20640 Rem LOCATION DATA
20650 Rem
20660 Rem " " is light or "X"
if dark, and directions.
20670 Data "A The main hall of a
y laboratory. -N S E W U D "
20680 Data "B The library. -N S
E W U D "
20690 Data "C The workshop. -N S
E W U D "
20700 Data "D The pharmacy. -N S
E W U D "
20710 Data "E The garden. -N S E
W U D "
20720 Data "F The cellar. -N S E
W U D "
20730 Data "G A dried well. -N S
E W U D "
20740 Data "H A Store. -N S E W
U D "
20750 Data "I A tunnel. -N S E W
U D "
20760 Data "J The garden lawn. -
N S E W U D "
20770 Data "K A path through flo
wers. -N S E W U D "
20780 Data "L A shrubbery. -N S
E W U D "
20790 Data "M A shed. -N S E W
U D "
20800 Data "N A rose garden. -N
S E W U D "
20810 Data "O A front garden. -N
S E W U D "
20820 Data "P A house. -N S E W
U D "
20830 Data "Q "
20840 Rem
20850 Rem
20860 Rem OBJECT DATA: Object le
tter, description, start loc.
20870 Rem penultimate W indicate
s it can be worn
20880 Rem A B before this indica
tes it can be burned
20890 Rem A X before this, kills
d, and a d destroyed.
20900 Rem A F before this means
it can be filled
20910 Rem any other letter means
it can be emptied,
20920 Rem and the letter is the
contents, unused for fill
20930 Rem the next five are the
ingredients required
20940 Rem to save this item. If
S, fill in with #
20950 Rem If they are all #, the
n it can't be made.
20960 Rem the next letter is an
E if the object is not
20970 Rem used up when making so
ething.
20980 Rem If instead of a W the
penultimate letter 'S' a
20990 Rem letter, it means the o
bject can be entered, and
21000 Rem the location of its in
terior is the letter. If
21010 Rem the letter is an "E" i
t simply means the object
21020 Rem cannot be taken. It ca
n only be taken if the
21030 Rem if a W or a -. The loc
ation W cannot be used for
21040 Rem for the inside of an o
bject.
21050 Data "A" A book. -N S E W
U D "
21060 Data "B" A hand drill with


```

<p>bit. -#####d-C-</p> <p>26030 Data "C" A very large wood on crate. -#####b-d-N"</p> <p>26040 Data "D" A pile of sand. 1 #####d-N"</p> <p>26050 Data "E" A metal drum, 6ft by 3ft dia. -#####d-R"</p> <p>26060 Data "F" Formaldehyde. 144 #####b-d-D"</p> <p>26070 Data "G" Chloroform. -#### b-b-d-D"</p> <p>26080 Data "H" Cement powder. 14 #####b-d-E"</p> <p>26090 Data "I" A coat. -#####b -d-N"</p> <p>26100 Data "J" A box of matches. 1#####b-d-N"</p> <p>26110 Data "K" An experimental b antam. -#####b-d-N"</p> <p>26120 Data "L" Gas cylinders. 14 #####b-d-N"</p> <p>26130 Data "M" A length of rubbe r tubing. (#####b-d-N"</p> <p>26140 Data "N" Gas torches. 1444 #####d-N"</p> <p>26150 Data "O" A welding set. 11. #####b-d -</p> <p>26160 Data "P" A steel sheet. -4 #####b-d -</p> <p>26170 Data "Q" A lid for the dru g. -#####d -</p> <p>26180 Data "R" A morphostatic co ffin -#####d -</p> <p>26190 Data "S" A wall. -##### -d -</p> <p>26200 Data "T" A hole in the wal l. -#####-1 -</p> <p>26210 Data "U" A pickaxe. (##### u---d-R"</p> <p>26220 Data "V" A dry wooden torr n. -#####-d-d"</p> <p>26230 Data "W" Tar oil. (#####u</p>	<p>-bd-R"</p> <p>26240 Data "X" A flaming torch. -#####d -</p> <p>26250 Data "Y" A red bottle -444 #####d-D"</p> <p>26260 Data "Z" A blue bottle -44 #####d-D"</p> <p>26270 Data "A" A person. -##### -d-d-R"</p> <p>26280 Data "C" A coded message. -#####b-d-G"</p> <p>26290 Data "Z" /"</p> <p>26296 Res</p> <p>27000 Res MESSAGES FOR LOOKS</p> <p>27002 Res</p> <p>27010 Data "A" It says that bodie s can be preserved"</p> <p>27020 Data "B" If they are submer ged in formaldehyde."</p> <p>27030 Data "C" It says this messa ge can be converted"</p> <p>27060 Data "C" by a code to be gi ven in a future"</p> <p>27070 Data "C" chronsauts to rev al more' - r'g'd"</p> <p>27080 Data "B" This person is a f riend of mine. I ask"</p> <p>27090 Data "B" that I may be plac ed in the morphostatic"</p> <p>27100 Data "B" coffin when I die, and that it be buried"</p> <p>27110 Data "B" in the well."</p> <p>27120 Data "E" The drug is so +" -----"</p> <p>27130 Data "E" solidly made.            </p> <p>27140 Data "E" that it must            </p> <p>27150 Data "E" last for            </p> <p>27160 Data "E" centuries' &lt;--</p>	<p>27170 Data "C</p> <p>27180 Data "C</p> <p>27190 Data "C</p> <p>27200 Data "C</p> <p>27230 Data "U</p> <p>27240 Data "U</p> <p>27250 Data "U</p> <p>27260 Data "U</p> <p>27270 Data "U</p> <p>27280 Data "U</p> <p>27290 Data "S :          "</p> <p>27300 Data "S</p> <p>27310 Data "S          "</p> <p>27320 Data "S</p> <p>27330 Data "S          "</p> <p>27340 Data "S</p> <p>27350 Data "S          "</p> <p>27360 Data "T</p> <p>27370 Data "T          "</p> <p>27380 Data "T</p> <p>27390 Data "T          "</p> <p>27400 Data "T</p> <p>27410 Data "T          "</p> <p>27420 Data "T</p> <p>27430 Data "T          "</p> <p>27440 Data "T</p> <p>27450 Data "T          "</p> <p>27460 Data "T</p> <p>27470 Data "T          "</p> <p>27480 Data "T</p> <p>27490 Data "T          "</p> <p>27500 Data "T</p> <p>Contact RTL, Westown, Porhstown, Thru, Cornwall for cassette details.</p>
---	---	---

**BLACK DONALD'S GAMES SHOP**

3 Waverley Place  
Aberdeen AB1 1XH  
Tel. 0224-643740

FOR GAMERS  
NORTH OF  
INCHTUTHIL



STOCKISTS OF ALL THE LATEST MICRO ADVENTURER GAMES FOR THE SPECTRUM, COMMODORE 64, DRAGON AND BBC + ALL THE USUAL FANTASY ROLE PLAYING SYSTEMS

**The No. 1\* Football Game**

**Football Manager**  
Designed by Kevin Toms

Some of the features of the game:

- ★ Matches in 3D graphics
- ★ Transfer market ★ Promotion and relegation ★ F.A. Cup matches ★ Injury problems ★ Full league tables ★ Four Divisions



★ Pick your own team for each match. ★ As many seasons as you like ★ Managerial rating ★ 7 skill levels ★ Save game facility.



**addictive**

\* ZX81 Chart  
Home Computing Weekly  
1.8.83 and 1.11.83

Comments about the game from press and our customers:

FOOTBALL MANAGER is the best game I have yet seen on the Spectrum and my personal favourite of all the games on my video... To the ordinary person it is an excellent use of which can be done in the field of computer games... The opening glory of this game is the start sequence of match highlights which show the game being running around a pitch, shooting, celebrating and scoring... It is a computer game but people who control take game sessions of 15 hours or so, which happened in one happy Sunday, will be grateful to know that there is a 'save to tape' option. FOOTBALL MANAGER has everything it could... The original, addictive football, certainly deserves the name. (Gaming World/Practical Computing - August 1983)

From software stockists nationwide, inc.  

Prices: Spectrum 48K £4.95  
256K £5.95

ADDICTIVE Games  
A. SCHACKE (UK)  
SCHAFFHAUSEN, CH

Decide! For urgent stocks send your needed message direct to our address.