

MSX

truuks en tips

deel 7

M. Buysse



MSX

truuks en tips

deel 7

M. Buysse

uitgeverij STARK-TEXEL

postbus 302 - 1794 ZG Oosterend tel. 02223 - 661

CIP-GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

MSX

MSX truuks en tips. – Oosterend: Stark-Textel

Deel 7 / M. Buysse

ISBN 90 6398 789 7

SISO 365.3 UDC 681.3.06

Trefw.: programmeren (computer) / MSX (computer)

1e druk 1986

ISBN 90 6398 789 7

© uitgeverij Stark-Textel, Oosterend Nh.

Niets uit deze uitgave mag worden vermenigvuldigd en/of openbaar gemaakt door middel van druk, fotokopie, microfilm of op welke andere wijze ook, zonder voorafgaande schriftelijke toestemming van de uitgever.

No part of this book may be reproduced in any form, by print, photo-print, microfilm or any other means, without prior written permission from the publisher.

Ondanks alle aan de samenstelling van de tekst bestede zorg kan noch de redaktie noch de uitgever aansprakelijkheid aanvaarden voor eventuele schade die zou kunnen voortvloeien uit enige fout die in deze uitgave zou kunnen voorkomen.

MSX is een handelsmerk van Microsoft.

Inhoudsopgave

	pagina
Inleiding	7
Tekst en scherm	
Kolommen	10
Knipperende woorden	12
Knipperende funktietoetsen	15
Karakters inverteren	17
Grote letters in screen 0	20
Schermbreedte	23
Screendump	25
Insert in string	28
Karakters gespiegeld	30
Peeks en pokes	
Behekste toets	34
Nieuwe resetknop	36
Onzichtbare cursor	37
Vernietigende cursor	38
Time & time	39
Grafische truuks	
Cursorlijnen	44
Gedetailleerd tekenen	48
Beweging-simulatie	53
Lopend mannetje	55
Verdwenen transparant	60
Scherms wisselen	62
Spirograph	65
Utilities	
Veilige karakterset	68
Hernummeraar	70
Speed sector run	73
RAM-disks	75
Supersnelle backup	77
Klokje	81
Hard copy van screen 6	83

Netjes programmeren	85
String manipulatie	87
Listing printer	89

Nieuwe schermmode 94

80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200

Inleiding

Dit alweer zevende deel van de MSX Truiks en tips reeks bevat niet alleen truiks voor de MSX1 computers maar ook voor de onlangs geïntroduceerde MSX2 computers.

Dat vele van deze MSX2 truiks met de grafische mogelijkheden te maken hebben, zal de MSX2 gebruikers onder de lezers niet verwonderen. MSX2 is immers een op grafisch vlak sterk uitgebreide MSX1 computer met (meestal) ingebouwde diskdrive en een iets groter geheugen.

Dat de MSX1 computers daardoor in dit deel van MSX Truiks en tips niet aan bod komen is ook geenszins waar. Het grootste gedeelte van de truiks is te gebruiken voor de MSX1 èn MSX2 computers.

Daar waar machinetaal gebruikt wordt, worden speciaal voor de machinetaalprogrammeur de mnemonics als REM regels in het programma opgenomen. Dit om het geheel overzichtelijk te houden.

Dankwoord

Mijn speciale dank gaat uit naar Chris Lanckriet voor het aanbrengen van enkele zeer interessante ideeën.

Tevens wil ik alle mensen bedanken die mij behulpzaam zijn geweest.

Tekst en scherm

Dat de MSX computers niet alleen in de grafische schermen veel te bieden hebben wordt in het volgende hoofdstuk duidelijk aangetoond. Zo kunt u in de tekstschermen alle karakters, behalve de cursor, veranderen in andere figuurtjes of andere lettertypen.

De volledige karakterset zelf opnieuw ontwerpen is een hele klus, daarvoor kunt u het beste enkele handige routines gebruiken. Van deze soort routines volgen er in dit hoofdstuk twee. De eerste dient om een deel van de karakterset te inverteren en de tweede om enkele hele grote karakters te maken.

Voor de MSX2 gebruikers zijn er enkele routines opgenomen die gebruik maken van de mogelijkheid om stukken tekst te laten knippen.

Verder nog een routine waarmee men het scherm kan versmallen of verbreden zonder dat het scherm gewist wordt.

Kolommen

De PRINT instructie in MSX Basic is redelijk uitgebreid. Zo kan men bijvoorbeeld in kolommen printen. Dit kan eenvoudig gedaan worden door tussen elk stukje dat men print een komma te plaatsen.

In screen 0 met 40 tekens krijgt men dan alles in twee kolommen op het scherm geprint. Nu is twee kolommen wel erg weinig. U kunt dat verhelpen door een eenvoudige Poke-opdracht te maken zodat er drie kolommen op het scherm uitgeprint worden.

In het volgende programmaatje ziet u daar een voorbeeld van:

De programmalijst:

```
10 REM *****
20 REM *      KOLOMMEN      *
30 REM *      -----      *
40 REM *                    *
50 REM * VOOR MSX1 EN MSX2 *
60 REM *****
70 REM
80 COLOR 15,4,4:SCREEN0:WIDTH40
90 PRINT"Dit is normaal:"
100 PRINT"truuks","en","tips"
110 POKE&HF3B2,20:PRINT
120 PRINT"Zo is het na de poke:"
130 PRINT"truuks","en","tips"
```

Bij de MSX2 computers heeft men natuurlijk meer dan twee kolommen. Voorwaarde daarvoor is wel dat u zich in screen 0 bevindt met een scherm breedte van 80 lettertekens. Als dat zo is beschikt u standaard over 5 kolommen. Ook dit kan vermeerderd worden met dezelfde Poke. Alleen het getal dat men in het geheugen moet poken is anders. Het volgende programma geeft een voorbeeld.

De programmalijsjt:

```
10 REM *****
20 REM *   KOLOMMEN   *
30 REM *   -----   *
40 REM *               *
50 REM * VOOR MSX2 *
60 REM *****
70 REM
80 COLOR 15,4,4:SCREEN0:WIDTH80
90 PRINT"Dit is normaal:"
100 PRINT"truuks","en","tips","deel","7"
    ,"MSX2"
110 POKE&HF3B2,61:PRINT
120 PRINT"Zo is het na de poke:"
130 PRINT"truuks","en","tips","deel","7"
    ,"MSX2"
```

Knipperende woorden

Dat de MSX2 computer ook in screen 0 veel meer mogelijkheden heeft dan de MSX1 computer is wel duidelijk. Een van deze mogelijkheden is dat men vier kleuren kan gebruiken: twee voorgrondkleuren en twee achtergrondkleuren. Dit wordt het meest gebruikt om met inverse karakters te kunnen werken. Nu is het nogal omslachtig om bij elk stukje tekst dat u in de alternatieve kleuren wilt zetten te gaan uitzoeken welke geheugenadressen in de video RAM op welke waarde moeten worden gezet.

Het volgende programma maakt al dat zoeken en uitrekenen overbodig. Door enkele variabelen te vullen met de juiste waarden en daarna de subroutine op regel 1000 aan te roepen krijgt u de gewenste tekst op de gewenste plaats. Het is zelfs mogelijk in te geven wanneer de tekst moet knipperen.

De variabelen die gevuld moeten worden met de gewenste waarden zijn de volgende:

KN\$= De gewenste tekst

X = De X-positie van de plaats waar de tekst moet worden geprint

Y = De Y-positie van de plaats waar de tekst moet worden geprint

De X-positie + de lengte van de tekst mag niet groter zijn dan 80.

De programmalijst:

```
10 REM *****
20 REM * KNIPPERENDE WOORDEN *
30 REM * ----- *
40 REM * *
50 REM * VOOR MSX2 *
60 REM *****
70 REM
```

```

80 SCREEN 0:WIDTH 80:COLOR 15,4,4:CLS:KEY
Y OFF
90 FOR I=BASE(1) TO BASE(1)+239
100 VPOKE I,0:NEXT I
110 INPUT "Geef de X-positie"; X
120 IF X>80 THEN 110
130 INPUT "Geef de Y-positie"; Y
140 IF Y>24 THEN 130
150 INPUT "Geef het woord";KN$
160 IF LEN(KN$)+X > 79 THEN 150
170 INPUT "Knippenen j/n";A$
180 IF A$="j" OR A$="J" THEN KN=1: GOTO
200
190 IF A$<>"n" AND A$<>"N" THEN 170 ELSE
KN=0
200 CLS:GOSUB 1050
210 END
1000 REM
1010 REM *****
1020 REM * DE KNIPPER ROUTINE *
1030 REM *****
1040 REM
1050 A$=SPACE$(X)+KN$+SPACE$(80-(X-1+LEN
(KN$)))
1060 FOR I=0 TO 9
1070 KN(I)=0
1080 FOR T=1 TO 8
1090 IF ASC(MID$(A$,T+I*8,1))<>32 THEN K
N(I)=KN(I)+2^(8-T)
1100 NEXT T
1110 NEXT I
1120 FOR I=0 TO 9
1130 VPOKEBASE(1)+(Y*10)+I,VPEEK(BASE(1)
+(Y*10)+I)OR KN(I)
1140 NEXT I
1150 VDP(13)=79

```

```
1160 IF KN=1 THEN VDP(14)=66 ELSE VDP(14
)=&HFO
1170 LOCATE X,Y:PRINTKN$
1180 RETURN
```

```
1190 INPUT "Best de X-position"; X
1200 IF X=0 THEN 110
1210 INPUT "Best de Y-position"; Y
1220 IF Y=0 THEN 110
1230 INPUT "Best del word"; KW
1240 IF LEN(KW) > 75 THEN 120
1250 INPUT "Kopiering 1/n"; N
1260 IF N=0 OR N=1 THEN 1011 GOTO
```

```
1270 IF A=C OR A=D OR A=C+D THEN 120 ELSE
KW=D
200 GOTO 1020
210 END
```

```
1000 REM *****
1010 REM *****
1020 REM * DE KNIPPER ROUTINE *
1030 REM *****
1040 REM
```

```
1050 A=SPACE(X)+KW+SPACE(80-X)+LEN
(KW)
```

```
1060 FOR I=0 TO 9
```

```
1070 K=I+0
```

```
1080 FOR T=1 TO 8
```

```
1090 IF ASC(MID(A,T,1))<=32 THEN K
M1=MID(A,2,8-T)
```

```
1100 NEXT T
```

```
1110 NEXT I
```

```
1120 FOR I=0 TO 9
```

```
1130 W=SPACE(X)+A+I+1+SPACE(80-X)
+C*10+I+OR(KM1)
```

```
1140 NEXT I
```

```
1150 VDP(13)=W
```


Knipperende funktietoetsen

De MSX computer heeft 10 funktietoetsen. De eerste vijf krijgt u door gewoon op de toetsen te drukken, de volgende vijf door op SHIFT en op die toetsen tegelijkertijd te drukken. Als u onderaan het scherm kijkt ziet u de betekenis van deze funktietoetsen.

Als u dat wenst, kunt u door 'KEY OFF' de betekenis van de funktietoetsen onzichtbaar maken. Het weer zichtbaar maken van de funktietoetsen gebeurt door 'KEY ON'.

Met het volgende programma worden de funktietoetsen zeer opvallend gemaakt door ze te laten knipperen.

De programmalijst:

```
10 REM *****
20 REM * KNIPPERENDE FUNKTIETOETSEN *
30 REM * ----- *
40 REM * *
50 REM *          VOOR MSX2          *
60 REM *****
70 REM
80 SCREEN 0:WIDTH 80:COLOR 15,4,4:KEY ON
:CLS
90 VDP(13)=79
100 FOR I=BASE(1) TO BASE(1)+239
110 VPOKE I,0: NEXT I
120 VPOKE BASE(1)+230,248
130 VPOKE BASE(1)+232,240
140 VPOKE BASE(1)+234,240
150 VPOKE BASE(1)+236,240
160 VPOKE BASE(1)+238,224
170 VDP(14)=66
180 REM
190 REM *****
```

```
200 REM * GEWOON INVERS MET: *
210 REM * *
220 REM * VDP(14)=&HFO *
230 REM *****
240 REM
250 REM *****
260 REM * TERUG NORMAAL MET: *
270 REM * *
280 REM * VDP(14)=0 *
290 REM *****
```

Karakters inverteren

De karakterset van de MSX computer is zoals u waarschijnlijk reeds gemerkt zult hebben zeer uitgebreid. Er zijn behalve de gewone cijfers en letters ook nog vreemde lettertekens, tekens met aksenten, grafische tekens enz. Wat echter niet in de normale karakterset te vinden is, zijn de inverse tekens. Hoewel deze niet altijd noodzakelijk zijn is het soms toch gemakkelijk over een karakterset te beschikken met deze inverse tekens.

Het volgende programma verdeelt de karakterset in twee delen, met in het eerste deel 'CHR\$(32)-CHR\$(127)' de normale cijfers en letters en in het tweede deel 'CHR\$(160)-CHR\$(254)' het inverse van het eerste deel. De speciale grafische tekens die men kan oproepen met 'PRINT CHR\$(1)+CHR\$(65-96)' worden eveneens geïverteerd. De nieuwe geïverteerde grafische tekens worden in de karaktercodes 128-159 geplaatst. Om de inverse tekens op te roepen moet er bij de normale karaktercode 128 opgeteld worden. Start eerst het programma en tik daarna:

```
PRINT CHR$(65):PRINT CHR$(65+128) [RETURN]
```

Nu ziet u eerst een gewone hoofdletter 'A' verschijnen met daaronder een inverse hoofdletter 'A'. Als u nu nog eens 32 daarbij optelt krijgt u kleine letters 'a'. Probeer maar:

```
PRINT CHR$(65+32):PRINT CHR$(65+128+32)  
[RETURN]
```

Het omzetten van de karakterset gebeurt in een klein stukje machinetaal zodat dit zeer snel gebeurt. In Basic kan het ook, maar dan duurt het wel iets langer.

De programmalijsjt:

```
10 REM *****
```

```

20 REM *   KARAKTERS INVERTEREN   *
30 REM *   -----                   *
40 REM *                               *
50 REM *   VOOR MSX1 EN MSX2       *
60 REM *                               *
70 REM *****
80 REM
90 COLOR15,4,4:SCREEN0:WIDTH40
100 FOR I=1 TO 255
110 IF I<32 THEN PRINT CHR$(I)+CHR$(I+65
);" ";:NEXT I
120 PRINT CHR$(I);" ";
130 NEXT I
140 FOR I=1 TO 30
150 READ A$
160 KI$=KI$+CHR$(VAL("&H"+A$))
170 NEXT
180 DEFUSR=PEEK(VARPTR(KI$)+1)+PEEK(VARP
TR(KI$)+2)*256
190 DUMMY=USR(0)
200 END
210 REM
220 REM *****
230 REM * MACHINETAAL DATA'S *
240 REM *****
250 REM
260 DATA 21,00,08:' LD HL,2048
270 DATA 01,F8,03:' LD BC,1016
280 DATA 11,00,0C:' LD DE,3072
290 DATA CD,4A,00:' CALL &H4A
300 DATA EE,FF :' XOR 255
310 DATA EB :' EX DE,HL
320 DATA CD,4D,00:' CALL &H4D "LOOP"
330 DATA EB :' EX DE,HL
340 DATA 23 :' INC HL
350 DATA 13 :' INC DE

```

```

360 DATA 0B      : ' DEC  BC
370 DATA AF     : ' XOR  A
380 DATA B0     : ' OR   B
390 DATA 20,EF  : ' JR   NZ, LOOP
400 DATA B1     : ' OR   C
410 DATA 20,EC  : ' JR   NZ, LOOP
420 DATA C9     : ' RET

```

Grote letters in screen 0

In de karakterset van de MSX computers zijn alle karakters van dezelfde grootte. Een mogelijkheid om deze karakters in verschillende formaten op het scherm te krijgen bestaat er niet.

Het volgende programma verandert een gedeelte van de karakterset zodanig dat men door het printen van twee karakters een groot karakter krijgt. De gewone kleine letters en cijfers blijven behouden. De vreemde karkaters zijn echter niet meer te gebruiken. Alleen de hoofdletters en de cijfers kunt u vergroot krijgen. Om ook nog eens de kleine letters vergroot te krijgen is de karakterset helaas niet groot genoeg. Om de juiste karaktercode te weten te komen van het gewenste karakter telt u bij de normale ASCII code 80 op, voor het tweede deel van een karakter telt u bij de normale ASCII code 128 op. Bijvoorbeeld:

```
PRINT CHR$(65) [RETURN]
PRINT CHR$(65+80);CHR$(65+128) [RETURN]
```

Bij het bovenstaande voorbeeld wordt eerst een gewone 'A' op het scherm geprint en daarna wordt er een grote 'A' onder geprint. Opgelet: Als u niet eerst onderstaand programma opgestart heeft werkt dit natuurlijk niet.

Om gemakkelijk volledige woorden of zinnen op het scherm vergroot te printen is er in het programma vanaf regel 1000 nog een kleine subroutine ingebouwd. Als u deze subroutine aanroept moeten er wel enkele variabelen bepaalde gegevens bevatten. Deze variabelen zijn de volgende:

GT\$= De gewenste tekst
X = De X-positie van de plaats waar de tekst moet worden geprint
Y = De Y-positie van de plaats waar de tekst moet worden geprint

In het programma wordt vanaf regel 240 tot regel 380 nog

maals getoond hoe de subroutine gebruikt kan worden. Het programma is gemaakt voor een MSX2 computer maar werkt ook op een MSX1 computer als u regel 80 als volgt verandert:

```
80 SCREEN 0:WIDTH 40:COLOR 15,4,4:CLS:KEY ON
```

Een leuk idee om dit programma te gebruiken is bijvoorbeeld een titel vergroot op het scherm te printen boven een menu of iets dergelijks.

De programmalijsjt:

```
10 REM *****
20 REM * GROTE LETTERS IN SCREEN0 *
30 REM * ----- *
40 REM * *
50 REM * VOOR MSX2 *
60 REM *****
70 REM
80 SCREEN 0:WIDTH 80:COLOR 15,4,4:CLS:KEY ON
90 FOR I=384 TO 767
100 A=VPEEK(BASE(2)+I)
110 B=0
120 IF (A AND 128)=128 THEN B=192
130 IF (A AND 64)=64 THEN B=B+48
140 IF (A AND 32)=32 THEN B=B+12
150 VPOKE BASE(2)+I+640,B
160 B=0
170 IF (A AND 16)=16 THEN B=192
180 IF (A AND 8)=8 THEN B=B+48
190 IF (A AND 4)=4 THEN B=B+12
200 VPOKE BASE(2)+I+1024,B
210 IF I/8=I\8 THEN PRINTCHR$(I/8);CHR$(
(I/8)+80);CHR$((I/8)+128);CHR$(32);
220 NEXT I
```

```

230 CLS
240 X=5:Y=2:GT$="TRUUKS EN TIPS"
250 GOSUB 1000
260 LOCATE 34,2:PRINT"DEEL 7"
270 LOCATE 10,4:PRINT"MET O.A."
280 X=6:Y=6:GT$="GRAFISCHE TRUUKS"
300 GOSUB 1000
310 X=6:Y=8:GT$="PEEKS EN POKES"
320 GOSUB 1000
330 X=6:Y=10:GT$="UTILITIES"
340 GOSUB 1000
350 X=6:Y=12:GT$="TRUUKS IN HET TEKSTSCH
ERM"
360 GOSUB 1000
370 X=6:Y=14:GT$="ENZ"
380 GOSUB 1000
390 END
950 REM
960 REM *****
970 REM * PRINT ROUTINE *
980 REM *****
990 REM
1000 LOCATE X,Y
1010 FOR I=1 TO LEN(GT$)
1020 IF ASC(MID$(GT$,I,1))=32 THEN PRINT
SPC(2);:GOTO 1040
1030 PRINT CHR$(ASC(MID$(GT$,I,1))+80);C
HR$(ASC(MID$(GT$,I,1))+128);
1040 NEXT I
1050 RETURN

```


Schermbreedte

De scherm breedte van de MSX computers is zoals u weet variabel. Bij MSX computers kunt u in screen 0 gaan tot een scherm breedte van 40 lettertekens. Bij de MSX2 computers kunt u, eveneens in screen 0, gaan tot een scherm breedte van 80 lettertekens. In screen 1 kunt u slechts gaan tot een scherm breedte van 32 lettertekens. Het wijzigen van die breedte gebeurt met de instructie 'WIDTH X'. Het grote nadeel van deze instructie is echter dat elke keer als u van scherm breedte verandert, het scherm gewist wordt.

Door rechtstreeks in het systeem RAM te poken kan dit echter voorkomen worden. De geheugenplaats die de waarde van de scherm breedte inhoudt, is adres &HF3B0.

PRINT PEEK(&HF3B0) [RETURN]

Door bovenstaande regel in te tikken krijgt u de tegenwoordige scherm breedte. Om de scherm breedte te veranderen tikt u:

POKE &HF3B0,X [RETURN]

Daarbij staat X voor de gewenste scherm breedte. Als u wilt dat na een screen opdracht deze waarde ook gehandhaafd blijft, moet u dezelfde waarde ook nog eens in de geheugenplaats &HF3AE poken. Bijvoorbeeld:

POKE &HF3AE,X [RETURN]

Voor screen 1 geldt een ander adres. Dan tikt u:

POKE &HF3AF,X [RETURN]

Als u een hogere waarde dan toegestaan voor de betreffende scherm mode in de geheugenplaatsen ingeeft, dan is het mogelijk dat de computer op tilt slaat. Bij de MSX2 computers is het niet mogelijk de scherm breedte te vergroten tot

meer dan 40 lettertekens als de tegenwoordige waarde onder de 41 ligt. Het is ook niet mogelijk de schermbreedte te verkleinen tot minder dan 41 als de tegenwoordige schermbreedte hoger is dan 40 lettertekens.

Het volgende programma toont nogmaals hoe u de schermbreedte gemakkelijk kunt aanpassen zonder het scherm te wissen.

De programmalijsjt:

```
10 REM *****
20 REM *   SCHERMBREEDTE   *
30 REM *   -----   *
40 REM *   *   *
50 REM * VOOR MSX1 EN MSX2 *
60 REM *****
70 REM
80 INPUT "Nieuwe schermbreedte";B
90 IF B>255 THEN 80
100 PRINT"Instellen vastleggen zodat het
na een screen opdracht behouden blijft
(j/n)"
110 I$=INKEY$:IF I$="" THEN 110
120 IF I$="j" OR I$="J" THEN IF PEEK(&HF
CAF)=0 THEN POKE&HF3AE,B:GOTO 140 ELSE P
OKE&HF3AF,B:GOTO 140
130 IF I$<>"N" AND I$<>"n" THEN 110
140 POKE&HF3B0,B
150 PRINT"Klaar!"
```

Screendump

Als men het heeft over een screendump denkt men direkt aan het uitprinten van alle gegevens die op het scherm staan op papier. Dat men ook een screendump op disk of naar een array kan maken is iets waar velen nog niet aan gedacht zullen hebben. Het kan toch zeer interessant zijn om een scherm tijdelijk op te slaan om het daarna als men het weer nodig heeft terug op het scherm te plaatsen.

In het eerste programma staan twee subroutines die in uw eigen programma ingebouwd kunnen worden. De eerste die begint op regel 1000 maakt een screendump op disk. De tweede subroutine begint op regel 2000 en zet het zojuist op disk geplaatste beeld terug op het scherm. De filename die gebruikt wordt kan eventueel aangepast worden.

De screendump die gemaakt wordt is een screendump van screen 0 WIDTH 40. Indien u een screendump van screen 1 wenst moeten de volgende regels als volgt veranderd worden:

```
1010 FOR I= 6144 TO 6911
2020 FOR I= 6144 TO 6911
```

Indien u werkt met een MSX2 computer en u wenst een screendump van SCREEN 0 WIDTH 80 dan moeten de volgende regels als volgt veranderd worden:

```
1010 FOR I= 0 TO 1919
2010 FOR I= 0 TO 1919
```

De programmalijsjt:

```
10 REM *****
20 REM * SCREENDUMP NAAR DISK *
30 REM * ----- *
40 REM * *
50 REM * VOOR MSX1 EN MSX2 *
```

Indien u in het bezit bent van een MSX2 computer en u wenst een screendump te maken van SCREEN 0 WIDTH 80 dan moeten de volgende regels als volgt aangepast worden:

```
1000 DIM SC(1919)
1010 FOR I= 0 TO 1919
2000 FOR I= 0 TO 1919
```

De programmalijsjt:

```
10 REM *****
20 REM * SCREENDUMP NAAR ARRAY *
30 REM * ----- *
40 REM * *
50 REM * VOOR MSX1 EN MSX2 *
60 REM *****
70 REM
996 REM *****
997 REM * VAN SCHERM NAAR ARRAY *
998 REM *****
999 REM
1000 DIM SC(959)
1010 FOR I=0 TO 959
1020 SC(I)=VPEEK(I)
1030 NEXT I
1040 RETURN
1995 REM
1996 REM *****
1997 REM * VAN ARRAY NAAR SCHERM *
1998 REM *****
1999 REM
2000 FOR I=0 TO 959
2010 VPOKE I,SC(I)
2020 NEXT I
2030 RETURN
```

Insert in string

Om strings te manipuleren zijn er in de MSX Basic een heleboel instructies. Zo kunt u onder andere met MID\$ een stuk in een string veranderen. Om een stuk tekst in een string te voegen bestaat er echter geen instructie.

In het volgende programma wordt met DEF FN een instructie gemaakt die een string in een andere string plaatst vanaf het gewenste karakter.

Indien u bij het ingeven van bijvoorbeeld het in te voegen woord een spatie voor of achteraan het woord wilt geeft u het woord (tussen aanhalingstekens) als volgt in:

run

Geef de zin? De computers

Geef het woord dat tussen gevoegd moet worden? "MSX "

Geef de plaats? 4

De MSX computers

Ok

In het bovenstaand voorbeeld wordt het programma gestart waarna eerst een kleine zin ingegeven wordt en daarna het woord MSX tussen aanhalingstekens met een spatie er achter moet worden ingeven. Zoals u ziet worden de aanhalingstekens niet in de string opgenomen maar de spatie er achteraan wordt dank zij de aanhalingstekens wel in de string opgenomen.

De programmalijst:

```
10 REM *****
20 REM * INSERT IN STRING *
30 REM * ----- *
40 REM * *
50 REM * VOOR MSX1 EN MSX2 *
```

```
60 REM *****
70 REM
80 SCREEN 0:COLOR 15,4,4:CLS:KEY OFF:WIDTH 40
90 DEF FNINSERT$(A$,B$,X)=LEFT$(A$,X)+B$+RIGHT$(A$,LEN(A$)-X)
100 INPUT"Geef de zin";Z$
110 INPUT"Geef het woord dat tussengevoegd moet worden";W$
120 INPUT"Geef de plaats";P
130 PRINT FNINSERT$(Z$,W$,P)
140 END
```

Karakters gespiegeld

Het volgende programma is weer een mooi voorbeeldje van een manier om de karakterset op eenvoudige wijze volledig te veranderen.

Na het opstarten van het programma wordt de volledige karakterset gespiegeld. Het programma loopt alleen goed in screen1 omdat in screen 0 nooit het volledige karakter op het scherm geplaatst wordt, doch slecht de hoogste zes bits. Om weer om te schakelen naar de gewone karakterset geeft u eenvoudig een screen opdracht.

De programmalijs:

```
10 ' *****
20 ' * KARAKTERS GESPIEGELD *
30 ' * ----- *
40 ' * *
50 ' * VOOR MSX 1 EN MSX 2 *
60 ' *****
70 '
80 SCREEN 1:KEY OFF:WIDTH 32:COLOR 15,4,
4
90 FOR I=0 TO 255
100 IF I<32 THEN PRINT CHR$(I)+CHR$(I+64
); ELSE PRINTCHR$(I);
110 NEXT I
120 PRINT:PRINT:PRINT"Even geduld a.u.b.
"
130 '
140 ' *****
150 ' * IN SPIEGELBEELD PLAATSEN *
160 ' *****
170 '
180 FOR I=VDP(4)*2048 TO VDP(4)*2048+204
7
```

```
190 A$=BIN$(VPEEK(I)):A$=STRING$(8-LEN(A
$, "0")+A$
200 B$=""
210 FOR T=1 TO 8
220 B$=MID$(A$,T,1)+B$
230 NEXT T
240 VPOKE I,VAL("&B"+B$)
250 NEXT I
260 END
```


Peeks en pokes

Dat er met het Poke kommando veel gedaan kan worden is reeds duidelijk gedemonstreerd in de vorige delen van Truuks en tips.

Hier volgen nog enkele voorbeelden. Wat dacht u bijvoorbeeld van de cursor die niet meer te zien is, en toch nog werkt, of van de tijd die op onverklaarbare wijze stopgezet wordt of dubbel zo snel gaat? En wat van één toets die bij het indrukken een beepje geeft?

Deze en enkele andere Poke truuks vindt u op de volgende bladzijden.

Behekste toets

Na het opstarten van het volgende programma lijkt het wel alsof de hoofdletter toets 'A' behekst is. Elke keer als u de toets indrukt zal er namelijk een beepje klinken. U kunt ook maken dat er bij het indrukken van een andere toets een beepje klinkt. Dit kan eenvoudig door in regel 80 het getal 65 dat de ASCII kode is voor hoofdletter 'A' in een andere ASCII kode te veranderen. Bijvoorbeeld:

```
80 POKE64962!,&HFE:POKE64963!,66
```

Als het programma zo opgestart wordt zal bij elke aanslag van de hoofdletter 'B' een beepje klinken. Op deze manier kunt u elke toets daarvoor kiezen. U kunt dit programma natuurlijk ook in een zelfgemaakt programma inbouwen. Zo kunt u bijvoorbeeld een kaartenbakprogramma maken dat elke keer als u naar het hoofdmenu springt door op een toets te drukken, een beepje laat horen.

Opgelet: Dit programma kan niet gebruikt worden in combinatie met het programma 'NIEUWE RESETKNOP' omdat het dezelfde POKE-adressen gebruikt.

De programmalijsjt:

```
10 REM *****
20 REM *   BEHEKSTE TOETS   *
30 REM *   -----         *
40 REM *                   *
50 REM * VOOR MSX1 EN MSX2 *
60 REM *****
70 REM
80 POKE64962!,&HFE:POKE64963!,65 ' ASCII
   CODE VAN DE GEWENSTE TOETS
90 POKE64964!,&HCA:POKE64965!,&HCO:POKE6
   4966!,0
```

```
100 REM
110 REM *****
120 REM *   UITZETTEN MET: *
130 REM *                                     *
140 REM *   POKE64962,&HC9 *
150 REM *****
```

Nieuwe resetknop

Veel van de tot nu toe uitgebrachte MSX computers hebben geen resetknopje. Dit komt doordat dit niet voorgeschreven is in de MSX standaard. Als u nu over zo'n MSX computer beschikt bestaat er wel een oplossing voor. U tikt heel eenvoudig het volgende programmaatje in en start het op. Nu zal de computer zich zelf resetten als u op de hoofdletter toets 'A' drukt. Nu wordt de hoofdletter 'A' nogal veel gebruikt, daarom is het beter regel 80 iets te wijzigen. Het is het gemakkelijkst om het getal 65 te veranderen in de ASCII kode van een karakter dat u nooit gebruikt. Bijvoorbeeld een van de grafische karakters. Als u nu de toets indrukt van het karakter dat u gekozen heeft, zal de computer zichzelf resetten.

De programmalijsjt:

```
10 REM *****
20 REM *   NIEUWE RESETKNOP   *
30 REM *   -----           *
40 REM *                       *
50 REM * VOOR MSX1 EN MSX2 *
60 REM *****
70 REM
80 POKE64962!,&HFE:POKE64963!,65 ' ASCII
   CODE VAN DE GEWENSTE TOETS
90 POKE64964!,&HCC:POKE64965!,0:POKE6496
   6!,0
100 REM
110 REM *****
120 REM *   UITZETTEN MET:   *
130 REM *                       *
140 REM *   POKE64962,&HC9  *
150 REM *****
```

Onzichtbare cursor

Zoals u wel zult weten is de cursor van de MSX computers een wit blokje dat de grootte heeft van een letterteken. Dit is zeer gemakkelijk daar u anders steeds zou moeten onthouden op welke positie het volgende karakter zal verschijnen. Nu zou het kunnen gebeuren dat u dit witte blokje beu bent; daarom het volgende programma. Dit programma maakt de cursor onzichtbaar.

De programmalijst:

```
10 REM *****
20 REM *   ONZICHTBARE CURSOR   *
30 REM *   -----               *
40 REM *                               *
50 REM *   VOOR MSX1 EN MSX2   *
60 REM *****
70 REM
80 POKE&HFDA9,&H33:POKE&HFDA A,&H33
90 POKE&HFDAE,&H33:POKE&HFDAF,&H33
100 REM
110 REM *****
120 REM * TERUG ZICHTBAAR MET: *
130 REM *                               *
140 REM *   POKE&HFDA9,&HC9   *
150 REM *   POKE&HFDAE,&HC9   *
160 REM *****
```

Vernietigende cursor

Het volgende programma maakt van de cursor een gevaarlijke vernietigende slang. Elke keer als de cursor van plaats verandert blijft hij namelijk staan en verschijnt hij op de nieuwe plaats.

De programmalijsjt:

```
10 REM *****
20 REM *   VERNIETIGENDE CURSOR   *
30 REM *   -----               *
40 REM *
50 REM *   VOOR MSX1 EN MSX2   *
60 REM *****
70 REM
80 POKE&HFDAE,&H33:POKE&HFDAF,&H33
90 REM
100 REM *****
110 REM * TERUG NORMAAL MET: *
120 REM *
130 REM *   POKE&HFDAE,&HC9   *
140 REM *****
```

Time & time

In de MSX computer is een speciale variabele gebouwd die 'TIME' heet. Hij wordt zo genoemd omdat hij bij elke interrupt van de Z80 met 1 verhoogd wordt. Dit gebeurt 50 maal per seconde. Zodoende wordt 'TIME' veel gebruikt om de tijd van iets te meten. Men kan het ook gebruiken om een wachtlus in een programma in te bouwen. Bijvoorbeeld:

```
100 TIME = 0
110 IF TIME < 500 THEN 110
```

Als u het bovenstaande voorbeeldje in een programma inbouwt zal de computer, elke keer als hij aan regel 110 komt, 10 seconden wachten met het verder uitvoeren van het programma.

De 'TIME' variabele is echter geen echte klok. Het optellen van de 'TIME' gebeurt slechts tot die een waarde van 65535 heeft. Daarna zal hij opnieuw beginnen met 0. Dit wil dus zeggen dat een wachtlus zoals hierboven maximum 131 seconden ofwel in ruim twee minuten kan worden doorlopen.

Nog een zeer nuttig gebruik van de variabele 'TIME' is de instructie 'ON INTERVAL GOSUB'. Met deze instructie is het mogelijk een bepaalde subroutine aan te roepen om de X seconden. Bijvoorbeeld:

```
10 ON INTERVAL=500 GOSUB 200
20 INTERVAL ON
30 PRINT "WACHT EVEN!"
40 GOTO 30
200 INTERVAL OFF
210 PRINT "TIEN SECONDEN ZIJN VOORBIJ"
220 RETURN 20
```

Het bovenstaande voorbeeldje laat zien hoe u deze instructie kunt gebruiken. Het is evenwel niet mogelijk deze periode groter te maken dan 131 seconden.

De derde mogelijkheid om de variabele 'TIME' te gebruiken is om echte random getallen te genereren. Dit doet u dan door de volgende regel vooran in uw programma op te nemen:

```
10 T=RND(-TIME)
```

Na het uitvoeren van deze regel zullen alle getallen die in een programma gebruikt worden echter random (willekeurige) getallen zijn.

Dit zijn zo de mogelijkheden die u heeft met de variabele 'TIME'.

Het volgende programma maakt dat de variabele 'TIME' niet meer verhoogd wordt. Hij blijft konstant op de waarde 1. Dit is erg leuk als u dit opstart voordat u een spelletje inleest dat met 'TIME' werkt. U heeft dan oneindig veel tijd om elk speelveld door te komen en krijgt zodoende een grote bonus. Het werkt niet bij alle spelletjes maar wel bij vele.

De programmalijsjt:

```
10 REM *****
20 REM *   BEVROREN TIJD   *
30 REM *   -----   *
40 REM *   *   *
50 REM * VOOR MSX1 EN MSX2 *
60 REM *****
70 REM
80 POKE&HFC9F,0
90 POKE&HFD9B,&H32:POKE&HFD9C,&H9E:POKE&
HFD9D,&HFC
100 POKE&HFD9A,&HAF
110 REM
120 REM *****
130 REM * TERUG LOPEND MET *
140 REM *   *   *
```

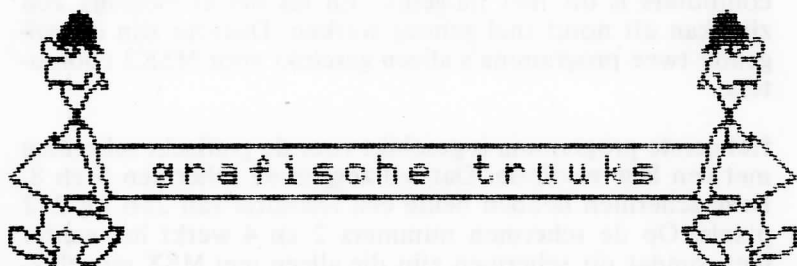
```
150 REM * POKE&HFD9A,&HC9 *
160 REM *****
```

Het volgende programma doet de tijd dubbel zo snel gaan. Dit geeft dus juist het tegenovergestelde effect als het voorgaande programma.

De programmalijst:

```
10 REM *****
20 REM * VERDUBBELDE TIJD *
30 REM * ----- *
40 REM * *
50 REM * VOOR MSX1 EN MSX2 *
60 REM *****
70 REM
80 CLEAR200,&HE200
90 POKE&HE200,&H2A:POKE&HE201,&H9E:POKE&
HE202,&HFC:POKE&HE203,&H23
100 POKE&HE204,&H22:POKE&HE205,&H9E:POKE
&HE206,&HFC:POKE&HE207,&HC9
110 POKE&HFD9B,0:POKE&HFD9C,&HE2
120 POKE&HFD9A,&HCD
130 REM
140 REM *****
150 REM * TERUG TRAG MET *
160 REM * *
170 REM * POKE&HFD9A,&HC9 *
180 REM *****
```


Grafische truuks



Het grote verschil tussen de MSX2 en MSX1 computers is in feite dat de MSX2 computers nog meer grafische mogelijkheden hebben en een groter geheugen. Daardoor zijn de meeste grafische truuks in dit deeltje voor MSX2.

Met het eerste programma wordt meteen een heel krachtige routine gegeven om met een lijncursor te werken, de lijncursor die men veel ziet in de grafische pakketten voor de personal computers. Het volgende programma geeft u de mogelijkheid om zeer eenvoudige en zeer gedetailleerde tekeningen te maken. Daarna wordt er voor wat beweging gezorgd met de programma's *beweging simulatie* en *lopend mannetje*. Deze programma's geven enkele zeer eenvoudige maar krachtige technieken om bewegende beelden te simuleren. Het programma dat daarop volgt geeft u de mogelijkheid een kleur meer te gebruiken.

Het laatste programma *spirograph* is ook geschikt voor MSX computer en toont dat men niet altijd kleur moet gebruiken om een mooie tekening te maken.

Cursor-lijnen

Als u naar de tekenprogramma's van grote computers kijkt ziet u daar vaak dat de cursor voorgesteld wordt als het kruispunt van twee lijnen die over het scherm lopen. Daar de MSX2 computers ook geen kleine computers meer zijn, is het ook zeer goed mogelijk dit effect te bereiken en in zelf gemaakte programma's in te bouwen. Voor de MSX computers is dit niet mogelijk. En als het al mogelijk zou zijn kan dit nooit snel genoeg werken. Daarom zijn de volgende twee programma's alleen geschikt voor MSX2 computers.

Het eerste programma is geschikt voor de grafische schermen met een lage resolutie. Dat wil zeggen de schermen 5 en 8. Deze schermen hebben beide een resolutie van 256 bij 212 pixels. Op de schermen nummers 2 en 4 werkt het echter niet, omdat dit schermen zijn die alleen met MSX mogelijkheden werken.

De eigenlijke lijnen worden gemaakt in een subroutine vanaf regel 290. Deze subroutine moet steeds als input parameter de X en de Y van de cursorpositie krijgen. De arrays die aangemaakt worden zijn zeer belangrijk, evenals de regels 170 en 180. Het overige van het programma dient alleen om het geheel direct te kunnen laten lopen. Dit mag dan ook vrij veranderd worden zoals u dit zelf wenst.

De programmalijst:

```
10 REM *****
20 REM *   CURSORLIJNEN   *
30 REM *   -----   *
40 REM *
50 REM *   VOOR   MSX2   *
60 REM *   lage resolutie *
70 REM *****
80 REM
```

```

90 R=RND(-TIME)
100 COLOR15,4,4:SCREEN8
110 C=15:X=0:Y=0:X1=0:Y1=0
120 DEFINTX,Y
130 DIM XB(130),YB(110)
140 FOR I=1 TO 100
150 LINE(RND(1)*255,RND(1)*211)-(RND(1)*
255,RND(1)*211),RND(1)*255
160 NEXT I
170 COPY (X,0)-(X,211) TO YB
180 COPY (0,Y)-(255,Y) TO XB
190 ON STICK(0) GOSUB 210,220,230,240,25
0,260,270,280
200 GOTO190
210 Y=Y-2:GOTO290
220 Y=Y-2:X=X+3:GOTO290
230 X=X+3:GOTO290
240 Y=Y+2:X=X+3:GOTO290
250 Y=Y+2:GOTO290
260 Y=Y+2:X=X-3:GOTO290
270 X=X-3:GOTO290
280 Y=Y-2:X=X-3
290 COPY YB TO (X1,0)
300 COPY XB TO (0,Y1)
310 COPY (X,0)-(X,211) TO YB
320 COPY (0,Y)-(255,Y) TO XB
330 LINE(0,Y)-(255,Y),C
340 LINE(X,0)-(X,211),C
350 X1=X:Y1=Y
360 RETURN

```

Het tweede programma is geschikt voor de grafische schermen met een hoge resolutie. Dit wil zeggen de schermen nummers 6 en 7. Deze schermen hebben beiden een resolutie van 512 bij 212 pixels.

De eigenlijke lijnen worden gemaakt in een subroutine vanaf regel 290. Deze subroutine moet steeds als input parameter

de X en de Y van de cursor positie krijgen. De arrays zijn zeer belangrijk, evenals de regels 170 en 180. Indien u in schermmode 6 werkt mag regel 130 als volgt aangepast worden:

```
130 DIM XB(130),YB(55)
```

Het overige van het programma dient alleen om het geheel direkt te kunnen laten lopen. Dit mag dan ook veranderd worden zoals u dit zelf wenst.

De programmalijsjt:

```
10 REM *****
20 REM *   CURSORLIJNEN   *
30 REM *   -----   *
40 REM *   *   *
50 REM *   VOOR MSX2   *
60 REM *   hoge resolutie *
70 REM *****
80 REM
90 R=RND(-TIME)
100 COLOR15,4,4:SCREEN7
110 C=15:X=0:Y=0:X1=0:Y1=0
120 DEFINTX,Y
130 DIM XB(260),YB(110)
140 FOR I=1 TO 100
150 LINE(RND(1)*511,RND(1)*211)-(RND(1)*
511,RND(1)*211),RND(1)*15
160 NEXT I
170 COPY (X,0)-(X,211) TO YB
180 COPY (0,Y)-(511,Y) TO XB
190 ON STICK(0) GOSUB 210,220,230,240,25
0,260,270,280
200 GOTO190
210 Y=Y-2:GOTO290
220 Y=Y-2:X=X+3:GOTO290
```

```
230 X=X+3:GOTO290
240 Y=Y+2:X=X+3:GOTO290
250 Y=Y+2:GOTO290
260 Y=Y+2:X=X-3:GOTO290
270 X=X-3:GOTO290
280 Y=Y-2:X=X-3
290 COPY YB TO (X1,0)
300 COPY XB TO (0,Y1)
310 COPY (X,0)-(X,211) TO YB
320 COPY (0,Y)-(511,Y) TO XB
330 LINE(0,Y)-(511,Y),C
340 LINE(X,0)-(X,211),C
350 X1=X:Y1=Y
360 RETURN
```


Gedetailleerd tekenen

Om tekeningen te maken in de grafische schermen zijn er een heleboel instructies zoals 'LINE', 'PSET', 'CIRCLE', 'DRAW' en nog enkele anderen. Dit werkt zeer handig maar als u de programmalijst bekijkt van een programma dat een zeer gedetailleerde tekening op het scherm doet verschijnen krijgt u al snel een erg onoverzichtelijk geheel te zien. Een hele ramp wordt het als er nog iets aan de tekening veranderd moet worden.

Daarom de volgende routine. De tekening geeft u in door middel van dataregels. Deze dataregels bevatten allemaal cijfers die elk op zichzelf 1 pixel van het beeldscherm voorstellen. Door de subroutine op regel 1000 aan te roepen worden die data-regels volledig vertaald in een tekening en op het scherm gezet.

Er moeten wel enkele variabelen gevuld worden met bepaalde gewenste waarden. Die variabelen zijn:

- X = X-positie van de linker bovenhoek vanwaar u de tekening wenst te beginnen
- Y = Y-positie van de linker bovenhoek waar u de tekening wenst te beginnen
- XL = Lengte van de dataregels
- YL = Aantal dataregels
- R = Richting (0 = gewoon, 1 = omgekeerd, 2 = gespiegeld, 3 = omgekeerd gespiegeld)
- K(1) tot K(9) = De gewenste kleur die voor elk cijfer moet staan

In het voorbeeld wordt slechts gebruik gemaakt van 4 kleuren. In regel 90 worden de kleuren bepaald die voor elk nummer staan. Op regel 140 wordt er door middel van de instructie 'RESTORE' voor gezorgd dat de data's nogmaals gelezen kunnen worden. Zodoende komt de tekening twee maal op het scherm. De tweede maal wordt de tekening gespiegeld weergegeven.

De nul in de dataregels betekent dat er op deze plaats niets getekend moet worden. De achtergrond blijft dus intact.

De programmalijsjt:

```
10 REM *****  
20 REM * GEDETAILEERD TEKENEN *  
30 REM * ----- *  
40 REM * *  
50 REM * VOOR MSX2 *  
60 REM *****  
70 REM
```


Beweging-simulatie

Als u een spelletje wilt programmeren heeft u daar meestal graag zo veel mogelijk beweging in. Die beweging wordt meestal verzorgd door sprites. Het nadeel van sprites is echter dat ze zo klein zijn. Daardoor kunt u ze niet gebruiken om stukken achtergrond te doen bewegen, of men zou een heleboel sprites moeten gebruiken, en om die allemaal tegelijk te laten bewegen is Basic niet snel genoeg. Daarom moet u een andere manier gebruiken om stukken achtergrond te doen bewegen of toch de illusie te geven dat ze bewegen. Nu is het niet nodig om daarvoor steeds zeer ingewikkelde machinetaal routines te maken. In Basic kunt u ook al veel bereiken als u de goede methoden gebruikt op de juiste plaats.

Een van die methoden wordt in het volgende programma gedemonstreerd. Daar krijgt men de indruk dat de verschillende onderdelen van de tekening bewegen. Dit wordt bereikt door drie kleuren in het kleurenpalet steeds van waarden te laten verwisselen.

Het eigenlijke omwisselen van de kleuren gebeurt vanaf regelnummer 360. Een voorwaarde om deze routine te gebruiken is wel dat de kleuren die gebruikt worden om de beweging te simuleren verder niet gebruikt worden in de tekening op het scherm. Doet u dit toch dan kunt u zeer eigenaardige effecten krijgen.

De programmalijst:

```
10 REM *****
20 REM * BEWEGING-SIMULATIE *
30 REM * ----- *
40 REM * *
50 REM *      VOOR MSX2      *
60 REM *****
70 REM
80 SCREEN7:COLOR 13,1,1:CLS
```

```

90 A$="F1 R2 F1 R2 F1 R2 F1 R2 F1 R2 F1
R2 G8 ND3 H1 L2 H1 L2 H1 L2 H1 L2 H1 L2
H1 L2 ND3 E8"
100 X=450:K=6:K1=2:K2=4
110 FOR Y=0 TO 200 STEP11
120 X=X-8:SWAPK,K1:SWAPK1,K2
130 PSET(X,Y)
140 DRAW "C=K;" +A$
150 PAINT(X+2,Y+2),K
160 NEXT Y
170 OPEN"GRP:" AS #1
180 PRESET(35,25):PRINT#1,"BEWEGING-SIMU
LATIE"
190 PRESET(35,33)
200 FOR I=1 TO 18
210 SWAPK,K1:SWAPK1,K2
220 COLOR K:PRINT#1,CHR$(207);
230 NEXT
240 CLOSE #1
250 FOR I=1 TO 12
260 SWAPK,K1:SWAPK1,K2
270 FOR T=5 TO 60 STEP3.5
280 CIRCLE(150,130),T,K,(I-1)*(6.283185#
/12),I*(6.283185#/12)
290 NEXT T
300 NEXT I
310 REM
320 REM *****
330 REM * DE EIGENLIJKE ROUTINE *
340 REM *****
350 REM
360 COLOR=(K,1,6,1)
370 COLOR=(K1,1,1,7)
380 COLOR=(K2,5,1,1)
390 SWAPK,K1:SWAPK1,K2
400 FORI=1TO50:NEXTI
410 GOTO 360

```

Lopend mannetje

Gewoonlijk als u een bewegend figuurtje wilt maken gebruikt u daarvoor sprites. Bij grote figuren die moeten bewegen komt u echter al gauw tot de ontdekking dat dit niet zo eenvoudig is. U moet dan al beginnen met een heleboel sprites naast elkaar te plaatsen. Als u die dan laat lopen komt u tot de ontdekking dat het lijkt alsof het figuurtje door elkaar geschud wordt. In MSX2 Basic bestaat er een instructie die veel oplost. Die instructie, 'COPY', maakt het mogelijk bepaalde delen van het scherm te kopiëren naar andere delen van het scherm of naar een array.

Het volgende programma gebruikt de instructie 'COPY' om een redelijk groot figuurtje over het scherm te laten lopen. De wandel-routines staan vanaf regel 800 in het programma. Door deze routines aan te passen en zelf een figuurtje te ontwerpen kunt u zelf bijvoorbeeld een klein tekenfilmpje maken.

De programmalijsjt:

```
10 REM *****
20 REM * LOPEND MANNETJE *
30 REM * ----- *
40 REM * *
50 REM * VOOR MSX2 *
60 REM *****
70 REM
80 COLOR15,0,0:SCREEN5
90 DEFINT A-Z
100 DIM D(413),T(413),L(384),H(564)
110 X=100:Y=100
120 SETPAGE0,1:CLS
130 REM
140 REM *****
```



```

150 REM * BEENTJES OPEN *
160 REM *****
170 REM
180 LINE(100,0)-(111,0),15
190 PSET(100,0)
200 DRAW"C15 D1 G1 D3 NF1 L1 G2 D2 L2 D2
  F1 D1 F3 D1"
210 DRAW"R2 E1 U2 H2 E1 R1 E4 F2 R2 D3 G
  1 F1 R1 NE1 D3"
220 DRAW"R1 E1 R1 E1 R3 U1 E1 U2 H1 L2 G
  2 L1 U3 R1 U3 H1 U1 H1 U2"
230 PAINT(105,3),15
240 PAINT(116,11),15
250 REM
260 REM *****
270 REM * BEENTJES DICT *
280 REM *****
290 REM
300 LINE(50,0)-(61,0),15
310 PSET(50,0)
320 DRAW"C15 G1 D1 F1 D1 F1 D1 F1 D4 L1
  D1 F1 D1 G1 D2"
330 DRAW"R3 U1 R1 F1 R6 E1 U1 H1 L2 H1 L
  1 U2 E1 U3 E2 U4"
340 PAINT(55,3),15
350 REM
360 REM *****
370 REM * HOEDJE *
380 REM *****
390 REM
400 PSET(X+5,Y+8)
410 DRAW"C2 R3 E1 R2 E1 R3 U2 H1 U1 H3 L
  2 G2 D1 G1 D4"
420 PAINT(X+8,Y+5),3,2
430 REM

```

```

440 REM *****
450 REM * HOOFD *
460 REM *****
470 REM
480 PSET(X+6,Y+9)
490 DRAW"C8 D2 G1 D1 G1 D3 F1 E1 BE1 U1
R1 D1 BG1 D1 F1 D1 F1 D4"
500 DRAW"BR1 R1 U1 E1 U1 E2 U2 H1 NL4 E1
R1 F1 R1 E1 U3"
510 DRAW"H1 U2 L1 G1 BE2 R1 E1 U3 L1 G1
L1 G1 L1 G2 L1 C1 U1 C8 L2 D1 L2"
520 PAINT(X+8,Y+11),9,8
530 PAINT(X+12,Y+21),9,8
540 REM
550 REM *****
560 REM * ARM EN BUIK *
570 REM *****
580 REM
590 PSET(X+8,X+27)
600 DRAW"C4 D1 G2 D2 G1 D3 G1 D4 G1 D2 R
3 U2 NH1 R1 U2 E1 U2 E1 U3 E1 U4 NH1 D2"
610 DRAW"R3 U2 H1 F3 D1 F1 D3 F1 D2 F1 D
3 G1 D2 L7 H1 L2"
620 PAINT(X+8,X+30),5,4
630 PAINT(X+12,X+35),5,4
640 LINE(X+5,Y+45)-(X+9,Y+45),15
650 REM
660 REM *****
670 REM * COPIEREN IN ARRAY'S *
680 REM *****
690 REM
700 COPY(78,0)-(120,18),1 TO O
710 COPY(28,0)-(70,18),1 TO T
720 COPY(X-16,Y-1)-(X+22,Y+26),1 TO H
730 COPY(X-16,Y+27)-(X+22,Y+45),1 TO L
740 CLS:SETPAGE0,0:CLS

```

```

750 REM
760 REM *****
770 REM * VAN LINKS NAAR RECHTS *
780 REM *****
790 REM
800 SETPAGE 1,0:P=0
810 P=P+8
820 COPY H,0 TO (P,Y),0
830 COPY L,0 TO (P,Y+27),0
840 COPY T,0 TO (P-2,Y+46),0
850 SET PAGE 0,1
860 P=P+8
870 COPY H,0 TO (P,Y),1
880 COPY L,0 TO (P,Y+27),1
890 COPY O,0 TO (P-2,Y+46),1
900 SETPAGE 1,0
910 IF P>200 THEN CLS:SETPAGE 0,1:CLS:GO
    TO 980
920 GOTO810
930 REM
940 REM *****
950 REM * VAN RECHTS NAAR LINKS *
960 REM *****
970 REM
980 SETPAGE 1,0:P=250
990 P=P-8
1000 COPY H,1 TO (P,Y),0
1010 COPY L,1 TO (P,Y+27),0
1020 COPY T,1 TO (P+1,Y+46),0
1030 SET PAGE 0,1
1040 P=P-8
1050 COPY H,1 TO (P,Y),1
1060 COPY L,1 TO (P,Y+27),1
1070 COPY O,1 TO (P+1,Y+46),1
1080 SETPAGE 1,0
1090 IF P<50 THEN CLS:SETPAGE 0,1:CLS:GO

```


Verdwenen transparant

Alhoewel de MSX2 computers beschikken over 512 kleuren kunt u in de meeste schermen slechts 16 kleuren tegelijk gebruiken. In screen 6 is dit zelfs beperkt tot 4 kleuren tegelijk. Daarbij komt dan nog eens dat de kleur met het nummer 0 niet kan worden veranderd. Dit is steeds transparant. Nu wordt transparant erg weinig gebruikt, waardoor u dus nog een kleur verliest.

Er bestaat echter wel een mogelijkheid om er voor te zorgen dat de kleur met het nummer 0 niet meer transparant is. Dit gaat door bit nr. 2 van het VDP register op adres 9 te zetten. Het volgende programma toont hoe dit gedaan kan worden.

De programmalijsjt:

```
10 REM *****
20 REM * VERDWENEN TRANSPARANT *
30 REM * ----- *
40 REM * *
50 REM *          VOOR MSX2          *
60 REM *****
70 REM
80 SCREEN 6:COLOR 1,2,2:CLS
90 ' TRANSPARANT AAN
100 VDP(9)=VDP(9)AND223
110 COLOR=(0,7,7,7)
120 COLOR=(1,1,1,1)
130 COLOR=(2,7,3,3)
140 COLOR=(3,2,7,2)
150 LINE(5,5)-(170,205),0,BF
160 LINE(175,5)-(340,205),1,BF
170 LINE(345,5)-(505,205),3,BF
180 FOR I=1 TO 1000:NEXT I
190 ' TRANSPARANT UIT
200 VDP(9)=VDP(9)OR 32
```

210 COLOR=(0,3,3,3)

220 I\$=INKEY\$: IF I\$="" THEN 220

Scherm wisselen

Bij het maken van een programma wilt u graag dat het programma ook op een aangename manier op het scherm gepresenteerd wordt. Daarom wordt er over het algemeen een introductiescherm of zelfs meerdere introductieschermen op het scherm geplaatst.

Bij MSX2 is dit helemaal leuk, daar u een veelheid aan grafische mogelijkheden heeft. Het is zelfs mogelijk een scherm te tekenen zonder dat de gebruiker het eigenlijke tekenen ziet. Als de computer klaar is met het tekenen laat u dan de tekening in een flits op het scherm zien. Dit doet u bijvoorbeeld zo:

```
10 SCREEN 8:COLOR 15,4,4:CLS
20 SETPAGE 1,1:CLS
30 SETPAGE 1,0
40 REM HIER TEKENT U WAT U WILT
100 SETPAGE 0,1
```

Dit is veel mooier dan als men het tekenen ziet. Nu kunt u ook beginnen met het tekenen van een andere tekening. Die zal dan niet zichtbaar zijn tot het programma de volgende regel tegenkomt:

```
200 SETPAGE 1
```

Op deze manier verdwijnt het eerste scherm en wordt plot-seling het volledige tweede scherm getoond.

Wat ook kan is de tekeningen langzaam laten overschakelen. Dit kan ook een erg mooi effect hebben. Het volgende programma toont hoe dit kan.

De programmalijst:

```
10 REM *****
20 REM * SCHERM WISSELEN *
30 REM * ----- *
```

```

40 REM * *
50 REM * VOOR MSX2 *
60 REM *****
70 REM
80 SCREEN 8: COLOR ,255: CLS
90 SETPAGE 0,0
100 OPEN"grp:" AS #1
110 FOR I=1 TO 5
120 LINE(20+I*5,10+I*3)-(200+I*5,50+I*3)
,33+4*I,B
130 NEXT I
140 PRESET(40+I*5,20+I*3):COLOR33+4*I:PR
INT#1,"TRUKS EN TIPS"
150 PRESET(41+I*5,20+I*3):COLOR33+4*I:PR
INT#1,"TRUKS EN TIPS"
160 LINE(20,90)-(235,200),,B
170 FOR I=21 TO 234
180 LINE(I,91)-(255-I,199),I
190 NEXT I
200 SETPAGE 0,1: COLOR,0: CLS
210 FOR I=1 TO 5
220 LINE(45-I*5,10+I*3)-(225-I*5,50+I*3)
,130+4*(I+2),B
230 NEXT
240 PRESET(64+I*5,20+I*3):COLOR130+4*(I+
2):PRINT#1,"DEEL 7"
250 PRESET(65+I*5,20+I*3):COLOR130+4*(I+
2):PRINT#1,"DEEL 7"
260 LINE(20,90)-(235,200),,B
270 FOR I=91 TO 199
280 LINE(21,I)-(234,289-I),I
290 NEXT I
300 CLOSE#1
310 FOR I=0 TO 7
320 FOR T=0 TO 255 STEP 8
330 COPY(T+I,0)-(T+I,105),1 TO (T+I,0),0

```



```
340 FOR W=1 TO 5:NEXT W,T
350 FOR T=255 TO 0 STEP -8
360 COPY(T-I,106)-(T-I,211),1 TO (T-I,10
6),0
370 FOR W=1 TO 5:NEXT W,T
380 NEXT I
390 I$=INKEY$: IF I$="" THEN 390
400 END
```

Spirograph

Dat ook de MSX1 computers veel grafische mogelijkheden hebben bewijst het volgende programma wel.

Het programma begint met het vragen van het aantal zijden dat de figuur moet hebben waarmee een spirograph op het scherm getekend moet worden. Daarna wordt de spirograph op het scherm getekend.

Door, nadat de spirograph volledig op het scherm verschenen is, op een toets te drukken begint het programma opnieuw en kunt u eventueel een andere spirograph bekijken. Om het programma te stoppen drukt u de [CTRL] en [STOP] toets tegelijk in.

De programmalijst:

```
10 REM *****
20 REM *           SPIROGRAPH           *
30 REM *           -----           *
40 REM *                                           *
50 REM * VOOR MSX1 EN MSX2 *
60 REM *****
70 REM
80 SCREEN 0:COLOR 15,1,1:CLS
90 INPUT "Aantal zijden";Z
100 A=80:PI=3.1415
110 SCREEN 2
120 FOR I=0 TO 2.3*PI STEP PI/20
130 FOR T=I TO 2*PI+I+.1 STEP 2*PI/Z
140 X=INT(A*1.3*SIN(T)+125)
150 Y=INT(A*COS(T)+100)
160 IF T>I THEN LINE-(X,Y) ELSE PSET(X,Y)
170 NEXT T
180 A=A-1.8
190 NEXT I
```

200 IF INKEY\$="" THEN 200
210 RUN

10 REM *****
20 REM *
30 REM *
40 REM *
50 REM * YOUR FILE IS HERE *
60 REM *****
70 REM
80 SCREEN 0:CLS:PRINT "INPUT FILE"
90 INPUT "FILE:" FILE\$
100 A\$=0:GOTO 110
110 SCREEN 0
120 FOR I=0 TO 255:STEP 1:PRINT
130 FOR J=0 TO 255:STEP 1:PRINT
140 PRINT "END OF FILE"
150 Y=INT(RND(1)*100)
160 IF Y=0 THEN LINE(Y) ELSE PRINT Y
170 NEXT J
180 NEXT I

Utilities

Behalve truuks die van uw programma's zeer krachtige routines maken staan er in dit deeltje ook enkele programma's die het programmeren gemakkelijker maken. Bij deze utilities zit onder andere een programma om zeer snel een backup van een diskette te maken met één drive. Dit gaat in vier keer, zodat elke schijf slechts vier maal in de drive geplaatst hoeft te worden.

Een andere interessante routine is de *hernummeraar*. Deze maakt het mogelijk om een stuk midden in een programma te henummeren. Dit in tegenstelling tot het commando RENUM waar het alleen mogelijk is vanaf een bepaalde plaats tot het einde van het programma te henummeren. Met het programma *speed sector run* wordt het mogelijk een diskette per sector te bekijken in ASCII formaat. Samen met nog enkele andere utilities zijn deze zeker het intikken waard, ook al zijn ze iets langer dan de meeste andere truuks in dit boekje.

Veilige karakterset

In de reeks 'Truiks en tips' zijn er reeds verschillende programmaatjes verschenen die iets deden met de karakterset. Steeds werden er op bepaalde wijze karakters van vorm veranderd. Het was echter jammer dat de vervormde karakterset na een screen-opdracht verdween zonder dat u hem terug kon roepen.

Het volgende programma brengt daar verandering in. Als u het inbouwt in een programma kunt u na elke screen opdracht de vervormde karakterset terug roepen door de volgende regels te plaatsen na de instructie 'SCREEN0':

```
200 DEFUSR0=PEEK(VARPTR(KV$)+1)+  
PEEK (VARPTR(KV$)+2)*256  
210 DUMMY=USR0(0)
```

De regelnummers die hierboven gebruikt worden mogen natuurlijk anders zijn.

Het programma werkt alleen als de schermmode 0 of de schermmode 1 geactiveerd is met minder dan 41 karakters op een regel.

Als u het programma wilt gebruiken in schermmode 0 met meer dan 40 karakters op een regel dan moeten de volgende regels als volgt veranderd worden:

```
280 DATA 21,00,10: ' LD HL,4096  
330 DATA 11,00,10: ' LD DE,4096
```

Het omschakelen tussen 40 en 80 karakters op een regel maakt ook dat de karakterset weer naar de normale waarden terug gezet wordt.

De programmalijst:

```
10 REM *****  
20 REM * VEILIG KARAKTERSET *
```

```

30 REM * ----- *
40 REM * *
50 REM * VOOR MSX1 EN MSX2 *
60 REM *****
70 REM
80 CLEAR200,&HDA00
90 FOR I=1 TO 13
100 READ A$
110 KH$=KH$+CHR$(VAL("&H"+A$))
120 NEXT
130 FOR I=1 TO 13
140 READ A$
150 KV$=KV$+CHR$(VAL("&H"+A$))
160 NEXT I
170 DEFUSR1=PEEK(VARPTR(KH$)+1)+PEEK(VAR
PTR(KH$)+2)*256
180 X=USR1(0)
190 SCREEN 0
200 DEFUSR0=PEEK(VARPTR(KV$)+1)+PEEK(VAR
PTR(KV$)+2)*256
210 X=USR0(0)
220 END
230 REM
240 REM *****
250 REM * MACHINETAAL DATA'S *
260 REM *****
270 REM
280 DATA 21,00,08: ' LD HL,2048
290 DATA 01,00,08: ' LD BC,2048
300 DATA 11,00,DA: ' LD DE,&HDA00
310 DATA CD,59,00: ' CALL &H59
320 DATA C9 : ' RET
330 DATA 11,00,08: ' LD DE,2048
340 DATA 01,00,08: ' LD BC,2048
350 DATA 21,00,DA: ' LD HL,&HDA00
360 DATA CD,5C,00: ' CALL &H5C
370 DATA C9 : ' RET

```

Hernummeraar

MSX computers hebben zoals u reeds gemerkt heeft een zeer sterke Basic. In die Basic heeft u onder andere het kommando 'RENUM'. Bij dit kommando kunt u drie waarden ingeven. De eerste waarde is de nieuwe beginregel, de tweede waarde is de oude beginregel, de derde waarde is de stapgrootte. Wat u echter niet kunt ingeven is een eindregel. Dit is erg jammer daar het niet steeds de bedoeling is om het volledige programma vanaf de beginregel te hernummeren. Met het volgende programma komt er een einde aan dit probleem.

Eerst moet het programma als ASCII file op tape of disk vastgelegd worden. Dit doet u met:

```
SAVE "HERNUM",A [RETURN]
```

Daarna kan het hernummerprogramma steeds als u het nodig heeft met MERGE achter het te hernummeren programma ingeladen worden. Dit doet u zo:

```
MERGE "HERNUM" [RETURN]
```

waarna het programma opgestart moet worden met:

```
RUN 60000 [RETURN]
```

Als het programma klaar is met het hernummeren van het stuk dat hernummerd moest worden dan kunt u het beste het hernummerprogramma uit het geheugen verwijderen met:

```
DELETE 60000-60350 [RETURN]
```

Het programma is zo klein mogelijk gehouden. Daardoor is er geen beveiliging ingebouwd die kijkt hoeveel regels er hernummerd moeten worden tussen de ingegeven beginregel en eindregel. Het kan dus gebeuren dat als u niet goed

uitkijkt het programma beschadigd wordt. Daarom is het beter uw programma eerst te save op tape of disk. Het programma past de 'GOTO' en 'GOSUB' kommando's niet aan. Dit moet u zelf doen.

De programmalijs:

```
60000 REM *****
60010 REM *   HERNUMMERAAR   *
60020 REM *   -----      *
60030 REM *                   *
60040 REM * VOOR MSX1 EN MSX2 *
60050 REM *****
60060 REM
60070 COLOR 15,4,4:SCREEN0:WIDTH40
60080 INPUT"oude beginregel";O
60090 INPUT"nieuwe beginregel";N
60100 IF N<O THEN RUN 60080
60110 INPUT"eind regel";E
60120 IF E<N THEN RUN 60080
60130 INPUT"stapgrote";S
60140 IF E<N+S THEN RUN 60080
60150 AT=&H8001
60160 RE=PEEK(AT+2)+256*PEEK(AT+3)
60170 IF RE=0 THEN 60200
60180 AT=PEEK(AT)+256*PEEK(AT+1)
60190 GOTO 60160
60200 POKE AT+2,NMOD256
60210 POKE AT+3,N\256
60220 AT=PEEK(AT)+256*PEEK(AT+1)
60230 RE=PEEK(AT+2)+256*PEEK(AT+3)
60240 IF RE=>E THEN60340
60250 IF N=>E THEN60280
60260 N=N+S
60270 GOTO60200
60280 PRINT"U hebt meer regels dan er tu
ssen de"
```



```
60290 PRINT"nieuwe beginregel en de eind  
regel"  
60300 PRINT"kunnen."  
60310 PRINT"Uw programma is waarschijnli  
jk"  
60320 PRINT"zwaar verminkt."  
60330 END  
60340 PRINT"het programma is hernummerd  
zoals u wou."  
60350 END
```

Speed sector run

De gegevensopslag van de diskette is zeer verschillend van de gegevensopslag van een cassette. Bij de diskette worden alle gegevens in sectoren weggeschreven. Deze sectoren hebben een grootte van 512 bytes. Met het volgende programma is het mogelijk deze sectoren een voor een van de diskette te lezen en in ASCII formaat op het scherm te bekijken.

Het programma toont steeds bovenaan het nummer van de sector met daaronder de inhoud van de sector. Door nu op de cursortoetsen te drukken kunt u het nummer van de sector veranderen.

pijltje omhoog : $sectornummer = sectornummer + 1$

pijltje omlaag : $sectornummer = sectornummer - 1$

Het is ook mogelijk zelf een sectornummer in te geven. Dan drukt u eerst op de '!' toets, waarna u het nieuwe sectornummer in kunt geven.

De programmalijst:

```
10 REM *****
20 REM * SPEED SECTOR RUN *
30 REM * ----- *
40 REM * *
50 REM * VOOR MSX1 EN MSX2 *
60 REM *****
70 REM
80 COLOR 15,4,4:SCREEN 0:WIDTH 40
90 FOR I=0 TO 12
100 READ A$
110 T$=T$+CHR$(VAL("&H"+A$))
120 NEXT
130 DATA 21,97,EB,11,28,00,01,00,02,CD,5
C,00,C9
```

```

140 S=1
150 GOSUB 260
160 I$=INKEY$: IF I$="" THEN 160
170 IF I$=CHR$(31) THEN IF S>1 THEN S=S-
1:GOSUB 260
180 IF I$=CHR$(30) THEN IF S<720 THEN S=
S+1:GOSUB 260
190 IF I$=CHR$(124) THEN 210
200 GOTO 160
210 LOCATE 0,0:PRINTSPC(39)
220 LOCATE 0,0:INPUT "SECTOR NUMMER";S
230 IF S>720 THEN 210
240 LOCATE 0,0:PRINTSPC(39)
250 GOSUB 260:GOTO 160
260 CLS
270 LOCATE 0,0:PRINT"Sector nummer";S
280 A$=DSKI$(0,S)
290 DEFUSR0=PEEK(VARPTR(T$)+1)+PEEK(VARP
TR(T$)+2)*256
300 DUMMY=USR0(0):BEEP
310 RETURN

```

RAM-disks

Het geheugen van de MSX computers lijkt op het eerste gezicht zeer omvangrijk. Bij MSX computers is er bijvoorbeeld over het algemeen 32k ROM + 16k VIDEO-RAM + 64k RAM, dit is in totaal 112k. Als u nu de computer opstart krijgt u slechts 28815 bytes vrij geheugen, als u geen diskdrive aangesloten heeft. Als er wel een diskdrive aangesloten is, krijgt u slechts 23432 bytes vrij geheugen. Nu heeft u in feite wel 16k geheugen meer vanwege het Video-RAM. Alles in totaal is dit echter nog niet zo veel. Dit komt doordat er 32k RAM niet gebruikt kan worden in Basic. Bij MSX1 was het zo en blijft het zo. Bij MSX2 is er echter een oplossing voor. Daar kunt u de 32k RAM gebruiken als RAM-disk. Een RAM-disk is net zo iets als een gewone disk maar dan met veel minder geheugenruimte, iets minder mogelijkheden en als u de computer uitzet verdwijnen alle gegevens. Het is wel veel sneller dan een gewone diskdrive.

Bij de MSX2 computers zijn er die geen 64k RAM hebben maar 128k RAM. Bijvoorbeeld de VG8235 van Philips. Om bij deze computer de 64k RAM ook als RAM-disk te gebruiken moet men een truukje gebruiken. Daardoor is het dan mogelijk om 3 RAM-disks te hebben van elk 32k.

De programmalijst:

```
10 REM *****
20 REM *   EXTRA RAM-disk's   *
30 REM *   -----           *
40 REM *                       *
50 REM *VOOR MSX2 MET 128K RAM*
60 REM *****
70 REM
80 OUT &HFC,4:OUT &HFD,5
90 CALL MEMINI
100 CALL MEMINI(0)
110 OUT &HFC,6:OUT &HFD,7
```

```

120 CALL MEMINI
130 CALL MEMINI(0)
140 OUT &HFC,3:OUT &HFD,2
150 CALL MEMINI
160 REM
170 REM *****
180 REM * Schakelen naar de *
190 REM * eerste RAM-disk met: *
200 REM * OUT &HFC,3:OUT &HFD,2 *
210 REM *****
220 REM
230 REM *****
240 REM * Schakelen naar de *
250 REM * tweede RAM-disk met: *
260 REM * OUT &HFC,4:OUT &HFD,5 *
270 REM *****
280 REM
290 REM *****
300 REM * Schakelen naar de *
310 REM * derde RAM-disk met: *
320 REM * OUT &HFC,6:OUT &HFD,7 *
330 REM *****

```

Het bovenstaande programma moet slechts eenmaal opgestart worden. Daarna kunt u steeds omschakelen van RAM-disk door de aangegeven 'OUT' instructie in te geven.

De twee extra RAM-disks hebben bovendien nog een speciaal voordeel. Ook na een reset van de computer blijft de inhoud van de twee extra RAM-disk's behouden. U moet dan wel na de reset het volgende intikken:

CALL MEMINI [RETURN]

en pas daarna mag u op de normale wijze overschakelen naar de gewenste RAM-disk.

Supersnelle backup

Als u al ooit geprobeerd heeft een backup van een volle schijf te maken met 1 diskdrive dan zult u waarschijnlijk een hele tijd bezig geweest zijn. Dit komt doordat het COPY kommando in disk Basic slechts sector voor sector een diskette kopieert. Daardoor moet u vaak van schijf wisselen. Als u met MSX-DOS werkt gaat dit al een heel stuk sneller, maar niet bij elke diskdrive of MSX computer met drive wordt er een MSX-DOS programma geleverd. Daarom het volgende programma.

Het programma is alleen geschikt voor MSX2 computers met 128 k RAM en kopieert een schijf in vier stukken. Dit wil zeggen dat elke schijf (origineel en backup) slechts vier maal in de drive geplaatst hoeft te worden. De methode is als volgt:

1. Tik het programma in en save het op disk of tape.
2. Reset de computer daarna en tik het volgende:

```
POKE &HF676,0:POKE&HF677,&HC0 [RETURN]
```

3. Laad nu het programma terug en tik:

```
GOTO 80 [RETURN]
```

waarna het programma gestart is en u de instructies opvolgt die gegeven worden voor het programma.

Het is steeds aan te raden om de originele schijf op WRITE PROTECT te zetten voor u begint met het kopiëren van de schijf.

De programmalijsjt:

```
10 REM *****  
20 REM * SUPERSNELLE BACKUP *  
30 REM * ----- *
```

```

40 REM * *
50 REM *VOOR MSX2 MET 128K.RAM*
60 REM *****
70 REM
80 CLEAR20,&HE000
90 SCREEN 0:COLOR 15,4,4:CLS:WIDTH 80
100 FOR I=0 TO 30
110 READ A$
120 POKE &HE000+I,VAL("&H"+A$)
130 NEXT I
140 DEFUSRO=&HE000
150 DEFUSRI=&HE00F
160 FOR I=1 TO 4
170 PRINT"Plaats de te copieren schijf i
n de drive a.u.b."
180 PRINT"[RETURN]"
190 I$=INKEY$:IF I$<>CHR$(13) THEN190
200 PRINT"Het programma leest nu een ged
eelte van de schijf."
210 REM
220 REM *****
230 REM * LEZEN *
240 REM *****
250 REM
260 FORB=1TO6
270 OUT 254,B
280 DUMMY=USR(0)'LEES 30 SECTOREN
290 S=PEEK(&HE005)+256*PEEK(&HE006)
300 S=S+30:POKE&HE005,SMOD256:POKE&HE006
,S\256
310 NEXTB
320 PRINT"Plaats de nieuwe schijf in de
drive a.u.b."
330 PRINT"[RETURN]"
340 I$=INKEY$:IF I$<>CHR$(13) THEN340
350 PRINT"Het programma schrijft een ged

```

```

eelte op de nieuwe schijf."
360 REM
370 REM *****
380 REM * SCHRIJVEN *
390 REM *****
400 REM
410 FORB=1T06
420 OUT 254,B
430 DUMMY=USR1(0)'SCHRIJF 30 SECTOREN
440 S=PEEK(&HE014)+256*PEEK(&HE015)
450 S=S+30:POKE&HE014,SMOD256:POKE&HE015
,S\256
460 NEXT B
470 NEXT I
480 PRINT"De schijf is gecopieerd."
490 END
500 REM
510 REM *****
520 REM * DISK NAAR RAM *
530 REM *****
540 REM
550 DATA 06,1E : ' LD B,30
560 DATA 0E,F8 : ' LD C,248
570 DATA 11,00,00: ' LD DE,0
580 DATA 21,00,80: ' LD HL,&H8000
590 DATA AF : ' XOR A
600 DATA CD,44,01: ' CALL &H144
610 DATA C9 : ' RET
620 REM
630 REM *****
640 REM * RAM NAAR DISK *
650 REM *****
660 REM
670 DATA 06,1E : ' LD B,30
680 DATA 0E,F8 : ' LD C,248
690 DATA 11,00,00: ' LD DE,0

```



```

700 DATA 21,00,80: ' LD    HL,&H8000
710 DATA AF      : ' XOR   A
720 DATA 37      : ' SCF
730 DATA CD,44,01: ' CALL  &H144
740 DATA C9      : ' RET

```

Klokje

In de meeste MSX2 computers zit een klok ingebouwd die, ook als de computer uit staat, steeds doorloopt.

In het volgende programma worden enkele voorbeelden gegeven van hoe men het klokje kan gebruiken. Bijvoorbeeld als alarmklokje.

De programmalijst:

```
10 REM *****
20 REM *   KLOKJE   *
30 REM *   ----- *
40 REM *           *
50 REM * V. MSX 2 *
60 REM *****
70 REM
80 SCREEN 0:WIDTH 80:COLOR 15,4,4:CLS:KEY
Y ON
90 ' *****
100 ' * FUNKTIE TOETSEN VULLEN *
110 ' * ZODANIG DAT BIJ HET   *
120 ' * INDRUKKEN VAN KEY 1 DE *
130 ' * TIJD GEGEVEN WORDT EN *
140 ' * BIJ KEY 6 DE DATUM.   *
150 ' *****
160 '
170 KEY 1,"GETTIME T$:?T$"+CHR$(13)
180 KEY 6,"GETDATE D$:?D$"+CHR$(13)
190 INPUT"Geef de alarmtijd a.u.b. (00:0
0:00)";A$
200 SETTIME A$,A
210 CLS
220 GETDATE D$   ' haal datum
230 GETTIME T$   ' haal tijd
240 GETTIME A$,A ' haal alarmtijd
```

```

250 LOCATE 0,0
260 PRINT "Datum      :";D$
270 PRINT "Tijd       :";T$
280 PRINT "Alarmtijd:";A$
290 IF RIGHT$(T$,5)=RIGHT$(A$,5) THEN 34
0
300 GOTO220
310 ' *****
320 ' * Alarm *
330 ' *****
340 SET BEEP 3,4
350 BEEP :GOTO 350

```

Hard copy van screen 6

Met een matrix printer is het veelal mogelijk om grafische afdrukken te maken.

Dit gaat echter niet erg eenvoudig. Een tekening uitprinten bijvoorbeeld is bijna niet te doen.

Daarom is het veel eenvoudiger een tekening op het scherm te maken met de grafische instructies en daarna een hard copy daarvan te maken met een programma.

Het volgende programma maakt een hard copy van screen 6. Daar er slechts vier kleuren beschikbaar zijn in screen 6 worden alle vier de kleuren afgedrukt door middel van grijs-tinten. Zodoende wordt elke kleur die op het scherm te zien is ook op de afdruk zichtbaar.

Het is mogelijk dat het programma niet op uw printer werkt, maar door de ESC codes aan te passen aan die van uw printer werkt het programma toch.

De programmalijs:

```
10 REM *****
20 REM * HARD COPY VAN SCREEN 6 *
30 REM * ----- *
40 REM * *
50 REM * VOOR MSX 2 *
60 REM *****
70 REM
80 ' *****
90 ' * SCHERM TEKENEN *
100 ' *****
110 '
120 SCREEN 6
130 COLOR 1,0,27:CLS
140 COLOR=(3,0,0,7)
150 FOR J=0 TO 200 STEP 10
```

```

160 FOR I=0 TO 440 STEP 40
170 A=((J/10)+(I/40))MOD 4
180 LINE (I,J)-(I+9,J+10),A,BF
190 LINE (I+10,J)-(I+19,J+10),(A+1)MOD 4
, BF
200 LINE (I+20,J)-(I+29,J+10),(A+2)MOD 4
, BF
210 LINE (I+30,J)-(I+39,J+10),(A+3)MOD 4
, BF
220 NEXT I
230 NEXT J
240 FOR I=0 TO 160 STEP 4
250 CIRCLE (256,106),I,(I/4)MOD 4
260 NEXT I
270 '
280 ' *****
290 ' * PRINT ROUTINE *
300 ' *****
310 '
320 LPRINT CHR$(27);"Q";CHR$(27);"T04"
330 FOR Y=0 TO 211
340 LPRINT CHR$(27);"S1024";
350 FOR X=0 TO 511
360 P=POINT(X,Y)
370 IF P=0 THEN LPRINTCHR$(0);CHR$(0);:G
OTO 410
380 IF P=1 THEN LPRINTCHR$(3);CHR$(3);:G
OTO 410
390 IF P=2 THEN LPRINTCHR$(1);CHR$(2);:G
OTO 410
400 LPRINTCHR$(1);CHR$(1);
410 NEXT X
420 LPRINTCHR$(13)
430 NEXT Y
440 SCREEN 0:WIDTH 80:COLOR 15,4,4:CLS
450 END

```

Netjes programmeren

Hoe vaak gebeurt het niet dat u een programmaatje maakt dat loopt zoals het hoort, u SAVet het en een volgende keer dat u het programma weer inleest en opstart lijkt er van alles mis te lopen. Fouten zoals veel kleinere sprites, tekst die heel anders op het scherm komt en heel andere kleuren dan er bedoeld werden.

Deze en een hele reeks andere fouten kunnen in uw programma's sluipen door niet in het begin van het programma enkele zeer belangrijke gegevens op de gewenste waarde te zetten.

In het volgende programma wordt getoond hoe u al die gegevens op de gewenste waarde zet. De waarden die gebruikt worden zijn dezelfde als bij het opstarten van de computer.

De programmlijst:

```
10 ' *****
20 ' * NETJES PROGRAMMEREN *
30 ' * ----- *
40 ' * *
50 ' * VOOR MSX 1 EN MSX 2 *
60 ' *****
70 '
80 '**BIJ DE TEKST SCHERMEN**
90 '
100 'De juiste schermmode instellen
110 SCREEN 0
120 '
130 'functie toetsen aan of uit ?
140 KEY ON
150 '
160 'schermbreedte, hoeveel kolommen
170 WIDTH 37
```

```
180 '
190 'de gepaste kleuren instellen
200 COLOR 15,4,4:CLS
210 '
220 '**BIJ DE GRAFISCHE SCHERMEN**
230 '
240 'De juiste schermmode instellen
250 '+ sprite groote
260 '+ weergavemodus (MSX 2)
270 SCREEN 2,0,,,0
280 '
290 'Bij het gebruik van 'DRAW'
300 'De juiste hoek instellen
310 DRAW"AO"
320 '
330 'De juiste schaal instellen
340 DRAW"SO"
```

String manipulatie

In de standaard ASCII tabel zijn er twee soorten lettertekens, de zogenaamde uppercase letters en de zogenaamde lowercase letters. Dit zijn respectievelijk de hoofdletters en de kleine letters.

Daar de karakterset in MSX Basic volledig compatible is met de ASCII standaard is het ook in MSX Basic mogelijk met deze lettertekens te werken.

De volgende twee programma's zijn handige routines om een string volledig om te zetten in uppercase of lowercase letters. Dit kan onder andere zeer interessant zijn na een input, zodat u niet hoeft te controleren of de string al dan niet in een bepaalde lettersoort ingegeven werd.

Het eerste programma dient om in een string alle lowercase letters om te zetten naar uppercase letters. De eigenlijke subroutine begint op regel 1000 en wordt aangeroepen met als input A\$ die de gewenste string moet bevatten.

De output is eveneens A\$ die dan de aangepaste string bevat.

De programmalijs:

```
10 ' *****
20 ' * STRING NAAR UPPERCASE *
30 ' * ----- *
40 ' * *
50 ' * Voor MSX 1 en MSX 2 *
60 ' *****
70 '
80 INPUT "Geef een regel tekst a.u.b.";R$
90 GOSUB1000
100 PRINTR$
110 END
1000 FOR I=1 TO LEN(R$)
```



```

1010 IF ASC(MID$(R$,I,1))<123 AND ASC(MI
D$(R$,I,1))>96 THEN MID$(R$,I,1)=CHR$(AS
C(MID$(R$,I,1))-32)
1020 NEXT I
1030 RETURN

```

Het tweede programma dient om in een string alle uppercase letters om te zetten naar lowercase letters. De eigenlijke subroutine begint op regel 1000 en wordt aangeroepen met als input A\$ die de gewenste string moet bevatten. De output is eveneens A\$ die dan de aangepaste string bevat.

De programmalijst:

```

10 ' *****
20 ' * STRING NAAR LOWERCASE *
30 ' * ----- *
40 ' * *
50 ' * Voor MSX 1 en MSX 2 *
60 ' *****
70 '
80 INPUT "Geef een regel tekst a.u.b.";R$
90 GOSUB1000
100 PRINTR$
110 END
1000 FOR I=1 TO LEN(R$)
1010 IF ASC(MID$(R$,I,1))<91 AND ASC(MID
$(R$,I,1))>64 THEN MID$(R$,I,1)=CHR$(ASC
(MID$(R$,I,1))+32)
1020 NEXT I
1030 RETURN

```

Listing printer

In computertijdschriften staan de listings veelal in twee kolommen geprint.

Om uw eigen listings ook in twee kolommen uit te printen moet u enkele malen nieuwe kantlijnen instellen en het papier opnieuw goed plaatsen. Vooral bij langere listings kan dit een zeer langdurig werkje zijn.

Met het volgende programma gaat het allemaal veel sneller zonder dat u iets hoeft te doen. Om een listing van een programma uit te printen doet u het volgende:

1. Tik onderstaand programma in en save het.
2. Laad nu het uit te printen programma in het computergeheugen en save het op disk als een ASCII file, door middel van: SAVE "A:prog.naam",A
3. Laad nu het onderstaand programma terug in het computergeheugen en start het op.
4. Geef antwoord op de vragen die in het programma gesteld worden.

Hierna zal het gewenste programma in twee kolommen van 37 karakters uitgeprint worden (als een printer aangesloten is).

Als u geen diskdrive heeft kunt u ook met een data recorder werken door regel 140 als volgt aan te passen:

```
140 OPEN "CAS:"+A$ FOR INPUT AS #1
```

Voor u het programma laat uitprinten is het goed de printer in te stellen op een 12 c/i lettertype. Een 10 c/i lettertype kan ook, het blad papier staat dan echter tot de randen toe bedrukt.

De programmalijst:

```
10 REM *****
```

```

20 REM * LISTING UITPRINTEN *
30 REM * ----- *
40 REM * *
50 REM * VOOR MSX1 EN MSX2 *
60 REM *****
70 REM
80 SCREEN 0:WIDTH 40:KEY OFF:COLOR 15,4,
4:CLS
90 CLEAR10000
100 PRINT"Geef de naam van het te printe
n"
110 INPUT"programma";A$
120 INPUT"Geef het aantal regels per bla
d";A
130 DIM K1$(A*2+6)
140 OPEN A$ FOR INPUT AS #1
150 '
160 ' *****
170 ' * EVENTUELE REST VAN DE VORIGE *
180 ' * PAGINA IN DE ARRAY PLAATSEN *
190 ' *****
200 '
210 B=0:T=A*2
220 IF K1$(T)<>" " THEN K1$(T-A*2)=K1$(T)
: K1$(T)=" ": B=B+1: T=T+1: GOTO 220
230 '
240 ' *****
250 ' * NIEUWE PAGINA INLEZEN *
260 ' *****
270 '
280 FOR I=B TO A*2-1
290 IF EOF(1)THEN 480
300 LINE INPUT#1,K1$(I)
310 PRINT K1$(I)
320 IF LEN(K1$(I))>37 THEN I=I+1: K1$(I)
=MID$(K1$(I-1),38): K1$(I-1)=LEFT$(K1$(I

```

```

-1),37): GOTO 320
330 NEXT I
340 '
350 ' *****
360 ' * UITPRINTEN *
370 ' *****
380 '
390 FOR I=0 TO A-1
400 LPRINT K1$(I);SPC(37-LEN(K1$(I)));
   ! ";K1$(I+A)
410 NEXT I:LPRINT CHR$(12)
420 IF V THEN END ELSE 210
430 '
440 ' *****
450 ' * LAATSTE PAGINA *
460 ' *****
470 '
480 FOR T=I TO A*2-1
490 K1$(T)=" "
500 NEXT T
510 V=1:GOTO 390

```

Nieuwe schermmode

Nieuwe schermmode

Bij de MSX computers werkt men steeds met schermmodes. Zo zijn er bij de MSX computers twee tekst schermen en twee grafische schermen. Bij de MSX2 computers komen daar nog eens vijf grafische schermen bij.

Een beetje jammer is wel dat bijvoorbeeld de grafische schermen niet gecombineerd kunnen worden met de tekst schermen. Daardoor is het niet mogelijk tegelijk met graphics en met de fullscreen editor van MSX Basic te werken. Met de volgende methode komt daar echter verandering in.

Wat er eigenlijk gedaan wordt in de volgende programma's is simpel. U gaat eerst over in screen 1:

```
SCREEN 1 [RETURN]
```

Daarna wordt er door middel van een directe adressering van de VDP overgeschakeld naar screen 2. Dat gebeurt als volgt:

```
VDP(0) = VDP(0) OR 2 [RETURN]
```

```
VDP(1) = &B11100111 [RETURN]
```

Als er nu eigenaardige dingen op het scherm gebeuren hoeft u niet te schrikken. Er is niets aan de hand.

Nu moeten er nog enkele VDP register een beetje aangepast worden door middel van:

```
VDP(4) = 3 [RETURN]
```

```
VDP(3) = 255 [RETURN]
```

Nadat deze reeks opdrachten uitgevoerd is, is de nieuwe schermmode ingesteld. Nu is het zeer goed mogelijk dat u weinig ziet, of dat uw karakterset vernietigd is. Dit is niet abnormaal en ligt aan het feit dat het Video RAM opnieuw ingedeeld is. Daarom hieronder een uitgebreide uitleg over de geheugenopbouw van het Video RAM in deze scherm-

mode. Voor de nog niet gevorderde lezer is het beter dit stuk over te slaan en meteen te beginnen met het intikken van de programma's.

In deze schermmode beschikt men zagezegd over drie verschillende karaktersets, een schermgeheugen, een kleurgeheugen (dat in drie gedeeld wordt), een spritepatroongeheugen, een sprite-kenmerkengeheugen en een palette table (alleen MSX2). De startadressen zijn als volgt:

Karakterset 1	= vanaf 0 tot 2047
Karakterset 2	= vanaf 2048 tot 4095
Karakterset 3	= vanaf 4096 tot 6143
Kleurgeheugen 1	= vanaf 8192 tot 10239
Kleurgeheugen 2	= vanaf 10240 tot 12287
Kleurgeheugen 3	= vanaf 12288 tot 14335
Schermageheugen	= vanaf 6144 tot 6911
Sprite-kenmerkengeheugen	= vanaf 6912 tot 7039
Sprite-patroongeheugen	= vanaf 14336 tot 16383
Palette tabel	= vanaf 7040 (alleen MSX2)

Het scherm is zagezegd in drie banden verdeeld die horizontaal op elkaar liggen. Nu kunt u gedeeltelijk zelf kiezen in welke van de banden elke karakterset van toepassing moet zijn. Het instellen doet u met het VDP register nummer 4 als volgt:

VDP(4) = 0: bovenste = set 1
 middelste = set 1
 onderste = set 1

VDP(4) = 1: bovenste = set 1
 middelste = set 2
 onderste = set 1

VDP(4) = 2: bovenste = set 1
 middelste = set 1
 onderste = set 3

VDP(4) = 3: bovenste = set 1

middelste = set 2
onderste = set 3

Met deze methode kunt u dus met verschillende karaktersets werken in het zelfde scherm. Het op de juiste plaats in het geheugen zetten van een zelf ontworpen of veranderde karakterset kan gedaan worden met het programma 'VEILIGE KARAKTERSET' dat eveneens in dit boekje staat. Regel 330 van het programma 'VEILIGE KARAKTERSET' moet dan wel als volgt aangepast worden:

Bij het plaatsen in karakterset nummer 1:

```
330 DATA 11,00,00: ' LD DE,0
```

Bij het plaatsen in karakterset nummer 2:

```
330 DATA 11,00,08: ' LD DE,2048
```

Bij het plaatsen in karakterset nummer 3:

```
330 DATA 11,00,10: ' LD DE,4096
```

Het volgende programma illustreert duidelijk het gebruik van meerdere karaktersets in de nieuwe schermmode.

De programmalijsjt:

```
10 REM *****
20 REM * DE TEKST IN HET *
30 REM * MIDDEN VAN HET *
40 REM * SCHERM OP ZIJN *
50 REM * KOP PLAATSEN. *
60 REM * ----- *
70 REM *****
80 REM
90 SCREEN 1:WIDTH 32:COLOR 15,4,4:CLS:KEY OFF
100 VDP(0)=VDP(0) OR 2
110 VDP(1)=&B11100111
120 VDP(3)=255
130 VDP(4)=1
140 FOR I=1 TO 12:READ A#
```



```

150 MK$=MK$+CHR$(VAL("&H"+A$))
160 NEXT I
170 DEFUSRO=PEEK(VARPTR(MK$)+1)+PEEK(VAR
PTR(MK$)+2)*256
180 DUMMY=USRO(0)
190 FOR I=2048 TO 4095 STEP8
200 FOR J=0 TO 7
210 A(J)=VPEEK(I+J):NEXT J
220 FOR J=0 TO 7
230 VPOKEI+J,A(7-J)
240 NEXT J,I
250 END
260 REM *****
270 REM * MACHINETAAL DATA'S *
280 REM *****
290 REM
300 DATA 21,00,20: ' LD HL,8192
310 DATA 01,00,18: ' LD BC,6144
320 DATA 3E,F4 : ' LD A,244
330 DATA CD,56,00: ' CALL &H56
340 DATA C9 : ' RET

```

In de regels 90 tot en met 130 wordt eerst overgeschakeld naar de nieuwe schermmode. Merk op dat VDP register nummer 4 op 1 gezet wordt en dat daardoor voor de middelste band van het scherm karakterset nummer 2 actief is en dat karakterset nummer 1 actief wordt voor de rest van het scherm. In de regels 140 tot en met 180 wordt eerst een stukje machinetaal ingelezen waarna het machinetaal programmaatje opgestart wordt. Het dient om het volledige kleurgeheugen op een bepaalde waarde te zetten. De kleuren die ingegeven worden zijn 15 (wit) als voorgrondkleur en 4 (blauw) als achtergrondkleur.

Als u een andere kleur verkiest kunt u dit zeer eenvoudig doen door regel 320 iets aan te passen. De tweede data in die regel bevat namelijk de hexadecimale waarde van de gewenste kleuren. Het getal berekent u eenvoudig als volgt:

ACHTERGROND KLEUR + (VOORGRONDKLEUR * 16)

Om nu bijvoorbeeld als achtergrond kleur zwart en als voorgrond kleur wit te gebruiken verandert u regel 320 als volgt:

```
320 DATA 3E,F1 : ' LD A,241
```

De regels 190 tot en met 240 ten slotte zorgen er voor dat de tweede karakterset omgekeerd geplaatst wordt.

Als u nu na het opstarten van het programma eens 'LIST' geeft zult u zien dat de regels op hun kop staan als ze door het middelste deel van het scherm scrollen. Zo kunt u natuurlijk ook een heleboel andere soorten karaktersets verzinnen. Zoals bijvoorbeeld inverse, waarvoor in dit boekje een routine gegeven is. U kunt zo het effect krijgen dat in het midden van het scherm de tekst invers staat en aan de randen gewoon.

Het kleurgeheugen is bij deze schermmode even groot als de drie kleurenkaraktersets tesamen. Dit wil dus zeggen dat apart een voorgrondkleur en een achtergrondkleur kan worden ingegeven. Daardoor kunt u zeer kleurrijke teksten op het scherm brengen. Om nu uit te rekenen waar de kleuren van elk karakter precies staan in het geheugen kunt u de volgende formule gebruiken:

```
PRINT 8192+ASC("A")* [RETURN]
```

Hier wordt de eerste geheugenplaats van de kleuren voor de hoofdletter A berekend. Om de eerste geheugenplaats van een ander karakter te berekenen plaatst u gewoon het gewenste karakter tussen de haakjes. Dit geldt voor de kleuren van de eerste karakterset. Als u de eerste geheugenplaats van een karakter in de tweede karakterset wilt berekenen moet u bij de uitkomst van het zo even gegeven voorbeeld 2048 optellen. Wilt u de geheugenplaats van een karakter in de derde karakterset berekenen, dan moet bij de uitkomst van het zojuist gegeven voorbeeld 4096 opgeteld worden.

Het volgende programma geeft elk karakter een verschillende

achtergrondkleur. De kleur wordt willekeurig berekend.

De programmalijs:

```
10 REM *****
20 REM * DE TEKST IN      *
30 REM * VELE KLEUREN.  *
40 REM * BEPAALD VOOR ELK *
50 REM * KARAKTER APPART *
60 REM * ----- *
70 REM *****
80 REM
90 SCREEN 1:WIDTH 32:COLOR 15,4,4:CLS:KEY OFF
100 VDP(0)=VDP(0) OR 2
110 VDP(1)=&B11100111
120 VDP(3)=255
130 VDP(4)=0
140 R=RND(-TIME)
150 FOR I=8192 TO 14335 STEP8
160 KL=RND(1)*13+2
170 PK=&H10 OR KL
180 FOR T=0 TO 7
190 VPOKE T+I,PK
200 NEXT T,I
210 END
```

In de regels 90 tot 130 wordt eerst overgeschakeld naar de nieuwe schermmode waarbij VDP register 4 op 0 gezet wordt waardoor over het gehele scherm karakterset nummer 1 actief is. Op regel 140 wordt de random functie verbonden met de tijd waardoor u echte random getallen krijgt. In de regels 150 tot 200 tenslotte worden dus steeds acht opeenvolgende geheugenplaatsen op dezelfde waarden gezet, hetgeen wil zeggen dat elk karakter acht geheugenplaatsen heeft in het kleurengeheugen.

Zoals reeds eerder vermeld kunt u in deze schermmode

tegelijktijd met tekst en met tekeningen werken. Zo kunt u bijvoorbeeld onderaan het scherm een tekening plaatsen en bovenaan met tekst werken. Het is niet mogelijk in dezelfde band van het scherm tekeningen en tekst te verwerken. U kunt echter wel bepalen per band of daar een tekening of tekst op moet komen. Om een tekening te maken die in een bepaalde band van het scherm geplaatst moet worden moet u deze tekening eerst in het grafische scherm tekenen (screen 2). Dit komt doordat u niet met de grafische kommando's kunt werken in deze schermmode. Doet u dit toch dan resulteert dit in een foutmelding. Het volgende programma maakt eerst een tekening waarna deze tekening op disk wordt weggeschreven.

De programmalijst:

```
10 REM *****
20 REM * EERST DE TEKENING *
30 REM * TEKENEN IN SCREEN 2 *
40 REM * EN OP SCHIJF SAVEN *
50 REM * ----- *
60 REM *****
70 REM
80 SCREEN2:COLOR2,1,1:CLS
90 FORI=32 TO 0 STEP-2
100 PRESET(96-(I*2),128+(32-I))
110 A=64+(I*4):K=IMOD15+1
120 DRAW"C=K;R=A;F=I;G=I;L=A;H=I;E=I;"
130 NEXT
140 OPEN "GRP:" AS #1
150 COLOR1,4
160 LINE(92,141)-(164,148),15,BF
170 LINE(92,170)-(164,177),15,BF
180 PSET(94,141),15
190 PRINT#1,"GRAFISCHE"
200 PSET(98,170),15
210 PRINT#1,"TEKENING"
220 CLOSE#1
```

```
230 BSAVE"TEKENING.PAT",4096,6143,S
240 BSAVE"TEKENING.COL",12288,14336,S
250 END
```

De tekening wordt in de eerste regels getekend waarna in de regels 230 en 240 het laatste stuk van het bitmap geheugen en het laatste stuk van het kleur geheugen op schijf gezet wordt.

Met het volgende programma wordt de tekening weer ingeladen en in de nieuwe schermmode op het scherm gezet.

De programmalijsjt:

```
10 REM *****
20 REM * DE TEKENING TERUG *
30 REM * INLEZEN EN OP HET *
40 REM * SCHERM PLAATSEN *
50 REM * IN SCREEN 1 *
60 REM * ----- *
70 REM *****
80 REM
90 SCREEN 1:WIDTH 32:COLOR 15,4,4:CLS
100 VDP(0)=VDP(0) OR 2
110 VDP(1)=&B11100111
120 VDP(3)=255
130 VDP(4)=3
140 FOR I=1 TO 12:READ A$
150 MK$=MK$+CHR$(VAL("&H"+A$))
160 NEXT I
170 FOR L=1 TO 3
180 FOR I=1 TO 13:READ A$
190 MP$(L)=MK$(L)+CHR$(VAL("&H"+A$))
200 NEXT I,L
210 DEFUSRO=PEEK(VARPTR(MK$)+1)+PEEK(VAR
PTR(MK$)+2)*256
220 DUMMY=USRO(0)
```

```

230 DEFUSR1=PEEK(VARPTR(MP$(1))+1)+PEEK(
VARPTR(MP$(1))+2)*256
240 DUMMY=USR1(0)
250 DEFUSR2=PEEK(VARPTR(MP$(2))+1)+PEEK(
VARPTR(MP$(2))+2)*256
260 DUMMY=USR2(0)
270 DEFUSR3=PEEK(VARPTR(MP$(3))+1)+PEEK(
VARPTR(MP$(3))+2)*256
280 DUMMY=USR3(0)
290 BLOAD"TEKENING.PAT",S
300 BLOAD"TEKENING.COL",S
310 POKE &HF3B1,16
320 FORI=0TO255
330 VPOKE 512+I+VDP(2)*1024,I
340 NEXTI
350 LOCATE 0,0
360 END
370 REM *****
380 REM * MACHINETAAL DATA'S *
390 REM *****
400 REM
410 DATA 21,00,20: ' LD HL,8192
420 DATA 01,00,18: ' LD BC,6144
430 DATA 3E,F4 : ' LD A,244
440 DATA CD,56,00: ' CALL &H56
450 DATA C9 : ' RET
460 '
470 DATA 21,BF,1B: ' LD HL,&H1BBF
480 DATA 01,00,18: ' LD BC,6144
490 DATA 11,00,00: ' LD DE,0
500 DATA CD,5C,00: ' CALL &H5C
510 DATA C9 : ' RET
520 '
530 DATA 21,BF,1B: ' LD HL,&H1BBF
540 DATA 01,00,18: ' LD BC,6144
550 DATA 11,00,08: ' LD DE,2048

```

```

560 DATA CD,5C,00: ' CALL &H5C
570 DATA C9      : ' RET
580 '
590 DATA 21,BF,1B: ' LD   HL,&H1BBF
600 DATA 01,00,18: ' LD   BC,6144
610 DATA 11,00,10: ' LD   DE,4096
620 DATA CD,5C,00: ' CALL &H5C
630 DATA C9      : ' RET

```

In de regels 90 tot en met 130 wordt eerst overgeschakeld op de nieuwe schermmode waarbij het VDP register 4 op 3 gezet wordt. Daardoor zijn alle drie de karaktersets actief op hun normale plaats. In de regels 140 tot 280 worden enkele machinetaalroutines ingelezen en uitgevoerd. Deze dienen om het kleurgeheugen volledig op dezelfde waarde te zetten en de karaktersets met de normale karkaterset te vullen. De regels 290 en 300 zorgen dat de tekening ingelezen wordt en de regels 320 tot 340 zorgen dat de tekening op scherm verschijnt.

De Poke op regel 310 zorgt er voor dat u, na het beëindigen van het programma, niet met de cursor over de tekening kunt wandelen en zodoende de tekening vernietigt.

Als u op [SHIFT] en [CLS HOME] drukt, verdwijnt de tekening en wordt het hele scherm gewist.

Om alle verdere nieuwe mogelijkheden en combinaties van nieuwe mogelijkheden te beschrijven is dit boekje niet bedoeld. Zodoende zult u met veel uitproberen nog veel meer mogelijkheden ontdekken in deze eigenaardige schermmode.



truuks en tips deel 7

Dat MSX Basic een sterke Basic is, is een feit. Om alle krachtige Basic kommando's om te zetten in een doeltreffend programma is een andere zaak. Daarvoor moet men reeds heel wat ervaring hebben of men moet over een scala van doeltreffende routines beschikken.

In de zes vorige delen van 'MSX Truuks en tips' werden reeds een aanzienlijke reeks van deze krachtige routines gegeven. In dit deel wordt de traditie vervolgd en zijn ook weer een heleboel krachtige truuks en tips opgenomen. In dit deel ook enkele speciale MSX2 truuks die de MSX2 computers nog meer het allure van een semi-professionele computer geven.

Een greep uit de inhoud:

- Schrijven kopiëren in slechts 4 schijf-wisselingen
- Beweging-simulatie programma's
- Kleine en grote karakters tegelijkertijd op het scherm
- Nieuwe schermmode die met drie verschillende karaktersets werkt, elk karakter kan 16 kleuren hebben
- Schermbreedte veranderen zonder het scherm te wissen
- Hard copy van grafisch scherm, met grijstinten
- Karakters supersnel inverteren

Dit en nog vele andere interessante truuks vindt u allemaal in dit deel van 'Truuks en tips'.

Als u ook door dit deel doorgeworsteld bent, zult u zeker meer en meer het gevoel hebben dat uw programma's een professioneel tintje krijgen.

Nederlandstalige MSX handboeken

MSX BASIC handboek voor iedereen, door A.C.J. Groeneveld

Een compleet nederlandstalig handboek voor iedere MSX computer-gebruiker. Dit handboek omvat een volledige behandeling van het MSX-basic in het Nederlands. Het handboek geeft een antwoord op elke vraag die een programmeur, van welke scholing ook, over het MSX-basic zou kunnen stellen. De volledige syntaxisbehandeling rekent af met onzekerheden of een bepaalde schrijfwijze nu wel of niet is toegestaan. De duidelijke beschrijving geeft per sleutelwoord aan, welke de functie hiervan is. De laatste mogelijk nog aanwezig onduidelijkheden worden vervolgens door de opgenomen, zinvolle voorbeelden weggenomen

ISBN 90 6398 1007

MSX ZAKBOEKJE door Wessel Akkermans

Een vlot geschreven naslagwerk na of naast het handboek. U vindt er o.a. in: niet computergerichte tabellen; de MSX-BASIC instructieset; diverse tabellen die het BASIC-programmeren kunnen versnellen; de Z80 instructieset; hardware-gegevens (connectoren) en een aantal programmaatjes

ISBN 90 6398 888 5

MSX DISK handboek voor iedereen, door A.C.J. Groeneveld

Handboek voor diskdrivebezitters om naast het grote handboek te gebruiken. Een zeer volledige behandeling van het disk-gebeuren zelf en de specifieke disk kommando's, uitgebreid met voorbeelden, tabellen en overzichten. Het handboek is aangevuld met interessante programma's, waaronder een tekenafelprogramma en een basisprogramma voor basisonderhoud

ISBN 90 6398 407 3

MSX PRAKTIJKPROGRAMMA'S door Wessel Akkermans

Praktische programma's met waar nodig eerst een stukje theorie. Erg handig bij het maken van uw programma's. Een greep uit de onderwerpen: priemgetallen; zoeken en sorteren; trefwoordenlijsten; converteren van getallen; enz.

ISBN 90 6398 437 5

MSX QUICK DISK handboek voor iedereen, door A.C.J. Groeneveld

Het handboek voor iedere QUICK DISK gebruiker. Uitvoerige behandeling van de sleutelwoorden aangevuld met duidelijke voorbeelden met listing

ISBN 90 6398 254 2

MSX DOS handboek voor iedereen, door A.C.J. Groeneveld

Dit handboek geeft u op een heldere wijze een totaalbeeld van de mogelijkheden van het MSX-DOS. Ook is dit handboek voorzien van een inleiding op het begrip 'operating system' en dus echt een handboek voor iedereen

ISBN 90 6398 674 2

MSX2 Basic handboek

A.C.J. Groeneveld, ISBN 90 6398 221 6, 507 pagina's, prijs f 56,50
Wie denkt over de aanschaf van een MSX2 computer, of er al een heeft, kan niet zonder het 507 pagina's tellende MSX2 Basic handboek. Alles over MSX2 Basic, de grafische- en geluidsmogelijkheden en de computer zelf. Met 288 voorbeeldprogramma's.

MSX2 Disk/Dos Uitbreidingshandboek

A.C.J. Groeneveld, ISBN 90 6398 222 4, 172 pagina's, prijs f 37,50
Omvat een volledige behandeling van het MSX2 Disk Basic en het MSX DOS operating system, voorafgegaan door een zeer duidelijke inleiding tot de fenomenen disk en operating system. Verder praktische tabellen, duidelijke afbeeldingen en zinvolle voorbeelden. Een standaardwerk dat naast elke MSX2 Disk computer zou moeten liggen.

MSX2 Utility-Toepassingshandboek

A.C.J. Groeneveld, ISBN 90 6398 223 2, 144 pagina's, prijs f 29,75
Een verzameling programma's die voor elke MSX-er onontbeerlijk zijn. Een aantal van de mogelijkheden met deze programma's: bestandsonderhoud met lijstwerk in iedere vorm op schijf en tape, staaf- en taartdiagrammen, programma's samenstellen met sprites en geluidseffecten, binair manipuleren binnen blokken op schijf. Alle programma's in dit boek zijn geschikt voor zowel MSX als MSX2 computers.

MSX en MSX2 mogelijkheden – Wat kan- wat mankeert - wat kan ik met mijn computer

Wessel Akkermans, ISBN 90 6398 606 8, 100 pagina's, prijs f 29,50
Welk type MSX computer heb ik? Doen mijn joysticks het wel goed? Werken alle toetsen naar behoren, maakt mijn schijfveenheid lees- of schrijffouten? Op al deze vragen krijgt u antwoord bij het uitvoeren van het in dit boek beschreven programma. Eigenlijk gaat het om een groot aantal afzonderlijke programma's, die als subroutine zijn opgenomen binnen een groot hoofdprogramma, van waaruit de subroutines gemakkelijk zijn te starten. Werkt zowel op MSX als op MSX2.
De beschreven programmatuur is leerzaam voor elke MSX-er, en bovendien een waardevol gereedschap bij het onderzoeken van de goede werking van MSX en MSX2 computers voor alle gebruikers.

MSX Basic voor kinderen deel 1

H.C. de Heer, ISBN 90 6398 084 1, 110 pagina's prijs f 19,50
De bedoeling is kinderen te vertellen hoe zij een programma kunnen begrijpen, maar ook hoe zij zelf kunnen leren programmeren. Alle onderwerpen die niet beslist nodig zijn worden niet behandeld. Allereerst echter maakt de lezer kennis met het verschijnsel computer en vervolgens met de taal van de computer. Voor kinderen vanaf 8 jaar.

deel 2 ISBN 90 6398 304 2, prijs f 24,50

MSX TRUUKS EN TIPS

Als MSX-er moet je wel even wat tijd uittrekken om het MSX Basic te leren kennen, maar beheers je het eenmaal, dan kun je met een paar instructies de meest ongelofelijke dingen doen.

Programmeren is en blijft echter een kunst. Het is niet genoeg om alle woorden van een programmeertaal uit het hoofd te kennen. Nee, de kunst is om vanuit deze bouwstenen een goed funktionerend en foutloos programma te schrijven. In het op een korte en krachtige wijze oplossen van problemen herkent men de ware programmeur.

En achter alle boekjes uit de serie MSX Truuks en tips gaan zulke 'ware programmeurs' schuil. MSX Truuks en tips: boordevol waardevolle truuks, tips, routines en programma's, soms in machinetaal; in elk geval altijd razend interessant. Voor iedere MSX-er een bron van lering en vermaak. Bij ieder deel opnieuw.

deel 1 ISBN 90 6398 900 8

deel 5 ISBN 90 6398 745 5

deel 2 ISBN 90 6398 340 9

deel 6 ISBN 90 6398 879 6

deel 3 ISBN 90 6398 910 5

deel 7 ISBN 90 6398 789 7

deel 4 ISBN 90 6398 897 4

MSX SOFTWARE PLUS

MSX INTROTAPE door A.C.J. Groeneveld

Heeft u nog maar net een MSX computer en wilt u graag weten wat hij kan en hoe u kunt leren hem te programmeren? Deze kassette introduceert MSX op een uiterst vriendelijke en onderwijzende manier. U krijgt volledige instructies om het programma te starten. Daarna volgt een demonstratie van de MSX mogelijkheden: tekenen met sprites, geluidsmogelijkheden... Tenslotte twee 'les' gedeeltes. In anderhalf tot drie uur weet u wat de MSX computer is, wat hij kan, en heeft u haast ongemerkt al wat regels geprogrammeerd. ISBN 90 6398 148 1

MSX SCRIPT door Ton Weijters

Een volledig menugestuurde nederlandstalige tekstverwerker, geschikt om efficiënt grotere of kleinere teksten te bewerken. De volledige pagina-indeling wordt door het programma verzorgd, evenals de telling, de toptitel en het invullen van regels. Ook corrigeren, zoeken, string-substitutie, blokken tekst verplaatsen, kopiëren of verwijderen, onderstrepen en vet zetten, is mogelijk. ISBN 90 6398 189 9

MSX DRAWS door A.C.J. Groeneveld

Een MSX tekenprogramma, waarmee u al binnen 10 minuten uw eerste tekening kunt maken. Draws werkt erg vriendelijk en maakt gebruik van alle grafische mogelijkheden van uw MSX computer, voor technische of creatieve tekeningen. Het effectief bereik is ruim 30000 bij 30000 puntjes met mogelijkheden als o.a. cirkels, krommen, verkleinen, verschuiven, verdraaien en substitueren. ISBN 90 6398 754 4

MSX LEERBOEKEN

door Wessel Akkermans en Piet den Heijer

De serie MSX leerboeken geeft een complete cursus MSX-BASIC programmeren, in drie delen. Deze leerboeken zijn gericht op de beginnende programmeur. De moeilijkheidsgraad van de leerstof wordt dan ook slechts geleidelijk hoger. De gebruikte voorbeelden zijn zo praktisch mogelijk gekozen. Hierdoor kunnen al in een vroeg stadium bruikbare programma's worden gemaakt. Dit zal de lezer/leerling er toe aansporen om verder te gaan. Aan het eind van ieder deel is een groot voorbeeldprogramma opgenomen. Dit programma laat zien waartoe de lezer/leerling na bestudering van het betreffende leerboek in staat zal zijn.

Bij ieder leerboek is een afzonderlijk –Opdrachten en uitwerkingen– boekje te verkrijgen. In deze boekjes staan, in volgorde van de hoofdstukken uit het leerboek, vragen en opdrachten met antwoorden en uitwerkingen. Een unieke serie leerboeken voor een ieder die meer over MSX wil weten en het betere werk met zijn computer wil maken.

- MSX Basic leerboek deel 1 - ISBN 90 6398 649 1
- Opdrachten bij deel 1 - ISBN 90 6398 596 7
- MSX Basic leerboek deel 2 - ISBN 90 6398 769 2
- Opdrachten bij deel 2 - ISBN 90 6398 556 8
- MSX DOS leerboek deel 3 - ISBN 90 6398 519 3
- Opdrachten bij deel 3 - ISBN 90 6398 516 9

MSX Verder uitgediept door H. Klopper

Eindelijk een Nederlandstalig boek over het altijd in de mist gehulde onderwerp – PEEKS EN POKES. In dit boek staan alle belangrijke RAM en VRAM adressen. De video chip en zijn registers worden volledig uitgelegd. Maar ook hoe men een machinetaal programma van cassette naar disk kan schrijven. Bovendien een diskloader utility en een uiterst geavanceerde programma beveiliging. Tenslotte zijn er een aantal interessante programma's opgenomen, waaronder een wereldkaart, waarmee verder kan worden geëxperimenteerd. Elke MSX gebruiker kan in dit boek iets van zijn gading vinden en nieuws leren.

ISBN 90 6398 447 2

MSX Machinetaal handboek door H. Klopper en M. Le Belle

Hoewel een MSX computer over een krachtig Basic beschikt, is het toch handig tijdens het programmeren de grondbeginselen van machinetaal te kennen. Daarvoor is dit boek een goede gids. De zaken worden niet puur theoretisch maar ook aan de hand van duidelijke voorbeelden, die direkt bruikbaar zijn, uitvoerig uitgelegd. Enkele onderwerpen zijn verder – scroll routine –machinetaal software (ook in disk Basic) op cassette zetten –disassembler –Z80 assembler instructies –lijst van ROM-routines –alle hook-adressen –bespreking van Basic tokens en een compleet token-overzicht. Het handboek voor iedere MSX programmeur die zijn computer ten volle wil benutten.

ISBN 90 6398 735 8