

SONY®

MSX

MSX-BASIC

HANDBOEK VOOR HET PROGRAMMEREN



HIT BIT

VOORWOORD

In dit handboek wordt het gebruik van alle MSX-BASIC bevelen en functies beschreven. Bovendien worden de bevelen van diverse groepen in een afzonderlijk hoofdstuk voorzien van nadere uitleg aan de hand van praktische voorbeelden.

U kunt dit handboek gebruiken om BASIC te leren, of als hulp bij het daadwerkelijk MSX-BASIC programmeren.

Het handboek is uitsluitend bedoeld voor gebruik met het Sony MSX systeem.

INHOUDSOPGAVE

Hoofdstuk 1 Bevelen, funkties, speciale variabelen en funkties, en foutmeldingen

| | | | |
|--|----|--------------------------------------|----|
| 1-1 Bevelen en funkties | 6 | ERL | 40 |
| ABS | 8 | ERR | 41 |
| ASC | 8 | ERROR | 42 |
| ATN | 9 | EXP | 43 |
| AUTO | 10 | FIX | 43 |
| BEEP | 11 | FOR—NEXT | 44 |
| BIN\$ | 11 | FRE | 46 |
| BLOAD | 12 | GOSUB—RETURN | 47 |
| BSAVE | 13 | GOTO | 50 |
| CALL | 14 | HEX\$ | 50 |
| CDBL | 14 | IF—THEN—ELSE | 51 |
| CHR\$ | 15 | INKEY\$ | 53 |
| CINT | 16 | INP | 53 |
| CIRCLE | 17 | INPUT | 54 |
| CLEAR | 19 | INPUT\$ | 55 |
| CLOAD | 20 | INPUT # | 56 |
| CLOAD? | 21 | INSTR | 57 |
| CLOSE | 22 | INT | 58 |
| CLS | 22 | INTERVAL { ON OFF STOP } | 58 |
| COLOR | 23 | KEY | 59 |
| CONT | 24 | KEY LIST | 60 |
| COS | 24 | KEY { ON OFF } | 61 |
| CSNG | 25 | KEY(n) { ON OFF STOP } | 62 |
| CSAVE | 25 | LEFT\$ | 63 |
| CSRLIN | 26 | LEN | 64 |
| DATA | 27 | LET | 65 |
| DEF FN | 28 | LINE | 65 |
| DEF { INT SNG DBL STR } | 29 | LINE INPUT | 67 |
| DEFUSR | 30 | LINE INPUT # | 68 |
| DELETE | 31 | LIST | 69 |
| DIM | 32 | LLIST | 70 |
| DRAW | 33 | LOAD | 70 |
| EOF | 38 | LOCATE | 71 |
| END | 39 | | |
| ERASE | 40 | | |

| | |
|------------------------|-----|
| LOG..... | 72 |
| LPOS | 72 |
| LPRINT..... | 73 |
| LPRINT USING..... | 73 |
| MERGE..... | 74 |
| MID\$..... | 75 |
| MID\$=Y\$..... | 76 |
| MOTOR..... | 77 |
| NEW | 77 |
| OCT\$..... | 78 |
| ON ERROR GOTO..... | 79 |
| ON—GOSUB | 80 |
| ON—GOTO | 80 |
| ON INTERVAL GOSUB..... | 82 |
| ON KEY GOSUB | 83 |
| ON SPRITE GOSUB | 84 |
| ON STOP GOSUB | 85 |
| ON STRIG GOSUB | 86 |
| OPEN..... | 87 |
| OUT | 88 |
| PAD..... | 88 |
| PAINT | 89 |
| PDL..... | 91 |
| PEEK | 91 |
| PLAY | 92 |
| PLAY (functie) | 96 |
| POINT..... | 97 |
| POKE..... | 98 |
| POS | 99 |
| PRESET..... | 100 |
| PRINT | 100 |
| PRINT USING | 102 |
| PRINT # | 105 |
| PRINT # USING | 106 |
| PSET | 106 |
| PUT SPRITE..... | 107 |
| READ | 109 |
| REM | 110 |
| RENUM..... | 111 |
| RESTORE..... | 112 |
| RESUME..... | 113 |
| RIGHT\$..... | 114 |
| RND | 114 |
| RUN | 116 |

| | |
|------------------------------------|-----|
| SAVE | 117 |
| SCREEN | 118 |
| SGN | 120 |
| SIN..... | 121 |
| SOUND..... | 122 |
| SPACES | 124 |
| SPC..... | 124 |
| SPRITE { ON OFF STOP } | 125 |
| SQR | 125 |
| STICK | 126 |
| STOP..... | 127 |
| STOP { ON OFF STOP } | 127 |
| STRIG | 128 |
| STRIG { ON OFF STOP } | 129 |
| STR\$ | 130 |
| STRING\$ | 131 |
| SWAP..... | 132 |
| TAB | 132 |
| TAN | 133 |
| TROFF | 134 |
| TRON..... | 134 |
| USR..... | 135 |
| VAL..... | 136 |
| VARPTR..... | 137 |
| VPEEK | 138 |
| VPOKE..... | 138 |
| WAIT | 139 |
| WIDTH | 139 |

1-2 Speciale variabelen en functies

| | |
|----------------|-----|
| BASE..... | 140 |
| MAXFILES..... | 142 |
| SPRITE\$ | 143 |
| TIME | 144 |
| VDP..... | 145 |

1-3 Foutmeldingen

Hoofdstuk 2 Gebruik van MSX-BASIC

| | |
|--|-----|
| 2-1 Gebruiksklaarstand | 154 |
| 2-2 Tekstopmaakprogramma | 155 |
| 2-3 Speciale toetsen | 158 |
| 2-4 Grafische voorstellingen en afbeeldingen | 159 |
| 2-5 Geluid | 172 |
| 2-6 Verwerken van bestanden | 179 |
| 2-7 Onderbrekingen | 188 |
| 2-8 Machinetaal subroutines | 194 |

Hoofdstuk 3 Tabellen

| | |
|--|-----|
| 1. Lettertekens | 198 |
| Lettertekens gebruikt in MSX-BASIC | 198 |
| 2. Konstanten en variabelen | 201 |
| Konstanten | 201 |
| Variabelen | 202 |
| 3. Uitdrukkingen en hun gebruik | 204 |
| Delen van uitdrukkingen | 204 |
| Evaluatie van rekenkundige uitdrukkingen | 204 |
| Vergelijkingen | 204 |
| Logische uitspraken | 205 |
| Uitdrukkingen met lettertekenrijen | 205 |
| 4. Kleurcodes | 206 |
| 5. Geheugenkaart | 206 |
| Konfiguratie gebruikersgebied | 207 |
| 6. I/O poort toewijzing | 208 |

HOOFDSTUK 1

BEVELEN, FUNKTIES, SPECIALE VARIABELEN EN FUNKTIES, EN FOUTMELDINGEN

1-1 BEVELEN EN FUNKTIES

In dit hoofdstuk wordt een beschrijving gegeven van de MSX-BASIC bevelen en funkties, gerangschikt in alfabetische volgorde.

OPMERKINGEN VOORAF

Naam van bevel of funktie

PSET (point set)

Zet een stip op het grafische scherm. — Toepassing

SCHRIJFWIJZE

PSET [STEP] (X-coördinaat, Y-coördinaat) [, kleur] — Schrijfwijze

X-coördinaat, Y-coördinaat **Invullen** Numerieke konstanten of variabelen, lijst-variabelen en uitdrukkingen hiermee, van -32768 tot 32767.

Kleur **Invullen** Gehele getallen van 0 tot 15. — Invulmogelijkheden

Weglaten Huidige kleur voorgrond

Bij weglaten invoer

TOEPASSINGEN

Aanvullende uitleg van bevel of funktie, en voorbeelden van de manier waarop het bevel of de funktie toegepast kan worden.

Waar het beschreven MSX-BASIC woord een funktie is, wordt voor de naam het woord **Funktie** geschreven.

(voorbeeld)

Funktie ABS (absolute)

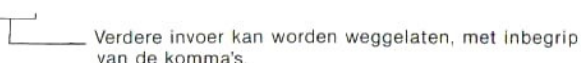
Weglaten van invoer

De invoer die onder het hoofdje SCHRIJFWIJZE tussen vierkante haken [] wordt gegeven kan weggelaten worden.

Voorbeeld

Wanneer in het geval van SCREEN [Modus], [Formaat sprite], [Intoets-signaal], [Snelheid in baud], [Soort afdrukeenheid] alleen de beeldschermmodus en het formaat van de "sprite" omschreven worden, luidt het volledig bevel als volgt:

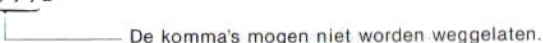
SCREEN 2,3



Verdere invoer kan worden weggelaten, met inbegrip van de komma's.

Wanneer slechts het soort afdrukeenheid nader wordt omschreven luidt het volledig bevel als volgt:

SCREEN , , , , 1



De komma's mogen niet worden weggelaten.

Deels herhalen van invoer

Voorbeeld

DATA Konstante [, Konstante]

Na het DATA bevel kunnen zoveel konstanten worden ingevoerd als gewenst is, tot het maximale invoertotaal per regel.

Functie **ABS** (absolute)

Geeft de absolute waarde van numerieke gegevens.

SCHRIJFWIJZE

ABS(X)

X **Invullen** Numerieke konstanten, variabelen, lijst-variabelen en uitdrukkingen hiermee.

Gegeven waarde: Getalswaarde

TOEPASSING

Geeft X wanneer $X \geq 0$ en $-X$ wanneer $X < 0$.

Voorbeeld gebruik

```
PRINT ABS (2)
```

```
2
```

```
PRINT ABS (3-10)
```

```
7
```

Functie **ASC** (ascii)

Geeft de ASCII cijfercode voor het eerste van een rij lettertekens.

SCHRIJFWIJZE

ASC(X\$)

X\$ **Invullen** Rij-konstanten, variabelen, lijst-variabelen en uitdrukkingen hiermee.

Gegeven waarde: Gehele getallen, decimale uitdrukkingen

TOEPASSING

Voorbeeld gebruik

```
PRINT ASC ("d")
```

```
100 ————— Cijfercode van "d".
```

```
PRINT ASC ("data")
```

```
100 ————— Cijfercode van "d"
```

Functie **ATN** (arc tangens)

Geeft de boogtangenswaarde van numerieke invoer.

SCHRIJFWIJZE

ATN(X)

X **Invullen** Numerieke konstanten, variabelen, lijst-variabelen en uitdrukkingen hiermee.

Gegeven waarde: Getalswaarde

TOEPASSING

De ATN functie geeft een numerieke waarde met drijvende komma voor de hoek waarvan de tangens, een trigonometrische waarde, gelijk is aan X. De eenheid waarin ATN de hoek uitdrukt is de radiaal.

● Om de waarde in booggraden te verkrijgen dient u de waarde die ATN geeft te vermenigvuldigen met $180/\pi$.

Voorbeeld gebruik

PRINT ATN(1)

.78539816339745 ————— Uitgedrukt in radialen

PRINT ATN(1)*180/3.14159

45.000038009905 ————— Uitgedrukt in graden

AUTO (auto)

De regelnummers worden automatisch toegewezen, te beginnen met een gekozen regelnummer, vermeerderd met een gekozen waarde.

SCHRIJFWIJZE

AUTO [nummer beginregel] [, verhoging]

Nummer beginregel **Invullen** Een geheel getal van 0 tot 65529.

Weglaten 0. Bij ook weglaten ", verhoging" wordt het regelnummer 10.

Verhoging **Invullen** Een geheel getal van 0 tot 65529.

Weglaten 10.

TOEPASSING

Handig om bij het invoeren van een programma niet telkens een regelnummer te hoeven intikken.

● Wanneer een automatisch toegewezen regelnummer al in gebruik is voor een eerder ingevoerd bevel, verschijnt links van dit regelnummer een asterisk " * ". Om het bestaande bevel te wijzigen verplaatst u de cursor naar " * " en verwijdert u de asterisk met een spatie, waarna u het nieuwe bevel invoert. Wanneer u het bestaande bevel niet wenst te wijzigen hoeft u slechts op **RETURN** te drukken.

● Om het automatisch toewijzen van regelnummers te stoppen drukt u tegelijk op **CTRL** en **STOP** of **CTRL** en **C**.

Voorbeeld gebruik

```
AUTO 100,50  
100 PRINT "12345"  
150*
```

↑ Geeft aan dat regel nummer 150 al bestaat.

BEEP (beep)

Geeft een pieptoon.

SCHRIJFWIJZE

BEEP

TOEPASSING

Voorbeeld gebruik

```
10 FOR I=0 TO 9
20 BEEP
30 NEXT I
```

Dit programma geeft tien maal achtereen een pieptoon weer.

Functie BIN\$ (binary dollar)

Zet een numerieke waarde om in een rij-waarde, en geeft deze weer in binaire code.

SCHRIJFWIJZE

BIN\$(X)

X Invullen Numerieke konstanten, variabelen, lijst-variabelen en uitdrukkingen hiermee, van -32768 tot 65535. Bij het invullen van een negatieve waarde is het resultaat hetzelfde als wanneer hier 65536 bij wordt opgeteld.

Gegeven waarde: Rij-waarde

TOEPASSING

Voorbeeld gebruik

```
PRINT BIN$(100)
1100100

PRINT BIN$(-32768)
1000000000000000
```

BLOAD (binary load)

Laadt een programma in machinetaal, of laadt het en verwerkt het tevens.

SCHRIJFWIJZE

BLOAD "apparaatnaam [bestandsnaam]" [,R] [, verschuiving]

Apparaatnaam **Invullen** CAS: ... Cassetterecorder

Bestandsnaam **Invullen** Rij van maximaal 6 lettertekens. Als 7 of meer tekens worden ingevoerd, zullen het 7de en volgende tekens worden genegeerd.

Weglaten Laadt het eerst gevonden bestand.

R parameter **Weglaten** Alleen laden.

Verschuiving **Invullen** Gehele getallen.

Weglaten 0

TOEPASSING

Laadt een programma in machinetaal dat met een BSAVE bevel is opgeslagen tussen een beginadres en een eindadres die beide in het BSAVE bevel zijn gegeven. De waarde die voor de verschuiving wordt ingevoerd wordt bij het beginadres en het eindadres opgeteld.

● Bij invoeren van de ";R" parameter wordt het programma na voltooiën van het laden verwerkt. Het beginadres voor de verwerking van het programma moet dan zijn vastgelegd in het BSAVE bevel.

BSAVE (binary save)

Slaat gegevens vanuit een nader omschreven gedeelte van het geheugen op in binaire code.

SCHERIJFWIJZE

BSAVE "apparaatnaam [bestandsnaam]", beginadres, eindadres
[, beginadres verwerking]

Apparaatnaam **Invullen** CAS: ...Cassetterecorder

Bestandsnaam **Invullen** Rij van maximaal 6 lettertekens. Als 7 of meer tekens worden ingevoerd, zullen het 7de en de volgende tekens worden genegeerd.

Weglaten Lege rij.

Beginadres, eindadres

Invullen Gehele getallen.

Beginadres verwerking

Invullen Gehele getallen van -32768 tot 65535.

Weglaten Verwerking start bij begin van het bestand.

TOEPASSING

Slaat gegevens uit een gedeelte van het geheugen dat is afgebakend met beginadres en eindadres op in binaire code, die gebruikt wordt voor het opslaan van machinetaal.

● Als een beginadres voor de verwerking wordt ingevoerd zal de verwerking bij dit adres starten zodra het machinetaal-programma wordt geladen met het BLOAD bevel met R parameter. Als geen beginadres voor de verwerking is ingevoerd, start de verwerking bij het begin van het bestand.

Voorbeeld gebruik

```
BSAVE "CAS:PRDG4",&HE000,&HE800,&HE100
```

CALL (call)

Voert een uitgebreid bevel van een ROM geheugenblok uit.

SCHRIJFWIJZE EN TOEPASSING

CALL uitgebreid bevel [(argument, argument ...)]

Argument **Invullen** Geheel-getals konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee.

Letterteken konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee.

Wanneer een uitgebreid bevel door het ROM geheugenblok wordt geleverd, kan dit met het CALL bevel ook verwerkt worden.

● (onderstreping) kan gebruikt worden in plaats van het woord "CALL".

Functie CDBL (convert to double precision)

Zet numerieke gegevens om in dubbele-precisie gegevens.

SCHRIJFWIJZE

CDBL(X)

X **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee.

Gegeven waarde: Dubbele-precisie getalswaarden

TOEPASSING

Na het geven van het CDBL bevel worden ingevoerde numerieke gegevens intern verwerkt met dubbele woordlengte voor grotere nauwkeurigheid.

Functie **CHR\$** (character dollar)

Geeft het letterteken dat bij een ingevoerde cijfercode hoort.

SCHRIJFWIJZE

CHR\$(X)

X **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van 0 tot 255.

Gegeven waarde: Rij-waarden

TOEPASSING

Voorbeeld gebruik

```
PRINT CHR$(100)
d
```

Zie de tabel met ASCII cijfercodes voor alle lettertekens (blz. 198).

Functie **CINT** (convert to integer)

Zet numerieke gegevens om in gehele getallen.

SCHRIJFWIJZE

CINT(X)

X **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van -32768 tot 32767.

Gegeven waarde: Gehele getalswaarden

TOEPASSING

Wanneer de numerieke gegevens X uit gehele getallen bestaan, wordt de waarde exakt gehandhaafd. Gegevens die bestaan uit getalswaarden met drijvende komma worden omgezet in gehele getallen door het weglaten van de waarde achter de komma. Het verschil met de INT functie is gelegen in de interne verwerking: INT geeft het grootste gehele getal kleiner dan X, terwijl CINT de ingevoerde X omzet in een geheel getal.

Voorbeeld gebruik

```
PRINT CINT(9/2)
```

```
4
```

```
PRINT CINT(12*200*55)
```

```
Overflow
```

CIRCLE (circle)

Tekent een cirkel, ellips, deel van een cirkelboog of een waaivorm in de voorgrond op het grafische scherm.

SCHRIJFWIJZE

CIRCLE [STEP] (coördinaten middelpunt), straal, [kleur], [beginhoek], [eindhoek], [aanzichtsverhouding]

X-coördinaat, Y-coördinaat middelpunt

| | | |
|---------------------|-----------------|---|
| | Invullen | Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van -32768 tot 32767. |
| Straal | Invullen | Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van -32768 tot 32767. |
| Kleur | Invullen | Gehele getallen van 0 tot 15. |
| | Weglaten | Huidige kleur voorgrond |
| Beginhoek | Invullen | Van -2π tot 2π (eenheid radialen). |
| | Weglaten | 0 |
| Eindhoek | Invullen | Van -2π tot 2π (eenheid radialen). |
| | Weglaten | 2π |
| Aanzichtsverhouding | Invullen | Positieve numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee. |
| | Weglaten | 1 |

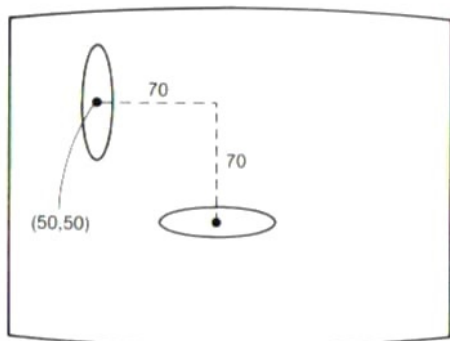
TOEPASSING

Tekent een cirkel met gekozen straal en met gekozen coördinaten voor het middelpunt. Wanneer bovendien waarden voor beginhoek en eindhoek worden ingevoerd, wordt slechts een deel van een cirkelboog getekend. Een waaivorm wordt verkregen door een negatieve waarde (-) voor beginhoek en eindhoek in te voeren. Een ellips wordt getekend wanneer u voor de aanzichtsverhouding de gewenste lengte van de verticale as uitdrukt als macht van de horizontale as.

● Zie blz. 164 voor de in te vullen waarden voor STEP.

Voorbeeld gebruik

```
10 CLS
20 SCREEN 2
30 CIRCLE (50,50),30,,,,,4
40 CIRCLE STEP(70,70),30,,,,,25
50 GOTO 50
```

Opmerking

Bij weglaten van de aanzichtsverhouding wordt een ellips getekend. Om een cirkel te tekenen vult u voor de aanzichtsverhouding 1.4 in.

CLEAR (clear)

Stelt alle variabelen op nul en bepaalt de grootte van het gebied voor lettertekens en het hoogste geheugenadres voor gebruik in BASIC. Sluit bovendien alle open bestanden.

SCHRIJFWIJZE

CLEAR [grootte gebied lettertekens] [, hoogste adres]

Grootte gebied lettertekens

Invullen Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee.

Weglaten Eerder ingestelde waarde (initiële waarde is 200). Het is niet mogelijk alleen de grootte van het gebied voor lettertekens weg te laten.

Hoogste adres **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee.

Weglaten Eerder ingestelde waarde.

TOEPASSING

Voorbeeld gebruik

```
CLEAR 400,55296
```

Alle variabelen worden met dit bevel op nul gesteld. Bovendien wordt het gebied voor rij-waarden ingesteld op 400 bytes en het hoogste geheugenadres voor gebruik in BASIC op 55296.

CLOAD (cassette load)

Laadt een MSX-BASIC programma vanaf een cassettebandje.

SCHRIJFWIJZE

CLOAD ["bestandsnaam"]

Bestandsnaam **Invullen** Rij van maximaal 6 lettertekens. Als 7 of meer tekens worden ingevoerd, zullen het 7de en volgende tekens worden genegeerd.

Weglaten Laadt het eerst gevonden programmabestand.

TOEPASSING

Voorbeeld gebruik

CLOAD "PROG1" _____ Laadt het programma genaamd PROG1 van cassette in het geheugen.

● Wanneer tijdens het laden iets mis gaat, spoelt u de band terug en laadt opnieuw.

CLOAD? (cassette load verify)

Vergelijkt het op de cassetteband opgeslagen programma met dat in het geheugen.

SCHRIJFWIJZE

CLOAD? ["bestandsnaam"]

Bestandsnaam **Invullen** Rij van maximaal 6 lettertekens. Als 7 of meer tekens worden ingevoerd, zullen het 7 de en volgende tekens worden genegeerd.

Weglaten Vergelijkt het eerst gevonden programma met dat in het geheugen.

TOEPASSING

Dit bevel dient ter controle of een programma geheel juist is opgeslagen. Bij uitvoeren van dit bevel wordt het programma in het geheugen vergeleken met het programma dat onder de gegeven bestandsnaam op cassetteband is opgeslagen.

● Als bij het vergelijken blijkt dat beide programma's precies overeenstemmen, verschijnt op het scherm "OK" en is de computer klaar voor verdere invoer. Als de programma's niet overeenstemmen verschijnt de foutmelding "Device I/O error" en wacht de computer op verdere invoer.

● Als de bestandsnaam wordt weggelaten of het bevel met een lege rij wordt ingevoerd (CLOAD? " _ "), dan wordt het eerst gevonden programabestand op cassetteband vergeleken met het programma in het geheugen. (_ staat hier voor een spatie.)

Voorbeeld gebruik

CLOAD? "PROG1"

CLOSE (close)

Sluit een bestand dat eerder geopend was met een OPEN bevel.

SCHRIJFWIJZE

CLOSE [#] [bestandsnummer] [, bestandsnummer]

Bestandsnummer **Invullen** $1 \leq \text{bestandsnummer} \leq \text{nummer gegeven in MAX-FILES} = \text{bevel}$

Weglaten Sluit alle bestanden.

TOEPASSING

Voorbeeld gebruik

```
10 MAXFILES=3
20 SCREEN 2
30 OPEN "GRP:" FOR OUTPUT AS #1 — Opent bestand 1
40 OPEN "GRP:" FOR OUTPUT AS #2 — Opent bestand 2
50 OPEN "GRP:" FOR OUTPUT AS #3 — Opent bestand 3
60 PRINT #1, "ABC"
70 PRINT #2, "DEF"
80 PRINT #3, "GHI"
100 CLOSE _____ Sluit alle bestanden.
```

CLS (clear screen)

Wist alle aanduidingen van het scherm.

SCHRIJFWIJZE

CLS

● Bij het grafische scherm verandert de achtergrondkleur als het CLS bevel wordt gegeven in combinatie met een COLOR bevel.

COLOR (color)

Bepaalt de kleur van de voorgrond, achtergrond en de randgebieden.

SCHRIJFWIJZE

COLOR [kleur voorgrond], [kleur achtergrond], [kleur randgebieden]

Kleur voorgrond, kleur achtergrond, kleur randgebieden

Invullen Gehele getallen van 0 tot 15. (Zie onderstaande kleuren-tabel.)

Weglaten Bestaande kleur

Kleurencode-tabel

| Code | Kleur | Code | Kleur | Code | Kleur | Code | Kleur |
|------|-------------|------|-------------|------|------------|------|-------------|
| 0 | Transparant | 4 | Donkerblauw | 8 | Middelrood | 12 | Donkergroen |
| 1 | Zwart | 5 | Lichtblauw | 9 | Lichtrood | 13 | Magenta |
| 2 | Middelgroen | 6 | Donkerrood | 10 | Donkergeel | 14 | Grijs |
| 3 | Lichtgroen | 7 | Hemelsblauw | 11 | Lichtgeel | 15 | Wit |

TOEPASSING

Voorbeeld gebruik

COLOR 6 ——— Alleen de kleur van de voorgrond (voor tekst de kleur van de lettertekens en voor grafische toepassingen de kleur van de grafische voorstellingen) verandert.

COLOR ,2 ——— Alleen de kleur van de achtergrond verandert.

COLOR , , 11 ——— Alleen de kleur van de randgebieden verandert.

COLOR 15 , 4 , 4 ——— Beginstand.

● Zie blz. 159 voor de schermconfiguratie.

● Om bij grafische toepassingen de kleur van de achtergrond te wijzigen kan niet volstaan worden met het slechts invoeren van de code voor de achtergrondkleur met een COLOR bevel; hiervoor dient eerst het CLS bevel uitgevoerd te worden.

CONT (continue)

Voor herstarten van een programma.

SCHERIJFWIJZE

CONT

TOEPASSING

Dit bevel dient voor het herstarten van de verwerking van een programma dat was onderbroken door indrukken van de **CTRL** en de **STOP** toetsen of door een STOP bevel in het programma. Door het geven van een CONT bevel start de verwerking bij de regel volgend op die waarin de onderbreking plaatsvond. Als het programma echter onderbroken was tijdens het uitvoeren van een INPUT bevel, zal bij herstarten eerst dit bevel uitgevoerd worden.

Functie **COS** (cosine)

Geeft de cosinuswaarde van numerieke gegevens.

SCHERIJFWIJZE

COS(X)

X **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee. (De eenheid is radialen.)

Gegeven waarde: Konstanten van -1 tot 1 met drijvende komma.

TOEPASSING

Voorbeeld gebruik

```
PRINT COS(3.14/3)
.50045968900814
```

```
PRINT COS(60*3.14/180)
.50045968900814
```

● Om X in graden uitgedrukt te krijgen gebruikt u de formule $\text{COS}(X * \pi/180)$.

Functie **CSNG** (convert to single precision)

Zet numerieke gegevens om in enkele-precisie gegevens.

SCHERIJFWIJZE

CSNG(X)

X **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee.

Gegeven waarde: Enkele-precisie getalswaarden

TOEPASSING

Voorbeeld gebruik

```
10 PRINT SQR (3)
20 PRINT CSNG (SQR (3))
RUN
1.7320508075688
1.73205
```

CSAVE (cassette save)

Slaat een MSX-BASIC programmabestand op een cassette op.

SCHRIJFWIJZE

CSAVE "bestandsnaam" [, snelheid in baud]

Bestandsnaam **Invullen** Rij van maximaal 6 lettertekens. Als 7 of meer tekens worden ingevoerd, zullen het 7de en volgende tekens worden genegeerd.

Snelheid in baud **Invullen** 1 (1200 baud) of 2 (2400 baud).

Weglaten 1 (1200 baud)

TOEPASSING

Hoewel voor een bestandsnaam 6 lettertekens naar wens gebruikt kunnen worden, mag op de eerste plaats geen cijfer komen. De snelheid is bij invoeren van een 1 gelijk aan 1200 baud, en bij invoeren van een 2 gelijk aan 2400 baud.

Voorbeeld gebruik

CSAVE "PROG1" ——— Slaat het BASIC programma dat zich in het geheugen bevindt op cassetteband op onder de bestandsnaam "PROG

Functie **CSRLIN** (cursor line)

Geeft de Y-coördinaat van de plaats waar de cursor zich bevindt.

SCHRIJFWIJZE

CSRLIN

TOEPASSING

Voorbeeld gebruik

```
10 CLS
20 INPUT A$
30 PRINT A$;
40 CL=CSRLIN
50 LOCATE 0,CL+3:PRINT "END"
```

De lettertekens waarin de gegevens van regel 30 zijn uitgedrukt nemen, al naar gelang de lengte, slechts een of enkele regels in beslag. De waarde die aan variabele CL wordt toegekend is gelijk aan de Y-coördinaat (de verticale uitdrukking van plaats) van de cursor, waarna "END" op het scherm verschijnt op een plaats met een Y-coördinaat die 3 groter is dan de waarde van CL. Daarom verschijnt "END" drie regels verder naar beneden, ongeacht de lengte van de A\$ gegevens.

DATA (data)

Bevat de gegevens die gelezen worden met een READ bevel.

SCHRIJFWIJZE

DATA konstante [, konstante]

Konstante **Invullen** Numerieke of rij-waarden.

TOEPASSING

- Wanneer meerdere gegevens in een DATA bevel gegeven worden, dienen zij gescheiden te worden door komma's (,).
- Zolang de gegevens in een DATA bevel maar dezelfde volgorde aanhouden als de variabelen in een READ bevel, is het mogelijk het DATA bevel op elke gewenste plaats te zetten en ook om de gegevens over diverse DATA bevelen te spreiden.
- Als een komma (,) of dubbele punt (:) deel uitmaakt van de gegevens in rij-waarden in een DATA bevel, of als aan het begin of aan het eind van de rij lettertekens een spatie hiervan deel uitmaakt, dan dient het leesteken of de spatie tussen aanhalingstekens (") te worden gezet.

Voorbeeld gebruik

```
10 CLS
20 SCREEN 2
30 READ A,B,C,D
40 LINE (A,B)-(C,D)
50 DATA 0,0,255,191
60 GOTO 60
```

DEF FN (define function)

Voor het definiëren van een gebruikersfunctie.

SCHERIJFWIJZE

DEF FN funktienaam [(parameter [, parameter])] = uitdrukking

Funktienaam **Invullen** Numerieke of rij-variabele (Soort variabele overeenkomstig de uitdrukking.)

Parameter **Invullen** Tot maximaal 9 variabelen.

Uitdrukking **Invullen** Numerieke of rij-konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee.

TOEPASSING

Voorbeeld gebruik

```
10 DEF FNA(X,Y) = (X*2+Y*3) / (X-Y)
20 B=FNA(4,2)
30 PRINT B
RUN
7
```

In regel 10 wordt de functie FNA(X,Y) gedefinieerd als de uitdrukking die volgt. In regel 20 worden aan de X en Y parameters de waarden 4 en 2 toegerekend, daarna wordt de functie uitgevoerd. Het resultaat, 7, wordt toegerekend aan variabele B.

DEFINT (define integer)
DEFSNG (define single precision)
DEFDBL (define double precision)
DEFSTR (define string)

Definieert de variabelen met het gegeven eerste letterteken als een bepaald soort variabele.

(INT: Geheel getal; SNG: Enkele precisie; DBL: Dubbele precisie; STR: Rij-variabele.)

SCHRIJFWIJZE

DEFINT letterteken [—letterteken]
 DEFSNG letterteken [—letterteken]
 DEFDBL letterteken [—letterteken]
 DEFSTR letterteken [—letterteken]
 Letterteken **invullen** Een letter van het alfabet.

TOEPASSING

DEFINT A-C

Als resultaat worden alle variabelen die beginnen met de letters A, B en C geheel-getals variabelen.

Voorrang van de definitietekens (% , ! , # , \$)

Na het geven van het bevel DEFINT A wordt A door het latere bevel met A# een dubbele-precisie variabele.

Voorbeeld gebruik

```

10 DEFINT A-C ————— Variabelen A t/m C zijn geheel-getals variabelen.
20 A=1.23456789 } — Variabelen A en ABC worden door regel 10 tot.
30 ABC=1.23456789 } — geheel-getals variabelen.
40 B#=1.23456789 ————— Krijgt dubbele precisie door het # teken.
50 C!=1.23456789 ————— Krijgt enkele precisie door het ! teken.
60 PRINT A;ABC;B#;C!
RUN
1 1 1.23456789 1.23457
```

DEFUSR (define user)

Geeft het beginadres wanneer een subroutine in machinetaal wordt opgeroepen met de USR functie.

SCHRIJFWIJZE

DEFUSR [X]=beginadres

X **Invullen** Gehele getallen van 0 tot 9.

Weglaten 0

Beginadres **Invullen** Numerieke konstanten, variabelen en uitdrukkingen hiermee, van 0 tot 65535.

TOEPASSING

Voorbeeld gebruik

```
DEFUSR1=&HE000
```

Hierna wordt de subroutine in machinetaal die begint vanaf adres &HE000 gedefinieerd als USR1.

● Het beginadres kan binnen een programma zo vaak als nodig is opnieuw gedefinieerd worden zonder de waarde van het gebruikersnummer (X) te wijzigen.

(Zie 2-8, Machinetaal subroutines.)

DELETE (delete)

Verwijdert een regel of regels uit een programma.

SCHRIJFWIJZE

DELETE [regelnummer] [-regelnummer]

Regelnummer **invullen** Gehele getallen van 0 tot 65529.

TOEPASSING

Voorbeeld gebruik

DELETE 40 ————— Verwijdert regel 40

DELETE 20-40 ————— Verwijdert regel 20 t/m 40.

DELETE -50 ————— Verwijdert alle regels van de beginregel t/m regel 50.

DELETE. ————— Verwijdert een regel die het laatst door een LIST
bevel werd aangegeven of een regel die door een
fout onderbroken was.

● Wanneer u slechts een regel wilt verwijderen kunt u ook volstaan met het intikken van het regelnummer en indrukken van **RETURN**.

DIM (dimension)

Bepaalt de naam van een lijstvariabele, het soort gegevens, het formaat en de afmetingen.

SCHRIJFWIJZE

DIM naam variabele (maximale waarde van een onder-index [, maximale waarde van een onder-index] ...) [, naam variabele (),]

Variabele **Invullen** Numerieke of rij-variabele.

Maximale waarde van een onder-index

Invullen Geheel-getals konstanten boven 0, variabelen, lijstvariabelen en uitdrukkingen hiermee.

TOEPASSING

Voorbeeld gebruik

DIM A(15) — Reserveert een deel van het geheugen voor 16 numerieke lijstvariabelen van A(0) t/m A(15). De beginwaarde van alle variabelen is 0.

DIM B\$(2,3) — Reserveert een deel van het geheugen voor 12 (rij-)variabelen zoals hieronder aangegeven. De beginwaarde van alle variabelen is een lege rij.

| | | |
|----------|----------|----------|
| B\$(0,0) | B\$(1,0) | B\$(2,0) |
| B\$(0,1) | B\$(1,1) | B\$(2,1) |
| B\$(0,2) | B\$(1,2) | B\$(2,2) |
| B\$(0,3) | B\$(1,3) | B\$(2,3) |

Definiëren van meerdere lijstvariabelen in een DIM bevel

DIM A(2), B\$(4,2), C(3,3)

Elke variabele wordt door komma's van de andere gescheiden.

Lijstvariabelen met meerdere dimensies

Lijstvariabelen met meerdere dimensies kunnen gedefinieerd worden door twee of meer maximale waarden voor de onder-index in te voeren.

DIM X(3,4,5) — 3 dimensies

Weglaten van het DIM bevel

Als u een lijstvariabele gebruikt zonder deze eerst met een DIM bevel te definiëren, dan wordt voor de onder-index een maximale waarde van 10 aangehouden.

DRAW (draw)

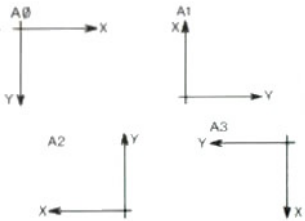
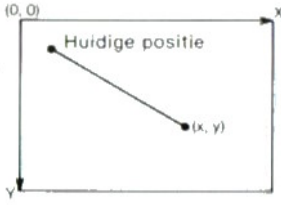
Tekent grafische voorstellingen op het scherm volgens de aanwijzingen in grafische deelinstrukties.

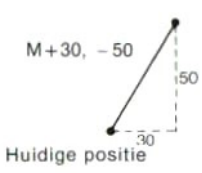
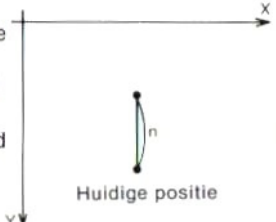
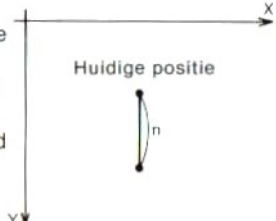
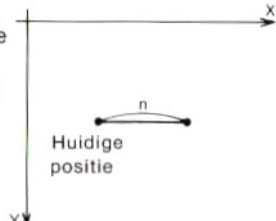
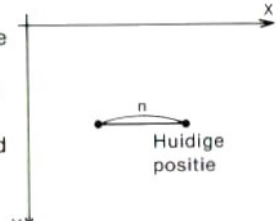
SCHRIJFWIJZE

DRAW deelinstruktie

Deelinstruktie **Invullen** Rij lettertekens (konstanten) tussen " " of rij-variabelen en uitdrukkingen hiermee. Hoofdletters of kleine letters.

Deelinstrukties

| Instructie | Invullen | Betekenis |
|-------------------|--|---|
| Sn (schaal) | $0 \leq n \leq 255$ | Geeft het aantal stippen waaruit de kleinste eenheid voor het trekken van lijnen bestaat. 1/4 punt bij $n=1$. Beginwaarde is S4 (een eenheid = 4 stippen). |
| An (hoek) | $0 \leq n \leq 3$ | Draait het coördinatensysteem in stappen van 90° vanaf de standaard coördinaat-assen (0°). Beginwaarde is A0.  |
| Cn (kleur) | $0 \leq n \leq 15$ | Geeft de kleurcode voor de kleur waarin een lijn getrokken wordt. Beginwaarde is C15. |
| Mx, y (beweeg) | $0 \leq x \leq 255$ $0 \leq y \leq 191$ | Trekt een lijn van de huidige positie naar een punt (x,y).  |

| Instructie | Invullen | Betekenis |
|------------------------------|--|--|
| $M \pm x, \pm y$ (beweeg) | $0 \leq x \leq 255$ $0 \leq y \leq 191$ | <p>Verplaatst $\pm x$ horizontaal van de huidige positie en $\pm y$ vertikaal. De eenheid voor x en y is het aantal stippen dat is bepaald in de S instructie.</p>  |
| Un (op) | | <p>Trekt een lijn in negatieve richting langs de Y as van de huidige positie naar een ander punt over een afstand n. De eenheid voor n is het aantal stippen dat is bepaald in de S instructie. (1 indien weggelaten.)</p>  |
| Dn (neer) | | <p>Trekt een lijn in positieve richting langs de Y as van de huidige positie naar een ander punt over een afstand n. De eenheid voor n is het aantal stippen dat is bepaald in de S instructie. (1 indien weggelaten.)</p>  |
| Rn (rechts) | | <p>Trekt een lijn in positieve richting langs de X as van de huidige positie naar een ander punt over een afstand n. De eenheid voor n is het aantal stippen dat is bepaald in de S instructie. (1 indien weggelaten.)</p>  |
| Ln (links) | | <p>Trekt een lijn in negatieve richting langs de X as van de huidige positie naar een ander punt over een afstand n. De eenheid voor n is het aantal stippen dat is bepaald in de S instructie. (1 indien weggelaten.)</p>  |

| Instructie | Invullen | Betekenis |
|------------|----------|---|
| En | | <p>Trekt een lijn in positieve richting langs de X as en in negatieve richting langs de Y as van de huidige positie naar een ander punt over een afstand n. De eenheid voor n is het aantal stippen dat is bepaald in de S instructie. (1 indien weggelaten.)</p> |
| Fn | | <p>Trekt een lijn in positieve richting langs de X as en in positieve richting langs de Y as van de huidige positie naar een ander punt over een afstand n. De eenheid voor n is het aantal stippen dat is bepaald in de S instructie. (1 indien weggelaten.)</p> |
| Gn | | <p>Trekt een lijn in negatieve richting langs de X as en in positieve richting langs de Y as van de huidige positie naar een ander punt over een afstand n. De eenheid voor n is het aantal stippen dat is bepaald in de S instructie. (1 indien weggelaten.)</p> |
| Hn | | <p>Trekt een lijn in negatieve richting langs de X as en in negatieve richting langs de Y as van de huidige positie naar een ander punt over een afstand n. De eenheid voor n is het aantal stippen dat is bepaald in de S instructie. (1 indien weggelaten.)</p> |

TOEPASSING

Bij intikken van een bevel of instructie voor het trekken van een lijn wordt altijd de huidige positie in het geheugen opgenomen, behalve bij instructies Sn, An en Cn.

Bijvoorbeeld,

```
DRAW "M100,120"
```

met het bovenstaande bevel wordt een lijn getrokken van een bepaalde positie naar punt (100,200), waarna dit punt de huidige positie wordt. Wanneer daarna opnieuw een bevel gegeven wordt om een lijn te trekken, dan loopt die lijn van deze huidige positie naar het nieuw gegeven punt. Een van de volgende twee instructies kan op de plaats voor de instructie om een lijn te trekken komen:

B....Hoewel de positie verandert naar het nieuw gegeven punt wordt er geen lijn getrokken. (Bijv.: BM0,0)

N....Hoewel er wel een lijn getrokken wordt verandert de positie niet naar het nieuw gegeven punt. (Bijv.: NU30,30NR30,30)

Uitdrukken van een deelinstuctie met een variabele

```
A$="BM100,150U50E50F50D50L100"  
DRAW A$
```

In dit voorbeeld wordt een deelinstuctie eerst toegewezen aan een rij-variabele A\$, waarna in het latere DRAW bevel de rij-variabele A\$ de plaats van de deelinstuctie inneemt.

Gedeeltelijk uitdrukken van een deelinstuctie met een variabele (X variabele;)

```
A$="U20R20D20L20"  
DRAW "BM50,50XA$;"  
DRAW "BM150,100XA$;"
```

Wanneer een deelinstuctie die is toegekend aan een rij-variabele tussen aanhalingstekens " " gebruikt wordt in een DRAW bevel, dan dient deze te worden voorafgegaan door X en gevolgd door ";" . In dit voorbeeld wordt de deelinstuctie die is toegekend aan de variabele A\$ gebruikt in twee DRAW bevelen.

Uitdrukken van n in een deelinstructie met een variabele (=variabele;)

De n, die staat voor de afstand van verplaatsing, de hoek en de kleurcode in de verschillende deelinstructies, kan in een DRAW bevel zowel een konstante als een variabele zijn. Als de n wordt uitgedrukt met een variabele, dan dient deze te worden voorafgegaan door = en gevolgd door " ; "

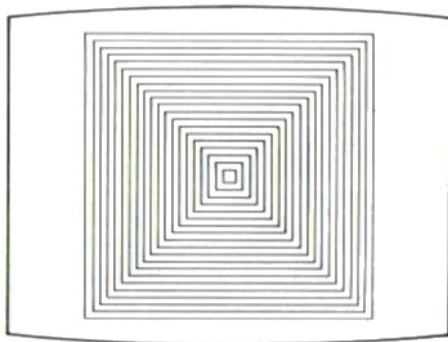
```
X=40  
DRAW "U=X ;"
```

is gelijk aan

```
DRAW "U40"
```

Voorbeeld gebruik

```
10 SCREEN 2 ----- Grafisch scherm voor gebruik van een DRAW bevel.  
20 DRAW "BM125,100" ----- Naar (125,100) zonder een lijn te trekken.  
30 FOR I=4 TO 240 STEP 12 }  
40 DRAW "S=I;BURD2L2U2RBD" } ----- Trekt een reeks vierkanten van  
50 NEXT I } verschillende afmetingen.  
60 GOTO 60
```



Functie **EOF** (end of file)

Wanneer de laatste gegevens in een bestand gelezen zijn wordt een - 1 gegeven, anders een 0.

SCHRIJFWIJZE

EOF (bestandsnummer)

Bestandsnummer **invullen** $1 \leq \text{bestandsnummer} \leq \text{nummer gegeven in MAX-FILES} = \text{bevel}$

Gegeven waarde: Geheel getal (- 1 of 0)

TOEPASSING

```
IF EOF (1) THEN CLOSE #1
```

Wanneer de laatste gegevens uit het bestand met het bestandsnummer 1 gelezen zijn, wordt dit bestand gesloten door het bovenstaande bevel.

END (end)

Beëindigt de verwerking van een programma en wacht op verdere bevelen.

SCHRIJFWIJZE

END

TOEPASSING

Het END bevel wordt gebruikt in de laatste regele van het hoofdprogramma wanneer hierna nog een subroutine volgt. Dit om te voorkomen dat de subroutine nogmaals verwerkt wordt nadat het hoofdprogramma geheel verwerkt is. Het bevel kan binnen een programma zo vaak gebruikt worden als gewenst is, bijvoorbeeld aan het eind van alle vertakkingen waarin de verwerking van het programma de resultaten afsplitst.

● Om de verwerking weer te starten moet een RUN of GOTO bevel gegeven worden. Het is niet mogelijk de verwerking te hervatten met het CONT bevel.

```
100 GOSUB 1000
    .
    .
190
200 END
1000 ' SUBROUTINE
    .
    .
1100 RETURN
```

Als in dit programma het END bevel van regel 200 niet zou voorkomen, dan zou ook zonder GOSUB bevel na regel 190 de subroutine van regel 1000 verwerkt worden, hetgeen tot een fout zou leiden.

ERASE (erase)

Wist een lijstvariabele.

SCHRIJFWIJZE

ERASE naam lijstvariabele [, naam lijstvariabele]

TOEPASSING

```
10 DIM A(100),B$(4,3)
```

```
100 ERASE A,B$
```

In dit voorbeeld worden de lijstvariabelen A en B\$, die gedefinieerd zijn in regel 10, gewist in regel 100. Daarna kan dat deel van het geheugen voor andere doeleinden gebruikt worden. Bovendien is het mogelijk hierna een andere lijstvariabele met dezelfde naam te definiëren met een DIM bevel.

Functie **ERL** (error line)

Geeft het nummer van de regel waarin een fout opgetreden is.

SCHRIJFWIJZE

ERL

Gegeven waarde: Numeriek.

TOEPASSING

Als er geen fout is opgetreden wordt 0 gegeven. Als een fout optreedt in een rechtstreeks gegeven bevel verschijnt 65535. Dit bevel kan nuttig gebruikt worden in combinatie met het ON ERROR of het ERROR bevel.

Functie **ERR** (error)

Geeft het foutnummer van een opgetreden fout.

SCHRIJFWIJZE

ERR

Gegeven waarde: Geheel getal.

TOEPASSING

Kan in combinatie met het ERROR bevel of de ERL functie gebruikt worden om de fouten in een programma te herstellen.

- Wanneer geen fouten zijn opgetreden wordt 0 gegeven.

Voorbeeld gebruik

```
PRINT 10/0
Division by zero
PRINT ERR
11
```

ERROR (error)

Simuleert de fout die behoort bij een gegeven foutnummer of definieert een fout aan de hand van een nummer.

SCHRIJFWIJZE

ERROR foutnummer

Foutnummer **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van 0 tot 255.

TOEPASSING

ERROR 1 ————— Geeft een "NEXT without FOR" fout. (Onderbreekt de verwerking van het programma.)

Definitie van een foutnummer door de gebruiker

IF A < 0 THEN ERROR 250

Als in een programma dat bovenstaande regel bevat aan variabele A een negatieve waarde wordt toegekend, dan treedt fout 250 op. (Aangezien alle foutnummers tot en met 59 in MSX-BASIC zijn gedefinieerd, dient u voor uw eigen definities van fouten nummers boven 59 te gebruiken.)

Voorbeeld gebruik

Wanneer in het volgende programma een negatieve numerieke waarde wordt ingevoerd verschijnt een foutmelding die aangeeft dat een positief getal vereist is, waarna de verwerking van het programma wordt hervat.

```
10 ON ERROR GOTO 90
20 FOR I=1 TO 10
30 INPUT A
40 IF A<0 THEN ERROR 250
50 SUM=SUM+A
60 NEXT I
70 PRINT SUM
80 END
90 IF ERR=250 THEN PRINT "Input a positive number.":RESUME 30
100 PRINT "Error!"
```

Functie **EXP** (exponential)

Geeft e^x ofwel de natuurlijke exponentiële functie van X.

SCHRIJFWIJZE

EXP(X)

X **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, tot een maximum van 145,06286085862.

Gegeven waarde: Numerieke waarde met drijvende komma.

- e (2,7182818284588) is het grondtal van de natuurlijke logaritme.

TOEPASSING

Voorbeeld gebruik

```
PRINT EXP(100)
2.6881171418087E+43
```

Functie **FIX** (fix)

Maakt gehele getallen van numerieke gegevens.

SCHRIJFWIJZE

FIX(X)

X **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee.

Gegeven waarde: Numeriek.

TOEPASSING

Geeft de waarde van numerieke gegevens voor X waarbij de cijfers achter de komma vervallen.

Voorbeeld gebruik

```
PRINT FIX(3);FIX(-3);FIX(3.58);FIX(-3.58)
3 -3 3 -3
```

FOR—NEXT (for—next)

Herhaalt het gedeelte van een programma tussen het FOR bevel en het NEXT bevel.

SCHRIJFWIJZE

FOR variabele=beginwaarde TO eindwaarde [STEP verhoging]

⋮

NEXT [variabele]

Variabele **Invullen** Numerieke variabele. De variabele in het NEXT bevel dient gelijk te zijn aan die in het FOR bevel.

Beginwaarde, eindwaarde

Invullen Numerieke constanten, variabelen en uitdrukkingen hiermee.

Verhoging **Invullen** Numerieke constanten, variabelen en uitdrukkingen hiermee.

Weglaten 1

TOEPASSING

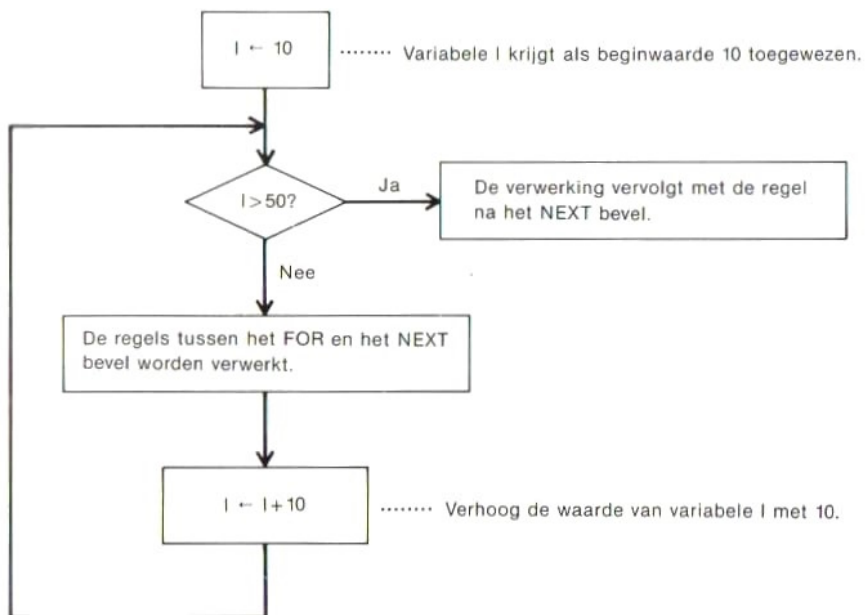
Het deel van een programma dat tussen een FOR en een NEXT bevel staat wordt net zo vaak herhaald tot de waarde van de variabele in het FOR bevel is verhoogd van een beginwaarde tot een eindwaarde. De waarde van de variabele wordt met een gegeven hoeveelheid, de verhoging, vermeerderd, telkens wanneer de verwerking van dit deel van het programma is voltooid.

● Alhoewel de variabele in het NEXT bevel kan worden wegelaten, zal de verbinding tussen het FOR bevel en het NEXT bevel duidelijk zijn wanneer de regels van het programma met een LIST bevel op het scherm worden gezet.

Voorbeeld gebruik

```
10 FOR I=10 TO 50 STEP 10
20 PRINT "I=";I
30 NEXT I
```

Dit programma wordt als volgt verwerkt:



Meervoudige lus

Een FOR—NEXT lus kan in zijn geheel binnen een andere FOR—NEXT lus worden geplaatst. Geen enkel onderdeel van de binnenste lus mag zich hierbij buiten de buitenste lus bevinden. Voor de verschillende lussen moeten verschillende variabelen worden gebruikt.

```

10 FOR I=1 TO 5
20 FOR J=1 TO I
30 PRINT "*" ;
40 NEXT J
50 PRINT
60 NEXT I
RUN
*
**
***
****
*****
  
```

Het is mogelijk verscheidene FOR bevelen te beëindigen met slechts een NEXT bevel. In dat geval kan de naam van de variabele in het NEXT bevel niet worden weggelaten. De variabelen voor de verschillende lussen dienen dan in de juiste volgorde te worden gezet, met de binnenste lus eerst, van elkaar gescheiden door komma's.

```
FOR I=0 TO 10
FOR J=0 TO 5
.
.
NEXT J,I
```

Functie **FRE** (free)

Geeft het aantal bytes in het geheugen dat niet gebruikt wordt en beschikbaar is voor gebruik in MSX-BASIC.

SCHRIJFWIJZE

FRE(X)

FRE(" ")

X **Invullen** Willekeurige numerieke waarde.

Gegeven waarde: geheel getal.

TOEPASSING

PRINT FRE (0) ————— Geeft het totaal aantal bytes in het geheugen dat niet in gebruik is.

PRINT FRE (" ") ————— Geeft het aantal bytes dat niet in gebruik is in dat deel van het geheugen gereserveerd voor rijen lettertekens.

GOSUB—RETURN (go to subroutine—return)

Start uitvoering van een gekozen subroutine. Het RETURN bevel geeft het eind van de subroutine aan waarna de verwerking vervolgt met de regel die volgt op het GOSUB bevel of met een andere, nader omschreven, regel.

SCHRIJFWIJZE

GOSUB regelnummer

:

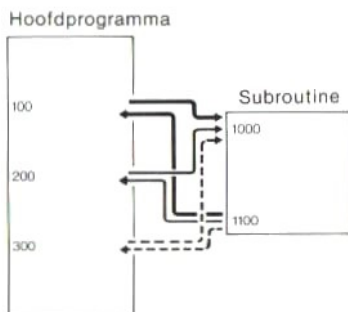
RETURN [regelnummer]

Regelnummer **Invullen** Gehele getallen van 0 tot 65529.

Weglaten Bij weglaten van het regelnummer in het RETURN bevel wordt vervolgd met de regel die volgt op het GOSUB bevel.

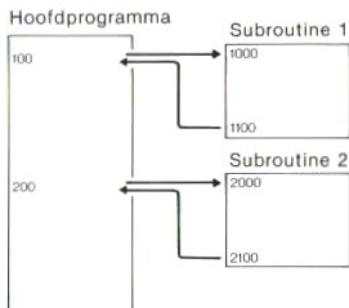
TOEPASSING

```
·  
·  
100 GOSUB 1000  
·  
·  
200 GOSUB 1000  
·  
·  
300 GOSUB 1000  
·  
·  
1000 SUBROUTINE  
·  
·  
1100 RETURN
```



Eenzelfde subroutine wordt gestart door meerdere GOSUB bevelen.

```
·  
·  
100 GOSUB 1000  
·  
·  
200 GOSUB 2000  
·  
·  
·  
1000 SUBROUTINE 1  
·  
·  
1100 RETURN  
·  
·  
2000 SUBROUTINE 2  
·  
·  
2100 RETURN
```

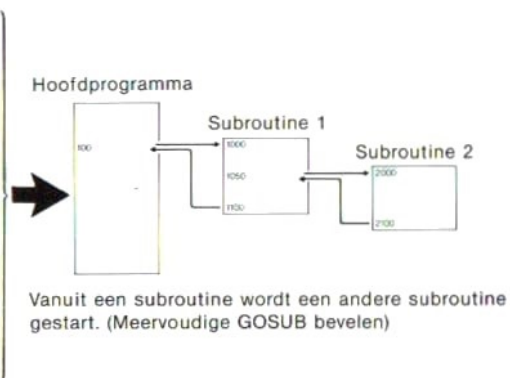


Binnen een programma bevinden zich meerdere subroutines.

```

.
.
100 GOSUB 1000
.
.
1000 SUBROUTINE 1
.
.
1050 GOSUB 2000
.
.
1100 RETURN
.
.
2000 SUBROUTINE 2
.
.
2100 RETURN

```



In hoeverre het mogelijk is meervoudige GOSUB bevelen te gebruiken is afhankelijk van het beschikbare geheugen.

GOTO (go to)

Doet de verwerking vervolgen met een gekozen regelnummer.

SCHRIJFWIJZE

GOTO regelnummer

Regelnummer **Invullen** Gehele getallen van 0 tot 65529.

TOEPASSING

De verwerking van het programma vervolgt met de regel waarvan het nummer omschreven is in het GOTO bevel.

● Wanneer dit bevel rechtstreeks gebruikt wordt vervolgt de uitvoering met een nader omschreven regel.

Funktie **HEX\$** (hexadecimal dollar)

Zet een numerieke waarde om in een rij-waarde, en geeft deze weer in hexadecimale code.

SCHRIJFWIJZE

HEX\$(X)

X **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van -32768 tot 65535. Bij het invullen van een negatieve waarde is het resultaat hetzelfde als wanneer hier 65535 bij wordt opgeteld.

Gegeven waarde: Rij-waarde

TOEPASSING

```
PRINT HEX$(100)
64
```

```
PRINT HEX$(-32768)
8000
```

```
PRINT HEX$(255)
FF
```

IF—THEN—ELSE (if—then—else)

Splitst de loop van de verwerking afhankelijk van de waarde van een uitdrukking.

SCHRIJFWIJZE

IF uitdrukking THEN bevel [ELSE bevel]

Uitdrukking **Invullen** Een vergelijking waarvan het resultaat een numerieke, logische of rekenkundige uitdrukking vormt.

ELSE bevel **Invullen** Vervolgen met het bevel na THEN als de vergelijking waar is, en met de volgende regel als de vergelijking onwaar is.

TOEPASSING

Als de waarde van een uitdrukking waar is (alles behalve 0) wordt het bevel na THEN uitgevoerd en als de waarde van de uitdrukking onwaar is (0) wordt het bevel dat volgt op ELSE uitgevoerd. Daarna wordt vervolgd met de volgende regel van het programma.

● Indien het ELSE bevel wordt weggelaten wordt het bevel na THEN uitgevoerd als de waarde van de uitdrukking waar is. Als deze onwaar is, wordt het bevel na THEN genegeerd en wordt de verwerking van het programma vervolgd met de volgende regel.

● Indien het bevel dat op THEN volgt GOTO luidt, dan kan of THEN, of GOTO worden weggelaten.

IF A=0 THEN 30]
IF A=0 GOTO 30] — betekenen hetzelfde

Na THEN volgt een bevel of regelnummer.

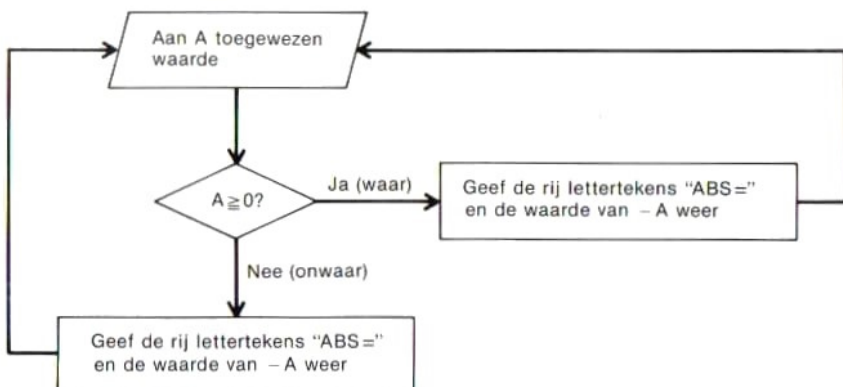
Na GOTO volgt een regelnummer.

● Indien na ELSE een GOTO bevel volgt, dan kan dit laatste worden weggelaten.

● Als na THEN of ELSE meerdere bevelen op dezelfde regel volgen, dan worden deze op volgorde uitgevoerd. De diverse bevelen dienen dan met een dubbele punt (:) te zijn gescheiden.

Voorbeeld gebruik

```
10 INPUT A
20 IF A >= 0 THEN PRINT "ABS=";A ELSE PRIN
T "ABS="; -A
30 GOTO 10
```



Meervoudige IF—THEN bevelen

Ook na THEN of ELSE kunt u opnieuw het IF—THEN bevel gebruiken. Een dergelijke meervoudige IF—THEN reeks kunt u herhalen zolang dit binnen de maximale lengte van een regel blijft.

Functie **INKEY\$** (inkey dollar)

Geeft het letterteken van de aangeslagen toets, of een lege rij bij het niet aanslaan van een toets.

SCHRIJFWIJZE

INKEY\$

Gegeven waarde: rij-waarde

TOEPASSING

Bij het indrukken van toetsen, behalve de **CTRL** + **STOP**, de **SHIFT** of de **CTRL** toets, wordt het letterteken hiervan als gegeven ingevoerd. Bij het indrukken van geen enkele toets wordt een lege rij gegeven.

Voorbeeld gebruik

```
10 CLS
20 PRINT "Press any key."
30 K$=INKEY$
40 IF K$="" THEN GOTO 30
50 PRINT K$;
60 GOTO 30
```

Wordt herhaald tot een toets wordt ingedrukt.

Bij het indrukken van een toets wordt het bijbehorende letterteken aan variabele K\$ toegewezen en met regel 50 op het scherm gebracht.

Functie **INP** (input)

Leest de gegevens van een nader opschreven I/O poort.

SCHRIJFWIJZE

INP (poortnummer)

Poortnummer **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van 0 tot 255.

TOEPASSING

Voor de invoer en het doorgeven van gegevens van een nader omschreven I/O poort. Zie blz. 208 voor de toewijzing van I/O poorten.

INPUT (input)

Voor invoer van de waarde van een variabele via het toetsenbord.

SCHRIJFWIJZE

INPUT ["Invoeraanwijzing";] variabele [, variabele] [, variabele]

Variabele **Invullen** Numerieke of rij-variabelen, lijstvariabelen.

"Invoeraanwijzing" **Invullen** Nadere aanwijzing voor de invoer van gegevens.

Weglaten Alleen "?" verschijnt op het scherm, zonder nadere aanwijzing.

TOEPASSING

Regelt de invoer van gegevens via het toetsenbord en wijst deze aan een variabele toe. Eventuele spaties voor de gegevens worden genegeerd.

● Wanneer INPUT gebruikt wordt voor het toewijzen van waarden aan numerieke variabelen, worden spaties ook binnen de gegevens genegeerd.

● Komma's die onderdeel uitmaken van de invoer worden beschouwd als scheidingstekens van verschillende gegevens, zodat hetgeen voor de komma staat wordt beschouwd als een gegeven, terwijl de komma zelf niet aan een variabele wordt toegewezen.

● Indien een invoeraanwijzing wordt ingevuld verschijnt deze op het scherm wanneer de computer om de invoer van gegevens verzoekt. Indien geen invoeraanwijzing wordt ingevuld verschijnt alleen ? op het scherm.

● Het aantal variabelen en het aantal afzonderlijke gegevens dient overeen te komen.

Voorbeeld gebruik

```
10 INPUT A  
RUN  
? } ————— Wanneer de invoeraanwijzing wordt weggelaten.
```

```
10 INPUT "A=" ; A  
RUN  
A=? } ————— Wanneer een invoeraanwijzing wordt ingevuld.
```

```
10 INPUT "A AND B " ; A,B  
RUN  
A AND B ? ?  
?? } ————— Aangezien het aantal ingevoerde gegevens  
minder is dan het aantal variabelen, volgt een  
verzoek om de ontbrekende gegevens met ??.
```

```
10 INPUT "A AND B " ; A,B  
RUN  
A AND B ? 1,2,3,4  
?Extra ignored } ————— Aanduiding dat meer gegevens worden ingev  
dan er variabelen beschikbaar zijn.  
(Overbodige gegevens worden genegeerd.)
```

Functie **INPUT\$** (input dollar)

1. Regelt de invoer van een bepaald aantal lettertekens via het toetsenbord.
2. Regelt de invoer van een bepaald aantal lettertekens uit een bestand.

SCHRIJFWIJZE

1. INPUT\$(X)
2. INPUT\$(X, [#] bestandsnummer)

X **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van 1 tot 255.

Bestandsnummer **Invullen** $1 \leq \text{bestandsnummer} \leq \text{nummer gegeven in MAXFILES} = \text{bevel}$

Gegeven waarde: Rij-waarde

TOEPASSING

Voorbeeld gebruik

```
10 X$=INPUT$(5)
20 PRINT X$
```

Nadat regel 10 is uitgevoerd, wordt gewacht op invoer via het toetsenbord. De eerste 5 lettertekens die worden ingevoerd, worden dan aan variabele X\$ toegewezen. Tijdens het invoeren via het toetsenbord worden de lettertekens niet op het scherm weergegeven.

```
10 OPEN "CAS:TEST" FOR INPUT AS #1
20 X$=INPUT$(50,#1)
30 CLOSE
```

In dit programma worden 50 lettertekens ingevoerd vanuit een bestand dat is opgeslagen op cassette, om vervolgens toegewezen te worden aan rij-variabele X\$. Daarna wordt het bestand gesloten.

Bereik van X

Indien X in de beginstand buiten het bereik van 1 tot 2000 ligt, volgt een foutmelding. Pas als met een CLEAR bevel het voor lettertekens beschikbare gebied is uitgebreid tot meer dan 255 kan voor X een waarde van 1 tot 255 worden gekozen.

INPUT # (input number)

Leest gegevens uit een bestand dat is geopend met een OPEN bevel, en zorgt dat de gegevens aan een variabele worden toegewezen.

SCHRIJFWIJZE

INPUT # bestandsnummer, variabele [, variabele]

Bestandsnummer **Invullen** $1 \leq \text{bestandsnummer} \leq \text{nummer gegeven in MAX-FILES} = \text{bevel}$

Variabele **Invullen** Numerieke of rij-variabele, lijstvariabelen.

TOEPASSING

Leest gegevens uit een bestand. Bij numerieke gegevens worden een eventuele spatie en codes voor terugkeer en regelopschuiven die voor de gegevens komen genegeerd. Als de gegevens bestaan uit een rij lettertekens worden deze allemaal, voor zover ze voor een spatie, komma, code voor terugkeer en regelopschuiven komen, als een gegeven gelezen. Wanneer aanhalingstekens " " gebruikt zijn, worden alleen de lettertekens zich hiertussen bevinden als gegevens gelezen.

Voorbeeld gebruik

```
10 OPEN "CAS:DATA" FOR INPUT AS #1-- Opent een bestand voor het
20 IF EOF(1) THEN GOTO 50          lezen van gegevens.
30 INPUT #1, A$:PRINT A$          Leest de gegevens en zorgt dat deze
40 GOTO 20                        aan variabele A$ worden toegewezen
50 CLOSE #1                       en op het scherm worden afgebeeld.
```

(Zie 2-6 Verwerken van bestanden.)

Functie **INSTR** (in string)

Vindt een nader omschreven rij lettertekens binnen een langere rij en geeft de plaats ervan weer.

SCHRIJFWIJZE

INSTR ([N,] X\$, Y\$)

N **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van 1 tot 255.

Weglaten 1

X\$, Y\$ **Invullen** Rij-konstanten, -variabelen, lijstvariabelen en uitdrukkingen hiermee.

Gegeven waarde: Geheel getal.

TOEPASSING

Geeft het nummer van een letterteken Y\$ binnen een uit numerieke gegevens bestaande rij X\$, van links af geteld. Indien N wordt ingevuld zoekt de functie een rij vanaf het letterteken op de N-de plaats binnen de X\$ rij.

Voorbeeld gebruik

```
PRINT INSTR(3, "WHAT IS THIS?", "IS")
6
```

● Wanneer de waarde voor N groter is dan de lengte van de X\$ rij of wanneer X\$ een lege rij is, of ook wanneer de Y\$ rij niet gevonden wordt, dan wordt 0 gegeven.

Functie **INT** (integer)

Geeft het grootste gehele getal dat kleiner is dan een gegeven numerieke waarde.

SCHRIJFWIJZE

INT(X)

X **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee.

Gegeven waarde: Numeriek

TOEPASSING

Voorbeeld gebruik

```
PRINT INT(3) ; INT(-3) ; INT(3.58) ; INT(-3.58)
      3 -3  3 -4
```

INTERVAL ON (interval on)

INTERVAL OFF (interval off)

INTERVAL STOP (interval stop)

Zorgt dat een onderbreking met een ingebouwde schakelklok ingeschakeld, uitgeschakeld resp. vastgehouden wordt.

SCHRIJFWIJZE

INTERVAL ON —Interval ingeschakeld.

INTERVAL OFF —Interval uitgeschakeld.

INTERVAL STOP—Interval vastgehouden.

TOEPASSING

Nadat een onderbreking met een ingebouwde schakelklok in het programma is aangegeven met het ON INTERVAL GOTO bevel, dient dit bevel om de onderbreking daadwerkelijk in te schakelen (INTERVAL ON), uit te schakelen (INTERVAL OFF) of vast te houden (INTERVAL STOP). (Zie 2-7 Onderbrekingen.)

KEY (key)

Definieert een rij lettertekens voor een funktietoets.

SCHRIJFWIJZE

KEY nummer funktietoets, rij lettertekens

Nummer funktietoets **Invullen** Geheel getal van 1 tot 10.

Rij lettertekens **Invullen** Rij van maximaal 15 lettertekens.

TOEPASSING

Nadat voor een funktietoets een rij lettertekens is gedefinieerd, is het mogelijk met een druk op die toets de gehele rij in een keer in te voeren.

● De funktietoetsnummers 1 t/m 5 geven de toetsen **F1**—**F5** aan, terwijl de nummers 6 t/m 10 staan voor het indrukken van deze zelfde vijf toetsen, tegelijk met de **SHIFT** toets.

● Bij indrukken van de **RESET** toets of het uitschakelen van de stroom worden de definities van de funktietoetsen gewist en in de beginstand gezet.

● Codes voor andere functies dan lettertekens (zoals bijv. terugkeer als met de **RETURN** toets) kunnen gedefinieerd worden met behulp van de CHR\$ functie.

Voorbeeld gebruik

KEY 1, "JAPAN" ————— Definieert "JAPAN" voor de **F1** toets.

KEY 2, "CLS"+CHR\$(13) ————— Definieert CLS **RETURN** voor de **F2** toets.

KEY LIST (key list)

Brengt de vastgelegde definities van de funktietoetsen op het scherm.

SCHRIJFWIJZE

KEY LIST

TOEPASSING

Bij uitvoeren van dit bevel komt voor alle funktietoetsen de vastgelegde rij lettertekens, die de definitie van de toets vormt, op het scherm.

Voorbeeld gebruik

```
KEY LIST
color _____ F1
auto _____ F2
goto _____ F3
list _____ F4
run _____ F5
color 15,4,4 _____ SHIFT + F1
cload" _____ SHIFT + F2
cont _____ SHIFT + F3
list. _____ SHIFT + F4
run _____ SHIFT + F5
```

Een voorbeeld van de beginstand. Hier ziet u dat de definitie van funktietoets nummer 6 (ofwel toets **F1**), tegelijk met de **SHIFT** toets ingedrukt luidt: "COLOR 15,4,4".

KEY ON, KEY OFF (key on, key off)

Brengt de definitie van een funktietoets op het scherm, resp. wist deze van het scherm.

SCHRIJFWIJZE

KEY ON of KEY OFF

TOEPASSING

Bij inschakelen wordt de definitie van alle funktietoetsen met een rij van vijf lettertekens op de onderste regel van het scherm weergegeven. Met het KEY OFF bevel kunt u deze aanuiding van het scherm wissen.

● Nadat de definities van de funktietoetsen van het scherm gewist zijn kunt u de onderste regel van het scherm gebruiken om willekeurige invoer weer te geven met het PRINT bevel.

● Met het KEY ON bevel kunt u de definities van de funktietoetsen weer doen verschijnen.

KEY (n) ON (key (n) on)
KEY (n) OFF (key (n) off)
KEY (n) STOP (key (n) stop)

Zorgt dat een onderbreking met een van de funktietoetsen ingeschakeld, uitgeschakeld resp. vastgehouden wordt.

SCHRIJFWIJZE EN FUNKTIE

KEY (nummer funktietoets) ON —Onderbreking ingeschakeld.

KEY (nummer funktietoets) OFF —Onderbreking uitgeschakeld.

KEY (nummer funktietoets) STOP—Onderbreking vastgehouden.

Nummerfunktietoets **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van 1 tot 5.

TOEPASSING

Om met een nummer aan te geven welke funktietoets u voor een onderbreking wilt gebruiken.

KEY (1) ON — Schakelt een onderbreking met de **F1** funktietoets in.

KEY (2) OFF — Schakelt een onderbreking met de **F2** funktietoets uit.

KEY (3) STOP — Houdt een onderbreking met de **F3** funktietoets vast.

(Zie 2-7 Onderbrekingen.)

Functie **LEFT\$** (left dollar)

Geeft een nader aangeluid aantal lettertekens vanaf de linkerkant van een langere rij als rij-gegevens weer.

SCHRIJFWIJZE

LEFT\$ (X\$, N)

X\$ Invullen Rij-konstanten, -variabelen, lijstvariabelen en uitdrukkingen hiermee.

N Invullen Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van 0 tot 255.

Gegeven waarde: Rij-waarden.

TOEPASSING

Voorbeeld gebruik

```
PRINT LEFT$("MSX-BASIC",3)
```

```
MSX
```

```
Ok
```

```
PRINT LEFT$("MSX-BASIC",3.8)
```

```
MSX
```

```
Ok
```

} Als N geen geheel getal is worden de cijfers achter de decimale punt genegeerd.

```
PRINT LEFT$("MSX-BASIC",0)
```

```
Ok
```

} Als N gelijk is aan 0 wordt een lege rij gegeven.

Functie **LEN** (length)

Geeft het aantal lettertekens in een rij (ofwel de lengte hiervan) numeriek weer.

SCHRIJFWIJZE

LEN(X\$)

X\$ **Invullen** Rij-konstanten, -variabelen, lijstvariabelen en uitdrukkingen hiermee.

Gegeven waarde: Geheel getal.

TOEPASSING

Voorbeeld gebruik

```
PRINT LEN("CHRISTMAS")
```

9

```
PRINT LEN("THE END")
```

7

Als zich in een rij lettertekens een spatie bevindt, wordt deze als 1 letterteken meegeteld.

● Ook een CHR\$ functie binnen een rij lettertekens wordt als 1 letterteken meegeteld.

LET (let)

Zorgt dat gegevens als waarden aan een variabele worden toegekend.

SCHRIJFWIJZE

[LET] variabele=X

Variabele **Invullen** Numerieke of rij-variabelen, lijstvariabelen.

X **Invullen** Numerieke of rij-konstanten, -variabelen, lijstvariabelen en uitdrukkingen hiermee.

TOEPASSING

Zorgt dat een waarde rechts van het is-gelijk teken aan de variabele links ervan wordt toegekend.

- Voor het toekennen aan een rij-variabele van rij-konstanten of lettertekens dienen de laatste tussen aanhalingstekens (") te worden gezet.
- Het woord LET zelf kan worden weggelaten.
- Wanneer een bepaald soort numerieke gegevens aan een ander soort numerieke variabele wordt toegekend, zullen de gegevens aangepast worden aan het soort variabele.

Voorbeeld gebruik

LET N=N+1 ————— Verhoogt de waarde van N met 1.

```
A%=45.6:PRINT A%
45
```

A\$=3+4
Type mismatch ————— Aangezien hier numerieke gegevens worden toegekend aan een rijvariabele, volgt een foutmelding.

LINE (line)

Trekt een rechte lijn of een vierkant op de voorgrond van het grafische scherm.

SCHRIJFWIJZE

LINE [[STEP] (coördinaten beginpunt)]-[STEP] (coördinaten eindpunt),
{kleur} {
 {I,B}
 {I,BF}}

Coördinaten beginpunt

Invullen Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van -32768 tot 32767.

Weglaten De laatste positie aangegeven in het laatste voorgaande grafische bevel.

Coördinaten eindpunt

Invullen Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van -32768 tot 32767.

Kleur **Invullen** Gehele getallen van 0 tot 15.

Weglaten De bestaande kleur van de voorgrond.

B, BF **Weglaten** Een rechte lijn wordt getrokken.

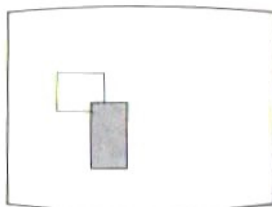
TOEPASSING

Trekt een rechte lijn van een beginpunt met gegeven coördinaten naar een eindpunt met gegeven coördinaten (wanneer B en BF worden weggelaten).

- Wanneer "B" wel wordt ingevoerd, wordt een vierkant getekend waarvan de diagonaal de twee met coördinaten omschreven punten verbindt.
- Wanneer "BF" wordt ingevoerd, wordt een vierkant getekend waarvan de diagonaal de twee met coördinaten omschreven punten verbindt, waarna het omringende gebied gekleurd wordt.
- Zie blz. 164 voor nadere aanwijzingen omtrent STEP.

Voorbeeld gebruik

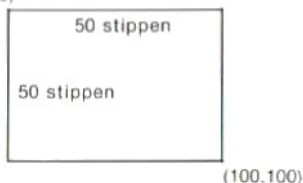
```
10 CLS
20 SCREEN 2
30 LINE (60,60)-(100,100),1,B
40 LINE STEP (-10,-10)-(120,160),8,BF
50 GOTO 50
```



Opmerking

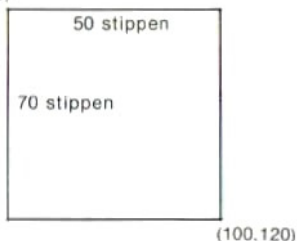
Om een vierkant met gelijke verticale en horizontale zijden op het scherm te tekenen kiest u de coördinaten van beginpunt en eindpunt **zo dat het aantal stippen van de verticale zijde 1,4 maal zo groot is als het aantal stippen van de horizontale zijde.**

Voorbeeld: (50,50)



Het aantal stippen is gelijk voor de verticale zijde en de horizontale zijde.

(50,50)



Het aantal stippen voor de verticale zijde is 1,4 maal het aantal voor de horizontale zijde.

LINE INPUT (line input)

Zorgt dat een regel van maximaal 254 lettertekens, die via het toetsenbord worden ingevoerd, als waarde aan rij-variabelen wordt toegekend.

SCHRIJFWIJZE

LINE INPUT ["invoeraanwijzing"]; variabele

"invoeraanwijzing" **Invullen** Nadere aanwijzing voor de invoer van gegevens.
Weglaten Alleen "?" verschijnt op het scherm, zonder nadere aanwijzing.

variabele **Invullen** Rij-variabelen, lijstvariabelen.

TOEPASSING

De terugkeer-code van de **RETURN** toets wordt beschouwd als niet meer dan een scheidingsteken tussen de gegevens, en dus als einde van een rij lettertekens die via het toetsenbord ingevoerd is en aan een variabele toegekend moet worden. Wanneer zich in een rij lettertekens een komma bevindt, wordt deze als onderdeel hiervan mede aan de variabele toegekend.

Voorbeeld gebruik

```
10 CLS
20 LINE INPUT "NAME,PHONE? ";N$
30 PRINT N$
RUN
NAME,PHONE? JACK,00-11-22
JACK,00-11-22
```

LINE INPUT # (line input number)

Leest een rij van maximaal 254 lettertekens uit een bestand en zorgt dat deze als waarde aan een rij-variabele wordt toegekend.

SCHRIJFWIJZE

LINE INPUT # bestandsnummer, variabele

Bestandsnummer **Invullen** $1 \leq \text{bestandsnummer} \leq \text{nummer gegeven in MAX-FILES} = \text{bevel}$

Variabele **Invullen** Rij-variabelen, lijstvariabelen.

TOEPASSING

Leest rij-gegevens uit een bestand. Bij dit bevel worden, anders dan bij het INPUT # bevel, een spatie, komma of code voor regelopschuiven niet beschouwd als scheidingstekens tussen de gegevens, en een rij lettertekens waarin deze voorkomen wordt in zijn geheel aan een rij-variabele toegekend als rij-waarde. Alleen de code voor terugkeer, gegeven met de **RETURN** toets, geldt bij dit bevel als scheidingstekens.

Voorbeeld gebruik

```
10 OPEN "CAS:DATA" FOR INPUT AS #1
20 IF EOF(1) THEN GOTO 60
30 LINE INPUT #1,A$
40 PRINT A$
50 GOTO 20
60 CLOSE #1:END
```

Wanneer een bestand met de naam DATA en de volgende regels is voorbereid:

```
PRINT #1,"ABC";",",";"DEF"
PRINT #1,"GHI JKL";
PRINT #1,"MNO"
PRINT #1,"PQR"
```

en wanneer dan de gegevens uit het bestand door het bovengenoemde programma worden gelezen en op het scherm worden gezet blijkt dat ze als drie rijen lettertekens beschouwd zijn, als volgt:

```
ABC,DEF
GHI JKLMNO
PQR
```

LIST (list out)

Zet de regels van het programma in de genummerde volgorde op het scherm.

SCHRIJFWIJZE

LIST [nummer beginregel] [-] [nummer laatste regel]

Nummer beginregel **Invullen** Gehele getallen van 0 tot 65529.

Weglaten Kleinste regelnummer

Nummer laatste regel **Invullen** Gehele getallen van 0 tot 65529.

Weglaten Grootste regelnummer

TOEPASSING

Druk op **STOP** om het weergeven van de regels op het scherm tijdelijk te onderbreken.

Druk nogmaals op **STOP** om het weergeven van de lijst te hervatten. Druk op **CTRL** en **STOP** om het geheel te stoppen.

Voorbeeld gebruik

LIST _____ Geeft alle regels weer.

LIST 40 _____ Geeft regel 40 weer.

LIST 20-40 _____ Geeft de regels van 20 t/m 40 weer.

LIST -50 _____ Geeft de regels van het begin t/m regel 50 weer.

LIST 30- _____ Geeft de regels van 30 t/m de laatste regel weer.

LIST . _____ Geeft de laatste regel weer van een eerder LIST bevel of de regel waarbij de verwerking van het programma vanwege een fout onderbroken was.

LLIST (line printer list out)

Laat de regels van het programma door de afdrukeenheid op papier zetten.

SCHRIJFWIJZE

LLIST [nummer beginregel] [-] [nummer laatste regel]

Nummer beginregel **Invullen** Gehele getallen van 0 tot 65529.

Weglaten Kleinste regelnummer

Nummer laatste regel **Invullen** Gehele getallen van 0 tot 65529.

Weglaten Grootste regelnummer

TOEPASSING

Dit bevel komt in het gebruik sterk overeen met het LIST bevel. Het is niet mogelijk tijdens het uitvoeren van het LLIST bevel tevens een lijst op het scherm weer te geven.

● Als een LLIST bevel wordt gegeven terwijl de afdrukeenheid niet is aangesloten of ingeschakeld, stopt de computer en reageert niet meer op invoer via het toetsenbord. U kunt de invoer pas weer hervatten na het tegelijk indrukken van de **CTRL** en **STOP** toetsen.

LOAD (load)

Laadt een BASIC bevel in het geheugen vanaf een nader omschreven apparaat.

SCHRIJFWIJZE

LOAD "apparaatnaam [bestandsnaam]"

Apparaatnaam **Invullen** CAS: ... Cassettespeler.

Bestandsnaam **Invullen** Rij van maximaal 6 lettertekens. Als er 7 of meer letters worden ingevoerd zullen deze vanaf nummer 7 worden genegeerd.

Weglaten Laadt het eerst gevonden bestand.

TOEPASSING

Als voor de apparaatnaam CAS: wordt ingevuld, wordt een programma geladen dat met een SAVE "CAS: bestandsnaam" in ASCII code op een cassette is opgeslagen.

Voorbeeld gebruik

LOAD "CAS:PROG2"

LOCATE (locate)

Verplaatst de cursor naar een nader omschreven plaats.

SCHRIJFWIJZE

LOCATE [X-coördinaat], [Y-coördinaat], [cursor aan/uit]

X-coördinaat **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van 0 tot 39.

Weglaten 0

Y-coördinaat **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van 0 tot 24.

Weglaten 0

Cursor aan/uit **Invullen** 0 ... Cursor wordt niet weergegeven.

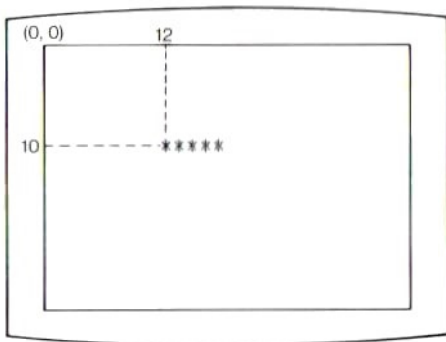
1 ... Cursor wordt weergegeven.

Weglaten 1

TOEPASSING

Voorbeeld gebruik

```
10 CLS
20 LOCATE 12,10
30 PRINT "*****"
```



Functie LOG (natural logarithm)

Geeft de waarde van de natuurlijke logaritme (\log_e).

SCHRIJFWIJZE

LOG (X)

X **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, groter dan 0.

Gegeven waarde: Getalswaarde.

TOEPASSING

De LOG functie geeft de waarde van de natuurlijke logaritme van X, d.w.z. met het grondtal e (2,7182818284588).

● De waarde van een logaritme $\log_a b$ ($b > 0$), waarbij a een positief grondtal is ($a \neq 1$), kan worden verkregen met $\text{LOG}(b)/\text{LOG}(a)$.

Voorbeeld gebruik

```
PRINT LOG(10)
2.302585092994
```

Functie LPOS (line printer position)

Geeft aan op welke plaats in de afdrukbuffer de drukkop van de printer zich bevindt.

SCHRIJFWIJZE

LPOS(X)

X **Invullen** Een willekeurig getal (symbolisch argument).

Gegeven waarde: Geheel getal.

TOEPASSING

Geeft bij het afdrukken met de printer de plaats in het buffergeheugen van het letterteken dat op dat moment wordt afgedrukt. (Start=0).

LPRINT (line print)

Stuurt de waarde van een uitdrukking voor weergave naar de afdrukeenheid.

SCHRIJFWIJZE

LPRINT [uitdrukking] [scheidingsteken] [uitdrukking] [scheidingsteken] [uitdrukking] ...

Uitdrukking **Invullen** Numerieke en rij-konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee.

Weglaten Regel opschuiven

Scheidingsteken **Invullen** , of ;

TOEPASSING

Het LPRINT bevel stuurt gegevens naar de afdrukeenheid om ze op papier te zetten, terwijl het PRINT bevel gegevens op het scherm doet verschijnen. Zie voor nadere bijzonderheden het PRINT bevel.

LPRINT USING (line print using)

Stuurt gegevens naar de afdrukeenheid voor weergave in een nader aangegeven opmaak of schrijfwijze.

SCHRIJFWIJZE

LPRINT USING opmaaksymbool; uitdrukking [uitdrukking]

Uitdrukking **Invullen** Numerieke en rij-konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee.

TOEPASSING

Het LPRINT USING bevel stuurt gegevens naar de afdrukeenheid om ze in een gekozen opmaak af te drukken, terwijl het PRINT bevel gegevens in een gekozen opmaak op het scherm doet verschijnen. Zie voor nadere bijzonderheden zoals de verschillende opmaaksymbolen het PRINT USING bevel.

MERGE (merge)

Laadt een programma dat is opgeslagen in ASCII code, en voegt het samen met een programma in het geheugen.

SCHRIJFWIJZE

MERGE "apparaatnaam [bestandsnaam]"

Apparaatnaam **Invullen** CAS: ... Cassettespeler.

Bestandsnaam **Invullen** Rij van maximaal 6 lettertekens. Als er 7 of meer letters worden ingevoerd zullen deze vanaf nummer 7 worden genegeerd.

Weglaten Laadt het eerst gevonden bestand en voegt dit samen met dat in het geheugen.

TOEPASSING

Voor de apparaatnaam kan alleen CAS: worden ingevuld. Dan wordt een programma geladen dat met een SAVE bevel in ASCII code op een cassette is opgeslagen. Het programma dat zich in het geheugen bevindt, wordt door het MERGE bevel in zijn bestaande vorm met het zojuist geladen programma samengevoegd.

● Als de regelnummers van het programma dat met het MERGE bevel geladen is gelijk zijn aan de regelnummers van het programma dat zich al in het geheugen bevond, dan worden de regelnummers van het eerstgenoemde, pas geladen programma gehandhaafd.

Voorbeeld gebruik

```
MERGE "CAS:PROG3"
```

Functie **MID\$** (middle dollar)

Zoekt en geeft een gedeelte van een rij lettertekens weer.

SCHRIJFWIJZE

MID\$(X\$, M [,N])

X\$ **Invullen** Rij-konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee.

M **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van 1 tot 255.

N **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van 1 tot 255.

Weglaten Geeft alle lettertekens van de rij weer, beginnend met dat op de M-de plaats.

Gegeven waarde: Rij-waarden.

TOEPASSING

Voorbeeld gebruik

```
PRINT MID$("JAPANUKFRANCE",6,2)
UK
```

```
PRINT MID$("JAPANUKFRANCE",6,2.6)
UK
```

} Als N geen geheel getal is, worden alle cijfers achter de decimale punt genegeerd.

```
PRINT MID$("JAPANUK",6,4)
UK
```

} Als de waarde van N groter is dan het aantal lettertekens vanaf dat op de M-de plaats, dan worden te beginnen met "nr.M" alle lettertekens gegeven.

```
PRINT MID$("JAPANUK",12,5)
PRINT MID$("JAPANUK",6,0)
```

} Wanneer de waarde van M groter is dan de lengte van de gehele rij X\$, of wanneer N gelijk is aan 0, dan wordt een lege rij gegeven.

MID\$=Y\$ (middle dollar)

Vervangt een gedeelte van een rij lettertekens door een ander rijtje lettertekens.

SCHRIJFWIJZE

MID\$(X\$,M[,N])=Y\$

X\$, Y\$ **Invullen** Rij-konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee.

M **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van 1 tot 255.

N **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van 1 tot 255.

Weglaten Alle lettertekens, te beginnen met dat op de M-de plaats, worden vervangen door de Y\$ rij.

TOEPASSING

Vervangt een N aantal lettertekens van de X\$ rij, te beginnen met de M-de plaats van links, door de lettertekens van de Y\$ rij. De totale lengte van de X\$ rij verandert bij het uitvoeren van dit bevel niet.

Voorbeeld gebruik

```
10 X$="ABCDEFGG"  
20 Y$="QRSTUVWXYZ"  
30 MID$(X$,4,2)=Y$  
40 PRINT X$  
RUN  
ABCQRFQ
```


MOTOR (motor)

Schakelt de motor van de cassette recorder in en uit.

SCHRIJFWIJZE

MOTOR [{ ON }
 { OFF }]

TOEPASSING

Sluit de "TAPE" uitgang van de computer aan op de afstandsbedienings-aansluiting van een cassette recorder en zet de cassette recorder in de stand voor afspelen. De bandloop start dan met het MOTOR ON bevel en stopt met het MOTOR OFF bevel.

Indien alleen een MOTOR bevel gegeven is, dan heeft dit het effect dat de bandloop met MOTOR wordt ingeschakeld als hij uit stond, en uitgeschakeld als de band liep.

NEW (new)

Wist een BASIC programma uit het geheugen en zet alle variabelen in de beginstand.

SCHRIJFWIJZE

NEW

TOEPASSING

Het NEW bevel wordt gegeven om voor het invoeren van een nieuw programma alle eerdere programma's te wissen en de computer klaar te zetten voor de komende bevelen.

● Een eventueel in het geheugen aanwezig programma in machinetaal blijft bij het uitvoeren van het NEW bevel intact.

Functie **OCT\$** (octonary dollar)

Zet een numerieke waarde om in een rij-waarde, en geeft deze weer in achttallige code.

SCHRIJFWIJZE

OCT\$(X)

X **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van -32768 tot 65535. Bij het invullen van een negatieve waarde is het resultaat hetzelfde als wanneer hier 65536 bij wordt opgeteld.

Gegeven waarde: Rij-waarde

TOEPASSING

Voorbeeld gebruik

```
PRINT OCT$(100)  
144
```

```
PRINT OCT$(65536-32768)  
100000
```

ON ERROR GOTO (on error go to)

Wanneer er een fout optreedt, vervolgt de verwerking met een nader aangegeven regelnummer.

SCHRIJFWIJZE

ON ERROR GOTO regelnummer

Regelnummer **Invullen** Gehele getallen van 0 tot 65529.

TOEPASSING

Dit bevel dient om oponthoud tengevolge van een fout tijdens verwerking van het programma te voorkomen. Wanneer na het ON ERROR GOTO bevel een fout optreedt, vervolgt de verwerking met een in het bevel aangegeven regelnummer. (Ook als de fout optreedt in een rechtstreeks gegeven bevel wordt de regel die in het bevel is aangegeven uitgevoerd.)

Voorbeeld gebruik

```
10 ON ERROR GOTO 100
20 INPUT A
30 B=SQR(A)
40 PRINT "SQR(A)=" ; B
50 END
100 IF ERR=5 AND ERL=30 THEN PRINT "Input a positive number."
110 RESUME 20
```

Dit END bevel scheidt het hoofdprogramma en de foutenherstelroutine.

} — Foutenherstelroutine

Annuleren van een ON ERROR GOTO bevel

Geef het bevel ON ERROR GOTO 0.

ON—GOSUB (on—go to subroutine)

Hiermee wordt de verwerking van het programma, afhankelijk van de waarde van een uitdrukking, afgeleid naar een subroutine beginnend met een nader aangegeven regelnummer.

SCHRIJFWIJZE

ON uitdrukking GOSUB regelnummer [, regelnummer]

Uitdrukking **Invullen** Numerieke variabelen, lijstvariabelen en uitdrukkingen hiermee, van 0 tot 255.

Regelnummer **Invullen** Gehele getallen van 0 tot 65529.

TOEPASSING

```
100 ON X GOSUB 500,600,700
```

Indien in dit programma de waarde van X gelijk is aan 1 wordt de verwerking van het programma afgeleid naar de subroutine die begint met regelnummer 500; als X de waarde 2 heeft, dan wordt vervolgd met de subroutine vanaf regelnummer 600, en als X de waarde 3 heeft wordt vervolgd met de subroutine vanaf regelnummer 700.

Om terug te keren naar het hoofdprogramma moet in de subroutine een RETURN bevel opgenomen zijn.

Waarde van een uitdrukking en resultaat van de verwerking

Indien de waarde van een uitdrukking geen geheel getal is ... Cijfers achter de decimale punt worden genegeerd.

Indien de waarde van de uitdrukking gelijk is aan 0 of, bij lijstvariabelen, groter dan het aantal regelnummers in het GOSUB bevel ... De verwerking vervolgt met de regel volgend op het ON—GOSUB bevel.

Indien de waarde van de uitdrukking negatief is of groter dan 255 ... Volgt een foutmelding.

ON—GOTO (on—go to)

Hiermee wordt de verwerking van het programma, afhankelijk van de waarde van een uitdrukking, afgeleid naar een nader aangegeven regelnummer.

SCHRIJFWIJZE

ON uitdrukking GOTO regelnummer [, regelnummer]

Uitdrukking **Invullen** Numerieke variabelen, lijstvariabelen en uitdrukkingen hiermee, van 0 tot 255.

Regelnummer **Invullen** Gehele getallen van 0 tot 65529.

TOEPASSING

```
100 ON X GOTO 120,130,180
```

Indien in dit programma de waarde van X gelijk is aan 1 wordt de verwerking van het programma afgeleid naar regel nummer 120; als X de waarde 2 heeft, dan wordt vervolgd met regel nummer 130, en als X de waarde 3 heeft wordt vervolgd met regel nummer 180.

Waarde van een uitdrukking en resultaat van de verwerking

Indien de waarde van een uitdrukking geen geheel getal is ... Cijfers achter de decimale punt worden genegeerd.

Indien de waarde van de uitdrukking gelijk is aan 0 of, bij lijstvariabelen, groter dan het aantal regelnummer in het GOSUB bevel ... De verwerking vervolgt met de regel volgend op het ON—GOSUB bevel.

Indien de waarde van de uitdrukking negatief is of groter dan 255 ... Volgt een foutmelding.

ON INTERVAL GOSUB

(on interval go to subroutine)

Door dit bevel wordt de verwerking van het programma bij het intreden van een onderbreking door de ingebouwde schakelklok afgeleid naar een subroutine.

SCHRIJFWIJZE

ON INTERVAL= Intervaltijd GOSUB regelnummer

Intervaltijd **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van -32768 tot 65535 en niet gelijk aan 0.

Regelnummer **Invullen** Gehele getallen van 0 tot 65529.

TOEPASSING

Dit bevel geeft de beginregel van een subroutine waarnaar de verwerking van een programma wordt afgeleid wanneer een onderbreking met een gegeven interval door de ingebouwde schakelklok wordt ingelast. De eenheid van de intervaltijd is ongeveer 1/60 seconde. Met andere woorden, wanneer u een intervaltijd van 60 invoert zal ongeveer elke seconde een onderbreking plaatsvinden (zie 2-7 Onderbrekingen).

Voorbeeld gebruik

```
10 ON INTERVAL=50 GOSUB 100
20 INTERVAL ON
30 SCREEN 2,1
40 SPRITE$(1)=CHR$(&H18)+CHR$(&H3C)+CHR$
(&H66)+CHR$(&HDB)+CHR$(&HE7)+CHR$(&H7E)+
CHR$(&H24)+CHR$(&H42)
50 GOTO 50
100 X=INT(RND(1)*256):Y=INT(RND(1)*192)
110 C=INT(RND(1)*14)+2
120 PUT SPRITE 1,(X,Y),C,1
130 RETURN 50
```

In dit programma vindt door regels 10 en 20 ongeveer elke gehele 1 seconde een onderbreking plaats, en telkens wanneer dat gebeurt gaat verwerking verder met de subroutine die start bij regel 100. Nadat door deze subroutine een patroon van sprites in de vorm van een UFO is afgebeeld, wordt vanwege het RETURN 50 bevel de verwerking van het hoofdprogramma hervat met regel 50.

● Wanneer voor de intervaltijd een negatief getal wordt gekozen heeft dit hetzelfde effect als wanneer bij dit negatieve getal 65536 wordt opgeteld. Dit resulteert dus in een lange intervaltijd.

ON KEY GOSUB (on key go to subroutine)

Leidt de verwerking van het programma af naar een subroutine wanneer met een funktietoets onderbroken is.

SCHRIJFWIJZE

ON KEY GOSUB regelnummer [, regelnummer]

Regelnummer **invullen** Gehele getallen van 0 tot 65529.

TOEPASSING

Dit bevel geeft de beginregel van een subroutine waarnaar de verwerking van een programma wordt afgeleid wanneer een onderbreking wordt veroorzaakt door het indrukken van een funktietoets. Na GOSUB kunnen hierbij maximaal 5 regelnummers worden ingevoerd, gescheiden door komma's, die dan korresponderen met **F1**, **F2**, enz.

Voorbeeld gebruik

```
10 ON KEY GOSUB 1000,,2000
20 KEY(1) ON:KEY(3) ON
```

Na de bovenstaande twee regels van het programma gaat bij het indrukken van funktietoets **F1** de verwerking van het programma verder met de subroutine die start bij regel 1000, en bij indrukken van funktietoets **F3** wordt de subroutine die start bij regel 2000 uitgevoerd.

Voor hervatten van het hoofdprogramma na uitvoeren van de subroutine is een RETURN bevel nodig.

(Zie 2-7 Onderbrekingen)

ON SPRITE GOSUB

(on sprite go to subroutine)

Leidt de verwerking van het programma af naar een subroutine wanneer door het overlappen van een sprite patroon onderbroken is.

SCHRIJFWIJZE

ON SPRITE GOSUB regelnummer [, regelnummer]

Regelnummer **invullen** Gehele getallen van 0 tot 65529.

TOEPASSING

Dit bevel geeft de beginregel van een subroutine waarnaar de verwerking van een programma wordt afgeleid wanneer een onderbreking wordt veroorzaakt door het overlappen van een patroon van sprites.

Voorbeeld gebruik

```
10 ON SPRITE GOSUB 1000
20 SPRITE ON
```

Na da bovenstaande twee regels van het programma gaat bij het overlappen door een patroon van sprites de verwerking van het programma verder met de subroutine die start bij regel 1000. Voor hervatten van het hoofdprogramma na uitvoeren van de subroutine is een RETURN bevel nodig.

(Zie 2-4 Grafische voorstellingen en afbeeldingen, en 2-7 Onderbrekingen)

ON STOP GOSUB (on stop go to subroutine)

Leidt de verwerking van het programma af naar een subroutine wanneer met de **CTRL** en **STOP** toetsen onderbroken is.

SCHRIJFWIJZE

ON STOP GOSUB regelnummer [, regelnummer]

Regelnummer **Invullen** Gehele getallen van 0 tot 65529.

TOEPASSING

Dit bevel geeft de beginregel van een subroutine waarnaar de verwerking van een programma wordt afgeleid wanneer een onderbreking wordt veroorzaakt door het indrukken van de **CTRL** en **STOP** toetsen.

Voorbeeld gebruik

```
10 ON STOP GOSUB 1000
20 STOP ON
```

Na de bovenstaande twee regels van het programma gaat bij het tegelijk indrukken van de **CTRL** en **STOP** toetsen de verwerking van het programma verder met de subroutine die start bij regel 1000. Voor hervatten van het hoofdprogramma na uitvoeren van de subroutine is een RETURN bevel nodig.

(Zie 2-7 Onderbrekingen)

Voorzorgsmaatregelen

Na het uitvoeren van een subroutine moet een programma op enigerlei wijze beëindigd worden. Het onderstaande programma kan slechts beëindigd worden door het indrukken van de **RESET** toets.

```
10 ON STOP GOSUB 100
20 STOP ON
30 PRINT "MAIN ROUTINE"
40 GOTO 40
100 PRINT "CTRL+STOP EXECUTED"
110 RETURN 30
```

ON STRIG GOSUB

(on stick trigger go to subroutine)

Leidt de verwerking van het programma af naar een subroutine wanneer door het indrukken van de spatiebalk of de trekkerknop van een joystick onderbroken is.

SCHRIJFWIJZE

ON STRIG GOSUB regelnummer [, regelnummer]

Regelnummer **Invullen** Gehele getallen van 0 tot 65529.

TOEPASSING

Dit bevel geeft de beginregel van een subroutine waarnaar de verwerking van een programma wordt afgeleid wanneer een onderbreking wordt veroorzaakt door het indrukken van de spatiebalk of de trekkerknop van een joystick. Na GOSUB kunnen hierbij maximaal 5 regelnummers worden ingevoerd, gescheiden door komma's.

ON STRIG GOSUB regelnummer ①, regelnummer ②, regelnummer ③, regelnummer ④, regelnummer ⑤.

Regelnummer ① ... Wordt als volgende uitgevoerd na het indrukken van de spatiebalk.

Regelnummer ② ... Joystick 1, trekkerknop 1

Regelnummer ③ ... Joystick 2, trekkerknop 1

Regelnummer ④ ... Joystick 1, trekkerknop 2

Regelnummer ⑤ ... Joystick 2, trekkerknop 2

Voorbeeld gebruik

```
10 ON STRIG GOSUB 1000,2000,3000
20 STRIG(0) ON:STRIG(1) ON:STRIG(2) ON
```

Bij het indrukken van de spatiebalk gaat de verwerking van het programma verder met de subroutine die start bij regel 1000. Bij het indrukken van trekkerknop 1 van joystick 1 gaat de verwerking verder met de subroutine die start bij regel 2000, en bij het indrukken van trekkerknop 1 van joystick 2 wordt de subroutine die start bij regel 3000 uitgevoerd.

Voor hervatten van het hoofdprogramma na uitvoeren van de subroutine is een RETURN bevel nodig.

(Zie 2-7 Onderbrekingen)

OPEN (open)

Opent een bestand en bepaalt hoe dit verwerkt wordt.

SCHRIJFWIJZE

OPEN "apparaatnaam [bestandsnaam]" FOR verwerking AS [#]
bestandsnummer

| | | |
|----------------|-----------------|--|
| Apparaatnaam | Invullen | CAS: ... Cassetterecorder CRT: ... Tekst-beeldscherm GRP: ... Grafisch beeldscherm LPT: ... Afdrukeenheid |
| Bestandsnaam | Invullen | Rij van maximaal 6 lettertekens. Als 7 of meer lettertekens worden ingevoerd, worden deze, te beginnen met het 7de, genegeerd. |
| | Weglaten | Lege rij |
| Verwerking | Invullen | OUTPUT ... Uitvoer, weergave INPUT ... Invoer |
| Bestandsnummer | Invullen | $1 \leq \text{bestandsnummer} \leq \text{nummer gegeven in MAXFILES} = \text{bevel}$ |

TOEPASSING

Met het OPEN bevel opent u een bestand met een nader omschreven bestandsnaam, om dit via een eveneens nader omschreven apparaat in of uit te voeren. Aangezien de apparaatnamen CRT:, GRP: en LPT: alle staan voor apparaten die alleen voor uitvoer kunnen dienen, kunt u voor de verwerking hiermee uitsluitend OUTPUT invullen. Met de apparaatnaam CAS: heeft u de keuze tussen in- en uitvoer, aan te geven met INPUT, resp. OUTPUT.

- De bestandsnaam die u gebruikt voor het invoeren van een bestand kunt u daarna ook gebruiken voor uitvoer of weergave van het bestand.
- Het bestandsnummer dient kleiner dan of gelijk te zijn aan het getal dat in het MAXFILES= bevel is gegeven om het maximaal aantal te openen bestanden aan te geven.

Voorbeeld gebruik

```
10 SCREEN 2
20 OPEN "GRP:" FOR OUTPUT AS #1
30 PSET (120,90)
40 PRINT #1,"ABC"
50 GOTO 50
```

Dit programma dient voor het op het grafisch beeldscherm (SCREEN 2) zetten van lettertekens.

(Zie 2-6 Verwerken van bestanden)

OUT (out)

Voert 1 byte gegevens uit, via een nader omschreven I/O poort.

SCHRIJFWIJZE

OUT poortnummer, uitdrukking

Poortnummer, uitdrukking **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van 0 tot 255.

TOEPASSING

Dit bevel dient om gegevens rechtstreeks naar een I/O poort te sturen. Zie blz. 208 voor de toewijzing van I/O poorten.

Functie PAD (pad)

Bepaalt het gebruik van het aanraakpaneel.

SCHRIJFWIJZE

PAD(N)

N **Invullen** Gehele getallen van 0 tot 7.
Gegeven waarde: getalswaarde.

TOEPASSING

De diverse in te vullen waarden voor N bepalen welke gegevens het aanraakpaneel bij aanraking doorgeeft. Wanneer N gelijk is aan 0, 1, 2 of 3 staat dit voor het gebruik van een aanraakpaneel aangesloten op regeling A. Wanneer N gelijk is aan 4, 5, 6 of 7 staat dit voor het gebruik van een aanraakpaneel aangesloten op regeling B.

| Waarde van N | Betekenis van een gegeven waarde |
|--------------|---|
| 0 of 4 | 0: Niet aangeraakt - 1: Wordt aangeraakt |
| 1 of 5 | X-coördinaat van de aangeraakte plaats |
| 2 of 6 | Y-coördinaat van de aangeraakte plaats |
| 3 of 7 | 0: Schakelaar wordt niet ingedrukt - 1: Schakelaar wordt ingedrukt |

PAINT (paint)

Kleurt een gebied omsloten door een randlijn.

SCHRIJFWIJZE

PAINT [STEP] (X-coördinaat, Y-coördinaat), [kleur vlak], [kleur lijn]

X-coördinaat **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van 0 tot 255.

Y-coördinaat **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van 0 tot 191.

Kleur vlak, kleur lijn

Invullen Gehele getallen van 0 tot 15.

Weglaten Bestaande kleur voorgrond.

TOETASSING

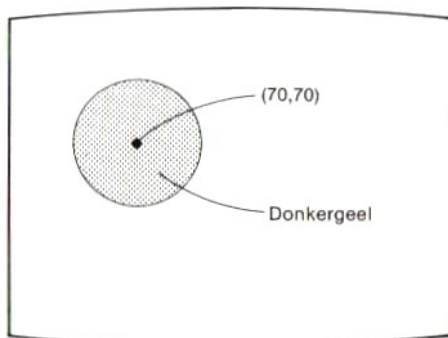
Kleurt een vlak een gekozen kleur binnen een randlijn van een gekozen kleur die door een punt loopt waarvan de X- en de Y-coördinaat zijn ingevoerd.

- Als de randlijn geen geheel gesloten kromme vormt wordt het gehele scherm gekleurd.
- Als bij het grafisch beeldscherm met hoge resolutie (SCREEN 2) de kleur van het vlak en de kleur van de randlijn niet gelijk zijn, dan wordt het gehele scherm gekleurd.
- Zie blz. 164 voor de mogelijke invoer voor STEP.

Voorbeeld gebruik

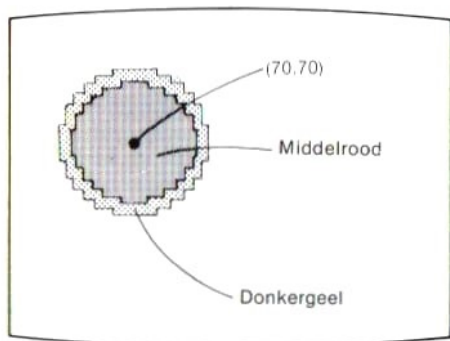
```
10 CLS
20 SCREEN 2
30 CIRCLE (70,70),40,10
40 PAINT (70,70),10,10
50 GOTO 50
```

Bij het grafische scherm (SCREEN 2) moet de zelfde kleur worden ingevoerd voor het vlak en de randlijn.



```
10 CLS
20 SCREEN 3
30 CIRCLE (70,70),40,10
40 PAINT (70,70),8,10
50 GOTO 50
```

Bij SCREEN 3 kunnen voor het vlak en de randlijn verschillende kleuren worden gekozen



Functie **PDL** (paddle)

Geeft de waarde ingevoerd met een zgn. peddel.

SCHRIJFWIJZE

PDL (N)

N **Invullen** Gehele getallen van 1 tot 12.

Gegeven waarde: Getalswaarde van 0 tot 255.

TOEPASSING

Geeft de waarde ingevoerd met een zgn. peddel weer als numerieke gegevens. Wanneer N een oneven getal is, dan komen de gegevens van de peddel aangesloten op regelingang A. Wanneer N een even getal is, dan komen de gegevens van de peddel aangesloten op regelingang B.

Functie **PEEK** (peek)

Geeft de inhoud van een nader omschreven geheugenadres.

SCHRIJFWIJZE

PEEK (adres)

Adres **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van - 32768 tot 65535. Bij het invullen van een negatieve waarde is het resultaat hetzelfde als wanneer hier 65536 bij wordt opgeteld.

Gegeven waarde: Getalswaarde, met decimalen.

TOEPASSING

Voorbeeld gebruik

$M = \text{PEEK}(50000)$ ————— Vindt de inhoud van geheugenadres 50000 en kent deze als waarde aan variabele M toe.

PLAY (play)




Geeft een geluid weer volgens de aanwijzingen in een aantal deel-instructies.

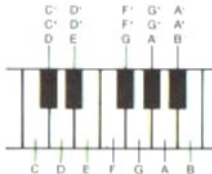





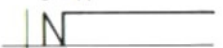

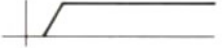

SCHRIJFWIJZE

PLAY deel-instructie

Deel-instructie **Invullen** Rij lettertekens (konstanten) tussen " " of rij-variabelen en uitdrukkingen hiermee. Hoofdletters of kleine letters.

Deel-instructies

| Instructie | Invullen | Betekenis |
|----------------|---|---|
| Tn (tempo) | Gehele getallen $32 \leq n \leq 255$ | Geeft het tempo van de muziek aan. De waarde van n geeft een minuut lang het tellen van een kwart noot. De beginwaarde is T120. |
| On (oktaaf) | Gehele getallen $1 \leq n \leq 8$ | Geeft een van de acht beschikbare oktaven aan. Bij het invullen van O4 wordt muziek gespeeld in het hieronder aangegeven toonbereik.  Voor een lagere waarde van n wordt een lagere oktaaf gekozen en voor een hogere waarde van n een hogere oktaaf. De beginwaarde is O4. |
| Ln (lengte) | $1 \leq n \leq 64$ | Geeft de lengte van een noot aan.  De beginwaarde is L4. |
| Nn (noot) | $0 \leq n \leq 96$ | Geeft de toon aan. N36 is  N0 is een rust. Bij verhogen van n wordt de toon ook hoger, volgens de chromatische toonladder. |

| Instructie | Invullen | Betekenis |
|----------------|--------------------|--|
| A—G An—Gn | $1 \leq n \leq 64$ | <p>Geeft de toon aan binnen de gekozen oktaaf.</p>  <p>Voor halve tonen kunt u # (of+) en – gebruiken. Hierbij kan ook de lengte van het geluid worden aangegeven met n. (C4 betekent hetzelfde als L4C.) Bij weglaten wordt de lengte bepaald door Ln.</p> |
| Rn (rust) | $1 \leq n \leq 64$ | <p>Geeft een rust aan.</p>  |
| . | | <p>Geeft een stip in muzieknotatie aan. Een noot met een stip wordt 1,5 maal zo lang aangehouden. C4.= $\underline{\cdot}$. R4.= $\underline{\cdot}$.</p> |
| Vn (volume) | $0 \leq n \leq 15$ | <p>Geeft het volume aan. Hoe groter n, des te groter ook het volume. De beginstand is V8.</p> |
| Sn (soort) | $0 \leq n \leq 15$ | <p>Kiest een golfvorm-patroon uit een van de volgende vormen.</p> <p>Magnitude S=0, 1, 2, 3, 9</p>  <p>S = 4, 5, 6, 7, 15</p>  <p>S = 8</p>  <p>S = 10</p>  <p>S = 11</p>  <p>S = 12</p>  <p>S = 13</p>  <p>S = 14</p>  <p>De beginwaarde is S1. Door het combineren van de deelinstrukties S en M zijn talloze geluiden mogelijk.</p> |

| Instructie | Invullen | Betekenis |
|------------|-----------------------|--|
| Mn | $1 \leq n \leq 65535$ | Geeft de lengte aan van de cyclus waarin de met S gekozen vorm wordt herhaald. Hoe groter n, des te langer de cyclus. De beginwaarde is M255. |

TOEPASSING

```
PLAY "T8003L4CDEFG2.RAB04CDC2."
```

Met het bovenstaande bevel worden de volgende tonen gespeeld.

Uitdrukken van een deelinstruktie met een variabele

```
M$="T8003L4CDEFG2.RAB04CDC2."
PLAY M$
```

Nadat aldus een deelinstruktie is toegewezen aan de rij-variabele M\$ kan deze variabele M\$ in het PLAY bevel als deelinstruktie gebruikt worden.

Deels uitdrukken van een deelinstruktie met een variabele (X variabele);

```
10 M$="CDEFG2.R"
20 PLAY "04L4X M$;GAGAG2.R"
30 PLAY "X M$;AB05CDC2."
```

Wanneer een deelinstruktie toegewezen aan een variabele in een PLAY bevel tussen " " wordt gebruikt, dient hij voorafgegaan te worden door X en gevolgd te worden door ;. In het bovenstaande voorbeeld wordt de deelinstruktie die is toegewezen aan de rij-variabele M\$ in twee PLAY bevelen gebruikt.

Uitdrukken van de n in een deelinstruktie met een variabele (=variabele;)

In een PLAY bevel kan de n van elke deelinstruktie een konstante of een variabele zijn. Als hiervoor een variabele gebruikt wordt dient deze voorafgegaan te worden door = en gevolgd te worden door ; .

```
10 FOR I=1 TO 8  
20 PLAY "O=I;CEG"  
30 NEXT I
```

Dit programma speelt een zelfde wijsje in 8 oktaven van PLAY "O1CEG" tot PLAY "O8CEG".

Spelen van akkoorden

De mogelijkheid bestaat in een bevel maximaal drie deelinstrukties tegelijk te spelen, zoals bijvoorbeeld

```
PLAY A$, B$, C$
```

```
10 A$="O4CD03B04E2R4"  
20 B$="O4EFDG2R4"  
30 C$="O4GAG05C2R4"  
40 PLAY A$, B$, C$
```

Dit programma speelt de volgende noten.



(Zie 2-5 Geluid.)

Funktie **PLAY** (play)

Om te controleren of de gekozen muziek al dan niet wordt gespeeld.

SCHRIJFWIJZE

PLAY (N)

N **Invullen** Gehele getallen van 0 tot 3.

Gegeven waarde: getalswaarde

TOEPASSING

In een PLAY bevel kunnen drie verschillende geluiden tegelijk worden gespeeld.

In het geval van PLAY A\$, B\$, C\$;

Het geluid van deelinstructie A\$ wordt uitgevoerd via kanaal 1, het geluid van deelinstructie B\$ via kanaal 2 en het geluid van deelinstructie C\$ via kanaal 3.

De PLAY funktie dient hierbij om te controleren of de gegevens in het buffergeheugen voor muziek van kanaal 1 aanwezig zijn wanneer $N=1$, idem voor kanaal 2 wanneer $N=2$ en voor kanaal 3 wanneer $N=3$. Indien de gegevens in het buffergeheugen aanwezig zijn wordt -1 gegeven, en zo niet dan wordt 0 gegeven. Wanneer $N=0$ wordt de logische som (OR) van de bufferstatus van alle kanalen gegeven. Met andere woorden, wanneer een van de drie -1 is wordt -1 gegeven.

Functie **POINT** (point)

Geeft de kleurcode van een stip op een nader omschreven plaats.

SCHRIJFWIJZE

POINT (X, Y)

X, Y **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van -32767 tot 32767.

Gegeven waarde: Getalswaarde (wanneer de gekozen plaats buiten het afbeeldingsgebied valt wordt -1 gegeven.)

TOEPASSING

Voorbeeld gebruik

```
10 SCREEN 3
20 FOR I=1 TO 250
30 X=INT(RND(1)*255)
40 Y=INT(RND(1)*191)
50 PSET (X,Y),1
60 NEXT I
70 FOR Y=0 TO 191 STEP 4
80 FOR X=0 TO 255 STEP 4
90 C=POINT(X,Y)
100 IF C=4 THEN PSET (X,Y),15
110 NEXT X,Y
120 GOTO 120
```

Voor de plaats met de coördinaten (X, Y) wordt de kleurcode in regel 90 toegekend aan variabele C, waarna deze in regel 100 gewijzigd wordt in wit indien C gelijk is aan 4 (donkerblauw).

POKE (poke)

Schrijft gegevens in een nader aangegeven geheugenadres.

SCHRIJFWIJZE

POKE adres, uitdrukking

Adres **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van -32768 tot 65535. Bij het invullen van een negatieve waarde is het resultaat hetzelfde als wanneer hier 65536 bij wordt opgeteld.

Uitdrukking **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van 0 tot 255.

TOEPASSING

Voorbeeld gebruik

POKE 50000 ,255 ——— Schrijft 255 als gegeven in geheugenadres 50000.

POKE &HD000 ,&HAB ——— Schrijft A8H als gegeven in geheugenadres D000H.

Functie **POS** (position)

Geeft de X-coördinaat van de plaats waar de cursor zich bevindt.

SCHRIJFWIJZE

POS (X)

X **Invullen** Een willekeurige getalswaarde (symbolisch argument)
Gegeven waarde: Geheel getal

TOEPASSING

Voorbeeld gebruik

```
10 INPUT A$
20 PRINT A$; : X=POS (X)
30 IF X>=5 THEN CLS
40 PRINT:GOTO 10
```

De waarde van de X-coördinaat van de plaats waar de cursor staat wordt in regel 20, X=POS (X), aan de variabele X toegekend. Resultaat hiervan is dat bij het invoeren van een rij van meer dan 5 lettertekens voor de variabele A\$ het scherm gewist wordt.

PRESET (point reset)

Zet of wist een stip op het grafische scherm.

SCHRIJFWIJZE

PRESET [STEP] (X-coördinaat, Y-coördinaat) [, kleur]

X; Y-coördinaat **Invullen** Numerieke constanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van -32768 tot 32767.

Kleur **Invullen** Gehele getallen van 0 tot 15.

Weglaten Bestaande kleur achtergrond.

TOEPASSING

Wanneer u de kleurcode uit dit bevel weglaat, wordt bij het uitvoeren een stip op het grafisch scherm gezet met de zelfde kleur als de achtergrond. Met andere woorden, als er op die plaats iets op het scherm stond met een andere kleur dan de achtergrond, dan wordt de indruk gegeven dat op die plaats een gebied ter grootte van een stip is gewist.

- Wanneer een kleurcode is ingevoerd heeft dit bevel hetzelfde effect als een PSET bevel met kleurcode.
- Zie blz. 164 voor de mogelijke invoer voor STEP.
- Zie voor een voorbeeld van het gebruik onder PSET.

PRINT (print)

Geeft numerieke gegevens of lettertekens op het scherm weer.

SCHRIJFWIJZE

PRINT uitdrukking [scheidingsteken] [uitdrukking] [scheidingsteken] [uitdrukking]

Uitdrukking **Invullen** Numerieke of rij-constanten, variabelen, lijstvariabelen en uitdrukkingen hiermee.

Scheidingsteken **Invullen** Komma (,) of puntkomma (;).

TOEPASSING

Schrijfwijze uitdrukkingen (gegevens)

Numerieke constanten en numerieke en rij-variabelen worden zo, zonder meer, geschreven, terwijl rij-constanten tussen aanhalingstekens (" ") geschreven worden.

Funktie scheidingstekens

Gegevens die onderling worden gescheiden door een komma (,) komen op het scherm met een tussenruimte van 14 spaties lengte, terwijl gegevens gescheiden door een puntkomma (;) op het scherm direkt gevolgd worden door de volgende gegevens.

Als aan het einde van de invoer geen scheidingstekens gegeven wordt, schuift na het laatste gegeven de weergave op het scherm een regel op. Als aan het einde wel een scheidingsteken ingevoerd wordt zullen de gegevens van het eerstvolgende PRINT bevel op de zelfde regel volgen zonder dat een regel opgeschoven wordt.

Numerieke gegevens en het teken hiervan

Het teken dat bij numerieke gegevens aanduidt of deze positief of negatief zijn wordt in het geval van een "+" weggelaten en in het geval van een "-" voor het gegeven op het scherm gezet. (Bij het gebruikmaken van het ";" scheidingstekens tussen positieve numerieke gegeven wordt op het scherm een ruimte van twee spaties tussen de gegevens gelaten als plaats voor het teken.)

Korter bevel

Hetzelfde resultaat wordt verkregen als u voor dit bevel in plaats van het woord PRINT een vraagteken "?" invoert.

Voorbeeld gebruik

```
10 A$="ABC" : B$="DEF"
20 PRINT A$;B$
30 PRINT A$,B$
40 PRINT
50 PRINT "MSX"
60 PRINT +50,-50
70 ?"PERSONAL COMPUTER"
RUN
ABCDEF _____ Resultaat van regel 20
ABC           DEF _____ Resultaat van regel 30
_____ Resultaat van regel 40
MSX_____ Resultaat van regel 50
 50           -50 _____ Resultaat van regel 60
PERSONAL COMPUTER _____ Resultaat van regel 70
```

PRINT USING (print using)

Geeft gegevens op het scherm weer in een nader omschreven opmaak.

SCHRIJFWIJZE

PRINT USING opmaaksymbool; uitdrukking [uitdrukking] ...

Uitdrukking **Invullen** Rij- en numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee.

TOEPASSING

De waarde van een uitdrukking wordt door gebruik van een opmaaksymbool weergegeven in een gekozen opmaak.

Opmaaksymbolen voor rijen lettertekens

| Symbool | Opmaak gegevens en voorbeeld weergave |
|----------------------------|--|
| "!" | Geeft alleen het eerste letterteken weer. PRINT USING "!" ; "United", "Nation" UN |
| "\ _ \\" ↑ n spaties | Geeft n+2 lettertekens weer. Wanneer de lengte van de gegevens minder bedraagt dan n+2 lettertekens wordt de rest aangevuld met spaties. PRINT USING "\ _ \" ; "ABCDEF", "GHI", "JKLMN" ABCDGHI JKLM |
| "&" | Geeft de hele rij lettertekens weer. 10 A\$="North":B\$="South" 20 PRINT USING "& Pole ";A\$,B\$ RUN North Pole South Pole |

Opmaaksymbolen voor numerieke gegevens

| Symbool | Opmaak gegevens en voorbeeld weergave |
|---------|--|
| " # " | <p>Plaats bij het weergeven van getalswaarden een cijfer voor elke #. De decimale punt dient door "." ingevoerd te worden.</p> <pre>PRINT USING "POINT:###.#";123.4 POINT:123.4</pre> <ul style="list-style-type: none"> ● Wanneer het aantal cijfers kleiner is dan het aantal ingevoerde " #" wordt de getalswaarde weergegeven met evenredige marges ter weerszijden, en wanneer het aantal cijfers groter is, wordt ervoor een "%" geplaatst. <pre>10 PRINT USING "####";12 20 PRINT USING "####";12345 RUN 12 %12345</pre> ● Wanneer het aantal cijfers achter de komma in een getalswaarde kleiner is dan het aantal ingevoerde " #" wordt het aangevuld met "0", en wanneer het groter is, wordt het afgerond op het dichtsbijzijnde gehele getal. <pre>10 PRINT USING "##.##";25.3 20 PRINT USING "##.##";25.345 RUN 25.30 25.35</pre> <p>Bij het weergegeven van numerieke gegevens wordt het "+" teken genegeerd en het "-" teken als een cijfer meegeteld.</p> <pre>10 PRINT USING "###";+123 20 PRINT USING "###";-123 RUN 123 %-123</pre> |
| " + " | <p>Bij het weergeven van positieve getalswaarden verschijnt "+" en bij negatieve getalswaarden "-" naar keuze voor of achter de numerieke gegevens.</p> <pre>10 PRINT USING "+####";123,-123 20 PRINT USING "####+";123,-123 RUN +123 -123 123+ 123-</pre> |
| " - " | <p>"-" wordt aan het eind van een negatieve getalswaarde weergegeven.</p> <pre>PRINT USING "###-";123,-123 123 123-</pre> |

| | |
|----------------------|--|
| <pre>"**"</pre> | <p>De ruimte voor numerieke gegevens wordt opgevuld met "*" tekens. Een enkel "*" opmaaksymbool staat voor een enkel cijfer.</p> <pre>10 PRINT USING "#####";123 20 PRINT USING "#####";-234 RUN ****123 ***-234</pre> |
| <pre>"££"</pre> | <p>Geeft "£" voor numerieke gegevens weer. Een "£" opmaaksymbool telt als een cijfer.</p> <pre>10 PRINT USING "££###";1234 20 PRINT USING "+££###";-1234 RUN £1234 -£1234</pre> |
| <pre>"**£"</pre> | <p>Geeft "£" op de plaats direct voor de getalswaarde, en de ruimte daarvoor wordt opgevuld met "*" tekens.</p> <pre>PRINT USING "**£###.##";12.34 ***£12.34</pre> |
| <pre>","</pre> | <p>Als de komma ergens voor de decimale punt wordt ingevoerd, verschijnt deze na elke drie cijfers, vanaf de decimale punt naar links toe gerekend.</p> <pre>PRINT USING "#,#####.##";12345.67 12,345.67</pre> |
| <pre>"^ ^ ^ ^"</pre> | <p>Geeft numerieke gegevens weer als getalswaarden met drijvende komma, volgens de wetenschappelijke notatie. "^ ^ ^ ^" staat hierbij voor de cijfers van het exponent-gedeelte.</p> <pre>PRINT USING "##.##^ ^ ^ ^";234.56 2.35E+02</pre> |

PRINT # (print number)

Schrijft gegevens naar een bestand dat is geopend met een OPEN bevel.

SCHRIJFWIJZE

PRINT # bestandsnummer, uitdrukking

Bestandsnummer **Invullen** $1 \leq \text{bestandsnummer} \leq \text{nummer gegeven in MAX-FILES= bevel}$

Uitdrukking **Invullen** Rij- en numerieke konstanten, variabelen, lijst-variabelen en uitdrukkingen hiermee.

TOEPASSING

Regelt de uitvoer van gegevens naar een bestand dat is geopend met een OPEN bevel.

Voorbeeld gebruik

```
10 OPEN "CAS:DATA" FOR OUTPUT AS #1 — Opent een bestand voor het
20 FOR I=0 TO 4                               invoeren van gegevens.
30 READ A$
40 PRINT #1,A$;" "; ————— Schrijft de gegevens naar het bestand.
50 NEXT I
60 CLOSE #1
70 DATA TOKYO,LONDON,PARIS,PEKING,NEW YORK
```

Dit programma schrijft de gegevens in regel 70 met dezelfde volgorde naar een bestand op cassette met de naam "DATA".
(Zie 2-6 Verwerken van bestanden.)

PRINT # USING (print number using)

Schrijft gegevens naar een bestand dat is geopend met een OPEN bevel, in een nader omschreven opmaak.

SCHRIJFWIJZE

PRINT # bestandsnummer USING opmaaksymbool; uitdrukking

Bestandsnummer **Invullen** $1 \leq \text{bestandsnummer} \leq \text{nummer gegeven in MAX-FILES=}$ bevel

Uitdrukking **Invullen** Rij- en numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee.

TOEPASSING

Met dit bevel kan de opmaak van gegevens die worden uitgevoerd naar een bestand geregeld worden. Zie onder PRINT USING voor het gebruik van de opmaaksymbolen.

PSET (point set)

Zet een stip op het grafische scherm.

SCHRIJFWIJZE

PSET [STEP] (X-coördinaat, Y-coördinaat) [, kleur]

X, Y-coördinaat **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van -32768 tot 32767.

Kleur **Invullen** Gehele getallen van 0 tot 15.

Weglaten Bestaande kleur voorgrond.

TOEPASSING

Voorbeeld gebruik

```
10 SCREEN 2
20 FOR X=0 TO 255
30 PSET (X+1,100) —— Zet een stip.
40 PRESET (X,100) —— Wist een eerder gezette stip.
50 NEXT X
```

Zie blz. 164 voor de mogelijkheden met STEP.

PUT SPRITE (put sprite)

Geeft een nader omschreven patroon van sprites weer op een gekozen plaats van een sprite-vlak naar keuze.

SCHRIJFWIJZE

PUT SPRITE spritevlak-nummer [[STEP] (X-coördinaat, Y-coördinaat)], [kleur], [sprite-nummer]

Sprite-vlaknummer **Invullen** Gehele getallen van 0 tot 31.

X-coördinaat **Invullen** Numerieke constanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van -32 tot 255.

Y-coördinaat **Invullen** Numerieke constanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van -32 tot 191.

STEP (X-coördinaat, Y-coördinaat)

Weglaten Vorige plaats, gegeven in het voorgaande grafische bevel.

Kleur **Invullen** Gehele getallen van 0 tot 15.

Weglaten Bestaande kleur voorgrond.

Sprite-nummer **Invullen** Voor 8×8 stippen een getal van 0 tot 255.

Voor 16×16 stippen een getal van 0 tot 63.

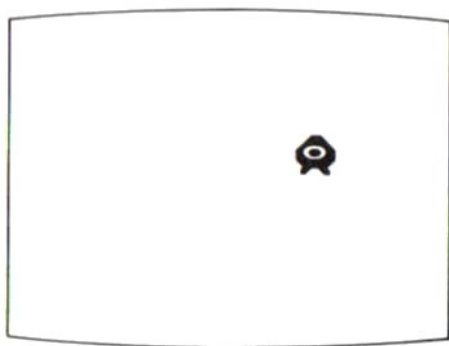
Weglaten Gelijk aan het spritevlak-nummer.

TOEPASSING

Voorbeeld gebruik

```
10 SCREEN 2
20 SPRITE$(1)=CHR$(&H18)+CHR$(&H3C)+CHR$
  (&H66)+CHR$(&HDB)+CHR$(&HE7)+CHR$(&H7E)+
  CHR$(&H24)+CHR$(&H42)
30 X=0:Y=0:DX=1:DY=1
40 PUT SPRITE 0,(X,Y),,1
50 X=X+DX:Y=Y+DY
60 IF X>250 OR X<0 THEN DX=-DX
70 IF Y>190 OR Y<0 THEN DY=-DY
80 GOTO 40
```

In regel 20 wordt een patroon van sprites in de vorm van een U FO. toege-
kend aan het spritenummer 1. Het patroon van sprites wordt door het
PUT SPRITE bevel van regel 40 op het scherm weergegeven. Het
spritevlak-nummer is 0. Aangezien de kleur is weggelaten wordt hiervoor
dezelfde waarde aangehouden als gebruikt voor de ingestelde voor-
grondkleur. Door het wijzigen van de waarden voor X en Y, de coördinaten
die de plaats aangeven, lijkt het of de U FO. over het scherm vliegt.



(Zie 2-4 Grafische voorstellingen en afbeeldingen.)

READ (read)

Leest de gegevens die met een DATA bevel zijn ingevoerd.

SCHRIJFWIJZE

READ variabele [, variabele] [, variabele]

Variabele **Invullen** Numerieke of rij-variabele.

TOEPASSING

Leest de ingevoerde gegevens op volgorde, beginnend met het eerste gegeven in het laagst genummerde DATA bevel in het programma, en wijst de gegevens in dezelfde volgorde toe aan de variabelen in het READ bevel.

- Meerdere variabelen in een READ bevel moeten gescheiden worden door komma's (,).
- Het soort variabele, numerieke of rij-variabele, dient overeen te komen met het soort gegevens in het DATA bevel.

```
10 READ A,B,C,D$,E$
20 PRINT A,B,C,D$,E$
100 DATA 5,10,20,ABC,XYZ
```

- Wanneer in een programma meerdere READ bevelen voorkomen, begint het tweede READ bevel de gegevens te lezen vanaf het eerste gegeven dat niet door het voorgaande READ bevel gelezen is.
- Na een RESTORE bevel keert het eerstvolgende READ bevel terug naar het laagst genummerde DATA bevel volgend op het RESTORE bevel, en begint daar met lezen van de gegevens.

Voorbeeld gebruik

```
10 READ A,B,C
20 READ D$,E$
30. PRINT A;B;C;D$;E$
100 DATA 10,20,30,ABC,DEF
RUN
10 20 30 ABCDEF
```

REM (remark)

Voor het in een programma plaatsen van een opmerking ter verduidelijking.

SCHRIJFWIJZE

REM opmerking

TOEPASSING

Het REM bevel is niet bedoeld voor de computer, maar uitsluitend om het programma in menselijke zin beter leesbaar te maken.

Voorbeeld gebruik

```
10 REM MUSIC _____
20 PLAY "T60CEGEC1"
```

Alle REM bevelen worden bij het op het scherm zetten (met LIST) van de regels van het programma wel weergegeven, maar bij het verwerken van het programma door de computer worden zij genegeerd.

```
10 'MUSIC _____
20 PLAY "T60CEGEC1"
```

In plaats van het woord REM kan ook een apostrof (') gebruikt worden.

```
10 PRINT "MSX":REM Output
20 PRINT "PERSONAL COMPUTER" 'Output
RUN
MSX
PERSONAL COMPUTER
```

Indien u een REM bevel na een ander bevel op dezelfde regel invoert dienen de twee gescheiden te worden door een dubbele punt (:). Bij gebruik van de apostrof (') in plaats van het woord REM is dit echter niet noodzakelijk.

RENUM (renumber)

Voor het opnieuw nummeren van de regels van een programma.

SCHRIJFWIJZE

RENUM [nieuw nummer beginregel], [oud nummer beginregel], [verhoging]

Nieuw nummer beginregel **Invullen** Gehele getallen van 0 tot 65529

Weglaten 10

Oud nummer beginregel **Invullen** Gehele getallen van 0 tot 65529

Weglaten Kleinste bestaande regelnummer.

Verhoging **Invullen** Gehele getallen van 0 tot 65529.

Weglaten 10

TOEPASSING

Om het programma na een correctie door opnieuw nummeren meer overzichtelijk te maken.

● Ook de regelnummers die zijn gebruikt voor de "sprongen" in een programma met GOTO of GOSUB bevelen worden met het RENUM bevel automatisch op de juiste wijze opnieuw genummerd. Als het regelnummer, gegeven in een GOTO of dergelijk bevel als bestemming voor een sprong, op het moment dat het RENUM bevel wordt gegeven nog niet werkelijk als regelnummer bestaat, wordt het niet automatisch gewijzigd en kan er een fout optreden.

Voorbeeld gebruik

```
RENUM _____ Nummert alle regels opnieuw vanaf regel 10 met een stap  
of verhoging van 10.  
RENUM 100.,.100 _____ Nummert alle regels opnieuw vanaf regel 100 met een  
stap of verhoging van 100.  
RENUM 100 _____ Nummert alle regels opnieuw vanaf regel 100 met een  
stap of verhoging van 10.  
RENUM 100,38,20 _____ Nummert alle regels vanaf de bestaande regel 38 op-  
nieuw vanaf regel 100, met een stap of verhoging van 20.  
LIST _____ Zet alle regels van het programma in een lijst op  
het scherm.  
15 FOR I=0 TO 10  
20 A=A+1  
23 PRINT A  
35 NEXT I  
Ok  
RENUM _____ Nummert alle regels opnieuw.  
Ok  
LIST _____ Zet opnieuw een lijst op het scherm.  
10 FOR I=0 TO 10  
20 A=A+1  
30 PRINT A  
40 NEXT I
```

RESTORE (restore)

Maakt een reeds gelezen DATA bevel weer geschikt voor lezen met een volgend READ bevel.

SCHRIJFWIJZE

RESTORE [regelnummer]

Regelnummer **Invullen** Gehele getallen van 0 tot 65529.

Weglaten Laagst genummerde DATA bevel wordt weer toegankelijk gemaakt.

TOEPASSING

Het RESTORE bevel kan tussentijds gegeven worden wanneer een reeks gegevens in een programma meerdere malen gelezen moet worden.

Na een RESTORE bevel keert het eerstvolgende READ bevel terug naar het laagst genummerde DATA bevel volgend op het regelnummer gegeven in het RESTORE bevel, en begint daar met lezen van de gegevens.

Voorbeeld gebruik

```
10 READ A,B,C
20 READ D,E,F
30 RESTORE 110
40 READ G,H,I
50 PRINT A;B;C;D;E;F;G;H;I
100 DATA 10,20,30
110 DATA 40,50,60
RUN
10 20 30 40 50 60 40 50 60
```


RESUME (resume)

Hervat de verwerking van het hoofdprogramma nadat een routine voor het herstellen van fouten geheel doorlopen is.

SCHRIJFWIJZE

RESUME [{ ⁰regelnummer }
NEXT]

Regelnummer **Invullen** Gehele getallen van 0 tot 65529.

Weglaten Regel waarin de fout voorkwam.

TOEPASSING

Voorbeeld gebruik

RESUME 0 of RESUME _____ Keert terug naar het bevel waarin een fout was opgetreden.

RESUME 100 _____ Keert terug naar regel 100

(Zie ook het programmavoorbeeld onder ON ERROR GOTO.)

Functie **RIGHT\$** (right dollar)

Geeft een nader aangeduid aantal lettertekens vanaf de rechterkant van een langere rij als rij-gegevens weer.

SCHRIJFWIJZE

RIGHT\$ (X\$, N)

X\$ Invullen Rij-konstanten, -variabelen, lijstvariabelen en uitdrukkingen hiermee.

N Invullen Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van 0 tot 255.

Gegeven waarde: Rij-waarden.

TOEPASSING

Voorbeeld gebruik

```
PRINT RIGHT$("I LOVE TOKYO",5)
TOKYO
```

```
PRINT RIGHT$("I LOVE TOKYO",5.3)
TOKYO
```

} — Als N geen geheel getal is worden de cijfers achter de decimale punt genegeerd.

```
PRINT RIGHT$("I LOVE TOKYO",0)
Ok
```

} — Als N gelijk is aan 0 wordt een lege rij gegeven.

Functie **RND** (random)

Geeft een willekeurig positief getal kleiner dan 1 (kan ook 0 zijn).

SCHRIJFWIJZE

RND (X)

X Invullen Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee.

Gegeven waarde: Getalswaarde.

TOEPASSING

Wanneer X groter is dan 0

Willekeurige getallen worden altijd in dezelfde volgorde gegenereerd.

```

10 FOR N=1 TO 10
20 RPINT RND(1)
30 NEXT N
RUN
.59521943994623
.10658628050158
.76597651772823
.57756392935958
.73474759503023
.18426812909758
.37075377905223
.94954151651558
.63799556899423
.47041117641358

```

Wanneer X negatief is

Willekeurige getallen worden gegenereerd in een reeks behorend bij de waarde van X, en daarna in willekeurige volgorde binnen die reeks.

```

10 PRINT RND(-1)
20 FOR N=1 TO 10
30 PRINT RND(N)
40 NEXT N
RUN
.04389820420821
.0962486816692
.21069655852301
.3265173630504
.47775124336581
.3409147084636
.12971184081661
.0977770174288
.35157860175541
.835389696666
.63902641386221

```

Wanneer X gelijk is aan 0

Geeft een willekeurig getal dat gelijk is aan het laatst gegenereerde getal.

```

10 PRINT RND(1)
20 PRINT RND(0)
30 PRINT RND(-1)
40 PRINT RND(0)
RUN
.59521943994623
.59521943994623
.04389820420821
.04389820420821

```

RUN (run)

Verwerkt een programma vanaf een nader aangeduide regel.

SCHRIJFWIJZE

RUN [regelnummer]

Regelnummer **Invullen** Gehele getallen van 0 tot 65529.

Weglaten Verwerkt het programma vanaf de beginregel.

TOEPASSING

Door het uitvoeren van het RUN bevel worden eerst alle variabelen in de beginstand gezet (numerieke variabelen krijgen de waarde 0, en rij-variabelen worden lege rijen). Vervolgens wordt het programma verwerkt en na voltooiing wordt gewacht op een volgend bevel.

● De verwerking van een programma kan tijdelijk worden onderbroken door indrukken van de **STOP** toets. Nogmaals indrukken hiervan doet de verwerking hervatten.

Druk de **CTRL** en **STOP** tegelijk in om de verwerking van een programma te staken. Daarna kan de verwerking hervat worden door het geven van een CONT bevel.

SAVE (save)

Voor het opslaan van een BASIC bestand op een nader omschreven apparaat.

SCHRIJFWIJZE

SAVE "apparaatnaam [bestandsnaam]"

Apparaatnaam **Invullen** CAS: ... Cassetterecorder

CRT: ... Tekst-beeldscherm

GRP: ... Grafisch beeldscherm

LPT: ... Afdrukeenheid

Bestandsnaam **Invullen** Rij van maximaal 6 lettertekens. Als 7 of meer lettertekens worden ingevoerd, worden deze, te beginnen met het 7de, genegeerd.

Weglaten Lege rij

TOEPASSING

Wanneer voor de apparaatnaam CAS: wordt ingevuld, wordt een BASIC programma uit het geheugen in ASCII code op cassette opgeslagen.

Voorbeeld gebruik

```
SAVE "CAS:PROG2"
```

● Een programma dat u later met een MERGE bevel wilt samenvoegen met een programma in het geheugen moet hiervoor in ASCII code worden opgeslagen.

SCREEN (screen)

Voor het kiezen van het soort scherm (de modus), het formaat van sprites, het intoetssignaal, de snelheid in baud voor in- en uitvoer van en naar cassette, en het soort afdrukeenheid.

SCHRIJFWIJZE

SCREEN [modus], [formaat sprite], [intoetssignaal], [snelheid in baud], [soort afdrukeenheid]

Modus **Invullen** 0, 1, 2 of 3.

Weglaten Bestaande modus.

Formaat sprite **Invullen** 0, 1, 2 of 3.

Weglaten Eerder gekozen formaat.

Intoetssignaal **Invullen** 0 of gehele getallen van 1 tot 255.

Weglaten Eerder gekozen toestand (aan/uit).

Snelheid in baud **Invullen** 1 of 2.

Weglaten Eerder gekozen snelheid.

Soort afdrukeenheid **Invullen** 0 of gehele getallen van 1 tot 255.

Weglaten Eerder gekozen soort afdrukeenheid.

Modus

| Ingevoerde waarde | Modus of soort scherm |
|-------------------|--|
| 0 | 40 lettertekens × 24 regels tekst-scherm |
| 1 | 32 lettertekens × 24 regels tekst-scherm |
| 2 | Gedetailleerd grafisch scherm |
| 3 | Grafisch scherm met meerdere kleuren |

Formaat sprite

| Ingevoerde waarde | Formaat |
|-------------------|-----------------------------------|
| 0 | 8 × 8 stippen zonder vergroting |
| 1 | 8 × 8 stippen met vergroting |
| 2 | 16 × 16 stippen zonder vergroting |
| 3 | 16 × 16 stippen met vergroting |

Intoetssignaal

| Ingevoerde waarde | Geluidssignaal bij indrukken toets |
|-------------------|------------------------------------|
| 0 | Nee |
| Ongelijk aan 0* | Ja |

*Bereik van 1 tot 255.

Snelheid in baud

| Ingevoerde waarde | Snelheid in baud* |
|-------------------|-------------------|
| 1 | 1200 baud |
| 2 | 2400 baud |

*Snelheid bij in- of uitvoeren van gegevens van en naar cassette.

Soort afdrukeenheid

| Ingevoerde waarde | Soort afdrukeenheid |
|-------------------|---------------------------------|
| 0 | MSX afdrukeenheid** |
| Ongelijk aan 0* | Andere dan MSX afdrukeenheid*** |

* Bereik van 1 tot 255.

** Een afdrukeenheid die bruikbaar is met een MSX persoonlijke computer, geschikt voor het afdrucken van grafische tekens.

***Andere dan MSX afdrukeenheden zetten spaties in de plaats van grafische tekens.

Weglaten invoer en instellen beginstand

Bij het weglaten van invoer voor de verschillende parameters wordt de bestaande toestand gehandhaafd. De oorspronkelijke of beginstand is als volgt:

Modus of soort scherm: 40 lettertekens \times 24 regels tekstschermb
(WIDTH 37)

Formaat sprite: 8 \times 8 zonder vergroting

Intoetssignaal: Geluidssignaal bij indrukken toets

Snelheid in baud: 1200 baud

Soort afdrukeenheid: MSX afdrukeenheid

Voorbeeld gebruik

10 SCREEN 0,,1 ——— 40 lettertekens \times 24 regels tekst-schermb, met intoetssignaal.

10 SCREEN ,, ,2 ——— De gekozen snelheid is 2400 baud.

10 SCREEN 2,3 ——— Grafisch scherm met hoge resolutie. Formaat van sprites is 16 \times 16 stippen met vergroting.

```
10 SCREEN 2
20 FOR I=0 TO 255
30 PSET (I,100)
40 NEXT I
50 GOTO 50
```

Nadat de verwerking van een programma voltooid is keert het scherm terug naar de tekst-modus (SCREEN 0 of 1). Daarom dient u voor het handhaven van het grafische scherm in het programma een bevel als in regel 50 van het bovenstaande voorbeeld opnemen.

Voor het staken van de verwerking van een programma drukt u tegelijk de **CTRL** en **STOP** toetsen in.

Functie **SGN** (sign)

Geeft 1 wanneer een numeriek gegeven positief is, 0 wanneer het gelijk aan 0 is, en -1 wanneer het negatief is.

SCHRIJFWIJZE

SGN (X)

X **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee.

Gegeven waarde: Geheel getal.

TOEPASSING

Voorbeeld gebruik

```
10 INPUT A
20 IF SGN(A)=-1 THEN PRINT "Negative"
30 GOTO 10
```

"Negative" in regel 20 wordt alleen weergegeven als de aan A toegekende waarde negatief is.

Functie **SIN** (sine)

Geeft de waarde van de sinus van een numeriek gegeven.

SCHRIJFWIJZE

SIN (X)

X **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee. (Eenheid: radialen)

Gegeven waarde: Konstanten met drijvende komma van -1 tot 1 .

TOEPASSING

Voorbeeld gebruik

```
PRINT SIN(3.14/3)
.86575983949239
```

```
PRINT SIN(60*3.14/180)
.86575983949239
```

● Om X als waarde in graden in te voeren, kunt u de formula $\text{SIN}(X * \pi/180)$ gebruiken.

SOUND (sound)

Brengt geluidseffecten voort door het rechtstreeks invoeren van gegevens in het PSG (Programmable Sound Generator) register.

SCHRIJFWIJZE

SOUND registernummer, uitdrukking

Registernummer **Invullen** Gehele getallen van 0 tot 13.

Uitdrukking **Invullen** Konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, binnen het vastgestelde bereik voor elk register.

PSG registerfuncties en bereik van in te voeren gegevens

| Registernummer | Functie | Bereik gegevens |
|----------------|---|---|
| 0 | Frekwentie kanaal A | 0—255 |
| 1 | | 0—15 |
| 2 | Frekwentie kanaal B | 0—255 |
| 3 | | 0—15 |
| 4 | Frekwentie kanaal C | 0—255 |
| 5 | | 0—15 |
| 6 | Ruisfrekwentie | 0—31 |
| 7 | Kiest een kanaal voor generatie van tonen en ruis | 0—63 |
| 8 | Volume kanaal A | 0—15 Bij invoeren van 16 treedt variatie van het volume op |
| 9 | Volume kanaal B | |
| 10 | Volume kanaal C | |
| 11 | Frekwentie van het patroon van volumevariaties | 0—255 |
| 12 | | 0—255 |
| 13 | Keuze van het patroon van volumevariaties | 0—14 |

TOEPASSING

Voorbeeld gebruik

| | | | | | |
|----|-------|------|---|---|---|
| 10 | SOUND | 0,56 | } | — | Stelt de frekventie van kanaal A in op 400 Hz. |
| 20 | SOUND | 1,1 | | | |
| 30 | SOUND | 7,62 | — | — | Kiest kanaal A voor het weergeven van een toon. |
| 40 | SOUND | 8,8 | — | — | Stelt het volume van kanaal A in. |

Bij het verwerken van dit programma wordt een ononderbroken toon van 400 Hz weergegeven.

Om deze te onderbreken drukt u tegelijk de **CTRL** en **STOP** toetsen in. (Zie 2-5 Geluid.)

Functie **SPACE\$** (space dollar)

Geeft een gekozen aantal spaties weer binnen een rij lettertekens.

SCHRIJFWIJZE

SPACE\$ (N)

N **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van 0 tot 255.

Gegeven waarde: Rij-waarde.

TOEPASSING

Voorbeeld gebruik

```
PRINT SPACE$(5) ; "ABC"
```

ABC
└──────────┘ 5 spaties

● Wanneer N geen geheel getal is, worden de cijfers achter de decimale punt genegeerd.

Functie **SPC** (space)

Geeft een gekozen aantal spaties weer.

SCHRIJFWIJZE

SPC (N)

N **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van 0 tot 255.

Gegeven waarde: Rij-waarde.

TOEPASSING

De SPC functie kan alleen gebruikt worden in PRINT en LPRINT bevelen.

Voorbeeld gebruik

```
PRINT "ABC" ; SPC(10) ; "DEF"
```

ABC ─────────── DEF
└────────────────┘ 10 spaties

● Wanneer N geen geheel getal is, worden de cijfers achter de decimale punt genegeerd.

SPRITE ON SPRITE OFF SPRITE STOP

Zorgt dat een onderbreking veroorzaakt door het overlappen van sprites ingeschakeld, uitgeschakeld of vastgehouden wordt.

SCHRIJFWIJZE

- SPRITE ON —Onderbreking ingeschakeld.
- SPRITE OFF —Onderbreking uitgeschakeld.
- SPRITE STOP—Onderbreking vastgehouden.

TOEPASSING

Nadat een onderbreking door het overlappen van sprites in het programma is aangegeven met het ON SPRITE GOSUB bevel, dient dit bevel om de onderbreking daadwerkelijk in te schakelen (PRINT ON), uit te schakelen (SPRITE OFF) of vast te houden (SPRITE STOP). (Zie 2-4 Grafische voorstellingen en afbeeldingen, en 2-7 Onderbrekingen.)

Functie **SQR** (square root)

Geeft de vierkantswortel uit numerieke gegevens.

SCHRIJFWIJZE

SQR (X)

X **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, groter dan 0.

Gegeven waarde: Getalswaarde.

TOEPASSING

Voorbeeld gebruik

```
PRINT SQR (100)  
10
```

Functie **STICK** (stick)

Geeft de richting aan bij het gebruik van de cursortoetsen of joysticks.

SCHRIJFWIJZE

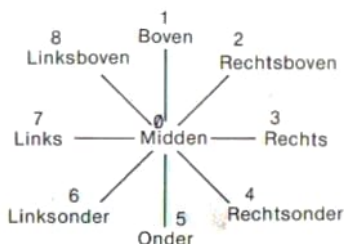
STICK (N)

N **Invullen** 0, 1 of 2.

Gegeven waarde: Geheel getal.

TOEPASSING

Geeft de richting voor de cursortoetsen wanneer N=0, voor joystick 1 wanneer N=1 en voor joystick 2 wanneer N=2. Het bereik van de gegeven waarde die de richting aangeeft loopt van 0 tot 8. Wanneer geen van de cursortoetsen wordt ingedrukt, en wanneer het pookje van de joystick precies in de middenpositie staat, wordt 0 gegeven.



Voorbeeld gebruik

```
10 CLS
20 X=14
30 LOCATE X,10:PRINT " ";
40 D=STICK(0)
50 IF D=0 THEN LOCATE X,10:PRINT "*"
60 IF D=3 THEN X=X+1:IF X>28 THEN X=28
70 IF D=7 THEN X=X-1:IF X<0 THEN X=0
80 LOCATE X,10:PRINT "*";
90 GOTO 30
```

Dit programma doet een "*" naar links en rechts over het scherm bewegen door indrukken van de linker en rechter cursor toetsen. De waarde van variabele D in regel 40 is afhankelijk van of al dan niet een cursortoets wordt ingedrukt. De gegeven waarde bepaalt in regel 50, 60 en 70 de X-coördinaat van de plaats waar "*" op het scherm verschijnt.

STOP (stop)

Onderbreekt de verwerking van een programma.

SCHRIJFWIJZE

STOP

TOEPASSING

Door het geven van het STOP bevel wordt de in gang zijnde verwerking van een programma onderbroken.

● De verwerking kan hierna hervat worden vanaf de regel na het onderbroken bevel door het rechtstreeks geven van het CONT bevel.

STOP ON (stop on)

STOP OFF (stop off)

STOP STOP (stop stop)

Zorgt dat een onderbreking door het indrukken van de **CTRL** en **STOP** toetsen ingeschakeld, uitgeschakeld resp. vastgehouden wordt.

SCHRIJFWIJZE

STOP ON —Onderbreking ingeschakeld.

STOP OFF —Onderbreking uitgeschakeld.

STOP STOP—Onderbreking vastgehouden.

TOEPASSING

Nadat een onderbreking door het indrukken van de **CTRL** en **STOP** toetsen in het programma is aangegeven met het ON STOP GOSUB bevel, dient dit bevel om de onderbreking daadwerkelijk in te schakelen (STOP ON), uit te schakelen (STOP OFF) of vast te houden (STOP STOP).

(Zie 2-7 Onderbrekingen.)

Functie STRIG (stick trigger)

Geeft -1 wanneer de spatiebalk of de trekkerknop van een joystick wordt ingedrukt, en geeft 0 wanneer deze niet ingedrukt worden.

SCHRIJFWIJZE

STRIG (N)

N **Invullen** Gehele getallen van 0 tot 4.

Gegeven waarde: Geheel getal.

TOEPASSING

Geeft met N=0 aan of de spatiebalk al dan niet wordt ingedrukt, met N=1 of N=3 de stand van de trekkerknop van joystick 1 en met N=2 of N=4 de stand van de trekkerknop van joystick 2. Wanneer de spatiebalk of de gekozen knop niet worden ingedrukt is de gegeven waarde 0 en wanneer ze wel worden ingedrukt is de gegeven waarde -1.

Voorbeeld gebruik

```
10 CLS
20 COLOR ,C,C
30 IF STRIG(0)=0 THEN GOTO 20
40 C=C+1:IF C>15 THEN C=0
50 GOTO 20
```

Dit programma wijzigt de kleur van het scherm telkens wanneer de spatiebalk wordt ingedrukt.

STRIG ON (stick trigger on)

STRIG OFF (stick trigger off)

STRIG STOP (stick trigger stop)

Zorgt dat een onderbreking met een ingebouwde schakelklok ingeschakeld, uitgeschakeld resp. vastgehouden wordt.

SCHRIJFWIJZE

STRIG(n) ON —Onderbreking ingeschakeld.

STRIG(n) OFF —Onderbreking uitgeschakeld.

STRIG(n) STOP—Onderbreking vastgehouden.

n **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van 0 tot 4.

TOEPASSING

Geeft aan dat de spatiebalk of de trekkerknop van joystick 1 of 2, nader aangeduid door de waarde van n, gebruikt kan worden voor een onderbreking. Het regelnummer van de subroutine die bij onderbreking de volgende bestemming vormt dient te zijn aangegeven met een ON STRIG GOSUB bevel.

| Waarde van n | Kiest |
|--------------|--------------------------|
| 0 | Spatiebalk |
| 1 | Joystick 1 trekkerknop 1 |
| 2 | Joystick 2 trekkerknop 1 |
| 3 | Joystick 1 trekkerknop 2 |
| 4 | Joystick 2 trekkerknop 2 |

Voorbeeld gebruik

STRIG(0) ON ————— Schakelt een onderbreking met de spatiebalk in.

STRIG(1) OFF — Schakelt een onderbreking met trekkerknop 1 van joystick 1 uit.

STRIG(2) STOP — Houdt een onderbreking met trekkerknop 1 van joystick 2 vast.

(Zie 2-7 Onderbrekingen.)

Functie **STR\$** (convert to string)

Zet numerieke gegevens om in rij-gegevens.

SCHRIJFWIJZE

STR\$(X)

X **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee.

Gegeven waarde: Rij-waarde

TOEPASSING

Wanneer numerieke gegevens negatief zijn wordt het eerste letterteken van de gegeven rij een min-teken. Wanneer de getalswaarde positief is of gelijk aan 0, wordt het eerste letterteken van de gegeven rij een spatie.

Voorbeeld gebruik

```
10 X=100:Y=200
20 X$=STR$(X):Y$=STR$(Y)
30 PRINT X+Y
40 PRINT X$+Y$
RUN
  300
 100 200
  X$  Y$
```


Functie **STRING\$** (string dollar)

Geeft het letterteken behorend bij een ingevoerde ASCII lettercode of het eerste van een rij lettertekens een gekozen aantal malen weer als een rij lettertekens.

SCHRIJFWIJZE

STRING\$ (N, J)

STRING\$ (N, X\$)

N **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van 0 tot 255.

J **Invullen** De gewenste ASCII lettercode (Zie de tabel van lettercodes op blz. 198.)

X\$ **Invullen** Rij-konstanten, -variabelen, -lijstvariabelen en uitdrukkingen hiermee.

Gegeven waarde: Rij-waarde.

TOEPASSING

Voorbeeld gebruik

```
PRINT STRING$(10,70)
FFFFFFFF
```

```
PRINT STRING$(5,"ABC")
AAAAA
```

SWAP (swap)

Verwisselt onderling de waarde van twee variabelen.

SCHRIJFWIJZE

SWAP variabele, variabele

variabele **Invullen** Numerieke of rij-variabelen en -lijstvariabelen. De beide variabelen moeten van hetzelfde type zijn.

TOEPASSING

Voorbeeld gebruik

```
10 A=3 : B=5
20 SWAP A, B
30 PRINT "A=" ; A
40 PRINT "B=" ; B
RUN
A= 5
B= 3
```

Functie TAB (tab)

Verplaatst de cursor van het begin van de regel over een nader omschreven aantal plaatsen naar rechts.

SCHRIJFWIJZE

TAB(N)

N **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van 0 tot 255.

TOEPASSING

De TAB tabulatorfunctie kan alleen gebruikt worden in combinatie met het PRINT of het LPRINT bevel. Wanneer N gelijk is aan 0 komt de cursor aan de uiterste linkerkant te staan, en wanneer N gelijk is aan het aantal lettertekens op een regel min 1 komt de cursor aan de uiterste rechterkant te staan.

Voorbeeld gebruik

```
PRINT TAB(5) ; "AAA"
  |
  | AAA
  |_____ 5 plaatsen open
```

Functie **TAN** (tangent)

Geeft de waarde van de tangens van een numeriek gegeven.

SCHRIJFWIJZE

TAN (X)

X **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee. (Eenheid: radialen)

Gegeven waarde: Konstante met drijvende komma.

TOEPASSING

Voorbeeld gebruik

```
PRINT TAN(3.14/3)  
1.72992922009
```

```
PRINT TAN(60*3.14/180)  
1.72992922009
```

● Om X als waarde in graden in te voeren, kunt u de formule $TAN(X * \pi/180)$ gebruiken.

TROFF (trace off)

Schakelt TRON en daarmee het op het scherm zetten van verwerkte regelnummers uit.

SCHRIJFWIJZE

TROFF

TOEPASSING

Wanneer na een eerder gegeven TRON bevel het tegengestelde TROFF bevel rechtstreeks of in een programma gegeven wordt, zullen de regelnummers niet meer op het scherm verschijnen.

TRON (trace on)

Geeft aan welk regelnummer verwerkt, welk bevel uitgevoerd is.

SCHRIJFWIJZE

TRON

TOEPASSING

Na het rechtstreeks of in een programma geven van een TRON bevel wordt het nummer van de hierna verwerkte regel op het tekstschermbinnen [] aangegeven. Dit kan zeer nuttig zijn voor het opsporen en corrigeren van fouten enz.

● Als door een SCREEN bevel het grafische scherm is ingeschakeld wordt het regelnummer niet aangegeven.

Voorbeeld gebruik

```
10 TRON
20 FOR I=0 TO 3
30 A=I+1:PRINT A
40 NEXT I
50 TROFF
RUN
[20] [30] 1
[40] [30] 2
[40] [30] 3
[40] [30] 4
[40] [50]
```

Functie **USR** (user)

Geeft het resultaat na uitvoering van een subroutine in machinetaal die begint bij een adres gegeven in een DEFUSR bevel.

SCHRIJFWIJZE

USR [X] (I)

X **Invullen** Gehele getallen van 0 tot 9.

Weglaten 0

I **Invullen** Numerieke of rij-konstanten, -variabelen en -lijstvariabelen.
Gegeven waarde: Afhankelijk van de gebruikersfunctie.

TOEPASSING

X is het nummer van een gebruikersprogramma. Hiervoor wordt het nummer gegeven in het DEFUSR bevel gebruikt. I is een variabele of konstante waarvan de waarde overgebracht wordt van BASIC naar de subroutine.

Voorbeeld gebruik

```
DEFUSR0=&HE000
```

```
X=USR0 (I)
```

Door deze beide bevelen wordt eerst de subroutine vanaf het adres &HE000 verwerkt, waarna de resulterende waarde wordt overgebracht naar het BASIC hoofdprogramma.
(Zie 2-8 Machinetaal subroutines.)

Functie **VAL** (value)

Geeft rij-gegevens als numerieke gegevens, getalswaarden, weer.

SCHRIJFWIJZE

VAL (X\$)

X\$ **Invullen** Rij-konstanten, -variabelen, -lijstvariabelen en uitdrukkingen hiermee, voor zover deze uit cijfers bestaan.

Gegeven waarde: Getalswaarde.

TOEPASSING

Voorbeeld gebruik

```
PRINT VAL ("5")  
5
```

```
PRINT VAL (" 5")
```

} — Eventuele spaties vooraan de rij-gegevens worden genegeerd.

Functie **VARPTR** (variable pointer)

Geeft het beginadres in het geheugen waar de gegevens die zijn toegekend aan een nader omschreven variabele zijn opgeslagen.

SCHRIJFWIJZE

VARPTR (variabele)

variabele **Invullen** Numerieke en rij-variabelen en -lijstvariabelen.

TOEPASSING

Geeft in een tientallige ("normale") getalswaarde het beginadres in het geheugen waar de waarde toegekend aan een nader omschreven variabele is opgeslagen. Deze waarde kan van -32768 tot 32767 lopen. Als de gegeven waarde negatief is, kunt u de plaats van het adres vinden door hier 65536 bij op te tellen. De **VARPTR** functie kan bijvoorbeeld gebruikt worden om de gegevens in een geheugenadres te gebruiken in een subroutine in machinetaal.

Voorbeeld gebruik

```
10 A%=15
20 X=VARPTR (A%)
30 M$=HEX$(X):PRINT M$
RUN
C032
```

Dit programma zoekt het adres in het geheugen op waar de waarde toegekend aan een variabele (A%) is opgeslagen, zet het adres om in het hexadecimale talstelsel en geeft het op het scherm weer.

Functie **VPEEK** (video RAM peek)

Leest de gegevens in het video RAM geheugen.

SCHRIJFWIJZE

VPEEK (adres)

Adres **Invullen** Gehele getallen van 0 tot 16383.

TOEPASSING

Geeft de gegevens die zijn opgeslagen in een nader omschreven adres in het video RAM geheugen weer.

Aangezien het basisadres van elke tabel gevonden kan worden met de BASE functie, kunt u deze BASE functie gebruiken voorafgaand aan de VPEEK functie om het gewenste adres in het video RAM geheugen te vinden.

VPOKE (video RAM poke)

Schrijft 1 byte gegevens naar het video RAM geheugen.

SCHRIJFWIJZE

VPOKE adres, uitdrukking

Adres **Invullen** Gehele getallen van 0 tot 16383.

Uitdrukking **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van 0 tot 255.

TOEPASSING

Schrijft willekeurige gegevens naar een nader omschreven adres in het video RAM geheugen.

Aangezien het basisadres van elke tabel gevonden kan worden met de BASE functie, kunt u deze BASE functie gebruiken voorafgaand aan de VPOKE functie om het gewenste adres in het video RAM geheugen te vinden.

WAIT (wait)

Wacht tot de invoer via een nader omschreven I/O poort een gekozen waarde bereikt.

SCHRIJFWIJZE

WAIT poortnummer, uitdrukking 1, [, uitdrukking 2]

Poortnummer, uitdrukking 1, uitdrukking 2

Invullen Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van 0 tot 255.

TOEPASSING

Na het geven van een WAIT bevel worden gegevens ingevoerd via een nader omschreven I/O poort, dan wordt hiermee en met de waarde van uitdrukking 2 een logische exclusieve OF-bewerking uitgevoerd, en vervolgens wordt met het resultaat en de waarde van uitdrukking 1 een EN-bewerking uitgevoerd en het daaruit resulterende logisch produkt gegeven. Als de aldus verkregen waarde gelijk is aan 0 wordt het invoeren van gegevens via de I/O poort voortgezet, en als de waarde niet gelijk is aan 0 wordt doorgedaan met het volgende regelnummer. Uitdrukking 2 kan weggelaten worden en hiervoor wordt dan de waarde 0 aangehouden.

WIDTH (width)

Geeft voor het tekstscherf het aantal lettertekens per regel aan.

SCHRIJFWIJZE

WIDTH (aantal lettertekens)

Aantal lettertekens **Invullen** Voor het SCREEN 0 tekstscherf een geheel getal van 1 tot 40. Voor het SCREEN 1 tekstscherf een geheel getal van 1 tot 32.

TOEPASSING

Voorbeeld gebruik

```
SCREEN 0  
WIDTH 40
```

Hierdoor wordt voor het SCREEN 0 tekstscherf een aantal van 40 lettertekens per regel aangehouden.

1-2 SPECIALE VARIABELEN EN FUNKTIES

BASE (base)

Voor het lezen en schrijven van het basisadres van een videotabel.

SCHRIJFWIJZE

BASE (N)

BASE (N)=uitdrukking

N Invullen Gehele getallen van 0 tot 19.

Uitdrukking Invullen Gehele getallen van 0 tot 65535.

TOEPASSING

Voor het lezen of herschrijven van het basisadres van een VDP videotabel in het geheugen.

BASE (N) komt overeen met een van de basisadressen in onderstaande tabel, afhankelijk van de waarde van N.

| Waarde van N | Tabel |
|--------------|---|
| 0 | 40 lettertekens × 24 lijnen tekstschermpatroonnaam tabel |
| 2 | 40 lettertekens × 24 lijnen tekstschermpatroongenerator tabel |
| 5 | 32 lettertekens × 24 lijnen tekstschermpatroonnaam tabel |
| 6 | 32 lettertekens × 24 lijnen tekstschermkleuren tabel |
| 7 | 32 lettertekens × 24 lijnen tekstschermpatroongenerator tabel |
| 8 | 32 lettertekens × 24 lijnen tekstschermsprite kenmerk tabel |
| 9 | 32 lettertekens × 24 lijnen tekstschermsprite patroon tabel |
| 10 | Gedetailleerd grafisch scherm patroonnaam tabel |
| 11 | Gedetailleerd grafisch scherm kleuren tabel |
| 12 | Gedetailleerd grafisch scherm patroongenerator tabel |
| 13 | Gedetailleerd grafisch scherm sprite kenmerk tabel |
| 14 | Gedetailleerd grafisch scherm sprite patroon tabel |

| Waarde van N | Tabel |
|--------------|--|
| 15 | Meerkleurenscherm patroonnaam tabel |
| 17 | Meerkleurenscherm patroongenerator tabel |
| 18 | Meerkleurenscherm sprite kenmerk tabel |
| 19 | Meerkleurenscherm sprite patroon tabel |

N=1, 3, 4, 16 worden niet gebruikt.

Opgelet

Het register van de TMS9918A videostuureenheid ofwel het LSI circuit dat de weergave op het beeldscherm regelt kan evenals het basisadres van de tabel direkt gewijzigd worden m.b.v. een BASE variabele en een VDP variabele. Er is echter wel een grondige kennis van het TMS9918A register voor nodig om dit naar behoren uit te voeren. Als het basisadres slordig of met vergissingen herschreven wordt kan er geen sprake zijn van normale beeldschermweergave. Bezint eer ge begint ...

MAXFILES (maxfiles)

Geeft aan hoeveel bestanden er maximaal tegelijk in een programma geopend kunnen worden.

SCHRIJFWIJZE

MAXFILES=uitdrukking

Uitdrukking **Invullen** Numerieke konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van 0 tot 15.

TOEPASSING

Geeft aan hoeveel bestanden er maximaal tegelijk in een programma geopend kunnen worden. Het tegelijk openen van bestanden betekent hier het geopend houden van een bestand terwijl een volgend bestand geopend wordt.

Voorbeeld gebruik

```
10 MAXFILES=3
20 OPEN "GRP:" FOR OUTPUT AS #1
30 OPEN "CRT:" FOR OUTPUT AS #2
40 OPEN "LPT:" FOR OUTPUT AS #3
.
.
.
1000 CLOSE
```

Aangezien in regel 10 het aantal bestanden dat tegelijk geopend kan worden is ingesteld op 3, kunnen in regel 20 en de volgende regels maximaal 3 bestanden tegelijk geopend worden.

Wanneer het aantal te openen bestanden niet met een MAXFILES= bevel is ingesteld, is het niet mogelijk meer dan een bestand tegelijk te openen.

● Bij het instellen van een onnodig groot aantal wordt het gebruikersgebied dienovereenkomstig veel kleiner.

SPRITE\$ (sprite dollar)

Voor het vastleggen van de gegevens van een sprite patroon.

SCHRIJFWIJZE

SPRITE\$ (spritenummer)

Spritenummer **Invullen** Voor 8×8 stippen—Gehele getallen van 0 tot 255.
Voor 16×16 stippen—Gehele getallen van 0 tot 63.

TOEPASSING

Wanneer een bepaald sprite patroon eenmaal met een SPRITE\$ variabele onder een gekozen spritenummer is vastgelegd, blijft dit zelfde patroon onder dat spritenummer bestaan. Zie "2-4 Grafische voorstellingen en afbeeldingen" voor nadere bijzonderheden betreffende het definiëren van sprites.

TIME (time)

Maakt de waarde van een ingebouwde schakelklok tot een variabele.

SCHRIJFWIJZE

TIME

TIME = uitdrukking

Uitdrukking **Invullen** Konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van 0 tot 65535.

TOEPASSING

Met deze variabele wordt de waarde van de ingebouwde schakelklok, die bij het inschakelen van BASIC begint te lopen, op een gekozen waarde ingesteld. Daarna loopt de schakelklok op in stappen van 1 ongeveer elke 1/50 seconde. Na het bereiken van de hoogste waarde, 65535, begint de schakelklok weer bij vanaf 0.

De waarde van de variabele kan worden gewijzigd met een LET bevel. Zolang de CPU verwerkingseenheid zich in een stand bevindt waarin onderbreken niet mogelijk is (zoals tijdens in- of uitvoer van of naar cassette) staat de schakelklok stil. Ook wanneer de stroom is uitgeschakeld loopt de schakelklok niet.

Voorbeeld gebruik

```
10 CLS:TIME=0
20 LOCATE 12,8:PRINT INT(TIME/50)
30 GOTO 20
```

Dit programma geeft onafgebroken de gehele getalswaarde weer van het quotiënt van de waarde van TIME en de deler 50, nadat de waarde van TIME eerst eenmaal op 0 is gezet. Het aangegeven getal zal ongeveer eens per seconde met 1 verhoogd worden.

VDP (video display processor)

Voor het lezen en schrijven van de inhoud van het video register.

SCHRIJFWIJZE

VDP (registernummer)

VDP (registernummer)= uitdrukking

Registernummer **Invullen** Gehele getallen van 0 tot 8.

Uitdrukking **Invullen** Konstanten, variabelen, lijstvariabelen en uitdrukkingen hiermee, van 0 tot 255.

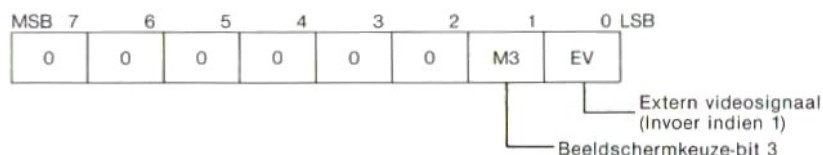
TOEPASSING

Kan gebruikt worden als functie om de registerinhoud van de TMS9918A videostuureenheid, het LSI circuit van de MSX persoonlijke computer dat de weergave op het beeldscherm regelt, te lezen. Ook te gebruiken als variabele om gegevens direct in het videoregister te schrijven.

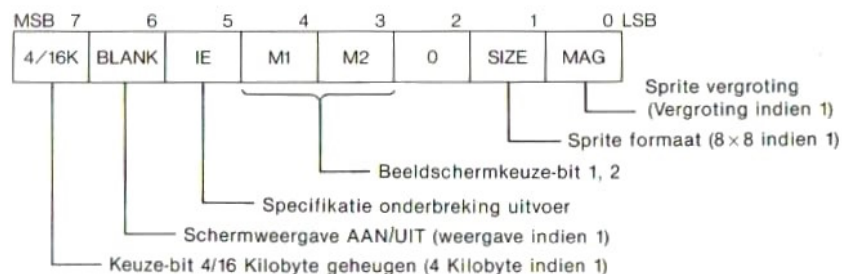
VDP registers

De onderstaande afbeeldingen tonen de bit-toewijzing van de videoregister.

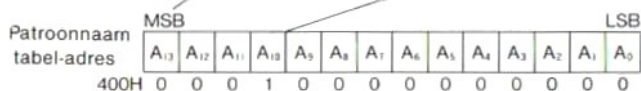
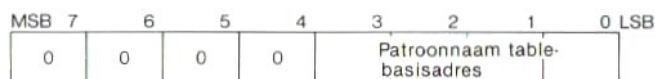
Register 0



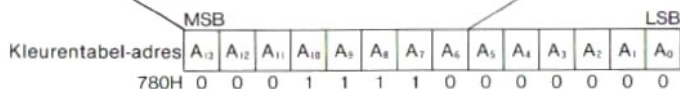
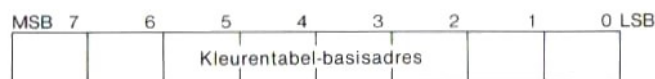
Register 1



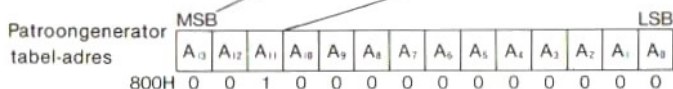
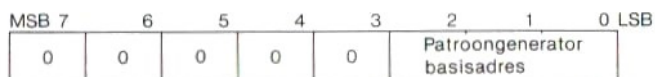
Register 2



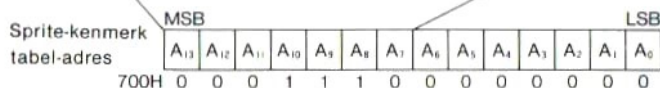
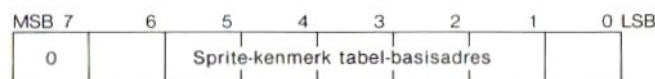
Register 3



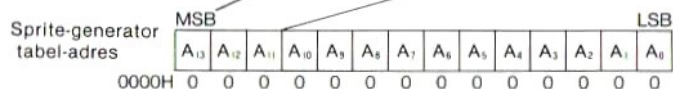
Register 4



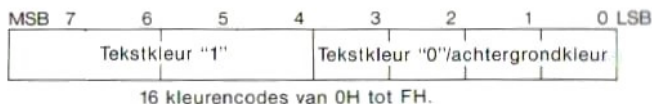
Register 5



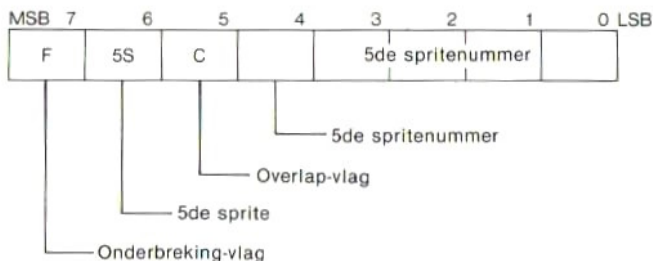
Register 6



Register 7



Register 8



Register 8 is een register speciaal bestemd voor uitlezen terwijl de andere registers speciaal bestemd zijn voor inschrijven.

Opgelet

Voor het bepalen van de weergave op het beeldscherm met een VDP variabele en door het herschrijven van de registerwaarden van de video-stuureenheid is een grondige kennis van het TMS9918A register onmisbaar. Als het videoregister slordig of met vergissingen herschreven wordt kan er van normale beeldschermweergave geen sprake zijn. U dient dus wel zeker van uw zaak te zijn voor u het register gaat wijzigen.

1-3 FOUTMELDINGEN

Wanneer er in een programma een fout optreedt wordt de verwerking onderbroken, er wordt gewacht op nadere bevelen en er wordt een foutmelding weergegeven. De oorzaak van een fout wordt in de foutmelding kort omschreven. Hieronder vindt u een uitleg van de foutmeldingen met een aantal praktijkvoorbeelden. De nummers tussen haakjes zijn foutnummers.

Bad file name (56)

- De ingevoerde bestandsnaam is onjuist.
- De ingevoerde apparaatnaam is niet bruikbaar met een OPEN, SAVE of LOAD bevel.

Bad file number (52)

- Het ingevoerde bestandsnummer ligt buiten het bereik dat is aangegeven in het MAXFILES= bevel.
- In het PRINT # bevel is een bestandnummer gebruikt van een bestand dat nog niet geopend is.

Can't CONTINUE (17)

- Na een onderbreking kan de verwerking van het programma niet hervat worden, aangezien er een wijziging is aangebracht.
- Het gevraagde programma bestaat niet.
- In het programma was al een CONT bevel gebruikt.

Device I/O error (19)

- Laden is niet mogelijk, er is iets mis met de band of de recorder.
- De niveau-instelling van de recorder is onjuist.
- Er heeft een onderbreking plaatsgevonden voor het laden was afgelopen.
- De omschreven I/O poort is niet de juiste.

Direct statement in file (57)

- Een van de bevelen in het ASCII programma dat geladen wordt is een rechtsteeks bevel, zonder regelnummer.
- Er wordt getracht een bestand te laden dat geen BASIC programma is (maar bijvoorbeeld een bestand met alleen gegevens).

Division by zero (11)

- Er wordt getracht door 0 te delen.
- Er wordt getracht te delen door een variabele die niet gedefinieerd is.

File already open (54)

- Er wordt getracht een bestand te openen dat al geopend is.

File not OPEN (59)

● In het uit te voeren bevel, PRINT #, INPUT # e.d., is een bestandnummer gebruikt van een bestand dat nog niet geopend is met een OPEN bevel.

Illegal direct (12)

● Er wordt getracht een bevel dat alleen in een programma gebruikt kan worden, zoals het DEFFN bevel, rechtstreeks uit te voeren.

Illegal function call (5)

- In het gegeven bevel wordt een onjuiste waarde gebruikt.
- De waarde van een functie ligt buiten het beschikbare bereik.

Input past end (55)

- Alhoewel alle gegevens reeds gelezen zijn, wordt getracht ze opnieuw te lezen.
- Het bestand bevat in 't geheel geen gegevens.

Internal error (51)

● Het BASIC vertolkingsprogramma werkt niet naar behoren.

Line buffer overflow (25)

● De lijnbuffer voor invoer is vol.

Missing operand (24)

- De nodige parameter van een bevel ontbreekt.
- De nodige parameters van een bevel zijn niet compleet.

NEXT without FOR (1)

- Er is een NEXT bevel zonder bijbehorend FOR bevel in een programma aanwezig.
- Bij de verwerking wordt door een GOTO bevel verwezen naar een regel die zich binnen een FOR—NEXT lus bevindt.

NO RESUME (21)

● De gebruikte foutenherstelroutine bevat geen RESUME bevel. (Een foutenherstelroutine dient te eindigen met een END, RESUME of ON ERROR GOTO 0 bevel.)

Out of DATA (4)

● Tijdens het uitvoeren van een READ bevel blijkt dat er niet voldoende of zelfs geen gegevens aanwezig zijn.

Out of memory (7)

- Het programma is te lang.
- Er zijn te veel variabelen gebruikt.
- Een lijstvariabele is te groot.
- De totaalstructuur van de FOR—NEXT of GOSUB—RETURN bevelen is te lang.

Out of string space (14)

- Het voor lettertekens beschikbare gebied is overschreden.
- Het met een CLEAR bevel voor lettertekens beschikbaar gemaakte gebied is onvoldoende.

Overflow (6)

- Numerieke gegevens of het resultaat van een berekening valt buiten het te hanteren bereik.
- De parameter van een adres ligt buiten het beschikbare bereik.

RESUME without error (22)

- Een RESUME bevel heeft geen bijbehorend ON ERROR bevel.
- Er wordt door een GOTO bevel verwezen naar een foutenherstelroutine.
- Aangezien het hoofdprogramma niet voorzien is van een END bevel wordt automatisch doorgegaan met een daarna volgende foutenherstelroutine.

RETURN without GOSUB (3)

- Een RETURN bevel heeft geen bijbehorend GOSUB bevel.
- Er wordt door een GOTO bevel verwezen naar een subroutine.
- Aangezien het hoofdprogramma niet voorzien is van een END bevel wordt automatisch doorgegaan met een daarna volgende subroutine.

Redimensioned array (10)

- Er wordt getracht overlappende lijstvariabelen te gebruiken met de zelfde naam.
- Er zijn ongedefinieerde lijstvariabelen gebruikt, die daarna pas met een DIM bevel gedefinieerd werden.

String formula too complex (16)

- Een rij lettertekens op een regel is te ingewikkeld.

String too long (15)

- Er is aan een rij-variabele een waarde toegekend die langer is dan 255 lettertekens.

Subscript out of range (9) _____

- Er wordt in een lijstvariabele een onder-indexnummer gebruikt dat het formaat zoals bepaald met een DIM bevel te boven gaat.
- Een onder-index groter dan 11 wordt gebruikt in een lijstvariabele die niet gedefinieerd is met een DIM bevel.

Syntax error (2) _____

- Een gegeven bevel voldoet niet aan de grammatikale regels van de BASIC taal.

Type mismatch (13) _____

- In een LET, INPUT of READ bevel worden links en rechts van het "—" teken ongelijksoortige eenheden (numerieke en rij-waarden) gebruikt.
- Er wordt getracht een rekenkundige of logische bewerking uit te voeren met een rij lettertekens.
- De gegevens die in een functie worden gebruikt zijn niet van de juiste soort.

Undefined line number (8) _____

- In een GOTO, GOSUB of RESUME bevel wordt verwezen naar een niet bestaand regelnummer.
- Bij het uitvoeren van een RENUM bevel blijkt dat in een GOTO of soortgelijk bevel een niet bestaand regelnummer is gebruikt.

Undefined user function (18) _____

- Er wordt getracht een gebruikersfunctie die niet gedefinieerd is met een DEFFN bevel te gebruiken.

Unprintable error (23, 26-49, 60-255) _____

- Er is een fout opgetreden waarvoor geen foutnummer bestaat.
- Er is een fout opgetreden die omschreven is in een ERROR bevel, waarbij gebruik gemaakt is van een van de hierboven gegeven "Unprintable error" foutnummers.

Verify error (20) _____

- Het programma op de cassette is niet gelijk aan het programma in het geheugen.

HOOFDSTUK 2

GEBRUIK VAN MSX-BASIC

2-1 GEBRUIKSKLAARSTAND

Na het inschakelen van MSX-BASIC verschijnt "Ok" op het scherm met hieronder de cursor. De stand waarin de computer dan staat, met de cursor en de letters "Ok" aan de linkerzijde van het scherm, is de **klaar-voor-bevelen** stand. Niet alleen in de beginstand, maar ook wanneer een programma wordt onderbroken door een STOP bevel of door gelijktijdig indrukken van de **CTRL** en **STOP** toetsen, en wanneer een programma geheel verwerkt is of een rechtstreeks gegeven bevel uitgevoerd, neemt de computer deze stand aan. De computer doet dan verder niets, maar wacht slechts tot u een volgend bevel invoert.

RECHTSTREEKS GEGEVEN BEVELEN

Wanneer de computer klaar-voor-bevelen staat kunt u een willekeurig bevel rechtstreeks invoeren. Dit bevel wordt dan door MSX-BASIC onmiddellijk uitgevoerd, waarna de computer weer op verdere bevelen blijft wachten.

INDIREKT GEGEVEN BEVELEN

Een indirect gegeven bevel is een regel tekst die bestaat uit een of meerdere bevelen, voorafgegaan door een regelnummer. Wanneer een bevel op deze manier gegeven wordt voert MSX-BASIC het niet onmiddellijk uit maar slaat het bevel samen met het regelnummer op en wacht dan weer op een volgend bevel. De in het geheugen opgeslagen bevelen met regelnummers worden daarna door het geven van een RUN of GOTO bevel op volgorde van de regelnummers uitgevoerd, en wanneer het gehele programma is verwerkt komt de computer opnieuw in de klaar-voor-bevelen stand te staan.




2-2 TEKSTOPMAAKPROGRAMMA

Met het tekstopmaakprogramma kunt u **de ingevoerde lettertekens wijzigen, van plaats doen veranderen en er nieuwe aan toevoegen, door de cursor naar elke gewenste positie op het scherm te verplaatsen.** Deze functie biedt u de mogelijkheid een programma op eenvoudige wijze te corrigeren en wijzigingen naar wens aan te brengen.

Bij gebruik van het tekstopmaakprogramma kunnen alle lettertekens op het scherm gewijzigd worden op de zelfde wijze als dit gebeurt bij het rechtstreeks geven van bevelen, waarbij de regel waarin de cursor zich bevindt door het indrukken van de **RETURN** toets in gewijzigde vorm in de computer wordt ingevoerd. Indien het eerste lettertekens van een regel een cijfer is wordt de regel als onderdeel van een programma in het geheugen opgenomen, en als de regel niet begint met een cijfer zullen de bevelen ervan onmiddellijk uitgevoerd worden.





In dit geval wordt de regel beschouwd als afgebroken door de **RETURN** toets. Dit is voor een regel op het scherm niet nodig maar het kan wel nodig zijn voor verscheidene regels. (De maximale lengte per regel in MSX-BASIC is 255 lettertekens.)

GEbruik OPMAAKTOETSEN

| Toets | Functie |
|--|---|
| Cursortoetsen (   ) | Verplaatsen de cursor een plaats omhoog, omlaag, naar links en naar rechts. |
| HOME | Verplaatst de cursor naar de uiterste linksbovenhoek van het scherm. |
| SHIFT + HOME | Wist alle lettertekens van het scherm en verplaatst de cursor naar de uiterste linksbovenhoek van het scherm. |
| INS | Maakt het invoegen van lettertekens mogelijk (de cursor wordt gehalveerd om dit aan te geven). De lettertekens die na het indrukken van de INS toets worden ingevoerd via het toetsenbord verschijnen op de plaats van de cursor tussen de bestaande tekst. De normale wijze van invoeren wordt hervat door het indrukken van de RETURN toets of een van de cursortoetsen, of door nogmaals indrukken van de INS toets. |
| DEL | Wist het letterteken dat zich op de plaats van de cursor bevindt, en verplaatst alle volgende lettertekens van de regel een plaats naar links. |
| BS | Wist het letterteken dat zich direct links van de cursor bevindt, en verplaatst alle volgende lettertekens van de regel een plaats naar links. |
| TAB | Verplaatst de cursor naar de volgende tabulatorpositie en voegt spaties in als er voor deze positie lettertekens stonden. |

CTRL TOETS FUNKTIES

Behalve de speciale opmaaktoetsen biedt MSX-BASIC ook een reeks speciale functies die eenvoudig te gebruiken zijn door het gelijktijdig indrukken van de **CTRL** toets en een van de lettertoetsen.

| Toets | Functie |
|------------------------|--|
| CTRL + B | Verplaatst de cursor naar het begin van een woord (of een groep lettertekens gescheiden van een voorgaande door een spatie). Als de cursor zich al aan het begin van een woord bevond, komt hij aan het begin van het vorige woord te staan. |
| CTRL + C | Schakelt de invoerstand of het automatisch genereren van regelnummers door het AUTO bevel uit, en zet de computer weer in de klaar-voor-bevelen stand. |
| CTRL + E | Wist alle lettertekens van de plaats waar de cursor zich bevindt tot en met de laatste regel van het programma. |
| CTRL + F | Verplaatst de cursor naar het begin van het volgende woord. |
| CTRL + G | Doet een pieptoon klinken. |
| CTRL + H | Gelijk aan de BS toets. |
| CTRL + I | Gelijk aan de TAB toets. |
| CTRL + J | Verplaatst de cursor 1 regel omlaag. |
| CTRL + K | Gelijk aan de HOME toets. |
| CTRL + L | Gelijk aan de SHIFT en HOME toets samen. |
| CTRL + M | Gelijk aan de RETURN toets. |
| CTRL + N | Verplaatst de cursor naar de positie volgend op het laatste letterteken van een regel. |
| CTRL + R | Gelijk aan de INS toets. |
| CTRL + U | Wist alle lettertekens van een regel. |
| CTRL + X | Gelijk aan de SELECT toets. Niet gedefinieerd in MSX-BASIC. |
| CTRL + \ | Gelijk aan de  cursortoets. |
| CTRL + | Gelijk aan de ESC toets. Niet gedefinieerd in MSX-BASIC. |
| CTRL +] | Gelijk aan de  cursortoets. |
| CTRL + ^ | Gelijk aan de  cursortoets. |
| CTRL + - | Gelijk aan de  cursortoets. |

2-3 SPECIALE TOETSEN

FUNKTIETOETSEN

De rij toetsen [F1] tot en met [F5] worden de funktietoetsen genoemd. Voor elke funktietoets kan een rij lettertekens vastgelegd worden, waarna het indrukken van de funktietoets hetzelfde resultaat heeft als het in zijn geheel invoeren van de rij lettertekens. In MSX-BASIC zijn voor de funktietoetsen de volgende rijen lettertekens vastgelegd, die te gebruiken zijn vanaf het moment dat MSX-BASIC ingeschakeld is.

| | | | |
|---------|---|-------|-------------------------------|
| [F1] | | color | _ |
| [F2] | | auto | _ |
| [F3] | | goto | _ |
| [F4] | | list | _ |
| [F5] | | run | [RETURN] |
| [SHIFT] | + | [F1] | color 15, 4, 4 [RETURN] |
| [SHIFT] | + | [F2] | load" |
| [SHIFT] | + | [F3] | cont [RETURN] |
| [SHIFT] | + | [F4] | list [RETURN] |
| [SHIFT] | + | [F5] | [SHIFT] + [HOME] run [RETURN] |

De voor de funktietoetsen vastgelegde rijen lettertekens kunnen naar wens gewijzigd worden met een KEY bevel.

OVERIGE SPECIALE TOETSEN

[STOP] toets

Voor het tijdelijk onderbreken van de verwerking van een BASIC programma. De verwerking kan hervat worden door het nogmaals indrukken van de [STOP] toets.

[CTRL] en [STOP] tegelijk ingedrukt

Voor het onderbreken van de verwerking van een BASIC programma. Als tot op dat moment de regelnummers automatisch gegenereerd werden door een AUTO bevel wordt bij onderbreking de klaar-voor-bevelen stand aangenomen.

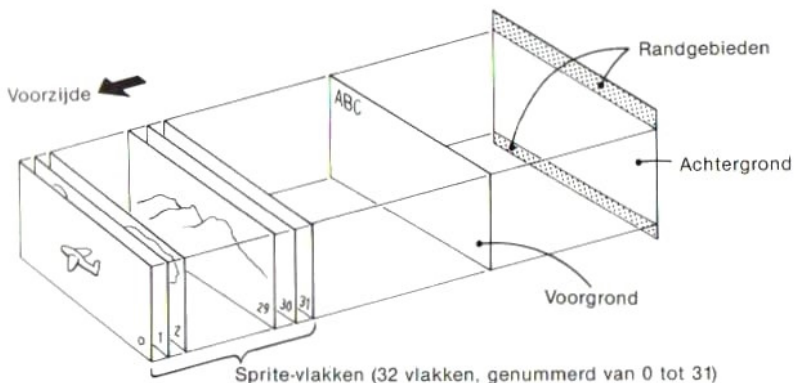
[SELECT] toets en [ESC] toets

De functie van deze toetsen is in MSX-BASIC niet gedefinieerd, ze hebben dus nog geen functie.

2-4 GRAFISCHE VOORSTELLINGEN EN AFBEELDINGEN

BEELDSCHERM-KONFIGURATIE

Onderstaande afbeelding toont de opbouw van de beeldscherm-konfiguratie voor MSX-BASIC.



Tekstschermben grafisch scherm

Het tekstschermben dient voor weergeven van lettertekens (zgn. alfanumerieke tekens) en het grafisch scherm voor het afbeelden van grafische voorstellingen (stippen, lijnen, cirkels, enz.). MSX-BASIC biedt twee verschillende tekstschermben en twee grafische scherm ben die gekozen kunnen worden met een SCREEN bevel, als volgt:

SCREEN soort scherm

De verschillende schermen die gekozen kunnen worden met het SCREEN bevel hebben de volgende eigenschappen.

| SCREEN bevel | Soort scherm | Sprite-vlakken | Eigenschappen |
|---|--------------|----------------|--|
| SCREEN 0 (Dit scherm verschijnt bij inschakelen van BASIC) | Tekst | Niet bruikbaar | De lettertekens zijn 6 stippen breed. Een aantal grafische tekens en KANA zijn 8 stippen breed en dus niet in hun geheel weer te geven. |
| SCREEN 1 | | Bruikbaar | Voor de lettertekens zijn 8 stippen beschikbaar. De meeste hebben echter maar 6 stippen nodig, wat het scherm beter leesbaar maakt dan SCREEN 0. |
| SCREEN 2 | Grafisch | Bruikbaar | Grafische voorstellingen worden met 1 stip tegelijk afgebeeld. |
| SCREEN 3 | | Bruikbaar | Grafische voorstellingen worden in blokken van 4×4 stippen tegelijk afgebeeld. |

De voorgrond, achtergrond en randgebieden zijn voor alle schermen gelijk. Wanneer lettertekens of grafische voorstellingen op de voorgrond zijn afgebeeld, kan alleen van de achtergrond en de randgebieden de kleur worden gewijzigd.

Voor alle schermen behalve dat van SCREEN 0 kunnen naast de bovenstaande gebieden ook een aantal sprite-vlakken worden gebruikt. Op een sprite-vlak kan een dynamisch beeld worden gevormd door een willekeurig te definiëren patroon van sprites, zoals uitgelegd zal worden in "Gebruik van sprites".

Kleurspecificatie

Voor het bepalen van de kleur van de voorgrond, achtergrond en randgebieden wordt het **COLOR** bevel gebruikt.

COLOR kleur voorgrond, kleur achtergrond, kleur randgebieden

Zowel lettertekens als grafische voorstellingen nemen de kleur van de voorgrond aan, tenzij uitdrukkelijk anders wordt gespecificeerd.

Voor het SCREEN 0 scherm is de kleur van de randgebieden altijd gelijk aan dat van de achtergrond.

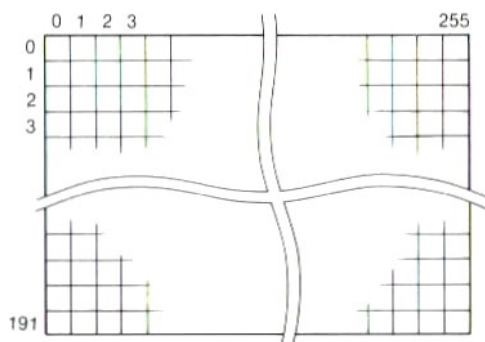
Zie blz. 206 voor de tabel met de kleurcodes.

SCREEN 2—GEDETAILLEERD GRAFISCH SCHERM

Op het grafisch scherm zijn met de volgende bevelen grafische voorstellingen te tekenen.

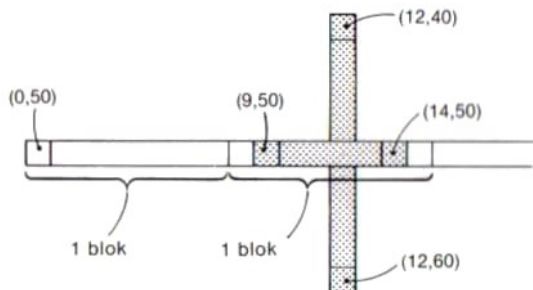
- PSET, PRESET ... Zet een stip of wist er een uit.
- LINE ... Trekt een rechte lijn of een vierkant.
- CIRCLE ... Trekt een cirkel.
- PAINT ... Voor kleuren.
- DRAW ... Tekent willekeurige grafische voorstellingen.

Bij het gebruik van deze bevelen moeten de coördinaten gegeven worden die de plaats op het scherm bepalen.



Bij het gedetailleerd grafisch scherm kan van elke stip de plaats en de kleur bepaald worden, voor in totaal van links naar rechts 256 stippen en van boven naar beneden 192 stippen, zoals aangegeven in bovenstaande afbeelding. Voor elk afzonderlijk blok van 8 stippen horizontaal kan echter slechts een kleur worden bepaald, en de laatst toegekende kleur wordt aangehouden.

```
10 SCREEN 2
20 LINE (9,50)-(14,50),15
30 LINE (12,40)-(12,60),1
40 GOTO 40
```



In het bovenstaande programma zien we een blok met coördinaten van 8 tot 15. De rechte lijn in regel 20, waarvan de kleur was bepaald op wit, wordt uiteindelijk zwart afgebeeld omdat de zwarte lijn van de volgende regel de eerste overlapt.

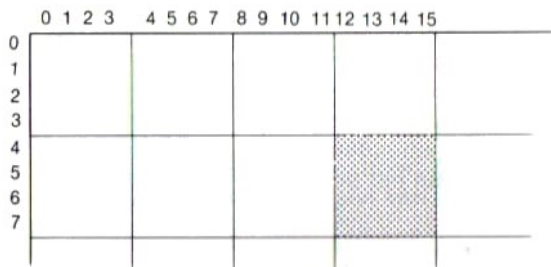
De gekozen witte kleur blijft wel gelden als het LINE bevel van regel 20 als volgt gewijzigd wordt:

```
LINE (8,50)-(15,50)
```

Hiermee wordt binnen het blok van 8 stippen een horizontale lijn van maximale lengte getrokken.

SCREEN 3—MEERKLEUREN GRAFISCH SCHERM

Ook het scherm van SCREEN 3 is geschikt voor het afbeelden van grafische voorstellingen met bevelen als PSET en LINE. De plaats wordt ook bepaald met de horizontale coördinaten van 0-255 en de verticale coördinaten van 0-191. De kleinste eenheid voor grafische voorstellingen is een blok van 4×4 stippen.



```
PSET (12,4),1
```

```
PSET (14,5),1
```

```
PSET (15,7),1
```

Alhoewel deze regels elk slechts betrekking hebben op een van de stippen in hetzelfde blok, is elk van de regels afzonderlijk voldoende om het gehele \otimes deel van bovenstaande afbeelding zwart te kleuren.

```
LINE (17,5)-(130,110)
```

Deze regel trekt een ruwe lijn die de blokken verbindt waarin de stippen (17,5) en (130,110) zich bevinden, met andere woorden trekt de verbindinglijn tussen

$\begin{matrix} 16 & 19 \\ 4 & \square \\ 7 & \end{matrix}$ en $\begin{matrix} 128 & 131 \\ 108 & \square \\ 111 & \end{matrix}$.

STEP—VOOR NIEUWE COÖRDINATEN

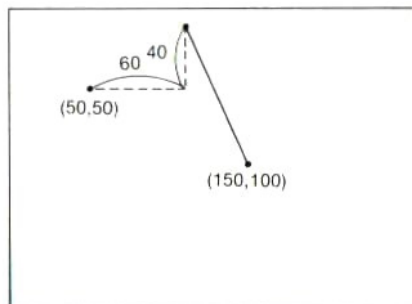
U heeft de mogelijkheid door het invoegen van een **STEP (X,Y)** specificatie een nieuw coördinatenstelsel voor de diverse grafische bevelen **CIRCLE**, **LINE**, **PAINT**, **PSET**, **PRESET** en **SPRITE** te bepalen.

Na het uitvoeren van een van deze grafische bevelen onthoudt de computer in **MSX-BASIC** altijd het laatst genoemde punt. Wanneer hierna een **STEP (X,Y)** specificatie wordt ingevoegd in een volgend grafisch bevel, wordt de plaats van punt (X,Y) bepaald aan de hand van een nieuw coördinatenstelsel met het laatst genoemde punt als punt van oorsprong (0,0). Bij het weglaten van de **STEP** specificatie kan een punt ook nog altijd bepaald worden op basis van het gewone coördinatenstelsel met de uiterste linkerbovenhoek van het scherm als punt van oorsprong.

Voorbeeld 1

```
10 SCREEN 2
20 PSET (50,50)
30 LINE STEP (60,-40)-(150,100)
40 GOTO 40
```

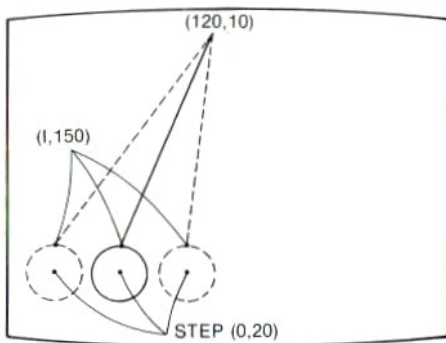
In dit programma worden de coördinaten (50, 50) van het **PSET** bevel in het geheugen van de computer vastgehouden wanneer deze verdergaat naar regel 30. Aangezien dan binnen het **LINE** bevel de **STEP (60, -40)** specificatie wordt gebruikt als beginpunt van de te trekken lijn, wordt als beginpunt het punt aangehouden dat horizontaal een afstand 60 en verticaal een afstand -40 van het punt (50, 50) als de nieuwe oorsprong verwijderd is.



Voorbeeld 2

```
10 SCREEN 2
20 FOR I=30 TO 240 STEP 20
30 LINE (120,10)-(I,150)
40 CIRCLE STEP (0,20),20
50 CLS
60 NEXT I
```

In dit programma worden de coördinaten van het eindpunt van de lijn in regel 30 telkens veranderd door een FOR—NEXT lus, waarna het middelpunt van de cirkel in regel wordt bepaald door de STEP (0, 20) specificatie in het CIRCLE bevel, op een vaste afstand van 20 van het eind van de rechte lijn.

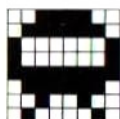


GEbruik VAN SPRITES

In MSX-BASIC kan een patroon (sprite-patroon genaamd) met willekeurige afmetingen worden afgebeeld op een van de 32 sprite-vlakken en naar wens worden verplaatst.

Sprite-patroon

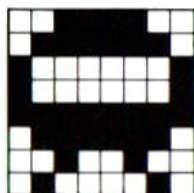
Een sprite-patroon bestaat uit 8×8 of 16×16 stippen met een naar keuze instelbaar formaat (vergroot of niet vergroot). Met vergroting is een sprite-patroon zowel horizontaal als vertikaal tweemaal zo groot als het patroon zonder vergroting.



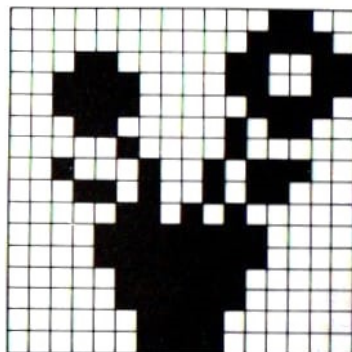
8×8 stippen zonder vergroting



16×16 stippen zonder vergroting



8×8 stippen met vergroting



16×16 stippen met vergroting

Het formaat van een sprite-patroon wordt bepaald in het SCREEN bevel. De tweede parameter van het SCREEN bevel dient voor het kiezen van het formaat van de sprites.

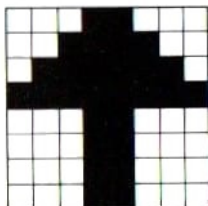
| Parameter | Sprite-formaat |
|-----------|--|
| 0 | 8×8 stippen zonder vergroting |
| 1 | 8×8 stippen met vergroting |
| 2 | 16×16 stippen zonder vergroting |
| 3 | 16×16 stippen met vergroting |

SCREEN 2,2

Dit bevel houdt in dat op het gedetailleerde grafische scherm een 16×16 sprite zonder vergroting wordt gebruikt. Na het aldus met een SCREEN bevel instellen van het formaat van de sprites blijft dit op alle spritevlakken konstant.

Definiëren van het sprite-patroon

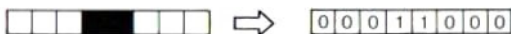
Bij het definiëren van een sprite-patroon van 8×8 stippen wordt dit eerst gescheiden in 8 horizontale banen. Een pijlvormig patroon wordt bijvoorbeeld gedefinieerd als in onderstaande afbeelding.



Een patroon dat zo in horizontale banen verdeeld is, kan weergegeven worden als kleinere patronen van 8 stippen elk.



Vervolgens wordt het patroon van elk baan weergegeven met een 1 in plaats van een stip en een 0 op elke plaats zonder stip. We verkrijgen dan een binair of tweetallig getal. De eerste regel wordt bijvoorbeeld 00011000 en de volgende 0011100.



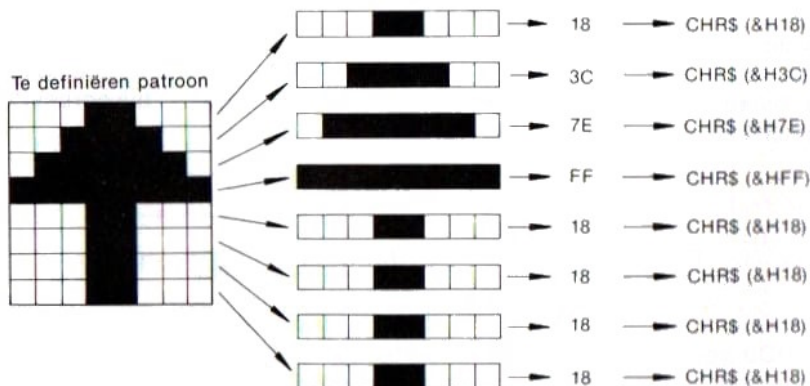
Deze binaire getallen kunnen dan omgezet worden in hexadecimale (of decimale) getallen. Voor de bovenste regel wordt dit: 00011000 (binair)=18 (hexadecimaal) of 24 (decimaal). Voor de tweede regel:

00111100 (binair)=3C (hexadecimaal) of 60 (decimaal). Voor de gebruiker die niet geheel vertrouwd is met het omzetten van binaire getallen in hexadecimale getallen is het eenvoudiger om het patroon van 8 stippen eerst te verdelen in 4 stippen links en 4 stippen rechts, om daarna deze groepen van 4 aan de hand van onderstaande tabel om te zetten in een hexadecimaal getal elk.

| Patroon | Hexadecimaal | Patroon | Hexadecimaal |
|---------|--------------|---------|--------------|
| | 0 | | 8 |
| | 1 | | 9 |
| | 2 | | A |
| | 3 | | B |
| | 4 | | C |
| | 5 | | D |
| | 6 | | E |
| | 7 | | F |

In het geval van het patroon zijn de linker 4 stippen dus en de rechter 4 stippen . Omgezet volgens bovenstaande tabel wordt dit samen 18.

Het letterteken waarvan het hexadecimale (of decimale) getal de ASCII code vormt kunt u vinden met behulp van de CHR\$ functie. Het definiëren van het eerder genoemde pijlvormige patroon gaat dan als volgt in zijn werk:



Bij een 8×8 sprite-patroon worden de gegevens die op de hierboven beschreven wijze verkregen zijn in volgorde van boven naar beneden wijze verkregen zijn in volgorde van boven naar beneden opgeteld en toegekend aan de **SPRITES** variabele als een rij lettertekens die het sprite-patroon definieert. Het pijlvormige patroon van het voorbeeld wordt dus uiteindelijk als volgt gedefinieerd.

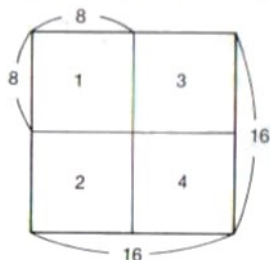
```
SPRITES(1)=CHR$(&H18)+CHR$(&H3C)+CHR$(&H7E)
+CHR$(&HFF)+CHR$(&H18)+CHR$(&H18)+CHR$(&H18)
+CHR$(&H18)
```

Het nummer waaraan dit gedefinieerde sprite-patroon herkenbaar is is 1, aangegeven tussen de haakjes van SPRITES(1).

Als voor de combinatie van de CHR\$ functie en de code een letterteken bestaat, dan kan dit ook direkt ingevoerd worden. In het voorbeeld komt CHR\$(&H3C) voor, wat staat voor "<", en CHR\$(&H7E), wat "~" betekent, zodat u ook het volgende kunt invoeren.

```
SPRITES(1)=CHR$(&H18)+"<"+"~"+CHR$(&HFF)+
CHR$(&H18)+CHR$(&H18)+CHR$(&H18)+CHR$(&H18)
```

Een sprite-patroon van 16×16 stippen kan op dezelfde wijze worden gedefinieerd. Een patroon van 16×16 stippen wordt echter beschouwd als opgebouwd uit vier sprite-patronen van 8×8 stippen, en deze vier blokken worden gedefinieerd in de hieronder aangegeven volgorde.



```
A$=CHR$(&H0)+CHR$(&H0)+CHR$(&H18)+CHR$(&H3C)
+CHR$(&H3C)+CHR$(&H18)+CHR$(&H04)+CHR$(&H22)
B$=CHR$(&H1A)+CHR$(&H06)+CHR$(&H0F)+
CHR$(&H0F)+CHR$(&H07)+CHR$(&H07)+CHR$(&H03)
+CHR$(&H03)
C$=CHR$(&H0C)+CHR$(&H1E)+CHR$(&H33)+
CHR$(&H33)+CHR$(&H1E)+CHR$(&H2C)+CHR$(&H20)
+CHR$(&H5C)
D$=CHR$(&H5B)+CHR$(&HA0)+CHR$(&HF0)+
CHR$(&HF0)+CHR$(&HE0)+CHR$(&HE0)+CHR$(&HC0)
+CHR$(&HC0)
SPRITES(2)=A$+B$+C$+D$
```

Aantal te definiëren sprite-patronen

De nummers van de 8×8 sprite-patronen lopen van 0 tot 255, en die van 16×16 patronen lopen van 0 tot 63. Met andere woorden, u kunt ten hoogste 256 sprite-patronen van 8×8 stippen definiëren, en ten hoogste 64 patronen van 16×16 stippen. (Overigens zijn deze aantallen soms niet haalbaar vanwege een tekort aan geheugenkapaciteit.)

Weergeven van een sprite-patroon

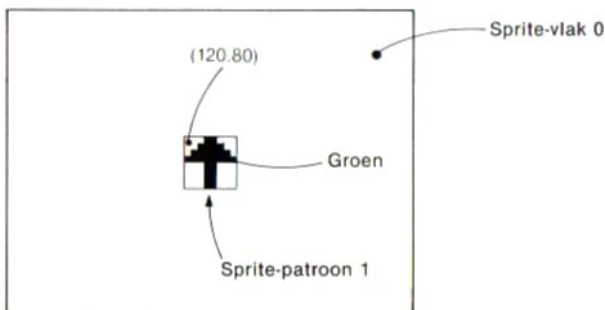
Na het definiëren kan een sprite-patroon weergegeven worden op een sprite-vlak met het **PUT SPRITE** bevel.

**PUT SPRITE nummer sprite-vlak, (X-coördinaat, Y-coördinaat),
kleurcode, nummer sprite-patroon**

Voor het in groen (kleurcode 2) weergeven van een aldus gedefinieerd sprite-patroon op positie (120, 80) van sprite-vlak 0 luidt de betreffende regel van het programma als volgt:

```
PUT SPRITE 0, (120,80), 2, 1
```

Op de aangegeven positie komt de stip in de linkerbovenhoek van het sprite-patroon te staan. De X- en Y-coördinaten worden gespecificeerd op basis van een coördinatenstelsel met punt (0, -1) van het grafische scherm als de oorsprong (0, 0).



```
PUT SPRITE 0, (120, 80), 2, 1
```


Regels voor het weergeven van sprites

- Op elk sprite-vlak kan slechts één sprite-patroon worden weergegeven.
- Wanneer sprite-patronen op verschillende sprite-vlakken elkaar overlappen, dan blijft dat op het meer naar voren gelegen vlak (lager nummer) zichtbaar en een deel van het andere patroon verdwijnt uit het gezicht.
- Ook al zijn er vijf of meer sprite-patroon horizontaal op een rij gezet, in dat geval zullen slechts de vier patronen met hogere prioriteit (op sprite-vlakken met een lager nummer) worden weergegeven.
- Bij het weglaten van de plaats-aanduiding wordt automatisch aangenomen dat de gewenste plaats in een voorgaand grafisch bevel is gegeven.
- Bij het weglaten van de kleurcode wordt aangenomen dat de kleur van de voorgrond is gekozen.
- Bij het weglaten van het nummer van het sprite-patroon wordt aangenomen dat het juiste nummer gelijk is aan dat van het sprite-vlak.

Verplaatsen van een sprite-patroon

Voor het verplaatsen van een sprite-patroon is het voldoende om in het PUT SPRITE bevel de X- en Y-coördinaten van de positie door variabelen te vervangen en het PUT SPRITE bevel meermalen te doen uitvoeren met wisselende waarden van de variabelen. Aangezien een eerder afgebeeld sprite-patroon automatisch van het sprite-vlak wordt gewist bij het uitvoeren van een PUT SPRITE bevel is het niet nodig het sprite-patroon in een programma afzonderlijk te wissen.

Het is mogelijk een vloeiende beweging te verkrijgen door het verplaatsen van het sprite-patroon met 1 stip tegelijk.

In het volgende programma vliegt een sprite-patroon in de vorm van een UFO over het scherm en verandert hierbij van richting.

```
10 SCREEN 2
20 SPRITE$(0)=CHR$(&H3C)+CHR$(&H7E)+CHR$(&H81)+CHR$(&H81)+CHR$(&HFF)+CHR$(&H7E)+CHR$(&H24)+CHR$(&H42)
30 X=100:Y=100
40 S=INT(RND(1)*80)
50 D=INT(RND(1)*4)
60 IF D=0 THEN VX=0:VY=-1
70 IF D=1 THEN VX=1:VY=0
80 IF D=2 THEN VX=0:VY=1
90 IF D=3 THEN VX=-1:VY=0
100 FOR I=0 TO S
110 PUT SPRITE 0,(X,Y),1,0
120 X=X+VX:Y=Y+VY
130 IF X>240 OR X<0 THEN VX=-VX
140 IF Y>175 OR Y<0 THEN VY=-VY
150 NEXT I
160 GOTO 40
```

Bepaalt de afstand waarover de UFO zich verplaatst.

Bepaalt de richting van de beweging.

Beweegt de sprite.

2-5 GELUID

MSX-BASIC EN GELUID

MSX-BASIC beschikt over twee bevelen voor het spelen van muziek, te weten **PLAY** en **SOUND**. **PLAY** is een bevel waarmee u via een aantal deel-instructies het LSI circuit stuurt voor het bepalen van de toon, het ritme en de klankkleur. Het geluid dat klinkt is het resultaat van het schrijven van verschillende gegevens in het LSI register. Met het **SOUND** bevel kunnen gegevens rechtstreeks in het LSI register geschreven worden. Daarom is het nodig voor het rechtstreeks bepalen van het geluid met een **SOUND** bevel dat u de functies van het LSI register kent en weet welke gegevens er in geschreven kunnen worden.

REGELEN VAN HET VOLUMENIVEAU MET EEN PLAY BEVEL

Het gebruik van het **PLAY** bevel is al beschreven in Hoofdstuk 1 onder "PLAY"; hier volgt een uitleg van de S deel-instructie en de M deel-instructie.

PLAY "CDEFG" ————— ❶

PLAY "S13M255CDEFG" ————— ❷ (Elke _ is een deel-instructie.)

Probeer eerst regel ❶ en dan regel ❷ in BASIC, en vergelijk bij uitvoering deze twee programma's met dezelfde klankkleur. Bij het uitvoeren van

PLAY "S8M900CDEFG"

klinkt het alsof onafgebroken in hoog tempo piano wordt gespeeld.

Sn — De deel-instructie die het **patroon van volumevariaties** bepaalt.

Mn — De deel-instructie die de **cyclus van het patroon** dat is gekozen met Sn bepaalt.

De beginwaarden van Sn en Mn zijn respectievelijk S13 en M255. Door de waarde van n in Sn en Mn te wijzigen wordt een andere klankkleur verkregen.

Kombinaties van patroon en cyclus

Met de Sn deel-instructie kan gekozen worden uit 8 verschillende patronen, zoals aangegeven in de tabel op blz. 00. Hoe kleiner de waarde van n in de Mn deel-instructie, des te korter wordt de cyclus. (Met andere woorden, het aantal malen dat een patroon binnen een bepaald periode herhaald wordt is groter voor een kleinere waarde van n.)

Dit kunt u ervaren door het volgende programma uit te voeren.

PLAY "S8M300CDEFG"

Luister maar eens naar de volgende twee bevelen en vergelijk ze.

PLAY "S8M900CDEFG"

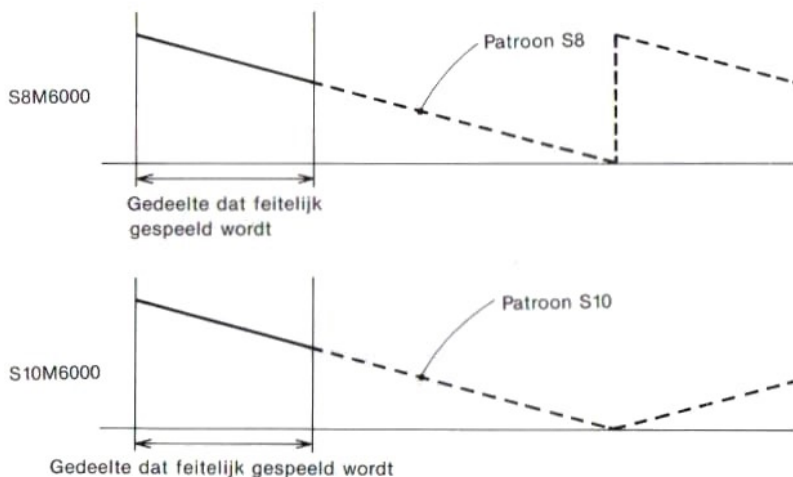
PLAY "S10M900CDEFG"

Het verschil in de patronen die u kunt bepalen met de Sn deelinstructie zal u duidelijk zijn. Met te grote waarden van n in de Mn deelinstructie wordt de cyclus te lang om het verschil nog duidelijk te kunnen horen.

PLAY "S8M6000CDEFG"

PLAY "S10M6000CDEFG"

Bij uitvoering klinken deze beide regels gelijk omdat het patroon horizontaal (in tijd) te lang werd uitgerekt, zodat bij het spelen van de toonreeks alleen de identieke gedeelten van verschillende patronen hoorbaar zijn.



Aangezien de lengte van de feitelijk gespeelde gedeelten in bovenstaande voorbeelden gewijzigd kan worden met de L deelinstructie zal het duidelijk zijn dat de kwaliteit van een muziekstuk voor een groot deel afhankelijk is van het kiezen van juiste combinatie van waarden voor de verschillende deelinstructies.

GELUID EN RUIS MET EEN SOUND BEVEL

SOUND is een bevel dat naar wens geluidseffekten of ruis voortbrengt door het schrijven van gegevens in een LSI register voor geluid, PSG genaamd (voor Programmable Sound Generator). De PSG is voorzien van 3 kanalen voor het voortbrengen van geluid (met een bepaalde frequentie). Ook zijn alle drie kanalen geschikt om als ruisgenerator te dienen. U kunt dus zowel drievoudige akkoorden als verschillende soorten ruis laten klinken. De PSG is uitgerust met 16 registers met een reeks uiteenlopende functies.

| Registrernummer | Functie |
|-----------------|--|
| 0, 1 | Bepaalt de frequentie van kanaal A. |
| 2, 3 | Bepaalt de frequentie van kanaal B. |
| 4, 5 | Bepaalt de frequentie van kanaal C. |
| 6 | Bepaalt de ruisfrequentie. |
| 7 | Kiest een kanaal voor het voortbrengen van geluid. |
| 8 | Bepaalt het volume van kanaal A. |
| 9 | Bepaalt het volume van kanaal B. |
| 10 | Bepaalt het volume van kanaal C. |
| 11, 12 | Bepaalt de frequentie van het patroon van volumevariaties. |
| 13 | Kiest het patroon van volumevariaties. |

(De registers 14 en 15 zijn niet van invloed op het voortgebrachte geluid.)

Elk register is in feite een plaats waar gegevens tijdelijk worden opgeslagen. Het geluid dat de PSG voortbrengt verandert afhankelijk van de gegevens die in de registers geschreven zijn.

Bepalen van de geluidsfrequentie

De frequenties van het geluid in de drie verschillende kanalen worden bepaald met de zes registers van 0 t/m 5. De gegevens die in een register geschreven moeten worden vindt u met de volgende formule:

$$\frac{1789772,5 \text{ (Hz)}}{16 \times (\text{uitgangsfrequentie (Hz)})} = 256 \times (\text{gegevens register 1, 3, 5}) \\ + (\text{gegevens register 0, 2, 4})$$

Wanneer u bijvoorbeeld via kanaal A een toon van 300 Hz wilt voortbrengen, wordt de uitdrukking hiervoor:

$$\frac{1789772,5}{16 \times 300} = 373 = 256 \times 1 + 117.$$

Daaruit volgt dat u in register 0 de waarde 160 moet schrijven en in register 1 de waarde 1.

Hiervoor voert u de volgende bevelen in:

```
SOUND 0,117
SOUND 1,1
```

In het geval van kanaal b worden de registers 2 en 3 gebruikt in plaats van 0 en 1, en de regels van het programma luiden dan als volgt:

```
SOUND 2,117
SOUND 3,1
```

Bepalen van de ruisfrequentie

In register 6 kunnen gegevens geschreven worden met een waarde van 0 tot 31 voor het bepalen van de ruisfrequentie. De verhouding tussen de gegevens en de frequentie kan als volgt uitgedrukt worden.

$$\text{Waarde gegevens} = \frac{1789772,5 \text{ (Hz)}}{16 \times \text{ruisfrequentie (Hz)}}$$

Wanneer bijvoorbeeld in register 6 als gegeven de waarde 15 geschreven wordt, luidt de uitdrukking

$$15 \approx \frac{1789772,5}{16 \times 7457}$$

Hieruit volgt dat de ruisfrequentie ongeveer 7457 Hz wordt.

Kanaalkeuze

Het voor weergave gebruikte kanaal wordt bepaald door de gegevens die in register 7 geschreven worden.

| Ruis | | | Geluid | | |
|----------|----|---|--------|---|---|
| Kanaal C | B | A | C | B | A |
| 32 | 16 | 8 | 4 | 2 | 1 |

Tel de getallen die u in bovenstaande tabel vindt voor de gebruikte kanalen op en trek de uitkomst af van 63. De uitkomst hiervan geeft de waarde van het gegeven dat in kanaal 7 moet worden geschreven.

Wanneer bijvoorbeeld via kanaal A en B alleen een toon, en via kanaal C zowel een toon als ruis moet worden voortgebracht, luidt de uitdrukking die de uiteindelijke waarde 24 geeft als volgt:

$$63 - (32 + 4 + 2 + 1) = 24$$

Bepalen van het volume en voortbrengen van geluid

Schrijf de gegevens voor het bepalen van het volume van kanaal A, B en C respectievelijk in register 8, 9 en 10. Waarden van 0 tot 15 kunnen gekozen worden, waarbij 15 staat voor het maximale volume.

De voorwaarden voor het voortbrengen van geluid zijn alle hierboven beschreven.

Het onderstaande programma geeft via kanaal A, B en C drie verschillende tonen weer.

| | | |
|-----------------|---|---|
| 10 SOUND 0,47 | } | Geeft de frekwentie van kanaal A als 200 Hz. |
| 20 SOUND 1,2 | | |
| 30 SOUND 2,140 | } | Geeft de frekwentie van kanaal B als 800 Hz. |
| 40 SOUND 3,0 | | |
| 50 SOUND 4,56 | } | Geeft de frekwentie van kanaal C als 2000 Hz. |
| 60 SOUND 5,0 | | |
| 70 SOUND 7,56 | } | Kiest de kanalen A, B en C voor het voortbrengen van geluid. |
| 80 SOUND 8,9 | | |
| 90 SOUND 9,10 | | |
| 100 SOUND 10,11 | } | Bepaalt het volume van elk kanaal en brengt het geluid voort. |

Wanneer het volumenniveau van de kanalen in regel 80, 90 en 100 van dit programma gewijzigd wordt, kunt u het geluid van de afzonderlijk kanalen onderscheiden.

Na het eenmaal uitvoeren van het programma blijft het geluid voortgebracht worden.

Druk voor het onderbreken van het geluid de **CTRL** en **STOP** toetsen tegelijk in.

Voeg aan het programma toe:

65 SOUND 6,31 ————— (Bepaalt de frekwentie van de ruis.)

en wijzig regel 70 als volgt:

70 SOUND 7,48 ——— (Geeft een toon en ruis via kanaal A en alleen een toon via kanaal B en C.)

Nu wordt een gemengd geluid van tonen en ruis voortgebracht:

Geluidseffekten met patronen van volumevariatiës

Met een SOUND bevel kunt u dezelfde funkties gebruiken als die van de Sn en Mn deelinstrukties van het PLAY bevel. De patronen van volumevariatiës worden bepaald door de gegevens die in register 13 worden geschreven, waarmee deze gegevens dus dezelfde funktie vervullen als de n waarde van de Sn deelinstruktie in het PLAY bevel.

Zie de tabel op blz. 93 voor de patronen die behoren bij de n waarden.

De cyclus van een patroon van volumevariatiës wordt bepaald door gegevens die in register 11 en 12 worden geschreven, volgens deze formule:

$$\frac{1789772,5 \text{ (Hz)}}{256 \times \text{cyclus(Hz)}} = 256 \times (\text{gegevens register 12}) + (\text{gegevens register 11})$$

Wanneer u bijvoorbeeld de cyclus wilt instellen op 10 Hz dient u de waarde 187 in register 11 te schrijven en de waarde 2 in register 12, aangezien de uitdrukking voor 10 Hz luidt:

$$\frac{1789772,5}{256 \times 10} = 699 = 256 \times 2 + 187$$

Geef de waarde 16 voor het volume van het kanaal waarop het hierboven beschreven patroon van toepassing is. Als de volumevariatiës bijvoorbeeld gebruikt worden voor kanaal C luidt het bevel

SOUND 10,16

Een groot aantal verschillende geluidseffekten kan worden bereikt door het patroon van volumevariatië toe te passen op de voortgebrachte ruis, en door het mengen van een zeer hoge toon met geluid van een zeer lage frequentie om zo een metalig klinkend of zoemend geluid voort te brengen. Het volgende programma geeft een geluid als van een stoomlokomotief weer door het periodiek wijzigen van het geluidsvolume.

```
10 FOR I=6 TO 13
20 READ J
30 SOUND I,J
40 NEXT I
50 DATA 31 _____ Ruisfrequentie
60 DATA 7 _____ Kiest kanalen A, B en C voor het weergeven van ruis.
70 DATA 16,16,16 _____ Wijzigt het volume van kanaal A, B en C.
80 DATA 71,2 _____ Cyclus van volumevariatië 12 Hz.
90 DATA 14 _____ Patroon van volumevariatië 14.
```


2-6 VERWERKEN VAN BESTANDEN

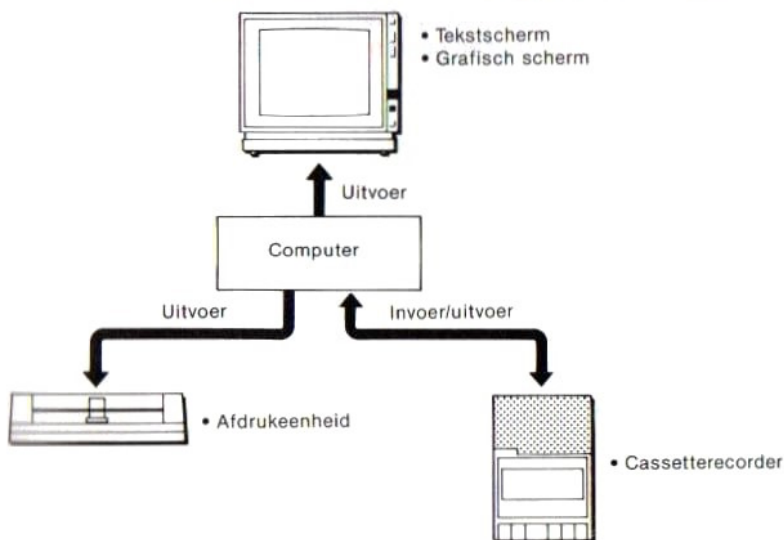
BESTANDEN EN APPARATEN

Het komt veelvuldig voor dat een blok gegevens in een programma wordt uitgewisseld tussen de computer en hierop aangesloten apparaten.

Dit is wel wat te vergelijken met het vinden van een boek op een boekenplank. Stel u houdt een dagboek bij. U heeft in uw kamer een aantal boekenplanken en op een ervan vinden we een notitieboek getiteld "dagboek". Voor het lezen of schrijven in uw dagboek begeeft u zich naar de juiste boekenplank en neemt het betreffende notitieboek van de plank.

Een computer gaat op soortgelijke wijze te werk voor het lezen of schrijven van een programma of een blok gegevens, te vergelijken met de inhoud van het dagboek. Het notitieboek waarin de gegevens komen te staan heet in computerterminologie een **bestand**. De titel "dagboek" waar we naar zoeken op de plank is bij de computer de naam die we aan een bestand gegeven hebben, de **bestandsnaam**. De boekenplanken waar we uit kiezen worden bij de computer gevormd door de aangesloten apparatuur. Als de verkeerde apparaatnaam wordt gegeven kan het gewenste bestand natuurlijk niet gevonden worden.

In MSX-BASIC zijn bevelen beschikbaar voor het uitwisselen van gegevens tussen de computer en vier verschillende aangesloten apparaten. Deze vier vormen van randapparatuur noemen we de **bestandsverwerkende apparaten**. De verbindingen tussen de apparaten en de centrale computer en de functies ziet u in de onderstaande afbeelding. de apparaten zijn in twee verschillende soorten te onderscheiden in die zin dat een soort alleen geschikt is voor uitvoer van een bestand terwijl de andere soort zowel voor uitvoer als voor invoer gebruikt kan worden.



Van de bestandsverwerkende apparaten met MSX-BASIC biedt alleen de cassetterecorder de mogelijkheid tot zowel invoer als uitvoer van bestanden, zoals aangegeven in bovenstaande afbeelding. Twee van de zgn. apparaten, het tekstschermb en het grafisch scherm, zijn gekombineerd in het beeldscherm van het TV-toestel of de videomonitor.

Apparaatnamen

Bij het uitwisselen van bestanden tussen de computer en de randapparatuur moet in het bevel altijd de naam van het gebruikte apparaat genoemd zijn. Hiervoor beschikt MSX-BASIC over een viertal **apparaatnamen**.

| Bestandsverwerkend apparaat | Apparaatnaam |
|-----------------------------|--------------|
| Cassetterecorder | CAS: |
| Tekstschermb | CRT: |
| Grafisch scherm | GRP: |
| Afdrukeenheid | LPT: |

Bestandsnamen

Elk bestand moet een naam hebben bestaande uit een rij van ten hoogste 6 lettertekens, beginnend met een letter van het alfabet. Als 7 of meer lettertekens worden ingevoerd, worden de overbodige tekens, te beginnen met het 7de, genegeerd.

Soms kan de bestandsnaam weggelaten worden, maar dit is niet aan te bevelen, aangezien dit het onmogelijk kan maken twee of meer bestanden bij in- of uitvoer van of naar cassette te onderscheiden.

PROGRAMMABESTANDEN

De volgende bevelen dienen voor het opslaan van een BASIC programma in een bestand, het laden vanuit een bestand, en het combineren van programma's.

CSAVE, CLOAD ... uitsluitend voor cassetterecorder.

SAVE, LOAD, BSAVE, BLOAD, MERGE ... apparaat naar keuze.

Voor het opslaan van een programma vanuit het geheugen op een cassette geeft u het bevel:

CSAVE "PROG1"

Bestandsnaam

of SAVE "CAS: PROG1"

Apparaatnaam Bestandsnaam

Overigens is er wel verschil tussen de beide bevelen: een programma wordt met CSAVE opgeslagen in een zgn. brugtaal, terwijl het SAVE bevel het programma in ASCII code opslaat.

Een programma dat opgeslagen is met CSAVE kan weer geladen worden met het CLOAD bevel in combinatie met dezelfde bestandsnaam. Een programma opgeslagen met SAVE kan geladen worden met LOAD. In dit laatste geval is met ook mogelijk een geladen programma samen te voegen met een programma dat zich al in het geheugen bevond. Deze mogelijkheid bestaat niet voor programma's die zijn opgeslagen met het CSAVE bevel.

Aangezien de bevelen LOAD en MERGE dienen voor invoer van een programma uit een bestand, kan hierbij als apparaatnaam alleen CAS: ingevoerd worden. Als met een SAVE bevel de apparaatnaam CRT: wordt ingevoerd is het resultaat hetzelfde als bij een LIST bevel. Een SAVE bevel met de apparaatnaam LPT: geeft hetzelfde resultaat als het uitvoeren van een LLIST bevel.

GEGEVENBESTANDEN

Ook voor het uitwisselen van gegevens tussen een BASIC programma en een bestandsverwerkend apparaat worden bestanden gebruikt.

De volgende bevelen worden gebruikt voor het invoeren en uitvoeren van gegevens van en naar bestanden.

| | |
|----------------------|---|
| OPEN | Opent een bestand. |
| PRINT # | } Voert gegevens uit naar een bestand. |
| PRINT # USING | |
| INPUT # | } Voert gegevens in vanuit een bestand. |
| LINE INPUT # | |
| CLOSE | Sluit een bestand. |

UITWISSELING MET BESTANDEN OP CASSETTE

Uitvoeren naar een bestand (Inschrijven)

Kort weergegeven is de gang van zaken voor uitvoer van gegevens als volgt:

- 1 Open een bestand met een OPEN bevel.
- 2 Schrijf gegevens naar het bestand met een PRINT # bevel.
- 3 Sluit het bestand met een CLOSE bevel.

De schrijfwijze van het OPEN bevel voor het uitvoeren van gegevens is:

```
OPEN "apparaatnaam [bestandsnaam]" FOR OUTPUT AS [#]  
bestandsnummer
```

Bij het verwerken hiervan wordt het uitvoeren van gegevens naar een nader omschreven apparaat met een nader omschreven bestandsnaam voorbereid. Wanneer daarna het bevel voor in- of uitvoer naar een bestand wordt gegeven slaat de computer eerst de gegevens op in zijn geheugen, om daarna over te gaan tot daadwerkelijke in- of uitvoer. Het in het geheugen voor de opslag van gegevens gereserveerde gebied wordt buffer genoemd. In MSX-BASIC kunnen ten hoogste 16 buffers worden gereserveerd. Het bestandsnummer in het OPEN bevel verwijst naar een van deze 16 buffers, waarvan het eerst geopende altijd het nummer 1 moet dragen.

Nadat een bestand is geopend met het OPEN bevel wordt de uitvoer van gegevens in gang gezet door een PRINT # bevel.

```
PRINT # bestandsnummer, uitdrukking [scheidingsteken] ...
```

Het bestandsnummer dient gelijk te zijn aan dat in het OPEN bevel.

Bij het uitvoeren van gegevens naar een bestand met een PRINT # bevel wordt automatisch een terugkeercodes (&H0D) en een regelopschuifcode (&H0A) na op tussen de gegevens geschreven. Bij het lezen van de gegevens gelden deze twee codes als scheidingstekens.

Wanneer de gegevens bestaan uit meerdere rijen lettertekens die worden uitgevoerd met een PRINT # bevel dienen de rijen onderling gescheiden te worden door een komma tussen aanhalingstekens " , ".

Het bevel zal dus zijn geheel bijvoorbeeld zo luiden:

```
PRINT #1,A$;" , ";B$
```

De komma fungeert hier als scheidingstekens en de A\$ en B\$ rijen worden bij het lezen van de gegevens uit het bestand als aparte gegevens beschouwd.

Wanneer de gegevens uit getalswaarden bestaan worden ze automatisch onderling gescheiden.

Na het uitvoeren van gegevens dient een bestand gesloten te worden met een CLOSE bevel.

```
CLOSE [#] bestandsnummer
```

Aangezien door het sluiten van een bestand de relatie van het bestand en het nummer verbroken wordt, is het hierna mogelijk een ander bestand met hetzelfde bestandsnummer te openen.

Voorbeeld programma

```
10 DIM A$(1,3)
20 OPEN "CAS:DATA" FOR OUTPUT AS #1
30 FOR I=0 TO 1
40 FOR J=0 TO 3
50 READ A$(I,J)
60 PRINT #1,A$(I,J);", ";
70 NEXT J
80 NEXT I
90 CLOSE #1
100 END
110 DATA JAPAN,ENGLAND,FRANCE,U.S.A
120 DATA TOKYO,LONDON,PARIS,NEW YORK
```

Bij het uitvoeren van dit programma worden de rij-gegevens "JAPAN", komma (,), "ENGLAND" enz. in die volgorde naar cassette geschreven. De gegevens worden in feite als volgt geschreven.

```
JAPAN,ENGLAND,FRANCE,U.S.A,TOKYO,LONDON,
PARIS,NEW YORK,
```

Regel 60 voegt als leesteken tussen de gegevens een komma in opdat de gegevens later bij het invoeren met een INPUT # bevel gemakkelijk van elkaar te onderscheiden zijn.

Invoeren vanuit een bestand (Uitlezen)

Kort weergegeven is de gang van zaken voor invoer van gegevens als volgt:

- 1 Open een bestand met een OPEN bevel.
- 2 Lees gegevens uit het bestand met een INPUT # of LINE INPUT # bevel (Kent de gegevens aan een variabele toe).
- 3 Sluit het bestand met een CLOSE bevel.

De schrijfwijze van het OPEN bevel voor het invoeren van gegevens is:

```
OPEN "apparaatnaam [bestandsnaam]" FOR INPUT AS [#]  
bestandsnummer
```

Bij het verwerken hiervan wordt het invoeren van gegevens vanuit een nader omschreven bestand voorbereid. Het eerst geopende bestand moet altijd het nummer 1 dragen.

Nadat een bestand is geopend met het OPEN bevel wordt de invoer van gegevens in gang gezet door een INPUT # bevel.

De gegevens die worden gelezen met een INPUT # bevel worden verwerkt op de wijze als aangegeven in onderstaande tabel.

| | Bij numerieke gegevens | Bij rijen lettertekens |
|---|---|--|
| Spatie, terugkeercode, regelopschuifcode voor de gegevens | Worden genegeerd | Worden genegeerd |
| Scheidingstekens of leestekens tussen de gegevens | Spatie, komma, terugkeercode, regelopschuifcode | Komma, terugkeercode, regelopschuifcode. Voor invoer van 255 lettertekens. |
| Aanhalingstekens die de gegevens omsluiten (" ") | — | Invoer tussen " " wordt als een gegeven beschouwd. |

Het LINE INPUT # bevel dient alleen voor het uitlezen van een rij letterteken-gegevens die worden ingevoerd met als enige scheidingstekens de terugkeercode.

Na het beëindigen van de invoer van gegevens wordt het bestand gesloten met een CLOSE bevel om de relatie tussen het bestand en het bestandsnummer te verbreken.

Voorbeeld programma

```
10 DIM A$(1,3)
20 OPEN "CAS:DATA" FOR INPUT AS #1
30 FOR I=0 TO 1
40 FOR J=0 TO 3
50 INPUT #1,A$(I,J)
60 NEXT J
70 NEXT I
80 CLOSE #1
90 FOR J=0 TO 3
100 PRINT A$(0,J),A$(1,J)
110 NEXT J
```

Dit programma dient voor het uitlezen van gegevens uit een bestand genaamd "DATA" dat door het vorige programma op cassette was uitgeschreven. De invoer vindt plaats in regels 20 t/m 80 en de gegevens worden door regels 90 t/m 110 op het scherm weergegeven. In regel 50 worden de gegevens alle toegekend aan de A\$(I,J) lijstvariabele.

```
10 OPEN "CAS:DATA" FOR INPUT AS #1
20 INPUT #1,A$
30 PRINT A$
40 GOTO 20
```

Wat gebeurt er als we proberen de gegevens in het bestand "DATA" met dit programma in te voeren? JAPAN, ENGELAND ... en alle andere gegevens worden toegekend aan de A\$ variabele en vervolgens op het scherm weergegeven. Na het invoeren van het laatste gegeven, NEW YORK, blijft het programma proberen meer gegevens in te voeren. Wanneer dit zich voordoet en het einde van een bestand wordt a.h.w. overschreden, dan treedt er een

Input past end

fout op. Dit kunt u voorkomen met de EOF functie.

```
10 OPEN "CAS:DATA" FOR INPUT AS #1
15 IF EOF(1)=-1 THEN GOTO 50
20 INPUT #1,A$
30 PRINT A$
40 GOTO 15
50 CLOSE #1
```

De EOF (bestandsnummer) functie geeft de waarde -1 wanneer het laatste gegeven van een bestand uitgelezen is. In dit programma controleert de functie telkens voor het invoeren van een gegeven of er inderdaad nog wel gegevens in het bestand over zijn.

WEERGEVEN VAN LETTERTEKENS OP HET GRAFISCHE SCHERM

Na het kiezen van een de grafische schermen met een SCREEN 2 of SCREEN 3 bevel is het niet meer mogelijk lettertekens op het scherm te brengen met een PRINT bevel.

Om desondanks lettertekens op het grafische scherm weer te geven bedienen we ons van een methode waarbij het grafische scherm wordt opgevat als een bestandsverwerkend apparaat, en de lettertekens die we willen afbeelden als een bestand met gegevens.

```
10 SCREEN 2
20 OPEN "GRP:" FOR OUTPUT AS #1
30 PRINT #1, "How do you do?"
40 GOTO 40
```

Bij het verwerken van dit programma wordt het grafische scherm gekozen en "How do you do?" weergegeven.

Voor het bepalen van de plaats waar de letters op het scherm komen is het nodig eerst een grafisch bevel te geven. Daarna wordt de laatst genoemde positie hiervan (een van de 256 horizontaal bij 192 vertikale stippen) gebruikt als de linkerbovenhoek van een 8x6 blok van stippen dat plaats biedt aan het eerste letterteken van de uitgevoerde rij.

```
10 SCREEN 2
20 OPEN "GRP:" FOR OUTPUT AS #1
30 PRESET (100,50)
40 PRINT #1, "How do you do?"
50 GOTO 50
```

In dit programma wordt de positie (100, 50) van het PRESET bevel in regel 30 gebruikt als de linkerbovenhoek van de rij lettertekens die wordt uitgevoerd in regel 40.

AANTAL TEGELIJK TE OPENEN BESTANDEN

Bij het inschakelen van MSX-BASIC is het niet mogelijk meer dan een bestandsnaam in te voeren. Met andere woorden, er kan in een programma maar een bestand tegelijk geopend zijn. Voor het openen van meerdere bestanden tegelijk moet het gewenste aantal eerst worden aangegeven met het bevel (bijv.):

```
MAXFILES=5
```

Deze regel zorgt dat er vanaf dat moment 5 bestanden met de bestandsnummers 1 t/m 5 tegelijk geopend kunnen zijn. Het grootste aantal bestanden dat op deze wijze tegelijk te openen is bedraagt 15. Overigens is een bestand met het bestandsnummer 0 speciaal bestemd voor gebruik met de CSAVE, CLOAD, CLOAD?, SAVE en LOAD bevelen, zodat na het bevel

`MAXFILES=0`

voor in- en uitvoer alleen de bevelen CSAVE, CLOAD, CLOAD?, SAVE en LOAD beschikbaar zijn.

2-7 ONDERBREKINGEN

Een onderbreking wordt veroorzaakt door een ingreep, naar keuze, van buitenaf en dient om een programma tijdens de verwerking tijdelijk te stoppen, om tussentijds andere bewerkingen uit te voeren. De reeks van handelingen die worden uitgevoerd na het onderbreken van een programma wordt een **onderbrekingsverwerkingsprogramma** of **onderbrekingsroutine** genoemd.

Een ingreep die soortgelijk is aan de onderbreking is de subroutine. Een subroutine wordt echter uitsluitend gestart door een GOSUB bevel in MSX-BASIC. Met andere woorden, een subroutine is van tevoren gepland en vindt geheel binnen het programma plaats.

Voor het veroorzaken van een onderbreking is echter altijd een externe ingreep nodig (zoals bijvoorbeeld het indrukken van de **F1** toets).

Na het beëindigen van een onderbrekingsroutine wordt de verwerking van het hoofdprogramma gewoonlijk op dezelfde wijze hervat als na het uitvoeren van een subroutine.

MSX-BASIC ONDERBREKINGEN

MSX-BASIC beschikt over een aantal bevelen die gebruikt kunnen worden om na het inschakelen van een onderbreking over te gaan tot het uitvoeren van een onderbrekingsroutine. Een onderbreking kan op de volgende wijzen ingeschakeld worden. Van te voren moet een onderbreking eerst met een van de hiervoor bestemde bevelen aangekondigd worden, waarbij de beginregel van de onderbrekingsroutine moet worden vastgelegd.

| | |
|--|--|
| Een onderbreking kan worden ingeschakeld door: | Bevel om onderbreking aan te kondigen |
| Indrukken van een regelnummer | ON KEY GOSUB regelnummer |
| Indrukken van spatiebalk of trekkerknop van een joystick | ON STRIG GOSUB regelnummer |
| Indrukken van CTRL en STOP | ON STOP GOSUB regelnummer |
| Overlappen van sprites | ON SPRITE GOSUB regelnummer |
| Het verstrijken van een bepaalde periode. | ON INTERVAL=periode GOSUB regelnummer |

Bijvoorbeeld,

```
ON KEY GOSUB 1000
```

is een bevel dat bij indrukken van een funktietoets de verwerking doet vervolgen met de routine vanaf regel 1000.

GEBRUIK VAN ONDERBREKINGEN

Een onderbreking kan niet daadwerkelijk worden ingeschakeld door het geven van een bevel met ON—GOSUB. Hierna moet nog een bevel worden gegeven dat de onderbreking geldig maakt. In het voorbeeld waar de onderbreking wordt ingeschakeld door het indrukken van de **F1** toets is dient u het bevel te geven:

```
KEY(1) ON
```

Er zijn vijf bevelen die dienen voor het geldig maken van een onderbreking.

| Bevel | Geldig gemaakte onderbreking |
|------------------------------|--|
| KEY (nummer funktietoets) ON | Onderbreking met een funktietoets |
| STRIG (nummer joystick) ON | Onderbreking met de spatiebalk of een joystick |
| STOP ON | Onderbreking met de CTRL en STOP toetsen |
| SPRITE ON | Onderbreking door het overlappen van sprites |
| INTERVAL ON | Onderbreking na een bepaalde periode. |

Voorbeeld programma

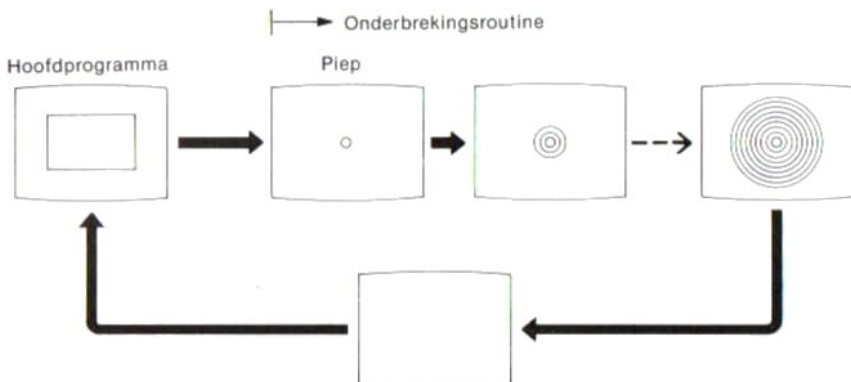
```
10 ON KEY GOSUB 100
20 KEY(1) ON
30 SCREEN 2
40 LINE (50,50)-(200,150),,B
50 GOTO 40
100 'SUBROUTINE
110 BEEP:CLS
120 FOR I=10 TO 90 STEP 10
130 CIRCLE (120,100),I
140 NEXT I
150 CLS
160 RETURN 40
```

Hoofdprogramma

Onderbrekingsroutine

Dit programma is door het invoegen van regel 10 en 20 zo opgezet dat bij het indrukken van de **F1** funktietoets wordt doorgegaan met een subroutine die loopt vanaf regel 110.

Bij het verwerken van dit programma wordt door regel 40 en 50 van het hoofdprogramma onafgebroken een rechthoek op het scherm afgebeeld. Wanneer echter de **F1** toets wordt ingedrukt vindt een onderbreking plaats waarna volgens het bevel van regel 10 de verwerking doorgaat met regel 100. Het merkbare resultaat is dat de rechthoek verdwijnt met een piepton (BEEP:CLS) en dat er 9 cirkels op het scherm worden getekend. Na het trekken van de laatste cirkel wordt het scherm gewist en verschijnt weer de rechthoek van regel 40 en 50.



ONGELDIG MAKEN VAN EEN ONDERBREKING

Laten we aan het bovenstaande programma eens de volgende regel toevoegen.

```
105 KEY(1) OFF
```

Verwerk het programma. Bij het eenmaal indrukken van de **F1** toets vindt een onderbreking plaats. Daarna volgt op het nogmaals indrukken van de **F1** toets geen enkele onderbreking meer.

De oorzaak hiervan is gelegen in het feit dat na het eenmaal uitvoeren van de onderbrekingsroutine het bevel van regel 105

```
KEY(1) OFF
```

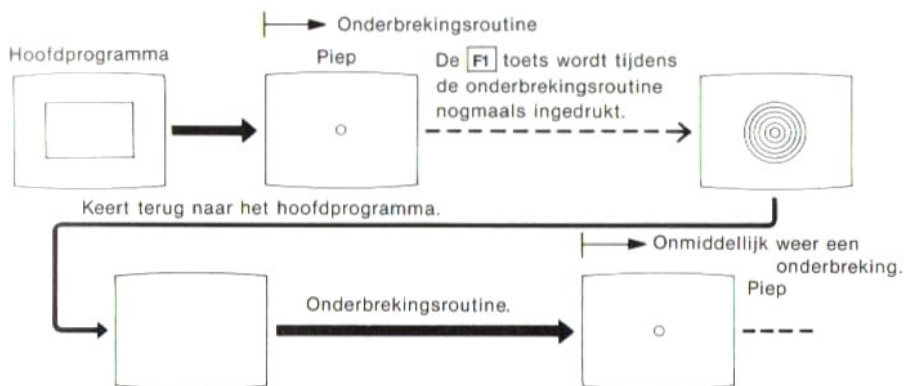
volgde, waarmee de onderbreking met de **F1** toets ongeldig werd gemaakt.

VASTHOUDEN VAN EEN ONDERBREKING

Wanneer bij een onderbreking de verwerking van een programma wordt overgebracht naar een onderbrekingsroutine, treedt de zogenaamde vasthoudstand van een onderbreking op. Wanneer in deze stand opnieuw getracht wordt de verwerking te onderbreken heeft dit geen direct effect, maar voor terugkeer naar het hoofdprogramma met een RETURN bevel wordt automatisch een – ON bevel ingelast. Het hoofdprogramma wordt in dat geval nog niet uitgevoerd, maar er wordt snel opnieuw overgegaan tot uitvoeren van de onderbrekingsroutine.

Anders gezegd, tijdens de onderbrekings-vasthoudstand wordt nooit tussentijds teruggedaan naar de beginregel van de onderbrekingsroutine, maar wordt een gewenste onderbreking vastgehouden tot de onderbrekingsroutine geheel beëindigd is, waarna deze in zijn geheel herhaald wordt.

Voor het programma op blz. 189 betekent dit dat bij eenmaal indrukken van de **F1** toets door de onderbrekingsroutine 9 cirkels worden getekend. Als echter voor het completeren van de laatste cirkel opnieuw de **F1** toets wordt ingedrukt gebeurt er ogenschijnlijk niets. Na de laatste cirkel wordt teruggegaan naar het hoofdprogramma. Dan vindt echter wel opnieuw een onderbreking plaats, voor de tweede maal dat de **F1** toets is ingedrukt, en er worden in plaats van de rechthoek opnieuw 9 cirkels getekend.



GELDIG MAKEN VAN EEN ONDERBREKING TIJDENS EEN ONDERBREKINGSROUTINE

Om een onderbreking ook in de vasthoudstand, dus tijdens het uitvoeren van een onderbrekingsroutine, geldig te maken geeft u een voor geldig maken bestemd bevel als `KEY(1) ON`. Het resultaat is dan dat als een onderbrekingsroutine op zijn beurt onderbroken wordt, deze onmiddellijk vanaf het begin herhaald zal worden.

Voorbeeld programma

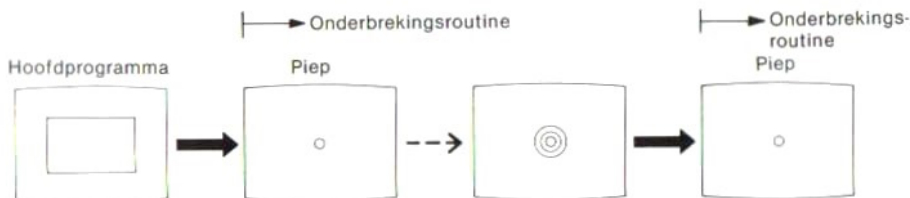
```

10 ON KEY GOSUB 100
20 KEY(1) ON
30 SCREEN 2
40 LINE (50,50)-(200,150),.B
50 GOTO 40
100 *SUBROUTINE
105 KEY(1) ON
110 BEEP:CLS
120 FOR I=10 TO 90 STEP 10
130 CIRCLE (120,100),I
140 NEXT I
150 CLS
160 RETURN 40

```

Dit programma is vrijwel gelijk aan het vorige, met als enige wijziging het KEY(1) ON bevel dat in regel 105 is ingevoegd.

Als gevolg van deze toevoeging brengt ook tijdens het tekenen van de cirkels het indrukken van de **F1** toets onmiddellijk een onderbreking teweeg, waarna de onderbrekingsroutine die begint met regel 100 in zijn geheel opnieuw wordt gestart.



TIJDELIJK VASTHOUDEN VAN EEN ONDERBREKING

Na het geldig maken van een onderbreking tijdens de onderbrekingsroutine met een—ON bevel is het mogelijk tijdelijk een vasthoudstand in te stellen door het invoegen van een—STOP bevel.

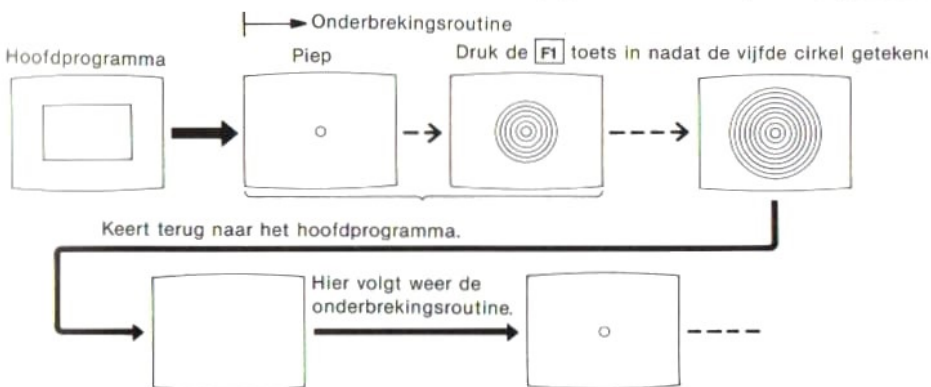
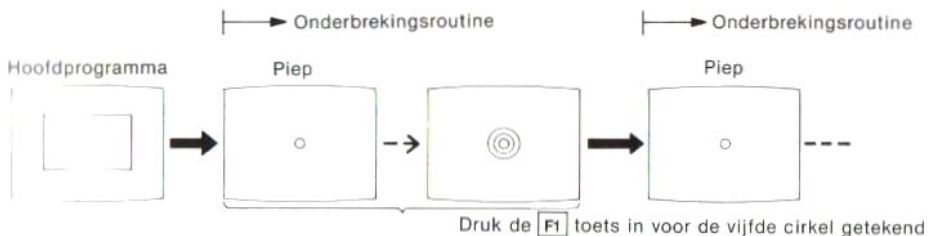
Voorbeeld programma

```

10 ON KEY GOSUB 100
20 KEY(1) ON
30 SCREEN 2
40 LINE (50,50)-(200,150),,B
50 GOTO 40
100 'SUBROUTINE
105 KEY(1) ON
110 BEEP:CLS
120 FOR I=10 TO 90 STEP 10
130 CIRCLE (120,100),I
135 IF I=50 THEN KEY(1) STOP
140 NEXT I
150 CLS
160 RETURN 40

```

Dit programma is weer bijna gelijk aan het vorige, maar deze keer met de extra bijzonderheid dat wanneer I de waarde 50 bereikt, in regel 135 het KEY(1) STOP bevel in werking treedt. Zodoende vindt bij het indrukken van de **F1** toets voor het tekenen van de vijfde cirkel onmiddellijk weer een onderbreking plaats, terwijl bij het indrukken na het tekenen van de vijfde cirkel de vasthoudstand wordt ingesteld, zodat de onderbreking pas later volgt.



ONDERBREKING DOOR OVERLAPPEN VAN SPRITES

Zodra twee of meer sprite-patronen elkaar met minimaal 1 stip overlappen kan hierop een onderbreking volgen door de ON STRITE GOSUB en SPRITE ON bevelen.

In het volgende programma komen UFO's zowel van links als van rechts gevlogen, en wanneer ze elkaar overlappen klinkt een piepton.

```

10 SCREEN 2
20 SPRITE$(0)=CHR$(&H3C)+CHR$(&H7E)+CHR$
(&H81)+CHR$(&H81)+CHR$(&HFF)+CHR$(&H7E)+
CHR$(&H24)+CHR$(&H42)
30 ON SPRITE GOSUB 100
40 SPRITE ON
50 FOR X=0 TO 255
60 PUT SPRITE 0,(X,100),15,0
70 PUT SPRITE 1,(255-X,100),10,0
80 NEXT X
90 END
100 SPRITE OFF
110 BEEP
120 SPRITE ON
130 RETURN

```

2-8 MACHINETAAL SUBROUTINES

MSX-BASIC biedt de mogelijkheid een programma te schrijven in de machinetaal van de Z-80A (de centrale verwerkingseenheid van de MSX persoonlijke computer). Er wordt dan vanuit BASIC overgeschakeld naar de machinetaal voor het sturen van de computer, en nadat het programma in machinetaal geheel verwerkt is kan het resultaat worden toegekend aan een variabele die in BASIC gedefinieerd is.

DEFINIËREN BEGINADRES VAN EEN MACHINETAAL SUBROUTINE

Reserveer eerst met een **CLEAR** bevel een gebied in het geheugen voor het schrijven van de machinetaal subroutine. Definieer dan het beginadres van de subroutine met een **DEFUSR** bevel.

DEFUSR N = beginadres

N kan een geheel getal zijn van 0 tot 9. Er kunnen 10 subroutines met hun beginadressen als **USR** functies gedefinieerd worden.

```
CLEAR 200,&HFFFF  
DEFUSR1=&HE000
```

In deze regels wordt een machinetaal subroutine vanaf adres &HE000 gedefinieerd als de **USR 1** functie.

UITVOEREN VAN MACHINETAAL SUBROUTINE

Variabele=USR N(I)

Door dit bevel wordt de verwerking van de machinetaal subroutine gestart. Nadat de verwerking beëindigd is wordt het eindresultaat aan de gekozen variabele toegekend, waarna zonder onderbreking het BASIC programma weer hervat wordt.

Bij het overschakelen vanuit BASIC naar een machinetaal subroutine wordt de waarde van "I", de parameter van de **USR** functie, naar de subroutine overgebracht.

```
X=USR1(I)
```

Door dit bevel wordt de waarde van de variabele I opgeslagen op een van de hieronder gegeven geheugenadressen, afhankelijk van het soort variabele dat I is, en tegelijk wordt deze laatste informatie, het soort of type, in register A vastgelegd. Het beginadres waar de waarde van I is opgeslagen wordt ingevoerd in het HL register.

| Soort of type variabele I | Gegeven in A* register | HL register adres-aanduiding | Adres waar de waarde van I is vastgelegd |
|-------------------------------|------------------------|------------------------------|--|
| Geheel getal | 2 | &HF7F6 | &HF7F8—&HF7F9 |
| Enkele precisie getalswaarde | 4 | | &HF7F6—&HF7F9 |
| Dubbele precisie getalswaarde | 8 | | &HF7F6—&HF7FD |

*Dezelfde waarde wordt vastgelegd in geheugenadres &HF663.

Wanneer I een rij-variabele is zijn de verschillende gegevens als volgt:

| Gegeven in A register | Gegeven in DE register | Beschrijving rij lettertekens |
|-----------------------|--|---|
| 3 | Beginadres beschrijving rij lettertekens | Eerst byte: lengte van de rij lettertekens Tweede en derde byte: beginadres van het gebied waar de rij lettertekens is opgeslagen. |

Na het volledig verwerken van de machinetaal subroutine wordt het eindresultaat aan variabele X toegewezen door het register en het geheugen tijdens het beëindigen in te stellen.

| Soort of type eindresultaat | &HF633 geheugen-adres | DE register | HL register | Opslagadres eindresultaat |
|-------------------------------|-----------------------|--|-------------|---|
| Geheel getal | 2 | | &HF7F6 | &HF7F8—&HF7F9 |
| Enkele precisie getalswaarde | 4 | | &HF7F6 | &HF7F6—&HF7F9 |
| Dubbele precisie getalswaarde | 8 | | &HF7F6 | &HF7F6—&HF7FD |
| Rij lettertekens | 3 | Beginadres beschrijving rij lettertekens | | Beginadres opslaggebied aangegeven door tweede en derde byte van rijbeschrijving. |

VOORBEREIDING MACHINETAAL

Een machinetaal subroutine kan in het geheugen geschreven worden met een **POKE** bevel.

Voor het terugkeren vanuit een machinetaal subroutine naar het BASIC hoofdprogramma moet een **RET** bevel gegeven worden.

HOOFDSTUK 3

TABELLEN

1. LETTERTEKENS

LETTERTEKENS GEBRUIKT IN MSX-BASIC

Alle tekens die in onderstaande codetabel voor lettertekens zijn opgenomen kunnen op het scherm weergegeven worden.

| Hexadecimale code | 00—1F | | 20—3F | | 40—5F | | 60—7F | |
|-------------------|-------|-------|-------|----------|-------|-------|-------|-------|
| | code | teken | code | teken | code | teken | code | teken |
| 0 | 0 | (nul) | 32 | (spatie) | 64 | @ | 96 | \ |
| 1 | 1 | ☺ | 33 | ! | 65 | A | 97 | a |
| 2 | 2 | ☹ | 34 | " | 66 | B | 98 | b |
| 3 | 3 | ♥ | 35 | # | 67 | C | 99 | c |
| 4 | 4 | ♦ | 36 | \$ | 68 | D | 100 | d |
| 5 | 5 | ♣ | 37 | % | 69 | E | 101 | e |
| 6 | 6 | ♠ | 38 | & | 70 | F | 102 | f |
| 7 | 7 | • | 39 | ' | 71 | G | 103 | g |
| 8 | 8 | ■ | 40 | (| 72 | H | 104 | h |
| 9 | 9 | ○ | 41 |) | 73 | I | 105 | i |
| A | 10 | ◻ | 42 | * | 74 | J | 106 | j |
| B | 11 | ♂ | 43 | + | 75 | K | 107 | k |
| C | 12 | ♀ | 44 | , | 76 | L | 108 | l |
| D | 13 | ♪ | 45 | — | 77 | M | 109 | m |
| E | 14 | ♫ | 46 | . | 78 | N | 110 | n |
| F | 15 | ✳ | 47 | / | 79 | O | 111 | o |
| 0 | 16 | † | 48 | ∅ | 80 | P | 112 | p |
| 1 | 17 | ‡ | 49 | 1 | 81 | Q | 113 | q |
| 2 | 18 | ⋈ | 50 | 2 | 82 | R | 114 | r |
| 3 | 19 | ‡ | 51 | 3 | 83 | S | 115 | s |
| 4 | 20 | ‡ | 52 | 4 | 84 | T | 116 | t |
| 5 | 21 | ‡ | 53 | 5 | 85 | U | 117 | u |
| 6 | 22 | ‡ | 54 | 6 | 86 | V | 118 | v |
| 7 | 23 | — | 55 | 7 | 87 | W | 119 | w |
| 8 | 24 | ┘ | 56 | 8 | 88 | X | 120 | x |
| 9 | 25 | ┘ | 57 | 9 | 89 | Y | 121 | y |
| A | 26 | ┘ | 58 | : | 90 | Z | 122 | z |
| B | 27 | ┘ | 59 | ; | 91 | [| 123 | { |
| C | 28 | ✕ | 60 | < | 92 | \ | 124 | ! |
| D | 29 | ／ | 61 | = | 93 |] | 125 | |
| E | 30 | ／ | 62 | > | 94 | ^ | 126 | ~ |
| F | 31 | + | 63 | ? | 95 | — | 127 | |

| Hexadecimale code | 80—9F | | A0—BF | | C0—DF | | E0—FF | |
|----------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| | code | teken | code | teken | code | teken | code | teken |
| 0 | 128 | Ç | 160 | á | 192 | ■ | 224 | α |
| 1 | 129 | ü | 161 | í | 193 | ■ | 225 | β |
| 2 | 130 | é | 162 | ó | 194 | ■ | 226 | Γ |
| 3 | 131 | â | 163 | ú | 195 | ■ | 227 | Π |
| 4 | 132 | ã | 164 | ñ | 196 | ■ | 228 | Σ |
| 5 | 133 | à | 165 | Ñ | 197 | ■ | 229 | σ |
| 6 | 134 | á | 166 | ä | 198 | ■ | 230 | μ |
| 7 | 135 | ç | 167 | ö | 199 | ■ | 231 | γ |
| 8 | 136 | ê | 168 | z | 200 | ■ | 232 | Φ |
| 9 | 137 | ë | 169 | Γ | 201 | ■ | 233 | Θ |
| A | 138 | è | 170 | Γ | 202 | ■ | 234 | Ω |
| B | 139 | ï | 171 | ½ | 203 | ▨ | 235 | δ |
| C | 140 | î | 172 | ¼ | 204 | ▨ | 236 | ∞ |
| D | 141 | ì | 173 | ì | 205 | ▼ | 237 | ∅ |
| E | 142 | Ä | 174 | « | 206 | ▲ | 238 | ε |
| F | 143 | Å | 175 | » | 207 | ▶ | 239 | ∩ |
| 0 | 144 | É | 176 | Ã | 208 | ◀ | 240 | ≡ |
| 1 | 145 | æ | 177 | ä | 209 | ⌘ | 241 | ± |
| 2 | 146 | Æ | 178 | ĩ | 210 | ⌘ | 242 | ≥ |
| 3 | 147 | ô | 179 | ī | 211 | ■ | 243 | ≤ |
| 4 | 148 | ö | 180 | Ö | 212 | ■ | 244 | ∫ |
| 5 | 149 | ò | 181 | ō | 213 | ■ | 245 | ∫ |
| 6 | 150 | û | 182 | Û | 214 | ■ | 246 | + |
| 7 | 151 | ù | 183 | ū | 215 | ■ | 247 | ≠ |
| 8 | 152 | ÿ | 184 | Ï | 216 | △ | 248 | ○ |
| 9 | 153 | Ö | 185 | Û | 217 | † | 249 | • |
| A | 154 | Ü | 186 | ¼ | 218 | ε | 250 | - |
| B | 155 | ç | 187 | √ | 219 | ■ | 251 | √ |
| C | 156 | £ | 188 | ◊ | 220 | ■ | 252 | η |
| D | 157 | ¥ | 189 | ‰ | 221 | ■ | 253 | ² |
| E | 158 | ₣ | 190 | ¶ | 222 | ■ | 254 | ■ |
| F | 159 | f | 191 | § | 223 | ■ | 255 | |

Tekens met een lettertekencode van 2 bytes

De lettertekens behorend bij de codes 1 t/m 31 (decimaal) in bovenstaande tabel hebben elk een lettertekencode die 2 bytes beslaat. De in de tabel genoemde codes moeten worden voorafgegaan door een 1, en verhoogd met 64.

In- en uitvoer van lettertekencodes

Invoer via het toetsenbord

Normale lettertekens de code van 1 byte wordt ingevoerd.

Voorbeeld: Code 65 (decimaal) voor de letter "A"

Tekens met 2-byte code een 1 en de rest van de code wordt ingevoerd.

Voorbeeld: Codes 1 en 67 voor het teken "♥"

Uitvoer met de CHR\$ functie

Normale lettertekens de code van 1 byte wordt gebruikt als parameter.

Voorbeeld: CHR\$(66) voor de letter "B"

Tekens met 2-byte code 2 CHR\$ functies worden gebruikt, de eerste CHR\$(1), en de tweede een CHR\$ functie met als parameter de bovengenoemde code.

Voorbeeld: CHR\$(1); CHR\$(68) voor het teken "◆".

2. KONSTANTEN EN VARIABELEN

KONSTANTEN

Konstanten ... geven altijd een gelijkblijvende waarde aan.

Konstanten die gebruikt worden in MSX-BASIC

| | | |
|--------------------------------|--|---|
| Numerieke konstanten | Gehele getallen | Decimale uitdrukkingen Hexadecimale uitdrukkingen Octale uitdrukkingen Binaire uitdrukkingen |
| | Vaste komma/ decimale punt Drijvende komma/ decimale punt | |
| Letterteken-konstanten (rijen) | | |

Numerieke konstanten

| | |
|---|--|
| Gehele getallen | Bereik van - 32768 tot 32767. Decimale uitdrukking: normale schrijfwijze (bijv. ... 123, - 4567) Hexadecimale uitdrukking: eerst &H (bijv. ... &HA0) Octale uitdrukking: eerst &O of & (bijv. &O146, &765) Binaire uitdrukking: eerst &B (bijv. ... &B11010110) |
| Drijvende komma (in feite decimale punt) | 6 (enkele precisie) of 14 (dubbele precisie) cijfers in uitdrukking. Exponentieel gedeelte: van - 64 tot + 62 (grondtal is 10) (Bijv.: 1.345E - 10) |

Enkele precisie en dubbele precisie

Zowel met vaste komma } Naar keuze enkele precisie of dubbele
als met drijvende komma } precisie (beginstand is dubbele precisie)

| | |
|---|---|
| Enkele precisie (6 cijfers in uitdrukking) | Macht of exponent uitgedrukt met E: bijv.: 12.34E3 Met aan het eind een uitroepteken (!): bijv.: 18.32! |
| Dubbele precisie (14 cijfers in uitdrukking) | Macht of exponent uitgedrukt met D: bijv.: 23.456789D10 Met aan het eind geen indikatie: bijv.: 3456789012.34 Met aan het eind het teken #: bijv.: 3456789012.34 # |

Rij-konstanten

Rijen van ten hoogste 255 lettertekens tussen " " (dubbele aanhalingstekens).

Rij-konstante die geen lettertekens, zelfs geen spatie bevat: ... Lege rij.

```
"HOW ARE YOU?"  
"@980"  
"" (lege rij)
```

VARIABELEN

Namen van variabelen

- Alleen de eerste twee lettertekens zijn van belang (het eerste teken moet een letter van het alfabet zijn).
- Noch een in MSX-BASIC gereserveerd woord (bevelnamen, funktienamen, enz.), noch een rij lettertekens die een dergelijk woord bevat kan als naam van een variabele gebruikt worden.

Definitie soort variabele

| Definitie met een definitieteken | Definitie met een DEF bevel | Soort variabele onder definitie |
|----------------------------------|-----------------------------|--------------------------------------|
| % achter de naam Bijv.: A% | DEFINT Bijv.: DEFINT A | Geheel getal-variabele |
| ! achter de naam Bijv.: B! | DEFSNG Bijv.: DEFSNG B | Enkele precisie-variabele |
| # achter de naam Bijv.: C# | DEFDBL Bijv.: DEFDBL C | Dubbele precisie-variabele |
| \$ achter de naam Bijv.: D\$ | DEFSTR Bijv.: DEFSTR D | Rij-variabele (voor lettertekens) |

- Wanneer bij definitie met een DEF bevel (DEFINT e.d.) achter de naam van de variabele bovendien een definitieteken wordt gezet, dan heeft dit laatste de prioriteit, m.a.w. het teken bepaalt de soort variabele.

Omzetting soort variabele

| Voorbeeld | Beschrijving |
|--|--|
| <pre>A%=100.5 PRINT A% 100</pre> | <p>Als aan een variabele gedefinieerd voor gehele getallen een konstante van een ander soort wordt toegekend, dan wordt bij de toekenning de soort van de variabele aangehouden.</p> |
| <pre>B=8!/7! PRINT B 1.1428571428571</pre> | <p>De uitkomst van de berekening wordt toegekend met dubbele precisie.</p> |
| <pre>B!=8/7 PRINT B! 1.14286</pre> | <p>Aangezien op de variabele een definitieteken voor enkele precisie volgt, wordt de uitkomst van de berekening met enkele precisie toegekend.</p> |
| <pre>PRINT 6.7 OR 4.3 6</pre> | <p>De logische bewerking wordt uitgevoerd nadat de gegevens zijn omgezet in gehele getallen.</p> |
| <pre>A!=1.2345678 B=A! PRINT B 1.23457</pre> | <p>Wanneer een gegeven met enkele precisie, dus met zes cijfers van belang, wordt toegewezen aan een variabele met dubbele precisie, dan blijft het aantal cijfers gelijk.</p> |

3. UITDRUKKINGEN EN HUN GEBRUIK

DELEN VAN UITDRUKKINGEN

Konstanten, variabelen, functies en rekenkundige symbolen.

EVALUATIE VAN REKENKUNDIGE UITDRUKKINGEN

| Rekenkundige operator | Betekenis | Voorbeeld | Voorrangnummer |
|-----------------------|-------------------------------------|---------------------|----------------|
| + | Optellen ($X + Y$) | $X + Y$ | 6 |
| - | Aftrekken ($X - Y$) | $X - Y$ | |
| * | Vermenigvuldigen ($X \times Y$) | $X * Y$ | 3 |
| / | Delen ($X \div Y$) | X / Y | |
| ^ | Machtsverheffen (X^2) | X^2 | 1 |
| - | Verandert het teken ($-X$) | $-X$ | 2 |
| \ | Delen met gehele getallen | $6.7 \setminus 2.3$ | 4 |
| MOD | Rest van deling met gehele getallen | $X \text{ MOD } 10$ | 5 |

VERGELIJKINGEN

De waarde van twee gegevens wordt vergeleken en het resultaat wordt als waar (-1) of onwaar gegeven.

| Vergelijkingsoperator | Betekenis | Voorbeeld |
|-----------------------|-------------------|------------------------|
| = | Gelijk | $X=Y, X\$=Y\$$ |
| < | Kleiner | $X < Y, X\$ < Y\$$ |
| > | Groter | $X > Y, X\$ > Y\$$ |
| <>, >< | Ongelijk | $X < > Y, X\$ > < Y\$$ |
| <=, =< | Kleiner of gelijk | $X < = Y, X\$ < = Y\$$ |
| >=, => | Groter of gelijk | $X > = Y, X\$ > = Y\$$ |

LOGISCHE UITSPRAKEN

Logische uitdrukkingen dienen voor het uitvoeren van logische bewerkingen met numerieke konstanten, variabelen en functies.

Logische bewerking ... Zet gegevens om in een geheel getal van 16 bits binair, en voert voor elk bit afzonderlijk een verwerking uit.

| Logische bewerking | Resultaat logische bewerking voor elk bit | | | | | | | | | | | | | | | |
|--|---|---------|-------|---------|---|---|---|---|---|---|---|---|---|---|---|---|
| NOT (logische ontkenning) | <table border="1"><thead><tr><th>X</th><th>NOT X</th></tr></thead><tbody><tr><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td></tr></tbody></table> | X | NOT X | 1 | 0 | 0 | 1 | | | | | | | | | |
| X | NOT X | | | | | | | | | | | | | | | |
| 1 | 0 | | | | | | | | | | | | | | | |
| 0 | 1 | | | | | | | | | | | | | | | |
| AND (logisch produkt) | <table border="1"><thead><tr><th>X</th><th>Y</th><th>X AND Y</th></tr></thead><tbody><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td></tr></tbody></table> | X | Y | X AND Y | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| X | Y | X AND Y | | | | | | | | | | | | | | |
| 1 | 1 | 1 | | | | | | | | | | | | | | |
| 1 | 0 | 0 | | | | | | | | | | | | | | |
| 0 | 1 | 0 | | | | | | | | | | | | | | |
| 0 | 0 | 0 | | | | | | | | | | | | | | |
| OR (logische som) | <table border="1"><thead><tr><th>X</th><th>Y</th><th>X OR Y</th></tr></thead><tbody><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr></tbody></table> | X | Y | X OR Y | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| X | Y | X OR Y | | | | | | | | | | | | | | |
| 1 | 1 | 1 | | | | | | | | | | | | | | |
| 1 | 0 | 1 | | | | | | | | | | | | | | |
| 0 | 1 | 1 | | | | | | | | | | | | | | |
| 0 | 0 | 0 | | | | | | | | | | | | | | |
| XOR (exklusieve OF-bewerking) | <table border="1"><thead><tr><th>X</th><th>Y</th><th>X XOR Y</th></tr></thead><tbody><tr><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr></tbody></table> | X | Y | X XOR Y | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| X | Y | X XOR Y | | | | | | | | | | | | | | |
| 1 | 1 | 0 | | | | | | | | | | | | | | |
| 1 | 0 | 1 | | | | | | | | | | | | | | |
| 0 | 1 | 1 | | | | | | | | | | | | | | |
| 0 | 0 | 0 | | | | | | | | | | | | | | |
| EQV (logische ontkenning exklusieve OF-bewerking) | <table border="1"><thead><tr><th>X</th><th>Y</th><th>X EQV Y</th></tr></thead><tbody><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td></tr></tbody></table> | X | Y | X EQV Y | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| X | Y | X EQV Y | | | | | | | | | | | | | | |
| 1 | 1 | 1 | | | | | | | | | | | | | | |
| 1 | 0 | 0 | | | | | | | | | | | | | | |
| 0 | 1 | 0 | | | | | | | | | | | | | | |
| 0 | 0 | 1 | | | | | | | | | | | | | | |
| IMP (logische implikatie) | <table border="1"><thead><tr><th>X</th><th>Y</th><th>X IMP Y</th></tr></thead><tbody><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td></tr></tbody></table> | X | Y | X IMP Y | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| X | Y | X IMP Y | | | | | | | | | | | | | | |
| 1 | 1 | 1 | | | | | | | | | | | | | | |
| 1 | 0 | 0 | | | | | | | | | | | | | | |
| 0 | 1 | 1 | | | | | | | | | | | | | | |
| 0 | 0 | 1 | | | | | | | | | | | | | | |

UITDRUKKINGEN MET LETTERTEKENRIJEN

De enige bewerking die kan worden uitgevoerd met rij-konstanten, -variabelen en -functies is "optellen", ofwel samenvoegen van de rijen lettertekens.

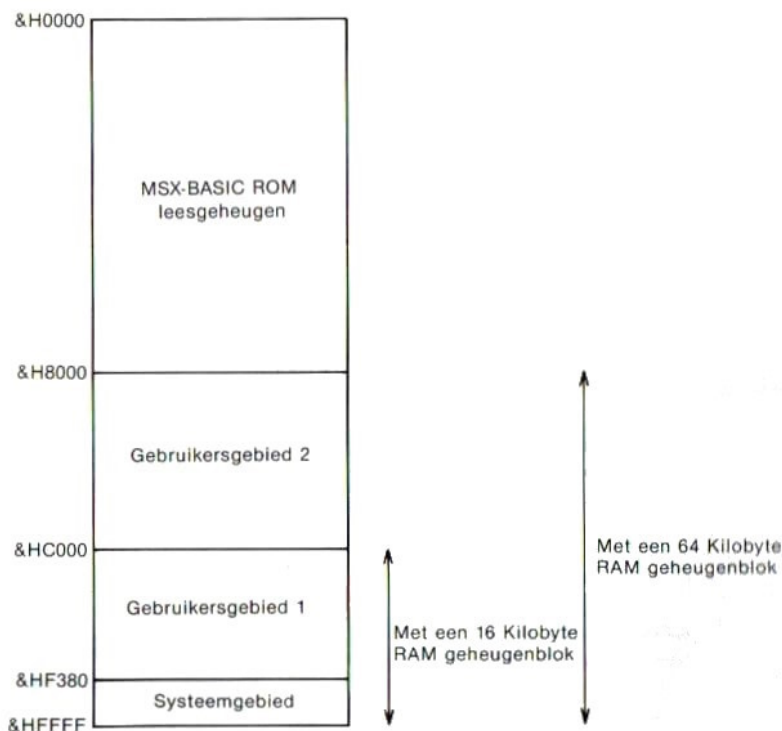
Voorbeeld

```
PRINT "ABC"+"XYZ"  
ABCXYZ
```

4. KLEURCODES

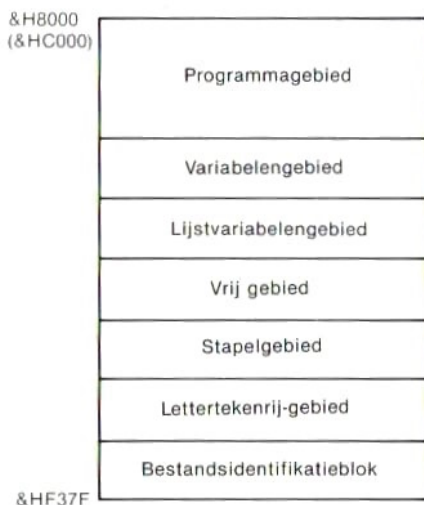
| Code | Kleur | Code | Kleur | Code | Kleur | Code | Kleur |
|------|-------------|------|-------------|------|------------|------|-------------|
| 0 | Transparant | 4 | Donkerblauw | 8 | Middelrood | 12 | Donkergroen |
| 1 | Zwart | 5 | Lichtblauw | 9 | Lichtrood | 13 | Magenta |
| 2 | Middelgroen | 6 | Donkerrood | 10 | Donkergeel | 14 | Grijs |
| 3 | Lichtgroen | 7 | Hemelsblauw | 11 | Lichtgeel | 15 | Wit |

5. GEHEUGENKAART



Zie voor de capaciteit van het RAM geheugenblok de gebruiksaanwijzing hiervan.

KONFIGURATIE GEBRUIKERSGEBIED



Programmagebied

Hier wordt een programma met regelnummers opgeslagen.

Variabelengebied

Voor het opslaan van numerieke gegevens en verwijzadressen van lettertekenrij-gegevens.

Lijstvariabelengebied

Voor het opslaan van gegevens in lijstvariabelen. Bij lijstvariabelen met lettertekens wordt hier het verwijzadres naar het lettertekenrij-gebied opgeslagen.

Vrij gebied

Ongebruikt gebied. Hoeveel gebied beschikbaar is kan gecontroleerd worden met de FRE functie.

Stapelgebied

In het stapelgebied worden terugkeeradressen opgeslagen.

Lettertekenrij-gebied

Voor het opslaan van rijen lettertekens die zijn toegekend aan rij-variabelen of -lijstvariabelen. Hoeveel gebied hiervoor beschikbaar is kan worden bepaald met een CLEAR bevel.

Bestandsidentifikatieblok

Wordt gebruikt tijdens in- en uitvoer van bestanden.

6. I/O POORT TOEWIJZING

| Gebruikt met | Nummer I/O poort | Toepassingen |
|---------------|------------------|---|
| RS-232-C | &H80 | Uitlezen/inschrijven van gegevens |
| | &H81 | Modus (bij inschrijven) Status (bij uitlezen) |
| Afdrukeenheid | &H90 | Signaalpoorten (bij inschrijven) Status (bij uitlezen) |
| | &H91 | Inschrijven van gegevens |
| VDP (video) | &H98 | Uitlezen/inschrijven van gegevens met video RAM geheugen. |
| | &H99 | Bevelen en adressen (bij inschrijven) Status (bij uitlezen) |
| PSG | &HA0 | Adres signaalbuffer (bij inschrijven) |
| | &HA1 | Inschrijven van gegevens |
| | &HA2 | Uitlezen van gegevens |
| PPI | &HA8 | Uitlezen/inschrijven van gegevens voor gebruik met poort A (keuze geheugenblok) |
| | &HA9 | Uitlezen/inschrijven van gegevens voor gebruik met poort B (lezen toetsenbord) |
| | &HAA | Uitlezen/inschrijven van gegevens voor gebruik met poort C (cassette) |
| | &HAB | Modus (bij inschrijven) |

● De I/O adressen van &H00 t/m &H7F worden niet gebruikt. De adressen in het bereik van &H80 t/m &HFF die hierboven niet genoemd zijn, worden door het systeem zelf gebruikt.

SONY®

MSX

**MSX-BASIC
OVERZICHTSKAART**

HIT BIT

*MSX is een handelsmerk van Microsoft Corp.

Printed in Japan

© 1984 by Sony Corporation

BEVELEN EN INSTRUKTIES

BEVELEN VOOR HET PROGRAMMEREN

| schrijfwijze | toepassing | voorbeeld |
|---|---|--------------------|
| AUTO [nummer beginregel] [, verhoging] | Automatisch genereren van regelnummers. | AUTO 100, 10 |
| DELETE [regelnummer] [-regelnummer] | Verwijderen van regels uit een programma. | DELETE 30-60 |
| LIST [nummer beginregel] [-] [nummer laatste regel] | Zet lijst van programmaregels op het scherm. | LIST |
| LLIST [nummer beginregel] [-] [nummer laatste regel] | Zet lijst van programmaregels op papier. | LLIST 100-200 |
| NEW | Programma wissen. | |
| RENUM [nieuw nummer beginregel], [oud nummer beginregel], [verhoging] | Regels hernummeren. | RENUM 100, 10, 10 |
| REM of ' | Opmerking invoegen. | REM---PROGRAM 1--- |
| KEY LIST | Geeft de functies van de funktietoetsen weer. | |

BEVELEN VOOR DEFINIËREN EN INSTELLEN

| schrijfwijze | toepassing | voorbeeld |
|--|---|-----------------------------|
| CLEAR [formaat gebied voor lettertekens] [, hoogste adres] | Geeft alle variabelen de beginwaarde en stelt de grens van het lettertekengebied en het geheugen in. | CLEAR 400, 55296 |
| DIM naam variabele (maximale waarde onder-index [maximale waarde onder-index] ...) | Instellen van de naam, soort en afmetingen van een lijstvariabele. | DIM A\$ (100) |
| DEF { INT } SNG } DBL } STR } lettertekens [-lettertekens] | Definiëren soort variabele aan de hand van de eerste letter van de naam. (INT: geheel getal, SNG: enkele precisie, DBL: dubbele precisie, STR: rij lettertekens) | DEFINT I-N |
| DEF FN funktienaam [(parameter)]=uitdrukking | Definiëren gebruikersfunctie. | DEF FNA (X)=A * X^2+B * X+C |
| ERASE [naam lijstvariabele] [, naam lijstvariabele] ... | Wissen lijstvariabelen. | ERASE A, B, C |
| KEY nummer funktietoets, rij lettertekens | Definiëren rijen voor funktietoetsen. | KEY 1, "LLIST"+CHR\$(13) |

BEVELEN VOOR IN- EN UITVOER VAN GEGEVENS

| schrijfwijze | toepassing | voorbeeld |
|--|--|------------------------------|
| DATA konstante [, konstante] [, konstante] ... | Invoeren van gegevens om te lezen met een READ bevel. | DATA 3, 4, 5, 6, ABC, "C, D" |
| INPUT ["invoeraanwijzing";] variabele [, variabele] [, variabele] ... | Invoeren van de waarde van een variabele via het toetsenbord. | INPUT "A\$=";A\$ |
| LINE INPUT ["invoeraanwijzing";] rij-variabele | Toewijzen van maximaal 254 lettertekens aan een rij-variabele via het toetsenbord. | LINE INPUT "C\$=";C\$ |
| [LET] variabele=uitdrukking | Gegevens toewijzen aan een variabele. | LET A=A+5 |
| MID\$(X\$, M[, N])=Y\$ | Vervangt lettertekens vanaf M-de letterteken van de rij X\$ door lettertekens vanaf begin tot en met N-de letterteken van rij Y\$. | MID\$(A\$, 2, 5)=B\$ |
| PRINT [uitdrukking] [scheidingstekens] [uitdrukking] [scheidingstekens] ... of ? [uitdrukking] [scheidingstekens] [uitdrukking] [scheidingstekens] ... | Zet gegevens op het scherm. Scheidingstekens zijn de komma (,), puntkomma (;) of een spatie. | PRINT A;B;C |

| | | |
|--|---|--|
| PRINT USING opmaaksymbool; schrijfwijze | Zet gegevens op het scherm in een gewenste opmaak. Opmaaksymbolen: "!" Geeft het eerste letterteken weer. "\n spaties \" Geeft n+2 lettertekens weer. "&" Geeft de gehele rij weer. "# " Geeft voor numerieke gegevens het aantal cijfers. "+ " Geeft + of - voor (na) numerieke gegevens. "- " Geeft - na negatieve numerieke gegevens. "* * " Vult de ruimte voor een getal op met *-jes. "££" Zet £ voor numerieke gegevens. "* * £" Zet £ voor numerieke gegevens en vult de ruimte ervoor op met *-jes. ", " Zet een komma tussen elke drie cijfers links van de decimale punt. "^^^" Geeft getalswaarden met drijvende decimale punt. | 10 A\$="ABCDEFGH" 20 PRINT USING "!";A\$ 30 PRINT USING "\ \ \";A\$ 40 PRINT USING "SS&TTT";A\$ PRINT USING "###.#";123.45,10.5 PRING USING "+###";100,-200 PRINT USING "###-";100,-200 PRINT USING "* * ###";100,-200 PRINT USING "££###";100,-200 PRING USING "* * £###";10,-20 PRINT USING "###.#.#";1234.56 PRINT USING "#.###^^^";123.98 |
| READ variabele [, variabele] [, variabele] ... | Lezen van de gegevens uit een DATA bevel. | READ A% |
| RESTORE [regelnummer] | Aangeven van de regel met het DATA bevel dat door het volgende READ bevel moet worden gelezen. | RESTORE100 |
| SWAP variabele, variabele | Uitwisselen van de waarde van twee variabelen. | SWAP A,B |

BEVELEN VOOR HET REGELEN VAN DE PROGRAMMAVERWERKING

| schrijfwijze | toepassing | voorbeeld |
|---|---|--|
| RUN [regelnummer] | Start de verwerking van een programma. | RUN 100 |
| STOP | Onderbreekt de verwerking van een programma. | |
| CONT | Herstart de verwerking van een programma. | |
| END | Beëindigt de verwerking van een programma. | |
| TRON | Geeft nummer verwerkte regel aan. | |
| TROFF | Annuleert TRON. | |
| FOR variabele=beginwaarde TO eindwaarde [STEP verhoging] NEXT [variabele] [, variabele] ... | Nogmaals verwerken van de programma-regels tussen FOR en NEXT. | FOR I=1 TO 10 STEP 2 NEXT I |
| GOSUB regelnummer RETURN [regelnummer] | Doorgaan met aangegeven subroutine. Bij RETURN terugkeren naar hoofdprogramma. | 100 GOSUB 100 1000 1100 RETURN |
| GOTO regelnummer | Doorgaan met de aangegeven regel. | GOTO 100 |
| IF uitdrukking { THEN { bevel regelnummer } GOTO regelnummer } { ELSE { bevel regelnummer } } | Verwerking afsplitsen afhankelijk van de waarde van een uitdrukking. | IF X=0 THEN 100 ELSE 200 |
| ON uitdrukking GOTO regelnummer [, regelnummer] ... | Verwerking afsplitsen afhankelijk van de waarde van een uitdrukking. | ON A GOTO 100, 200, 300 |
| ON uitdrukking GOSUB regelnummer [, regelnummer] ... | Verwerking afsplitsen afhankelijk van de waarde van een uitdrukking. | ON SGN (A)+2 GOSUB 1000, 2000, 3000 |

BEVELEN VOOR WEERGAVE OP HET SCHERM

| schrijfwijze | toepassing | voorbeeld |
|---|---|-------------------------------|
| SCREEN [modus], [formaat sprite], [intoetssignaal], [snelheid in baud], [soort afdrukeenheid] | Bepalen van de scherm-weergavekarakteristieken. Modus 0: 40 × 24 tekens tekstschermb 1: 32 × 24 tekens tekstschermb 2: gedetailleerd grafisch scherm 3: meerkleuren grafisch scherm Formaat sprite 0: 8 × 8 stippen zonder vergoting 1: 8 × 8 stippen met vergroting 2: 16 × 16 stippen zonder vergroting 3: 16 × 16 stippen met vergroting Intoetssignaal 0: Intoetssignaal onderdrukt 1: Intoetssignaal klinkt Snelheid in baud 0: 1200 baud 1: 2400 baud Soort afdrukeenheid 0: MSX afdrukeenheid 1: Andere dan MSX afdrukeenheid | SCREEN 2, 0,0 |
| WIDTH aantal lettertekens per regel op het scherm | Geeft het aantal lettertekens per regel op het tekstschermb. | WIDTH 28 |
| CLS | Gehele scherm wissen. | |
| LOCATE [x-coördinaat], [y-coördinaat], [cursor aan/uit] | Cursor verplaatsen. Cursor aan/uit 0: Cursor onzichtbaar 1: Cursor zichtbaar | LOCATE 10, 12, 1 |
| COLOR [kleur voorgrond], [kleur achtergrond], [kleur randgebieden] | Bepalen kleuren voor- en achtergrond en randgebieden. | COLOR 8, 15, 2 |
| PUT SPRITE nummer sprite-vlak [, [STEP] (x-coördinaat, y-coördinaat)], [kleurcode], [spritenummer] | Weergeven van een gekozen sprite-patroon op de gekozen plaats op een gekozen sprite-vlak. | PUT SPRITE 0, (100, 50), 7, 2 |
| CIRCLE [STEP] (x-coördinaat, y-coördinaat), straal, [kleurcode], [beginhoek], [eindhoek], [aanzichtshoek] | Cirkel tekenen. | CIRCLE (80, 60), 15, 8 |
| DRAW "grafische deelinstrukties" | Willekeurige grafische voorstelling tekenen. | DRAW "S40U5R5D5L5" |
| LINE [[STEP] (x-coördinaat, y-coördinaat)]-[STEP] (x-coördinaat, y-coördinaat), [kleurcode] { [B], [BF] } | Lijn of vierkant tekenen. | LINE -STEP (20, 50),, B |
| PAINT [STEP] (x-coördinaat, y-coördinaat), [kleurcode vlak], [kleurcode randlijn] | Kleuren van een gebied binnen een randlijn. | PAINT (120, 100) |
| PSET [STEP] (x-coördinaat, y-coördinaat) [, kleurcode] | Stip zetten. | PSET STEP (10, 10), 14 |
| PRESET [STEP] (x-coördinaat, y-coördinaat) [, kleurcode] | Stip zetten of wissen. | PRESET (100, 100) |
| KEY { ON OFF } | Funkties van de funktietoetsen weergeven of wissen. | KEY OFF |

Grafische deelinstrukties (Met B wijzigt een deelinstrukctie alleen het beginpunt zonder een lijn te trekken. Met N trekt de deelinstrukctie een lijn maar wijzigt niet het beginpunt.)

| deelinstrukctie | toepassing | beginwaarde | deelinstrukctie | toepassing | beginwaarde |
|------------------|--|-------------|------------------|--|-------------|
| Mx, y | Naar een absolute positie (x, y) | | Fn | Naar rechtsonder | n=1 |
| $M \pm x, \pm y$ | Verplaats over afstand $\pm x, \pm y$ vanuit de huidige positie. | | Gn | Naar linksonder | n=1 |
| Un | Naar boven | n=1 | Hn | Naar linksboven | n=1 |
| Dn | Naar beneden | n=1 | An | Draai het coördinatenstelsel | |
| Rn | Naar rechts | n=1 | Cn | Kies een kleur | n=15 |
| Ln | Naar links | n=1 | Sn | Kies het aantal stippen per eenheid. | n=4 |
| En | Naar rechtsboven | n=1 | X rij-variabele; | Voer het deelbevel toegekend aan de rij-variabele uit. | |

BEVELEN VOOR HET SPELEN VAN MUZIEK

| schrijfwijze | toepassing | voorbeeld |
|--|---------------------------------------|--------------------|
| BEEP | Laat een pieptoon klinken. | BEEP: BEEP: BEEP |
| SOUND PSG registernummer, uitdrukking | Schrijf gegevens in het PSG register. | SOUND 7, 7 |
| PLAY "muziek deelinstrukcties" [, "muziek deelinstrukcties"] [, "muziek deelinstrukcties"] | Muziek spelen. | PLAY "O4L4CEGEL1C" |

Muziek deelinstrukcties

| deelinstrukctie | toepassing en bereik | beginwaarde | deelinstrukctie | toepassing en bereik | beginwaarde |
|--|--|-------------|-----------------|---|-------------|
| $A \begin{bmatrix} \# \\ + \\ - \end{bmatrix} -G \begin{bmatrix} \# \\ + \\ - \end{bmatrix}$ | Muzieknoten | | Tn | Tempo $32 \leq n \leq 255$ | n=120 |
| On | Oktaaf $1 \leq n \leq 8$ | n=4 | Vn | Volume $0 \leq n \leq 15$ | n=8 |
| Nn | Toonhoogte $0 \leq n \leq 96$ | | Mn | Frekwentie omhullende $1 \leq n \leq 65535$ | n=255 |
| Ln | Lengte $1 \leq n \leq 64$ | n=4 | Sn | Patroon omhullende $1 \leq n \leq 15$ | n=1 |
| Rn | Rust $1 \leq n \leq 64$ | n=4 | | Stip | |
| X rij-variabele; | Voer het deelbevel toegekend aan de rij-variabele uit. | | | | |

BEVELEN VOOR PROGRAMMA- EN DATA-BESTANDEN

| schrijfwijze | toepassing | voorbeeld |
|---|---|------------------------------------|
| MAXFILES=uitdrukking | Instellen van het aantal bestanden dat in een programma geopend kan worden. | MAXFILES=3 |
| OPEN "apparaatnaam [bestandsnaam]" [FOR modus] AS [#] bestandsnummer | Openen van een bestand en kiezen van een modus. Modus: OUTPUT Schrijven INPUT Lezen | OPEN "CRT : TEST" FOR OUTPUT AS #1 |
| PRINT # bestandsnummer, uitdrukking | Gegevens op volgorde in een bestand schrijven. | PRINT #1, "ABC" |
| PRINT # bestandsnummer, USING opmaaksymbool; uitdrukking | Gegevens op volgorde met een gekozen opmaak in een bestand schrijven. (Zie PRINT USING.) | PRINT #1, USING "\ \";A\$ |
| INPUT # bestandsnummer, variabele [, variabele] ... | Gegevens op volgorde uit een bestand lezen en aan variabelen toewijzen. | INPUT #1, A, B, C |
| LINE INPUT # bestandsnummer, rij-variabele | Rij van maximaal 254 lettertekens uit een bestand lezen en aan een variabele toewijzen. | LINE INPUT #1, A\$ |
| CLOSE [#] [bestandsnummer] [, bestandsnummer] ... | Bestand(en) sluiten. | CLOSE #1, 2 |
| SAVE "apparaatnaam [bestandsnaam]" | Programma opslaan. | SAVE "CAS:PROGRAM" |
| LOAD "apparaatnaam [bestandsnaam]" | Programma laden. | LOAD "CAS:PROGRAM" |
| MERGE "apparaatnaam [bestandsnaam]" | Programma in ASCII code laden en samenvoegen met een programma in het geheugen. | MERGE "CAS:PROG2" |
| BSAVE "apparaatnaam [bestandsnaam]", beginadres, eindadres [, beginadres verwerking] | Geheugeninhoud binnen het aangegeven adres opslaan. | BSAVE "CAS:GAME", &H3000, &H3FFF |
| BLOAD "apparaatnaam [bestandsnaam]" [, R] [, verschuiving] | Programma in machinetaal laden. Met ,R programma vervolgens uitvoeren. De verschuiving geldt voor het geheugenadres bij het laden. | BLOAD "CAS:GAME", R |
| CSAVE "bestandsnaam" [, snelheid in baud] | Programma op cassette opslaan. Snelheid in baud: 1 1200 baud 2 2400 baud | CSAVE "STAR" |
| CLOAD ["bestandsnaam"] | Programma vanaf cassette laden. | CLOAD "STAR" |
| CLOAD? ["bestandsnaam"] | Programma op cassette met programma in geheugen vergelijken. | CLOAD? "STAR" |

Apparaatnaam

CAS: cassetterecorder
 CRT: tekstscherf
 GRP: grafisch scherm
 LPT: afdrukeenheid
 CAT: data cartridge

BEVELEN VOOR ONDERBREKING

| schrijfwijze | toepassing | voorbeeld |
|---|---|-------------------------------|
| ON KEY GOSUB regelnummer, regelnummer ... | Onderbreken met een funktietoets. | ON KEY GOSUB 1000, 2000, 3000 |
| KEY (nummer funktietoets) ON | Onderbreking met een funktietoets geldig maken. | KEY (1) ON |
| KEY (nummer funktietoets) OFF | Onderbreking met een funktietoets ongeldig maken. | KEY (2) OFF |
| KEY (nummer funktietoets) STOP | Onderbreking met een funktietoets vasthouden. | KEY (3) STOP |
| ON STRIG GOSUB regelnummer, regelnummer ... | Onderbreken met de trekkerknop van een joystick. | ON STRIG GOSUB 1000,, 2000 |
| STRIG (nummer joystick) ON | Onderbreking met een joystick geldig maken. Nummer joystick: 0 spatiebalk 1 joystick 1 2 joystick 2 | STRIG (1) ON |
| STRIG (nummer joystick) OFF | Onderbreking met een joystick ongeldig maken. | STRIG (2) OFF |
| STRIG (nummer joystick) STOP | Onderbreking met een joystick vasthouden. | STRIG (0) STOP |
| ON STOP GOSUB regelnummer | Onderbreken met de CTRL en STOP toetsen. | ON STOP GOSUB 1000 |
| STOP ON | Onderbreking met de CTRL en STOP toetsen geldig maken. | |
| STOP OFF | Onderbreking met de CTRL en STOP toetsen ongeldig maken. | |
| STOP STOP | Onderbreking met de CTRL en STOP toetsen vasthouden. | |
| ON SPRITE GOSUB regelnummer | Onderbreken met een sprite-overlapping. | ON SPRITE GOSUB 1000 |
| SPRITE ON | Onderbreking met een sprite-overlapping geldig maken. | |
| SPRITE OFF | Onderbreking met een sprite-overlapping ongeldig maken. | |
| SPRITE STOP | Onderbreking met een sprite-overlapping vasthouden. | |
| ON INTERVAL=periode GOSUB regelnummer | Onderbreken na een bepaalde periode. De tijd tussen de onderbrekingen is de periode (het interval) × 1/50 seconde. | ON INTERVAL=100 GOSUB 1000 |
| INTERVAL ON | Onderbreking na een periode geldig maken. | |
| INTERVAL OFF | Onderbreking na een periode ongeldig maken. | |
| INTERVAL STOP | Onderbreking na een periode vasthouden. | |

BEVELEN VOOR AANGESLOTEN APPARATUUR

| schrijfwijze | toepassing | voorbeeld |
|--|---|--------------------------|
| LPRINT [uitdrukking] [scheidingsteken] [uitdrukking] [scheidingsteken] ... | Gegevens op papier zetten. | LPRINT A, B, C |
| LPRINT USING opmaaksymbool; uitdrukking | Gegevens met een gekozen opmaak op papier afdrukken. (Zie PRINT USING.) | LPRINT USING "###"; A, B |
| MOTOR [{ ON } { OFF }] | In- en uitschakelen van de motor van de cassette recorder. | MOTOR OFF |

BEVELEN VOOR HET VERWERKEN VAN FOUTEN

| schrijfwijze | toepassing | voorbeeld |
|---|---|--------------------------------------|
| ERROR foutcode | Geeft een fout die behoort bij de genoemde foutcode. | ERROR 3 IF A > 100 THEN ERROR 250 |
| ON ERROR GOTO regelnummer | Doorgaan met verwerking bij de aangegeven regel met de fout. | ON ERROR GOTO 1000 |
| RESUME [{ 0 } NEXT regelnummer] | Doorgaan met verwerking van het hoofdprogramma na verwerking van de foutenherstelroutine. | RESUME 10 |

BEVELEN VOOR SUBROUTINES IN MACHINETAAL

| schrijfwijze | toepassing | voorbeeld |
|---------------------------|---|-------------------|
| DEFUSR [getal]=beginadres | Beginadres van een gebruiker-subroutine definiëren. | DEFUSR0=53248 |
| POKE adres, uitdrukking | Gegevens in het geheugen schrijven. | POKE &HA400, &HFF |

BEVELEN VOOR I/O POORTEN EN GEHEUGEN

| schrijfwijze | toepassing | voorbeeld |
|--|---|----------------|
| OUT nummer I/O poort, uitdrukking | Gegevens naar I/O poort uitvoeren. | OUT &H90, 3 |
| WAIT nummer I/O poort, uitdrukking [, uitdrukking] | Verwerking van programma vasthouden tot via I/O poort ingevoerde gegevens een bepaalde waarde bereiken. | WAIT &H90, 255 |
| VPOKE adres, uitdrukking | Gegevens naar video RAM geheugen uitvoeren. | VPOKE 263, 01 |

BEVELEN VOOR OPROEPEN BEVELENREEKSEN

| schrijfwijze | toepassing | voorbeeld |
|--|--|-----------|
| CALL naam subroutine of _naam subroutine CALL bevelenreeks [argument, argument ...] of _bevelenreeks [argument, argument ...] | Doorgaan met verwerking van subroutine in machinetaal, of met verwerking van een bevelenreeks vanuit het ROM geheugenblok. | CALL SUB |

FUNKTIES

NUMERIEKE FUNKTIES

| | |
|----------|--|
| ABS (X) | : Geeft een absolute waarde. |
| ATN (X) | : Geeft de boogtangens. |
| CDBL (X) | : Omzetten in dubbele-precisie waarde. |
| CINT (X) | : Omzetten in geheel getal. ($-32768 \leq X \leq 32767$) |
| COS (X) | : Geeft de cosinus van X radialen. |
| CSNG (X) | : Omzetten in enkele-precisie waarde. |
| ERL | : Geeft het nummer van een regel met een fout. |
| ERR | : Geeft het foutnummer. |
| EXP (X) | : Geeft e^x . |
| FIX (X) | : Geeft het geheel getal van X. |
| INT (X) | : Geeft het grootste gehele getal kleiner dan of gelijk aan X. |
| LOG (X) | : Geeft de natuurlijke logaritme. |
| RND (X) | : Geeft een willekeurig getal. |
| SGN (X) | : Geeft 1 als $X > 0$, 0 als $X = 0$ en -1 als $X < 0$. |
| SIN (X) | : Geeft de sinus van X radialen. |
| SQR (X) | : Geeft de vierkantswortel. |
| TAN (X) | : Geeft de tangens van X radialen. |

RIJ-FUNKTIES

| | |
|---------------------|--|
| LEFTS (X\$, N) | : Geeft N lettertekens vanaf de linkerkant van de X\$ rij. |
| MIDS (X\$, M [, N]) | : Geeft N lettertekens te beginnen met het M-de teken van links in de X\$ rij. |
| RIGHTS (X\$, N) | : Geeft N lettertekens vanaf de rechterkant van de X\$ rij. |
| SPACE\$ (N) | : Geeft N spaties. |
| STRING\$ (N, J) | : Geeft N lettertekens met de ASCII code J. |
| STRING\$ (N, X\$) | : Geeft N maal het eerste letterteken van de X\$ rij. |
| TAB (N) | : Verplaatst de cursor naar de N-de plaats. |
| SPC (N) | : Geeft N spaties. |

FUNKTIES VOOR HET OMZETTEN VAN GETALLEN IN RIJEN EN VV.

| | |
|-----------------------|--|
| ASC (X\$) | : Geeft de ASCII code voor het eerste letterteken van de X\$ rij. |
| BIN\$ (X) | : Geeft X als een binaire rij weer. ($-32768 \leq X \leq 65535$) |
| CHR\$ (X) | : Geeft het letterteken waarvan de ASCII code X is. |
| HEX\$ (X) | : Geeft X als een hexadecimale rij weer. ($-32768 \leq X \leq 65535$) |
| INSTR ([N,] X\$, Y\$) | : Geeft de plaats van de Y\$ rij na het N-de letterteken van de X\$ rij. |
| LEN (X\$) | : Geeft het aantal lettertekens waaruit de X\$ rij bestaat. |
| OCT\$ (X) | : Geeft X als een octale rij weer. ($-32768 \leq X \leq 65535$) |
| STR\$ (X) | : Omzetten in een rij-waarde. |
| VAL (X\$) | : Omzetten in een getalswaarde. |

OVERIGE FUNKTIES

| | |
|----------|--|
| PLAY (N) | : Controleren of er muziek gespeeld wordt. Als N gelijk is aan 1, 2 of 3 en er wordt muziek gespeeld dan wordt -1 gegeven; zo niet dan wordt 0 gegeven. Als N=0 dan wordt met de status van de deelinstructies voor muziek een OF-bewerking uitgevoerd en de uitkomst hiervan gegeven. |
|----------|--|

FUNKTIES VOOR DE INVOER VAN GEGEVENS

| | |
|---|---|
| Van het scherm | |
| CSRLIN | : Geeft de y-coördinaat van de cursor. |
| POS (X) | : Geeft de x-coördinaat van de cursor. |
| POINT (X, Y) | : Geeft de kleurcode van punt (X, Y). |
| Van een bestand met gegevens | |
| EOF (bestandsnummer) | : Geeft -1 als de laatste waarde van het bestand is gelezen; zo niet, dan wordt 0 gegeven. |
| INPUT\$ (N, [#] bestandsnummer) | : Voert N lettertekens vanuit een bestand in en geeft ze weer. |
| Van de afdrukeenheden | |
| LPOS (X) | : Geeft de positie van de drukkop in de afdrukbuffer. |
| Van het geheugen | |
| FRE (0) | : Geeft aan hoeveel geheugen beschikbaar is. |
| FRE (" ") | : Geeft aan hoeveel geheugen beschikbaar is voor rijen lettertekens. |
| PEEK (adres) | : Geeft de inhoud van een geheugenadres. |
| VARPTR (variabele) | : Geeft het beginadres van het gebied in het geheugen waar de variabele opgeslagen is. |
| VPEEK (adres) | : Geeft de inhoud van een adres van het video RAM geheugen. |
| Van het toetsenbord | |
| INKEY\$ | : Geeft het letterteken van de ingedrukte toets. |
| INPUTS (X) | : Voert X lettertekens via het toetsenbord in. |
| Van een I/O poort | |
| INP (nummer I/O poort) | : Voert gegevens in via een I/O poort. |
| Van een subroutine in machinetaal | |
| USER $\begin{cases} 0 \\ \text{tot} \\ 9 \end{cases}$ (X) | : Geeft de uitkomstwaarde van de gebruiker-subroutine. |
| Van een joystick, peddel of aanraakpaneel | |
| STICK (N) | : Geeft de richting van de joystick. (Bij N=0 die van een cursortoets) (Midden=0, Boven=1, Rechtsboven=2, Rechts=3, Rechtsonder=4, Onder=5, Linksonder=6, Links=7, Linksboven=8) |
| STRIG (N) | : Geeft -1 als de trekkerknop van een joystick is ingedrukt; zo niet, dan wordt 0 gegeven. (Bij N=0 geldt hetzelfde voor de spatiebalk.) |
| PDL (N) | : Voert gegevens van een peddel in. |
| PAD (N) | : Geeft de toestand van een aanraakpaneel. Als N=0 of 4, dan wordt -1 gegeven als het paneel wordt aangeraakt; zo niet, dan wordt 0 gegeven. Als N=1 of 5 wordt de x-coördinaat van de aangeraakte plaats gegeven. Als N=2 of 6 wordt de y-coördinaat van de aangeraakte plaats gegeven. Als N=3 of 7 wordt -1 gegeven als de schakelaar wordt ingedrukt; zo niet, dan wordt 0 gegeven. |

KONSTANTEN EN VARIABELEN

FOUTMELDINGEN

| | | |
|------------|--------------------------------------|---|
| Konstanten | Rij-konstanten | Rij van 0 tot 255 lettertekens (tussen aanhalingstekens) |
| | Gehele getallen | - 32768 tot + 32767 |
| | Getallen met drijvende decimale punt | Aantal cijfers van belang: 6 (enkele precisie) of 14 (dubbele precisie) Exponentieel gedeelte: - 64 tot + 63 |
| | Hexadecimale uitdrukking | Met voorvoegsel "&H" |
| | Octale uitdrukking | Met voorvoegsel "&O" of "&O" |
| | Binaire uitdrukking | Met voorvoegsel "&B" |

| | | |
|------------|-----------------|--|
| Variabelen | Naam variabele | De eerste twee lettertekens zijn van belang |
| | Definitietekens | Volgt direct na naam variabele % : Gehele getallen ! : Enkele precisie # : Dubbele precisie S : Rij-variabelen |

SPECIALE VARIABELEN

TIME : Voor vasthouden van een schakelklok-waarde. Kan herschreven worden.

SPRITE\$ (spritenummer) : Voor vasthouden van een spritepatroon.

[Voorbeeld] `SPRITE$(1)=CHR$(&H18)+CHR$(&H3C)+CHR$(&H7E)+CHR$(&HFF)+CHR$(&H18)+CHR$(&H18)+CHR$(&H18)+CHR$(&H18)`

● Speciale bevelen en functies voor de VDP videostuureenheid

BASE (uitdrukking) : Voor het lezen of schrijven van het basisadres van de VDP tabel.

VDP (getalswaarde) : Voor het lezen of schrijven van de inhoud van het VDP register.

- | | |
|-------------------------------|--|
| 1 NEXT without FOR | : NEXT bevel zonder bijbehorend FOR bevel. |
| 2 Syntax error | : Taalfout in het bevel. |
| 3 RETURN without GOSUB | : RETURN bevel zonder bijbehorend GOSUB bevel. |
| 4 Out of DATA | : Geen gegevens meer over om te lezen. |
| 5 Illegal function call | : Onjuistheid in functie of bevel. |
| 6 Overflow | : Waarde van gegevens te klein of te groot. |
| 7 Out of memory | : Geen geheugen meer beschikbaar. |
| 8 Undefined line number | : Er wordt verwezen naar een niet bestaande regel. |
| 9 Subscript out of range | : Onder-index van lijstvariabele valt buiten gedefinieerd bereik. |
| 10 Redimensioned array | : De lijstvariabele in het DIM bevel was al gebruikt. |
| 11 Division by zero | : Deling door nul. |
| 12 Illegal direct | : Dit bevel kan niet rechtstreeks gegeven worden. |
| 13 Type mismatch | : Niet passende gegevens. |
| 14 Out of string space | : Geen geheugen meer beschikbaar voor rijen lettertekens. |
| 15 String too long | : De rij lettertekens is te lang. |
| 16 String formula too complex | : De rij lettertekens is te ingewikkeld. |
| 17 Can't CONTINUE | : De verwerking van het programma kan niet hervat worden. |
| 18 Undefined user function | : Er wordt een functie gebruikt die niet met een DEF FN bevel gedefinieerd is. |
| 19 Device I/O error | : Er is iets mis met de aangesloten apparatuur. |
| 20 Verify error | : Er is verschil tussen het programma op cassette en dat in het geheugen. |
| 21 No RESUME | : ON ERROR bevel zonder bijbehorend RESUME bevel. |
| 22 RESUME without error | : RESUME bevel zonder bijbehorend ON ERROR bevel. |
| 23 Unprintable error | : Er is een fout opgetreden waarvoor geen foutnummer bestaat. |
| 24 Missing operand | : Een parameter ontbreekt. |
| 25 Line buffer overflow | : Het ingevoerde programma is te lang voor het buffergeheugen. |
| 51 Internal error | : Er is iets mis met de inhoud van het geheugen of de tekst. |
| 52 Bad file number | : Het bestandsnummer is niet juist. |
| 54 File already open | : Dit bestand was al geopend. |
| 55 Input past end | : De gegevens zijn allemaal al gelezen. |
| 56 Bad file name | : De gegeven bestandsnaam is onjuist. |
| 57 Direct statement in file | : Tijdens het laden van het bestand is een bevel rechtstreeks gegeven. |
| 59 Fine not OPEN | : Het bestand moet eerst geopend worden. |

KLEURCODE

| code | kleur | code | kleur |
|------|-------------|------|-------------|
| 0 | Transparant | 8 | Middelgroen |
| 1 | Zwart | 9 | Lichtrood |
| 2 | Middelgroen | 10 | Donkergeel |
| 3 | Lichtgroen | 11 | Lichtgeel |
| 4 | Donkerblauw | 12 | Donkergroen |
| 5 | Lichtblauw | 13 | Magenta |
| 6 | Donkerrood | 14 | Grijs |
| 7 | Hemelsblauw | 15 | Wit |

OPERATORS

| | | |
|------------------------|--|--|
| Rekenkundige operators | \wedge $-$ $*, /$ \backslash MOD $+, -$ | machtsverheffen teken veranderen vermenigvuldigen, delen delen met gehele getallen restwaarde na deling optellen, aftrekken (In volgorde van prioriteit) |
| Vergelijkingsoperators | $< > =$ | vergelijken |
| Logische operators | NOT AND OR XOR EQV IMP | logische ontkenning logisch produkt logische som exklusieve logische som ontkenning van eksklusieve logische som logische implikatie |

