

**MSX**

# Computers en printers

Aansluiten en gebruiken

Wessel Akkermans



**MSX**  
**Computers en Printers**



**MSX**

# **Computers en printers**

Aansluiten en gebruiken

Wessel Akkermans

**uitgeverij STARK-TEXEL**

postbus 302 - 1794 ZG Oosterend tel. 02223 - 661

CIP-GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Akkermans, W.

MSX computers en printers : aansluiten en gebruiken / W. Akkermans.

— Oosterend : Stark-Textel

ISBN 90-6398-405-7

SISO 365.2 UDC 681.327.11

Trefw.: printers / MSX (computer).

---

1e druk 1986

ISBN 90 6398 405 7

©by uitgeverij Stark-Textel, Oosterend Nh.

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

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

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

MSX is een handelsmerk van Microsoft

# Voorwoord

Zogoed als er voor bijna ieder merk computer een ander dialect van BASIC is, zo zijn er voor bijna ieder merk printer andere commando's. Soms hebben de printers wel dezelfde functies, doch gebruiken ze verschillende commando-codes, soms hebben ze niet dezelfde functies. Het gevolg is, dat het voor de meeste mensen moeilijk is, om met printers veel meer te doen, dan het afdrukken van een listing. Dit betekent echter, dat van alle mogelijkheden die de printer heeft, er slechts een paar worden gebruikt. Om u in staat te stellen meer mogelijkheden van de printer te gebruiken, heb ik dit boekje samengesteld.

Voor wie nog geen printer heeft, is hoofdstuk 1 opgenomen. Daarin kunt u lezen welke soorten printers er zijn, en wat hun voor en nadelen zijn. Deze informatie kunt u gebruiken bij het maken van een uiteindelijke keuze.

Een groot probleem, dat in mindere mate ook voor MSX-computers geldt, is het aansluiten van de printer op de computer. Alle MSX-computers hebben een Centronics-interface, waarop een printer, die ook met een Centronics-interface is uitgerust, probleemloos kan worden aangesloten. MSX2-computers hebben echter ook een seriele poort. Wilt u een printer, met een seriele poort, aansluiten op de seriele poort van een MSX2-computer, dan dient u wat meer achtergrond kennis te hebben van de serie-interface. Het aansluiten van de printer op de computer wordt in hoofdstuk 2 behandeld.

Is de printer eenmaal aangesloten, dan kunnen er gegevens van de computer naar de printer worden gestuurd. Om een print-programma met verstand van zaken te schrijven, is het echter belangrijk iets meer te weten van de interne werking van de printer. Vandaar, dat er in hoofdstuk 3 nog wat theorie wordt gegeven, voordat in hoofdstuk 4 met het eigenlijke programmeren wordt begonnen. Na de eerste programmeer-

stappen in hoofdstuk 4 wordt in hoofdstuk 5 dieper op het programmeren ingegaan. Na het bestuderen van hoofdstuk 5 zult u praktisch ieder gewenst print-programma voor iedere gewenste printer kunnen schrijven.

Er zijn echter nog twee belangrijke onderwerpen, die moeilijk genoeg zijn, om er aparte hoofdstukken aan te wijden. Dat zijn de grafische mode van de printer en het definiëren van een eigen tekenset. Grafisch werk is op praktisch iedere printer mogelijk, zelfs op daisy wheel printers, zoals u in hoofdstuk 6 zult zien. Het definiëren van een eigen tekenset is slechts op een beperkt aantal printers mogelijk. In hoofdstuk 7 wordt het definiëren en gebruiken van een eigen tekenset voor een Epson printer behandeld.

Het boek wordt afgesloten met een appendix, waarin u een vertaaltabel aantreft, waarmee u programma's, die zijn geschreven voor een andere printer, kunt aanpassen aan uw eigen printer.

Voor het testen van de verschillende mogelijkheden van de printers heb ik van vele verschillende printers gebruik moeten maken. Elke dag leerde ik er weer iets nieuws bij, soms noodgedwongen door schade en schande, daar de bij de printer geleverde documentatie niet voldoende of niet helemaal juiste informatie bevatte. Zowel het experimenteren met de printers, als het op een praktische (hopelijk duidelijke) manier beschrijven van de mogelijkheden en het aansturen daarvan door middel van een MSX-computer, waren een plezierige bezigheid. Ik hoop dat u bij het lezen van dit boek, en het proberen van de gegeven programma's evenveel plezier zult hebben.

W. Akkermans

Februari 1986



# Inhoudsopgave

<b>1</b>	<b>Introductie</b>	<b>9</b>
1.1	Soorten printers	11
1.2	Impact printers	12
1.2.1	De matrix printer	14
1.2.2	De daisy wheel printer	17
1.3	Non impact printers	19
1.3.1	De ink jet printer	19
1.3.2	De elektrostatische printer	21
1.3.3	De thermische printers	21
1.4	Opties	23
1.5	Conclusie	24
<b>2</b>	<b>Aansluiten aan de computer</b>	<b>26</b>
2.1	De RS232C-interface	26
2.1.1	Transmissiecode	28
2.1.2	En als er nu toch iets fout gaat	30
2.1.3	Het ready/busy protocol	31
2.1.4	Het Xon/Xoff protocol	36
2.2	Centronics parallel interface	39
2.2.1	Het Centronics protocol	40
2.2.2	De fysieke verbinding	41
<b>3</b>	<b>Hoe komt een teken van computer tot papier?</b>	<b>44</b>
3.1	Codes en tekens	44
3.2	Wat doet de printer met de codes?	52
3.3	Hoe worden de tekens afgedrukt?	54
<b>4</b>	<b>Het programmeren van printers</b>	<b>59</b>
4.1	Carriage Return en Line Feed	59
4.2	Horizontale tabulatiestoppen	62

4.3	Normale, smalle en brede tekens	65
4.4	Diverse printer commando's	67
<b>5</b>	<b>Programmeren met ESC-reeksen</b>	<b>68</b>
5.1	Geprogrammeerde tabulatiestoppen	69
5.2	De MSX-printer standaard	72
5.3	Programma's en files afdrukken	76
5.4	Verschillende print-modes	83
<b>6</b>	<b>Grafische print modes</b>	<b>91</b>
6.1	Hard copy in SCREEN 0 en 1 modes	92
6.2	Grafische schermen naar Daisy Wheel printer	94
6.3	Grafische mode van matrix printers	100
6.4	Kleurenafdruk in grafische mode	114
<b>7</b>	<b>Down loading van tekens</b>	<b>123</b>
7.1	De theorie	123
7.2	Een praktisch voorbeeld	125
7.3	Een down load character editor	131
<b>Appendix A</b>	<b>MSX printer standaard</b>	<b>138</b>
<b>Appendix B</b>	<b>Vertaaltabellen printer commando's</b>	<b>142</b>
	<b>Alfabetische trefwoordenlijst</b>	<b>150</b>

# 1 Introductie

In dit inleidende hoofdstuk zal een kort overzicht van de verschillende soorten printers worden gegeven. Het indelen van printers in soorten is een moeilijke zaak, het hangt namelijk geheel af van de gebruikte criteria. Wij zullen er echter vanuit gaan, dat er "impact" en "non-impact" printers zijn en beide soorten zullen verder worden onderverdeeld in een aantal typen. Van alle bestaande soorten en typen printers zijn er slechts twee, die op dit moment voor de hobbyist betaalbaar zijn. In dit boek zullen we dan ook alleen deze twee typen verder behandelen.

De meeste printers worden alleen in een basisuitvoering geleverd. Meestal is het mogelijk om de basisuitvoering van een printer uit te breiden met een aantal opties. Ook die opties zullen in dit inleidende hoofdstuk de revue passeren.

Nadat we een aantal basisbegrippen hebben vastgelegd, zullen we verder ingaan op de details. We beginnen dan bij het aansluiten van de printer op de computer. De meeste fabrikanten, van zowel printers als computers, gebruiken een gestandaardiseerde aansluiting. Computers met een specifieke printer-interface zullen meestal zijn uitgerust met een "Centronics"-interface. Computers, die geen specifieke printer-interface hebben, zijn meestal uitgerust met een RS232C-interface. Dit is een serie-interface, die voor zowel datacommunicatiedoeleinden als voor het aansluiten van de meeste printers kan worden gebruikt. Voorwaarde is dan wel, dat de printer ook van een RS232C-interface is voorzien.

Na het aansluiten van de printer, zult u ontdekken dat er in de printer een aantal schakelaartjes zitten. Afhankelijk van de computer, waaraan u de printer wilt gebruiken, en afhankelijk van de toepassing van de printer, zullen die schakelaartjes in een bepaalde stand moeten worden gezet. Vaak hebben die schakelaartjes een directe relatie met de gebruikte soft-

ware. Om de schakelaars in de goede stand te kunnen zetten, zult u minstens moeten weten wat de functie ervan is.

Nu staat de printer geheel klaar om de door u gewenste informatie af te drukken. U zou echter tot de ontdekking kunnen komen, dat de printer bepaalde tekens, die wel op het beeldscherm worden afgedrukt, niet op papier kan zetten. Dit heeft dan te maken met de teken-set (character-set) van de printer en die van de computer. Die hoeven namelijk niet altijd dezelfde te zijn. Alleen wanneer u een MSX-computer hebt gekocht, en daarbij een MSX-printer, zullen beide dezelfde tekens hebben.

Voordat we dan kunnen overgaan tot het werkelijk toepassen van de printer (we willen de printer voor meer zaken gebruiken dan alleen voor het maken van een listing), is het goed iets meer te weten van de printer-buffers. Sommige printers hebben hele grote buffers, andere hebben slechts een klein buffertje, dat net groot genoeg is voor het opslaan van een regel.

Nu begint het echte programmeren. We zullen beginnen met eenvoudige voorbeelden. Deze voorbeelden zullen met praktisch alle printers mogelijk zijn. De opgenomen voorbeelden zijn in MSX-BASIC geschreven en getest op verschillende printers. In de appendix is een tabel opgenomen met daarin de software-commando's en instellingen voor een aantal printers. U zult zien dat niet alle printers dezelfde mogelijkheden hebben. Bij printers, die wel dezelfde mogelijkheden hebben, moeten die mogelijkheden vaak door middel van andere commando's worden aangestuurd.

Een speciale mogelijkheid, die bij de meeste matrix-printers aanwezig is, is de grafische mode. Daar dit onderwerp tamelijk gecompliceerd is, is daaraan een speciaal hoofdstuk gewijd. In dat hoofdstuk zal een programma, waarmee een "hard-copy" van het grafisch beeldscherm van de MSX-computer kan worden gemaakt, worden behandeld.

Een andere speciale mogelijkheid, die bij lang niet alle prin-

ters aanwezig is, is het zogenaamde "down loading" van tekensets. Dit wil zeggen, dat de printer een RAM-geheugen heeft, waarin het de vorm van de letters, cijfers en leestekens kan bewaren. Dit RAM-geheugen kan vanuit de computer worden geladen, met een tekenset, die door onszelf is bedacht. Ook dit onderwerp is nogal gecompliceerd, zodat ook daaraan een speciaal hoofdstuk is gewijd.

Hoewel alle hoofdstukken, die daarvoor in aanmerking komen, van zoveel mogelijk programmavoorbeelden zijn voorzien, vindt u vooral in de laatste twee hoofdstukken gebruiksklare programma's. De programma's uit die hoofdstukken (6 en 7) dienen echter tevens als voorbeeld van het programmeren van printers.

## **1.1 Soorten printers.**

Zoals we, aan het begin van dit hoofdstuk, al hebben vastgesteld, ligt het helemaal aan de criteria die we gebruiken, wat we onder een soort van printer verstaan. Je zou kunnen zeggen dat een matrix-printer (een printer waarbij de tekens worden opgebouwd uit een groot aantal puntjes) een andere soort printer is dan een daisy wheel printer (een printer waarbij het teken in een keer in zijn geheel op het papier wordt afgedrukt, net als bij een normale typemachine). Met hetzelfde gemak kun je echter ook stellen dat deze beide printers van dezelfde soort zijn, namelijk van het soort "impact" printers. Impact printers zijn printers waarbij de leestekens door middel van mechanische kracht op het papier worden gedrukt. Dit soort printers maakt over het algemeen veel lawaai, hoewel er natuurlijk onderscheid is.

Als impact printers een soort is, dan is de andere soort die groep printers, die de tekens zonder mechanische kracht op het papier zet. In die groep treffen we onder andere de ink jet printers, de electrostatische- en thermische matrix printers en de (dure) laser printers. Dit laatste type printer valt voor verdere behandeling in dit boek af, daar de prijs voor de hob-

byist nog te hoog is. Dit is erg jammer, omdat dit soort printer een afdrukkwaliteit levert, die kan wedijveren met professioneel drukwerk. Ze maken namelijk gebruik van een matrix van 300 tot 400 puntjes per inch.

Wij zullen er nu vanuit gaan, dat we met de volgende soorten printers te maken kunnen hebben:

- impact printers.
- non impact printers.

Deze soorten zullen we nader bekijken. We zullen zien wat hun voordelen en hun nadelen zijn, en voor welke toepassing we een bepaalde printer het beste kunnen gebruiken.

## **1.2 Impact printers.**

Van de vele soorten impact printers die er zijn, zullen we hier alleen de twee meest gangbare behandelen. Dit zijn de daisy wheel printer, ook wel aangeduid met margrietwielprinter, en de matrix printer. Voor beide printers geldt, dat zij de letters door middel van mechanische kracht via een inktlint op het papier drukken. Een gevolg van dit mechanisme is, dat dit soort printers een behoorlijke hoeveelheid lawaai produceert. Weliswaar zorgen de ontwikkelingen van de laatste jaren er voor dat deze printers hoe langer hoe minder lawaai maken, maar toch is het geluidsniveau voor een huiskamer nog erg hoog.

Het belangrijkste verschil tussen beide typen impact printers zit hem in de kwaliteit van de letters. De daisy wheel printer is in staat om letters van drukkwaliteit te leveren, afhankelijk van het gekozen letterwiel. De matrix printer maakt letters, die uit een matrix van puntjes bestaan. De meeste moderne matrix printers gebruiken echter zoveel puntjes om een letter mee op te bouwen, dat de puntjes bijna niet meer zijn te onderscheiden. Dit wordt dan aangeduid met NLQ (Near Letter Quality), ofwel een kwaliteit, die de kwaliteit van daisy wheel letters benadert.

Dit boek is gedrukt op een daisy wheel printer en is dus een voorbeeld van de te bereiken kwaliteit van een daisy wheel printer. Daarbij is gebruik gemaakt van de mogelijkheid om proportioneel af te drukken. Dit wil zeggen, dat er voor iedere letter niet meer ruimte wordt gebruikt dan nodig is. Zo zal de ruimte voor de letter i aanzienlijk kleiner zijn dan de ruimte voor de letter m. Dit is duidelijk in de volgende regels te zien:

iiiiiiiiii  
mmmmmmmmmm

Wanneer van de mogelijkheid tot proportioneel afdrucken geen gebruik wordt gemaakt, zal de ruimte voor iedere letter even groot zijn. Dus zal de letter i evenveel ruimte innemen als de letter m. In dat laatste geval kan een regellengte worden aangegeven met het aantal tekens per regel. Bij proportioneel afdrucken is het aantal tekens per regel afhankelijk van de gebruikte tekens. In dat geval kan de regellengte alleen worden aangegeven met centimeters of inches.

Dit is een voorbeeld van een standaard matrix letter.  
Dit is lettertype ELITE, op dezelfde printer.  
*Het is ook mogelijk om de letters schuin te drukken.*  
**Dit is twee keer zo breed.**  
En dit is smal, hetgeen ook wel CONDENSED wordt genoemd.

Dit is een voorbeeld van een NLQ-mode matrix letter.  
Dit is lettertype ELITE, op dezelfde printer.  
*Het is ook mogelijk om de letters schuin te drukken.*  
**Dit is twee keer zo breed.**  
En dit is smal, hetgeen ook wel CONDENSED wordt genoemd.

Afb. 1-1 Afdrukvoorbeelden van een matrix printer.

Een voorbeeld van de afdruk van een matrix printer vindt u in afbeelding 1-1. In dat voorbeeld ziet u eerst de normale letters (met weinig puntjes), daarna wordt de NLQ-mode van diezelfde printer getoond. Deze voorbeelden zijn gemaakt op een printer van nog geen 1000 gulden. De snelheid is 80

tekens per seconde voor de grove matrix letters en 40 tekens per seconde voor de NLQ-mode. In de NLQ-mode wordt iedere regel namelijk twee keer afgedrukt, waarbij de puntjes in de tweede slag iets verschoven zijn ten opzichte van de eerste.

Voor wat betreft de snelheid van daisy wheel printers, kunt u er van uitgaan, dat dit soort printers snelheden haalt die tussen de 12 en 50 tekens per seconde liggen. De langzaamste printers liggen in de prijsklasse van 1500 gulden en de snelste printers komen op 5000 gulden of meer.

Na dit inleidende verhaaltje zullen we de principes van de beide typen impact printers eens nader bekijken. We beginnen met de matrix printer en behandelen daarna de daisy wheel printer.

### **1.2.1 De matrix printer.**

Met "de matrix printer" bedoelen we de impact printer, die de tekens met behulp van matrices van puntjes opbouwt. Ook onder de non impact printers vinden we matrix printers, doch die printers worden in het algemeen aangeduid met een naam, die betrekking heeft op het principe van afdrukken op papier (thermisch, elektrostatisch of ink jet).

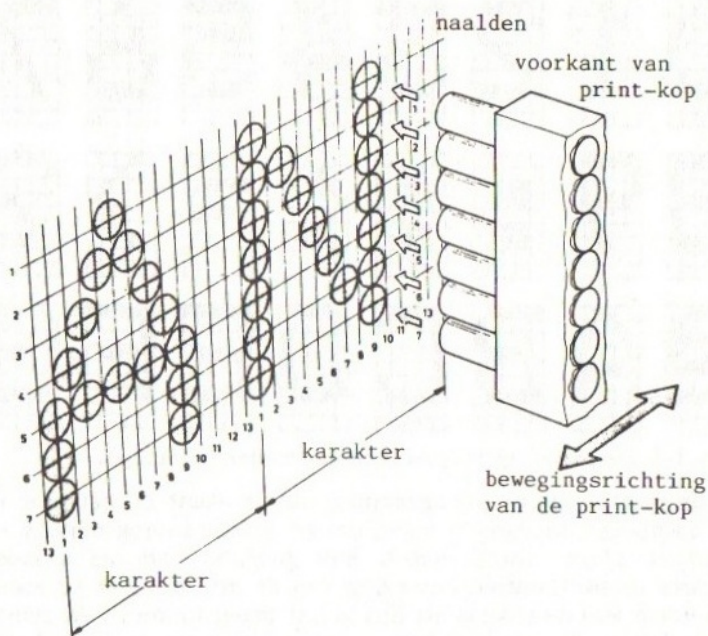
Om een letter, cijfer of leesteken op papier te krijgen, dient de printer over minstens de volgende onderdelen te beschikken:

- Een papieropvoermechanisme (verticale bewegingen).
- Een drukkoptransportmechanisme (horizontale bewegingen).
- Een inktlintransportmechanisme.
- Naalden, waarmee het inklint tegen het papier kan worden gedrukt.

Met het papieropvoermechanisme en het drukkoptransportmechanisme kan de plaats van een teken op het papier worden bepaald. Of er een teken wordt afgedrukt, en hoe dat teken



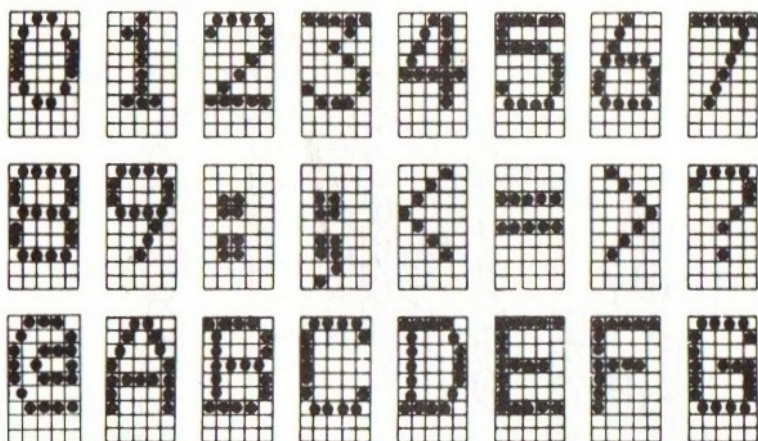
eruit zal zien, wordt bepaald door het al of niet afvuren van 1 of meer naalden. Afbeelding 1-2 laat de verschillende delen van de printer zien en geeft de functie van ieder deel aan.



Afb. 1-2 Het principe van de matrix printer.

De printkop bestaat uit een verticale rij naaldjes, die door middel van elektromagnetische kracht naar voren kunnen worden geschoten. Tijdens die voorwaartse beweging zullen de naaldjes het inktlint tegenkomen, en uiteindelijk tegen het papier drukken. Door dit afvuren van de naaldjes te doen, terwijl de printkop zich naar rechts beweegt, kan de vorm van een letter worden verkregen. Afbeelding 1-3 laat een aantal op deze manier te verkrijgen tekens zien. Het aantal naaldjes varieert per printer. Minimaal zullen er 7 naaldjes zijn, meestal zijn er 9 naaldjes, doch sommige printers hebben wel 18 naaldjes. In het algemeen kun je stellen, dat hoe meer

naaldjes de printkop heeft, hoe hoger de kwaliteit van de afgedrukte tekens is. De tekens uit het voorbeeld van afbeelding 1-3 worden met behulp van 9 naaldjes gemaakt.



Afb. 1-3 Samenstelling van tekens uit een matrix van puntjes.

In de printer zit een programma, dat in staat is, om een van de computer ontvangen code, om te zetten in een matrix van puntjes. Deze matrix wordt dan gebruikt, om de naalden, tijdens de horizontale beweging van de printkop, af te vuren. De vorm van de tekens zit dus in het programma in de printer opgeslagen. Sommige printers hebben een groot aantal verschillende vormen, die door de gebruiker (via de computer) kunnen worden geselecteerd.

We hebben nu al een paar keer gezien, dat het afdrukken van tekens tijdens het bewegen van de printkop gebeurt. De printkop dient zich met een constante snelheid te bewegen. Is de snelheid niet constant, dan zullen de letters er misvormd gaan uitzien. De vorm en de grootte van de tekens is helemaal afhankelijk van de software in de printer.

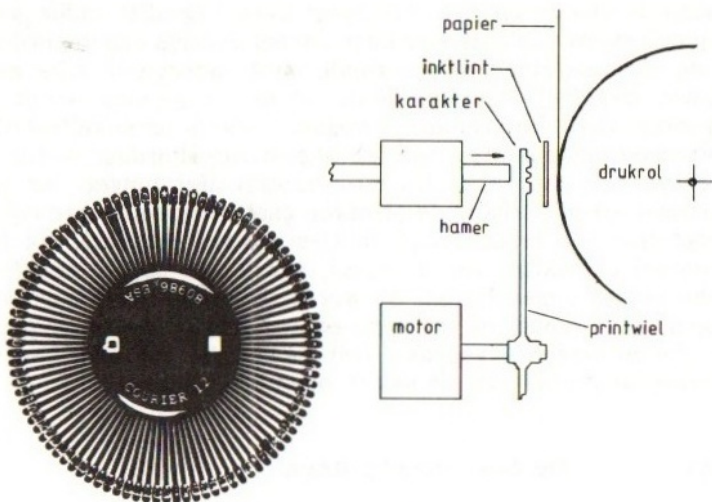
De tekens van de matrix printer bestaan uit puntjes. Het aantal puntjes, waaruit de tekens zijn opgebouwd, bepaalt de

kwiteit van de letter. In de Near Letter Quality mode worden de tekens vaak uit vier keer zoveel puntjes opgebouwd als in de normale mode. Die normale mode wordt ook vaak aangeduid met Data Quality mode. In de NLQ-mode wordt de printkop vaak langzamer bewogen, terwijl er tegelijkertijd per teken meer puntjes worden afgedrukt. Hierdoor wordt de breedte van een teken (in millimeters) niet langer dan een normaal teken (immers de printkop gaat langzamer, terwijl er gedurende een langere tijd puntjes worden gedrukt). Na het eenmaal afdrukken van de regel, wordt het papier een heel klein stukje opgeschoven, en wordt dezelfde regel nog eens afgedrukt. De puntjes vallen nu echter precies tussen de puntjes die al waren afgedrukt. Het gevolg is, dat alle puntjes tesamen een praktisch ononderbroken lijn vormen.

### **1.2.2 De daisy wheel printer.**

Een daisy wheel printer heeft veel meer weg van een type machine. De tekens zitten allemaal op armpjes. Deze armpjes worden tegen het papier geslagen, en indien er tussen het armpje en het papier een inktlint zit, zal er op het papier een afdruk van de letter verschijnen. Bij een conventionele type machine zit aan iedere toets, via een mechanische verbinding, een arm met daarop een teken. Bij een daisy wheel printer gaat dit anders. Daar wordt gebruik gemaakt van een wieltje waarop de armpjes met tekens zijn bevestigd als spaken. Aan het uiteinde van die spaken zitten de tekens. (zie afbeelding 1-4.)

Het principe van de daisy wheel printer is nu als volgt. Het wiel met tekens wordt tussen een hamer en het papier geplaatst. De hamer is zo klein, dat er maar 1 arm van het daisy wheel tegelijk mee kan worden aangetikt. Door nu het wiel met het gewenste teken tussen de hamer en het papier te draaien, en daarna de hamer te activeren, wordt het teken tegen het papier gedrukt. Door tussen het af te drukken teken en het papier een inktlint te plaatsen, zal op het papier een afdruk verschijnen van het teken op het daisy wheel.



Afb. 1-4 Het principe van een daisy-wheel printer.

Voor ieder af te drukken teken dient dus het wiel te worden gedraaid, totdat het gewenste teken tussen de hamer en het papier staat. Hierna kan worden gehamerd. Staat het teken op het papier, dan zal de printkop (hamer plus wiel) naar de volgende positie moeten worden verplaatst, waar dan opnieuw een teken kan worden geselecteerd en afgedrukt. De afstand tussen twee tekens is vast. Wel kan er op de meeste daisy wheel printers gekozen worden tussen een aantal vaste afstanden. Vaak zijn dit de afstanden 10, 12 en 15 tekens per inch. Daarnaast kunnen de tekens op de meeste daisy wheel printers ook nog proportioneel worden afgedrukt. In dat geval zal de printer de afstand tussen twee tekens laten afhangen van de breedte van die tekens.

Zowel voor de matrix printer als voor de daisy wheel printer geldt, dat het papier vaak met iedere gewenste hoeveelheid kan worden opgeschoven. Standaard waarden hiervoor zijn afstanden waarbij er 6 of 8 regels per inch kunnen worden geschreven. Meestal is het echter ook mogelijk om het papier

maar een heel klein stukje op te schuiven, bijvoorbeeld een honderdste van een inch. Deze mogelijkheid is erg belangrijk voor het afdrucken van grafische voorstellingen. Wilt u grafische voorstellingen afdrucken met uw printer, dan zult u er op dienen te letten dat de printer van uw keuze zowel de printkop als het papier over hele kleine afstanden kan verplaatsen. Later in dit boek zullen we zien dat het ook met daisy wheel printers mogelijk is om tekeningen af te drukken. Ook geldt voor beide soorten printers, dat ze gebruik maken van gewoon (goedkoop) papier en gewone (goedkope) inktlinten. Doordat de letters met mechanische kracht op het papier worden gedrukt, is het ook mogelijk om meervoudig papier te gebruiken, er kunnen dus doorslagen worden gemaakt, met behulp van bijvoorbeeld carbonpapier. Bij veel drukwerk kan dit een beslissende factor zijn.

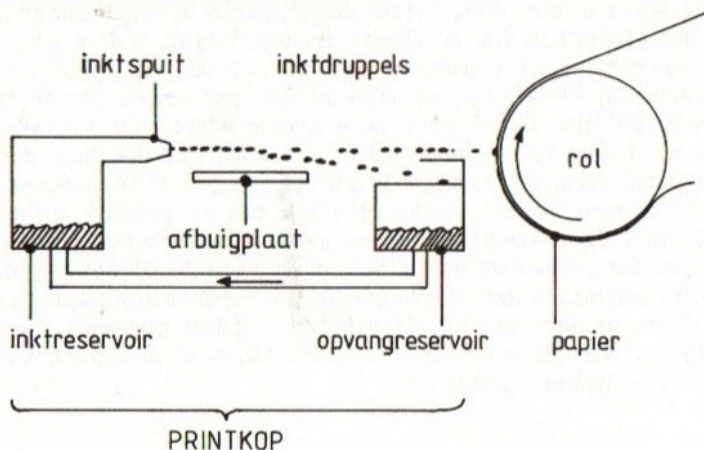
### **1.3 Non impact printers.**

In deze categorie zullen we drie typen printers bekijken. Alle drie hebben ze een groot voordeel gemeen, en wel, geruisloosheid. Zeker voor de hobbyist, die zijn hobby bedrijft in de huiskamer, kan dit een zeer belangrijke factor zijn. Wat ze verder nog gemeen hebben, is dat ze de tekens op dezelfde manier op het papier zetten als de reeds behandelde matrix printer. Een matrix van puntjes dus.

#### **1.3.1 De ink jet printer.**

De ink jet printer is nog een vrij nieuwe ontwikkeling. Met deze printer worden kleine druppeltjes inkt van uit een reservoir naar het papier geschoten. De weggeschoten inktdruppeltjes zijn statisch geladen. Onderweg, tussen de "inktspuit" en het papier, zijn magnetische velden aangebracht, waarmee de richting van het inktdruppeltje een beetje kan worden gewijzigd. Dit werkt eigenlijk net als de elektronenstraal van een beeldbuis van de televisie, waarbij je dan de elektronenstraal vervangen kunt denken door de straal met inktdruppeltjes en de fosforlaag van de beeldbuis door het papier. Het principe is

weergegeven in afbeelding 1-5.



Afb. 1-5 Het principe van de ink jet printer.

Het voordeel van deze printer is, dat er van normaal papier gebruik kan worden gemaakt. Dit in tegenstelling tot de twee andere typen non impact printers, die in de volgende paragrafen zullen worden behandeld.

Bij al deze voordelen zult u zich waarschijnlijk afvragen, waarom dit dan niet de meest verkochte en meest gebruikte printer is. Een reden daarvoor is, dat deze printer nog niet zo lang op de markt is, en dat er nog niet zo veel keus in uitvoeringen is. Een andere reden kan zijn, dat de inkt, kort nadat het op het papier is geschoten, nog niet helemaal droog is. Hierdoor kunnen, bij aanraking, vlekken ontstaan. In de drukwereld worden echter kranten gedrukt met een onvoorstelbaar hoge snelheid. Die kranten worden onmiddellijk na het bedrukken gevouwen en op elkaar gestapeld. Toch blijft die krant altijd goed leesbaar en ontstaan er nooit vlekken. Het ligt dus in de lijn der verwachtingen, dat de ink jet printer een grote toekomst tegemoet gaat.

### **1.3.2**

#### **De elektrostatische printer.**

Bij dit type printer wordt gebruik gemaakt van zeer speciaal papier. Dat papier is elektrisch geleidend. De naald(en) gaan vlak langs het papier, doch worden niet tegen het papier aangeschoten. Door nu de naald(en) een ander potentiaal te geven dan het papier, zal er een stroom gaan lopen, van de naald(en) naar het papier. Op de plaats waar zich een naald bevindt met een ander potentiaal dan het papier, zal een vonk ontstaan. Deze vonk brandt een vlekje in het papier. Vele vlekjes samen, mits ze het juiste patroon vertonen, resulteren in een teken.

Het nadeel van de elektrostatische printers is, dat het papier duur is, en dat het eindresultaat niet fraai is. Het papier is namelijk niet wit, maar ziet er metaalachtig uit.

Daar er van deze printers niet zo heel erg veel verschillende modellen te verkrijgen zijn, en daar er naast het grote voordeel van geruisloosheid toch wel een aantal belangrijke nadelen aan kleven, zullen we er hier niet verder op ingaan.

### **1.3.3**

#### **De thermische printers.**

In deze categorie printers hebben zich de laatste jaren nogal wat positieve ontwikkelingen voorgedaan. Deze printer is goed te vergelijken met een normale matrix printer, waarbij de printkop met naaldjes is vervangen door een printkop met punten die warm kunnen worden. De kop glijdt langs het papier en daar, waar een puntje moet worden geschreven, wordt de betreffende punt in de printkop warm gemaakt. Zou met normaal papier in de printer leggen, dan zou die warmte geen effect hebben. Steekt men er echter speciaal geprepareerd papier in, dan ontstaat op de plaats waar het papier wordt verwarmd een donker puntje.

U ziet, er komt geen inktlint aan te pas. Dit is een voordeel. Echter, een nadeel is, dat de afdruk na verloop van tijd gaat vervagen. Bovendien is het speciaal geprepareerde papier veel

duurder dan normaal papier. Dit laatste nadeel wordt echter weer teniet gedaan door het gebruik van speciale inktlinten, waarmee de thermische printer op normaal papier kan afdrucken.

Bijna alle thermische printers hebben de mogelijkheid om een speciaal inktlint te gebruiken. Door de hitte wordt de inkt van het lint aan het papier geplakt. Als daarna het lint weer van het papier wordt getrokken, blijft alleen op de plaatsen die verhit zijn geweest de inkt aan het papier kleven. Op die manier is het zelfs mogelijk om gebruik te maken van inktlinten met verschillende kleuren en daarmee een kleurenafdruk te maken.

Het gebruik van dit soort inktlinten neemt het bezwaar van duur papier, dat bovendien na verloop van tijd vervaagt, weg. De inktlinten zelf zijn echter niet erg economisch in het gebruik. Zeker de kleurenlinten niet. De linten worden namelijk steeds een hele regel opgeschoven, voordat er een regel of een deel van een regel wordt afgedrukt. Zelfs al wordt er maar 1 letter op een regel gedrukt, dan zal het lint toch voor een hele regel worden getransporteerd. Voor een goede afdruk kwaliteit dient bovendien mooi glad papier te worden gebruikt, hetgeen toch weer wat duurder is dan normaal papier. Alles bij elkaar kunnen we dus stellen, dat wanneer er veel drukwerk is te doen, de thermische printer duurder in gebruik is dan een normale matrix printer. Daar staat wel tegenover dat de thermische printers in aanschafprijs vaak goedkoper zijn dan normale matrix printers.

De keus tussen een normale matrix printer en een thermische matrix printer zal dus worden bepaald door enerzijds de hoeveelheid drukwerk, die er mee moet worden gedaan, en anderzijds de plaats waar dat drukwerk moet worden gedaan (aparte kamer of huiskamer). Het grote voordeel van geruisloosheid mag zeker niet worden uitgevlakt.



## 1.4 Opties.

Op praktisch alle printers kan het papier worden getransporteerd met behulp van de zogenaamde **friction feed**. Dit is dezelfde manier als het papiertransport op een conventionele typemachine. Het papier zit geklemd tussen de rol en een aantal transportwielen. Wanneer de rol een stukje wordt gedraaid, gaat het papier automatisch mee. Deze manier van papiertransport is uitstekend geschikt voor losse velletjes. Zoals u weet, gebruiken computerprinters vaak papier van een rol of van een stapel. Papier van een rol wordt altijd op dezelfde manier getransporteerd als losse velletjes. Papier van een stapel wordt, zoals we straks zullen zien, op een heel andere manier getransporteerd.

De kans dat het papier een beetje scheef wordt getrokken is bij friction feed echter groot. Vandaar dat het printen op papier aan de rol nooit erg opgang heeft gemaakt. Om friction feed transport bij papier aan de rol toch goed te laten verlopen, dient een speciale rolhouder te worden gebruikt. De rol papier kan in deze houder, die achter aan de printer wordt bevestigd, worden gelegd en vervolgens kan het papier zonder verdere wrijving aan de printer worden toegevoerd. Hierdoor is de kans dat het papier scheef trekt gering geworden. De kans dat het papier echter van het begin af aan scheef in de printer zit is wel groot. Het gevolg van het scheef in de printer zitten van het papier (ook al is het maar een heel klein beetje scheef) is, dat de afgedrukte tekst steeds dichter naar een van beide randen van het papier gaat lopen.

Papier van de stapel wordt meestal aangeduid met **kettingbaanformulieren**. Dit soort papier heeft aan de randen gaatjes met een gestandaardiseerde afmeting op gestandaardiseerde afstand van elkaar. Dit papier wordt in de printer over een rol, met aan beide zijden van de printer een wiel met uitstekende nokken, gelegd. De nokken passen precies in de gaatjes in de randen van het papier. Het gevolg is, dat het papier nooit kan verschuiven. Het blijft altijd door de nokken op de juiste plaats zitten. Deze manier van papier transport wordt vaak aangeduid met **tractor feed**.

Niet alle printers zijn standaard van deze tractor feed faciliteit voorzien. Heeft een printer deze faciliteit niet standaard ingebouwd, dan is die faciliteit meestal wel als optie te koop. Of u bij de aanschaf van een printer er op moet letten of die printer al of niet een tractor feed faciliteit heeft, of dat die faciliteit los is bij te kopen, ligt aan het drukwerk waarvoor u de printer aanschaf. Denkt u veel lange listings te maken, of ander drukwerk waarbij lange stukken achter elkaar moeten worden afgedrukt, dan is het zeker goed om op de mogelijkheid van tractor feed te letten. Weet u daarentegen zeker dat u alleen maar korte stukken (brieven bijvoorbeeld) zult afdrucken, dan is friction feed voldoende en kunt u gebruik maken van losse velletjes.

Voor sommige printers is er nog een andere optie te verkrijgen. Stel, dat u uw printer hoofdzakelijk gebruikt voor het afdrucken van brieven en dat die brieven in meervoud moeten worden afgedrukt. In dat geval zult u graag losse velletjes willen gebruiken. Elke keer dat er aan een nieuwe brief wordt begonnen, zult u echter naar uw printer moeten lopen, om er een nieuw papiertje in te leggen. Het zou in dat geval erg handig zijn, indien u een hele stapel papieren in de printer kon leggen, en dat de printer zelf, steeds wanneer dat nodig is, een papiertje invoert. Dit wordt aangeduid met **sheet feeder**.

Sommige sheet feeders zijn alleen geschikt voor papieren van een bepaalde afmeting en bepaalde dikte. Andere sheet feeders kunnen allerlei maten van papieren aan. Zo is het in sommige sheet feeders mogelijk om er enveloppen in te leggen. Voor gebruik bij een club, zou het wel eens handig kunnen zijn, om direct op enveloppen te kunnen printen. In dat geval hoeft u niet meer eerst adressen op stickers te printen, om ze daarna op de enveloppe te plakken.

## 1.5 Conclusie

We hebben nu in vogelvlucht gezien welke printers er zijn, hoe ze werken, en wat de voor- en nadelen van iedere printer zijn. Daarnaast hebben we een aantal opties gezien. Om de

juiste printer te kunnen kiezen, zult u goed moeten weten wat u er mee wilt doen. Pas daarna kunt u bepalen of een printer wel of niet aan uw wensen tegemoet zal komen.

Bij alle soorten en typen van printers hebt u keuze in snelheid. Hoe hoger de snelheid, hoe hoger de prijs. Het kan voorkomen, dat twee printers ongeveer dezelfde prijs hebben, doch dat de ene printer veel sneller is dan de andere. Let dan goed op de kwaliteit van de afgedrukte letters.

Bijna alle printers hebben tegenwoordig de mogelijkheid om in twee richtingen te printen. Dit wil zeggen, dat de tekst voor een regel eerst in een in de printer ingebouwd geheugen wordt opgeslagen. Zit de regel eenmaal in de printer, dan kan de printer bepalen of er van links naar rechts of van rechts naar links wordt afgedrukt. Deze mogelijkheid wordt aangeduid met **bi-directional printing**. In welke richting er wordt afgedrukt is afhankelijk van de positie van de drukkop. Staat de drukkop aan de rechterkant, dan zal de volgende regel van rechts naar links worden afgedrukt. Staat de drukkop aan de linkerkant, dan wordt er van links naar rechts afgedrukt.

Veel van de moderne printers hebben ook nog de mogelijkheid van **logical seeking**. Daarbij legt de printkop, indien dat niet nodig is, niet de weg van de hele regel af, maar gaat slechts zover als nodig is. Hierdoor kunnen grote stukken spaties als het ware worden overgeslagen. Dit betekent een belangrijke snelheidsverbetering.

Wat we nog niet hebben besproken, is de aansluiting aan de computer. Dit komt echter in het volgende hoofdstuk uitgebreid aan de orde.

## 2 Aansluiten aan de computer

Bij het aansluiten van een printer aan een computer zijn twee manieren te onderscheiden. Bij de ene manier worden de gegevens bit voor bit (over 1 lijntje) van de computer naar de printer gestuurd. Bij de andere manier worden alle bitjes van een ASCII-code tegelijkertijd (over 8 lijnen) van de computer naar de printer gestuurd. De eerste manier wordt aangeduid als een **seriele** verbinding. De tweede manier wordt aangeduid als een **paralelle** verbinding. Voor de seriele verbinding wordt meestal gebruik gemaakt van de Internationaal gestandaardiseerde **RS232C**-verbinding. Deze verbinding zullen we in de volgende paragraaf behandelen. Voor de paralel-verbinding wordt meestal de **Centronics**-interface gebruikt. Deze interface is in de loop der jaren uitgegroeid tot een Internationale standaard, doch werd oorspronkelijk door een printerfabrikant voor zijn eigen printers ontworpen. Ook deze interface zullen we verderop in dit hoofdstuk behandelen.

Behalve de tot nu toe genoemde interfaces, zijn er nog tal van andere. Die andere interfaces worden echter slechts op bepaalde computers, in combinatie met bepaalde printers, gebruikt, en zijn te specifiek om in een algemeen printerboek te behandelen.

Indien u al een computer hebt, en u wilt een printer kopen, dan zult u moeten uitzoeken of uw computer een RS232C- of een Centronics-interface heeft. Daarna kunt u een printer met een overeenkomstige interface uitzoeken. Heeft uw computer geen van beide interfaces, dan zult u moeten kijken of er voor uw computer een uitbreidingsset bestaat, waarmee een van beide interfaces wordt verkregen.

### 2.1 De RS232C-interface.

Zoals in het begin van dit hoofdstuk reeds opgemerkt, is de

RS232C-interface een serie-interface. Alle bitjes van een code worden na elkaar over dezelfde draad verzonden. Bij een verbinding tussen een computer en een printer zouden dus in principe twee draadjes voldoende zijn, een massadraad en een gegevens draad. De gegevens worden dan van de computer naar de printer gestuurd. De printer hoeft niets anders te doen dan de ontvangen gegevens op papier af te drukken.

Hierin zit echter een probleem, namelijk het volgende. Hoe snel worden de gegevens naar de printer gestuurd, en hoe snel kan de printer die gegevens op papier zetten. Een bijkomend probleem zou kunnen zijn, dat er tijdens de overdracht van de gegevens een beetje verloren gaat. De printer heeft dan een verkeerd teken ontvangen, of weet helemaal niet wat te doen met het ontvangen teken. In beide voorgaande gevallen zou het handig zijn wanneer de printer ook een manier had om de computer te laten weten of de gegevens goed zijn ontvangen en of de printer in staat is nog meer gegevens te ontvangen.

De printer heeft dus een terugmeldmogelijkheid nodig. Hoe deze mogelijkheid ook wordt gerealiseerd, het zal altijd betekenen dat er nog een draad (of meerdere draden) nodig is.

Om te voorkomen, dat er tijdens de gegevensoverdracht leesfouten worden gemaakt, dienen beide zijden (computer en printer) zich aan een aantal afspraken te houden. Zo dienen ze beide dezelfde "taal" te spreken. Hiermee wordt bedoeld, dat ze beide dezelfde code, die uit hetzelfde aantal bitjes bestaat, dienen te gebruiken voor dezelfde tekens. Ook dienen ze beide met dezelfde snelheid te "spreken". Heeft u wel eens geprobeerd om een gesprek dat op normale snelheid is opgenomen op een bandrecorder af te draaien met de dubbele snelheid? U zult dan ontdekt hebben, dat het gesprek niet meer te volgen was. Dit geldt ook voor computer en printer. Indien ze niet beide de zelfde snelheid gebruiken, de een voor het zenden (spreken) en de ander voor het ontvangen (luisteren), dan zal er geen zinnig gegeven over komen. Voor een nadere uitleg van voorgaande problemen zullen we een de RS232C-standaard eens nader gaan bekijken.

### 2.1.1 Transmissiecode.

Zoals gezegd, dienen beide zijden dezelfde taal te spreken, ofwel dezelfde code te gebruiken. Een zeer gebruikelijke code is de ASCII-code. Praktisch alle computers en alle printers gebruiken deze code. Sommige grote systemen gebruiken andere codes (EBCDIC voor IBM-mainframes, een eigen code voor bijvoorbeeld de ZX81 van Sinclair). Het gebruik van de ASCII-code op zich is echter nog geen garantie voor goede gegevensoverdracht. Dit houdt verband met de gebruikte HW-componenten (de zender en ontvanger).

Een code bestaat uit nulletjes en eentjes. Dit zijn de twee toestanden die de computer kan onderscheiden. Hoe zou je dan aan moeten geven dat er niets wordt verzonden? Geef je dat met een nulletje aan of met een eentje?.

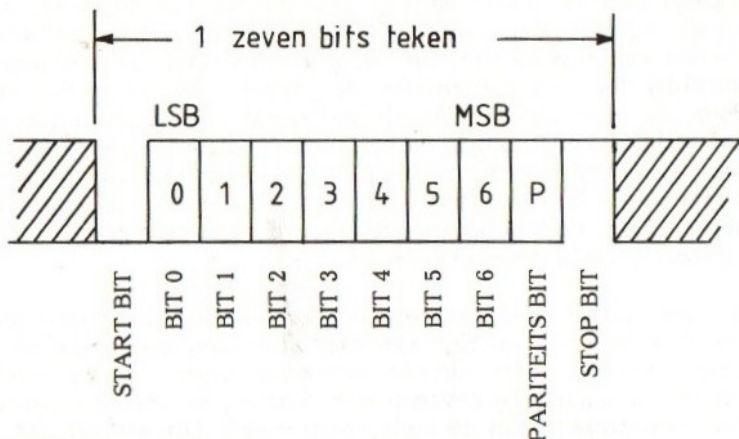
Om dit probleem op te lossen is de volgende afspraak gemaakt. Zolang er geen gegevens worden verstuurd, wordt de lijn, waarover de gegevens moeten worden verstuurd, op logisch 1 gehouden. Om nu aan te geven dat er een tekencode gaat worden verzonden, wordt een **startbit** verstuurd. Dit bit is een logische 0. Na dit startbit kunnen de bitjes van de ASCII code worden verzonden.

Om de printer de mogelijkheid te geven, te controleren of de ontvangen code correct is ontvangen, wordt onmiddellijk na de code een **pariteitsbit** verzonden. Een pariteitsbit wordt door de computer gegenereerd uit de verzonden codebits en meegestuurd met de code. Er zijn verschillende manieren van pariteitsberekening mogelijk. Stel dat de bitjes in de verzonden code uit een oneven aantal enen bestaan. We kunnen dan het pariteits bit 1 maken, om het totaal aan enen even te maken. In dat geval spreken we van **even pariteit** (EVEN parity). Zouden we daarentegen het totaal aan bitjes met de waarde 1 altijd oneven willen laten zijn, dan zouden we bij een oneven aantal eentjes in de code het pariteitsbit 0 moeten maken, terwijl bij een even aantal enen in de code het pariteitsbit 1 zou moeten zijn. In dat geval spreken we van

**oneven pariteit** (ODD parity). Een laatste mogelijkheid, die bij de meeste home-computers wordt toegepast, is het ontbreken van een pariteitsbit. In dat geval mag de ontvanger (printer) ook niet controleren op pariteit. Zou deze dat wel doen, dan zou het stopbit als pariteitsbit worden gezien en dat zou betekenen dat in de helft van de codes de pariteit als fout zou worden gezien. Het niet aanwezig zijn van pariteit wordt aangeduid met **NO parity**. Het zal duidelijk zijn, dat beide zijden (computer en printer) van dezelfde manier van pariteit gebruik dienen te maken.

Het pariteitsbit kan dus, afhankelijk van de verzonden code, een 0 of een 1 zijn. Het kan dus nooit een goede indicatie geven over het einde van het verzonden teken. Om het einde van een teken aan te geven wordt daarom, na het pariteitsteken, een **stopbit** aan de code toegevoegd. Dit stopbit (er is overigens vaak de keuze uit 1 of 2 stopbits) heeft de logische waarde 1. Hierdoor ontstaat er een goed verschil tussen het einde van het ene teken (logisch 1) en het begin van het volgende teken (logisch 0), in het geval dat beide tekens direct na elkaar zouden worden verzonden. Daar het aantal stopbits niet vast ligt, zult u er voor dienen te zorgen, dat zowel de computer als de printer hetzelfde aantal stopbits gebruiken. Meestal betekent dit dat er op de printer een schakelaartje in een bepaalde stand moet worden gezet.

Afbeelding 2-1 laat zien, hoe een teken er, met in achtname van het voorgaande er gaat uitzien. U ziet nu dat een teken, waarvan de code uit 7 bits bestaat, wordt verzonden als een teken dat uit 10 bits bestaat. Behalve de code zelf, met daarin een start- en een stopbit, is er geen enkele controle over de snelheid. De snelheid kan aan beide zijden (computer en printer) worden gekozen. Heel belangrijk is natuurlijk dat aan beide zijden dezelfde snelheid wordt gekozen. Deze snelheid wordt aangegeven in bits per seconde. Een historisch hiervoor gegroeide term is **Baud**. Een heel gebruikelijke snelheid is 9600 Baud. Bij aansluiting van een printer via een telefoonlijn en modems zou een snelheid van 1200 Baud zijn te adviseren, terwijl aansluiting via akoestische modems een snelheid van 300 Baud verdient. Deze snelheid kan in de computer meestal



Afb. 2-1 Samenstelling van een teken voor seriele overdracht.

met behulp van een commando worden ingesteld. In de printer moeten echter een aantal schakelaars op de juiste stand worden ingesteld.

### 2.1.2 En als er nu toch iets fout gaat?

Zelfs als we alle instellingen, zoals besproken in de vorige paragraaf, goed hebben gedaan, is het nog niet zeker dat er niets meer fout kan gaan. Meestal zullen we de gegevens sneller naar de printer sturen dan die ze op papier kan schrijven. Ook al heeft de printer een intern geheugen, waarin de ontvangen gegevens kunnen worden opgeslagen, dan nog kan dat buffer vollopen. De printer zal dus een manier wensen om de zender te vertellen dat deze even moet wachten met het verzenden van nog meer gegevens, totdat er weer ruimte in het printerbuffer is voor het opslaan van meer gegevens.

Er zijn meerdere manieren, waarop de printer aan de computer kan laten weten dat deze even moet wachten. Er zijn echter twee zeer veel gebruikte manieren, die we hier zullen



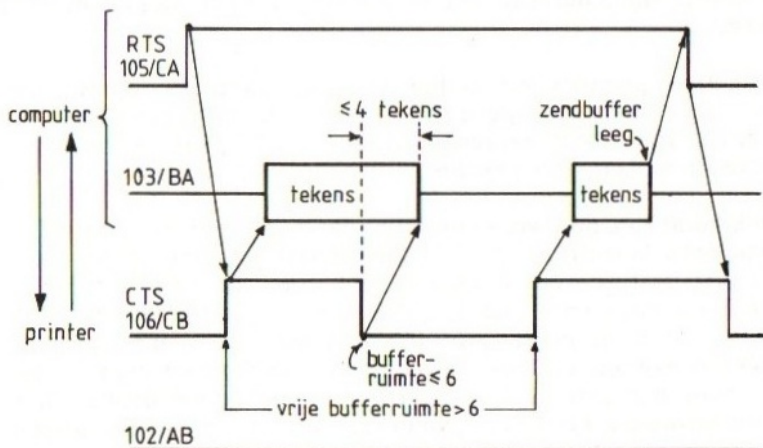
gaan bespreken. Die twee manieren zijn:

- Het ready/busy protocol.
- Het Xon/Xoff protocol.

Bij het eerstgenoemde protocol wordt een extra draad tussen de printer en de computer aangesloten. Bij de tweede manier dient de printer in staat te zijn gegevens naar een computer te versturen.

### 2.1.3 Het ready/busy protocol.

In dit protocol worden een aantal extra draden tussen de computer en de printer aangesloten. Met behulp van deze draden kan de computer de printer vragen, of deze gereed is om gegevens te ontvangen en kan de printer aangeven of dit inderdaad het geval is. Dit **protocol** is in afbeelding 2-2 weergegeven.



Afb. 2-2 Het ready/busy protocol.

In afbeelding 2-2 zijn, naast de data-lijn, twee lijnen getekend. De computer zal de lijn met de naam CA/105 activeren (RTS=Ready To Send), hetgeen door de printer wordt geïnter-

preteerd als: "Hallo printer, ik heb gegevens te versturen, bent u er klaar voor?". Indien de printer inderdaad klaar staat om gegevens van de computer te ontvangen, zal deze de lijn, waar CB/106 bij staat, activeren (CTS=Clear To Send). De computer ziet dit en zal het interpreteren als: "Ja computer, ik ben klaar om uw gegevens te ontvangen, stuur ze maar.". Hierop zal de computer beginnen gegevens over de lijn, waar BA/103 (TxD=Transmitted Data) bij staat, te sturen en hiermee door te gaan totdat de printer de lijn CB/106 deactiveert. Zolang de computer nog meer gegevens te versturen heeft, zal de lijn CA/105 geactiveerd blijven, opdat de printer weet, dat deze lijn CB/106 moet activeren zodra er weer gegevens kunnen worden aangenomen. Deactiveert de computer de lijn CA/105, dan zal de printer de lijn CB/106 ook deactiveren, er valt dan immers geen data meer te verwachten. Onder alle omstandigheden zal de lijn AB/102 tussen de computer en de printer aanwezig dienen te zijn. Dit is de gemeenschappelijke aarde van de beide apparaten, die nodig is om de signaalniveau's op de andere lijnen te kunnen detecteren.

Sommige printers en computers gebruiken in plaats van CA/105 een andere lijn, en wel de lijn CD/108, terwijl dan de lijn CB/106 wordt vervangen door CC/107. Het gebruik van deze lijnen is echter precies gelijk.

Ook komt het nog wel eens voor, dat computer- of printerfabrikanten besluiten, om de lijnen BA/103 (transmitted data) en BB/104 (received data) om te keren. Normaal gesproken moet de data over lijn BA/103 worden verzonden. Aan de ontvangst kant moet deze lijn dan met het connectorpunt BB/104 worden verbonden. Sommige fabrikanten denken dat dit voor een gebruiker te moeilijk is, en geven daarom het connectorpunt BB/104 de naam BA/103. U hoeft dan alleen maar de punten BA/103 aan beide zijden met elkaar te verbinden. Zouden echter zowel de fabrikant van de computer als van de printer deze truuk hebben uitgehaald, dan zit het toch weer fout. Met andere woorden, deze truuk zorgt alleen maar voor extra verwarring. Voor u is alleen van belang te weten, op welk connectorpunt uw computer de data zet die naar de

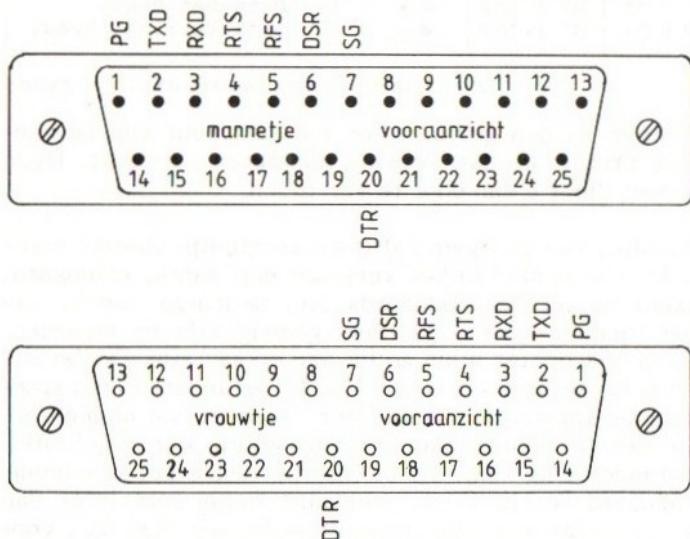
CCITT-naam	RS232C-naam	DIN-naam	Afkorting van de functienaam	DCE Richting DTE	Pin-nummer	Omschrijving
101	AA	E1	PG		1	Protective Ground Signal Ground
102	AB	E2	SG		7	
103	BA	D1	TXD	↑	2	Transmitted data
104	BB	D2	RXD	↓	3	Received data
105	CA	S2	RTS	↑	4	Request To Send
106	CB	M2	RFS	↓	5	Ready For Sending
107	CC	M1	DSR	↓	6	Data Set Ready
108	CD	S1	DTR	↑	20	Data Terminal Ready

Tabel 2-1 De meest gebruikte RS232C-signalen.

printer moet worden gestuurd, en op welk punt van de connector de printer de data van de computer verwacht. Deze twee punten dient u dan door te verbinden.

De aanduiding van de lijnen zal u waarschijnlijk vreemd voorkomen. Er zijn echter in het verleden een aantal standaards ontwikkeld en in die standaards zijn bepaalde namen aan bepaalde lijnen gegeven. Jammer genoeg zijn er meerdere standaards, die echter bijna gelijk aan elkaar zijn. De letteraanduiding bij de lijnen is uit de RS232C-standaard, een standaard van de Amerikaanse computer industrie, die momenteel door de meeste home computerfabrikanten wordt gebruikt. De cijferaanuiding komt uit de internationale datacommunicatiestandaard (V.24). Deze standaard kwam iets later dan RS232C, en bevat iets meer mogelijkheden dan RS232C. Voor het grootste deel overlappen beide standaards elkaar echter. In tabel 2-1 wordt een overzicht van de meest gebruikte signalen uit verschillende standaards gegeven.

Voor RS232C is geen connector in de standaard opgenomen. Wel is er een advies gegeven. In de V.24 standaard is wel een omschrijving van de te gebruiken connector opgenomen. Deze connector komt overeen met de in RS232C geadviseerde connector. De connector is weergegeven in afbeelding 2-3. In diezelfde afbeelding is ook aangegeven, op welke pinnen de verschillende signaallijnen dienen te worden aangesloten. Er zijn echter zoveel zaken die niet bindend zijn voorgeschreven, of die voor tweeerlei uitleg vatbaar is, dat zowel computers als printerfabrikanten wel eens van de geest van de standaard afwijken. Hierdoor ontstaan veel aansluitproblemen voor de gebruiker, voor u dus. Voor een juiste aansluiting zult u dan ook zowel de specificatie van de computer als van de printer dienen te bestuderen, om daarna te beslissen welke draad aan welk connectorpunt dient te worden aangesloten.



Afb. 2-3 De standaard RS232C-connector.

De hiernavolgende beschrijving geeft de functie van de meestgebruikte signaallijnen weer. Hierbij dient u te bedenken, dat de meeste lijnen zijn ontworpen voor het gebruik van

**modems.** Modem is de afkorting van MODulator/DEModulator. Dit is een apparaat, dat de digitale computer om moet zetten naar analoge signalen. Deze analoge signalen kunnen via een telefoonlijn van de ene abonnee naar de andere worden verzonden. Modems zijn dus nodig, om verbindingen over langere afstanden te realiseren. In dit boek wordt het gebruik van modems niet uitgelegd, omdat we ervan uitgaan, dat u een printer naast de computer hebt staan.

### **101**

Protective Ground vormt de massaverbinding tussen beide met elkaar verbonden apparaten en wordt meestal verbonden met de afscherming van de kabel.

### **102**

Signal Ground dient als gemeenschappelijke terugvoerlijn voor de zenders van beide zijden en dus voor de ontvangers van beide zijden.

### **103**

De door de computer te verzenden gegevens worden via deze lijn verzonden. Twee begrippen verdienen nadere uitleg.

DTE (Data Terminal Equipment) is de aanduiding voor apparatuur, die aan het einde van de datacommunicatieverbinding zit (computer en printer).

DCE (Data Circuit Equipment) is de aanduiding voor apparatuur, die tussen de DTE en de telefoonlijn zit (modems).

Wanneer zowel de computer als de printer als DTE zijn uitgevoerd (en dit is lang niet altijd het geval), dient lijn 103 van de ene kant te worden verbonden met lijn 104 van de andere kant.

### **104**

Op deze lijn ontvangt de printer de af te drukken gegevens. Sommige printerfabrikanten hebben hun printer echter uitgevoerd als DCE. In dat geval dient de lijn 103 van de computer te worden verbonden met lijn 103 van de printer (zie ook onder 103).

### **105**

Oorspronkelijk is dit signaal bedoeld om de modulator in het modem aan te zetten. Bij gebruik zonder modem wordt dit signaal tussen beide DTE's als volgt gebruikt:

De computer activeert 105.

De printer detecteert dit en, in plaats van de modulator aan te zetten, signaleert de printer door middel van signaal 106 dat deze gereed is om de gegevens te ontvangen.

### **106**

Oorspronkelijk wordt dit signaal door het modem gegenereerd, om aan te geven dat de draaggolf is opgebouwd, en dat de gegevens kunnen worden verzonden. Bij gebruik zonder modem wordt dit signaal van de printer naar de computer teruggestuurd, om aan te geven dat de computer gegevens mag versturen.

### **107**

Met behulp van deze lijn geeft het modem aan de DTE aan, dat het modem met de lijn is verbonden, en dat met het verzenden van gegevens kan begonnen.

In het ready/busy-protocol, bij gebruik van de lijn zonder modems, wordt deze lijn soms op dezelfde manier gebruikt als lijn 106.

### **108**

Aan deze lijn zijn twee functies toegekend (108.1 en 108.2). In RS232 is alleen de functie 108.2 bekend. Met signaal 108.2 kan de DTE aan het modem kenbaar maken, dat deze (de DTE) gereed is om te zenden (of ontvangen).

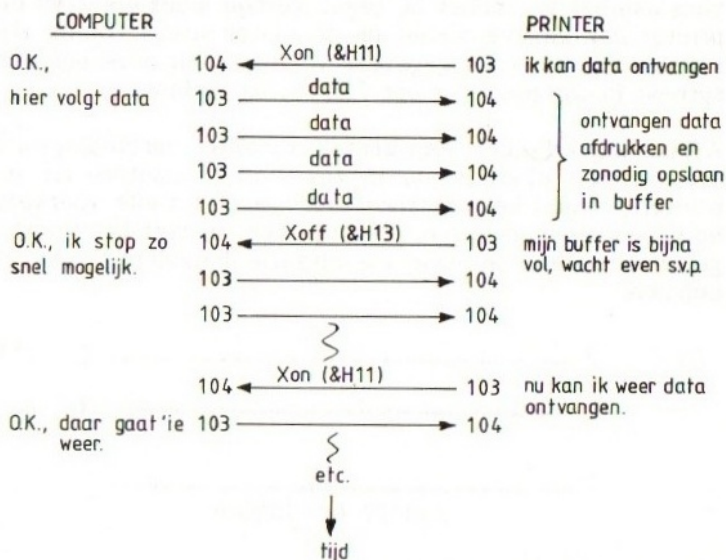
In het ready/busy-protocol, bij gebruik van de lijn zonder modems, wordt deze lijn soms op dezelfde manier gebruikt als lijn 105.

## **2.1.4 Het Xon/Xoff-protocol**

Bij gebruik van dit protocol dienen zowel de computer als de printer te kunnen zenden en ontvangen. De computer kan dus gegevens naar de printer sturen, maar de printer kan ook

gegevens naar de computer sturen. Beide kunnen ze gegevens ontvangen. Bij gebruik van het Xon/Xoff protocol zijn geen verdere "handshaking" signalen nodig. De signalen 103 van de zenderzijde dienen te worden verbonden met de signalen 104 van de ontvangstzijde. Bovendien dienen de signalen 102 van beide zijden met elkaar te worden verbonden. Met deze drie signalen kan in principe het Xon/Xoff protocol worden afgewikkeld.

Dit protocol wordt niet alleen voor verbindingen tussen computers en printers gebruikt, doch hier zullen we alleen het gebruik van dit protocol voor printers beschrijven.



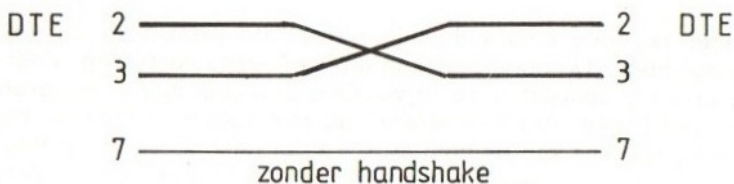
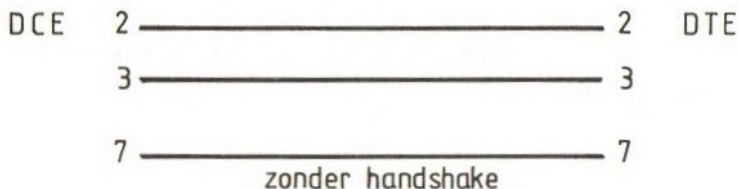
Afb. 2-4 Het Xon/Xoff-protocol.

Onder normale omstandigheden staat de printer in de ontvangstmode. De computer kan dus gegevens versturen. Zodra de printer, doordat deze bijvoorbeeld papier moet opvoeren, en dus tijdelijk niet kan afdrukken, niet langer in staat is nog meer gegevens van de computer te ontvangen, zal de printer een Xoff signaal naar de computer sturen (zie afbeelding 2-4).

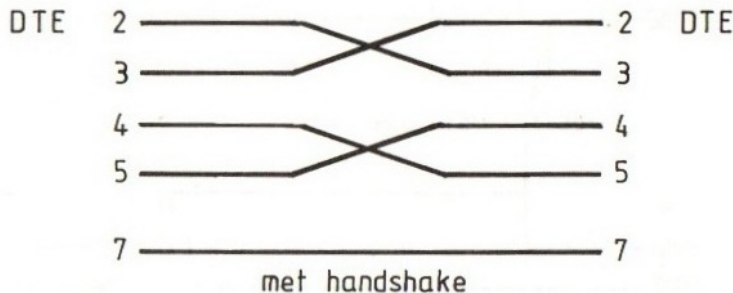
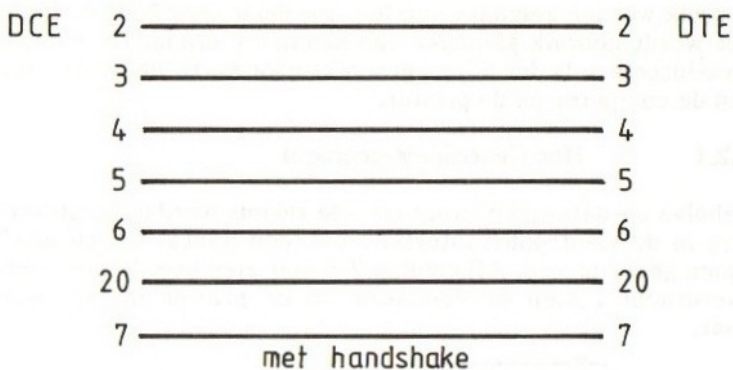
Het Xoff-signaal heeft de hexadecimale code X'13'. In de meeste ASCII-tabellen staat in plaats van Xoff de naam DC3. Zodra de computer dit Xoff-signaal ontvangt, zal deze het versturen van verdere gegevens stoppen. Wanneer de printer weer in staat is om gegevens van de computer te ontvangen, zal deze een Xon-signaal naar de computer sturen. In de meeste ASCII-tabellen staat, in plaats van Xon, de naam DC1. De hexadecimale code is X'11'. De computer zal na ontvangst van Xon weer verder gaan met het versturen van gegevens.

U ziet, dat voor dit protocol de computer, terwijl deze gegevens aan het verzenden is, tegelijkertijd moet opletten of de printer ook gegevens naar de computer stuurt. Beide zijden kunnen tegelijkertijd gegevens versturen en ontvangen. Men spreekt in dat geval van een full duplex verbinding.

Afbeelding 2-5 geeft een aantal mogelijke verbindingen. Het grote aantal afwijkingen, in zowel de computers als in de printers, maakt het praktisch onmogelijk om alle voorkomende kabels weer te geven. Hopelijk kunt u, met behulp van de gegeven uitleg, de voor uw situatie benodigde kabel zelf bepalen.







Afb. 2-5 Veel gebruikte manieren van verbindingen.

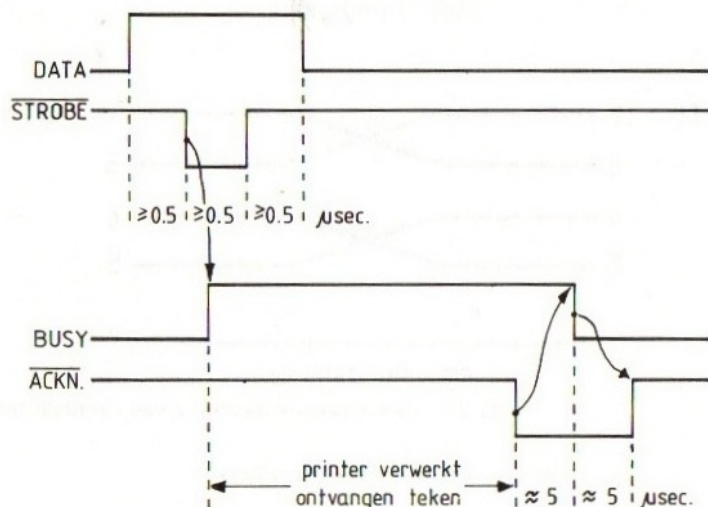
## 2.2 Centronics parallel-interface.

In tegenstelling tot wat we bij RS232C hebben gezien, waarbij alle bitjes van een teken over dezelfde lijn werden verstuurd, wordt bij de Centronics-interface gebruik gemaakt van 8 datalijnen, zodat alle bitjes van een teken tegelijkertijd kunnen worden verzonden. Het groten voordeel hiervan is, dat er een veel hogere overdrachtssnelheid kan worden bereikt. Hiertegenover staat als nadeel, dat er een kabel met veel meer draadjes moet worden gebruikt. Deze kabel zal daardoor veel duurder zijn dan de kabel uit de RS232C-interface. Bovendien kan met behulp van de Centronics-interface nooit

gebruik worden gemaakt van het openbaar telefoonnet. In dat net wordt gebruik gemaakt van slechts 1 draad. De Centronics-interface is dus alleen geschikt voor korte afstanden tussen de computer en de printer.

### 2.2.1 Het Centronics-protocol.

Behalve de datalijnen (waarover de tekens worden verstuurd), zijn in de Centronics-interface ook een aantal "handshake"-lijnen gedefinieerd. Afbeelding 2-6 laat zien hoe de gegevensoverdracht tussen de computer en de printer in zijn werk gaat.



Afb. 2-6 Het Centronics-protocol.

De computer zet de bits van het te versturen teken op de acht datalijnen. Ongeveer 1 microseconde nadat de data op de datalijnen is gezet, wordt door de computer het STROBE-sig-naal geactiveerd. De printer reageert hier op door het signaal BUSY te activeren. Dit wil zeggen, dat de printer bezig is de data van de datalijnen in te lezen en te verwerken. Zolang het BUSY-sig-naal actief is, mag de computer geen

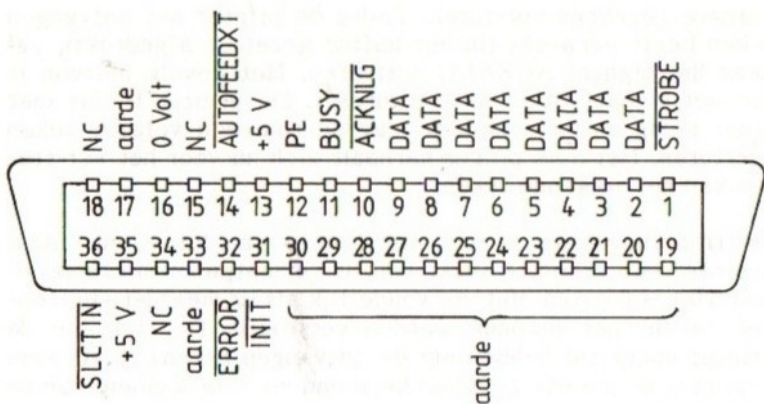
verdere gegevens versturen. Zodra de printer het ontvangen teken heeft verwerkt (in het buffer gezet of afgedrukt), zal deze het signaal ACKNLG activeren. Het gevolg hiervan is dat het signaal BUSY wordt gereset. De printer is dus niet meer bezig en de computer mag daarom een volgend teken versturen. Het hele proces herhaalt zich nu voor het versturen van het volgende teken.

De tijd, die nodig is om met behulp van het Centronics-protocol een teken te versturen, is te meten in microseconden. Het is daarom niet verwonderlijk als er meerdere duizenden tekens per seconde worden verstuurd. De tijd, die de printer nodig zal hebben om de ontvangen tekens te verwerken, is in de meeste gevallen bepalend voor de snelheid van de overdracht.

## **2.2.2 De fysieke verbinding.**

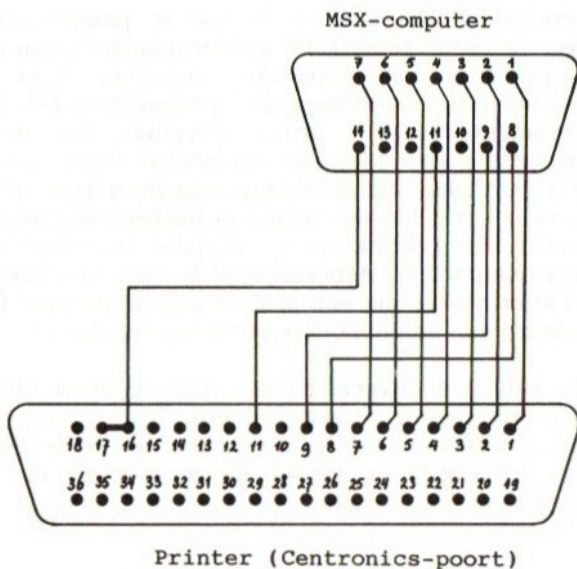
In de Centronics-standaard is de toe te passen connector vastgelegd. Er dient gebruik te worden gemaakt van een 36-polige AMPHENOL connector. Deze connector, met de signaalnamen per pin, is weergegeven in afbeelding 2-7. Het zal de MSX-computerbezitter direct opvallen, dat de MSX-computer slechts een 14-polige connector bezit, terwijl er toch wordt gesproken van de Centronics-interface. Inderdaad wordt er, voor zover het de connector betreft, afgeweken van de standaard. Het gebruik van de signalen is echter volledig Centronics compatibel. Afbeelding 2-8 laat zien hoe u zelf een kabel kunt maken om een printer met werkelijke Centronics-standaard aan een MSX-computer aan te sluiten.

Hieronder volgt een overzicht van alle signalen, die in de Centronics-interface zijn gedefinieerd. De richting van de signalen is gezien vanuit de printer. Staat er in het overzicht dus dat de richting IN is, dat wil dat zeggen dat het van de computer naar de printer gaat.



vooraanzicht vrouwtje = achteraanzicht mannetje

Afb. 2-7 De Centronics-connector.



Printer (Centronics-poort)

Afb. 2-8 De verbinding tussen MSX-computer en printer.

signaal PIN	massa PIN	SIGNAAL	RICHTING
1	19	<u>STROBE</u>	in
2	20	DATA1	in
3	21	DATA2	in
4	22	DATA3	in
5	23	DATA4	in
6	24	DATA5	in
7	25	DATA6	in
8	26	DATA7	in
9	27	DATA8	in
10	28	<u>ACKNLG</u>	uit
11	29	BUSY	uit
12	30	PE	uit
13	-	+5 V	-
14	-	AUTOFEEDXT	in
15	-	NC	-
16	-	0 V	-
17	-	Ground	-
18	-	NC	-
31	-	<u>INIT</u>	in
32	-	<u>ERROR</u>	uit
33	-	Ground	-
34	-	NC	-
35	-	+5 V	-
36	-	<u>SLCTIN</u>	in

Tot slot van dit hoofdstuk dient nog te worden opgemerkt, dat lang niet alle Centronics-signalen altijd worden gebruikt. In praktisch alle gevallen worden alleen de datalijnen, STROBE, BUSY en ACKNLG gebruikt. In tegenstelling tot de RS232C-interface, waar het maken van een werkende verbinding vaak op grote problemen stuit, is het verbinden van een printer via de Centronics-interface bijna altijd probleemloos. Zelfs wanneer er, zoals bij MSX-computers, enigszins van de standaard wordt afgeweken, dan nog werkt de verbinding probleemloos, zolang de signalen van beide kanten, met dezelfde naam, maar met elkaar worden verbonden.

## 3 Hoe komt een teken van computer tot papier?

In dit hoofdstuk zullen we een aantal zaken onder de loep nemen. Ten eerste, wat is een teken (zoals verstuurd door de computer). Ten tweede, wat gebeurt er met dat teken in de printer. En ten derde, hoe wordt dat teken omgezet in een letter en op papier gezet. De eerste twee vragen kunnen voor zowel de matrix- als voor de daisy wheel printer gezamenlijk worden beantwoord. Voor het beantwoorden van de laatste vraag zal onderscheid moeten worden gemaakt tussen matrix- en daisy wheel printer.

### 3.1 Codes en tekens.

De gegevens, zoals die van de computer naar de printer worden gestuurd, zijn gecodeerd. Zoals u bekend zal zijn, kent de computer alleen het verschil tussen 0 en 1. We hebben echter ook gezien, dat er voor het versturen van een teken een serie nulletjes en eentjes wordt gebruikt. Niet alle computerfabrikanten maken printers en niet alle printerfabrikanten maken computers. Daarom dienen er tussen beide soorten fabrikanten goede afspraken te bestaan, opdat printer en computer beide dezelfde taal spreken, ofwel dezelfde codes gebruiken voor dezelfde tekens.

In hoofdstuk 2 zagen we reeds, dat er voor de fysieke verbinding inderdaad een groot aantal afspraken zijn. Bovendien zagen we dat er nog al wat (kleine) afwijkingen op die afspraken bestaan, met als gevolg dat het aansluiten vaak niet zonder problemen gaat. Is de verbinding echter eenmaal gemaakt, dan moet de printer nog kunnen verstaan wat de computer zegt. De meest gebruikte code is de ASCII-code. Volgens deze code kunnen alle letters van het alfabet, de

cijfers en de leestekens worden omgezet naar een serie van nulletjes en eentjes. Praktisch alle home computers en printers maken van deze code gebruik. In tabel 3-1 is evenwel niet de ASCII-code weergegeven, doch de code die wordt aangeduid met IA-5 (Internationaal Alfabet nummer 5). Deze code komt praktisch overeen met de ASCII-code, doch is een internationaal gestandaardiseerde code, terwijl de ASCII-code in feite alleen een Amerikaanse standaard is.

MSB's bin hex		0000 0	0001 1	0010 2	0011 3	0100 4	0101 5	0110 6	0111 7
LSB's bin hex									
0000	0	NUL		SP					
		0	16	32	48	64	80	96	112
0001	1	SOH	DC1	!	1	A	Q	a	q
		1	17	33	49	65	81	97	113
0010	2	STX	DC2	"	2	B	R	b	r
		2	18	34	50	66	82	98	114
0011	3	ETX	DC3	#	3	C	S	c	s
		3	19	35	51	67	83	99	115
0100	4	EOT	DC4	\$	4	D	T	d	t
		4	20	36	52	68	84	100	116
0101	5	ENQ	NAK	%	5	E	U	e	u
		5	21	37	53	69	85	101	117
0110	6	ACK	SYN	&	6	F	V	f	v
		6	22	38	54	70	86	102	118
0111	7	BEL	ETB	'	7	G	W	g	w
		7	23	39	55	71	87	103	119
1000	8	BS	CAN	(	8	H	X	h	x
		8	24	40	56	72	88	104	120
1001	9	HT	EM	)	9	I	Y	i	y
		9	25	41	57	73	89	105	121
1010	A	LF	SUB	*	:	J	Z	j	z
		10	26	42	58	74	90	106	122
1011	B	VT	ESC	+	;	K	[	k	{
		11	27	43	59	75	91	107	123
1100	C	PF	FS	,	<	L	\	l	:
		12	28	44	60	76	92	108	124
1101	D	CR	GS	-	=	M	]	m	}
		13	29	45	61	77	93	109	125
1110	E	SO	RS	.	>	N	^	n	~
		14	30	46	62	78	94	110	126
1111	F	SI	US	/	?	O	_	o	DEL
		15	31	47	63	79	95	111	127

Tabel 3-1 De ASCII-code tabel (IA5).

Het lezen van de tabel gaat als volgt. Stel dat u wilt weten welke code wordt gebruikt voor de hoofdletter A. U zoekt dan die letter op in de tabel. U leest dan het (hexadecimale) cijfer dat boven de kolom staat en vervolgens het (hexadecimale) cijfer dat links van de rij staat. Voor de letter A zou u dus

eerst de 4 lezen en daarna de 1. Hieruit volgt, dat de hexadecimale code voor de letter A het getal 41 is. Dit is echter een hexadecimaal getal. Zou u willen weten wat de decimale waarde is, dan kunt u dat direct aflezen in de tabel. De decimale code staat namelijk in het kleine rechthoekje rechts onderin ieder vakje van de tabel. Voor de hoofdletter A leest u dus de waarde 65 af.

Behalve de decimale en hexadecimale waarde kunt u ook nog de binaire waarde uit de tabel halen. De binaire waarden staan boven en links van de hexadecimale waarden. Voor de hoofdletter A lezen we dus de binaire waarden 0100 0001.

Uit de tabel kan natuurlijk ook een teken bij een gegeven code worden gezocht. Indien de hexadecimale code van een gezocht teken 5A is, dan zoeken we in de tabel naar de kolom waarboven de 5 staat, en vervolgens dalen we in de kolom af tot we de rij, waar aan de linkerkant de letter (het hexadecimale cijfer) A staat, hebben bereikt. We zien dan dat het gezochte teken de letter Z is.

Bij die binaire waarden zal het u opvallen, dat het meest significante bitje (het meest linkse bitje bovenaan de tabel) steeds een 0 is. Met andere woorden, in de ASCII-code worden slechts 7 bitjes gebruikt voor het coderen van tekens. De meeste moderne home computers en printers gebruiken dat achtste bitje wel. Hierdoor worden buiten de reeds in ASCII gestandaardiseerde 128 tekens, nog eens 128 tekens beschikbaar gemaakt. Deze toegevoegde groep tekens is echter veel minder gestandaardiseerd. Binnen MSX is er een goede standaard, die zowel door MSX-computers als door MSX printers wordt aangehouden. Daarbuiten komen echter een groot aantal verschillende codes voor. De laatste tijd komt er enige standaardisatie voor bij printers die speciaal gemaakt zijn om aan een IBM-PC te worden aangesloten.

In afbeelding 3-1 ziet u de tekensets van twee verschillende printers. Boven in de afbeelding staat de tekenset van een MSX-printer, daaronder de tekenset van een IBM-PC compatibele printer. Beide tekensets zijn afgedrukt op volgorde van



code. U ziet, dat de verschillen vooral in de tweede helft van de tekensets zitten.

⊕	⊗	♥	♦	♣	♠	•	◻	○	◼	♂	♀	♃	♄	♅	
†	‡	⋈	‡	†	†		-	┌	┐	└	┘	/	\	+	
!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
p	q	r	s	t	u	v	w	x	y	z	{		}	~	Δ
Ç	ú	é	à	â	ä	ç	ê	ë	è	é	í	î	ã	Ä	Å
É	æ	Æ	ö	ø	ñ	ú	û	y	ÿ	Û	Ü	½	¼	⅓	℥
á	í	ó	ú	ñ	ñ	ú	û	ÿ	ÿ	Û	Ü	½	¼	⅓	℥
Ä	ä	Y	ÿ	Û	Ü	½	¼	⅓	℥	~	%	‰	¶	§	
←	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
α	β	Γ	π	Σ	σ	μ	τ	Φ	θ	Ω	δ	∞	∅	ε	∩
≡	±	≥	≤	↑	J	÷	≈	°	.	.	√	n	2	.	.

MSX

!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
p	q	r	s	t	u	v	w	x	y	z	{		}	~	
á	í	ó	ú	ñ	ñ	ú	û	ÿ	ÿ	Û	Ü	½	¼	⅓	℥
Ä	ä	Y	ÿ	Û	Ü	½	¼	⅓	℥	~	%	‰	¶	§	
←	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
α	β	Γ	π	Σ	σ	μ	τ	Φ	θ	Ω	δ	∞	∅	ε	∩
≡	±	≥	≤	↑	J	÷	≈	°	.	.	√	n	2	.	.

IBM-compatible

Afb. 3-1 Tekensets van verschillende printers.

Wanneer u een niet-MSX printer wilt gebruiken (misschien zelfs nog aanschaffen), dan doet u er goed aan de MSX-tekenset eens goed te vergelijken met de tekenset van de door u te gebruiken printer. Zitten daar weinig verschillen in, of zijn de verschillen voor uw toepassing niet van belang, dan is er geen bezwaar tegen het gebruiken van die printer. Zijn de verschillen echter van vitaal belang, dan doet u er waarschijnlijk beter aan een andere printer of een MSX-printer te kiezen.

De oplettende lezer zal in de ASCII-tabel (tabel 3-1) nog iets zijn opgevallen. In de twee linker kolomen staan geen afdrukbare tekens, maar afkortingen van, voor de meesten onder u, onbegrijpelijke termen. Hierna volgt een opsomming van al die afkortingen.

0	NUL	niets
1	SOH	Start Of Heading (DC)
2	STX	Start of TeXt (DC)
3	ETX	End of TeXt (DC)
4	EOT	End Of Transmission (DC)
5	ENQ	ENQuiry (DC)
6	ACK	ACKnowledgement (DC)
7	BEL	BEL
8	BS	Back Space
9	HT	Horizontal Tabulation
A	LF	Line Feed
B	VT	Vertical Tabulation
C	FF	Form Feed
D	CR	Carriage Return
E	SO	Shift Out
F	SI	Shift In
10	-	
11	DC1	Device Control 1 (Xon)
12	DC2	Device Control 2
13	DC3	Device Control 3 (Xoff)
14	DC4	Device Control 4
15	NAK	Not AcKnowledged (DC)
16	SYN	SYNchronise (DC)
17	ETB	End of Text Block (DC)
18	CAN	CANcel

19	EM	End of Message (DC)
1A	SUB	SUBstitute character (DC)
<b>1B</b>	<b>ESC</b>	<b>ESCape</b>
1C	FS	File Separator (DC)
1D	GS	Group Separator (DC)
1E	RS	Record Separator (DC)
1F	US	Unit Separator (DC)

Zoals u ziet staan er in deze tabel nogal wat codes, waarachter tussen haakjes DC staat. Deze tekens worden voor het controleren van gegevensoverdracht tussen computers gebruikt. Het gebruik van deze tekens is nogal gecompliceerd. We zullen er daarom niet verder op ingaan. Voor de aansturing van onze printer hebben we die kennis niet nodig.

De vetgedrukte tekens in de lijst zijn speciaal voor printers van belang. Laten we daarom de betekenis van deze codes nu eens nader toe lichten.

### **BEL**

Indien deze code naar de printer wordt gestuurd, zal de printer, mits deze daartoe is uitgerust, een pieptoonje laten horen.

### **BS**

Door deze code naar de printer te sturen, wordt de printkop 1 positie teruggezet. De printkop komt dan dus boven de laatst afgedrukte letter te staan. Hierdoor ontstaat de mogelijkheid om bijvoorbeeld een accent op een letter te plaatsen, door eerst de letter af te drukken, dan een BS-code te versturen en tenslotte het accent af te drukken.

### **HT**

Op typemachines zitten tabulatorstoppen. Door het indrukken van de tabulatorstoets wordt automatisch naar de volgende tabulatorstop gegaan. De code HT heeft hetzelfde gevolg voor de printer. De plaatsen van de tabulatorstoppen zijn printer-afhankelijk.

## **LF**

Met deze code wordt het papier een regel opgeschoven.

## **VT**

Door deze code naar de printer te sturen, wordt 1 of meer regels papier opgeschoven, nadat de nog in het buffer staande tekens zijn afgedrukt. Het aantal regels, dat wordt opgeschoven is op de meeste printers met behulp van een print-commando instelbaar.

## **FF**

Wanneer de printer deze code ontvangt, zal het papier een hele pagina worden opgeschoven. Hoe lang (hoeveel regels) een pagina is, kan meestal met behulp van een aantal DIP-switches in de printer worden ingesteld. Bovendien kan de instelling van die DIP-switches meestal door middel van een print-commando worden overschreven met iedere andere gewenste waarde.

## **CR**

Deze code laat de printkop teruggaan naar het begin van de regel. Indien een aantal tekens naar een printer met een buffer (intern geheugen) zijn gestuurd, zullen die tekens bij ontvangst van de CR-code op het papier worden afgedrukt, voordat de printkop teruggaat naar het begin van de regel. Veel printers hebben een DIP-switch, waarmee kan worden ingesteld, dat de printer, bij ontvangst van een CR-code, het papier automatisch een regel opschuift.

## **SO**

De meeste matrix printers zullen, na ontvangst van deze code, overgaan naar de mode, waarin de letter twee keer zo breed worden afgedrukt. Uiteraard werkt dit alleen voor matrix printers. De letters van daisy wheel printers worden bepaald door het gebruikte letterwiel, en kunnen dus niet plotseling breder worden. Voor de meeste printers geldt, dat deze mode (breed printen) kan worden verlaten door de code DC4, of een line feed commando te geven. Voorbeelden van line feed commando's zijn: LF, VT, FF. Ook geldt voor een aantal printers dat de reeks codes **ESC W n** de printer terug

zet naar de normale tekenbreedte. Voor MSX-printers geldt, dat de mode, die met SO is ingesteld, weer moet worden teruggezet naar normaal met de code SI.

## **SI**

Voor de meeste matrix printers geldt, dat de ontvangst van deze code de printer in de zogenaamde "condensed character mode" zet. Dit wil zeggen dat de tekens smal worden afgedrukt. Om deze mode weer te verlaten dient de code DC2 naar de printer te worden gestuurd. Dit geldt echter niet voor MSX-printers. Voor die printers betekent de code SI, dat de printer de mode, waarin de tekens breed worden geprint, weer moet verlaten.

## **DC1**

De code DC1 wordt ook wel eens aangeduid met de term Xon. Na ontvangst van deze code zal de printer, zo deze in de "off line" (niet gereed) mode is, naar de "on line" mode (gereed ofwel ready) gaan. Op de meeste printers is een lampje aanwezig, waaraan kan worden afgelezen of de printer gereed is, om tekens van de computer te ontvangen. De schakelaar, waarmee de printer ready kan worden gemaakt, heeft dezelfde functie als de code DC1.

## **DC2**

Na ontvangst van deze code zal de printer de mode, waarin de tekens smal worden afgedrukt, verlaten, en weer tekens van normale breedte afdrukken. Meestal geldt, dat deze mode was gezet door de code SI. MSX-printers kennen deze code in het geheel niet.

## **DC3**

De code DC3 wordt ook wel eens aangeduid met de term Xoff. Ontvangst van deze code zal de printer naar de off line mode zetten, ofwel "niet gereed" maken.

## **DC4**

Voor de meeste printers geldt, dat deze code de printer, die op het moment van ontvangst van de code in de mode is, waarin de tekens met dubbele breedte worden afgedrukt

(gezet door SO), weer in de normale mode zet. Dit wil zeggen, dat de tekens na ontvangst van deze code weer de normale breedte zullen hebben. Voor MSX-printers geldt dit niet. In het volgende hoofdstuk zal een voorbeeld voor MSX-printers worden gegeven.

## **CAN**

Na ontvangst van deze code zal de printer alle tekens, die nog in het interne geheugen staan en die nog moeten worden afgedrukt, uit het geheugen wissen. Dit betreft echter alleen afdrubare tekens, en dus niet de eventueel naar de printer gezonden commando-codes. Met andere woorden, de mode waarin de printer werkte voor ontvangst van CAN zal na ontvangst van CAN nog steeds dezelfde zijn.

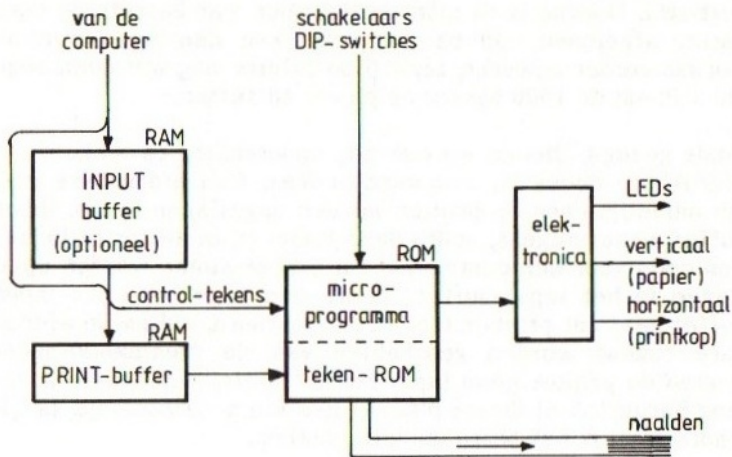
## **ESC**

Moderne printers hebben veel meer mogelijkheden, dan met de zojuist aangegeven codes kunnen worden aangestuurd. De ASCII-code heeft echter niet meer codes vrij voor gebruik door printers. Vandaar dat de ESC-code wordt gebruikt. Na ontvangst van de ESC-code weet de printer dat het volgende teken of de volgende tekens een commando bevat(ten). Dat volgende teken moet dus worden uitgevoerd, in plaats van afgedrukt op papier. In het volgende hoofdstuk zullen we zien hoe dit in zijn werk gaat.

### **3.2 Wat doet de printer met de codes?**

Bij de vraag, wat de printer doet met de ontvangen codes, dienen we onderscheid te maken tussen printers met een "input-buffer" en printers zonder zo'n buffer. Daarbij dienen we niet uit het oog te verliezen, dat iedere printer een zogenaamd "print-buffer" heeft (zie afbeelding 3-2). Bovendien moeten we onderscheid maken tussen normale afdrubare tekens (letters, cijfers en leestekens) en besturingscodes (zoals CR, LF, ESC, etc.).

Iedere printer heeft, zoals gezegd, een print-buffer. In dit buffer, dat niet groter is dan nodig voor het opslaan van alle



Afb. 3-2 Logisch blokdiagram van een printer.

afdrukbare tekens van 1 regel, worden alleen de afdrukbare tekens opgeslagen. Deze tekens blijven opgeslagen, totdat een CR-code wordt ontvangen. Bij ontvangst van de CR-code wordt het printbuffer op papier afgedrukt. Hierna is het printbuffer weer leeg, en kan het met nieuwe tekens worden gevuld.

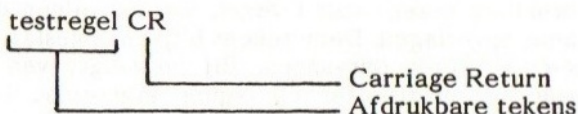
Sommige printers (de laatste tijd zijn dat er steeds meer) hebben naast dit print-buffer ook nog een input-buffer. Dat input-buffer is dan vaak veel groter dan nodig is voor het opslaan van 1 regel. Input-buffers van 2, 4 of 8kBytes zijn gangbaar. Het grote voordeel van deze input-buffers is, dat er een groot aantal regels tegelijkertijd in opgeslagen kunnen worden.

Dit is vooral voordelig, wanneer de gegevensoverdracht naar de printer via een Centronics-interface gaat, omdat over die interface in enkele seconden een heel A4-blad vol tekst kan worden gestuurd. Stel dat u 30 regels van 60 tekens naar de printer stuurt, dan zijn dat in totaal 1800 tekens. Die tekens zullen in ongeveer een seconde naar het input-buffer zijn

gestuurd. Daarna is de print-actie, voor wat betreft de computer, afgelopen. Op de computer kan dan weer normaal worden verder gewerkt, terwijl de printer nog wel even bezig zal zijn om de 1800 tekens op papier te zetten.

Zoals gezegd, dienen we ook nog onderscheid te maken naar afdruckbare codes en commando-codes. Een afdruckbare code zal namelijk door de printer worden opgeslagen in het input-buffer en vervolgens, zodra daar plaats is, in het print-buffer. Een code voor een commando zal echter alleen worden opgeslagen in het input-buffer. Zodra een regel van het input-buffer naar het print-buffer wordt gestuurd, zullen de afdruckbare tekens worden gescheiden van de commando-codes. (Indien de printer geen input-buffer heeft, vindt deze scheiding natuurlijk al direct plaats.) Een klein voorbeeldje zal dit waarschijnlijk het beste duidelijk maken.

Stel dat de volgende tekst naar de printer wordt gestuurd:



Het eerste teken, dat wordt verstuurd is de letter t. Vervolgens de letter e, enz. De afdruckbare tekens worden in het print-buffer opgeslagen. Op een gegeven moment staan alle afdruckbare tekens in het printbuffer. Het volgende teken dat door de printer wordt ontvangen, is de code CR. Het gevolg van deze code is, dat de printer alle tekst in het print-buffer op het papier gaat afdrucken, voordat de printkop wordt teruggezet naar het begin van de regel. De code CR (het commando Carriage Return) wordt dus direct uitgevoerd, terwijl de afdruckbare tekens eerst werden opgeslagen.

### 3.3 Hoe worden de tekens afgedrukt?

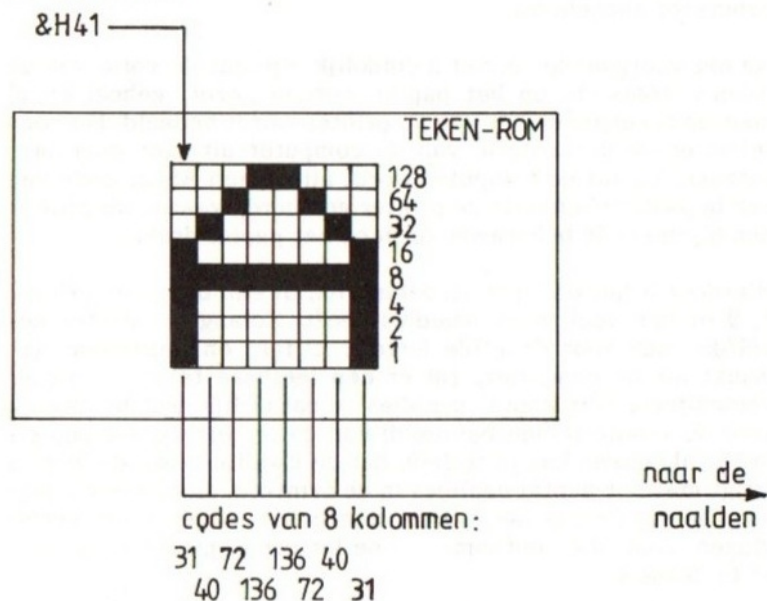
Bij de manier waarop tekens op papier worden afgedrukt, moet onderscheid worden gemaakt tussen matrix printers en



daisy wheel printers. We zullen eerst gaan kijken hoe een en ander in zijn werk gaat bij matrix printers, daarna zullen we de daisy wheel printer aan een nader onderzoek onderwerpen.

In de matrix printer zit een zogenaamde "tekengenerator". Dit is een ROM, waarin het patroon van ieder afdrukbaar teken is opgeslagen. Stel dat een bepaalde matrix printer de tekens afdruckt in een matrix van 8 bij 8 puntjes. In dat geval bestaat ieder afdrukbaar teken uit 8 verticale kolommen van elk 8 puntjes (zie afbeelding 3-3).

ASCII-code voor  
de letter A =



Afb. 3-3 Hoe tekens naar de printkop gaan.

De code van het afdruckbare teken (de ASCII-code) is als het ware het adres voor het blok van 8 kolommen binnen de teken generator. Met andere woorden, de ASCII-code gaat er in, en

het patroon van bitjes komt er uit, steeds met een kolom van 8 bitjes tegelijk. Deze 8 bitjes gaan naar de printkop, waar ze de 8 naalden van de printkop aansturen. Elk bitje dat een 1 is, zal er voor zorgen dat de betreffende naald wordt afgevuurd en een stipje op het papier achter laat. Alle 8 naalden kunnen dus in principe tegelijk worden afgevuurd. Dit zal het geval zijn als er een verticaal streepje moet worden gedrukt. Na het afdrukken van 1 kolom wordt de printkop een heel klein stukje verplaatst. Daarna wordt de volgende kolom vanuit de tekengenerator via de printkop op het papier gedrukt. Dit gaat door, totdat alle 8 kolommen vanuit de tekengenerator op het papier zijn afgedrukt. Daarna wordt de volgende ASCII-code vanuit het print-buffer als adres aan de teken-generator aangeboden.

Na het voorgaande zal het u duidelijk zijn dat de vorm van de tekens, zoals die op het papier worden gezet, geheel en al door de tekengenerator van de printer wordt bepaald. Hoe een teken er op het scherm van de computer uit ziet doet niet terzake. Vanuit de computer wordt alleen een ASCII-code van een bepaald teken naar de printer gestuurd, waarna de printer een bij die code behorende vorm op het papier drukt.

Hierdoor is het ook niet zo belangrijk, of een bepaalde printer 8, 9 of nog veel meer naaldjes heeft. Zolang de printer dezelfde code voor dezelfde letters, cijfers en leestekens gebruikt als de computer, zal er een leesbare tekst op papier verschijnen. Het aantal naaldjes is natuurlijk wel belangrijk voor de kwaliteit (leesbaarheid) van de letters op het papier. In het algemeen kun je stellen, dat de kwaliteit van de letters stijgt met het aantal naaldjes in de printkop. Het meest gangbare aantal naaldjes is 9. Daarmee is het mogelijk om verlagen, zoals die voorkomen in de letters g, j, p, q en y goed af te drukken.

Bijna alle matrix printers hebben de mogelijkheid om, in plaats van via de tekengenerator, direct kolommen vanuit het print-buffer aangeboden te krijgen. Dit wordt aangeduid met de grafische mode. Vanuit de computer sturen we een kolom van 8 bitjes naar de printer (8 is het maximum, omdat een

byte nu eenmaal uit niet meer dan 8 bitjes bestaat), nadat we de printer hebben verteld, dat de volgende codes niet als ASCII-code moeten worden gezien, maar de printer ze direct naar de printkop moet sturen. De grafische mode zullen we uitgebreid behandelen aan de hand van een aantal voorbeelden, in een van de volgende hoofdstukken.

Sommige printers hebben de mogelijkheid om, door de programmeur gedefinieerde tekens, die in een RAM-geheugen in de printer worden opgeslagen, op dezelfde manier af te drukken als tekens uit de tekengenerator. Het speciale RAM-geheugen heeft dan de functie van een tekengenerator. Later in het boek zullen we hierop terugkomen en dit onderwerp aan de hand van een voorbeeldprogramma uitleggen.

De hiervoor beschreven principes gelden voor alle matrix printers. Bij een thermische printer zijn er weliswaar geen naaldjes die moeten worden afgevuurd, doch daar zullen de betreffende plaatsen op de printkop verhit worden en zodoende een puntje op het papier achter laten.

De daisy wheel printer heeft niet eenzelfde soort tekengenerator als de matrix printer. De vormen van de tekens zitten namelijk al op het letterwiel. Toch is er wel enige overeenkomst. Ook de daisy wheel printer ontvangt de ASCII-codes van de computer en drukt een door de printer gedefinieerde vorm van een teken af op het papier. Eigenlijk zou het beter zijn om te zeggen, dat de vorm in dit geval wordt bepaald door het letterwiel dat, in de printer zit.

De ASCII-code wordt door de elektronica in de printer omgezet in een aantal pulsjes, die naar de motor, die het letterwiel moet ronddraaien, worden gestuurd. Het letterwiel wordt daardoor in een bepaalde stand gebracht (gedraaid), met het teken dat bij de gegeven ASCII-code hoort tussen de hamer en het papier. Hierna wordt de hamer geactiveerd en verschijnt het teken op het papier. Nu kan de volgende ASCII-code vanuit het print-buffer worden opgehaald, om er het letterwiel weer mee in een bepaalde stand te zetten, waarna de hamer weer wordt geactiveerd. Dit gaat zo door totdat het hele

print-buffer afgedrukt is op papier.

Het zal u nu duidelijk zijn, dat een daisy wheel printer geen grafisch mode kent. De afdrubare tekens zijn vastgelegd door het gebruikte letterwiel. Ook is het niet mogelijk om zelfgedefinieerde tekens te laten afdrukken. Het is echter wel mogelijk om een ander letterwiel in de printer te zetten en zodoende een andere vorm van de tekens op het papier te krijgen.

Dat er geen grafische mode is op daisy wheel printers wil nog niet zeggen, dat er geen grafisch werk mogelijk is. Dat is wel degelijk mogelijk. In de volgende hoofdstukken zullen we daarvan een voorbeeldprogramma laten zien. Het kan zelfs zo zijn, dat een bepaalde grafiek op de daisy wheel printer net zo vlug wordt gedrukt als op een matrix printer. Dit is dan afhankelijk van het aantal puntjes dat op papier moet worden gezet.

## 4 Het programmeren van printers

Na alle theorie uit de vorige hoofdstukken zult u waarschijnlijk zitten te springen om aan de slag te gaan. Voordat het echter zover is, nog even het volgende.

Vanaf dit moment zal steeds worden begonnen met een voorbeeldprogramma, dat daarna zal worden toegelicht. In die toelichting vindt u dan de uitleg van de specifieke printer besturingstekens, die in het betreffende programma worden gebruikt. De voorbeeldprogramma's zullen steeds op slechts 1 printer zijn getest. Er zal ook worden vermeld, op welke printer het programma is getest. Indien u een andere printer hebt, kan het zijn, dat het programma niet ongewijzigd werkt op uw printer. Om het programma aan uw printer aan te passen, hebt u een aantal mogelijkheden, en wel:

De theorie uit de vorige hoofdstukken heeft u waarschijnlijk voldoende kennis gegeven, om de juiste codes uit de bij uw printer geleverde documentatie te halen en in te vullen in het voorbeeldprogramma.

Aan het einde van dit boek is een tabel gegeven, waarin voor de verschillende functies de codes van een aantal printers zijn terug te vinden. Ook die tabel kunt u eventueel raadplegen.

### 4.1 Carriage Return en Line Feed

Dan zijn we nu eindelijk aan het eerste programma toe. Met dit programma kunt u tekst intikken op de computer, waarna die tekst op de printer wordt afgedrukt. Het programma is geschreven voor en getest op een Seikosha printer, de Seikosha SP800. De gebruikte besturingscodes zijn echter zo algemeen, dat het programma op elke willekeurige printer werkt.

```

10 INPUT "aantal tekens per regel";T
20 CLS:A=0
30 I$=INKEY$:IF I$="" GOTO 30
40 PRINT I$;
50 LPRINT I$;
60 A=A+1
70 IF A=T THEN LPRINT CHR$(13);CHR$(10);
80 IF A=T THEN CLS:A=0
90 GOTO 30

```

Met de regels 50 en 70 worden codes van tekens en commando's naar de printer gestuurd. U ziet op die regels het LPRINT-statement. Door gebruik te maken van het LPRINT-statement, worden de codes automatisch naar de centronics parallelpoort van de computer gestuurd.

Zou men een LPRINT-statement zonder verdere parameters geven, dan zou dat resulteren in het versturen van een Carriage Return (CR) en een Line Feed (LF). Om het versturen van deze codes in eigen hand te houden, doen we er goed aan het LPRINT-statement altijd te laten eindigen met een puntkomma (;). Het gevolg daarvan is, dat er nooit een CR of LF zal worden verstuurd, tenzij wij dat uitdrukkelijk in het LPRINT-statement vermelden.

De code voor CR is de decimale waarde 13. De code voor LF is 10. In regel 70 van het programma ziet u dat die codes met behulp van het LPRINT-statement naar de printer worden gestuurd. Daarbij is gebruik gemaakt van de functie CHR\$(x). Dit is nodig, omdat er voor de codes voor CR en LF geen toets op het toetsenbord zit. Om de code dan toch te kunnen versturen, zetten we die code in de CHR\$-functie tussen de haakjes. Zou het een code van een letter betreffen, dan kunnen we die letter gewoon achter het LPRINT-statement zetten (tussen aanhalingstekens). Ook is het mogelijk om de letters in een stringvariabele te zetten, en deze string-variabele achter het LPRINT-statement te zetten. Moeten er meerdere niet afdrubbare tekens worden afgedrukt, of een combinatie van afdrubbare en niet afdrubbare tekens, dan kunnen die tekens aan elkaar worden geplakt door er een puntkomma of

een plusje tussen te zetten. Hierna volgen enkele voorbeelden van de toegestane manieren van programmeren van het LPRINT-statement:

statement:	resultaat:
LPRINT "abcABC012";	abcABC012
LPRINT "abcABC012"	abcABC012 en vervolgens gaat de printkop terug naar het begin van de regel en wordt het papier een regel opgeschoven.
LPRINT CHR\$(13);	De printkop gaat terug naar het begin van de regel.
LPRINT CHR\$(13); CHR\$(10);	De printkop gaat terug naar het begin van de regel en het papier wordt een regel opgeschoven.
A\$="abcABC012" LPRINT A\$;	abcABC012
A\$="abcABC012" B\$=" ABC " LPRINT A\$;CHR\$(13);B\$;	abcABC012 Na het afdrucken van A\$ gaat de printkop terug naar het begin van de regel en wordt B\$ over A\$ heen afgedrukt.
LPRINT "abc"+"ABC";	abcABC
LPRINT CHR\$(&H41);	A
LPRINT CHR\$(13)+CHR\$(10);	De printkop gaat terug naar het begin van de regel en het papier wordt een regel opgeschoven.

In voorgaande voorbeelden is bijna altijd de decimale notatie gebruikt. Het is echter ook toegestaan de hexadecimale notatie te gebruiken. In dat geval zou bijvoorbeeld CHR\$(13) moeten worden geschreven als CHR\$(&HD) en CHR\$(10) als CHR\$(&HA).

Om zelf volledige controle te houden over het al of niet teruggaan van de printkop en het opschuiven van het papier, moet de printer zo zijn ingesteld, dat deze niet automatisch een regelopschuif uitvoert bij ontvangst van de CR-code. In de meeste printers zit een schakelaartje waarmee deze instelling kan worden gewijzigd. Die schakelaar wordt meestal aangegeven met de tekst "AUTOFEEDXT", daarde schakelaar dezelfde functie heeft als het Centronics signaal AUTOFEEDXT.

U merkt echter gauw genoeg, of de schakelaar in de door u gewenste stand staat, als u de zojuist gegeven commando's uitvoert. Vooral het commando LPRINT "A";CHR\$(13);"A"; is een goede test. Krijgt u daarmee twee regels onder elkaar, dan staat de schakelaar AUTOFEEDXT aan. Nadat u deze schakelaar in de andere stand hebt gezet, zal er nog maar 1 regel worden afgedrukt.

## 4.2 Horizontale tabulatiestoppen.

Het volgende programma maakt het mogelijk, gegevens netjes in kolommen onder elkaar af te drukken. Hiertoe wordt de code voor Horizontale Tabulatie (HT) gebruikt. Ook dit programma is op de Seikosha SP800 getest.

```
10 LPRINT "01234567890123456"  
20 CLS  
30 INPUT "Eerste letter";A$  
40 INPUT "Tweede letter";B$  
50 LPRINT CHR$(9);  
60 LPRINT A$;  
70 LPRINT CHR$(9);
```



```
80 LPRINT B$;  
90 LPRINT CHR$(13);CHR$(10);  
100 GOTO 20
```

Met regelnummer 10 wordt als het ware een positiebalk afgedrukt. Daarna worden twee alfanumerieke gegevens opgevraagd en in de variabelen A\$ en B\$ gezet. Op regel 50 wordt de code 9 (code voor HT) naar de printer gestuurd. Hierdoor zal het eerstvolgende afdruckbare teken op de eerstvolgende tabulatorstop worden afgedrukt. Na het aanschakelen van de Seikosha-printer is elke achtste tekenpositie een tabulatorstop. Het eerste teken uit A\$ zal dus op de achtste positie worden afgedrukt. Nu wordt met regel 70 naar de volgende tabulatorstop gepositioneerd. Die stop ligt weer 8 posities verderop. Vanaf positie 16 zal dan ook de inhoud van variabele B\$ worden afgedrukt. Een en ander is in het voorbeeld van de output van het programma te zien.

Met regelnummer 90 wordt de printkop naar het begin van de volgende regel gestuurd (CR en LF). Hierna worden de volgende gegevens opgevraagd en begint het geheel weer opnieuw. U ziet dat dit nogal eenvoudig is. Later zullen we nog terugkomen op het tabuleren. Op een aantal printers is het namelijk ook mogelijk om de plaats van de tabulatorstoppen te wijzigen.

Het nu volgende programma doet precies hetzelfde als het vorige programma. Het enige verschil zit hem daarin, dat in dit programma het statement PRINT # is gebruikt in plaats van het LPRINT-statement.

```
10 OPEN "LPT:" AS #1  
20 PRINT #1,"01234567890123456"  
30 CLS  
40 INPUT "Eerste letter";A$  
50 INPUT "Tweede letter";B$  
60 PRINT #1,CHR$(9);  
70 PRINT #1,A$;  
80 PRINT #1,CHR$(9);
```

```
90 PRINT #1,B$;  
100 PRINT #1,CHR$(13);CHR$(10);  
110 GOTO 30
```

Om het statement PRINT # te kunnen gebruiken, moeten we eerst een bestand op de printer openen. Dit doen we met het OPEN-statement op regel 10. Zolang we de systeemvariabele MAXFILES niet hebben gewijzigd, staat deze op de waarde 1. We kunnen dan alleen bestandsnummer 1 gebruiken. Zou u, omdat u bijvoorbeeld nummer 1 al hebt gebruikt voor het grafisch scherm, bestandsnummer 2 willen gebruiken, dan zult u voor het OPEN-statement nog de volgende regel moeten toevoegen:

```
5 MAXFILES=2
```

Nadat het bestand op de printer een nummer heeft gekregen (met behulp van het OPEN-statement) kunnen we het statement PRINT # gaan gebruiken. Dit ziet u in de regels 20, 60, 70, 80, 90 en 100. Als u goed kijkt, ziet u dat er in die regels niets anders is gedaan dan het vervangen van LPRINT door PRINT #1,.

Zoals in het begin van dit hoofdstuk al kort werd gezegd, kan het LPRINT-statement (of het PRINT #n statement) ook zonder parameters, ofwel zonder een puntkomma aan het einde, worden gegeven. U ziet dit in regel 20. Het gevolg van het niet plaatsen van de puntkomma is, dat er automatisch een CR en een LF code naar de printer worden gestuurd.

Net als bij het programma, aan de hand waarvan CR en LF werden toegelicht, geldt ook voor HT, dat deze code voor praktisch alle printers gelijk is. Alleen de plaats van de tabulatiestoppen kan van printer tot printer variëren. Dat merkt u echter gauw genoeg, bij het uitvoeren van een van voorgaande programma's.

### 4.3 Normale, smalle en brede tekens.

Deze paragraaf geldt alleen voor matrix printers. Met matrix printers kunnen de tekens namelijk op een aantal manieren worden afgedrukt. Drie van deze manieren zullen we met behulp van het volgende programma zien:

```
100 KEY OFF
110 FOR I=1 TO 10
120 KEY(I) OFF
130 NEXT I
140 SCREEN 0: WIDTH 40: CLS
150 KEY 1,"breed"
160 KEY 2,"normaal"
170 KEY 3,"smal"
180 KEY 4,"normaal"
190 KEY ON
200 ON KEY GOSUB 290,320,350,380
210 FOR I=1 TO 4
220 KEY(I) ON
230 NEXT I
240 I$=INKEY$: IF I$="" GOTO 240
250 IF I$=CHR$(13) THEN PRINT: LPRINT: G
OTO 240
260 PRINT I$;
270 LPRINT I$;
280 GOTO 240
290 PRINT "-breed-";
300 LPRINT CHR$(14);
310 RETURN
320 PRINT "-normaal-";
330 LPRINT CHR$(20);
340 RETURN
350 PRINT "-smal-";
360 LPRINT CHR$(15);
370 RETURN
380 PRINT "-normaal-";
390 LPRINT CHR$(18);
400 RETURN
```

Voorbeeld van het resultaat van dit programma:

Dit is een regel normale tekst.

**Dit is breed.**

Deze regel is smal geprint.

Nu printen we **breed** en *smal* door elkaar.

Na het aanschakelen van de printer zullen de tekens worden afgedrukt met een normale breedte. Deze breedte is vaak in te stellen met behulp van een aantal schakelaartjes in de printer (DIP-switches). De meest gebruikelijke instelling is 10 tekens per inch, hetgeen vaak ook wel wordt aangeduid met de term PICA-tekenmode. Vaak is er keuze tussen PICA, ELITE en PROPORTIONEEL.

PICA wil zeggen, dat er 10 tekens per inch worden afgedrukt. ELITE wil zeggen, dat er 12 tekens per inch worden afgedrukt. PROPORTIONEEL wil zeggen, dat de tekens ieder een eigen breedte hebben. Ieder teken zal zoveel plaats innemen, als nodig is, om het teken duidelijk af te drukken, zonder een gedrongen schrift te krijgen. De letter M is dus veel breder dan de letter I. Het gevolg van proportioneel schrift is, dat de letters niet meer recht onder elkaar staan. Sommige regels zullen 90 tekens bevatten, terwijl andere regels maar 70 tekens bevatten. Het aantal tekens per regel (of per inch) is afhankelijk van de afgedrukte tekens.

Onafhankelijk van de ingestelde tekendichtheid, kan de printer door middel van een aantal eenvoudige commando's worden opgedragen de tekens breder of smaller af te drukken. Dit wordt met het voorgaande programma aangetoond. Om deze functies van de printer eenvoudig te kunnen testen, zijn in dat programma nieuwe functies aan de functietoetsen 1 tot en met 4 toegekend. Hiertoe dienen de regels 100 tot en met 230.

De regels 240 tot en met 280 accepteren tekens, die op het beeldscherm worden afgedrukt en naar de printer worden

gestuurd. In plaats van een afdrukbaar teken kunt u ook een functietoets indrukken. Drukt u functietoets 1 in, dan wordt de subroutine van regel 500 gestart. Hiermee wordt op het beeldscherm de tekst -breed- afgedrukt, waarna de code SO (CHR\$(14)) naar de printer wordt gestuurd. Deze code heeft tot gevolg, dat alle volgende tekens twee keer zo breed worden afgedrukt. Dit gaat door, totdat een CR-code wordt gegeven, of totdat de code DC4 (CHR\$(20)) wordt gegeven. De code DC4 kan worden gegeven door op functietoets F2 te drukken. Alle daarna in te tikken tekens zullen weer op normale breedte worden afgedrukt. Het voorbeeld, na de listing, laat enkele mogelijkheden van het programma zien.

#### 4.4 Diverse printer commando's.

Naast de, in de vorige paragrafen behandelde, printer commando's zijn er nog een aantal commando's, die uit slechts 1 byte bestaan. Van die overige commando's zal geen voorbeeldprogramma worden gegeven. Wel volgt hier de manier, waarop ze in een programma kunnen worden opgenomen.

LPRINT CHR\$(7)	- Bel
LPRINT CHR\$(8)	- Back Space
LPRINT CHR\$(11)	- Verticale tabulatie
LPRINT CHR\$(12)	- Form Feed
LPRINT CHR\$(24)	- Cancel

De functies van de hiervoor genoemde commando's zijn al in hoofdstuk 3 beschreven. We zullen die dan ook niet herhalen. Waar nodig, zullen we nog terugkomen op deze codes. In het volgende hoofdstuk zullen we nader ingaan op de zogenaamde "Escape sequences", die we hierna zullen aanduiden met ESC-reeksen. ESC-reeksen zijn printer commando's, die uit een ESC-code, gevolgd door 1 of meer bytes, bestaan.

## 5 Programmeren met ESC-reeksen

Daar de meeste printers veel meer mogelijkheden hebben, dan er in de ASCII-code tabel zijn gedefinieerd, is er een uitvluchtmogelijkheid geschapen. Het Engelse woord voor uitvlucht is Escape. Dit is terug te vinden in de ASCII-code tabel onder de naam ESC. De waarde, die bij die code hoort is decimaal 27 ofwel hexadecimaal &H1B. Steeds wanneer de code CHR\$(27) naar de printer wordt gestuurd, verwacht deze dat de daaropvolgende tekens een commando zullen zijn.

Vaak zal de ESC-code door slechts 1 code worden gevolgd. Het is echter ook mogelijk dat de ESC-code door meerdere codes wordt gevolgd. Het aantal codes en de betekenis daarvan is afhankelijk van het eerste teken dat volgt op de ESC-code. Dit zal waarschijnlijk het best kunnen worden aange-toond met behulp van een praktisch voorbeeld.

```
LPRINT CHR$(27);CHR$(14);
```

De code 14, onmiddellijk volgend op de code 27 (ESC), laat de printer weten, dat er slechts 1 code volgt op de ESC-code, en dat die code (14) moet worden uitgevoerd. De printer weet, dat alle tekens, die volgen op de code 14, afgedrukt moeten worden. Alleen de code 14 moet worden uitgevoerd. De code 14 is de code voor Shift Out (SO). Deze code zal er voor zorgen dat de printer de volgende tekens breed zal afdrukken. In feite is het gevolg van deze code dus precies gelijk aan de code 14 zonder voorafgaande ESC-code.

```
LPRINT CHR$(27);CHR$(87);CHR$(1);
```

De code 87, onmiddellijk volgend op de code 27 (ESC), laat de printer weten, dat 87 een commando is, dat nog wordt ge-

volgd door een parameter. Het commando (87), gevolgd door de parameter 1, vertelt de printer dat de tekens, volgend op de ESC-reeks, met dubbele breedte moeten worden afgedrukt. Dit komt bijna overeen met het SO-commando. Het verschil is echter, dat na het instellen van breed afdrucken met deze ESC-reeks, de CR-code er niet voor zorgt, dat er weer op normale breedte wordt afgedrukt. De printer blijft net zo lang in de mode waarin de tekens met dubbele breedte worden afgedrukt, totdat de ESC-reeks CHR\$(27); CHR\$(87);CHR\$(0); wordt gegeven.

U ziet, dat de ESC-reeks uit het eerste voorbeeld uit een ESC-code gevolgd door 1 code bestaat, terwijl de ESC-reeks uit het tweede voorbeeld uit een ESC-code gevolgd door 2 codes bestaat. Beide voorbeelden gelden voor de Seikosha SP800 printer. Voor andere printers kunnen de functies van deze codes enigszins afwijken. Sommige printers zullen deze functies in het geheel niet kennen en sommige printers zullen deze functies wel hebben, doch gebruiken daarvoor totaal andere codes.

De code CHR\$(87) is de code voor een afdrukbaar teken, namelijk de hoofdletter W. Zodra er sprake is van een afdrukbaar teken, mag, in plaats van bijvoorbeeld CHR\$(87), ook de betreffende letter worden geschreven (in ons voorbeeld de hoofdletter W), zolang deze letter maar tussen aanhalingstekens wordt geplaatst. De ESC-reeks uit het tweede voorbeeld gaat er dan als volgt uit zien:

```
LPRINT CHR$(27);"W";CHR$(1);
```

Dit is niets anders dan een andere schrijfwijze, die is toegestaan, doch die niet verplicht is. Daar deze schrijfwijze echter korter is, zullen de meeste programmeurs hiervoor kiezen.

## **5.1 Geprogrammeerde tabulatiestoppen.**

In hoofdstuk 4 werd een voorbeeld gegeven van het gebruik

van de in de printer aanwezige tabulatiestoppen. Daar werd al opgemerkt, dat het op een aantal printers mogelijk is, de plaats van de tabulatiestoppen zelf te bepalen. Het volgende programma laat hiervan een voorbeeld zien. Het programma is geschreven voor en getest op een Seikosha SP1000MX printer. Dit is een MSX-compatibele printer, met andere woorden, deze printer voldoet aan de MSX-standaard voor printers. Straks zullen we nader ingaan op die standaard.

```

100 '*****
110 '* RESET OUDE TABULATIESTOPPEN *
120 '* SET NIEUWE TABULATIESTOPPEN *
130 '*****
140 '
150 LPRINT CHR$(27);"2";
160 LPRINT CHR$(27);"(005,015,029."
170 '
180 '*****
190 '* HOOFDPROGRAMMA MET TABULEREN *
200 '*****
210 '
220 LPRINT "0123456789012345678901234567
8901234567890"
230 INPUT "WELK GETAL";G
240 IF G<1 OR G>999.9 THEN GOTO 230
250 G1=G: G2=G^2: G3=SQR(G)
260 LPRINT CHR$(9);
270 LPRINT USING "###.##";G1;
280 LPRINT CHR$(9);
290 LPRINT USING "#####.##";G2;
300 LPRINT CHR$(9);
310 LPRINT USING "##.#####";G3
320 GOTO 230

```

Voorbeeld van het programmaresultaat:



01234567890123456789012345678901234567890		
999.9	999800.01	31.621195
4.0	16.00	2.000000
64.0	4096.00	8.000000
128.0	16384.00	11.313708
625.0	390625.00	25.000000
23.0	529.00	4.795832
7.0	49.00	2.645751

Het programma berekent het kwadraat en de vierkantswortel van het getal, dat met behulp van het INPUT-statement op regel 230, wordt ingegeven. Vervolgens worden het getal zelf, het berekende kwadraat en de vierkantswortel met de regels 260 tot en met 310 afgedrukt. Op de regels 260, 280 en 300 ziet u, dat de HT-code (CHR\$(9)) naar de printer wordt gestuurd. Zonder verdere maatregelen zouden de resultaten echter niet netjes onder elkaar worden geprint, omdat sommige resultaten voorbij de eerstvolgendestandaard tabulatiestop zouden gaan.

Om alle resultaten toch netjes in een kolom te krijgen, zullen we nieuwe tabulatiestoppen moeten programmeren. Doch, voordat we nieuwe tabulatiestoppen programmeren, doen we er goed aan de oude tabulatiestoppen eerst te verwijderen. Met regel 150 wordt een commando naar de printer gestuurd, waarmee alle op dat moment actieve tabulatiestoppen worden gereset. Hierna worden met regel 160 drie nieuwe tabulatiestoppen gezet, en wel op de posities 5, 15 en 29.

Volgens de MSX-printer standaard is het echter niet toegestaan om alleen het cijfer 5 te geven (de eerste tabulatiestop). Iedere stop moet met behulp van drie cijfers worden aangegeven. Vandaar dat er in regel 160 het getal 005 staat in plaats van 5. Het "haakje-openen" geeft aan dat het hier het tabulatiestoppen zetten commando betreft. De reeks van tabulatiestoppen moet worden afgesloten met een punt (.). De gehele reeks dient tussen aanhalingstekens te worden gezet. Later in dit hoofdstuk zullen we zien hoe we, in plaats van vaste waarden, ook variabelen in de reeks kunnen opnemen.

Als laatste opmerking over dit programma, wil ik nog even wijzen op de regels 270, 290 en 310. In die regels is het LPRINT USING formaat gebruikt. Door de toevoeging USING te gebruiken kunnen de resultaten netjes worden uitgelijnd, met de decimale punten onder elkaar. Het gebruik van USING is bij LPRINT aan dezelfde regels gebonden als bij PRINT (naar het beeldscherm).

## 5.2 De MSX printer standaard

Binnen de MSX-standaard is ook aandacht geschonken aan de voorwaarden, waaraan printers moeten voldoen, willen deze het predicaat MSX-printer mogen dragen. Deze voorwaarden zijn in twee delen opgesplitst. Het eerste deel bevat de minimum eisen waaraan een printer moet voldoen. Het tweede deel bevat een groot aantal control-reeksen, en wel alle control-reeksen, zoals die ook voor de PC-8023 printer gelden. De MSX printer standaard vindt u achter in dit boek in de appendix. Naast deze voorwaarden moet een MSX-printer ook beschikken over de MSX-tekenset.

De MSX-tekenset wordt met het volgende programma afgedrukt. De oplettende lezer zal echter opmerken, dat er nog een aantal tekens ontbreken.

```
10 SCREEN , , , , 0
20 FOR J=2 TO 15
30 FOR I=0 TO 15
40 LPRINT CHR$(I+16*J); " ";
50 NEXT I
60 LPRINT
70 NEXT J
80 END
```

```
! " # $ % & ' ( ) * + , - . /
0 1 2 3 4 5 6 7 8 9 : ; < = > ?
@ A B C D E F G H I J K L M N O
```

P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_	o
`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	~	Δ
p	q	r	s	t	u	v	w	x	y	z	{		}	~	Δ	Δ
Ç	ú	é	â	ä	à	á	ç	ê	ë	è	í	î	ï	Ä	Å	Å
É	æ	Æ	ô	ö	ò	ó	ù	ý	ö	ü	ç	£	¥	R	f	f
á	í	ó	ú	ñ	Ñ	æ	ó	¿	¡	¡	½	¼	;	«	»	»
Ä	ä	ÿ	ÿ	ö	ö	ü	ü	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ
■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
◀	⌘	⌘	■	■	■	■	⊗	Δ	‡	ω	■	■	■	■	■	■
α	β	Γ	π	Σ	σ	μ	τ	ϕ	θ	Ω	δ	∞	∅	Ε	Π	Π
≡	±	≥	≤	∫	J	÷	≈	°	.	.	√	n	z	■	■	■

Met regel 40 worden achtereenvolgens alle codes van 32 tot en met 255 naar de printer gestuurd. De printer drukt deze tekens vervolgens af. Regel 10 komt u misschien vreemd voor. Daarin wordt de vijfde parameter van het SCREEN-statement op 0 gezet. Dit wil zeggen, dat MSX-BASIC aanneemt dat de aangesloten printer een MSX-printer is. Straks zullen we zien wat dit tot gevolg heeft.

De eerste twee kolommen van de ASCII-tabel bevatten printer- en DC-controle tekens. Deze tekens zijn niet afdruktbaar. Vandaar, dat deze twee kolommen in het voorbeeldprogramma niet zijn afgedrukt.

Het volgende programma drukt de tekenset van een **niet** MSX-printer af. Het betreft hier een Seikosha SP800 printer. In deze printer bevatten de kolommen 8 en 9 printer controle codes, die niet afdruktbaar zijn. Vandaar dat het programma iets is aangepast, ten opzichte van het vorige programma.

```

10 SCREEN ,,,,1
20 FOR J=2 TO 7
30 FOR I=0 TO 15
40 LPRINT CHR$(I+16*J);" ";
50 NEXT I
60 LPRINT
70 NEXT J
72 LPRINT

```

```

74 LPRINT
80 FOR J=10 TO 15
90 FOR I=0 TO 15
100 LPRINT CHR$(I+16*J); " ";
110 NEXT I
120 LPRINT
130 NEXT J
140 END

```

```

! " # $ % & ' ( ) * + , - . /
0 1 2 3 4 5 6 7 8 9 : ; < = > ?
@ A B C D E F G H I J K L M N O
P Q R S T U V W X Y Z [ \ ] ^ _
` a b c d e f g h i j k l m n o
p q r s t u v w x y z { | } ~

```

```

á î ó û ü ñ ñ ã ö ç ÿ ½ ¼ ¼ ¡ « »
⌈ ⌋ ⌌ ⌍ ⌎ ⌏ ⌐ ⌑ ⌒ ⌓ ⌔ ⌕ ⌖ ⌗ ⌘ ⌙ ⌚ ⌛ ⌜ ⌝ ⌞ ⌟ ⌠ ⌡ ⌢ ⌣ ⌤ ⌥ ⌦ ⌧ ⌨ 〈 〉 ⌫ ⌬ ⌭ ⌮ ⌯ ⌰ ⌱ ⌲ ⌳ ⌴ ⌵ ⌶ ⌷ ⌸ ⌹ ⌺ ⌻ ⌼ ⌽ ⌾ ⌿ Ⓚ Ⓛ Ⓜ Ⓨ Ⓩ ⓐ ⓑ ⓓ ⓔ ⓖ ⓗ ⓙ ⓚ ⓛ ⓞ ⓟ ⓠ ⓡ ⓢ ⓣ ⓤ ⓥ ⓦ ⓧ ⓨ ⓩ ⓪ ⓫ ⓬ ⓭ ⓮ ⓯ ⓰ ⓱ ⓲ ⓳ ⓴ ⓵ ⓶ ⓷ ⓸ ⓹ ⓺ ⓻ ⓼ ⓽ ⓾ ⓿ Ⓚ Ⓛ Ⓜ Ⓨ Ⓩ ⓐ ⓑ ⓓ ⓔ ⓖ ⓗ ⓙ ⓚ ⓛ ⓞ ⓟ ⓠ ⓡ ⓢ ⓣ ⓤ ⓥ ⓦ ⓧ ⓨ ⓩ ⓪ ⓫ ⓬ ⓭ ⓮ ⓯ ⓰ ⓱ ⓲ ⓳ ⓴ ⓵ ⓶ ⓷ ⓸ ⓹ ⓺ ⓻ ⓼ ⓽ ⓾ ⓿
α β γ π Σ σ μ τ ϖ θ Ω δ ∞ ∅ ε ∩
≡ ± ≥ ≤ ∫ ∫ ÷ ≈ ° · - √ n z ■

```

Ook in dit programma wordt gestart met het SCREEN-statement. Nu wordt de vijfde parameter van het SCREEN-statement echter op 1 gezet. Dit geeft aan MSX-BASIC aan, dat er een printer is aangesloten, die niet aan de MSX-standaard voldoet. Als u de tekensets uit de zojuist gegeven voorbeelden bekijkt, zult u opmerken, dat die voor een groot deel overeenkomen, terwijl de verschillen zich beperken tot de kolommen 8, 9, 11, 12 en 13.

De MSX-tekenset bestaat echter uit nog meer tekens, dan er in het eerste voorbeeld al werden afgedrukt. Om die tekens af te kunnen drukken moet een speciale code naar de printer

worden gestuurd. Dit ziet u in het volgende voorbeeld.

```
10 SCREEN ,,,,0
20 FOR J=0 TO 1
30 FOR I=0 TO 15
40 LPRINT CHR$(1);CHR$(I+16*J+64);" ";
50 NEXT I
60 LPRINT
70 NEXT J
80 END
```

☺ ☻ ♥ ♦ ♣ · ■ ○ ◻ ♂ ♀ ♪ ♫ \*  
† ‡ ▴ † ‡ † | - ▭ ▮ † ‡ × / \ +

Met regel 40 wordt eerst een code 1 (SOH uit de ASCII-tabel) naar de printer gestuurd, direct gevolgd door de code van het gewenste teken. Deze code mag variëren tussen 64 en 95 (&H40 - &H5F). U ziet nu, dat de tekens, die in de MSX-standaard tekenset in de kolommen 0 en 1 zijn gegeven, worden afgedrukt. Ook in dit programma is de vijfde parameter van het SCREEN-statement op 0 gezet, om aan te geven, dat er een MSX-printer is aangesloten.

MSX-BASIC weet nu, dat er een MSX-printer is aangesloten. Hierdoor zal de string CHR\$(1);CHR\$(65); met een LPRINT-statement ongewijzigd naar de printer worden gestuurd. Zou de vijfde parameter van het SCREEN-statement aangeven, dat er een niet MSX-printer is aangesloten, dan zal de string CHR\$(1);CHR\$(65); door MSX-BASIC worden vervangen door CHR\$(0);. In dat geval zal dus een zogenaamde NULL-string naar de printer worden gestuurd. De meeste printers reageren hierop met het verplaatsen van de printkop, over 1 tekenpositie (alsof er een spatie wordt geprint). Dit is de enige functie van de vijfde parameter van het SCREEN-statement. U kunt dit gemakkelijk testen, door het SCREEN-statement in het laatste voorbeeld te wijzigen in: SCREEN ,,,,1.

### 5.3

### Programma's en files afdrukken

Als u uw printer al eens hebt gebruikt, om listings van BASIC-programma's af te drukken, dan zal het u waarschijnlijk zijn opgevallen, dat de regels op het papier veel langer kunnen zijn dan ze op het beeldscherm zijn. Soms zijn ze zo lang, dat ze van het papier aflopen. Dit is erg hinderlijk, doch er is een simpele oplossing voor. Het volgende programma geeft die oplossing. Dit programma is geschreven voor gebruik met een MSX-printer. Het werd getest op een Seikosha SP1000MX printer.

```
100 '*****
110 '* INITIALISATIE PRINTER *
120 '*****
130 '
140 INPUT "Linker kantlijnpositie      ";
LK
150 IF LK>80 THEN GOTO 140
160 INPUT "Aantal tekens per regel    ";
TR
170 IF TR+LK>120 OR TR<2 THEN GOTO 140
180 RK=LK+TR
190 L1$=STR$(LK):L1=LEN(L1$)-1:LK$=RIGHT
$("000"+RIGHT$(L1$,L1),3)
200 R1$=STR$(RK):R1=LEN(R1$)-1:RK$=RIGHT
$("000"+RIGHT$(R1$,R1),3)
210 PRINT "ZET DE PRINTER GEREED!!
        DRUK DAARNA OP RETURN!!"
220 IF INKEY$<>CHR$(13) THEN GOTO 220
230 LPRINT CHR$(27);"@";
240 LPRINT CHR$(27);"L"+LK$;
250 LPRINT CHR$(27);"/"+RK$;
260 '
270 '*****
280 '* HOOFDPROGRAMMA *
290 '*****
300 '
310 SCREEN 0:WIDTH 40:KEY OFF:FILES:X=0:
```

```

Y=0
320 PRINT:PRINT:PRINT"Wijs de gewenste f
ile aan met de cursor.Druk vervolgens op
RETURN."
330 LOCATE X,Y,1:I$=INKEY$
340 IF I$=CHR$(30) THEN Y=Y-1:IF Y<0 THE
N Y=0
350 IF I$=CHR$(28) THEN X=X+13:IF X>26 T
HEN X=0:Y=Y+1:IF Y>20 THEN Y=20
360 IF I$=CHR$(31) THEN Y=Y+1:IF Y>20 TH
EN Y=20
370 IF I$=CHR$(29) THEN X=X-13:IF X<0 TH
EN X=26:Y=Y-1:IF Y<0 THEN Y=0
380 IF I$<>CHR$(13) THEN 330
390 LOCATE 0,23,0:F$=""
400 FOR I=0 TO 11
410 F$=F$+CHR$(VPEEK(BASE(0)+X+I+Y*40))
420 NEXT I
430 OPEN "a:"+F$ FOR INPUT AS #1
440 IF EOF(1)=-1 THEN GOTO 480
450 LINE INPUT #1,R$
460 LPRINT R$
470 GOTO 440
480 CLS: KEY ON
490 CLOSE #1
500 END

```

We kunnen de printer opgeven waar de linker- en de rechter kantlijn van de af te drukken tekst zich moeten bevinden. Hebben we dit eenmaal aan de printer laten weten, dan zal alle tekst, die daarna naar de printer wordt gestuurd, tussen die kantlijnen worden afgedrukt.

Als we een BASIC-programma vanuit het geheugen, als een ASCII-file naar schijf of cassette schrijven, dan kunnen we die file later openen en er records (=BASIC-regels) uitlezen. Door die regels naar de printer te sturen, nadat die printer is verteld waar de kantlijnen zich moeten bevinden, zal de BASIC-regel tussen de gegeven kantlijnen worden afgedrukt.

Zouden we bijvoorbeeld voor de linker kantlijn positie 11 opgeven, en voor de rechter kantlijn positie 50, dan kunnen er precies 40 tekens tussen de kantlijnen worden afgedrukt. Dit komt overeen met de maximale regellengte op het beeldscherm.

Met regel 240 wordt de linker kantlijn gezet. Met regel 250 wordt de rechter kantlijn gezet. In die regels ziet u dat er met variabelen wordt gewerkt. Die variabelen worden gevuld in de regels 140 tot en met 200. Dat vullen van variabelen zit nogal ingewikkeld in elkaar. De reden daarvan is, dat de waarden van de linker- en rechter kantlijnpositie beslist uit drie cijfers moeten bestaan. Zouden we de linker kantlijn op positie 0 willen hebben, dan zou het LPRINT-statement er als volgt uit moeten zien:

```
LPRINT CHR$(27);"L000";
```

Net als bij het programmeren van de tabulatiestoppen, willen we ook hier liever geen vaste waarde. We zullen dus de cijfers 000 moeten vervangen door een variabele. Hoe dit in zijn werk gaat is in het voorbeeldprogramma in regel 240 te zien. Daarbij dienen we er echter wel op toe te zien, dat de inhoud van de variabele uit precies drie cijfers bestaat, zonder voorafgaande spaties. Vandaar, dat het opvragen van de variabele waarden voor de kantlijnen nogal ingewikkeld is.

Met de regels 140 tot en met 180 worden de waarden opgevraagd en wordt er gecontroleerd of de ingegeven waarden wel binnen redelijke grenzen liggen. Een belangrijk punt hierbij is, dat de regellengte minimaal 2 moet zijn. Is deze korter, dan wordt de controlereeks voor de rechter kantlijn als niet gegeven beschouwd en zal de rechter kantlijn blijven wat hij was. Regels 190 en 200 maken tenslotte van de numerieke variabelen LK en RK alfanumerieke variabelen (LK\$ en RK\$). De inhoud van deze alfanumerieke variabelen zal altijd uit drie cijfers bestaan, ook als er slechts 1 cijfer werd ingegeven bij het INPUT-statement. In dat geval wordt dat cijfer namelijk aangevuld met nullen.



Regel 230 bevat het **reset**-commando. Na ontvangst van dit commando zal de printer naar dezelfde mode gaan, waarin deze ook terecht komt na het aanschakelen. Het is een goed idee, om ieder programma, dat gebruik maakt van de printer, te laten beginnen met het resetten van die printer. Voordat u iets naar de printer stuurt, doet u er goed aan deze eerst aan te zetten. Om niet te vergeten dat de printer moet worden aangezet, kunt u een bericht op het beeldscherm zetten, zoals dit ook is gedaan in het voorbeeld op regels 210 en 220.

Nadat de kantlijnen op de printer zijn ingesteld, zijn we gereed om er tekst naartoe te sturen. Dit zou een BASIC-programma kunnen zijn, dat in het geheugen staat. In het voorbeeldprogramma is er echter vanuit gegaan, dat dit programma het afdrukprogramma is, waarmee andere programma's of files kunnen worden afgedrukt. Het programma is geschreven voor gebruik van een schijfveeneenheid. De regels 310 tot en met 420 stellen u in staat om een ASCII-file, die op de schijf staat, aan te wijzen. Deze ASCII-file (dit kan een bestand zijn of een BASIC-programma dat met het statement `SAVE "xxx",A` is weggeschreven) wordt met de regels 430 tot en met 500 naar de printer gestuurd.

Met regel 430 wordt de ASCII-file geopend. Regel 440 controleert of het einde van de file is bereikt, en indien dit zo is, wordt naar regel 480 gesprongen. De regels 480 tot en met 500 sluiten het programma netjes af. Met regel 450 wordt, indien het einde van de file nog niet is bereikt, een record uit die file gelezen. Indien de file een BASIC-programma is, zal zo'n record uit een complete BASIC-regel bestaan. Het gelezen record wordt met regel 460 naar de printer gestuurd.

De lengte van het record is niet belangrijk. De printer zal het record namelijk altijd binnen de opgegeven kantlijnen afdrukken. Daarbij zal de printer zelf de records afbreken aan het einde van een regel, om op de volgende regel verder te gaan met het afdrukken van het record. De listing uit het voorbeeld is met behulp van het voorbeeldprogramma zelf gemaakt. U ziet dat de kantlijnen zo zijn gekozen, dat er precies 40 tekens per regel werden afgedrukt.

Zou u dit programma ook willen gebruiken op een systeem met cassetterecorder, dan dient u de volgende wijzigingen aan te brengen:

Verwijder de regels 310 tot en met 420.

Voeg een nieuwe regel 420 toe:

420 PRINT "Zet recorder op afspelen"

Wijzig regel 430 in:

430 OPEN "CAS:" FOR INPUT AS #1

Het vorige programma gaat er vanuit, dat de te gebruiken printer een aantal faciliteiten biedt, die niet door alle printers worden geboden. Zo zijn er printers, die geen kettingbaanformulieren kunnen verwerken. Een voorbeeld daarvan is de Brother CE50BT daisy wheel (typemachine) printer. Deze printer bestaat in feite uit een typemachine met een aantal printer-eigenschappen. Op een typemachine is het de typist, die erop toeziet, dat de letters op papier komen en dat er een nieuw papier wordt ingezet, zodra het vorige vol is. Logisch dus, dat op dit soort printers geen detectie aanwezig is, waarmee de printer ziet dat het papier op is.

Wie wel eens tekst heeft ingtikt op een typemachine, die zal het zijn opgevallen, dat een typemachine geen "wrap-around" faciliteit zit. Dit wil zeggen, dat tekst, die niet op de regel past, niet automatisch op de volgende regel wordt gezet. Ook dit verschil, met de meeste matrix printers, maakt, dat voorgaand programma niet zonder meer voor een typemachineachtige printer kan worden gebruikt.

Het volgende programma heeft precies dezelfde functie als het vorige programma. Er is echter nu rekening mee gehouden, dat de te gebruiken printer niet kan zien of het papier op is ("paper low detection") en dat de printer geen "wrap around" kent. Het voorbeeld is gemaakt voor en getest op de Brother CE50BT.

```
100 '*****
110 '* INITIALISATIE PRINTER *
```

```

120 '*****
130 '
140 INPUT "Aantal regels per pagina";RP
150 INPUT "Linker kantlijnpositie ";
LK
160 IF LK>80 THEN GOTO 150
170 INPUT "Aantal tekens per regel ";
TR
180 IF TR+LK>120 OR TR<2 THEN GOTO 150
190 PRINT "ZET DE PRINTER GEREED!!
        DRUK DAARNA OP RETURN!!"
200 IF INKEY$<>CHR$(13) THEN GOTO 200
210 LPRINT CHR$(27);"C";:'clear margins
and tabs
220 FOR I=0 TO LK
230 LPRINT " ";
240 NEXT I
250 LPRINT CHR$(27);"9";:'left margin
260 LPRINT CHR$(27);"@";'flush margin
270 '
280 '*****
290 '* HOOFDPROGRAMMA *
300 '*****
310 '
320 SCREEN 0:WIDTH 40:KEY OFF:FILES:X=0:
Y=0
330 PRINT:PRINT:PRINT"Wijs de gewenste f
ile aan met de cursor.Druk vervolgens op
RETURN."
340 LOCATE X,Y,1:I$=INKEY$
350 IF I$=CHR$(30) THEN Y=Y-1:IF Y<0 THE
N Y=0
360 IF I$=CHR$(28) THEN X=X+13:IF X>26 T
HEN X=0:Y=Y+1:IF Y>20 THEN Y=20
370 IF I$=CHR$(31) THEN Y=Y+1:IF Y>20 TH
EN Y=20
380 IF I$=CHR$(29) THEN X=X-13:IF X<0 TH
EN X=26:Y=Y-1:IF Y<0 THEN Y=0
390 IF I$<>CHR$(13) THEN 340
400 LOCATE 0,23,0:F$=""

```

```

410 FOR I=0 TO 11
420 F$=F$+CHR$(VPEEK(BASE(0)+X+I+Y*40))
430 NEXT I
440 RT=0
450 CLS
460 OPEN "a:"+F$ FOR INPUT AS #1
470 IF EOF(1)=-1 THEN GOTO 530
480 LINE INPUT #1,R$
490 IF LEN(R$)>TR THEN RL$=LEFT$(R$,TR):
R$=RIGHT$(R$,LEN(R$)-TR): RT=RT+1: LPRI
NT RL$: GOTO 490
500 LPRINT R$: RT=RT+1
510 IF RT=>RP THEN GOSUB 560
520 GOTO 470
530 CLS: KEY ON
540 CLOSE #1
550 END
560 RT=0
570 PRINT "Voer, zodra de printer stopt,
nieuw papier in en druk daarna op
RETURN"
580 IF INKEY$="" THEN GOTO 580
590 CLS
600 RETURN

```

Het zetten van de linker kantlijn gaat anders dan op de meeste andere printers. De printkop moet eerst op de kolom, waar de linker kantlijn wordt gewenst, worden gezet. Met regel 150 wordt de linker kantlijnpositie opgevraagd. Voordat de linker kantlijn wordt gezet, wordt eerst het commando "Clear Margins and Tabs" gegeven (zie regel 210). Hierna staat de printkop op de eerste positie van de regel. Met de regels 220 tot en met 240 worden nu zoveel spaties afgedrukt als er nodig zijn om de in regel 150 opgegeven linker kantlijnpositie te bereiken. Door nu de linker kantlijn te zetten (zie het commando in regel 250), wordt de nieuwe linker kantlijn op de opgegeven plaats gezet. Door nu het commando "Flush Right Margin" te geven, wordt de rechter kantlijn op de maximale waarde gezet.

Om nu het probleem van het ontbreken van een "wrap around" faciliteit te omzeilen, zullen we het aantal tekens dat op een regel wordt afgedrukt in het programma zelf moeten bijhouden. Het af te drukken record wordt met regel 480 van schijf gelezen. Met regel 490 wordt gekeken of dit record wel op 1 regel van TR tekens past (TR werd in regel 170 ingegeven). Is het record langer, dan wordt het opgedeeld in een stuk met een lengte TR en een overblijvend stuk. Het overblijvende stuk wordt weer gecontroleerd op lengte en eventueel opgedeeld. Dit gaat door, totdat het hele record is afgedrukt. Iedere keer, dat er een record of een gedeelte van een record wordt afgedrukt, wordt de regelteller (RT) met 1 verhoogd. Is de regelteller (RT) hoger of gelijk aan het aantal regels per pagina (RP), dan is de pagina vol en moet een nieuw papier in de printer worden gezet, voordat er wordt doorgeprint.

Met regel 510 wordt, indien het papier vol is, naar de subroutine op regel 560 gesprongen. In die subroutine wordt de regelteller weer op 0 gezet, wordt de operator ervan op de hoogte gesteld dat het papier moet worden verwisseld, en wordt gewacht totdat de operator een toets heeft ingedrukt. Na terugkeer uit de subroutine wordt het volgende record van disk gelezen. Dit gaat door totdat de hele file is gelezen en afgedrukt.

#### **5.4                   Verschillende print-modes**

De meeste printers kennen verschillende print-modes. Dit wil zeggen, dat ze in staat zijn om de tekens in verschillende diktes en grootten af te drukken. Matrix printers hebben wat dit betreft veel meer mogelijkheden dan daisy wheel printers. Bij daisy wheel printers kan vaak alleen worden gekozen voor vet of normaal afdrukken. Daarnaast kan dan nog worden gekozen voor een bepaald aantal tekens per inch (10, 12, 15 of proportioneel). Afhankelijk van het gewenste aantal tekens per regel moet echter ook een bepaald letterwiel in de printer aanwezig zijn.

Om een goed beeld van de mogelijkheden te geven, is daarom het volgende voorbeeldprogramma geschreven voor een matrix printer. Het programma werd geschreven voor en getest op de IBM-compatible Seikosha SP800 printer.

```
1000 '*****
1010 '* PRINTERBESTURINGSPROGRAMMA *
1020 '*****
1030 '
1040 '*****
1050 '* INITIALISATIE *
1060 '*****
1070 '
1080 SCREEN 0: WIDTH 40: CLS
1090 B=0: H=0: L=0
1100 KEY OFF
1110 FOR I=1 TO 10
1120 KEY(I) OFF
1130 NEXT I
1140 KEY 1, "BOLD"
1150 KEY 2, "HOOG"
1160 KEY 3, "LAAG"
1170 FOR I=4 TO 10
1180 KEY I, ""
1190 NEXT I
1200 FOR I=1 TO 3
1210 KEY(I) ON
1220 NEXT I
1230 ON KEY GOSUB 2000,3000,4000
1240 KEY ON
1250 '
1260 '*****
1270 '* HOOFDPROGRAMMA *
1280 '*****
1290 '
1300 I$=INKEY$:IF I$="" GOTO 1300
1310 IF I$=CHR$(13) THEN LPRINT CHR$(10)
;CHR$(13);:PRINT
1320 PRINT I$;
1330 LPRINT I$;
```

```

1340 I$="": GOTO 1300
2000 '
2010 '*****
2020 '* VET AFDRUKKEN *
2030 '*****
2040 '
2050 PRINT "^B";
2060 IF B=0 THEN LPRINT CHR$(27);"E"; EL
SE LPRINT CHR$(27);"F";
2070 B=NOT(B)
2080 RETURN
3000 '
3010 '*****
3020 '* SUPERSCRIPT *
3030 '*****
3040 '
3050 PRINT "^H";
3060 IF H=0 THEN LPRINT CHR$(27);"S";CHR
$(0); ELSE LPRINT CHR$(27);"T";
3070 H=NOT(H)
3080 RETURN
4000 '
4010 '*****
4020 '* SUBSCRIPT *
4030 '*****
4040 '
4050 PRINT "^L";
4060 IF L=0 THEN LPRINT CHR$(27);"S";CHR
$(1); ELSE LPRINT CHR$(27);"T";
4070 L=NOT(L)
4080 RETURN

```

Voorbeeld van het resultaat:

Enkele testregels

Deze regel is normaal afgedrukt.  
De woorden **vet gedrukt** zijn vet gedrukt.

Het woordje `superscript` is hoger afgedrukt.

Het woordje `subscript` is lager afgedrukt.

Het programma zelf is erg eenvoudig. Met regel 1300 wordt een teken van het toetsenbord opgevraagd. Regel 1310 controleert of de RETURN-toets is ingedrukt. Is dit inderdaad het geval, dan wordt een CR en een LF code naar de printer gestuurd. Is dit niet het geval, dan wordt het ingetikte teken naar de printer gestuurd (regel 1330). De ingetikte tekens worden ook op het beeldscherm afgedrukt (regel 1320).

Aan het begin van het programma worden nieuwe functies aan de functietoetsen toegekend. Met regel 1230 wordt aangegeven, waar naartoe moet worden gesprongen na het indrukken van een bepaalde functietoets. Stel nu dat functietoets 1 werd ingedrukt. Er zal dan naar de subroutine, die op regel 2000 start, worden gesprongen.

In de subroutine wordt de tekst **B** naar het beeldscherm gestuurd. Vervolgens wordt, op voorwaarde dat de variabele `B=0`, de ESC-reeks `CHR$(27);"E"` naar de printer gestuurd. Deze zal daarop reageren, door alle volgende tekens vet af te drukken. Voordat de subroutine wordt verlaten wordt de variabele `B` ginvertteerd. Hierdoor zal, de volgende keer dat de subroutine wordt aangeroepen, het tweede deel van het `IF...THEN`-statement worden uitgevoerd (`CHR$(27);"F"`; waarmee het vet afdrukken weer wordt gereset.). De overige twee subroutines werken op dezelfde manier, doch veroorzaken het afdrukken in super- en subscript.

Het volgende programma doet ongeveer hetzelfde als het vorige programma, doch dit programma is geschreven voor gebruik met een MSX-printer. Bovendien zijn er een aantal functies aan toegevoegd, zoals:

- Meer functietoetsen zijn geprogrammeerd.
- De kantlijnen kunnen worden ingesteld.
- De ingetikte regel kan worden gecorrigeerd.



De regel wordt pas gedrukt nadat RETURN is ingetikt.  
De subroutines zijn iets gewijzigd.

Daar de meeste functies van dit programma al in andere programma's zijn gebruikt, is geen verdere toelichting nodig.

```
1000 '*****
1010 '* PRINTERBESTURINGSPROGRAMMA *
1020 '*****
1030 '
1040 '*****
1050 '* INITIALISATIE PROGRAMMA *
1060 '*****
1070 '
1080 SCREEN 0: WIDTH 40: CLS
1090 CLEAR 5000
1100 B=0: H=0: L=0
1110 KEY OFF
1120 FOR I=1 TO 10
1130 KEY(I) OFF
1140 NEXT I
1150 KEY 1,"bold"
1160 KEY 2,"hoog"
1170 KEY 3,"laag"
1180 KEY 4,"italic"
1190 KEY 5,"nlq"
1200 FOR I=6 TO 10
1210 KEY I,""
1220 NEXT I
1230 FOR I=1 TO 5
1240 KEY(I) ON
1250 NEXT I
1260 ON KEY GOSUB 2000,2100,2200,2300,24
00
1270 KEY ON
1280 '
1290 '*****
1300 '* INITIALISATIE PRINTER *
1310 '*****
1320 '

```

```

1330 INPUT "Linker kantlijnpositie   NN"
;LK$
1340 IF LEN(LK$)<>2 THEN GOTO 1330
1350 INPUT "Aantal tekens per regel  NN"
;TR$
1360 IF LEN(TR$)<>2 THEN GOTO 1350
1370 TR=VAL(TR$): LK=VAL(LK$):RK=LK+TR
1380 IF RK>100 THEN GOTO 1330
1390 RK$=RIGHT$(STR$(RK),2)
1400 PRINT "ZET DE PRINTER GEREED!!
          DRUK DAARNA OP RETURN!!"
1410 IF INKEY$<>CHR$(13) THEN GOTO 1410
1420 LPRINT CHR$(27);"@";
1430 LPRINT CHR$(27);"L0"+LK$;
1440 LPRINT CHR$(27);"/0"+RK$;
1450 '
1460 '*****
1470 '* HOOFDPROGRAMMA *
1480 '*****
1490 '
1500 DIM T$(20,TR)
1510 CLS
1520 IF TR>39 THEN ML=39 ELSE ML=TR
1530 LOCATE 1,0: PRINT STRING$(ML,"*")
1540 FOR I=1 TO 20
1550 FOR J=1 TO TR
1560 LOCATE J,I,1
1570 T$(I,J)=INKEY$: IF T$(I,J)="" GOTO
1570
1580 IF T$(I,J)=CHR$(8) THEN J=J-1: GOTO
1560
1590 IF T$(I,J)=CHR$(13) THEN J=TR: PRIN
T
1600 PRINT T$(I,J);
1610 NEXT J
1620 FOR J=0 TO TR
1630 LPRINT T$(I,J);
1640 NEXT J
1650 LPRINT
1660 NEXT I

```

```

1670 END
2000 '
2010 '*****
2020 '* VET AFDrukKEN *
2030 '*****
2040 '
2050 IF B=0 THEN KEY 1,"BOLD" ELSE KEY 1
,"bold"
2060 IF B=0 THEN T$(I,J-1)=T$(I,J-1)+CHR
$(27)+"CB" ELSE T$(I,J-1)=T$(I,J-1)+CHR$
(27)+"Cb"
2070 B=NOT(B)
2080 RETURN
2100 '
2110 '*****
2120 '* SUPERSCRIPT *
2130 '*****
2140 '
2150 IF H=0 THEN KEY 2,"HOOG" ELSE KEY 2
,"hoog"
2160 IF H=0 THEN T$(I,J-1)=T$(I,J-1)+CHR
$(27)+"CS" ELSE T$(I,J-1)=T$(I,J-1)+CHR$
(27)+"Cs"
2170 H=NOT(H)
2180 RETURN
2200 '
2210 '*****
2220 '* SUBSCRIPT *
2230 '*****
2240 '
2250 IF L=0 THEN KEY 3,"LAAG" ELSE KEY 3
,"laag"
2260 IF L=0 THEN T$(I,J-1)=T$(I,J-1)+CHR
$(27)+"CU" ELSE T$(I,J-1)=T$(I,J-1)+CHR$
(27)+"Cu"
2270 L=NOT(L)
2280 RETURN
2300 '
2310 '*****
2320 '* ITALIC *

```

```

2330 '*****
2340 '
2350 IF IT=0 THEN KEY 4,"ITALIC" ELSE KE
Y 4,"italic"
2360 IF IT=0 THEN T$(I,J-1)=T$(I,J-1)+CH
R$(27)+"CI" ELSE T$(I,J-1)=T$(I,J-1)+CHR
$(27)+"Ci"
2370 IT=NOT(IT)
2380 RETURN
2400 '
2410 '*****
2420 '* NLQ *
2430 '*****
2440 '
2450 IF N=0 THEN KEY 5,"NLQ" ELSE KEY 5,
"nlq"
2460 IF N=0 THEN T$(I,J-1)=T$(I,J-1)+CHR
$(27)+"!" ELSE T$(I,J-1)=T$(I,J-1)+CHR$(
27)+CHR$(34)
2470 N=NOT(N)
2480 RETURN

```

In de volgende tekst zullen de be-  
treffende woorden zo worden afge-  
drukt als ze zelf aangeven.

Met dit programma is het mogelijk  
om woorden **vet** af te drukken, om ze  
*schuin* af te drukken, of in **Near**  
**Letter Quality** af te drukken. Bo-  
vendien kunnen ze als *superscript* o  
of als *subscript* worden afgedrukt.

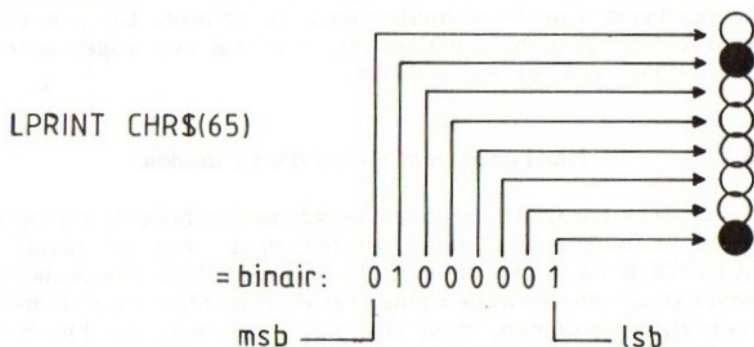
Het programma is geschreven voor de  
**Seikosha SP1000MX** printer.

Aan de naam van de printer in de  
vorige regel ziet u trouwens, dat  
het ook mogelijk is om **vet** en  
*schuin* tegelijk af te drukken.

## 6 Grafische print modes

Bijna alle moderne matrix printers hebben een speciale grafische mode. In die mode kunnen, in plaats van tekens die via de zogenaamde Character PROM worden samengesteld, de naalden van de printkop rechtstreeks worden aangestuurd. Een code, die van de computer naar de printer wordt gestuurd, bestaat uit 8 bitjes. Door nu ieder bitje aan een naaldje te relateren, kan de waarde van het betreffende bitje het al of niet afvuren van de bijbehorende naald veroorzaken. Met andere woorden, een teken, dat van de computer naar de printer wordt gestuurd, veroorzaakt een verticale kolom van maximaal 8 puntjes op het papier.

Indien de code van het verzonden teken 0 is, zal geen enkele naald worden afgevuurd. Is de waarde 255 (hexadecimaal FF) dan zullen alle 8 naaldjes worden afgevuurd en zodoende een verticaal lijntje vormen. Een en ander is weergegeven in afbeelding 6-1.



Afb. 6-1 Een kolom in grafische mode.

Nadat de kolom is afgedrukt, wordt de printkop een heel klein stukje verplaatst. Nu kan de volgende door de computer verzonden code naar de printkop worden gestuurd. En weer ont-

staat een verticale rij puntjes. Door op deze manier een aantal kolommen te printen, ontstaat een patroon van puntjes, vergelijkbaar met het patroon van puntjes op een beeldscherm. Het is zodoende mogelijk om tekeningen en teksten, net als op een beeldscherm, af te drukken.

Daisy wheel printers hebben de hiervoor genoemde mogelijkheid niet. Toch is het, zoals we later in dit hoofdstuk zullen zien, op de meeste daisy wheel printers mogelijk om grafische afbeeldingen af te drukken. In plaats van acht puntjes tegelijk drukken we dan 1 puntje tegelijk af. Op elk print-wiel staat immers wel een punt (.) en een spatie. Bovendien is het op de meeste daisy wheel printers mogelijk om zowel de horizontale positionering van de printkop, als de verticale positionering van het papier, in hele kleine stapjes te laten geschieden. Verderop in dit hoofdstuk zullen we aan de hand van een paar voorbeeldprogramma's zien hoe dit in zijn werk gaat.

Om een aantal goede voorbeelden van de programmering van de printers te kunnen geven, en om bovendien een aantal direct bruikbare programma's, die niet al te lang zijn, te kunnen geven, heb ik besloten om alle voorbeelden in dit hoofdstuk te baseren op het copieren van de inhoud van het beeldscherm van de computer naar de printer. De gegeven voorbeeldprogramma's maken dus allemaal een zogenaamde "hard copy" van het beeldscherm.

## **6.1 Hard copy in SCREEN 0 en 1 modes.**

De in deze paragraaf gegeven programma's hebben eigenlijk niets te maken met de grafische mode van de printer. SCREEN 0 en 1 zijn immers tekstmodes. Het beeldscherm bevat in die modes alleen maar tekst. Dat deze programma's toch zijn opgenomen, vindt zijn oorzaak daarin, dat hiermee de hard copy mogelijkheden voor MSX-computers compleet maken.

Daar het hier alleen om teksten gaat, is het heel goed mogelijk om zowel op een matrix printer als op een daisy wheel

printer een hard copy te maken. Bij gebruik van een daisy wheel printer moet er echter op worden gelet dat de teksten op het beeldscherm alleen letters, cijfers en leestekens bevatten, die op het print-wiel voorkomen. Voor het overige is er geen verschil tussen de matrix- en de daisy wheel printers.

Wel is er verschil tussen beide schermmodes. Er worden dan ook twee programma's gegeven, het eerste voor SCREEN 0 en het tweede voor SCREEN 1.

```
10 '*****
20 '* Hardcopy - SCREEN 0 *
30 '*****
40 '
50 FOR J=0 TO 23
60 FOR I=0 TO 39
70 LPRINT CHR$(VPEEK(BASE(0)+J*40+I));
80 NEXT I
90 LPRINT
100 NEXT J
110 END
```

```
10 '*****
20 '* Hardcopy - SCREEN 1 *
30 '*****
40 '
50 FOR J=0 TO 23
60 FOR I=0 TO 31
70 LPRINT CHR$(VPEEK(BASE(5)+J*32+I));
80 NEXT I
90 LPRINT
100 NEXT J
110 END
```

Beide voorgaande programma's zijn in principe gelijk aan elkaar. Het enige verschil is, dat de tekentabellen in het video RAM op een andere plaats starten en een verschillende

lengte hebben. Daarom worden de variabelen I en J (in regels 50 en 60) met verschillende waarden geladen. Voor SCREEN 0 geldt, dat er 24 regels van elk 40 tekens in het video RAM staan en voor SCREEN 1 geldt dat er 24 regels van elk 32 tekens in het video RAM staan.

De regels 70 lezen een positie uit het video RAM. Voor SCREEN 0 staat het beginadres van de tekentabel in het video RAM in systeemvariabele BASE(0) en voor SCREEN 1 staat het beginadres van de tekentabel in systeemvariabele BASE(5). De formule in regels 70 zorgt ervoor dat alle tekens uit het video RAM stuk voor stuk worden uitgelezen en op de juiste plaats op het papier worden afgedrukt.

De gegeven programma's werken op praktisch alle printers, zolang die printers als codes voor CR (Carriage Return) en LF (Line Feed) respectievelijk 13 en 10 gebruiken (hexadecimaal D en A).

## 6.2 Grafische schermen naar Daisy Wheel Printer.

Het volgende programma maakt een hard copy van een grafisch beeldscherm, dat is beschreven in de high resolution mode (SCREEN 2). Het werd geschreven voor en getest op de Brother CE50BT. Na de listing ziet u het resultaat van het gegeven programma.

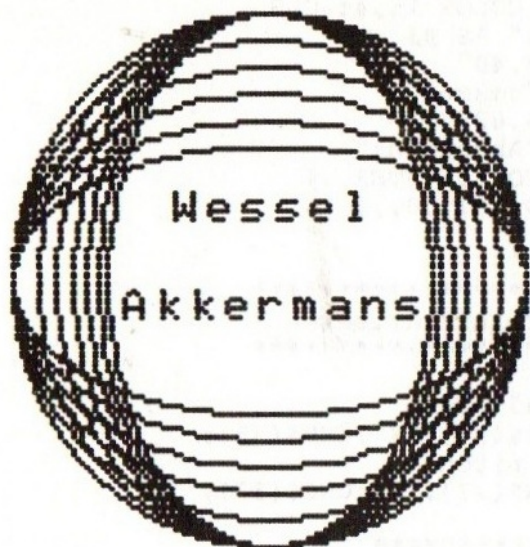
```
100 '*****
110 '* Hardcopy - SCREEN 2 *
120 '*****
130 '
140 SCREEN 0: WIDTH 40: CLS
150 INPUT "Welke kleur moet worden gepri
nt (nr.)";K
160 '
170 '*****
180 '* Iets op het scherm zetten *
190 '*****
```



```

200 '
210 SCREEN 2: COLOR 15,4: CLS
220 OPEN "grp:" AS #1
230 DRAW "BM38,40"
240 PRINT #1,"Wessel"
250 DRAW "BM26,62"
260 PRINT #1,"Akkermans"
270 FOR I=.5 TO 1.6 STEP .1
280 CIRCLE (60,60),60,,,,I
290 NEXT I
300 '
310 '*****
320 '* Printer initialiseren *
330 '*****
340 '
350 'Horizontal pitch
360 LPRINT CHR$(27);"K";CHR$(35);
370 'Vertical pitch
380 LPRINT CHR$(27);"L";CHR$(37);
390 '
400 '*****
410 '* Afdruk routine *
420 '*****
430 '
440 FOR Y=0 TO 191
450 FOR X=0 TO 255
460 IF POINT(X,Y)=K THEN LPRINT "."; ELSE
LPRINT " ";
470 NEXT X
480 LPRINT
490 NEXT Y
500 END

```



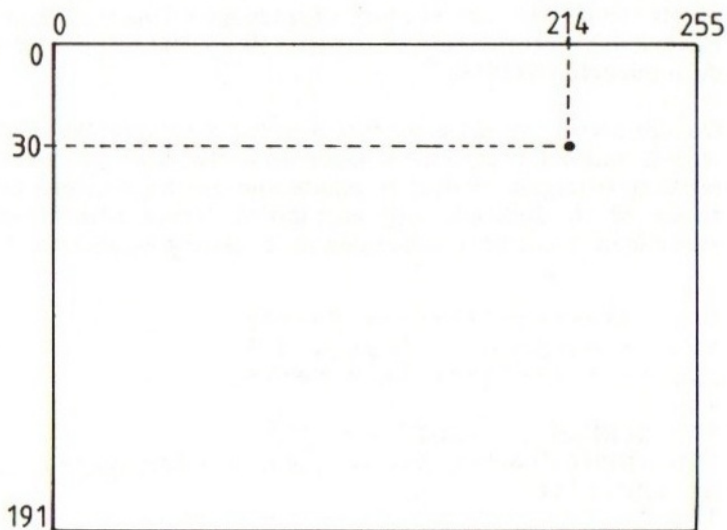
Voordat we naar het werkelijke afdrucken gaan kijken, eerst even het volgende. Met regel 460 wordt de kleur van een beeldpunt opgevraagd. Het betreffende beeldpunt wordt aangegeven met behulp van de coördinaten X en Y. Afbeelding 62 geeft het coördinatenstelsel van het beeldscherm aan. De kleur van het opgegeven pixel wordt vergeleken met het in K staande kleurnummer. Dit kleurnummer is in regel 150 opgevraagd.

Indien de kleur van het pixel gelijk is aan de opgegeven kleur, wordt een punt gedrukt. Is de kleur niet gelijk, dan wordt een spatie gedrukt. Dit alles wordt in regel 460 gedaan. Deze regel zit in een FOR...NEXT-lus, waardoor achtereenvolgens alle beeldpuntjes van het hele scherm worden bekeken.

Met de regels 210 tot en met 290 wordt een plaatje op het beeldscherm getekend. Voordat dit plaatje wordt afgedrukt moet de printer worden ingesteld, en wel zodanig, dat de

printkop na het afdrukken van een puntje ongeveer de afstand van 1 puntje of spatie opschuift en dat het papier na een horizontale lijn van puntjes en spaties ongeveer een afstand van 1 puntje wordt opgeschoven. Dit gebeurt in de regels 360 en 380. Horizontal pitch wil zeggen: de verplaatsing van de printkop. Vertical pitch is de verplaatsing van het papier.

POINT(214,30) :



Afb. 6-2 Het coördinatenstelsel van een grafisch scherm.

De horizontale pitch kan op de Brother CE50BT worden ingesteld in stapjes van 1/120-ste inch. Geen verplaatsing (0/120-ste inch) wordt aangegeven met de decimale waarde 32. 1/120-ste inch wordt aangegeven met 33, enzovoorts. De maximale verplaatsing is 31/120-ste inch, hetgeen wordt aangegeven met de decimale waarde 63.

De verticale pitch kan worden ingesteld in stapjes van 1/192-ste inch. Geen papier opschuif (0/192-ste inch) wordt aangegeven met de decimale waarde 32. De maximale papierop-

schuif is 94/192-ste inch, hetgeen met de decimale waarde 126 wordt aangegeven. Ook hier geldt dat de opschuif kan worden ingesteld in stapjes van 1/192-ste inch.

Het volgende programma laat zien hoe het beeldscherm, dat in de mode SCREEN 3 is beschreven, op een daisy wheel printer kan worden afgedrukt. Het verschil zit hem voornamelijk in het feit, dat er nu veel minder tekens op een regel kunnen worden geprint (zie regels 240 en 250) en dat er veel minder pixels behoeven te worden afgevraagd. Daarom zijn de FOR...NEXT-statements in regels 410 en 420 uitgebreid met de toevoeging STEP 4.

Ook de listing van dit programma wordt weer gevolgd door de output van het programma. Door de horizontale en verticale pitch te wijzigen, en door in plaats van een punt (.) een ander teken af te drukken, zijn verrassend fraaie resultaten te verkrijgen, zoals de voorbeelden na de listing laten zien.

```
100 '*****
110 '* Hardcopy - SCREEN 3 *
120 '*****
130 '
140 SCREEN 0: WIDTH 40: CLS
150 INPUT "Welke kleur moet worden gepri
nt (nr.)";K
160 '
170 '*****
180 '* Iets op het scherm zetten *
190 '*****
200 '
210 SCREEN 3: COLOR 15,4: CLS
220 OPEN "grp:" AS #1
230 PSET (0,0)
240 PRINT #1,"Grafisch"
250 PRINT #1," scherm"
270 '
280 '*****
290 '* Printer initialiseren *
```

```

300 '*****
310 '
320 'Horizontal pitch
330 LPRINT CHR$(27);"K";CHR$(35);
340 'Vertical pitch
350 LPRINT CHR$(27);"L";CHR$(37);
360 '
370 '*****
380 '* Afdruk routine *
390 '*****
400 '
410 FOR Y=0 TO 191 STEP 4
420 FOR X=0 TO 255 STEP 4
430 IF POINT(X,Y)=K THEN LPRINT "."; ELS
E LPRINT " ";
440 NEXT X
450 LPRINT
460 NEXT Y
470 END

```

# Grafisch scherm

## 6.3 Grafische mode van matrix printers.

Matrix printers dienen op een geheel andere manier te worden aangestuurd dan daisy wheel printers. Bij een daisy wheel printer kan steeds slechts 1 puntje tegelijk worden afgedrukt. Bij matrix printers kunnen echter 8 puntjes tegelijk worden afgedrukt. Het probleem daarbij is, dat de waarde van die acht puntjes van de computer naar de printer moet worden gestuurd. Zoals we echter al hebben gezien, stellen alle combinaties van acht bitjes al een teken voor. De acht bits code wordt in de printer, met behulp van de "tekengenerator", omgezet naar een aantal kolommen van puntjes, die samen een teken vormen.

Om nu toch in staat te zijn om een code van acht bitjes rechtstreeks naar acht naaldjes van de printkop te sturen, moet er een mogelijkheid zijn om de tekengenerator te omzeilen. Deze mogelijkheid wordt geboden door een speciaal commando, waarmee de printer in de grafische mode wordt gezet.

Dat commando bestaat uit een ESC-code, gevolgd door een code waarmee de printer in de grafische mode wordt gezet en waarmee de printer wordt verteld hoeveel grafische codes (combinaties van acht bitjes) er zullen worden verstuurd. Stel dat we een regel van het grafisch scherm van de computer naar de matrix printer willen sturen. Het beeldscherm bestaat horizontaal uit 256 beeldpuntjes. De bovenste lijn van beeldpuntjes moet uiteindelijk terecht komen bij de bovenste naald in de printkop. De tweede lijn van 256 beeldpuntjes bij de tweede naald, enz. Er is dus sprake van 256 verticale kolommen van 8 beeldpuntjes. We willen dus 256 grafische codes (combinaties van acht bitjes) naar de printer sturen.

Om die 256 grafische codes door de printer als grafische codes te kunnen laten interpreteren, dienen we de printer eerst in de grafische mode te zetten en erbij te vertellen dat er 256 codes zullen volgen. Voor een MSX-printer komt het commando, waarmee we dit kunnen aangeven er dan als volgt uit te zien:

```
LPRINT CHR$(27);"G0600256";
```

De letter G verteld de printer dat deze naar de grafische mode moet overgaan. De cijfers 060 geven aan welke grafische mode we wensen. De meeste matrix printers hebben meerdere grafische modes, met verschillende resoluties. Dit is te vergelijken met de grafische modes van de MSX-computer (hoge en lage resolutie). De cijfers 0256, ten slotte, geven aan hoeveel codes er zullen volgen. Als we vervolgens 256 codes naar de printer sturen, zullen deze codes rechtstreeks naar de printkop worden gestuurd. Zodoende zal iedere code slechts 1 verticale kolom van 8 puntjes tengevolge hebben. Na de 256-ste code schakelt de printer weer automatisch terug naar de normale print-mode. Dit is belangrijk om in de gaten te houden. Als je opgeeft dat er 10 grafische codes zullen worden gestuurd, dan interpreteert de printer de eerstvolgende 10 codes, niet meer en niet minder, als grafische codes. De eventuele elfde code die we sturen, wordt weer als een normaal teken beschouwd.

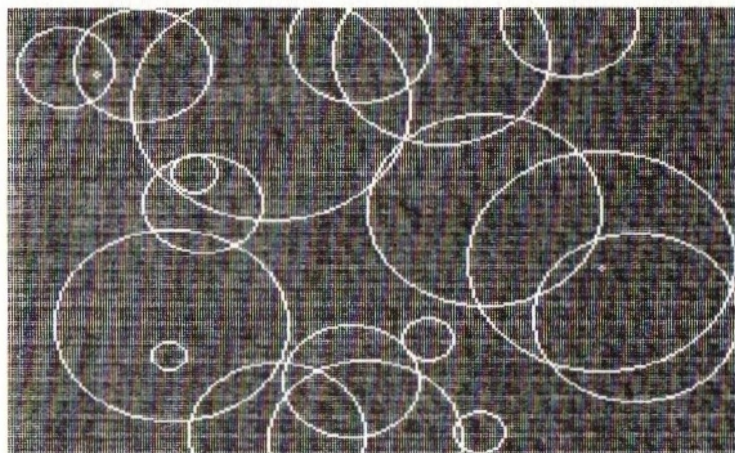
Zoals gezegd, worden in de grafische mode altijd 8 naaldjes van de printkop gebruikt. De meeste printkoppen hebben echter negen naaldjes. Bovendien wordt er bij normaal afdrucken altijd enige ruimte tussen de regels opengelaten. Bij het maken van een tekening willen we echter niet dat de lijnen worden onderbroken. We zullen daarom de regelafstand zodanig moeten instellen, dat de bovenste naald van de printkop zijn puntjes vlak onder het onderste puntje van de vorige regel plaatst. Dit houdt voor de meeste printers in, dat er een regelafstand van 1/9 inch moet worden gekozen. Op MSX-printers is hiervoor een speciaal commando aanwezig, namelijk LPRINT CHR\$(27);"B";.

Het volgende programma maakt een hard copy van het beeldscherm. Alle hiervoor besproken theorie is in dit programma terug te vinden op de regelnummers 290 en 310. Het resultaat van het programma is te zien in het plaatje dat na de listing volgt.

```
100 DEFINT A-Z
110 CLEAR 2000
120 SCREEN 2
130 FOR I=1 TO 20
140 X=RND(1)*256
150 Y=RND(1)*192
160 Z=RND(1)*50
170 CIRCLE (X,Y),Z
180 NEXT I
185 LPRINT CHR$(27);"@";
190 TIME=0
200 GOSUB 250
210 LPRINT CHR$(27);"@";
220 LPRINT:LPRINT
230 LPRINT "TEKENTIJD =";INT(TIME/50);"S
ECONDEN"
240 BEEP:GOTO 240
250 '*****
260 '* GRAFISCHE PRINT-ROUTINE *
270 '*****
280 '
290 LPRINT CHR$(27);"B";
300 FOR L=0 TO 23
310 LPRINT CHR$(27);"G0600256";
320 FOR X=0 TO 255
325 B$=""
330 FOR I=7 TO 0 STEP -1
340 Y=L*8+I
350 IF POINT(X,Y)=15 THEN B$=B$+"0" ELSE
    B$=B$+"1"
360 NEXT I
420 LPRINT CHR$(VAL("&B"+B$));
430 NEXT X
```



```
450 LPRINT  
460 NEXT L  
470 RETURN
```



TEKENTIJD = 861 SECONDEN

Met de regels 100 tot en met 240 wordt een plaatje op het beeldscherm getekend. Het eigenlijke afdrucken begint pas op regel 290, waar de regelafstand wordt ingesteld. Op regelnummer 300 start een FOR...NEXT lus, die voor iedere regel op het beeldscherm eenmaal wordt uitgevoerd. Het beeldscherm bestaat uit 24 regels, van 0 tot en met 23. U ziet, dat de printer met regelnummer 310 voor iedere af te drukken beeldschermregel opnieuw in de grafische mode wordt gezet.

Met regel 350 wordt de status van een beeldpuntje opgevraagd. We dienen echter nu de status van acht onder elkaar liggende beeldpuntjes te kennen, voordat we een code naar de printer kunnen sturen. Vandaar, dat er voor ieder horizontaal beeldpuntje een FOR...NEXT-lus (regelnummer 320) wordt gestart, waarbinnen de waarden van de acht verticaal onder

elkaar liggende beeldpuntjes (de FOR...NEXT-lus van regel 330) worden verzameld in B\$. Uiteindelijk wordt de totale waarde van B\$ naar de printer gestuurd (regelnummer 420). Nadat er 256 verticale kolommen zijn verwerkt, wordt de lus van regelnummer 320 verlaten en wordt het LPRINT-statement van regelnummer 450 uitgevoerd, hetgeen een regelopschuif ten gevolge heeft. Daarna wordt de volgende beeldschermregel afgetast.

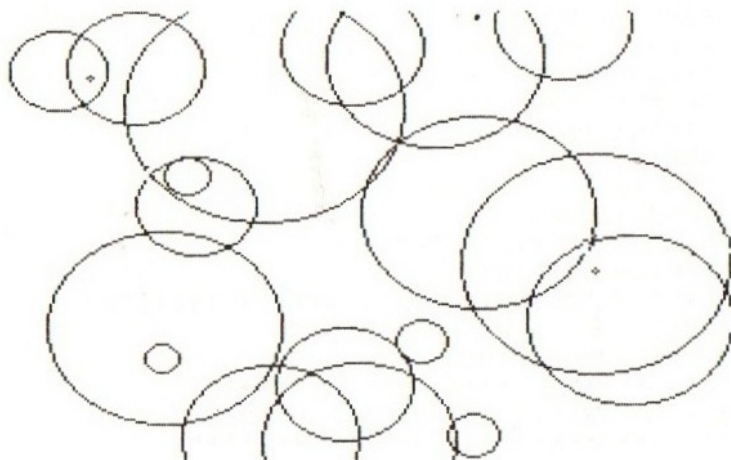
Er doet zich bij het gebruiken van de grafische mode van een printer altijd een klein probleempje voor, en wel, welk bitje van de naar de printer te versturen code komt bij welke naarld van de printkop terecht. Stel dat we de code CHR\$(1) naar de printer sturen, wordt dan de bovenste of de onderste naald van de printkop afgevuurd? Per printermerk kan dit nogal eens verschillen. In het voorbeeldprogramma voor de MSX-printer ziet u, dat we eerst het onderste beeldpuntje van een kolom van 8 in B\$ zetten, daarna het een na onderste, enz. Dit betekent, dat het onderste puntje in de uiteindelijke code de hoogste waarde (128) zal hebben. Het meest significante bitje uit een code zal dus op de MSX-printer op de onderste naald van de printkop worden afgedrukt (Zie ook afbeelding 6-1). De Seikosha SP800 werkt echter precies andersom. Daar zal het meest significante bitje op de bovenste naald van de printkop worden afgedrukt. Voor uw printer zult u, in het bij de printer geleverde handboek, moeten nakijken hoe uw printer de codes in de grafische mode verwerkt.

U ziet, dat het afdrukken van een hard copy van het beeldscherm met voorgaand programma 861 seconden duurt. Dit is bijna een kwartier. Het volgende programma doet hetzelfde als het vorige, doch nu is er geen gebruik gemaakt van de functie POINT. Bovendien is het volgende programma geschreven voor en getest op een Seikosha SP800 printer. Door in dit programma gebruik te maken van het schuiven met strings, in plaats van het werken met POINT, is de snelheid enigszins opgevoerd. Ik kan u melden, dat ik hetzelfde programma ook nog heb geschreven met gebruikmaking van rekenkundige functies voor het omzetten van beeldpuntjes naar printer codes. Dat was echter zo veel langzamer, dat ik

het hier niet meer heb willen opnemen.

```
5 DEFINT A-Z
10 CLEAR 2000
20 SCREEN 2
30 FOR I=1 TO 20
40 X=RND(1)*256
50 Y=RND(1)*192
60 Z=RND(1)*50
70 CIRCLE (X,Y),Z
80 NEXT I
85 TIME=0
90 GOSUB 30000
93 LPRINT CHR$(27);"@";
94 LPRINT: LPRINT
95 LPRINT "TEKENTIJD =";INT(TIME/50);"SE
CONDEN"
100 BEEP: GOTO 100
30000 '*****
30010 '* GRAFISCHE PRINT-ROUTINE *
30020 '*****
30030 '
30040 LPRINT CHR$(27);"A";CHR$(8);
30050 FOR L=0 TO 23
30060 LPRINT CHR$(27);"*";CHR$(0);CHR$(0
);CHR$(1);
30070 FOR K=0 TO 31
30080 A=L*256+K*8
30090 FOR I=0 TO 7
30100 I$(I)=RIGHT$("00000000"+BIN$(VPEEK
(A+I)),8)
30110 NEXT I
30120 FOR I=1 TO 8
30130 B$=""
30140 FOR J=0 TO 7
30150 B$=B$+MID$(I$(J),I,1)
30160 NEXT J
30170 LPRINT CHR$(VAL("&B"+B$));
30180 NEXT I
```

```
30190 NEXT K
30200 LPRINT CHR$(10);
30210 NEXT L
30220 RETURN
```



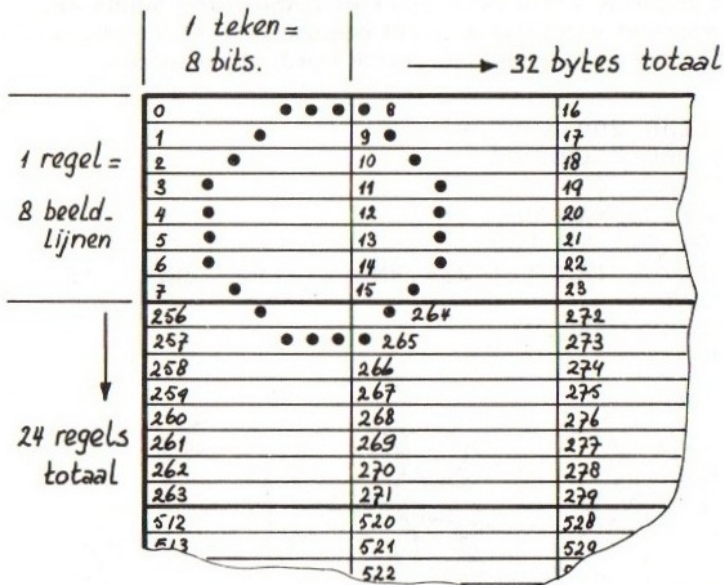
TEKENTIJD = 663 SECONDEN

U ziet, dat de grafische afdrukroutine nu van hoge regelnummers is voorzien. In feite is het een opzichzelfstaande subroutine, die vanuit het hoofdprogramma op ieder gewenst moment kan worden aangeroepen. Door de afdrukroutine als zelfstandige ASCII-file naar schijf of cassette te schrijven, kan deze later aan een bestaand (teken)programma worden gekoppeld. Het wegschrijven van de subroutine kan met het statement **SAVE "CAS:HC.ASC",A** en het later weer aan een ander programma koppelen gaat dan met het statement **MERGE "CAS:HC.ASC"**. Daarbij dient u er alleen op te letten, dat het hoofdprogramma niet dezelfde regelnummers gebruikt als de subroutine.

Met regelnummer 30040 wordt de regelafstand ingesteld op

8/72-ste inch. U ziet, dat dit weer neerkomt op 1/9-de inch, ofwel 9 regels per inch. Met regelnummer 30060 wordt de printer in de grafische mode gezet. De asterisk (\*) verteld de printer, welke grafische mode wordt gewenst. Daarna kan in twee bytes worden aangegeven hoeveel grafische codes zullen worden gestuurd. In het eerste byte kan het aantal worden gegeven, dat wordt opgeteld by 256 keer de inhoud van het tweede byte. In het voorbeeldprogramma wordt de printer dus verteld dat er  $0 + 256*1 = 256$  bytes zullen worden verstuurd.

In grote lijnen volgt dit programma hetzelfde algoritme als het vorige programma. We kijken nu echter direct in het videogeheugen naar de status van de beeldpunten. Het uitlezen van het videogeheugen werkt echter niet puntje voor



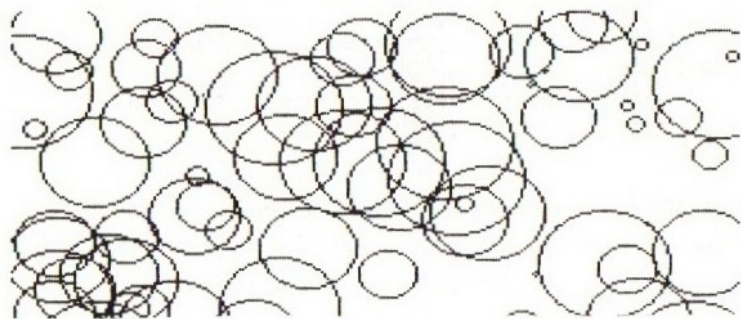
Het eerste byte (nummer 0) staat op adres  $BASE(12) + 0$   
 Het tweede byte (nummer 1) in het Video RAM staat op het beeldscherm onder het eerste byte (nummer 0).

Afb. 6-3 De relatie tussen beeldscherm en videogeheugen.

puntje, doch er wordt steeds een horizontale rij van 8 bitjes uitgelezen (Zie afbeelding 6-3). We zullen echter een verticale kolom van 8 bitjes naar de printer sturen. We zullen dus eerst 8 onder elkaar liggende horizontale rijen van 8 bitjes moeten lezen (zie regelnummers 30090 t/m 30110), om daarvan vervolgens, van links naar rechts, verticale kolommen te maken, die naar de printer worden gestuurd.

Van voorgaande BASIC-subroutine heb ik ook nog een machinetaalversie gemaakt. Deze machinetaalroutine is dus ook geschreven voor gebruik van de Seikosha SP800 printer. De routine is echter gemakkelijk te wijzigen voor andere printers. We zullen dat gaan bekijken bij de behandeling van van de subroutine. Voordat de listing van de machinetaalroutine wordt gegeven, wordt eerst een kort BASIC-programma gegeven, waarmee een plaatje op het beeldscherm wordt gemaakt, en waarmee de machinetaalroutine wordt aangeroepen.

```
10 CLEAR 200,&HBFFF
20 BLOAD "SCRNDU"
30 SCREEN 2
40 DEFUSR0=&HC000
50 FOR I=1 TO 100
60 CIRCLE (RND(1)*256,RND(1)*192),RND(1)*25
70 NEXT I
80 DUMMY=USR0(0)
90 END
```



```

10 ;GRAPHIC PRINTER DRIVER
20 ;
00A8 30 LPTSTT: EQU #A8
00A5 40 LPTOUT: EQU #A5
0059 50 LDIRMV: EQU #59
C000 60 ORG #C000
70 ;SET LINE SPACING
C000 80 LD A,#1B
C002 90 CALL LPTOUT
C005 100 LD A,#41
C007 110 CALL LPTOUT
C00A 120 LD A,#08
C00C 130 CALL LPTOUT
C00F 140 LD HL,0
C012 150 LD B,24
C014 160 REGEL: PUSH BC
C015 170 CALL SETGR
180 ;COPY LINE VRAM->RAM
C018 190 LD DE,BUFFER
C01B 200 LD BC,#0100
C01E 210 CALL LDIRMV
C021 220 LD DE,#0100
C024 230 ADD HL,DE
C025 240 PUSH HL
C026 250 LD HL,BUFFER
C029 260 LD B,32
C02B 270 TEKEN: PUSH BC
C02C 280 LD C,8
C02E 290 KOLDM: LD DE,7
C031 300 LD B,0
C033 310 LD A,(HL)
C034 320 RLC A
C036 330 JR NC,L6
C038 340 SET 7,B
C03A 350 L6: LD (HL),A
C03B 360 INC HL
C03C 370 LD A,(HL)
C03D 380 RLC A
C03F 390 JR NC,L5
C041 400 SET 6,B

```

C043	77	410	L5:	LD	(HL),A
C044	23	420		INC	HL
C045	7E	430		LD	A,(HL)
C046	CB07	440		RLC	A
C048	3002	450		JR	NC,L4
C04A	CBE8	460		SET	5,B
C04C	77	470	L4:	LD	(HL),A
C04D	23	480		INC	HL
C04E	7E	490		LD	A,(HL)
C04F	CB07	500		RLC	A
C051	3002	510		JR	NC,L3
C053	CBE0	520		SET	4,B
C055	77	530	L3:	LD	(HL),A
C056	23	540		INC	HL
C057	7E	550		LD	A,(HL)
C058	CB07	560		RLC	A
C05A	3002	570		JR	NC,L2
C05C	CBD8	580		SET	3,B
C05E	77	590	L2:	LD	(HL),A
C05F	23	600		INC	HL
C060	7E	610		LD	A,(HL)
C061	CB07	620		RLC	A
C063	3002	630		JR	NC,L1
C065	CBD0	640		SET	2,B
C067	77	650	L1:	LD	(HL),A
C068	23	660		INC	HL
C069	7E	670		LD	A,(HL)
C06A	CB07	680		RLC	A
C06C	3002	690		JR	NC,L0
C06E	CBC8	700		SET	1,B
C070	77	710	L0:	LD	(HL),A
C071	23	720		INC	HL
C072	7E	730		LD	A,(HL)
C073	CB07	740		RLC	A
C075	3002	750		JR	NC,GRBYTE
C077	CBC0	760		SET	0,B
C079	77	770	GRBYTE:	LD	(HL),A
C07A	78	780		LD	A,B
C07B	CDA500	790		CALL	LPTOUT
C07E	ED52	800		SBC	HL,DE



C080	0D	810	DEC	C
C081	97	820	SUB	A
C082	B9	830	CP	C
C083	20A9	840	JR	NZ, KOLOM
C085	19	850	ADD	HL, DE
C086	23	860	INC	HL
C087	C1	870	POP	BC
C088	10A1	880	DJNZ	TEKEN
C08A	CD9CC0	890	CALL	MOVELIN
C08D	E1	900	POP	HL
C08E	C1	910	POP	BC
C08F	1083	920	DJNZ	REGEL
		930	; RESET PRINTER	
C091	3E1B	940	LD	A, #1B
C093	CDA500	950	CALL	LPTOUT
C096	3E40	960	LD	A, #40
C098	CDA500	970	CALL	LPTOUT
C09B	C9	980	RET	
C09C	3E0D	990	MOVELIN:	LD A, #0D
C09E	CDA500	1000	CALL	LPTOUT
C0A1	CDA500	1010	CALL	LPTOUT
C0A4	3E0A	1020	LD	A, #0A
C0A6	CDA500	1030	CALL	LPTOUT
C0A9	C9	1040	RET	
C0AA	3E1B	1050	SETGR:	LD A, #1B
C0AC	CDA500	1060	CALL	LPTOUT
C0AF	3E4B	1070	LD	A, #4B
C0B1	CDA500	1080	CALL	LPTOUT
C0B4	3E00	1090	LD	A, 0
C0B6	CDA500	1100	CALL	LPTOUT
C0B9	3E01	1110	LD	A, 1
C0BB	CDA500	1120	CALL	LPTOUT
C0BE	C9	1130	RET	
C0BF		1140	BUFFER:	EQU \$

De machinetaalroutine lijkt erg lang. Dit valt echter heel erg mee, want het grootste deel van de tekst bestaat uit ondersteunende gegevens. De linkse kolom hexadecimale getallen bevatten de adressen van de rechts daarvan gegeven machinetaalinstructies. Die instructies zijn ook weer hexadecimaal

gegeven. De rest van de informatie is een vertaling van de hexadecimale instructiecodes. Het enige wat wij behoeven in te tikken zijn de machinetaalinstructies zelf. Een programma waarmee we de machinetaalroutine eenvoudig in het geheugen kunnen zetten is het volgende:

```
100 CLEAR 200,&HBFFF
110 FOR I=0 TO 191
120 READ MC$: POKE &HC000+I,VAL("&H"+MC$)
130 NEXT I
140 CLS: PRINT "ZET CASSETTERECORDER OP
OPNAME          DRUK DAARNA OP RETURN"
150 IF INKEY$<>CHR$(13) THEN GOTO 150
160 BSAVE "CAS:SCRNDU",&HC000,&HC100:END
200 DATA 3E,1B,CD,A5,00,3E,41,CD,A5,00
210 DATA 3E,08,CD,A5,00,21,00,00,06,18
220 DATA C5,CD,AA,C0,11,BF,C0,01,00,01
230 DATA etc. etc.
```

De machinetaalinstructies zijn als items in de DATA-statements geplaatst. Door die items een voor een uit de DATA-regels te lezen en ze vervolgens in het geheugen, dat eerst is vrijgemaakt, te POKEN, wordt de machinetaalroutine in het geheugen gezet. Nu kan het geheugendeel, waarin de machinetaalroutine staat, naar cassette of schijf worden geschreven met het BSAVE-statement.

Staat de routine eenmaal op cassette of schijf, dan kan deze met behulp van het BLOAD statement weer worden teruggelezen. Een voorbeeld hiervan is te zien in het BASIC-programma waarmee een beeldscherm wordt opgebouwd. In regel 10 van dat programma wordt een stuk geheugen vanaf adres &HBFFF vrijgemaakt voor machinetaal gebruik. Met regel 20 wordt de machinetaalroutine vanaf cassette of schijf geladen en in het vrijgemaakte geheugendeel gezet. Met regel 40 wordt het startadres van de machinetaalroutine vastgelegd. Die machinetaalroutine wordt in regelnummer 80 gestart.

In de machinetaalroutine zelf worden een aantal commando's naar de printer gestuurd, zoals het instellen van de regelaf-

stand, het resetten van de printer naar initiële instellingen en het instellen van de grafische mode.

De derde kolom van de machinetaal-listing bevat regelnummers. In het vervolg van deze uitleg zal ik refereren naar die regelnummers.

Op regelnummer 70 start het instellen van de regelafstand. U ziet, dat er achtereenvolgens de codes 1B, 41 en 08 naar de printer worden gestuurd (CALL LPTOUT). Zouden we, in plaats van een Seikosha SP800, een MSX-printer willen gebruiken, dan dienen we de codes 1B en 42 naar de printer te sturen. Dit is te bereiken door de machinetaalinstructiecodes van de regels 100 tot en met 130 als volgt te wijzigen:

is:	wordt:
3E41	3E42
CDA500	CDA500
3E08	0000
CDA500	000000

Op regelnummer 1050 start het instellen van de grafische mode. Voor de MSX-printer moet achtereenvolgens CHR\$(27);"G0600256"; worden verstuurd. Dit zijn aanzienlijk meer tekens dan in het geven machinetaalroutine. De nodige wijzigingen vanaf regelnummer 1070 zijn als volgt:

is:	wordt:	
3E4B	3E47	(47 = G)
CDA500	CDA500	
3E00	3E30	(30 = 0)
CDA500	CDA500	
3E01	3E36	(36 = 6)
CDA500	CDA500	
	3E30	(30 = 0)
	CDA500	
	3E30	(30 = 0)
	CDA500	
	3E32	(32 = 2)
	CDA500	

	3E35	(35 = 5)
	CDA500	
	3E36	(36 = 6)
	CDA500	
C9	C9	

Er moeten, zoals u ziet een aantal machinetaalroutines worden toegevoegd. Dit is op zich geen probleem, echter, na deze toegevoegde machinetaal instructies start ht buffer. We zullen daarom het startadres van het buffer overal waar dat wordt gebruikt ook moeten aanpassen. Vandaar dat de volgende regelnummers nog moeten worden gewijzigd:

	is:	wordt:
190	11BFC0	11D8C0
250	21BFC0	21D8C0

Met deze wijzigingen werkt de routine ook op een MSX-printer. Mocht u deze machinetaalroutine proberen, nadat u eerst de in BASIC geschreven afdruk routines hebt geprobeerd, dan zult u versteld staan van de snelheid. Er valt met de machinetaalroutine een snelheid van ongeveer 120 regels per minuut te bereiken. Met andere woorden, een afdruk van een scherm van 24 regels duurt slechts 12 seconden, tenzij uw printer langzamer is dan 120 regels per minuut.

#### **6.4 Kleurenafdruk in grafische mode.**

In deze paragraaf volgt nog een voorbeeld van een afdruk routine waarmee een kleurenafdruk van het scherm wordt verkregen op een OKIMATE 20 printer. Deze routine is weer geheel in BASIC en daardoor niet zo heel erg snel. Er zijn twee versies gegeven, een voor de high resolution mode en een voor de low resolution mode. Het maken van een hard copy van een beeldscherm in high resolution mode duurt ongeveer 28 minuten, voor de low resolution mode geldt een afdruktijd van ongeveer 7 minuten. Hier volgen de beide versies van het programma.

```

10 '*****
20 '* AFDRUKKEN VAN KLEUREN OP DE *
30 '* OKIMATE 20 KLEUREN PRINTER. *
40 '*****
50 '
60 DEFINT A-Z
70 CLEAR 1500
80 '
100 '*****
110 'MSX:                OKI:
120 '
130 '                geel rood blauw
140 'transp  0000  0    0    0    wit
150 'zwart   0001  1    1    1    zwart
160 'groen   0010  1    0    1    groen
170 'l.groen 0011  1    0    1    groen
180 'd.blauw 0100  0    1    1    paars
190 'l.blauw 0101  0    0    1    blauw
200 'd.rood  0110  0    1    1    paars
210 'blauw   0111  0    0    1    blauw
220 'rood    1000  0    1    0    rood
230 'l.rood  1001  1    1    0    oranje
240 'geel    1010  1    0    0    geel
250 'l.geel  1011  1    0    0    geel
260 'd.groen 1100  1    0    1    groen
270 'rood    1101  0    1    0    rood
280 'grijs   1110  1    1    1    zwart
290 'wit     1111  0    0    0    wit
300 '*****
310 '
320 DIM KTAB$(2,16)
330 FOR J=0 TO 2
340 FOR I=0 TO 15
350 READ KTAB$(J,I)
360 NEXT I:NEXT J
370 'geel
380 DATA 0,1,1,1,0,0,0,0,0,1,1,1,1,0,1,0
390 'rood
400 DATA 0,1,0,0,1,0,1,0,1,1,0,0,0,1,1,0
410 'blauw

```

```

420 DATA 0,1,1,1,1,1,1,1,0,0,0,0,1,0,1,0
430 '
500 '*****
510 '*      OPBOUWEN GRAFISCH SCHERM      *
520 '*****
530 '
540 SCREEN 2
550 COLOR 1,10
560 CLS
570 OPEN "grp:" AS #1
580 DRAW "bm0,0"
590 PRINT #1,"HARD COPY VAN HIGH RES SCHERM"
600 FOR I=1 TO 30
610 X=RND(1)*256:Y=RND(1)*152+40
620 Z=RND(1)*50:K=RND(1)*16
630 CIRCLE (X,Y),Z,K
640 NEXT I
650 CLOSE #1
660 COLOR 4,11
670 '
700 '*****
710 '* INITIALISEREN PRINTER EN AF- *
720 '* DRUKKEN VAN DE INHOUD VAN HET *
730 '* SCHERM (IN GRAFISCHE MODE). *
740 '*****
750 '
760 LPRINT CHR$(27);"3";CHR$(24);
770 LPRINT CHR$(13);CHR$(10);
780 FOR L%= 0 TO 23
790 DIM P$(2)
800 GOSUB 1100:'lees pixels
810 GOSUB 2110:'print pixels
820 ERASE P$
830 NEXT L%
840 END
850 '
1000 '*****
1010 '*      LEZEN VAN EEN PIXEL      *
1020 '*-----*
```

```

1030 '* Van ieder pixel wordt de kleur*
1040 '* bepaald en geconverteerd naar *
1050 '* (m.b.v. KTAB) printer kleuren *
1060 '* De printerkleuren worden in *
1070 '* arrays P$ opgeslagen. *
1080 '*****
1090 '
1100 FOR X%=0 TO 250
1110 PG$="":PR$="":PB$=""
1120 FOR Y%=0 TO 7
1130 K%=POINT(X%,Y%+8*L%)
1140 PG$=PG$+KTAB$(0,K%)
1150 PR$=PR$+KTAB$(1,K%)
1160 PB$=PB$+KTAB$(2,K%)
1170 NEXT Y%
1180 P$(0)=P$(0)+CHR$(VAL("&B"+PG$))
1190 P$(1)=P$(1)+CHR$(VAL("&B"+PR$))
1200 P$(2)=P$(2)+CHR$(VAL("&B"+PB$))
1210 NEXT X%
1220 RETURN
1230 '
2000 '*****
2010 '* KLEUREN AFDRUK ROUTINE *
2020 '*-----*
2030 '* INPUT: P$(0)=GEEL *
2040 '* P$(1)=ROOD *
2050 '* P$(2)=BLAUW *
2060 '*-----*
2070 '* De lengte van P$ mag variëren *
2080 '* tussen 1 en 255. *
2090 '*****

2100 '
2110 LPRINT CHR$(27);CHR$(25);:'zet lint
    op begin
2120 FOR K%=0 TO 2
2130 LPRINT TAB(10);
2140 G%=LEN(P$(K%))
2150 LPRINT CHR$(27);"K";CHR$(G%);CHR$(0
);:'zet grafisc

```

```

he mode voor g tekens
2160 LPRINT P$(K%);CHR$(13);
2170 NEXT K%
2180 LPRINT CHR$(10);
2190 RETURN

```

```

10 '*****
20 '* AFDRUKKEN VAN KLEUREN OP DE *
30 '* OKIMATE 20 KLEUREN PRINTER. *
40 '*****
50 '
60 DEFINT A-Z
70 CLEAR 1500
80 '
100 '*****
110 'MSX:                OKI:
120 '
130 '                geel rood blauw
140 'transp  0000  0    0    0    wit
150 'zwart   0001  1    1    1    zwart
160 'groen   0010  1    0    1    groen
170 'l.groen 0011  1    0    1    groen
180 'd.blauw 0100  0    1    1    paars
190 'l.blauw 0101  0    0    1    blauw
200 'd.rood  0110  0    1    1    paars
210 'blauw   0111  0    0    1    blauw
220 'rood    1000  0    1    0    rood
230 'l.rood  1001  1    1    0    oranje
240 'geel    1010  1    0    0    geel
250 'l.geel  1011  1    0    0    geel
260 'd.groen 1100  1    0    1    groen
270 'rood    1101  0    1    0    rood
280 'grijs   1110  1    1    1    zwart
290 'wit     1111  0    0    0    wit
300 '*****

```



```

310 '
320 DIM KTAB$(2,16)
330 FOR J=0 TO 2
340 FOR I=0 TO 15
350 READ KTAB$(J,I)
360 NEXT I:NEXT J
370 'geel
380 DATA 0,1,1,1,0,0,0,0,0,1,1,1,1,0,1,0
390 'rood
400 DATA 0,1,0,0,1,0,1,0,1,1,0,0,0,1,1,0
410 'blauw
420 DATA 0,1,1,1,1,1,1,1,0,0,0,0,1,0,1,0
430 '
500 '*****
510 '*      OPBOUWEN GRAFISCH SCHERM      *
520 '*****
530 '
540 SCREEN 3
550 COLOR 1,10
560 CLS
570 OPEN "grp:" AS #1
580 DRAW "bm0,0"
590 FOR I=0 TO 15:COLOR I:PRINT #1,"****
";:NEXT I
600 CLOSE #1
610 COLOR 4,11
620 '
700 '*****
710 '* INITIALISEREN PRINTER EN AF- *
720 '* DRUKKEN VAN DE INHOUD VAN HET *
730 '* SCHERM (IN GRAFISCHE MODE). *
740 '*****
750 '
760 LPRINT CHR$(27);"3";CHR$(24);
770 LPRINT CHR$(13);CHR$(10);
780 FOR L%= 0 TO 23
790 DIM P$(2)
800 GOSUB 1110:'lees pixels
810 GOSUB 2110:'print pixels

```

```

820 ERASE P$
830 NEXT L%
840 END
850 '
1000 '*****
1010 '*           LEZEN VAN EEN PIXEL           *
1020 '*-----*
1030 '* Uit ieder blok van 4*4 pixels *
1040 '* wordt de kleur van een pixel *
1050 '* bepaald en geconverteerd naar *
1060 '* (m.b.v. KTAB) printer kleuren *
1070 '* De printerkleuren worden in *
1080 '* arrays P$ opgeslagen.           *
1090 '*****
1100 '
1110 FOR X%=0 TO 250 STEP 4
1120 PG$="":PR$="":PB$=""
1130 FOR Y%=0 TO 7 STEP 4
1140 K%=POINT(X%,Y%+8*L%)
1150 PG$=PG$+KTAB$(0,K%)+KTAB$(0,K%)+KTA
B$(0,K%)+KTAB$(
0,K%)
1160 PR$=PR$+KTAB$(1,K%)+KTAB$(1,K%)+KTA
B$(1,K%)+KTAB$(
1,K%)
1170 PB$=PB$+KTAB$(2,K%)+KTAB$(2,K%)+KTA
B$(2,K%)+KTAB$(
2,K%)
1180 NEXT Y%
1190 FOR I=0 TO 3
1200 P$(0)=P$(0)+CHR$(VAL("&B"+PG$))
1210 P$(1)=P$(1)+CHR$(VAL("&B"+PR$))
1220 P$(2)=P$(2)+CHR$(VAL("&B"+PB$))
1230 NEXT I
1240 NEXT X%
1250 RETURN
1260 '
2000 '*****
2010 '*           KLEUREN AFDRUK ROUTINE           *
2020 '*-----*

```

```

2030 '*      INPUT:  P$(0)=GEEL      *
2040 '*          P$(1)=ROOD      *
2050 '*          P$(2)=BLAUW      *
2060 '*-----*
2070 '* De lengte van P$ mag variëren *
2080 '* tussen 1 en 255.          *
2090 '*****
2100 '
2110 LPRINT CHR$(27);CHR$(25);:'zet lint
      op begin
2120 FOR K%=0 TO 2
2130 LPRINT TAB(10);
2140 G%=LEN(P$(K%))
2150 LPRINT CHR$(27);"K";CHR$(G%);CHR$(0
);:'zet grafisc
he mode voor g tekens
2160 LPRINT P$(K%);CHR$(13);
2170 NEXT K%
2180 LPRINT CHR$(10);
2190 RETURN

```

Om het programma eenvoudig te houden, en dit is nodig in verband met de uitvoeringssnelheid, worden er slechts acht verschillende kleuren op de printer afgedrukt. De printer heeft een inktlint, waarop stukken geel, blauw en rood lint zitten. Door de verschillende kleuren over elkaar heen te drukken ontstaan nieuwe kleuren (geel + blauw = groen, etc.). In totaal zijn op deze manier echter slechts 8 kleuren te maken. De MSX-computer heeft echter 16 kleuren. De 16 kleuren van de computer moeten dus worden geconverteerd naar de 8 kleuren van de printer. Daartoe wordt in het programma gebruik gemaakt van een conversietabel (zie regelnummers 380, 400 en 420 en de REM-regels 100 tot en met 300).

Daar het er in dit boek niet om gaat u te leren programmeren, zal ik u de verdere uitleg van het programma besparen. Met het in het programma opgenomen commentaar hoop ik dat u er uit kunt komen. De besturing van de printer verdient

nog wel enige verdere uitleg.

In regelnummer 760 wordt de regelafstand ingesteld. Hierna wordt met regelnummer 770 naar de volgende regel gegaan. Voor iedere af te drukken regel wordt de afdrukroutine eenmaal aangeropen. Deze routine start op regelnummer 2110 met het klaarzetten van het lint. Het lint bestaat namelijk uit stukken geel, rood en blauw, die zich in diezelfde volgorde herhalen. Ieder stuk lint heeft de lengte van een hele regel. Is er met geel afgedrukt, dan zal het lint worden verplaatst totdat er een stuk rood voor het papier zit. Daarna wordt blauw voor het papier gebracht, enz. Er moet echter altijd met dezelfde kleur worden begonnen, en wel met geel. Met het commando uit regel 2110 wordt het gele lintstuk voor het papier gebracht.

Met regel 2150 wordt de printer in de grafische mode gebracht. De letter K selecteert de grafische mode, terwijl de variabele G% het aantal kolommen, dat zal worden verstuurd, bevat. Dit werkt in grote lijnen dus net als in de vorige paragraaf werd uiteengezet. Nadat er drie kleuren zijn afgedrukt (K=0 to 2), wordt een regel opgeschoven (regelnummer 2180) en wordt teruggekeerd naar het hoofdprogramma.

Daar een kleurenafdruk in dit boekje de prijs van dit boekje te ver zou opdrijven, is dat hier weggelaten. Mocht u echter over een OKIMATE 20 beschikken, dan raad ik u aan bovenstaande programma's eens in te tikken en uit te voeren. Zelf was ik verbaasd over de prachtige kleuren die er ontstonden.

## 7 Down loading van tekens

In dit laatste hoofdstuk zullen we nader ingaan op een faciliteit die op lang niet alle printers aanwezig is. Bovendien is de manier van aansturen van deze faciliteit van printer tot printer nogal verschillend. Waarom dan toch dit onderwerp zo uitgebreid behandelen? Omdat het zulke fraaie effecten tot gevolg heeft. In dit hoofdstuk zullen we zien hoe we een Epson FX80+ printer van een door ons zelf gedefinieerde tekenset kunnen voorzien. Straks zullen we een programma zien, waarmee op eenvoudige en zeer snelle wijze een nieuwe tekenset kan worden gewaardeerd. Nu al wil ik verwijzen naar het te verkrijgen resultaat. Zie daarvoor afbeelding 7-1. De daarin afgedrukte tekst is samengesteld uit letters, waarvan ik zelf de vorm heb bepaald. U ziet, dat er een grote variatie mogelijk is. Het definiëren van andere vormen voor alle kleine letters van het alfabet kostte me ongeveer 20 minuten tijd. Dit is, zoals u straks zult zien, slechts een fractie van de tijd die nodig zou zijn, wanneer er niet van een programma.

Deze tekst is geschreven in een lettertype dat is gedefinieerd in ongeveer 20 minuten. De letters staan in een file op de schijf. Het laden van de letters uit de file naar de printer toe (met functie 4 uit het menu) duurt slechts enkele seconden.

Afb. 7-1 Zelf gedefinieerde lettervormen.

### 7.1 De theorie

In hoofdstuk 3 werd uitgelegd hoe de matrix printer een teken op het papier afdrukt. De van de computer ontvangen ASCII-code wordt gebruikt bij het adresseren van de teken ROM. In die ROM zit de matrix van puntjes, die de vorm van de bij de

ASCII-code behorende letter bepaalt. De matrix van puntjes wordt kolom voor kolom naar de naaldjes van de printkop gestuurd.

Het kenmerkende van een ROM is, dat er alleen maar gegevens uit kunnen worden gelezen. De door de fabriek bepaalde inhoud van de ROM bepaalt dus de vorm van de afgedrukte tekens. Zouden de matrices van puntjes in een RAM-geheugen staan, dan zouden we de inhoud van het RAM en dus de vorm van de afgedrukte tekens kunnen wijzigen. Het nadeel van een RAM-geheugen is echter, dat het zijn inhoud verliest zodra de spanning wordt uitgeschakeld. Vandaar, dat de tekens altijd zijn opgeslagen in een ROM.

Er zijn echter een aantal printers, die zowel een tekenset in ROM hebben, als een stuk RAM-geheugen, waarin nieuwe tekenvormen kunnen worden opgeslagen. Die nieuwe tekens moeten dan vanuit de computer naar de printer worden gestuurd. Staan die tekens eenmaal in het RAM van de printer, dan zitten er in feite twee verschillende tekensets parallel aan elkaar in de printer. De vraag is nu, welke van de twee tekensets zal worden gebruikt voor het afdrukken van de letters. Wij zullen de printer moeten vertellen welke van beide tekensets moet worden gebruikt. Met andere woorden, we moeten de down load character set activeren.

Daar er, zowel voor wat betreft de interne organisatie van tekenmatrices als ten aanzien van de aansturing van ROM en RAM, nogal aanzienlijke verschillen bestaan tussen printers van verschillende fabrikaten, zullen we ons vanaf nu beperken tot het bespreken van de manier, waarop down loading van tekens werkt, op de Epson printers.

Omdat we meestal niet de gehele tekenset (van 128 tekens) opnieuw willen definiëren, heeft Epson de mogelijkheid geschapen om de ROM-inhoud te copieren naar het RAM. Vervolgens kunnen we met behulp van een commando (ESC-reeks) opgeven welke tekens we willen herdefiniëren, waarna de printer de opnieuw te definiëren tekens vanuit de computer verwacht. Zijn alle gewenste tekens naar de printer ver-

stuurd, dan dient nog een commando (ESC-reeks) naar de printer te worden gezonden, waarmee de printer wordt verteld, of de af te drukken tekens vanuit het ROM of vanuit het RAM moeten worden geselecteerd. Het gebruik van down load characters heeft dus drie stappen nodig, en wel:

- Copieer ROM naar RAM.
- Definieer nieuwe tekens in het RAM.
- Selecteer matrices uit RAM of uit ROM.

Tot zover de theoretische achtergronden. De volgorde van de drie stappen is belangrijk. U doet er goed aan die drie stappen goed te onthouden. Hoe we die drie stappen toe passen zullen we in de volgende paragraaf zien, aan de hand van een voorbeeld programma.

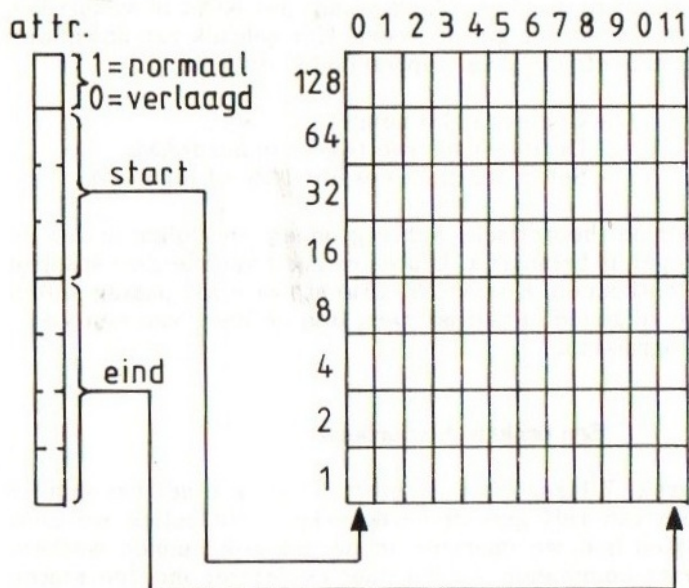
## **7.2 Een praktisch voorbeeld**

In paragraaf 7.1 zagen we de grote lijnen van het bepalen en gebruiken van zelf gedefinieerde tekens. Nu zullen we eens gaan kijken hoe we daarmee in de praktijk kunnen werken. Het eerste commando dat we naar de printer moeten sturen (indien we tenminste slechts een gedeelte van de gehele tekenset willen herdefinieren), is de opdracht om de tekenset vanuit ROM naar RAM te copieren. Dat commando ziet er als volgt uit:

```
LPRINT CHR$(27);":";CHR$(0);CHR$(0);CHR$(0);
```

Na dit commando zit de standaard tekenset van de printer zowel in het ROM van de printer als in het RAM van de printer. Zouden we nu een tekst afdrukken, dan zouden we nog geen enkel verschil zien, zelfs wanneer de printer de tekens vanuit het RAM zou afdrukken. We kunnen nu echter een letter (of indien gewenst meerdere letters) een andere vorm geven, door een nieuwe matrix van puntjes in het RAM te zetten op de plaats van een uit het ROM gecopieerde matrix. Voordat we dat gaan doen, zullen we moeten weten hoe een matrix van puntjes voor een teken er in het ROM precies

uitziet. Zo'n matrix is in afbeelding 7-2 weergegeven.



$$\text{eind} \leq 11 ; (\text{eind} - \text{start}) \geq 5$$

Afb. 7-2 Tekenmatrix voor down load characters.

We zien in afbeelding 7-2, dat een teken is opgebouwd uit een matrix van 12 kolommen (van 0 tot en met 11), die wordt voorafgegaan door een kolom waarin nadere gegevens over de volgende tekendefinitie staan. Die extra kolom wordt het attribuut genoemd. Verder zien we, dat iedere kolom uit 8 bitjes bestaat. Het is dus mogelijk om de kolommen van de computer naar de printer te sturen (er kunnen immers steeds acht bits tekens van de computer naar de printer worden gestuurd).

Stel nu eens, dat we de hoofdletter A willen veranderen. We dienen dan een commando samen te stellen, waarmee we de printer vertellen welke letter we willen veranderen en hoe die



letter er moet gaan uitzien. Dat commando zou er dan als volgt uitzien:

```
LPRINT CHR$(27);"&";CHR$(0);"AA";  
LPRINT CHR$(attribuut);  
LPRINT CHR$(kolom 1);....;CHR$(kolom n);
```

Met de eerste regel wordt de printer verteld dat er tekens moeten worden gedefinieerd en wel de tekens A tot en met A. In dit geval dus alleen het teken A. Zou u alle hoofdletters van het alfabet willen herdefinieren, dan zou u "AA" moeten vervangen door "AZ". Voor ieder opnieuw te definiëren teken dienen de tweede en derde regel uit voorgaand voorbeeld te worden herhaald. Er dient dus voor ieder te definiëren teken een attribuut byte plus de bytes met de kolommeninformatie te worden verstuurd.

Is een teken eenmaal opnieuw gedefinieerd, dan staat dat nieuwe teken in het RAM van de printer. Er wordt echter nog niets mee gedaan. Indien we nu een tekst zouden afdrukken, zouden we nog steeds de oude vorm van het teken te zien krijgen. We moeten de printer immers nog vertellen dat de vorm van de tekens uit het RAM moet worden gelezen, en niet meer uit het ROM. Daartoe geven we de printer het volgende commando:

```
LPRINT CHR$(27);"%" ; CHR$(1);CHR$(0);
```

Alles wat we nu afdrukken zal vanuit het RAM worden samengesteld. Het gevolg zal dus zijn, dat de door ons gedefinieerde vorm van de hoofdletter A zal worden afgedrukt. Willen we dat de printer de tekens weer samenstelt vanuit het ROM, dan dienen we het volgende commando te geven:

```
LPRINT CHR$(27);"%" ; CHR$(0);CHR$(0);
```

Het volgende programma laat zien hoe een en ander precies in zijn werk gaat. Het programma is zo eenvoudig mogelijk gehouden, opdat u de principiële werking gemakkelijk kunt doorgronden.

```

100 LPRINT "Akkermans"
110 LPRINT CHR$(27);".";CHR$(0);CHR$(0);CHR$(0);
120 LPRINT CHR$(27);"&";CHR$(0);"AA";
130 LPRINT CHR$(139);
140 RESTORE
150 FOR I=1 TO 11
160 READ K: LPRINT CHR$(K);
170 NEXT I
180 LPRINT CHR$(27);"%";CHR$(1);CHR$(0);
190 LPRINT "Akkermans"
200 LPRINT "AAAAAAAAAAAA"
210 END
220 DATA "255,0,225,0,139,48,139,0,225,0,255

```

Voorbeeld van het resultaat van dit programma:

```

Akkermans
Akkermans
AAAAAAAAAAAA

```

Wat hebben we nu precies gedaan in voorgaand programma? Met regel 110 hebben we de inhoud van het ROM naar RAM gecopieerd. Met regel 120 hebben we de printer verteld dat we de hoofdletter A een nieuwe vorm willen geven. Met regel 130 versturen we het attribuut-byte. In dit geval is de waarde 139 opgegeven.

De waarde 139 bestaat uit  $128 + 8 + 2 + 1$ . Vult u deze waarden in in het attribuut byte in afbeelding 7-2, dan ziet u dat de waarde 128 wil zeggen, dat de negende naald van de printkop niet wordt gebruikt. Dit wil zeggen dat alleen de bovenste acht naaldjes worden gebruikt voor het afdrucken van het teken. Zou de waarde 128 niet in het attribuut zitten, dan zouden de onderste acht naaldjes worden gebruikt. In dat geval zou het hele teken dus wat zakken. Men spreekt dan van een descender-teken, ofwel een verlaagd teken (nodig voor de letters g, j, p, q en y).

Verder zien we, bij het invullen van de waarden in afbeelding

ATTR	0	1	2	3	4	5	6	7	8	9	0	1
	128											
	64											
	32											
	16											
	8											
	4											
	2											
	1											

ATTR	0	1	2	3	4	5	6	7	8	9	0	1
	128											
	64											
	32											
	16											
	8											
	4											
	2											
	1											

ATTR	0	1	2	3	4	5	6	7	8	9	0	1
	128											
	64											
	32											
	16											
	8											
	4											
	2											
	1											

ATTR	0	1	2	3	4	5	6	7	8	9	0	1
	128											
	64											
	32											
	16											
	8											
	4											
	2											
	1											

ATTR	0	1	2	3	4	5	6	7	8	9	0	1
	128											
	64											
	32											
	16											
	8											
	4											
	2											
	1											

ATTR	0	1	2	3	4	5	6	7	8	9	0	1
	128											
	64											
	32											
	16											
	8											
	4											
	2											
	1											

ATTR	0	1	2	3	4	5	6	7	8	9	0	1
	128											
	64											
	32											
	16											
	8											
	4											
	2											
	1											

ATTR	0	1	2	3	4	5	6	7	8	9	0	1
	128											
	64											
	32											
	16											
	8											
	4											
	2											
	1											

ATTR	0	1	2	3	4	5	6	7	8	9	0	1
	128											
	64											
	32											
	16											
	8											
	4											
	2											
	1											

ATTR	0	1	2	3	4	5	6	7	8	9	0	1
	128											
	64											
	32											
	16											
	8											
	4											
	2											
	1											

ATTR	0	1	2	3	4	5	6	7	8	9	0	1
	128											
	64											
	32											
	16											
	8											
	4											
	2											
	1											

ATTR	0	1	2	3	4	5	6	7	8	9	0	1
	128											
	64											
	32											
	16											
	8											
	4											
	2											
	1											

7-2, dat we de kolommen 0 tot en met 11 willen definiëren. Dit zijn 12 kolommen. Toch versturen we er maar 11 (met regel 160). De twaalfde kolom mogen we namelijk niet versturen. Die moet altijd leeg zijn en wordt automatisch door de printer ingevuld.

Het opgeven van de start- en eindkolommen is op twee punten van belang. Ten eerste wordt er de plaats van het teken binnen de twaalf beschikbare kolommen mee bepaald. Bij normale afdruk worden alle twaalf kolommen afgedrukt. Een lege kolom heeft dan alleen een verplaatsing van de printkop tot gevolg. Bij het proportioneel afdrucken worden echter niet meer kolommen afgedrukt dan in het attribuut byte staan aangegeven. Indien de startkolom 3 is, en de eindkolom 9, dan zullen alleen de kolommen 3 tot en met 9 worden afgedrukt.

De zo afgedrukte letter zal dan slechts 7 kolommen breed zijn. Een andere letter kan van kolom 0 tot en met 11 gaan en 12 kolommen breed zijn. Op die manier kunnen verschillende tekens ook werkelijk verschillende breedtes op het papier innemen. De minimale breedte van een teken is 5 kolommen. Hiermee dient u bij het opgeven van begin- en eindkolom rekening te houden.

De waarden van de kolommen staan in het DATA-statement van regel 220. Die waarden worden met regel 160 ingelezen en naar de printer gestuurd. Vult u die waarden in in de kolommen van afbeelding 7-2, dan zult u de vorm van het teken herkennen. Als u goed oplet, ziet u nog iets. In een horizontale rij zijn nooit twee punten vlak naast elkaar gedefinieerd. Er moet namelijk altijd een blanco punt tussen twee punten staan. De reden hiervoor is, dat de blanco plaatsen worden gebruikt bij het afdrucken van tekst in "double width".

Na het definiëren van de nieuwe tekenvorm, wordt het RAM geactiveerd met behulp van regel 180. Vervolgens worden een paar teksten afgedrukt, waaruit blijkt wat het resultaat van het programma is. Indien u hierna een listing van het programma zou maken, zonder de printer tussentijds uit te zetten, dan zou u zien, dat alle hoofdletters A in die listing

vervangen zijn door de nieuw gedefinieerde tekenvorm.

Het definiëren van tekens is een tamelijk moeizame bezigheid. Om u enigszins te helpen bij het bepalen van nieuwe vormen heb ik een pagina met invul-matrices opgenomen. U zou dit ook op een ruitjesblok kunnen maken. Na het invullen (met potlood, om nog correcties te kunnen aanbrengen) dient u de waarden van de kolommen en het attribuut byte te bepalen. Al gauw zult u tot de ontdekking komen, dat dit een langdurig karwei is. Bovendien bent u de nieuwe vormen weer kwijt zodra u de printer uitzet. Om het geheel wat gemakkelijker te maken wordt dit hoofdstuk besloten met een programma, waarmee de tekens kunnen worden samengesteld op het beeldscherm en opgeslagen op schijf (of met en kleine wijziging op cassette).

### 7.3 Een down load character editor

Daar ik zelf niet in het bezit ben van een Epson printer, heb ik het volgende programma in zeer korte tijd moeten schrijven en testen, toen ik in de gelegenheid was om zo'n printer voor een paar dagen te lenen. Het gevolg is dan ook, dat het programma niet een toonbeeld van goede programmering is geworden. Het werkt echter goed, en doet wat er van wordt verwacht.

```
10 MAXFILES=3
20 DIM P$(12,8)
30 DATA 0,0,62,62,62,62,62,0
40 S$=""
50 FOR I=1 TO 8
60 READ S: S$=S$+CHR$(S)
70 NEXT I
80 SCREEN 0: WIDTH 40
90 CLS
100 PRINT "DOWN LOAD CHARACTER CODES:"
110 PRINT:PRINT
120 PRINT "1  afdrukken op beeldscherm"
130 PRINT "2  afdrukken op de printer"
```

```

140 PRINT "3   opslaan op disk"
150 PRINT "4   copieren van disk naar pr
inter"
160 PRINT:PRINT
170 INPUT "Nummer van uw keuze";F
180 IF F<1 OR F>4 THEN GOTO 90
190 IF F<4 THEN GOSUB 1740
200 OPEN "GRP:"AS #1
210 IF F=2 THEN OPEN "LPT:"AS #2
220 IF F=3 THEN GOSUB 290
230 IF F=4 THEN GOTO 1850
240 FOR L=ASC(E$) TO ASC(L$)
250 GOSUB 430
260 ON F GOSUB 1210,1210,1430
270 NEXT L
280 END
290 INPUT "Te gebruiken file-naam";N$
300 OPEN "a:"+N$+".dlc" FOR OUTPUT AS #3
310 PRINT #3,27
320 PRINT #3,ASC("&")
330 PRINT #3,0
340 PRINT #3,ASC(E$)
350 PRINT #3,ASC(L$)
360 RETURN
370 '
380 '*****
390 '* Opbouwen grafisch scherm voor *
400 '*   herdefinitie van tekens.   *
410 '*****
420 '
430 SCREEN 2
440 SPRITE$(1)=S$
450 A=0: B=0: E=0
460 DRAW "BM10,10"
470 PRINT #1,"verlaagde letter? ";
480 I$=INKEY$: IF I$="" THEN GOTO 480
490 PRINT #1,I$
500 IF I$="n" OR I$="N" THEN A=A+128
510 DRAW "BM10,25"
520 PRINT #1,"Eerste kolom?      ";

```

```

530 I$=INKEY$: IF I$="" THEN GOTO 530
540 IF I$<"0" OR I$>"9" THEN GOTO 530
550 PRINT #1,I$
560 B=VAL(I$)
570 A=A+VAL(I$)*16
580 DRAW "BM10,40"
590 J$=""
600 PRINT #1,"Laatste kolom? ";
610 FOR I=1 TO 2
620 I$=INKEY$: IF I$="" THEN GOTO 620
630 IF I$<"0" OR I$>"9" THEN GOTO 620
640 J$=J$+I$
650 PRINT #1,I$;
660 NEXT I
670 E=VAL(J$)
680 IF E-B<5 THEN CLS: GOTO 450
690 IF E>11 THEN LINE (150,40)-(170,50),
4,BF: GOTO 580
700 A=A+VAL(J$)
710 DRAW "BM10,55"
720 PRINT #1,"Attribute byte = ";A
730 FOR I=1 TO 500: NEXT I
740 CLS
750 FOR I=20 TO 100 STEP 10
760 LINE (20+B*8,I)-(20+(E+1)*8,I)
770 NEXT I
780 FOR I=20+B*8 TO 20+(E+1)*8 STEP 8
790 LINE (I,20)-(I,100)
800 NEXT I
810 DRAW "BM22,10"
820 PRINT #1,"012345678901"
830 DRAW "BM0,160"
840 PRINT #1," VERPLAATSEN MET CURSOR-TO
ETSSEN"
850 DRAW "BM0,180"
860 PRINT #1," RETURN = PUNT ESC = EIND
E DEF"
870 '
880 '*****
890 '* Initialiseren van de array P$ *
```

```

900 '*****
910 '
920 FOR I=0 TO 12
930 FOR J=0 TO 7
940 P$(I,J)="0"
950 NEXT J
960 NEXT I
970 '
980 '*****
990 '* Aan/uit zetten van de punten *
1000 '* van het down-load character. *
1010 '*****
1020 '
1030 I=B: J=0
1040 PUT SPRITE 1,(20+8*I,20+10*J),3
1050 I$=INKEY$: IF I$="" THEN GOTO 1050
1060 IF I$=CHR$(13) THEN GOSUB 1560
1070 IF I$=CHR$(28) AND I<E THEN I=I+1
1080 IF I$=CHR$(29) AND I>B THEN I=I-1
1090 IF I$=CHR$(30) AND J>0 THEN J=J-1
1100 IF I$=CHR$(31) AND J<7 THEN J=J+1
1110 IF I$=CHR$(27) THEN GOTO 1130
1120 GOTO 1040
1130 CLS: PUT SPRITE 1,(0,209)
1140 RETURN
1150 '
1160 '*****
1170 '* Afdrukken van de down-load *
1180 '* attribuut en character codes *
1190 '*****
1200 '
1210 DRAW "BM0,0"
1220 PRINT #F,"De codes voor ";CHR$(L)
1230 PRINT #F,"-----"
1240 PRINT #F,"CHR$(";"A;") = attribuut"
1250 DRAW "BM0,40"
1260 FOR I=B TO E
1270 C=0
1280 FOR J=7 TO 0 STEP -1
1290 IF P$(I,J)="1" THEN C=C+2^(7-J)

```



```

1300 NEXT J
1310 PRINT #F, "CHR$(" ; C ; ")"
1320 NEXT I
1330 PRINT #F,
1340 PRINT #1, "RETURN = volgende definit
ie"
1350 IF INKEY$ <> CHR$(13) GOTO 1350
1360 CLS
1370 RETURN
1380 '
1390 '*****
1400 '* kolom-codes naar disk-file *
1410 '*****
1420 '
1430 PRINT #3, A
1440 FOR I=B TO E-1
1450 C=0
1460 FOR J=7 TO 0 STEP -1
1470 IF P$(I, J)="1" THEN C=C+2^(7-J)
1480 NEXT J
1490 PRINT #3, C
1500 NEXT I
1510 DRAW "BM0,0"
1520 PRINT #1, "RETURN = volgende definit
ie"
1530 IF INKEY$ <> CHR$(13) THEN GOTO 1530
1540 CLS
1550 RETURN
1560 '
1570 '*****
1580 '* Punt zetten of terugzetten *
1590 '*****
1600 '
1610 IF P$(I, J)="1" THEN P$(I, J)="0": GO
TO 1660
1620 IF I>0 THEN GOTO 1640
1630 IF P$(I, J)="0" AND P$(I+1, J)="0" TH
EN P$(I, J)="1": GOTO 1660
1640 IF I=E THEN GOTO 1680
1650 IF P$(I, J)="0" AND P$(I-1, J)="0" AN

```

```

D P$(I+1,J)="0" THEN P$(I,J)="1"
1660 IF P$(I,J)="0" THEN LINE (21+I*8,21
+J*10)-(27+I*8,29+J*10),4,BF
1670 IF P$(I,J)="1" THEN LINE (21+I*8,21
+J*10)-(27+I*8,29+J*10),15,BF
1680 RETURN
1690 '
1700 '*****
1710 '* Tekens definieren of lezen *
1720 '*****
1730 '
1740 PRINT "Welke tekens wilt u herdefin
ieren?"
1750 INPUT "Eerste teken";E$
1760 INPUT "Laatste teken";L$
1770 IF LEN(E$)>1 OR LEN(L$)>1 THEN GOTO
80
1780 CLS
1790 RETURN
1800 '
1810 '*****
1820 '* Download characters to printer *
1830 '*****
1840 '
1850 INPUT "Welke file moet worden gelez
en";N$
1860 OPEN "a:"+N$+".dlc" FOR INPUT AS #3
1870 OPEN "LPT:" AS #2
1880 PRINT #2,CHR$(27);": ";CHR$(0);CHR$(
0);CHR$(0);: 'copy ROM to RAM
1890 IF EOF(3)=-1 THEN GOTO 1930
1900 INPUT #3,R
1910 PRINT #2,CHR$(R);
1920 GOTO 1890
1930 PRINT #2,CHR$(27);"%";CHR$(1);CHR$(
0);: 'select character RAM
1940 END

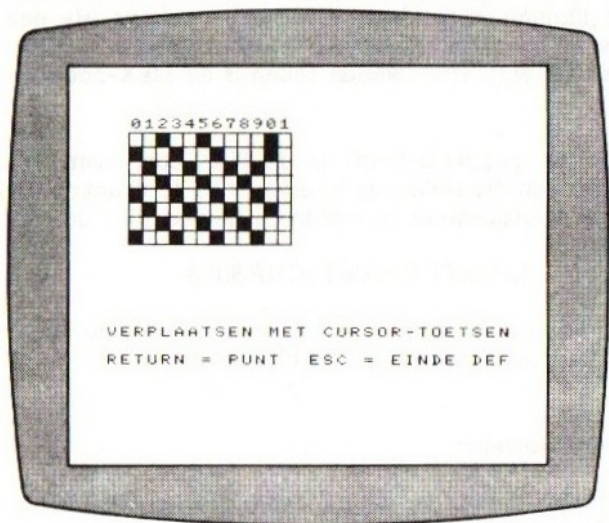
```

Het programma biedt u verschillende mogelijkheden, die u

kunt kiezen van een menu. U kunt de gedefinieerde kolom-codes op het beeldscherm afdrukken, of op de printer. Ook kunt u de kolom-codes naar een file op schijf schrijven. Later kunt u dan met de vierde mogelijkheid een file met kolom-codes van schijf lezen en naar de printer sturen.

Heeft u geen schijf-eenheid, dan kunt u de regels 300 en 1860 aanpassen voor gebruik van cassette, door de aanduiding "a:" te wijzigen in "cas:". De file-naam mag dan niet langer dan 6 tekens zijn.

De volgende afbeelding laat zien wat er op het scherm wordt getoond bij uitvoering van dit programma. U ziet de matrix, met daarin een cursor. De cursor wordt gevormd door middel van een sprite, die zo klein is gekozen, dat hij duidelijk zichtbaar is, terwijl hij toch duidelijk laat zien wat er onder de cursor staat. De cursor kan worden verplaatst met behulp van de cursor-control toetsen. Het vakje waarin de cursor staat kan worden gezet of teruggezet met behulp van de RETURN-toets. Heeft u een teken naar uw zin opgebouwd, dan kunt u dit aangeven door de ESC-toets in te drukken.



# Appendix A

## MSX-printer standaard

Printers, die het predicaat "MSX-printer" willen voeren, dienen aan bepaalde eisen te voldoen. Naast de tekenset van de printer worden ook eisen gesteld aan de commando's die de printer moet kunnen uitvoeren. In deze appendix worden alle eisen, waaraan een MSX-printer moet voldoen, opgesomd.

### De tekenset.

De tekenset van de printer moet precies dezelfde zijn als die van de MSX-computer zelf. Dit houdt in, dat alle MSX teken-codes van 0 tot en met 254 door de printer moeten kunnen worden afgedrukt. De codes 0 tot en met 31 worden echter door de printer gezien als commando's. Om die MSX-codes toch te kunnen afdrucken, dienen die tekens als een twee bytes code naar de printer te worden gestuurd. Het eerste teken is CHR\$(1). Het tweede teken is de MSX-code verhoogd met 64.

Het lachende poppetje heeft in de MSX-computer de code 1 (CHR\$(1)). Om dit teken op de printer af te drukken dient dus het volgende statement te worden uitgevoerd op de computer:

```
LPRINT CHR$(1);CHR$(65)
```

De code 65 bestaat dan uit de waarde 64 plus de waarde van de code van het poppetje in de MSX-tekenset (1).

### De commandocodes

MSX-printers dienen de codes, zoals die gelden voor de NEC PC-8023 printer, te volgen. Dit wil niet zeggen, dat een MSX-

printer alle codes moet kennen die de NEC PC-8023 kent. Er is echter wel een minimum gesteld aan het aantal commando's dat de printer moet kunnen uitvoeren, om nog als MSX-printer te worden bestempeld. In de volgende tabel worden alle commandocodes van de NEC PC-8023 printer opgesomd, in volgorde van ASCII-code. Indien van toepassing, wordt aangegeven of een commando tot de minimum commandoset behoort of niet, door het betreffende commando vet te drukken.

Commando	omschrijving
CHR\$(8)	Back Space
CHR\$(9)	Horizontal Tab
<b>CHR\$(10)</b>	<b>Line Feed</b>
CHR\$(11)	Vertical Tab
<b>CHR\$(12)</b>	<b>Form Feed</b>
<b>CHR\$(13)</b>	<b>Carriage Return</b>
CHR\$(14)	Double width
CHR\$(15)	Normal width
CHR\$(27);"!"	Near Letter Quality
CHR\$(27); CHR\$(34)	Verlaat Near Letter Quality
CHR\$(27);"\$"	Katakana
CHR\$(27);"&"	Hiragana
CHR\$(27);"(nnn"	Zet horizontale tabulatiepositie(s)

CHR\$(27);"nnn"	Verwijder aangegeven horizontale Tab(s)
CHR\$(27);"2"	Verwijder alle horizontale tabposities
CHR\$(27);"["	Incremental printing
CHR\$(27);"]"	Logical seeking bidirectional printing
<b>CHR\$(27);"A"</b>	<b>1/6 inch Line Feed</b>
<b>CHR\$(27);"B"</b>	<b>1/8 inch Line Feed</b>
CHR\$(27);"E"	Elite (12 tekens/inch)
CHR\$(27);"Lnnn"	Set Left margin op positie nnn
CHR\$(27);"N"	Normal spacing (10 tekens/inch)
CHR\$(27);"P"	Proportioneel
CHR\$(27);"Q"	Condensed (136 tekens/regel)
<b>CHR\$(27);"Snnnn"</b>	<b>Grafische mode; nnnn kolommen</b>
CHR\$(27);"Tnn"	nn/144 inch Line Feed
CHR\$(27);"X"	Start onderstrepen
CHR\$(27);"Y"	Beeindig onderstrepen
CHR\$(27);"f"	Normale papieropschuif
CHR\$(27);"r"	Terugwaartse papieropschuif
CHR\$(29)	Start Vertical Form control setting

CHR\$(30)                      End Vertical Form control  
setting

CHR\$(31);  
CHR\$(16+n)                    n regels opschuiven  
(n = 1 to/m 15)

CHR\$(31);  
CHR\$(n)                      Vertical Tab channel n  
(n = 2 t/m 6)

## Appendix B

# Vertaaltabellen printer commando's

De in deze appendix gegeven tabellen geven een overzicht van een aantal commando's voor alle printers die ik tijdens het schrijven van dit boek heb gebruikt. Het gaat daarbij om de printers: Seikosha SP1000MX (een MSX-printer), Seikosha SP800, Epson FX80+, OKI-mate 20, Brother CE50BT en Juki 6100.

De tabellen bevatten niet de complete commando-sets van de printers. Ik heb een keuze gemaakt van die commando's die het meest worden gebruikt. Ook is een overweging geweest, dat de opgenomen commando's een zo groot mogelijk toepassingsgebied moesten vormen.

Het nut van deze tabellen ligt daarin, dat programma's, waarin printers worden aangestuurd, met behulp van deze tabellen gemakkelijk kunnen worden geconverteerd naar een programma voor uw eigen printer. Daarom is er ook een lege kolom opgenomen. U kunt daarin de overeenkomstige printercommando's voor uw eigen printer aantekenen, indien u een printer bezit, die niet in de tabel voorkomt. Staat uw printer al in de tabel, dan kunt u wellicht de gegevens van een andere printer te pakken krijgen en in de tabel invullen.



	MSI SP1000MI	Seikosa SP000	Epson FX00+	DKI-mate 20	Brother CES0BT	Juki	
<b>Regelafstand:</b>							
6 regels/inch	CHR\$(27) "A"	CHR\$(27) "2"	CHR\$(27) "2"	-	CHR\$(27) CHR\$(30) CHR\$(9)	CHR\$(27) CHR\$(30) CHR\$(9)	
8 regels/inch	-	CHR\$(27) "0"	CHR\$(27) "0"	CHR\$(27) "0"	-	CHR\$(27) CHR\$(30) CHR\$(7)	
9 regels/inch	CHR\$(27) "B"	-	-	-	-	-	
48/n regels/inch	-	-	-	-	-	CHR\$(27) CHR\$(30) CHR\$(n)	
72/7 regels/inch	-	CHR\$(27) "1"	CHR\$(27) "1"	-	-	-	
72/n regels/inch	-	CHR\$(27) "A" CHR\$(n)	CHR\$(27) "A" CHR\$(n)	CHR\$(27) "A" CHR\$(n)	-	-	
144/n regels/inch	CHR\$(27) "T" CHR\$(n)	-	-	-	-	-	
192/(n-32) reg./inch	-	-	-	-	CHR\$(27) "L" CHR\$(n)	-	
216/n regels/inch	CHR\$(27) "2" CHR\$(n)	CHR\$(27) "3" CHR\$(n)	CHR\$(27) "3" CHR\$(n)	CHR\$(27) "3" CHR\$(n)	-	-	
<b>Teknafstand:</b>							
10 tekens/inch (PICA)	CHR\$(27) "N"	-	-	CHR\$(18)	CHR\$(27) CHR\$(31) CHR\$(13)	CHR\$(27) CHR\$(31) CHR\$(13)	
12 tekens/inch aan (ELITE)	CHR\$(27) "E"	CHR\$(27) "M"	CHR\$(27) "M"	CHR\$(27) ":"	CHR\$(27) CHR\$(31) CHR\$(11)	CHR\$(27) CHR\$(31) CHR\$(11)	

	MSI SP1000MI	Seikosha SPB00	Epson FX00+	OKI-mate 20	Brother CE50BT	Juki
<b>Tekenafstand (vervolg):</b>						
12 tekens/inch uit (ELITE)	-	CHR\$(27) "p"	CHR\$(27) "p"	-	-	-
15 tekens/inch	-	-	-	-	CHR\$(27) CHR\$(31) CHR\$(9)	CHR\$(27) CHR\$(31) CHR\$(9)
17 tekens/inch aan (CONDENSED)	CHR\$(27) "q"	CHR\$(15)	CHR\$(15)	CHR\$(15)	-	-
17 tekens/inch uit (CONDENSED)	-	CHR\$(18)	CHR\$(18)	-	-	-
Proportioneel aan	CHR\$(27) "p"	CHR\$(27) "p" CHR\$(1)	CHR\$(27) "p" CHR\$(1)	-	CHR\$(27) "p"	CHR\$(27) "p"
Proportioneel uit	-	CHR\$(27) "p" CHR\$(0)	CHR\$(27) "p" CHR\$(0)	-	-	CHR\$(27) "q"
<b>Print codes:</b>						
Bold printing aan	CHR\$(27) "CB"	CHR\$(27) "E"	CHR\$(27) "E"	-	-	CHR\$(27) "0"
Bold printing uit	CHR\$(27) "Cb"	CHR\$(27) "F"	CHR\$(27) "F"	-	-	CHR\$(27) "b"
Double strike aan	CHR\$(27) "CD"	CHR\$(27) "G"	CHR\$(27) "G"	-	-	-
Double strike uit	CHR\$(27) "Cd"	CHR\$(27) "H"	CHR\$(27) "H"	-	-	-
Double width aan	CHR\$(14)	CHR\$(14) of CHR\$(27) "y" CHR\$(1)	CHR\$(14) of CHR\$(27) "y" CHR\$(1)	CHR\$(14) of CHR\$(27) "y" CHR\$(1)	-	-

	MSX SP1000MI	Seikoha SP000	Epson FX80+	OKI-mate 20	Brother CE50BT	Juki	
<b>Printmodes (vervolg):</b>							
Double width uit	CHR\$(15)	CHR\$(13) of CHR\$(27) "U" CHR\$(0)	CHR\$(13) of CHR\$(27) "U" CHR\$(0)	CHR\$(13) of CHR\$(27) "U" CHR\$(0)	-	-	
Italic aan	CHR\$(27) "CI"	CHR\$(27) "4"	CHR\$(27) "4"	CHR\$(27) "56"	-	-	
Italic uit	CHR\$(27) "CI"	CHR\$(27) "5"	CHR\$(27) "5"	CHR\$(27) "5H"	-	-	
NLQ mode aan	CHR\$(27) CHR\$(33)	CHR\$(27) "x" CHR\$(1)	-	CHR\$(27) "12"	-	-	
NLQ mode uit	CHR\$(27) CHR\$(34)	CHR\$(27) "x" CHR\$(0)	-	CHR\$(27) "11"	-	-	
Subscript aan	CHR\$(27) "CU"	CHR\$(27) "S" CHR\$(1)	CHR\$(27) "S" CHR\$(1)	CHR\$(27) "S" CHR\$(1)	-	-	
Subscript uit	CHR\$(27) "Cu"	CHR\$(27) "T"	CHR\$(27) "T"	CHR\$(27) "T"	-	-	
Superscript aan	CHR\$(27) "CS"	CHR\$(27) "S" CHR\$(0)	CHR\$(27) "S" CHR\$(0)	CHR\$(27) "S" CHR\$(0)	-	-	
Superscript uit	CHR\$(27) "Cs"	CHR\$(27) "T"	CHR\$(27) "T"	CHR\$(27) "T"	-	-	
Underline aan	CHR\$(27) "X"	CHR\$(27) "_" CHR\$(1)	CHR\$(27) "_" CHR\$(1)	CHR\$(27) "_" CHR\$(1)	CHR\$(27) "E"	CHR\$(27) "E"	
Underline uit	CHR\$(27) "Y"	CHR\$(27) "_" CHR\$(0)	CHR\$(27) "_" CHR\$(0)	CHR\$(27) "_" CHR\$(0)	CHR\$(27) "R"	CHR\$(27) "R"	

	MSI SP1000MI	Seikosha SP800	Epson FX80+	DKI-mate 20	Brother CE50BT	Juki	
<b>Horizontaal tabulieren:</b>							
Naar volgende TAB	CHR\$(9)	CHR\$(9)	CHR\$(9)	CHR\$(9)	CHR\$(9)	CHR\$(9)	
TABs aanbrengen	CHR\$(27) "(* "nnn,nnn"	CHR\$(27) "D" CHR\$(n) CHR\$(0)	CHR\$(27) "D" CHR\$(n) CHR\$(0)	CHR\$(27) "D" CHR\$(n) CHR\$(0)	CHR\$(27) "1"	CHR\$(27) "1" of CHR\$(27) CHR\$(9) CHR\$(n)	
Een TAB verwijderen	CHR\$(27) ")" "nnn,nnn"	-	-	-	CHR\$(27) "8"	CHR\$(27) "8"	
Alle TABs verwijderen	CHR\$(27) "2"	-	-	-	CHR\$(27) "2"	CHR\$(27) "2"	
<b>Verticaal tabulieren:</b>							
Naar volgende TAB	CHR\$(11)	CHR\$(11)	CHR\$(11)	-	-	CHR\$(11)	
Zet verticale TAB	-	-	CHR\$(27) "B" CHR\$(n) CHR\$(0)	-	-	CHR\$(27) "_" of CHR\$(27) CHR\$(11) CHR\$(n)	
<b>Pagina formatering:</b>							
Kantlijn links zetten	CHR\$(27) "L" "nnn"	CHR\$(27) "1" CHR\$(n)	CHR\$(27) "1" CHR\$(n)	-	CHR\$(27) "9"	CHR\$(27) "9"	
Kantlijn rechts zetten	CHR\$(27) "/" "nnn"	CHR\$(27) "Q" CHR\$(n)	CHR\$(27) "Q" CHR\$(n)	-	CHR\$(27) "0"	CHR\$(27) "0"	
Paginalengte in regels	CHR\$(27) "D" "nnn"	CHR\$(27) "C" CHR\$(n)	CHR\$(27) "C" CHR\$(n)	CHR\$(27) "C" CHR\$(n)	-	CHR\$(27) CHR\$(12) CHR\$(n)	
Paginalengte in inches	CHR\$(27) "O1" "nn"	CHR\$(27) "C" CHR\$(0) CHR\$(n)	CHR\$(27) "C" CHR\$(0) CHR\$(n)	CHR\$(27) "C" CHR\$(0) CHR\$(n)	-	-	

	MSX SP1000MI	Seikosha SP800	Epson FX80+	DKI-mate 20	Brother CE50BT	Juki	
<b>Paginaformatering (vervolg):</b>							
Margin a/h begin	-	-	-	-	-	CHR\$(27) "T"	
Margin a/h einde	-	-	-	-	-	CHR\$(27) "L"	
Wissen v/d margins	-	-	-	-	-	CHR\$(27) "C"	
Skip perforatie (regels)	CHR\$(27) "OS" "nn"	CHR\$(27) "N" CHR\$(n)	CHR\$(27) "N" CHR\$(n)	CHR\$(27) "N" CHR\$(1)	-	-	
Clear skip perforatie	CHR\$(27) "OS00"	CHR\$(27) "0"	CHR\$(27) "0"	CHR\$(27) "0"	-	-	
<b>Grafische modes:</b>							
Standard density	CHR\$(27) "6060"	CHR\$(27) "K" CHR\$(n) CHR\$(n)	CHR\$(27) "K" CHR\$(n) CHR\$(n)	CHR\$(27) "K" CHR\$(n) CHR\$(n)	-	-	
Double density	CHR\$(27) "6120"	CHR\$(27) "L" CHR\$(n) CHR\$(n)	CHR\$(27) "L" CHR\$(n) CHR\$(n)	CHR\$(27) "L" CHR\$(n) CHR\$(n)	-	-	
Quadruple density	CHR\$(27) "6240"	CHR\$(27) "Z" CHR\$(n) CHR\$(n)	CHR\$(27) "Z" CHR\$(n) CHR\$(n)	CHR\$(27) "Z" CHR\$(n) CHR\$(n)	-	-	
Grafische mode aan	-	-	-	-	-	CHR\$(27) "3"	
Verplaats naar rechts	-	-	-	-	-	" "	
Verplaats naar links	-	-	-	-	-	CHR\$(8)	
Verplaats omlaag	-	-	-	-	-	CHR\$(10)	
Verplaats omhoog	-	-	-	-	-	CHR\$(27) CHR\$(10)	

	MSX SP1000MX	Seikosha SP800	Epson FX80+	DKI-mate 20	Brother CE50BT	Juki	
<b>Grafische modes (vervolg):</b>							
Grafische mode uit	-	-	-	-	-	CHR\$(27) "4"	
<b>Download Characters:</b>							
Copy ROM naar RAM	-	-	CHR\$(27) ";" CHR\$(0) CHR\$(0) CHR\$(0)	-	-	-	
Definieer tekens  (herdefinieer A-Z) (attribuut + kolommen)	-	-	CHR\$(27) "3" CHR\$(0) "AZ" CHR\$(n) etc.	-	-	-	
Selecteer RAM	-	-	CHR\$(27) "x" CHR\$(1) CHR\$(0)	CHR\$(27) "x2"	-	-	
Selecteer ROM	-	-	CHR\$(27) "x" CHR\$(0) CHR\$(0)	CHR\$(27) "11" of CHR\$(27) "12"	-	-	
Set downline loading	-	-	-	CHR\$(27) CHR\$(1) "1"	-	-	
Laad normaal teken (herdefinieer a) (kolom-waarden)	-	-	-	CHR\$(27) "\$Aa" CHR\$(n) etc.	-	-	
Laad verlaagd teken (herdefinieer g) (kolom waarden)	-	-	-	CHR\$(27) "\$Dg" CHR\$(n) etc.	-	-	

	MSX SP1000MX	Seikosha SP000	Epson FX80+	DKI-mate 20	Brother CES0BT	Juki	
<b>Download characters (vervolg):</b>							
Vis download tekens	-	-	-	CHR\$(27) CHR\$(1) CHR\$(0)	-	-	
<b>Kleurencommando's:</b>							
Zet lint gereed	-	-	-	CHR\$(27) CHR\$(25)	-	-	
Begin nieuwe regel	-	-	-	CHR\$(13)	-	-	
<b>Diverse commando's:</b>							
Back space	CHR\$(8)	CHR\$(8)	CHR\$(8)	-	CHR\$(8)	CHR\$(8)	
Buffer Clear	CHR\$(24)	CHR\$(24)	CHR\$(24)	CHR\$(24)	-	-	
Buzzer	CHR\$(7)	CHR\$(7)	CHR\$(7)	CHR\$(7)	CHR\$(7)	CHR\$(7)	
Carriage Return (CR)	CHR\$(13)	CHR\$(13)	CHR\$(13)	CHR\$(13)	CHR\$(13)	CHR\$(13)	
Form Feed (FF)	CHR\$(12)	CHR\$(12)	CHR\$(12)	CHR\$(12)	-	CHR\$(12)	
MSX-symbolen	CHR\$(1) CHR\$(n)	-	-	-	-	-	
Line Feed (LF)	CHR\$(10)	CHR\$(10)	CHR\$(10)	CHR\$(10)	CHR\$(10)	CHR\$(10)	
Reset printer	CHR\$(27) "g"	CHR\$(27) "g"	CHR\$(27) "g"	-	-	CHR\$(27) CHR\$(13) "p"	

# Alfabetische trefwoordenlijst

+ 61

; 60

14-polige connector 41

36-polige connector 41

101 35

102 32, 35

103 32, 35

104 35

105 31, 36

106 32, 36

107 36

108 36

## A

aansluiten 26

AB 32

ACKNLG 41

afdrukbare code 54

Amphenol connector 41

ASCII code 28

ASCII-file 77, 79

ASCII-tabel 45

attribuut 126

AUTOFEEDXT 62

## B

BA 32

Back Space 49, 67, 149

Baud 29

BEL (buzzer) 49, 67, 149

besturingscodes 52

bi-directional printing 25

bits per seconde 29

brede tekens 65

BS-code 49



Buffer Clear (cancel) 149  
BUSY 40  
Buzzer (Bel) 149

## C

CA 31  
CAN-code 52  
Cancel (buffer clear) 52, 67, 149  
Carriage Return 50, 59, 60  
CB 32  
CCITT 33  
Centronics connector 42  
Centronics interface 9, 26, 39  
Centronics protocol 40  
character editor 131  
character set 10  
CHR\$(1) 75  
CHR\$(7) 67  
CHR\$(8) 67  
CHR\$(11) 67  
CHR\$(12) 67  
CHR\$(14) 65  
CHR\$(15) 65  
CHR\$(18) 65  
CHR\$(20) 65  
CHR\$(24) 67  
CHR\$(27) 68  
Clear To Send 32  
commando code 54  
coördinatenstelsel 97  
copieren ROM naar RAM 124  
CR-code 50  
CTS 32

## D

daisy wheel printer 11, 17  
Data Circuit Equipment 35  
Data Quality 17  
Data Terminal Equipment 35  
DC1 38, 51

DC2 51  
DC3 38, 51  
DC4 50, 51  
DCE 35  
decimale notatie 62  
definieren van tekens 131  
descenders 128  
DIP-switch 66  
diverse commando's 149  
double width 130  
down loading 11  
download characters 123, 148  
drukkoptransport 14  
DTE 35

## E

EBCDIC code 28  
elektrostatistische printer 21  
ELITE 66  
ESC-code 52, 68  
ESC-reeksen 67  
even pariteit 28

## F

FF-code 50  
Form Feed 50, 67  
friction feed 23  
full duplex 38

## G

geen pariteit 29  
gemeenschappelijke aarde 32  
geruisloosheid 19, 22  
grafische codes 100  
grafische mode 56, 91, 122, 133, 147  
grafische voorstelling 19

## H

handshaking 37, 40  
hard copy 92

hardcopy SCREEN 0	93
hardcopy SCREEN 1	93
hardcopy SCREEN 2	94
hardcopy SCREEN 3	98
hexadecimale notatie	62
horizontaal tabuleren	49, 146
horizontaal pitch	97
HT-code	49, 62

## I

IA-5	45
IBM PC tekenset	46
impact printer	11
ink jet printer	19
inktlinten	22
input buffer	52
Internationaal Alfabet	45
ITALIC	89

## K

kettingbaanformulieren	23
kleuren commando's	149
kleurenafdruk	114
kleureninktlint	122
kolom in grafische mode	91

## L

lawaai	12
lettervormen definiëren	123
LF	50, 60
Line Feed	50, 59, 60, 149
linker kantlijn	77, 82
listings afdrukken	76
logical seeking	25
LPRINT USING statement	72
LPRINT-statement	60

## M

margrietwiel printer	12
matrix printer	014

MAXFILES-statement	64
minimum commandoset	139
modem	35
MSX commandocodes	138
MSX printer standaard	72, 138
MSX symbolen	149
MSX tekenset	46, 72, 138

## N

naaldjes	15
Near Letter Quality	12, 17, 90
nieuwe tabs zetten	71
NLQ	12, 90
no parity	29
non impact printers	19

## O

odd parity	29
oneven pariteit	29
OPEN-statement	64

## P

pagina formateren	146
paper low detection	80
papier aan de rol	23
papier op detectie	80
papier van de stapel	23
papieropvoer	14
parallele verbinding	26
parallel interface	39
pariteitsbit	28
physieke verbinding	41
PICA	66
plusje	61
PRINT # statement	63
print buffer	52
print modes	83, 144
printkop	15, 18
proportioneel	13, 66, 130
Protective ground	35

protocol 31  
puntkomma 60

## R

Ready To Send 31  
ready/busy protocol 31  
rechter kantlijn 77, 82  
regelafstand 101, 143  
regelafstand instellen 106, 113, 122  
regellengte 13  
regels per inch 18  
reset printer 79, 149  
RS232C 33  
RS232C connector 34  
RS232C interface 9, 26  
RTS 31

## S

schuin afdrukken 90  
SCREEN-statement 73, 75  
seriele verbinding 26  
sheet feeder 24  
Shift In 51  
Shift Out 50, 68  
SI 51  
Signal ground 35  
smalle tekens 65  
snelheid 13  
SO 50, 52, 68  
SOH-code 75  
start bit 28  
stop bit 29  
STROBE 40  
subscript 85, 89  
superscript 85, 89

## T

tabs toevoegen 71  
tabs verwijderen 71  
tabulatiestoppen 62, 70

teken ROM 123  
tekenafstand 143  
tekengenerator 55  
tekenmatrix 126  
tekens per inch 18  
tekenset 10  
thermische printer 21  
tractor feed 23  
transmissie code 28  
Transmitted Data 32  
TxD 32

## V

V.24 33  
Variabele waarden 78  
verbinding 26  
verlaagde tekens 128  
verticaal tabuleren 50, 146  
vertical pitch 97  
Vertical Tab 67  
vet afdrukken 85, 89  
VT-code 50

## W

wrap around 80, 83

## X

Xon/Xoff protocol 036

## Nederlandstalige MSX handboeken

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

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

ISBN 90 6398 1007

### **MSX ZAKBOEKJE** door Wessel Akkermans

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

ISBN 90 6398 888 5

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

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

ISBN 90 6398 407 3

### **MSX PRAKTIJKPROGRAMMA'S** door Wessel Akkermans

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

ISBN 90 6398 437 5

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

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

ISBN 90 6398 254 2

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

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

ISBN 90 6398 674 2

## **MSX LEERBOEKEN**

door Wessel Akkermans en Piet den Heijer

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

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

MSX Basic leerboek deel 1 - ISBN 90 6398 649 1

Opdrachten bij deel 1 - ISBN 90 6398 596 7

MSX Basic leerboek deel 2 - ISBN 90 6398 769 2

Opdrachten bij deel 2 - ISBN 90 6398 556 8

MSX DOS leerboek deel 3 - ISBN 90 6398 519 3

Opdrachten bij deel 3 - ISBN 90 6398 516 9

## **MSX Verder uitgediept** door H. Klopper

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

ISBN 90 6398 447 2

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

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

ISBN 90 6398 735 8



## **MSX TRUKS EN TIPS**

door A.C.J. Groeneveld

Hoe laat ik de computer een cirkel arceren, hoe tover ik mijn computer om in een elektronisch orgeltje, hoe maak ik een mooie intro voor een spelletje. Allemaal vraagstukken die zich lastig laten programmeren maar die iedere MSX-er toch graag opgelost wil zien.

Dit boekje staat boordevol truuks en tips, allemaal in gewoon MSX basic geschreven. Bladerend door dit boek komt u tot de ontdekking dat er voornamelijk korte maar uiterst krachtige en bijzonder goed bruikbare routines zijn opgenomen. Dit boekje geeft kort maar krachtig een antwoord op al uw programmeervragen.

deel 1 ISBN 90 6398 900 8

deel 2 ISBN 90 6398 340 9

## **SOFTWARE PLUS IN MSX**

### **INTROTAPE MSX** door A.C.J. Groeneveld

Heeft u nog maar net een MSX computer gekocht en wilt u graag weten wat de computer kan en hoe u hem kunt leren programmeren? Deze cassette introduceert MSX op een uiterst vriendelijke en onderwijzende manier. U krijgt instructies hoe u de computer aan moet sluiten en de tape laden. Daarna volgt een demonstratie van de mogelijkheden in MSX, zoals het tekenen van sprites en het werken met de driestemmige toongenerator. Het geheel wordt afgesloten met twee 'les' gedeeltes. In anderhalf à drie uur weet u wat de MSX computer is, wat hij kan, en heeft u haast ongemerkt al wat regels geprogrammeerd.

ISBN 90 6398 148 1

### **MSX SCRIPT** door Ton Weijters

Een menugestuurde nederlandstalige tekstverwerker. Het programma is geschikt om efficiënt grotere of kleinere teksten te bewerken. Pagina-indeling (regellengte, paginalengte, marge, inspringen, centreren, enz.) wordt door het programma verzorgd. Dit geldt ook voor de paginatelling, toptitel en het eventueel uitvullen van de regels. Ook corrigeren, zoeken, string-substitutie, blokken tekst verplaatsen, kopiëren of verwijderen, onderstrepen en vet zetten, is mogelijk met dit programma.

ISBN 90 6398 189 9

### **MSX DRAWS** door A.C.J. Groeneveld

Een tekenprogramma in MSX basic, waarmee u al binnen 10 minuten uw eerste tekening kunt maken. Draws werkt erg vriendelijk en maakt gebruik van alle grafische mogelijkheden van de MSX computer. U kunt met Draws zowel technisch als creatieve tekeningen maken. Het programma heeft een effectief bereik van ruim 30.000 bij 30.000 puntjes met mogelijkheden als lijnen, cirkels, krommen, inkleuren, vergroten, verkleinen, verschuiven, verdraaien en andere tekeningen invoegen

ISBN 90 6398 754 4





Het aansluiten van printers, of dat nu MSX-printers zijn of één van de vele andere soorten printers, aan een MSX-computer, wil nogal eens problemen geven. Om vervolgens alle mogelijkheden, die de printers bieden, te kunnen benutten, is meer kennis van het programmeren in MSX-Basic nodig.

In dit boek wordt het aansluiten van allerhande printers aan zowel MSX- als MSX2-computers behandeld. Wie zich nog een printer moet aanschaffen, vindt in dit boek voldoende gegevens om een goede keuze te maken. Wie al een printer heeft, leest in dit boek de aansluitmogelijkheden.

Wilt u uw printer meer laten doen dan alleen een listing van een programma laten maken, dan zult u iets meer van de manier, waarop in MSX-Basic een printer dient te worden aangestuurd, moeten weten. Er worden daarom in dit boek, aan de hand van talloze voorbeelden, in MSX-Basic voor zowel de MSX- als de MSX2-computers, programmeervoorbeelden gegeven. Met behulp van de gegeven theorie en de praktijkvoorbeelden, kunt u de mogelijkheden van uw printer ten volle benutten.

Een korte greep uit de inhoud:

- Soorten printers
- Aansluiten van printers
- Programmeren van printers
- Printen in de grafische modes
- Definiëren van eigen tekens
- Gebruiksklare voorbeeldprogramma's