## 9.3 DATA OP DE MDCR

## 9.3.1 Karakteristieken van de gebruikte recorder

Cassette : Philips Minicassette, tape 3,81 mm breed Opnamekop : enkele spleet, enkel spoor, halve breedte

Sporen : twee, een in elke richting

Opname mode : teken en bit serieel, phase encoded, het minst significante

bit voorop

Dichtheid : 12 - 20 bits/mm

Capaciteit : in een aaneengesloten blok: 64 kbyte per kant in blokken van 1024 byte : 40 kbyte per kant

in blokken van 256 byte : 32 kbyte per kant

Snelheid : 6000 bits/sec.

30 cm/sec. - 50 cm/sec.

Doorlooptijd : 95 seconden nominaal zonder stops voor 35 meter tape

Starttijd : in dezelfde richting: binnen 100 msec.

in tegenovergestelde richting: binnen 150 msec.

Stoptijd : 30 - 100 msec.

Startafstand : in dezelfde richting: 15 - 50 mm

in tegenovergestelde richting: 30 - 65 mm

Startgat : 1 sec. voor het eerste record

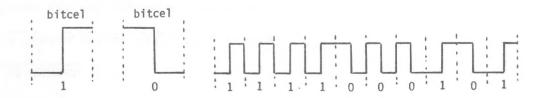
Tussen records: 450 msec voor alle volgende records

End of data gat: 1,8 sec na het laatste record of na het startgat

#### 9.3.2 Phase encoding

Phase encoding is een veel gebruikte manier om digitale informatie op een tape te schrijven. Professionele computertapes worden meestal zo beschreven. De tijd wordt hiervoor ingedeeld in gelijke stukjes, z.g. bitcellen. In elke bitcel wordt een bit geschreven. Op de MDCR van de P2000 komen de bitcellen met een snelheid van 6000 cellen per seconde.

Bij phase encoding wordt een 1 op de tape geschreven met een wisseling in magnetisatie van de tape in een bepaalde richting, bij voorbeeld van zuid naar noord, in het midden van de bitcel. Een 0 wordt geschreven met een wisseling in de tegenovergestelde richting, dus van noord naar zuid, ook in het midden van de bitcel. Wanneer er twee dezelfde bits achter elkaar staan, moet er op de scheiding tussen de twee bitcellen dus een extra magnetisatie-wisseling in de tegenovergestelde richting worden tussengevoegd.

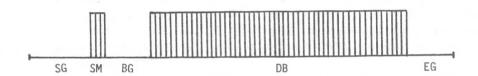


## 9.3 DATA OP DE MDCR

### 9.3.2 Phase encoding (vervolg)

Zoals is te zien, komt er op de cassette een toon van 6 kHz wanneer er een stel nullen achter elkaar wordt geschreven, maar ook als er een aantal enen achter elkaar staat. Phase encoding heeft het voordeel dat er uit de datastroom gemakkelijk een kloksignaal is af te leiden en dat er geen echte lage frequenties in voorkomen. Een klein nadeel is dat de polariteit van het signaal essentieel is. Als het signaal wordt geïnverteerd, dan veranderen alle nullen in enen en omgekeerd. Daarom is phase encoding ongeschikt voor gebruik op normale cassetterecorders. Daarbij is namelijk niets vastgelegd omtrent de polariteit van het weergegeven signaal.

### 9.3.3 Hoe een blok in elkaar zit



SG = Startgat gewiste tape

SM = Startmerk

BG = Blokgat gewiste tape

DB = Datablok

EG = Eindgat gewiste tape

#### 9.3.3.1 Startmerk

Het startmerk bestaat uit 4 bytes:

Sync byte : &H AA
Eerste checksom byte : &H 00
Tweede checksom byte : &H 00
Eindsync byte : &H AA

#### 9.3.3.2 Datablok

Het datablok bestaat uit totaal 1060 bytes, en wel als volgt:

Sync : &H AA

Header : 32 bytes, zie hierna

Data bytes: 1024 bytes, transparante data

Checksom : 2 bytes volgens het polynoom X16 + X15 + X2 +1

Eindsync : &H AA

#### 9.3 DATA OP DE MDCR

#### 9.3.4 Header

De 32 bytes in de header hebben de volgende betekenis:

Byte nr.	Betekenis	Codering
0 en 1 2 en 3 4 en 5 6 t/m 13 14 t/m 16 17 18 19 en 20 21 en 22 23 t/m 30	Beginadres Aantal bytes in de file Aantal geldige databytes Naam, eerste 8 tekens Uitbreiding (BAS, INT, enz) Soort (B, F, V, W, enz) Taalgroep (D, U, N, enz) Machinetaal startadres Machinetaal laad-beginadres Naam, volgende 8 tekens Bloknummer	Binair Binair Binair ASCII ASCII ASCII Binair Binair Binair Binair ASCII Binair, aflopend tot 01
Soort:	<pre>B = Basic W = Wordprocessor V = Viewdata en BIS-editor F = Familiegeheugen-bestanden.</pre>	

Deze letter verschijnt op het scherm wanneer men in BASIC op 'ZOEK' drukt.

Taalgroep:

D = Duits en Oostenrijks

N = Nederlands

U = Engels (en Nederlands)

S = Zweeds

Het is mogelijk dat niet alle bytes in een file geldig zijn. Dit mechanisme is nodig om het mogelijk te maken van te voren ruimte te reserveren voor een groter wordende file. De tekstverwerker reserveert op deze manier 5 kb voor elke pagina. Basic gebruikt het alleen wanneer een kleiner programma over een oud, groter programma heen geschreven wordt. Wat er dan op de cassette geschreven wordt, blijft even groot, de bytes aan het einde worden overschreven met nullen en deze zijn niet geldig. Aan de bytes nr. 2 en 3 is dus te zien hoe groot de werkelijke file ooit geweest is.



### 9.4 CASSETTE-FOUTEN

### 9.4.1 Cassette-fouten bij Basic-NL

Af en toe treden er fouten op bij het gebruik van de minicassette. Deze worden meestal veroorzaakt door een vervuilde, versleten of beschadigde band in de cassette. Maar er kunnen ander oorzaken zijn. Hier is een lijst van alle fouten, die kunnen optreden. Gegeven wordt het errornummer, de bijbehorende letter en de melding van Basic-NL, als die er is.

- 65, Cassettefout A, Geen cassette Er is geen cassette in de recorder of het schakelaartje dat naar de cassette voelt, is ontregeld.
- 66, Cassettefout B
  Begin van de band. Dit wijst op een veel te stroef lopende cassette. De
  P2000 heeft het begin van een programma gevonden en wil weer naar het
  begin van het programma spoelen. Dat gaat echter zo stroef, dat het lijkt
  alsof de cassette al aan het begin staat.
- 67, Cassettefout C, Leesfout
  Checksom fout in datablok. Elk blok op de cassette wordt gecontroleerd met
  een checksom. Als deze fout gelezen wordt, komt dat meestal door een
  beschadigde band. Soms is een stukje per ongeluk gewist. Zet nooit de
  P2000 uit als de cassette (nog) loopt!
- 68, Cassettefout D
  Checksom fout van het startmerk. Elk datablok wordt voorafgegaan door een startmerk. De checksom hiervan kan door dezelfde oorzaken als die bij een leesfout verkeerd uitvallen.
- 69, Cassettefout E
  Einde van de cassette. Bij het inlezen betekent dit vrijwel altijd dat er
  data verloren zijn gegaan. Bij schrijven kan het voorkomen dat toch alles
  is weggeschreven. Dit is op de meest eenvoudige wijze te controleren door
  het programma of de data nogmaals weg te schrijven op dezelfde plaats.
  Gaat het de tweede keer zonder problemen, dan is alles in orde.
- 70, Cassettefout F Cassette is vol (Full). Het programma is in zijn geheel weggeschreven.
- 71, Cassettefout G, Geen stopje Er zit geen schrijfstopje in de cassette terwijl er iets op geschreven moet worden.

## 9. CASSETTERECORDER

### 9.4 CASSETTE-FOUTEN

## 9.4.1 Cassette-fouten bij Basic-NL (vervolg)

- 73, Cassettefout I
  Terugspoeltijd verstreken. De minicassette behoort een spoeltijd van
  ongeveer 90 seconden te hebben. Wanneer een cassette langer spoelt dan
  100 seconden treedt de fout op. Oorzaken kunnen zijn een gebroken
  band of een cassette met een te lange tape erin.
- 74, Cassettefout J
  Te kort datablok gelezen, maar de checksom klopt wel. Deze fout treedt
  zeer zelden op. Het resultaat is hetzelfde als dat bij een leesfout, data
  is niet te gebruiken.
- 75, Cassettefout K
  Verkeerde functiecode. De machinetaalroutine voor de cassette wordt
  bestuurd door een getal mee te geven. Dit bepaalt wat er moet gebeuren.
  Als er een getal wordt gebruikt, dat niets betekent, dan komt deze
  foutmelding terug van de machinetaalroutine. Omdat de interpreter nooit
  een verkeerde functiecode meegeeft komt deze melding in BASIC niet voor.
- 76, Cassettefout L
  Te kort datablok gelezen, checksom ook fout. Meestal is er in dit geval
  een klein stukje van de cassette gewist. Het resultaat is gelijk aan data
  van een leesfout. Schakel de P2000 nooit uit als de recorder nog loopt!
- 78, Cassettefout M, Niet gevonden Geen startmerk gevonden binnen een tijd van 1,5 sec. Dit geeft aan dat de cassette (verder) leeg is.
- 79, Cassettefout N
  Geen datablok gevonden. Na elk startmerk moet een datablok volgen. Dit
  ontbreekt blijkbaar geheel omdat er een stuk tape gewist is of de
  cassette tijdens het opnemen is uitgenomen. Deze fout komt niet vaak voor.

### 9.4.2 Cassette-fouten bij Basic-UK

Bij het gebruik van de oude Basic-UK komen ook nog voor:

Cassette error 0
Te weinig geheugen beschikbaar om het gevonden programma of array in op te slaan. Bij BASIC NL is deze melding vervangen door Out of memory = error 7.

Cassette error W Verkeerde informatie op de cassette. Wat er op de cassette staat is geen BASIC. Dit wordt herkend aan de B in de header. BASIC NL loopt hier over heen en kijkt of er verderop nog iets bruikbaars staat.

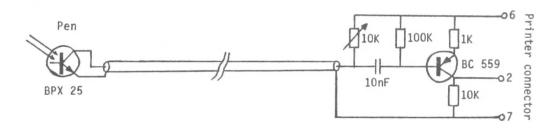


### 13.8.1 Inleiding

De lichtpen is een zeer eenvoudig hulpmiddel, waarmee het kiezen in een programma veel gebruikersvriendelijker kan worden gemaakt. Ook de constructie is eenvoudig. Een lichtgevoelige fototransistor wordt gemonteerd in een geschikt buisje, bij voorbeeld een lege balpenhouder. De lichtpen wordt via een eenvoudige schakeling aangesloten op de printerconnector. Wanneer de pen op het scherm van de monitor of TV wordt gehouden, op een plaats waar een oplichtend teken staat (een wit blokje), dan kan de P2000 dit via de printerconnector signaleren. Hierbij is een klein stukje machinetaal nodig.

Het maken van een keuze uit een menu met behulp van de lichtpen is aanzienlijk veel prettiger dan het kiezen met het toetsenbord. Het sluit meer aan bij de menselijke manier van denken: "Dat daar wil ik hebben".

### 13.8.2 Schakelschema



Door de rasterscan van een monitor is het zo dat elke plaats van het scherm 50 keer per seconde wordt geraakt door de electronenstraal. Het licht is er dus niet continu, maar komt 50 keer per seconde als impuls. De impuls wordt versterkt door de BC 559 en toegevoerd aan pen 2 van de printerconnector. Omdat niet ieder monitorscherm evenveel licht afgeeft, is de gevoeligheid met een regelweerstand instelbaar gemaakt.

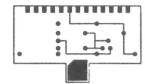
#### 13.8.3 Printje

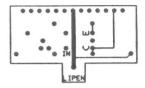
Het kleine aantal onderdelen maakt het mogelijk de gehele schakeling onder te brengen in de kap van de toe te passen RS232-steker, dit buiten de lichtpen. Het rechthoekige printje is bestemd voor inbouw in de kap van een Philips-connector, het trapeziumvormige printje behoort bij een connectorkap van het merk AMP.



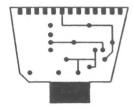
# 13.8.3 Printje (vervolg)

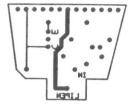
Philips |





AMP





Onderdelenzijde

Sporenzijde

### 13.8.4 Montage

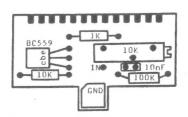
Het monteren van het printje gaat vrijwel net zo als bij de cassetterecorder interface. Raadpleeg hoofdstuk 13.6.3 daarvoor.

### Onderdelen

- 1 Weerstamd
- 1 Weerstand
- 1 Weerstand
- 1 Condensator, keramisch
- 1 Multiturn instelweerstand
- 1 LF PNP transistor met hoge versterking
- 1 Fototransistor met lensje
- 100 kOhm, 0,25 Watt (bruin-zwart-geel)
- 10 kOhm, 0,25 Watt (bruin-zwart-oranje)
- 1 kOhm, 0,25 Watt (bruin-zwart-rood)
- 10 nF
- 10 kOhm

BC 109, BC 559 o.i.d.

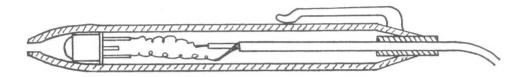
BPX 25





# 13.8.4 Montage (vervolg)

Gebruik dun en soepel afgeschermd snoer van voldoende lengte voor de lichtpen. In de pen wordt alleen de fototransistor BPX 25 ondergebracht. Gebruik voor de pen een oude balpenhouder van niet-doorzichtig plastic. Het gaatje voorin, waar de fototransistor "doorheen kijkt", moet meestal iets groter worden gemaakt, tot ongeveer 3 mm. Dit gat bepaalt grotendeels de kijkhoek.



## 13.8.5 Hulproutine

Er is een klein stukje machinetaal nodig om de lichtpen te kunnen gebruiken.

0.000	01 00	0.2		ID BC 030D	Marrimala kijiktiid - 1/50 aaa
	01 8D	03		LD BC,038D	Maximale kijktijd = $1/50$ sec.
0003	AF		input	XOR A	Anti-streepjes
0004	DB 20			IN 20	Kijk naar lichtpen
0006	2F			CPL	Inverteer alle bits
0007	E6 01			AND 01	Isoleer bit 0
0009	20 05			JR NZ klaar	Licht gezien, dan al klaar
000B	OB			DEC BC	Laag kijktijd-teller af
000C	78			LD A,B	
000D	B1			OR C	Tijd al om?
000E	20 F3			JR NZ input	Nee, nog eens
0010	77		klaar	LD (HL),A	Bewaar een 0 of een 1
0011	3E 02			LD A,02	A=2 voor een integer
0013	C9			RET	Return

Dit stukje machinetaal van 20 bytes kijkt gedurende maximaal 1/50 seconde naar de lichtpen. Als er geen licht werd gezien in deze tijd dan komt hij terug met een 0, was er wel licht, dan geeft hij een 1.

De routine is relocatable en kan ongewijzigd op elk adres worden geplaatst. In een Basic-programma kan hij als volgt worden ingevoegd:

- 10 RESTORE 30 : A=&H6030
- 20 FOR I=0 TO 18 : READ A\$:POKE A+I,VAL("&H"+A\$):NEXT
- 30 DATA 01,8D,03,AF,DB,20,2F,E6,01,20,05,0B,78,B1,20,F3,77,3E,02,C9
- 40 DEF USR=A

Op regel 10 wordt het beginadres van de routine gedefinieerd. Er is gebruik gemaakt van een stukje geheugen dat ook door de cassetterecorder wordt gebruikt. Staat hier wat anders, dan komt de routine op een andere plaats.



#### 13.8.6 Testen en afregelen

Voeg de volgende paar regels toe aan het programma van 13.8.5

100 PRINT CHR\$(12) CHR\$(2) CHR\$(4) CHR\$(12) CHR\$(18) CHR\$(127)

110 A=USR(0)

120 PRINT CHR\$(4) CHR\$(12) CHR\$(20) A

130 IF PEEK("")=0 THEN 110 ELSE A=INP(""): END

Merk op hoe de routine in regel 110 wordt aangeroepen en hoe A de waarde O of 1 krijgt wanneer er geen of wel licht wordt gezien. Run het totale programma. Midden op het scherm verschijnt een blokje en daar achter het cijfer O. Als het goed is, verandert de O in een 1 wanneer de lichtpen op het blokje wordt gehouden. Meestal zal dat niet meteen zo zijn. Handel dan als volgt:

- 1. Regel de monitor zo af dat de achtergrond net geen licht geeft en de tekst niet "opblaast", maar wel goed helder is.
- 2. Regel de instelweerstand door het gaatje in de connectorkap nu zo af, dat er bedrijfszeker een 1 verschijnt als de pen op het blokje wordt gehouden.

De lichtpen reageert alleen op een monitor- of TV-scherm en niet op lampen of daglicht. Het licht moet duidelijk een pulserend karakter hebben.

#### 13.8.7 Gebruik in programma's

Met deze lichtpen kan er alleen worden geken of er licht is onder de pen, maar niet waar dat licht op het scherm is. Dat hoeft geen bezwaar te zijn. Bij voorbeeld in een keuzemenu kunt u dit als volgt opvangen:

Maak een scherm met keuzemenu, met bij elke keuze een blokje en een cijfer. Wacht dan tot er licht wordt gezien door de lichtpen of tot er een toets wordt ingedrukt. Zet in het eerste geval achtereenvolgens de verschillende blokjes even uit en controleer steeds of de output van de lichtpen meegaat met het uitzetten van het blokje. Zo niet, zet dan het blokje weer aan en probeer het met het volgende blokje. Wanneer de juiste keuze gevonden is, zet dan nog een paar keer het blokje aan en uit en controleer of de licht-output echt meegaat. Zo is een zeer bedrijfszekere lichtpen ontstaan, die relatief erg goedkoop is en gemakkelijk in een programma is in te passen.



### 14.4.9 Koppelen van bestanden Jan Gieles

Bestanden gemaakt m.b.v. MINITEXT kunnen op verschillende manieren aan elkaar gekoppeld worden. We behandelen eerst een zuinige manier wat betreft tekstruimte maar die wat kennis en kunde vraagt van de gebruiker en daarna een veilige manier, die altijd werkt en door iedereen te gebruiken is.

Eerst de zuinige manier. Als een bestand wordt ingeladen wordt het standaard in een gereedliggend array geplaatst, dat vlak daarvoor leeg gemaakt is. Onderstel eens, dat we dat leegmaken verhinderen en dat we een aanwezig bestand d.m.v. ingevoegde blanco regels zover naar onderen verplaatst hebben dat er een ander bestand boven past (elke regel is 80 bytes). Een nieuw in te laden bestand zal dan de eerste (lege) regels van het oude overschrijven en we hebben de twee bestanden aan elkaar gekoppeld in de machine ter beschikking.

De werkwijze is als volgt:

Het wissen van het aanwezige bestand gebeurt in regel 940 door de instructies: J=FNT:I=2\*X:GOSUB37:PRINTL\$;

(Sub 37 plaatst I spaties vanaf adres J en zet de cursor aan. FNT is het begin van het array en X het aantal elementen ervan.)
Het is voldoende om deze instructies te verwijderen. Dit levert bovendien een dikke 20 bytes extra ruimte op.

U loopt dan wel tegen enkele problemen op. Elk bestand wordt n.l. voor het wegschrijven voorzien van een staartje van 8 bytes, dat na het inladen wordt uitgelezen en verwijderd. Deze procedure loopt in het bovenstaande geval mis, maar de gevolgen zijn niet ernstig en te overzien. Het zijn de volgende:

1. Als dat nodig is dient U zelf een aanwezig bestand te wissen via STOP en START. Het programma doet dat niet meer.

2. Het staartje van het nieuw ingeladen bestand wordt niet gewist. Overschrijf dus na het inladen de laatste 8 bytes daarvan met spaties.

3. Inplaats daarvan worden de laatste 8 bytes van het onderste, reeds in de machine aanwezige programma wel gewist. Zet dus van tevoren op minstens 8 posities na dat bestand de een of andere letter neer.

4. De linker marge en de regelbreedte worden fout ingelezen. Pas deze dus direct na het inladen weer even aan.

Aangezien het koppelen niet alle dagen zal voorkomen, is het heel goed te doen om deze aanpassing niet te CSAVE'n, maar per keer dat het nodig is uit te voeren. Punt 1 hierboven komt dan te vervallen.



### 14.4.9 Koppelen van bestanden (vervolg)

Voor wie het koppelen regelmatig gebruikt volgt hieronder nog een methode, die zich o.a. van de eerste onderscheidt, doordat een tweede bestand niet voor, maar achter een reeds aanwezig bestand wordt geplaatst. Hiertoe moet het hoofdprogramma worden gewijzigd volgens bijgaande regels. Het kost ca 160 bytes of twee tekstregels. Geef eerst CLOAD"M", wijzig dan en schrijf tenslotte weer weg met CSAVE"MNT Nx.x + kopp."

850 P(4)=1:I=FNT+X*2:J=T*40-X+3:POKEI,2:
POKEI+1,82:POKEI+2,0:POKEI+3,(2*J+3)MODS
:POKEI+4, (2*J+3) ÷S:POKEI+5, 1:POKEI+6, JMO
DS:POKEI+7, J÷S:I=X:GOTO880

In 850 moeten 3 poke's erbij. S = 256, T is de vroegere NT.

900 Y=0:GOSUB90:IFXTHENX=X+4:PRINT" Best and omvat"2\*X"bytes":PRINT:PRINT" Daar o verheen ? (J/N) "CHR\$(1); ELSE910 904 A=INP("")AND95:IFA=78THENIFX+40>T\*40 THENPRINTF\$;:GOTO904ELSEY=1ELSEIFA<>74TH EN904 906 PRINTCHR\$(A)

900 wordt helemaal anders.

940 IFYTHENX=40\*((X-1)÷40)+36:GOSUB850:C

904 en 906 komen erbij.

LOAD\*ROCHR\$(A):GOSUB870ELSEJ=FNT:I=2\*X:G OSUB37:PRINTL\$;:P(3)=0:CLOAD\*TaCHR\$(A)

940 krijgt een stuk ervoor. L\$ = CHR\$(2)

65523 REM Versie Nx.x+Kopp dd \*\*-\*\*-\*\*

Documentatie

Voor het laden wordt gevraagd of het nieuwe bestand over het oude heen moet komen of er achteraan. Na het laden zijn de linker marge en de regelbreedte van het laatst ingeladen bestand in de instelling opgenomen.

Als beide methoden tegelijk worden gebruikt, kan een extra bestand naar believen voor of achter een aanwezig bestand worden gekoppeld.



### 14.4.10 Controle op cassettefouten Jan Gieles

Als een bestand op cassette is weggeschreven, is het alleen maar veilig op eventuele cassettefouten te controleren door het op een andere P2000 te bekijken, b.v. via MINITEXT of via cassettehulp.

De eigen machine is nl. nog bezet en als we uit het programma springen, is het bestand verloren. De ZOEK-toets geeft ook geen uitkomst, want die leest alleen het eerste blok van elke file en slaat de rest over. In de oude UK-BASIC werd bij de inhoudsopgave wel alles gelezen, maar er kwam ook geen foutmelding als er iets mis ging.

Onderstaande aanpassing maakt het mogelijk om binnen het programma een controle op cassettefouten te laten uitvoeren. Het menu van de cassette is daartoe uitgebreid met de mogelijkheid om via de bliksemtoets een cassette op mogelijke fouten te laten doorzoeken.

Deze aanpassing vraagt wijzigingen in zowel het voor- als het hoofdprogramma. In het voorlooppprogramma "printer XXX" wordt een klein stukje machinetaal weggezet. Het regelnummer is onbelangrijk, als het maar groter dan 30 en kleiner dan 100 is.

#### 40 A=&H6350:A\$="7EDF77C9":GOSUB4080:DEF USR=A

In het hoofdprogramma moeten twee dingen gebeuren. In de eerste plaats moet het menu van de cassette als volgt worden aangepast:

820 PRINTFNP\$(2,5)CHR\$(22)TAB(5)"Plaats de juiste c"S\$".":PRINT:PRINTTAB(5)"Inho udsopgave"TAB(30)C\$"ZOEK":PRINT:PRINTTAB(5)"C"S\$" wissen"TAB(30)K\$
830 PRINT:PRINTTAB(5)D\$"inlezen"TAB(30)C\$"INL":PRINT:PRINTTAB(5)D\$"wegschrijven"TAB(30)C\$"OPN":PRINT:PRINTTAB(5)"Controle"TAB(30)C\$"(-/-)"
835 POKEW+5,13:POKEW+6,10:PRINTW\$;
840 GOSUB72:IFI=139THEN9OOELSEIFI=136THEN950ELSEIFI=131THENI=USR5(0)ELSEIFI=123THENI=USR6(0)ELSEIFI=130THENGOSUB3000ELSE



14.4.10 Controle op cassettefouten (vervolg)

Verder moet onderstaande subroutine vanaf 3000 in zijn geheel worden opgenomen:

3000 PRINTC\$"Controle";:I=0:A=USR(0):A=USR(1):IFATHEN3050
3020 POKEF+35,0:POKEF+36,0:POKEF+37,1:POKEF+38,0:POKEF+66,1:A=USR(6):IFATHEN3050
3030 A=PEEK(F+41):IFA<>ITHENPRINT:PRINTY
\$;
3040 I=A:PRINTCHR\$(I)C\$"Ok"STRING\$(3,8);
:GOTO3020
3050 IFA=77THENPRINT:RETURNELSEPRINTM\$"C
"S\$"fout"A"("CHR\$(A)")":RETURN

Deze aanpassing kost ca 240 bytes of 3 tekstregels aan ruimte. Bewaar de LIST-ings, pas de staart aan en CSAVE het programma onder "MINITEXT+CtrNx.x" of iets dergelijks ten behoeve van uw eigen administratie.

Met dank aan Hans Pennings.

14.4.11 Regelinvoer repeteren Jan Gieles

Onderstaande aanpassing in het hoofdprogramma "M" maakt, dat DEF + op analoge wijze reageert als DEF - op het vasthouden van de laatste toets:

220 GOSUB91:IFX<TTHENGOSUB330:P(1)=P(0): P(0)=Z:P(2)=P:I=FNP:G=0:GOSUB82:IFPEEK(F)=42THEN220ELSE186ELSE65



#### 14.4.12 Cassettelabels voor audiocassettes Jan Gieles

Hieronder staat een bestand in MINITEXT, dat als het uitgeprint wordt een papiertje oplevert met de inhoudsopgave van een normale audiocassette, dat bovendien precies in het doosje daarvan past. Door de rug van het doosje heen is dan het nummer en de naam van de cassette en door de platte kant heen de inhoud ervan te lezen. Een tijdbesparende manier dus, om Uw audiocassettes te updaten en van een nieuw jasje te voorzien. De breedte in het bestand hangt af van het aantal tekens per inch

waarop de printer is ingesteld (CPI). Het meest rechtse teken valt:

bij CPI: 10 onder nr: 41 (zie voorbeeld) 12 49 15 61 17 69

Hoe groter CPI is, hoe meer tekst U kwijt kunt maar hoe slechter deze te lezen zal zijn.

In de getoonde bestanden zijn de hekjes in werkelijkheid blokjes. (CHR\$(127), op toetsenbord boven het hek)

De bedoeling is n.l., om iets te laten onderstrepen, wat zelf niet uitgeprint mag worden. Maar juist omdat die blokjes niet gezien kunnen worden, zijn ze hier even vervangen door hekjes. De onderste rij blokjes kan voor zover nodig overschreven worden met tellerstand en titel van het zevende nummer op kant B. Laat daarbij in elk geval het eerste en het laatste blokje staan, dan komt de streep netjes op zijn plaats. Het bestand is bedoeld voor een matrixprinter, omdat deze categorie gewoonlijk bij een reeks onderstreeptekens een stippellijn produceert en alleen bij echt onderstrepen een doorlopende lijn. Bij een letterwielprinter kan met voordeel een iets gewijzigd bestand worden gebruikt, zoals dat waarvan de eerste regels daaronder zijn afgedrukt. De horizontale streepjes zijn onderstreeptekens en geen mintekens. De rest is gelijk aan het eerste bestand.

Onderaan staat een voorbeeld van zo'n label, zoals dat na printen eruit behoort te zien. Er gaan 3 van zulke labels op een A4-tje en als U dat dwars gebruikt zelfs 4. Ik gebruik zelf maar één bestandje en type telkens de tekst over de vorige heen, maar U kunt ze desgewenst natuurlijk ook wegschrijven.



## 14.4.12 Cassettelabels voor audiocassettes (le vervolg)

Bij gebruik van een matrixprinter dient het volgende te worden ingegeven:

```
3
                                  4
                                          5
                                                 6
    $OS
1
2
                                   ļ
                   weg
3
      -Nr.- Titel of naam v.d. cassette
4
    $OS ( # = blokje )
5
6
    7
     000 Eerste nummer kant A
8
     nnn Tweede nummer kant A
9
         enz.
10
11
12
13
14
    ______
15
     000 Eerste nummer kant B
16
     nnn Tweede nummer kant B
17
         enz.
18
19
20
21
       ( # = blokje )
    $OS
22
    23
        Hierboven laatste nummer kant B
```

Voor een letterwielprinter worden de eerste zes regels aldus:

	1	2	3	4	5	6
\$EM-nn	(eventu	ele extr	a marge)			
1		weg		1		
					knip	
-Nr	Titel o	of naam v	.d. cassett	e		
					vouw	
enz	z. zie bo	oven				



14.4.12 Cassettelabels voor audiocassettes (2e vervolg)

Het produkt dat uit de machine komt, hoort er dan zo uit te zien:

! weg	!	knip
-Nr Titel of naam v.d. cassette		vouw
000 Eerste nummer kant A nnn Tweede nummer kant A enz.		
000 Eerste nummer kant B nnn Tweede nummer kant B enz.	==	

Hierboven laatste nummer kant B



### 14.6 SNELZOEKBOEK

### 14.6.2 Hernummeraar voor snelzoekboek Jan Gieles

Voor bestandenprogramma's, waarin de gegevens in REM- of DATA-regels zijn opgenomen is een eenvoudige, ingebouwde hernummeraar vaak erg plezierig.

Onderstaande hernummeraar laat de regels op hun plaats staan, tast het programma niet aan en past zichzelf aan, als de gekozen afstand tussen de regels te groot blijkt te zijn. Hij kan zo in alle ZOEK-boeken worden opgenomen. De aanroep gebeurt met "RUN 10000" in de directe mode. Voor andere bestandsprogramma's kunnen kleine aanpassingen nodig zijn, die m.b.v. de onderstaande LISTing snel duidelijk zullen zijn.

4 2 2 2 2 2	gins, pinne g B B	
10000	REM Hernummer	
10010	DEFINT B-Z:B=0:S=10:E=7999	B=Begin, S=Stap
10020	V=S:A=PEEK(&H625C)+ 256* PEEK	E=Eind, A=Adres
	(%H625D)	V=Verschuiving
10030	N=PEEK(A+2)+ 256* PEEK(A+3):IF	N=Nummer regel
	N<=B THEN 10050 ELSE IF N>E THEN	
	PRINT: PRINT"Klaar": CLEAR: END ELSE	
	IF B+V>E THEN S=S/2:PRINT"-";:	B+V=Nieuw nr.
	G0T010020	
10040	POKE A+2, (B+V) MOD256: POKE A+3,	
	(B+V) =256: V=V+S: PRINT"*";	

De te behandelen regelnummers mogen niet boven de 32000 liggen.

10050 A=PEEK(A)+ 256\* PEEK(A+1):GOTO

10030

A=Nieuw adres