

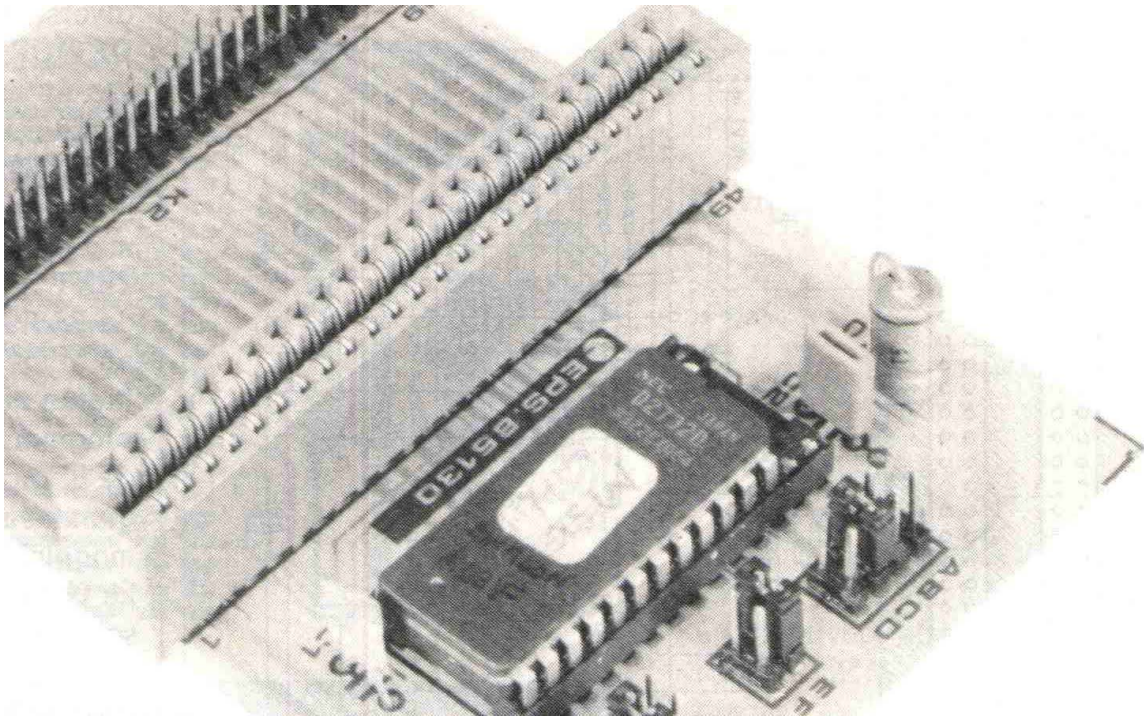
# MSX UITBREIDINGEN

## DEEL II

### MSX Cartridge print

*Scanned by Hans Oranje, ocr'ed and converted to PDF by HansO, 2001*

**Elektuur april 1986, Elektuur MSX special 1986**



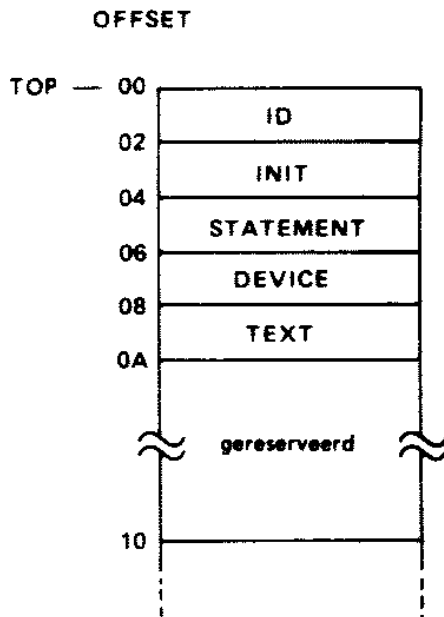
Mits slim ontworpen kan een zelfgemaakte MSX-cartridge-print veel meer functies vervullen dan je zo op het eerste gezicht zou zeggen. Ons tweede verhaal over MSX en de buitenwereld is dan ook helemaal aan de mogelijkheden van een dergelijke print gewijd.

In deel I over MSX en de buitenwereld (zie het februari-nummer van dit jaar) lieten we al een beetje zien dat je zelfs met een fabrieks-cartridge meer kunt doen dan alleen maar spelletjes eruit peuren. Maar zelfs wij waren een beetje verrast van wat je allemaal kunt realiseren wanneer je een speciale (cartridge-)print in het cartridgeslot of de I/O-expansionport prikt. Lees maar eens mee:

1. Probleemloos aansluiten van allerlei hardware-uitbreidingen zoals bijvoorbeeld de universele I/O-bus.
2. De I/O-expansionport van de Spectravideo gebruiken als cartridgeslot.
3. Door het monteren van een kontrakonnektor kan de print worden gebruikt als 90°-verloopstuk.
4. De print kan dienen als koppelstuk voor een Yamaha synthesizer module (o.a. verkrijgbaar in België).
5. Uiteraard kan de print ook als standaard MSX-cartridge aangesloten worden, maar dan met naar keuze (E)PROM's van verschillend "kaliber" (16.. .. 256 Kbit).
6. De cartridge-print is een uitstekend kopieer-hulpstuk (op deze feature gaan we uiteraard niet nader in...)

Nogal wat, nietwaar? En het enige dat men voor dat alles nodig heeft is de door ons ontworpen (dubbelzijdige) print.

**1**



*Figuur 1. Deze data dienen helemaal aan het begin van de EPROM te staan, zodat de software weet wat er moet worden gedaan.*

## De MSX-cartridge

U hebt hierboven al kunnen lezen voor welke toepassingen onze cartridge-print kan worden ingezet. De print zelf zorgt in feite voor het "mechanische" aanpassings- en uitbreidingswerk maar wat komt er allemaal kijken bij het gebruik als cartridge? Om deze vraag te kunnen beantwoorden, is het niet onverstandig om eens wat dieper in de MSX-materie te duiken.

- Hoe start de machine?
- Wat is het verschil tussen een ROM-cartridge en een RAM-uitbreiding?

- Hoe liggen de geheugengebieden binnen de memory-map?

Laten we eens kijken:

- Allereerst bekijkt de MSX-BASIC de aanwezige hoeveelheid RAM tussen de adressen 8000 en BFFF en activeert dan het grootste aaneensluitende RAM-gebied. Daarna wordt hetzelfde gedaan voor het gebied C000... FFFF en tot slot kijkt de BASIC nog naar een aaneensluitend RAM-blok over de beide geheugengebieden. Nu weet de computer wat RAM is en kan zich vervolgens gaan bezighouden met de al dan niet aanwezige cartridges.

- Er wordt nu gekeken naar de inhoud van de slots in het adresbereik 4000.. BFFF. Elk slot omvat 16 K, welke weer onderverdeeld is in vier pages.

Aan het begin van de pagina kijkt de MSX-BASIC naar een bepaalde identifikatiecode, welke informatie verschaft over de aard van de inhoud (een soort visitekaartje dus). Dit zijn een aantal bytes die altijd voor in de ROM of EPROM worden geplaatst.

In figuur 1 zien we de samenstelling van de code.

**ID** (identifikatie): dit zijn twee bytes die onderscheid maken tussen een lege pagina en een ROM. In het laatste geval bevat ID de waardes 41 en 42 (hex), oftewel de ASCII-karakters "A" en "B".

**INIT** (initialisation): deze bevat het adres van de specifieke initialisatie-routine voor de cartridge. Wanneer dit niet nodig is, dan staat er de waarde 0.

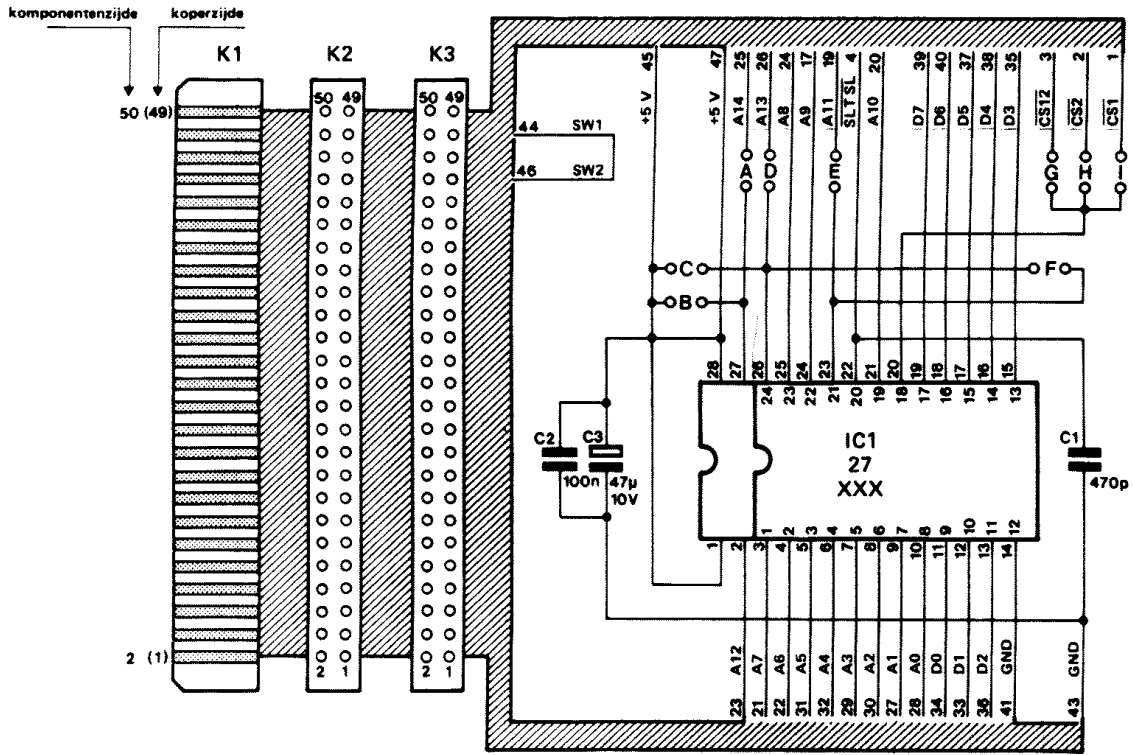
**STATEMENT**: hierin staat het adres van de uitgebreide statement-handler in de cartridge voor zover dit nodig is (zo niet, dan staat er de waarde 0). Voor meer bijzonderheden hierover verwijzen we naar de diverse MSX-handleidingen.

**DEVICE**: deze bevat het adres van de uitgebreide device-handler in de cartridge, weer met default-waarde 0. Ook hier verwijzen we naar het MSX-manual.

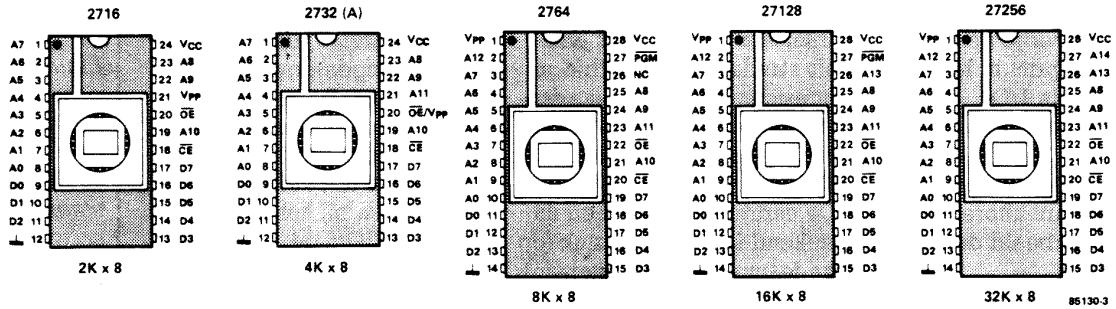
**TEXT** bevat het startadres van de BASIC-tekst (in ASCII-tekens en in tokens) in de cartridge. Indien het geen BASIC-programma betreft, is de waarde 0. Dit is voor BASIC-programmeurs een uitstekende mogelijkheid om hun specifieke programma's in EPROM te schieten. Hierop komen we later terug.

Alle bovengenoemde adressen behoren in de EPROM te staan met het lage byte vooraan en daaropvolgend het hoge byte. Dit formaat is standaard voor machinetaal; de programmeur zal dit dan ook zeker herkennen.

2



Figuur 2. De schakeling van de cartridge. Er kunnen EPROM's van 16... 256 Kbit worden gebruikt.



Figuur 3. De aansluitgegevens van de mogelijke EPROM's. Door middel van jumpers kan het gewenste type EPROM worden gekozen.

## De schakeling

Wanneer we figuur 2 aanschouwen, dan is het woord "schakeling" wat ongelukkig gekozen. Eigenlijk is het meer een "veredelde IC-voet", maar dan wel een die aangepast is op een MSX-slot met al zijn mogelijkheden! IC1 is de grote variabele. Voor de aanduiding: -XXX kunnen we kiezen uit een 2-K-uitvoering, een 4-K-uitvoering (2732), oplopend tot maar liefst 32 K (27256). Ter oriëntatie geven we in figuur 3 de aansluitgegevens van de hele reeks EPROMS. De bedrading van de IC-voet is zodanig uitgevoerd dat de verschillende geheugen-kalibers altijd "passen" (eventueel na het leggen van een jumper-tje).

We nemen de verschillende functies van de jumpers één voor één door:

**Jumper A:** deze dient voor de selectie tussen een 27256 en een 27128. Een 27256 gebruikt de adreslijn A14 zodat dan jumper A moet worden aangebracht.

**Jumper B:** hiermee wordt een 27128 op pen 27 aan +5 V gelegd. De keuze 27256/27128 bepaalt dus het respectievelijk aanbrengen van A of B. **Jumper C:** hiermee wordt pen 24 (de Vcc-aansluiting) aan + 5 V gelegd in geval van 24-pens-IC's (2716, 2732).

**Jumper D:** deze verbindt pen 26 met adreslijn A13 door (voor een 28-pens-uitvoering). Deze jumper moet worden aangebracht bij de 27128 en de -256. In het geval 2764 moet pen 26 aan de plus worden gelegd; draadbrug C dus, niet C en D tegelijk.

**Jumper E:** Deze legt pen 23 (28-pens-konfiguratie) en pen 21 (bij een 2732) aan adreslijn All en dient voor alle EPROM's behalve de 2716 te worden aangebracht.

**Jumper F:** Hiermee wordt de VPF van een 2716 aan de plus gelegd.

Dan hebben we verder nog de jumpers G, H en I. De CE-aansluiting van de EPROM kan hiermee aan CS12, CS2 of CS1 worden gelegd. CST is het ROM-select-sig-naal voor adresbereik 4000... 7FFF, CS2 voor 8000... BFFF en CS12 voor beide gebieden tegelijk (4000... BFFF). Tot en met de 27128 moet CST of C2 gekozen worden, voor de 256-versie behoort men CS12 te gebruiken. In tabel I hebben we deze gegevens vertaald naar een jumper"recept" voor de diverse EPROM's. Deze multi-EPROM-voet mag gerust universeel genoemd worden, nietwaar?

We hebben tot nu toe alleen nog maar gesproken over EPROM's: vanzelfsprekend kunnen ook (pin-compatible) PROM's of ROM's worden toegepast.

Tabel 1

	A	B	C	D	E	F	G	H	I
27256	—○—			—○—	—○—		—○—		
27128		—○—		—○—	—○—			—○*	—○*
2764		—○—	—○—		—○—			—○*	—○*
2732			—○—		—○—			—○*	—○*
2716			—○—			—○—		—○*	—○*

—○— = doorverbinding

—○\* = ter keuze (of H of I)

85130-T1

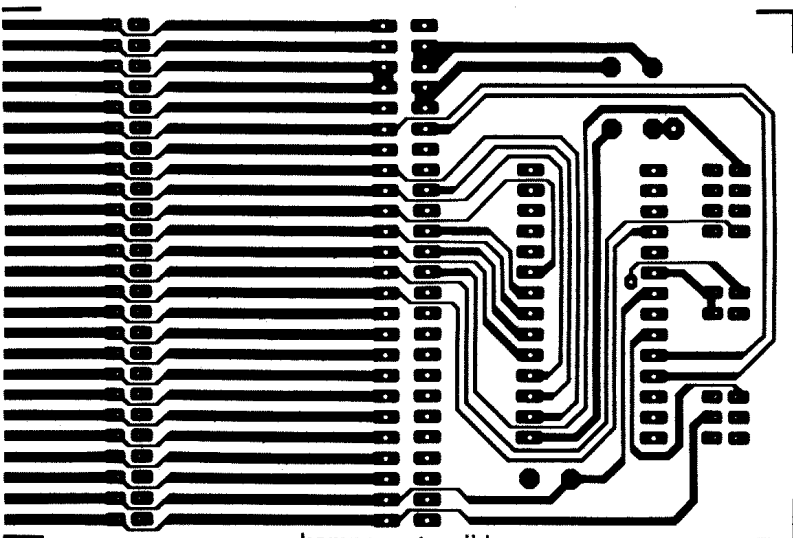
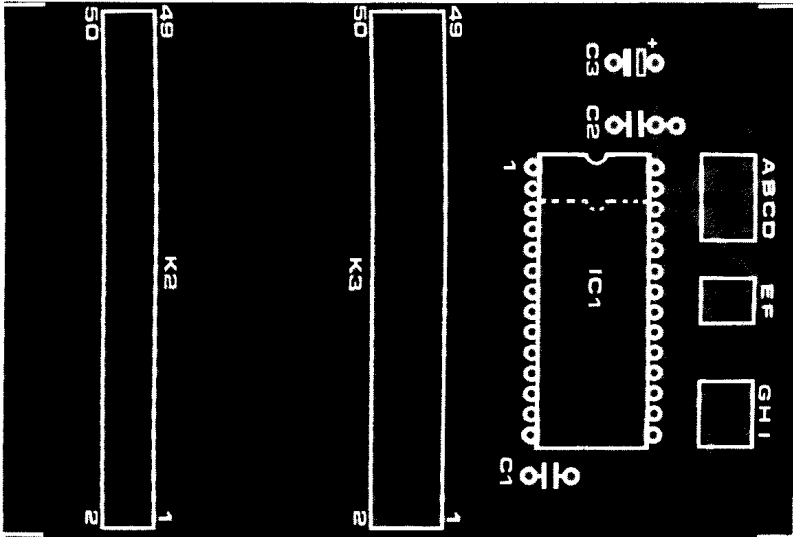
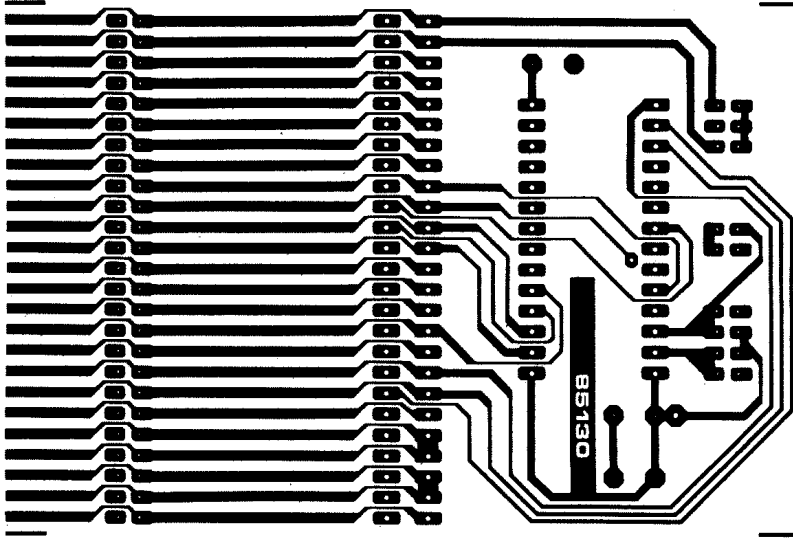
Tabel 1. Aan de hand van deze tabel kunnen we de diverse jumpers plaatsen voor het gekozen geheugenformaat. De keuze tussen H en I hangt af van de interne geheugenhuishouding (zie tekst).

SW1 en SW2 zijn op de print met elkaar doorverbonden en dienen als detektie voor insert/remove-protectie: een beveiliging waardoor er niets mis kan gaan bij het insteken en uitnemen van de cartridge (mits aanwezig).

Verder bevat de print nog een aantal konnektoren. KI is een directe (print)konnektor die in het cartridge-slot van de computer wordt gestoken. K2 is een 2x25-polige konnektor (male) terwijl K3 dient om een cartridge in te steken. Dit moet dezelfde konnektor zijn als in het slot van de MSX, en wel met een 0,1-inch-raster. Dit is omgerekend 2,54 mm. Let hierbij op dat de steek niet — volgens Europese norm — 2,5 mm bedraagt. Het verschil is dan toch te groot om 25 pennen precies te laten passen.

4

soldeerzijde



komponentenziide

*Figuur 4. Hoe klein de printplaat ook zijn mag, ze biedt toch nog plaats aan een "dikke" EPROM en twee uitbreidingskonnektors.*

## Onderdelenlijst

Kondensatoren:

C1 = 470 pF

C2 = 100 n

C3 = 47 n/10 V

Halfgeleiders:

IC1 = 2716,2732(A), 2764,27128,27256 (EP)ROM naar keuze

Diversen:

K2 = konnektor, 2x25 pins male

K3 = slot-konnektor, 2x25-polig female, 0,1 inch raster

A. . I = 2x9 pins male

konnektor

4 kortsluitjumper

## De opbouw

In figuur 4 vinden we de koper-layout en componenten-plaatsing van de cartridgeprint. Zoals u ziet, is de print dubbelzijdig uitgevoerd met doorgemetalliseerde gaten. Hierdoor kan de afmeting van de print minimaal worden gehouden. De contactstrippen van konnektor K1 zijn, evenals de soldeereilanden, voorver-tind. Er hoeft maar één IC te worden gemonteerd, dus is het een kleine moeite om een voetje van goede kwaliteit te kiezen. Wanneer de EPROM's vaak moeten worden uitgewisseld, is het te overwegen om een ZIF-voetje (Zero Insertion Force) te nemen. Met het solderen zelf hoeven we geen moeilijkheden te verwachten mits de punt van de soldeerbout niet al te grof is.

## Het gebruik

Wat doen we ermee? Net als bij iedere andere cartridge kunnen we ook hier bepaalde programma's in kwijt die vaker gebruikt moeten worden en waarbij het steeds opnieuw laden van cassette lastig is. Dit kunnen spelletjes zijn, maar ook bepaalde Utilities die men door simpelweg inprikken steeds bij de hand heeft. Hoe gaat nu het opslaan van een BASIC-programma in zijn werk? We hebben al gezien hoe het begin van een cartridge eruit ziet. Voor een BASIC-programma zijn daarvan twee dingen belangrijk: ID, liggend op adres 0000 en 0001 van de EPROM en TEXT, het startadres van het BASIC-



programma op de adressen 0008 en 0009. Aangezien de eerste 16 bytes van de cartridge-ROM zijn gereserveerd kan het startadres liggen op 0010 van de (EP)ROM. Bij de genoemde adressen moet de (verplichte) ligging van de (EP)ROM, te weten 8000, nog worden toegevoegd. Vanaf nu praten we alleen nog over adres 8000 en hoger. Om een BASIC-programma in cartridge te kunnen zetten, zullen we eerst moeten weten hoe zo'n programma in het geheugen wordt opgeslagen. Dat gaat als volgt: Op het startadres staat altijd de waarde 00 (één byte), gevolgd door een zogenaamd link-adres (twee bytes) en een regelnummer (eveneens twee bytes). Daarna staat een regel BASIC-tekst in token-vorm, d.w.z. de statements worden niet in ASCII in het geheugen gezet, maar in een byte-kode. De regel wordt afgesloten met 00. Direkt hierachter volgt weer een link-adres, een nieuw regelnummer enzovoort. Voordat we een BASIC-programma in EPROM kunnen zetten, moet er eerst wat werk verricht worden. Allereerst moet het BASIC-programma in hexadecimale vorm worden afgedrukt. Dat kan door het bijgaande "LDUMP" - programma achter het bestaande programma te knopen (zie tabel 2), bijvoorbeeld vanaf regel 10000. Na "RUN 10000" vraagt het programma om een begin- en een eindadres. Het beginadres is in ons geval altijd 8000; het eindadres is natuurlijk afhankelijk van de lengte van het programma. Het resultaat van dit programma zien we in tabel 3: een afdruk van zijn "eigen" geheugengebied.

Ten tweede moeten alle linkadressen met 10(hex) worden verhoogd. In de hele tabel moeten alle link-adressen worden opgezocht. Moeilijk is dat niet, want ieder link-adres verwijst naar de geheugenplaats waar het volgende link-adres is te vinden. Let wel even op: het link-adres en het regelnummer staan beide "omgekeerd" in het geheugen, dus eerst het LSB en dan het MSB. Het einde van het programma is gekenmerkt door een linkadres met de waarde 0000. Omdat het dumpprogramma achter het te onderzoeken programma staat, is het einde van het totale programma ongeveer 160 bytes "te ver". Het reële einde van het programma moet dus nog worden gezocht, bijvoorbeeld door te zoeken naar regelnummer 10000 (2710(hex)). Het direkt daarvoor liggende link-adres moet worden vervangen door 0000.

Tabel 2

```

10 CLS
20 INPUT "start" ;A
30 INPUT "end";B
40 FOR C = A TO B
50 LPRINTUSING"\ \";HEX$(C);:LPRINT" ";
60 FOR D=0 TO 15
70 LPRINTUSING"\\";HEX$(PEEK(C+D));:LPRINT" ";i
80 NEXT
90 C=C+15:LPRINT" ":LPRINT" "
100 NEXT
110 END

```

*Tabel 2. De listing van een BASIC-hexdump-programma.*

Om een indruk te krijgen hoe de hele procedure in zijn werk gaat, kan men eerst eens het dumpprogramma intypen en vervolgens te laten runnen met startadres &H8000 en eindadres &H8100. Uit een vergelijking van het afgedrukte resultaat met tabel 3 zal dit alles wel duidelijk worden. Wie niet de beschikking heeft over een printer, kan de dump naar het beeldscherm laten voeren door alle LPRINT-statements te vervangen door PRINT. Bovendien moet in regel 60 en 90 het getal 15 worden vervangen door 7 in verband met het maximaal aantal karakters per regel. In tabel 4 vinden we de hex-dump van de te programmeren EPROM. De eerste 16 bytes (de bovenste regel) zijn bestemd voor de identifikatie-kode en moeten in dit geval altijd opgenomen worden. Op adres 8010 komt het eigenlijke BASIC-programma. De gerasterde link-adressen zijn precies 10(hex) opgehoogd (in vergelijking met tabel 3). Het aldus verkregen ROM-BASIC-programma zal dan automatisch starten, maar — let op — er kan niet ge-edit of toegevoegd worden! We werken nu eenmaal niet in RAM. Wél worden in RAM de variabelen-waarden bijgehouden. In tabel 3 gebeurt dit direkt na het einde van het BASIC-programma (in dit voorbeeld vanaf 80A4). Wanneer de zelf geprogrammeerde EPROM in het slot steekt, dan begint het RAM-bereik bij C000. De variabelen starten dan op C010 (na te gaan met het DUMP-programma).

## **Kabel**

De print hoeft niet zonder meer in het cartridge-slot geplaatst te worden: de Spectravideo-MSX — om maar eens een voorbeeld te noemen — bezit ook een uitbreidings-konnektor waarop de print "past" door middel van een verbindingkabel. Zodoende blijft het cartridge-slot zelf vrij voor andere toepassingen. In figuur 5 zien we hoe deze flatcable-verbinding eruit ziet.

Het kabeltje zelf maken we van twee perskonnektoren (female, 2x25-pens) en een kort stukje 50-aderige flatcable. Let even op bij de Spectra-video-konnektor: de nummering hiervan komt weliswaar overeen met de nummering van het cartridge-slot, maar is in spiegelbeeld bedraad. Dit heeft tot gevolg dat het merkteken op deze konnektor (een klein pijltje, uiterst rechts boven) niet, zoals gebruikelijk, wijst naar pen 1 maar naar pen 50. Even opletten dus, maar als men zich houdt aan het voorbeeld in figuur 5, dan zit het wel snor. De print kan ook als adapterstekker worden gebruikt tussen de MSX en de Yamaha-synthesizer. De print moet dan worden doorgezaagd tussen K2 en K3, zodat alleen een male-male-verloopteel overblijft.

**Tabel 3**

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8000	0	7	80	A	0	9F	0	16	80	14	0	85	22	87	74	61
			L8007	*10		Tk	eol	L8016	*20			Tk		s	t	a
8010	72	74	22	38	41	0	23	80	1E	0	85	22	65	6E	64	22
	r	t	“	;	A	eol	L8023.	*30			Tk	“	e	n	d	“
8020	38	42	0	33	88	28	i0	*4#	82	20	43	20	EF	20	41	M
		B	eol	L8833					sp	C	sp	Tk	^E-	A	jsp	Tk
8030	20	42	0	4E	80	32	0	9D	E4	22	5C	20	20	5C	22	3B
	sp	B	eol	L804E	*50			Tk	Tk		\	sp	sp	\	“	;
8040	FF	9B	28	43	29	3B	9D	E4	22	20	20	22	3B	0	5D	80
	Tk	Tk	(	C	)	;	Tk	Tk	“	sp	sp	“	;	eol	L80	50
8050	3C	0	82	20	44	EF	11	20	D9	20	F	F	0	78	80	46
	*60		Tk	sp	D	Tk	0	3p	Tk	sp	Tk	15	eol	L8078		
8060	0	9D	E4	22	5C	5C	22	3B	FF	9B	28	FF	97	28	43	F1
	70	Tk	Tk	“	\	\	“	;	Tk	Tk	(	Tk	Tk	(	C	Tk
8070	44	29	29	3B	3A	90	22	20	22	3B	0	81	80	50	0	83
	D	)	)	;	:	Tk	“	sp	”	;	eol	L8081	*80			Tk
8080	0	96	80	5A	0	43	EF	43	F1	F	F	3A	9D	22	20	22
	eol	L8096	*90			C	Tk	C	Tk	Tk	15	:	Tk	“	sp	“
8090	3A	9D	22	20	22	0	9C	80	64	0	83	0	A2	80	6E	0
	:	Tk	“	sp	“	eol	L809C	*100			Tk	eol	L80A2	*110		
80A0	81	0	0	0												
	Tk	eol	eop	eop												

- regelnummer nn
- L link adres llhh
- Sp spatie
- Eol einde van regel
- Eop einde van programma

*Tabel 3. Zó staat het programma in RAM.*

**Tabel 4**

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
8000	41	42	0	0	0	0	0	0	10	80	0	0	0	0	0	0
8010	0	17	80	A	0	9F	0	26	8D	14	0	85	22	73	74	61
8020	72	74	22	3B	41	0	33	80	1E	0	85	22	65	6E	64	22
8030	3B	42	0	43	80	28	0	82	20	43	20	EF	20	41	20	D9
8040	20	42	0	5E	80	32	0	9D	E4	22	5C	20	20	5C	22	3B
8050	FF	9B	28	43	29	3B	3A	9D	22	20	20	22	3B	0	6D	80
8060	3C	0	82	20	44	EF	11	20	D9	20	F	F	0	8B	80	46
8070	0	9D	E4	22	5C	5C	22	3B	FF	9B	28	FF	97	28	43	F1
8080	44	29	29	38	3A	9D	22	20	22	3B	0	9D	80	50	0	83
8090	0	A6	80	5A	0	43	EF	43	F1	F	F	3A	9D	22	20	22
80A0	3A	9D	22	20	22	0	AC	80	64	0	83	0	B2	80	6E	0
80B0	81	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

*Tabel 4*

*De hexdump van het programma in EPROM. Let op: ten opzichte van tabel 3 zijn en de geheugenplaatsen en de linkadressen met 10 (hex) opgehoogd.*