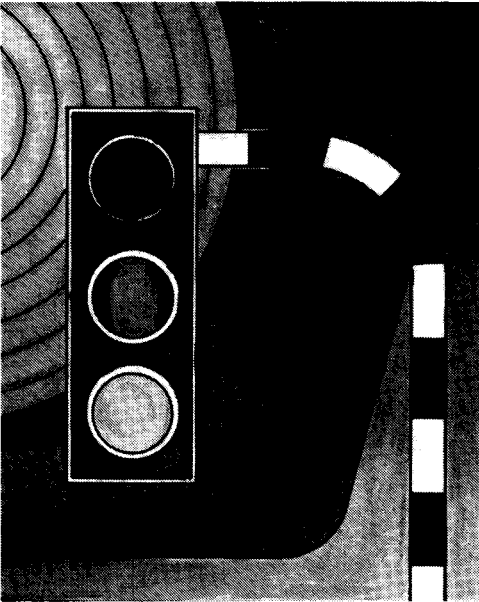


All Elektuur Junior articles 1983

- 5-60 morse-dekoder
- 5-37 junior-programmatester
- 6-48 RTTY-dekoder
- 7-73 busy-indicator voor junior
- 7-79 vektor-aansturing voor junior
- 9-58 VDU-kaart
- 9-66 de muzikale junior
- 9-74 64K op de dynamische RAM-kaart
- 10-57 basicode-2 voor de junior
- 10-72 EPROMmer zonder interfacekaart

De klassieke opgave voor een microprocessor is het met behulp van een programma (software) verzorgen van een procesbesturing, waarvoor anders een grote hoeveelheid hardware nodig zou zijn. Wat in het groot al lang is gerealiseerd, kan natuurlijk ook in het klein (modelbouw). Voor de microcomputer nemen we dan de junior computer.

verkeerslichtensturing...



De hardware kan gemakkelijk op een stuk gaatjesbord worden gemonteerd. Voor de verbindingen naar de verkeerslichten gebruikt men twee stukken drie-aderig snoer. De positieve voedingsspanning kan worden betrokken van de basisvoeding van de junior computer (van elke C5). Elk ander, ongestabiliseerd voedingsapparaat voor 12 V is eveneens bruikbaar. Lampjes met een andere werkspanning zijn uiteraard ook te gebruiken als men de voeding daar op aanpast. Wil men LED's gebruiken, dan moeten alle LED's een eigen voorschakelweerstand hebben (1k2, ½ W).

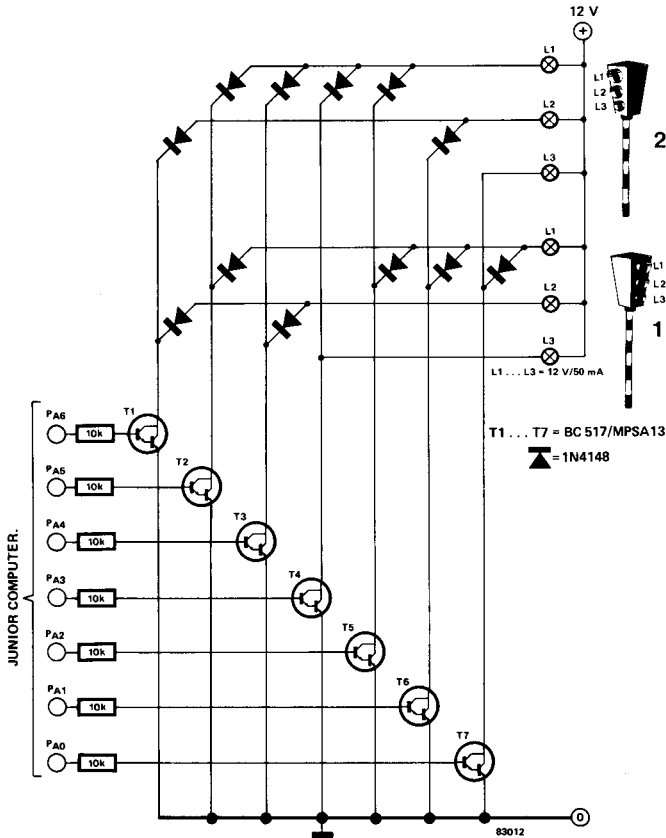
... met de
junior
computer

D. Herzberg

Zoals bij alle systemen waarin de hardware een ondergeschikte rol speelt, beperken we ons ook in dit artikel tot enkele opmerkingen over de opbouw van de schakeling. Zo bevat figuur 1 alleen maar de stuurtrappen voor de gloeilampjes in twee verkeerslichten, samen met de toevoerleidingen en een diodenmatrix die tegelijk een beveiligende functie vervult. De beide verkeerslichten worden gestuurd door het programma in tabel 1. Met de start op adres 0200 begint de computer een "1" door te schuiven van PA1 via PA2 enz. naar PA5 en vervolgens naar PA0. PA6 wordt daarbij overgeslagen. De lichtfasen verlopen dan als in tabel 2 is aangegeven. Na fase 0 begint het zaakje weer bij fase 1 enz. Een ander type signalering begint bij start-adres 023F. In dat geval knipperen de gele lampen in een ritme van: 1s aan, 1s uit, 1s aan enz. Daarvoor wordt PA6 gebruikt. Verdere geheimen heeft het programma niet. In de hex-dump-listing (tabel 3) is nog eens de in te toetsen data samengevat.

Als er om een of andere reden een storing in de programma-afloop optreedt, zorgen de dioden er voor dat minstens één van de verkeerslichten op "rood" springt. Op die manier kan geen verkeerschaos ontstaan en zijn ongelukken als gevolg daarvan praktisch uitgesloten.

1



0010:	0200		ORG	\$0200	
0020:					
0030:					
0040:			TRAFFIC LIGHT SIMULATION WITH THE JUNIOR COMPUTER		
0050:					
0060:					
0070:			WRITTEN BY DIETER HERZBERG BERLIN		
0080:					
0090:			DEFINITION OF ADDRESSES		
0100:					
0110:	0200	PA	*	\$1A80	PORT A DATA REGISTER
0120:	0200	PADD	*	\$1A81	PORT A DATA DIRECTION
0130:					
0140:	0200	TIMERD	*	\$1A97	
0150:	0200	END	*	\$1A85	
0160:					
0170:					
0180:	0200 A9 7F	START	LDAIM	\$7F	INITIALIZE PORT A
0190:	0202 8D 81 1A		STA	PADD	
0200:					
0210:	0205 A0 02	LOOP	LDYIM	\$02	LOOP COUNTER
0220:	0207 8C 80 1A		STY	PA	BEGIN WITH PHASE 1 =A1:RED/A2:AMBER
0230:					
0240:	020A 20 2F 02	NEXT	JSR	SHORT	WAIT FOR 2 SECONDS (PHASE 1+2)
0250:	020D 0E 80 1A		ASL	PA	PHASE 2+3
0260:	0210 88		DEY		
0270:	0211 D0 F7		BNE	NEXT	
0280:	0213 20 2C 02		JSR	LONG	WAIT FOR 10 SEC.AT PHASE 3
0290:	0216 A0 02		LDYIM	\$02	PHASE 4+5
0300:					
0310:	0218 0E 80 1A	NEXTA	ASL	PA	
0320:	021B 20 2F 02		JSR	SHORT	WAIT FOR 2 SEC. (PHASE 4+5)
0330:	021E 88		DEY		
0340:	021F D0 F7		BNE	NEXTA	
0350:	0221 A9 01		LDAIM	\$01	PHASE 6=PHASE 1
0360:	0223 8D 80 1A		STA	PA	
0370:	0226 20 2C 02		JSR	LONG	WAIT FOR 10 SEC. AT PHASE 1
0380:	0229 4C 05 02		JMP	LOOP	
0390:					
0400:					
0410:	022C A2 28	LONG	LDXIM	\$28	DELAY FOR 10 SEC.
0420:	022E 2C		=	\$2C	
0430:					
0440:	022F A2 08	SHORT	LDXIM	\$08	DELAY FOR 2 SEC.
0450:					
0460:	0231 A9 F4	LOAD	LDAIM	\$F4	250 MS
0470:	0233 8D 97 1A		STA	TIMERD	DIVISION FACTOR 1024 MS
0480:					
0490:	0236 2C 85 1A	TIMEND	BIT	END	TIME OUT?
0500:	0239 10 FB		BPL	TIMEND	NO
0510:	023B CA		DEX		YES LOAD TIMER WITH 2ND DELAY
0520:	023C 10 F3		BPL	LOAD	2ND TIME OUT?
0530:	023E 60		RTS		
0540:					
0550:					
0560:	023F A9 7F	US	LDAIM	\$7F	ONLY IF LIGHTS ARE OUT OF ORDER
0570:	0241 8D 81 1A		STA	PADD	
0580:	0244 A9 40		LDAIM	\$40	
0590:					
0600:	0246 8D 80 1A	PORT	STA	PA	BOTH AMBER LIGHTS FLASHING
0610:	0249 A2 04		LDXIM	\$04	1 SEC. ON/OFF
0620:	024B 20 31 02		JSR	LOAD	
0630:	024E AD 80 1A		LDA	PA	
0640:	0251 49 40		EORIM	\$40	INVERT PORT
0650:	0253 4C 46 02		JMP	PORT	
0660:					
SYMBOL TABLE 3000 3054					
END	1A85	LOAD	0231	LONG	022C
NEXT	020A	NEXTA	0218	PA	1A80
PORT	0246	SHORT	022F	START	0200
TIMERD	1A97	US	023F	LOOP	0205
				PADD	1A81
				TIMEND	0236

Tabel 2

Tabel 1. Met dit programma kan de Junior computer twee verkeerslichten (modelbouw) sturen.

licht	1	2	3	4	5	0	6
I	rood	rood	groen	geel	rood	rood	geel
II	geel	rood	rood	rood	rood	groen	geel

Tabel 2. In deze tabel is het verloop van de lichtfasen van de beide verkeerslichten aangegeven.

M															
HEXDUMP: 200,255															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E
0200:	A9	7F	8D	81	1A	A0	02	8C	80	1A	20	2F	02	0E	80
0210:	88	D0	F7	20	2C	02	A0	02	0E	80	1A	20	2F	02	88
0220:	F7	A9	01	8D	80	1A	20	2C	02	4C	05	02	A2	28	2C
0230:	08	A9	F4	8D	97	1A	2C	85	1A	10	FB	CA	10	F3	60
0240:	7F	8D	81	1A	A9	40	8D	80	1A	A2	04	20	31	02	AD
0250:	1A	49	40	4C	46	02									
JUNIOR															

Tabel 3. Deze hex-dump-listing geeft een samenvatting van de in te toetsen data.

P. Wild

Het zelf ontwikkelde programma is klaar. Even intypen, starten en de zaak loopt, . . . tenminste dat hoopt men. Doch dan blijkt maar al te vaak de executie van het programma uit te blijven. Zou er dan toch iets met het programma fout zijn? Even kijken. Maar waar? In ieder geval de inhoud van de akku, het X- en Y-register en het statusregister van de processor controleren. Aan de hand van de registerinhouden kan men het vlugste de oorzaak(en) opsporen. Dat kan op een eenvoudige en snelle wijze met de Junior-programmatester, een combinatie van een weinig hardware en een kort hulpprogrammaatje.

junior-programmatester

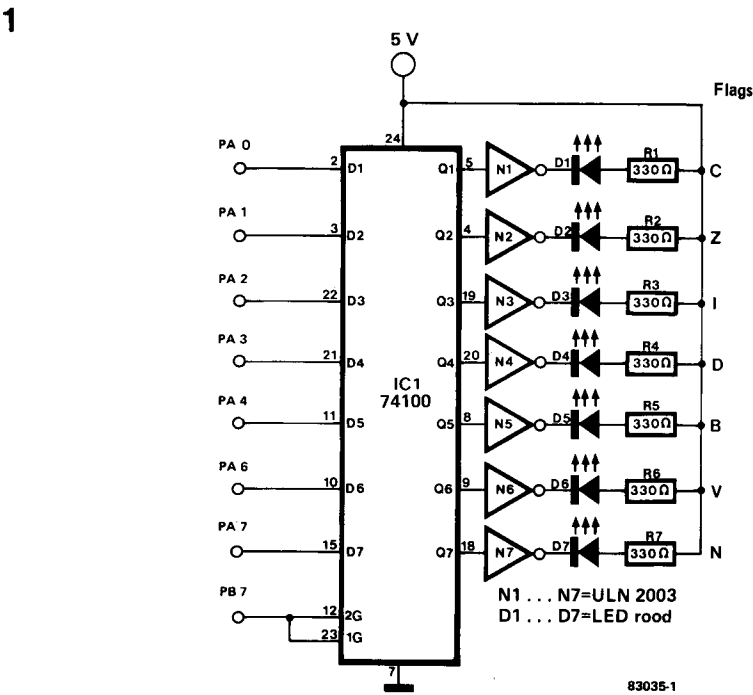
hulp voor
software-
ontwikkelaars

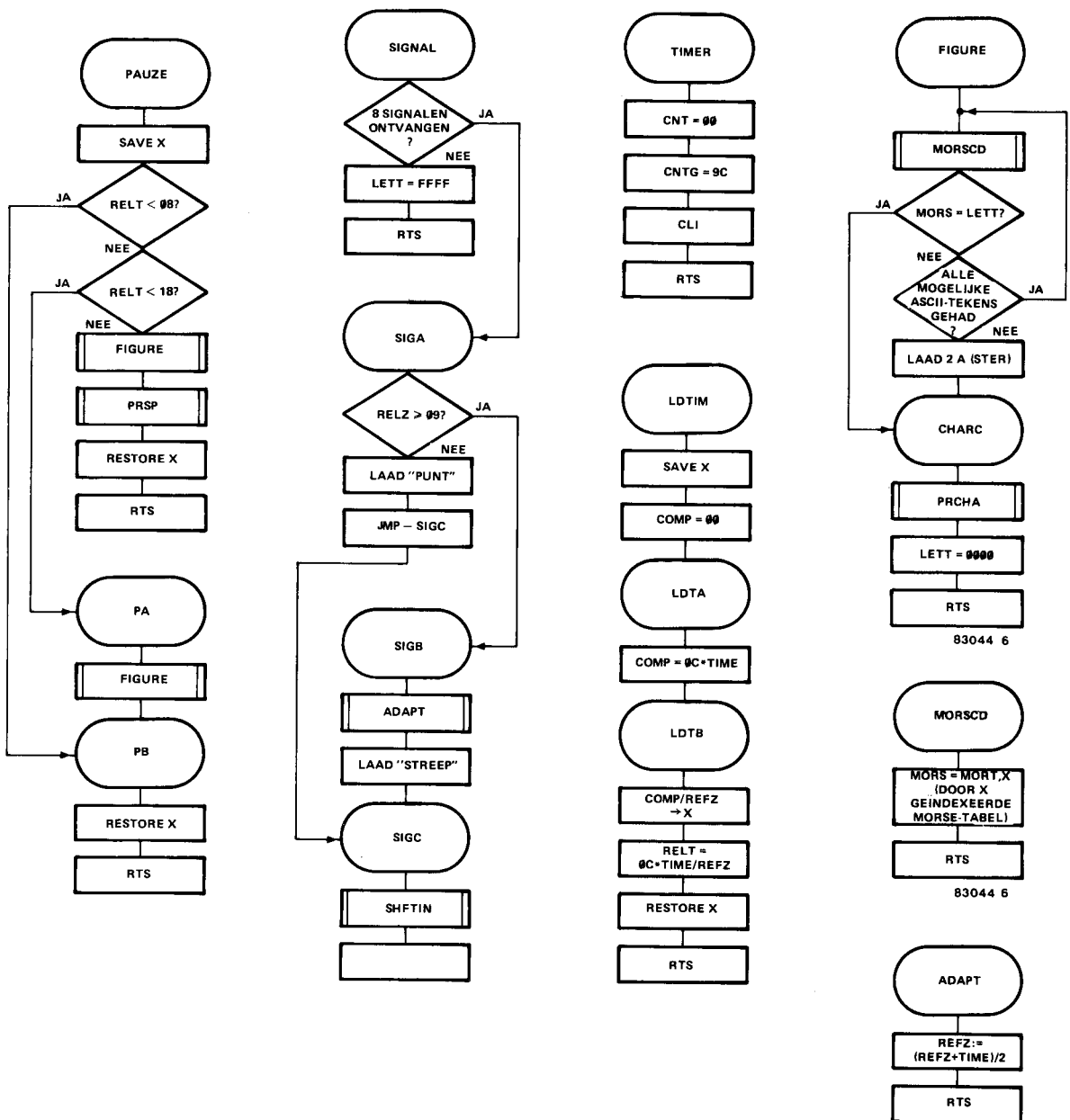
Het controleren van de registerinhouden betekent geen groot probleem en vergt slechts een kort en eenvoudig hulpprogramma. Dit programma bevat enkele instructies voor het uitlezen van de akku-inhoud. Ze zorgen er voor dat op het display van links naar rechts de inhouden van de akku, het X- en het Y-register verschijnen. Ook bevat het programma instructies voor het uitlezen van het processor-statusregister. Naast dit hulpprogramma is nog wat hardware nodig. Figuur 1 laat zien waaruit die bestaat. Aangezien het display geen plaats biedt voor de hexadecimale uitlezing van het processor-statusregister, worden de flags bitsgewijs uitgelezen (een hexadecimale uitlezing van

de flags is bovendien niet praktisch). Via het hulpprogramma wordt de inhoud van het processor-statusregister in poort A geladen. Van daaruit neemt het 8-bits latch-register (IC1) de logische nivo's van de flags over en maakt ze via de drivers N1 t/m N7 en de LED's D1 t/m D7 zichtbaar. De schakeling uit figuur 1 bouwt men op een apart printje, dat dan voor het testen in de poort-konnektor gedrukt wordt. Het te testen programma dient op een geschikte plaats een BRK-instructie te bevatten. De BRK-vektor moet uiteraard naar het begin van het hulpprogramma wijzen. Het hulpprogramma kan men alleen via de RST-toets verlaten.

Hulpprogramma	85	FB	STA POINTH	
	86	FA	STX POINTL	
	84	F9	STY INH	
	68		PLA	
	8D	80 1A	STA PAD	haal inhoud processor-statusregister
	A9	FF	LDA # FF	en laad die in poort A
	8D	81 1A	STA PADD	poort A uitgang
	8D	83 1A	STA PBDD	poort B uitgang = clock voor FF1 t/m FF7
	20	8E 1D	JSR SCANDS	
	4C	XX XX	JMP	terug naar /

Figuur 1. De hardware, die nodig is om de inhoud van het processor-statusregister bitsgewijs uit te lezen.





processor naar de lus onder label ML, waar hij wacht op het begin van een nieuw teken. De subroutine LDTIM zet de in TIME opgeslagen tijd om in een aan REFT gerelateerde waarde: $REFT = \$0C * TIME / REFT$. De faktor 0C is toegevoegd om afrondingsfouten bij de deling zo klein mogelijk te houden. De subroutine FIGURE vergelijkt het morse-teken in FIG A, B met de morse-kode voor de ASCII-tekenen \$22 ... \$5A. Als een morse-teken is geïdentificeerd wordt het bijbehorende ASCII-teken naar printer of video-interface gestuurd; niet geïdentificeerde morse-tekenen (bijvoorbeeld fout geseinde tekens) worden met een sterretje aangegeven (*). Het vergissingsteken (8 of meer punten) wordt aangeduid met het

ASCII-teken \$23 (#).

Subroutine SHFTIN schuift een punt of streep die door de processor is herkend in de geheugenplaatsen FIG A en FIG B. Zoals figuur 7 laat zien betekent 00: leeg, 01: punt, 10: streep en 11: fout (door LETT).

Handleiding voor het programma

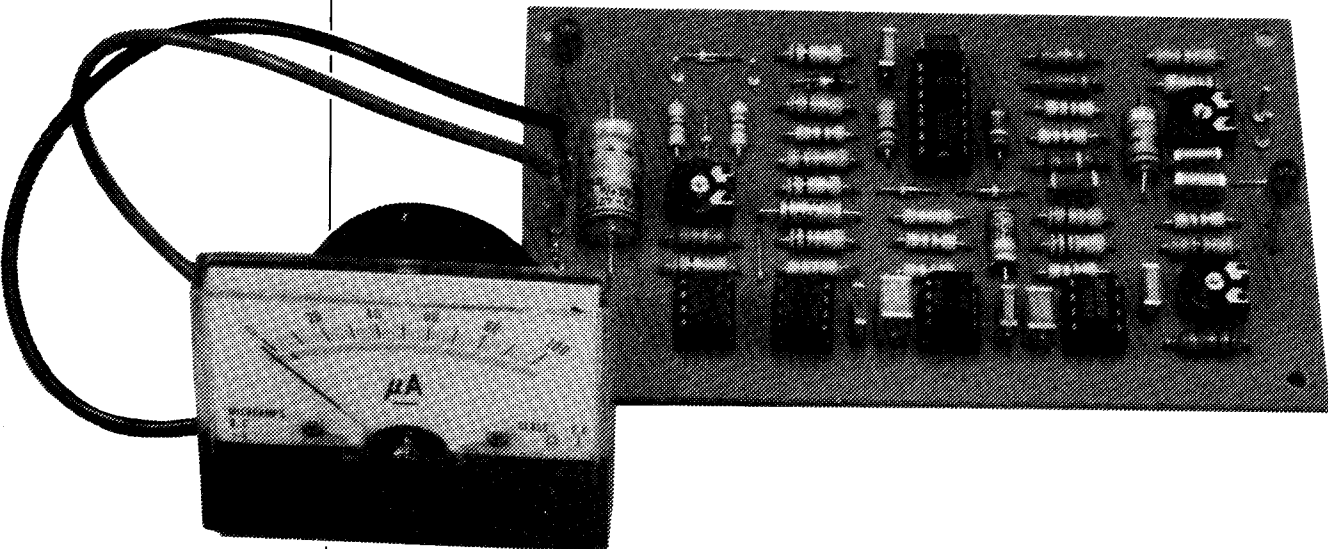
Het morse-programma maakt gebruik van de RAM-adressen 4000 ... 7FFF. Hiervoor kan bijvoorbeeld een dynamische 16 K-RAM-kaart gebruikt worden. Het startadres van het programma is 4000.

Aangezien de DOS-junior een andere geheugenindeling heeft dan de uitgebreide junior

Figuur 6. Stroomschema van het morse-decodeer-programma, waarmee de junior computer morse-tekenen kan omzetten in leesbare tekst.

Veel korte-golf-luisteramateurs zijn geïnteresseerd in de ontvangst van morse-uitzendingen. Het grote probleem daarbij is echter de morse-code, want die moet men eerst leren voordat zo'n uitzending kan worden gevolgd. En dat is niet zo gemakkelijk! Als men in het bezit is van een junior computer kan het ook op een eenvoudiger manier: men "leert" zijn computer morse, met behulp van een programma en een kleine hulpschakeling.

morse-dekoder



morse
ontcijferen
met
de computer

Wie kent tegenwoordig nog morse? In de amateurwereld zijn dat niet zoveel mensen. De enigen die morse moeten kennen zijn de zendamateurs met de licenties A en B, en zelfs die maken er niet altijd gebruik van. Morse leren is een vrij eentonige en vermoeiende bezigheid. Voordat men een echte morse-uitzending goed kan volgen zijn er dan ook maanden training aan vooraf gegaan. Niet iedereen heeft dat er voor over. Er zijn echter een hoop amateurs die zo'n uitzending wel interessant vinden, maar daarvoor niet meteen een cursus morse willen volgen. Nou, tegenwoordig hoeft dat ook niet meer, dankzij de computer en wat software.

Zo'n schakeling voor het "vertalen" van morsetekens noemt men een morse-dekoder. In principe kan zo'n dekoder worden gemaakt van elke microcomputer, met behulp van een kleine hardware-interface en een morse-dekodeerprogramma. In dit artikel wordt een universele morse-interface beschreven en een heel doordacht dekodeerprogramma voor de junior computer.

Morse dekoderen

Zoals (waarschijnlijk wel) bekend bestaat een morse-sigitaal uit een reeks korte en lange "piepjes", die strepen en punten

worden genoemd. Een streep heeft de lengte van drie punten. Minder bekend is de samenstelling van die punten en strepen. Daarmee bedoelen we niet het aantal punten en strepen van een teken, maar de pauzen tussen de punten en strepen. De lengte van zo'n pauze geeft namelijk ook informatie. Tussen de punten en strepen van een teken zijn de pauzes gelijk aan de duur van één punt, tussen twee tekens is de lengte van de pauze drie punten en tussen twee woorden bedraagt de pauzelengte vijf punten.

Het probleem bij het dekoderen van morsetekens zit 'm in het feit dat al die mooie verhoudingen in de praktijk niet zo nauwkeurig worden aangehouden. De punt- en streepduur kan variëren, maar ook de pauzetijden, de verhoudingen van die verschillende tijden en de seïnsnelheid. Kortom, alle grootheden bij morse zijn variabel.

Een goede telegrafist is in staat om zelfs bij zeer grote afwijkingen van punt-, streep- en pauzeduur de tekens nog duidelijk te herkennen. Bij een computer is het niet zo eenvoudig om deze te "leren" hoe hij de diverse signalen moet interpreteren. Hij kan echter wel eenvoudig morse leren. Een andere moeilijkheid voor de computer wordt gevormd door de vele storingen die voorkomen op de korte golf: atmosferische

storingen, radiostoringen, interferentie-fluitjes, storingen van naburige stations, ruis, enzovoorts. Een goede telegrafist kan zelfs nog zeer sterk verruiste morse-signalen herkennen. Zoiets lukt een computer-morse-dekoder (nog) niet. Voor de luisteramateur die de morse-kode niet beheerst is zo'n computer-dekoder echter een prima hulpmiddel voor de ontvangst van de wat sterkere stations. Alleen zeer zwakke of onregelmatig seinende stations geven moeilijkheden bij de dekodering. De computer kan natuurlijk niets doen met de morsetonen die uit de ontvanger komen. Er is een interface nodig die de morse-piepen omzet in een voor de computer begrijpelijk digitaal signaal en tevens de aanwezige stoorsignalen onderdrukt. Daarna kan de computer de lengte (tijdsduur) van de door de interface geleverde blokspanningen en pauzen gaan meten. Door middel van enkele routines kan hij dan nagaan of hij de gemeten waarde moet interpreteren als punt, streep, korte, middel-lange of lange pauze. Als alle elementen op die wijze zijn herkend kunnen ze worden samengesteld tot binair gekodeerde tekens, die vervolgens worden omgezet in ASCII-kode.

Figuur 1 toont de blokschematische opzet van een μ P-morse-dekoder. In plaats van een video-interface met monitor kan men natuurlijk ook een printer op de computer aansluiten.

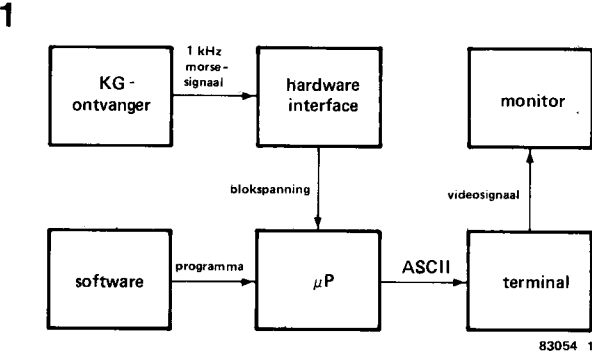
De morse-interface

De taak van de morse-interface is in principe vrij eenvoudig: als aan de ingang een 1 kHz-signaal staat moet de uitgang een logische één leveren. Staat geen signaal aan de ingang, dan is de uitgang nul. Een onderbroken

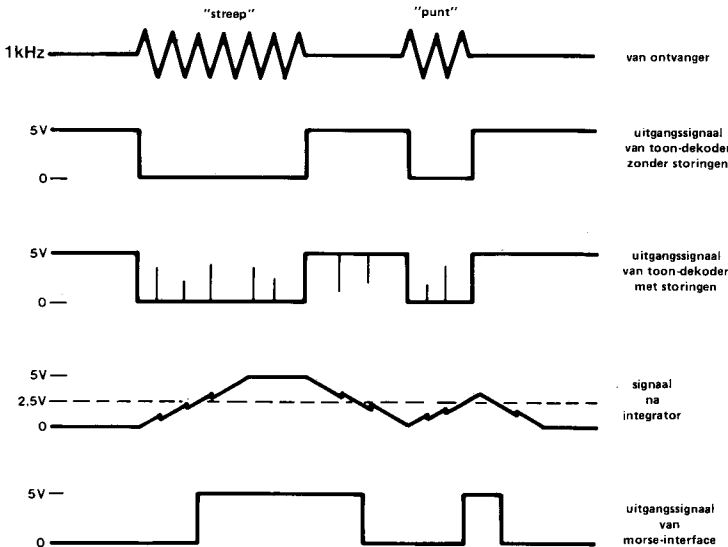
1 kHz-toon (morse-signaal) levert dus aan de uitgang van de interface een blokspanning waarvan de "1"-lengte overeenkomt met de lengte van de 1 kHz-tonen. Met behulp van de BFO (Beat Frequency Oscillator) van de KG-ontvanger kan men de frekwentie van de morse-tonen instellen op precies 1 kHz. Aangezien de morse-interface alleen reageert op deze frekwentie worden stoorsignalen met afwijkende frekwenties zo behoorlijk onderdrukt. Tegen impuls-vormige storingen is deze frekwentie-selektiviteit van de toon-dekoder in de morse-interface echter niet opgewassen, omdat zo'n impuls een vrij breed frekwentie-spektrum bezit. Een stoorimpuls die midden in een pauze verschijnt kan dan (foutief) als punt worden herkend. Voor het uifilteren van zulke storingen bevat de interface nog een integrator die er voor zorgt dat alleen impulsen met een bepaalde minimumlengte (duur) aan de uitgang van de toon-dekoder een logische één leveren. Op die wijze worden de meeste stoorimpulsen onderdrukt. Figuur 2 verduidelijkt de werking van de

morse-dekoder
elektuur mei 1983

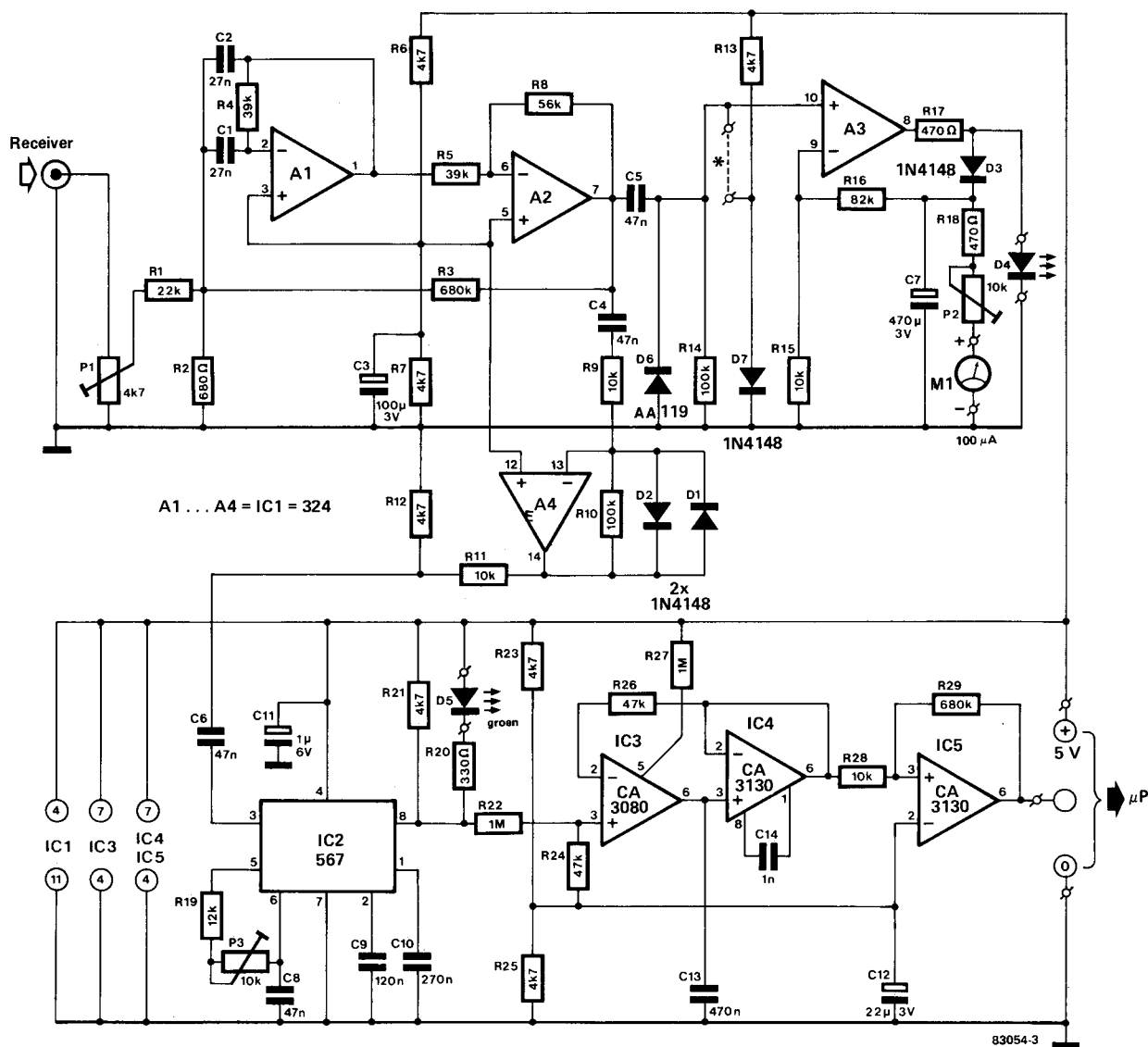
Figuur 1. Blokschema van een combinatie van een morse-dekoder en een computer. De belangrijkste delen zijn hier de hardware-interface en de decodeer-software.



2



Figuur 2. Vereenvoudigd tijdvolgordediagram voor de interface-schakeling. De verschillende morse-signalen worden door een toon-dekoder omgezet in blokspanningen. Een integrator en komparator zorgen voor het onderdrukken van korte stoorsignalen



Figuur 3. Het schema van de morse-interface. De morse-tonen worden door de toon-decoder IC2 "omgezet" in blokspanningen. Een vrij uitgebreide nivo-indikatie vergemakkelijkt het afstemmen van de ontvanger bij de ontvangst van morse-tekens.

morse-interface aan de hand van diverse signalen uit de schakeling. Het schema van de morse-interface is in figuur 3 afgebeeld, terwijl de print in figuur 4 te zien is. De interface wordt aangesloten op de opname-uitgang van de ontvanger. De ingangsgoedigheid van de interface kan worden ingesteld met P1. Rond de opamps A1 en A2 is een 1 kHz-filter opgebouwd. Daarna volgt A4, die als tienmaal-versterker is geschakeld. De dioden D1 en D2 in de tegenkoppeling van A4 zorgen er voor dat het uitgangssignaal van de opamp wordt begrensd tot + en -600 mV. Via een verzwakker (R11, R12) en een condensator (C6) komt het uitgangssignaal van A4 terecht bij de ingang van het toon-decoder-IC 567 (IC2). De uitgang van dit IC (pen 8) wordt logisch nul als een 1 kHz-signaal op de ingang staat. De groene LED licht dan op. Daar de toon-decoder ook reageert op korte stoorimpulsen met een nul aan zijn uitgang is de schakeling rond IC3... IC5 toegevoegd voor het onderdrukken van zulke impulsen. IC3, een OTA, is als integrator geschakeld, waarbij

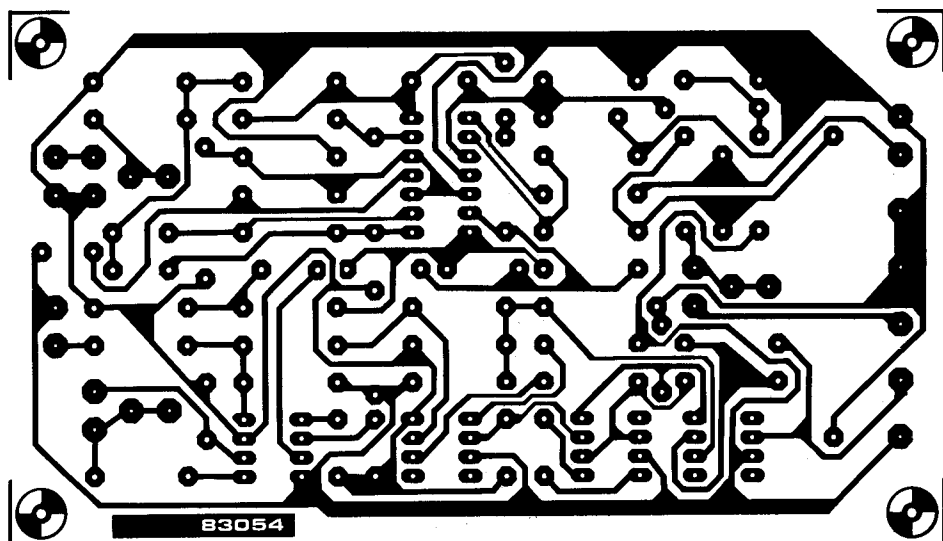
de integrator-tijdconstante wordt bepaald door de stroom, die via R27 naar pen 5 van de OTA gaat, en door condensator C13 aan de uitgang. De uitgangssignalen van IC2 worden door IC3 geïntegreerd, zodat aan de uitgang van de OTA een signaal staat zoals in figuur 2 getekend is. IC4 dient als buffer, zodat C13 niet belast wordt. IC5 werkt als komparator, waarbij de komparatorspanning op 2,5 V is ingesteld. Boven 2,5 V geeft de uitgang (pen 6) een "1", beneden 2,5 V een "0".

Het gefilterde 1 kHz-signaal (aan de uitgang van A2) gaat ook naar A3, een actieve enkelzijdige gelijkrichter. Deze stuurt de draaispoelmeter die de signaalsterkte aangeeft. De rode LED D4 is als oversturings-indikator toegevoegd.

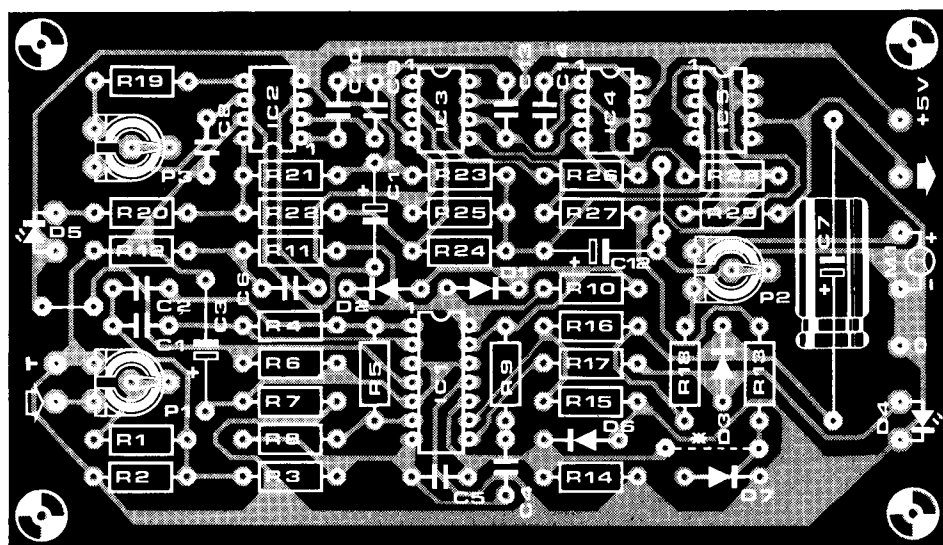
Diode D7 en weerstand R13 worden alleen gebruikt voor het afregelen van de meter. De diode levert een "referentiespanning" van 0,6 V aan opamp A3 als de met een sterretje aangegeven draadbrug wordt gelegd.

Het afregelen van de morse-interface

Eerst wordt de signaalsterktemeter geijkt.



Figuur 4. De print waarop de schakeling van de morse-interface kan worden ondergebracht.



Onderdelenlijst

Weerstanden:

R1 = 22 k
R2 = 680 Ω
R3, R29 = 680 k
R4, R5 = 39 k
R6, R7, R12, R13, R21, R23, R25 = 4k7
R8 = 56 k
R9, R11, R15, R28 = 10 k
R10, R14 = 100 k
R16 = 82 k
R17, R18 = 470 Ω
R19 = 12 k
R20 = 330 Ω
R22, R27 = 1 M
R24, R26 = 47 k
P1 = 5 k instelpotmeter
P2, P3 = 10 k instelpotmeter

Kondensatoren:

C1, C2 = 27 n
C3 = 100 μ /3 V
C4, C5, C6, C8 = 47 n
C7 = 470 μ /3 V
C9 = 120 n
C10 = 270 n
C11 = 1 μ /6 V
C12 = 22 μ /3 V
C13 = 470 n
C14 = 1 n

Halfgeleiders:

D1, D2, D3, D7 = 1N4148
D4 = LED rood
D5 = LED groen
D6 = AA 119
IC1 = LM 324
IC2 = LM 567
IC3 = CA 3080
IC4, IC5 = CA 3130

Diversen:

M1 = draaispoelmeter
100 μ A

Daartoe wordt de draadbrug bij A3/D7 aangebracht en stelt men P2 zo in dat de meter volle uitslag geeft. Draadbrug weer verwijderen en de schakeling vervolgens op de ontvanger aansluiten. Zoek nu een sterke zender op (liefst eentje die alleen een draag-golf uitzendt), zet P1 in de middenstand en verdraai dan de BFO totdat de meter M1 maximale uitslag geeft. Als de meter niet meer kan worden afgelezen (omdat de naald in de linker of rechter hoek zit) moet de gevoeligheid van de schakeling met P1 worden veranderd. Als zo de optimale stand van de BFO is gevonden wordt P1 zo ingesteld dat de draaispoelmeter volle uitslag geeft.

Vervolgens is de toon-dekoder aan de beurt. Zoek een zender op die duidelijke en regelmatige morse-tonen geeft en stel P3 dan zo in dat de groene LED (D5) knippert in het ritme van de morse-tonen. Het "vangbereik" van de toon-dekoder is vrij groot, zodat men een breed bereik vindt bij de potmeter waarbij de LED nog goed reageert. De juiste instelling van P3 ligt in het midden van dit bereik.

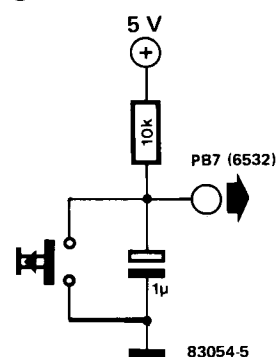
Morse-dekoder-software

Het dekoderprogramma is gemaakt voor de junior computer. De in een 2716 opgeslagen software kan worden gebruikt bij een "uitgebreide" junior en bij een DOS-junior.

De morse-interface wordt aangesloten op pen PB7 van de junior computer. Voor demonstraties en sein-oefeningen kan men ook een seinsleutel op PB7 aansluiten door middel van de anti-denderschakeling uit figuur 5.

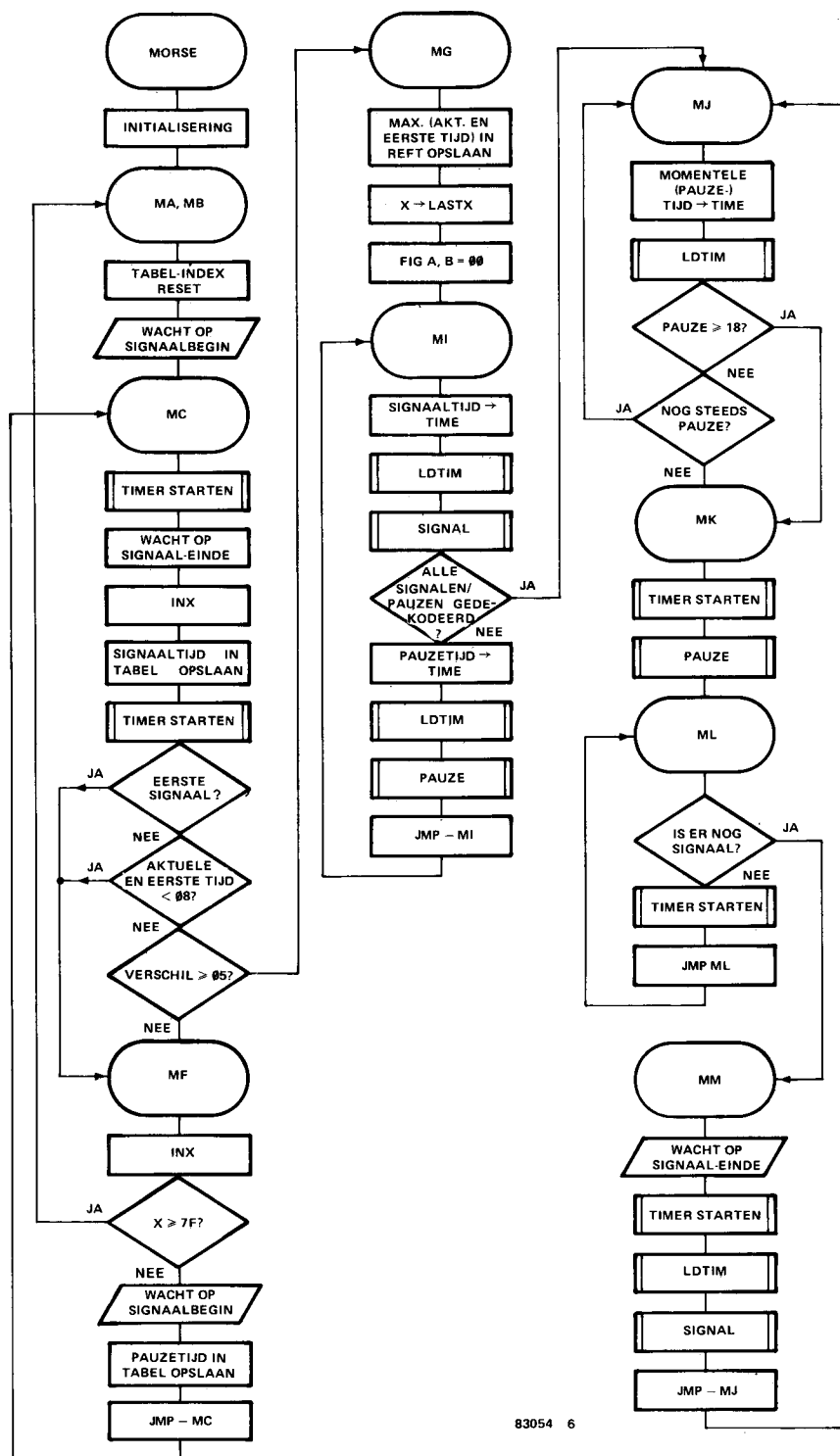
Na het starten van het programma verschijnt eerst de programmaam "MORSE-DECODER". Daarna gaat de computer de ontvangen morsetekens zo lang met elkaar vergelijken totdat hij een lengteverschil van minstens 50 ms vindt. Signalen die korter zijn dan 80 ms worden geïnterpreteerd als storingen. Als de computer een tijdsverschil van meer dan 50 ms heeft gevonden gaat hij beginnen met het decoderen van de tot dan toe ontvangen en opgeslagen morse-signalen, waarbij de eerste opgeslagen streep als tijd-referentie wordt genomen. Bij elke ont-

5



Figuur 5. Via deze dender-
onderdrukkingsschakeling
kan men een morse-sleutel
aansluiten op PB7 van de
junior computer.

6



83054 6

vangen streep wordt deze referentie gekorri-
geerd, zodat veranderingen in de seinsnelheid
door de computer direkt worden gevolgd.
De morse-dekoder geeft 64 tekens per regel
uit en zendt daarna automatisch een carriage
return-(CR)- en line feed-(LF)kommando
naar de printer of de video-interface (bij-
voorbeeld Elekterminal).
Voor het herkennen van de morse-tekens
maakt het programma gebruik van de
volgende punten:

- Bepaald minimaal verschil tussen punten
en strepen bij de start van het programma.
- Minimale pauzeduur tussen de morse-
tekens.
- Minimale pauzeduur tussen twee karak-
ters.

■ Minimale lengte van een streep.
Het kan gebeuren dat de dekoder buiten zijn
"vangbereik" komt; men kan het programma
dan opnieuw starten door het indrukken van
de NMI-toets.

Het stroomschema in figuur 6 toont de
volledige opbouw van het programma. We
geven een korte toelichting bij de belang-
rijkste punten.

Nadat de computer een eerste referentietijd
REFT heeft gevonden, bevindt hij zich in
de lus met label MJ. Tussen label MJ en
label MK wacht hij op een morse-signaal,
maar niet langer dan 18 referentie-eenheden,
anders zou het laatste ontvangen teken van
een uitzending niet gekodeerd worden. Als
de tijd langer is dan 18 eenheden gaat de

Tabel 1. Startadressen van de kopieerroutines

junior computer versie	kopieer- routine startadres	gekopieerd	
		van adres	naar adres
uitgebreid	0E56	0800	4000
DOS	EB2D	E800	4000

Tabel 2. Veranderingen bij gebruik DOS-junior met morse-programma

adres	data
4238	A3
4239	FE
426D	A3
426E	FE
4284	A3
4285	FE
428C	A3
428D	FE

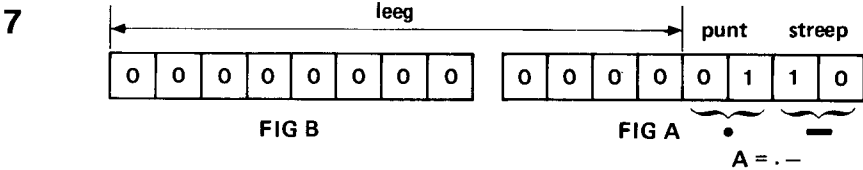
Tabel 3. Veranderingen bij gebruik uitgebreide junior met morse-programma

adres	data	adres	data
4013	1A	4234 ...	EA,EA,EA
4018	1A	4238	34
401B	1A	4239	13
4022	1A	4245 ...	EA,EA,EA
402C	1A	4251	1A
4031	1A	4268	1A
4039	1A	426D	34
403C	1A	426E	13
4048	1A	4280 ...	EA,EA,EA,
404D	1A	4284	34
40E1	1A	4285	13
40F8	1A	4288 ...	EA,EA,EA
414F	1A	428C	34
41F5 ...	EA,EA,EA	428D	13

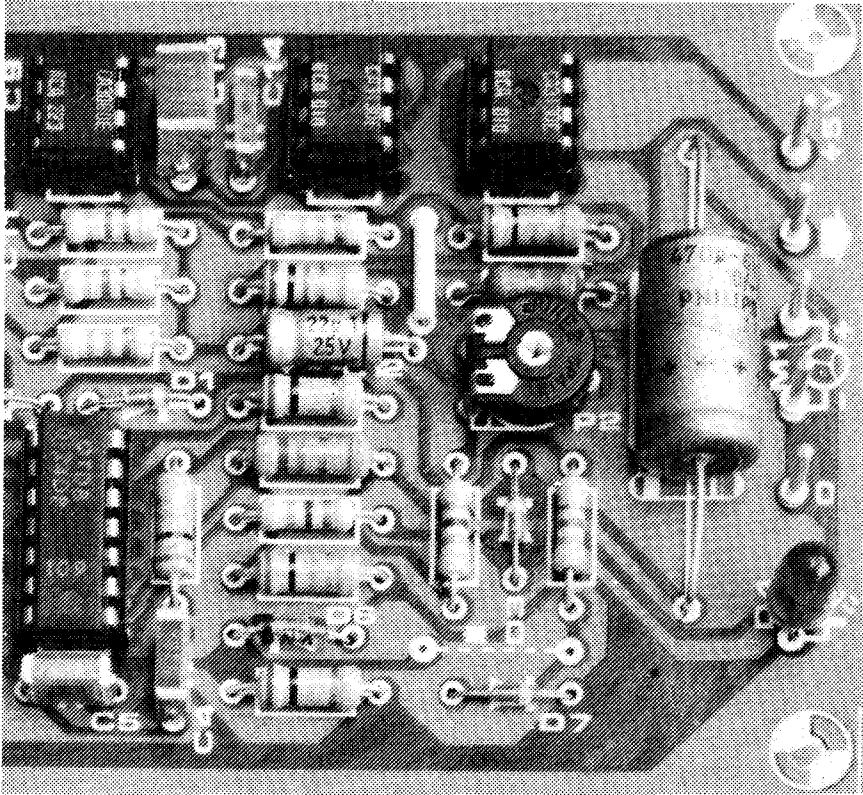
Tabel 4. De hexdump van het morse-dekoder-programma.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
800	4C	11	40	00	00	00	00	00	00	00	00	00	00	00	00	00
810	00	AD	82	FA	29	DF	8D	82	FA	AD	83	FA	09	20	29	7F
820	8D	83	FA	A9	00	8D	0F	40	A9	3F	8D	7A	FA	A9	42	8D
830	7F	FA	20	13	BC	A9	81	8D	F7	FA	2C	D5	FA	10	FB	A0
840	00	20	2D	42	A9	4C	8D	7F	FA	A9	42	8D	7F	FA	A2	FF
850	20	22	41	20	46	41	20	34	41	E8	AD	03	40	9D	2C	43
860	20	46	41	20	46	41	20	46	41	20	46	41	20	46	41	20
870	C9	08	30	12	38	BD	2C	43	ED	2C	43	B0	05	49	FF	18
880	69	01	C9	05	10	11	E8	E0	7F	10	C3	20	22	41	AD	03
890	40	9D	2C	43	4C	53	40	BD	2C	43	CD	2C	43	10	03	AD
8A0	2C	43	8D	04	40	8E	05	40	A2	00	8E	06	40	8E	07	40
8B0	BD	2C	43	8D	04	40	20	52	41	20	B2	41	EC	05	40	F0
8C0	11	E8	BD	2C	43	8D	04	20	52	41	20	91	C9	18	10	11
8D0	60	40	AD	03	40	8D	04	20	52	41	20	91	C9	18	10	11
8E0	82	FA	29	8D	00	EC	20	5C	42	D0	E7	20	5C	42	D0	E2
8F0	20	46	41	20	91	41	AD	82	FA	29	8D	00	EC	20	46	41
900	4C	F6	40	20	5C	42	D0	E7	20	5C	42	D0	E9	20	34	41
910	AD	03	40	8D	00	40	20	46	41	20	52	41	20	B2	41	4C
920	D2	40	20	66	42	D0	FB	20	5C	42	D0	F6	20	5C	42	D0
930	F1	4C	66	42	20	66	42	F0	FB	20	5C	42	F0	F6	20	5C
940	42	F0	F1	4C	66	42	A9	00	8D	03	40	A9	9C	8D	FE	FA
950	58	60	8E	09	40	A2	00	8E	0A	40	8E	08	40	A2	0C	18
960	AD	0A	40	6D	08	40	8D	0A	40	AD	0B	40	69	00	8D	0B
970	40	CA	D0	EB	38	AD	0A	40	ED	04	40	8D	0A	40	AD	0B
980	40	E9	00	8D	0B	40	E8	0B	EC	8A	8E	0C	40	AE	09	40
990	60	8E	09	40	AD	0C	40	C9	0A	30	13	C9	18	30	0C	20
9A0	D7	41	78	20	43	42	58	AE	09	40	60	20	D7	41	AE	09
9B0	40	60	AD	07	40	29	C0	F0	09	A9	FF	8D	06	40	8D	07
9C0	40	60	AD	0C	40	C9	09	10	05	A9	02	4C	D3	41	20	12
9D0	42	A9	01	20	1E	42	60	A2	22	20	05	42	AD	00	40	CD
9E0	60	40	D0	08	AD	0E	40	CD	07	40	F0	07	E8	E0	5B	30
9F0	E8	A2	2A	8A	78	8D	63	23	20	6C	42	58	A2	E0	8E	06
A00	40	8E	07	40	60	8D	6F	42	8D	0D	40	8F	42	8D	0E	
A10	40	60	18	AD	04	40	6D	08	40	4A	8D	04	40	60	4A	2E
A20	06	40	2E	07	40	4A	2E	06	40	2E	07	40	60	89	0F	43
A30	C9	00	F0	8A	8D	63	23	20	43	23	C8	4C	2D	42	60	78
A40	4C	11	40	A9	20	8D	63	23	20	6C	42	60	48	A9	9C	8D
A50	FE	FA	EE	03	40	D0	03	CE	03	40	68	40	A9	7F	8D	10
A60	40	CE	10	40	D0	FB	AD	82	FA	29	80	60	20	43	23	EE
A70	0F	40	A9	3F	CD	0F	40	D0	15	A9	00	8D	0F	40	A9	00
A80	8D	63	23	20	43	23	A9	0A	8D	63	23	20	43	23	60	00
A90	00	59	55	00	00	00	A9	69	A6	00	00	5A	56	66	59	AA
AA0	AA	6A	5A	56	55	55	A5	A9	95	99	00	56	00	A5	00	
AB0	06	95	99	25	01	59	29	55	05	6A	26	65	0A	09	2A	69
AC0	A6	19	15	02	16	56	1A	96	9A	A5	00	00	00	00	00	00
AD0	00	06	55	00	00	00	06	02	09	00	00	0A	09	06	02	02
AE0	01	01	01	01	01	02	02	02	02	0A	09	00	02	00	05	00
AF0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
B00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
B10	0A	0A	0A	4D	4F	52	53	45	20	44	45	43	4F	44	45	52
B20	0D	0A	0A	52	45	41	44	59	0D	0A	00	00	00	A9	00	A2
B30	E8	85	00	86	01	A9	00	A2	40	85	02	86	03	20	43	E8
B40	40	00	FC	A2	04	A0	00	H1	00	91	02	88	D0	F9	E6	01
B50	E6	03	CA	D0	F0	60	A9	00	A2	88	05	00	86	01	A9	00
B60	A2	48	85	02	86	03	20	6C	EB	4C	1D	1C	A2	04	A0	00
B70	B1	00	91	02	88	D0	F9	E6	01	E6	03	CA	D0	F0	60	00

Figuur 7. Met behulp van de subroutine wordt een "herkende" punt of streep in de adressen FIG A en FIG B geschoven.



83054-7

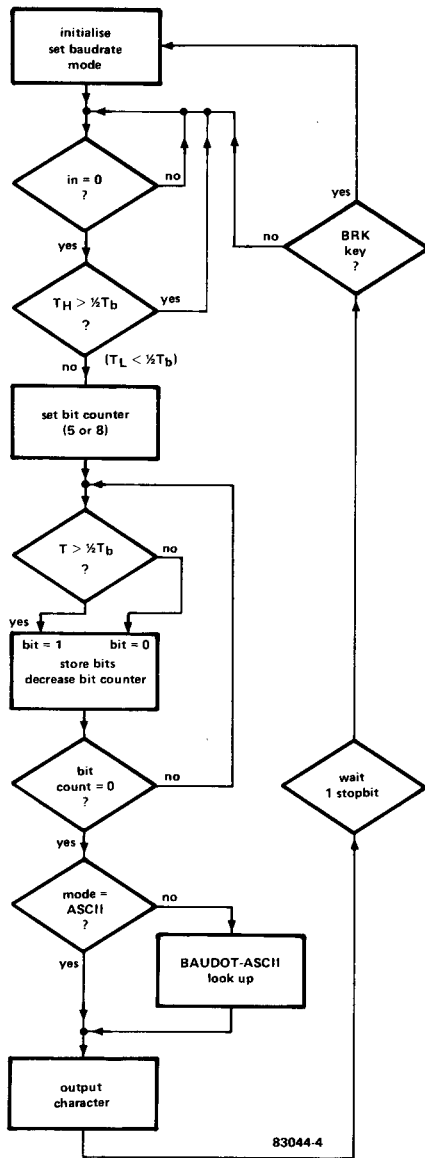


zijn in één EPROM twee verschillende versies van het morse-programma opgeslagen. De EPROM moet in het voetje IC4 op de uitbreidingskaart worden gestoken. Bij de uitgebreide junior ligt de EPROM in het bereik 0800...0FFF, bij de DOS-junior in het bereik E800...EFFF. Voordat men het programma start moet dit eerst van de EPROM in het RAM-bereik worden gekopieerd. De benodigde kopieerroutines zitten al in de EPROM. De startadressen voor de kopieerroutines zijn in tabel 1 gegeven.

Nadat het programma in RAM gezet is moet men nog enkele bytes met de hand veranderen. Ook dit is weer afhankelijk van de soort junior die gebruikt wordt. In tabel 2 zijn de veranderingen aangegeven voor de DOS-junior en in tabel 3 de veranderingen voor de uitgebreide junior. Daarna kan het programma gestart worden.

Als het (aangepaste) programma eenmaal in RAM staat kan men het op cassette opslaan of op diskette zetten (bij de DOS-junior), zodat men niet meer de veranderingen hoeft in te brengen als het programma op een later tijdstip weer wordt gebruikt. Wie de EPROM met het morse-programma zelf wil programmeren kan dat doen aan de hand van de hexdump uit tabel 4.

4



Figuur 4. Vereenvoudigd stroomschema van het RTTY-programma. Het "hart" van het programma wordt gevormd door de bitteller. Het programma bekijkt of het ingangssignaal langer dan de halve bitduur logisch één is. Aan de hand daarvan wordt beslist of een bit een "0" of een "1" vertegenwoordigt. Op deze wijze worden storingen op een bijzonder effectieve manier onderdrukt.

Tabel 2. Print-out na het starten van het RTTY-programma.

```
BAUDRATE:
0=45.45 BAUD
1=50
2=57
3=75
4=100
5=110
DO YOU LIKE TO CHANGE IT? <Y/N>Y
SELECT THE BAUDRATE: 1
ASCII RECEIVER? <Y/N>N
FILE BUFFER? <Y/N>Y
AUTO LETTER MODE? <Y/N>
LIST THE FILE BUFFER? <Y/N>
::
```

Tabel 3. Startadressen voor de kopieerroutines.

junior versie	start- adres	gekopieerd van	naar adres
uitgebreid	0E88	0800	4000
DOS	EE72	E800	4000

men gewoon alle standen van S1 totdat een stand wordt gevonden waarbij D3 fel oplicht en D4 zo weinig mogelijk oplicht. Als zo'n instelling niet kan worden gevonden in een van de standen 2... 6 van S1 hebben we waarschijnlijk te maken met een afwijkende shift. In dat geval wordt S1 in stand 1 gezet en kan men met P1 de dekoder instellen op de gebruikte shift. Als de LED's in het juiste ritme knipperen duidt dat op een goede ontvangst en een goede werking van de interface. Daarna moet nog de juiste baud rate (met de computer) en de polariteit (met S2) worden ingesteld. Dit is een kwestie van proberen, want daarvoor kunnen geen vaste regels worden gegeven.

Het RTTY-decodeerprogramma

Het programma voor de RTTY-decodering kan worden ondergebracht in een 2716-EPROM. Deze EPROM kan zowel bij de uitgebreide junior computer als bij de DOS-junior worden gebruikt. De RTTY-interface wordt aan pen PB7 van de junior aangesloten.

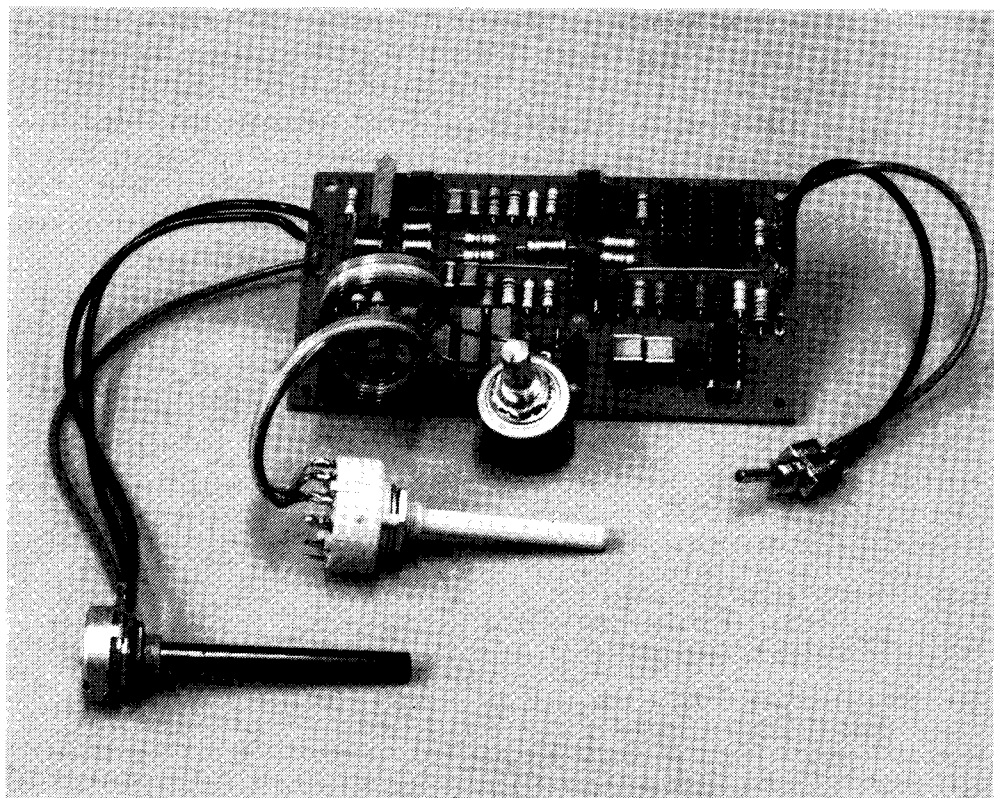
Het RTTY-programma is zodanig opgezet dat hiermee 5 bits baudot en 7 bits ASCII ontvangen en gedecodeerd kunnen worden. Bovendien zijn er zeven verschillende baud-rates (overdrachtssnelheid). De ontvangen data kan eerst in een file-buffer worden opgeslagen. Als de buffer vol is wordt een foutmelding door de computer gegeven. De inhoud van de buffer kan natuurlijk worden uitgelezen.

Een andere bijzonderheid van het programma is de "auto letter mode". Bij de baudotkode wil het nogal eens voorkomen dat het letter-teken verloren gaat. Dat heeft dan tot gevolg dat de navolgende letters als cijfers worden gedecodeerd. In de "auto letter mode" wordt dit zoveel mogelijk vermeden doordat na de ontvangst van een spatieteken door de computer altijd naar de "letter mode" wordt teruggeschakeld.

In figuur 4 is een vereenvoudigd stroomschema van het programma gegeven. Na het starten van het programma op adres 4000 geeft de computer de mogelijke baud-rates, zoals in tabel 2 te zien is. Daarna stelt hij enkele vragen waarop met Y (yes), N (no) of een getal tussen 0 en 5 (voor de baud-rate) geantwoord moet worden. Bij de ontvangst van een ASCII-uitzending moet men de vraag "ASCII-RECEIVER" met Y beantwoorden. Als op deze plaats N wordt ingegeven schakelt de dekoder over op baudot-kode.

Nadat alle vragen zijn beantwoord is de computer klaar voor de ontvangst van een serieel signaal via PB7. Dit wordt kenbaar gemaakt doordat hij het teken ": " geeft. Als men de eerste vraag ("DO YOU LIKE TO CHANGE IT?") beantwoordt met het indrukken van de RETURN-toets wordt de startprocedure verkort. Het dekoderprogramma start dan meteen in de baudot-mode, met een baud-rate van 50 baud. Het teken ": " vervalt dan.

Nadat het programma is gestart kan men de gekozen "mode" veranderen door het bedienen van de BREAK-toets op het ASCII-keyboard. Een reset is mogelijk door middel van de NMI-toets.



RTTY- dekoder

Telex-
ontvangst
met de
computer

In de vorige Elektuur hebben we uitgebreid aandacht besteed aan het dekoderen van morsetekens met behulp van de junior computer en de Z80A-kaart. Nu is het de beurt aan de telex-ontvangst via de radio. Benodigd voor deze toepassing is een uitgebreide junior computer, die met behulp van een eenvoudige interface en een bijbehorend programma in een EPROM wordt "omgetoverd" tot een echte RTTY-dekoder. En ziedaar... het onverstaanbare RTTY-gebrabbel op de korte golf wordt als duidelijke tekst op het beeldscherm zichtbaar gemaakt.

Het principe van de overdracht en de dekodering is bij radio-telex eigenlijk hetzelfde als bij morse. Door het onderbreken van het zendersignaal, ook wel gesleutelde draaggolf genoemd, wordt de digitaal gekodeerde informatie uitgezonden. Bij morse maakt men gebruik van de morse-code en bij telex van een 5 bits code, ook wel baudotcode genoemd. Deze baudotcode (officieel CCITT-code nr. 2) kunt u vinden op Infokaart 84 in het meinumner van Elektuur.

Behalve het verschil in code is er ook nog een verschil in de overdracht tussen morse en

In de laatste jaren is de belangstelling voor het radio-telex-verkeer sterk toegenomen. Dat komt onder andere door het feit dat steeds meer hobbyisten thuis een computer hebben staan. En een computer leent zich heel goed voor deze telex-ontvangst. De Elektuur-computer bij uitstek, de junior computer, is ook uitstekend als RTTY-dekoder te gebruiken, met behulp van een kleine interface-schakeling en een speciaal voor deze toepassing geschreven programma.

telex. Bij morse wordt één draaggolf uitgezonden, die in het ritme van de punten en strepen wordt onderbroken (dus uitgeschakeld). Bij telex gaat het uitzenden op een andere manier. Daar maakt men gebruik van twee draaggolven, waarbij de ene draaggolf de "enen" vertegenwoordigt en de andere draaggolf de "nullen". Je zou dat kunnen voorstellen door twee zenders die naast elkaar staan en elk op een andere frekwentie uitzenden. Tijdens een "1" wordt de ene zender ingeschakeld en tijdens een "0" de andere zender. In werkelijkheid gebeurt het uitzenden natuurlijk met één enkele zender. Deze methode noemt men FSK (Frequency Shift Keying).

Bij radio-telex wordt een logische één gewoonlijk "mark" genoemd en een logische nul "space". Het frekwentieverschil tussen "mark" en "space" bij het uitzenden noemt

men "shift". Het frekventieverschil is vrij klein.
 Bij telex-ontvangst komen uit de luidspreker van de ontvanger twee verschillende tonen, die de logische nullen (space) de eenen (mark) vertegenwoordigen. Als de beide tonen tegelijk hoorbaar zijn is er sprake van een fout in de overdracht.

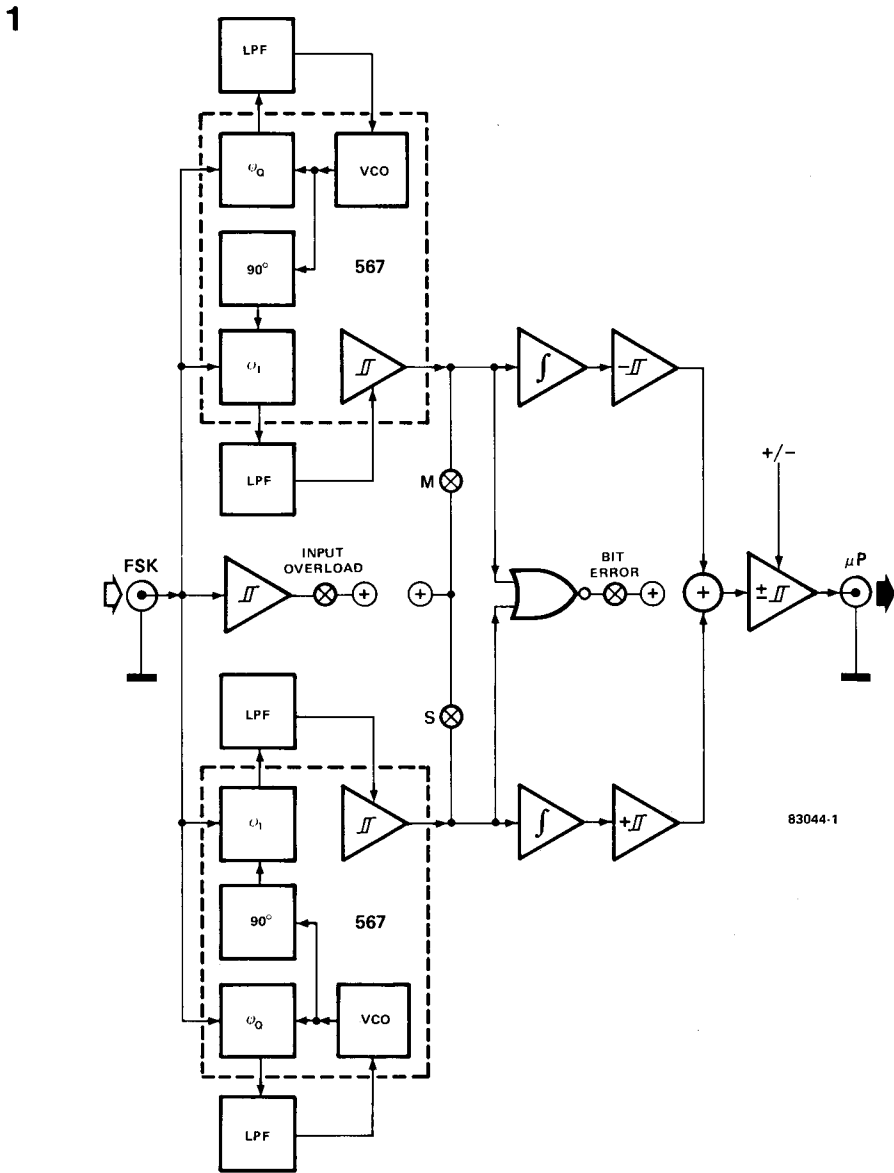
De RTTY-interface

Het "gepiep" uit de kortegolf-ontvanger kan niet zonder meer door de computer worden verwerkt. Er is nog een interface-schakeling nodig die de tonen uit de ontvanger omzet in blokgolven die de computer wel kan verwerken. Die interface moet duidelijk onderscheid kunnen maken tussen de twee ontvangers frekwenties. Daartoe gebruiken we een toondekoder die gevolgd wordt door een integrator en een schmitt-trigger. Dat alles is twee maal aanwezig, namelijk voor beide frekwenties. In figuur 1 is het blokschema van deze interface afgebeeld. Het schema van de complete RTTY-interface is afgebeeld in figuur 2.
 Aan de ingang van de interface bevindt zich potmeter P7 voor de nivo-instelling. Daar-

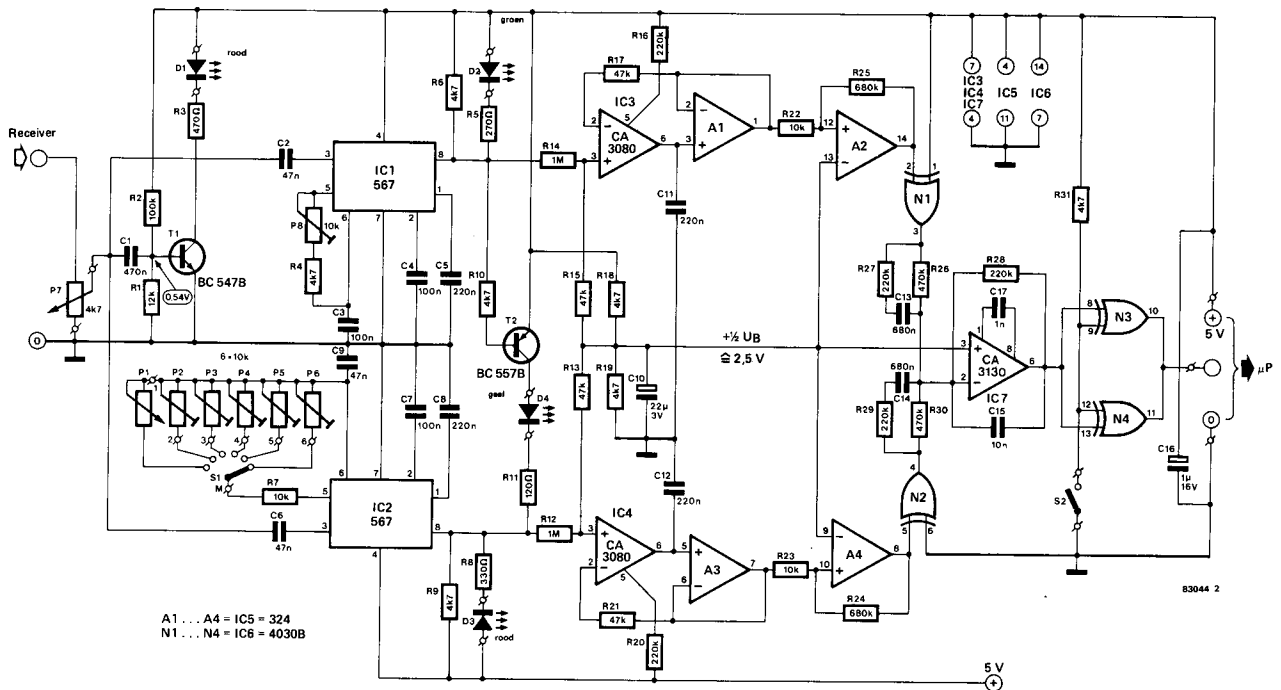
Tabel 1. Meest gebruikte frekwenties en shift-waarden bij RTTY-verkeer.

Signaal	Instellen met:	Frekwentie (Hz)	Frekwentie-shift (Hz)
mark	P8	1275	0
space 1	P1	var.	var.
space 2	P2	1445	170
space 3	P3	1575	300
space 4	P4	1700	425
space 5	P5	2125	850
space 6	P6	2275	1000

achter zit een nivo-indikator die rond T1 en D1 is opgebouwd. Hetingangssignaal wordt verder naar de ingangen van de twee toondekoders IC1 en IC2 gevoerd. IC1 kan met behulp van P8 op een bepaalde frekwentie worden afgeregeld, terwijl IC2 door middel van S1 en P1...P6 op maar liefst zes verschillende frekwenties kan worden afgeregeld. Dit is gedaan om de dekodeer te kunnen omschakelen op de bij een telex-uitzending gebruikte frekwenties. Met P8 wordt IC1 ingesteld op een frekwentie van 1275 Hz. De frekwentie waarop IC2 moet reageren is gelijk aan 1275 Hz plus de "shift". Men kan ook 1275 Hz minus



Figuur 1. Blokschema van de RTTY-interface. De schakeling bestaat uit twee toondekoders die elk worden gevolgd door een integrator en een schmitt-trigger voor het onderdrukken van stoormapen. Aan de uitgang bevindt zich een sommeer-schakeling die zo is opgezet dat ze ook nog een bruikbaar signaal levert als een van de twee signalen (mark of space) uitvalt. Het "bit error"-gedeelte geeft een indicatie van de kwaliteit van het ontvangen signaal.



Figuur 2. Het schema van de RTTY-interface, die de ontvangen telex-signalen omzet in TTL-kompatible blokspanningen die door de junior computer verwerkt kunnen worden.

"shift" aanhouden, dat maakt voor de werking verder niets uit. In tabel 1 zijn de verschillende bij het telex-verkeer gebruikelijke shift-waarden gegeven, zodat men aan de hand van die gegevens de potmeters P1 ... P6 kan instellen.

Aan de uitgangen van de toondekoders bevinden zich drie LED's. D2 licht op bij de ontvangst van een mark-sigitaal en D3 licht op bij de ontvangst van een space-sigitaal. D4 signaleert een foutconditie. Dat wil zeggen dat D4 oplicht als er tegelijkertijd een mark- en een space-sigitaal aanwezig zijn. Bij goede ontvangst zal deze LED niet of nauwelijks oplichten, terwijl sterk oplichten wijst op een slechte ontvangst, een storing of een foutieve instelling van de interface. Na elke toondekoder volgt een integrator die rond een OTA is opgebouwd (IC3 en IC4), een buffertrap (A1 en A3) en een schmitt-trigger (A2 en A4). Het integrator- en triggergedeelte is identiek aan dat van de morse-interface uit het meinumner van Elektuur.

De daaropvolgende schakeling zorgt voor het "optellen" van de mark- en space-signalen op een heel bijzondere manier. Hierbij wordt gebruik gemaakt van het feit dat altijd één van de twee signalen (mark of space) aanwezig moet zijn. Valt één van de twee signalen uit, dan kan het andere sigitaal nog worden gebruikt voor het verkrijgen van de telex-informatie. Als mark "1" is moet space "0" zijn, en omgekeerd. Het mark-sigitaal wordt daarom eerst door N1 geïnverteerd, terwijl het space-sigitaal via N2 onveranderd wordt doorgegeven. De uitgangssignalen van N1 en N2 worden via R26, R30 en opamp IC7 gesommeerd. Op die wijze levert de uitgang van de opamp toch de juiste informatie als het space- of mark-sigitaal uitvalt. Kondensator C15 zorgt er voor dat IC7 ook nog een integrerende werking heeft, zodat eventuele op deze plaats nog aanwezige stoorimpulsen onderdrukt worden.

De poorten N3 en N4 dienen voor het verbeteren van de flanksteilheid van de door IC7 geleverde blokspanningen. Tevens worden deze twee poorten gebruikt om het uitgangssigitaal naar behoefte te kunnen inverteren (via schakelaar S2). Als S2 is geopend inverteren de twee poorten; is S2 gesloten, dan werken ze alleen maar als buffer. De stand van S2 is afhankelijk van het telex-station waarnaar men luistert.

De praktijk

Als de interface-schakeling is opgebouwd op de in figuur 3 afgebeelde print kan ze worden afgeregeld. Daarvoor zijn een generator en een frekwentiemeter nodig. Sluit de generator en de frekwentiemeter beide aan op de ingang van de interface-schakeling. Zet P7 in de middenstand, draai de generator op een frekwentie van 1275 Hz (aflezen op de frekwentiemeter) en regel de uitgangsspanning van de generator vervolgens omhoog totdat D1 juist oplicht. Voor P8 moet nu een bepaald instelbereik te vinden zijn waarbij LED D2 oplicht. De juiste stand van P8 is het midden van het gevonden bereik. Men kan ook het ingangssigitaal steeds wat kleiner maken en daarbij proberen nog een stand van P8 te vinden waarbij de LED oplicht. Tenslotte zal er nog één stand overblijven: dat is de optimale instelling. Daarna komt de afregeling van de tweede toondekoder. Regel P2 ... P6 volgens de hiervoor beschreven procedure af, waarbij de generator wordt ingesteld op de space-frekwenties die in tabel 1 vermeld zijn (space-frekwentie = frekwentie + shift). Zonder generator en frekwentiemeter is een afregeling vrij moeilijk. Men kan dan het beste P7 in de middenstand zetten en de juiste shift-frekwentie van elke uitzending experimenteel bepalen door P1 te verdraaien (S1 in stand 1).

Dan kan de interface op de laagfrekwent-



uitgang (bijv. opname-uitgang) van de kortegolf-ontvanger worden aangesloten. Zoek een telex-zender op en stel P7 zo in dat LED D1 net oplicht. Stem de ontvanger

nu zo af dat LED D2 zo fel mogelijk oplicht in het ritme van het ontvangen RTTY-sigitaal. Kies vervolgens de juiste shift met S1. Als de shift niet bekend is probeert

Onderdelenlijst

Weerstanden:

R1 = 12 k
R2 = 100 k
R3 = 470 Ω
R4, R5, R9, R10, R18, R19,
R31 = 4k7
R5 = 270 Ω
R7, R22, R23 = 10 k
R8 = 330 Ω
R11 = 120 Ω
R12, R14 = 1 M
R13, R15, R17, R21 = 47 k
R16, R20,
R27 ... R29 = 220 k
R24, R25 = 680 k
R26, R30 = 470 k
P1 = 10 k tienslagen-
potmeter
P2 ... P6, P8 = 10 k
tienslagen-instelpotmeter
P7 = 4k7 (5 k) potmeter

Kondensatoren:

C1 = 470 n
C2, C6, C9 = 47 n
C3, C4, C7 = 100 n
C5, C8, C11, C12 = 220 n
C10 = 22 μ /3 V
C13, C14 = 680 n
C15 = 10 n
C16 = 1 μ /6 V
C17 = 1 n

Halfgeleiders:

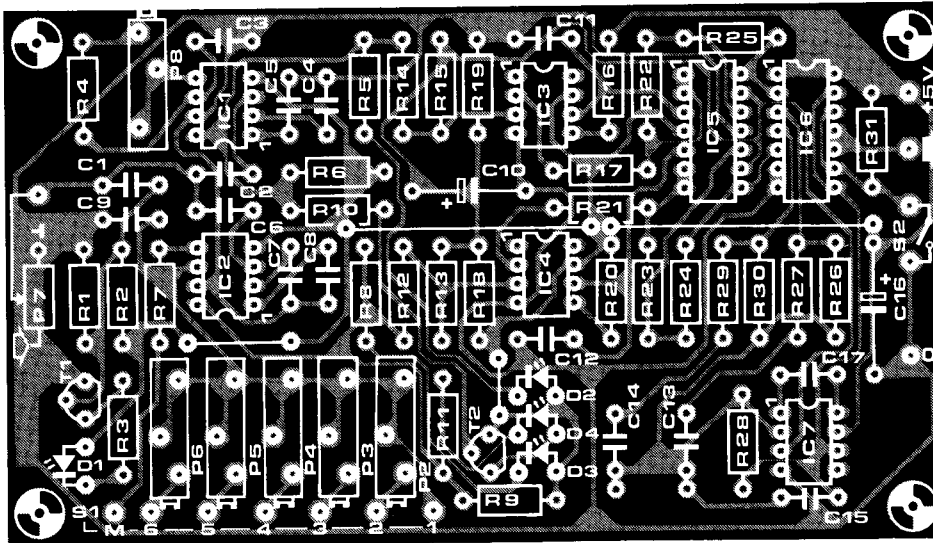
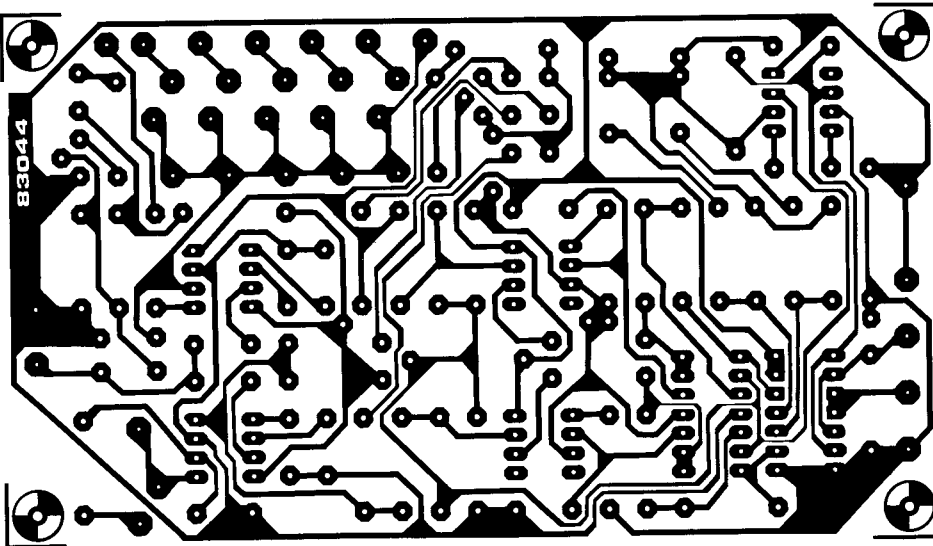
D1, D3 = LED rood
D2 = LED groen
D4 = LED geel
T1 = BC 547B
T2 = BC 557B
IC1, IC2 = LM 567
IC3, IC4 = CA 3080
IC5 = LM 324
IC6 = 4030B
IC7 = CA 3130

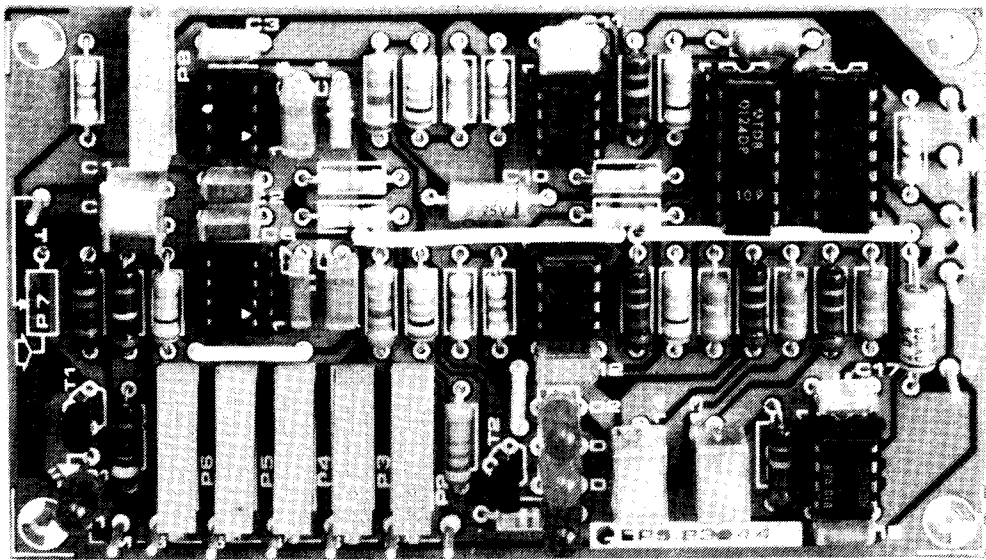
Diversen:

S1 = draaischakelaar
1 moederkontakt,
6 standen
S2 = enkelpolige schakelaar

**Figuur 3. Layout en
komponentenopstelling
van de print voor de
interface-schakeling. Met
de instelpotmeters worden
de toondekoders afgeregeld
op de diverse bij het telex-
verkeer gebruikelijke
frekwenties.**

3





Tabel 6. De hexdump van het RTTY-decodeer-programma.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
800	4C	BD	42	42	01	80	00	00	00	00	00	00	00	00	05	00
810	4E	FF	7F	01	00	00	00	00	00	3F	35	00	00	F0	55	20
820	4E	88	44	15	34	10	27	83	23	4C	3C	40	8D	63	23	48
830	C9	00	F0	06	20	F8	44	20	43	23	68	60	AD	06	40	2D
840	80	FA	F0	F8	A2	FF	9A	20	EB	41	4C	19	44	AD	14	40
850	48	A9	00	8D	14	40	20	1B	FE	20	2C	40	AA	68	8D	14
860	48	8A	60	68	85	F8	68	85	F9	AD	14	40	AA	A9	00	8D
870	14	40	E6	F8	D0	02	E6	F9	A0	00	B1	F8	C9	00	F0	06
880	20	2C	40	4C	72	40	68	8D	14	40	A5	F9	48	A5	F8	48
890	60	20	63	40	0D	0A	42	41	55	44	4F	54	20	00	60	20
8A0	63	40	0D	0A	41	53	43	49	49	20	00	60	20	63	40	52
8B0	45	43	45	49	56	45	52	0D	0A	60	20	63	40	20	42	
8C0	41	55	44	0D	0A	60	20	63	40	34	35	2E	34	35	00	
8D0	60	20	63	40	35	30	00	60	20	63	40	35	37	00	60	20
8E0	63	40	37	35	00	60	20	63	40	31	30	30	00	60	20	63
8F0	40	31	31	30	00	60	20	63	40	4E	4F	20	00	60	20	63
900	40	0D	0A	52	45	43	45	49	56	45	44	20	43	48	41	52
910	41	43	54	45	52	53	20	41	52	45	00	0A	53	54	4F	52
920	45	44	20	49	4E	20	42	55	46	46	45	52	0D	0A	00	60
930	20	63	40	41	55	54	4F	20	4C	45	54	54	54	52	20	4D
940	4F	44	45	00	60	20	63	40	0D	0A	00	60	AD	14	40	8D
950	16	40	A9	00	8D	14	40	AD	0F	40	AE	10	40	C5	FA	D0
960	1D	E4	FB	D0	19	20	63	40	0D	0A	46	49	4C	45	20	45
970	4D	50	54	59	0D	0A	00	AD	16	40	8D	14	40	60	85	FA
980	86	FB	20	45	41	A9	00	8D	1B	40	A0	00	B1	FA	C9	FF
990	F0	E5	48	20	2C	40	20	E4	41	38	A5	FA	ED	11	40	A5
9A0	FB	ED	12	40	68	90	16	20	63	40	0D	0A	46	49	4C	45
9B0	20	42	56	45	52	46	4C	4F	57	0D	0A	60	C9	0D	0D	
9C0	C9	20	2C	40	EE	1B	40	AE	1B	40	EC	1C	40	90	89	20
9D0	4D	40	C9	45	F0	A1	D0	AD	A9	0A	D1	FA	F0	AD	20	2C
9E0	40	4C	8A	41	E6	FA	D0	02	E6	FB	60	20	45	41	20	45
9F0	41	AD	03	40	C9	42	D0	06	20	91	40	4C	01	42	20	9F
A00	40	20	AC	40	AD	04	0A	D0	06	20	C7	40	4C	3A	42	C9
A10	81	D0	06	20	D1	40	4C	3A	42	C9	02	D0	06	20	D8	40
A20	4C	3A	42	C9	03	D0	06	20	DF	40	4C	3A	42	C9	04	D0
A30	06	20	E6	40	4C	3A	42	20	EE	40	20	BB	40	AD	14	40
A40	F0	0C	20	FE	40	AD	17	40	F0	20	30	41	60	20	F6	
A50	40	4C	42	42	20	F6	40	4C	4A	42	C9	F1	F0	19	C9	1B
A60	F0	0F	AE	13	40	F0	05	AA	BD	7D	42	60	AA	BD	9D	42
A70	60	A9	00	8D	13	40	60	8D	13	40	A9	00	60	00	45	0A
A80	41	20	53	49	55	0D	40	52	4A	4E	46	43	4B	54	3A	4C
A90	57	48	59	50	51	4F	42	47	00	4D	58	56	00	00	55	0A
AA0	2D	20	27	38	37	0D	24	34	07	2C	00	3A	28	35	2B	29
AB0	32	00	36	30	31	39	3F	00	00	40	58	56	00	20	A0	44
AC0	A2	FF	9A	A9	05	0D	0E	40	A9	42	8D	03	40	A2	01	8E
AD0	04	40	8E	13	40	CA	8E	14	40	8E	17	40	20	63	40	0D
AE0	04	45	4C	45	46	54	4F	52	20	52	54	54	59	21	44	45
AF0	43	4F	44	45	52	0D	0A	42	41	55	44	52	41	54	45	
B00	3A	0D	0A	30	3D	34	35	2E	34	35	2D	42	41	55	44	
B10	0D	0A	31	3D	35	30	0D	0A	32	3D	35	37	0D	0A	33	3D
B20	37	35	0D	0A	34	3D	31	30	3D	0A	35	3D	31	31	30	
B30	0D	0A	44	4F	20	59	4F	52	20	4C	49	4B	45	20	54	4F
B40	20	63	48	41	4E	47	45	20	49	54	20	3C	59	2F	4E	
B50	3E	00	20	4D	40	C9	59	D0	5A	20	63	40	0D	0A	53	45

B60	4C	45	43	54	20	54	48	45	20	42	41	55	44	52	41	54
B70	45	3A	20	00	20	4D	40	C9	30	90	DE	C9	36	B0	DA	29
B80	0F	8D	04	40	20	63	40	0D	0A	41	53	43	49	49	20	52
B90	45	43	45	49	56	45	52	3F	20	3C	59	2F	4E	3E	00	20
BA0	4D	40	C9	59	D0	10	A9	07	8D	0E	40	A9	41	8D	03	40
BB0	4C	C0	43	4C	74	44	A9	05	8D	0E	40	A9	42	8D	03	40
BC0	20	63	40	0D	0A	46	49	4C	45	20	42	55	46	46	45	52
BD0	3F	20	3C	59	2F	4E	3E	00	20	4D	40	C9	59	D0	06	8D
BE0	14	40	4C	EA	43	29	00	8D	14	40	20	63	40	0D	0A	41
BF0	55	54	4F	20	4C	45	54	54	45	52	20	4D	4F	44	45	3F
C00	20	3C	59	2F	4E	3E	00	20	4D	40	C9	59	D0	06	8D	17
C10	40	4C	19	44	29	00	8D	17	40	20	63	40	0D	0A	40	49
C20	53	54	20	54	48	45	20	46	49	4C	45	20	42	55	46	46
C30	45	52	3F	20	3C	59	2F	4E	3E	00	20	4D	40	C9	59	D0
C40	03	20	4C	41	20	63	40	0D	0A	3A	3A	0D	0A	C0	4C	74
C50	44	A9	00	8D	18	40	20	75	45	AE	03	40	E0	42	0D	03
C60	20	5A	42	AE	18	40	EC	19	40	B0	09	EC	1A	40	90	11
C70	C9	20	D0	0D	A9	0D	20	2C	40	A9	0A	20	2C	40	4C	51
C80	44	C9	0A	F0	0C	C9	0D	D0	02	A9	20	20	2C	40	EE	18
C90	40	AE	17	40	F0	07	C9	20	D0	03	8D	13	40	4C	56	44
CA0	78	AD	0F	40	AE	10	40	85	FA	86	FB	A0	FF	98	C8	91
CB0	FA	20	E4	41	AE	11	40	E4	FA	D0	F4	AE	12	40	E4	FB
CC0	D0	ED	91	FA	8C	15	40	8C	14	40	AD	0F	40	AE	10	40
CD0	85	FA	86	FB	A9	3C	A2	40	8D	7C	FA	8E	7D	FA	A9	BD
CE0	A2	42	8D	7A	FA	8E	7B	FA	A9	07	8D	82	FA	A9	3F	8D
CF0	83	FA	A9	00	8D	81	FA	60	AC	15	40	D0	1E	C9	00	F0
D00	1A	AC	14	40	F0	15	A4	FA	CC	11	40	D0	07	A4	FB	CC
D10	12	40	F0	08	A0	00	91	FA	20	E4	41	60	A0	01	8C	15
D20	40	20	A7	41	20	63	40	0D	0A	42	55	46	46	45	52	20
D30	4F	55	54	50	55	54	3F	20	3C	59	2F	4E	3E	00	20	4D
D40	40	C9	59	D0	09	20	45	41	20	45	41	20	4C	41	20	45
D50	40	0D	0A	53	59	53	54	45	40	20	52	45	53	45	54	3F
D60	20	3C	59	2F	4E	3E	00	20	4D	40	C9	59	D0	01	60	4C
D70	BD	42	4C	3C	40	8A	48	98	48	AE	0E	40	AD	06	40	2D
D80	80	FA	F0	EE	AD	05	40	2D	B2	FA	D0	F0	20	4A	46	20
D90	CC	45	B0	EE	82	20	4A	46	A9	40	2D	D0	F8	F0	F9	20
DA0	45	20	CC	45	6E	0D	40	CA	D0	F4	38	A9	08	ED	0E	40
DB0	AA	4E	0D	40	CA	D0	FA	20	1D	46	A9	40	2D	D0	F8	F0
DC0	F9	68	AE	68	AA	AD	0D	40	8D	63	23	60	AD	05	40	2D
DD0	82	FA	F0	1D	EE	08	40	D0	03	EE	0C	40	A9	40	2D	0D
DE0	F8	F0	E9	38	AD	0B	40	ED	09	40	AD	0C	40	ED	0A	40
DF0	60	EE	09	40	D0	03	EE	0A	40	4C	DC	45	AD	04	00	0A
E00	A8	B9	1D	40	8D	04	FB	C8	B9	1D	40	8D	05	F8	A9	00
E10	8D	09	40	8D	0B	40	8D	0A	8D	0C	40	8D	0C	40	60	AD
E20	0A	A8	B9	1D	40	8D	07	40	C8	B9	1D	40	8D	08	40	4E
E30	08	40	6E	07	40	4E	08	40	6E	07	40	AD	07	40	8D	04
E40	F8	AD	08	40	8D	05	F8	4C	VE	46	AD	40	0A	0A	A8	B9
E50	1D	40	8D	07	40	C8	B9	1D	40	8D	08	40	4C	35	46	A2
E60	08	40	00	B1	00	91	02	88	D0	F9	E6	01	E6	03	CA	0D
E70	F0	60	A9	00	A2	E8	85	00	86	01	A9	00	A2	40	85	E2
E80	86	03	20	5F	EE	4C	00	FC	A9	08	A2	00	85	00	86	01
E90	A9	00	A2	40	85	02	86	03	20	9E	0E	4C	1D	1C	A2	8E
EA0	A0	00	B1	00	91	02	88	D0	F9	E6	01	E6	03	CA	D0	F0
EB0	60															

modellen overeenkomen: dat zijn de 8 data-bits (vergeet niet dat de Centronics-interface een parallel-interface is!), een "data geldig" signaal (DATA STROBE), de BUSY-lijn (die in actieve toestand aangeeft dat de printer geen data kan ontvangen) en de SELECT-lijn waarmee de printer aangeeft of hij wel of niet de data van de computer wil ontvangen. Uiteraard zijn de onmisbare massa-lijn en de afschermingslijnen niet vergeten. De sekundaire signalen zijn Paper Out of Paper Empty, Fault, Input Prime, . . . onnodig om ze allemaal op te noemen omdat ze hier toch niet gebruikt worden.

Op de interface-kaart van de Junior Computer bevindt zich de VIA 6522 waarvan de twee poorten zonder voorafgaande ingreep gebruikt kunnen worden. Zoals te zien in het schema blijven de lijnen PA0 . . . PA5 (voor de verwerking van speciale signalen bijvoorbeeld) ongebruikt. Voor de rest is de zaak vrij eenvoudig: de 8 databits (PB0 . . . PB7) geven niet alleen de komplette ASCII-karakterset (7 bits), maar geven ook toegang tot speciale printer-karakters (Griekse, Japanse, Cyrillische en Franse tekens, graphics etc.). De lijn CB2 op de 6522 geeft het "data-valid"-signaal. Dit is een impuls van een mikroseconde of minder, die wordt uitgezonden binnen een mikroseconde na de stabilisatie van de databits op een poort B. Het uitzenden van die puls geschiedt automatisch en volgt na elke schrijfoperatie op poort B. De pen PA7 is een ingang die het BUSY-signaal ontvangt. Via ingang PB6 wordt infor-

matie naar de computer verstuurd voor wat betreft de modus waarin de printer verkeert; wanneer SELECT logisch laag is, drukt de printer niets af. Ter informatie zij nog gezegd, dat het logische nivo van de SELECT-lijn ook beïnvloed wordt door de detector die het eind van het papier signaleert; wanneer het papier in de printer bijna op is "de-SELECTeert" deze zichzelf. Volgt nu alleen nog het aansluit-schema van de Centronics-konnector en een klein programmaatje, geschreven voor de DOS Junior, waarmee u het beste uit deze parallel-interface kunt halen. Daarbij valt op, dat de configuratie van de in- en uitgangspoorten bij het begin van elk karakter wordt hernieuwd. De ASCII-kode van het af te drukken teken moet niet in de akkumulator zijn op het moment

dat de routine begint, maar de temp.-buffer AHOLD; de inhoud van de X-en Y-registers wordt namelijk niet "gesaved". De instructie LDA AHOLD, die te vinden is aan het einde van de routine PAROUT, is noodzakelijk bij de DOS-versie van de Junior Computer die deze routine aanroept om het af te drukken karakter uit de buffer AHOLD te halen. Voor de gebruiker van de DOS Junior Computer merken we op dat het niet voldoende is deze routine te plaatsen vanaf lokatie \$9E op adres \$2317 te vervangen door \$E1.

De I/O-instructie, beschreven in het gebruikershandboek van Ohio Scientific, laat een erg flexibele besturing van de printer toe; I0,08 alleen al is voldoende om de printer te activeren; I0,09 activeert de seriële en parallelle uitgang van het systeem. ■

CENTRONICS INTERFACE WITH VIA/6522

PB0 . . . PB7 : 8 BITS DATA OUTPUT
PA7 : BUSY INPUT
PA6 : SELECT INPUT
CB2 : DATA STROBE OUTPUT

24E2	A2	FF	PAROUT	LDXIM	\$FF	
24E4	BE	02	FB	STX	PBDD	PB IS OUTPUT
24E7	E8			INX		
24E8	BE	03	FB	STX	PADD	PA6 & PA7 ARE INPUTS
24EB	A2	A0		LDXIM	\$A0	
24ED	BE	0C	FB	STX	PCR	CB2 OUTPUTS DATA STROBE
24F0	2C	0	FB BUSY	BIT	PAD	GET SELECT BIT
24F3	10	08		BPL	RTS	EXIT IF SELECT IS LOW
24F5	70	F9		BVS	BUSY	WAIT FOR BUSY LINE LOW
24F7	AD	63	23	LDA	AHOLD	GET CHARACTER
24FA	BD	00	FB	STA	PBD	PRINT IT WITH STROBE
24FD	60		RTS	RTS		

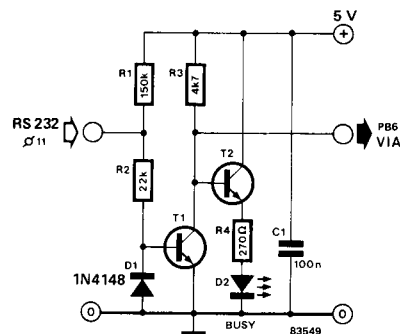
49

W. Schaaij

busy-indikator voor junior

Op de junior computer kan vrij eenvoudig een busy-indikator voor een printer worden aangesloten. Alles wat daarvoor nodig is, is een kleine schakeling die wordt toegevoegd aan de printer-interface. Verder moet een kleine modifikatie worden aangebracht in de PM-software. Het schema van de schakeling is bijzonder eenvoudig. Het busy-sig-naal van de printer is beschikbaar aan pen 11 van de RS 232-aansluiting.

Het gedeelte rond T1 zorgt voor de nivo-aanpassing van RS 232 naar TTL. Bij een positieve spanning aan de ingang ("0") gaat T1 geleiden, zodat op aansluiting PB6 van de VIA een logische nul komt te staan. Bij een negatieve spanning ("1") spert T1, zodat PB6 +5 V krijgt toegevoerd via R3. Diode D1 zorgt er voor dat de basis-emitterspanning van T1 in dat geval wordt begrensd tot -0,6 V. Via T2 gaat de LED branden als de printer een busy-sig-naal (negatieve) spanning geeft. Weerstand R1 zorgt er voor dat de LED niet brandt en ook geen busy-sig-naal aan de junior wordt gegeven als niets is aange-



T1, T2 = BC 547B

sloten op de RS 232-konnektor. In de PM-EPR0M moeten de volgende plaatsen veranderd/toegevoegd worden:

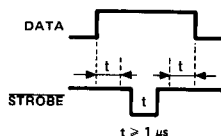
133A 20 F4 14 JSR **BUSY** (veranderen)

14F4 BUSY	AD 00	18	LDA PBDVIA (toevoegen)
14F7	29 40		ANDIM \$40
14F9	D0 F9		PB6 VIA = 0?
14FB	AD 82	1A	LDA PBDPIA
14FE	60		RTS

50

centronics-interface

Ondanks de fantastische eigenschappen van het beeldscherm is de oudere terminal met papier toch nog niet in onbruik geraakt. Vorig jaar nog maakten wij onze lezers blij met een ontwerpidee voor een koppeling van een printer aan de Elektterminal (zie Elektuur nr. 225/226 op pagina 7-72). We komen hier op dit onder-

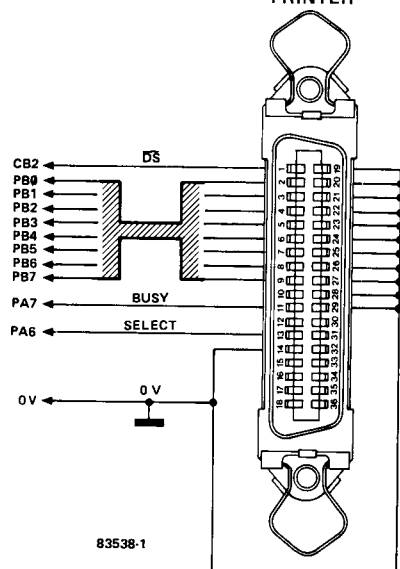


werp terug met een vollediger en flexibeler ontwerp, zeggend beter aangepast aan de Centronics-standaard, waaraan de huidige printers voldoen.

Er wordt nogal makkelijk gesproken van *Centronics-norm*, maar als men de gebruikershandboeken van de diverse fabrikanten (Epson, Seikosha, NEC... om de bekendste maar op te noemen) raadpleegt, stelt men vast dat elk model zijn eigenaardigheden heeft, van afwijkingen in de pinning tot het gewoonweg ontbreken van bepaalde signalen. Daarom stellen wij voor pas blind vertrouwen te schenken in de konnektor van de printer na een grondige controle van de aansluitpennen. Voor onze

VIA 6522

PRINTER



interface hebben we slechts de onmisbare signalen genomen waarvan de aansluitpennen bij alle gangbare

58

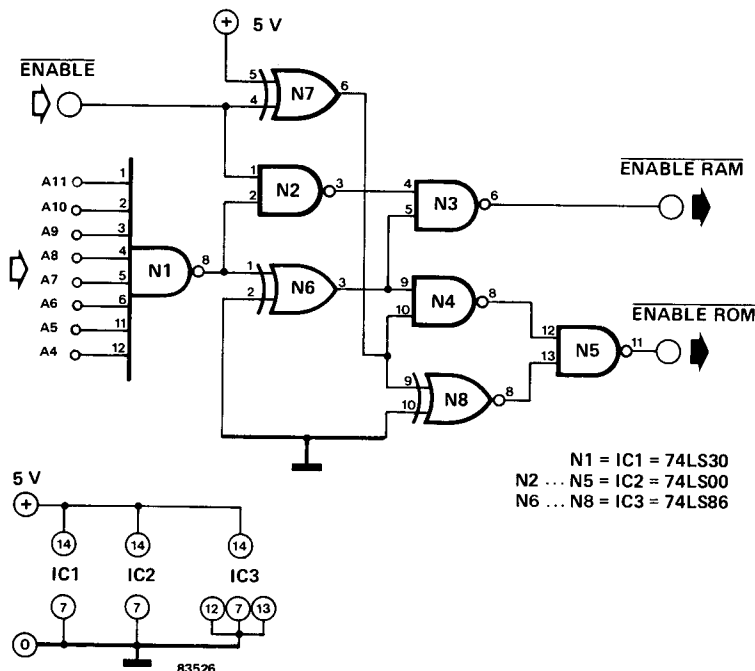
N. Humphreys

vektor- aansturing voor junior

Met de hier afgebeelde schakeling is het mogelijk bij de junior computer de noodzakelijke vektordata uit de standaard-EPROM te lezen zonder dat daarvoor een flink gedeelte van het adresbereik hoeft te worden ingeleverd en/of een extra ROM nodig is. De vektoren voor NMI, RES en IRQ staan op de adressen FFFA...FFFF. Bij de gewone oplossing voor het halen van die vektoren uit de standaard-EPROM, zoals in het junior computer boek deel 3 is beschreven, wordt het hele geheugenbereik van F000...FFFF voor 6 bytes "opgeofferd". Dat is 4 K geheugenruimte! Bij de hier beschreven oplossing worden voor het lezen van die 6 bytes slechts 16 bytes gereserveerd. De schakeling is natuurlijk alleen nodig als men RAM wil leggen in het bovenste geheugenbereik van F000...FFFF; bij gebruik van de mini-EPROM-kaart,

bijvoorbeeld (Elektuur april '82) is er sowieso geen opoffer-probleem. De schakeling bestaat uit slechts acht poorten. Ze genereert uit het "gewone" enable-sigitaal voor het hoogste 4 K-bereik twee aparte enable-signalen. Bij de adressen F000...FFFF wordt een nieuw ENABLE RAM-sigitaal gegeven, terwijl bij de adressen FFF0...FFFF geen ENABLE RAM wordt gegeven, maar een ENABLE EPROM-sigitaal naar de standaard-EPROM van de junior wordt gestuurd. De schakeling kan worden bijgebouwd op de betreffende RAM-

kaart, het gaat hier om slechts drie IC's. De oude ENABLE-lijn voor het bereik F000...FFFF van de adresdekoder op de RAM-kaart (bij de dynamische RAM-kaart is dat uitgang F van IC11) wordt op de "toevoeg"-schakeling aangesloten en de acht ingangen van N1 worden verbonden met de adreslijnen A4...A11. Naar de RAM-kaart gaat dan een nieuw ENABLE RAM-sigitaal (bij de dynamische RAM-kaart wordt die lijn dus aan één van de punten V, W, X of Y gelegd). De ENABLE ROM-lijn moet worden verbonden met aansluiting K7 (pen 14a) van de konnektor.



De CRTC levert tegelijkertijd aan de 2732 de rij-adressen (RA0...RA3) van elk "om te zetten" karakter, zodat voor ieder karakter telkens één rij dots wordt uitgelezen voor de videolijn die op dat moment geschreven wordt op het scherm. Door IC20 wordt de dot-informatie omgezet van een parallel naar een serieel signaal. Het toegepaste schuifregister is een synchroon type om timing-fouten bij de gebruikte hoge frekwenties uit te sluiten. Het klok-signaal voor IC20 wordt via N19 en N20 direkt afgeleid van de oscillator. De seriële dot-informatie is beschikbaar aan uitgang Y van het IC. De video-mengtrap bestaat uit N34, N31, N22 en het gedeelte rond T1 en T2. Hier wordt het Y-signaal van IC20 gemengd met de door de CRTC geleverde lijn- en rastersynchronisatiepulsen (pen 39 en 40 van IC11). Met de instelpotmeters P1 en P2 kan men de grootte van de sync-pulsen en de dot-amplitude instellen. Hierbij moet nog worden opgemerkt dat de twee potmeters elkaar beïnvloeden.

Twee belangrijke signalen van de CRTC moeten nog even apart worden besproken, namelijk DEN en CUR. Uitgang CUR geeft de plaats van de cursor op het scherm en uitgang DEN (display enable) geeft aan wanneer de CRTC in het "aktieve bereik" van het scherm zit (zie ook de beeldopbouw in het beschrijvende artikel). Dat laatste signaal is nodig om het scherm buiten het aktieve bereik helemaal zwart te houden. Die twee signalen moeten nu met het video-signaal worden gemengd (gebeurt via N34 en N31), maar dat kan niet direkt in verband met de tijd die verstrijkt tussen het geven van een adres aan de RAM en het verschijnen van de dot-informatie aan de uitgangen van de EPROM. Die vertragingstijd bedraagt verscheidene honderden nanosekonden. Dat zou tot gevolg hebben dat het cursor- en display-enable-signaal iets te vroeg zouden verschijnen ten opzichte van het dot-signaal. Om dit probleem te vermijden worden het DEN- en CUR-signaal door middel van de

flipflops FF1...FF4 twee hele karaktertijden vertraagd voordat ze aan het video-signaal worden toegevoegd.

Met de draadbruggen bij N33 kan men kiezen tussen een oplichtende of een donkere cursor op het scherm. Dit betekent tevens dat hiermee kan worden omgeschakeld van gewone naar "inverse" video, aangezien bij een zwarte cursor ook alle dot-signalen door N34 worden geïnverteerd. Bij draadbrug T krijgt men een gewoon beeld (zwarte achtergrond) en bij draadbrug S een geïnverteerd beeld (lichte achtergrond).

N15, N25, N28 en N29 vormen samen de Z80-interface. Deze poorten maken uit de door de Z80 geleverde signalen ekwivalenten van de door de 6502 geleverde signalen R/W en enable. In het geval van de Z80 moeten de draadbruggen U-V en X-Y worden gelegd. Met de draadbruggen U-W en X-Z is de kaart geschikt voor 6502 en de 6800-familie.

Opbouw

Het bouwen van de video-kaart zal voor de hobbyist die ook al andere computer-delen heeft gebouwd (bijvoorbeeld van de junior computer) beslist geen moeilijkheden geven. Vooral niet als men gebruik maakt van de in figuur 5 afgedrukte print. In dit artikel is alleen de componentenopdruk gegeven; de koper-layout van de (dubbelzijdige) print kan men vinden op de print-layout-pagina's in deze Elektuur.

Het verdient natuurlijk aanbeveling voor alle IC's een goede kwaliteit voetjes te gebruiken. Zeker bij IC3 en IC20 is dat belangrijk, eventueel kan men deze IC's rechtstreeks op de print solderen. Voor T1 is in de onderdelenlijst een BSX 20 aangegeven, maar dat mag ook een BC 547B zijn. Vergeet tenslotte niet de drie draadbruggen (twee voor het type processor en een voor gewoon of geïnverteerd beeld) op de juiste plaats te leggen. Vergeet ook niet de verbindingen voor de adresdekodering.

Als men een kristal in de oscillator toepast kunnen L1, C1 en C2 worden weggelaten op de print. Voor het gebruik van de VDU-kaart met de uitgebreide junior zijn drie EPROM's nodig: een 2732 die de karaktergenerator bevat en twee 2716's (TMV en PMV) met de video-routines. Die laatste twee vervangen de EPROM's met TM en PM(E). In de nieuwe EPROM's zit tevens de TM- en PM-software, zodat er kwa bediening van de junior niets verandert. Voor de DOS-junior zijn twee nieuwe EPROM's nodig: een 2732 met de karaktergenerator en een 2716 met de video-routines (DOSVT). De 2716 komt op het voetje voor IC5 op de interface-kaart. Bovendien is bij de DOS-junior nog een CMOS-RAM 6116 nodig die geplaatst moet worden in de voet voor IC4 op de interface-kaart. Verder moeten er bij de DOS-junior nog enige verbindingen op de interface-kaart worden gelegd:

- pen 18 van IC4 moet worden doorverbonden met pen 20
- leg de volgende draadbruggen: M-J, G-I, I'-G', J'-L', O'-M'

In de ESS zijn de programma's als set opgenomen. Voor de uitgebreide junior dient

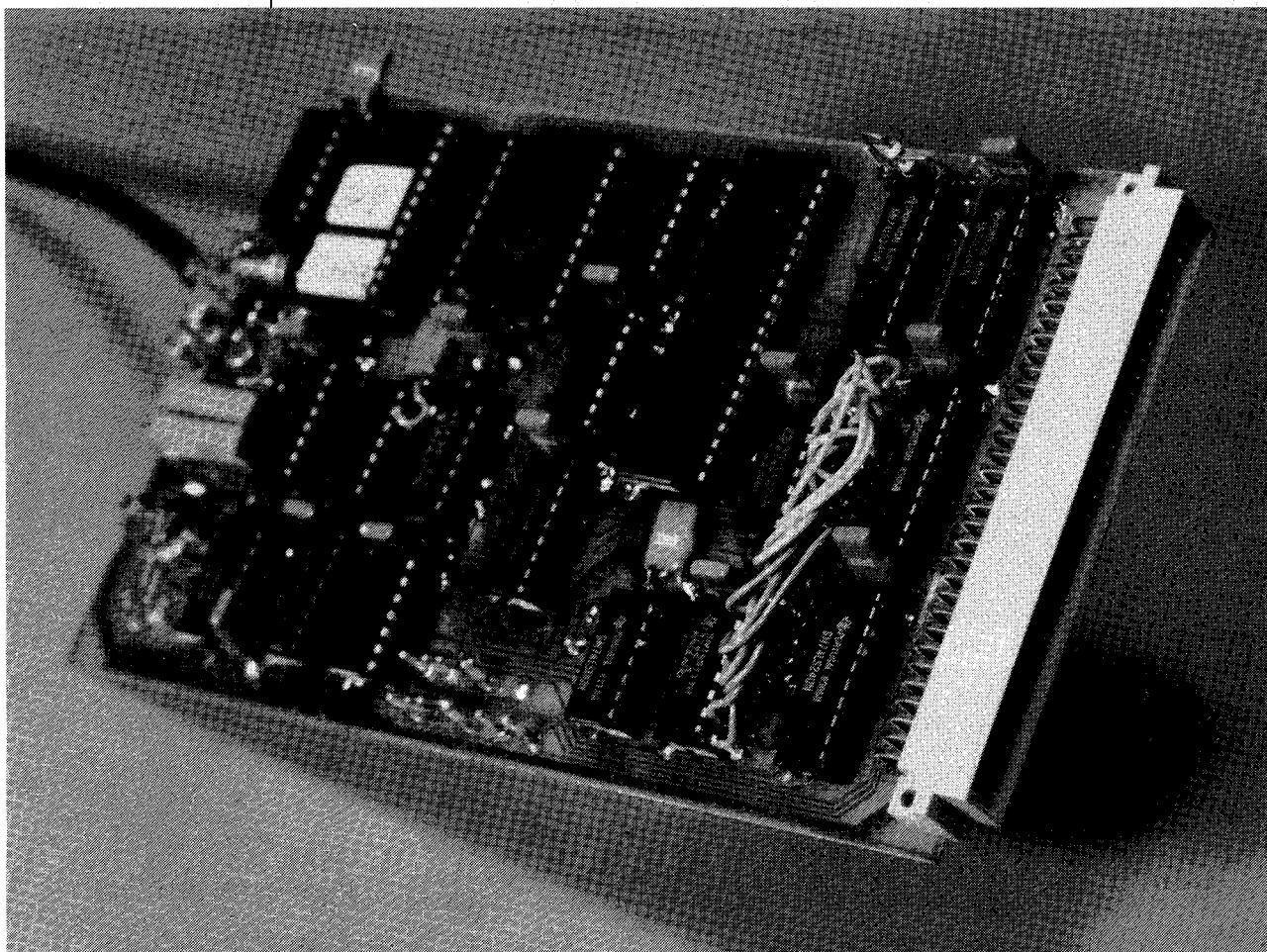
Ok
RUN*BEXEC#

OS-650 Tutorial disk five - Sept. 16, 1981

- 1 > Directory
- 2 > Create a new file
- 3 > Change a file name
- 4 > Delete file from diskette
- 5 > Create blank data diskette
- 6 > Create data diskette with files
- 7 > Create buffer space for data files
- 8 > Single or dual disk drive copier
- 9 > Enter OS-650 system

Type the number of your selection
and depress RETURN ?

VDU - kaart



in samenwerking
met H. Vermeulen

video voor
computers

Eindelijk is het zover: we hebben een nieuwe video-kaart! De hier voorgestelde VDU-kaart vormt niet zo maar een nieuwerwetse vervanger voor de oude en alom bekende Elekterminal, nee deze kaart is duidelijk berekend op de moderne computer-mogelijkheden. 24 regels van 80 karakters kunnen op het scherm worden gezet, graphics zijn aanwezig en zo zijn er nog meer mogelijkheden. Talloze junior-computer-bezitters hebben waarschijnlijk al maanden zitten wachten om hun computersysteem met een "eigen" video-kaart te kunnen uitrusten. Maar deze VDU-kaart is niet alleen geschikt voor de junior. Ook andere processors kunnen worden aangesloten, zoals de 6800-familie en de Z 80.

Wie het bijbehorende artikel "karakters op het scherm" in deze Elektuur heeft gelezen, weet wat hij hier kan verwachten: de beschrijving van de opzet en bouw van een moderne, goed doordachte VDU-kaart. Wie de algemene werking van een VDU-kaart met monitor nog niet kent, wordt verwezen naar het zojuist genoemde beschrijvende artikel. Hier wordt alleen het schema van de VDU-kaart besproken. Tevens moeten we nog vertellen wat de verdere mogelijkheden van deze kaart zijn. We beginnen met het laatste.

VDU-kaart . . . en terminal?

In dit artikel wordt alleen de VDU-kaart besproken als zelfstandig geheel. In deze vorm kan de VDU-kaart zonder meer worden aangesloten op de uitbreidingsbus van de junior computer. Het enige dat verder nog nodig is, is een 2716-EPROM met een VDU-output-programma dat in de plaats komt van het printer-monitor-programma. In figuur 1 is in een tekening aangegeven uit welke belangrijke componenten de VDU-kaart is opgebouwd. Ten eerste is daar de

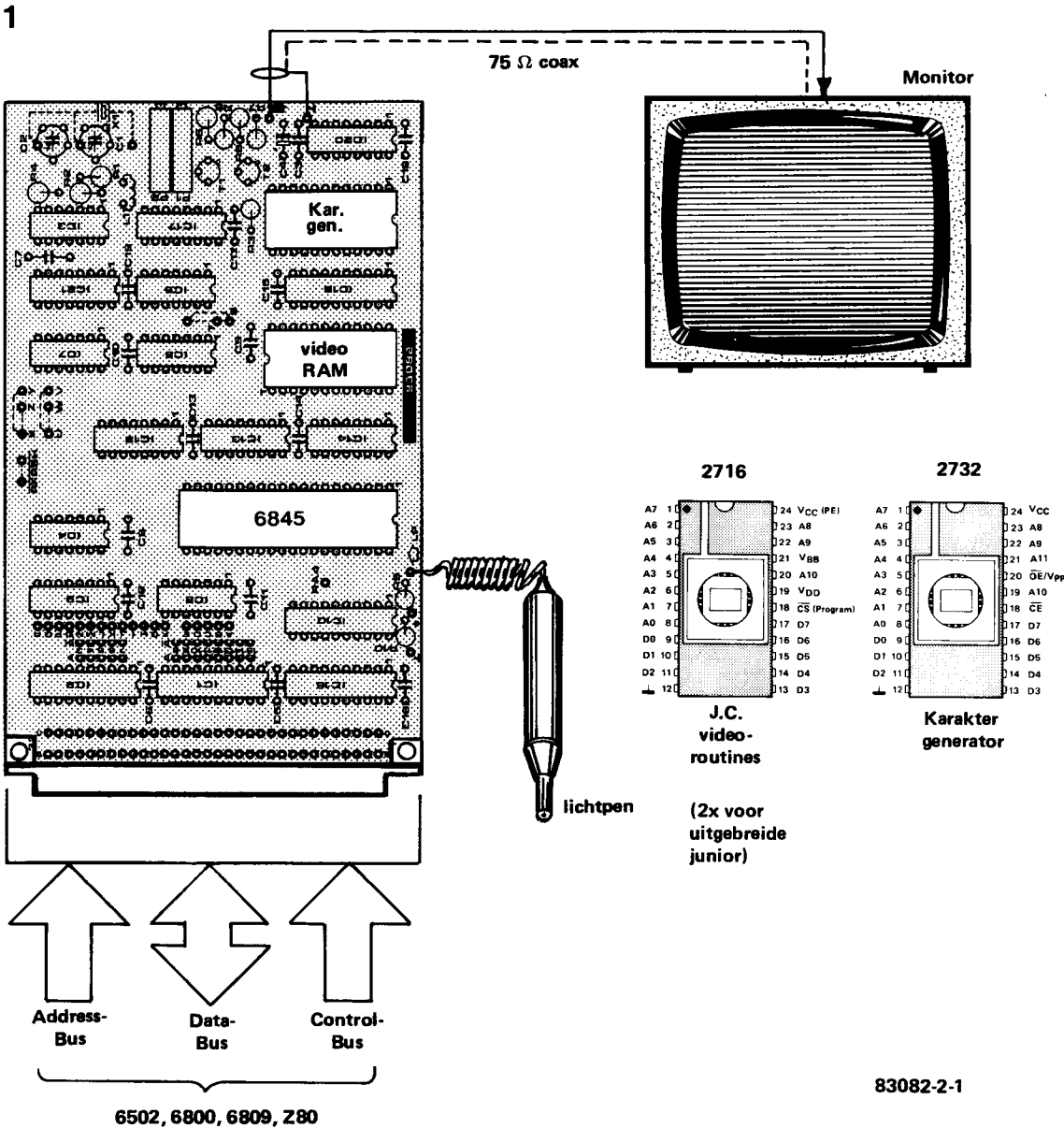
VDU-kaart zelf, met het controller-IC 6845, een 2 K-video-RAM 6116 en de karakter-generator (zie ook het blokschema in het beschrijvende artikel). De karaktergenerator bestaat uit een 2732-EPROM waarin alle ASCII- en grafische tekens zijn opgeslagen in de betreffende punt-matrix-vorm (tussen haakjes: graphics zijn mogelijk door middel van poke-kommando's, maar daar komen we nog een andere keer op terug). Via een 75 Ω -video-uitgang kan de kaart worden verbonden met een monitor. Op de kaart is een aansluiting voor een lichtpen aanwezig, hoewel daarbij moet worden opgemerkt dat in de basis-uitvoering van de VDU-kaart nog geen software voor lichtpen-gebruik aanwezig is. Deze kan later wel vrij eenvoudig worden toegevoegd. In de tekening is ook nog de 2716 te zien die de video-routines voor de junior bevat.

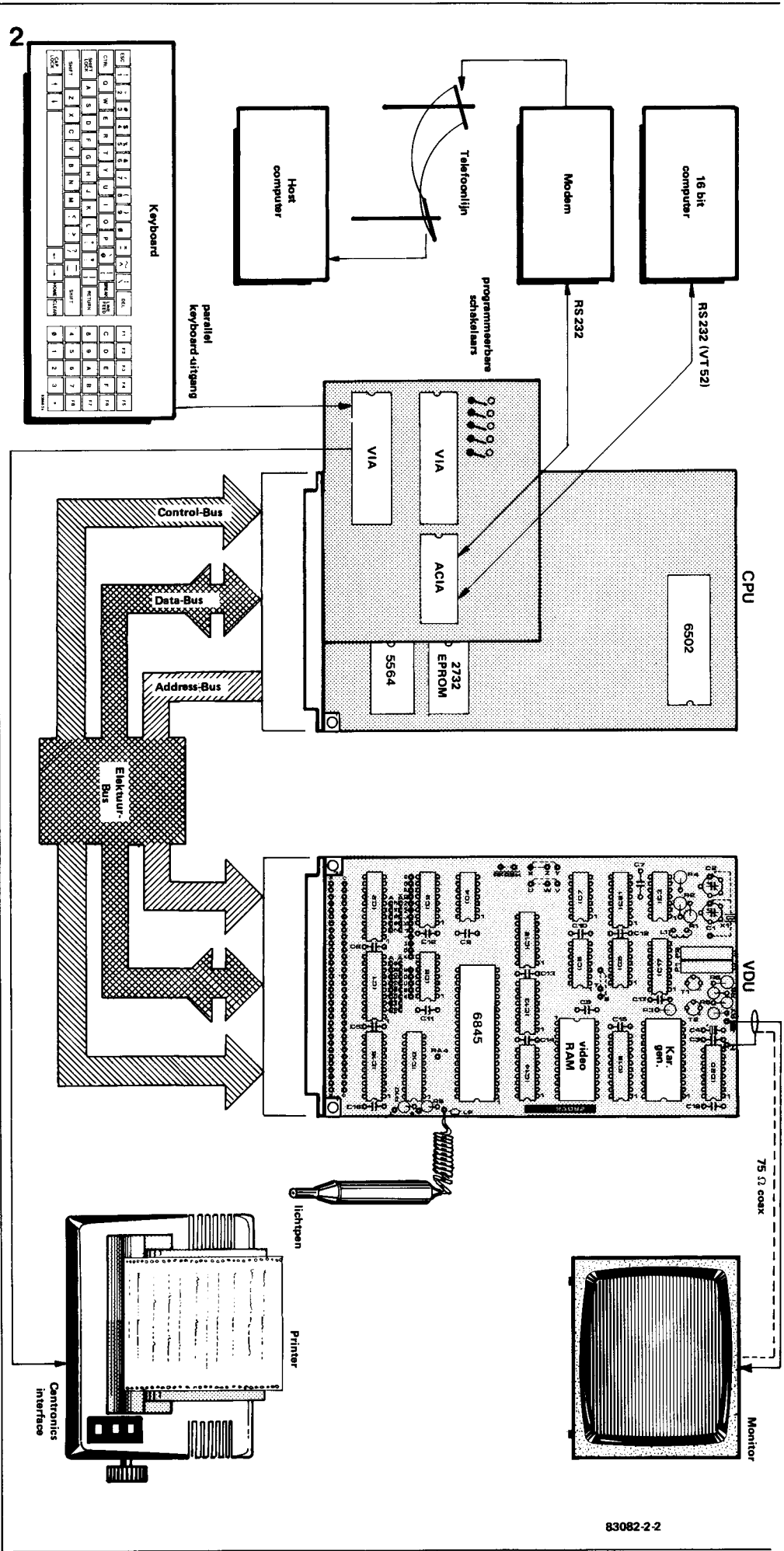
Het standaard-formaat op het scherm is 24 regels van 80 karakters. In verband met de hiervoor benodigde bandbreedte is dan ook echt een monitor nodig of een TV met een video-ingang (dus niet de gewone antenne-ingang).

Op de kaart is een interface voor de Z 80-processor aanwezig, zodat de VDU-kaart ook geschikt is voor dit type processor. Ook andere 6502-computers kunnen worden aangesloten, evenals de 6800-familie. Door de aanwezigheid van een volledige adresdekodering op de kaart kan deze worden toegepast bij praktisch elke moderne computer waarin een van de genoemde processor-typen wordt toegepast: AIM-65, SYM, VC-20, VC-64, enzovoorts. Men dient er dan wel rekening mee te houden dat de VDU-kaart gebruik maakt van de Elektuur-bus. Bij andere systemen moet men dus zelf de verbindingen "uitknobbelen" en zelf voor de video-routines zorgen.

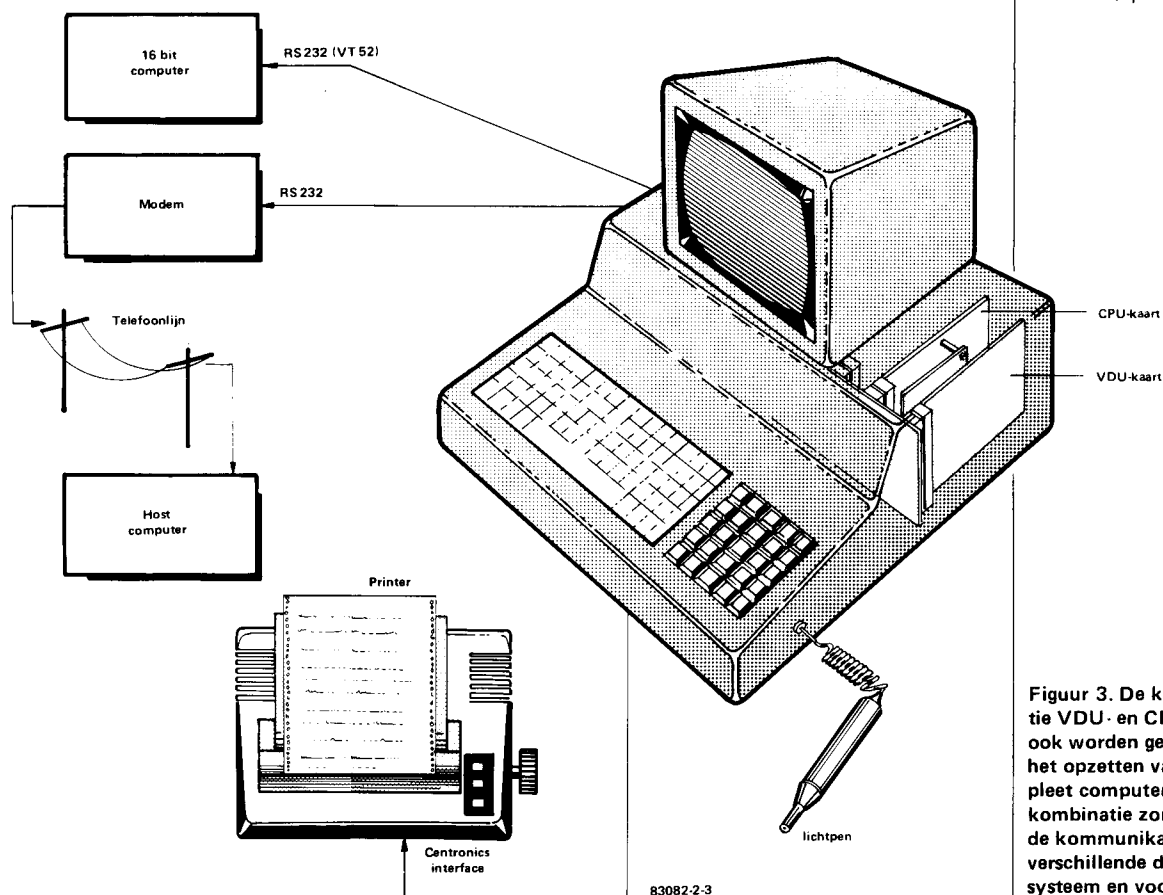
Het door de VDU-kaart geproduceerde "composite video"-signaal kan worden aangepast aan elke monitor. Zowel de synchronisatie-impuls als het contrast zijn instelbaar. Het hele beeld kan ook geïnverteerd worden, zodat zwarte karakters op een oplichtende achtergrond zichtbaar zijn. De cursor kan naar keuze knippen of gewoon oplichten. De VDU-kaart kan worden gebruikt met een vrijlopende oscillator, maar

Figuur 1. Een "grove schets" van de samenstelling van de VDU-kaart, waarop de belangrijkste delen van deze kaart zijn aangegeven.





Figuur 2. De toekomstverwachting: een universele terminal, bestaande uit een VDU-kaart en een CPU-kaart. Alle andere apparaten (computer, modem, printer, toetsenbord etc.) kunnen op deze terminal worden aangesloten.



Figuur 3. De combinatie VDU- en CPU-kaart kan ook worden gebruikt voor het opzetten van een compleet computersysteem. De combinatie zorgt dan voor de communicatie tussen de verschillende delen van het systeem en voor het zichtbaar maken van de gewenste informatie op het scherm.

het is ook mogelijk in de tijdbasis een 15MHz-kristal toe te passen. In dat geval is er werkelijk sprake van een "rotsvast" beeld. Een bijzonderheid van de VDU-kaart is nog, dat de hele timing op de kaart werkt met synchroon geklokte TTL-schakelingen. Dat heeft het voordeel dat er bij deze hoge frequenties geen timing-fouten kunnen optreden.

Deze opsomming laat zien dat zowel de opzet als de mogelijkheden van de VDU-kaart bepaald niet mis zijn. Maar er komt nog meer.

Als aanvulling op de VDU-kaart brengt Elektuur binnenkort nog een speciaal hiervoor ontworpen CPU-kaart. Deze twee kaarten vormen samen het hart van een universele terminal met RS232-interface en VT52-protocol, waarop werkelijk van alles kan worden aangesloten. Figuur 2 geeft hiervan een indruk. De terminal kan in de eerste plaats natuurlijk worden verbonden met elke computer die een RS232-interface bezit. De CPU-kaart bevat een 6502-microprocessor, 2 VIA's (Versatile Interface Adapter), een ACIA (Asynchronous Communications Interface Adapter), een EPROM en een RAM. Met behulp van een stel doorverbindingen op de print kan men overdrachtsformaat, snelheid, aantal start- en stopbits en het soort en aantal controlebits aanpassen aan de op de terminal aangesloten computer. Bij het beeldformaat kan worden gekozen uit zo'n acht verschillende formaten.

Een complete terminal bestaat uit een VDU-

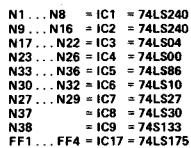
kaart, een CPU-kaart, een monitor en een toetsenbord. Via een modem kan de terminal bijvoorbeeld over een telefoonlijn communiceren met een computer op een andere plaats op deze wereldbol, maar het is ook mogelijk om dankzij het VT52-protocol een 16-bits computer op de terminal aan te sluiten. Natuurlijk is ook voorzien in een aansluiting voor een printer. Het is verder nog mogelijk om de CPU-kaart en de VDU-kaart samen te gebruiken als basis voor een compleet computersysteem, zoals figuur 3 laat zien. Het voorbeeld toont een samenstelling met een "16-bitter", maar in principe kan dat elk type zijn.

De terminal-software is opgeslagen in een 2716-EPROM op de CPU-kaart; op deze kaart is plaats voor maximaal 8 Kbytes RAM en 16 Kbytes ROM.

Mogelijkheden genoeg dus met deze twee-kaart-kombinatie. Verdere bijzonderheden bewaren we tot het betreffende artikel, maar nu weet men tenminste al welke kanten men allemaal uit kan met de VDU-kaart. En er zijn nog meer mogelijkheden dan de genoemde te bedenken...

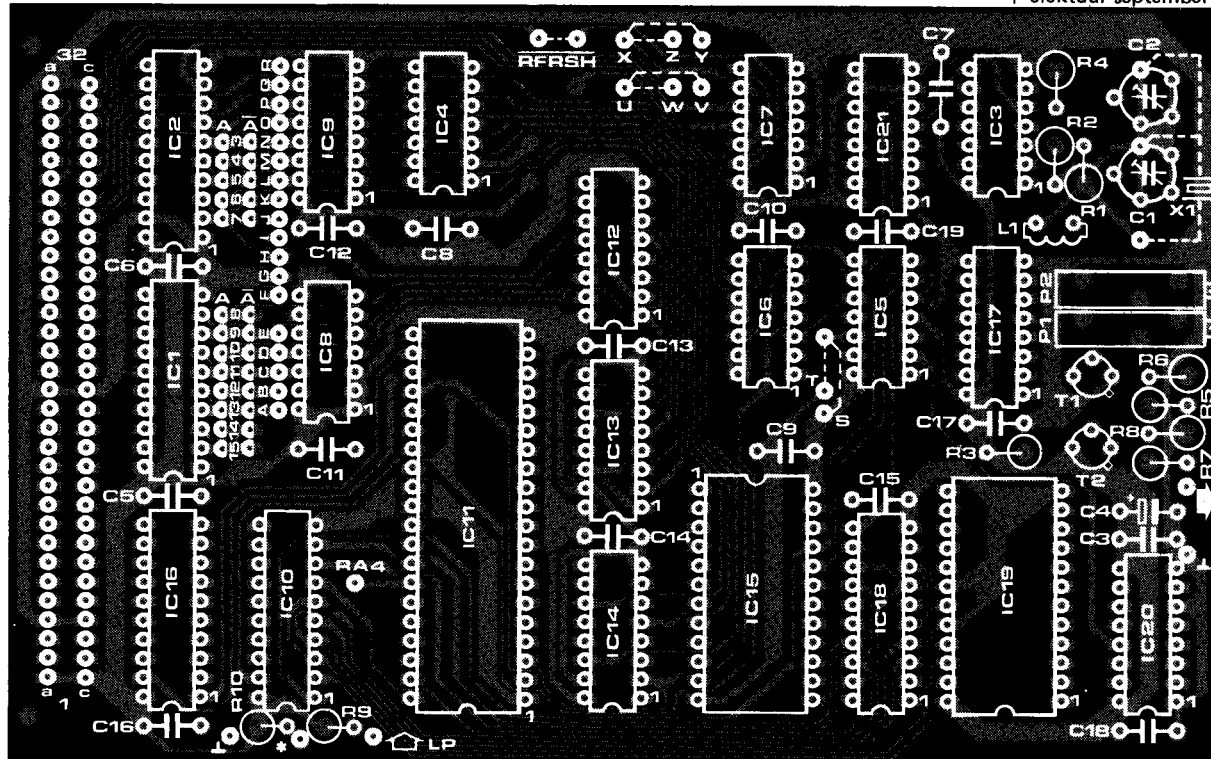
De VDU-kaart op de korrel

In figuur 4 zien we dan (eindelijk) het schema van de VDU-kaart. Links is de systeem-bus getekend. De adreslijnen A0...A10 zijn verbonden met de B-ingangen van de 2-naar-1-multiplexers IC12...IC14. Verder worden de adreslijnen A3...A15 geïnverteerd door N1...N13. Een vol-



Figuur 5. De komponentenopstelling op de print voor de VDU-kaart. De koper-layouts van de dubbelzijdige print zijn afgedrukt op de Print-layout-pagina's achter in deze Elektuur.





ledige adresdekodering is zo mogelijk omdat de adressen zowel normaal als geïnverteerd aanwezig zijn aan de aansluitpunten A3...A15 en $\bar{A}3$... $\bar{A}15$. Poort N37 zorgt voor de adresdekodering van de video-RAM en N38 voor de dekodering van de CRTC IC11. Bij de poorten is aangegeven welke verbindingen moeten worden gelegd bij gebruik van de junior computer. Video-RAM ligt in dat geval in het bereik D000...D7FF en de CRTC ligt in het bereik D800...D80F.

Als de video-RAM IC15 vanuit de systeembus wordt geadresseerd door de microprocessor, dan geeft N37 een chip-select-sigitaal. Daardoor worden de adres-ingangen van de 6116 via de A-ingangen van de multiplexers IC12...IC14 verbonden met de adres-bus van het systeem (select-ingangen van multiplexers logisch nul). Tegelijkertijd wordt databusbuffer IC10 ge-enabled via N14 en N24. Het logische nivo van de R/W-lijn (aansluiting 29c op de konektor) zorgt er voor dat IC15 via N32 en N23 op lezen of schrijven wordt geschakeld (WE-ingang). Tevens wordt zo de doorlaatrichting van de databusbuffer IC10 ingesteld.

Als vanuit de systeembus de CRTC IC11 wordt geadresseerd geeft N38 een logische nul aan de CS-ingang. Vanuit de systeembus heeft men (die "men" is dan de processor) dan toegang tot de interne registers van de 6845. Via N16 en N26 wordt de doorlaatrichting van databusbuffer IC16 ingesteld. IC16 is trouwens alleen nodig als men later een lichtpen bij de VDU-kaart wil toepassen. Is dat niet het geval, dan wordt alleen maar van de bus naar de CRTC geschreven en is IC16 overbodig. Door middel van 8 draadbruggen kan men dan de 6845 verbinden met de datalijn.

De "adresdekoder" N37 zorgt tevens voor het resetten van de flipflops FF1...FF4,

zodat men geen "onregelmatigheden" op het scherm ziet als de processor toegang tot de video-RAM heeft.

De hele timing van de VDU-kaart wordt afgeleid van een rond N17 en N18 opgebouwde oscillator. Deze levert de zogenaamde dot-frequentie die bij het hier gebruikte schermformaat 15 MHz bedraagt. De spoel is toegevoegd om de oscillator bij zo'n frequentie nog betrouwbaar te laten werken. Het is ook mogelijk een 15 MHz-kristal in de oscillator toe te passen. Het door de oscillator geleverde signaal wordt door IC21 door 8 gedeeld. Dit IC is een synchrone teller die door middel van poort N30 bij de tellerstand 7 wordt gereset. Doordat deze reset pas wordt "verwerkt" in het IC bij de volgende klokpuls telt het IC op deze manier toch in werkelijkheid tot acht. Uitgang QC levert de zogenaamde karakterfrequentie voor de controller. In het ritme van dit signaal telt de CRTC continu van 000 tot 7FF (het hele adresbereik van de video-RAM). De adres-uitgangen MA0...MA10 van IC11 zijn (als de processor geen toegang heeft tot de video-RAM) via de multiplexers verbonden met de adres-ingangen van de 6116, zodat voortdurend alle RAM-adressen doorlopen worden. De RAM geeft dan continue data uit die in de latch IC18 wordt opgeslagen. Deze latch is nodig om de video-RAM tijd te geven tot alle data aan de uitgangen stabiel is. De latch wordt pas geklokt als de data van de RAM stabiel is. De uitgangs-data van de latch kan dan gebruikt worden terwijl ondertussen een nieuw adres aan de RAM wordt aangeboden. De RAM heeft dan tijd om nieuwe data uit te geven. De klokpulsen voor de latch worden geleverd door N21.

De informatie in de latch dient nu als adres voor de karaktergenerator IC19.

Onderdelenlijst

Weerstanden:

R1, R2 = 470 Ω
R3, R4, R8 = 100 Ω
R5, R6 = 4k7
R7 = 68 Ω
R9, R10 = 2k2

Kondensatoren:

C1 = 40 p trimmer
C2 = 10 p trimmer
C3, C5...C19 = 100 n
C4 = 1 μ /6 V

Halfgeleiders:

T1, T2 = BSX 20
IC1, IC2 = 74LS240
IC3 = 74LS04
IC4 = 74LS00
IC5 = 74LS86
IC6 = 74LS10
IC7 = 74LS27
IC8 = 74LS30
IC9 = 74LS133
C10, IC16 = 74LS245
IC11 = 6545 of 6845
IC12, IC13, IC14 = 74LS157
IC15 = 6116
IC17 = 74LS175
IC18 = 74LS273
IC19 = 2732
IC20 = 74LS166
IC21 = 74LS163

Diversen:

X1 = Xtal 15 MHz (bij een display-konfiguratie van 80 x 24 karakters; bij gebruik van kristal vervallen C1, C2 en L1)
L1 = 4 μ 7
Konektor volgens DIN 41612, 64-polig male, A- en C-rijen bezet

men dus één 2732 en twee 2716's te sturen (ESS 522) en voor de DOS-junior een 2732 en één 2716 (ESS 521).

Bij de uitgebreide junior kan het systeem na het inschakelen van de voedingsspanning worden gestart door het indrukken van de Reset-toets.

Bij de DOS-junior moeten er nog enkele dingen op diskette worden aangepast. Daarvoor hebben we een op de junior aangepaste V3.3-diskette nodig met een Elekterminal of een andere serieel I/O-apparaat. Via Utility 8 wordt nu eerst een kopie gemaakt van deze diskette. Deze kopie wordt dan in drive A gestoken. Vervolgens volgt men de procedure die in tabel 1 is gegeven. Daarna gaat men verder op het toetsenbord van de junior:

<RST>

<AD> A200

<DA>

waarna de data uit tabel 2 wordt ingegeven (dit is de modifikatie van de bootstrap). Dan gaan we als volgt verder:

<AD> A311

<DA> FFEF (video output 1)

FFEF (video output 2)

A2FE (serial output 1)

EIF3 (Centronics output 1)

Hierna kunnen we verder gaan met tabel 3. Als dat is beëindigd hebben we een nieuwe, op de VDU-kaart aangepaste V3.3-diskette.

Tabel 2.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A200 :	A9	01	8D	5E	26	20	BC	26	A9	2A	85	FF	20	54	27	86
A210 :	FE	20	67	29	20	79	2E	A0	BF	20	EC	22	F0	03	88	D0
A220 :	F8	8C	00	23	A2	01	8E	C6	2A	4C	41	22	EA	EA	EA	EA
A230 :	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA
A240 :	EA	A9	00	8D	F7	EF	8D	D2	EF	20	35	F4	20	30	F3	20
A250 :	61	27	20	73	2D	0D	0A	0A	2A	44	4F	53	20	4A	55	4E
A260 :	49	4F	52	20	43	4F	4D	50	55	54	45	52	20	20	56	32
A270 :	2E	30	2A	0D	0A	0A	43	4F	50	59	52	49	47	48	54	20
A280 :	42	59	20	45	4C	45	4B	54	4F	52	00	A9	2E	8D	7C	FA
A290 :	A9	FF	8D	7D	FA	A9	00	8D	7A	FA	A9	FC	8D	7B	FA	4C
A2A0 :	E6	2A														

Het is trouwens de bedoeling dat binnenkort in de Paperware-service uitgebreide informatie voor geïnteresseerden verkrijgbaar zal zijn, waarin dieper wordt ingegaan op de werking van de CRTC en de bijbehorende video-software.

De schakeling heeft alleen maar een voedingsspanning van 5 V nodig. De stroomopname is ongeveer 450 mA.

Bij het instellen van P1 en P2 wordt uitgegaan van de middenstand van deze potmeters. Daarna zoekt men een instelling waarbij een duidelijk beeld op het scherm wordt verkregen. Bij een TV moet het contrast helemaal terug worden gedraaid, aangezien dan de bandbreedte meestal het grootst is. Met de trimmers C1 en C2 kan men de frekwentie zodanig instellen dat het beeld goed stil blijft staan op het scherm en niet meer "trekt". Bij gebruik van een 15 MHz-kristal is deze laatste instelling natuurlijk overbodig.

Tabel 1.

A*CA 0200 = 39,1

A*CA 2000 = 39,2

A*GO 0200

— DISKETTE UTILITIES —

SELECT ONE:

1) COMPAR

2) TRACK 0 READ/WRITE

? 2

— TRACK ZERO READ/WRITE UTILITY —

COMMANDS:

Rnnnn — READ INTO LOCATION nnnn

Wnnnn/gggg,p — WRITE FROM nnnn FOR p PAGES

WITH gggg AS THE LOAD VECTOR

E — EXIT TO OS-65D

COMMAND? RA200

— TRACK ZERO READ/WRITE UTILITY

COMMANDS:

Rnnnn — READ INTO LOCATION nnnn.

Wnnnn/gggg,p — WRITE FROM nnnn FOR p PAGES

WITH gggg AS THE LOAD VECTOR

E — EXIT TO OS-65D

COMMAND? E

A*CA AA00 = 01,1

A*

Tabel 1. Zo wordt track 0 van de floppy overgebracht naar het RAM-geheugen vanaf adres \$A200 en track 1 naar RAM vanaf adres \$AA00

Tabel 3.

A*GO 0200

— DISKETTE UTILITIES —

SELECT ONE:

1) COMPAR

2) TRACK 0 READ/WRITE

? 2

— TRACK ZERO READ/WRITE UTILITY —

COMMANDS:

Rnnnn — READ INTO LOCATION nnnn.

Wnnnn/gggg,p — WRITE FROM nnnn FOR p PAGES

WITH gggg AS THE LOAD VECTOR

E — EXIT TO OS-65D

COMMAND? WA200/2200,8

— TRACK ZERO READ/WRITE UTILITY

COMMANDS:

Rnnnn — READ INTO LOCATION nnnn.

Wnnnn/gggg,p — WRITE FROM nnnn FOR p PAGES

WITH gggg AS THE LOAD VECTOR

E — EXIT TO OS-65D

COMMAND? E

A*SA 01,1 = AA00/8

A*

Tabel 2. De data voor het modifieren van het bootstrapgedeelte.

Tabel 3. En zo wordt de gemodificeerde bootstrap weer teruggeschreven op de floppy disk.

junior
computer
speelt
zijn eigen lied

Een zingende uitlezing

A. Bricart

De frekwentie wordt afgeleid van de 1 MHz-kristalfrekwentie van het processorsysteem.



Uitgaande van de A van 440 Hz dient de periodeduur precies 2,28 ms te bedragen. Bij een symmetrische blok is de "1"-tijd dan 1,14 ms. De bij de junior beschikbare vertragsingslus (DELAY) geeft een vertraging van 14 μ s. Voor een A moet deze lus dus 81 keer worden doorlopen voor het verkrijgen van de halve periodeduur ($81 \times 14 \mu\text{s} = 1,134 \text{ ms}$). In hexadecimale vorm uitgedrukt bedraagt de "toonhoogte" van de A dan \$51.

Eenvoudige opzet

De eenvoud van het programma heeft tot gevolg dat de melodie alleen in pagina 0300 kan worden gezet. Dat houdt in dat de lengte van de melodie niet meer dan 127 noten mag zijn. Het tempo wordt bepaald door de inhoud van adres MULT (\$0002). De toonduur hangt af van de inhoud van de oneven adressen, waarbij nog moet worden opgemerkt dat de toonduur samenhangt met de toonhoogte. Tabel 1 geeft de kodes voor toon en toonduur. Als de processor in een even adres de waarde 00 tegenkomt, dan ziet hij dat als een pauze. De inhoud van het volgende adres bepaalt in dat geval de lengte van de pauze. Als het oneven adres de data 00 bevat wordt de melodie op dat punt afgebroken en begint de processor weer vooraan. In tabel 3 is tenslotte een muzikaal voorbeeld gegeven waarbij de junior zich van zijn beste kant laat horen.

Tabel 1

noot	Hz	toon- hoogte		toonduur	
E	1318.5	1B		84	47
D#	1244.5	1D		F9	7C
D	1174.6	1E		EB	76
C#	1108.7	20		DE	6F
C	1046.5	22		D1	68
B	988	24		C6	63
A#	932.3	26		BA	5D
A	880	29		B0	58
G#	830.6	2B		A6	53
G	784	2E		9D	4E
F#	740	30		94	4A
F	698.4	33		8C	46
E	659.2	36		84	42
D#	622.2	39	F9	7C	3E
D	587.3	3D	EB	75	3B
C#	554.3	41	DE	6F	37
C	523.2	44	D1	69	34
B	494	48	C6	63	31
A#	466.1	4D	BA	5D	2F
A	440	51	B0	58	2C
G#	415.3	56	A6	53	2A
G	392	5B	9D	4E	27
F#	370	61	94	4A	25
F	349.2	66	8C	46	23
E	329.6	6C	84	42	21
D#	311.1	73	7C	3E	1F
D	293.6	79	75	3A	1D
C#	277.2	81	6F	37	1C
C	261.6	89	69	34	1A
B	247	91	63	31	19
A#	233.1	99	5D	2F	17
A	220.6	A2	58	2C	16
G#	207.6	AC	53	2A	15
G	196	B6	4E	27	14
		00	E0	70	38
			00		

de muzikale junior
elektuur september 1983

Tabel 1. De codes voor toonhoogte en toonduur, waarmee men een stuk muziek kan "omzetten" in junior-taal.

Tabel 2

JUNIOR

M

HEXDUMP: 200,25D

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0200:	A9	7F	8D	81	1A	A9	08	8D	82	1A	A9	00	85	00	A9	02
0210:	85	02	A6	00	BD	01	03	85	01	F0	E5	A9	40	8D	80	1A
0220:	20	50	02	A6	00	BC	00	03	F0	08	A9	BF	8D	80	1A	20
0230:	50	02	C6	01	D0	E5	C6	02	D0	D8	E6	00	E6	00	A2	FF
0240:	CA	EA	EA	EA	D0	FA	4C	0E	D2	00	00	00	00	00	00	00
0250:	A6	00	BC	00	03	A2	02	CA	D0	FD	88	D0	F8	60		

Tabel 2. Het programma dat via de 6532 en de ULN 2003 het geluidssignaal voor het luidsprekertje opwekt. Een interface voor het luidsprekertje is hier niet nodig.

Tabel 3

JUNIOR

M

HEXDUMP: 300,36B

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0300:	51	58	3D	EA	41	DE	3D	75	36	84	51	58	48	63	5B	9C
0310:	61	4A	5B	4E	6C	84	61	94	79	3A	51	58	3D	EA	41	DE
0320:	3D	75	36	84	51	58	48	63	5B	9C	61	4A	5B	4E	6C	84
0330:	61	94	79	3A	61	94	5B	4E	51	B0	51	58	48	63	48	63
0340:	56	53	51	B0	51	58	3D	EA	48	63	41	6F	41	6F	51	58
0350:	3D	75	3D	75	48	63	41	DE	3D	75	51	58	48	63	5B	9C
0360:	61	4A	5B	4E	6C	84	79	74	00	70	00	00				

Tabel 3. En hier is een voorbeeld van een melodietje gegeven. Op de even adresnummers staat de toonhoogte en op de oneven nummers de toonduur. De code 00 op adres \$036B zorgt er voor dat de processor weer terugkeert naar het begin. Het muziekstuk wordt dus eindeloos herhaald totdat het programma wordt onderbroken.

... en dan weer opbouwen

Vervolgens gaan we enkele nieuwe verbindingen leggen tussen de volgende punten:

- pen 8 van IC12 ... IC19 en pen 1a/1c van de konnektor (+5 V);
- pen 6 van IC7 (N29) en pen 10 van van IC8 (N31);
- pen 8 van IC8 (N31) en pen 5 van IC2 (N47);

- pen 8 van IC6 en pen 2 van IC5 (N19);
- pen 4 van IC10 en pen 2 van IC4 (N18);
- pen 2 van IC10 en pen 19c van de konnektor (A14);
- pen 3 van IC10 en pen 19a van de konnektor (A15);
- pen 18 van IC4, pen 18 van IC5 en pen 9 van IC12 . . . IC19 (A7);
- pen 9 en pen 10 van IC7 (V-W);
- pen 12 en pen 13 van IC7 (X-Y);
- Afhankelijk van de gewenste adresdekodering kunnen de uitgangen van de adresdekoder IC1 worden verbonden met de ingangen V/W en X/Y. Twee pull-up weerstanden van 470 Ω moeten nog worden toegevoegd aan de ingangen van N27 en N28, zoals in het schema is aangegeven. Volgens de in het schema getekende verbindingen ligt de kaart in het adresbereik \$0000 . . . \$BFFF (dat is 8 Kbyte, de

64 K op de dynamische RAM-kaart

één enkele voedingsspanning voor
524288 bits

naar een idee van
K.D. Lorig

Figuur 1. Aansluitgegevens van de dynamische RAM 4164. Het verschil tussen dit IC en de 4116 zit kwa aansluitpunten in pen 1, 8 en 9. Bij de 4164 is adres A7 toegevoegd en zijn de -5 V- en +12 V-aansluitingen vervallen.

Tegenwoordig zijn dynamische RAM's met een capaciteit van 64 Kbits goed verkrijgbaar. Bovendien zijn deze RAM's de laatste tijd zodanig in prijs gedaald dat ze voor de hobbyist zeer aantrekkelijk zijn geworden. Tel daar nog bij dat de meeste 64 K-RAM's slechts één enkele voedingsspanning nodig hebben, dan is er eigenlijk nog maar een konklusie mogelijk: De 16 K RAM-kaart moet 64 K worden! Zo'n ombouw brengt nogal wat voordelen met zich mee. Zo wordt de prijs per bit veel lager dan bij een 16 K-kaart, er is nog maar een enkele voedingsspanning nodig en voor het hele RAM-bereik is nog maar een enkele konnektor nodig, zodat de andere konnektoren vrij blijven (een 8-bits microprocessor kan gewoonlijk maar 64 K adresseren, dus één

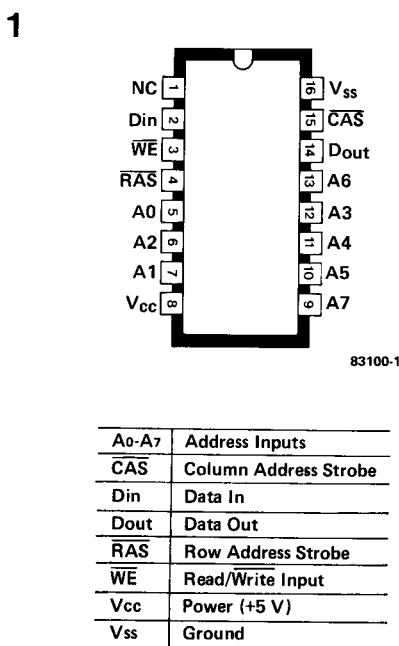
Ruim een jaar na de publikatie van de dynamische RAM-kaart in *Elektuur* (april 1982) kan deze geheugenkaart zich nog steeds in een levendige belangstelling verheugen. Dit komt vooral door het feit dat vele lezers deze 16 K-kaart willen gebruiken als 64 K-kaart, door de 16 Kbit-RAM's te vervangen door 64 Kbit-typen. Hierbij stoten ze dan op moeilijkheden omdat er in dat geval nog verschillende veranderingen op de print moeten worden aangebracht. In dit artikel wordt een lijst met veranderingen gegeven waarmee elke lezer zijn (oude) kaart kan ombouwen tot een even goed lopende kaart met een geheugenkapaciteit van 64 kilobytes.

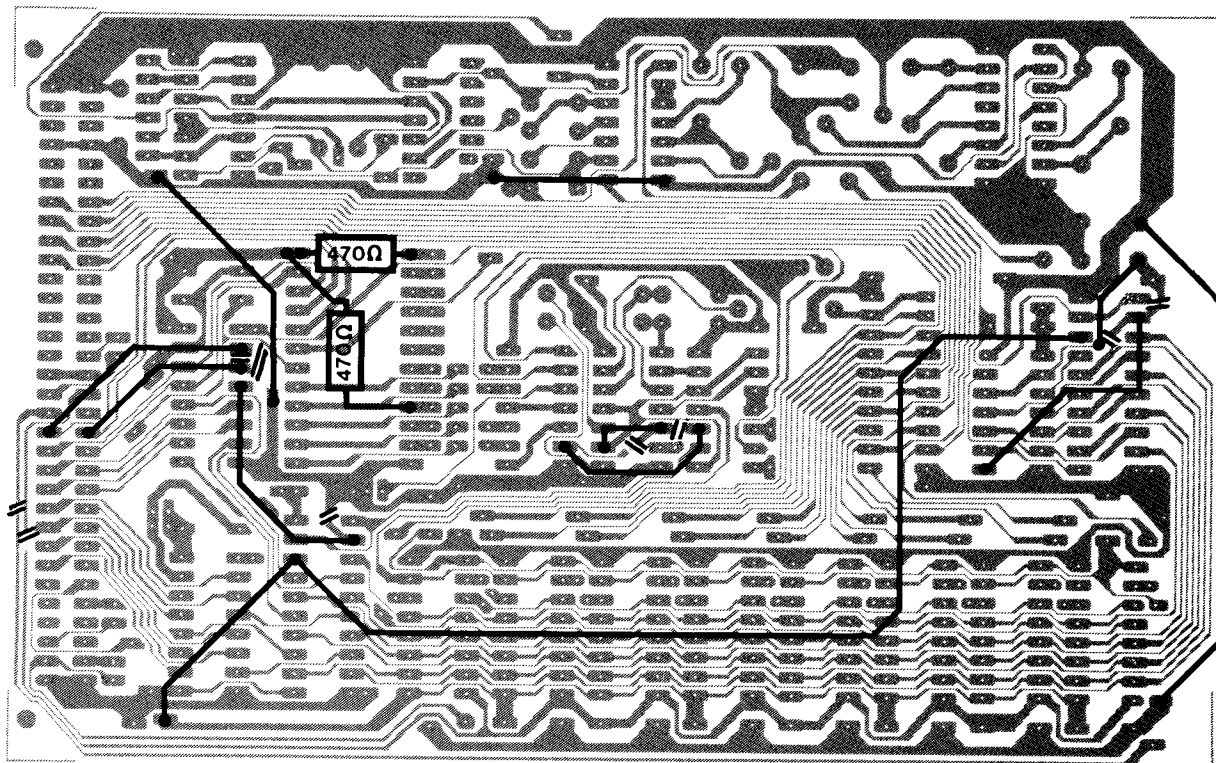
RAM-kaart is dan voldoende voor het hele bereik). Het enige nadeel: er moet hier en daar wat veranderd worden op de dynamische RAM-kaart. Enkele koperbanen moeten worden doorgekrast en er moeten enkele nieuwe verbindingen worden gelegd. Maar al die moeite resulteert wel in een viervoudige geheugenkapaciteit op dezelfde print-oppervlakte.

Eerst afbreken . . .

Om niets aan het toeval over te laten en fouten uit te sluiten hebben we alle veranderingen mooi op een rijtje gezet. Als men zich daaraan houdt bij de rekonstruktie en alles punt voor punt uitvoert kan er niets mis gaan. In figuur 2 en 3 zijn de veranderingen in het schema en op de print voor de duidelijkheid ook nog eens aangegeven. We beginnen met het verwijderen van verschillende onderdelen en het doorkrassen van diverse printbanen.

- Verwijder IC11, IC12 . . . IC19 uit hun voetjes.
- De condensatoren C3, C12 . . . C15, C19 en C20 worden los gesoldeerd en van de print gehaald.
- Verwijder de draadbrug die vlak naast pen 9 . . . 16 van IC9 ligt.
- Onderbreek de volgende koperbanen:
 - de verbinding tussen pen 2 van IC4 (N18) en massa;
 - de verbinding tussen pen 2 van IC5 (N19) en massa (vergeet niet pen 2 en 12 van IC6 daarna weer apart met massa te verbinden);
 - de verbinding tussen pen 8 van IC12 . . . IC19 en +12 V;
 - de verbinding tussen pen 1 van IC12 . . . IC19 en -5 V;
 - de verbinding tussen pen 6 van IC7 (N29) en pen 5 van IC2 (N47);
 - de verbinding tussen pen 5 van IC2 en pen 10 van IC8 (N31);





83100-3

rest van het adresbereik is nodig voor ROM van de junior). De dekodering zoals aangegeven wordt gebruikt bij de DOS-junior.

■ Nu moeten nog twee massaverbindingen worden gelegd, uitgaande van de open 4a/4c van de konnektor aangesloten koperbaan. Dit is duidelijk in figuur 3 aangegeven.

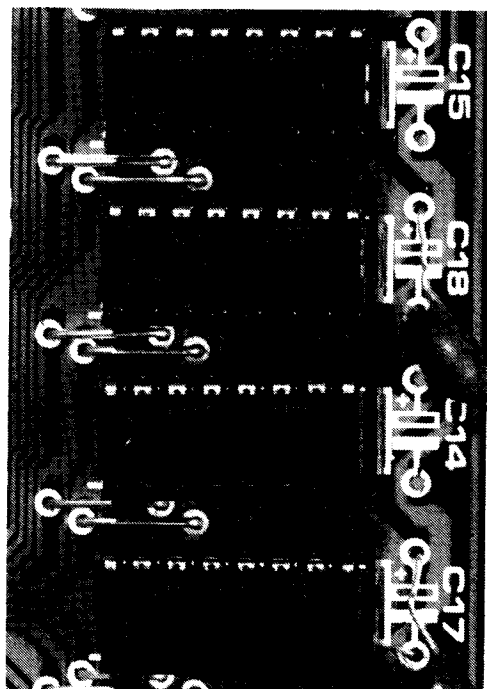
Enkele nieuwe componenten

Als alle onderbrekingen en nieuwe verbindingen goed zijn uitgevoerd zit het zwaarste gedeelte van het karwei erop. Nu hoeven alleen nog maar enkele nieuwe IC's op de print te worden geprikt. Op de plaats van IC11 komt een 74LS159 (een type met open-kollektor-uitgangen, vandaar de aanwezigheid van die twee 470 Ω-weerstand). Daarna volgt nog een laatste inspectie om te zien of men niets vergeten is. Kijk alles zorgvuldig na!

Tenslotte kunnen de nieuwe RAM's in de voetjes worden geplaatst. De 64 K-RAM's worden geleverd door verschillende (voornamelijk Japanse) fabrikanten, waarvan de typenummers verschillen maar de laatste twee cijfers altijd "64" zijn: F4164 (Fairchild), MB8264 (Fujitsu), HM4864 (Hitachi), ITT4164, M5K4164 (Mitsubishi), MK4564 (Mostek), NMC4164 (National Semiconductor), UPD4164C/D (NEC), enzovoorts. Keus genoeg!

Het lijkt ons nuttig om tot slot nog eens te wijzen op het programma voor het testen van het geheugen, zoals beschreven op pagina 4-56 van Elektuur april 1982. Bij 524288 bits is het testen van het geheugen beslist geen overbodige luxe!

Figuur 3. Hier ziet men welke koperbanen moeten worden onderbroken en welke nieuwe verbindingen moeten worden gelegd. Vergeet niet de draadbrug vlak bij IC9 te verwijderen!



worden de OUTPUT-vektor en de JMP op adres 0003 weer goed gezet en keren we terug naar BASIC.

Het leesprogramma

Na het aanroepen van dit programma via X = USR(X) wordt het Basicode-programma van de cassette gelezen en in tabelvorm in RAM opgeslagen. Het programma staat daar dus weer in LIST-formaat. Na het inlezen van het "end of text"-karakter en de checksum staat het hele programma in de tabel, waarna de INPUT-vektor (in de junior-BASIC) wordt vervangen door het startadres van de LDIND-routine en de computer terugkeert in de normale BASIC. De computer zou nu eigenlijk een input van de terminal verwachten (de INPUT-vektor wijst gewoonlijk naar de "receive character"-routine), maar omdat de INPUT-vektor nu naar de LDIND-routine wijst worden de karakters een voor een via deze routine door de junior-BASIC uit de tabel gehaald (en tevens geprint). Voor de BASIC lijkt het dus alsof er in een heel snel tempo een programma wordt ingetypt. Het zo uit de tabel gelezen programma wordt dan door de junior-BASIC op de normale manier verwerkt en opgeslagen. Daarna wordt de INPUT-vektor weer teruggezet en meldt de computer zich met "OK". Men kan dan op de normale manier verder werken met het ingelezen programma.

De BASIC-subroutines

Naast het vertaalprogramma zijn er ook nog enkele standaard-subroutines nodig, die in het Basicode-2-protocol zijn omschreven. Hiervoor wordt verwezen naar het Basicode-artikel in deze Elektuur. Van deze subroutines zijn er drie niet realiseerbaar bij de combinatie junior computer/Elekterminal. Dat zijn de routines 120, 200 en 250. Subroutine 120 heeft betrekking op de plaats van de cursor op het scherm en subroutine 200 moet kijken of er op een bepaald moment een toets is ingedrukt. Beide zijn niet mogelijk door de opzet van de Elekterminal. Subroutine 250 geeft gewoon een piepje, dus dat is niet zo'n gemis. Als vanuit het BASIC-hoofdprogramma de subroutine 120 of 250 wordt aangeroepen gebeurt er eenvoudig niets, want deze routines bestaan bij de junior uit het kommando "RETURN". Bij subroutine 200 wordt IN\$ een lege string bij het verlaten van de routine, zodat het lijkt alsof op dat moment geen enkele toets is gedrukt. In tabel 4 en 5 zijn de standaard-subroutines gegeven voor respectievelijk de uitgebreide junior met KB-9-BASIC en de DOS-junior, beide met Elekterminal. Let op de subroutines 350 en 360. Deze moeten officieel betrekking hebben op een printer, maar in ons geval hebben deze subroutines betrekking op de terminal. De subroutines kunnen voor of na het inlezen van het Basicode-programma worden ingelezen, dat maakt niets uit. Als ze maar aanwezig zijn als men het programma RUNt. Als men bijvoorbeeld het Basicode-programma al heeft ingelezen, dan kunnen de subroutines aan dat programma worden toegevoegd door ze gewoon in te lezen

via POKE . . . : POKE . . . : X = USR(X). We zien hier dat het mogelijk is om twee programma-delen samen te voegen tot één programma, door ze beide apart in te lezen. Voorwaarde is hierbij wel dat de delen geen identieke regelnummers hebben.

Praktijkopmerkingen

Na het inlezen van een Basicode-programma is het verstandig om het hele programma nog eens goed na te kijken. Het komt vaak voor dat er enkele kleinigheidjes in zitten die op uw computer een andere betekenis hebben dan op de computer waarop het programma is ontwikkeld, waardoor een programma dan toch niet loopt. Een voorbeeld: In een Basicode-2-programma dat een doolhof tekent zitten de nodige PRINT-statements. Als nu een deel van de doolhof op het scherm is getekend en het programma vervolgens iets midden in de doolhof wil zetten via een PRINT . . . , dan wordt na het print-statement automatisch een carriage return en line feed gegeneerd. Een carriage return betekent echter bij de Elekterminal dat alles na dit print-statement op de regel wordt gewist. In dit voorbeeld kan het programma gemakkelijk worden aangepast door de desbetreffende PRINT's te laten volgen door een ;. Dan volgt namelijk geen CR en LF en loopt het programma goed. Verder kan het natuurlijk gebeuren dat een van de subroutines wordt aangeroepen die de junior/elekterminal-kombinatie niet kent (120, 200 en 250). Subroutines 200 en 250 zijn niet zo problematisch en kunnen eenvoudig worden omzeild, maar het ontbreken van routine 120 kan wel eens lastig zijn. Als sub 120 bijvoorbeeld in een spelletje wordt gebruikt om de plaats van de cursor op het scherm te bepalen, dan kan het erg moeilijk worden om het programma aan te passen. Subroutine 120 wordt echter ook vaak gebruikt in programma's om de schermgrootte te bepalen. Dit is dan eenvoudig op te lossen door de betreffende programmaregels weg te laten

basicode-2 voor de junior
elektuur oktober 1983

Tabel 4. De standaard-subroutines voor de uitgebreide junior met KB-9-BASIC.

Tabel 5. De standaard-subroutines voor de DOS-junior.

Tabel 4.

LIST
10 GOTO 1000
20 GOTO 1010
100 PRINT
101 POKE6745,200:PRINT CHR\$(12);
102 POKE6745,3
103 RETURN
110 IF HO<63 THEN RETURN
111 IF VE>15 THEN RETURN
112 POKE6745,200:PRINT CHR\$(20);
113 POKE6745,3
114 PRINT
115 IF HO=0 GOTO 117
116 FOR OD=1 TO HO:PRINT CHR\$(9);:NEXT
117 FOR OF=-1 TO 15-VE:PRINT CHR\$(11);:NEXT
118 RETURN
120 RETURN
200 IN\$="":RETURN
210 OS=PEEK(8256):OT=PEEK(8257)
211 POKE8256,(10*16+14):POKE8257,(1*16+2)
212 O=USR(0)
213 POKE8256,OS:POKE8257,OT
214 OX=(PEEK(6754) AND 127)
215 IN\$=CHR\$(OX)
216 RETURN
250 RETURN
260 RV=RND(1):RETURN
270 PR=PEE(0):RETURN
300 IF SR<.01 AND SR>-.01 THEN SR=0
301 IF SGN(SR)=1 THEN SR\$=STR\$(SR):RETURN
302 SR\$=MID\$(STR\$(SR),2):RETURN
310 OS=ABS(SR)+.5*10^-CN:OI=INT(OS):OD=OS-OI+1
311 SR\$=""
312 IF OS>1E9 THEN 321
313 IF CN=0 THEN OD\$="":GOTO 317
314 IF OD=1 THEN OD\$="":GOTO 316
315 OD\$=MID\$(STR\$(OD),3,CN+1)
316 IF LEN(OD\$)<CN+1 THEN OD\$=OD\$+"0":GOTO 316
317 SR\$=MID\$(STR\$(OI),2)+OD\$
318 IF SR<0 AND VAL(SR\$)<0 THEN SR\$="-"+SR\$
319 IF LEN(SR\$)<CT THEN SR\$=" "+SR\$:GOTO 319
320 IF LEN(SR\$)>CT THEN SR\$=""
321 IF LEN(SR\$)<CT THEN SR\$=SR\$+"*":GOTO 321
322 RETURN
350 PRINT SR\$:RETURN
360 PRINT :RETURN

OK

Tabel 5.

LIST
10 GOTO 1000
20 GOTO 1010
100 PRINT
101 POKE64089,200:PRINT CHR\$(12);
102 POKE64089,3
103 RETURN
110 IF HO<63 THEN RETURN
111 IF VE>15 THEN RETURN
112 POKE64089,200:PRINT CHR\$(28);
113 POKE64089,3
114 PRINT
115 IF HO=0 GOTO 117
116 FOR OD=1 TO HO:PRINT CHR\$(9);:NEXT
117 FOR OF=-1 TO 15-VE:PRINT CHR\$(11);:NEXT
118 RETURN
120 RETURN
200 IN\$="":RETURN
210 OS=PEEK(574):OT=PEEK(575)
211 POKE574,(1*16+11):POKE575,(15*16+14)
212 O=USR(0)
213 POKE574,OS:POKE575,OT
214 OX=PEEK(9859)
215 IN\$=CHR\$(OX)
216 RETURN
250 RETURN
260 RV=RND(1):RETURN
270 PR=PEE(0):RETURN
300 IF SR<.01 AND SR>-.01 THEN SR=0
301 IF SGN(SR)=1 THEN SR\$=STR\$(SR):RETURN
302 SR\$=MID\$(STR\$(SR),2):RETURN
310 OS=ABS(SR)+.5*10^-CN:OI=INT(OS):OD=OS-OI+1
311 SR\$=""
312 IF OS>1E9 THEN 321
313 IF CN=0 THEN OD\$="":GOTO 317
314 IF OD=1 THEN OD\$="":GOTO 316
315 OD\$=MID\$(STR\$(OD),3,CN+1)
316 IF LEN(OD\$)<CN+1 THEN OD\$=OD\$+"0":GOTO 316
317 SR\$=MID\$(STR\$(OI),2)+OD\$
318 IF SR<0 AND VAL(SR\$)<0 THEN SR\$="-"+SR\$
319 IF LEN(SR\$)<CT THEN SR\$=" "+SR\$:GOTO 319
320 IF LEN(SR\$)>CT THEN SR\$=""
321 IF LEN(SR\$)<CT THEN SR\$=SR\$+"*":GOTO 321
322 RETURN
350 PRINT SR\$:RETURN
360 PRINT :RETURN

OK

Na de beschrijving van de Basicode-2 ligt het natuurlijk voor de hand dat we ook een praktische uitwerking geven van de Basicode-software met alles wat daar verder bij hoort. En dat we daarbij uitgaan van onze eigen vertrouwde junior computer zal ook niemand verbazen. Door het gebruik van de Basicode-standaard is de junior computer nu in staat om zonder moeilijkheden BASIC-programma's door middel van cassettes uit te wisselen met andere computers. Bovendien kunnen "ontvangen" programma's zonder problemen lopen op de junior, zodat de BASIC-taal in combinatie met de Basicode werkelijk universeel en uitwisselbaar wordt.

basicode-2

voor de junior

junior
kommuniceert
met andere
computers

Aangezien we de Basicode reeds in alle facetten hebben beschreven en er nog heel wat te vertellen valt over de junior-Basicode, zullen we maar gelijk van start gaan. De Basicode-vertaalprogramma's en -subroutines zijn niet gelijk voor de uitgebreide junior en de DOS-junior, in verband met het verschil in geheugenindeling en kleine verschillen in de gebruikte BASIC's. Voor het gebruik van de Basicode is een uitgebreide junior met KB-9-BASIC en Elekterminal nodig of een DOS-junior met Elekterminal.

De vertaalprogramma's

De vertaalprogramma's voor de beide junior-versies zijn in machinetaal geschreven. In tabel 1 is een source-listing afgedrukt van het complete vertaalprogramma, samen met een verklarende (Engelse) tekst. Deze source-listing is voor de uitgebreide junior met KB-9-BASIC. In verband met plaatsgebrek is de source-listing voor de DOS-junior niet afgedrukt, maar deze is toch vrijwel hetzelfde als de hier afgebeelde listing (alleen zijn enkele adressen anders). De hexdump van het vertaalprogramma is te zien in tabel 2 (junior met KB-9-BASIC) en tabel 3 (DOS-junior).

Bij de uitgebreide junior met KB-9-BASIC ligt het vertaalprogramma op de adressen \$0200...\$059B, bij de DOS-junior op de adressen \$E000...\$E39B. Deze bereiken zijn gebruikt omdat daar gewoonlijk altijd RAM aanwezig is. De programma's moeten voor een goede werking trouwens in RAM staan (kunnen dus niet in EPROM worden gezet). Als het programma een keer is ingetypt kan men het gewoon op cassette of floppy wegschrijven, zodat men het een volgende keer alleen maar hoeft in te lezen. Het programma bestaat uit een schrijf- en een leesgedeelte. Aan de hand van het programma voor de uitgebreide junior zullen we het gebruik van het programma beschrijven. De afwijkende punten voor de DOS-junior staan hierbij steeds tussen haakjes.

Lezen

Eerst wordt het Basicode-vertaalprogramma

ingetypt (of ingelezen, als men het al op cassette of floppy heeft staan). Lees- en schrijfprogramma kunnen trouwens als één file op cassette worden gezet: SA = 0200, EA = 059C (DOS-junior: SA = E000, EA = E39C). Daarna wordt de KB-9-BASIC ingelezen van de cassette (zie Elektuur maart 1982). Bij de DOS-junior wordt de BASIC van de floppy gelezen. Als dat is gebeurd kan men de BASIC op de normale manier starten.

Vervolgens kan men een Basicode-programma laden. Hiervoor is wel de kleine interface-schakeling nodig die aan het einde van dit artikel wordt beschreven. Het laden van een Basicode-programma gebeurt als volgt:

Eerst wordt NEW ingetypt om oude programma's te wissen. Vervolgens typen we: POKE 8256,0 : POKE 8257,4 : X = USR(X) (POKE 574,0 : POKE 575,226 : X = USR(X))

gevolgd door een (carriage) return.

Op het hex-display van de junior verschijnt nu het teken $\equiv$ dat aangeeft dat er *geen* synchronisatie is. Men kan dan de cassette-recorder starten. Als het programma (willekeurige) signalen ontvangt gaat het $\equiv$ -teken heen en weer springen op de twee rechter displays. Wordt nu de 2400 Hz-header herkend, dan verschijnt op de twee rechter displays een langzaam overspringend $\diagup$ -teken. Hiermee geeft het programma aan dat het bezig is met synchroniseren. Dit springen op de displays duurt ongeveer 2 s, daarna wordt het teken continu zichtbaar op beide displays gedurende de rest van de "leader". Als de leader voorbij is en de eigenlijke programma-overdracht begint verschijnt op de twee displays het teken $\diagup$. Zolang de data goed wordt ontvangen blijft dit teken mooi gelijkmatig op beide displays oplichten. Als het programma in zijn geheel is ontvangen geeft de computer automatisch een listing van het programma op het beeldscherm of de printer. Aan het einde van de listing meldt op computer zich met "OK". Als er een fout is opgetreden bij het inlezen verschijnt na de listing de melding "CHECKSUM ERROR". In dat geval moet het programma worden gecontroleerd of

Tabel 1.

```

0010: BASICODE WRITE PROGRAM FOR JUNIOR COMPUTER
0020: WITH KB-9 BASIC
0030:
0040:
0050: DATE:14-6-'83
0060:
0070: 0200      ORG      $0200
0080:
0090: POINTERS IN PAGE ZERO (BASIC)
0100:
0110: 0200      INTER *   $0004
0120: 0200      BFRSPL *   $007E BEGINNING OF FREE SPACE POINTER
0130: 0200      BFRSPH *   $007F
0140: 0200      HIMEML *   $0084 END OF RAM POINTER
0150: 0200      HIMEMH *   $0085
0160:
0170: 6522-IC REGISTERS
0180:
0190: 0200      TOCL *    $1804 TIMER ONE COUNTER
0200: 0200      TOCH *    $1805
0210: 0200      TOLL *    $1806 TIMER ONE LATCH
0220: 0200      TOLM *    $1807
0230: 0200      ACR *    $180B AUXILIARY CONTROL REGISTER
0240: 0200      IFR *    $180D INTERRUPT FLAG REGISTER
0250: 0200      IER *    $180E INTERRUPT ENABLE REGISTER
0260:
0270: OUTPUT VECTOR (BASIC)
0280:
0290: 0200      OUTVL *   $2A52 OUTPUT VECTOR
0300: 0200      OUTVH *   $2A53
0310:
0320: TEMPORARY DATA BUFFERS
0330:
0340: 0200      SOVL *    $03F0
0350: 0200      SOVH *    $03F1
0360: 0200      SAVINL *   $03F2
0370: 0200      SAVINH *   $03F3
0380: 0200      CHSUM *   $03F4
0390: 0200      DESTFL *   $03F8
0400:
0410: EXTERNAL SUBROUTINE
0420:
0430: 0200      MESSY *   $0559
0440:
0450:
0460: : INITIALISE POINTERS
0470: : TABLE ROUTINE STARTADDRESS TO OUTPUT VECTOR
0480: :
0490:
0500: 0200 AD 52 2A SAVE LDA OUTVL
0510: 0200 AC 53 2A LDY OUTVH
0520: 0200 8D F0 03 STA SOVL
0530: 0200 8C F1 03 STY SOVH SAVE OUTPUT VECTOR
0540: 0200 A9 42 LDAIM TABLE
0550: 0200 A0 02 LDYIM TABLE /256
0560: 0210 8D 52 2A STA OUTVL
0570: 0211 8C 53 2A STY SOVL NEW OUTPUT VECTOR
0580: 0216 A5 04 LDA INTER
0590: 0218 A4 05 LDY INTER +01
0600: 021A 8D F2 03 STA SAVINL
0610: 021D 8C F3 03 STY SAVINH SAVE INTER VECTOR
0620: 0220 A9 77 LDAIM SVECAS
0630: 0222 A0 02 LDYIM SVECAS /256
0640: 0224 05 04 STA INTER
0650: 0226 84 05 STY INTER +01 NEW INTER VECTOR
0660: 0228 A5 7E LDA BFRSPL
0670: 022A A4 7F LDY BFRSPH
0680: 022C 8D 49 02 STA STTABL +01 BEGINNING FREE SPACE POINTER
0690: 022F 8C 4A 02 STY STTABL +02 TO INDIRECT ADDRESSES STTABL AND
0700: 0232 8D A0 02 STA LDABL +01 LDABL ( TABLE POINTERS )
0710: 0235 8C A1 02 STY LDABL +02
0720: 0238 A9 00 LDAIM S00
0730: 023A 8D F4 03 STA CHSUM CLEAR CHECKSUM
0740: 023D 8D F0 03 STA DESTFL CLEAR DESTROY FLAG
0750: 0240 A9 02 LDAIM S02 START OF TEXT CHARACTER
0760:
0770:
0780: : THIS ROUTINE IS CALLED BY BASIC AND PUTS THE
0790: : LISTING OF THE PROGRAM IN A TABLE
0800: :
0810:
0820: 0242 C9 0A TABLE CMPIM S0A SKIP LF'S
0830: 0244 F0 30 SED OK
0840: 0246 09 00 ORAIM S00
0850: 0248 8D FF FF STTABL STA SFFFF CHARACTER TO TABLE
0860: 024D A0 F4 03 EOR CHSUM
0870: 024E 8D F4 03 STA CHSUM UPDATE CHECKSUM
0880: 0251 E5 49 02 INC STTABL +01 INCREMENT TABLE POINTER
0890: 0254 D0 03 BNE OUTOFM
0900: 0256 E6 4A 02 INC STTABL +02
0910: 0259 A5 05 LDA HIMEMH
0920: 025B CD 4A 02 CMP STTABL +02
0930: 025E D0 16 BNE OK NOT END OF RAM? THEN BRANCH
0940: 0260 A5 84 LDA HIMEML
0950: 0262 CD 49 02 CMP STTABL +01
0960: 0265 D0 0F BNE OK
0970: 0267 A9 FF LDAIM SFF
0980: 0269 8D F0 03 STA DESTFL SET DESTROY FLAG
0990: 026C A5 78 LDA S0078 GO ON AT BEGINNING OF BASIC WORKSPACE
1000: 026E 8D 49 02 STA STTABL +01 AND REWRITE BASIC PROGRAM BY TABLE
1010: 0271 A5 79 LDA S0079
1020: 0273 8D 4A 02 STA STTABL +02
1030: 0276 60 OK RTS
1040:
1050:
1060: : PROGRAM TO TAPE
1070: :
1080:
1090: 0277 48 SVECAS PHA
1100: 0278 98 TPA
1110: 0279 48 PHA
1120: 027A A9 0D LDAIM S0D
1130: 027C 20 42 02 JSR TABLE CARRIAGE RETURN TO TABLE
1140: 027F A9 03 LDAM S03
1150: 0281 20 42 02 JSR TABLE END OF TEXT TO TABLE
1160: 0284 A9 F3 JSR CHSUM
1170: 0287 20 48 02 JSR STTABL CHECKSUM TO TABLE (WITHOUT SETTING BIT7)
1180: 028A A9 7F LDAM S7F
1190: 028C 8D 0E 18 STA IER SET INTERRUPT DISABLE MODE
1200: 028F A9 C0 LDAM S0B
1210: 0291 8D 0B 18 STA ACR SET SQUARE WAVE OUTPUT (PB7)
1220: 0294 A9 01 LDAM S01
1230: 0296 8D 05 18 STA TOCH START TIMER ONE
1240: 0299 20 2B 03 JSR HEADER
1250: 029C 20 39 03 LDATB JSR ZERO
1260: 029F AD FF FF LDATBL LDA SFFF
1270: 02A2 A0 08 LDYIM S08
1280: 02A4 4A OUTCHA LGRA
1290: 02A5 40 PHA
1300: 02A6 80 05 03 BCS HIGH
1310: 02A8 20 39 03 JSR ZERO ONE PERIOD OF 1200HZ (=0)
1320: 02AB 70 03 BVS NEXT BRANCH ALWAYS
1330: 02AD 20 46 03 JSR ONE TWO PERIODS OF 2400HZ (=1)
1340: 02B0 68 PLA
1350: 02B1 88 DEY
1360: 02B2 D0 F0 BNE OUTCHA BITCOUNTER NOT ZERO? THEN BRANCH
1370: 02B4 20 46 03 JSR ONE STOPBIT=1
1380: 02B7 20 46 03 JSR ONE STOPBIT=1
1390: 02BA E8 02 INC LDATBL +01 INCREMENT TABLE POINTER
1400: 02BD D0 03 BNE LDT
1410: 02BF EE A1 02 INC LDATBL +02
1420: 02C2 AC A1 02 LDY LDATBL +01
1430: 02C5 AB 02 LDY LDATBL +01
1440: 02C8 CA 05 BNE CMPARE NOT END OF RAM? THEN BRANCH
1450: 02CA D0 0E BNE CMPARE NOT END OF RAM? THEN BRANCH
1460: 02CC E4 04 CPX HIMEMH
1470: 02CE D0 0A BNE CMPARE
1480: 02D0 A5 78 LDA S0078
1490: 02D2 8D A0 02 STA LDATBL +01 GO ON AT BEGINNING OF BASIC WORKSPACE
1500: 02D5 A5 79 LDA S0079
1510: 02D7 8D A1 02 STA LDATBL +02
1520: 02DA CC 4A 02 CMPARE CPY STTABL +02
1530: 02DD D0 BD BNE LDATBL NOT END OF TABLE? THEN BRANCH
1540: 02DF EC 49 02 CPX STTABL +01
1550: 02E2 D0 B8 BNE LDATBL
1560: 02E4 20 2B 03 JSR HEADER
1570: 02E7 8C 0B 18 STY ACR DISABLE PB7
1580: 02EA AD F2 03 LDA SAVINL
1590: 02ED AC F3 03 LDY SAVINH
1600: 02F0 85 04 INTER INTER
1610: 02F2 84 05 STY INTER +01
1620: 02F4 AD F0 03 LDA SOVL
1630: 02F7 AC F1 03 LDY SOVH
1640: 02FA 8D 52 2A STA OUTVL RESET OUTPUT VECTOR
1650: 02FD 8C 53 2A STY OUTVH
1660: 0300 AD F0 03 LDA DESTFL
1670: 0303 F0 20 BEQ RTRN DESTROY FLAG NOT SET? THEN BRANCH
1680: 0305 A5 79 LDA S0079 DESTROY BASIC PROGRAM IF DESTFL IS SET
1690: 0307 85 7B STA S007B
1700: 0309 85 7D STA S007D
1710: 030B 85 7E STA S007E
1720: 030D A4 78 LDY S0078
1730: 030F C8 INY
1740: 0310 C8 INY
1750: 0311 C8 INY
1760: 0312 84 7A STY S007A
1770: 0314 84 7C STY S007C
1780: 0316 84 7E STY S007E
1790: 0318 A9 00 LDAM S00
1800: 031A A8 TAY
1810: 031B 91 78 STAIY S0078
1820: 031D C8 INY
1830: 031E 91 78 STAIY S0078
1840: 0320 A8 2C LDAM S2C
1850: 0322 20 59 05 JSR MESSY "NEW" MESSAGE
1860: 0325 68 RTRN PLA
1870: 0326 A8 PLA
1880: 0327 68 PLA
1890: 0328 4C 03 00 JMP INTER -01 RETURN TO BASIC
1900:
1910:
1920: : 5 SEC 2400HZ
1930:
1940:
1950: 0328 A2 78 HEADER LDYIM S70 SET X- AND Y-REGISTER FOR
1960: 032D A0 17 LDYIM S17 12000 PERIODS OF 2400HZ (=5 SEC 2400HZ)
1970: 032F 20 46 03 HDR JSR ONE TWO PERIODS OF 2400HZ
1980: 0332 CA DEX
1990: 0333 D0 FA BNE HDR
2000: 0335 88 DEY
2010: 0336 D0 F7 BNE HDR
2020: 0338 60 RTS
2030:
2040:
2050: : ZERO-ONE PERIOD OF 1200HZ
2060: : ONE -TWO PERIODS OF 2400HZ
2070:
2080:
2090: 0339 A9 9F ZERO LDAM S9F
2100: 033B 8D 06 18 STA TOLL
2110: 033E A9 01 LDAM S01
2120: 0340 8D 07 18 STA TOLH SET TIMER ONE FOR 417 MICRO SEC.
2130: 0343 4C 53 03 JMP PERIOD
2140: 0346 28 49 03 JSR SUBONE
2150: 0349 A9 C0 SUBONE LDAM S0C
2160: 034B 8D 06 18 STA TOLL
2170: 034E A9 00 LDAM S00
2180: 0350 8D 07 18 STA TOLH SET TIMER ONE FOR 208 MICRO SEC.
2190: 0353 AD 04 18 PERIOD LDATBL TOCL
2200: 0356 2C 0D 18 PER BIT IFR
2210: 0359 50 FB BVC PER NOT TIME-OFF?
2220: 035B AD 04 18 LDA TOCL
2230: 035E 2C 0D 18 PRD BIT IFR
2240: 0361 50 FB BVC PRD
2250: 0363 60 RTS

```

kan men proberen het programma opnieuw in te lezen.

Het listen mag absoluut niet worden onderbroken door het indrukken van een toets. Doet men dit wel, dan bestaat de kans dat zowel de BASIC als het Basicode-programma opnieuw moeten worden ingelezen. Ook als men ziet dat de listing fouten geeft, bijvoorbeeld regels over elkaar schrijft (dit kan gebeuren bij flinke storingen op de band) moet men wachten tot de computer klaar is met de listing en zich meldt met "OK" of "CHECKSUM ERROR". Daarna kan men weer gewoon in BASIC werken, het programma listen en eventueel controleren en wijzigen. Het kan ook wel eens voorkomen dat de computer het einde van een programma niet herkent. In dat geval meldt hij zich niet terug en blijft het hex-display oplichten.

We kunnen dan de RST-toets indrukken, maar dat heeft tot gevolg dat de BASIC opnieuw moet worden ingelezen. Niet zo fijn dus. Een betere oplossing is het opzoeken van het einde van een ander Basicode-programma op de band en dit af te spelen. De computer herkent dan het einde en meldt zich wel terug op het scherm. Het laatste gedeelte van het ingelezen programma klopt nu natuurlijk niet meer, maar in elk geval kan men dan het goed ingelezen gedeelte bekijken en hoeft men in dat geval de BASIC niet opnieuw in te lezen.

Als het programma dat ingelezen moet worden niet in de beschikbare geheugenruimte past, meldt de computer zich terug met "OUT OF MEMORY". Er verschijnt dan geen listing. Als men toch een listing wil hebben van het gedeelte dat is ingelezen,

```

0010: BASICODE READ PROGRAM FOR JUNIOR COMPUTER
0020: WITH KB-9 BASIC
0030:
0040: DATE:14-6-'83
0050:
0060:
0070: 0400 ORG $0400
0080:
0090: POINTERS IN PAGE ZERO (BASIC)
0100:
0110: 0400 BFRSP * $007E BEGINNING OF FREE SPACE POINTER
0120: 0400 BFRSP * $007F
0130: 0400 HIMEM * $0084 END OF RAM POINTER
0140: 0400 HIMEM * $0085
0150:
0160: 6522-IC REGISTERS
0170:
0180: 0400 PCR * $180C PERIPHERAL CONTROL REGISTER
0190: 0400 IFR * $180D INTERRUPT FLAG REGISTER
0200: 0400 IER * $180E INTERRUPT ENABLE REGISTER
0210:
0220: 6532-IC REGISTERS
0230:
0240: 0400 PAD * $1A80 DATA REGISTER OF PORT A
0250: 0400 PBD * $1A82 DATA REGISTER OF PORT B
0260: 0400 CNTB * $1AF5 CLKBT (INTERRUPT DISABLE)
0270:
0280: INPUT VECTOR (BASIC)
0290:
0300: 0400 INVECL * $2457 INPUT VECTOR
0310: 0400 INVECH * $2458
0320:
0330: TEMPORARY DATA BUFFERS
0340:
0350: 0400 CHSUM * $03F4 CHECKSUM
0360: 0400 PRCLM * $03F5 PERIOD COUNTER
0370: 0400 PRCLM * $03F6
0380: 0400 ZERO * $03F7 PERIODTIME
0390: 0400 HLFTPT * $03F8 HALF PERIODTIME
0400: 0400 SIVL * $03F9
0410: 0400 SIVH * $03FA
0420:
0430: EXTERNAL SUBROUTINE
0440:
0450: 0400 PRCHA * $1334 PRINT CHARACTER ROUTINE
0460:
0470:
0480: INITIALISE AND RECEIVE
0490:
0500:
0510: 0400 A9 7F READ LDAM $7F
0520: 0402 8D 0E 18 STA IER SET INTERRUPT DISABLE MODE
0530: 0405 A9 00 LDAM $00
0540: 0407 8D 0C 18 STA PCR SET CA1 NEGATIVE EDGE DETECT
0550: 040A 8D F4 03 STA CHSUM CLEAR CHECKSUM
0560: 040D A9 73 LDAM $73
0570: 040F 8D 82 1A STA PBD
0580: 0412 A5 7E LDA BFRSP BEGINNING OF FREE SPACE POINTER
0590: 0414 8D 65 04 STA STAIND +01 TO INDIRECT ADDRESS STAIND
0600: 0417 A6 7F LDX BFRSP
0610: 0419 E8 INX
0620: 041A E4 05 CPX HIMEM 256 BYTE FOR EXPANSION
0630: 041C B0 63 BCS STOP
0640: 041E 8E 66 04 STX STAIND +02
0650:
0660: 0421 A9 10 HEADER LDAM $10
0670: 0423 8D F6 03 STA PRCLM SET PERIOD COUNTER
0680: 0426 A9 36 LDAM $36
0690: 0428 2D 4D 05 JSR NOSYNC DISPLAY NOSYNC CHARACTER
0700: 042B 2D 0D 04 JSR PERIOD
0710: 042E C9 22 CMPIM $22
0720: 0430 90 EF BCC HEADER PERIODTIME << 2400HZ PERIODTIME
0730: 0432 C9 4E CMPIM $4E
0740: 0434 B0 EB BCS HEADER PERIODTIME >> 2400HZ PERIODTIME
0750: 0436 EE F5 03 INC PRCLM
0760: 0439 D0 F0 BNE HDR NOT 256 PERIODS? THEN BRANCH
0770: 043B A8 TAY
0780: 043C 2D 4B 05 JSR SYNC DISPLAY SYNC CHARACTER
0790: 043E 98 TYA
0800: 0440 CE F6 03 DEC PRCLM
0810: 0443 D0 E5 BNE HDR NOT 16*256 PERIODS OF 2400HZ? THEN BRANCH
0820: 0445 0A ASLA
0830: 0446 38 SEC
0840: 0447 E9 8C SBCIM $8C
0850: 0449 8D F7 03 STA ZERO
0860: 044C 2D 4B 05 JSR SYNC
0870: 044E D0 05 BNE HDR FIND STARTBIT
0880: 0450 C9 F7 03 CMP ZERO
0890: 0453 90 F5 BCC STABIT 10 1200HZ PERIOD? THEN BRANCH
0900: 0457 B0 08 BCS READYBY BRANCH ALWAYS
0910: 0459 2D 06 75 JSR HLFFPT FIND STARTBIT
0920: 045C D0 F7 03 CMP ZERO
0930: 045E 90 F8 BCC STRTBT
0940: 0461 2D 24 05 JSR RBYT READ ONE BYTE
0950:
0960: 0464 BD FF FF STAIND STA SFFF CHARACTER TO TABLE
0970: 0467 C9 03 CMPIM $03
0980: 0469 F0 21 BEQ EOT END OF TEXT CHARACTER? THEN BRANCH
0990: 046B EE 65 04 INC STAIND +01 INCREMENT POINTER
1000: 046E D0 03 BNE EORAM
1010: 0470 EE 66 04 INC STAIND +02
1020: 0473 A5 05 LDA HIMEM
1030: 0475 CD 66 04 CMP STAIND +02
1040: 0478 D0 0F BNE STRTBT NOT END OF RAM? THEN BRANCH
1050: 047A A5 04 LDA HIMEM
1060: 047C CD 65 04 CMP STAIND +01
1070: 047F D0 0F BNE STRTBT
1080: 0481 A0 00 LDYIM $00
1090: 0483 2D 59 05 JSR MESSY "OUT OF MEMORY" MESSAGE
1100: 0486 A9 67 LDAM $67
1110: 0488 8D 82 1A STA PBD
1120: 048B 60 RTS RETURN TO BASIC
1130:
1140: 048C 2D 00 05 EOT JSR HLFFPT
1150: 048F CD F7 03 CMP ZERO
1160: 0492 90 F8 BCC EOT
1170: 0494 2D 24 05 JSR RBYT READ CHECKSUM
1180: 0497 A9 67 LDAM $67
1190: 0499 8D 82 1A STA PBD
1200: 049C A5 7E LDA BFRSP TABLE STARTADDRESS TO LDIND
1210: 049E 8D BF 04 STA LDIND +01
1220: 04A1 A6 7F LDX BFRSP
1230: 04A3 E8 INX
1240: 04A4 8E C0 04 STX LDIND +02
1250: 04A7 AD 57 24 LDA INVECL
1260: 04AA AC 58 24 LDY INVECH
1270: 04AD 8D F9 03 STA SIVL SAVE INPUT VECTOR
1280: 04B0 8C FA 03 STY SIVH
1290: 04B3 A9 BE LDAM LDIND
1300: 04B5 A0 04 LDYIM LDIND /256
1310: 04B7 8D 57 24 STA INVECL NEW INPUT VECTOR
1320: 04BA 8C 58 24 STY INVECH
1330: 04BD 60 RTS
1340:
1350:
1360: THIS ROUTINE IS CALLED BY BASIC AND TRANSFERS
1370: EVERY CHARACTER FROM THE TABLE TO BASIC
1380: IT ALSO GIVES A LISTING OF THE PROGRAM
1390:
1400:
1410: 04BE AD FF FF LDIND LDA SFFF CHARACTER FROM TABLE
1420: 04C1 A8 TAY
1430: 04C2 2D 34 13 JSR PRCHA
1440: 04C5 EE BF 04 INC LDIND +01 INCREMENT POINTER
1450: 04C8 D0 03 BNE INCPNT
1460: 04CA EE C0 04 INC LDIND +02
1470: 04CD AD 66 04 INC PBD
1480: 04D0 CD C0 04 CMP LDIND +02
1490: 04D3 D0 26 BNE RETBAS NOT END OF TABLE? THEN BRANCH
1500: 04D5 AD 65 04 LDA STAIND +01
1510: 04D8 CD BF 04 CMP LDIND +01
1520: 04DB D0 1E BNE RETBAS
1530: 04DD AD F9 03 LDA SIVL
1540: 04E0 8D 57 24 STA INVECL RESET INPUT VECTOR
1550: 04E3 AD FA 03 LDA SIVH
1560: 04E6 8D 58 24 STA INVECH
1570: 04E9 AD F4 03 STY CHSUM
1580: 04EC F0 05 BEQ RETURN NO CHECKSUM ERROR? THEN BRANCH
1590: 04EE A0 12 LDYIM $12
1600: 04F0 2D 59 05 JSR MESSY "CHECKSUM ERROR" MESSAGE
1610: 04F3 A0 25 LDYIM $25
1620: 04F5 2D 59 05 JSR MESSY "OK" MESSAGE
1630: 04F8 A9 0D LDAM $0D
1640: 04FA 60 RTS CARRIAGE RETURN
1650: 04FB 90 RETBAS TYA
1660: 04FC 60 RTS
1670:
1680:
1690: MEASURE ONE PERIODTIME
1700:
1710:
1720: 04FD 2D 00 05 PERIOD JSR HLFFPT
1730: 0500 A9 82 JSR LDAM $82
1740: 0502 2C 0D 18 HLF BIT
1750: 0505 F0 F8 BBEQ HLF NO ACTIVE EDGE ON CA1-INPUT? THEN BRANCH
1760: 0507 8D 0D 18 STA IFR
1770: 050A A9 01 LDAM $01
1780: 050C D0 8C 18 BCR PCR
1790: 050F 8D 0C 18 STA PCR OPPOSITE ACTIVE CA1 EDGE DETECT
1800: 0512 A9 FF LDAM $FF
1810: 0514 AA TAX
1820: 0516 AD F6 1A STX SIAF6 GET ELAPSED TIME IN ACCU
1830: 0518 8E F5 1A STX CNTB RESET TIMER
1840: 051B AA TAX
1850: 051C 18 CLC
1860: 051D 6D F8 03 ADC HLFTPT FULL PERIODTIME IN ACCU
1870: 0520 8E F8 03 STX HLFTPT SAVE LAST HALF PERIODTIME
1880: 0523 60 RTS
1890:
1900:
1910: READ ONE BYTE
1920:
1930:
1940: 0524 A9 55 RBYT LDAM $55
1950: 0526 2D AD 05 JSR LDAM $05
1960: 0529 A0 08 LDYIM $08 SET BITCOUNTER
1970: 052B 48 RB FRA SAVE ACCU
1980: 052C 2D FD 04 JSR PERIOD
1990: 052F 2C 0D 18 CMP ZERO
2000: 0532 8D 06 BCR FNDZRO 1200HZ PERIOD? THEN BRANCH
2010: 0534 2D FD 04 JSR PERIOD SECOND 2400HZ PERIOD
2020: 0537 38 SEC
2030: 0538 8D 01 BCS SHIFT BRANCH ALWAYS
2040: 053A 18 RORA
2050: 053B 68 SHIFT PLA
2060: 053C 6A RORA NEXT BIT TO ACCU
2070: 053D 88 DEY
2080: 053E D0 EB BNE RB NOT 8 BITS? THEN BRANCH
2090: 0540 48 PHA
2100: 0541 4D F4 03 EOR CHSUM
2110: 0544 8D F4 03 STA CHSUM UPDATE CHECKSUM
2120: 0547 60 PLA
2130: 0548 2D 7F ANDIM $7F CLEAR BIT 7
2140: 054A 60 RTS
2150:
2160:
2170: DISPLAY SYNC-CHARACTER ON 7-SEGM.DISPLAY
2180:
2190:
2200: 054B A9 69 SYNC LDAM $69
2210: 054D 8D 80 1A NOSYNC STA PAD
2220: 0550 AD 82 1A LDA PBD
2230: 0553 A9 02 EORIM $02
2240: 0555 8D 82 1A STA PBD
2250: 0558 60 RTS
2260:
2270:
2280: OUTPUT MESSAGE (MESS+Y)
2290:
2300:
2310: 0559 B9 68 05 MESSY LDAM MESS LOAD CHARACTER
2320: 055C C9 03 CNPIM $03
2330: 055E 8E 87 BBEQ MESEND END OF TEXT CHARACTER?
2340: 0560 2D 34 13 JSR PRCHA
2350: 0563 C8 INY
2360: 0564 4C 59 05 JMP MESSY
2370: 0567 60 MESEND RTS
2380:
2390: 0568 8D MESS = $8D
2400: 0569 8A = $8A
2410: 056A 4F = 'O'
2420: 056B 55 = 'U'
2430: 056C 54 = 'T'
2440: 056D 20 = ' '
2450: 056E 4F = 'O'
2460: 056F 4E = 'F'
2470: 0570 20 = ' '
2480: 0571 4D = 'M'
2490: 0572 45 = 'E'
2500: 0573 4D = 'M'
2510: 0574 4F = 'O'
2520: 0575 52 = 'Y'
2530: 0576 59 = 'Y'
2540: 0577 8A = $8A
2550: 0578 8D = $8D
2560: 0579 03 = $03
2570: 057A 8D = $8D
2580: 057B 8A = $8A
2590: 057C 43 = 'C'
2600: 057D 48 = 'H'
2610: 057E 45 = 'E'
2620: 057F 43 = 'C'
2630: 0580 4B = 'K'
2640: 0581 53 = 'S'
2650: 0582 55 = 'M'
2660: 0583 4D = 'U'
2670: 0584 20 = ' '
2680: 0585 45 = 'E'
2690: 0586 52 = 'R'
2700: 0587 52 = 'R'
2710: 0588 4F = 'O'
2720: 0589 52 = 'R'
2730: 058A 0A = $0A
2740: 058B 8D = $8D
2750: 058C 43 = $03
2760: 058D 8D = $8D
2770: 058E 8A = $8A
2780: 058F 4F = 'O'
2790: 0590 4B = 'K'
2800: 0591 8A = $8A
2810: 0592 8D = $8D
2820: 0593 03 = $03
2830: 0594 8D = $8D
2840: 0595 8A = $8A
2850: 0596 4E = 'N'
2860: 0597 45 = 'E'
2870: 0598 57 = 'W'
2880: 0599 8A = $8A
2890: 059A 8D = $8D
2900: 059B 03 = $03

```

Tabel 1. De source-listing van het complete vertaal-programma. Deze listing geldt voor de uitgebreide junior met KB-9-BASIC. De listing van het programma voor de DOS-junior is praktisch identiek aan de hier afgedrukte listing, met uitzondering van enkele adressen.

HEXDUMP: 0200,0363																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0200:	AD	52	2A	AC	53	2A	8D	F0	03	8C	F1	03	A9	42	A0	02
0210:	8D	52	2A	8C	53	2A	A5	04	A4	05	8D	F2	03	8C	F3	03
0220:	A9	77	A0	02	85	04	84	05	A5	7E	A4	7F	8D	49	02	8C
0230:	4A	02	8D	A0	02	8C	A1	02	A9	00	8D	F4	03	8D	FB	03
0240:	A9	02	C9	0A	F0	30	09	80	8D	FF	FF	4D	F4	03	8D	F4
0250:	03	EE	49	02	D0	03	EE	4A	02	A5	85	CD	4A	02	D0	16
0260:	A5	84	CD	49	02	D0	0F	A9	FF	8D	FB	03	A5	78	8D	49
0270:	02	A5	79	8D	4A	02	60	48	98	48	A9	0D	20	42	02	A9
0280:	03	20	42	02	AD	F4	03	20	48	02	A9	7F	8D	0E	18	A9
0290:	C0	8D	0B	18	A9	01	8D	05	18	20	2B	03	20	39	03	AD
02A0:	FF	FF	A0	08	4A	48	B0	05	20	39	03	70	03	20	46	03
02B0:	68	88	D0	F0	20	46	03	20	46	03	EE	A0	02	D0	03	EE
02C0:	A1	02	AC	A1	02	AE	A0	02	C4	85	D0	0E	E4	84	D0	0A
02D0:	A5	78	8D	A0	02	A5	79	8D	A1	02	CC	4A	02	D0	BD	EC
02E0:	49	02	D0	B8	20	2B	03	8C	0B	18	AD	F2	03	AC	F3	03
02F0:	85	04	84	05	AD	F0	03	AC	F1	03	8D	52	2A	8C	53	2A
0300:	AD	FB	03	F0	20	A5	79	85	7B	85	7D	85	7F	A4	78	C8
0310:	C8	C8	84	7A	84	7C	84	7E	A9	00	A8	91	78	C8	91	78
0320:	A0	2C	20	50	05	68	A8	68	4C	03	08	A2	70	A0	17	20
0330:	46	03	CA	D0	FA	88	D0	F7	60	A9	9F	8D	06	18	A9	01
0340:	8D	07	18	4C	53	03	20	49	03	A9	CE	8D	06	18	A9	00
0350:	8D	07	18	AD	04	18	2C	0D	18	50	FB	AD	04	18	2C	0D
0360:	18	50	FB	60												

JUNIOR

M	HEXDUMP: 0400,059B															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0400:	A9	7F	8D	0E	18	A9	00	8D	0C	18	8D	F4	03	A9	73	8D
0410:	82	1A	A5	7E	8D	65	04	A6	7F	E8	E4	85	B0	63	8E	66
0420:	A0	A9	10	8D	F6	03	A9	36	20	4D	05	20	FD	04	C9	22
0430:	90	EF	C9	4E	B0	EB	EE	F5	03	D0	F0	A8	20	4B	05	98
0440:	CE	F5	03	D0	E6	0A	38	E9	0C	8D	F7	03	20	4B	05	20
0450:	00	05	CD	F7	03	90	F5	B0	08	20	00	05	CD	F7	03	90
0460:	F8	20	24	05	8D	FF	FF	C9	03	F0	21	EE	65	04	D0	03
0470:	EE	66	04	A5	85	CD	66	04	D0	DF	A5	84	CD	65	04	D0
0480:	D8	A0	00	20	59	05	A9	67	8D	82	1A	60	20	00	05	CD
0490:	F7	03	90	F8	20	24	05	A9	67	8D	82	1A	A5	7E	8D	BF
04A0:	04	A6	7F	E8	EE	C0	04	AD	57	24	AC	58	24	8D	F9	03
04B0:	8C	FA	03	A9	EE	A0	04	8D	57	24	8C	58	24	60	AD	FF
04C0:	FF	A8	20	34	13	EE	BF	04	D0	03	EE	C0	04	AD	66	04
04D0:	CD	C0	04	D0	26	AD	65	04	CD	BF	04	D0	1E	AD	F9	03
04E0:	8D	57	24	AD	FA	03	8D	58	24	AD	F4	03	F0	05	A0	12
04F0:	20	59	05	A0	25	20	59	05	A9	0D	60	98	60	20	00	05
0500:	A9	02	2C	0D	18	F0	FB	8D	0D	18	A9	01	4D	0C	18	8D
0510:	0C	18	A9	FF	AA	4D	F6	1A	8E	F5	1A	AA	18	6D	F8	03
0520:	BE	F8	03	60	A9	55	20	4D	05	A0	08	48	20	FD	04	CD
0530:	F7	03	B0	06	20	FD	04	38	B0	01	18	68	6A	88	D0	EB
0540:	48	4D	FA	03	8D	F4	03	68	29	7F	60	A9	69	8D	80	1A
0550:	AD	82	1A	49	02	8D	82	1A	60	B9	68	05	C9	03	F0	07
0560:	20	34	13	C8	4C	59	05	60	0D	0A	4F	55	54	20	4F	46
0570:	20	4D	45	4D	4F	52	59	0A	0D	03	0D	0A	43	48	45	43
0580:	48	53	55	4D	20	45	52	52	4F	52	0A	0D	03	0D	0A	4F
0590:	4B	0A	0D	03	0D	0A	4E	45	57	0A	0D	03				

JUNIOR

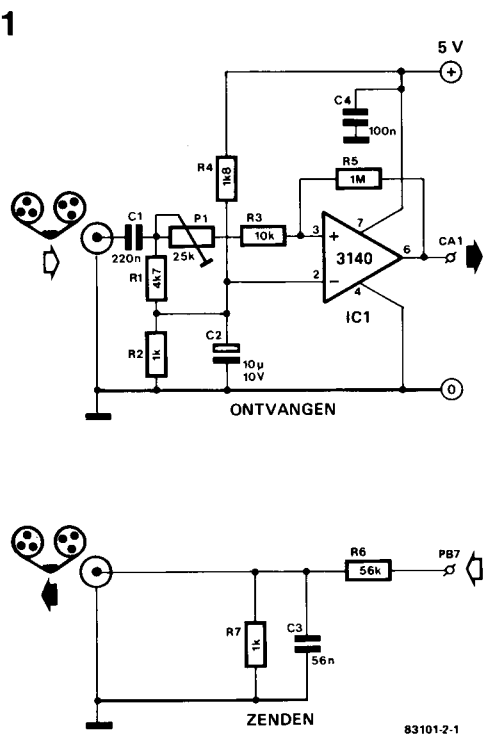
HEXDUMP:	E000,E163															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
E000:	AD	11	23	AC	12	23	A5	04	E1	8C	F1	E1	A9	41	A0	E0
E010:	8D	11	23	8C	12	23	A5	04	A4	05	8D	F2	E1	8C	F3	E1
E020:	A9	77	A0	E0	85	04	84	05	A5	7E	A4	7F	8D	49	E0	8C
E030:	4A	E0	8D	A0	E0	8C	A1	E0	A9	00	8D	F4	E1	8D	FB	E1
E040:	A9	02	C9	0A	F0	30	09	80	8D	FF	FF	4D	F4	E1	8D	F4
E050:	E1	EE	49	E0	D0	03	EE	4A	E0	A5	85	CD	4A	E0	D0	16
E060:	A5	84	CD	49	E0	D0	0F	A9	FF	8D	FB	E1	A5	78	8D	49
E070:	02	A5	79	8D	4A	E0	60	48	98	48	A9	0D	20	42	02	A9
E080:	03	20	42	E0	AD	F4	E1	20	48	02	A9	7F	8D	0E	18	A9
E090:	C0	8D	0B	F8	A9	01	8D	05	F8	20	2B	E1	20	39	E1	AD
E0A0:	FF	FF	A0	08	4A	48	B0	05	20	39	E1	70	03	20	46	E1
E0B0:	68	88	D0	F0	20	46	E1	20	46	E1	EE	A0	E0	D0	03	EE
E0C0:	A1	E0	AC	A1	E0	AE	A0	E0	C4	85	D0	0E	E4	84	D0	0A
E0D0:	A5	78	8D	A0	E0	A5	79	8D	A1	E0	CC	4A	E0	D0	BD	EC
E0E0:	49	E0	D0	B8	20	2B	E1	8C	0B	F8	AD	F2	E1	AC	F3	E1
E0F0:	85	04	84	05	AD	F0	E1	AC	F1	E1	8D	11	23	8C	12	23
E100:	AD	FB	E1	F0	20	A5	79	85	7B	85	7D	85	7F	A4	78	C8
E110:	C8	C8	84	7A	84	7C	84	7E	A9	00	A8	91	78	C8	91	78
E120:	A0	2C	20	59	E3	68	A8	68	4C	03	08	A2	70	A0	17	20
E130:	46	E1	CA	D0	FA	88	D0	F7	60	A9	9F	8D	06	F8	A9	01
E140:	8D	07	F8	4C	53	E1	20	49	E1	A9	CE	8D	06	F8	A9	00
E150:	8D	07	F8	AD	04	F8	2C	0D	F8	50	FB	AD	04	F8	2C	0D
E160:	F8	50	FB	60												

JUNIOR

HEXDUMP:	E200,E39B															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
E200:	A9	7F	8D	0E	F8	A9	00	8D	0C	F8	8D	F4	E1	A9	73	8D
E210:	82	FA	A5	7E	8D	65	E2	A6	7F	E8	E4	85	B0	63	8E	66
E220:	A0	A9	10	8D	F6	E1	A9	36	20	4D	E3	20	FD	E2	C9	22
E230:	90	EF	C9	4E	B0	EB	EE	F5	E1	D0	F0	A8	20	4B	E3	98
E240:	CE	F5	E1	D0	E6	0A	38	E9	0C	8D	F7	E1	20	4B	E3	20
E250:	00	E3	CD	F7	E1	90	F5	B0	08	20	00	E3	CD	F7	E1	90
E260:	F8	20	24	E3	8D	FF	FF	C9	03	F0	21	EE	65	E2	D0	03
E270:	EE	66	E2	A5	85	CD	66	E2	D0	DF	A5	84	CD	65	E2	D0
E280:	D8	A0	00	20	59	E3	A9	67	8D	82	1A	60	20	00	E3	CD
E290:	F7	E1	90	F8	20	24	E3	A9	67	8D	82	1A	A5	7E	8D	BF
E2A0:	E2	A6	7F	E8	EE	C0	E2	AD	01	23	AC	58	24	8D	F9	E1
E2B0:	8C	FA	E1	A9	EE	A0	E2	AD	01	23	8C	58	24	60	AD	FF
E2C0:	FF	A8	20	34	E3	EE	BF	E2	D0	03	EE	C0	E2	AD	66	E2
E2D0:	CD	C0	E2	D0	26	AD	E5	E2	CD	BF	E2	D0	1E	AD	F9	E1
E2E0:	8D	01	23	AD	FA	E1	8D	02	23	AD	F4	E1	F0	05	A0	12
E2F0:	20	59	E3	A0	25	20	59	E3	A9	0D	60	98	60	20	00	E3
E300:	A9	02	2C	0D	F8	F0	FB	8D	0D	F8	A9	01	4D	0C	F8	8D
E310:	0C	F8	A9	FF	AA	4D	F6	1A	8E	F5	1A	AA	18	6D	F8	E1
E320:	BE	F8	E1	F0	A9	55	20	4D	E3	A0	08	48	20	FD	E2	CD
E330:	F7	E1	B0	06	20	FD	E2	38	B0	01	18	68	6A	88	D0	EB
E340:	48	4D	FA	E1	8D	F4	E1	68	29	7F	60	A9	69	8D	80	1A
E350:	AD	82	FA	E1	02	8D	82	FA	E1	60	B9	68	E3	C9	F0	07
E360:	20	34	23	C8	4C	59	E3	60	0D	0A	4F	55	54	20	4F	46
E370:	20	4D	45	4D	4F	52	59	0A	0D	03	0D	0A	43	48	45	43
E380:	4B	53	55	4D	20	45	52	52	4F	52	0A	0D	03	0D	0A	4F
E390:	4B	0A	0D	03	0D	0A	4E	45	57	0A	0D	03				

JUNIOR

Tabel 3. De hexdump van
het vertaalprogramma voor
de DOS-junior.



Figuur 1. Het schema van de interface-schakeling die tussen cassette-recorder en junior computer moet worden opgenomen.

Onderdelenlijst

Weerstanden:

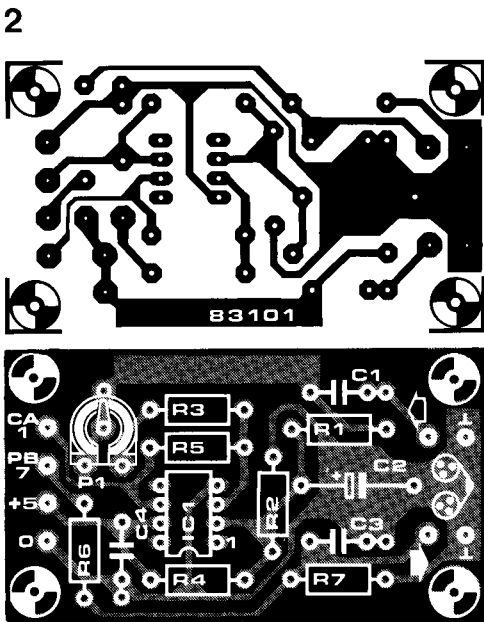
- R1 = 4k7
- R2, R7 = 1 k
- R3 = 10 k
- R4 = 1k8
- R5 = 1 M
- R6 = 56 k
- P1 = 25 k instelpotmeter

Kondensatoren:

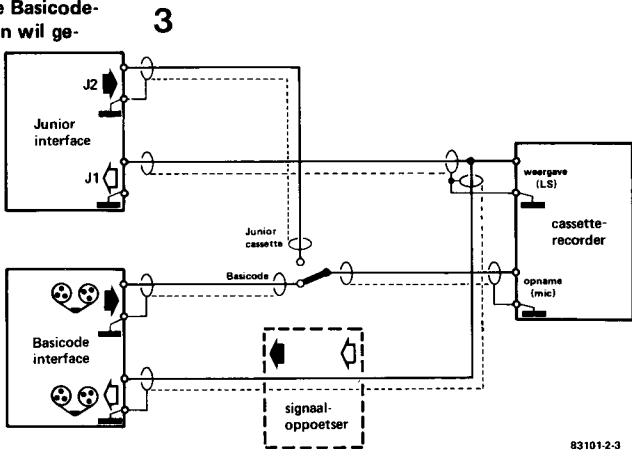
- C1 = 220 n
- C2 = 10 μ /10 V
- C3 = 56 n
- C4 = 100 n

Halfgeleider:
IC1 = 3140

Figuur 2. Het printje voor de interface-schakeling.



Figuur 3. Dit bedradings-schema moet worden aangehouden als men de gewone junior-cassette-interface en de Basicode-interface samen wil gebruiken.



en op de vrijgekomen regels gewoon in te geven hoe groot het scherm is (16 regels van 64 karakters bij de Elekterminal). Een voorbeeld: in een programma-gedeelte wordt de grootte van het scherm bepaald, waarbij de variabelen VV en HH na het verlaten van dit gedeelte de hoogte en de breedte van het scherm moeten bevatten. In ons geval wordt dit programmadeel dan gewoon vervangen door: VV = 15: HH = 63 (denk eraan dat de eerste plaats altijd nummer nul heeft). Let tenslotte ook op het teken @ bij KB-9-BASIC. Als de computer dat teken tegenkomt wordt de hele regel gewist, waarna CR en LF wordt gegeven.

De hardware

De hardware voor de Basicode bestaat uit een kleine aanpassingsschakeling die tussen de cassette-recorder en de junior computer wordt opgenomen.

In figuur 1 is de schakeling afgebeeld. Ze bestaat uit een zend- en ontvanggedeelte. Het ontvanggedeelte bestaat uit een enkel IC (3140), dat is geschakeld als schmitt-trigger/nivo-aanpasser. Met behulp van P1 kan het triggernivo binnen zekere grenzen worden ingesteld, maar meestal werkt de schakeling naar behoren als de potmeter gewoon in de middenstand staat. Het zendgedeelte doet niets anders dan het uitgangssignaal van de junior verzwakken en de hogere harmonischen uit het signaal filteren. Het printje voor de interface (figuur 2) is zo opgezet dat men twee cinch-bussen (voor in- en uitgang) direkt op het printje kan solderen met behulp van een paar draadstukjes. De punten CA1 en PB7 worden verbonden met de overeenkomstige punten van de VIA-konnektor op de interface-print.

Als men zowel de normale junior-cassette-interface als de Basicode-interface wil aansluiten (die eerste is nodig voor het inlezen van machinetaalprogramma's), dan moet worden opgepast bij het bedraden van de interfaces. In figuur 3 is een bedradings-schema gegeven voor het aansluiten van beide interfaces. Wijk niet van dit bedradings-schema af, anders ontstaan er aardlussen en is de kans op oscillatie zeer groot. In figuur 3 is ook nog een blokje "signaal-oppoetser" te zien. Deze schakeling, die eveneens in deze Elektuur wordt beschreven, hoeft alleen te worden tussengevoegd op de aangegeven plaats als het signaal dat van de recorder (of de radio) komt erg slecht van kwaliteit is. Het is aan te raden om alles eerst zonder signaal-oppoetser te proberen. Als het op die manier niet lukt kan de schakeling nog altijd worden toegevoegd.

Regelmatig ontvangen we lezersbrieven met de vraag: hoe kan ik de EPROMmer (Elektuur januari 1982) gebruiken bij een junior computer *zonder* interface-kaart? Welnu, met de hier beschreven modifikaties geven we een duidelijk antwoord op die vraag!

M. Seiler en R. Kisse

EPROMmer zonder interface-kaart

2716's
eenvoudig
programmeren
met de
junior
computer

EPROM's kunnen voor de meest uiteenlopende toepassingen worden gebruikt: niet alleen voor het opslaan van veelgebruikte computerprogramma's, maar ook voor tabellen, omzetskodes en dergelijke. Denk maar aan de junior computer met al zijn uitbreidingen, het nieuwe ASCII-toetsenbord, de video-kaart en de in dit nummer beschreven Quantisizer. Een bijzonder populaire EPROM is wel de 2716, een type dat 2 Kbyte niet-vluchtige data kan bevatten. Dit IC koppelt een flinke opslagcapaciteit aan een lage prijs.

Voor het programmeren van zo'n EPROM is een programmer eigenlijk onontbeerlijk, anders is het niet mogelijk om zelf iets in een EPROM te zetten.

Een eenvoudige combinatie

Een eenvoudige combinatie voor het programmeren van EPROM's kan bestaan uit een junior computer met de in januari 1982 gepubliceerde EPROMmer, alleen hebben we nog nooit beschreven hoe die twee printen aan elkaar kunnen worden gekoppeld. Twee weerstanden vormen de enige componenten die daarvoor moeten worden toegevoegd, verder is er niets extra's nodig. In tegendeel zelfs, sommige onderdelen op de EPROMmer-kaart vervallen zelfs: R1...R4, S3...S6 en IC5. Als men het IC wil laten zitten (omdat het bijvoorbeeld rechtstreeks op de print is gesoldeerd), dan kan men ook de verbinding onderbreken tussen pen 6 van IC5 en pen 5 van IC10 (N7) en de verbinding tussen pin 6 van de 74LS85 en pen 2 en 12 van IC8.

Daarmee is de originele adresdekodering dan voorgoed buiten werking gesteld. Deze wordt vervangen door het gedeelte dat boven in figuur 1 is getekend. Die schakeling maakt een chip-select-sigitaal uit twee K-signalen, die geleverd worden door IC6 op de basisprint van de junior computer.

De twee poorten voor de nieuwe adresdekoder zijn reeds aanwezig op de EPROMmer-kaart. De EXOR-poort is nog vrij in IC12, terwijl de AND-poort in IC9 zit. De twee

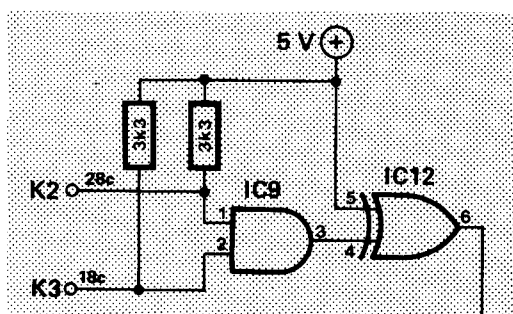
ingangen van de AND-poort uit IC9 moeten via twee weerstandjes van 3k3 met +5 V worden verbonden.

Als men deze veranderingen op de EPROMmer-kaart heeft aangebracht, dan hoeven alleen nog maar de verbindingen tussen de ingangen van de nieuwe adresdekoder te worden verbonden met twee K-uitgangen van de junior-basisprint. Daartoe wordt eerst tabel 1 geraadpleegd. Aan de hand daarvan kan men kiezen in welk (nog vrij) adresbereik de EPROMmer wordt gelegd. In de tabel zijn ook de betreffende konnektor-aansluitingen gegeven. In deze opzet kan de EPROMmer alleen worden gebruikt voor het programmeren van 2716's. Voor het type 2732 zou er nog veel meer veranderd moeten worden, en dat leek ons niet de moeite waard.

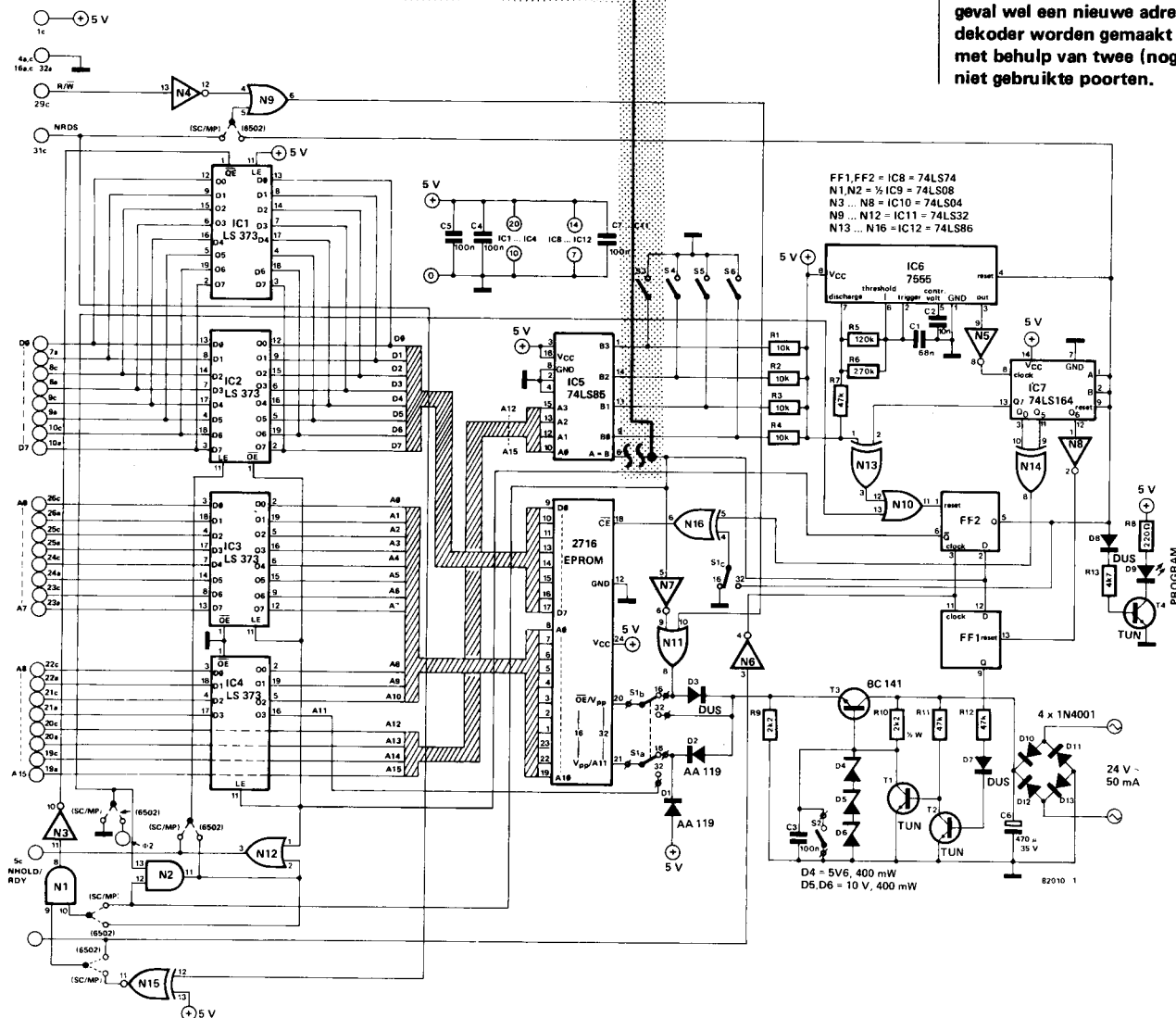
In figuur 2 is getekend hoe de twee kaarten met elkaar kunnen worden verbonden met behulp van twee 64-polige konnektors. Het verdient aanbeveling om voor de doorverbindingen tussen de "c"-aansluitingen geïsoleerde draad te gebruiken.

Aanvullende informatie over de EPROMmer kan men vinden in het EPROMmer-artikel uit Elektuur januari 1982. M

Tabel 1. Hier is aangegeven op welke adresbereiken men de EPROMmer-kaart kan leggen.



Figuur 1. Voor het programmeren van een 2716 kan men volstaan met de EPROMmer en een standaard-junior. Een interfacekaart is dus niet nodig. Op de EPROMmer moet in dat geval wel een nieuwe adresdekoder worden gemaakt met behulp van twee (nog) niet gebruikte poorten.

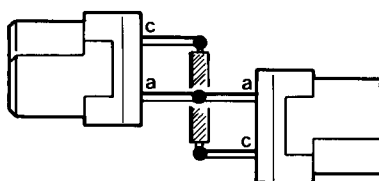


83119-1

Tabel 1.

Adres	Dekodering	
0800 ... 0FFF	K2-28c	K3-18c
0C00 ... 13FF	K3-18c	K4-17a
1000 ... 17FF	K4-17a	K5-15a
1400 ... 1BFF	K5-15a	K6-15c

2



83119-2

Figuur 2. Zo kunnen de twee kaarten met elkaar worden verbonden. Zorg dat er geen sluitingen ontstaan tussen de verbindingsdraden!