

6502

KENNER

KIM GEBRUIKERS CLUB
NEDERLAND

5de Jaargang nr. 16

Mei 1981

KIM GEBRUIKERS CLUB een club van 6502 gebruikers **NEDERLAND**

Het doel van de vereniging is het bevorderen van de kennis uitwisseling tussen de gebruikers van 6502 computers zoals KIM, SYM, JUNIOR, AIM65, System-65, Acorn, PET en CBM, Apple, Atari, ITT2020, PC100, Ohio Scientific Challengers, etc.

Dit doel wordt onder andere gerealiseerd door vijf maal per jaar een clubbijeenkomst te houden op de derde zaterdag van oneven maanden uitgezonderd de maand juli. Ook wordt vijf maal per jaar het huisorgaan " DE 6502 KENNER " gepubliceerd, welke verschijnt op de derde zaterdag van de maanden februari, mei, augustus, oktober en december.

Naast deze vaste activiteiten kunnen de leden gebruik maken van club faciliteiten zoals de KIM-CLUB-KIM , de KIM-CLUB-JUNIOR en de cassette bibliotheek.

De KIM gebruikers club Nederland is een volledig onafhankelijke vereniging met statuten en een bestuur. De club is ingeschreven bij de Kamer van Koophandel en Fabrieken voor Hollands Noorderkwartier te Alkmaar onder nummer 634305

De samenstelling van het bestuur is thans als volgt :

Dagelijks bestuur	Voorzitter	Anton Müller
	Secretaris	Ruud Uphoff
	Penningmeester	Ted Schouten
	Accomodatie	Riche van Steen
	Hardware	Peter Visser
	Ledenadministratie	Bob van de Oudewetering
	Organisatie	Rinus Vleesch Dubois
	Redactie	Hans Otten
	Software	Sebo Woldringh

Het adres van de vereniging is KIM GEBRUIKERS CLUB NEDERLAND
Voorburgpad 10
6843 EM ARNHEM
tel. 085 - 816935 op werkdagen van 19.00 tot 20.00

Penningmeester : giro nummer 3757649 tnv T.Schouten Junoplt 57 Haarlem
Lidmaatschapskosten f 40,- per kalenderjaar

DE 6502 KENNER

De 6502 KENNER is
een uitgave van
de KIM Gebruikers
club Nederland.

Adres voor het in-
zenden van en re-
akties op artike-
len voor de 6502

KENNER:
Redaktiesekretaris
Hans Otten
Ottoborrengoed 33
3871 MJ HOEVELAKEN
Tel.: 03495-36551
(alleen op werk-
dagen tussen 19.00
en 20.00 uur).

Redactie 6502 KENNER:
Anton Müller
Hans Otten
Willem van Pelt

Geheel of gedeelte-
lijke overname van
de inhoud van de
6502 KENNER zonder
toestemming van het
bestuur is verboden.
Toepassen van gepu-
bliceerde programma's,
hardware etc. is al-
leen voor persoonlijk
gebruik toegestaan.

COPYRIGHT © 1981
KIM Gebruikers club
Nederland.

Verschijnt vijf maal
per jaar.

INHOUDSOPGAVE VAN DE 6502 KENNER nr. 16 mei 1981

1. Inhoudsopgave 6502 KENNER mei 1981 .
2. Van het bestuur . Anton Mueller .
3. Van de redactie (cassette bibliotheek)
Hans Otten .
4. Clubbijeenkomst 17 januari 1981
Anton Mueller .
6. Verkeersregeling . D.J. Draal .
14. Ascii toetsenbord encoder .
Hans Otten .
24. Ervaringen met de JUNIOR .
W. van Pelt en J. van Sprang .
26. Automatische register uitlezing voor de JUNIOR.
30. Shooting stars voor de SYM.
J.F. van Woudenberg
39. Browse voor de JUNIOR .
43. Het slangetje en de 6532.
J.H. Vennimmen
45. Vraag en aanbod .
46. CBM geheugen uitbreiding.
R. Uphoff
48. Agenda
49. Patches op Microsoft Basic . Wim Blonk

Tijdens de afgelopen clubbijeenkomst/buitengewone algemene ledenvergadering ben ik officieel verkozen tot voorzitter van onze vereniging. Samen met de overige bestuursleden hoop ik er het beste van te maken. Onze gedachten gaan uit naar verdere groei van de vereniging, o.a. door nauwere kontakten met andere verenigingen en tijdschriften die zich bewegen op het gebied van de 6502 micro-processoren. Verder zal meer aandacht worden besteed aan de inhoud van de 6502 KENNER en de clubbijeenkomsten van een zodanig programma te voorzien dat deze voor een ieder interessant zijn, d.w.z. zowel voor de beginners als ook voor de gevorderden. U kunt ons daarbij helpen. Immers is de 6502 KENNER een weerspiegeling van wat er onder de leden leeft. Als zowel de beginners als de gevorderden ons van kopij voorzien dan kan de redactie deze zodanig over de komende 6502 KENNERS verdelen dat de zo broodnodige gevarieerdheid gewaarborgd is. Wat mij de laatste tijd wel opvalt is dat er nauwelijks "KIM"-artikelen in de KIM- en/of 6502 KENNER hebben gestaan. Het wordt hoog tijd dat onze KIM-broeders dus weer eens voor wat kopij zorgen en die is er echt wel als ik zo af en toe mijn oor eens te luisteren leg op de clubbijeenkomsten. Bovendien is m.i.v. dit nummer het aantal redactionele pagina's opgevoerd van 40 naar 48 teneinde U nog meer waar voor Uw geld te geven. Tijdens de clubbijeenkomst op 19 september a.s. willen we de stunt uithalen een twintigtal JUNIOR computers (die door de leden zelf meegebracht moeten worden) via een current loop verbinding met z'n allen te koppelen aan de KIM-club-KIM die dan als master funktioneert. Op de KIM draait dan een aangepaste versie van tiny BASIC waarvan de resultaten zichtbaar kunnen worden gemaakt op de JUNIOR's, ieder op hun eigen via de toetsenboardjes te regelen snelheid en condities.

Als alles volgens plan verloopt treft U bij dit nummer ook het financiële jaarverslag over 1980 aan. Zoals daaruit blijkt gaat het goed met onze vereniging. Zo goed zelfs, dat wij in januari j.l. hebben besloten een elektrische schrijfmachine aan te schaffen voor de redactie, alsmede hebben besloten tot aanschaf van een club-JUNIOR. Deze laatste komt ons vooral van pas bij het uittesten van ingezonden handgeschreven programmaatjes voor de JUNIOR. Ook kunnen onze software- en hardware adviseurs door bestudering van de JUNIOR de JUNIOR-leden beter van advies dienen.

Tot slot nog even iets over het "markt"-wezen. Zo rond een uur of drie in de middag tijdens clubbijeenkomsten, houden we een half uurtje markt. Over het algemeen wordt dit half uurtje gevuld door een aantal clubleden die zich beroepsmatig met de micro-elektronica bezig houden. De markt staat echter open voor alle clubleden en het zou ontzettend fijn zijn als U zo af en toe Uw overvloedige hardware te koop kwam aanbieden. Overigens kunt U daarvoor ook gebruik maken van de rubriek "Vraag en aanbod" in de 6502 KENNER. Alles wat maar even met micro's te maken heeft is welkom: micro's, terminals, onderdelen, ponsbandlezers, ponsbandponzers, digitale cassette recorders, floppy diskdrives, boeken, tijdschriften alsmede zelf ontwikkelde programmatuur. Het adverteren in de rubriek vraag en aanbod is gratis voor clubleden! Uiteraard kan de KIM-club c.q. het bestuur niet verantwoordelijk of aansprakelijk worden gesteld voor de goede werking van het aangeboden goed, defecten en/of garantie, plaats van herkomst en eventuele gevolgen en afspraken betreffende levering, levertijd en betaling. Waarschijnlijk is dit voor U vanzelfsprekend.

Met vriendelijke groeten: Anton Müller

Allereerst een goed bericht : Elektuur heeft de langbeloofde en noodzakelijke uitbreiding voor de JUNIOR gepubliceerd in het aprilnummer. Helaas zonder print ontwerp maar dat komt wel. De JUNIOR komt nu op het niveau van de KIM met een terminal en een cassette interface en geheugenuitbreiding. De cassette interface kan het KIM formaat aan en de voorbeelden in het artikel wijzen erop dat de JUNIOR cassettes volledig uitwisselbaar worden met de KIM tapes. Voor de uitwisseling in onze club een goed uitgangspunt. Overigens, de KIM gebruikers werken bijna allemaal met Hypertape wat dezelfde snelheid is als het Elektuur artikel voorspelt. De software ontbreekt ook nog, maar komt ongetwijfeld in Boek 3. Misschien ten overvloede, de ontwikkelingen van Elektuur wijzen erop dat het niet zinvol is zelf eigen oplossingen te bedenken. Eigen oplossingen isoleren U van de medegebruikers en in de KIM club hebben we de afgelopen jaren gemerkt dat contact met gebruikers van identieke systemen enorm vruchtbaar is . Hoed U voor afwijkende systemen !

De redactie heeft besloten de publicatie van software te ondersteunen met een cassette service. Deze service komt in de plaats van de cassette bibliotheek. Omdat de redactie leden de meeste software die wordt gepubliceerd in bezit heeft kunnen we de cassette software service het best uitvoeren.

De voorwaarden zijn als volgt :

Ieder kan een cassette met software uit de 6502 kenner bestellen.

De prijs van een cassette is f 12,- , dit is om de kosten van cassette, verpakking verzend- en administratie kosten te dekken. Betalen gebeurt door bij de schriftelijke bestelling een getekende girobetaalkaart of bankcheque (Euro of 'groene') met het bedrag en de handtekening ingevuld mee te sturen. Verder oningevuld laten. De bestelling stuurt U naar het redactie secretariaat (H.J.C.Otten, zie blz 1). Andere wijzen van bestellen bestaan niet.

De levertijd zal ongeveer 2 of 3 weken duren.

Er zijn op het moment twee cassettes beschikbaar :

- | | |
|---------------------------|--|
| 1. KIM met o.a. | 2. Junior (KIM of Kanssa City tape !) |
| Verkeerslichten 6502 K 16 | Lotto 6502 K 17 |
| Talenstudiehulp KIM K 13 | Moonlander K 15 |
| Target 1 + 2 KIM K 8 | Browse K 15 |
| Microchess openingen K 8 | Aut reg uitl K 15 |
| KIM schaakprogramma K 11 | Stopw K 15 |
| Supertape K 12 | Hex displayerK 14 |
| Reactiesnelh meter K 12 | |
| Microchess versnellenK 12 | |

Geeft U bij het bestellen het cassette nummer op en voor de junior het tape type.

Vaak zijn programma's in assembler source op de band gezet (Micro Ade) en in de uit te voeren machine code . Een inhoudsopgave staat vermeld in de cassette .



CLUB BIJeenkomst

Samenvatting van de KIM-clubbijeenkomst d.d. 17 januari 1981.

De gehele dag stond in het kader van de programmeertaal FORTH. De computer taal FORTH is uitgevonden door Charles Moore, als een uitbreidbaar, multi-level omgeving bevattende elementen van een operating systeem, een machine monitor en faciliteiten voor programma ontwikkeling en testen. De lezing werd gegeven door Dr. Nieuwenhuizen van de Rijks sterrenwacht in Utrecht, waarvoor langs deze weg nogmaals onze hartelijke dank.

Het FORTH systeem bestaat uit een aantal elementen van hardware structuren. Verder bestaat het systeem uit een tekst interpreter, een opdracht interpreter, een woordenboek, een stuk geheugen en een stapel (stack) alsmede een opbergkast voor geneste structuren. Woorden in FORTH zijn onderverdeeld in andere woorden die uiteindelijk resulteren in elementen. De FORTH syntax, is een opeenvolging van woorden, gescheiden door een spatie, mogelijk beëindigd met een carriage return. Een woord kan een ASCII character bevatten, behalve spaties, CR, of BS. Het uniek zijn: woorden worden van alle andere onderscheiden door hun lengte (dus het aantal characters) en de eerste drie characters (drie of meer, afhankelijk van de implementatie). Voorbeelden van FORTH woorden: FORTH (wordt uitgevoerd); 123 (wordt omgezet naar binair en opgeborgen op de stack); . (opdracht: afdrukken van een getal van de stack en dit weggooien). OK = opdracht succesvol uitgevoerd.

Voorbeeld: 1 2 3 CR OK. Op de stack staat nu van boven naar onderen: 3 2 1. .CR print dan 3, nog een .CR print dan 2 en nog een .CR print dan 1. Doen we nog een .CR dan krijgen we een boodschap dat de stack leeg is. Het verwijderen van getallen van de stack is essentieel in FORTH. Wil je een getal na dat je het van de stack gehaald hebt bewaren dan moet je het eerst dupliceren (DUP). Operatoren en operanden: Operatoren nemen hun inputs van de stack en laten hun outputs op de stack. 6 3 + CR (Reversed Polish Notation) geeft aan dat we 6 en 3 bij elkaar willen optellen. 6 komt op de stack, 3 komt op de stack, + is een operator die zijn inputs van de stack haalt en zijn outputs, in dit geval 9 op de stack zet. Overigens is + in eerste instantie een woord en wordt pas later gepromoveerd tot operator. Zoals al eerder genoemd werkt FORTH volgens de RPN of te wel post fix notatie (ordering) in operanden en operatoren. Degenen die een Hewlett Packard calculator hebben zullen hier wel geen problemen mee hebben. Nog enige voorbeelden: 1 2 + . CR 3 OK; 7 2 x 1 + . CR 15 OK; 3 2 - . CR 1 OK. Volgens Dr. Nieuwenhuizen is aangetoond dat babies ook zo rekenen doch dat ze dit later door hun onderwijzers wordt afgeleerd! Het FORTH woordenboek (dictionary) bevat de namen van de woorden benevens een gecompileerde vorm van hun definities in de volgorde waarin zij werden gedefiniëerd. Definitie van een nieuw woord: :nieuw woord definitie; (eerder gedefiniëerde woorden of getallen). Voorbeelden: 8X wordt gedefiniëerd als: :8X 2X 2X 2X; We kunnen 8X nu bijvoorbeeld gebruiken door 7 8X . CR en krijgen dan 56 OK. Nog een voorbeeld: :% 100 */; gebruik: 200 5% . CR geeft als resultaat 10 OK.

Overigens is Dr. Nieuwenhuizen van mening dat grote computers maar veel te duur zijn en de manier van werken er mee veel te omslachtig is. Hij voelt veel meer voor micro's die qua prijs in de orde van grootte zo tussen de 5 à 10.000 gulden liggen, zoals de KIM, PET en APPLE e.a. In FORTH kun je overal bij en je hoeft niet eerst de moeilijke salg te maken van editen, vertalen, link editen, testen, debuggen etc. Ook in FORTH is het mogelijk om gestructureerde programma's te schrijven. Control structures als IF, ELSE, THEN, DO-WHILE en CASE zijn standaard aanwezig. Dr. Nieuwenhuizen ziet FORTH helemaal zitten, in combinatie met LISP en ADA. Tot zover FORTH. Ik hoop dat U van deze summier samenvatting e.e.a. heeft begrepen. Overigens kan ik U mededelen dat ik zelf inmiddels de FORTH compiler heb besteld bij de FORTH Interest Group, P.O. Box 1105, te San Carlos CA 94070 in de USA door overmaking van \$20. Je krijgt dan een listing van FORTH voor de 6502 die je zelf moet aanpassen en een handleiding. Voor de mensen die een AIM hebben is er een kant en klare FORTH beschikbaar voor + f. 250,- bij FAMATRA.

Parallel aan de demonstratie van FORTH was er een lezing door Frans van de Berg over PASCAL op SYSTEM/65. Helaas door de overweldigende belangstelling voor FORTH is Frans z'n lezing een beetje in de mist geraakt, in zoverre dat er slechts vier belangstellenden waren. Ik hoop dat die vier er het e.e.a. van hebben opgestoken. Frans weet intussen aardig wat van PASCAL af. Uitgaande van de compiler van Mark Rustad die destijds door Gerard van der Grinten werd gepresenteerd, heeft Frans hier het nodige aan verbeterd, c.q. geoptimaliseerd en toegevoegd, zoals bijv. een otherwise clause voor de case, die er oorspronkelijk niet in zat, zodat als je niet zeker van de cases was maar beter geneste IF's kon gebruiken. Ook heeft Frans de interpreter compleet herschreven. Gerard van der Grinten heeft destijds de interpreter vertaald of te wel geconverteerd van de 6800 naar de 6502, zonder gebruik te maken van de eigenschappen van de 6502 (13 addressing modes, instructie repertoire etc.). Frans heeft dit wel gedaan. Bovendien heeft hij met een hardware monitor bekeken welke de meest uitgevoerde PASCAL P-code instructies zijn en afhankelijk daarvan ook de interpreter geoptimaliseerd. Kortom een hoop werk. De PASCAL compiler van Frans van der Berg is te koop voor SYSTEM/65 bij DATEQ in Almere en voor overige gebruikers bij Ingenieursbureau Schröder te Eindhoven. Siep de Vries heeft destijds eveneens deze compiler en interpreter aangepast voor de AIM65 en deze is te koop bij FAMATRA. De prijs is ongeveer f 800,- à f. 900,-. De hobbyist is echter beter af. Deze krijgt voor f. 100,- het oorspronkelijke werk van Gerard van der Grinten in handen, "AS IS", hetgeen betekent dat dit werkt op een KIM met minimaal 32K RAM. De aanvullingen voor de cassette i/o zijn te bekomen bij Willem van Gelderen, die er eigenlijk helemaal niets voor rekent. Daarbij wordt er wel vauit gegaan dat je MICRO-ADE van Peter Jennings hebt aangeschaft, om je PASCAL sources aan te maken, waarvoor Willem van Gelderen dan weer een aanpassing heeft dat dit op record niveau gebeurd.

Verder hebben we een aardige demonstratie gehad van de heer Dral uit Den Helder over een maquette van een verkeersplein met stoplichten dat aan een KIM hing. Onder de weg waren metaaldetectoren ingebouwd die constateerden of er soms een auto o.i.d. stond te wachten. Dit systeem sloeg zo aan, dat hij het zonder meer aan een aantal rijsschoolhouders in zijn woonplaats kwijt kon.

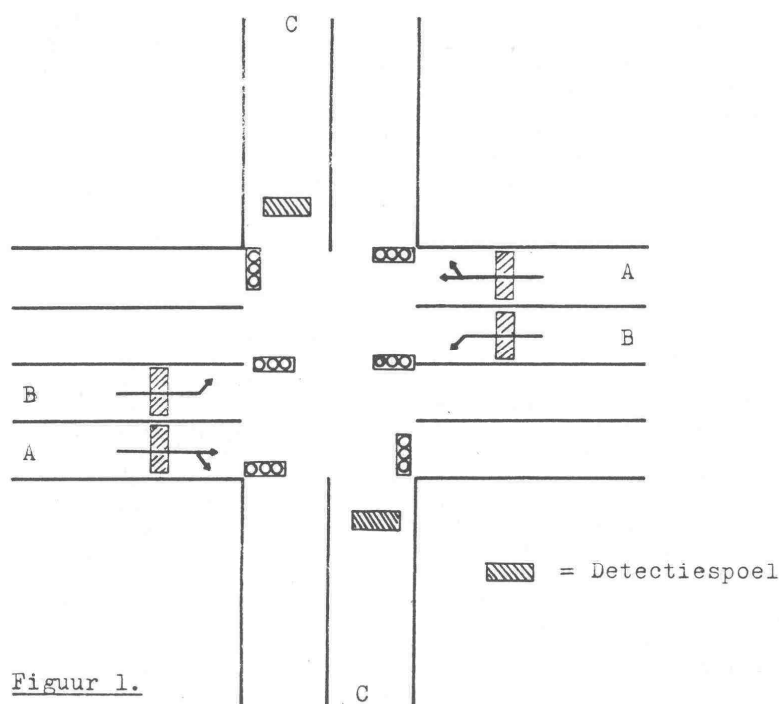
Al met al een bijzonder geslaagde dag.

Anton Müller

VERKEERS REGELING.

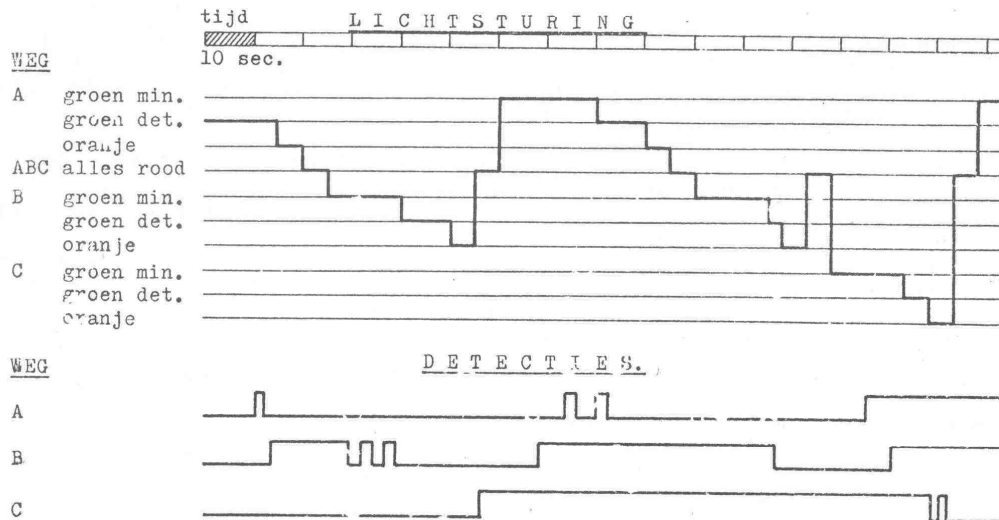
Deze verkeersafhankelijke regeling is ontworpen voor het kruispunt zoals voorgesteld in fig. 1, bestaande uit een hoofdweg (A), afslaandverkeer (B) en zijweg (C). De detectiespoelen bevinden zich onder het wegdek.

D.J.Dral
Ijsselstraat 15
1784 VN Den Helder



Figuur 1.

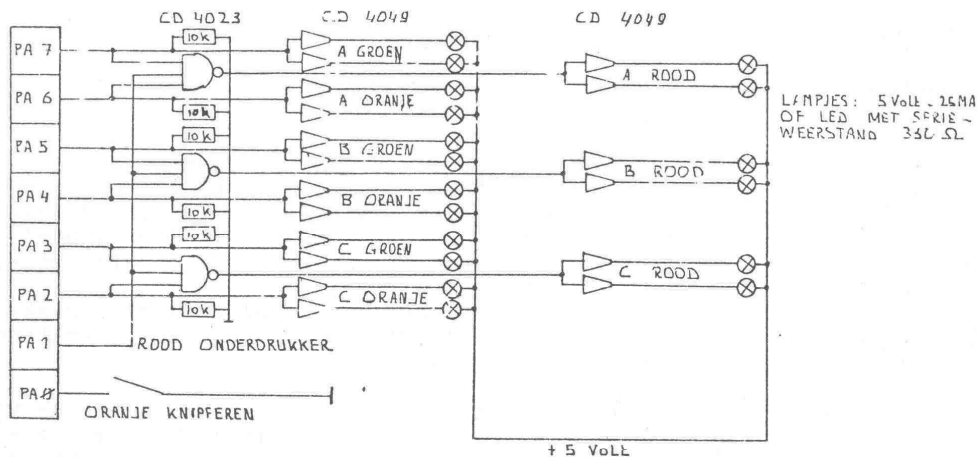
De regeling is als volgt: Buiten de verkeersdrukte ('s nachts) knippen de oranje lampen om beurten. Indien de installatie wordt aangezet dan branden de groene lampen op de hoofdweg continu totdat er een detectie op één der wegen B en/of C wordt waargenomen. Het licht op de hoofdweg springt op oranje en daarna op rood. Daarna springt het licht op de gedetecteerde weg op groen gedurende een ingestelde tijd, verhoogd met een tijdseenheid afhankelijk van de in de eerste periode aantal gepasseerde voertuigen. Hierna herhaalt zich de gehele cyclus, ook voor de hoofdweg; zie fig. 2 (tyddiagram).



FIGUUR 2

Dit voorbeeld dient alleen om te laten zien hoe het geheel werkt; de tijden, zoals weergegeven, zijn allen variabel.

Om het programmeren te vergemakkelijken zijn de lichten via een stukje hardware aangesloten (fig. 3). Hierdoor is het mogelijk om de lampen via één datapoort allemaal te sturen. De oranje en groene signalen (van één weg) worden aan een EN poort toegevoerd die hierdoor de rode lamp stuurt. De roodonderdrukker zorgt ervoor dat indien de oranje lampen knipperen, de rode lamp niet mee gaat branden. Het voordeel van dit systeem is, dat als de computer wordt aanzet (het programma is dan nog niet geladen) de rode lampen allemaal gaan branden.

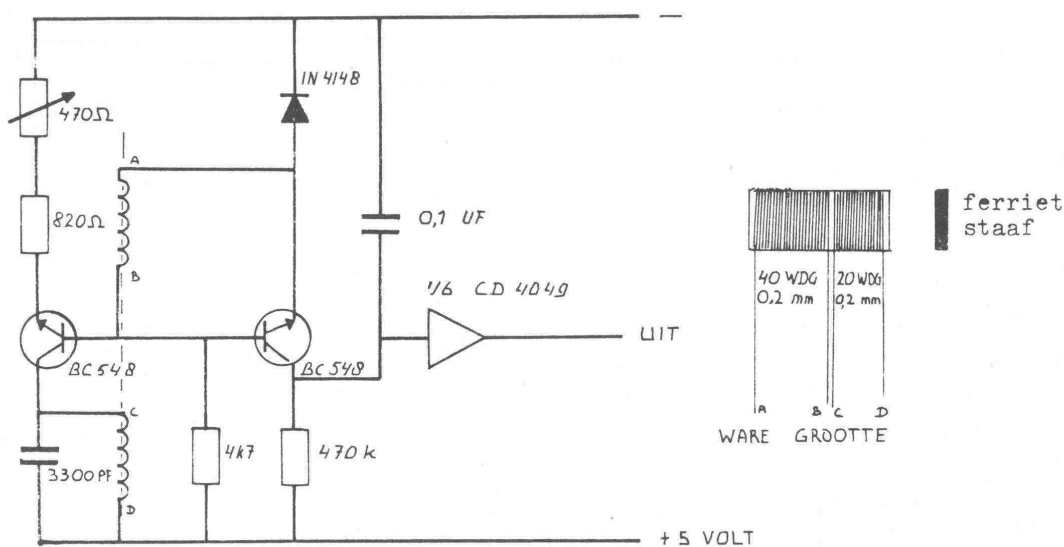


FIGUUR 3

Inplaats van gloeilampjes, 5 volt- 30 mA, kunnen ook leds worden toegepast; men dient dan wel een serieweerstand van 330 ohm op te nemen. Als de lampjes, of leds, tegen de nul moeten branden dan kunnen de buffers CD 4049 worden vervangen door inverter- buffers CD 4050.

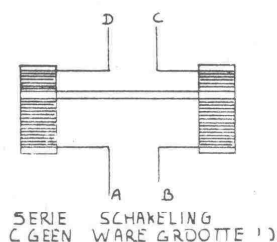
De detectie van de voertuigen worden waargenomen door detectiespoelen, welke onder het wegdek zijn gemonteerd. Zorg er in ieder geval voor, dat de bevestiging niet met behulp van metalen delen gebeurd, aangezien de schakeling hiervoor zeer gevoelig is.

Het schema voor de detectie is weergegeven in figuur 4. In de L-C kring wordt een frequentie opgewekt van ca. 500 khz. Zodra het veld rond de ferrietstaaf, waar de windingen omheen zijn gewikkeld, wordt verstoord door een metalen voorwerp, dan wordt de uitgang 'hoog'.



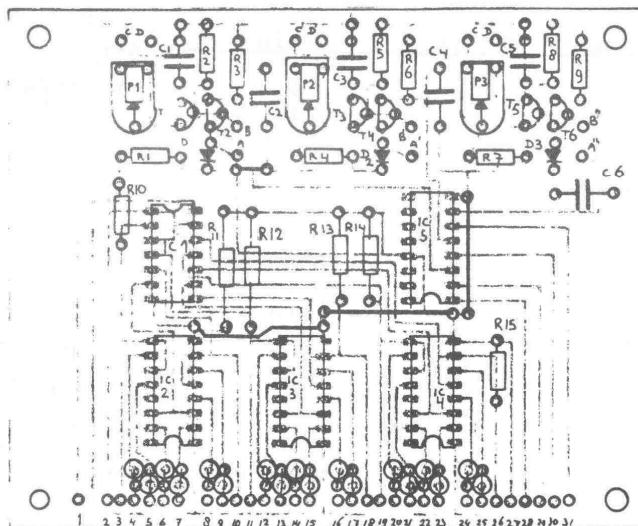
Figuur 4

De schakeling is erop berekend, dat twee van deze spoelen in serie zijn geschakeld, voor twee wegen bijvoorbeeld A en A' of B en B' enz. Hieronder is geschetst hoe de schakeling in serie moet worden gemonteerd. Alleen op deze manier werkt het geheel; de polariteit van de spoelen is erg belangrijk.



Indien één spoel in de schakeling wordt toegepast dient men de waarde van de vaste weerstand van 470 ohm aan te passen.

Van het geheel is een print ontworpen waarop kan worden aangesloten voor 2 maal drie wegen detectie spoelen en in totaal 6 stoplichten; dus zoals geschetst in figuur 1.

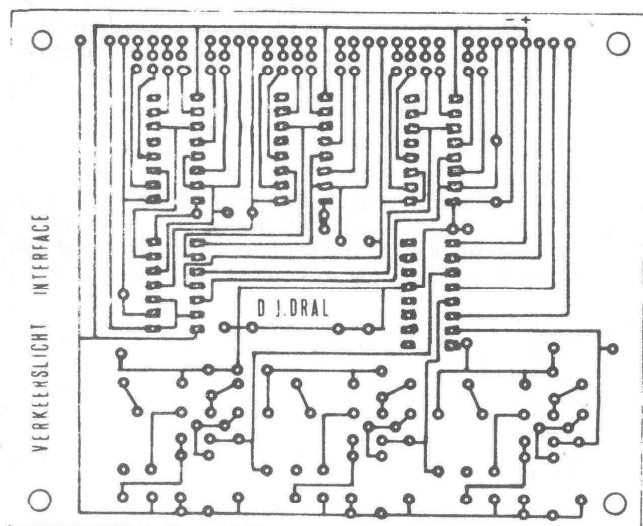


ONDERDELENLIJST

R1,4,7	820 ohm
R2,5,8	4k7
R3,6,9	470k
R10 t/m 15	10k
R16 t/m 33	330 ohm
P1,2,3	470 ohm
D1,2,3	1N4148
C1,3,5	3300 pf
C2,4,6	100 nf
T1 t/m 6	BC548
IC 1	CD 4023
IC 2,3,4,5	CD 4049

COMPONENTEN ZIJDE

Alle genoemde weerstanden zijn 1/8 watt.
De weerstanden R16 t/m R 33 zijn voorschakelweerstanden indien LED's worden toegepast en zijn dus bij toepassing van gloeilampjes 5 Volt- 30 mA overbodig (doorverbinden op de print).



PRINTZIJDE

Aansluitingen van de print

Nr	kleur	Weg	I/O	KIM	DIV.
1			I	PA 1	+ 5 Volt
2			I	PA 2	Rood onderdr.
3	oranje	C	I		
4	oranje	C	O		
5	oranje	C	O		
6	rood	B	O		
7	rood	B	O		
8	oranje	B	O		
9	oranje	B	O		
10	oranje	B	I	PA 4	
11	groen	B	I	PA 5	
12	groen	B	O		
13	groen	B	O		
14	rood	C	O		
15	rood	C	O		
16	groen	C	O		
17	groen	C	O		
18	groen	C	I	PA 3	
19	groen	A	I	PA 7	
20	groen	A	O		
21	groen	A	O		
22	rood	A	O		
23	rood	A	O		
24	oranje	A	O		
25	oranje	A	O		
26	oranje	A	I	PA 6	
27					- (massa)
28					+ 5 Volt
29		A	O	PB 0	detectie
30		B	O	PB 1	detectie
31		C	O	PB 2	detectie

Het Programma.

Het programma wordt gestart op adres 0200.

In het eerste deel van het programma wordt de data gelezen en in de Z page gezet. De variabelen, tijd, uitsturing e.d., kan men veranderen door een andere waarde op de betreffende adressen te zetten. Het programma dient daarna wel te worden gestart op 020A.

De belangrijkste tijden zijn:

Roodtijd	0008
Oranjetijd	0009
Oranjeknippertijd	000A
Min. groentijd A	000B
Min. groentijd B	000C
Min. groentijd C	000D

```
0010: *****
0020: VERKEERSAFHANKELYKE REGELING
0030: AUTEUR: D.J.DRAL
0040: IJSSELSTRAAT 15
0050: 1784 UN DEN HELDER
0060: *****
```

```
0070:
0080:
0090: 0200 STOPL ORG $0200
0100: 0200 GROEN + $0000
0110: 0200 ORABC + $0003
0120: 0200 ORAC + $0006
0130: 0200 TRO + $0008
0140: 0200 TOR + $0009
0150: 0200 TORK + $000A
0160: 0200 MINGRO + $000B
0170: 0200 AANUX + $000E
0180: 0200 AANVD + $0011
0190: 0200 AANVDB + $0012
0200: 0200 AANVDC + $0013
0210: 0200 VERTR + $0014
0220: 0200 VERTRB + $0015
0230: 0200 VERTRC + $0016
0240: 0200 NEND + $0017
0250: 0200 NENT + $0018
0260: 0200 DREGA + $1700
0270: 0200 DDREGA + $1701
0280: 0200 DREGB + $1702
0290: 0200 DDREGB + $1703
0300:
0310: 0200 A2 16 LDXIM $16 VUL ZPAGE MET DATA
0320: 0202 B0 B0 02 VUL LDAX DATA
0330: 0205 95 00 STAZX GROEN
0340: 0207 0A DEX
0350: 0208 10 F8 SPL VUL ALLE DATA GEHAD?
0360: 020A A9 FE LDAIM $FE
0370: 020C 8D 01 17 STA DDREGA
0380: 020F A9 B8 LDAIM $B8
0390: 0211 8D 03 17 STA DDREGB
0400: 0214 20 0A 02 JSR DANA DAG OF NACHT
0410: 0217 A2 00 NIEUW LDXIM $00
0420: 0219 A9 FF DOOR LDAIM $FF ROOD
0430: 021B 8D 00 17 STA DREGA
0440: 021E A5 00 LDAX TRO TYD ROOD
0450: 0220 20 8F 02 JSR DREMP
0460: 0223 B5 00 LDAXX GROEN
0470: 0225 8D 06 17 STA DREGA
0480: 0228 B5 00 LDAXX MINGRO MINIMUM GROENTYD
0490: 022A 20 0F 02 JSR DREMP
0500: 022D B5 14 HOOFD LDAXX VERTR DETECTIE VERTRAGING
0510: 022F 20 8F 02 JSR DREMP
0520: 0232 20 5A 02 JSR DANA DAG OF NACHT
0530: 0235 A9 00 LDAIM $00
0540: 0237 95 14 STAZX VERTR RESET VERTRAGING
0550: 0239 95 11 STAZX AANVD
0560: 023B A5 15 LDAX VERTRB
```

0570:	023D	05	1C		ORAZ	VERTRO	
0580:	023F	D0	04		BNE	ANUR	ANDERE DETECTIE?
0590:	0241	E0	00		CPXIM	#00	
0600:	0243	F0	E8		BEQ	HOOFD	
0610:	0245	05	03	ANUR	LDZK	ORABC	ORANJE
0620:	0247	0D	00	17	STA	DREGA	
0630:	024A	A5	09		LDZ	TOR	ORANJE TYD
0640:	024C	20	0F	02	JSR	DREMP	
0650:	024F	E8		NUL	INX		
0660:	0250	E0	03		CPXIM	#03	
0670:	0252	F0	03		BEQ	NIEUW	ALLE RICHTINGEN GEHAD ?
0680:	0254	05	14		LDZK	VERTR	ANDERE DETECTIE ?
0690:	0256	F0	F7		BEQ	NUL	
0700:	0258	D0	0F		BNE	DOOR	
0710:	025A	AD	00	17	LDA	DREGA	
0720:	025D	29	01		ANDIM	#01	
0730:	025F	F0	18		BEQ	DAG	IS HET DAG ?
0740:	0261	A2	00	NACHT	LDXIM	#00	
0750:	0263	05	06	KNIP	LDZK	ORAC	ORANJE
0760:	0265	0D	00	17	STA	DREGA	
0770:	0268	A5	0A		LDZ	TORK	KNIPPEREN
0780:	026A	20	7A	02	JSR	TYD	
0790:	026D	E8			INX		
0800:	026E	E0	02		CPXIM	#02	
0810:	0270	D0	F1		BNE	KNIP	EN WISSEL ORANJE
0820:	0272	AD	00	17	LDA	DREGA	
0830:	0275	29	01		ANDIM	#01	
0840:	0277	D0	E8		BNE	NACHT	IS HET DAG ?
0850:	0279	00		DAG	RTS		
0860:	027A	05	10	TYD	STA	MENT	TYD MAAL 1/4 SEC.
0870:	027C	0A			TXA		
0880:	027D	40			PHA		X OP STACK
0890:	027E	A2	D0	WEERA	LDXIM	#00	TYDINSTELLING
0900:	0280	A0	FF	WEERX	LDXIM	#FF	1/4 SECONDE
0910:	0282	00		WEERY	DEY		
0920:	0283	D0	F0		BNE	WEERY	
0930:	0285	0A			DEX		
0940:	0286	D0	F8		BNE	WEERX	
0950:	0288	06	18		DECZ	MENT	
0960:	028A	10	F2		BPL	WEERA	
0970:	028C	68			FLA		X VAN STACK
0980:	028D	AA			TAX		
0990:	028E	60			RTS		
1000:	028F	05	17	DREMP	STA	MENT	DETECTIE ROUTINE
1010:	0291	0A			TXA		
1020:	0292	40			PHA		X OP STACK
1030:	0293	A2	00	WEERZ	LDXIM	#00	
1040:	0295	AD	02	17	LDA	DREGB	
1050:	0298	35	0E	WEERD	ANDZK	ANUX	
1060:	029A	F0	0F		BEQ	NIET	GEEN AANVRAAG
1070:	029C	A9	01		LDXIM	#01	
1080:	029E	D5	11		CMPZK	ANUD	
1090:	02A0	F0	0B		BEQ	REEDS	
1100:	02A2	10			CLC		
1110:	02A3	05	14		LDZK	VERTR	
1120:	02A5	69	04		ADCM	#04	1 SECONDE VERTR. PER AUTO

```

1130: 02A7 95 14      STAZX VERTR  TOTALE TYDVERTRAGING
1140: 02A9 A3 01      LDRAIM #01
1150: 02AB 95 11      NIET STAZX AANUD  RESET AANVRAAG
1160: 02AD E8         REEDS INX
1170: 02AE E0 03      CPXIM #03
1180: 02B0 D0 E3      BNE WEERD
1190: 02B2 A9 00      LDRAIM #00
1200: 02B4 20 7A 02   JSR TYD
1210: 02B7 C6 17      DEC MEMD
1220: 02B9 10 D8      BPL WEERZ
1230: 02BB 68         PLA X VAN STACK
1240: 02BD AA         TAX
1250: 02BD 60         RTS
1260: 02BE 7F      DATA = $7F
1270: 02BF DF      = $DF
1280: 02C0 F6      = $F6
1290: 02C1 BF      = $BF
1300: 02C2 EF      = $EF
1310: 02C3 FA      = $FA
1320: 02C4 B9      = $B9
1330: 02C5 ED      = $ED
1340: 02C6 06      = $06
1350: 02C7 08      = $08
1360: 02C8 02      = $02
1370: 02C9 20      = $20
1380: 02CA 10      = $10
1390: 02CB 10      = $10
1400: 02CC 01      = $01
1410: 02CD 02      = $02
1420: 02CE 04      = $04
1430: 02CF 00      = $00
1440: 02D0 00      = $00
1450: 02D1 00      = $00
1460: 02D2 00      = $00
1470: 02D3 00      = $00
1480: 02D4 00      = $00

```

SYMBOL TABLE 3000 30F6

AANUD 0011	AANUD8 0012	AANUDC 0013	AANVR 0245
AANUX 000E	DAG 0279	DANA 025A	DATA 02BE
DDREGA 1701	DDREGB 1703	DOOR 0219	DREGA 1700
DREGB 1702	DREMF 028F	GROEN 0000	HOOFD 022D
KHIP 0263	MEMD 0017	MENT 0018	NINGRO 000B
NACHT 0261	NIET 02AB	NIEUW 0217	NUL 024F
ORABC 0003	ORAC 0006	REEDS 02AD	STOPL 0200
TORK 000A	TOR 0009	TRO 0008	TYD 027A
VERTR 0014	VERTRB 0015	VERTRC 0016	UUL 0202
WEERA 027E	WEERD 0295	WEERX 0280	WEERY 0282
WEERZ 0293			

ASCII Toetsenbord

ASCII toetsenbord encoder.

H.J.C. Otten

Om opdrachten en gegevens in een computer te brengen, is een toetsenbord zeer geschikt. De ingetypte karakters worden tevens naar een printer of een videodisplay gestuurd, zodat de programmeur de invoer kan volgen. Het indrukken van een toets moet voor de computer vertaald worden in een digitale code. De meest gebruikte code is de ASCII code. De vertaalschakeling, de encoder, van toetsenbord naar de volledige ASCII code wordt hier beschreven.

Inleiding

Een ASCII toetsenbord is onmisbaar om informatie, bestaande uit gegevens en stuurcommando's, aan een computer toe te voeren. Deze ASCII toetsenborden zijn geheel compleet te koop, inclusief de encoderschakeling, maar ook worden er losse toetsenborden aangeboden. Men kan ook zelf een toetsenbord opbouwen uit losse keyboard schakelaars. Om van het indrukken van een toets tot de 7 bits ASCII code te komen is een encoder schakeling nodig. Hiertoe zijn er een aantal mogelijkheden:

1. De computer kan zelf aftasten van het toetsenbord verzorgen met een PIA (Peripheral Interface Adapter) op een wijze die sterk lijkt op de hier beschreven schakeling.
Deze mogelijkheid vervalt als het toetsenbord deel moet uitmaken van een teletyp-achtige in- en uitvoereenheid, die op de seriële teletyp in- en uitgang van de computer wordt aangesloten.
2. Er bestaan IC's die het opwekken van de ASCII code geheel verzorgen. Deze IC's hebben als nadeel dat ze niet zo makkelijk te verkrijgen en vrij prijzig zijn. Verder vereisen ze een negatieve voedingsspanning van -12V extra.
3. Een uit TTL IC's opgebouwde encoder, zoals hierna beschreven wordt. De onderdelen hiervoor zijn goedkoop en makkelijk te verkrijgen. De flexibele opzet maakt uitbreidingen en wijzigingen eenvoudig. Zo is de encoder ook geschikt om andere codes, zoals de Baudot code op te wekken.

De ASCII code

De ASCII code (American Standard For Coded Information Interchange) is een veel gebruikte digitale code in computers. Ook de KIM-1 gebruikt deze code als er bijvoorbeeld informatie over de teletyp in- en uitgang wordt verstuurd. De ASCII code bestaat uit 7 bits, er zijn zo 128 combinaties mogelijk van enen en nullen. In tabel 1 is te zien hoe de verschillende karakters in de ASCII code voorgesteld worden. Allereerst is de eigenlijke ASCII code, de zeven digitale getallen, te zien, van links naar rechts het meest significante bit, b_6 , tot het laagste bit, b_0 . Dit binaire getal is in de kolom ernaast omgezet in het hexa-decimale getal en daarnaast in het decimale getal.

De volgende kolom vermeldt of het een normaal toetsenkarakter is of met een speciale manipulatie, waarover later meer, wordt opgewekt. De naam van het ASCII karakter is vervolgens vermeld met daarachter eventuele opmerkingen.

De 128 mogelijke combinaties zijn ruim voldoende om behalve de letters van het alfabet, de getallen 0 tot en met 9 en vele leestekens en speciale karakters, ook een groot aantal stuurtekens mogelijk te maken. Velen daarvan zijn bedoeld om een teletype, een elektrische schrijfmachine die ook door de computer kan worden bestuurd, te bedienen, zoals carriage return (wagen terug), line feed (volgende regel) en space (spatie). Aan de stuurtekens wordt in ieder computerprogramma een andere betekenis gehecht, zodat een ASCII toetsenbord ze allemaal moet kunnen opwekken.

Het ASCII toetsenbord

Om ASCII karakters in de computer te voeren, wordt een toetsenbord gebruikt. Indrukken van een toets heeft als gevolg dat op de zeven uitgangen de bij de toets behorende ASCII code verschijnt, terwijl op een achtste uitgang een strobesignaal, een digitale puls, verschijnt om aan te geven dat er een toets is ingedrukt.

Om de gehele ASCII code op deze wijze op te wekken zouden er 128 toetsen nodig zijn. Omdat dit erg onpraktisch is, worden de meest gebruikte karakters als toets uitgevoerd zoals letters, cijfers en leestekens en wordt aan de meeste toetsen een dubbelfunctie en soms drie functies gegeven. Bij een gewone schrijfmachine hebben veel toetsen ook een dubbelfunctie. Het ASCII toetsenbord lijkt erg op het toetsenbord van een schrijfmachine maar er zijn toch wel een paar verschillen. De indeling is ongeveer hetzelfde, letters in het midden, boven een rij cijfertoeetsen en onder een spatiebalk. Toegevoegd zijn een aantal toetsen voor besturing, waaronder altijd carriage return en line feed.

Gewoonlijk zijn de kleine letters niet te krijgen met de gewone lettertoetsen, maar zijn de hoofdletters de normale functie. De kleine letters zijn de shiftfunctie, dat wil zeggen dat een kleine letter verkregen wordt door de desbetreffende lettertoets in te drukken en tegelijk de shift toets ingedrukt te houden. De niet als aparte toets uitgevoerde stuurtoetsen, de control karakters, zijn te verkrijgen als controlfunctie van de lettertoetsen, dat wil zeggen dat een control karakter verkregen wordt door een lettertoets in te drukken terwijl de controltoets ingedrukt gehouden wordt.

In tabel 1 is te zien in de kolom "soort" of het karakter een normale toets is (aangegeven door "N"), de shiftfunctie (aangegeven door SH/ gevolgd door de naam van de toets), of de controlfunctie (aangegeven door CTL/ gevolgd door de naam van de toets).

Sommige control toetsen zijn als losse toetsen uitgevoerd, maar ook als controlfunctie beschikbaar. Zo wordt het spatiekarakter (\$ 20) verkregen met de spatiebalk, maar ook als controlfunctie van het "C" teken.

De encoder

Uit het voorgaande is duidelijk geworden dat het indrukken van een toets

op het toetsenbord omgezet moet worden in de zeven bits ASCII code, een strobesignaal opgewekt moet worden om aan te geven dat er een karakter beschikbaar is en de shift en controlfuncties moeten worden uitgevoerd. Al deze functies worden door de encoder schakeling uitgevoerd. Het blokschema is in afb. 2 te zien, het principeschema in afb. 2. Een klok oscillator rond een 7413 stuurt een 8-teller (7493). De uitgangen van deze 8-teller worden naar een 8 bits-multiplexer gevoerd. Door het lopen van de teller worden elk van de ingangen van de multiplexer, die door trekweerstand normaal hoog zijn, naar de uitgang van de multiplexer gebracht. De uitgang van de multiplexer zal dus normaal hoog zijn. De overloop van de 8-teller stuurt een 16-teller. De uitgangen van deze 16-teller worden naar een 16-bits demultiplexer gevoerd. De uitgangen van deze demultiplexer worden achtereenvolgens door het lopen van de 16-teller even laag. We hebben nu 16 stuurlijnen die achtereenvolgens even laag worden en 8 aftastlijnen die achtereenvolgens worden bekeken, door de trekweerstand is de uitgang van de multiplexer hoog. Als nu een verbinding tussen een stuurlijn en een aftastlijn wordt gemaakt en dit gebeurt als een toets wordt ingedrukt, wordt de uitgang van de multiplexer laag op het moment dat zowel de stuurlijn als de aftastlijn aan de beurt zijn. Het laag worden van de uitgang gebeurt dus bij een bepaalde stand van de beide tellers. De uitgang van de multiplexer heeft via een poortschakeling tot gevolg dat er geen telpulsen meer aan de tellers worden toegevoegd. De tellers stoppen en de stand van de tellers identificeert ondubbelzinnig welke toets is ingedrukt. De tellerstand, 7 bits, wordt naar een latch (2 x 7475) gevoerd en daarin opgeslagen. Het indrukken van twee toetsen tegelijkertijd, sluit twee uitgangen kort als ze bij dezelfde aftastlijn behoren. De uitgangen zijn hiertegen beschermd door dioden in de uitgangslijnen op te nemen. Welke code opgewekt wordt bij het gelijktijdig indrukken van twee toetsen is volledig willekeurig, het zogenaamde "roll-over" principe.

Antidender schakeling.

Na het indrukken van een toets wordt de uitgang van de multiplexer laag en stopt de teller. Het uitgangssignaal van de multiplexer, een 1 naar 0 overgang wordt met het kloksignaal in een nandpoort gebracht. Daar de tellers reageren op een 1 naar 0 overgang zal bij een coïncidentie, doe optreedt bij het indrukken van een toets, het kloksignaal laag zijn. Pas als het kloksignaal weer hoog wordt, zal het uitgangssignaal van de multiplexer doorgegeven worden aan de antidenderschakeling. Door het uitpoorten met het kloksignaal, zijn de schijncoïncidenties tengevolge van kontaktdender en het gebruik van asynchrone tellers uitgeschakeld. Het coïncidentiesignaal wordt vervolgens aan twee "one-shots", twee monostabiele multivibratoren die hertriggerbaar zijn (74123), toegevoegd. De ene one-shot levert het strobesignaal dat de tellers blokkeert en ook naar buiten gevoerd wordt. De andere one-shot blokkeert door het hertriggeren de strobe one-shot tot de toets minimaal 50 msec losgelaten is. Kontaktdender is zo effectief voorkomen zonder de typesnelheid te beperken. De antidenderschakeling is door R.E. Martens al eerder in Radio Bulletin beschreven, in het julinumnummer van 1976. Het strobesignaal wordt via N3 naar buiten gevoerd en is een puls van 50 msec.

Tijdens de strobepuls is de ASCII code beschikbaar op de uitgangen van de latch en door het inklokken van het strobesignaal blijft de code ook na de strobepuls beschikbaar tot een nieuwe toets wordt ingedrukt. Via N4 is ook een geïnverteerde strobepuls beschikbaar.

De repeat toets

Na de strobepuls blijft de code op de uitgangen van de latch beschikbaar. Als nu hetzelfde karakter weer opgewekt moet worden, wat bijvoorbeeld met spaties vaak zal voorkomen, kan de toets weer ingedrukt worden. Het is veel handiger en sneller om een repeat toets te gebruiken. Het indrukken en ingedrukt houden van deze toets wekt namelijk een reeks strobepulsen op, terwijl de code op de uitgangen van de latch blijft staan. Het indrukken van de repeat toets zal dus een oscillator moeten starten die stopt als de repeat toets weer losgelaten wordt. Een dergelijke oscillator is rond een timer IC 555 opgebouwd. De schakeling is die van een normale oscillator rond een 555 als de repeat toets is ingedrukt. Wordt deze toets losgelaten, dan wordt de condensator van 10 uF niet meer opgeladen en blijft de uitgang hoog. Door de keuze van weerstanden en condensator is de zo ontstane puls, die in N3 gemengd wordt met de gewone strobepulsen, ongeveer even lang als de strobepulsen, zodat het voor de computer lijkt of er iedere keer dezelfde toets wordt ingedrukt. Er bestaat een kleine kans dat de repeat toets wordt losgelaten als de condensator nog opgeladen is. De uitgang is dan laag en zal dat blijven en blokkeert de gewone strobepulsen. Om dit te voorkomen is over de condensator een weerstand van 470K geplaatst die de condensator ontlaadt, zodat de strobepulsen niet geblokkeerd blijven.

De control toets

Uit tabel 1 blijkt dat er een groot aantal stuurcodes zijn, de control toetsen, waarvan er maar een paar als toets worden uitgevoerd. De andere controlcodes kunnen opgewekt worden met de controltoets. Als we in tabel 1 de codes van de letters (hex 40 - 5F) vergelijken met die van de controlcodes, dan zien we dat deze gelijk zijn op bit b_6 na. Als we b_6 nu van een 1 naar een 0 veranderen en de lettertoets indrukken, krijgen we de controlcode. b_6 wordt alvorens naar de latch te worden gevoerd, naar een exclusive or poort gevoerd. De andere ingang van de poort is aangesloten op de controltoets. Als de controltoets niet is ingedrukt, staat er een 0 op de ingang. De exor poort laat het signaal op de andere ingang dan onveranderd door naar de uitgang van de latch. Is de control toets ingedrukt, dan staat er een 1 op de ingang en de exor poort inverseert b_6 . Was b_6 een 1, wat bij de letters het geval is, dan wordt er een 0 in de latch gezet, zodat de controlcode in de latch komt. Tabel 2 verduidelijkt dit nog.

De shift toets

Veel toetsen hebben door de control toets een dubbele functie. Door de shift toets krijgen sommige toetsen een drievoudige functie. Zo wordt bijvoorbeeld een lettertoets gebruikt om een hoofdletter code te krijgen,

met de controltoets wordt een controlcode verkregen, de shifttoets met de lettertoets levert een kleine letter code.

In tabel 1 is te zien welke codes als shift functie van gewone toetsen worden verkregen. In de kolom "soort" is te zien bij welke toets de shiftcode behoort. Vergelijken van de toetscode en de shiftcode geeft als resultaat:

1. De codes 2X moeten omgezet worden in de codes 3X, de codes 3X moeten omgezet worden in de codes 2X. Dit is mogelijk door b_4 te inverteren ten gevolge van het indrukken van de shifttoets. Hierbij is b_6 voor alle codes 0.
2. De codes 4X en 5X moeten worden omgezet naar de codes 6X respectievelijk 7X. Hiertoe moet b_5 geïnverteerd worden. b_6 is in dit geval 1.

Het bovenstaande is samengevat in tabel 3.

Zowel b_4 als b_5 moeten geïnverteerd kunnen worden, twee exorpoorten N_{11} en N_{12} , maken dit mogelijk op dezelfde wijze als bij de control toets¹¹. Alleen is het al of niet inverteren nu afhankelijk van de shifttoets en de toestand van b_6 , zoals in tabel 3 te zien is. Het stuursignaal voor het inverteren wordt met N_5 , N_6 en N_9 samengesteld.

Het toetsenbord

Het toetsenbord moet uit normaal open drukschakelaars opgebouwd zijn. Dit kan een gewone keyboard schakelaar zijn, maar ook toetsen met micro-switches of reed kontakten zijn bruikbaar.

Er zijn reeds gemonteerde toetsenborden te koop in vele uitvoeringen en prijzen en daarmee samenhangende kwaliteit.

Het kan echter voordeliger en handiger zijn om zelf een toetsenbord samen te stellen uit losse toetsen, waar een ruim aanbod van is, ook weer verschillend in prijs en kwaliteit. Het voordeel van het zelf bouwen van een toetsenbord is de vrijheid die we dan hebben in de opstelling van de toetsen.

De meeste toetsen zullen wel op de normale (schrijfmachine) manier geplaatst worden, maar de stuurtoetsen, de controltoetsen, kunnen naar eigen inzicht geplaatst worden. Veel gebruikte toetsen kunnen als aparte toets in plaats van als controlfunctie opgenomen worden.

Een ander voorbeeld van de vrijheid die we hebben is de "?" toets. Vaak is dit de shiftfunctie van de "/" toets.

Omdat in vele (Microsoft) basic implementaties het vraagteken als verkorte notatie van "PRINT" wordt gebruikt, is het handiger het vraagteken als normale toets uit te voeren, het "/" is dan de shiftfunctie.

Een aantal toetsen kunnen we ook voor hardware besturingssignalen reserveren. Zo kan de toets "BREAK" zowel CTL/C, een ASCII stuurcode, danwel STOP (NON Maskable Interrupt), een hardware stop signaal als functie hebben.

Meestal zijn toetsenbord schakelaars bedoeld voor printmontage, zodat het opbouwen van een toetsenbord eenvoudig is door een print te maken. De toetsen komen aan de eerst met een verfspuit bewerkte componentenzijde en worden aan de koperzijde vastgesoldeerd. Op de print kunnen al een groot aantal verbindingen tussen de toetsen (op dezelfde stuur- of tastlijn) via printbanen gemaakt worden, de rest moet met losse bedrading, ook aan de koperzijde, gebeuren. Zoals uit het schema blijkt, heeft elke stuurlijn en elke aftastlijn een (hexadecimaal) nummer.

6502 HARDWARE

De in tabel 1 opgenomen code komt overeen met deze nummers, eerst het aftastlijnnummer en daarna het stuurlijnnummer.

Afwijkende ASCII codes

Vaak wordt de zeven bits ASCII code uitgebreid tot 8 bits. Het aantal mogelijkheden neemt hierdoor toe tot 256. De hier beschreven schakeling kan hiertoe uitgebreid worden door een 16-teller in plaats van de 8-teller te gebruiken en ook een 16-bit multiplexer. Een achtste latch is nog over in een 7475.

Bij videodisplay's dient het achtste bit meestal als stuurteken voor het invertieren van het karakter, dat wil zeggen dat in plaats van een wit teken op een zwarte achtergrond, een zwart teken op een witte ondergrond wordt geschreven.

Toevoegen van een "reverse" toets die de gewenste 0 of 1 in het achtste bit plaatst (ook weer in achtste latch) maakt ook dit mogelijk.

Geheel afwijkende codes, zoals de Baudot code kunnen ook met deze schakeling opgewekt worden. De shift en control logica is dan niet nodig, het aantal bits waaruit de code bestaat, bepaalt wat voor soort tellers en (de-) multiplexers toegepast moeten worden. De toetsen worden natuurlijk ook op andere knooppunten van stuur- en aftastlijnen geplaatst, afhankelijk van de definitie van de code.

Tabel 3. De werking van de shift toets

b6	shift toets	b4 in latch	b5 in latch
0	0	b4	b5
1	0	b4	b5
0	1	$\overline{b4}$	$\overline{b5}$
1	1	b4	$\overline{b5}$

Tabel 2. De werking van de control toets

b6	control toets	b6 in latch
0	0	0
1	0	1) b6 doorgelaten
0	1	1
1	1	0 control functie

Tabel 1. De ASCII code

binair	hex	decimaal	soort	naam	opmerkingen
0 0 0 0 0 0 0 0	00	000	CTRL / C	NUL	Null
0 0 0 0 0 0 0 1	01	001	CTRL / A	SOH	Start of heading
0 0 0 0 0 0 1 0	02	002	CTRL / B	STX	Start of text
0 0 0 0 0 0 1 1	03	003	CTRL / C	ETX	End of text, break
0 0 0 0 0 1 0 0	04	004	CTRL / D	EOT	End of transmission
0 0 0 0 0 1 0 1	05	005	CTRL / E	ENQ	Enquiry, Here is
0 0 0 0 0 1 1 0	06	006	CTRL / F	ACK	Acknowledge
0 0 0 0 0 1 1 1	07	007	CTRL / G	BEL	Bell
0 0 0 0 1 0 0 0	08	008	CTRL / H	BS	Backspace
0 0 0 0 1 0 0 1	09	009	CTRL / I	HT	Hor. tabulation

0 0 0 1 0 1 0	0A	010	N	LF	Line feed
0 0 0 1 0 1 1	0B	011	CTRL / K	VT	Vert. tabulation
0 0 0 1 1 0 0	0C	012	CTRL / I	FF	Form feed
0 0 0 1 1 0 1	0D	013	N	CR	Carriage return
0 0 0 1 1 1 0	0E	014	CTRL / H	SO	Shift out
0 0 0 1 1 1 1	0F	015	CTRL / O	SI	Shift in
0 0 1 0 0 0 0	10	016	CTRL / P	DLE	Data link escape
0 0 1 0 0 0 1	11	017	CTRL / Q	DC1	Device control 1
0 0 1 0 0 1 0	12	018	CTRL / R	DC2	Device control 2
0 0 1 0 0 1 1	13	019	CTRL / S	DC3	Device control 3
0 0 1 0 1 0 0	14	020	CTRL / T	DC4	Device control 4
0 0 1 0 1 0 1	15	021	CTRL / U	NAK	Neg. acknowledge
0 0 1 0 1 1 0	16	022	CTRL / V	SYN	Synchronous idle
0 0 1 0 1 1 1	17	023	CTRL / W	ETB	End of transm. block
0 0 1 1 0 0 0	18	024	CTRL / X	CAN	Cancel
0 0 1 1 0 0 1	19	025	CTRL / Y	EM	End of medium
0 0 1 1 0 1 0	1A	026	CTRL / Z	SUB	Substitute
0 0 1 1 0 1 1	1B	027	CTRL /	ESC	Escape
0 0 1 1 1 0 0	1C	028	CTRL	FS	File separator
0 0 1 1 1 0 1	1D	029	CTRL /	GS	Group separator
0 0 1 1 1 1 0	1E	030	CTRL /	RS	Record separator
0 0 1 1 1 1 1	1F	031	CTRL /	US	Unit separator
0 1 0 0 0 0 0	20	032	N	SP	Space
0 1 0 0 0 0 1	21	033	SH/ 1	!	
0 1 0 0 0 1 0	22	034	SH/ 2	"	
0 1 0 0 0 1 1	23	035	SH/ 3	#	
0 1 0 0 1 0 0	24	036	SH/ 4	\$	
0 1 0 0 1 0 1	25	037	SH/ 5	%	
0 1 0 0 1 1 0	26	038	SH/ 6	&	
0 1 0 0 1 1 1	27	039	SH/ 7	'	
0 1 0 1 0 0 0	28	040	SH/ 8	(	
0 1 0 1 0 0 1	29	041	SH/ 9	)	
0 1 0 1 0 1 0	2A	042	SH/ :	*	
0 1 0 1 0 1 1	2B	043	SH/ ;	+	
0 1 0 1 1 0 0	2C	044	N	,	
0 1 0 1 1 0 1	2D	045	N	-	
0 1 0 1 1 1 0	2E	046	N	.	
0 1 0 1 1 1 1	2F	047	SH/ ?	/	
0 1 1 0 0 0 0	30	048	N	0	
0 1 1 0 0 0 1	31	049	N	1	
0 1 1 0 0 1 0	32	050	N	2	
0 1 1 0 0 1 1	33	051	N	3	
0 1 1 0 1 0 0	34	052	N	4	
0 1 1 0 1 0 1	35	053	N	5	
0 1 1 0 1 1 0	36	054	N	6	
0 1 1 0 1 1 1	37	055	N	7	
0 1 1 1 0 0 0	38	056	N	8	
0 1 1 1 0 0 1	39	057	N	9	
0 1 1 1 0 1 0	3A	058	N	:	
0 1 1 1 0 1 1	3B	059	N	;	
0 1 1 1 1 0 0	3C	060	SH/ ,	<	
0 1 1 1 1 0 1	3D	061	SH/ -	=	
0 1 1 1 1 1 0	3E	062	SH/ .	>	
0 1 1 1 1 1 1	3F	063	N	?	
1 0 0 0 0 0 0	40	064	N	6	
1 0 0 0 0 0 1	41	065	N	A	begin upper case
1 0 0 0 0 1 0	42	066	N	B	
1 0 0 0 0 1 1	43	067	N	C	
1 0 0 0 1 0 0	44	068	N	D	

6502

HARDWARE

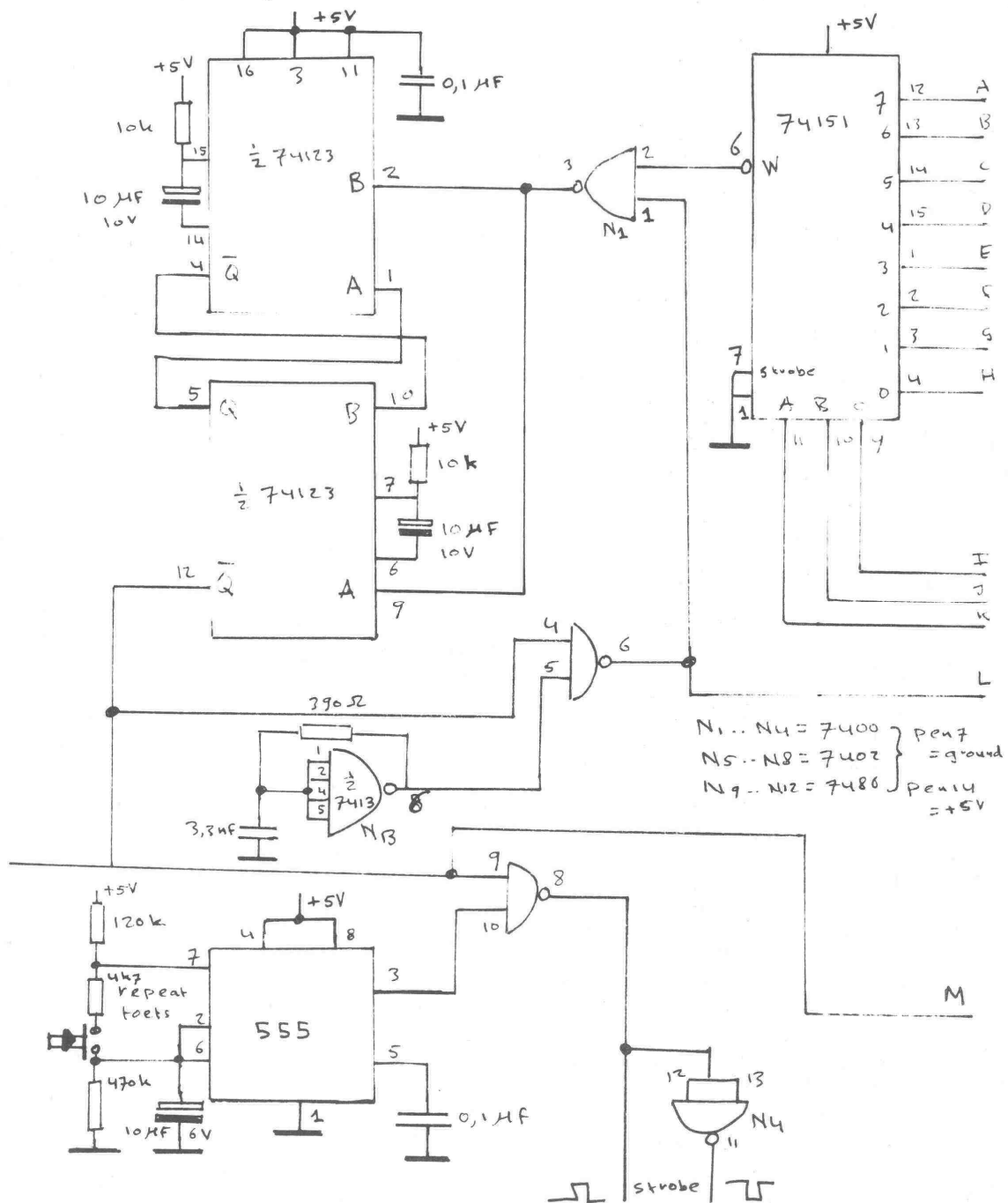
1 0 0 0 1 0 1	45	069	N	E
1 0 0 0 1 1 0	46	070	N	F
1 0 0 0 1 1 1	47	071	N	G
1 0 0 1 0 0 0	48	072	N	H
1 0 0 1 0 0 1	49	073	N	I
1 0 0 1 0 1 0	4A	074	N	J
1 0 0 1 0 1 1	4B	075	N	K
1 0 0 1 1 0 0	4C	076	N	L
1 0 0 1 1 0 1	4D	077	N	M
1 0 0 1 1 1 0	4E	078	N	N
1 0 0 1 1 1 1	4F	079	N	O
1 0 1 0 0 0 0	50	080	N	P
1 0 1 0 0 0 1	51	081	N	Q
1 0 1 0 0 1 0	52	082	N	R
1 0 1 0 0 1 1	53	083	N	S
1 0 1 0 1 0 0	54	084	N	T
1 0 1 0 1 0 1	55	085	N	U
1 0 1 0 1 1 0	56	086	N	V
1 0 1 0 1 1 1	57	087	N	W
1 0 1 1 0 0 0	58	088	N	X
1 0 1 1 0 0 1	59	089	N	Y
1 0 1 1 0 1 0	5A	090	N	Z
1 0 1 1 0 1 1	5B	091	N	[
1 0 1 1 1 0 0	5C	092	N	\
1 0 1 1 1 0 1	5D	093	N	^
1 0 1 1 1 1 0	5E	094	N	~
1 0 1 1 1 1 1	5F	095	N	—
1 1 0 0 0 0 0	60	096	SH/ C	
1 1 0 0 0 0 1	61	097	SH/ A	a
1 1 0 0 0 1 0	62	098	SH/ B	b
1 1 0 0 0 1 1	63	099	SH/ C	c
1 1 0 0 1 0 0	64	100	SH/ D	d
1 1 0 0 1 0 1	65	101	SH/ E	e
1 1 0 0 1 1 0	66	102	SH/ F	f
1 1 0 0 1 1 1	67	103	SH/ G	g
1 1 0 1 0 0 0	68	104	SH/ H	h
1 1 0 1 0 0 1	69	105	SH/ I	i
1 1 0 1 0 1 0	6A	106	SH/ J	j
1 1 0 1 0 1 1	6B	107	SH/ K	k
1 1 0 1 1 0 0	6C	108	SH/ L	l
1 1 0 1 1 0 1	6D	109	SH/ M	m
1 1 0 1 1 1 0	6E	110	SH/ N	n
1 1 0 1 1 1 1	6F	111	SH/ O	o
1 1 1 0 0 0 0	70	112	SH/ P	p
1 1 1 0 0 0 1	71	113	SH/ Q	q
1 1 1 0 0 1 0	72	114	SH/ R	r
1 1 1 0 0 1 1	73	115	SH/ S	s
1 1 1 0 1 0 0	74	116	SH/ T	t
1 1 1 0 1 0 1	75	117	SH/ U	u
1 1 1 0 1 1 0	76	118	SH/ V	v
1 1 1 0 1 1 1	77	119	SH/ W	w
1 1 1 1 0 0 0	78	120	SH/ X	x
1 1 1 1 0 0 1	79	121	SH/ Y	y
1 1 1 1 0 1 0	7A	122	SH/ Z	z
1 1 1 1 0 1 1	7B	123	SH/ [	
1 1 1 1 1 0 0	7C	124	SH/ \	
1 1 1 1 1 0 1	7D	125	SH/]	
1 1 1 1 1 1 0	7E	126	SH/ ^	
1 1 1 1 1 1 1	7F	127	N	DEL

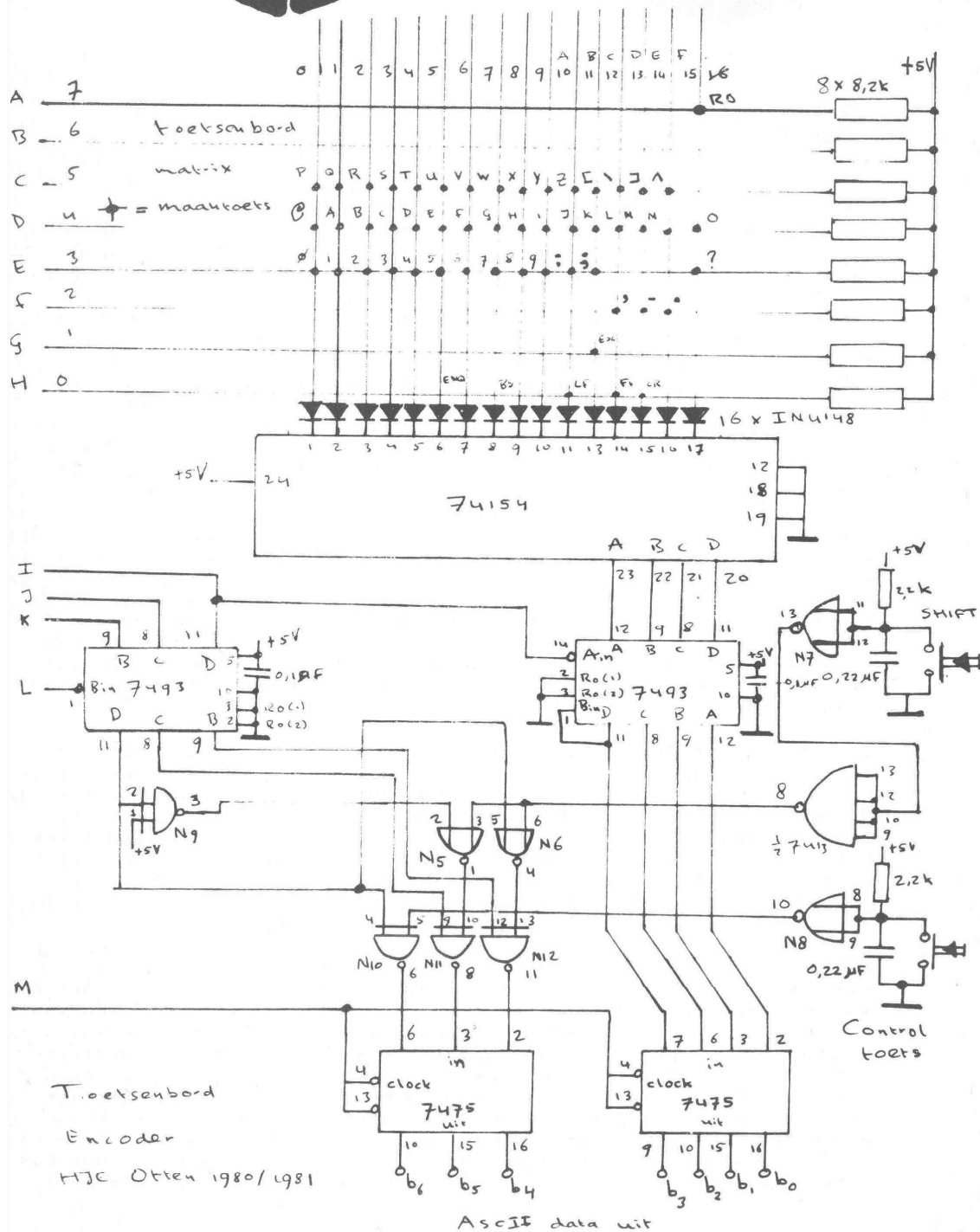
einde upper case

ook wel ↑
ook wel ←
ongedefinieerd
begin lower case

einde lower case

delete of rub out





W.L. van Pelt
Jacob Jordaenstraat 15
Krimpen a.d. IJssel

J. van Sprang
Tulp 71
Krimpen a.d. IJssel

ERVARINGEN MET DE JUNIOR

We willen nog ééns terugkomen op ons extern geheugen. En dan in de eerste plaats op de al veel besproken cassette-interface van Music-print, ook wel bekend onder de naam CHIP. Wie kiest in deze tijd zo'n naam? Verwarrend in een patatcultuur als de onze en nogal gevoelig, denkend aan het aureool van magie dat velen nog aan zo'n stom met elektronica omhangen rekentuijg toedenken en er tegelijker-tijd allerlei gevaren bijfantaseren. Maar laten we ons niet verliezen in het wel en wee van filosofieën over het verschijnsel chip. We willen iets kwijt over de interface. Die werkt nu naar behoren. Een hele opluchting, als je terugdenkt aan alle aanpassingen. Toch is er nog een kleinigheid te melden, ons ter ore gekomen van een enthousiaste medewerker. Wat blijkt? Bij de aankoop van de interface wordt een soort handleiding meege-zonden, waarop schema, componentenlijst en -opstelling. In de lijst zowel als in de componentenopstelling komt een condensator nr.4 met een waarde van 47k voor. Deze ontbreekt echter in het schema. Nu is dat natuurlijk geen probleem, de print van de firma wijst wel de weg. Maar je vraagt je wel af waarom zo' fout in het schema zit. Heeft men dat, evenals het afschrappen van de 74123 type aanduiding, gedaan ter voorkoming van nabouw? Als dat zo is, dan getuigt dat toch wel van ongeremde naieve gedachten. Voor alle zekerheid nog dit: bedoelde condensator nr.4 ligt aan aarde en via R9 en draad brug aan de plus.

In verband met de vele pogingen van anderen om met deze interface toch maar te gaan werken, het wachten op veelbeloofde en veelbeloven-de Elektuur hardware nu toch wel moe, nog even een opmerking terzijde. De in een vorige KIM KENNER gepubliceerde schrijf- en leesroutines van Sebo Woldringh zijn korrekt. Hierover bestond nog enige onzekerheid. Het is waar, we kunnen ons uitgebreider software voorstellen maar deze routines zijn een prima basis om te starten en vergen nauwelijks geheugenruimte. Als het schrijven en lezen nog niet lukt dan moet de fout in de hardware worden gezocht.

Wat het extern geheugen betreft, de uitvoering van de voeding van Hans Otten uit Radio Bulletin maart 1980 bleek op een gegeven moment de oorzaak van veel storing, zo zelfs dat het uitlezen van de band op een gegeven moment onmogelijk bleek. De voeding was opgezet op een pertinax montage strook met koperrondjes, de trafo was een Ormatu type. De oorzaak van de storingen is tot op heden onbekend. Wel is dezelfde voeding nu met een ingegoten Spitznagel trafo op een eigen printontwerp geplaatst, bestemd eigenlijk voor de tweede Twenthe recorder. Het geheel functioneert weer uitstekend. Wel is het gevoel een beetje blijven hangen dat de oorzaak ook kan hebben gezeten in een inmiddels vervangen IC voetje op de interface.

GEBRUIKERS CLUB **NEDERLAND**

Geachte leden,

Hierbij wordt u het financiële overzicht voor het jaar 1980 aangeboden. dit is een bijzonder jaar in het klubgebeuren, want de omzet is meer dan verdubbeld ten opzichte van het vorige jaar. Het ledental is in dit jaar enorm toegenomen. De in 1980 nog genoemde KIM-kenner is behoorlijk verbeterd. De bijeenkomsten zijn, dank zij het vaak gratis beschikbaar stellen van zaalruimte, met een behoorlijk positief resultaat vertegenwoordigd.

De kosten zijn aanzienlijk opgelopen, maar het eindresultaat is met een welkom batig saldo uitgekomen. En dat is zonder meer prettig. De klub zocht al lang naar een behoorlijke schrijfmachine. Ook voor andere plannen zijn er mogelijkheden met een goede financiële positie. Hierbij denken wij bv. aan het komende lustrum, de aanschaf van een klubjunior, enz.

In het verleden zijn er geen financiële mogelijkheden geweest maar eigenlijk is het noodzakelijk om over een reservefonds te beschikken.

Dit om het onverwachte, zoals bv. een ledenterugval op te kunnen vangen, om bv. een kontrakt af te kunnen sluiten om de 6502-kenner een heel jaar te drukken, enz. Hierbij stel ik namens het bestuur voor om in de komende jaren een reservefonds op te bouwen van Fl. 5000.- groot.

In de hoop dat 1981 in alle opzichten even gunstig voor onze klub mag verlopen, verblijf ik als uw penningmeester,

Ted Schouten.

DE 6502 KENNER

KIM GEBRUIKERS CLUB NEDERLAND

BALANS per 31 december 1980.

<u>AKTIVA</u>			<u>PASSIVA</u>		
	<u>31-12-79</u>	<u>31-12-80</u>		<u>31-12-79</u>	<u>31-12-80</u>
Postgiro	1.977,85	9.695,25	Transitoir	850,-	3.450,05
Bank	811,05	371,01	Investering	0,-	887,-
Kas	0,-	424,48	Kapitaalrekening	4773,90	8.257,69
Transitoir	1.670,-	300,-			
Inventaris	1.165,-	1.804,-			
	<u>5.623,90</u>	<u>12.594,74</u>		<u>5.623,90-</u>	<u>12.594,74</u>

EXPLOITATIEREKENING over 1980.

<u>BATEN:</u>		<u>LASTEN:</u>	
Kontributies	12.552,70	Drukkosten KIM-kenner	7.538,26
Bijeenkomsten-netto	2.726,60	Verzendkosten KIM-kenn.	910,47
Oude KIM-kenners	1.046,41	Aanschaf flip-over	309,50
Advertenties	325,-	Uitbreiding KIM-koffer	577,47
Rente	94,24	Porti-kosten	1.476,20
Diversen	70,82	Telef.en autorijkosten	590,79
<u>Van vorig boekjaar:</u>		Kopiëerkosten	235,88
Kontributies	900,-	Materiaalkosten	576,51
Kursusgeld	870,-	Bestuurskosten	965,90
		Diversen	53,-
		Afschrijving inventaris	248,-
		<u>Vorig boekjaar:</u>	
		Lesgelden	800,-
		Batig saldo	<u>4.303,79</u>
TOTAAL	<u>18.585,77</u>		<u>18.585,77</u>

GEBRUIKERS CLUB **NEDERLAND**

I Beknopte toelichting op de balans.

AKTIVA:

Transitoir:

f 300,-

Dit is een vordering i.v.m. het verlenen van een voorschot voor bestuurskosten.

Inventaris:

	Boekwaarde 31-12-79	Afschrijving	Boekwaarde 31-12-80
Inbrandmachine	300,-	100,-	200,-
KIM-koffer	865,-	148,-	717,-
-"- uitbreiding			577,-
Flip - over			310,-
	<u>1.165,-</u>	<u>248,-</u>	<u>1.804,-</u>

PASSIVA:

Transitoir:

Dit zijn de nog te betalen schulden, zoals:

Drukkosten KIM-kenner 14	1.449,55
Verzendkosten KIM-kenner 14	232,78
Gemaakte bstuurskosten voor 1980	<u>1.767,72</u>
	3.450,05

GEBRUIKERS CLUB **NEDERLAND**

II Beknopte toelichting op de exploitatierekening 1980

BATEN:

Kontributies:

12.567,55

De opbrengst voor de kontributies is door 351 betalende leden ingebracht, waarvan 65 leden een gedeeltelijke bijdrage hebben betaald.

Bijeenkomsten.

2.726,60

Dit is het netto-resultaat van de 5 gehouden bijeenkomsten.

LASTEN:

Uitbreiding KIM-koffer

De KIM-koffer is uitgebreid met een 8k. RAM-kaart en 2 schakelbare EPROM-voeten.

400,-

177,47

577,47

Haarlem, 14 maart 1981.

Penningmeester


Ted Schouten.

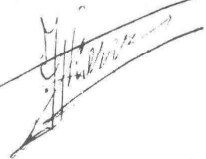
KASKONTROLE.

Hierbij verklaren de leden van de kaskontrole-commissie dat de kas en de jaarstukken voor 1980 zijn gecontroleerd en in orde worden bevonden.

H.L. Burgers.

J. Tervoorren.





Haarlem 13 maart 1981

Nog meer over het extern geheugen. De in Elektuur gepubliceerde 8K RAM kaart met EPROM is niet zo gemakkelijk zelf te vervaardigen. De print is dubbelzijdig en doorgemetalliseerd. Daarom zijn we gaan zoeken naar een redelijk alternatief. Nu hebben we dat wel gevonden maar gezien de mogelijke problemen rond bedrading, inbouw: bezint eer gij begint. Nu hebben wij het gezegde ook niet omgedraaid, dat is waar.

In de eerste plaats hebben we rekening gehouden met mogelijke noodzakelijke vervanging op later tijdstip. Wie weet waar Elektuur in de nabije of verre toekomst mee komt... Tegelijkertijd speelt enige terughoudendheid ook een rol. Als computerconsument wordt je per slot van rekening geconfronteerd met een tweetal tegenstrijdigheden. Ten eerste staat men te trappelen om u allerlei hardware aan te praten, en terecht. Meestal blijkt de belangstelling groter dan de voorraad. Ten tweede wil je best het geld er voor uitgeven ware het niet dat steeds meer een beroep wordt gedaan op het geweten om toch vooral maar meer aan inkomen in te leveren om de economie op poten te houden. Wij zijn geen econoom en wij klagen niet, maar u hoort ons denken: moeten de ondernemers ook nog wat verkopen om aan de kost te komen? Goed, dat was er even uit. Terug naar het redelijk alternatief. We kunnen natuurlijk uitgaan van het idee dat de componenten hergebruikt kunnen worden. De verhoogde kostprijs wordt bepaald door wat bedrading en zelfgemaakte printen. Dat is weer goed te praten met het aantal uren extra knutselen. We hebben ons dan ook met volle overgave op de september en oktober nummers 1979 van Radio Bulletin, waarin de 4K geheugen-uitbreiding van de KIM door Hans Otten, u weet wel. Nauwkeurige bestudering van verschil in connector is vereist. Op die manier ontdek je ook dat tabel 1 van de 31 polige connector niet geheel juist is. Pen 14 moet zijn S3 en pen 16 S4 terwijl pen 30 5 V is. (Gelukkig bij een volle kaart geen probleem) De ervaring heeft ook geleerd dat Hans terecht signaleert dat bufferen van data en adres lijnen "wel aan te bevelen" is. De noodzakelijke bufferkaart is in hetzelfde artikel opgenomen. De uitbreiding zelf met niet al te serieus kort gehouden bedrading bracht nogal wat capacitaire storing met zich mee. De buffering verhielp dat.

Eenmaal de uitbreiding eraan en een busprint op gaatjes board met koperbanen, dan ook een EPROM kaart erbij. En zo geschiedde. Die kaart vonden we bij J.M. v.d. Peijl in een CB Special van Radio Bulletin, ook gepubliceerd in Radio Bulletin juli 1978. En dat heeft als voordeel dat met de welwillende medewerking van een KIM-eigenaar in de buurt nu de schrijf- en leesroutine direct uit het geheugen kunnen worden gestart. Denk nu niet: wat lopen die hard van stapel. We hebben al gezien dat elders de boel klaar staat om op de monitor te gaan. Wij hopen van die leden via het blad veel te vernemen.

Hebben we een nieuwe voeding nodig? Begint de voeding tekort te schieten? Nog niet. Maar wel spoedig. Dus uitgezien naar andere oplossingen. We hebben een echte oplossing nog niet gevonden, we behelpen ons nog maar even. We hebben gedacht aan de 3 A voeding uit Elektuur januari 1978 als voorlopige oplossing, we knutselen wat met een 3 A voeding van de Nascom. Maar we zijn zeker zo benieuwd naar de oplossing die Hans Otten heeft beloofd.

(Radio Bulletin van juni 1981 wijdt aan 4K EPROM print en voeding met dezelfde connector een artikel voor KIM en Junior)

6502 SYSTEEM SOFTWARE

AUTREG 6502 KENNER SOFTWARE LIBRARY PAGE 01

```

0010: 1A00      AUTREG ORG $1A00  AUTOMATISCHE REGISTERUITLEZING JUNIOR
0020:
0030:      OORSPRONKELIJK EEN PROGRAMMA VOOR DE KIM-1, GEPUBLI-
0040:      CEERD DOOR D. DE BOER IN RADIO BULLETIN FEBRUARI 1978
0050:      NU VOOR DE JUNIOR AANGEPAST DOOR:
0060:
0070:      D. BLOK
0080:      RIJKSSTRAATWEG 594 II
0090:      2026 RD  HAARLEM
0100:
0110:      BIJ HET ONTWIKKELEN VAN PROGRAMMA'S KUNNEN WE GEBRUIK
0120:      MAKEN VAN DE SINGLE STEP MOGELIJKHEID. NA ELKE DRUK
0130:      OP DE "GO" TOETS WORDT ER EEN INSTRUCTIE UITGEVOERD.
0140:      OP HET DISPLAY STAAT DAN HET ADRES VAN DE VOLGENDE
0150:      UIT TE VOEREN INSTRUCTIE. NA ELKE STAP KUNNEN WE DE
0160:      INHOUD VAN DE VERSCHILLENDE REGISTERS CONTROLLEREN. EEN
0170:      NADEEL IS, DAT VOOR ELK REGISTER EEN ADRES INGEVOERD
0180:      MOET WORDEN, WAARDOOR HET CONTROLLEREN WAT OMSLACHTIG
0190:      WORDT. NAVOLGEND PROGRAMMA ZORGT ER VOOR DAT TIJDENS
0200:      HET INDRIKKEN VAN DE "GO" TOETS DE INHOUD VAN DRIE
0210:      GEHEUGENPLAATSEN NAAR KEUZE, TEGELIJKERTIJD OP HET
0220:      DISPLAY VERSCHIJNEN. BIJ HET LOSLATEN VAN DE TOETS
0230:      VERSCHIJNT DAN WEER HET ADRES VAN DE VOLGENDE UIT TE
0240:      VOEREN INSTRUCTIE.
0250:
0260:      GEBRUIKSAANWIJZING:
0270:
0280:      OM HET PROGRAMMA TE GEBRUIKEN MOETEN DE VOLGENDE HAN-
0290:      DELINGEN WORDEN VERRICHT:
0300:
0310:      A) EENMALIG.
0320:      1. ER MOET EEN SCHAKELAAR TUSSEN PEN 12A (IRQ) EN
0330:      PEN 12C (NMI) VAN DE EXPANSION CONNECTOR WORDEN
0340:      GESOLDEERD.
0350:      2. HET PROGRAMMA MOET WORDEN GELADEN OP DE ADRESSEN
0360:      1A00 T/M 1A77, OF, INDIEN DEZE GEHEUGENRUIMTE
0370:      VOOR EEN ANDER PROGRAMMA (BIJV. SCHRIJFRoutine
0380:      VOOR CASSETTE-INTERFACE) GEBRUIKT IS OP EEN WIL-
0390:      LEKEURIGE ANDERE PLAATS IN HET GEHEUGEN. (NA-
0400:      TUURLIJK MAG HET PROGRAMMA NIET DE GERESERVEERDE
0410:      RUIMTEN VAN HET GEHEUGEN OVERLAPPEN).
0420:      3. DE NMI-VECTOR MOET INGEVOERD WORDEN. WANNEER
0430:      HET PROGRAMMA WORDT GELADEN VANAF ADRES 1A00
0440:      ZETTEN WE 00 OP ADRES 1A7A EN 1A OP ADRES 1A7B.
0450:      4. GEHEUGENPLAATS "VLAG" MOET SCHOONGEMAAKT WORDEN,
0460:      DUS 00 OP ADRES 00E0 ZETTEN.
0470:
0480:      B) BEDIENING:
0490:
0500:      DE STEPSCHAKELAAR IN STAND "NORMAAL", DUS NIET IN

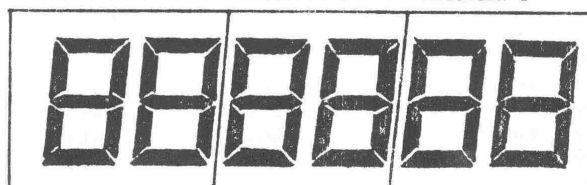
```

6502 SYSTEEM SOFTWARE

UTREG 6502 KENNER SOFTWARE LIBRARY PAGE 02

510: STEPPPOSITIE. OMDAT HET VOOR KAN KOMEN DAT NIET DI-
520: RECT DE FUNKTIE VAN HET PROGRAMMA ALS INTERRUPT-
530: ROUTINE WERKT, WORDEN DE VOLGENDE HANDELINGEN AAN-
540: BEVOLEN:
550: CONTROLEER 00E0=00, 1A7A=00, 1A7E=1A,
560: 1A7E=00, 1A7F=1C.
570: ZET BEGINADRES INTERRUPT ROUTINE 1A00 EN MAAK
580: ENIGE STEPS DOOR INDRIKKEN 'GO' TOETS.
590: DRUK 'AD' TOETS IN EN ZET BEGINADRES VAN HET TE
600: TESTEN PROGRAMMA.
610: DRUK 'GO' TOETS IN EN HOUDT DEZE INGEDRUKT. OP HET
620: DISPLAY VERSCHIJNT LINKS DE INHOUD VAN DE ACCUMU-
630: LATOR, IN HET MIDDE DE INHOUD VAN HET X-REGISTER
640: EN RECHTS DE INHOUD VAN HET Y-REGISTER. (ZIE FIG 1)
650: ACCUMULATOR REGISTER X REGISTER Y
660:
670:
680:
690:
700:
710:
720:

Fig. 1: DISPLAY



730: DOOR HET WIJZIGEN VAN DE ADRESSEN 1A3F, 1A44 EN
740: 1A49 KUNNEN DRIE ANDERE PAGE ZERO LOCATIES WORDEN
750: UITGELEZEN. DOOR HET WIJZIGEN VAN DE OPCODES OP
760: LOCATIES 1A3E, 1A43 EN 1A48 IN 'AD' IN PLAATS VAN
770: 'A5' EN HET WIJZIGEN VAN DE ADRESSEN OP LOCATIES
780: 1A3F/1A40, 1A44/1A45 EN 1A49/1A4A KUNNEN DRIE ANDE-
790: RE WILLEKEURIGE GEHEUGENPLAATSEN NAAR KEUZE WORDEN
800: UITGELEZEN.

810:
820: * ADRESSEN IN PAGINA 00
830: *

840:		KEY	*	\$00E1	
850:	1A00	WSTORG	*	KEY	WERK GEHEUGEN EINDIGT HIER
860:	1A00	VLAG	*	WSTORG	-01
870:	1A00	PCL	*	KEY	+0E
880:	1A00	PCH	*	PCL	+01
890:	1A00	PREG	*	PCH	+01
900:	1A00	SPUSER	*	PREG	+01
910:	1A00	ACC	*	SPUSER	+01
920:	1A00	YREG	*	ACC	+01
930:	1A00	XREG	*	YREG	+01
940:	1A00	BYTES	*	XREG	+01
950:	1A00	COUNT	*	BYTES	+01
960:	1A00	INL	*	COUNT	+01
970:	1A00	INH	*	INL	+01
980:	1A00	POINTL	*	INH	+01
990:	1A00	POINTH	*	POINTL	+01
000:	1A00				

6502 SYSTEEM SOFTWARE

AUTREG 6502 KENNER SOFTWARE LIBRARY PAGE 03

```

1010.
1020.
1030.
1040.
1050. 1A00
1060. 1A00
1070. 1A00
1080. 1A00
1090.
1100.
1110.
1120. 1A00
1130. 1A00
1140.
1150.
1160.
1170.
1180. 1A00
1190. 1A00
1200. 1A00
1210. 1A00
1220.
1230. 1A00 48
1240. 1A01 8A
1250. 1A02 48
1260. 1A03 98
1270. 1A04 48
1280. 1A05 A2 04
1290. 1A07 86 F1
1300. 1A09 BD 7F 1A
1310. 1A0C 48
1320. 1A0D CA
1330. 1A0E D0 F9
1340. 1A10 8E 81 1A
1350. 1A13 AD 83 1A
1360. 1A16 29 E1
1370. 1A18 09 1E
1380. 1A1A 8D 83 1A
1390. 1A1D A9 03
1400. 1A1F 85 F6
1410. 1A21 D8
1420. 1A22 78
1430. 1A23 20 F9 1D
1440. 1A26 C9 13
1450. 1A28 D0 30
1460. 1A2A A5 E0
1470. 1A2C D0 0C
1480. 1A2E A9 02
1490. 1A30 85 E0
1500. 1A32 A9 28

* 6532 LOCATIES
* -----
PAD * $1A80 DATA REGISTER VAN POORT A
PADD * PAD +01 DATA DIR. REG. VAN POORT A
PBD * PADD +01 DATA REGISTER VAN POORT B
PBDD * PBD +01 DATA DIR. REG. VAN POORT B

* TIMER LOCATIES MET IRQ ENABLED
CNTE * $1AFC CLKIT
CNTH * $1AFF CLKIKT

* MONITOR ROUTINES
* -----
SAVE * $1C00
GOEXEC * $1C45
SCANB * $1D8E
GETKEY * $1DF9

INTER PHA ZET INHOUD ACCUMULATOR
TXA X REGISTER
PHA
TYA Y REGISTER
PHA OP DE STACK
LDXIM $04
STX PREG RESET STATUS REGISTER
LDAAX PAD +01 INHOUD
PHA VIER
DEX GEHEUGENPLAATSEN
BNE SAVEA OP STACK ZETTEN.
STX PADD ZET PA0..PA7 OP INPUT (VOOR GETKEY)
LDA PBDD ZET PB1..PB4
ANDIM $E1 OP OUTPUT
ORAIM $1E EN LAAT PBO EN
STA PBDD PBO..PB7 ONGEWIJZIGD
LDAIM $03 ZET
STA BYTES DISPLAY POINTH, POINTL, INH
CLD CLEAR DECIMAL MODE
SEI SET INTERRUPT DISABLED
JSR GETKEY 'GO' INGEDRUKT?
CMPIM $13 NEE
BNE ESC NAAR ESC.
LDA VLAG ALS VLAG=0
BNE VLAG0 NAAR VLAG0
LDAIM $02 ZET VLAG
STA VLAG OP TWEE
LDAIM $28 TIMER OP

```

6502 SYSTEM SOFTWARE

AUTREG 6502 KENNER SOFTWARE LIBRARY PAGE 04

1510:	1A34	8D	FC	1A		STA	CNTE	40 MICROSECONDEN
1520:	1A37	4C	49	1C		JMP	GOEXEC	+04 NAAR MONITOR
1530:	1A3A	C6	E0		VLAGE	DEC	VLAGE	VERMINDER VLAGE
1540:	1A3C	D0	1C			BNE	ESC	BIJ 0 NAAR ESC.
1550:	1A3E	A5	F3		LINKS	LDA	ACC	ACCUMULATOR OP DE 2
1560:	1A40	EA				NOF		
1570:	1A41	85	FB			STA	POINTH	LINKER DISPLAYS
1580:	1A43	A5	F5		MIDDEN	LDA	XREG	REGISTER X OP DE 2
1590:	1A45	EA				NOF		
1600:	1A46	85	FA			STA	POINTL	MIDDELSTE DISPLAYS
1610:	1A48	A5	F4		RECHTS	LDA	YREG	REGISTER Y OP DE 2
1620:	1A4A	EA				NOF		
1630:	1A4B	85	F9			STA	INH	RECHTER DISPLAYS
1640:	1A4D	20	8E	1D	WACHT	JSR	SCANDS	DISPLAY AAN. EN
1650:	1A50	D0	FB			BNE	WACHT	WACHT TOT "GO"
1660:	1A52	A5	EF			LDA	PCL	IS LOSGELATEN
1670:	1A54	85	FA			STA	POINTL	VOL DISPLAYBUFFER
1680:	1A56	A5	F0			LDA	PCH	MET OORSPRONKELIJKE
1690:	1A58	85	FB			STA	POINTH	INHOUD
1700:	1A5A	A2	00		ESC	LDXIM	#00	INHOUD
1710:	1A5C	68			TERUG	PLA		VIER
1720:	1A5D	9D	80	1A		STAAX	PAD	GEHEUGENPLAATSEN
1730:	1A60	E8				INX		VAN
1740:	1A61	E0	04			CPXIM	#04	DE
1750:	1A63	D0	F7			BNE	TERUG	STACK HALEN.
1760:	1A65	68				PLA		INHOUD
1770:	1A66	A8				TAY		X EN Y
1780:	1A67	68				PLA		VAN DE
1790:	1A68	AA				TAX		STACK
1800:	1A69	A9	02			LDXIM	#02	TIMER OP
1810:	1A6B	8D	FF	1A		STA	CNTH	2048 MICROSECONDEN
1820:	1A6E	A5	E0			LDA	VLAGE	ALS VLAGE=0
1830:	1A70	F0	04			BEG	RETURN	NAAR RETURN
1840:	1A72	68				PLA		ACCUMULATOR VAN STACK
1850:	1A73	4C	00	1C		JMP	SAVE	NAAR MONITOR
1860:	1A76	68			RETURN	PLA		ACCUMULATOR VAN STACK
1870:	1A77	40				RTI		TERUG NAAR ONDERBROKEN PROGRAMMA

SYMBOL TABLE 3800 38DE

ACC	00F3	AUTREG	1A00	BYTES	00F6	CNTE	1AF0
CNTH	1AFF	COUNT	00F7	ESC	1A5A	GETKEY	1DF9
GOEXEC	1C45	INH	00F9	INL	00F8	INTER	1A00
KEY	00E1	LINKS	1A3E	MIDDEN	1A43	PADD	1A81
PAD	1A80	PBDD	1A83	FBD	1A82	PCH	00F0
PCL	00EF	POINTH	00FB	POINTL	00FA	PREG	00F1
RECHTS	1A48	RETURN	1A76	SAVE	1C00	SAVEA	1A09
SCANDS	1D8E	SPUSER	00F2	TERUG	1A5C	VLAGE	00E0
VLAGE	1A3A	WACHT	1A4D	WSTORG	00E1	XREG	00F5
YREG	00F4						

STARS 6502 KENNER SOFTWARE LIBRARY PAGE 01

0010. 0000

STARS ORG \$0000

0020.

PROGRAMMA: SHOOTING STARS

0030.

----- EEN SPEL VOOR DE SYM-1

0040.

0050.

AUTEUR. JAN FAAS VAN WOUDEBERG

0060.

----- SILCHERSTRASSE 5

0070.

D 7145 MARKGRONINGEN

0080.

WEST DUITSLAND

0090.

0100.

----- GEBRUIKSAANWIJZING EN SPELREGELS.

0110.

0120.

0130.

ER WORDT GEBRUIK GEMAAKT VAN DE LINKER DRIE DISPLAY'S
OM EEN FIGUUR VOOR TE STELLEN EN VAN DE TWEE RECHTER
DISPLAY'S OM HET AANTAL BEURTEN BIJ TE HOUDEN.

0140.

0150.

0160.

0170.

0180.

0190.

0200.

0210.

0220.

0230.

0240.

0250.

0260.

0270.

0280.

0290.

0300.

0310.

0320.

0330.

0340.

0350.

0360.

0370.

0380.

0390.

0400.

0410.

0420.

0430.

0440.

0450.

0460.

0470.

0480.

0490.

0500.

DE SPELREGELS.

GEPROBEERD MOET WORDEN OM ALLE 9 SEGMENTEN BEHALVE DE
MIDDELSTE (5) AAN TE KRIJGEN. DAN BEGINT AFHANKELIJK
VAN HET PROGRAMMA NA ADRES 02AA DE SPEAKER TE PIEPEN
OF HET DISPLAY TE KNIPPEREN TOTDAT EEN TOETS WORDT
INGEDRUKT OM OPNIEUW TE STARTEN. IN HET PROGRAMMA
IS STANDAARD OP ADRES 02AA E.V. OPGENOMEN DAT DE
SPEAKER PIEFT. OM DIT TE VERANDEREN IN EEN KNIP-
PEREND DISPLAY MOET U HET VOLGENDE WIJZIGEN.

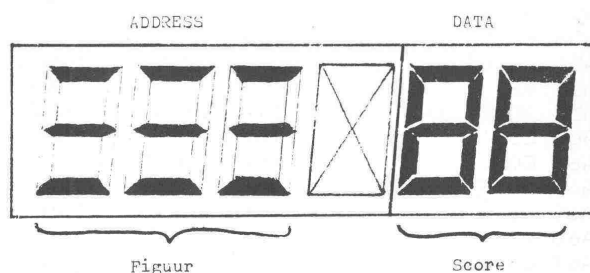
02AA C6 45

DEC LENGTE

02AD D0 E0

BNE AAN

Fig. 1 Display



```

0510: 02AE A2 18          LDXIM #18      GOEDE COMBINATIE +
0520: 02B0 20 D0 02      JSR  DISPLY  STAND OP DISPLAY
0530: 02B3 4C 00 02      JMP  LOOPA
0540: OM DE FIGUUR TE VERANDEREN WORDEN DE TOETSEN 1 T/M 9
0550: GEBRUIKT. HET TOETSNUMMER CORRESPONDEERT MET DE NUM-
0560: MERING VAN DE SEGMENTEN (ZIE FIG. 2). EEN TOETS IN-
0570: DRUKKEN GEEFT ALLEEN REAKTIE ALS HET CORRESPONDERENDE
0580: SEGMENT AAN IS.
0590:
0600:

```

SEGMENT NUMMERING

1	2	3
4	5	6
7	8	9

(FIG. 2)

HET OM FLIPPEN VAN DE SEGMENTEN GAAT ALS VOLGT.

	SEGMENTEN								
	1	2	3	4	5	6	7	8	9
T	1	X	X		X	X			
	2	X	X	X					
O	3		X	X		X	X		
	4	X			X			X	
E	5	X		X		X		X	X
	6			X			X		
T	7				X	X		X	X
	8						X	X	X
S	9					X	X		X

(FIG. 3) X = WISSEL
SEGMENTEN DIE AAN WAREN GAAN UIT
SEGMENTEN DIE UIT WAREN GAAN AAN

DE VERANDERINGSPROCEDURES ZIJN (ZIE FIG. 2 EN 3):

- ALS EEN VAN DE HOEKSEGMENTEN (NRS. 1, 3, 7 OF 9) WORDT INGEDRUKT, WISSELEN (D.W.Z. EEN SEGMENT DAT UIT IS GAAT AAN EN EEN SEGMENT DAT AAN IS GAAT UIT) DE DRIE AANLIGGENDE SEGMENTEN EN HET SEGMENT ZELF.
- ALS EEN TUSSENLIIGGEND SEGMENT AAN DE RAND (2, 4, 6 OF 8) WORDT INGEDRUKT WISSELEN DE TWEE SEGMENTEN DIE IN DEZELFDE RIJ OF KOLOM LIGGEN AAN DE BUITEN-RAND EN HET SEGMENT ZELF.
- ALS HET MIDDELSTE SEGMENT (NR. 5) WORDT INGEDRUKT

```

1010:          WISSELEN DE HOEKSEGMENTEN EN HET SEGMENT ZELF.
1020:
1030:
1040:          VERDERE BEPALINGEN.
1050:          -----
1060:
1070:          ALS EEN '0' TOETS WORDT INGEDRUKT KAN OPNIEUW WORDEN
1080:          BEGONNEN. ALS ALLE DISPLAY'S UIT ZIJN IS HET SPEL AFG-
1090:          GELOPEN EN VERSCHIJNT "FOUT" OP HET DISPLAY.
1100:
1110:          PAGE ZERO LOCATIES.
1120:          -----
1130:
1140: 0000 78      TOETS  =    $78    T
1150: 0001 3F      "      =    $3F    O
1160: 0002 79      "      =    $79    E
1170: 0003 78      "      =    $78    T
1180: 0004 ED      "      =    $ED    S.
1190: 0005 80      "      =    $80    .
1200: 0006 40      STOP  =    $40    -
1210: 0007 6D      "      =    $6D    S
1220: 0008 78      "      =    $78    T
1230: 0009 3F      "      =    $3F    O
1240: 000A 73      "      =    $73    F
1250: 000B 40      "      =    $40    -
1260: 000C 00      ASUBRA =    $00    ADDR LOW SUBR VOOR TOETS 1
1270: 000D 00      ASUBRB =    $00    ADDR LOW SUBR VOOR TOETS 2
1280: 000E 00      ASUBRC =    $00    ADDR LOW SUBR VOOR TOETS 3
1290: 000F 00      ASUBRD =    $00    ADDR LOW SUBR VOOR TOETS 4
1300: 0010 00      ASUBRE =    $00    ADDR LOW SUBR VOOR TOETS 5
1310: 0011 00      ASUBRF =    $00    ADDR LOW SUBR VOOR TOETS 6
1320: 0012 00      ASUBRG =    $00    ADDR LOW SUBR VOOR TOETS 7
1330: 0013 00      ASUBRH =    $00    ADDR LOW SUBR VOOR TOETS 8
1340: 0014 00      ASUBRI =    $00    ADDR LOW SUBR VOOR TOETS 9
1350: 0015 49      KYK   =    $49    1STE SEGMENT
1360: 0016 09      "     =    $09    2DE SEGMENT
1370: 0017 49      "     =    $49    3DE SEGMENT
1380: 0018 00      CODE  =    $00
1390: 0019 00      "     =    $00
1400: 001A 00      "     =    $00
1410: 001B 00      "     =    $00    (ALTIJD)
1420: 001C 00      "     =    $00    BEURT-
1430: 001D 00      "     =    $00    TELLER
1440: 001E 71      FOUT  =    $71    F
1450: 001F 3F      "     =    $3F    O
1460: 0020 3E      "     =    $3E    U
1470: 0021 F8      "     =    $F8    T.
1480: 0022 00      "     =    $00    BEURT-
1490: 0023 00      "     =    $00    TELLER
1500: 0024 00      BLOK  =    $00    SEGMENT 1

```



AMUSEMENT

STARS 6502 KENNER SOFTWARE LIBRARY PAGE 04

1510:	0025 00	=	\$00	SEGMENT 2
1520:	0026 00	=	\$00	SEGMENT 3
1530:	0027 00	=	\$00	SEGMENT 4
1540:	0028 00	=	\$00	SEGMENT 5
1550:	0029 00	=	\$00	SEGMENT 6
1560:	002A 00	=	\$00	SEGMENT 7
1570:	002B 00	=	\$00	SEGMENT 8
1580:	002C 00	=	\$00	SEGMENT 9
1590:	002D 00	TELLRA =	\$00	
1600:	002E 00	TELLRB =	\$00	(BEURTTELLER)
1610:	002F 00	MEMO =	\$00	
1620:	0030 00	KEY =	\$00	
1630:	0031 00	LENGTA =	\$00	
1640:	0032 00	LENGTB =	\$00	
1650:	0033 01	SEG =	\$01	SEGMENT 1
1660:	0034 01	=	\$01	SEGMENT 2
1670:	0035 01	=	\$01	SEGMENT 3
1680:	0036 40	=	\$40	SEGMENT 4
1690:	0037 40	=	\$40	SEGMENT 5
1700:	0038 40	=	\$40	SEGMENT 6
1710:	0039 08	=	\$08	SEGMENT 7
1720:	003A 08	=	\$08	SEGMENT 8
1730:	003B 08	=	\$08	SEGMENT 9

1740: SYM-1 MONITOR ROUTINES:
1750: -----

1760:				
1780:	003C	GETKEY *	\$88AF	LEES EEN TOETS 0 T/M F = HEX 30 T/M 3F
1790:	003C	SCAND *	\$8906	LAAT DISBUF OP DE DISPLAY'S ZIEN
1800:	003C	KEYQ *	\$8923	KIJK OF ER EEN TOETS IS INGEDRUKT
1810:	003C	BEEP *	\$8972	LAAT DE SPEAKER PIEFEN
1820:	003C	TABEL *	\$8C29	HEX TO 7 SEGMENT CONVERSION TABEL
1830:	003C	ACCESS *	\$8B86	UNWRITE PROTECT SYSTEM RAM

1840: TIMER LOCATIES:
1850: -----

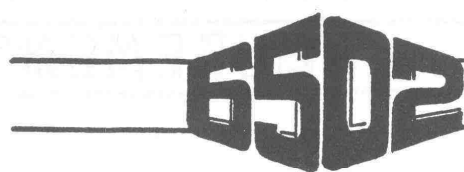
1860:				
1870:				
1880:	003C	TICL *	\$A004	
1890:	003C	TICH *	\$A005	
1900:	003C	TILL *	\$A006	

1910: VIA APPLICATION CONTROL REGISTER:
1920: -----

1930:				
1940:				
1950:	003C	ACR *	\$A00B	

1960: DISPLAY BUFFER:
1970: -----

1980:				
1990:				
2000:	003C	DISBUF *	\$A640	(6 BYTES, EEN VOOR ELK DISPLAY)



```

0010: 0200          ORG    $0200
0020: 0200 A9 2F      START LDAIM SUBRA OPBOUWEN SUBROUTINE TABEL VOOR:
0030: 0202 85 0C      STA  ASUBRA TOETS 1
0040: 0204 85 0E      STA  ASUBRC TOETS 3
0050: 0206 85 12      STA  ASUBRG TOETS 7
0060: 0208 85 14      STA  ASUBRI TOETS 9
0070: 020A A9 4B      LDAIM SUBRE
0080: 020C 85 0D      STA  ASUBRB TOETS 2
0090: 020E 85 13      STA  ASUBRH TOETS 8
0100: 0210 A9 60      LDAIM SUBRC
0110: 0212 85 0F      STA  ASUBRD TOETS 4
0120: 0214 85 11      STA  ASUBRF TOETS 6
0130: 0216 A9 7B      LDAIM SUBRD
0140: 0218 85 10      STA  ASUBRE TOETS 5
0150: 021A A9 C0      LOOPA LDAIM $C0
0160: 021C 8D 0B A0    STA  ACR
0170: 021F A9 64      LDAIM $64    TIMER STARTEN. TELT VAN 0 - 64 (HEX)
0180: 0221 8D 06 A0    STA  TILL                      0 - 100 (DEC)
0190: 0224 A9 00      LDAIM $00
0200: 0226 8D 05 A0    STA  TICH
0210: 0229 A2 12      LDXIM $12
0220: 022B 95 23      LOOPB STAZX BLOK    -01 MAAK ADRES 30 - 41 NUL
0230: 022D CA          DEX
0240: 022E D0 FB      BNE  LOOPB
0250: 0230 20 86 8B    JSR  ACCESS UNWRITE PROTECT SYSTEM RAM
0260: 0233 A2 00      LOOPC LDXIM $00    "TOETS" OP DISPLAY
0270: 0235 20 D0 02    JSR  DISPLY
0280: 0238 A6 2D      LDX  TELLRA
0290: 023A AD 04 A0    LDA  TICI    TIMER UITLEZEN
0300: 023D C9 5A      CMPIM $5A    IS WAARDE < HEX 5A = 90 DEC?
0310: 023F 30 04      BMI  VERDRA ZO JA, VERDER-A
0320: 0241 B5 33      LDAZX SEG    ZO NEE, ZET DE GOEDE DISPLAY
0330: 0243 95 24      STAZX BLOK    CODE IN BLOK
0340: 0245 E8          VERDRA INX
0350: 0246 86 2D      STX  TELLRA
0360: 0248 E0 09      CPXIM $09    ZIJN ALLE 9 HOKJES AL GEWEEST?
0370: 024A D0 E7      BNE  LOOPC    ZO NEE, LOOP-C
0380: 024C A2 06      LDXIM $06    ZO JA: "STOP" OP DISPLAY
0390: 024E 20 D0 02    JSR  DISPLY
0400: 0251 20 11 03    LOOPD JSR  TELQP
0410: 0254 20 F6 02    JSR  BEURT
0420: 0257 A2 18      LOOPE LDXIM $18    CODE + STAND OP DISPLAY
0430: 0259 20 D0 02    JSR  DISPLY
0440: 025C F0 BC      BEQ  LOOPA    IS ER "0" INGEDRUKT? ZO JA, LOOP-A
0450: 025E 85 30      STA  KEY    INGEDRUKTE TOETS (ASCII) ---> KEY
0460: 0260 AA          TAX          KIJK OF SEGMENT WEL AAN IS
0470: 0261 B5 23      LDAZX BLOK    -01 ZO NIET, LOOP-E
0480: 0263 18          CLC
0490: 0264 F8          SED
0500: 0265 A5 2E      LDA  TELLRB BEURTENTELLER DECIMAAL OPHOGEN

```

```

0510: 0267 69 01      ADCIM $01
0520: 0269 85 2E      STA  TELLRB
0530: 026B D8         CLD
0540: 026C A0 00      LDYIM $00      Y NUL MAKEN I.V.M. SUBROUTINE
0550: 026E B5 0B      LDAZX ASUBRA -01 ADRES VOOR SUBROUTINE
0560: 0270 8D 74 02    STA  JSUBRX +01 GOEDZETTEN
0570: 0273 20 2F 03    JSUBRX JSR  SUBRA  XX=SUBRA ALS 1,3,7,9 INGEDRUKT
0580: 0276 20 11 03    JSR  TELOP  XX=SUBRB ALS 2,8 INGEDRUKT
0590: 0279 20 F6 02    JSR  BEURT  XX=SUBRC ALS 4,6 INGEDRUKT
0600: 027C A2 FF      LDXIM $FF      XX=SUBRD ALS 5 INGEDRUKT
0610: 027E E8          LOOPF INX
0620: 027F E0 03      CPXIM $03      ALS ALLE DRIE DE SEGMENTEN
0630: 0281 F0 15      BEQ  VERDRB  GOED ---> VERDER-B
0640: 0283 B5 18      LDAZX CODE
0650: 0285 D5 15      CMPZX KYR      ALS GELIJK TERUG TOT ALLEMAAL VERGELEKI
0660: 0287 F0 F5      BEQ  LOOPF
0670: 0289 A2 FF      LDXIM $FF      VERDER ALS 1 FOUT
0680: 028B E8          LOOPG INX
0690: 028C E0 03      CPXIM $03      ALS ALLE DRIE DE SEGMENTEN UIT --->
0700: 028E F0 38      BEQ  VERDRB  VERDER-C
0710: 0290 A9 00      LDAIM $00
0720: 0292 D5 18      CMPZX CODE
0730: 0294 F0 F5      BEQ  LOOPG
0740: 0296 D0 B9      BNE  LOOPD  ALTIJD!
0750: 0298 A9 FF      VERDRB LDAIM $FF      *** PROCEDURE ALS GOEDE RESULTAAT **
0760: 029A 85 31      STA  LENGTA
0770: 029C A9 0A      LDAIM $0A
0780: 029E 85 32      STA  LENGTB
0790: 02A0 A2 00      LDXIM $00
0800: 02A2 B5 18      LOOPH LDAZX CODE
0810: 02A4 9D 40 A6    STAAX DISBUF
0820: 02A7 E8          INX
0830: 02A8 E0 06      CPXIM $06      DISBUF LADEN MET GOEDE COMBINATIE
0840: 02AA D0 F6      BNE  LOOPH      EN STAND (BEURTENTELLER)
0850: 02AC 20 06 89    AAN  JSR  SCAND
0860: 02AF C6 31      DEC  LENGTA  DISPLAY AAN
0870: 02B1 D0 F9      BNE  AAN
0880: 02B3 A0 00      UIT  LDYIM $00
0890: 02B5 88          LOOPI DEY
0900: 02B6 D0 FD      BNE  LOOPI  DISPLAY UIT
0910: 02B8 C6 31      DEC  LENGTA
0920: 02BA D0 F7      BNE  UIT
0930: 02BC 20 72 89    BLEEP JSR  BEEP
0940: 02BF 20 23 89    JSR  KEYQ      KIJK OF TOETS INGEDRUKT
0950: 02C2 F0 E8      BEQ  AAN      ZO JA OPNIEUW BEGINNEN
0960: 02C4 4C 1A 02    JMP  LOOPA
0970: 02C7 EA          NOP      ***** DEZE NOP NIET VERWIJDEREN.
0980: 02C8 A2 1E      VERDRB LDXIM FOUT      *** PROCEDURE ALS FOUT ***
0990: 02CA 20 D0 02    JSR  DISPLY  FOUT + STAND OP DISPLAY
1000: 02CD 4C 1A 02    JMP  LOOPA

```



AMUSEMENT

6502 KENNER SOFTWARE LIBRARY PAGE 07

```
1010: 02D0 A0 00      DISPLY LDYIM $00
1020: 02D2 A1 00      DSPLYA LDAIX $0000  HAAL DATA VOOR DISPLAY UIT
1030: 02D4 99 40 A6    STAAY DISBUF ZERO PAGE ADRES BEGINNED MET X WAARDE
1040: 02D7 E8          INX
1050: 02D8 C8          INY
1060: 02D9 C0 06      CPYIM $06      ZIJN ER 6 DATA WORDS GEWEEST?
1070: 02DB D0 F5      BNE DSPLYA (1 * DISPLAY VOL)
1080: 02DD 20 06 89    DSPLYB JSR SCAND  DISPLAY VOLSCHRIJVEN
1090: 02E0 D0 FB      BNE DSPLYB
1100: 02E2 20 06 89    DSPLYC JSR SCAND  IS ER EEN TOETS INGEDRUKT +
1110: 02E5 F0 FB      BEQ DSPLYC
1120: 02E7 20 06 89    JSR SCAND  CONTROLE
1130: 02EA F0 F6      BEQ DSPLYC
1140: 02EC 20 AF 88    JSR GETKEY  ZO JA, WELKE?
1150: 02EF C9 3A      CMPIM $3A  IS KLEINER DAN 10
1160: 02F1 10 DF      BPL DSPLYA  ZO NEE NIEUWE TOETS
1170: 02F3 29 0F      ANDIM $0F  HIGH ORDER NIBBLE 0 MAKEN
1180: 02F5 60          RTS
1190: 02F6 A5 2E      BEURT LDA TELLRB ZET DE 7 SEGMENT
1200: 02F8 4A          LSRA      CODE VAN HET
1210: 02F9 4A          LSRA      LINKER DIGIT
1220: 02FA 4A          LSRA      VAN DE BEURTENTELLER
1230: 02FB 4A          LSRA      IN CODE- EN FOUT VELD
1240: 02FC 85 1C      STA CODE  +04
1250: 02FE 85 22      STA FOUT  +04
1260: 0300 A8          TAY
1270: 0301 B9 29 8C    LDAAY TABEL
1280: 0304 A5 2E      LDA TELLRB ZET DE 7 SEGMENT
1290: 0306 29 0F      ANDIM $0F  CODE VAN HET
1300: 0308 A8          TAY      RECHTER DIGIT
1310: 0309 B9 29 8C    LDAAY TABEL  VAN DE BEURTENTELLER
1320: 030C 85 1D      STA CODE  +05 IN CODE- EN FOUT VELD
1330: 030E 85 23      STA FOUT  +05
1340: 0310 60          RTS
1350: 0311 A9 00      TELOP LDAIM $00
1360: 0313 85 2F      STA MEMO  MEMO NUL MAKEN
1370: 0315 AA          TAX      X NUL MAKEN
1380: 0316 A9 00      LOOPJ LDAIM $00  A NUL MAKEN
1390: 0318 A8          TAY      Y NUL MAKEN
1400: 0319 15 24      LOOPK ORAZX BLOK
1410: 031B E8          INX
1420: 031C E8          INX
1430: 031D E8          INX
1440: 031E C8          INY
1450: 031F C0 03      CPYIM $03  OPTELLEN VAN 3 GEHEUGENPLAATSEN
1460: 0321 D0 F6      BNE LOOPK  VAN BLOK
1470: 0323 A6 2F      LDX MEMO
1480: 0325 95 18      STAZX CODE  TOTAAL WEGZETTEN OP CODE
1490: 0327 E8          INX
1500: 0328 86 2F      STX MEMO
```

```

1510: 032A E0 03      CFXIM $03
1520: 032C D0 E8      BNE  LOOPJ
1530: 032E 60          RTS
1540:
1550:                  I. V. M. ADRESBEREKENING BIJ LABEL JSUBRX MOETEN DE
1560:                  SUBROUTINES SUBRA T/M SUBRD ALLEN OP PAGE 3 STAAN!
1570:
1580:                  SUBROUTINE VOOR DE TOETSEN 1 - 3 - 7 - 9
1590:
1600: 032F 46 30      SUBRA  LSR  KEY
1610: 0331 A6 30      LDX  KEY
1620: 0333 E8          LOOPL  INX
1630: 0334 B5 23      LOOPM  LDAZX BLOK  -01 KIJK OF WAARDE 0 WAS
1640: 0336 F0 04      BEQ  VERDRD  ZO JA VERANDER, ZO NEE OOK
1650: 0338 A9 00      LDAIM $00
1660: 033A F0 02      BEQ  VERDRE  ALTIJD
1670: 033C B5 32      VERDRD  LDAZX SEG  -01
1680: 033E 95 23      VERDRE  STAZX BLOK  -01
1690: 0340 C8          INY
1700: 0341 E8          INX
1710: 0342 C0 02      CPYIM $02
1720: 0344 F0 ED      BEQ  LOOPPL
1730: 0346 C0 04      CPYIM $04
1740: 0348 D0 EA      BNE  LOOPM
1750: 034A 60          RTS
1760:                  SUBROUTINE VOOR DE TOETSEN 2 EN 8
1770:
1780: 034B A6 30      SUBRB  LDX  KEY
1790: 034D B5 22      LOOPN  LDAZX BLOK  -02
1800: 034F F0 04      BEQ  VERDRF
1810: 0351 A9 00      LDAIM $00
1820: 0353 F0 02      BEQ  VERDRG  ALTIJD
1830: 0355 B5 31      VERDRF  LDAZX SEG  -02
1840: 0357 95 22      VERDRG  STAZX BLOK  -02
1850: 0359 E8          INX
1860: 035A C8          INY
1870: 035B C0 03      CPYIM $03
1880: 035D D0 EE      BNE  LOOPN
1890: 035F 60          RTS
1900:                  SUBROUTINE VOOR DE TOETSEN 4 EN 6
1910:
1920: 0360 A9 FC      SUBRC  LDAIM $FC
1930: 0362 18          CLC
1940: 0363 65 30      ADC  KEY      ALS 4 = HEX 00 (BEGIN)
1950: 0365 AA          TAX          ALS 6 X=02
1960: 0366 B5 24      LOOPO  LDAZX BLOK
1970: 0368 F0 04      BEQ  VERDRH
1980: 036A A9 00      LDAIM $00
1990: 036C F0 02      BEQ  VERDRI
2000: 036E B5 33      VERDRH  LDAZX SEG

```



```
2010: 0370 95 24   VERDRI STAZX BLOK
2020: 0372 E8             INX
2030: 0373 E8             INX
2040: 0374 E8             INX
2050: 0375 C8             INY
2060: 0376 C0 03         CPYIM $03
2070: 0378 D0 EC         BNE  LOOPF0
2080: 037A 60             RTS
2090:                   SUBROUTINE VOOR TOETS 5
2100:
2110: 037B A2 00         SUBRD  LDXIM $00
2120: 037D B5 24         LOOPP  LDAZX BLOK
2130: 037F F0 04             BEQ  VERDRJ
2140: 0381 A9 00             LDAIM $00
2150: 0383 F0 02             BEQ  VERDRK ALTIJD
2160: 0385 B5 33         VERDRJ  LDAZX SEG
2170: 0387 95 24         VERDRK  STAZX BLOK
2180: 0389 E8             INX
2190: 038A E8             INX
2200: 038B E0 0A         CPXIM $0A
2210: 038D D0 EE         BNE  LOOPP
2220: 038F 60             RTS
```

SYMBOL TABLE 3800 39C8

STARS 0000	TOETS 0000	STOP 0006	ASUBRA 000C
ASUBRB 000D	ASUBRC 000E	ASUBRD 000F	ASUBRE 0010
ASUBRF 0011	ASUBRG 0012	ASUBRH 0013	ASUBRI 0014
KYK 0015	CODE 0018	FOOT 001E	BLOK 0024
TELLRA 002D	TELLRB 002E	MEMO 002F	KEY 0030
LENGTA 0031	LENGTB 0032	SEG 0033	START 0200
LOOPFA 021A	LOOPFB 022B	LOOPC 0233	VERDRA 0245
LOOPD 0251	LOOPE 0257	JSUBRX 0273	LOOPF 027E
LOOPG 028B	VERDRB 0298	LOOPH 02A2	AAN 02AC
UIT 02B3	LOOPI 02B5	BLEEP 02BC	VERDRC 02C8
DISPLY 02D0	DSPLYA 02D2	DSPLYB 02DD	DSPLYC 02E2
BEURT 02F6	TELOP 0311	LOOPJ 0316	LOOPK 0319
SUBRA 032F	LOOPL 0333	LOOPL 0334	VERDRD 033C
VERDRE 033E	SUBRB 034B	LOOPN 034D	VERDRF 0355
VERDRG 0357	SUBRC 0360	LOOPO 0366	VERDRH 036E
VERDRI 0370	SUBRD 037B	LOOPP 037D	VERDRJ 0385
VERDRK 0387	GETKEY 88AF	SCAND 8906	KEYQ 8923
BEEP 8972	ACCESS 8B86	TABEL 8C29	TICL A004
TICH A005	TILL A006	ACR A00B	DISBUF A640

6502 SYSTEM SOFTWARE

BROWSE 6502 KENNER SOFTWARE LIBRARY PAGE 01

```
0010: *****
0020: *
0030: * BROWSE FOR THE JUNIOR COMPUTER
0040: * -----
0050: *
0060: * ORIGINALLY A KIM-1 VERSION, MODIFIED BY
0070: *
0080: * WIM VAN PELT JOHN VAN SPRANG
0090: * JACOB JORDAENSSTR 15 TULP 71
0100: * KRIMPEN AD IJSSEL KRIMPEN AD IJSSEL
0110: *
0120: * IN A PREVIOUS EDITION A PROGRAM IS PUBLISHED,
0130: * WITH WHICH ONE COULD DISPLAY AUTOMATICALLY THE
0140: * MEMORY CONTENTS FROM A CERTAIN START ADDRESS.
0150: * THE SO CALLED AUTOMATIC HEX-DISPLAYER.
0160: * THIS PROGRAM GIVES YOU THE SAME, BUT NOW YOU
0170: * CAN ALSO STEP BACKWARD BY USING THE PC KEY.
0180: * STOP BY USING THE AD KEY AND PROVIDE THE START
0190: * ADDRESS FROM WHERE ON YOU WANT A READOUT. CHANGE
0200: * MEMORY CONTENTS WITH DA AND STEP FORWARDS WITH
0210: * THE + KEY.
0220: * IT IS ALSO POSSIBLE TO KEY IN A PROGRAM WITHOUT
0230: * USING THE + KEY!
0240: *
0250: * LOAD BROWSE ANYWHERE IN MEMORY - IT IS FULLY
0260: * RELOCATABLE - START IT UP, AND PRESTO!
0270: * IT DOESN'T SEEM TO DO ANYTHING.
0280: *
0290: * BROWSE IS A MINI/MONITOR THAT PERFORMS MOST OF
0300: * THE FUNCTIONS OF THE REGULAR JUNIOR MONITOR, BUT
0310: * YOU'LL FIND IT HANDY FOR ENTERING AND PROOF-READ-
0320: * ING PROGRAMS. MOST OF THE KEYS WORK THE SAME AS
0330: * USUAL, BUT PC, +, AND DA ARE SLIGHTLY DIFFERENT.
0340: *
0350: * WHEN YOU HIT + YOU GO TO THE NEXT ADDRESS AS
0360: * USUAL ... BUT THEN YOU KEEP ON GOING! GREAT FOR
0370: * PROOF-READING A PROGRAM YOU'VE JUST ENTERED. IT
0380: * LETS YOU BROWSE THROUGH MEMORY.
0390: *
0400: * HIT PC AND THE PROGRAM STEPS BACKWARDS, SO YOU
0410: * CAN LOOK AT A VALUE YOU'VE JUST PASSED. ALL OTHER
0420: * KEYS INSTANTLY FREEZE THE BROWSING PROCESS, YOU
0430: * CAN HIT AD OR DA TO STOP ON A GIVEN ADDRESS, OR
0440: * JUST ENTER A NEW ADDRESS IF YOU WISH.
0450: *
0460: * KEY DA OPERATES A LITTLE DIFFERENTLY FROM THE
0470: * REGULAR JUNIOR FUNCTION. TO ENTER DATA, FIRST SET
0480: * UP THE ADDRESS BEFORE THE ONE YOU WANT TO CHANGE.
0490: * AS YOU ENTER THE DATA, BROWSE WILL AUTOMATICALLY
0500: * STEP FORWARD TO THE NEXT ADDRESS, AND THEN THE
```

6502 SYSTEM SOFTWARE

BROWSE 6502 KENNER SOFTWARE LIBRARY PAGE 02

```

0510:      * NEXT ONE, AND SO ON. YOU NEVER NEED TO HIT THE +
0520:      * KEY DURING ENTRY; AND THE DISPLAY WILL SHOW THE
0530:      * LAST VALUE YOU HAVE ENTERED.
0540:      *
0550:      * AUTHOR: JIM BUTTERFIELD
0560:      * ----- 14 BROOKLYN AVENUE
0570:      * TORONTO - ONTARIO M4M2X5
0580:      * CANADA
0590:      *
0600:      * WORK AREA'S IN PAGE ZERO:
0610:      * -----
0620:      *
0630: 0100 BROWSE ORG $0100
0640: 0100 WAIT * $00F3
0650: 0100 DIGIT * $00F4
0660: 0100 POINTL * $00FA
0670: 0100 POINTH * $00FB
0680: 0100 TEMP * $00FC
0690: 0100 TEMPX * $00FD
0700: 0100 NIBBLE * $00FE
0710: 0100 MODE * $00FF
0720:
0730:      * REFERENCES TO JUNIOR MONITOR:
0740:      * -----
0750:      *
0760: 0100 GOEXEC * $1C49
0770: 0100 SCAND * $1D88
0780: 0100 GETKEY * $1DF9
0790:      *
0800:      * MAIN PROGRAM STARTS HERE
0810:      *
0820: 0100 D8 START CLD CLEAR DECIMAL MODE
0830: 0101 A9 13 LDAIM #13 GO KEY IMAGE
0840: 0103 85 FE STA NIBBLE
0850: 0105 A9 00 LDAIM #00 VALUE ZERO ..
0860: 0107 85 FA STA POINTL .. TO ADDRESSPOINTER
0870: 0109 85 FB STA POINTH
0880: 010B C6 F3 LOOP DEC WAIT MAIN PROGRAM LOOP
0890: 010D D0 0E BNE LOOPA PAUSE 1 SEC.
0900: 010F A5 FD LDA TEMPX UP OR DOWN?
0910: 0111 F0 0A BEQ LOOPA NEITHER
0920: 0113 10 69 BPL UP
0930: 0115 A5 FA LDA POINTL DOWN, DECREMENT
0940: 0117 D0 02 BNE DOWN NEXT PAGE?
0950: 0119 C6 FB DEC POINTH
0960: 011B C6 FA DOWN DEC POINTL
0970: 011D 20 88 1D LOOPA JSR SCAND LIGHT DISPLAY
0980: 0120 20 F9 1D JSR GETKEY CHECK KEYS
0990: 0123 C5 FE CMP NIBBLE SAME KEY AS LAST TIME?
1000: 0125 F0 E4 BEQ LOOP

```

6502 SYSTEM SOFTWARE

BROWSE 6502 KENNER SOFTWARE LIBRARY PAGE 03

```

1010: 0127 85 FE      STA  NIBBLE  NOTE NEW KEY INPUT
1020: 0129 C9 15      CMPIM #10   NO KEY?
1030: 012B F0 DE      BEQ  LOOP   YES, SKIP
1040: 012D A2 00      LDXIM #00
1050: 012F 86 FD      STX  TEMPX   CLEAR UP/DOWN FLAG
1060: 0131 C9 10      CMPIM #10   NUMERIC?
1070: 0133 90 1C      BCC  NUM    YES, BRANCH
1080: 0135 86 F4      STX  DIGIT
1090: 0137 C9 11      CMPIM #11   DA
1100: 0139 F0 01      BEQ  OVER   YES, LEAVE X=0
1110: 013B E8         INX          NO SET X=1
1120: 013D 86 FF      OVER  STX  MODE   0021 INTO MODE
1130: 013E C9 12      CMPIM #12   PLUS KEY?
1140: 0140 D0 02      BNE  PASS   NO, SKIP
1150: 0142 E6 FD      INC  TEMPX   YES, SET BROWSE
1160: 0144 C9 14      PASS  CMPIM #14   PC?
1170: 0146 D0 02      BNE  PASSA  NO, SKIP
1180: 0148 C6 FD      DEC  TEMPX   YES, DOWN-BROWSE
1190: 014A C9 13      PASSA CMPIM #13   GO?
1200: 014C D0 CF      BNE  LOOPA  NO, LOOP
1210: 014E 20 49 1C   JSR  GOEXEC  START PROGRAM
1220:                *  NUMERIC (HEX) ENTRY COMES HERE
1230: 0151 0A         NUM   ASLA          POSITION
1240: 0152 0A         ASLA          DIGIT
1250: 0153 0A         ASLA          TO
1260: 0154 0A         ASLA          LEFT
1270: 0155 85 FC      STA  TEMP
1280: 0157 A2 04      LDXIM #04   FOUR BITS TO MOVE
1290: 0159 A4 FF      LDY  MODE   AD OR DA?
1300: 015B D0 17      BNE  ADDR   BRANCH IF AD MODE
1310: 015D C6 F4      DEC  DIGIT  TIME TO STEP?
1320: 015F 10 07      BPL  SAME   NO, SKIP
1330: 0161 20 84 01   JSR  INCPT  YES, STEP
1340: 0164 E6 F4      INC  DIGIT  .. AND RESTORE
1350: 0166 E6 F4      INC  DIGIT  .. DIGIT COUNT
1360: 0168 B1 FA      SAME  LDAIY POINTL GET DATA
1370: 016A 06 FC      DADA  ASL  TEMP  MOVE A BIT ..
1380: 016C 2A         ROLA          .. INTO DATA
1390: 016D 91 FA      STAIY POINTL
1400: 016F CA         DEX
1410: 0170 D0 F8      BNE  DADA   LAST BIT?
1420: 0172 F0 A9      BEQ  LOOPA  YES, EXIT
1430: 0174 0A         ADDR  ASLA          MOVE BITS ..
1440: 0175 26 FA      ROL  POINTL  .. INTO ADDRESS
1450: 0177 26 FE      ROL  POINTH
1460: 0179 CA         DEX
1470: 017A D0 F8      BNE  ADDR
1480: 017C F0 9F      BEQ  LOOPA
1490:                *  INCREMENT ADDRESS FOR BROWSING
1500: 017E 20 84 01   UP    JSR  INCPT

```

6502 SYSTEEM SOFTWARE

BROWSE 6502 KENNER SOFTWARE LIBRARY PAGE 04

```

1510: 0181 AA          TAX
1520: 0182 10 99      BPL  LOOPA
1530:                  *  SUBROUTINE INCPT
1540: 0184 E6 FA      INCPT INC  POINTL
1550: 0186 D0 02      BNE  INCPTA
1560: 0188 E6 FB      INC  POINTH
1570: 018A 60          INCPTA RTS
    
```

```

SYMBOL TABLE 3800 38A2
ADDR  0174  BROWSE 0100  DADA  016A  DIGIT  00F4
DOWN   011B  GETKEY 1DF9  GOEXEC 1C49  INCPT  0184
INCPTA 018A  LOOP   010B  LOOPA  011D  MODE   00FF
NIBBLE 00FE  NUM    0151  OVER   013C  PASS   0144
PASSA  014A  POINTH 00FB  POINTL 00FA  SAME   0168
SCAND  1D88  START  0100  TEMP   00FC  TEMPX  00FD
UP      017E  WAIT   00F3  J      0100
    
```

NIEUWS

Een nieuwe importeur voor de SYM
 en voor SYNERTEK :
 Elcres Systems
 C.Vockstraat 63
 2613 DD Delft Tel.015-141182
 Prijzen : SYM-1 f 599,-
 KTM-2 f 799,-
 KTM-80 998,- (80 kar)
 Excl.btw franko huis

VRAAG EN AANBOD

Wie kan mij helpen aan
 een monitor listing
 van de PC100 of AIM - 65 .
 Willem van Pelt
 J.Jordaenstraat 15
 Krimpen a.d. IJssel

6502 SYSTEEM SOFTWARE

HET SLANGETJE EN DE 6532

=====

J. H. Vernimmen
Van IJsendijkstraat 128
1442 CS PURMEREND

Dit programma is bedoeld voor beginners met de Junior-computer om enige oefening te krijgen met het aansturen van de PIA. Het stuurt beurtelings alle segmenten van alle displays aan; het eindeffect is zoiets als een slange-tje van één segment groot dat zich sprongsgewijze van links naar rechts voortbeweegt.

10	SLANGETJE	18	CLC	
20		A9 FF	LDA IMM \$FF	
30		8D 81 1A	PA0 .. PA7	output
40		A9 FF	LDA IMM \$FF	zie tekst
50		8D 83 1A	PB0 .. PB7	output
60	LABEL 01	A9 08	LDA IMM \$08	
70	LABEL 02	8D 82 1A	STA-PBD	selecteer Di 1
80		A9 FE	LDA IMM \$FE	alleen bit 0=0
90		8D 80 1A	STA-PAD	sel.segm,a van Disp.
100	LABEL 03	A2 FF	LDX IMM \$FF	
110	LABEL 04	A0 FF	LDY IMM \$FF	
120	LABEL 05	88	DEY	
130		D0 05	BNE LABEL 05	indien niet 0
140		CA	DEX	
150		D0 04	BNE LABEL 04	indien niet 0
160		38	SEC	..maar carry "1"..
170		2E 80 1A	ROL-PAD	schuif 0 links, sel.
180		18	CLC	vlg segm. van disp.
190		AD 80 1A	LDA-PAD	nr.segm. naar ACC
200		C9 7F	CMP IMM \$7F	0 nog in bit 0 .. 6?
210		D0 03	BNE LABEL 03	ja, spring LABEL 03
220		18	CLC	
230		AD 82 1A	LDA-PBD	Adres disp. naar ACC
240		69 02	ADC IMM \$02	verhoog ACC met 2
250		C9 14	CMP IMM \$14	Disp. nr. 7?
260		D0 02	BNE LABEL 02	nee, spring
270		4C 01 00	JMP-LABEL 01	ja, overnieuw disp. 1
280		77	END	

U vult het startadres in op E2-E3; een eindadres op E4-E5 en toetst het programma in via de JUNIOR-editor om de labels te verwerken; u assembleert het, en ziet, het werkt! (of niet natuurlijk).

Dit programma is echt niet bedoeld om de top 10 van de 6502-library te halen. Wat is het nut er dan van? Laten we het eens bekijken. Wat doet het eigenlijk?

Een hoop dingen die we in wezen wel weten, maar die je als beginner een keer gedaan moet hebben om ze echt tot je door te laten dringen. Om ze, zogezegd, je geestelijk eigendom te laten worden. Bijvoorbeeld: we sturen de displays aan zonder gebruik te maken van SCANDS, en wel direkt via de PIA 6532. Dit laatste doet anders SCANDS natuurlijk ook. Bekijken we het programma eens regel voor regel. Het programma starten met CLC is hier niet noodzakelijk, maar nooit weg. Evenmin misstaan hier de instructies CLD en CLI (SEI). Bij een "koude" start staan deze registers voor dit programma echter al in de juiste stand. Maar wanneer we programma's schrijven

6502 SYSTEEM SOFTWARE

die later mogelijk deel gaan uitmaken van een groter geheel, vergeten we niet ze dan toe te voegen, indien nodig. In regel 20 en 30 worden alle bitjes van de PIA-A kant output gemaakt. In wezen hebben we alleen de bits 0 t/m 6 nodig. Waarom dan bit 7 ook op output? Probeert u het eens: regel 20: A9 7F. Het eerste display doet het nog wel, wat er daarna gebeurt, daar zullen we nog even naar moeten gissen. Voot de PIA-B kant geldt eigenlijk hetzelfde verhaal. We hebben alleen PB 1 t/m PB 4 nodig (zie JC-boek, deel 2, p. 109). Mogen de bits 0, 5, 6 en 7 op input staan? We kunnen het proberen door van regel 40 A9 1E te maken (overigens is in bovenstaande listing regel 40 overbodig; de ACC bevat op dit moment nog FF. Ze is toegevoegd om in dit stadium de inhoud van 1A81 en die van 1A83 onafhankelijk van elkaar te kunnen veranderen). De clou is het volgende: onze PIA 6532 heeft géén 3-STATE uitgangen, dit in tegenstelling tot bijv. de PIA 6520. De bits die op input staan, zijn ook input! Een niet-aangesloten uitgang van een PIA is van nature hoog (= 1). Dit betekent, dat we ééntjes staan in te lezen en wel in bit 7 voor de A-kant in lokatie 1A80 en via PB0, PB5, PB6 en PB7 in lokatie 1A82. Wanneer op regel 40 A9 1E staat, bevat na regel 70 het byte 1A82 geen 08, meer E9. Dit bemerken we echter pas in een later stadium. Van 7F in 1A81 krijgen we eerder last. Het geheugen 1A80 zal nooit 7F bevatten. Het laatste displaysegment wordt aangestuurd door BF (eventueel 3F, alleen hier niet). Na SEC en ROL (regel 160 en 170) hadden we toch 7F mogen verwachten, edoch, bit 7 staat op input, de in bit 7 geschoven 0 wordt direkt door de ingelezen 1 overschreven. De lokatie bevat FF! Het verhaal voor lokatie 1A82 loopt natuurlijk hetzelfde. Hier zijn voor de output alleen de bits 1 t/m 4 van belang. Verhogen we de inhoud van 1A82 met 2, dan wordt het volgend display aangestuurd (Tabel 1, p. 109, JC-boek, deel 2), ook al starten we met E9 als inhoud. Maar na display 6 wordt dan ineens de hele toestand donker ...! Programma's die ineens en onverklaarbaar stoppen met output, werken op mij als een lap op een rooie stier! Het zou niet de eerste keer zijn dat ik daarna een fraaie rookpluim zie verschijnen vanaf daar waar ik het programma vermoed. Dan rest alleen nog met de puinhoop de vraag: was dit nu oorzaak of alleen maar gevolg?

Nu valt het hier allemaal nogal mee. Na enig geduld begint de hele toestand weer vrolijk opnieuw bij disp. nr. 1. Wat deed het programma ondertussen? We kunnen natuurlijk even in de STEP MODE gaan kijken. Alleen ... die lussen bij LABEL 03, 04 en 05. Hoe vaak moet ik daar op de steptoets drukken? 2 x FF voor LABEL 05, en dat FF maal! Toch maar iets anders verzinnen!

Enig binair rekenwerk levert al vrij snel resultaat. Het duffe ding probeert de displays 7 t/m 15 aan te sturen! Vervolgens begint 'ie weer met disp. nr. 1. De compare-instructie op regel 250 (C9 14) is nu volledig de mist ingegaan. Tja, een computer doet nu eenmaal exact datgene wat het opgedragen wordt, en nu maar hopen dat dat ook hetgene is wat je eigenlijk wilt! Overigens wordt display 6 aangestuurd met F3 in 1A82. Wanneer wij bij 1E in 1A83 nu regel 250 veranderen in C9 F5, dan werkt alles weer normaal en verklaarbaar. Evenzo: wanneer 1A81 nu 7F bevat, moet regel 200 in C9 FF gewijzigd worden.

6502 SYSTEEM SOFTWARE

Zouden de PIA's wel ergens op zijn aangesloten, dan kunnen we met de AND-instructie de juiste bits eruit ziften en dan deze testen.

De reeds genoemde loops in regels 100 t/m 150 dienen om de segmenten enige tijd te laten branden. Dit zou ook via de aanwezige timerlokaties mogelijk zijn, maar zo gaat het ook. Dat dit in de registers X en Y gebeurt, komt o.a. daardoor dat de instructieset geen DECREMENT ACC bevat (ook geen INCREMENT ACC, waar van akte!). Dat in regel 160 de carry 1 moet worden is inmiddels wel duidelijk. De Rotate to the left op de volgende regel schuift de carry in bit 0 en deze moet 1 zijn om te voorkomen dat segment a van de displays gaat of blijft branden (volgt uit Tabel 2, p. 109, JC-boek, deel 2). De CLC instructie op regel 180 is het logisch gevolg hiervan, maar achteraf bekeken overbodig! De compare instructie hierna kan de carry beïnvloeden, maar trekt zich van deze carry zelf niets aan. Mede daarom is de CLC in regel 220 zonder meer verplicht. Willen we tenminste in regel 240 de inhoud van de ACC met 2 verhogen. Men bedenke, dat de ADC IMM instructie (69 02) niet alleen de ACC met 2 verhoogt, maar ook de inhoud van de carry! (Dit geldt voor alle ADC instructies, niet voor ADC IMM alleen).

De rest is duidelijk: nadat display 6 is aangestuurd, wordt in regel 270 naar LABEL 01 gesprongen en begint alles opnieuw. De instructie 77 wordt door de editor zelf gegenereerd als END en staat, ten rechte of ten onrechte, nergens in de instructieset omschreven.

VRAAG EN AANBOD

AANGEBODEN :

- SYM-1 met 4K RAM en 4K monitor ROM, 3 x VIA ,geheel in fraaie behuizing , prijs n.o.t.k.
- voeding bij SYM-1 12 V 1 A , 2x 5 V 5 A , -12 V 1 A kortsluitvast prijs f 200,-
- zelfontwikkelde videoschakeling low resolution incl. software prijs f 25,-

Inlichtingen : J.v.d. Zijp
L.J. Costerstraat 14
Krommenie
tel. 075 - 284676 's avonds

6502 SYSTEM SOFTWARE

CBM GEHEUGENUITBREIDING

1.RAM 8K naar 16K en vervolgens 32K

In de 8K machines met dynamisch ram kan men het beste de aanwezige 8k chips, meestal TMS-4108 van TI vervangen door de steeds goedkoper wordende typen TMS-4116. Vervolgens moeten de doorverbindingsblokjes worden uitgesoldeerd (weggooien) en de navolgende doorverbindingen m.b.v. massief montagedraad op de print worden aangebracht:

C-D-F-I-J-M-P-S

Om later naar 32K te gaan moet de tweede rij op de print, nu nog leeg, worden gevuld met eveneens 8stuks TMS-4116, waarbij ook de ontbrekende ontkoppelcondensatoren (0.1 uF tantaal) moeten worden aangebracht. Vervolgens moeten de bovengenoemde doorverbindingen opnieuw worden veranderd en wel:

C-D-F-H-I-J-P-S

16K naar 32K

Dit is in principe slechts mogelijk door de aanwezige 16 stuks TMS-4108 door TMS-4116 te vervangen en vervolgens weer de doorverbindingen te veranderen (zie boven voor 32K)

ECHTER..... !!!! U kunt iets gratis krijgen!

Onlangs ontdekte ik bij toeval dat veel fabrikanten geen 8K type dynamisch RAM produceren maar stomweg, om economische redenen slechts 16K produceren. Om aan de vraag naar 8K te voldoen worden:

OF de 16 k typen slechts als 8K getest en gestempeld.

EN/OF degradeert een 16K type met een foutje in de "tweede helft" tot 8K. In verr. eweg de meeste gevallen heeft dat foutje dan betrekking op de "accestime" (toegangstijd) van de "tweede helft" en daaraan stelt de 6502 minimale eisen zodat veel van deze IC's toch als 16K type in een 6502 systeem bruikbaar zijn.

U kunt dus éérsst proberen, met een grote kans van slagen, alleen de doorverbindingen te veranderen. Door enige tijd te experimenteren kunt U er snel achter komen of de truck is geslaagd. Schakel de machine enige tijd steeds weer opnieuw aan en uit en kijk naar het resultaat in het start-up display! Als dat de correcte waarde blijft geven voor een 16 resp 32K CBM dan is de truck geslaagd, als ook het navolgen- nog netjes klopt: Zet ~~00~~ in bytes \$2~~000~~ resp. \$4~~000~~ en controleer of alle overige bytes nog netjes \$AA bevatten. Is dat zo , dan is het gelukt!

2EPROM

De 2716 EPROM (2K) van het Japanse merk FUJITSU is reeds in veel electronica-zaken verkrijgbaar voor ca f 50,-- en kan zonder meer in een vrije IC-voet van de CBM worden gestoken! Ook kan men nu een interessante wijziging in het operatingsysteem aanbrengen: Een kleine wijziging in de initialisator voor de BASIC-interpreter maakt RESET

6502 SYSTEEM SOFTWARE

zonder aantasting van de RAM-inhoud mogelijk. Men kan deze verandering nu aanbrengen door het tweede ROM van links op locatie \$E000-E7ff door een eeprom te vervangen. Tevens is een sprong naar een ander EPROM mogelijk. Het is om technische redenen niet mogelijk het geheel hier uitvoerig te publiceren. Wel wil ik KIM-CLUBleden het navolgende aanbieden:

- Op elke CLUB-bijeenkomst zal ik voorlopig mijn systeem en eeprom-programmer meenemen. U wordt dan, na het officiële gedeelte of in de pauze in de gelegenheid gesteld Uw 2716 door mij te laten programmeren onder navolgende voorwaarden (kosteloos)
 - a. Uw EPROM dient gewist te zijn.
 - b. Het door U gewenste programma dient op cassette te staan en wel zodanig dat het:
 - Geen gebruik maakt van RAMPagina \$05
 - Geen aanpassing van JMP's, JSR's etc. meer behoeft.
 - c. Door auteursrecht beschermde software wordt alleen in EPROM gezet indien men kan aantonen op legale wijze in bezit van het origineel te zijn gekomen.

SYS en USR met een hexadecimaal adres:

Een interessante routine om in EPROM te zetten is een uitbreiding op de CBM-BASIC-interpreter: SYS:\$033A voert een routine op \$033A uit. A=USR(X): \$033A evalueert de parameter X via een routine op \$033A (adressen slechts als voorbeeld.)

De adressen bij de routine laten we aan uzelf over.

```

start  a0 01    ldyim  #01      ;Set index Y
        a2 05    ldxim  #05      ;Set index X
        b1 77    ldaz   ($77)'y  ;Get 2nd next ch from basictext.
        c9 24    cmpim  "$"      ;Check for "$"
        d0 27    bne    def.      ;If not, go to default USR routine.
loop    20 70    ccjsr  CHRGOT    ;Get next ch from basic text.
        20 e0    e7jsr  HEXIT     ;Convert ascii to hex.
        95 87    staz   $87,x     ;RND accu used for storage.
        ca      inx              ;
        d0 f5    bne    loop      ;Repeat untill 5 ch done.
        a5 8b    ldaz   $8b       ;Get first number in hex.
        0a      asl  a           ;Go shift to first adress-digit.
        0a      asl  a           ;
        0a      asl  a           ;
        0a      asl  a           ;
        18      clc              ;
        65 8a    adcz   $8a       ;Add 2nd number, to provide first byte
        85 10    staz   $10       ;;of adress (high)
        a5 89    ldaz   $89       ;Store high byte of adresspointer.
        0a      als  a           ;Get 3th number in hex.
        0a      asl  a           ;Go shift to 3th adress-digit.
        0a      asl  a           ;
        0a      asl  a           ;
        18      clc              ;
        65 88    adcz   $88       ;Add 4th number, to provide 2nd byte
                                   ;;of adress(low).

```

6502 SYSTEM SOFTWARE

```
85 0f      staz  $0f      ;Store low byte from adress-pointer
20 70 00   jsr   CHRGOT    ;Adjust pointer to next basic ch.
6c 0f 00   jmp   ($0f)     ;JMP to adress,just computed.
def. 4c xx xx jmp   $xxxx   ;JMP to default USR-routine.
```

De subroutines CHRGOT en HEXIT maken deel uit van de in de CBM reeds aanwezige software. \$0f en \$10 zijn ongebruikte zeropage locaties. De laatste JMP dient te wijzen naar Uw meest gebruikte USR-routine in eprom. Zelf heb ik daarvoor de cassette FFWD gekozen. In de bestaande USR-vector op \$0001 dient het adres gezet te worden van deze routine. Dat kan gebeuren door het gewijzigde initialiseringsprogramma voor de basic-interpreter.

De routine kan zowel in direct mode als in een basicprogramma worden toegepast. Na de routine kan, gescheiden door ":" op de normale manier een ander basic-statement worden gezet. Basic zal foutmeldingen geven onder dezelfde condities als een normaal statement. Het is echter niet mogelijk voor het hexadecimaal adres een variabele in te vullen.

R.Uphoff

AGENDA

Agenda van bijeenkomsten gewijd aan de 6502 :

16 mei 1981 KIM gebruikers club Nederland bijeenkomst
Forsom B.V. Coevorden

19 september 1981 KIM gebruikers club Nederland bijeenkomst
Amersfoort

21 november 1981 KIM gebruikers club Nederland bijeenkomst
plaats wordt nog bekend gemaakt

Bijeenkomsten 6500 gebruikers groep Delft E-cafe 20.00 uur :

5 februari , 5 maart , 2 april , 7 mei , 17 juni 1981

Inlichtingen over 6500 ga Delft door briefje met naam te zenden
aan : J.W.Oomen , T.H. Delft Afd Electrotechniek
Mekelweg 4 Delft

28 november 1981 Landelijke bijeenkomst HCC (Hobby Computer
Club)

Patches op Microsoft BASIC

Bug in KIM KENNER 12

Wim Blonk

C.Fagel straat 61
2613 GV Delft

De in KIM KENNER 12 gepubliceerde patch op Microsoft Basic waarmee het mogelijk werd in het begin van een regel spaties te laten staan heeft een vervelend bij effect. Als het DATA statement wordt gebruikt met spaties ervoor geeft het programma de OUT OF DATA foutmelding. De spaties maken de data onbereikbaar. De volgende patch op de spatie patch zorgt ervoor dat spaties voor een DATA statement weer worden weggegooid.

Verander eerst :

4510 A2 45 (een byte meer te verplaatsen)

Op 1780 ev komt te staan :

1780	452E	20 B8 28	JSR \$28B8	
1783	4531	A6 C7	LDX C7	
1785	4533	B5 00	LDA 00,X	
1787	4535	C9 44	CMPI \$44	KARAKTER D ?
1789	4537	D0 06	BNE LOOP	
178B	4539	B5 01	LDA 01,X	
178D	453B	C9 41	CMPI \$41	KARAKTER A ?
178F	453D	F0 0A	BEQ TERUG	
1791	453F	CA LOOP	DEX	
1792	4540	B5 00	LDA 00,X	
1794	4542	C9 20	CMPI \$20	SPATIE ?
1796	4544	F0 F9	BEQ LOOP	
1798	4546	E8	INX	
1799	4547	86 C7	STX \$C7	
179B	4549	60 TERUG	RTS	
179C	454A	A5 87	LDA \$87	verder gelijk, één byte opschuiven !!

Op \$26A1 komt te staan :

26A1 20 9C 17 JSR \$179C

De verplaatsroutine in KIM KENNER 12 is niet correct.
Vervang het volgende :

```
4510 A2 5A
4512 BD 25 45
4515 9D 7F 17
```

DE **KIM** GEBRUIKERS CLUB:
EEN CLUB VAN **6502** KENNERS