

**** DE 6502 KENNERS ** -- EEN CLUB VOOR 65xx GEBRUIKERS**

De vereniging heeft leden in Nederland, België, Duitsland, Frankrijk, Spanje, Portugal, Amerika, Zambia. Het doel van de vereniging is: het bevorderen van de kennisuitwisseling tussen gebruikers van 65xx-computers, zoals KIM, JUNIOR, COMMODORE-64, APPLE, CHE-1, PEARCOM, AIM-65, SYM, PET, BBC ATARI, VIC-20, BASIS 108, PROTON-computers, ITT-2020, OSI, ACC 8000, ACORN ELECTRON, SYSTEM 65, PC-100, PALLAS, MINTA FORMOSA, ORIC-1, STARLIGHT, CV-777, ESTATE III, SBC65/68, NCS 6502, KEMPAC System-4, Elektuur SAMSON-65 DOS computer, LASER, etc., etc.

De kennisuitwisseling wordt o.a. gerealiseerd door 5 maal per jaar DE 6502 KENNER te publiceren (1984 en 1985 6 maal zonder contributieverhoging), door het houden van clubbijeenkomsten, door het instandhouden van een cassette-bibliotheek en door het verlenen van paperware-service. Regionale bijeenkomsten worden door leden georganiseerd.

**Verschijningsdata
DE 6502 KENNER 1985**

derde zaterdag van
februari, april, juni,
augustus, oktober, december.

**Inlichtingen over de regio-
bijeenkomsten:**

Gerard van Roekel
Van der Palstraat 11 - C
3135 LK Vlaardingse
Tel.: 010 - 351101

De vereniging is volledig onafhankelijk, is statutair ogericht en ingeschreven bij de Kamer van Koophandel en Fabrieken voor Hollands Noorderkwartier te Alkmaar, onder nummer 634305.

Voorzitter:
Rinus Vleesch-Dubois
Fl. Nightingalestraat 212
2037 NG Haarlem
Tel.: 023 - 330993

Penningmeester:
John F. van Sprang
Tulp 71
2925 EW Krimpen/IJssel.
Tel.: 01807 - 20589

Leden:
Adri Hankel (05490 - 51151) Hardware/software
Erwin Visschedijk (05490 - 71416) Hardware/software
Promotie

Nico de Vries (010 - 502239) Hardware/software/PET
Erevoorzitter: Siep de Vries
Ereleden : Mw. H. de Vries - Van der Winden
Anton Mueller

Lidmaatschap : Hfl. 45,- per kalenderjaar, postrekening 3757649 t.n.v. Penningmeester KIM Gebruikers Club Ned., Krimpen/IJssel.

Lidm. schap 86: Te voldoen uiterlijk in december 1985.
Advertenties : Tarieven op aanvraag bij de redactie.

Bijeenkomsten van de club

derde zaterdag van
januari, maart, mei,
september, november.

**Redactie-adres en informa-
ties over paperware etc.**

Willel L. van Pelt
Jacob Jordaensstraat 15
2923 CK Krimpen/IJssel.
Tel.: 01807 - 19881

Sekretaris:
Gert Klein
Diederweg 119
6706 CM Wageningen
Tel.: 08370 - 23646

Redactie DE 6502 KENNER:
Willel L. van Pelt
Jacob Jordaensstraat 15
2923 CK Krimpen/IJssel.
Tel.: 01807 - 19881

**** DE 6502 KENNER ** -- EEN BLAD VOOR 65xx GEBRUIKERS**

DE 6502 KENNER is een uitgave van de KIM Gebruikers Club Nederland. Het blad wordt verstrekt aan leden van de club. DE 6502 KENNER wordt van kopij voorzien door leden van de club, bij de opmaak van een publikatie bijgestaan door de redactie. De inzendingen van programma's dienen voorzien te zijn van commentaar in de listings en zo mogelijk door een inleiding voorafgegaan. Publikatie van een inzending betekent niet dat de redactie of het bestuur enige aansprakelijkheid aanvaardt voor de toepassing ervan. De inzendingen kunnen geschieden in assembly-source-listings, in Basic, in Basicode, Forth, Focal, Comal, Pascal, Fortran, Cobol, Logo Elan, etc. etc.

De leden schrijven ook artikelen over de door hen ontwikkelde hardware en/of aanpassingen daarop. Zij schrijven tevens artikelen van algemene aard of reageren op publikaties van andere inzenders.

DE 6502 KENNER IS EEN BLAD VAN EN DOOR DE LEDEN

Micro-ADE Assembler/Disassembler/Editor is een produkt van Micro Ware Ltd., geschreven door Peter Jennings en bestemd voor alle 6502-computers. De KIM Gebruikers Club Ned. heeft de copyrights verworven nadat ons lid Sebo Woldringh de 4 K KIM-1 versie uitbreidde tot 8 K KIM-1 versie. Adri Hankel paste deze aan voor de JUNIOR. Willel L. van Pelt stelde een nieuwe 8 K source-listing voor de JUNIOR samen.

De implementatie op andere systemen dan de KIM-1 en JUNIOR kan eenvoudig gebeuren door het aanpassen van de I/O-adressen, welke in de source-listing gemakkelijke te vinden zijn

FATE Format-lister/cond. Assembler/Tape-utilities/Editor is de door ons lid Rob Banen geschreven source-listing van een 12 K universeel systeem voor de JUNIOR-computer aan de hand van het universele disk operating system van de fa. Proton Electronics te Maarden, nu geschikt voor werken met tapes. FATE wordt beschikbaar gesteld met toestemming van Proton.

In de edities van DE 6502 KENNER worden regelmatig mededelingen gedaan over de door de club georganiseerde bijeenkomsten. Ook worden bestuurlijke mededelingen gedaan, naast informatie over hetgeen in de handel te koop is. Leden die iets te koop hebben of iets zoeken kunnen dit in de edities van DE 6502 KENNER bekend maken. Ook worden wel brieven aan redactie gepubliceerd, evenals specifieke vragen van leden. De edities worden samengesteld op basis van een groot aantal prioriteiten, welke door een redactievergadering worden gehanteerd. Deze vergadering bestaat uit de vaste medewerkers zoals in de colofon vermeld. Het aantal inzendingen is groter dan in een enkele editie van minimaal 48 pagina's is te verwerken. Hierdoor kan het voorkomen dat een inzending eerst na enige tijd kan worden gepubliceerd.

DE CLUB HEEFT BEHOEFTE AAN MEER LEDEN. WIJ WILLEN MEER AAN KUNNEN BIJEDEN DAN NU AL HET GEVAL IS. WERF DAAROM EEN LID!

WILT U EEN PRIJSLIJST? STUUR EEN GEFRANKEERDE ENVELOP AAN HET REDAKTIE-ADRES.

Een onafhankelijke jury kent jaarlijks een aantal aanmoedigingspremie's toe aan auteurs van gepubliceerde artikelen in DE 6502 KENNER.

De 6502 KENNER is een uitgave van de KIM gebruikers Club Nederland.

Adres voor het inzenden van en reacties op artikelen voor DE 6502 KENNER:
 Willem L. van Pelt
 Jacob Jordaensstraat 15
 2923 CK Kriepen a/IJssel
 Tel.: 01807 - 19881

Vaste medewerkers:

Willem L. van Pelt
 Gerard van Roekel
 Frans Smeehuijzen
 Jaap van Toledo

Freelance medewerkers:
 Frans Bakx
 Rob Banen
 Fridus Jonkman
 Gert Klein
 Roger Langeveld
 Anton Mueller
 Gert van Oobroek
 Ruud Uphoff

Gehele of gedeeltelijke overname van de inhoud van DE 6502 KENNER zonder toestemming van het bestuur is verboden. Toepassing van gepubliceerde programma's, hardware etc. is alleen toegestaan voor persoonlijk gebruik.

DE 6502 KENNER verschijnt 6 x per jaar en heeft een oplage van 500 exemplaren.

Copyright (C) 1985 KIM Gebruikers Club Nederland.

De voorpagina is een aquarel van een KIM, geschilderd door:
 Rinus Vleesch Dubois.

In verband met auteurswetgeving en andere maatregelen op het gebied van bescherming van software kan de redactie geen aansprakelijkheid aanvaarden voor inzendingen.

Inzendingen dienen afkomstig te zijn van de inzender, tenzij anders aangegeven.

INHOUDSOPGAVE DE 6502 KENNER NR. 38 JUNI 1985

1. Van de redactie	2.
2. Ervaringen met de Elektuur SAMSON65 DOS computer ... Wouf van Dinther	3.
3. Vragen van leden	3.
4. Bestuursmededeling	3.
5. JUNIOR met Printer Monitor (PM) Dubbele Dobbelsteen ... Maarten Lamey Patches on Printer Monitor ... Fernando Lopez, Portugal SORTING, subroutine to arrange in alphabetic or numeric order ... Fernando Lopez, Portugal	11. 32. 40.
6. BASIC Methode uitwisselen SYM en AIM Basic programma's ... Ronald Vink Het Thomas-Kilman Instrument ... HAVISOF Invullen LOTTO-formulieren Spiegelbeeld op beeldscherm Torens van Hanoi Tokenised Microsoft Basic Keywords and addresses VIC-20 ... Nico de Vries ID on KB-9 Microsoft Basic Programs ... Gamiel de Ly, Belgium	4. 5. 10. 10. 15. 20. 22.
7. HARDWARE VIA-ACIA-PRINT ... M.A. van der Laan	23.
8. FORTH Assembler for FORTH ... Gert van Oobroek Het veranderen van TAPE-FORTH naar DISC-FORTH ... Marcel Breukink Zichtbaar maken controle-karakters in screens ... Wout van Dinther	17. 45. 48.
9. DIVERSEN Mysterieuze hang-ups Diskettes en tapes zijn vergeetachtig Vraag en Aanbod	19. 31. 4, 49.

Het is wellicht niet zo voor de hand liggend dat de redacteur zich geroepen voelt andere mededelingen te doen dan een mededeling in de vorm van een verantwoording aangaande de inhoud van de editie die voor U ligt. Natuurlijk kunnen we over de inhoud het nodige vertellen. Maar dat gebeurt pas als aan mijn andere behoefte gevolg is gegeven: het af en toe toegeven aan de drang om de lezers te vertellen dat het in de wereld van onze hobby ook wel eens om dingen gaat die verder reiken dan ons gebruikelijk interesseveld veelal bestrijkt. Zo staan we niet altijd stil bij het feit dat andere computers ons best wel eens zouden kunnen leren welke dingen we zelf steeds weer over het hoofd zien. Nou, de gebruikers ervan, bedoel ik. Veel lezers van ons blad hebben de neiging alleen of allereerst te zoeken naar die artikelen welke direct voor de eigen computer van toepassing kunnen zijn. Bijgevolg worden soms uiterst aardige software-kneepjes over het hoofd gezien. Maar ook worden de mogelijkheden van betere toepassingen dan niet opgemerkt. Toegegeven, beginnende hobbyisten met bijvoorbeeld een Commodore - 64 kunnen moeilijk de edities oordeelend doornemen. Zij hebben in de eerste plaats behoefte aan iets dat hen kan bezig houden: het intypen van b.v. Basic-programma's die speciaal voor hun computer is geschreven. Althans, dat kan hun eerste reactie zijn. Ik ben echter van mening dat daarmee over het hoofd wordt gezien dat ons blad tevens beoogt een blad te zijn waaruit lering kan worden getrokken. Juist door het aandacht hebben voor programma's voor andere computers worden we gedwongen allerlei aanpassingen daarop te maken, omdat we op de eigen computer ernaar kunnen werken. Het is een uiterst leerzame activiteit uit te proberen op de eigen computer wat een ander reeds op zijn (andere) computer heeft ontwikkeld. Als we daarin zijn geslaagd, moeten we in elk geval niet vergeten dat de redactie zit te springen om die oplossingen die we bedacht hebben. Immers, dat is weer voor publikatie vatbaar, en daarmee helpen we weer die leden die er (nog) niet in slaagden oplossingen te vinden. Het is heus niet zo gek om ervan uit te gaan dat we elkaar eigenlijk nodig hebben, en dat de club haar erkenning weet te vinden juist door deze instelling te presenteren.

Op de landelijke bijeenkomst te Almelo op 18 mei jongstleden heeft de heer Nachtaam van het tijdschrift Elektuur een SAMSON65 DOS-computer en een Big Board II-computer gedemonstreerd. Met behulp van een overhead-projector hield hij een lezing over de mogelijkheden van de SAMSON65, welke software daarvoor beschikbaar zal komen, waaronder PASCAL en veel 'public domain software'. In zijn verhaal liet hij doorschemeren dat de SAMSON65 de samenwerking met andere processoren zoals Z-80, 6809 en 68000 niet zal schuwen, en dat ook harddisk als extern geheugen tot de mogelijkheden behoort. Een vergelijking tussen de processoren leert dat de manier van programmeren van elk van hen zo op elkaar lijkt dat er eigenlijk van geen verschil in programmeren kan worden gesproken, of het moet zijn dat het gaat om een ietwat ander gebruik van de registers. Kenners van 6502-machinetaal moeten geen moeite hebben de wereld van die andere processoren te doorgronden. Bijgevolg trek ik de conclusie dat een grensverleggende waarneming meer perspectief te bieden heeft. Met andere woorden: het kennis nemen van hetgeen anderen doen met andere computers, met name computers uit een andere dan de 65xx-familie, ook een bijdrage kan leveren aan het verhogen van het nivo van programmeren bij onze leden. Ik bedoel hier allereerst mee dat we ook zouden moeten stimuleren tot de aanschaf of de bouw van een anderssoortige computer. Integendeel, voor mij staat nog steeds als een paal boven water dat het gebruik van een computer met een 65xx-processor als 6502 en 6510, of 65SC802, of 65SC816, etc. voor de hobbyist nog de meeste perspectieven te bieden heeft, al dan niet, zoals bij de APPLE, gekombineerd met Z-80, omdat de software ondersteuning ongekend oavangrijk is en in hobbykringen misschien wel het meest onderworpen aan de onweerstaanbare drang tot het oplossen van uiterst ingewikkelde problemen.

Welke praktische vertaling zou je aan een deroelike gedachte in de zin van bestuurlijke besluitvorming moeten geven? Dat is niet zo eenvoudig te zeggen. Allereerst heb je te maken met het feit dat een bestuurder de verantwoordelijkheid heeft uit te dragen hetgeen tegenover de leden als verantwoord kan worden betiteld. Als zodanig heb ik te maken met het onderscheiden van de eigen prive-wensen ten opzichte van datgene wat de leden akseptabel vinden. Hoewel dit laatste niet zo eenvoudig liot, het is niet zo, dat het niet met enige zekerheid in te schatten is. Een behoorlijk contact met de leden kan daaraan een aardige bijdrage leveren. Zo ben ik zo ijdel om te veronderstellen dat de leden mijn gedachten kunnen volgen als ik het toejuich eens na te denken over de toekomst van de club. Nu moet ik eerlijkheidshalve erkennen dat ik ervan uitga dat de toekomst van DE 6502 KENNER en de toekomst van de club ontzettend veel met elkaar van doen hebben. Heel eenvoudig gesteld komt het erop meer dat ik redeneer dat, waar de instroom van koudje aan kracht inboet, het aanzwengelen van die instroom uit andere bronnen niet alleen geoorloofd maar zelfs noodzakelijk is. Het aksepteren of aemchtig reageren op een zwakker kloppend hart is het tekenen van het eigen vonnis. Om deze reden heb ik het altijd toegejuicht als het reeds van onze club deel uit makende aantal buitenlandse leden sterk zou worden opgevoerd. In het buitenland wonen niet alleen hobby-vrienden, we kunnen er misschien nog iets van leren. Ik ben niet van oordeel dat ik alwetend ben, maar ik weet wel graag alles. Daarnaast doen we als bestuur al een paar jaar ons best het ledenaantal op te schroeven. Vijfhonderd maal Fl. 45,- geeft altijd meer mogelijkheden wat voor de leden te doen dan vierhonderd maal dat bedrag. Mede door het stomme feit dat we aan chronisch geldoebrek leiden kunnen geen andere wervingsacties op touw worden gezet dan enkele leden te vragen een vriend lid te maken, wat op zich echt wel positieve gevolgen heeft, maar te weinig om groei op te leveren. Dus komt het ons goed uit als Elektuur de SAMSON65 in Duitsland het levenslicht schenkt en onze naam noemt in haar Elektor Computing Sonderheft. Dat heeft ons al nieuwe leden opgeleverd. En niet alleen in Duitsland, maar zelfs in Zaambia! Inmiddels is het besef doorgedrongen dat we het op die manier niet redden om te groeien. Vandaar dat in het bestuur een voornamelijk door Adri Hankel geschreven nota een eerste overpeinzing over de toekomst heeft opgeleverd. Rond dat thema is er besloten dat er ook ondersteuning moet komen voor het DOS65 voor JUNIOR met VDU, in de vorm van op CAD/CAM nivo ontwikkelde controllerkaart en de distributie van speciaal daarvoor ontwikkelde software met full-screen editing, met een behoorlijke handleiding. Voor dit werk is HAVISOF verantwoordelijk, dit zijn Adri Hankel en Erwin Visschedijk, terwijl de kennis van Nico de Vries tegelijk wordt ingeschakeld. Maar dat is niet alles. Het bestuur was al lang van mening dat de redactie over eigen apparatuur moest beschikken. Er staat nu ook een APPLE-compatible met Z-80 kaart voor draaien met CP/M en op 80 koloms en nog wat leuk. Dat wil geenszins zeggen dat ik niet meer met 6502 werk, het wil wel zeggen dat een 6502-computer heel goed te combineren is met een andere processor. Ik ken al een APPLE gebruiker in onze club die een combinatie met 68000 heel goed ziet zitten.

Enige tijd geleden heb ik andere leden van het bestuur erg zachtjes in het oor gefluisterd dat ik ervan gecharmeerd ben dat onze club zich open zal stellen voor bijvoorbeeld andere computers dan alleen de eerder genoemde uit de 65xx-familie, waarbij ik echter niet direkt verder denk dan computers uit de 68xx-familie. Enige jaren geleden stapten enkele van onze leden over naar een dergelijke computer. Zij waren en zijn niet de minste in het programmeren. Die kennis hebben we uit onze handen laten glijden, vind ik. Dat moet ons niet meer gebeuren. We moeten alles op alles zetten onze club nog meer te maken dan het al is. In Almelo heb ik de aanwezigen gezeed dat we de toekomst op twee manieren kunnen tegemoet zien: we maken hem zelf, of hij wordt ons opgelegd. Laten we de slag voor blijven! v.p.

 *
 * ERVARINGEN MET DE ELEKTUUR SAMSON65 COMPUTER *
 *

Door : WOUT van DINTHER

Reeds enige tijd heb ik mijn oude vertrouwde JUNIOR omgetuurd naar een SAMSON65, want een mens wil ook wel eens wat anders. Ik wil hier alleen maar een vergelijking treffen tussen de JUNIOR en de SAMSON65 daar andere vergelijkingen mijns inziens niet helemaal passen. Eerst even dit: hardwarematig draait de SAMSON65 met de tot op heden gepubliceerde ontwerpen van Elektuur en er komt niets extra's bij (uitgezonderd een klein tracerprintje). Uitbreidingen in de toekomst zullen waarschijnlijk in eerste instantie op de SAMSON65 slaan (zowel hardware- als softwarematig). De hardware berust op de CPU-kaart, de VDU-KAART, de floppy-controller en een 64K dynamische RAM-kaart of meerdere 16K RAM-kaarten natuurlijk. Het toetsenbord wordt volledig parallel ingelezen, dus geen gedonder meer met het inlezen van de I/O-baudrate (rubout werd hiervoor gebruikt) waardoor de RS-232 volledig vrij ter beschikking is gekomen (voor modem e.d.).

Kijken we even naar de monitor. Er zijn mensen voor wie iedere byte buiten de bootup-routine teveel is, daar zij hun systeem alleen maar gebruiken om kant en klare software te verwerken, maar iemand met de ware 'JUNIOR'-geest nog in zich zal inzien dat hij met deze monitor een knap stuk gereedschap er weer bij heeft. Niet alleen een hex- en ascii-dump maar ook een heel krachtige tracer en nog diverse andere routines verschuilen zich hierin. En er is nog plaats voor eigen ontwikkelingen c.o. veranderingen. In dit artikel moet ik mij bedenken daar er een uitgave van Elektuur over zal verschijnen, waarin alles uitgebreid beschreven staat. (Red.: vermoedelijk na de zomervakantie) In Basic zijn opgenomen de shorthand-commando's (gepubliceerd in Elektuur, december '84) en een full-screen editor en tevens de DIR en PUT commando's.

Alles bij elkaar is het er een stuk gebruikersvriendelijker op geworden voor weinig werk (over geld praat ik niet, dat doet Elektuur wel). De assembler die er nu bij zit is gedeeltelijk afgeleid van Micro-ADE (wat commando's betreft) en is m.i. ook iets gemakkelijker gemaakt. De wordprocessor die onder SAMSON65 werkt is door de heer Nachtmann geschreven, maar deze heeft van Elektuur niet alle gelegenheid gekregen om haar tot in de puntjes af te werken. Diverse routines zijn al in de wordprocessor geplaatst, doch kunnen via een commando nog niet aangeroeven worden. Hier is dus weer werk aan de winkel voor leden van de club! Dat moet weer copy opleveren voor de redactie van Nogmaals: ten opzichte van de JUNIOR een hele verbetering. Er is een mooi stukje gereedschap bijgekomen, edoch een persoonlijke inbreng (verbetering, aanpassing) is ook hier weer opengelaten! Kijken we naar andere systemen, dan moet ik er eerlijk bijzeggen dat ik nog maar weinig systemen heb gezien waar zo gemakkelijk overal in te komen is. En: we hebben indertijd de JUNIOR aangeschaft (de meeste althans) om er wat van te leren.

Rest mij nog te vertellen dat er met Duitsland een link is gelegd, alwaar SAMSON65 al een tijdje draait. Van die zijde valt te verwachten, door de toetreding in onze club van al enige in dat land woonachtige SAMSON65-bezitters, dat er ook van hen het een en ander aan ervaringen kan worden opgestoken.

Kortom, diegenen die een CPU-kaart van Elektuur hebben moeten mijns inziens niet meer twifelen, maar overstappen naar SAMSON65. Alles dat op JUNIOR draaide, doet het nu ook op SAMSON65.

De redactie wacht met enthousiasme jullie inzendingen in.

V R A G E N V A N L E D E N

- Ik ben in het bezit van een APPLE II. Het 'ready'-signaal op mijn printer-interface is een 'aan/uit'-schakelaar. Kan zo'n signaal dienen als een interrupt-signaal? Ik moet hierbij omerken dat het signaal 'uit' gaat (printer not ready) wanneer ik op reset druk. Dit is de reden dat ik een dummy-karakter (null) naar de printer stuur bij het PR#-commando. Volgens mij zoroet het interrupt-signaal dus voor de eigen uitschakeling, maar dit is slechts een vermoeden.
- Wat gebeurt er met een interrupt request wanneer interrupts zijn 'disabled'? Gaat de interrupt verloren of wordt hij bewaard totdat een interrupt enable instructie is gegeven?
- Voor de APPLE II PLUS zijn allerlei shift-key en autorepeat modificaties verschenen, zo ook in HCC nummer 66. Deze modificaties zijn echter alleen voor de revision 7 boards en ik heb een revision 4. Nu is mijn vraag, zijn er ook shift-key en auto repeat modificaties voor deze revisie?

Stuur uw antwoord naar de redactie. Deze zal voor doorzending zorgdragen.

Marcel Visser.

Ik heb van een bedrijf een IBM 3742 gekregen. Dit apparaat bestaat uit twee 8" drives, twee toetsenborden, een monitor welke met behulp van een spiegel in tweeen is gedeeld, een flinke voeding, en een microprocessor met bijbehorend geheugen en I/O. Alle IC's zijn eigen fabrikaat, zodat ik niet weet wat wat is.

Nu wil ik zo'n drive aan m'n Junior hangen, maar ik weet niet hoe. Wat ik wil weten is het volgende:

- wat doet een floppy disk controller, en kan het ook zonder?
- wat is het verschil tussen 5" en 8" ?
- wat is het verschil tussen Ohio en IBM-formaat?
- is er ook informatie speciaal gericht op een 8" drive gekoppeld aan een BBC, model B ?

Ik heb nog en ander probleem:

ik heb de volgende wijziging in mijn JUNIOR aangebracht:

- de adresdekodering is vervangen door een volledige adresdekoder in blokken van 4K, waarbij twee blokken zijn uitgevoerd tot 8 blokken van 1K (K0..K7)
- de standaard-monitor zit nu op FC00..FFFF
- op de plaats van de eeprom op de standaardkaart zit nu een 2016 (2K statische RAM)
- de 6502 is vervangen door de 65C02 van Rockwell.

Als ik nu een programma oosla op pagina 2, 3, 1C, 1D, 1E of 1F (2016), dan gaat dat uitstekend. Als ik het programma laat lopen, dan worden er echter een heleboel bytes gewijzigd! Dit gebeurt telkens met dezelfde geheugenplaatsen. Een programma in PIA-RAM heeft echter nergens last van!!! Hoe kan dit, en hoe herstel ik dit? De voedingsspanning is gecontroleerd en heeft een waarde van 4,84V.

Als iemand mij met deze problemen kan helpen, laat het dan even weten. Ik ben er zeker een eind verder mee.

Stuur het naar de redactie. Die stuurt het mij door.

Marcel van Doro.

BESTUURSMEEDELING

Het bestuur zoekt op korte termijn iemand die veel enthousiasme voor de hobby weet te combineren met wat bestuurlijke activiteiten op het terrein van 'marketing' ten behoeve van landelijke bijeenkomsten en DE 6502 KENNER. Heeft U belangstelling, schrijf of bel de redactie dan heel even.

 * METHODE 'DOOR HET UITWISSELEN VAN PROGRAMMA'S TUSSEN *
 * SYM EN AIM-65 BASICS ZONDER AL TE VEEL PROBLEMEN *

DOOR: RONALD VINK

Zelf heb ik een AIM-65 Basic. De hex- en ASCII-dump zijn hierop gebaseerd, maar voor de SYM-Basic is dit ongeveer hetzelfde, alleen is het startadres anders. Stel: we hebben een klein programma en we lezen daarna het geheugen uit, dan krijgen we het volgende beeld:

TEST PROGRAMMA: 10 PRINT "ABCDEF"
 20 PRINT "123456"

HEXDUMP:2000,202F

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2000:	XX	15	20	0A	00	97	22	41	42	43						
2010:	44	45	46	22	00	23	20	14	00	97	22	31	32	33	34	35
2020:	36	22	00	00	00	AA										

ASCII-DUMP:2000,202F

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
2000:														"	A	B	C
2010:	D	E	F	"							"	1	2	3	4	5	
2020:	6	"															

0073 07 +
 0074 20 +
 0075 25 +
 0076 20

VERKLARING

Op de adressen 0073 en 0074 staat de pointer die wijst naar het begin van het Basic-werkgeheugen. Bij de AIM-65 Basic is dit 2007. De adressen 2000-2006 worden gebruikt voor de cassette-routines. Het begin- en eindadres van het programma wordt hier opgeslagen. 0075 en 0076 geven het einde van het Basic-programma aan, nu 2025 en wijzen naar het adres waar een regelnummer wordt verwacht, dus na de laatste 00 die aan het einde van elk Basic-programma staat. Op 2007 en 2008 wordt het adres bewaard waar de volgende Basic-regel begint, in dit geval 2015. 2009 en 200A geven het regelnummer van de komende Basic-regel. 000A (hex) geeft regel 10. 200B bevat het Basic-token '97 (PRINT). 200C ASCII-waarde van een " 200D-2012 de ASCII-waarden van de te printen string ABCDEF en wordt weer afgesloten door de ASCII-waarde van " De Basic-regel gaat niet verder en wordt daarom afgesloten door 00 op adres 2014. 2015 en 2016 geven weer het begin van de volgende Basic-regel op adres 2023. Nu is ook te zien dat de pointer op adres 2007 en 2008 naar het adres wijst waar deze pointer staat. 2017 en 2018 geven het regelnummer. 0014 = regel 20. 2019 PRINT token 201A ASCII-waarde van " 201B t/m 2020 de ASCII-waarde van de string 123456 De regel wordt nu weer afgesloten met 00 op adres 2022. Er volgt nu geen Basic-regel meer, dus de regelpointer wordt op 0000 gezet en wel op adres 2023 en 2024. 2025-202F zijn gevuld met AA wat een restant is van de geheugentest die de Basic uitvoert bij een koude start.

Bij de SYM-Basic is de opbouw van het programma hetzelfde. Alleen het begin van het vrije Basic-geheugen is lager gekozen, en wel vanaf adres 2001. Een voorwaarde voor het uitwisselen van programma's is nu dat elk programma begint begint met een 'dummy'-regel, bijvoorbeeld: 1 REM***** Het doel van deze regel is om ruimte te maken om het begin van het programma te kunnen verschuiven.

Methode van inlezen:

De Basic wordt eerst koud opgestart en daarna wordt teruggegaan naar de JUNIOR-mode. Het programma wordt nu gewoon ingelezen, het adres vlak na de laatste 00 moet nu op adres 0075 (low byte) en 0076 (high byte) worden ingevuld. Nu moet het begin van het Basic-programma aangepast worden aan de eigen Basic door het begin te verplaatsen. In het geval van een SYM-programma met AIM-65 Basic wordt adres 2001 tot 2007 verplaatst naar adres 2007 tot 200C waarbij het aantal sterren (hex 2A) kleiner wordt. Na het warm opstarten van de Basic is het aangepaste programma gereed om te draaien.

Bij een AIM-65 programma met SYM-Basic moet adres 2007 tot 200C verplaatst worden naar adres 2001 tot 2007 waarbij de vrij gekomen plaatsen opgevuld moeten worden met sterren. De pointer die het einde van het Basic-werkgeheugen aangeeft staat bij de SYM-Basic op adres 007D (low byte) en 007E (high byte).

 * 12 K FATE for JUNIOR-computer *
 * ***** *
 * * *
 * Intelligent programmers has the capability to change *
 * the complete source-listing of this professional and *
 * very attractive Format Lister/Assembler/Tape-utili- *
 * ties to their own computers. *
 * * *
 * Owners of the Junior-computer need an extended area. *
 * From \$2000 until \$5000 for FATE itself, and a work- *
 * space until \$9000 or more is the best. With FATE it *
 * is possible to assemble from one or two controlled *
 * tape-recorders. You can convert source-files to ob- *
 * ject-files, from cassette to cassette, from memory *
 * to memory, from cassette to memory and vice versa. *
 * The merge and solitfunctions make it possible to *
 * merge two editorfiles into one file, or to solit in- *
 * to two files. *
 * * *
 * FATE is originally based on the object-code of the *
 * Proton Electronics (Naarden, The Netherlands) disk- *
 * operating system SENIOR for the JUNIOR-computer as *
 * described in edition number 26 of DE 6502 KENNER pa- *
 * ges 6 - 15, May, 1983. *
 * This routine was developed by Rob Banen. Copyrights *
 * on assembler and editor by Proton Electronics. Both *
 * the patches and debugging on assembler and editor as *
 * well as writing the Dutch version of the manual and *
 * comments in source-listing by Rob Banen. *
 * Publication of this routine with permission of Pro- *
 * ton Electronics. *
 * Further distribution by any way of copying is not *
 * allowed. *
 * * *
 * The manual of FATE has been translated in English by *
 * Jaap de Hood, May 1985, Longjumeau, France. *
 * * *
 * Eurocheque *
 * Manual Dutch version HFL 35,00 *
 * Manual English version HFL 35,00 *
 * 12K Source-listing English version HFL 110,00 *
 * Cassette for JUNIOR-computer HFL 12,50 *
 * (only with Manual or source-listing) *

```

10 REM ----- 'HET THOMAS-KILMAN INSTRUMENT'
20 REM ----- COMPUTER-BEWERKING DOOR :
30 REM ----- ++++ H A V I S O F T ++++
40 REM ----- HANkel VISSchedijk SOFTware
50 REM ----- (V1.3 ; Version 19-01-85)
60 PRINTCHR$(12)
70 PRINT"H E T T H O M A S - K I L M A N I N S T R U M E N T "
80 PRINT"-----":PRINT
90 PRINT"Inleiding:"
100 PRINT"Het Thomas-Kilman Instrument is een test, die gebruikt wordt"
110 PRINT"in het bedrijfsleven bij psychologische onderzoek in het"
120 PRINT"kader van personeelselectie en -werving."
130 PRINT"De methode wordt voornamelijk toegepast bij hogere"
140 PRINT"functies, en analyseert de leidinggevende eigenschappen van"
150 PRINT"de betrokkene. Niettemin is het ook heel aardig om deze test"
160 PRINT"in 'huiselijke' kring te doen, bijv. met je partner."
170 PRINT"Wel dienen de uitkomsten dan gerelativeerd te worden !!"
180 INPUT"Tik spatie + return voor vervolg";DUMMY$:PRINT
190 PRINT"De methode:"
200 PRINT"Er worden 30 multiple-choice vragen gesteld, waarop"
210 PRINT"antwoord gegeven moet worden. Na de beantwoording verwerkt het"
220 PRINT"programma de verkregen antwoorden tot eindresultaten.
230 PRINT"De verkregen antwoorden worden verdeeld in score's per groep."
240 PRINT"Er zijn vijf groepen, namelijk:"
250 PRINT"*** DOORDRUKKEN ***"
260 PRINT"*** SAMENWERKEN ***"
270 PRINT"*** COMPROMIS ZOEKEN ***"
280 PRINT"*** VERMIJDEN ***"
290 PRINT"*** TOEGEVEN ***"
300 INPUT"Tik spatie + return voor vervolg";DUMMY$:PRINT
310 PRINT"Wanneer er in 1 groep een score van 7 of hoger gescoord"
320 PRINT"wordt, mag genoemde groep als relevant voor Uw karakter"
330 PRINT"gezien worden."
340 PRINT"Bovendien geeft het programma een opsomming van eigenschappen"
350 PRINT"die gelden voor een bepaalde groep."
360 INPUT"Tik spatie en return voor start .....";DUMMY$
370 REM HOOFDPROGRAMMA BEGINT HIER !
380 GOSUB1000: REM KREEER & INITIALISEER
390 GOSUB1090: REM INPUT
400 GOSUB430: REM VERWERK VERKREGEN GEGEVENS & TOELICHTING
410 END
420 REM SUBROUTINE VERWERK VERKREGEN GEGEVENS
430 PRINT CHR$(12)
440 PRINT"De verkregen gegevens worden nu verwerkt .":PRINT
450 FOR I= 1 TO 30
460 IF Z(I)=2 THEN 500
470 REM 'A'-BEHANDELING
480 Q=A(I) : R(Q)=R(Q)+1 : GOTO 510
490 REM 'B'-BEHANDELING
500 Q=B(I) : R(Q)=R(Q)+1
510 NEXT
520 PRINT CHR$(12)
530 PRINTTAB(25);"U W S C O R E ' S "
540 PRINTTAB(25);"-----" : PRINT

```

```

550 PRINT"Doordrukken";SPC(8);"Samenwerken";SPC(8);
560 PRINT"Compromis";SPC(8);"Vermijden";SPC(8);"Toegeven"
570 PRINTCHR$(11);
580 PRINTSPC(19);"(exploreren)";SPC(7);"zoeken";SPC(11);
590 PRINT"(terug-"
600 PRINTSPC(38);"(delen)";SPC(10);"trekken)":PRINT
610 PRINTTAB(4);R(1);TAB(23);R(2);TAB(39);R(3);
620 PRINTTAB(57);R(4);TAB(74);R(5)
630 PRINT"Score's van 7 en hoger kunnen als relevant beschouwd worden.":PRINT
640 PRINT" Tik spatie + return voor toelichting op de score's";
650 INPUT DUMMY$:PRINT CHR$(12)::I=0
660 I=I+1: IF I=6 THEN RETURN
670 IF R(I)<7 THEN 660
680 ON I GOSUB 690,750,800,860,920: PRINT : GOTO660
690 PRINT"** DOORDRUKKEN **"
700 PRINT" - zelf problemen van de medewerkers oplossen"
710 PRINT" - het oplossen van oplossingen"
720 PRINT" - zelf alle verantwoordelijkheid"
730 PRINT" - nemen voor de oplossing"
740 PRINT" - instrueren" : RETURN
750 PRINT"** SAMENWERKEN **"
760 PRINT" - ruimte geven en ruimte invullen"
770 PRINT" - oplossingen bereiken door gelijkwaardige inbreng"
780 PRINT" - wederzijdse openheid"
790 PRINT" - oplossingen bereiken door samenspel" : RETURN
800 PRINT"** COMPROMIS ZOEKEN **"
810 PRINT" - suggesties aan de hand doen"
820 PRINT" - advies geven op basis van deskundigheid"
830 PRINT" - als 'objectieve buitenstaander'"
840 PRINT" - oplossingen aanreiken"
850 PRINT" - de ander keuze-alternatieven laten" : RETURN
860 PRINT"** VERMIJDEN **"
870 PRINT" - niet diep ingaan op de problemen"
880 PRINT" - je terughoudend opstellen"
890 PRINT" - problemen op z'n beloop laten"
900 PRINT" - communicatie beperken"
910 PRINT" - afwachten" : RETURN
920 PRINT"** TOEGEVEN **"
930 PRINT" - de medewerkers bepalen zelf hoe ze"
940 PRINT" - aan het probleem werken"
950 PRINT" - veel luisteren, laten uitspreken"
960 PRINT" - de begeleiding hangt af van de medewerkers"
970 PRINT" - frustraties vermijden"
980 PRINT" - geen eigen meningen geven" : RETURN
990 REM KREEER & INITIALISEER
1000 DIM A(30) : REM 'A'-TABEL
1010 DIM B(30) : REM 'B'-TABEL
1020 DIM Z(30) : REM ANTWOORDEN-TABEL
1030 DIM R(5) : REM RESULTAAT-ARRAY
1040 REM INITIALISEER DE ARRAY'S
1050 FOR I = 1 TO 30 : READ A(I) : NEXT
1060 FOR I = 1 TO 30 : READ B(I) : NEXT
1070 RETURN
1080 REM INPUT
1090 GOSUB2910
1100 PRINT"A. Soms laat ik anderen de verantwoordelijkheid nemen om"

```

```
1110 PRINT" het probleem op te lossen.":PRINT
1120 PRINT"B. Liever dan te onderhandelen over de dingen waarover we"
1130 PRINT" het niet eens zijn, probeer ik de nadruk te leggen op"
1140 PRINT" de dingen waarover we het wel eens zijn."
1150 GOSUB2840
1160 GOSUB2910
1170 PRINT"A. Ik probeer een compromis-oplossing te vinden.":PRINT
1180 PRINT"B. Ik probeer zijn en al mijn belangen te behartigen."
1190 GOSUB2840
1200 GOSUB2910
1210 PRINT"A. Ik ben doorgaans vastberaden in het nastreven van"
1220 PRINT" mijn doeleinden.":PRINT
1230 PRINT"B. Soms probeer ik de ander te sussen om onze relatie"
1240 PRINT" goed te houden."
1250 GOSUB2840
1260 GOSUB2910
1270 PRINT"A. Ik probeer een compromis-oplossing te vinden.":PRINT
1280 PRINT"B. Soms geef ik mijn eigen wensen op ten behoeve"
1290 PRINT" van de ander."
1300 GOSUB2840
1310 GOSUB2910
1320 PRINT"A. Ik vraag konsekwent de hulp van een ander bij het"
1330 PRINT" bedenken van een oplossing.":PRINT
1340 PRINT"B. Ik probeer datgene te doen wat noodzakelijk is om"
1350 PRINT" overbodige spanningen te vermijden."
1360 GOSUB2840
1370 GOSUB2910
1380 PRINT"A. Ik probeer te vermijden het voor mijzelf onprettig"
1390 PRINT" te maken":PRINT
1400 PRINT"B. Ik probeer mijn standpunt af te dwingen."
1410 GOSUB2840
1420 GOSUB2910
1430 PRINT"A. Ik probeer de kwestie uit te stellen tot ik enige"
1440 PRINT" tijd gehad heb om er over na te denken.":PRINT
1450 PRINT"B. Ik geef sommige punten op in ruil voor andere."
1460 GOSUB2840
1470 GOSUB2910
1480 PRINT"A. Ik ben doorgaans vastberaden in het nastreven van"
1490 PRINT" mijn doeleinden.":PRINT
1500 PRINT"B. Ik probeer alle belangen en geschilpunten onmiddellijk"
1510 PRINT" boven tafel te krijgen."
1520 GOSUB2840
1530 GOSUB2910
1540 PRINT"A. Ik vind verschillen niet altijd de moeite waard om je"
1550 PRINT" druk over te maken.":PRINT
1560 PRINT"B. Ik span mij wel enigszins in om mijn zin te krijgen."
1570 GOSUB2840
1580 GOSUB2910
1590 PRINT"A. Ik ben vastberaden in het nastreven van mijn"
1600 PRINT" doeleinden.":PRINT
1610 PRINT"B. Ik probeer een compromis-oplossing te vinden."
1620 GOSUB2840
1630 GOSUB2910
1640 PRINT"A. Ik probeer alle belangen en geschilpunten onmiddellijk"
1650 PRINT" boven tafel te krijgen.":PRINT
1660 PRINT"B. Soms probeer ik de ander te sussen om onze relatie"
```

```
1670 PRINT" goed te houden."
1680 GOSUB2840
1690 GOSUB2910
1700 PRINT"A. Soms vermijd ik het innemen van standpunten die tot"
1710 PRINT" controverse zouden kunnen leiden.":PRINT
1720 PRINT"B. Ik laat de ander sommige van zijn standpunten, als"
1730 PRINT" hij mij sommige van mij laat."
1740 GOSUB2840
1750 GOSUB2910
1760 PRINT"A. Ik stel voor elkaar halverwege te ontmoeten.":PRINT
1770 PRINT"B. Ik ga er stevig tegenaan om mijn visie over te dragen."
1780 GOSUB2840
1790 GOSUB2910
1800 PRINT"A. Ik vertel de ander mijn gedachten en vraag hem de"
1810 PRINT" de zijne.":PRINT
1820 PRINT"B. Ik probeer hem de logica en voordelen van mijn"
1830 PRINT" standpunt te tonen."
1840 GOSUB2840
1850 GOSUB2910
1860 PRINT"A. Soms probeer ik de ander te sussen om onze relatie"
1870 PRINT" goed te houden.":PRINT
1880 PRINT"B. Ik probeer datgene te doen wat noodzakelijk is om"
1890 PRINT" spanningen te vermijden."
1900 GOSUB2840
1910 GOSUB2910
1920 PRINT"A. Ik probeer de gevoelens van de ander niet te kwetsen.":PRINT
1930 PRINT"B. Ik probeer de ander te overtuigen van de voordelen"
1940 PRINT" van mijn standpunt."
1950 GOSUB2840
1960 GOSUB2910
1970 PRINT"A. Ik ben doorgaans vastberaden in het nastreven van"
1980 PRINT" mijn doeleinden.":PRINT
1990 PRINT"B. Ik probeer datgene te doen wat noodzakelijk is om"
2000 PRINT" overbodige spanningen te vermijden."
2010 GOSUB2840
2020 GOSUB2910
2030 PRINT"A. Als het de ander gelukkig maakt, laat ik hem zijn"
2040 PRINT" gezichtspunten.":PRINT
2050 PRINT"B. Ik laat hem sommige van zijn standpunten, als hij mij"
2060 PRINT" sommige van de mijne laat."
2070 GOSUB2840
2080 GOSUB2910
2090 PRINT"A. Ik probeer alle belangen en geschilpunten onmiddellijk"
2100 PRINT" boven tafel te krijgen.":PRINT
2110 PRINT"B. Ik probeer de kwestie uit te stellen tot ik enige"
2120 PRINT" tijd heb gehad om erover na te denken."
2130 GOSUB2840
2140 GOSUB2910
2150 PRINT"A. Ik probeer onze verschillen onmiddellijk grondig door"
2160 PRINT" te nemen.":PRINT
2170 PRINT"B. Ik probeer een billijke combinatie van winst en"
2180 PRINT" verlies voor ons beiden te vinden."
2190 GOSUB2840
2200 GOSUB2910
2210 PRINT"A. In mijn aanpak van onderhandelingen probeer ik"
2220 PRINT" rekening te houden met de wensen van de ander.":PRINT
```

```
2230 PRINT"B. Ik neig er altijd toe het probleem direkt te bespreken."  
2240 GOSUB2840  
2250 GOSUB2910  
2260 PRINT"A. Ik probeer een standpunt te vinden dat het midden houdt"  
2270 PRINT" tussen het zijne en het mijne.":PRINT  
2280 PRINT"B. Ik kom op voor mijn wensen."  
2290 GOSUB2840  
2300 GOSUB2910  
2310 PRINT"A. Ik stel mij erg vaak ten doel al onze wensen te"  
2320 PRINT" bevredigen":PRINT  
2330 PRINT"B. Soms laat ik anderen de verantwoordelijkheid nemen om"  
2340 PRINT" het probleem op te lossen."  
2350 GOSUB2840  
2360 GOSUB2910  
2370 PRINT"A. Als het standpunt van de ander erg belangrijk voor hem"  
2380 PRINT" lijkt, zal ik proberen aan zijn wensen tegemoet te"  
2390 PRINT" komen.":PRINT  
2400 PRINT"B. Ik probeer hem ertoe te krijgen genoegen te nemen met"  
2410 PRINT" een compromis."  
2420 GOSUB2840  
2430 GOSUB2910  
2440 PRINT"A. Ik probeer hem de logica en de voordelen van mijn"  
2450 PRINT" standpunten te tonen.":PRINT  
2460 PRINT"B. In mijn aanpak van onderhandelingen probeer ik"  
2470 PRINT" rekening te houden met de wensen van de ander."  
2480 GOSUB2840  
2490 GOSUB2910  
2500 PRINT"A. Ik stel voor elkaar halverwese te ontmoeten.":PRINT  
2510 PRINT"B. Ik stel mij bijna altijd ten doel al onze wensen te"  
2520 PRINT" bevredigen."  
2530 GOSUB2840  
2540 GOSUB2910  
2550 PRINT"A. Soms vermijd ik het innemen van standpunten die tot"  
2560 PRINT" controversen zouden leiden.":PRINT  
2570 PRINT"B. Als het de ander gelukkig maakt, laat ik hem wellicht"  
2580 PRINT" zijn ideeën over de zaak."  
2590 GOSUB2840  
2600 GOSUB2910  
2610 PRINT"A. Ik ben gewoonlijk vastberaden in het nastreven van"  
2620 PRINT" mijn doeleinden.":PRINT  
2630 PRINT"B. Gewoonlijk vraag ik de hulp van de ander bij het"  
2640 PRINT" uitdenken van een oplossing."  
2650 GOSUB2840  
2660 GOSUB2910  
2670 PRINT"A. Ik stel voor elkaar halverwese te ontmoeten.":PRINT  
2680 PRINT"B. Ik vind verschillen niet altijd de moeite waard om je"  
2690 PRINT" druk over te maken."  
2700 GOSUB2840  
2710 GOSUB2910  
2720 PRINT"A. Ik probeer de gevoelens van de ander niet te kwetsen.":PRINT  
2730 PRINT"B. Ik maak de ander altijd deelgenoot van het probleem,"  
2740 PRINT" zodat we het uit kunnen werken."  
2750 GOSUB2840  
2760 RETURN  
2770 REM DATA VOOR 'A'-TABEL  
2780 DATA 4, 3, 1, 3, 2, 4, 4, 1, 4, 1, 2, 4, 3, 2, 5
```

```

2790 DATA 5,1,5,2,2,5,3,2,5,1,3,4,1,3,5
2800 REM DATA VOOR 'B'-TABEL
2810 DATA 5,2,5,5,4,1,3,2,1,3,5,3,1,1,4
2820 DATA 1,4,3,4,3,2,1,4,3,5,2,5,2,4,2
2830 REM SUBROUTINE DIE OM ANTWOORD VRAAGT EN DIT OPSLAAT IN TABEL
2840 PRINT:PRINT : INPUT"Uw antwoord (A of B) :";Y$
2850 Y$=LEFT$(Y$,1)
2860 Y=ASC(Y$) : Y=Y-64
2870 IF Y(<)1 AND Y(<)2 THEN 2840
2880 Z(NU)=Y : RETURN
2890 REM SUBROUTINE OM HET BEELDSCHERM SCHOON TE MAKEN
2900 REM EN VRAAGNUMMER TE PRINTEN.
2910 PRINT CHR$(12);
2920 PRINT"Vraag :";
2930 NU=NU+1
2940 PRINT NU : PRINT : RETURN

```

OK

Voor velen is het een wekelijks bezigheid om de lotto formulieren in te vullen. Indien u niet steevast dezelfde getallen invult is misschien dit programma een leuk hulpmiddel om de computer deze getallen voor u te laten "invullen".

<pre> 10 REM SCHOONMAKEN BEELDSCHERM 20 H=0 30 PRINT"VOOR HOEVEEL WEKEN WILT U DE" 40 INPUT"LOTTO GETALLEN OP HET SCHERM";G 50 A=INT(41*RND(1)+1) 60 B=INT(41*RND(1)+1) 70 C=INT(41*RND(1)+1) 80 D=INT(41*RND(1)+1) 90 E=INT(41*RND(1)+1) 100 F=INT(41*RND(1)+1) 110 IFB=ATHENS0 120 IFC=AORD=BTHEN70 130 IFD=AORD=BORD=CTHEN80 140 IFE=ADRE=BORE=CORE=DTHEN90 150 IFF=AORF=BORF=CORF=DORF=ETHEN100 160 H=H+1 170 PRINTA"--"B"--"C"--"D"--"E"--"F 180 IFH=GTHENEND 190 GOT050 </pre>	<pre> 37 - 22 - 6 - 5 - 7 - 41 40 - 13 - 21 - 30 - 5 - 28 30 - 18 - 29 - 16 - 11 - 10 27 - 9 - 32 - 13 - 28 - 33 5 - 10 - 30 - 4 - 35 - 13 8 - 9 - 41 - 12 - 25 - 15 14 - 2 - 5 - 39 - 10 - 25 35 - 18 - 14 - 22 - 16 - 23 37 - 39 - 2 - 10 - 20 - 17 25 - 40 - 32 - 38 - 20 - 35 37 - 7 - 16 - 22 - 24 - 34 5 - 17 - 33 - 26 - 37 - 3 29 - 38 - 7 - 37 - 8 - 41 19 - 13 - 26 - 11 - 5 - 25 27 - 41 - 18 - 34 - 38 - 25 23 - 20 - 1 - 28 - 21 - 26 13 - 25 - 26 - 24 - 40 - 20 24 - 38 - 27 - 9 - 18 - 3 </pre>
--	---

.DLEEBLEGEIPS NI SELLA TFJIRHCS AMMARGORP ETROK REEZ TID

Nee dit is geen drukfout of rariteiten van mijn tekstverwerker. Als u het volgende miniprogrammaatje intoetst en 'RUN't kan u tekst invoeren. Daarna de 'CR' of 'RETURN' toets indrukken en het geheel verschijnt in spiegelbeeld op het beeldscherm.

```

1 INPUTA$:FORA=LEN(A$)TO1STEP-1:PRINTMID$(A$,A,1);:NEXTA

```

STONE

PAGE 01

```

0010: *****
0020: *** DUBBELE DOBBELSTEEN ***
0030: *****
0040: DOOR: M. LAMEIJ
0050: PRINS CLAUSLAAN 81
0060: 7242 GN LOCHEM
0070: TEL.: 05730 - 4427.
0080: *****
0090:
0100: >>>> GEBRUIKSAANWIJZING <<<<
0110: HET PROGRAMMA SCHRIJFT OP HET SCHERM
0120: DE AFBEELDING VAN TWEE DOBBELSTENEN
0130: NAAST ELKAAR. HET TOTAAL AANTAL OGEN
0140: WORDT ONDER DE RECHTER-DOBBELSTEEN
0150: AFGEDRUKT. WANNEER ER EEN DUBBEL UIT DE
0160: BUS KOMT WORDT DIT AANGEGEVEN. OP POORT
0170: PBO VAN DE VIA KAN EEN LUIDSPREKER-
0180: SCHAKELING WORDEN AANGESLOTEN: EEN
0190: PIEPTOONTJE GEEFT AAN DAT DE COMPUTER
0200: KLAAR IS VOOR DE VOLGENDE GOOI. MET DE
0210: TOETSEN 'J' EN 'N' KAN DAN GEKOZEN
0220: WORDEN VOOR DOORGAAN OF STOPPEN.
0230: DE DATA VOOR HET BEELD WORDT DOOR DE
0240: COMPUTER OPGEBOUWD OP PAGINA 9F. BIJ
0250: MIJN SYSTEEM DE LAATSTE RAM-PAGINA. ER
0260: KAN NATUURLIJK EEN ANDERE PAGINA
0270: GEKOZEN WORDEN.
0280:
0290: PAGINA NUL LOKATIES:
0300: 09 00 SWITCH * $0009 EERSTE/TWEDE GOOI
0310: 12 00 GETAL * $0012 GETAL GOOI
0320: 70 00 COUNT * $0070 FAKTOR SOUND
0330:
0340: PIA LOKATIES:
0350: D5 1A RDFLAG * $1AD5
0360: F7 1A TIMER * $1AF7
0370:
0380: VIA LOKATIES:
0390: 00 18 ORB * $1800 I/O POORT
0400: 02 18 DDRB * $1802 RICHTINGS REGISTER
0410:
0420: MONITOR SUBROUTINES:
0430: 5F 10 LABJUN * $105F
0440: E8 11 CRLF * $11E8
0450: A4 12 NIBASC * $12A4
0460: AE 12 RECCHA * $12AE
0470: 34 13 PRCHA * $1334
0480:
0490: PROGRAMMA LOKATIES:
0500: FE 9F SCOREA * $9FFE SCORE
0510: FF 9F SCOREB * $9FFF SCORE
0520:
0530: PROGRAMMA-BEGIN
0540: 0200 STONE ORG $0200
0550: 0200 A9 00 START LDAM $00
0560: 0202 85 09 STA SWITCH RESET SWITCH
0570: 0204 A9 0C LDAM $0C
0580: 0206 20 B0 03 JSR CLRSCR WIS SCHERM
0590: 0209 20 A9 03 JSR RANBEE GETAL + AFBEELDEN
0600: 020C AD FE 9F LDA SCOREA
0610: 020F 8D FF 9F STA SCOREB VERPLAATS EERSTE GOOI
0620: 0212 20 FF 02 JSR DISPLA DRUK AF
0630: 0215 E6 09 INC SWITCH VERZET SWITCH
0640: 0217 A9 1C LDAM $1C

```

```

0650: 0219 20 B0 03 JSR CLRSCR CURSOR HOME
0660: 021C 20 A9 03 JSR RANBEE GETAL + BEELD
0670: 021F 20 2B 03 JSR TOTAAL BEPAAL TOTAAL
0680: 0222 20 FF 02 JSR DISPLA DRUK AF
0690: 0225 20 63 03 JSR DUBBEL DUBBEL?
0700: 0228 20 BE 03 JSR SOUND LAAT TOON HOREN
0710: 022B 20 0B 04 JSR QUEST NOG EENS?
0720: 022E 20 AE 12 STARTB JSR RECCHA WACHT OP TOETS
0730: 0231 C9 4A CMPIM 'J IS HET J?
0740: 0233 F0 CB BEQ START ZOJA OPNIEUW
0750: 0235 C9 4E CMPIM 'N IS HET N?
0760: 0237 F0 02 BEQ STARTC ZOJA STOP
0770: 0239 D0 F3 BNE STARTB ZONEE NAAR TOETS?
0780: 023B 4C 5F 10 STARTC JMP LABJUN STOP ERMEE
0790:
0800: 023E A0 C0 PREDIS LDYIM $C0
0810: 0240 A9 20 LDAIM $20 SCHRIJF SPATIES
0820: 0242 8D 00 9F STA $9F00 OP DE
0830: 0245 99 00 9F PREA STAA Y $9F00 BEELDPAGINA
0840: 0248 88 DEY
0850: 0249 D0 FA BNE PREA
0860: 024B A9 03 LDAIM $03
0870: 024D 8D 9F 9F STA $9F9F PLAATS EOT-TEKEN
0880: 0250 A0 00 LDYIM $00
0890: 0252 A9 5F LDAIM $5F BOUW BEELD OP
0900: 0254 99 10 9F PREB STAA Y $9F10
0910: 0257 C8 INY
0920: 0258 C0 10 CPYIM $10
0930: 025A D0 F8 BNE PREB
0940: 025C A0 00 LDYIM $00
0950: 025E A9 5F LDAIM $5F
0960: 0260 99 71 9F PREC STAA Y $9F71
0970: 0263 C8 INY
0980: 0264 C0 0E CPYIM $0E
0990: 0266 D0 F8 BNE PREC
1000: 0268 A9 49 LDAIM 'I
1010: 026A 8D 20 9F STA $9F20
1020: 026D 8D 30 9F STA $9F30
1030: 0270 8D 40 9F STA $9F40
1040: 0273 8D 50 9F STA $9F50
1050: 0276 8D 60 9F STA $9F60
1060: 0279 8D 70 9F STA $9F70
1070: 027C 8D 2F 9F STA $9F2F
1080: 027F 8D 3F 9F STA $9F3F
1090: 0282 8D 4F 9F STA $9F4F
1100: 0285 8D 5F 9F STA $9F5F
1110: 0288 8D 6F 9F STA $9F6F
1120: 028B 8D 7F 9F STA $9F7F
1130:
1140: 028E A5 12 HOWMUC LDA GETAL HAAL GOOI OP
1150: 0290 8D FE 9F STA SCOREA EN BERG OP
1160: 0293 C9 01 CMPIM $01 IS HET EEN 1?
1170: 0295 F0 17 BEQ PLACEA ZOJA PLAATSSEN
1180: 0297 C9 02 CMPIM $02 IS HET EEN 2?
1190: 0299 F0 19 BEQ PLACEB ZOJA PLAATSSEN
1200: 029B C9 03 CMPIM $03 IS HET EEN 3?
1210: 029D F0 1E BEQ PLACEC ZOJA PLAATSSEN
1220: 029F C9 04 CMPIM $04 IS HET EEN 4?
1230: 02A1 F0 26 BEQ PLACED ZOJA PLAATSSEN
1240: 02A3 C9 05 CMPIM $05 IS HET EEN 5?
1250: 02A5 F0 31 BEQ PLACEE ZOJA PLAATSSEN
1260: 02A7 C9 06 CMPIM $06 IS HET EEN 6?
1270: 02A9 F0 3F BEQ PLACEF ZOJA PLAATSSEN
1280: 02AB 4C A9 03 JMP RANBEE ZONEE OPNIEUW

```

```

1290:
1300: 02AE A9 4F          PLACEA LDAIM '0
1310: 02B0 8D 48 9F      STA $9F48  PLAATS EEN 0
1320: 02B3 60              RTS
1330: 02B4 A9 4F          PLACEB LDAIM '0
1340: 02B6 8D 62 9F      STA $9F62
1350: 02B9 8D 2D 9F      STA $9F2D  PLAATS 2 0'S
1360: 02BC 60              RTS
1370: 02BD A9 4F          PLACEC LDAIM '0
1380: 02BF 8D 48 9F      STA $9F48
1390: 02C2 8D 62 9F      STA $9F62
1400: 02C5 8D 2D 9F      STA $9F2D  PLAATS 3 0'S
1410: 02C8 60              RTS
1420: 02C9 A9 4F          PLACED LDAIM '0
1430: 02CB 8D 62 9F      STA $9F62
1440: 02CE 8D 22 9F      STA $9F22
1450: 02D1 8D 2D 9F      STA $9F2D
1460: 02D4 8D 6D 9F      STA $9F6D  PLAATS 4 0'S
1470: 02D7 60              RTS
1480: 02D8 A9 4F          PLACEE LDAIM '0
1490: 02DA 8D 48 9F      STA $9F48
1500: 02DD 8D 62 9F      STA $9F62
1510: 02E0 8D 22 9F      STA $9F22
1520: 02E3 8D 2D 9F      STA $9F2D
1530: 02E6 8D 6D 9F      STA $9F6D  PLAATS 5 0'S
1540: 02E9 60              RTS
1550: 02EA A9 4F          PLACEF LDAIM '0
1560: 02EC 8D 28 9F      STA $9F28
1570: 02EF 8D 2D 9F      STA $9F2D
1580: 02F2 8D 62 9F      STA $9F62
1590: 02F5 8D 22 9F      STA $9F22
1600: 02F8 8D 67 9F      STA $9F67
1610: 02FB 8D 6D 9F      STA $9F6D  PLAATS 6 0'S
1620: 02FE 60              RTS
1630:
1640: 02FF A0 00          DISPLA LDYIM $00
1650: 0301 98          DISPLB TYA
1660: 0302 29 0F          ANDIM $0F
1670: 0304 F0 0E          BEQ DISPLD BEPAAL EERSTE/TWEEDE
1680: 0306 B9 00 9F      DISPLC LDAAY $9F00 HAAL BEELD OP
1690: 0309 C9 03          CMPIM $03 TOT HET EOT-TEKEN
1700: 030B F0 1D          BEQ DISPLG DAN STOPPEN
1710: 030D 20 34 13      JSR PRCHA ANDER AFDRUKKEN
1720: 0310 C8              INY
1730: 0311 4C 01 03      JMP DISPLB EN DOORGAAN
1740: 0314 20 E8 11      DISPLD JSR CRLF
1750: 0317 A5 09          LDA SWITCH BEPAAL WELKE GOOI
1760: 0319 29 01          ANDIM $01
1770: 031B F0 0A          BEQ DISPLF EERSTE GOOI
1780: 031D A2 23          LDXIM $23 TWEEDE GOOI
1790: 031F A9 09          DISPLE LDAIM $09
1800: 0321 20 34 13      JSR PRCHA DRUK HOR. TAB. AF
1810: 0324 CA              DEX 35 KEER
1820: 0325 D0 F8          BNE DISPLE
1830: 0327 4C 06 03      DISPLF JMP DISPLC DRUK TWEEDE GOOI
1840: 032A 60          DISPLG RTS GA DOOR
1850:
1860: 032B A9 54          TOTAAL LDAIM 'T PLAATS TOTAAL
1870: 032D 8D 96 9F      STA $9F96
1880: 0330 8D 98 9F      STA $9F98
1890: 0333 A9 4F          LDAIM '0
1900: 0335 8D 97 9F      STA $9F97
1910: 0338 A9 41          LDAIM 'A
1920: 033A 8D 99 9F      STA $9F99

```

1930:	033D	8D	9A	9F		STA	\$9F9A	
1940:	0340	A9	4C			LDAIM	'L	
1950:	0342	8D	9B	9F		STA	\$9F9B	
1960:	0345	F8				SED		BEREKEN TOTAAL
1970:	0346	18				CLC		
1980:	0347	AD	FF	9F		LDA	SCOREB	
1990:	034A	6D	FE	9F		ADC	SCOREA	
2000:	034D	D8				CLD		
2010:	034E	48				PHA		
2020:	034F	4A				LSRA		
2030:	0350	4A				LSRA		
2040:	0351	4A				LSRA		
2050:	0352	4A				LSRA		
2060:	0353	20	A4	12		JSR	NIBASC	ZET TOTAAL
2070:	0356	8D	9D	9F		STA	\$9F9D	OM IN ASCII EN
2080:	0359	68				PLA		ZET HET IN BEELD
2090:	035A	29	0F			ANDIM	\$0F	
2100:	035C	20	A4	12		JSR	NIBASC	
2110:	035F	8D	9E	9F		STA	\$9F9E	
2120:	0362	60				RTS		GA DOOR
2130:								
2140:	0363	AD	FF	9F	DUBBEL	LDA	SCOREB	HAAL GOOI OP
2150:	0366	CD	FE	9F		CMP	SCOREA	GELIJK?
2160:	0369	D0	3D			BNE	DUBBF	ZONEE GA DOOR
2170:	036B	20	E8	11		JSR	CRLF	NIEUWE REGEL
2180:	036E	A2	16		DUBBA	LDXIM	\$16	
2190:	0370	A9	09		DUBBB	LDAIM	\$09	DRUK HOR. TAB. AF
2200:	0372	20	34	13		JSR	PRCHA	
2210:	0375	CA				DEX		22 KEER
2220:	0376	D0	F8			BNE	DUBBB	
2230:	0378	A0	00			LDYIM	\$00	
2240:	037A	B9	1F	04	DUBBC	LDAAY	MESSAG	
2250:	037D	C9	03			CMPIM	\$03	
2260:	037F	F0	07			BEQ	DUBBD	
2270:	0381	20	34	13		JSR	PRCHA	DRUK AF 'DUBBEL'
2280:	0384	C8				INY		
2290:	0385	4C	7A	03		JMP	DUBBC	
2300:	0388	20	B3	03	DUBBD	JSR	DELAY	WACHT EVEN
2310:	038B	E6	09			INC	SWITCH	BEPAALT OOK HET
2320:	038D	A5	09			LDA	SWITCH	AANTAL MALEN
2330:	038F	C9	07			CMPIM	\$07	DAT DUBBEL WORDT
2340:	0391	F0	15			BEQ	DUBBF	AFGEDRUKT
2350:	0393	A2	0A			LDXIM	\$0A	
2360:	0395	A9	08		DUBBE	LDAIM	\$08	
2370:	0397	20	34	13		JSR	PRCHA	WIS DUBBEL
2380:	039A	CA				DEX		
2390:	039B	D0	F8			BNE	DUBBE	
2400:	039D	A9	0D			LDAIM	\$0D	
2410:	039F	20	34	13		JSR	PRCHA	BEGIN VOORAAN REGEL
2420:	03A2	20	B3	03		JSR	DELAY	
2430:	03A5	4C	6E	03		JMP	DUBBA	OPNIEUW
2440:	03A8	60			DUBBF	RTS		GA DOOR
2450:								
2460:	03A9	20	DC	03	RANBEE	JSR	RAND	BEPAAAL GETAL
2470:	03AC	20	3E	02		JSR	PREDIS	MAAK BEELD KLAAR
2480:	03AF	60				RTS		GA DOOR
2490:								
2500:	03B0	20	34	13	CLRSCR	JSR	PRCHA	WIS SCHERM
2510:	03B3	A9	80		DELAY	LDAIM	\$80	WACHT EVEN
2520:	03B5	8D	F7	1A		STA	TIMER	
2530:	03B8	2C	D5	1A	DELA	BIT	RDFLAG	TIJD OM?
2540:	03BB	10	FB			BPL	DELA	ZONEE WACHT NOG
2550:	03BD	60				RTS		ZOJA WEER VERDER
2560:								

STONE

PAGE 05

```

2570: 03BE A9 01      SOUND LDAIM $01
2580: 03C0 8D 00 18   STA ORB      STEL POORTEN IN
2590: 03C3 8D 02 18   STA DDRB
2600: 03C6 A9 7F      LDAIM $7F
2610: 03C8 85 70      STA COUNT    STEL VERTRAGING IN
2620: 03CA EE 00 18   SOUNA INC ORB      ZET STEEDS
2630: 03CD A6 70      LDX COUNT    DE UITGANG AAN
2640: 03CF E8          SOUNB INX      EN WEER UIT
2650: 03D0 D0 FD      BNE SOUNB
2660: 03D2 C6 70      DEC COUNT
2670: 03D4 C6 70      DEC COUNT
2680: 03D6 30 03      BMI SOUNC
2690: 03D8 4C CA 03   SOUNC JMP SOUNA  OPNIEUW
2700: 03DB 60          SOUNC RTS      GA DOOR
2710:
2720: 03DC 38          RAND SEC
2730: 03DD A5 00      LDAZ $00     HAAL INHOUD 0000
2740: 03DF 65 16      ADC GETAL    +04 TEL 0016 ERBIJ
2750: 03E1 65 17      ADC GETAL    +05 TEL 0017 ERBIJ
2760: 03E3 85 12      STA GETAL    ZET OP 0012
2770: 03E5 A2 04      LDXIM $04
2780: 03E7 B5 12      RANDA LDAZX GETAL  VERSCHUIF GETALLEN
2790: 03E9 95 13      STAZX GETAL  +01
2800: 03EB CA          DEX
2810: 03EC 10 F9      BPL RANDA
2820: 03EE A2 10      LDXIM $10
2830: 03F0 A0 C0      LDYIM $C0
2840: 03F2 84 11      STY GETAL    -01
2850: 03F4 A0 06      LDYIM $06
2860: 03F6 C5 11      RANDB CMP GETAL  -01
2870: 03F8 90 02      BCC RANDC
2880: 03FA E5 11      SBC GETAL    -01
2890: 03FC 46 11      RANDC LSR GETAL    -01
2900: 03FE 88          DEY
2910: 03FF D0 F5      BNE RANDB
2920: 0401 18          CLC
2930: 0402 29 09      ANDIM $09
2940: 0404 95 00      STAZX GETAL  -12
2950: 0406 C6 05      DEC GETAL    -2
2960: 0408 10 D2      BPL RAND
2970: 040A 60          RTS
2980:
2990: 040B 20 E8 11   QUEST JSR CRLF    NIEUWE REGEL
3000: 040E A0 09      LDYIM $09
3010: 0410 B9 1F 04   QUESTA LDAAY MESSAG  HAAL TEKST
3020: 0413 C9 03      CMPIM $03    TOT EOT-TEKEN
3030: 0415 F0 07      BEQ QUESTB   AFDRUKKEN
3040: 0417 20 34 13  JSR PRCHA
3050: 041A CB          INY
3060: 041B 4C 10 04   QUESTA JMP QUESTA  VOLGENDE LETTER
3070: 041E 60          QUESTB RTS
3080:
3090: 041F 44          MESSAG = 'D     DUBBEL!!
3100: 0420 55          = 'U
3110: 0421 42          = 'B
3120: 0422 42          = 'B
3130: 0423 45          = 'E
3140: 0424 4C          = 'L
3150: 0425 21          = '!'
3160: 0426 21          = '!'
3170: 0427 03          = $03
3180: 0428 4E          = 'N     NOG EENS? (J/N)
3190: 0429 4F          = 'O
3200: 042A 47          = 'G
    
```

```

3210: 042B 20
3220: 042C 45
3230: 042D 45
3240: 042E 4E
3250: 042F 53
3260: 0430 3F
3270: 0431 20
3280: 0432 28
3290: 0433 4A
3300: 0434 2F
3310: 0435 4E
3320: 0436 29
3330: 0437 03
    
```

TORENS VAN HANDI

Een der beroemste puzzels uit het verleden is toch wel de TORENS VAN HANDI. Volgens de overlevering zouden bij een klooster aldaar drie palen staan. Om de linkerpaal lasen oorspronkelijk 64 gouden ringen, allen van verschillende diameter en van beneden naar boven van groot naar klein gerangschikt. De taak van de kloosterlingen is om elke dag een ring van een paal naar een andere te verplaatsen, zodanig dat uiteindelijk alle 64 ringen een toren vormen om een andere paal. Nooit mag daarbij een ring op een ring met kleinere diameter komen te liggen. Is de taak volbracht dan versaat de wereld. Gelukkig waren er in die tijd nog geen computers anders hadden we dit spel nooit kunnen spelen.



```

10 REM SCHONMAKEN BEELDSCHERM
20 INPUT "AANTAL SCHIJVEN ? " : N
30 DIM I(N), J(N), B(N)
40 I(N)=1: J(N)=3: K=N
50 IF K=1 THEN GOTO 100
60 I1=I(K): J1=J(K): K=K-1
70 IF B(K+1)=1 THEN GOTO 90
80 I(K)=I1: J(K)=6-I1-J1: GOTO 50
90 I(K)=6-I1-J1: J(K)=J1: GOTO 50
100 IF B(K)=0 THEN GOTO 140
110 B(K)=0: K=K+1
120 IF K=N THEN GOTO 100
130 END
140 PRINT "SCHIJF: " ; K ; " VAN: " ; I(K) ; " NAAR: " ; J(K)
150 B(K)=1: GOTO 50
    
```

```

AANTAL SCHIJVEN ?
SCHIJF: 1 VAN: 1 NAAR: 2
SCHIJF: 2 VAN: 1 NAAR: 3
SCHIJF: 1 VAN: 2 NAAR: 3
SCHIJF: 3 VAN: 1 NAAR: 2
SCHIJF: 1 VAN: 3 NAAR: 1
SCHIJF: 2 VAN: 3 NAAR: 2
SCHIJF: 1 VAN: 1 NAAR: 2
SCHIJF: 4 VAN: 1 NAAR: 3
SCHIJF: 1 VAN: 2 NAAR: 3
SCHIJF: 2 VAN: 2 NAAR: 1
SCHIJF: 1 VAN: 3 NAAR: 1
SCHIJF: 3 VAN: 2 NAAR: 3
SCHIJF: 1 VAN: 1 NAAR: 2
SCHIJF: 2 VAN: 1 NAAR: 3
SCHIJF: 1 VAN: 2 NAAR: 3
    
```

ASSEMBLER FOR FORTH.

G. van Opbroek
Hooglanden 20
9801 LB Zuidhorn

This is an assembler to use 6502 code within FORTH. It can be used with the 79-standard FORTH from the cassette library for the JUNIOR without drives. For other versions of 79-standard FORTH some changes in the addresses of the subroutines may be necessary.

The assembler supports the following instructions:

CLC,	SEC,	CLD,	SED,	CLI,	SEI,
CLV,	NOP,	DEX,	INX,	DEY,	INY,
TAX,	TXA,	TAY,	TYA,	TXS,	TSX,
RTS,	RTI,	PHA,	PLA,	PHP,	PLP,
LDA,	STA,	ADC,	SBC,	CMP,	EOR,
AND,	ORA,	DEC,	INC,	ASL,	LSR,
ROL,	ROT,	BIT,	STX,	STY,	CPX,
CPY,	LDX,	LDY,	JMP,	JSR,	JMP,

The (optional) addressing mode is placed BEFORE the instruction and is coded by:

(immediate)
.A (accumulator)
,X (indexed-X)
,Y (indexed-Y)
X) (indexed-indirect)
)Y (indirect-indexed)

Absolute and zero-page addressing is default.

Conditional branches are coded by the construction:

test IF, ... (optional ELSE, ...) ENDFIF,

or the construction:

BEGIN, ... test UNTIL,

For 'test' you can use:

= (equal)
< (minus)
CS (carry set)
VS (overflow set)

which tests can be followed by an optional NOT.

A code definition has the construction:

CODE name ENDCODE

For code subroutines you can use the construction:

LABEL name RTS,

Labels can be defined with:

ASSEMBLER value ALABEL name

e.g.

ASSEMBLER HEX 1334 ALABEL PRCHA

```
SCR # 96
0 ( ##### 6502 ASSEMBLER ##### GEVOP: 19/11/84 )
1 VOCABULARY ASSEMBLER ASSEMBLER DEFINITIONS
2 VARIABLE MODE FORTH DEFINITIONS DECIMAL
3 ' ASSEMBLER 6 + 32 +ORIGIN ! ( LOCK VOCABULARY ASSEMBLER )
4 : ASSEMBLER [ ASSEMBLER ] 0 MODE ! ASSEMBLER ;
5 : CODE ASSEMBLER CREATE LATEST PFA DUP CFA ! ;
6 : LABEL ASSEMBLER CREATE ;
7 : ENDCODE [COMPILE] FORTH ;
8 : ;CODE ASSEMBLER [COMPILE] ;CODE ; IMMEDIATE
9
10 ASSEMBLER DEFINITIONS HEX
11
12 --)
13
14
15
```

```
SCR # 97
0 ( ##### 6502 ASSEMBLER ##### GEVOP: 29/10/84 )
1 : MODES <BUILDS C, DOES> C0 MODE ! ;
2 1 MODES # 2 MODES .A 3 MODES ,X
3 4 MODES ,Y 5 MODES X) 6 MODES )Y
4
5 : PAGE_ZERO ( TEST ON ADDRESS IN PAGE ZERO )
6 OVER 000F SWAP UC IF 10 MODE +! THEN ;
7
8 : IMM ( TEST ON IMMEDIATE ADDRESSING )
9 MODE @ 1 = IF 9 + C, C, 0 ELSE 1 THEN ;
10 : ACCU ( TEST ON ACCU ADDRESSING )
11 MODE @ 2 = IF 9 + C, 0 ELSE 1 THEN ;
12 : Z,X ( TEST ON ZERO,X ADDRESSING )
13 MODE @ 3 = IF 15 + C, C, 0 ELSE 1 THEN ;
14 : Z,Y ( TEST ON ZERO,Y ADDRESSING )
15 MODE @ 4 = IF 15 + C, C, 0 ELSE 1 THEN ; --)

```

```
SCR # 98
0 ( ##### 6502 ASSEMBLER ##### GEVOP: 03/01/85 )
1 : (IND,X) ( TEST ON X-INDIRECT ADDRESSING )
2 MODE @ 5 = IF 1 + C, C, 0 ELSE 1 THEN ;
3 : (IND,Y) ( TEST ON INDIRECT-Y ADDRESSING )
4 MODE @ 6 = IF 11 + C, C, 0 ELSE 1 THEN ;
5 : Z&ABS ( TEST ON ZERO AND ABSOLUTE ADDRESSING )
6 MODE @ -DUP 0 = IF 5 + C, C, 0
ELSE 10 = IF D + C, 0 ELSE 1 THEN THEN ;
7
8 : A,X ( TEST ON ABSOLUTE-X ADDRESSING )
9 MODE @ DUP 13 = SWAP 3 = OR IF 10 + C, , 0 ELSE 1 THEN ;
10 : A,Y ( TEST ON ABSOLUTE-Y ADDRESSING )
11 MODE @ DUP 14 = SWAP 4 = OR IF 19 + C, , 0 ELSE 1 THEN ;
12 --)
13
14
15
```

```
SCR # 99
0 ( ##### 6502 ASSEMBLER ##### GEVOP: 29/10/84 )
1 : SINGLE <BUILDS C, DOES> MODE DUP @ 0 ROT ! 3 ?ERROR
2 ( TEST ON CORRECT MODE AND RESET MODE )
3 C0 C, ;
4
5 10 SINGLE CLC, 30 SINGLE SEC, D0 SINGLE CLD, F0 SINGLE SED,
6 50 SINGLE CLI, 70 SINGLE SEI, B0 SINGLE CLV, EA SINGLE NOP,
7 CA SINGLE DEX, E0 SINGLE INX, 80 SINGLE DEY, C0 SINGLE INY,
8 AA SINGLE TAX, BA SINGLE TXA, A0 SINGLE TAY, 90 SINGLE TYA,
9 9A SINGLE TXS, BA SINGLE TSX,
10 40 SINGLE PHA, 60 SINGLE PLA, 00 SINGLE PHP, 20 SINGLE PLP,
11 60 SINGLE RTS, 40 SINGLE RTI,
12 --)
13
14
15
```

Labels and macro's:

Within the assembler there are some predefined labels and macro's. The following description is taken from the fig-FORTH MODEL from the fig FORTH INSTALLATION MANUAL.

- IP address of the interpretive pointer in zero page.
- W address of the code field pointer in zero page.
- N address of an 8 byte scratch area in zero page.
- XSAVE address of a temporary register for X in zero page.
- UP address of the user pointer in zero page.
- BOT address of low byte of a 16-bit stack item with ,X address mode. X register locates computation stack in zero page, relative to address \$0000.
- BOT 1+ address of the high byte of the bottom stack item, ,X mode present.
- SEC address of the second stack item low byte as for BOT.
- SEC 1+ address of the high byte of the second stack item as for BOT 1+.
- R address of the low byte of the return stack with ,X mode present.
- R 1+ address of the high byte of the return stack with ,X mode present.

NOTE:

Other items on the computation and return stack can be accessed with:

- BOT n +
- TSX, R n +

For accessing items on the return stack use the instruction TSX, first and restore X to the top of the computation stack afterwards.

PREDEFINED ROUTINES:

- PUT address of routine to replace the present computation stack high byte from the accumulator, and put from the machine stack one byte which replaces the present low stack byte; continue on to NEXT.
- PUSH address of routine to repeat PUT but creating a new bottom item on the computation stack.
- PUT0A address of routine to PUT the accumulator at low stack byte with the high stack byte zero.
- PUSH0A address of routine to PUSH the accumulator at low stack byte with the high stack byte zero.
- POP address of routine to remove one 16-bit item from the computation stack.
- POPTWO address of routine to remove two 16-bit items from the computation stack.

```
SCR # 100
0 ( ##### 6502 ASSEMBLER ##### GEVOP: 29/10/84 )
1 : TYPE1 <BUILDS C, DOES> C0
2 IMM IF PAGE_ZERO Z&ABS
3 IF 2,X IF A,X IF A,Y IF <IND,X> IF <IND>,Y 3 ?ERROR
4 THEN THEN THEN THEN 0 MODE ! ;
5
6 A0 TYPE1 LDA, 80 TYPE1 /STA/ 60 TYPE1 ADC, E0 TYPE1 SBC,
7 C0 TYPE1 CMP, 40 TYPE1 EOR, 20 TYPE1 AND, 00 TYPE1 ORA,
8
9 : STA, MODE @ 1 = 3 ?ERROR /STA/ ;
10
11 : TYPE2 <BUILDS C, DOES> C0
12 ACCU IF PAGE_ZERO Z&ABS
13 IF 2,X IF A,X 3 ?ERROR
14 THEN THEN THEN 0 MODE ! ;
15 -->
```

```
SCR # 101
0 ( ##### 6502 ASSEMBLER ##### GEVOP: 29/10/84 )
1 C1 TYPE2 /DEC/ E1 TYPE2 /INC/ 01 TYPE2 ASL, 41 TYPE2 LSR,
2 21 TYPE2 ROL, 61 TYPE2 ROR,
3
4 : DEC, MODE @ 2 = 3 ?ERROR /DEC/ ;
5 : INC, MODE @ 2 = 3 ?ERROR /INC/ ;
6
7 : BIT, IF PAGE_ZERO Z&ABS 3 ?ERROR 0 MODE ! ;
8
9 : STX, 81 PAGE_ZERO Z&ABS IF 2,Y 3 ?ERROR THEN
10 0 MODE ! ;
11 : STY, 7F PAGE_ZERO Z&ABS IF 2,X 3 ?ERROR THEN
12 0 MODE ! ;
13 -->
14
15
```

```
SCR # 102
0 ( ##### 6502 ASSEMBLER ##### GEVOP: 29/10/84 )
1 : CPX, D7 IMM IF DROP DF PAGE_ZERO Z&ABS 3 ?ERROR
2 THEN 0 MODE ! ;
3 : CPY, B7 IMM IF DROP BF PAGE_ZERO Z&ABS 3 ?ERROR
4 THEN 0 MODE ! ;
5
6 : LDX, 99 IMM IF DROP A1 PAGE_ZERO Z&ABS IF 2,Y 3 ?ERROR
7 THEN THEN 0 MODE ! ;
8 : LDY, 97 IMM IF DROP 9F PAGE_ZERO Z&ABS IF 2,X 3 ?ERROR
9 THEN THEN 0 MODE ! ;
10
11 : JMP, MODE @ 3 ?ERROR 4C C, , ;
12 : JSR, MODE @ 3 ?ERROR 20 C, , ;
13 : )JMP, MODE @ 3 ?ERROR 6C C, , ;
14
15 -->
```

```
SCR # 103
0 ( ##### 6502 ASSEMBLER ##### GEVOP: 29/10/84 )
1 : IF, HERE SWAP , 2 ;
2 : ENDIF, 2 ?PAIRS DUP C0 4C = IF HERE SWAP 1+ !
3 ELSE HERE OVER - 2 - SWAP 1+ C!
4 THEN ;
5 : ELSE, >R HERE 0 JMP, SWAP R) ENDIF, 2 ;
6
7 : BEGIN, HERE 1 ;
8 : UNTIL, SWAP 1 ?PAIRS C, HERE 1+ - C, ;
9
10 : NOT 20 XOR ;
11
12 D0 CONSTANT 0= 10 CONSTANT 0<
13 90 CONSTANT CS 50 CONSTANT VS
14
15 -->
```

BINARY address of routine to remove one 16-bit item of the computation stack and put the accumulator (high) and machine stack (low) over what was second.

NEXT address of the inner-interpreter to which all code routines must return. NEXT fetches indirectly referred to IP the next compiled FORTH word address. It then jumps indirectly to pointed machine code.

All the above routines continue with NEXT and must be called with JMP, the next routine is a subroutine and must be called with JSR,

SETUP address of routine to move 16-bits items to zero-page. Item quantity is in accumulator.

```

SCR # 104
0 ( ##### 6502 ASSEMBLER #####          GEVOP: 19/11/84 )
1 : ALABEL <BUILDS , DOES> @ ;
2
3 00A6 ALABEL N          00AE ALABEL IP          00B1 ALABEL W
4 00B3 ALABEL UP        00B5 ALABEL XSAVE       203D ALABEL PUSH
5 203F ALABEL PUT       2044 ALABEL NEXT       2104 ALABEL SETUP
6 21EC ALABEL POPTWO   21EE ALABEL POP          23A3 ALABEL BINARY
7 23DC ALABEL PUSH0A
8
9 : BOT  0 ,X ;          : SEC  2 ,X ;
10 : R   101 ,X ;      ( STACK POINTER ON 1ST EMPTY POS.)
11
12 -->
13
14
15
    
```

```

SCR # 105
0 ( ##### 6502 ASSEMBLER #####          GEVOP: 29/10/84 )
1 LABEL PUT0A BOT 1+ STA, 0 # LDA, BOT STA, NEXT JMP,
2
3 FORTH DEFINITIONS DECIMAL
4
5 CR ." ASSEMBLER LOADED "
6
7
8
9
10
11 LATEST  12 +ORIGIN !
12 HERE   28 +ORIGIN !
13 HERE   30 +ORIGIN !
14
15 HERE FENCE ! ;S
    
```

MYSTERIEUZE HANG-UPS

Iedereen die regelmatig achter een computer zit, heeft het meer dan eens meesemaakt: plotselings "hangt" het systeem. Om onverklaarbare redenen zit de hele boel muurvast. Op het toetsenbord wordt niet meer gereageerd en ook bonken op de beeldbuis helpt niet. De computer is "down" gesaan, zoals dit in vakjargon heet. Het is al een heel oud zeer, dat zelfs bijna dagelijks bij grote computers voorkomt. Computer hang-ups hebben altijd iets raadselachtigs gehad. De oorzaak kan zowel in de hardware als in de software zitten of in een combinatie van beide. Niet zo bekend is dat de oorzaak ook in ons openbare elektriciteitsnet kan liggen. De spanningen kunnen nogal wisselen met regelmatige "dips". Wanneer wordt oversgeschakeld van het ene net op het andere, doen moderne thyristor schakelingen dat heel snel en voor de mens onmerkbaar.

Maar niet voor de computer. Als sedurende een onderdeel van een seconde de spanning sterk varieert, reageren de chips daar onmiddellijk op en in sommige gevallen vervalt de computer in "idle loop": een soort coma toestand, waaruit geen ontsnappings mogelijk is, tenzij de gebruiker hem met de reset- of aan/uit knop uit zijn lijden verlost.



COMMODORE BASIC V2 (as present in the Commodore VIC-20).

Tokenised Microsoft BASIC keywords and addresses N. de Vries
 Analogous to the publication for Commodore-64 by A. Mueller.
 DE 6502 KENNER, December 1983, pages 5-8.

```

-----
:      COMMANDS
:
:      KEYWORDS CORRESPONDING TO C09E
:      ADDRESSES CORRESPONDING TO C00C
:      THE ADDRESSES OF ROUTINES FOR COMMANDS ARE THE ADDRESSES
:      MINUS 1, BECAUSE THE ROUTINES ARE INVOKED THROUGH RTS.
:
:
-----
    
```

KEYWORD	TOKEN	ADDR-1			
END	80	C830	WAIT	92	D82C
FOR	81	C741	LOAD	93	E164
NEXT	82	CD1D	SAVE	94	E152
DATA	83	CBF7	VERIFY	95	E161
INPUT#	84	C8A4	DEF	96	D3B2
INPUT	85	C8BE	POKE	97	D823
DIM	86	D080	PRINT#	98	CA7F
READ	87	CC05	PRINT	99	CA9F
LET	88	C9A4	CONT	9A	C856
GOTO	89	C89F	LIST	9B	C69B
RUN	8A	C870	CLR	9C	C65D
IF	8B	C927	CMD	9D	CA85
RESTORE	8C	CB1C	SYS	9E	E126
GOSUB	8D	C882	OPEN	9F	E1BA
RETURN	8E	C8D1	CLOSE	A0	E1C3
REM	8F	C93A	GET	A1	CB7A
STOP	90	C82E	NEW	A2	C641
ON	91	C94A	GO	CB	(C812)

```

-----
:
:      MISCELLANEOUS KEYWORDS
:
:      KEYWORDS CORRESPONDING TO C129
:
:
-----
    
```

KEYWORD	TOKEN
TAB (A3
TO	A4
FN	A5
SFC (A6
THEN	A7
NOT	A8
STEP	A9

COMMODORE BASIC V2 (as present in the Commodore VIC-20).

Tokenised Microsoft BASIC keywords and addresses

```

:
:   DYADIC OPERATORS
:
:   KEYWORDS CORRESPONDING TO C140
:   PRIORITIES AND ADDRESSES CORRESPONDING TO C0B0
:
:   THE ADDRESSES OF ROUTINES FOR DYADIC OPERATORS ARE THE
:   ADDRESS MINUS 1, BECAUSE THE ROUTINES ARE INVOKED THROUGH
:   A RTS INSTRUCTION.
:

```

KEYWORD	TOKEN	ADDR-1	PRTY	
+	AA	D869	79	addition
-	AB	D852	79	subtraction
*	AC	DA2A	7B	multiplication
/	AD	DB11	7B	division
↑	AE	DF7A	7F	exponentiation
AND	AF	CFE8	50	logical AND
OR	BA	CFE5	46	logical OR
monadic "-"	AB	DFB3	7D	negation
monadic NOT	AB	CE03	5A	logical NOT
>	B1	D015	64	comparison
=	B2	D015	64	comparison
<	B3	D015	64	comparison

```

:
:   FUNCTIONS
:
:   FUNCTIONS CORRESPONDING TO C14D
:   ADDRESSES CORRESPONDING TO C052
:

```

KEYWORD	TOKEN	ADDR	KEYWORD	TOKEN	ADDR
SGN	B4	DC39	TAN	C0	E2B1
INT	B5	DCCC	ATN	C1	E30B
ABS	B6	DC58	PEEK	C2	D80D
USR	B7	0000	LEN	C3	D77C
FRE	B8	D37D	STR\$	C4	D465
POS	B9	D39E	VAL	C5	D7AD
SQR	BA	DF71	ASC	C6	D78B
RND	BB	E094	CHR\$	C7	D6EC
LOG	BC	D9EA	LEFT\$	C8	D700
EXP	BD	DFED	RIGHT\$	C9	D72C
COS	BE	E261	MID\$	CA	D737
SIN	BF	E268			

- Note: 1) Bit 7 in the last character of each keyword is set to determine the end of a keyword.
 2) Keywords are tokenised by adding \$80 to their relative (hex) position in the table.

YASS L

```

0010      : ID ON KB-9 MICROSOFT BASIC PROGRAMS
0020      : =====
0030      :
0040      : BIJ HET SCHRIJVEN VAN BASIC PROGRAM-
0050      : MA'S OP CASSETTE EN HET LATER TERUG-
0060      : LEZEN MISTE IK DE MOGELIJKHEID DE
0070      : PROGRAMMA'S GEEN IDENTIFICATIENUM-
0080      : MER MEE TE KUNNEN GEVEN. ZOALS DAT
0090      : WEL KAN IN HET MONITORPROGRAMMA PM
0100      : VAN DE JUNIOR-COMPUTER.
0110      : IK HEB DIT EENVOUDIG OP KUNNEN LOS-
0120      : SEN. MEN KAN WELISWAAR SLECHTS 1 CY-
0130      : FER OF LETTER MEEGEVEN, MAAR DIT IS
0140      : NIET ZO EEN BEZWAAR ALS MEN +/- 64
0150      : MOGELIJKHEDEN HEEFT.
0160      :
0170      : DOOR : GAMIEL DE LY
0180      :       E. BLAESSTRAAT 7
0190      :       B-9218 GENT.
0200      :
0210      .BA $1A00
0220      :
0230 RDTAPE .DE $0B02      READ THE TAPE
0240 DUMP   .DE $09DF      WRITE ON TAPE
0250 RECCHA .DE $12AE      RECEIVE CHARACTER FROM KEYBOARD
0260 RESTTY .DE $14BC      RESET THE TTY
0270 ID    .DE $1A79      IDENTIFICATION OF THE DUMPT PROGRAM
0280 WSTRT  .DE $2348      WARM START BASIC
0290      :
1A00- 20 AE 12 0300      JSR RECCHA      WAIT FOR A KEY
1A03- 8D 79 1A 0310      STA ID        DEPRESSED KEY BECOMES IDENTIFICATION
1A06- 20 DF 09 0320      JSR DUMP      WRITE PROGRAM ON TAPE
1A09- 20 BC 14 0330      JSR RESTTY    RESET THE TTY
1A0C- 4C 48 23 0340      JMP WSTRT     RETURN TO BASIC
1A0F- 20 AE 12 0350      JSR RECCHA    WAIT FOR A KEY
1A12- 8D 79 1A 0360      STA ID
1A15- 20 02 0B 0370      JSR RDTAPE    READ PROGRAM ON TAPE WHIT THIS ID
1A18- 20 BC 14 0380      JSR RESTTY
1A1B- 4C A9 27 0390      JMP $27A9
0400      :
0410      : AANPASSEN IN KB-9 BASIC :
0420      : $27A3 - JMP $1A0F
0430      :
0440      : (Red.: Diegenen die. zoals Gamiel de Lv. snel met
0450      :       z'n oas oekochte KB-9 Basic programma's
0460      :       wil kunnen bewaren. zal hiermede snel oeh-
0470      :       holoen ziin. Wie de Basic noo meer oebru-
0480      :       kersvriendelijk wil maken doet er oood aan
0490      :       de Collected Patches on KB-9 aan te schaf-
0500      :       fen.)
0510      .en

```

VIA-ACIA-PRINT

PIN-AANSLUITINGEN VIA+ACIA PRINT

25 POLIGE MALE CONNECTOR VIA AANSLUITINGEN

PA0	1	GROEN
PA1	2	WIT
PA2	3	PAARS
PA3	4	DRANJE
PA4	5	ROOD/WIT
PA5	6	GEEL
PA6	7	GROEN/WIT
PA7	8	BRUIN
PB0	9	GRIJS
PB1	10	ZWART
PB2	11	GROEN
PB3	12	WIT
PB4	13	PAARS
PB5	14	DRANJE
PB6	15	ROOD/WIT
PB7	16	GEEL
CA1	17	ZWART
CA2	18	GRIJS
CB1	19	GROEN/WIT
CB2	20	BRUIN
+5V	21	ROOD
+5V	22	
GND	23	BLAUW
GND	24	
GND	25	

25 POLIGE FEMALE CONNECTOR ACIA

TXD	2	DRANJE	> UIT INTERFACE	TRANSFER DATA
RTS	4	PAARS	> UIT INTERFACE	REQUEST TO SEND
RXD	3	WIT	< IN INTERFACE	RECEIVE DATA
DCD	8	GROEN	< IN INTERFACE	DATA CARRIER DETECT
CTS	5	ZWART	< IN INTERFACE	CLEAR TO SEND
GND	7	BLAUW		
GND	1	SIGNAL GROUND		
+12V	20	ROOD		

BESTE WILLEM,

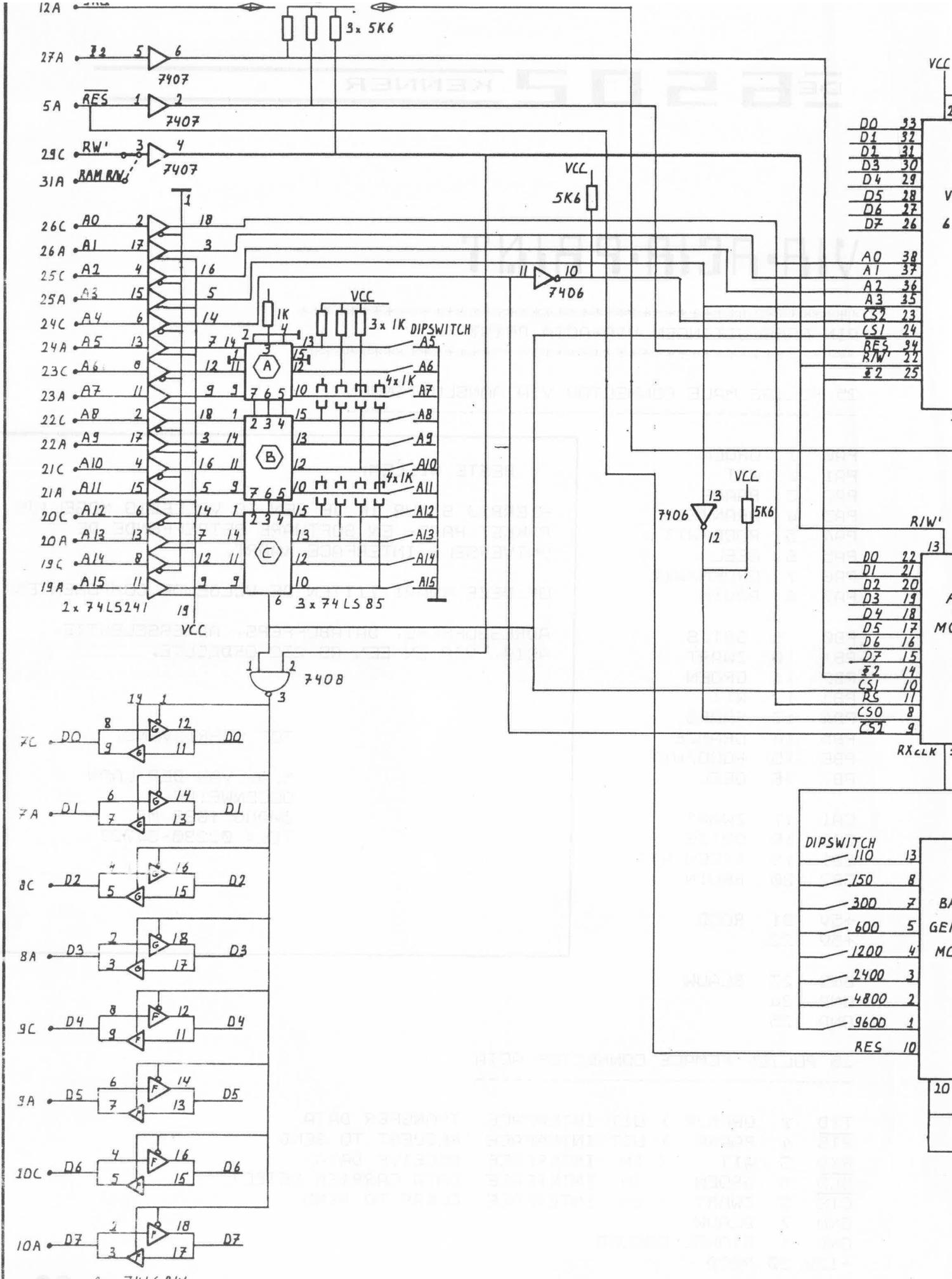
HIERBIJ STUUR IK JE EEN ZO VOLLEDIG MOGELIJK
PAKKET HARD- EN SOFTWARE BETREFFENDE DE
UNIVERSELE INTERFACE KAART.

OP DEZE KAART ZITTEN DE VOLGENDE COMPONENTEN:

ADRESBUFFERS, DATABUFFERS, ADRESSELEKTIE,
ACIA, VIA EN EEN RS 232 GEDEELTE.

TOT SCHRIJVENS,

M. A. VAN DER LAAN
OSSENWEIDE 2
ZWAAG 1589 MR
TEL: 02290-34723

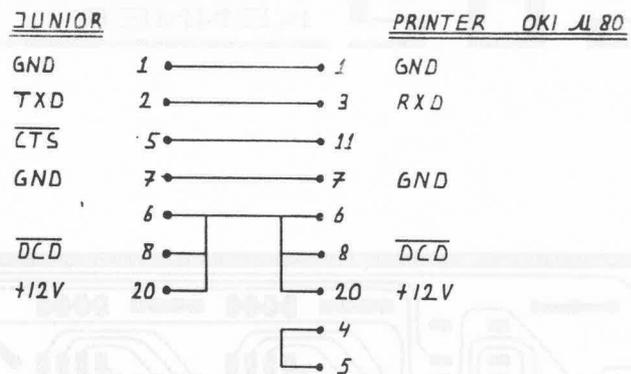
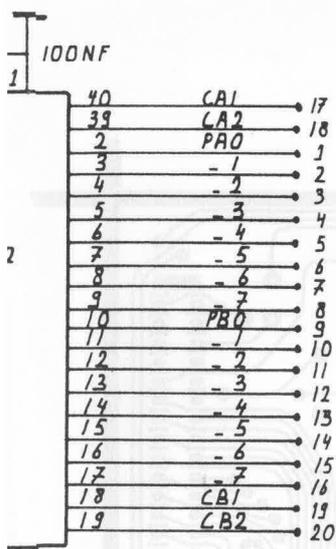


DO	33
D1	32
D2	31
D3	30
D4	29
D5	28
D6	27
D7	26

A0	38
A1	37
A2	36
A3	35
CS2	23
CS1	24
RES	34
RW'	22
#2	25

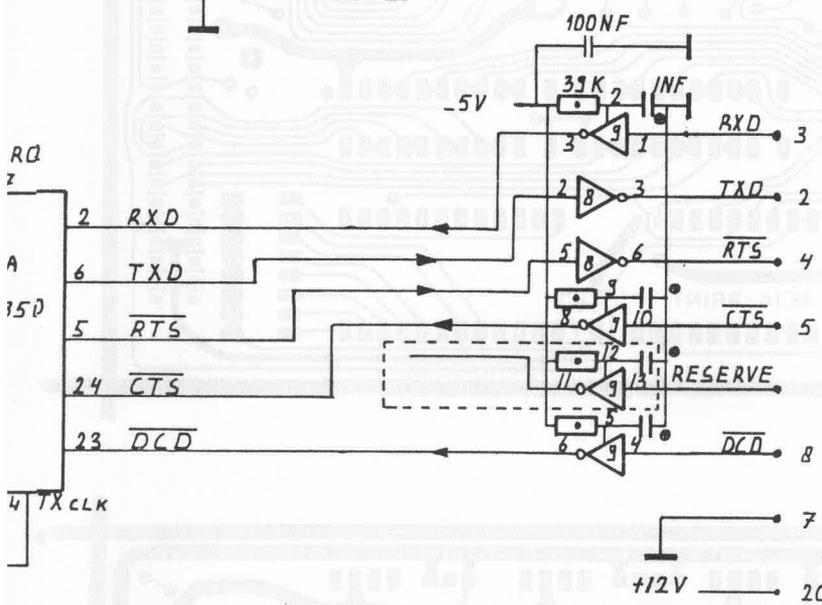
R/W'	13
DO	22
D1	21
D2	20
D3	19
D4	18
D5	17
D6	16
D7	15
#2	14
CS1	10
RS	11
CS0	8
CS2	9
RXCLK	2

DIPSWITCH	
110	13
150	8
300	7
600	5
1200	4
2400	3
4800	2
9600	1
RES	10

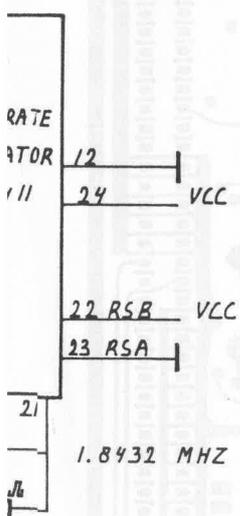


25 POLIGE MALE CONN.

ADRES-SELEKTIE
 X000 - CONTROL REG. ACIA
 X001 - DATA REG. ACIA
 X010 - X01F - 16x VIA ADRESSEN

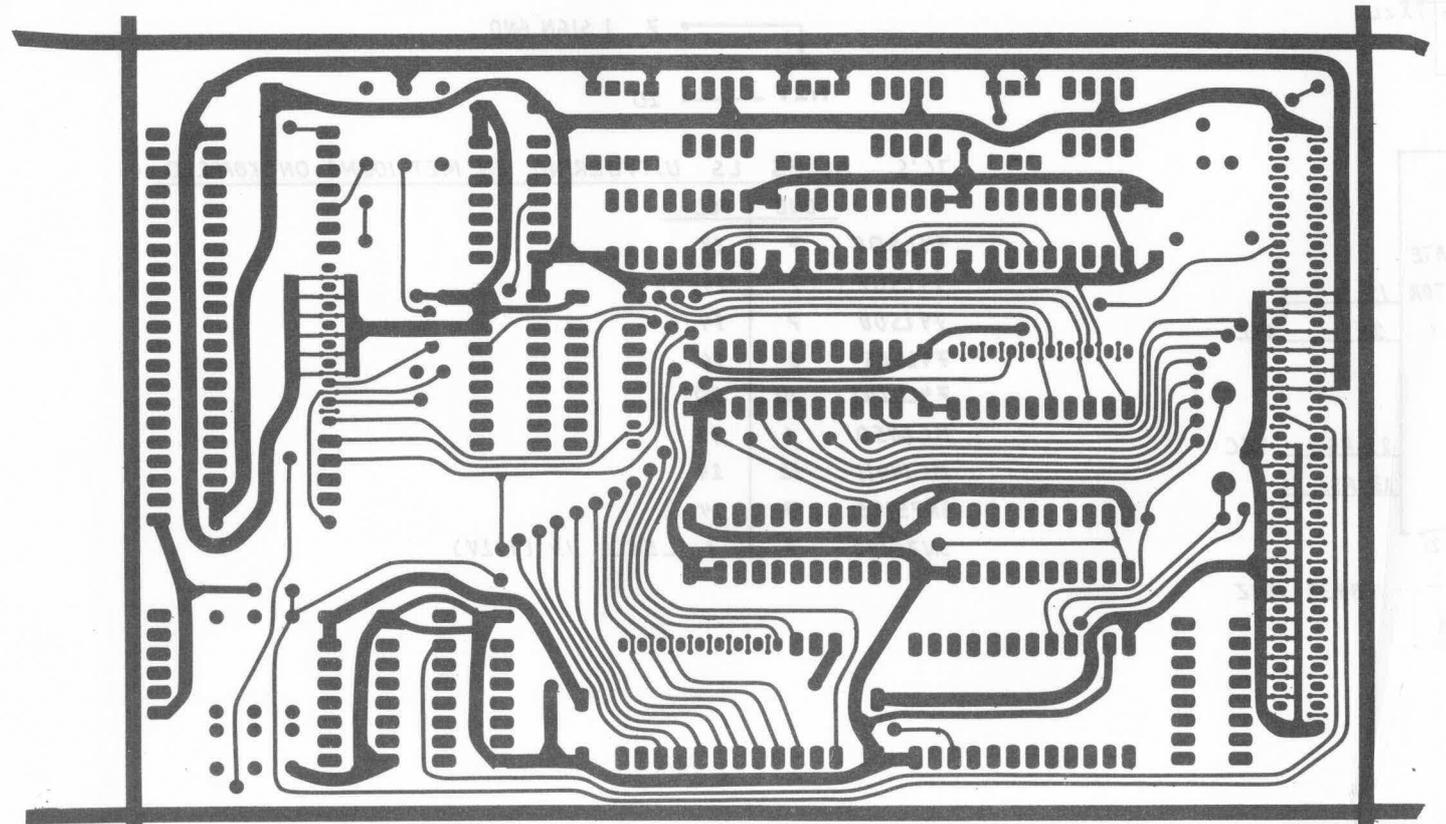
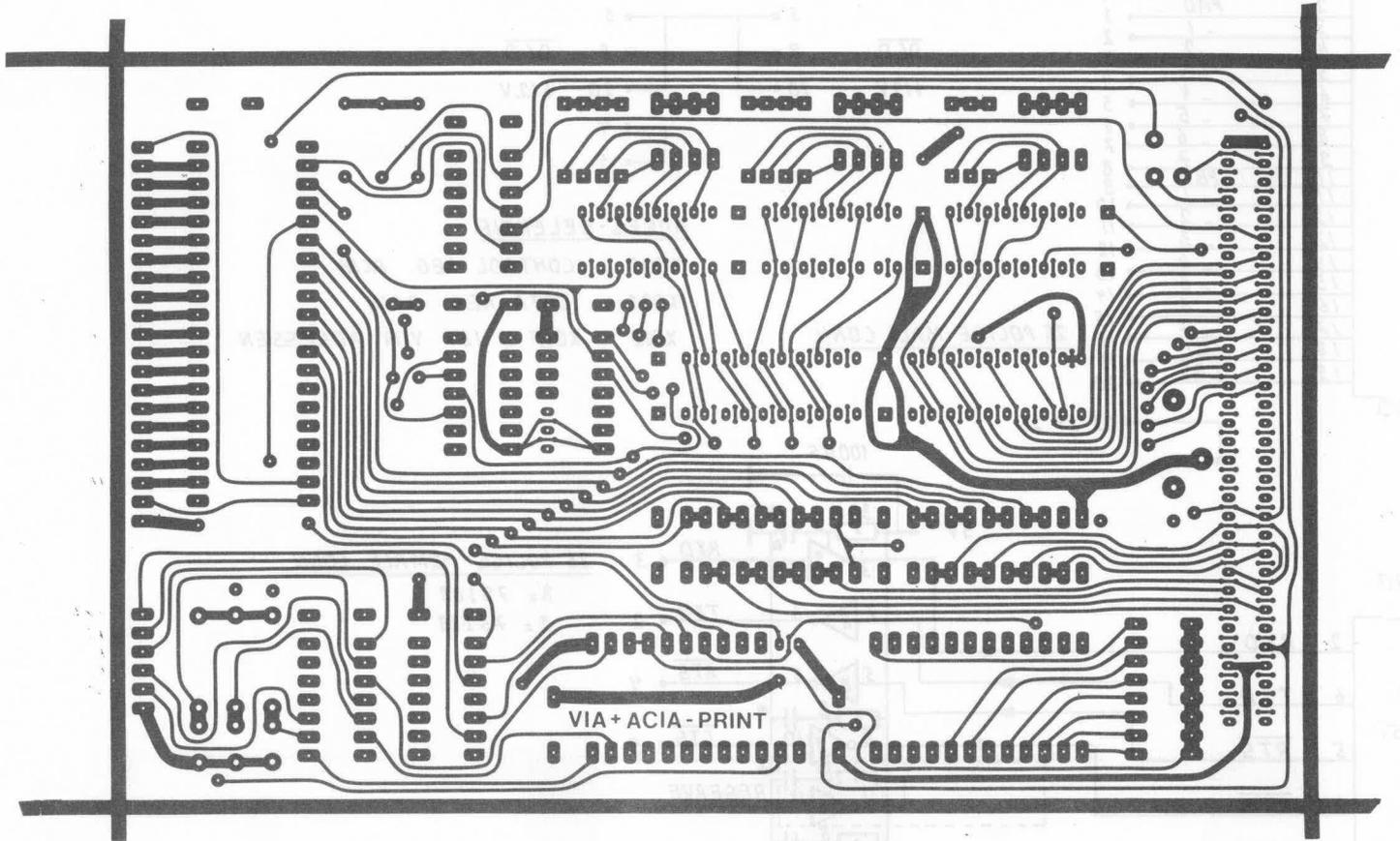


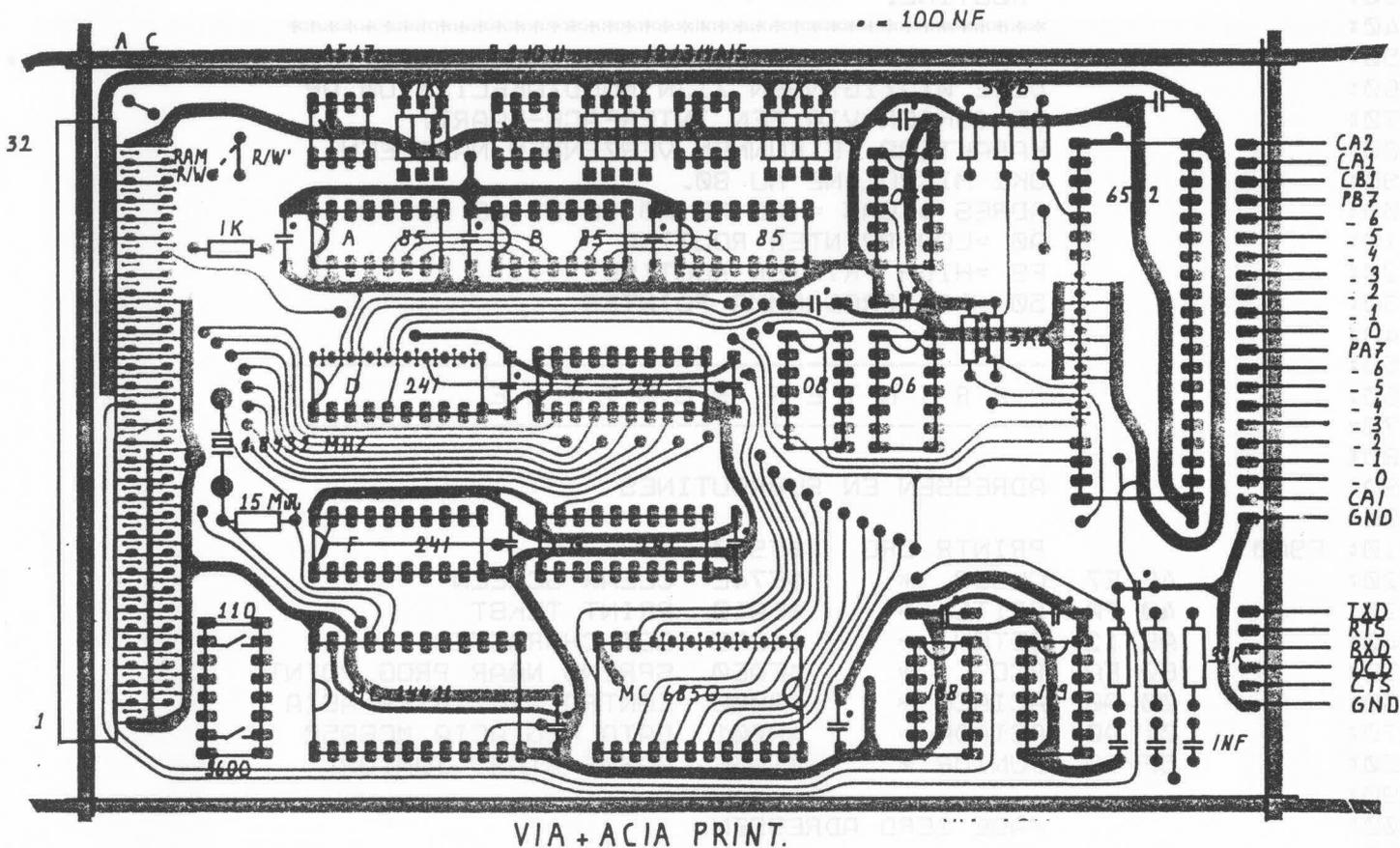
25 POLIGE FEMALE CONN.
 8 = 75188
 9 = 75189



JL'S ALLES LS UITVOERING EN MET 100NF ONTKOPPELD

	GND	VCC
74LS06	7	14
74LS07	7	14
74LS08	7	14
74LS85	8	16
74LS241	10	20
MC6850	1	12
MC14411	12	14
SN75189	7	14
SN75188	7	1 (-5V) 14 (+12V)





COMPONENTEN-LIJST VIA+ACIA PRINT.

- 3* 4 POLIGE DIPSWITCH
- 1* 8 POLIGE DIPSWITCH
- 3* 74LS85
- 4* 74LS241
- 1* 74LS06
- 1* 74LS07
- 1* 74LS08
- 1* SN 75188
- 1* SN 75189
- 1* MC 6850
- 1* MC 14411
- 1* 6522
- 1* KRISTAL 1.8432 MHZ
- 12* 1K KOOL
- 1* 15M KOOL
- 3* 39K KOOL
- 5* 5K6 KOOL
- 3* 1 NF
- 16* 100 NF
- 1* 64 POLIGE DIN COONECTOR
- 1* 25 POLIGE FEMALE CONNECTOR
- 1* 25 POLIGE MALE CONNECTOR

```

0010: *****
0020: WIJZIGINGEN IN PM TEN BEHOEVE VAN PRINTER-
0030: ROUTINE.
0040: *****

```

```

0050:
0060: DEZE WIJZIGINGEN ZIJN NOODZAKELIJK OM OP
0070: DE JUNIOR VIA EEN INTERFACE-KAART,
0080: KARAKTERS TE KUNNEN VERZENDEN NAAR EEN
0090: OKI MICROLINE MU 80.
0100: ADRES $10A3 = A5 F8 A0 WORDT 4C A0 F9
0110: A0 =LOW PRINTER ROUTINE
0120: F9 =HIGH PRINTER ROUTINE
0130: 50 ='P' PROGRAMMA POINTER
0140:

```

P R I N T E R R O U T I N E

```

0150:
0160:
0170:
0180:
0190: ADRESSEN EN SUBROUTINES
0200:

```

```

0210: F9A0          PRINTR ORG      $F9A0
0220:          4E E7 CLSCR *        $E74E CLEAR SCREEN
0230:          40 FA PRITEK *       $FA40 PRINT TEKST
0240:          AE 12 GETCH *       $12AE GET CHARACTER
0250:          50 FA PCC *         $FA50 SPRING NAAR PROG POINT
0260:          00 A0 ACIACD *      $A000 CONTROL REGISTER ACIA
0270:          01 A0 ACIADA *      $A001 DATA REG.ACIA MC6850
0280:          5F 10 JUNIOR *      $105F WARMTE START JUNIOR
0290:

```

```

0300: PAGE ZERO ADRESSEN
0310:

```

```

0320:          31 00 SALTEK *       $0031 START ADRES LOW TEKST
0330:          32 00 SAHTEK *       $0032 START ADR. HIGH TEKST
0340:          FF 00 DATBUF *       $00FF DATA BUFFER
0350:
0360:

```

```

0370: F9A0 20 4E E7 START JSR CLSCR CLEAR SCREEN
0380: F9A3 A9 00          LDAIM $00 INIT PRINT TEKST
0390: F9A5 85 31          STAZ SALTEK
0400: F9A7 A9 FA          LDAIM $FA
0410: F9A9 85 32          STAZ SAHTEK
0420: F9AB 20 40 FA JSR PRITEK PRINT TEKST UIT
0430: F9AE 20 AE 12 JSR GETCH PAK KARAKTER
0440: F9B1 C9 31 CMPIM $31 TOETS 1 BINNEN?
0450: F9B3 D0 03 BNE PRIAAN PRINTER AAN
0460: F9B5 4C 50 FA JMP PCC
0470: F9B8 C9 32 PRIAAN CMPIM $32 TOETS 2 BINNEN?
0480: F9BA D0 11 BNE PRIUIT PRINTER UIT
0490: F9BC A9 03 LDAIM $03 RESET ACIACD MC6850
0500: F9BE 8D 00 A0 STA ACIACD
0510: F9C1 A9 51 LDAIM $51 INIT PRINTER AAN
0520: F9C3 8D 00 A0 STA ACIACD
0530: F9C6 A9 01 LDAIM $01 INIT DATA BUFFER
0540: F9C8 85 FF STAZ DATBUF
0550: F9CA 4C 5F 10 JMP JUNIOR
0560: F9CD C9 33 PRIUIT CMPIM $33 TOETS 3 BINNEN?
0570: F9CF D0 CF BNE START
0580:
0590: HIER KOMEN WE VANUIT DE RESETROUTINE
ADRES $FF00 TERECHT.
0600: F9D1 A9 03 LDAIM $03 RESET ACIACD

```

```

0610: F9D3 8D 00 A0          STA  ACIACD
0620: F9D6 85 FF          STAZ DATBUF PRINTER START NU UIT
0630: F9D8 4C 5F 10        JMP  JUNIOR
0640:
0650:          TEKST BEGINT OP ADRES $FA00 TOT $FA3B
0660:
0670:          KEUZE: 1=PROGRAMMA POINTER 2=PRINTER AAN
0680:          3=PRINTER UIT
0690:

```

```

0700:          INHOUD DATA BUFFER =01 ) PRINTER IS AAN
0710:          INHOUD DATA BUFFER =03 ) PRINTER IS UIT
0720:
0730:

```

```

0010:          *****
0020:          WIJZIGINGEN IN PM TEN BEHOEVE VAN
0030:          SUBROUTINE PRINTER EN SUBROUTINE PRINTTEKST
0040:          *****
0050:

```

```

0060:          VERANDERING IN OUTCH
0070:          ADRES 1337 8D 62 1A WORDT 20 E0 F9 JSR
0080:          PRINTER SUBROUTINE
0090:          VERANDERING IN GETCH
0100:          ADRES 12DC AE 61 1A WORDT 20 DD F9 JSR
0110:          PRINTER SUBROUTINE
0120:

```

0130: GEBRUIKTE ADRESSEN EN SUBROUTINES

0140:					
0150:	61 1A	SAVEX *	\$1A61	SAVE X-REG.	
0160:	62 1A	CHAR *	\$1A62	DATA BUFFER CHARACTER	
0170:	00 A0	ACIACD *	\$A000	CONTR REG ACIA	
0180:	01 A0	ACIADA *	\$A001	DATA REG ACIA	
0190:	34 13	OUTCH *	\$1334	WEGSCHR. KARAKTER	

0200: PAGE ZERO ADRESSEN

0210:					
0220:					
0230:	30 00	AD *	\$0030	LDA MET KARAKTER	
0240:	31 00	SALTEK *	\$0031	START ADR LOW TEKST	
0250:	32 00	SAHTEK *	\$0032	START ADR HIGH TEKST	
0260:	33 00	E0 *	\$0033	RTS TERUG UI SUBR.	
0270:	FF 00	DATBUF *	\$00FF	DATA BUFFER PRINTER	

0300: -----
0310: S U B R O U T I N E P R I N T E R
0320: -----

```

0330:
0340:
0350:          VANUIT DE SUBR. GETCH KOMT MEN OP ADRES
0360:          $F9DD TERCHT EN VANUIT SUBR. OUTCH KOMT
0370:          MEN OP ADRES $F9E0 TERECHT.
0380:

```

```

0390: F9DD          SUBPRI  ORG  $F9DD
0400: F9DD AE 61 1A  SUBROU  LDX  SAVEX  X-REG TERUG
0410: F9E0 8D 62 1A  SUB     STA  CHAR  SAVE ACCU
0420: F9E3 A5 FF          LDAZ  DATBUF
0430: F9E5 C9 03          CMPIM $03  PRINTER UIT ?

```

```

0440: F9E7 F0 00          BEQ   KAR
0450: F9E9 AD 00 A0  KARAK LDA   ACIACO
0460: F9EC 29 02          ANDIM $02
0470: F9EE F0 F9          BEQ   KARAK   NOG GEEN NIEUW KARAKTER
0480: F9F0 AD 62 1A      LDA   CHAR
0490: F9F3 8D 01 A0      STA   ACIADA  SCHRIJF KARAKTER WEG
0500: F9F6 AD 62 1A  KAR  LDA   CHAR   KARAKTER TERUG IN ACCU
0510: F9F9 60          RTS
0520:
0530:
0540:
0550:

```

S U B R O U T I N E P R I N T T E K S T

```

0580: FA40          PRITEK ORG   $FA40
0590: FA40 A9 AD      START LDAIM $AD   INIT SUBROUTINE
0600: FA42 85 30      STAZ  AD
0610: FA44 A9 60      LDAIM $60
0620: FA46 85 33      STAZ  60
0630: FA48 20 30 00  NIEUW JSR  AD   KARAKTER OPHALEN
0640: FA4B C9 1B      CMPIM $1B   ESCAPE TOETS
0650: FA4D F0 08      BEQ   RTS
0655: FA4F 20 34 13  JSR  OUTCH  WEGSCHRIJVEN KARAKTER
0670: FA52 E6 31      INC   SALTEK
0680: FA54 4C 48 FA  JMP   NIEUW
0690: FA57 60          RTS   RTS
0700:
0710:
0720:
0730:

```

WIJZIGING IN SYM-BASIC TBV DE PRINTER
OUTCH OP ADRES \$0C96 WORDT
JMP OUTCH ADRES 4C 34 13

WIJZIGING IN INIT RESET OP ADRES \$FF00
ADRES \$FF4C 4C 5F 10 WORDT 4C D1 F9 JMP
PRINTER ROUTINE.

HET PROGRAMMA PCC (DIT IS HET UITPRINTEN VAN DE
PROGRAM POINTER.

```

0060: FA60          PCC   ORG   $FA60
0070:
0080:          F8 00  INL   *   $00F8
0090:          A7 10  PNT+S *   $10A7
0100:
0110: FA60 A5 F8          LDAZ  INL
0120: FA62 A0 00          LDYIM $00
0130: FA64 4C A7 10      JMP   PNT+S

```

```

0140:
0150:
0160:      VOOR HET AUTOMATISCH RESETTEN VAN DE JUNIOR
0170:      KUN JE EVENTUEEL HET PROGRAMMA "AUTOMA"
0180:      UIT DE 6502-KENNER NO. 23 NEMEN, MAAR DAN MOET
0190:      JE WEL DE REGELS 600, 610, 620 VAN DE
0200:      PRINTERROUTINE HIERAAN TOEVOEGEN
0210:      EN DEZE DRIE REGELS OOK IN DE PRINTER-
0220:      ROUTINE LATEN ZITTEN.

```

```

0010:      *****
0020:      SUBROUTINE CLEAR SCREEN.
0030:      *****
0040:
0050: E74E      CLESCR   ORG      $E74E
0060:
0070:          34 13  DUTCH  *      $1334  KARAKTER WEGSCHRIJVEN.
0080:
0090: E74E A9 0C      LDAIM  $0C      SCHOONMAAKKARAKTER JUNIOR
0100: E750 20 34 13  JSR     DUTCH
0110: E753 A2 80      LDXIM  $80
0120: E755 A0 FF      LAB08  LDYIM  $FF
0130: E757 88        LABEL  DEY
0140: E758 D0 FD      BNE    LABEL
0150: E75A CA        DEX
0160: E75B D0 F8      BNE    LAB08
0170: E75D 50        RTS

```

DISKETTES EN TAPES ZIJN VERGEETACHTIG

Veel computersebruikers bewaren de originelen van hun dierbare programma's op floppy disks of kassettebandjes in de veronderstelling dat de informatie daarop vastligt voor de eeuwigheid. In werkelijkheid kan de houdbaarheid van de soms kostbare programma's op deze magnetische informatiedragers slechts voor een beperkte tijd worden gesarandeerd. Aangenomen wordt dat een goede kwaliteit diskette de informatie slechts voor ten hoogste twee jaar kan vasthouden en op kassettebandjes iets langer. Op de sporen van de diskettes en bandjes lissen programma's en sesevens vast in de vorm van kleine semagnetiseerde plekjes. Dit magnetisme is op den lange duur onderhevig aan een vorm van slijtase. De zogenaamde Brownse bewesing van de moleculen doet de informatie langzaam maar zeker vervalsen, terwijl ook het konstante bombardement van kosmische deeltjes een bedreiging vormt voor de bits en de bytes.

Tenminste om de twee jaar moeten originele diskettes worden sekopieerd om de informatie daarop weer nieuw magnetisch leven in te blazen. Het magnetisme op banden is iets minder kwetsbaar, maar doordat ze soms stijf zijn opgerold, is het raadzaam de banden regelmatig om te spoelen. Men voorkomt daarmee dat de bandjes gaan vastplakken en in de kassette rekorder kapot worden getrokken.



Fernando Lopes
 Colégio dos Orfãos
 Lq. Baltazar Guedes
 4300 PORTO PORTUGAL

```

0010: 11E8          CRLF   ORG   $11E8
0020:
0030:          **** PRINT BACKSPACE PATCH IN PRINTER MONITOR ****
0040:
0050:          CRLF PRINT A CARRIAGE RETURN & LINE FEED
0060:          PRBS PRINT A BACK SPACE
0070:          PRSP PRINT A SPACE
0080:
0090:          *****

0150: 11E8 A9 0D          LDAIM $0D   CR
0160: 11EA 20 BA 16      JSR NPRCHA
0170:
0180: 11ED A9 0A          PRLF   LDAIM $0A   LF
0190:
0200: 11EF 2C              =   $2C   (BIT) JUMP NEXT TWO BYTES
0210: 11F0 A9 08          PRBS   LDAIM $08   BS
0220:
0230: 11F2 2C              =   $2C   (BIT)
0240: 11F3 A9 20          LDAIM $20   SP
0250:
0260: 11F5 4C BA 16      JMP   NPRCHA PRINT AND RETURN
0270:

0300:          *****
0310:
0320:          INITIALIZE PRINT FLAG
0330:          JUNIOR PRINT 'JUNIOR'
0340:          EDITOR PRINT 'EDITOR'
0350:          ASSEM PRINT 'ASSEMBLER'
0360:
0370:          *****
0380:

0410:          58 1A SERFLG *   $1A58
0420:          D6 11 MESSY  *   $11D6
0430:          BA 16 NPRCHA *   $168A
0440:
0450:          1246          ORG   $1246
0460:
0470: 1246 A9 00          JUNIOR LDAIM $00
0480: 1248 BD 58 1A      STA   SERFLG SET PRINT FLAG TO VIDEO DISPLAY ONLY
0490:
0500: 124B A0 00          JUN   LDYIM $00
0510:
0520: 124D 2C              =   $2C   (BIT) JUMP NEXT TWO BYTES
0530: 124E A0 07          EDITOR LDYIM $07
0540:
0550: 1250 2C              =   $2C
0560: 1251 A0 0E          ASSEM LDYIM $0E
0570:
0580: 1253 20 D6 11      JSR   MESSY
0590: 1256 4C E8 11      JMP   CRLF AND RETURN
    
```



```

0010: ***** RECCHA *****
0020:
0030: RECCHA ) RECEIVES 1 ASCII CHARACTER FROM ELEKTERMINAL
0040: ) SAVES X REGISTER: DOESN'T USE Y
0050: ) RETURNS WITH ASCII CHARACTER IN ACCUMULATOR
0060: * MODIFIED:
0070: INCLUDES 'GEN. INP. WITH ECHO TO OUT. DEVICES'
0080: SETS N AND V FLAGS ACCORDING TO ACC (NOT TO X)
0090: AND DOESN'T MAKE BIT 7 ZERO: RECEIVE 256 CHAR!
0100:
0110: 03 13 DELHBT * $1303
0120: 12 13 DELBIT * $1312
0130: 8A 16 NPRCHA * $168A
0130: 41 1A DUPLEX * $1A41
0150: 61 1A TEMPB * $1A61
0160: 62 1A CHA * $1A62
0170: 80 1A PAD * $1A80
0180:
0190: 12AE RECCHA ORG $12AE
0200:
0210: 12AE 2C 80 1A RECCHA BIT PAD WAIT FOR START BIT
0220: 12B1 30 FB BMI RECCHA
0230:
0240: 12B3 8E 61 1A STX TEMPB SAVE INDEX X
0250: 12B6 A2 08 LDXIM $08 BIT COUNTER IS 8
0260: 12B8 20 03 13 JSR DELHBT DELAY HALF BIT TIME
0270:
0280: 12BB 20 12 13 RECA JSR DELBIT DELAY ONE BIT TIME
0290:
0300: 12BE 2C 80 1A RECD/B BIT PAD ONE/ZERO CHECK
0310: 12C1 10 02 BPL BIT-0-
0320: 12C3 38 BIT-1- SEC BIT IS '1'
0330: 12C4 24 = $24 (BIT) JUMP NEXT BYTE
0340: 12C5 18 BIT-0- CLC BIT IS '0'
0350: 12C6 6E 62 1A ROR CHA ROTATE CARRY INTO CHARACTER
0360: 12C9 CA DEX SET UP FOR NEXT BIT
0370: 12CA D0 EF BNE RECA ALL BITS READ?
0380:
0390: 12CC 20 12 13 RECC JSR DELBIT WAIT FOR STOP BIT TIME: V=0. C=0
0400: 12CF AD 62 1A LDA CHA GET THE CHARACTER JUST RECEIVED
0410:
0420: 12D2 48 INALL PHA SAVE IT
0430: 12D3 2C 41 1A BIT DUPLEX 00 NO ECHO / FF ECHO
0440: 12D6 50 03 BVC INALLX (V=0)
0450: 12D8 20 8A 16 JSR NPRCHA (V=0) ECHO THE CHARACTER TO OUTPUT DEVICES
0460: 12DB AE 61 1A INALLX LDX TEMPB RESTORE INDEX X
0470: 12DE 68 PLA SET N AND Z FLAGS ACCORDING TO CHARACTER RECEIVED
0480: 12DF 60 RTS DON'T MASK BIT 7: V=0 C=0
0490:
0500:
0510:
0520: 12E0 18 COMTIM CLC ...
0530:

```

DE 6502 KENNER

PRCHA THE JUNIOR PROMOTING CY (W&J) 07/04/84 PAGE 01

```

0010: ***** PRCHA *****
0020:
0030: >TRANSMIT AN ASCII CHARACTER STORED IN ACCUMULATOR
0040: TO THE ELEKTERMINAL
0050: >ADAPTED FROM THE ORIGINAL AND MODIFIED TO 8 BITS
0060: (ALSO MODIFIED IS THE BIT FORMAT OF THE UART)
0070: >SAVE INDEX X: INDEX Y NOT USED
0080:
0090:      12 13 DELBIT *      $1312 ***** AT RTS *****
0100:      CF 16 BRKTST *      $16CF *           =X *
0110:      58 1A SERFLG *      $1A58 *           =Y *
0120:      59 1A STPBIT *      $1A59 *           Acc=FF *
0130:      60 1A TEMPA *       $1A60 *           V=0 *
0140:      62 1A CHA *         $1A62 *           C=0 *
0150:      80 1A PAD *         $1A80 *           Z=? *
0160:      82 1A PBD *         $1A82 ***** N=?*****
0170:
0180:
0190: 1334 PRCHA ORG $1334
0200:
0210: 1334 8E 60 1A STX TEMPA SAVE INDEX X
0220: 1337 8D 62 1A STA CHA STORE CHARACTER IN 'SHIFT REGISTER'
0230:
0240:
0250: 133A 48 PRCHA* PHA SAVE CHARACTER...
0260: 133B 18 CLC SET FOR -0-
0270: 133C 20 61 13 JSR ONEBIT TRANSMIT START BIT (LOGICAL 0)
0280: 133F A2 08 LDXIM #08 SET UP FOR 8 DATA BITS
0290:
0300: 1341 4E 62 1A PRA LSR CHA SEND CHARACTER BIT BY BIT (SHIFT CHARACTER)
0310: 1344 20 61 13 JSR ONEBIT IF C=0, BIT-0-; IF C=1, BIT-1-;
0320:
0330: 1347 CA PRB DEX SET UP FOR NEXT BIT
0340: 1348 D0 F7 BNE PRA ALL BITS TRANSMITTED?
0350:
0360: 134A AE 59 1A LDX STPBIT AMOUNT OF STOP BITS
0370: 134D 38 SEC SET FOR -1-
0380: 134E 20 61 13 JSR ONEBIT SEND STOP BIT (LOGICAL 1); Acc=FF; CARRY=0
0390: 1351 CA DEX
0400: 1352 D0 F9 BNE PRD ALL STOP BITS?
0410:
0420: 1354 2C 80 1A BREAK? BIT PAD TEST FOR BREAK (PAD=X000.0000); OVERFLOW=0
0430: 1357 30 03 BMI END N=0 IF BREAK KEY NV-- ---- Z
0440: 1359 4C CF 16 JMP BRKTST (NPRCHA)
0450: 135C AE 60 1A END LDX TEMPA RESTORE INDEX X
0460: 135F 68 PLA RESTORE CHARACTER
0470: 1360 60 RTS C AND V FLAGS ARE ALWAYS CLEARED
0480:
0490: 1361 AD 82 1A ONEBIT LDA PBD
0500: 1364 90 03 BCC BIT-0-
0510: 1366 09 01 BIT-1- ORAIM #01 SEND A LOGICAL 1
0520: 1368 2C = $2C (BIT) JUMP NEXT TWO BYTES
0530: 1369 29 FE BIT-0- ANDIM #FE SEND A LOGICAL 0
0540: 136B 8D 82 1A STA PBD
0550: 136E 4C 12 13 JMP DELBIT DELAY 1 BIT TIME (AND RTS); Acc=FF; CARRY=0
0560:
0570: 1371 FF = $FF (22) BYTES LEFT FOR PATCH OR USER PROGRAM
0580: 1372 FF = $FF
0590: 1373 FF = $FF BUT 10 TIMES A JSR/RTS (ONEBIT) TIME MORE...
0600: 1374 FF = $FF
0610: 1375 FF = $FF
0620: 1376 FF = $FF
0630: 1377 FF = $FF
0640: 1378 FF = $FF
0650: 1379 FF = $FF
0660: 137A FF = $FF
0670: 137B FF = $FF
0680: 137C FF = $FF
0690: 137D FF = $FF
0700: 137E FF = $FF
0710: 137F FF = $FF
0720: 1380 FF = $FF
0730: 1381 FF = $FF
0740: 1382 FF = $FF
0750: 1383 FF = $FF
0760: 1384 FF = $FF
0770: 1385 FF = $FF
0780: 1386 FF = $FF
0790:
0800: 87 13 INPAR * $1387

```

```

0010: ***** ASHETT *****
0020:
0030: CONVERT AN ASCII CHARACTER INTO AN HEX DATA NIBBLE.
0040:
0050: RETURN WITH CONVERTED HEX NUMBER IN ACCUMULATOR
0060: WITH CARRY CLEAR IF VALID HEX NUMBER
0070: WITH CARRY SET IF NOT VALID HEX NUMBER
0080: WITHOUT CHANGING Y OR X REGISTERS
0090:
0100: 141E ORG $141E
0110:
0120: 141E C9 30 ASHETT CMPIM $30 IGNORE 00 → 2F
0130: 1420 30 0C BMI NOTVAT
0140: 1422 C9 3A CMPIM $3A
0150: 1424 30 0A BMI VALIT
0160: 1426 C9 41 CMPIM $41 IGNORE 3A → 40
0170: 1428 30 04 BMI NOTVAT
0180: 142A C9 47 CMPIM $47 IGNORE 47 → 7F (FF)
0190: 142C 30 02 BMI VALIT
0200:
0210: 142E 38 NOTVAT SEC
0220: 142F 60 RTS ERROR EXIT WITH CARRY SET
0230:
0240: 1430 C9 40 VALIT CMPIM $40 ASCII → HEX CONVERSION
0250: 1432 30 03 BMI VALT
0260: 1434 18 CLC
0270: 1435 69 09 ADCIM $09
0280:
0290: 1437 29 0F VALT ANDIM $0F
0300: 1439 60 RTS NORMAL EXIT WITH CARRY CLEAR
0310:
0320: 143A FF = $FF ONE BYTE LEFT
0330:
0340:
0350:
0360: *** ROUTINES WHERE ASHETT IS INCLUDED AND WHERE THE
0370: FLAGS STATUSES WILL BE DIFFERENT:
0380: *****
0390:
0400: *TAPE MONITOR
0410: RDBYT $0BFE: B0 INSTEAD OF 30 AFTER ASCHEX (ASHETT)
0420: $0C11: B0 IDEM
0430: RETURN WITH CARRY SET IF NOT VALID HEX CHARACTER
0440:
0450: *PRINTER MONITOR
0460: HEXNUM $1272: B0 INSTEAD OF 30 AFTER ASHETT
0470:
0480: *PRINTER MONITOR EDITOR
0490: SKIP $1580: B0 INSTEAD OF 30 AFTER READIN
0500: SEARCH $159B: B0 IDEM
0510: INPUT $1650: B0 INSTEAD OF 30 AFTER BYT
0520: $1655: B0 INSTEAD OF 30 AFTER READ
0530: BYTIN $16B4: B0 INSTEAD OF 30 AFTER ASHETT
0540: $16C2: B0 IDEM
0550: $16C6: 60 RTS WITH C. CLEAR INSTEAD OF LDXIM $00
0551: RETURN WITH CARRY SET IF ERROR EXIT
0560: READIN $16CC: B0 INSTEAD OF 30 AFTER BYTIN
0570: $16E1: B0 IDEM
0580: $16EF: B0 IDEM
0590: $16F3: 18 CLC
0600: $16F4: 60 RTS WITH C. CLEAR INSTEAD OF LDXIM $00
0610: RETURN WITH CARRY SET IF ERROR EXIT

```

```

0010: ***** PRINTER *****
0020:
0030: Routine to output an ASCII character to an EPSON FX-80
0040: using the Centronics interface, through 8 bit, parallel
0050: Includes messages about the printer status (power off,
0060: off line, paper end and error).
0070: The BUSY line is activated during data entry, printing,
0080: OFF LINE and ERROR (active low).
0090: The ERROR line is active during PAPER END, OFF LINE or
0100: printer ERROR.
0110:
0120: The VIA 6522 is programmed this way:
0130:
0140: IRA 0 - PAPER END active high - C FLAG (AFTER LSR)
0150: IRA ;
0160: IRA 6 - ERROR active low - V FLAG (AFTER BIT)
0170: IRA 7 - BUSY (READY) - N FLAG
0180:
0190: ORB - DATA OUT
0200: CB2 - DATA OUT STROBE handshake
0210:
0220:
0230: F4 00 YREG * $00F4 *****
0240: FA 00 COUNT0 * $00FA *
0250: FB 00 COUNT1 * $00FB * AT RTS *
0260: CF 16 BRKTST * $16CF * =Acc *
0270: C0 17 MESSAGE * $17C0 * =X *
0280: 00 18 ORB * $1800 * =Y *
0290: 01 18 IRA * $1801 * V=0 *
0300: 52 1A AHOLD2 * $1A52 * C=0 *
0310: 56 1A CENFLG * $1A56 *
0320: 80 1A PAD * $1A80 *****
0330:
0340: SUBROUTINES
0350:
0360: F8 16 DELAY1 * $16F8 DELAY AMOUNTS # IN ACCUMULATOR
0370: 5F 10 LABJUN * $105F PM "JUNIOR"
0380: E8 11 CRLF * $11E8
0390: 8A 16 NPRCHA * $168A
0400:
0410:
0420:
0430: 1559 PRINTER ORG $1559
0440: 1559 8D 52 1A STA AHOLD2 SAVE CHARACTER
0450: 155C 84 F4 STY YREG SAVE Y REGISTER
0460:
0470: 155E 2C 01 18 BUSY? BIT IRA N=IRA BIT 7; V=IRA BIT 6
0480: 1561 30 11 BMI OFFPOWR N=1 IF POWER OFF OR BUSY
0490: N=0 IF THE PRINTER IS READY
0500: 1563 AD 52 1A PRINT LDA AHOLD2 GET CHARACTER
0510: 1566 8D 00 18 STA ORB SEND IT TO PRINTER
0520:
0530: ACKNOWLEDGE NOT NECESSARY BUT POSSIBLE
0540:
0550: 1569 2C 80 1A BREAK? BIT PAD TEST FOR BREAK KEY BEFORE RETURN
0560: 156C 30 03 BMI RETURN N=0 IF BREAK KEY DEPRESSED; IF NOT, N=1
0570: 156E 4C CF 16 STOP JMP BRKTST OF NPRCHA SUBROUTINE
0580:
0590: 1571 A4 F4 RETURN LDY YREG RESTORE Y REGISTER. Acc HAS STILL THE CHAR.
0600: 1573 60 RTS RETURN WITH V=0 AND C=0 AND ACC WITH CHARACTER
0610:
0620:
0630:
0640: 1574 AD 01 18 OFFPOWR LDA IRA GET PORT
0650: 1577 C9 FF CMPIM #FF ARE ALL BITS SET? (HIGH IMPEDANCE)
0660: 1579 D0 13 BNE ERROR? IRA=FF IF THE PRINTER IS OFF POWER
0670: 157B A0 00 LDYIM #00
0680: 157D 20 C2 15 JSR MESSY PRINT "POWER OFF"
0690:

```

```

0700: 1580 AD 01 18 WAIT-1 LDA IRA WAIT FOR SWITCHING ON THE PRINTER
0710: 1583 C9 FF CMPIM $FF
0720: 1585 F0 F9 BEQ WAIT-1
0730: 1587 A9 07 LDAIM $07 RETURN AFTER DELAY #7 (PRINTER POWER / SET UP)
0740: 1589 20 F8 16 OK-1 JSR DELAY1
0750: 158C 90 D0 BCC BUSY?
0760:
0770:
0780: 158E 70 CE ERROR? BVS BUSY? V=1 IF NOT ERROR
0790: 1590 4A LSRA C= IRA BIT0
0800: 1591 90 0F BCC OFFLIN C=0 IF NOT PAPER END
0810:
0820: 1593 A0 0A PAPEND LDYIM $0A
0830: 1595 20 C2 15 JSR MESSY' PRINT "INSERT MORE PAPER"
0840:
0850: 1598 AD 01 18 WAIT-2 LDA IRA WAIT FOR MORE PAPER
0860: 159B 4A LSRA
0870: 159C B0 FA BCS WAIT-2 C=1 IF STILL PAPER END
0880:
0890: 159E A9 03 OK-2 LDAIM $03 RETURN AFTER DELAY #3 (SET UP AFTER PAPER
0900: 15A0 D0 E7 BNE OK-1 RELOAD)
0910:
0920:
0930: 15A2 A0 1C OFFLIN LDYIM $1C
0940: 15A4 20 C2 15 JSR MESSY' PRINT "PRESS ON LINE"
0950: 15A7 85 FA STA COUNT0 SET COUNT TO WAIT TIME
0960: 15A9 85 FB STA COUNT1
0970:
0980: 15AB 2C 01 18 WAIT-3 BIT IRA STILL ERROR?
0990: 15AE 70 EE BVS OK-2 V=0 MEANS ERROR V=1 MEANS NOT ERROR
1000: 15B0 88 DEY
1010: 15B1 D0 F8 BNE WAIT-3
1020: 15B3 C6 FA DEC COUNT0
1030: 15B5 D0 F4 BNE WAIT-3 ... WAIT FOR SWITCHING ON LINE
1040: 15B7 C6 FB DEC COUNT1
1050: 15B9 D0 F0 BNE WAIT-3
1060:
1070: IF AFTER ALL THAT TIME
1080: THERE IS NO ANSWER...
1090: 15BB A0 2A PRTERR LDYIM $2A
1100: 15BD 20 C2 15 JSR MESSY' PRINT "PRINTER ERROR"
1110: 15C0 50 AC BVC STOP (GO TO BREAK POINT)
1120:
1130:
1140: 15C2 AD 56 1A MESSY' LDA CENFLG SAVE CENTRONICS FLAG STATUS
1150: 15C5 48 PHA
1160: 15C6 A9 00 LDAIM $00
1170: 15C8 8D 56 1A STA CENFLG PRINT MESSAGE ONLY ON VIDED!
1180:
1190: 15CB 20 E8 11 JSR CRLF PRINT A CARRIAGE RETURN AND A LINE FEED
1200: 15CE B9 C0 17 PRTMES LDAAY MESSAGE
1210: 15D1 C9 03 CMPIM $03 ETX
1220: 15D3 F0 06 BEQ PRTEAD
1230: 15D5 20 8A 16 JSR NPRCHA
1240: 15D8 C8 INY
1250: 15D9 D0 F3 BNE PRTMES
1260:
1270: 15DB 68 PRTEAD PLA
1280: 15DC 8D 56 1A STA CENFLG RESTORE CENTRONICS FLAG STATUS
1290: 15DF 60 RTS
1300:
1310:
1320:
1330: MESSAGE $17C0
1340:
1350: 504F574552204F464603494E53455254 - POWER OFF.INSERT
1360: 204D4F52452050415045520350524553 - MORE PAPER.PRES
1370: 53204F4E204C494E45035052494E5445 - S ON LINE.PRINTE
1380: 52204552524F5203 - R ERROR.

```

NPRCHA THE JUNIOR PROMOTING CY (W&J) 08/04/84 PAGE 01

```

0010: ***** NEW PRCHA ROUTINE *****
0020:
0030: FC 00 RAMPTR * $00FC
0040: 80 0C VIDEO * $0C80
0050: 3A 13 PRCHA* * $133A
0060: 59 15 CENOUT * $1559
0070: 29 1A PZPTR * $1A29
0080: 2B 1A YHOLD * $1A2B
0090: 2C 1A XHOLD * $1A2C
0100: 2D 1A AHOLD * $1A2D
0110: 56 1A CENFLG * $1A56
0120: 58 1A PRIFLG * $1A58
0130: 60 1A TEMPA * $1A60
0140: 62 1A CHA * $1A62
0150: 7C 1A BRKT * $1A7C
0160: 80 1A PAD * $1A80
0161:
0162:
0170:
0180: 168A NPRCHA ORG $168A
0190: 168A 8E 60 1A STX TEMPA SAVE X REGISTER (FOR PRCHA IF CALLED)
0200: 168D 8D 62 1A STA CHA SAVE CHARACTER (FOR PRCHA)
0210:
0220: 1690 8D 2D 1A OUTALL STA AHOLD SAVE REGISTERS
0230: 1693 8E 2C 1A STX XHOLD
0240: 1696 8C 2B 1A STY YHOLD
0250:
0260: 1699 A4 FC LDY RAMPTR SWAP SCREEN POINTER (OS DOS)
0270: 169B A6 FD LDX RAMPTR +01
0280: 169D 8C 29 1A STY PZPTR
0290: 16A0 8E 2A 1A STX PZPTR +01
0300:
0310: 16A3 20 80 0C JSR VIDEO OUTPUT TO SCREEN
0320: 16A6 2C 58 1A BIT PRIFLG IS THE SERIAL PORT SELECTED?
0330: 16A9 50 03 BVC OUTALA
0340: 16AB 20 3A 13 JSR PRCHA* OUTPUT THE CHARACTER ON THE SERIAL PORT
0350: 16AE 2C 56 1A OUTALA BIT CENFLG IS THE CENTRONICS PORT SELECTED
0360: 16B1 50 03 BVC OUTALB
0370: 16B3 20 59 15 JSR CENOUT OUTPUT THE CHARACTER TO CENTRONICS
0380:
0390: 16B6 AC 29 1A OUTALB LDY PZPTR SWAP PAGE ZERO BYTES
0400: 16B9 AE 2A 1A LDX PZPTR +01
0410: 16BC 84 FC STY RAMPTR
0420: 16BE 86 FD STX RAMPTR +01
0430:
0440: 16C0 AE 2C 1A LDX XHOLD RESTORE THE REGISTERS
0450: 16C3 AC 2B 1A LDY YHOLD
0460: 16C6 AD 2D 1A LDA AHOLD
0470:
0480: 16C9 2C 80 1A BIT PAD
0490: 16CC 10 01 BPL BRKTST
0500: 16CE 60 RTS V=0, C=0, N=1, Z DEPENDS ON CHARACTER PRINTED
0510:
0520:
0530: *** EXIT FOR BREAK KEY (NPRCHA, PRCHA AND CENOUT) *****
0540:
0550:
0560: 16CF 2C 80 1A BRKTST BIT PAD
0570: 16D2 10 FB BPL BRKTST WAIT UNTIL KEY IS RELEASED
0580: 16D4 A2 FF LDXIM $FF
0590: 16D6 9A TXS RESET THE STACK POINTER
0600: 16D7 6C 7C 1A JMI BRKT JUMP TO AN USER SELECTABLE VECTOR (1A7C,1A7D)

```

DE 6502 KENNER

RECBYT THE JUNIOR PROMOTING CY (W&J) 15/04/84 PAGE 01

```

0010: ***** RECBYT *****
0020:
0030: Receive a byte is a subroutine to get a byte from the
0040: Elekterminal. equal or less than a value set at maxval
0050: -it will not return until a perfect input be executed
0060: -and a not hexadecimal (0→F) character, or a byte grea-
0070: ter than the preset value will cause the printing of
0080: backspace(s) and a restart.
0090: Adapted from BYTIN ($16AE-PME)
0100:
0110: *****
0120:
0130: A4 00 RECVD * $00A4 FLAG: 1111.1110 none character: V=1
0140: A6 00 MAXVAL * $00A6 1111.1111 one V=1
0150: FE 00 NIBBLE * $00FE 0000.0000 two V=0
0160: F0 11 PRBS * $11F0 Print a backspace
0170: AE 12 RECCHA * $12AE Monitor input routine
0180: 1E 14 ASHETT * $141E ASCII → HEXA (modified)
0190:
0200: 06A0 RECBYT ORG $06A0
0210: 06A0 A9 FE LDAIM $FE Set flag for nibble count
0220: 06A2 85 A4 STA RECVD
0230:
0240: 06A4 E6 A4 RECB INC RECVD
0250: 06A6 20 AE 12 JSR RECCHA Receive one character
0260: 06A9 20 1E 14 JSR ASHETT Convert it into a Hex nibble
0270: 06AC 90 10 BCC OK
0280:
0290: INPUT ERROR (carry set from ASHETT)
0300:
0310: 06AE 24 A4 BIT RECVD
0320: 06B0 70 05 BVS ONE
0330: 06B2 20 F0 11 TWO JSR PRBS V=0 Two character received. two BS to print
0340: 06B5 C6 A4 DEC RECVD
0350: 06B7 20 F0 11 ONE JSR PRBS V=1 One character received. one BS to print
0360: 06BA C6 A4 DEC RECVD
0370: 06BC 50 E6 BVC RECB Branch always
0380:
0390: NIBBLE RECEIVED
0400:
0410: 06BE 24 A4 OK BIT RECVD First or second nibble received?
0420: 06C0 50 08 BVC SECONDD
0430:
0440: 06C2 0A FIRST ASLA V=1 First (higher) data nibble
0450: 06C3 0A ASLA
0460: 06C4 0A ASLA
0470: 06C5 0A ASLA
0480: 06C6 85 FE STA NIBBLE
0490: 06C8 70 DA BVS RECB Branch always
0500:
0510: 06CA 05 FE SECONDD DRA NIBBLE V=0 Second (lower) data nibble
0520:
0530: 06CC C5 A6 CMP MAXVAL Byte equal or less than
0540: 06CE F0 04 BEQ DONE receive only equal or less than max.value
0550: 06D0 90 02 BCC DONE
0560: 06D2 B0 DE BCS TWO .... greater than max.value → repeat
0570:
0580: 06D4 60 DONE RTS V=0

```

SORTNG THE JUNIOR PROMOTING CY (W&J) 15/04/84 PAGE 01

0010: E000 SORTNG ORG \$E000

0020: SORT(ING) is a subroutine to arrange in alphabetic or
 0030: numeric order with increasing values. formatted files
 0040: of no more than 256 bytes each, and with a variable
 0050: amount of strings. Each individual string begins in the
 0060: same relative address in each file; the strings can be
 0070: numbers, birthday dates, names... or simple bits or
 0080: bytes with some meaning.
 0090: The strings chosen to one particular sorting (names,
 0100: numbers, dates...), are defined by its relative start
 0110: and end addresses in the file (YSTART, YSTOP).
 0120: The "last" file is the ETX (\$03) character for alphabe-
 0130: tic strings, or \$FE for numeric strings.
 0140: The distinction between alpha and numeric is only need-
 0150: ed if extra characteres are included (é, ç, ã...).
 0160: The sorting process is done by comparing the correspon-
 0170: ding bytes in pairs, and exchanging the files of the
 0180: pairs out of order. Adapted from "8-Bit Sort", 6502
 0190: Assembly Language Programming by L.A. Leventhal
 0200: * The program is slow, it takes about 5 minutes to sort
 0210: 600 Files of 64 bytes, due to the repeated exchange mo-
 0220: vements of the files on each pass...

EXAMPLE:

```

-----
[] BEGAD          [] YSTART  YSTOP []          FIELD []
$0 1 02 3 04 5 06--10-20-30-435 36 37 38 9 A B 3C D E F
1234 01 A 12 M Nelson...          72.04.31 I S A 0567 S N
2569 08 C 22 F Margarida...      66.12.01 F E R 5310 N N
0002 12 B 30 M Artur José...     83.06.22 I E A 2489 N S
:::
[] ENDAD
-----
    
```

In this example the files are \$40 bytes long and one is just sorting them by the string "names" (\$06+36).

```

0380: A6 00 MAXVAL * $00A6
0390: D4 00 ASCREG * $00D4
0400: D5 00 FIELD * $00D5 Bytes of each (formatted) file
0410: D7 00 YSTART * $00D7 Begin of string in file to sort
0420: D8 00 YSTOP * $00D8 End of string in file to sort
0430: D9 00 SRFLAG * $00D9 Sorting flag
0440: DA 00 ANFLAG * $00DA Alphanumeric flag
0450: E4 00 ENDADL * $00E4 Address of last file (ETX)
0460: E5 00 ENDADH * $00E5
0470: E6 00 CURADL * $00E6 Address of one pair's first file
0480: E7 00 CURADH * $00E7
0490: E8 00 CENDL * $00E8 Address of one pair's second file
0500: E9 00 CENDH * $00E9
0510: F0 00 MOVADL * $00F0 Address of buffer file
0520: F1 00 MOVADH * $00F1
0530: 63 1A PARA * $1A63 Address of first file
0540:
0550: A0 06 RECBYT * $06A0 Receive one byte equal or less than maxval
0560: D5 06 ASC 'S * $06D5 DATA: Real ASCII value of masked characters
0570: A5 07 PARPNT * $07A5 (1A00).Y → (Point).X
0580: 80 07 XZZ+Y * $07B0 Add Y to an address in page 00 pointed by X
0590: B2 07 XZZ+FD * $07B2 Add the value in Field...
0600: 75 0C XPARAY * $0C75 (Point).X → (Point).Y
0610: AE 12 RECCHA * $12AE Monitor input routine
0620: 8A 16 NPRCHA * $168A Monitor output routine
0630: 69 F6 INITAD * $F669 Input parameters PARA, PARB
0640: 39 F7 MESSA * $F739 Print "DONE"
0650: F2 FA PRINTV * $FAF2 Print characters that follow until ETX
0660:
0670: E000 20 F2 FA JSR PRINTV Print "ALPHANUMERIC SORTING"
0680: E003 0D = $0D CR
0690: E004 0A = $0A LF
0700: E005 0D = $0D CR
0710: E006 0A = $0A LF
0720: E007 41 = 'A
0730: E008 4C = 'L
0740: E009 50 = 'P
0750: E00A 48 = 'H
    
```

```

0760: E00B 41 = A
0770: E00C 4E = N
0780: E00D 55 = U
0790: E00E 4D = M
0800: E00F 45 = E
0810: E010 52 = R
0820: E011 49 = I
0830: E012 43 = C
0840: E013 20 =
0850: E014 53 = S
0860: E015 4F = O
0870: E016 52 = R
0880: E017 54 = T
0890: E018 49 = I
0900: E019 4E = N
0910: E01A 47 = G
0920: E01B 0D = $0D CR
0930: E01C 0A = $0A LF
0940: E01D 03 = $03 ETX
0950: E01E 88 = DEY →Y=FF
0960: E01F 84 A6 = STY MAXVAL Set Max. value
0970:
0980: E021 20 69 F6 SORT JSR INITAD PARA=BEGAD=CURAD, PARB=ENDAD=CEND
0990: E024 B1 E4 ETX LDAIY ENDADL Get begin and end addresses
1000: E026 C9 03 CMPIM $03 *****
1010: E028 F0 28 BEQ FIELD? If End of Text at ENDAD, it's OK.
1020: E02A C9 FE CMPIM $FE
1030: E02C F0 24 BEQ FIELD? If 'End of number' at ENDAD, it's OK too.
1040:
1050: E02E 20 F2 FA ENDAD? JSR PRINTV Print "ENDAD!? (03.FE): "
1060: E031 0D = $0D CR
1070: E032 0A = $0A LF
1080: E033 45 = E
1090: E034 4E = N
1100: E035 44 = D
1110: E036 41 = A
1120: E037 44 = D
1130: E038 21 = ?
1140: E039 3F = ?
1150: E03A 20 = (
1160: E03B 28 = 0
1170: E03C 30 = 3
1180: E03D 33 = .
1190: E03E 2C = F
1200: E03F 46 = E
1210: E040 45 = )
1220: E041 29 = :
1230: E042 3A =
1240: E043 20 =
1250: E044 03 = $03 ETX : Y comes =00
1260: E045 20 A0 06 JSR RECBYT Receive byte (any value)
1270: E048 85 E5 STA ENDADH
1280: E04A 20 A0 06 JSR RECBYT Get the address of the last file (ETX)
1290: E04D 85 E4 STA ENDADL *****
1300: E04F C8 INY →Y=00
1310: E050 F0 D2 BEQ ETX Check again for ETX at ENDAD
1320:
1330: E052 20 F2 FA FIELD? JSR PRINTV Print "FIELD?: "
1340: E055 0D = $0D CR
1350: E056 0A = $0A LF
1360: E057 46 = F
1370: E058 49 = I
1380: E059 45 = E
1390: E05A 4C = L
1400: E05B 44 = D
1410: E05C 3F = ?
1420: E05D 3A =
1430: E05E 20 =
1440: E05F 03 = $03 ETX
1450: E060 20 A0 06 JSR RECBYT Get the length of the formatted files
1460: E063 85 D5 STA FIELD *****
1470:
1480: E065 20 F2 FA STRSTP JSR PRINTV Print "YSTART?,YSTOP?: "
1490: E068 0D = $0D CR
1500: E069 0A = $0A LF

```

```

1510: E06A 59      =      Y
1520: E06B 53      =      S
1530: E06C 54      =      T
1540: E06D 41      =      A
1550: E06E 52      =      R
1560: E06F 54      =      T
1570: E070 3F      =      ?
1580: E071 2C      =      Y
1590: E072 59      =      S
1600: E073 53      =      T
1610: E074 54      =      O
1620: E075 4F      =      P
1630: E076 50      =      ?
1640: E077 3F      =      :
1650: E078 3A      =
1660: E079 20      =
1670: E07A 03      =      $03      ETX
1680: E07B A5 D5    LDA      FIELD      Set maxvalue for YSTART and YSTOP.
1690: E07D 85 A6    STA      MAXVAL     but = or < than FIELD
1700: E07F 20 A0 06 JSR      RECBYT     Get begin address of strings to sort
1710: E082 85 D7    STA      YSTART     *****
1720: E084 A9 2C    LDAIM   $2C
1730: E086 20 BA 16 JSR      NPRCHA     Print "."
1740: E089 20 A0 06 JSR      RECBYT     Get end address of strings to sort
1750: E08C 85 D8    STA      YSTOP     *****
1760:
1770: E08E 20 F2 FA  ALFNUM JSR      PRINTV     Print "ALPHA(A) OR NUMERIC(9)?:"
1780: E091 0D        =      $0D      CR      Enter a letter for alphabetic.
1790: E092 0A        =      $0A      LF      and a figure for numerical sorting
1800: E093 41        =      A
1810: E094 4C        =      L
1820: E095 50        =      P
1830: E096 48        =      H
1840: E097 41        =      A      NV-BDIZC
1850: E098 28        =      (      A=01000001 or any letter V=1
1860: E099 41        =      A      9=00111001 or any figure V=0
1870: E09A 29        =      )
1880: E09B 20        =
1890: E09C 4F        =      O
1900: E09D 52        =      R
1910: E09E 20        =
1920: E09F 4E        =      N
1930: E0A0 55        =      U
1940: E0A1 4D        =      M
1950: E0A2 45        =      E
1960: E0A3 52        =      R
1970: E0A4 49        =      I
1980: E0A5 43        =      C
1990: E0A6 28        =      (
2000: E0A7 39        =      9
2010: E0A8 29        =      )
2020: E0A9 3F        =      ?
2030: E0AA 3A        =      :
2040: E0AB 20        =
2050: E0AC 03        =      $03      ETX
2060: E0AD 20 AE 12  JSR      RECCHA     Set alpha/numeric flag
2070: E0B0 85 DA      STA      ANFLAG     *****
2080:
2090: E0B2 20 F2 FA  JSR      PRINTV     Print "CR,LF. *"
2100: E0B5 0D        =      $0D      CR
2110: E0B6 0A        =      $0A      LF
2120: E0B7 2A        =      *      ... This means computer working!
2130: E0B8 08        =      $08      BS
2140: E0B9 03        =      $03      ETX
2150:
2160:
2170:      Set parameters for the sorting process, first pass
2180:      *****
2190:
2200: E0BA A0 63      SORT 'G LDYIM $63      (PARAL)
2210: E0BC A2 E6      LDXIM $E6      (CURADL)
2220: E0BE 20 A5 07  JSR      PARPNT     CURAD ← PARA
2230: E0C1 A0 00      LDYIM $00
2240: E0C3 84 D9      STY      SRFLAG     Reset sorting flag
2250: E0C5 A0 E8      LDYIM $E8      Y=CEND; X is still $E6. CURADL

```

```

2260: E0C7 20 75 0C      JSR  XPARAY  CURAD + CEND
2270: E0CA A2 E8          LDXIM $E8   CENDL
2280: E0CC 20 B2 07      JSR  XZZ+FD  CEND + CEND+FIELD
2290:
2300:
2310:
2320:
2330:
2340:
2350: E0CF A4 D7          CMPARE LDY   YSTART
2360:
2370: E0D1 24 DA          COMPRE BIT   ANFLAG Alphabetic or Numeric?
2380: E0D3 70 06          BVS      ALPHA
2390:
2400: E0D5 B1 E6          NUMERI LDAIY  CURADL
2410: E0D7 D1 E8          CMPIY  CENDL
2420: E0D9 50 24          BVC     EQUAL?
2430:
2440: E0DB B1 E8          ALPHA  LDAIY  CENDL Get the ASCII value of the char. at CEND
2450: E0DD 10 0A          BPL    lower
2460: E0DF 29 7F          ANDIM  $7F   Mask bit 7 of extra characteres (#80-9F)
2470: E0E1 C9 20          CMPIM  $20   Filter extra charact. from italic characteres
2480: E0E3 B0 04          BCS    lower Italic characteres are #A0-#FE
2490: E0E5 AA
2500: E0E6 BD D5 06      LDAAX  ASC'S * AAAAAAIIUAAAaAaAaUaC...
2510:
2520: E0E9 29 DF          lower ANDIM  $DF   Mask bit 5 of lower case characteres
2530: E0EB 85 D4          STA   ASCREG Save it for comparation
2540:
2550: E0ED B1 E6          LDAIY  CURADL Get the ASCII value of the charac. at CURAD
2560: E0EF 10 0A          BPL    lower2
2570: E0F1 29 7F          ANDIM  $7F
2580: E0F3 C9 20          CMPIM  $20
2590: E0F5 B0 04          BCS    lower2
2600: E0F7 AA
2610: E0F8 BD D5 06      LDAAX  ASC'S
2620: E0FB 29 DF          lower2 ANDIM  $DF
2630:
2640: E0FD C5 D4          CMP   ASCREG Compare the actual ASCII values
2650:
2660:
2670: E0FF D0 07          EQUAL? BNE   NOTEQU First check Z flag
2680:
2690:
2700:
2710:
2720:
2730:
2740: E101 C8
2750: E102 C4 D8          EQUAL! INY
2760: E104 D0 CB          CPY   YSTOP
2770: E106 F0 02          BNE   COMPRE
2780:
2790: E108 B0 1C          BEQ   GO-ON Same value at CURAD/CENDAD + get next pair
2800:
2810:
2820:
2830:
2840:
2850:
2860: E10A A2 E6          GO-ON LDXIM  $E6   CURAD
2870: E10C 20 B2 07      JSR   XZZ+FD  CURAD + CURAD+FIELD ; up-date CURAD
2880: E10F A2 E8          LDXIM  $E8   CEND
2890: E111 20 B2 07      JSR   XZZ+FD  CEND + CEND+FIELD ; up-date CEND
2900: E114 A5 E8          LDA   CENDL
2910: E116 C5 E4          CMP   ENDADL
2920: E118 D0 B5          BNE   COMPARE Last file? (CEND=ENDAD)
2930: E11A A5 E9          LDA   CENDH
2940: E11C C5 E5          CMP   ENDADH
2950: E11E D0 AF          BNE   COMPARE
2960: E120 A5 D9          LDA   SRFLAG Sorting flag * There was any change of pairs?
2970: E122 D0 96          BNE   SORT'G ... if flag set, pass once more
2980: E124 F0 29          BEQ   DONE   ... if flag clear, the files are ordered
2990:

```

```

3000:
3010:
3020:
3030:
3040: E126 A0 01
3050: E128 84 D9
3060: E12A 88
3070: E12B 84 F0
3080: E12D 84 F1
3090: E12F A4 D5
3100: E131 88
3110: E132 B1 E8
3120: E134 91 F0
3130: E136 88
3140: E137 10 F9
3150: E139 A4 D5
3160: E13B 88
3170: E13C B1 E6
3180: E13E 91 E8
3190: E140 88
3200: E141 10 F9
3210: E143 A4 D5
3220: E145 88
3230: E146 B1 F0
3240: E148 91 E6
3250: E14A 88
3260: E14B 10 F9
3270:
3280: E14D 30 BB
3290:
3300: E14F A0 1F
3310: E151 4C 39 F7
3320:
3330:
3340:
3350:
3360: E154 84 D5
3370:
3380: E156 18
3390: E157 85 00
3400: E159 65 D5
3410: E15B 95 00
3420: E15D B5 01
3430: E15F 69 00
3440: E161 95 01
3450: E163 60
3460:
3470:
3480: E164 85 00
3490: E166 99 00 00
3500: E169 85 01
3510: E16B 99 01 00
3520: E16E 60
3530:
3540:
3550: E16F B9 00 1A
3560: E172 95 00
3570: E174 B9 01 1A
3580: E177 95 01
3590: E179 60
3600:

```

GREATR LDYIM #01
 STY SRFLAG Sorting flag ← 01 : one pair is changed!
 DEY →Y=00
 STY MOVADL
 STY MOVADH MOVAD ← 0000 Buffer address for exchange
 LDY FIELD movements. If the files aren't too long,
 DEY the move address can be the stack.

LOADCH LDAIY CENDL
 STAIY MOVADL File at CEND → MOVAD
 DEY
 BPL LOADCH
 LDY FIELD
 DEY

LOADC LDAIY CURADL
 STAIY CENDL File at CURAD → CEND
 DEY
 BPL LOADC
 LDY FIELD
 DEY

LOAD LDAIY MOVADL
 STAIY CURADL File at MOVAD → CURAD
 DEY
 BPL LOAD

LODEND BMI GO-ON

DONE LDYIM #1F
 JMP MESSA Print "DONE" and RTS

***** SUBROUTINES *****

STY FIELD Add Y to point
 CLC Add FIELD to point
 LDAZX #00
 ADC FIELD
 STAZX #00
 LDAZX #01
 ADCIM #00
 STAZX #01
 RTS

LDAZX #00 Move a point in page 00, pointed by X
 STAY #0000 register, to another one, pointed by Y
 LDAZX #01
 STAY #0001
 RTS

LDAAY #1A00 Move one parameter in page 1A, pointed by Y
 STAZX #00 To an address in page 00 pointed by X
 LDAAY #1A01
 STAZX #01
 RTS

50 63 MLIST

SCR # 50

```

0 *****
1 *
2 *       HET VERANDEREN VAN TAPE-FORTH NAAR DISC-FORTH
3 *
4 *****
5
6 DOOR : M.C. BREUKINK
7       SCHOUTENDREEF 66
8       2542 LN DEN HAAG
9       TEL.: 070 - 666497
10
11
12       (RED.: zie ook eerdere publikatie van Gert Klein)
13
14
15

```

SCR # 51

```

0 NA EEN TIJD MET FIG-FORTH 1.1 GEWERKT TE HEBBEN KWAM TOCH HET
1 VERLANGEN OM IN PLAATS VAN EEN CASSETTERECORDER EEN DISC-DRIVE
2 AAN TE SLUITEN.
3 NA HET KOPEN VAN EEN DISC-DRIVE EN HET BOUWEN VAN EEN INTERFACE.
4 BEN IK GAAN UITZOEKEN HOE FORTH NU EIGENLIJK MET EEN DISC WERKT.
5 DE MANIER WAAROP FORTH MET EEN DISC WERKT IS HEEL ANDERS ALS BIJ
6 EEN DISC OPERATING SYSTEM (DOS). NAMELIJK:
7
8 A. FORTH ZIET DE DISC SIMPELWEG ALS EEN UITBREIDING V.H. GEHEUGEN
9 B. FORTH MAAKT GEEN GEBRUIK VAN EEN DIRECTORY. ZODAT DE PLAATS
10 VAN EEN SCREEN (1024 BYTES) OP DE DISC ALTIJD GELIJK IS EN
11 DAT DE GEHELE DISC GEBRUIKT KAN WORDEN.
12
13 FORTH MAAKT GEBRUIK VAN DISC-BUFFERS IN HET RAM-GEHEUGEN.
14 IN FIG-FORTH V1.1 IS ER RUIMTE IN DE BUFFERS VOOR 1 SCREEN.
15 DE BUFFERS ZIJN ALS VOLGT OPGEBOUWD:

```

SCR # 52

```

0 BYTE 1 = HET BLOCKNUMMER VAN DE BUFFER.
1 BYTE 2 = DEZE BYTE BEVAT HET UPDATE BIT WELKE AANGEEFT DAT DE
2 DATA UIT DEZE BUFFER NAAR DISC GESCHREVEN MOET WORDEN
3 ( ALS BIT 7 GESET IS )
4 BYTE 3 T/M 258 = DE EIGENLIJKE SCREEN INFORMATIE
5 ( ALTIJD EEN VEELVOUD VAN 128 )
6 BYTE 259 EN 260 = DEZE BYTES ZIJN ALTIJD 00 EN DIENEN ALS
7 STOPBYTES VOOR DE COMPILER.
8
9 IN MIJN GEVAL (MET 256 BYTES/SECTOR) ZIJN ER 4 BUFFERS AANWEZIG.
10 STANDAARD MAAKT DE FIG-FORTH V1.1 GEBRUIK VAN 128 BYTES/SECTOR.
11 OM FORTH AAN TE PASSEN AAN ANDERE SECTORLENGTEN MOETEN DE
12 VOLGENDE CONSTANTEN AANGEPAST WORDEN :
13
14 FIRST = HET BEGINADRES VAN DE DISC-BUFFERS ( BIJ MIJ 5B60 )
15 LIMIT = HET EINDADRES VAN DE DISC-BUFFERS ( BIJ MIJ 5F70 )

```

SCR # 53

```

0 B/BUF = HET AANTAL BYTES PER BUFFER ( BIJ MIJ 256 )
1 B/SCR = HET AANTAL BUFFERS PER SCREEN ( BIJ MIJ 4 )
2 VERVOLGENS MOETEN ER EEN AANTAL PRIMITIVES EN HIGH-LEVEL
3 WOORDEN AANGEPAST WORDEN. TE BEGINNEN MET DE PRIMITIVES :
4 HET WOORD -DISC BESTAAT UIT MACHINETAAL EN LESTEST OF SCHRIJFT
5 EEN SECTOR NAAR DISC.
6 342B 85 : BEGIN WOORD -DISC ( LENGTE IS 5 )
7 342C 2D 44 49 53 C3 : NAAM -DISC
8 3431 CF 33 : LINKADRES NAAR --)
9 3433 35 34 : BEGIN MACHINETAAL - DISCDISC
10 3435 B5 00 LDA 0,X : HAAL SECTOR NUMMER
11 3437 85 02 STA FSEC
12 3439 B5 02 LDA 2,X : HAAL TRACK NUMMER
13 343B 85 03 STA FTRK
14 343D 86 B5 STX XSAVE : RED X-REGISTER
15 343F A5 05 LDA RWF : HAAL R/W FLAG

```

SCR # 54

```

0 3441 D0 06      BNE 3449      : R/W=1 LEZEN
1 3443 20 00 E0 JSR SWRITE     : SCHRIJF EEN SECTOR NAAR DISC
2 3443 4C 4C 34 JMP 344C
3 3449 20 0A E0 JSR SREAD      : LEES EEN SECTOR VAN DE DISC
4 344C A6 B5      LDX B5       : HAAL X-REGISTER WEER OP
5 344E EA EA EA EA EA EA EA
6 3456 EA EA EA EA EA EA EA
7 345E EA EA EA EA          : NOP
8 3462 AD FC DF LDA STREG      : STATUS REGISTER FDC
9 3465 95 04      STA 04.X     : EVT ERROR OP DE STACK
10 3467 4C EC 21 JMP POPTWO
11
12 HET VRIJE GEDEELTE VAN ADRES 344E - 3461 KAN GEBRUIKT WORDEN
13 ALS MEN 2 EN/OF DUBBELZIJDIGE DRIVES GEBRUIKT.
14 ( BIJ 2 DRIVES IS HET DRIVE NUMMER TE VINDEN OP ADRES 4.X )
15

```

SCR # 55

```

0 DE ROUTINE SREAD START DE DRIVE. BEWEEGT DE KOP NAAR DE JUISTE
1 TRACK EN LEEST DE GEWENSTE SECTOR.
2 DE SECTOR WORDT IN HET GEHEUGEN GEZET VANAF HET ADRES WAT ZICH
3 OP ADRES 00 EN 01 BEVINDT.
4 SWRITE DOET BIJNA HETZELFDE MAAR SCHRIJFT EEN SECTOR NAAR DISC.
5 HET STUK MACHINETAAL VAN -DISC IS ERG AFHANKELIJK VAN HET TOE-
6 GEPASTE DISC-SYSTEEM.
7
8 VERVOLGENS MOETEN EEN AANTAL HIGH-LEVEL WOORDEN Aangepast worden
9 HET WOORD DAT VOLGT NA -DISC IS -BCD.
10 HET WOORD -BCD ZET EEN BINAIRE WAARDE OM NAAR BCD.
11 ( ALS SREAD EN SWRITE GENOEGEN NEMEN MET EEN BINAIRE WAARDE KAN
12 DIT WOORD VERVALLEN )
13 HET WOORD -BCD IS ALS VOLGT OPGEBOUWD :
14
15 346A 84          :: RECHTER NIBBLE = LENGTE WOORD

```

SCR # 56

```

0 346B 2D 42 43 C4      : NAAM -BCD
1 346F 2B 34          : LINKADRES NAAR -DISC
2 3471 D6 25          : DOCOL ( : )
3 3473 5D 26          : ZERO ( CONSTATE 0 )
4 3475 67 20          : CLIT ( BYTE VALUE )
5 3477 0A            : 10
6 3478 5B 23          : USLAS ( U/ )
7 347A 67 20          : CLIT
8 347C 10            : 16
9 347D 44 31          : STAR ( * )
10 347F AD 23         : OR
11 3481 0C 24         : SEMIS ( : )
12
13 DIT IS ONGEVEER GELIJK AAN : -BCD 0 10 U/ 16 * OR :
14
15

```

SCR # 57

```

0 HET WOORD R/W ZORGT VOOR HET LEZEN EN SCHRIJVEN VAN EN NAAR DISC
1 ADRES N F R/W ADRES=SOURCE OF DESTINATION ADRES N=BLOCK NUMMER
2 F=R/W FLAG 0=SCHRIJVEN 1=LEZEN
3 3483 83          : RECHTER NIBBLE = LENGTE WOORD
4 3484 52 2F D7     : NAAM R/W
5 3487 6A 34          : LINKADRES NAAR -BCD
6 3489 D6 25          : DOCOL ( : )
7 348B 73 24          : ZEQU ( 0= )
8 348D 2A 20          : LIT
9 348F 00 05          : ADRES 5 = R/W FLAG
10 3491 B5 25         : Cstor ( C! )
11 3493 22 25         : SWAP
12 3495 5D 26         : ZERO CONSTATE 0
13 3497 9D 25         : STORE ( ! )
14 3499 5D 26         : ZERO
15 349B 08 25         : OVER

```

```

SCR # 58
0 349D 3C 28 : GREAT ( ) )
1 349F 08 25 : OVER
2 34A1 2A 20 : LIT
3 34A3 FF 04 : SECTL-1 ( SECTL = SECTOREN 2 DRIVES )
4 34A5 3C 28 : GREAT
5 34A7 AD 23 : OR
6 34A9 67 20 : CLIT
7 34AB 06 : 6
8 34AC 11 29 : QERR ( 6 ?ERROR = DISC RANGE? )
9 34AE 5D 26 : 0
10 34B0 2A 20 : LIT
11 34B2 80 02 : SECTR ( SECTR=SECTOREN PER DRIVE )
12 34B4 5B 23 : USLAS ( U/ )
13 34B6 A1 27 : ONEP ( 1+ )
14 34B8 22 25 : SWAP
15 34BA 5D 26 : ZERO

```

```

SCR # 59
0 34BC 67 20 : CLIT
1 34BE 10 : 16 SECTOREN PER TRACK
2 34BF 5B 23 : USLAS
3 34C1 71 34 : DBCD ( -BCD KAN EVT VERVALLEN )
4 34C3 22 25 : SWAP
5 34C5 A1 27 : ONEP
6 34C7 71 34 : DBCD ( KAN EVT VERVALLEN )
7 34C9 33 34 : DDISC ( -DISC )
8 34CB 67 20 : CLIT
9 34CD 08 : 8
10 34CE 11 29 : QERR ( 8 ?ERROR= DISC ERROR )
11 34D0 0C 24 : SEMIS ( : )
12 34D2 0C 24 : SEMIS
13
14 SECTR EN SECTL ZIJN GEBASEERD OP EEN DISC-SYSTEEM MET 16 SECTORS
15 EN 40 TRACKS PER DISC.

```

```

SCR # 60
0 NU MOETEN WE IN DE VOLGENDE FORTH WOORDEN WAARDEN VERANDEREN :
1 : DR1 640 OFFSET ! : ( 640 SECTOREN PER DRIVE )
2 WERK ALLEEN MET DE TWEEDE DRIVE. ( ONBELANGRIJK BIJ 1 DRIVE )
3
4 : +BUF 260 + DUP LIMIT = IF DROP FIRST THEN DUP PREV @ - :
5 +BUF LEVERT HET ADRES VAN DE VOLGENDE BUFFER
6
7 IN DE EDITOR BEVINDT ZICH NOG EEN FLUSH :
8 : FLUSH 4 0 DO 32767 BUFFER DROP LOOP :
9 MET FLUSH WORDEN DE BUFFERS WAARVAN DE UPDATE BIT GESET IS NAAR
10 DISC GESCHREVEN.
11 HET SETTEN VAN DE UPDATE BIT VAN EEN BUFFER GEBEURT AUTOMATISCH
12 ALS MEN MET DE EDITOR IN EEN BUFFER IETS VERANDERD.
13 HET SETTEN KAN OOK GEBEUREN DOOR UPDATE IN TE TYPEN.
14 ( DIT GEBEURT DAN ALLEEN MAAR IN DE HUIDIGHE BUFFER !!! )
15

```

```

SCR # 61
0 VARIABLE PREV = HET ADRES VAN DE MEEST RECENT AANGESPROKEN
1 BUFFER ( LEZEN OF SCHRIJVEN )
2 VARIABLE USE = HET ADRES VAN DE MINST RECENT AANGESPROKEN
3 BUFFER ( HIER KOMT DE NIEUW GELEZEN DISC-SECTOR TERECHT )
4 ( USE EN PREV MOETEN 5B60 5C64 5D68 OF 5E6C ZIJN BIJ OPSTARTEN )
5
6 BUFFER = N --- ADRES
7 N IS HET BLOCK NUMMER VAN DE BUFFER.
8 ALS HET UPDATE BIT VAN DE BUFFER GESET IS WORDT DEZE BUFFER NAAR
9 DISC GESCHREVEN. ( BUFFER SCHRIJFT ALLEEN NAAR DISC )
10 HET ADRES IS HET ADRES VAN BUFFER N .
11
12 BLOCK = N --- ADRES
13 N IS WEER HET BLOCK NUMMER VAN DE BUFFER.
14 DIT WOORD GEEFT HET ADRES VAN HET BLOCK NUMMER N.
15 ALS DIT BLOCK NOG NIET IN HET GEHEUGEN STAAT WORDT DIT VAN DISC

```

SCR # 62

```

0 GELEZEN. ALS DE UPDATE FLAG VAN DE BUFFER WAAR DE SECTOR NAAR
1 TOE MOET GESET IS. WORDT DEZE BUFFER EERST NOG NAAR DISC
2 GESCHREVEN VOORDAT DE NIEUWE SECTOR WORDT INGELEZEN.
3 HET WOORD BLOCK MAAKT O.A. GEBRUIK VAN DE VARIABELEN PREV. USE
4 EN DE WOORDEN +BUF EN BUFFER.
5 ALS MEN OVER EEN ZGN DECOMPILER BESCHIKT KAN MEN BEKIJKEN HOE
6 DE WOORDEN, VARIABELEN EN CONSTANTEN ERUIT ZIEN DIE MET DISC TE
7 MAKEN HEBBEN.
8
9 DE 'NIEUWE' DISC-FORTH WERKT AL MAANDENLANG ZONDER PROBLEMEN.
10 MAAR PAS OP : NA EENMAAL MET DISC-FORTH GEWERKT TE HEBBEN. WIL
11 JE DE TAPE-FORTH NOOIT MEER GEBRUIKEN!
12
13
14 VOOR MEER INFORMATIE OVER HET AANPASSEN VAN TAPE NAAR DISC KUNT
15 U MIJ ALTIJD BELLEN OF SCHRIJVEN.

```

SCR # 63

```

0 REFERENTIES :
1 FIG-FORTH 6502 ASSEMBLY SOURCE LISTING V1.1
2 FORTH INTEREST GROUP, P.O. BOX 1105, SAN CARLOS
3 CA 94070.
4
5 STARTING FORTH. LEO BRODIE, FORTH INC.
6
7 DIVERSE FORTH DIMENSIONS. FORTH INTEREST GROUP.
8
9
10
11
12
13
14
15

```

OK

0 LIST

SCR # 0

```

0
1 ( Het gebeurt wel eens dat een FORTH-screen geladen wordt )
2 ( en het oefft een foutboodschap. Je kunt echter niet ont- )
3 ( dekken waar de fout zit. Er kunnen controle-karakters in )
4 ( het screen zitten, die je niet ziet. Met de volgende de- )
5 ( finitie worden ze wel zichtbaar ).
6
7
8
9
10 ( Wout van Dinther, Molenriinselaan 27, 5262 TN Vuucht )
11
12
13
14
15

```

OK

1 LIST

SCR # 1

```

0 ( Zoek in screen n1 naar controle karakters )
1 : ZOEK ( n1 -- )
2   BLOCK
3   1024 0 DO DUP C@ DUP 32 <
4     IF CR ." ^ " 64 + EMIT
5     ." @ : " DUP U. ELSE DROP
6   ENDIF 1+ LOOP DROP :
7 : REINIG ( n1 -- )
8   BLOCK
9   1024 0 DO DUP C@ 32 <
10    IF DUP 32 SWAP C!
11    ENDIF 1+ LOOP DROP :
12
13 ( ZOEK laat het karakter zien )
14 ( REINIG maakt er een spatie van )
15

```

OK

VRAAG EN AANBOD

TE KOOP AANGEBODEN

8" FDOS controllerkaart met FD 1771 fl. 90,00
 2 x BK Ram/Rom kaart met 2 x BK Ram en in
 Euro's FDOS 1.2 voor 8" fl. 100,00
 Verkoop ten behoeve van een Belgisch lid.
 Voor informatie of bezichtiging:
 Tel.: 01807 - 19881
 Verzendkosten zijn niet in de prijs inbe-
 grepen.

SD+DD DOS DOSAV2.0A

JUNIOR-computer met PM en TM. Single density + double den-
 sity disk operating system voor 5" met FDC 1770. Van onze
 Belgische vriend Ir. Koen van Nieuwenhove. Ontwikkeld op
 eigen JUNIOR-computer. Al enkele jaren draaiend op de re-
 daktie.

Informatiepakket met complete becommentarieerde source-
 listing. Met patch-routine voor werken met Microsoft KB-9
 Basic voor JUNIOR. Patch-routine voor werken met assembler
 van Carl Moser voor JUNIOR (verkrijgbaar bij Ingenieursbu-
 ro Schroeder te Eindhoven). Patch-routine voor werken met
 Pascal (indien men daarover beschikt). Overdrukken van arti-
 kelen uit BYTE en diverse informatie voor het systeem
 met clubfaciliteiten. Met schema's en command summary in
 inleidend artikel.

Hans Buurman publiceerde inmiddels al een patch die het
 DOS 250% sneller maakt.

De controllerkaart voor DOSAV2.0A dient aan de hand van
 schema's zelf te worden ontwikkeld door middel van road-
 running of wire-wrap met eventueel speciale VERO-experi-
 menteerkaart Eurocard No. 10 - 0581 B, of dubbelzijdige
 doorgemetalliseerde kaart maken).

De floppy-disk controller FDC 1770 is in elk geval ver-
 krijgbaar bij M.C.A. Tronix, Route du Condroz 513, B-4200
 Luik, België.

Informatiepakket fl. 97,50
 Inleidend artikel uit informatiepakket fl. 10,00

Bestellingen te plaatsen door overschrijving van het ver-
 schuldigde bedrag op postrekening 841433 t.n.v. W.L. van
 Pelt te Kriemden a.d. IJssel, met vermelding van hetgeen
 gewenst wordt.

 * De redaktie zoekt voor publikatie in DE 6502 KENNER *
 * artikelen van leden die in het bezit zijn van een BBC- *
 * computer, een APPLE, alle COMMODORE machines, de ATARI *
 * en de OSI, en alle compatibles. *
 * Heeft u zelf gemaakte programma's ontwikkeld, klein of *
 * groot, ingewikkeld of niet, in assembler, Basic, Forth *
 * of Comal, in Pascal, Fortran, Cobol, in C, Logo, Elan, *
 * en wat dies meer zij, stuur het op naar de redaktie en *
 * draag uw steentje bij tot het verbeteren van de samen- *
 * stelling van Uw eigen clubblad. *
 * De redaktie helpt u met het publikatierijp maken en de *
 * onmaak. Heeft u geen printer, geen nood, wij typen het *
 * programma of uw artikel wel uit. *
 * Schroom niet, andere leden zitten op uw bijdrage al te *
 * wachten en zij kunnen ook van u leren. *
 * Jongeren nodigen we ook uit hun bedenkensel in te zenden *
 * naar de redaktie. En vergeet niet een jury beoor- *
 * deelt welke van de gepubliceerde programma's/artikelen *
 * in aanmerking kan komen voor de door onze club ter be- *
 * schikking gestelde prijzen! *

DUDE EDITIES "DE 6502 KENNER"

Reeds verschenen edities van DE 6502 KENNER kunnen nog be-
 steld worden door fl. 9,00 per editie over te schrijven op
 postrekening 3757649 ten name van de penningmeester van de
 KIM Gebruikersclub Nederland te Kriemden a.d. IJssel, onder
 vermelding van de gewenste editie(s).
 De edities worden afgeleverd door de sekretaris van de club

De volgende edities zijn nog verkrijgbaar:

NR. 16 MEI 1981	NR. 17 AUG 1981
NR. 18 OKT 1981	NR. 19 DEC 1981
NR. 20 NIET MEER LEVERBAAR	NR. 21 NIET MEER LEVERBAAR
NR. 22 AUG 1982	NR. 23 OKT 1982
NR. 24 DEC 1982	NR. 25 FEB 1983
NR. 26 MEI 1983	NR. 27 AUG 1983
NR. 28 OKT 1983	NR. 29 DEC 1983
NR. 30 FEB 1984	NR. 31 APR 1984
NR. 32 JUN 1984	NR. 33 AUG 1984
NR. 34 OKT 1984	NR. 35 DEC 1984

Bestellen van de gehele jaargang 1984, waarin zes edities
 zijn verschenen, kost slechts fl. 45,= !!

 * U kent vast wel iemand uit Uw omgeving of familie die *
 * in het bezit is van een computer met een 65xx processor *
 * en wellicht overgehaald kan worden om lid te worden van *
 * onze unieke club. Stuur de redaktie zijn/haar adres op *
 * dat wij betrokkene kunnen voorzien van informatie. Ook *
 * op Uw werk kunt U eens wat moeite doen. We hebben sterk *
 * behoefte aan meer leden. We willen graag nog meer voor *
 * onze leden doen. Daarvoor zijn echter veel meer leden *
 * nodig. Het adres van de redaktie is: Redaktie DE 6502 *
 * KENNER, o/a Willem L. van Pelt, Jacob Jordaanstraat 15 *
 * 2923 CK Kriemden a.d. IJssel. Tel.: 01807 - 19881 *

ASM65 65CXXX-CROSS-ASSEMBLER for JUNIOR with Proton DOS.
 Complete source-listing developed by Rob Banen. fl. 97,50
 English version.

CLUB ASSEMBLER TEXTEDITOR C.A.T. for COMMODORE 64 computer
 With IF-THEN-ELSE, REPEAT-UNTIL, CASE-WHEN-OTHERWISE, etc.
 Developed by Ruud H. Ushoff (C) 1984.
 Manual (Dutch version) with cassette fl. 47,50

RAW QUICK TAPE for COMMODORE 64 computer
 Developed by Ruud H. Ushoff (C) 1984.
 Cassette with directions for use (Dutch version) fl. 12,50

SUPERMON for COMMODORE 64 computer
 Cassette fl. 12,50

FORTH-79 STANDARD papers fl. 20,00
 fig-FORTH 6502 Assembly Source Listing fl. 65,00
 fig-FORTH Installation Manual with Glossary, fl. 45,00
 Model, Editor
 FORTH-79 STANDARD conversion fl. 20,00

CFAST5 Controlling digital cassette recorder.
 Written by Koen van Nieuwenhove for JUNIOR-comp.
 This program controls several functions of the
 recorder over the free VIA 6522. Data transfer
 is about 6000 baud.
 Assembly source listing, Dutch version fl. 12,50

Prices only for members in Holland. Members in other coun-
 tries have to pay fl. 7,50 extra transfers, unless they pay
 with eurocheque.

** DE 6502 KENNERS ** — A CLUB FOR ALL 65xx USERS

Information about the club.

Most people are not well-informed about the first micro-computer club in The Netherlands. It was called The KIM Users Club The Netherlands. Today, the name of the club is not quite covering all the other members with a computer from the 65xx-family. So, the club appears also with the name DE 6502 KENNERS, and offers accommodation to all users of the 65xx-computers, like APPLE, COMMODORE, BBC, ACORN, ATARI, OSI, ITT 2020, AIM-65, CHE-1, CV-777, BASIS 108, PEARCOM, SYM-1, PET, VIC-20, Proton-computers, ORIC-1, ACC 1000, SYSTEM 65, PC-100, PALLAS, MINTA, FORMOSA, STARLIGHT ESTATE III, SBC 65/68, NCS 6502, KEMPAC System 4, Elektor JUNIOR, Elektor SAMSON65, KIM-1, etc., etc.

International

The club has members in The Netherlands, Belgium, Germany, France, Spain, Portugal, America, India, Zambia, etc. etc.

Purpose

The club has been established 29th of January 1977 with the purpose to promote the exchange of knowledge among the users of systems mentioned before.

Meetings

In The Netherlands the club organises meetings each third saturday of the odd months, except in July. The club delivers lectures of all kinds, about programming machine language, Basic, Forth, etc. etc.

Cassette Libr.

The cassette library is free available for all members by ordering cassettes. Programs on cassette has been developed by members of the club, mostly published in our club magazin DE 6502 KENNER. All rights of the authors are reserved.

Paperware

The paperware service exists of providing source-listings and articles published in our club magazin DE 6502 KENNER. There are also listings in English.

Magazine

The magazin DE 6502 KENNER appears 5 times a year. The articles are written by members of the club from all countries, aided by the redaction. Articles of all kind, hardware as well as source-listings in assembly with comments, Basic, Forth, Comal, Focal, Logo, Elan, Pascal, Cobol, Fortran, C, etc. etc. Used assemblers: Micro-ADE, ASS/TED, Proton assembler, FATE, Big Mac, and so on. Each edition counts 48 pages at least. Some source-listings are too long to be published, like a 12K FATE. They will be available in the paperware-service. An independent jury awards prizes for some contributions.

Membership

Membership is open for all natural persons. Both for novice and advanced. Subscribe on Eurocheque HFL. 45,== and send to: Editorial Office DE 6502 KENNER
c/o Willem L. van Pelt
Jacob Jordaensstraat 15
2923 CK Krimpen a.d. IJssel
The Netherlands.
If not paying with Eurocheque: HFL. 52,50 !

Paperware-service

Micro-ADE 8K Assembler/Disassembler/Editor. For use with any 6502 microcomputer. Micro-ADE was developed by Peter Jenninos of Micro-Ware Ltd Canada, (C) 1977. (C) 1982 by KIM Users Club.

FATE 12K Format Lister/Assembler/Tape-utilities/Editor. For use with Elektor's JUNIOR-computer with PM + TM. Also for use with KIM-1 computer (separate I/O patches). We now try to make a FATE working on APPLE-computer. FATE was developed by our member Rob Banen, based on object information and published with permission of Proton Electr. Naarden, The Netherlands. (C) 1984. We recommend free memory spaces of at least 16K, better 32K

FATEC 12K Format Lister/Assembler/Tape-utilities/Editor. Same FATE, but now for use with 65C02-processor.

ASM65C CrossAssembler for use with 65C02-processor. Developed by Rob Banen for Elektor's JUNIOR-computer with Proton Disk Operating System (Senior).

CAT 8K two pass Club Assembler Texteditor for C-64. Developed by our member Ruud H. Uhoff. Assembler with IF-THEN-ELSE, REPEAT-UNTIL, WHILE-DO, CASE-WHEN-OTHERWISE, MOVE, COPY, DELETE, RENUMBER, CHANGE, etc. With build-in structuremacro's and controlstructures. English manual not available yet.

FORTH-79 STANDARD
fig-FORTH 6502 Assembly Source Listing
fig-FORTH Instalation Manual with Glossary, Model, Editor
FORTH 79 STADARD Conversion

fig-FORTH for JUNIOR with Proton Disk Operating System. Developed by our member Gert van Dobroek.

FYS-FORTH for APPLE II
2nd FYS-FORTH VSN 0.3 Release of July 4th. 1984.

Disassembler for the Rockwel 65C02. Developed by Rob Banen for JUNIOR with Proton DOS.

CFAST5 Controlling digital recorder 6000 baud. Dutch version source-listing developed by our member Koen van Nieuwenhove for JUNIOR-computer with PM + TM.

JUNIOR on 5" Floppy. Dutch manual and English source-listing developed by Koen van Nieuwenhove, with scheme, Byte-articles and other informations. Ask for prices.

Floppy disk on AIM-65
Article published in DE 6502 KENNER nr. 28 Oct. 1983. Dutch version, with scheme. Author: D.A. Gerritsen.

Read basicode with your Microsoft SYM-1 Basic on Elektor's JUNIOR-computer with PM + TM, and Elektor's interface.

Read basicode with your Microsoft KB9-P Basic on Elektor's JUNIOR-computer with PM + TM, and Elektor's interface.

* Send your program listings to the editors office *
* c/o Willem L. van Pelt *
* Jacob Jordaensstraat 15 *
* 2923 CK Krimpen a.d. IJssel *
* The Netherlands *
