

KIM GEBRUIKERS CLUB NEDERLAND

Zesde Jaargang Nr. 20

Februari 1982

# LUSTRUMNUMMER

# KIM GEBRUIKERS CLUB

## NEDERLAND

een club van 6502 gebruikers

Het doel van de vereniging is: het bevorderen van de kennisuitwisseling tussen de gebruikers van 6502-computers, zoals KIM, SYM, JUNIOR, AIM 65, System-65, ACORN, PET en CBM, APPLE, ATARI, ITT 2020, PC 100, OHIO Scientific Challengers, etc. etc.

Dit doel wordt o.a. gerealiseerd door vijf maal per jaar het huisorgaan " DE 6502 KENNER " te publiceren en vijf maal per jaar een clubbijeenkomst te houden, en wel als volgt:

### Verschijningsdata DE 6502 KENNER

derde zaterdag  
van de maanden:  
februari,  
mei,  
augustus,  
oktober,  
december.

### Bijeenkomsten van de club

derde zaterdag  
van de maanden:  
januari,  
maart,  
mei,  
september,  
november.

Naast deze activiteiten kunnen de leden gebruik maken van clubfaciliteiten, zoals de KIM-Club-KIM, de KIM-Club-JUNIOR, en de cassettebibliotheek.

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

De samenstelling van het bestuur is thans als volgt:

Voorzitter:

Sekretaris:

Penningmeester:

Organisator:

Redactie:

Accomodatie:

Adres van de vereniging:

Lidmaatschappen:

### Dagelijks bestuur:

Anton Muller  
Sinj. Semeynstr. 78 - 1  
1061 GM Amsterdam (020 - 860245)

Ruud H. Uphoff  
Voorburgpad 10 (085 - 816935)  
6843 EM Arnhem (19.00 - 20.00 )

Herman C. Burgers  
Manus Peetstraat 57  
1183 LH Amstelveen (020 - 419958)

Leden:  
Rinus Vleesch-Dubois  
F. Nightingalestr. 212  
2037 NG Haarlem (023 - 330993)

Willem L. van Pelt  
Jacob Jordaensstr. 15  
2923 CK Krimpen a.d. IJssel  
(01807 - 19881)

Riche A. van Steen  
Vingerhoed 11  
6953 BZ Dieren (08330 - 21004)

zie adres v.d. sekretaris

f1. 40,= per kalenderjaar  
gironummer 3757649 t.n.v.  
H.C. Burgers te Amstelveen.

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

Adres voor het inzenden van en reakties op artikelen voor de 6502 KENNER:

Willem van Pelt  
J. Jordaensstr. 15  
Krimpen a/d IJssel  
(postcode 2923 CK)

Redactie 6502 KENNER:

Vaste medewerkers:

- Anton Müller
- Willem van Pelt
- Frans Smeehuijzen

Freelance medewerkers:

- Frans Engelsman
- Renee de Hoop

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

De 6502 KENNER verschijnt 5 x per jaar.

Copyright (C) 1982  
KIM Gebruikers Club  
Nederland.

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

Inhoud.....	1
Van het bestuur ...door Anton Muller.....	2
Van de redactie ...door Willem van Pelt.....	5
Structured programming (deel II) ...door Anton Muller.....	6
Programmeerwedstrijd ...samengesteld door Anton Muller.....	21
Schaakprogramma ...door Theo Kortekaas.....	36
Disassembler van Apple (Allen Baum/Stephen Wozniak) ...aangepast voor de KIM... ...door Siep de Vries en Coen van Nieuwenhove.....	63
PASCAL als assembler hulpmiddel ...door Hans Otten.....	73
SYM Microsoft BASIC aangepast voor de JUNIOR ...door Jaron Borensztajn.....	79
Aanvullingen op Elektuur EPROM programmer ...door John van Sprang.....	81
Aanpassingen Junior print voor 2716 ...door J. Vollerings.....	85
Bugs.....	86
Patches Micro ADE deel 11 voor de Junior ... door A. Hankel.....	87
Voedingsschema voor +5V, +12V en -12V, ...door Frans Smeehuijzen.....	91
Memorytest aangepast voor de Junior ... door Bert van Tiel.....	92
Bugs.....	93
Ledenlijst.....	94
Inhoudsopgave nrs. 1 t/m 19.....	98
Agenda...en...Bugs.....	100
Literatuurverwijzing VIA (6522).....	39
Uitnodiging KIM Club Lustrumbijeenkomst 200382 Middenpas. Uitnodiging KIM Club bijeenkomst 150582.....Middenpas.	

Voor U ligt het uitbundige jubileum nummer van de 6502 Kenner. Uitbundig vanwege het feit dat wij het vijfjarige bestaan vieren van de KIM Gebruikers Club Nederland die op 29 januari 1977 werd opgericht, op initiatief van Sier de Vries en Co Filmer, als eerste microcomputer gebruikers club in Nederland. Hadden we bij de oprichting van de vereniging een ledental van 36, dat zo gelijdelijk aan doorgroeide naar 200. Begin 1980 hebben wij (voor ons doen) eens flink aan de weg getimmerd, hetgeen er toe heeft geleid, dat we uit kwamen op zo'n 350 leden en met onze deelname aan de HCC dag hebben we er nog eens zo'n stuk of 50 leden bijgekregen, hetgeen neerkomt op een totaal van 400 leden per 1 januari 1982. Waren het in de begin periode hoofdzakelijk leden die een KIM hadden thans is de verscheidenheid van systemen onder de leden zo groot. Zo zijn we de volgende systemen in de ledenadministratie tegengekomen:

KIM - Junior - PET - CBM - Apple - ITT 2020 - Challenger - Superboard - TIM - AIM 65 - PC 100 - System/65 - SYM - Acorn - JOLT. De paar systemen die we nog missen (voor zover mij bekend) zijn de VIC 20, de ATARI 400 en 800 en de PEARCOM (het broertje van de Apple). Oh, vergat ik bijna onze gebroeders Henk en Jaap Berkhoudt, die al vanaf de oprichting lid zijn en een 8008 systeem hebben, maar zuiver lid zijn en blijven vanwege de gezelligheid tijdens de clubbijeenkomsten, waar je uiteraard ook het nodige opsteekt.

Van die clubbijeenkomsten (in totaal 25 over de afgelopen 5 jaar) heb ik er slechts een gemist en dat was de bijeenkomst in Coevorden. Je mag dus zeggen dat ik een aardige kijk heb op wat er al zo allemaal gebeurt tijdens die club bijeenkomsten. Daar wil ik wel iets over zeggen. Er is een keiharde kern van zo'n man of 25, die je op elke bijeenkomst aanwezig is, ongeacht waar die wordt gehouden en ongeacht wat het programma te bieden heeft. Fanatici dus, net als ik. Ik wil gewoon niets missen. Die ene bijeenkomst in Coevorden heb ik gemist omdat toevallig diezelfde dag een aantal kopstukken uit de informatica wereld (waaronder Niklaus Wirth), in Utrecht een aantal lezingen gaven over een aantal in de belangstelling staande onderwerpen, die ik voor geen goud wilde (en had willen) missen. Vandaar dus..

Verder zijn er nog eens zo'n man of 25, met iedere keer andere gezichten. Afgezien van de meestal bijzonder professionele lezingen, waar ook de beginners enorm veel van kunnen opsteken, is juist het aardige

van deze bijeenkomsten het informele karakter tussen de lezingen door, waar je kontakten kunt leggen met mede clubgenoten, die je vaak kunnen helpen met het oplossen van je problemen, of in ieder geval naar iemand verwijzen, die er wel verstand van heeft. En dat soort dingen is toch altijd leuk meegenomen.

En dan niet te vergeten het onderwerp MARKT, waarbij een ieder die iets te koop heeft zijn zegje kan doen, niet alleen de commercie, maar ook de gewone clubleden. En daar zijn vaak leuke dingen te koop, tegen hele schappelijke prijsjes.

Onze jubileum clubbijeenkomst in Amersfoort, op 20 maart a.s. is echter een heel andere bijeenkomst, dan tot nu toe gebruikelijk. We hebben een enorme ruimte tot onze beschikking en het is de bedoeling, dat zoveel mogelijk mensen hun systeem eens meenemen en eens laten zien wat zij er mee doen of zo. Verder proberen we nog eens een aantal toppers te laten zien, zoals de demonstratie van twee met een KIM bestuurd diaprojectoren van Theo Kortekaas, de besturing van een aantal stoplichten op een verkeerskruispunt van de heer Dral, het opwekken van lichtbeelden op een paneel van 2,5 x 2,5 meter met daarop gemonteerd 4096 fietslampjes, hetgeen gekoppeld zit aan de APPLE van Uwe Schröder. Hierbij is het de bedoeling dat het paneel met de lichtbeelden op de achtergrond staat en op de voorgrond (met muziek) een balletstuk wordt opgevoerd, waarbij er sprake is van enige overeenkomst tussen de lichtbeelden en de ballethoudingen.

Of de dansers ook mee zullen komen is nog niet bekend, maar op zich is de techniek die hier achter zit voor ons waarschijnlijk ook al genoeg om naar te kijken. Op de HCC-dag kreeg ik iemand aan onze stand, wiens naam mij zo een twee drie niet te binnen wil schieten, maar wel wat hij te vertellen had. Hij had namelijk een oud elektronisch orgel op de kop getikt, daaruit alle elektonica gesloopt en er voor in de plaats een SYM met 8 toongeneratoren van General Instruments gezet. Dok die meneer halen we met zijn orgel naar onze lustrum bijeenkomst.

U neemt toch zelf ook Uw systeem mee? Hoe meer toepassingen we die dag kunnen laten, des te leuker zal de hele happening verlopen. Indien U met vervoersproblemen zit, kijkt U dan een in de ledenlijst die in deze 6502 Kenner is afgedrukt. Wellicht woont er iemand bij U in de omgeving die U een lift wil geven.

De toegang voor deze lustrumbijeenkomst is gratis, U kunt komen en gaan wanneer U maar wilt, vrouwen en kinderen

en andere familieleden of kennissen zijn eveneens van harte welkom. We maken er een gezellige dag van. Mocht U willen deelnemen aan de gemeenschappelijke lunch, dan kan dit tegen een geringe vergoeding a f. 12,50

In tegenstelling tot het bezoeken van de bijeenkomsten, is het gros van de leden echter zuiver en alleen lid van de KIM club vanwege de 6502 Kenner. Het samenstellen van die 6502 Kenner loopt tegenwoordig op rolletjes, getuige het produkt dat thans voor U ligt. We hebben echter nooit genoeg kopij. Gezien de dikte van dit nummer, zijn wij bijna door alle kopij heen. We hebben echter voldoende vertrouwen in onze leden dat zij het pijl van de 6502 Kenner hoog houden en zorgen dat hij ook gevuld blijft. Voor de redactie zou het echter prettig zijn dat we een aantal 6502 Kenners vooruit zouden kunnen werken, zodat niet (zoals nu) altijd alles op het nipper-tje nog even klaar moet.

Mijn oproep in de vorige 6502 Kenner, voor mensen die op freelance basis willen meewerken aan het uitwerken van programmaatjes ter publikatie, is niet voor niets geweest. Maar liefst 2 leden hebben zich aangemeld, te weten Frans Engelsman, uit Hoogezand, die in het bezit is van een ITT 2020 met floppy disk en printer en bereidt is een deel van zijn vrije tijd te besteden aan het uitwerken van artikelen voor de 6502 Kenner. En Renee de Hoop uit Voorburg, met een uitgebreide SYM en printer, die eveneens bereidt is een deel van zijn vrije tijd ter beschikking te stellen om mee te helpen ons het allemaal een beetje gemakkelijker te maken. U ziet hun namen genoemd in de colofon. Zorg er voor dat ook Uw naam daarbij komt. En denk er om: het mes snijdt aan twee kanten. Je krijgt er iets voor terug....



Anton Muller,  
voorzitter

Op dit tijdstip nog de allerbeste wensen overbrengen lijkt wat aan de late kant, maar beter laat dan nooit. Dus, dan toch maar: Beste lezers, mogen alle goede dingen die denkbaar zijn uw deel worden. Veel computer-geroegen in het komend jaar, namens de leden van de redactie van DE 6502 KENNER.

De totstandkoming van deze editie is voor uw nieuwe redactie-sekretaris een hele opgave geweest. Niet dat hij het niet zag zitten, dat niet. Er was natuurlijk al veel geleerd van Hans Otten, wiens vertrek uit de redactie toch wel een situatie van "even wennen" met zich bracht. Ik neem aan dat ik vanaf deze plaats Hans nog een goede tijd mag toewensen. Overigens, hij blijft ons nog behulpzaam bij de werkzaamheden omtrent de cassette-bibliotheek. Cassettes moeten voorlopig nog bij Hans Otten worden besteld.

Het zal niet tot een steeds weerkerend ritueel gaan behoren om iedereen te bedanken voor de verleende hulp bij de totstandkoming van elke toekomstige editie. Elke vorm van hulp blijft per definitie iets om dankbaar voor te zijn. Maar in dit geval wil ik een uitzondering maken voor de ondersteuning die van John van Sprang mocht worden ondervonden. Zonder dat zou het werk aan de toekomst van het blad toch wat minder vlot verlopen zijn. John, bedankt voor het printen van veel plezierwerk.

Deze editie is er een met een bijzonder karakter. Het enthousiasme van de inzenders was van niet te beschrijven omvang. Zo zelfs, dat de redactievergadering het moeilijk heeft gehad om een zodanige uitgave samen te stellen dat zoveel mogelijk lezers het ook zouden ervaren als een nummer met een bijzonder karakter, zonder daarbij iemand te kort te doen. Uiteraard slagen we er nooit in het iedereen volledig naar de zin te maken. Ook ditmaal zal er wel reden zijn voor ons en voor andere leden het jammer te vinden dat een bepaalde inzending (nog) niet werd opgenomen. De keus kan echter niet worden bepaald zonder de niet geringe financiële last in de saten te houden. Desalniettemin hopen we een "goed nummer" te hebben kunnen maken. Waarvoor de inzenders alle lof verdienen.

\*\*\*\*\*  
\* LET OP HET NIEUWE \*  
\* REDAKTIE-ADRES \*  
\*\*\*\*\*

W.L. van Pelt.

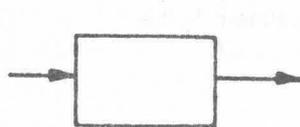
## STRUCTURED PROGRAMMING (Deel 2).

---

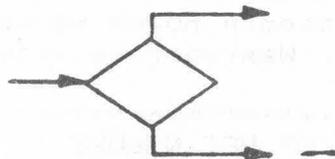
Door: Anton Muller  
 Sinj. Semeynsstr. 78 I  
 1061 GM AMSTERDAM  
 Tel.: 020 - 860245

Na lang wachten dan nu eindelijk het vervolg op mijn serie artikelen over structured programming, o.m. dat ik uit diverse gesprekken heb vernomen dat hier-voor binnen de club toch wel belangstelling bestaat. Voor degenen die nog niet zo lang lid zijn: in KIM Kenner 5 van 30 september 1978, heb ik deel 1 van bovengenoemd artikel geschreven, waarin werden behandeld alle control structures in Nassi/Schneiderman diagram notatie (ref. 1) en in de flow diagram notatie, alsmede het gebruik van deze control structures in de vorm van (denkbeeldige) macro's in assembly language. Voor de volledigheid laat ik de toen behandelde stof nog even in het kort de revue passeren. Mocht U geïnteresseerd zijn in een herdruk van deel 1 van dit verhaal uit KIM Kenner 5, stuur dan een gefrankeerde en aan Uzelf geadresseerde envelop (f. 1,30 postzegels) aan ondergetekende en de kopie komt per omslaande bij U in de bus.

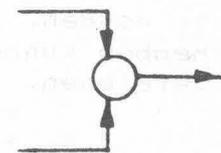
We kunnen een programma weergeven in de vorm van een stroomdiagram. Een stroomdiagram is een richting-aangevend diagram dat de loop beschrijft van de besturings uitvoer van het programma. Voor ons doel zullen we stroomdiagrammen beschouwen als drie types knooppunten, genaamd functie knooppunt, beslissings knooppunt en verzamelings knooppunt. De bovenste en onderste lijnen van beslissings knooppunten worden altijd verondersteld respectievelijk waar en niet-waar te zijn.



FUNKTIE



BESLISSING



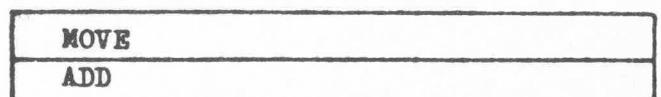
VERZAMELING

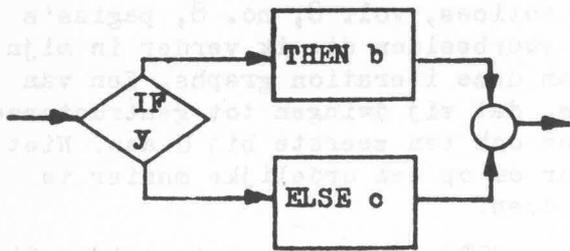
### Stroomdiagram



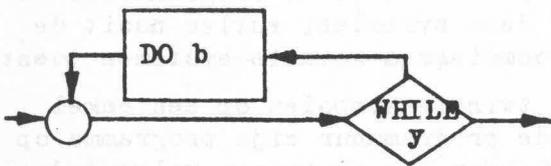
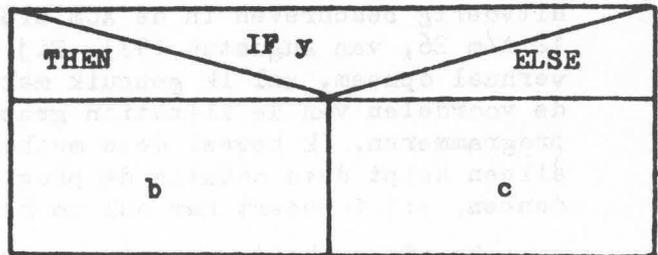
SEQUENCE

### Nassi/Schneiderman-diagram

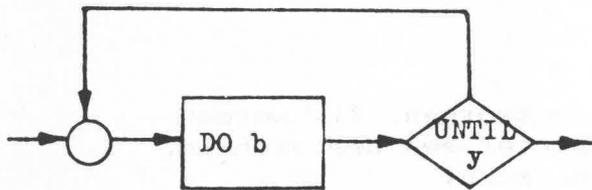
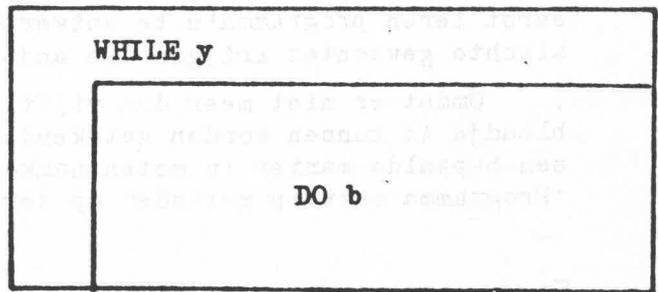




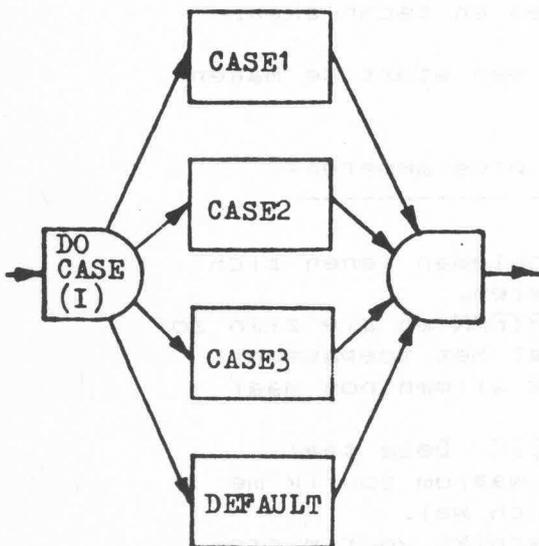
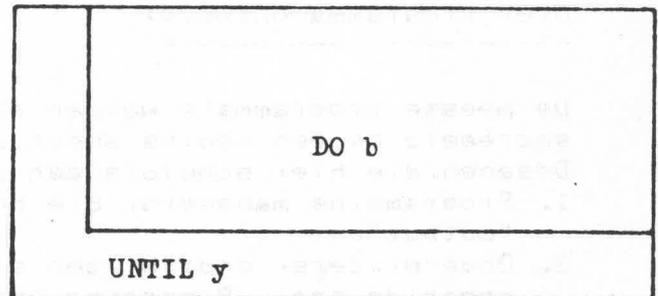
IFTHENELSE



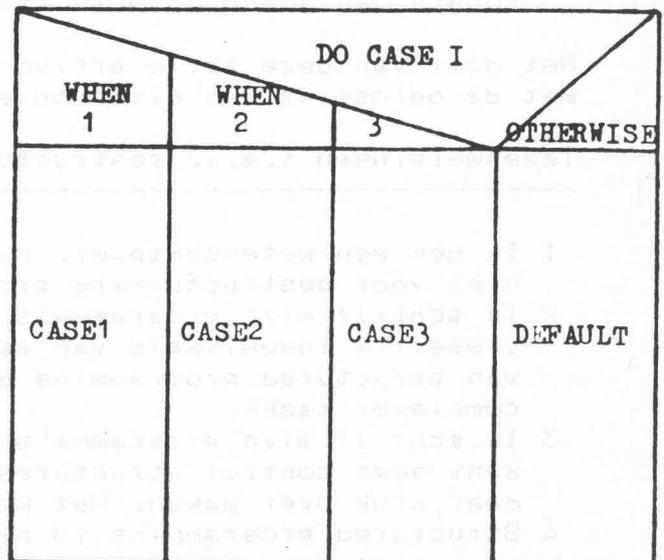
DOWHILE



DOUNTIL



CASE



De Nassi/Schneiderman diagrammen (ook wel "iteration graphs" genoemd) staan uitvoerig beschreven in de ACM SIGPLAN Notices, vol. 8, no. 8, pagina's 12 t/m 26, van Augustus 1973. Bij alle voorbeelden die ik verder in mijn verhaal opneem, zal ik gebruik maken van deze iteration graphs. Een van de voordelen van de iteration graphs is, dat zij dwingen tot gestructureerd programmeren. Ik beveel deze methode dan ook ten zeerste bij U aan. Niet alleen helpt deze notatie de programmeur om op een ordelijke manier te denken, zij forceert hem ook om het te doen.

De afwezigheid van enige vorm van de GOTO of branch statement heeft tot gevolg dat de programmeur zonder ze moet werken: een taak die in toenemende mate gemakkelijker wordt bij het praktisch doen. Programmeurs die eerst leren programma's te ontwerpen met deze symbolen, zullen nooit de slechte gewoontes krijgen die andere stroomdiagram-notatie-systemen toestaan.

Omdat er niet meer dan vijftien tot twintig symbolen op een enkel blaadje A4 kunnen worden getekend, moet de programmeur zijn programma op een bepaalde manier in moten hakken. Hoe? Daar kom ik in een volgend deel "Programma ontwerp methode" op terug.

Eerst nog even iets over het waarom en dan straks verder met het hoe.

Over programma ontwerp:

De meeste programma's worden niet ontworpen. Zij worden gecreeerd op een codings sheet of achter een beeldscherm. Degenen die hier schuldig aan zijn, zijn:

1. Programming managers, die hoofdzakelijk letten op de "output".
2. Onderwijzers, onderwijzen slechts het coderen in een bepaalde taal. Programma ontwerp wordt bijna altijd volledig buiten beschouwing gelaten.
3. Programmeurs. De meeste programmeurs zijn niet op de hoogte van goede ontwerp strategieën en technieken.

Het doel van deze serie artikelen is een start te maken met de oplossing van dit probleem.

Tegenwerpingen t.a.v. gestructureerd programmeren:

1. Ik ben een wetenschapper. Mijn problemen lenen zich niet voor gestructureerd programmeren.
2. Ik schrijf mijn programma's in FORTRAN en die zijn zo vreselijk ingewikkeld van aard, dat het toepassen van structured programming de zaak alleen nog maar complexer maakt.
3. Ik schrijf mijn programma's in BASIC. Deze taal kent geen control structures, dus waarom zou ik me daar druk over maken. Het werkt toch wel.
4. Structured programming is niet geschikt voor micro-computers.
5. Structured programming is alleen maar geschikt voor administratieve problemen.

## Tegenwerpingen t.a.v. commentaar en dokumentatie:

---

Vaak hoor ik de volgende tegenwerpingen t.a.v. dokumentatie:

- 1 "Ik heb geen tijd genoeg".
- 2 "Mijn programma is zelf-documenterend".
- 3 "Elke competente programmeur kan mijn coding begrijpen".
- 4 "Mijn programma is een eenmalig programma, dus is er geen dokumentatie nodig".
- 5 "Het programma zal tijdens het debuggen drastisch worden gewijzigd, zodat de dokumentatie waardeloos is geworden tegen de tijd dat het programma klaar is.
- 6 "Ik begrijp ontzettend goed wat mijn programma doet, dus waarom zou ik dokumentatie maken"?
- 7 "Ik houd niet van dokumenteren".
- 8 "Het is niet goed om teveel commentaar te hebben. Men ziet daardoor het belangrijke commentaar over het hoofd.
- 9 "Commentaar neemt alleen maar onnodig geheugenruimte in beslag en het vertalen duurt veel langer".
- 10 "Als het programma goed werkt leest toch niemand de dokumentatie".
- 11 "Ik ben een microcomputer hobbyist. Wat moet ik met al die gestructureerde onzin en commentaar in mijn programma's".
- 12 "Mijn labels en operanden zijn self-explaining, dus waarom zou ik commentaar toevoegen en dokumentatie maken? Bovendien heb ik daar toch geen ruimte voor, omdat mijn labels en operanden maximaal 31 posities beslaan".

Als U de volgende zaken probeert te voorkomen, dan komen we al een aardig eind in de goede richting:

- 1 Coding zonder commentaar. Dus liever kortere labels met commentaar dan lange labels zonder commentaar.
- 2 Gebruik van ongedefinieerde opcodes.
- 3 Onnodig gebruik van multi-tasking.
- 4 "Pervers" misbruik van de programmeertaal.
- 5 Programma's die zichzelf modificeren.
- 6 Het gemeenschappelijk gebruik van variabelen tussen meerdere modules.
- 7 Onnodig (veelvuldig) gebruik van GOTO's (unconditional JMP's). Gebruik ze alleen in uiterste noodzaak, om efficiency redenen, zoals bijvoorbeeld abnormal end situaties, error condities e.d. Overigens geldt ook hier: "In eerste instantie trekken we ons bij het programma ontwerp niets aan van efficiency. Het enige wat belangrijk is, is de programmastructuur". In tweede instantie, als de zaak te traag blijkt te zijn omdat we het bit dat we wilden lezen al voorbij is, gaan we optimaliseren! Eerder niet. Punt uit.
- 8 Complexe macro's in assembly language.
- 9 Niet-mnemonische variabelen.

## Programmeren in pseudo code:

Pseudo code is geen compileerbare taal. Pseudo code is een informele methode voor het weergeven van structured programming logic. Het lijkt een beetje op een programmeertaal (zoals PL/1 of PASCAL), maar is niet gebonden aan enige formele syntax regels. De enige regels van de pseudo code zijn die, welke betrekking hebben op de structured control statements en het inspringen, hetgeen de logica beter tot uitdrukking laat komen.

Het primaire doel van pseudo code is je in staat te stellen je ideeën uit te drukken op een natuurlijke manier, met gebruikmaking van normaal Nederlands (of normaal Engels).

Het idee wat er achter zit is, dat pseudo code je in staat stelt om je te concentreren op logische oplossingen van de problemen, in plaats van syntactische regels waarin e.e.a. uiteindelijk moet worden uitgedrukt.

Pseudo code heeft een programma ontwerp tot gevolg dat zeer goed leesbaar is en gemakkelijk kan worden geconverteerd naar uitvoerbare coding.

De syntactische regels zijn simpele gestructureerde woorden (die altijd worden geschreven in KAPITALEN), zoals:

IF, THEN, ELSE, DOWHILE, DOUNTIL, SELECT, CASE, OTHER, GET, PUT, READ, WRITE, CALL etc.

Even een voorbeeldje. Ontwerp een programma voor het teruggeven van wisselgeld door een automaat. Het terug te geven bedrag bedraagt nooit meer dan 99 cent.

```
wisselgeld := betaaldbedrag - verkoopprijs
```

```
DOUNTIL wisselgeld nul is
```

```
  IF wisselgeld niet minder dan 25 ct
```

```
    geef een kwartje
```

```
    trek 25 ct af van wisselgeld
```

```
  ELSE
```

```
    IF wisselgeld niet minder dan 10 ct
```

```
      geef een dubbeltje
```

```
      trek 10 ct af van wisselgeld
```

```
    ELSE
```

```
      IF wisselgeld niet minder dan 5 ct
```

```
        geef een stuiver
```

```
        trek 5 ct af van wisselgeld
```

```
      ELSE
```

```
        geef een cent
```

```
        trek 1 ct af van wisselgeld
```

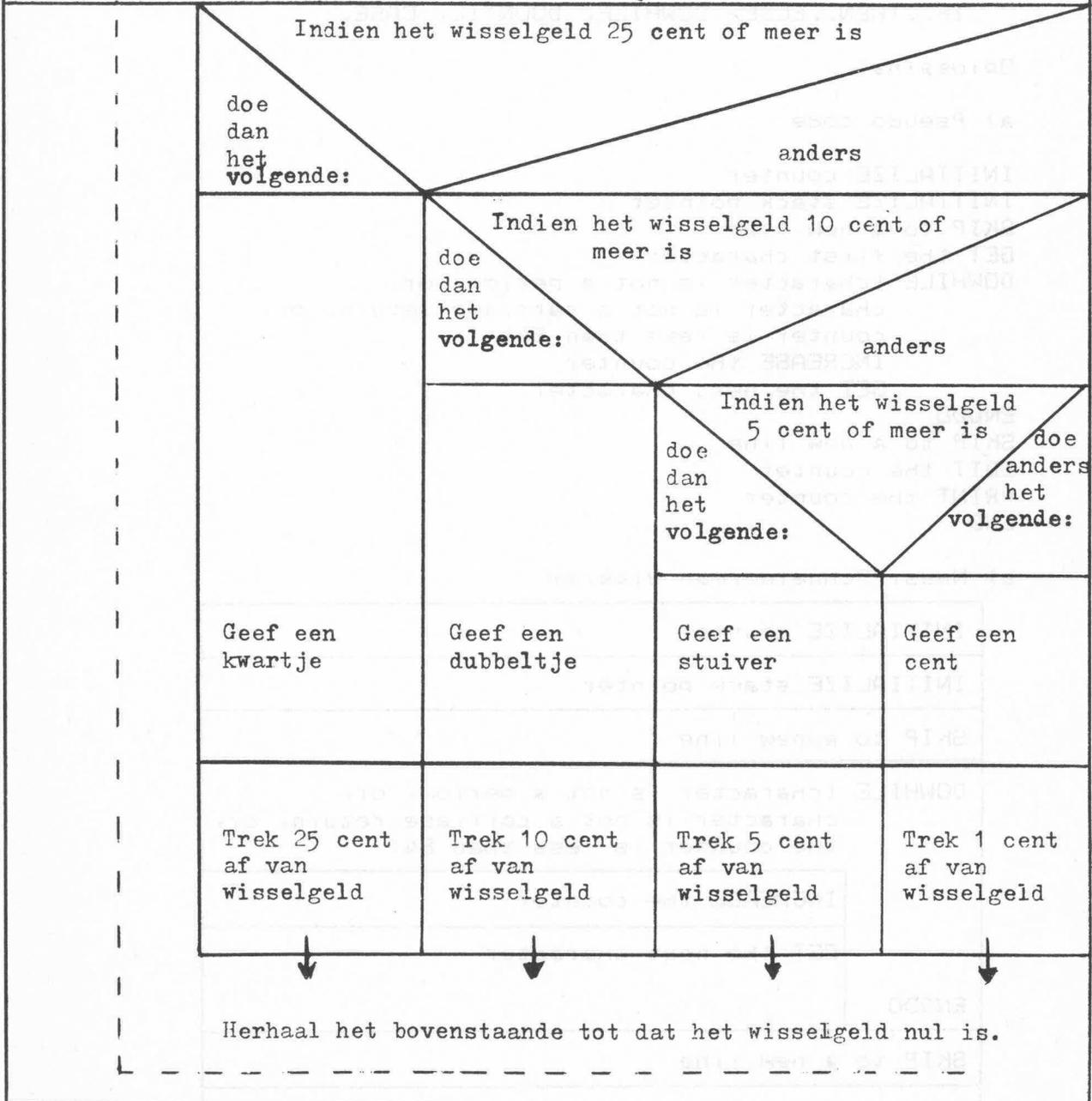
```
      ENDIF
```

```
    ENDIF
```

```
  ENDIF
```

```
ENDDO
```

Wisselgeld := Betaaldbedrag minus verkoopprijs



Nog een voorbeeld:

Gegeven: Een regel tekst van een TTY, maximaal 63 characters gevolgd door een carriage return of een punt.

Gevraagd: Tel het aantal characters en druk dit af op de TTY. Doe dit in:

- a) Pseudo code
- b) Nassi/Schneiderman diagram
- c) Micro ADE assembler zonder gebruik te maken van compare, branch en jump instructies, doch in plaats hiervan de eerder besproken control structures, zoals IF..THEN..ELSE, DOWHILE, DOUNTIL, CASE.

Oplossing:

- a) Pseudo code

```
INITIALIZE counter
INITIALIZE stack pointer
SKIP to a new line
GET the first character
DOWHILE (character is not a period, or,
        character is not a carriage-return, or,
        counter is less than 64)
    INCREASE the counter
    GET the next character
ENDDO
SKIP to a new line
EDIT the counter
PRINT the counter
END
```

- b) Nassi/Schneiderman diagram

INITIALIZE counter
INITIALIZE stack pointer
SKIP to a new line
DOWHILE (character is not a period, or, character is not a carriage return, or, the counter is less than 64)
INCREASE the counter
GET the next character
ENDDO
SKIP to a new line
EDIT the counter
PRINT the counter
END

```

0010: 0200          TTYCHR ORG   $0200
0020: 0200 00      COUNT   =     $00    COUNT FIELD
0030: 0201          KIMMON  *     $1C00  KIM MONITOR START ADDRESS
0040: 0201          CRLF    *     $1E2F  KIM MONITOR CARRIAGE RETURN
0050: 0201          GETCH   *     $1E5A  KIM MONITOR CHAR INPUT FROM TTY
0060: 0201          OUTCH   *     $1EA0  KIM MONITOR CHAR OUTPUT TO TTY
0070: 0201 D8      START   CLD        CLEAR DECIMAL MODE
0080: 0202 A9 00    LDAIM  $00    INITIALIZE
0090: 0204 8D 00 02 STA   COUNT  COUNT FIELD
0100: 0207 A2 FF    LDXIM $FF    INITIALIZE
0110: 0209 9A      TXS        STACK POINTER
0120: 020A 20 2F 1E JSR   CRLF   SKIP TO NEW LINE
0130: 020D 20 5A 1E JSR   GETCH  GET FIRST CHARACTER
0140:              DOWHILE (((A), NE, ', , IM), OR,
0150:              ((A), NE, $0D, IM), OR,
0160:              (COUNT, LT, $64, IM))
0170: 021F F8      SED        SET DECIMAL MODE
0180: 0220 AD 00 02 LDA   COUNT  FETCH COUNT
0190: 0223 69 01    ADCIM $01    INCREASE IT
0200: 0225 8D 00 02 STA   COUNT  BY ONE
0210: 0228 D8      CLD        CLEAR DECIMAL MODE
0220: 0229 20 5A 1E JSR   GETCH  GET NEXT CHARACTER
0230:              ENDDO
0240: 022E 20 2F 1E JSR   CRLF   SKIP TO NEW LINE
0250: 0231 AD 00 02 LDA   COUNT  GET COUNT FIELD
0260: 0234 4A      LSRA        SHIFT
0270: 0235 4A      LSRA        IT
0280: 0236 4A      LSRA        HALF A
0290: 0237 4A      LSRA        NIBBLE
0300: 0238 69 30    ADCIM $30    MAKE IT ASCII
0310: 023A 20 A0 1E JSR   OUTCH  AND PRINT IT
0320: 023D AD 00 02 LDA   COUNT  GET IT AGAIN
0330: 0240 29 0F    ANDIM $0F   RIGHT NIBBLE
0340: 0242 69 30    ADCIM $30    MAKE IT ASCII
0350: 0244 20 A0 1E JSR   OUTCH  AND PRINT IT
0360: 0247 00      BRK

```

```

0200 00D8A900 8D0002A2 FF9A202F 1E205A1E *..... /.. Z. *
0210 C92EF01A C90DF016 AD0002C9 64B00FF8 *..... D... *
0220 AD000269 018D0002 D8205A1E D0E2202F *... I..... Z.... /*
0230 1EAD0002 4A4A4A4A 693020A0 1EAD0002 *... JJJJIO..... *
0240 290F6930 20A01E00 00000000 00000000 *). IO..... *

```

Nog maar een voorbeeldje:

Gegeven: Temperaturen in graden Celsius verhouden zich tot graden Farenheit met de volgende formule:

$$C := 5/9*(F-32).$$

Gevraagd: Ontwerp een compleet programma dat in Celsius de overeenkomstige graden Farenheit afdruckt van F -40 tot F +120. Doe dit in:

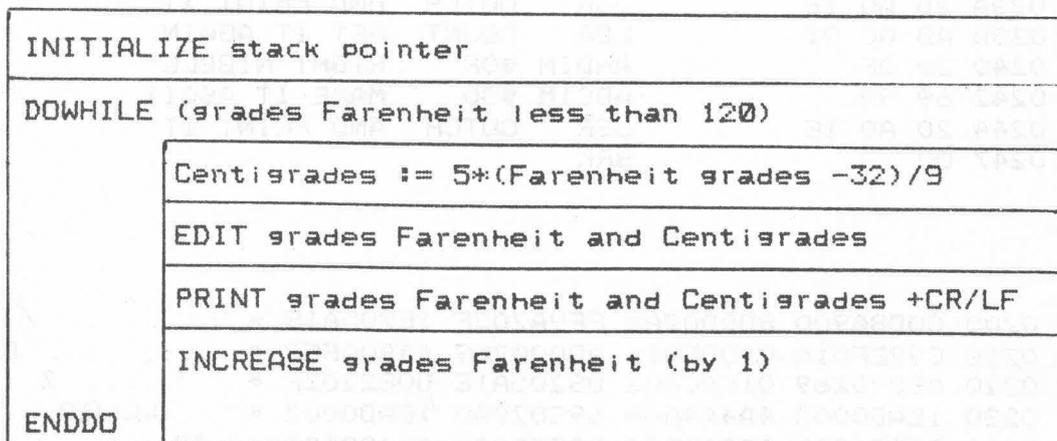
- Pseudo code
- Nassi/Schneiderman diagram

(Programmering van dit voorbeeld laat ik graag aan Uzelf over. Stuur Uw oplossing in aan het redaktiesekretariaat).

- Pseudo code

```
SET grades Farenheit to -40
INITIALIZE stack pointer
DOWHILE (grades Farenheit less than 120)
    Centigrades = 5*(Farenheit grades - 32) / 9
    EDIT grades Farenheit and Centigrades
    PRINT grades Farenheit and Centigrades + CR/LF
    INCREASE grades Farenheit (by 1)
ENDDO
```

- Nassi/Schneiderman diagram



Dan nu een voorbeeld hoe de CASE structuur gebruikt wordt. De CASE is eigenlijk overbodig. We zouden in plaats van de CASE net zo goed nested IF statements kunnen gebruiken. Dit wordt alleen wat lastig bij het inspringen, als we erg veel geneste IF's hebben. Voor de coding is dit bij de CASE geen bezwaar. We kunnen in de diepte net zoveel CASE's hebben als we willen.

Voor het Nassi/Schneiderman diagram hebben we bij de CASE hetzelfde probleem als bij geneste IF's, n.l. dat het papier bij een zekere hoeveelheid condities niet breed genoeg is. De heren I. Nassi en B. Schneiderman hebben dit probleem kennelijk gebasatelliseerd onder het motto: "Als je ontwerp niet op een standaard blaadje A4 (297 x 210 mm) past, dan wordt het te complex en onoverzichtelijk".

Echter is juist de CASE voor dit soort problemen uitgevonden. Je werkt dan namelijk niet meer in de breedte, maar in de diepte. Dan zou je de CASE's eigenlijk ook in de diepte moeten tekenen in de Nassi/Schneiderman diagrammen.

In een volgend artikel kom ik daar wel eens op terug. Voor de geïnteresseerden heb ik de oplossing voor handen. Geef maar even een belletje.

Dan nu het voorbeeld voor een CASE:

Gegeven: Lees een character vanaf de teletype o.i.d.  
Store dit character in een lokatie genaamd SYSTEM.  
Het character kan zijn:

- 1 = JUNIOR
- 2 = KIM
- 3 = PET
- 4 = CBM
- 5 = VIC 20

Gevraagd: Ontwerp een routine die een character vanaf de teletype o.i.d. leest. Dit character opslaat in een lokatie genaamd SYSTEM. Als het een 1 is de JUNIOR routine uitvoeren, als het een 2 is de KIM routine uitvoeren, als het een 3, 4 of 5 is de COMMODORE routine uitvoeren of als het niet een 1, 2, 3, 4 of 5 is, de ERROR routine uitvoeren.

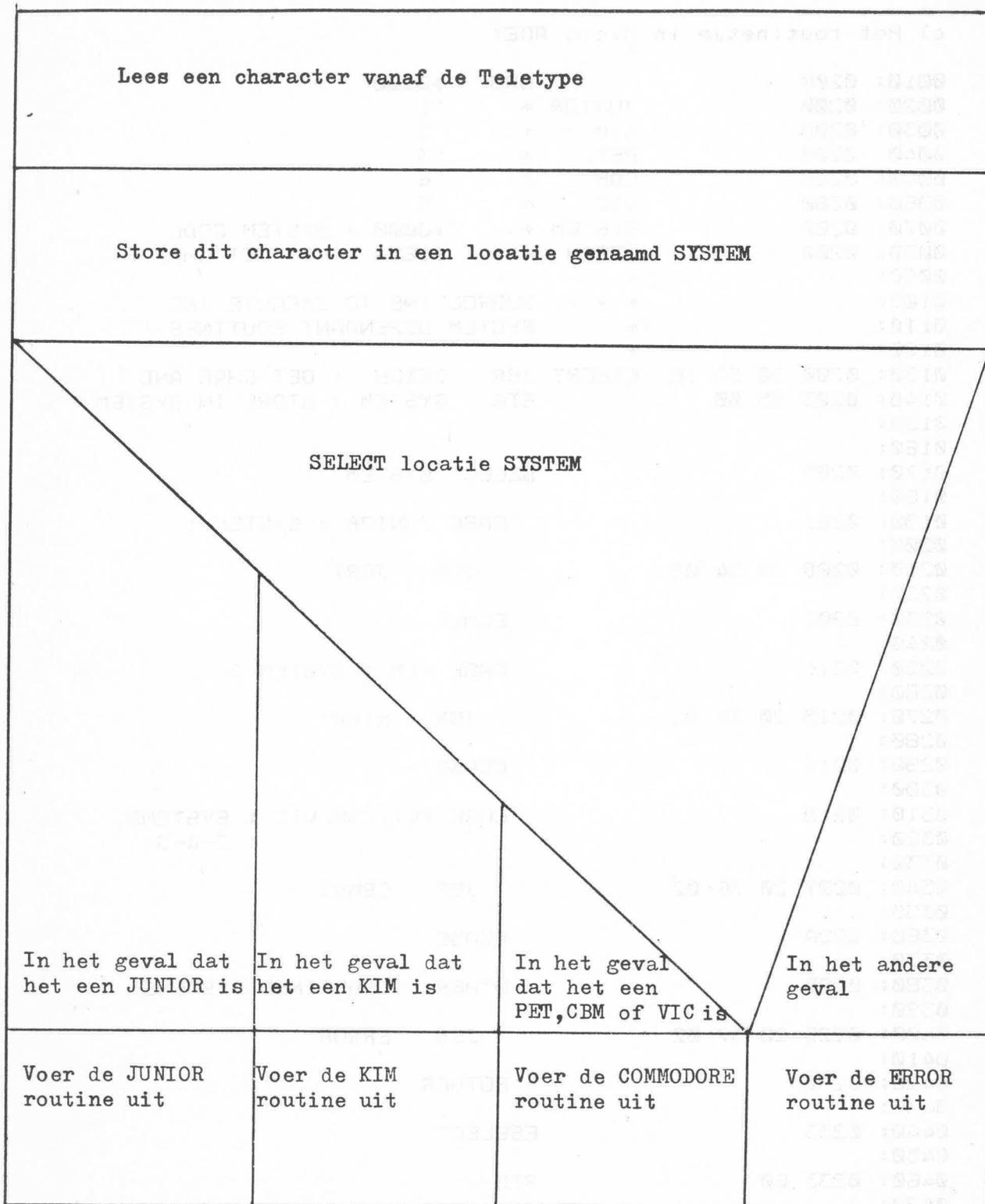
Geef de oplossing in:

- a) Pseudo code
- b) Nassi/Schneiderman diagram
- c) Assembly language met virtuele structured programming macro's.

a) Pseudo code

```
GET a character from the TTY
STORE the character in location SYSTEM
SELECT (SYSTEM)
    CASE JUNIOR
        CALL Junior routine
    ECASE
    CASE KIM
        CALL Kim routine
    ECASE
    CASE PET,CBM,VIC
        CALL Commodore routine
    ECASE
    OTHER
        CALL error routine
    EOTHER
ESELECT
RETURN
```

b) Het Nassi/Schneiderman diagram



## c) Het routinetje in Micro ADE:

```

0010: 0200
0020: 0200
0030: 0200
0040: 0200
0050: 0200
0060: 0200
0070: 0200
0080: 0200
0090:
0100:
0110:
0120:
0130: 0200 20 5A 1E EXECRT JSR GETCH ; GET CHAR AND
0140: 0203 85 00 STA SYSTEM ; STORE IN SYSTEM
0150:
0160:
0170: 0205 SELECT SYSTEM
0180:
0190: 0207 CASE JUNIOR ; SYSTEM 1
0200:
0210: 020B 20 34 02 JSR JCRT
0220:
0230: 020E ECASE
0240:
0250: 0211 CASE KIM ; SYSTEM 2
0260:
0270: 0215 20 35 02 JSR KIMRT
0280:
0290: 0218 ECASE
0300:
0310: 021B CASE PET,CBM,VIC ; SYSTEMS
0320: ; 3-4-5
0330:
0340: 0227 20 36 02 JSR CBMRT
0350:
0360: 022A ECASE
0370:
0380: 022D OTHER ; UNDEFINED SYSTEMS
0390:
0400: 022D 20 37 02 JSR ERROR
0410:
0420: 0230 EOTHER
0430:
0440: 0233 ESELECT
0450:
0460: 0233 60 RTS
0470:
0480: 0234 60 JCRT RTS
0490: 0235 60 KIMRT RTS
0500: 0236 60 CBMRT RTS
0510: 0237 60 ERROR RTS

```

Om te kunnen zien wat voor coding de macro's hebben gegenereerd, laat ik hieronder ook nog even dezelfde listing in geëxpandeerde vorm zien. De expansie van de macro's wordt weergegeven door het ontbreken van een regelnummer en de aanwezigheid van een '+' teken voorafgaande aan het adres-veld.

```

0010: 0200                                ORG    $0200
0020: 0200                JUNIOR *    '1
0030: 0200                KIM    *    '2
0040: 0200                PET    *    '3
0050: 0200                CBM    *    '4
0060: 0200                VIC    *    '5
0070: 0200                SYSTEM *    $0000 ; SYSTEM CODE
0080: 0200                GETCH *    $1E5A ; KIM GET CHAR RT
0090:
0100:                * * * SUBROUTINE TO EXECUTE THE
0110:                * SYSTEM DEPENDANT ROUTINES
0120:                *
0130: 0200 20 5A 1E EXECRT JSR    GETCH ; GET CHAR AND
0140: 0203 85 00                STA    SYSTEM ; STORE IN SYSTEM
0150:
0160:
0170: 0205                                SELECT SYSTEM
      +0205 A5 00                LDA    SYSTEM
0180:
0190: 0207                                CASE JUNIOR ; SYSTEM 1
      +0207 C9 31                L0002A CMPIM JUNIOR
      +0209 D0 0E                BNE   L0004A
0200:
0210: 020B 20 34 02                JSR    JCRT
0220:
0230: 020E                                ECASE
      +020E 4C 33 02            JMP    L0010A ; JMP TO END

0240:
0250: 0211                                CASE KIM ; SYSTEM 2
      +0211 C9 32                L0004A CMPIM KIM
      +0213 D0 0E                BNE   L0006A
0260:
0270: 0215 20 35 02                JSR    KIMRT
0280:
0290: 0218                                ECASE
      +0218 4C 33 02            JMP    L0010A ; JMP TO END
0300:
0310: 021B                                CASE PET,CBM,VIC ; SYSTEMS
0320:                ; 3-4-5
0330: +021B C9 33                L0006A CMPIM PET
      +021D F0 08                BEQ   L0006B
      +021F C9 34                CMPIM CBM
      +0221 F0 04                BEQ   L0006B
      +0223 C9 35                CMPIM VIC
      +0225 D0 0E                BNE   L0008A

```

```

+0227          L0006B  *
0330:
0340: 0227 20 36 02          JSR   CBMRT
0350:
0360: 022A          ECASE
+022A 4C 33 02          JMP   L0010A ; JMP TO END
0370:
0380: 022D          OTHER ; UNDEFINED SYSTEMS
0380: +022D          L0008A  *
0390:
0400: 022D 20 37 02          JSR   ERROR
0410:
0420: 0230          EOTHER
+0230 4C 33 02          JMP   L0010A ; JMP TO END
0430:
0440: 0233          ESELECT
+0233          L0010A  *
0450:
0460: 0233 60          RTS
0470:
0480: 0234 60          JCRT   RTS
0490: 0235 60          KIMRT  RTS
0500: 0236 60          CBMRT  RTS
0510: 0237 60          ERROR  RTS

```

Het nut van dit programmaatje zal U wel niet helemaal duidelijk zijn. Mij ook niet. Het enige nut is het aantonen van de implementatie van de CASE structuur in assembly language. Meer niet.

Copyright (C) 1982, by Anton Muller, Sinj. Semeynsstr. 78 Amsterdam (The Netherlands).

No part of this article may be reproduced in any form, by print, photoprint, microfilm or any other means without written permission from the author.

Niets uit deze uitgave mag worden verveelvoudigd en/of openbaar gemaakt door middel van druk, fotocopie, microfilm of op welke andere wijze ook, zonder voorafgaande schriftelijke toestemming van de schrijver.

#### Referenties:

- (1) Nassi, I. en Schneiderman, B. "Flowchart techniques for structured programming". ACM SIGPLAN notices, vol. 8, no 8, (Aug. 1973), p. 12-26.

Programmeerwedstrijd.

Samengesteld door: Anton Muller  
Sinjeur Semeynsstr. 78 I  
1061 GM AMSTERDAM

met dank aan Hans Otten voor het lezen van het script en het aanbrensen van de nodige correcties.

Eigenlijk is deze programmeerwedstrijd ontstaan uit mijn vervolg artikel over structured programming (deel 2), hetwelk elders in deze 6502 Kenner is opgenomen. In dat artikel had ik oorspronkelijk twee opgaves opgenomen, voor desenen die een beetje wilden oefenen met de daarin behandelde stof en de stof over hetzelfde onderwerp in KIM Kenner 5, alwaar werden behandeld de control-structures sequence, if...then...else, do while en do until. Tezamen met de case structuur die in mijn artikel over structured programming in deze 6502 Kenner wordt behandeld, heeft U nu een compleet beeld over hoe we gestructureerd kunnen programmeren in assembleer taal. De opgaves die ik samengesteld had in het structured programming artikel, waren die van de vermenigvuldigings en delings routines, zoals deze thans in deze programmeerwedstrijd zijn opgenomen. Het geheel werd echter dusdanig omvangrijk, dat ik heb besloten er maar een apart artikel van te maken en dan wat uitgebreider, met ook optellen, aftrekken en vergelijken van decimale getallen.

De bedoeling van deze programmeerwedstrijd, is dus eigenlijk een oefening in het gestructureerd programmeren, met als resultaat dat het een rekenpakket oplevert, dat kan:

- 1e) vermenigvuldigen
- 2e) delen
- 3e) optellen
- 4e) aftrekken
- 5e) vergelijken
- 6e) pakken van zoned decimale getallen
- 7e) unpacken van packed getallen

Elke functie moet afzonderlijk als een subroutine aan te roepen zijn, waarbij de gegevens (ook wel genoemd parameters) die de subroutines nodig hebben, zoals de lengtes van de getallen waar we mee werken en de begin adressen van die getallen (ook wel genoemd operanden), op de stack staan. Hierover later meer.

Ik gebruik in dit verhaal nogal wat vakjargon en daarom lijkt het mij nuttig eerst maar een paar definities te geven van kreten die U misschien afschrikken, als U niet weet wat ze betekenen.

Het is voor mij echter ondoenlijk om een dergelijk verhaal als dit, in het normale algemeen beschaafd Nederlands te schrijven, omdat ik met vaktermen zit, die eenvoudig weg niet in het Nederlands te vertalen zijn, omdat de Nederlandse taal daar geen woorden voor heeft, doch slechts zeer uitgebreide omschrijvingen van die begrippen.

Zo betekent het woord nibble, een halve byte, of te wel vier bits uit een 8 bits byte. We kennen een linker nibble en een rechter nibble. Het woord bit kende U waarschijnlijk al. Dat is een afkorting van binary digit, of te wel een geheugen cel, die de waarde 0 of 1 kan bevatten. Een digit is een cijfer uit een talstelsel. Zo kennen we binaire digits 0 en 1, decimale digits 0 t/m 9, hexadecimale digits 0 t/m F etc.

Het woord sign, betekent teken, hetgeen plus of min kan zijn en een bepaalde afgesproken codering heeft.

Een integer is een geheel getal.

Low order betekent laagste waarde, high order betekent hoogste waarde. Least significant betekent meest rechtse, most significant betekent meest linkse. (Eigenlijk betekent het minst waardevolle en meest waardevolle).

Het least significant bit in een binair getal van 8 bits, is het meest rechtse bit. Het most significant bit is in dat geval het meest linkse bit.

Het least significant digit in een getal van meerdere cijfers, is het meest rechtse. Het most significant digit is in dat geval het meest linkse.

Als we het hebben over de object code, dan is dat het resultaat van het omzetten van source coding in machine taal, hetgeen we normaliter met een assembler doen.

Source coding is de invoer voor de assembler, waarbij we de instructies schrijven in de vorm van LDA, STA, CMP, etc. en labels gebruiken in plaats van de werkelijke geheugen adressen etc.

Een hex dump is een weergave van de geheugen inhoud op een bepaald medium (papier, cassette, beeldscherm e.d.)

ASCII betekent American Standard Code for Information Interchange (uit te spreken als ASKEY, waarbij

AS wordt uitgesproken als as van verbrande turf en KEY als het Engelse woord voor sleutel).

Hex betekent hexa decimaal of te wel het zestien tallige stelsel bestaande uit de digits 0 t/m 9 en A t/m F.

In de opgaves wordt de ene keer gesproken over operand, de andere keer over veld en weer een andere keer over getal. In al deze gevallen wordt hetzelfde bedoeld, n.l. een stukje geheugen waar wat in staat en waar we wat mee gaan doen in een van de routines. Ook hebben we het over getal (A) of operand A. Hiermee wordt bedoeld het linker operand en met (B) het rechter operand.

Tenslotte hebben we nog het woord parameter of parameters of parameter list. Hiermee wordt bedoeld een of meer waarden, die subroutines nodig hebben om te kunnen werken, zoals de lengtes van de operanden (is een parameter)

het begin adres van het (A) operand (is een parameter) en het begin adres van het (B) operand (is ook een parameter) en al deze parameters tezamen noemen we een parameterlist.

Tot zover de moeilijke woorden. U ziet, dat het met enige uitleg best meevalt. Of niet? Dan gaan we nu uitlegen wat er verder moet gebeuren.

De bedoeling is, dat U deze routines ontwerpt en uitwerkt, in:

- a) Pseudo code.....en
- b) Nassi/Schneiderman diagram.....en
- c) Mos Technology assembler, of  
Micro ADE assembler, of  
Carl Moser's assembler (ASM/TED),  
in de stijl die in mijn artikelen over structured programming (deel 1) in KIM Kenner 5 en (deel 2) in de 6502 Kenner 20 wordt gebruikt.  
Zij die niet de beschikking hebben over een assembler, kunnen de routines schrijven in de syntax van een van deze assemblers en daarna de routines met de hand coderen naar machine taal.

Stuur bij Uw oplossingen de volgende zaken op:

- a) De source coding in machine leesbare vorm, op cassette, floppy disk, papertape o.i.d., als U over een assembler beschikt. Is dat niet het geval dan kunt U dit achterwege laten.
- b) De object code in machine leesbare vorm, op cassette, floppy disk, papertape o.i.d.
- c) De source listing, hetzij afgedrukt op een regeldrukker, hetzij met de hand geschreven en vertaald in machine taal.
- d) Een hex dump van de object code op papier, hetzij afgedrukt op een regeldrukker, hetzij met de hand geschreven.
- e) Een listing van Uw test set
- f) Een listing van de testresultaten
- g) Een listing van Uw test-driver
- h) Programma ontwerp in Pseudo code
- i) Programma ontwerp in Nassi/Schneiderman diagram.

Stuur Uw oplossingen in voor 23 augustus 1982, naar het verenigings sekretariaat, p/a Voorburgpad 10, 6843 EM ARNHEM. Oplossingen, ontvangen op of na 23 augustus 1982, met een datum postmerk die op of na 23 augustus 1982 list, worden niet meer in behandeling genomen.

U kunt mee doen in een van de twee categorieën:

**Categorie I:** Is voor desenen die het volledige rekenpakket maken.

Voor deze categorie stellen wij drie prijzen beschikbaar, t.w.:

1e Prijs een cadeaubon ter waarde van honderd gulden

2e Prijs een cadeaubon ter waarde van vijftig gulden

3e Prijs een cadeaubon ter waarde van vijftientig gulden

**Categorie II:** Is voor desenen die slechts een deel van het rekenpakket maken.

Voor deze categorie stellen wij een hoofdprijs ter beschikking in de vorm van een cadeaubon ter waarde van vijftientig gulden.

Uw inzendingen worden beoordeeld door een deskundige commissie, bestaande uit de volgende leden:

Anton Muller  
Tom Offringa  
Hans Otten  
Ruud Uphoff  
Siep de Vries  
Sebo Woldringh

Uitgesloten van deelname aan deze wedstrijd, zijn: de leden van de commissie, de bestuursleden en de leden van de redactie, alsmede hun naaste familieleden. Tevens zijn uitgesloten die personen, die per 23 augustus 1982 geen lid (meer) zijn van de KIM gebruikers club.

Bij de beoordeling wordt gelet op:

- a) originaliteit
- b) structuur
- c) netheid
- d) volledigheid
- e) betrouwbaarheid
- f) snelheid
- g) geheugenruimte
- h) onderhoudbaarheid
- i) robuustheid
- j) commentaar
- k) dokumentatie
- l) test in- en uitvoer
- m) de gebruikte algoritmes (met literatuur verwijzing)

De winnaars worden bekend gemaakt in het december nummer van 1982. Over de uitslag valt niet te corresponderen. Door het inzenden van Uw oplossingen verleent U tevens toestemming aan de vereniging, dat deze gepubliceerd kunnen worden in de 6502 Kenner. Verder behoudt U zelf het copyright.

Decimale getallen formaten: (Ref. 1)

Decimale getallen kunnen in zoned en/of packed formaat voorkomen. Beide decimale-getallen formaten bestaan uit minimaal 1 en maximaal 16 bytes, bestaande uit een paar van vier-bit codes. De vier-bit codes bestaan uit decimale digit codes, sign codes en een zone code.

Zoned formaat:

```
+-----+-----+-----+-----+
! Z ! N ! Z ! N !...! Z ! N ! S ! N !
+-----+-----+-----+-----+
! byte 0! byte 1!...! byte 14! byte 15!
+-----+-----+-----+-----+
```

In het zoned formaat, worden de meest rechtse vier bits van een byte de numerieke bits genoemd (N) en bestaan normaler wijze uit een code die een decimale digit weergeeft. De meest linkse vier bits van een byte, worden de zone bits (Z) genoemd, met uitzondering van de meest rechtse byte van een decimaal operand, waarbij deze bits de sign bits (S) worden genoemd. In alle bytes behalve de meest rechtse, staat dus de normale ASCII code voor de cijfers 0 t/m 9. Hierover later meer.

Decimale digits in het zoned formaat, zijn onderdeel van een grotere character set, die ook alfabetische en speciale characters bevat. Het zoned formaat is daarom geschikt voor input, editing en output van numerieke data in menselijk-leesbare vorm. Het rekenpakket werkt niet direkt met de decimale getallen in het zoned formaat; deze getallen moeten eerst worden omgezet naar het packed formaat.

Packed formaat:

```
+-----+-----+-----+-----+
! D ! D ! D ! D ! . ! D ! D ! D ! S !
+-----+-----+-----+-----+
!byte 0!byte 1!...!byte 14!byte 15!
+-----+-----+-----+-----+
```

In het packed formaat, bevat iedere byte twee decimale digits (D), behalve de meest rechtse byte, die naast het digit in de linker nibble, in de rechter nibble een sign bevat. (Een nibble is een halve byte (vier bits)). Ons decimale rekenpakket werkt met operanden in het packed formaat en genereert resultaten in het packed formaat.

#### Decimale codes:

-----

De decimale digits 0-9 hebben de binaire codering van 0000-1001.

De sign codes of te wel de codes voor de plus en min tekens, zijn respectievelijk 1100 voor de plus en 1101 voor de min.

Dit zijn ook de sign codes welke gegenereerd worden voor de resultaten van de rekenkundige bewerkingen.

De zone code 0011 komt voor in de linker vier bit posities van iedere byte die een decimale digit weer-geeft in zoned formaat resultaten, met uitzondering van de meest rechtse byte, die in deze posities de sign code heeft (1100 of 1101).

Zoned formaat resultaten worden geproduceerd door de UNPACK routine, met uitzondering van de linker vier bit posities van de meest rechtse byte, welke ongecontroleerd worden unpacked.

De rechter vier bits van iedere byte in het zoned formaat bevatten allen een decimale digit code.

Indien een ongeldige code wordt ontdekt in hetzij de digits, hetzij de tekens, kunnen we niet verder. Derhalve plaatsen we een code hex. 01 in de accumulator en breken we de betreffende routine af met een BRK instructie en laten het verder over aan degene die voor deze rommel heeft gezorgd, om uit te zoeken wat er aan de hand is, hetgeen we al min of meer aangeven met onze code hex. 01 in de accumulator.

#### Overzicht zoned cijfers:

30 = 0, 31 = 1, 32 = 2, 33 = 3, 34 = 4

35 = 5, 36 = 6, 37 = 7, 38 = 8, 39 = 9

(Dit zijn de cijfers in normale ASCII notatie).

C0 = +0, C1 = +1, C2 = +2, C3 = +3, C4 = +4

C5 = +5, C6 = +6, C7 = +7, C8 = +8, C9 = +9

(Deze cijfers kunnen dus voorkomen in de eenheden-positie van zoned decimale getallen en hebben dezelfde waarde als de normale zoned decimale cijfers (30 - 39) zoals hiervoor omschreven).

D0 = -0, D1 = -1, D2 = -2, D3 = -3, D4 = -4  
D5 = -5, D6 = -6, D7 = -7, D8 = -8, D9 = -9

Deze cijfers kunnen dus voorkomen in de eenhedenpositie van zoned decimale getallen en hebben dezelfde waarde als de normale zoned decimale cijfers (30 - 39) zoals hiervoor omschreven, echter is het zoned decimale getal in dit geval negatief).

## Decimale rekenkundige instructies:

De decimale rekenkundige instructies bestaan uit optellen, aftrekken, vermenigvuldigen, delen en vergelijken.

Operanden van de decimale rekenkundige instructies, zijn in het packed formaat en worden beschouwd als signed decimale integers. Een decimaal integer wordt weergegeven in het werkelijke formaat als een absolute waarde met een apart plus of min teken. Het bestaat uit een oneven aantal decimale digits, van 1 tot en met 31 en het teken; hetse correspondeert met een operand lengte van 1 tot en met 16 bytes.

Een decimale nul heeft normaal een plus teken, doch door vermenigvuldigen en delen, kan dit ook een min teken worden. Zorg er daarom altijd voor, dat als het resultaat van een bewerking nul is, het teken altijd een plus teken is.

De lengtes van de twee operanden, die gespecificeerd zijn in de lengte parameter, hoeven niet gelijk te zijn. Indien nodig, wordt het kortere operand beschouwd als zijnde uitgebreid met nullen links van de high-order digit. De resultaten kunnen echter nooit groter zijn dan de lengte van het eerste operand.

Indien een carry of een high-order nonzero digit van het resultaat verloren gaat, omdat het eerste operand te klein is, om het resultaat te bevatten, wordt de code hex. 02 in de accumulator gezet en de betreffende routine afgebroken d.m.v. een BRK instructie, als zijnde een zgn. decimal overflow. De lengtes van de operanden op zich zijn geen indicatie voor een overflow. Significante digits moeten verloren zijn gegaan tijdens een bewerking, wil er sprake zijn van een decimal overflow.

Parameter list:

De parameters voor de rekenkundige bewerkingen staan op de stack en zien er als volgt uit:

```
+-----+-----+-----+-----+
! La  Lb !   A1   !   A2   !   B1   !   B2   !
+-----+-----+-----+-----+
```

waarbij La de lengte van het eerste operand (A) is,  
Lb de lengte van het tweede operand (B) is,  
A1 low-order deel van operand A is,  
A2 high-order deel van operand A is,  
B1 low order deel van operand B is en  
B2 high-order deel van operand B is.

De opgegeven lengtes, zijn de lengtes min 1, dus lengte nul bestaat niet, lengte een wordt 0, lengte zestien wordt F.

Als u met een PLA de parameters van de stack haalt, krijgt u dus als eerste B2 binnen en als laatste de lengtebyte La-Lb.

Het eerste probleem: vermenigvuldigen.

Gegeven:

Een geheel getal A en een geheel getal B, beiden in packed formaat.

Gevraagd:

Ontwerp een routine die het getal A met het getal B vermenigvuldigt en het produkt plaatst in A, rekeninghoudend met de lengtes van de operanden en het maximale resultaat en rekening houdend met de tekens (positief of negatief. Nul is positief!).

Na uitvoering van de vermenigvuldiging wordt de status byte als volgt gezet:

	N	Z
	-	-
Resultaat is negatief	set	reset
Resultaat is nul	reset	set
Resultaat is positief	reset	reset

Voorbeeld:

Operand A bevat een negatieve waarde 38460, opgeslagen in 5 bytes, en wel als volgt:

```
00 00 38 46 0D
```

Operand B bevat eveneens een negatieve waarde 321, opgeslagen in 2 bytes, en wel als volgt:

```
32 1D
```

Na uitvoering van de vermenigvuldigings routine bevat A het produkt, dat er dan als volgt uit ziet:

```
01 23 45 66 0C
```

De status byte wordt als volgt gezet:

```
N=0, Z=0.
```

Een simpele vermenigvuldigings procedure:

Een procedure voor vermenigvuldigen die daadwerkelijk werd gebruikt door Egyptische wiskundigen in het jaar 1800 voor Christus, is als volgt gedefinieerd: (Ref.2)

```
A := vermenigvuldigtal
B := vermenigvuldiger
P := 0 (* tussenresultaat van het produkt *)
WHILE (B <> 0) DO
  IF (B = oneven) THEN
    P := P + A
  END
  A := A + A (* verdubbel A *)
  B := (B gedeeld door 2)
ENDDO
Bepaal het sign van het produkt (P)
A := P
```

Dit is een praktische methode voor vermenigvuldigen, omdat ze alleen simpele operaties nodig heeft, zoals verdubbelen, halveren en optellen. En dat kan onze 6502 uitstekend.

Deze methode wordt vaak de "Russische boeren methode" van vermenigvuldigen genoemd, omdat Westerse bezoekers aan Rusland in de negentiende eeuw deze methode daar op grote schaal in gebruik vonden.

(Uiteraard bent U vrij om te kiezen of U dit algoritme al dan niet wenst te gebruiken. Ik geef het hier alleen maar als voorbeeld, hoe het zou kunnen).

Het tweede probleem: Delen.

Gegeven:

Een geheel getal A en een geheel getal B, beiden in packed formaat.

Gevraagd:

Ontwerp een routine die het getal A deelt door B en het quotient en de rest van de deling plaatst in A. Zowel de operanden als het resultaat zijn in packed decimale vorm. Het aantal bytes van het quotient is gelijk aan het verschil tussen de lengte van het deeltal (A) en de deler (B). De rest heeft dezelfde lengte als de deler. Het teken van het quotient wordt vastgesteld door de regels van het Algebra los te laten op de tekens van het deeltal en de deler. Het teken van de rest van de deling heeft dezelfde waarde als die van het deeltal.

Indien het quotient 0 is of de rest 0 is, is het teken altijd positief.

Indien de deler 0 is dan hex 03 in de accumulator plaatsen (ook bij nul gedeeld door nul) en de routine afbreken m.b.v. een BRK instructie.

Na uitvoering van de deling wordt de status byte als volgt gezet:

	N	Z
Resultaat is negatief	set	reset
Resultaat is nul	reset	set
Resultaat is positief	reset	reset

Voorbeeld:

Operand A bevat een positief getal 12345678, dat als volgt is opgeslagen:

01 23 45 67 8C

Operand B bevat een negatief getal 321, dat als volgt is opgeslagen:

32 1D

Na uitvoering van de delings routine, bevat A het quotient en de rest, dat er als volgt uit ziet

38 4E 0D 01 8C

De status byte wordt als volgt gezet:

N=0, Z=0.

Het derde probleem: optellen.

-----  
Gegeven:

-----  
Een geheel getal A en een geheel getal B.

Gevraagd:

-----  
Ontwerp een routine die getal B optelt bij getal A en het resultaat plaatst in A, rekening houdend met de lengtes van de operanden en het maximale resultaat en rekening houdend met de tekens (positief of negatief. Nul is positief!).

Na uitvoering van de optellings wordt de status byte als volgt gezet:

	N	Z
Resultaat is negatief	set	reset
Resultaat is nul	reset	set
Resultaat is positief	reset	reset

Voorbeeld:

-----  
Operand A bevat een negatieve waarde 38460, opgeslagen in 3 bytes, en wel als volgt:

38 4E 0D

Operand B bevat een positieve waarde 112345, opgeslagen in 4 bytes, en wel als volgt:

01 12 34 5C

Na uitvoering van de optel routine bevat A de som, die er als volgt uit ziet:

73 88 5C

De status byte wordt als volgt gezet:

N=0, Z=0.

Het vierde probleem: aftrekken.

-----

Geseeven:

-----

Een geheel getal A en een geheel getal B.

Gevraagd:

-----

Ontwerp een routine die het getal B aftrekt van het getal A en het verschil plaatst in A, rekening houdend met de lengtes van de operanden en het maximale resultaat en rekening houdend met de tekens (positief of negatief. Nul is positief!).

Na uitvoering van de aftrekking wordt de status byte als volgt gezet:

	N	Z
Resultaat is negatief	set	reset
Resultaat is nul	reset	set
Resultaat is positief	reset	reset

Voorbeeld:

-----

Operand A bevat een positieve waarde 73885, opgeslagen in 3 bytes, en wel als volgt:

73 88 5C

Operand B bevat een positieve waarde 112345, opgeslagen in 4 bytes, en wel als volgt:

01 12 34 5C

Na uitvoering van de aftrekkings routine bevat A het verschil, dat er dan als volgt uit ziet:

38 46 0D

De status byte wordt als volgt gezet:

N=1, Z=0.

Het vijfde probleem: vergelijken van twee getallen.

-----

Het eerste operand wordt vergeleken met het tweede operand, en het resultaat wordt weergegeven in de status byte. De operanden zijn in het packed formaat.

Vergelijking is algebraïsch en volgt de procedure van decimale aftrekking, met dat verschil, dat beide operanden ongewijzigd blijven. Indien het verschil nul is, zijn de operanden gelijk. Indien een niet-nul verschil positief of negatief is, is het eerste operand respectievelijk hoog of laag.

Overflow kan niet voor komen, omdat het verschil verder buiten beschouwing wordt gelaten.

Na uitvoering van de vergelijkingsroutine wordt de status byte als volgt gezet:

	N	C	Z
1e operand < 2e operand	afh. inh.	reset	reset
1e operand = 2e operand	reset	set	set
1e operand > 2e operand	afh. inh.	set	reset

Voorbeeld:

-----

A bevat een negatieve waarde 172535 die er in packed formaat als volgt uit ziet:

17 25 35 6D

B bevat eveneens een negatieve waarde, 72142, die er in packed formaat als volgt uit ziet:

72 14 2D

Na uitvoering van de vergelijkingsroutine wordt de status byte als volgt gezet:

N=1, C=0, Z=0, hetgeen er op duidt dat het eerste operand negatief is en kleiner is dan het tweede operand.

Dus, om te controleren of het eerste operand kleiner is dan het tweede operand, wordt de vergelijkingsroutine call gevolgd door een BCC; om te controleren of ze gelijk zijn door een BEQ; en om te controleren of het eerste operand groter is dan het tweede operand, wordt de call gevolgd door een BEQ en een BCS.

1e operand groter dan of gelijk aan 2e operand wordt gecontroleerd door een BCS.

(Zie de overeenkomst op pagina 46 van het 6502 programming manual van MOS-Technology).

Het zesde probleem: packen van een zoned decimal getal.

---

Gegeven:

---

Een resultaat veld A en een operand B met daarin een getal in zoned decimale notatie.

Gevraagd:

---

Ontwerp een routine die het zoned decimale getal van operand B converteert naar een packed decimaal getal in operand A volgens de hiervoor omschreven indeling van packed en zoned decimale getallen.

De status byte blijft ongewijzigd.

Voorbeeld:

---

Operand A heeft een lengte van vier bytes.  
Operand B bevat een positieve waarde 1234, in zoned decimaal formaat, dat als volgt wordt weergegeven:

31 32 33 C4

Na uitvoering van de pack routine, bevat A het gepackte equivalent van de oorspronkelijke zoned decimale waarde 1234, hetseen er als volgt uit ziet:

00 01 23 4C

Het zevende probleem: unpacken van een packed getal.

---

Gegeven:

---

Een resultaat veld A en een operand B met daarin een getal in packed decimale notatie.

Gevraagd:

---

Ontwerp een routine die het packed decimale getal van operand B converteert naar een zoned decimaal getal in operand A, volgens de hiervoor omschreven indeling van zoned- en packed decimale getallen.

De status byte blijft ongewijzigd.

### Voorbeeld:

Operand A heeft een lengte van vier bytes.  
Operand B bevat een positieve waarde 1234, in packed  
decimaal formaat, dat als volgt wordt weergegeven:

00 01 23 4C

Na uitvoering van de unpack routine, bevat A het ge-  
unpakte equivalent van de oorspronkelijke packed deci-  
male waarde 1234, hetgeen er als volgt uit ziet:

31 32 33 C4

### Referenties:

1. IBM System/370 Principles of Operation (GA22-7000-8).
2. The art of computer programming, vol. 2, 2nd edition, Seminumerical algorithms, by Donald E. Knuth, Addison-Wesley Publishing Company, Reading Massachusetts. (ISBN 0-201-03822-6).

### Aanbevolen literatuur:

1. The art of computer programming, vol. 1, 2nd edition, Fundamental Algorithms, by Donald E. Knuth, Addison-Wesley Publishing Company, Reading Massachusetts.
2. The art of computer programming, vol. 2, 2nd edition, Seminumerical Algorithms, by Donald E. Knuth, Addison-Wesley Publishing Company, Reading Massachusetts. (ISBN 0-201-03822-6).
3. Programming a microcomputer, 6502, by Caxton C. Foster, Addison-Wesley Publishing Company, Reading Massachusetts. (ISBN 0-201-01995-7).
4. How to program microcomputers, by William Barden Jr., Howard W. Sams & Co. Inc., 4300 West 62nd Street, Indianapolis, Indiana 46268. (ISBN 0-672-21459-8).
5. 6502 Assembly Language Programming, by Lance A. Leventhal, Osborne/McGraw-Hill, 630 Bancroft Way, Berkeley, CA 94710. (ISBN 0-931988-27-6).
6. Scelbi's 6502 Software Gourmet Guide & Cook Book, by Robert Findley, Scelbi Computer Consulting Inc., 1322 Rear Boston Post Road, Milford, CT 06460.

KIM - SCHAAKPROGRAMMA.

De doelstellingen voor dit programma waren:  
 Een schaakprogramma te ontwikkelen, dat kon worden uitgevoerd op de standaard 1K-versie van de KIM.  
 Voorts dat het programma alle geldige zetten zou kunnen genereren, waaronder rochade, en-passent slaan en minor-promoties, en alle geldige zetten van de tegenpartij zou accepteren (en ook alleen maar geldige).

Uiteraard mag van de kwaliteit van een dergelijk programma niet teveel worden verwacht, maar wellicht is het toch wel aardig voor KIM/JUNIOR gebruikers met dit programma kennis te maken. Mogelijk kan dit programma als basis dienen voor verdere ontwikkelingen (maar dan met meer geheugen).  
 Als iemand belangstelling heeft dit programma samen met mij verder te ontwikkelen, dan gaarne een berichtje of een tele oontje aan:

Theo Kortekaas,  
 Kleine Poellaan 26,  
 Rijsenhout.  
 Tel. 02977 - 21888.

Gebruikers-handleiding.

Er zijn twee versies van dit schaakprogramma: een versie, waarbij de computer wit speelt, en een waarbij de computer zwart speelt.

Het laden van het programma.

Nadat de computer in gereedheid is gebracht, en de cassetterecorder is aangesloten, wordt de cassette in de recorder geplaatst en voor zover nodig ge-rewind. Op de KIM-computer drukt men de volgende toetsen in:

```
-RS          (reset)
-AD          (adres-selektie)
-0, 0, F, 1 (selektie adres 00F1)
-DA          (data)
-0, 0        (op adres 00F1 wordt 00 geplaatst)
-AD
-1, 7, F, 9
-DA
-0, 0        (op adres 17F9 wordt 00 geplaatst)
-/D
-1, 8, 7, 3 (startadres laadprogramma)
-G0
```

\* Het adres is steeds op de vier linker posities van het display te zien, de data op de twee rechter posities.  
 Na het indrukken van G0 wordt het display donker.

Hierna kan de cassetterecorder worden gestart, en wordt het programma ingelezen. Het programma bestaat uit twee delen, die afzonderlijk worden ingelezen.

Als het eerste gedeelte correct is ingelezen, verschijnt op het display; 0000 4C (inlezen eerste deel duurt + 2½ min.)

Door nu binnen vijf seconden op de toets GO te drukken, wordt automatisch het tweede gedeelte van het programma ingelezen.

Binnen een minuut verschijnt nu op het display: 0000 EA.

Het programma is nu ingelezen en gebruiksklaar.

Als binnen een minuut het display niet oplicht, of indien op het display FFFF xx verschijnt, dan is er met het inlezen iets fout gegaan. Begin dan opnieuw.

#### Programma starten en aflezen display.

Nadat op het display 0000 EA is verschenen, kan het programma worden gestart. Dit geschiedt door op de toets GO te drukken.

Als de computer zwart speelt, dan verschijnt op het display: C0dE 00. De tegenspeler is nu in staat zijn eerste zet in te toetsen (zie intoetsen zet).

Als de computer wit speelt, dan begint deze na het indrukken van de toets GO aan het berekenen van de eerste zet. Het display wordt gedoofd, maar vaak licht één positie helder op. Het berekenen van een zet duurt gemiddeld 3 minuten, maar dit kan wel oplopen tot 6 minuten, afhankelijk van de stelling. Wanneer de computer gereed is met de berekening, dan wordt het resultaat op het display vermeld.

De eerste twee posities van het display (van links af) bevatten een aanduiding van het van-veld en de derde en vierde positie een aanduiding van het naar-veld voor het stuk, dat de computer wil spelen. De normale notatie wordt gebruikt (bijv. E2 - E4). De velden G1 t/m G8 worden aangegeven door 01 t/m 08, en de velden H1 t/m H8 door 11 t/m 18 (de G en H kunnen niet op normale wijze op het display worden getoond!). Aanduidingen 'slaan' en 'schaak geven' worden niet gegeven! Bij EN-passent slaan wordt gewoon het van- en naar-veld gegeven. Uit de beweging van de pion kan wel worden afgeleid, dat het om een EN-passent-situatie gaat.

Bij rochade wordt alleen de beweging van de Koning op het display getoond. Ook hier kan uit de beweging van het stuk worden opgemaakt, dat het geen gewone zet is. Tevens ziet men daaruit of het lange, dan wel korte rochade is.

Bij promotie spelen de twee rechter posities van het display een rol. Hierop verschijnt een code voor het stuk tot welk de pion is gepromoveerd.

Code 83	betekent:	Een wit paard
" 84	"	Een witte toren
" 85	"	Een witte loper
" 86	"	Een witte Dame

Code C3	betekent:	Een zwart paard
" C4	"	Een zwarte toren
" C5	"	Een zwarte loper
" C6	"	Een zwarte Dame

Wanneer de computer geen geldige zet meer kan doen (door Pat of Mat), danwel wanneer zeer spoedig mat wordt gegeven, dan verschijnt op het display CODE xx. xx heeft hier geen betekenis. Het spel is dan afgelopen.

### Het intoesten van een zet.

Na het displayen van een zet door de computer staat het toetsenbord geblokkeerd. Dit toetsenbord kan worden vrijgemaakt door de toets reset (RS) in te drukken.

Daarna toetst men in:AD (address-selektie) en de zet die men wil verrichten. Hiertoe kan men de gewone notatie gebruiken. (zie ook het vorige blad). Voor de G moet men de 0 gebruiken, voor de H een 1.

Rochade en en-passant slaan worden op de zelfde manier ingegeven, als ze worden gedisplated door de computer.

Bij promotie moet men een code meegeven om aan te geven tot welk stuk de pion promoveert. Dit kan als volgt:

- AD (address - selektie)
- 0, 0, 0, 0 (address 00 00)
- DA (data invoer)
- code van het stuk (zie vorige blad)
- AD

\* in de rechter twee posities van het display is de ingegeven code te zien.

Hierna kan de zet op de normale manier worden ingegeven.

Wanneer de zet is ingegeven drukt men op de toets ST waarmee het programma weer gestart wordt voor het berekenen van de volgende zet.

Als de ingegeven zet onjuist is of ongeldig, dan verschijnt vrij snel na het indrukken van de toets ST op het display: CODE FF. Hierna kan men op de gebruikelijke manier een juiste zet ingeven.

Het mat of pat staan vande tegenspeler wordt niet afzonderlijk aangegeven. In zo'n geval is het eenvoudigweg niet meer mogelijk een nieuwe zet in te geven, en leidt elke ingegeven zet tot het verschijnen van CODE FF op het display.

JUNIOR-aanpssingen.

In de code zijn de (geringe) patches voor de Junior aangegeven. Verder moet men behalve het stukje initialisering ook de beginstand met de hand intoetsen (ik heb nog geen cassette), te weten 0015 t/m 0025 (lege velden doet het programma zelf in het deel 0003 t/m 000B).

Tenslotte kan men het initialiseren nog inkorten door 002E t/m 003D te laten vervallen met de volgende toevoegingen:

```
002E 4C 03 00
1A7A 92
1A7B 00
1A7E 00
1A7F 1A
```

003E t/m 0043 vervalt dan ook

NB voor computer speelt zwart :

```
0010 84 F9
0012 4C 54 1A
1A64 C0 3B
```

*(Lees verder op pagina 40)*

VOOR U GELEZEN

HET TIJDSCHRIFT ELEKTUUR HEEFT DE JUNIOR-BEZITTER VERRIJKT MET EEN INTERFACE-KAART VOOR HET KONTAKT MET DE BUITENWERELD VIA EXTRA I/O. OP DIE INTERFACE-KAART ZIT EEN ZG. 'VERSATILE INTERFACE ADAPTER' OFWEL -VIA- VAN HET TYPE 6522.

IN APRIL 1981 GING ELEKTUUR SUMMIER IN OP DIE 6522 EN HET GEBRUIK ERVAN. TE WEINIG OM OPERATIONEEL TE LATEN ZIJN EN TE VEEL VOOR VELEN OM ERVAN AF TE ZIEN. EN WAAR OM ZOU MEN OOK, ELEKTUUR BELOOFDE: "DE VIA IC 1, VAN HET TYPE 6522, IS EEN HOOFDSTUK APART. EN WEL EEN HOOFDSTUK VAN DE TWEDE HELFT VAN BOEK 3". IN DEEL 3 WORDT DEZE BELOFTE NIET VERVULD. IN DEEL 4 WORDT HET BELOOFDE HOOFDSTUK OOK NIET GEPUBLICEERD, MAAR WORDT VERWEZEN NAAR "ERG VEEL GOEDE LITERATUUR", T.W. R6522 - DATASHEET ROCKWELL; SY6522/SY6522A-DATASHEET SYNERTEK; SECTION 6 R6500 HARDWARE MANUAL ROCKWELL.

WIJ WILLEN U NOG WAT LITERATUUR OPGEVEN, EN WEL: 6502 ASSEMBLY LANGUAGE PROGRAMMING, PAG. 11-23 T/M PAG. 11-38; PROGRAMMING & INTERFACING THE 6502, PAG. 218 EV., MET VOLLEDIGE DATASHEET VAN, DE VIA, OPGENOMEN IN APPENDIX C, EN VOORBEELDEN VAN PROGRAMMERING; RODNAY ZAKS, 6502 APPLICATIONS BOOK; RODNAY ZAKS, PROGRAMMING THE 6502; LEO J. SCANLON, 6502 SOFTWARE DESIGN. DE BOEKEN ZIJN VERKRIJGBAAR BIJ COMPUTERBOEKHANDEL WOLFKAMP TE AMSTERDAM. OOK HET TIJDSCHRIFT MICRO THE 6502 JOURNAL VAN JANUARI 1981 BESTEEDDE AANDACHT AAN DE VIA MIDDELS EEN ARTIKEL "INTERFACING THE 6522 VERSATILE INTERFACE ADAPTER".

HJ00 81 6502 ASSEMBLER V2.0 PAGE 81

```

0010: *****
0020: *
0030: * SCHAAR PROGRAMMA          JUNIOR EN KIM          *
0040: *
0050: * AUTEUR : THEO KORTEKAAS   *
0060: *****
0070: *****
0080: ;
0090: 0000  CHES  ORG  $0000 ;
0100: ;
0110: ;
0120: ; ZERO PAGE LOKATIES
0130: ;
0140: 0000  ZP    *    $0000 ;
0150: 0000  ZETI *    ZP  +00 ; INTERNE REPRESENTATIE ZET
0160: 0000  ZETII *   ZP  +01 ; ZETI BEVAT EUT PROMOTIE STUK
0170: 0000  ZETIII *  ZP  +02 ; ZETII NAAR VELD EN ZETIII VAN VELD
0180: 0000  PROM  *   ZP  +03 ; STUK CODE AS RESULTAAT PROMOTIE
0190: 0000  NAAR  *   ZP  +04 ; NAAR VELD : HEX 00 - 3F
0200: 0000  VAN   *   ZP  +05 ; VAN VELD   : HEX 00 - 3F
0210: 0000  NZET  *   ZP  +06 ; AANTAL GELDIGE ZETTEN PER NIVO
0220: 0000  ITZ   *   ZP  +07 ; INDEX TABEL ZET MOGELIJKHEDEN
0230: 0000  ZWRD  *   ZP  +08 ; ZET WAARDE PER NIVO
0240: 0000  STUK  *   ZP  +09 ; CODE VAN STUK DAT WORDT GEZET
0250: 0000  ROCO  *   ZP  +0A ; ROCHADE CODE
0260: 0000  WRDE  *   ZP  +0B ; WAARDE STELLING
0270: 0000  NIVO  *   ZP  +0C ; AANTAL HALVE ZETTEN
0280: 0000  CZA   *   ZP  +0D ; CODE ZET AANBRENGEN
0290: 0000  CZO   *   ZP  +0E ; CODE ZET ONGELDIG
0300: 0000  PZET  *   ZP  +0F ; HOOGSTE AANTAL GELDIGE ZETTEN NIVO 2
0310: 0000  CZET  *   ZP  +10 ; CODE SOORT ET
0320: 0000  MAXI  *   ZP  +11 ; MAX AANTAL ZETTEN NORMAAL
0330: 0000  MAXII *   ZP  +12 ; MAX AANTAL HALVE ZETTEN VOOR SLAAN
0340: 0000  EPS   *   ZP  +13 ; VELDNUMMER PION ENFASSANT TE SLAAN
0350: 0000  HULP  *   ZP  +14 ;
0360: 0000  WIS   *   ZP  +90 ; WISSEL STADIUM BEREKENING
0370: 0000  OZET  *   ZP  +91 ;
0380: 0000  DSPI  *   ZP  +F9 ; DISPLAY BUFFER
0390: 0000  DSPII *   ZP  +FA ;
0400: 0000  DSPIII *  ZP  +FB ;
0410: ;
0420: ; MONITOR ADRESSEN
0430: ;
0440: 0000  LOAD  *   $1873 ; BASIS JUNIOR NIET VAN TOEPASSING
0450: 0000  MONI  *   $1C00 ;
0460: 0000  SCANDS *  $1F1F ; JUNIOR : $1D8E
0470: ;
0480: ; PIA TIMER ADRES
0490: ;
0500: 0000  TIMER *   $1704 ; JUNIOR : $1AF4
0510: ;
0520: ; NMI VECTOR
0530: ;
0540: 0000  VECT  *   $17FA ; JUNIOR : $1A7A
0550: ;
0560: ; ZERO PAGE CONSTANTEN

```

CHESS HJCO 81 6502 ASSEMBLER V2.0 PAGE 02

```

0570:      ;
0580: 0015      ORG    $0015 ;
0590:      ;
0600: 0015 FF    CKAZ   =    $FF    ; CODE KLEUR AAN ZET
0610: 0016 84    BORD   =    $84    ; WEERGAVE SCHAKBORD 64 POSITIES
0620: 0017 83      =    $83    ; GENUMMERD HEX 00 - 3F
0630: 0018 85      =    $85    ;
0640: 0019 86      =    $86    ; POS 00 = A1 , POS 01 = A2
0650: 001A 82      =    $82    ; POS 08 = B1 , POS 3F = H8
0660: 001B 85      =    $85    ;
0670: 001C 83      =    $83    ; LEEG VELD = 00
0680: 001D 84      =    $84    ; WITTE PION = 80
0690: 001E 80      =    $80    ;      KONING = 82
0700: 001F 80      =    $80    ;      PAARD = 83
0710: 0020 80      =    $80    ;      TOREN = 84
0720: 0021 80      =    $80    ;      LOPER = 85
0730: 0022 80      =    $80    ;      DAME = 86
0740: 0023 80      =    $80    ; ZWARTE PION = C1
0750: 0024 80      =    $80    ;      KONING = C2
0760: 0025 80      =    $80    ;      PAARD = C3
0770: 0026 80      =    $80    ;      TOREN = C4
0780: 0027 80      =    $80    ;      LOPER = C5
0790: 0028 80      =    $80    ;      DAME = C6
0800: 0029 00      =    $00    ;
0810: 002A 00      =    $00    ;
0820: 002B 00      =    $00    ;
0830: 002C 00      =    $00    ;
0840: 002D 00      =    $00    ;
0850: 002E 00      =    $00    ;
0860: 002F 00      =    $00    ;
0870: 0030 00      =    $00    ;
0880: 0031 00      =    $00    ;
0890: 0032 00      =    $00    ;
0900: 0033 00      =    $00    ;
0910: 0034 00      =    $00    ;
0920: 0035 00      =    $00    ;
0930: 0036 00      =    $00    ;
0940: 0037 00      =    $00    ;
0950: 0038 00      =    $00    ;
0960: 0039 00      =    $00    ;
0970: 003A 00      =    $00    ;
0980: 003B 00      =    $00    ;
0990: 003C 00      =    $00    ;
1000: 003D 00      =    $00    ;
1010: 003E 00      =    $00    ;
1020: 003F 00      =    $00    ;
1030: 0040 00      =    $00    ;
1040: 0041 00      =    $00    ;
1050: 0042 00      =    $00    ;
1060: 0043 00      =    $00    ;
1070: 0044 00      =    $00    ;
1080: 0045 00      =    $00    ;
1090: 0046 C1      =    $C1    ;
1100: 0047 C1      =    $C1    ;
1110: 0048 C1      =    $C1    ;
1120: 0049 C1      =    $C1    ;

```

# 6502

# AMUSEMENT

HJCO 81 6502 ASSEMBLER V2.0 PAGE 03

1130:	004A	C1	=	#C1	;	
1140:	004B	C1	=	#C1	;	
1150:	004C	C1	=	#C1	;	
1160:	004D	C1	=	#C1	;	
1170:	004E	C4	=	#C4	;	
1180:	004F	C3	=	#C3	;	
1190:	0050	C5	=	#C5	;	
1200:	0051	C6	=	#C6	;	
1210:	0052	C2	=	#C2	;	
1220:	0053	C5	=	#C5	;	
1230:	0054	C3	=	#C3	;	
1240:	0055	C4	=	#C4	;	
1250:	0056	C0	FTZ	=	#C0	;
1260:	0057	C5	=	#C5	;	
1270:	0058	0A	=	#0A	;	
1280:	0059	13	=	#13	;	
1290:	005A	1C	=	#1C	;	
1300:	005B	4F	=	#0E	;	
1310:	005C	0A	=	#0A	;	
1320:	005D	01	TZW	=	#01	;
1330:	005E	01	=	#01	;	
1340:	005F	00	=	#00	;	
1350:	0060	03	=	#03	;	
1360:	0061	05	=	#05	;	
1370:	0062	03	=	#03	;	
1380:	0063	00	=	#00	;	
1390:	0064	05	TZET	=	#05	;
1400:	0065	0F	=	#0F	;	
1410:	0066	A4	=	#A4	;	
1420:	0067	24	=	#24	;	
1430:	0068	00	=	#00	;	
1440:	0069	15	=	#15	;	
1450:	006A	1F	=	#1F	;	
1460:	006B	B4	=	#B4	;	
1470:	006C	34	=	#34	;	
1480:	006D	00	=	#00	;	
1490:	006E	06	=	#06	;	
1500:	006F	16	=	#16	;	
1510:	0070	22	=	#22	;	
1520:	0071	A2	=	#A2	;	
1530:	0072	26	=	#26	;	
1540:	0073	36	=	#36	;	
1550:	0074	A6	=	#A6	;	
1560:	0075	B6	=	#B6	;	
1570:	0076	00	=	#00	;	
1580:	0077	2E	=	#2E	;	
1590:	0078	3E	=	#3E	;	
1600:	0079	AE	=	#AE	;	
1610:	007A	BE	=	#BE	;	
1620:	007B	66	=	#66	;	
1630:	007C	76	=	#76	;	
1640:	007D	E6	=	#E6	;	
1650:	007E	F6	=	#F6	;	
1660:	007F	00	=	#00	;	
1670:	0080	06	=	#06	;	
1680:	0081	16	=	#16	;	

# 6502

# AMUSEMENT

HJCO 81 6502 ASSEMBLER V2.0 PAGE 04

```

1690: 0082 22          = $22 ;
1700: 0083 A2          = $A2 ;
1710: 0084 80          RCTI = $80 ;
1720: 0085 04          = $04 ;
1730: 0086 07          = $07 ;
1740: 0087 38          = $38 ;
1750: 0088 3C          = $3C ;
1760: 0089 3F          = $3F ;
1770: 008A 40          RCTII = $40 ;
1780: 008B C0          = $C0 ;
1790: 008C 80          = $80 ;
1800: 008D 10          = $10 ;
1810: 008E 30          = $30 ;
1820: 008F 20          = $20 ;
1830: ;
1840: ; INITIALISERING
1850: ;
1860: 0000              ORG $0000 ;
1870: ;
1880: 0000 4C 26 00     INIT JMP INII ;
1890: 0003 A0 00        INIV LDYIN $00 ;
1900: 0005 A2 1F        LDYIN $1F ;
1910: 0007 94 26        VNI2 STVZX BORD +10 ;
1920: 0009 CA          DEX ;
1930: 000A 10 FB        BPL VNI2 ;
1940: 000C 84 0A        STY RODO ;
1950: 000E 84 0B        STY WRDE ;
1960: 0010 86 90        STX WIS ;
1970: 0012 4C 00 17     JMP BREAK ; JUNIOR : 4C 00 1A
1980: ;
1990: ; INITIALISERING DEEL 2
2000: ;
2010: 0026              ORG $0026 ;
2020: ;
2030: 0026 A9 EA        INII LDYIN $EA ; NOP VERVANG OPDRACHT OF 0000
2040: 0028 85 00        STA INII ;
2050: 002A A9 85        LDYIN $85 ; STAZ
2060: 002C 85 01        STA INII +01 ;
2070: 002E EA          NOP ;
2080: 002F EA          NOP ;
2090: 0030 EA          NOP ;
2100: 0031 A2 05        LDYIN $05 ;
2110: 0033 B5 3E        INIIII LDZKX STOP ; ZET NMI IRQ EN BRK VECTOR
2120: 0035 9D FA 17     STAZKX VECT ; JUNIOR : 9D 7A 1A
2130: 0038 CA          DEX ;
2140: 0039 10 F8        BPL INIIII ;
2150: 003B 4C 73 18     JMP LOAD ; JUNIOR : 4C 03 00 , GEEN LOAD BASIS JUNIOR
2160: ;
2170: 003E 92          STOP = $92 ; NMI VECTOR
2180: 003F 00          = $00 ;
2190: 0040 00          RSET = $00 ; ST VECTOR
2200: 0041 1C          = $1C ;
2210: 0042 80          BRK = $80 ; JUNIOR : $80
2220: 0043 17          = $17 ; $1A BRK VECTOR
2230: ;
2240: ;

```

# 6502

# AMUSEMENT

HJ00 01 6502 ROSENBLER V2.0 PAGE 00

```

2250:          ; HOOFDROUTINE
2260:          ;
2270: 0092          ORG  $0092 ;
2280:          ;
2290: 0092 A0 00      HFD  LDYIM #00 ; KOM HIER NA STOP TOETS
2300: 0094 A2 01      LDYIM #01 ;
2310: 0096 B5 FA      HI   LDZX DSP11 ;
2320: 0098 18          CLC ;
2330: 0099 69 5F      ADCIN #5F ;
2340: 009B 96 FA      STAZY DSP11 ;
2350: 009D 4A          LSRA ;
2360: 009E 4A          LSRA ;
2370: 009F 4A          LSRA ;
2380: 00A0 4A          LSRA ;
2390: 00A1 05 14      STA  HULP ;
2400: 00A3 B5 FA      LDZX DSP11 ;
2410: 00A5 29 07      ANDIN #07 ;
2420: 00A7 0A          ASLA ;
2430: 00A8 0A          ASLA ;
2440: 00A9 0A          ASLA ;
2450: 00AA 05 14      ORA  HULP ;
2460: 00AC 95 01      STAZX ZET11 ;
2470: 00AE 0A          DEX ;
2480: 00AF 10 E5      BPL  NI ;
2490: 00B1 06 00      STX  WIS ;
2500: 00B3 40 C2 17   JNP  HV ; JUNIOR : 40 42 1A
2510:          ;
2520:          ; HOOFDROUTINE DEEL BRAKE
2530:          ; JUNIOR : ROUTINE FLANTSEN VAN #1A00 T/M #1A61
2540:          ;
2550: 1700          ORG  $1700 ;
2560:          ;
2570: 1700 04 00      BREAK STY  ZET1 ; KOM HIER NA BRK INSTRUCTIE
2580: 1702 A6 90      LDY  WIS ;
2590: 1704 F0 50      BEQ  DISP ;
2600: 1706 9A          TWS ;
2610: 1707 E0          IRS ;
2620: 1708 06 00      STX  CZR ;
2630: 170A 06 90      STX  WIS ;
2640: 170C 06 0F      STX  FZET ;
2650: 170E 06 0C      STX  HIVO ;
2660: 1710 A2 02      LDYIM #02 ;
2670: 1712 06 11      STX  NAX1 ;
2680: 1714 A2 03      LDYIM #03 ;
2690: 1716 06 12      STX  NAX11 ;
2700: 1718 20 00 02   JSR  CALC ; BIJ UITBREIDING HIER PATCH
2710: 171A A5 03      LDA  ZWRD ;
2720: 171C 05 F9      STA  DSP1 ;
2730: 171E 04 06      COPY NZET ;
2740: 1720 17A1 F0 31  BEQ  CODE ;
2750: 1722 17A3 A2 01      LDYIM #01 ;
2760: 1724 17A5 B5 01      MIII LDZX ZET11 ;
2770: 1726 17A7 4A          LSRA ;
2780: 1728 17A8 4A          LSRA ;
2790: 172A 17A9 4A          LSRA ;
2800: 172C 17AA 05 14      STA  HULP ;

```

HJCO 81 6502 ASSEMBLER V2.0 PAGE 06

```

2810: 17AC B5 01          LDAZX ZETII ;
2820: 17AE 29 07          ANDIM #07 ;
2830: 17B0 0A             ASLA ;
2840: 17B1 0A             ASLA ;
2850: 17B2 0A             ASLA ;
2860: 17B3 0A             ASLA ;
2870: 17B4 05 14        ORA HULP ;
2880: 17B6 18             CLC ;
2890: 17B7 69 A1         ADCIM #A1 ;
2900: 17B9 95 FA         STAZX DSPII ;
2910: 17BB CA             DEX ;
2920: 17BC 10 E7         BPL KIII ;
2930: 17BE A5 00         LDA ZETI ;
2940: 17C0 85 F9         STA DSPI ;
2950: ;
2960: ; HOOFDRoutine AANBRENGEN ZET
2970: ; X BEVAT #FF
2980: ;
2990: 17C2 86 0D        MV STX CZR ;
3000: 17C4 9A             TXS ;
3010: 17C5 E8             INX ;
3020: 17C6 86 0C        STX NIUG ;
3030: 17C8 E8             INX ;
3040: 17C9 86 11        STX MAXI ;
3050: 17CB 86 12        STX MAXII ;
3060: 17CD 20 00 02     JSR CALC ;
3070: 17D0 A9 FF        FOUT LDAIM #FF ;
3080: 17D2 85 F9        STA DSP1 ;
3090: 17D4 A9 00        CODE LDAIM #00 ;
3100: 17D6 85 FB        STA DSPIII ;
3110: 17D8 A9 DE        LDAIM #DE ;
3120: 17DA 85 FA        STA DSPII ;
3130: 17DC 20 1F 1F     DISP JSR SCANDS ;
3140: 17DF 4C DC 17     JMP DISP ; JUNIOR : 4C 8E 1D
3150: ;
3160: ; SUBROUTINE MOVE
3170: ; ROCHADE
3180: ;
3190: 00B6              ORG #00B6 ;
3200: ;
3210: 00B6 A9 01        MVRO LDAIM #01 ;
3220: 00B8 C5 0C        CMP NIUG ;
3230: 00BA D0 02        BNE MVY ;
3240: 00BC 84 0F        STV PZET ;
3250: 00BE 20 D0 00     MVY JSR MOVE ; JUNIOR : 20 C2 00
3260: 00C1 60             RTS ;
3270: ;
3280: ; JUNIOR : PLAATSEN VAN #1A62 T/M #1A6E
3290: 00C2 A4 05        PTCH LDY VAN ; HOODMATEREGEL
3300: 00C4 C0 3B        CPYIM #3B ; OM TE VOORKOMEN
3310: 00C6 D0 02        BNE BRC ; DAT DAME TE SNEL
3320: 00C8 86 91        STX OZET ; IN SPEL WORDT
3330: 00CA A6 0F        BRC LDX PZET ; BETROKKEN
3340: 00CC E4 91        CPX OZET ;
3350: 00CE EA             NOP ;
3360: 00CF 60             RTS ;

```

# 6502

# AMUSEMENT

HJCO 81 6502 ASSEMBLER V2.0 PAGE 07

```
3370: ;
3380: ; MOVE
3390: ; JUNIOR : PLAATSEN VAN $00C2 T/M $00E0
3400: ;
3410: 00D0 84 0E MOVE STY C20 ;
3420: 00D2 A6 04 LDX NAAR ;
3430: 00D4 B5 16 LDAZX BORD ;
3440: 00D6 85 09 STA STUK ;
3450: 00D8 D0 26 BNE BIO ; JUNIOR : D0 34
3460: 00DA C4 06 CPY NZET ;
3470: 00DC F0 33 BEQ BO ; JUNIOR : F0 41
3480: 00DE A5 11 LDA MAXI ;
3490: 00E0 C5 0C CMP NIVO ;
3500: 00E2 B0 2D BCS BO ; JUNIOR : B0 38
3510: 00E4 E6 06 INC NZET ;
3520: 00E6 A5 0B LDA WRDE ;
3530: 00E8 C5 08 CMP ZWRD ;
3540: 00EA 30 02 BMI BJ ;
3550: 00EC 85 08 STA ZWRD ;
3560: 00EE 60 BJ RTS ;
3570: ;
3580: 0100 ; ORG $0100 ;
3590: ;
3600: 0100 45 15 BIO EOR OKAZ ;
3610: 0102 29 40 ANDIM $40 ;
3620: 0104 F0 0A BEQ EXM ;
3630: 0106 B5 16 LDAZX BORD ;
3640: 0108 29 07 ANDIM $07 ;
3650: 010A C9 02 CMPIM $02 ;
3660: 010C D0 03 BNE BO ;
3670: 010E C6 0E DEC C20 ;
3680: 0110 60 EXM RTS ;
3690: ;
3700: 0111 A2 08 BO LDXIM $08 ;
3710: 0113 B5 03 BOI LDAZX PROM ;
3720: 0115 48 PHA ;
3730: 0116 CA DEX ;
3740: 0117 10 FA BPL BOI ;
3750: 0119 A0 06 LDYIM $06 ;
3760: 011B A5 0A LDA ROCC ;
3770: 011D B6 83 BIA LDXZY RCTI -01
3780: 011F E4 05 CPX VAN ;
3790: 0121 F0 04 BEQ BIB ;
3800: 0123 E4 04 CPX NAAR ;
3810: 0125 D0 03 BNE BIC ;
3820: 0127 19 89 00 BIB ORARY RCTII -01
3830: 012A 88 BIC DEY ;
3840: 012B D0 F0 BNE BIA ;
3850: 012D 85 0A STA ROCC ;
3860: 012F A6 05 LDX VAN ;
3870: 0131 B5 16 LDAZX BORD ;
3880: 0133 94 16 STYZX BORD ;
3890: 0135 A6 04 LDX NAAR ;
3900: 0137 95 16 STAZX BORD ;
3910: 0139 A5 09 LDA STUK ;
3920: 013B F0 0A BEQ BID ;
```

# 6502

# AMUSEMENT

HJCO 6502 ASSEMBLER V2.0 PAGE 08

```
3930: 013D 29 07      ANDIM #07      ;
3940: 013F AA           TAX            ;
3950: 0140 A5 0B      LDA  WRDE     ;
3960: 0142 18           CLC           ;
3970: 0143 75 5D      ADCZX TSW     ;
3980: 0145 05 0B      STA  WRDE     ;
3990: 0147 A5 0B      LDA  WRDE     ;
4000: 0149 05 00      STA  ZWRD     ;
4010: 014B A5 12      LDA  MAXII    ;
4020: 014D 05 0C      CNP  NIVO     ;
4030: 014F 90 03      BCC  BRA      ;
4040: 0151 20 00 02    JSR  CALC     ;
4050: 0154 A2 00      BRA  LDXIM #00 ;
4060: 0156 68           BIE  PLA      ;
4070: 0157 95 03      STAZX PROM    ;
4080: 0159 E8           INX          ;
4090: 015A E8 05      CPXIM #05     ;
4100: 015C D8 F8      BNE  BIE      ;
4110: 015E 24 0E      BIT  C20      ;
4120: 0160 E6 0E      INC  C20      ;
4130: 0162 68           PLA          ;
4140: 0163 70 3E      BVS  BIK      ;
4150: 0165 E6 06      INC  NZET     ;
4160: 0167 24 05      BIT  VAN      ;
4170: 0169 70 38      BVS  BIK      ;
4180: 016B A6 0C      LDX  NIVO     ;
4190: 016D E8 01      CPXIM #01     ;
4200: 016F D8 2E      BNE  BIJ      ;
4210: 0171 24 0D      BIT  C2A      ;
4220: 0173 70 1E      BVS  BIH      ;
4230: 0175 05 08      CNP  ZWRD     ;
4240: 0177 D8 09      BNE  BIF      ;
4250: 0179 20 02 00    JSR  PTOH     ; JUNIOR : 20 62 1A
4260: 017C EA           NOP          ;
4270: 017D D8 03      BNE  BIF      ;
4280: 017F CD 04 17    CNP  TIMER    ; JUNIOR : CD F4 1A
4290: 0182 10 1F      BIF  BPL      ;
4300: 0184 A6 91      LDX  OZET     ;
4310: 0186 86 0F      STX  PZET     ;
4320: 0188 A2 02      LDXIM #02     ;
4330: 018A B5 03      LDAZX PROM    ;
4340: 018C 95 00      STAZX ZETI    ;
4350: 018E CA           DEX          ;
4360: 018F 10 F9      BPL  BIG      ;
4370: 0191 30 12      BMI  BIL      ;
4380:                   ;
4390: 0193 A2 02      BIH  LDXIM #02 ;
4400: 0195 B5 03      BII  LDAZX PROM ;
4410: 0197 D5 00      CNPZX ZETI    ;
4420: 0199 D8 04      BNE  BIJ      ;
4430: 019B CA           DEX          ;
4440: 019C 10 F7      BPL  BII      ;
4450: 019E 00           BRK          ;
4460:                   ;
4470: 019F C5 08      BIJ  CNP  ZWRD ;
4480: 01A1 30 02      BMI  BIL      ;
```

# 6502

# AMUSEMENT

R200 81 6502 ASSEMBLER V2.0 PAGE 83

```

4490: 01A5 95 08      BIK   STA   ZWRD   ;
4500: 01A5 A6 04      BIL   LDX   NAAR   ;
4510: 01A7 B4 16              LDYXZ  BORD   ;
4520: 01A9 68              PLA                    ;
4530: 01AA 95 16              STAZX  BORD   ;
4540: 01AC A6 05              LDX   VAN    ;
4550: 01AE 94 16              STYXZ  BORD   ;
4560: 01B0 68              PLA                    ;
4570: 01B1 85 0A              STA   R000   ;
4580: 01B3 68              PLA                    ;
4590: 01B4 85 0B              STA   WRDE   ;
4600: 01B6 A0 00              LDYIM  #00   ;
4610: 01B8 60              RTS                    ;
4620:                      ;
4630:                      ; SUBROUTINE CALL
4640:                      ;
4650: 0200                      ORG   $0200
4660:                      ;
4670: 0200 E6 0C      CALC  INC   R100   ;
4680: 0202 A9 00              LDYIM  #00   ;
4690: 0204 85 08              STA   ZWRD   ;
4700: 0206 45 15              EOR   CRAL   ;
4710: 0208 85 15              STA   CRAL   ;
4720: 020A 38              TYP                    ;
4730: 020B 85 06              STA   NZET   ;
4740: 020D 38              SEC                    ;
4750: 020E E5 0E              SEC   WRDE   ;
4760: 0210 85 0E              STA   WRDE   ;
4770: 0212 A0 3F              LDYIM  #3F   ;
4780: 0214 85 05              STA   VAN    ;
4790: 0216 A6 05      BOZ   LDX   VAN    ;
4800: 0218 B5 16              LDZAX  BORD   ;
4810: 021A F0 00              BEQ   B0N    ;
4820: 021C 45 15              EOR   CRAL   ;
4830: 021E 29 40              ANDIM  #40   ;
4840: 0220 D0 66              BNE   B0N    ;
4850: 0222 B5 16              LDZAX  BORD   ;
4860: 0224 29 07              ANDIM  #07   ;
4870: 0226 AA              TAX                    ;
4880: 0227 B5 56              LDZAX  PTZ   ;
4890: 0229 85 07              STA   ITZ   ;
4900: 022B A6 05      B0A   LDX   VAN    ;
4910: 022D 86 04              STX   NAAR   ;
4920: 022F 46 04      B0C   LSR   NAAR   ;
4930: 0231 46 04              LSR   NAAR   ;
4940: 0233 46 04              LSR   NAAR   ;
4950: 0235 8A              TXA                    ;
4960: 0236 29 07              ANDIM  #07   ;
4970: 0238 AA              TAX                    ;
4980: 0239 A4 07              LDY   ITZ   ;
4990: 023B B9 64 00          LDYAY  TZET   ;
5000: 023E A0 02              LDYIM  #02   ;
5010: 0240 0A      B0D   ASLA                    ;
5020: 0241 B0 0E              BCS   B0G    ;
5030: 0243 0A              ASLA                    ;
5040: 0244 90 01              BCC   B0E    ;

```

# 6502

# AMUSEMENT

HJCO 01 6502 ASSEMBLER V2.0 PAGE 10

5050:	0246	E8		INX					
5060:	0247	0A	BOE	ASLA					
5070:	0248	90 01		BCC	BOF				
5080:	024A	E8		INX					
5090:	024B	E8 00	BOF	CPXIN	#88				
5100:	024D	90 0A		BCC	BOZZ				
5110:	024F	B0 2F		BOS	BOL				
5120:	0251	0A	BOG	ASLA					
5130:	0252	90 01		BCC	BOH				
5140:	0254	0A		DEX					
5150:	0255	0A	BOH	ASLA					
5160:	0256	0A		DEX					
5170:	0257	30 27		BMI	BOL				
5180:	0259	88	BOZZ	DEY					
5190:	025A	F0 38		BEG	BOD				
5200:	025C	86 14		STX	HULP				
5210:	025E	A6 04		LDX	NAAR				
5220:	0260	10 DE		BFL	BOD				
5230:	0262	20 D0 00	BOJ	JSR	MOVE		JUNIOR : 20 C2 00		
5240:	0265	60	BOK	FLA					
5250:	0266	85 13		STA	EPS				
5260:	0268	24 0E		BIT	CZO				
5270:	026A	70 25		BUS	BON				
5280:	026C	A5 08		LDA	ZWRD				
5290:	026E	C9 41		CMPI	#41				
5300:	0270	F0 1F		BEG	BON				
5310:	0272	A6 05		LDX	VAN				
5320:	0274	B5 16		LDAZX	BORD				
5330:	0276	29 04		ANDIN	#04				
5340:	0278	F0 06		BEG	BOL				
5350:	027A	A6 04		LDX	NAAR				
5360:	027C	B5 16		LDAZX	BORD				
5370:	027E	F0 AF		BEG	BOD				
5380:	0280	E6 07	BOL	INC	ITZ				
5390:	0282	A6 07		LDX	ITZ				
5400:	0284	B5 64		LDAZX	TZET				
5410:	0286	D0 A3		BNE	BOR				
5420:	0288	C6 05	BOM	DEC	VAN				
5430:	028A	10 8A		BFL	BOZ				
5440:	028C	A0 00		LDYIM	#00				
5450:	028E	4C 3A 03		JMP	BN				
5460:	0291	4C CE 03	BON	JMP	BBD				
5470:	0294	85 10	BOO	STA	CZET				
5480:	0296	8A		TXA					
5490:	0297	0A		ASLA					
5500:	0298	0A		ASLA					
5510:	0299	0A		ASLA					
5520:	029A	05 14		ORA	HULP				
5530:	029C	85 04		STA	NAAR				
5540:	029E	A5 13		LDA	EPS				
5550:	02A0	48		PHA					
5560:	02A1	84 03		STY	PROM				
5570:	02A3	24 10		BIT	CZET				
5580:	02A5	30 02		BMI	BIIZ				
5590:	02A7	50 12		BUC	BA				
5600:	02A9	84 13	BIIZ	STY	EPS				

```

5610: 02AB 10 4C          BPL  BG      ;
5620: 02AD 50 B3          BVC  B0J     ;
5630: 02AF A5 04          LDA  NAAR    ;
5640: 02B1 24 15          BIT  OKAZ    ;
5650: 02B3 09 20          CNPIN $20   ;
5660: 02B5 50 35          BVC  BE      ;
5670: 02B7 90 AC          BCC  BOK     ;
5680: 02B9 B0 33          BCS  BF      ;
5690:                    ;
5700:                    ; PION SCHUIN SLAAN
5710:                    ;
5720: 02BB A6 04          BA   LDW  NAAR ;
5730: 02BD B5 16          LDAZX BORD   ;
5740: 02BF F0 04          BEG  BB      ;
5750: 02C1 84 13          STY  EPS     ;
5760: 02C3 D0 3A          BNE  BH      ;
5770:                    ;
5780:                    ; PION EN PASSANT SLAAN
5790:                    ;
5800: 02C5 E4 13          BB   CPX  EPS ;
5810: 02C7 D0 6B          BNE  BL      ;
5820: 02C9 A0 08          LDYIN $08   ;
5830: 02CB 24 15          BIT  OKAZ    ;
5840: 02CD E8          BC   INX     ;
5850: 02CE 70 02          BVC  BU      ;
5860: 02D0 CA          DEX         ;
5870: 02D1 CA          DEX         ;
5880: 02D2 88          BD   DEY     ;
5890: 02D3 D0 F8          BNE  BC      ;
5900: 02D5 84 13          STY  EPS     ;
5910: 02D7 B5 16          LDAZX BORD   ;
5920: 02D9 94 16          STYZX BORD   ;
5930: 02DB 48          PHA         ;
5940: 02DC 8A          TXA         ;
5950: 02DD 48          PHA         ;
5960: 02DE E6 0B          INC  WRDE    ;
5970: 02E0 20 D0 00        JSR  MOVE    ; JUNIOR : 20 02 00
5980: 02E3 C6 0B          DEC  WRDE    ;
5990: 02E5 68          PLA         ;
6000: 02E6 AA          TAX         ;
6010: 02E7 68          PLA         ;
6020: 02E8 95 16          STAZX BORD   ;
6030: 02EA D0 48          BNE  BL      ;
6040:                    ;
6050:                    ; PION TWEE VELDEN VOORUIT
6060:                    ;
6070: 02EC B0 46          BE   BCS  BL  ;
6080: 02EE 18          BF   CLC     ;
6090: 02EF 65 05          ADC  VAN     ;
6100: 02F1 4A          LSRA        ;
6110: 02F2 AA          TAX         ;
6120: 02F3 B5 16          LDAZX BORD   ;
6130: 02F5 D0 3D          BNE  BL      ;
6140: 02F7 86 13          STX  EPS     ;
6150: 02F9 A6 04          BG   LDX  NAAR ;
6160: 02FB B5 16          LDAZX BORD   ; PION EEN VELD VOORUIT

```

# KIM CLUB BIJeenKOMST

## UITNODIGING KIM CLUB LUSTRUM BIJeenKOMST

Datum: Op 20 maart 1982 vieren wij het heugelijke  
----- feit, dat onze vereniging VIJF JAAR bestaat.

Lokatie: Cultureel Centrum "De Drietand",  
----- Neptunusplein 34 te Amersfoort.

Reisroute:  
-----

- per auto vanaf snelweg A1 Amsterdam-Apeldoorn-Hengelo  
-----

Bij kruispunt Hoewelaken richting Utrecht,  
afslag Amersfoort nemen, bij stoplicht r.a.,  
bij 2e stoplicht l.a., na 100 meter ligt  
rechts het Neptunusplein.

vanaf Arnhem - Utrecht - Den Bosch  
-----

Bij Utrecht borden richting Amersfoort volgen,  
langs Zeist en Soesterberg naar rotonde ten  
Zuiden van Amersfoort richting Apeldoorn.  
Snelweg volgen tot afslag Hoewelaken, onder-  
aan links onder viaduct richting Amersfoort.  
Bij stoplicht verder route als boven volgen.

- per trein Naar Amersfoort. Vanaf station Amersfoort ver-  
trekken om de 15 min. afwisselend bus 5 en 6.  
Halte Neptunusplein uitstappen.

Toegang: De toegang is gratis voor leden en hun  
----- eventuele introducees.

Lunch: Van 12.30 tot 13.30 is er een gemeenschapper-  
----- lijke lunch. De kosten zijn f. 12,50 p/p.

Programma: 09.30 Ontvangst met koffie.  
----- 10.00 Opening door de voorzitter.  
10.15 Opstellen apparatuur. (Zorst U er wel  
voor dat U voldoende verlengsnoeren e.d.  
meeneemt om Uw apparatuur aan te sluiten  
11.00 Koffie pauze  
11.30 Aanvang demonstraties e.d.  
12.30 Lunch  
13.30 Benoeming Siep de Vries tot  
Ere-voorzitter  
Benoemins Hanny de Vries tot  
Ere-lid  
14.00 Vervolg demonstraties  
15.00 Theepauze  
15.30 Markt.  
16.00 Vervolg demonstraties  
17.00 Sluiting



UITDAGING KIM CLUB LUSTUM BLANKOMST



Deze pagina en de volgende kunt U met gemak uit het clubblad lichten, door er licht aan te trekken of door de nietjes even terug te buigen, de 4 middenpagina's (2 bladen) er uit te lichten en de nietjes weer om te buigen.

17.00 Stijling	17.00
16.00 Vervolg demonstraties	16.00
15.00 Luchtmusiek	15.00
14.00 Vervolg demonstraties	14.00
13.30 Lunch	13.30
13.00 Vervolg demonstraties	13.00
12.00 Koffie pauze	12.00
11.00 Vervolg demonstraties	11.00
10.00 Vervolg demonstraties	10.00
9.00 Vervolg demonstraties	9.00
8.00 Vervolg demonstraties	8.00
7.00 Vervolg demonstraties	7.00
6.00 Vervolg demonstraties	6.00
5.00 Vervolg demonstraties	5.00
4.00 Vervolg demonstraties	4.00
3.00 Vervolg demonstraties	3.00
2.00 Vervolg demonstraties	2.00
1.00 Vervolg demonstraties	1.00
0.00 Vervolg demonstraties	0.00

# KIM CLUB BIJeenKOMST

## UITNODIGING KIM CLUB BIJeenKOMST

Datum: 15 mei 1982.  
Lokatie: Gemeenschapshuis "De Zes Gehuchten",  
Papenvoort 10 te Geldrop (bij Eindhoven).

### Reisroute:

- per auto Vanaf E3 Belgie - Venlo

Afslag Geldrop, onder E3 door bij Sheraton hotel. 1e stoplicht l.a., na 800 meter bij kruising l.a., die weg 1 km volgen tot iets voorbij de kerk. Hiernaast is het.

Vanuit Eindhoven.

Rondweg, afslag Geldrop bij DAF. Bij stoplicht voor spoorwagviadukt in Geldrop r.a. Weg 1 km volgen tot iets voorbij de kerk. Hiernaast is het.

Vanuit Nuenen - Helmond.

Richting Eindhoven door Geldrop volgen. Bij 1e stoplicht na spoorwagviadukt l.a. Verder als bij bovenstaande route.

- per trein Stoptrein Eindhoven - Weert (Ieder half en  
heel uur vanuit Eindhoven). In Geldrop uit-  
stappen. Voetsanserstunnel naast station door.  
Deze weg (Toernooiveld) oversteken en r.a.  
slaan. L.a. (Hertosenlaan). R.a. (Gildestraat)  
L.a. (Diepenvaart). Aan het einde bij de  
kruising is het (rechts).

Toegang: De toegangsprijs bedraagt f. 25,- inclusief  
koffie, lunch en frisdrank.

Programma: 09.30 Ontvangst met koffie.  
10.00 Opening door de voorzitter.  
10.15 Gestructureerd programmeren met enige  
praktijkvoorbeelden (Fred Harthoorn).  
11.00 Koffiepauze  
11.30 Forum.  
12.00 Behandeling instructieset 6502.  
12.30 Lunch  
13.30 Opzet en uitwerking van een leden-  
administratiesysteem op een 6502  
systeem met 32K RAM (Ruud Uphoff).  
14.30 Markt.  
15.00 Theepauze.  
15.15 Demonstratie van meegebrachte apparatuur  
informele discussies etc. etc.  
17.00 Sluiting.



INHOUD VAN DE STUBBENBLAD

1. Inhoud van de Stubbenblad	1
2. Bericht van de voorzitter	2
3. Bericht van de penningmeester	3
4. Bericht van de secretaris	4
5. Bericht van de commissie	5
6. Bericht van de jury	6
7. Bericht van de wedstrijd	7
8. Bericht van de club	8
9. Bericht van de leden	9
10. Bericht van de vrienden	10
11. Bericht van de sponsors	11
12. Bericht van de medewerkers	12
13. Bericht van de vrijwilligers	13
14. Bericht van de ouders	14
15. Bericht van de kinderen	15
16. Bericht van de sport	16
17. Bericht van de natuur	17
18. Bericht van de kunst	18
19. Bericht van de literatuur	19
20. Bericht van de muziek	20
21. Bericht van de film	21
22. Bericht van de televisie	22
23. Bericht van de radio	23
24. Bericht van de pers	24
25. Bericht van de uitgeverij	25
26. Bericht van de distributie	26
27. Bericht van de verkoop	27
28. Bericht van de abonnementen	28
29. Bericht van de reclame	29
30. Bericht van de publiciteit	30
31. Bericht van de sponsoring	31
32. Bericht van de schenkingen	32
33. Bericht van de erfenissen	33
34. Bericht van de giften	34
35. Bericht van de overname	35
36. Bericht van de fusie	36
37. Bericht van de overname	37
38. Bericht van de fusie	38
39. Bericht van de overname	39
40. Bericht van de fusie	40
41. Bericht van de overname	41
42. Bericht van de fusie	42
43. Bericht van de overname	43
44. Bericht van de fusie	44
45. Bericht van de overname	45
46. Bericht van de fusie	46
47. Bericht van de overname	47
48. Bericht van de fusie	48
49. Bericht van de overname	49
50. Bericht van de fusie	50

Deze pagina en de vorige kunt U met gemak uit het clubblad lichten, door er licht aan te trekken of door de nietjes even terug te buigen, de 4 middenpagina's (2 bladen) er uit te lichten en de nietjes weer om te buigen.

```

6170: 02FD D0 35      BNE BL ;
6180: ;
6190: ; PION PROMOVEREN
6200: ;
6210: 02FF E0 08      BH CPXIM #08 ;
6220: 0301 90 04      BCC BI ;
6230: 0303 E0 38      CPXIM #38 ;
6240: 0305 90 38      BCC BM ;
6250: 0307 A6 05      BI LDX VAN ;
6260: 0309 B5 16      LDAZX BORD ;
6270: 030B 48          PHA ;
6280: 030C 09 03      ORAIM #03 ;
6290: 030E 05 03      STA PROM ;
6300: 0310 A5 03      BJZZ LDA PROM ;
6310: 0312 A6 05      LDX VAN ;
6320: 0314 95 16      STAZX BORD ;
6330: 0316 29 07      ANDIM #07 ;
6340: 0318 C9 07      CMPIM #07 ;
6350: 031A F0 15      BEQ BK ;
6360: 031C AA          TAX ;
6370: 031D A5 0B      LDA WRDE ;
6380: 031F 48          PHA ;
6390: 0320 18          CLC ;
6400: 0321 75 5D      ADCZX TSW ;
6410: 0323 85 0B      STA WRDE ;
6420: 0325 C6 0B      DEC WRDE ;
6430: 0327 20 D0 00   JSR MOVE ; JUNIOR : 20 C2 00
6440: 032A 68          PLA ;
6450: 032B 85 0B      STA WRDE ;
6460: 032D E6 03      INC PROM ;
6470: 032F D0 BD      BNE BJ ;
6480: 0331 68          BK PLA ;
6490: 0332 95 16      STAZX BORD ;
6500: 0334 4C 65 02   BL JMP BOK ;
6510: 0337 4C 62 02   BM JMP BOJ ;
6520: ;
6530: ; TEST OP SCHRAAK
6540: ;
6550: 033A A5 13      BN LDA EPS ;
6560: 033C 48          PHA ;
6570: 033D 84 13      STY EPS ;
6580: 033F A9 40      LDRAIM #40 ;
6590: 0341 85 05      STA VAN ;
6600: 0343 85 04      STA NAAR ;
6610: 0345 06 06      ASL NZET ;
6620: 0347 20 D0 00   JSR MOVE ; JUNIOR : 20 C2 00
6630: 034A A2 BF      LDXIM #BF ;
6640: 034C A5 06      LDA NZET ;
6650: 034E F0 79      BEQ BBB ;
6660: 0350 46 06      LSR NZET ;
6670: 0352 90 77      BCC BBC ;
6680: ;
6690: ; KORTE ROCHADE
6700: ;
6710: 0354 20 F0 03   JSR RC ;
6720: 0357 B0 30      BCS BONZ ;

```

# 6502

# AMUSEMENT

HJ00 91 6502 ASSEMBLER V2.0

PAGE 13

```
6730: 0359 E8          INX          ;
6740: 035A B5 16        LDAZX BORD  ;
6750: 035C 96 04        STX NAAR   ;
6760: 035E E8          INX          ;
6770: 035F 15 16        ORAZX BORD  ;
6780: 0361 D0 26        BNE BONZ   ;
6790: 0363 06 06        ASL NZET   ;
6800: 0365 20 D0 00     JSR MOVE   ; JUNIOR : 20 C2 00
6810: 0368 46 06        LSR NZET   ;
6820: 036A 90 1D        BCC BONZ   ;
6830: 036C A6 05        LDX VAN    ;
6840: 036E E8          INX          ;
6850: 036F E8          INX          ;
6860: 0370 9C 04        STX NAAR   ;
6870: 0372 E8          INX          ;
6880: 0373 B5 16        LDAZX BORD  ;
6890: 0375 94 16        STYZX BORD  ;
6900: 0377 CA          DEX         ;
6910: 0378 CA          DEX         ;
6920: 0379 95 16        STAZX BORD  ;
6930: 037B 20 D6 00     JSR NURD   ;
6940: 037E A6 05        LDX VAN    ;
6950: 0380 E8          INX          ;
6960: 0381 B5 16        LDAZX BORD  ;
6970: 0383 94 16        STYZX BORD  ;
6980: 0385 E8          INX          ;
6990: 0386 E8          INX          ;
7000: 0387 95 16        STAZX BORD  ;
7010:                   ;
7020:                   ; LANGE ROCHADE
7030:                   ;
7040: 0389 20 F0 03     BONZ JSR RC ;
7050: 038C 0A          ASLA       ;
7060: 038D B0 36        BCS BBA    ;
7070: 038F CA          DEX         ;
7080: 0390 86 04        STX NAAR   ;
7090: 0392 B5 16        LDAZX BORD  ;
7100: 0394 CA          DEX         ;
7110: 0395 15 16        ORAZX BORD  ;
7120: 0397 CA          DEX         ;
7130: 0398 15 16        ORAZX BORD  ;
7140: 039A D0 29        BNE BBA    ;
7150: 039C 06 06        ASL NZET   ;
7160: 039E 20 D0 00     JSR MOVE   ; JUNIOR : 20 C2 00
7170: 03A1 46 06        LSR NZET   ;
7180: 03A3 90 20        BCC BBA    ;
7190: 03A5 A6 05        LDX VAN    ;
7200: 03A7 CA          DEX         ;
7210: 03A8 CA          DEX         ;
7220: 03A9 86 04        STX NAAR   ;
7230: 03AB CA          DEX         ;
7240: 03AC CA          DEX         ;
7250: 03AD B5 16        LDAZX BORD  ;
7260: 03AF 94 16        STYZX BORD  ;
7270: 03B1 E8          INX          ;
7280: 03B2 E8          INX          ;
```

# 6502

# AMUSEMENT

HJ00 81 6502 ASSEMBLER V2.0 PAGE 14

```
7290: 03B3 E8          INX          ;
7300: 03B4 95 16        STAZX BORD  ;
7310: 03B6 20 B6 00      JSR  NURO   ;
7320: 03B9 A6 05        LDX  VAN    ;
7330: 03BB CA          DEX         ;
7340: 03BC B5 16        LDAZX BORD  ;
7350: 03BE 94 16        STYZX BORD  ;
7360: 03C0 CA          DEX         ;
7370: 03C1 CA          DEX         ;
7380: 03C2 CA          DEX         ;
7390: 03C3 95 16        STAZX BORD  ;
7400:                   ;
7410:                   ; EINDROUTINE CALC
7420:                   ;
7430: 03C5 A6 06        BBA  LDX  NZET ;
7440: 03C7 D6 02        BNE  BBC      ;
7450: 03C9 86 08        BBB  STX  ZWRD ;
7460: 03CB 68          BBC  PLA      ;
7470: 03CD 85 13        STA  EPS     ;
7480: 03CE 98          BBD  TYA     ;
7490: 03CF 38          SEC         ;
7500: 03D0 E5 08        SBC  ZWRD   ;
7510: 03D2 85 08        STA  ZWRD   ;
7520: 03D4 98          TYA         ;
7530: 03D5 38          SEC         ;
7540: 03D6 E5 08        SBC  WRDE   ;
7550: 03D8 85 08        STA  WRDE   ;
7560: 03DA A5 15        LDA  CKAZ   ;
7570: 03DC 49 FF        EORIN #FF   ;
7580: 03DE 85 15        STA  CKAZ   ;
7590: 03E0 C6 0C        DEC  NIVO   ;
7600: 03E2 A6 0C        LDX  NIVO   ;
7610: 03E4 E0 02        CPXIM #02   ;
7620: 03E6 D0 04        BNE  EXCL   ;
7630: 03E8 A5 06        LDA  NZET   ;
7640: 03EA 85 91        STA  OZET   ;
7650: 03EC 60          EXCL RTS     ;
7660:                   ;
7670: 03ED EA          NOP         ;
7680: 03EE EA          NOP         ;
7690: 03EF EA          NOP         ;
7700:                   ;
7710:                   ; INSPECTEREN ROCHADE
7720:                   ; CODE ROCC
7730:                   ;
7740: 03F0 A2 04        RC  LDXIM #04 ;
7750: 03F2 A5 0A        LDA  ROCC   ;
7760: 03F4 24 15        BIT  CKAZ   ;
7770: 03F6 50 04        BVC  ROI    ;
7780: 03F8 A2 3C        LDXIM #3C   ;
7790: 03FA 0A          ASLA       ;
7800: 03FB 0A          ASLA       ;
7810: 03FC 9A 05        ROI  STX  VAN ;
7820: 03FE 0A          ASLA       ;
7830: 03FF 60          RTS         ;
7840:                   ;
```

HJCO 81 6502 ASSEMBLER V2.0 PAGE 15

```

7850:                ; PATCHES VOOR COMPUTER SPEELT ZWART
7860:                ;
7870: 0010                ORG    $0010
7880:                ;
7890: 0010 84 F9                STY    DSP1    ;
7900: 0012 4C D4 17            JMP    CODE    ;
7910:                ;
7920:                ;
7930:                ;
7940:                ; SUBROUTINE OM OPENINGEN IN TE VOEREN
7950:                ; OPENINGEN IN TABEL OTAB1 OTABII EN OTABD
7960:                ;
7970:                ; PATCH 1798 JSR OPEN 20 31 05
7980:                ;
7990: 0500                ORG    $0500
8000:                ;
8010: 0500 00                PTR    =    $00    ; COMPUTER SPEELT ZWART : PTR = 01
8020:                ;
8030: 0501 AE 00 05  SROPEN LDX    PTR    ;
8040: 0504 BD 00 07  OII    LDAA   OTAB1    ;
8050: 0507 85 14                STA    HULP    ;
8060: 0509 29 BF                ANDIM $BF    ;
8070: 050B C5 02                CMP    ZETIII   ;
8080: 050D D0 2B                BNE    OIV     ;
8090: 050F BD 00 08            LDAA   OTABII   ;
8100: 0512 C5 01                CMP    ZETII    ;
8110: 0514 D0 24                BNE    OIV     ;
8120: 0516 BD 00 09            LDAA   OTABD    ;
8130: 0519 F0 29                BEQ    OVI     ;
8140: 051B AA                TAX     ;
8150: 051C BD 00 07  OIII   LDAA   OTAB1    ;
8160: 051F 85 14                STA    HULP    ;
8170: 0521 24 14                BIT    HULP    ;
8180: 0523 70 05                BUS    US     ;
8190: 0525 CD 04 17            CMP    TIMER   ; OF ROL TNER
8200: 0528 30 17                BMI    OV     ; OF BCC OV
8210: 052A 29 BF                US    ANDIM $BF    ;
8220: 052C 85 02                STA    ZETIII   ;
8230: 052E BD 00 08            LDAA   OTABII   ;
8240: 0531 85 01                STA    ZETII    ;
8250: 0533 BD 00 09            LDAA   OTABD    ;
8260: 0536 8D 00 05            STA    PTR     ;
8270: 0539 60                RTS     ;
8280: 053A 24 14                OIV   BIT    HULP    ;
8290: 053C 70 06                BUS    OVI     ;
8300: 053E E8                INX     ;
8310: 053F D0 C3                BNE    OII     ;
8320: 0541 E8                OV    INX     ;
8330: 0542 D0 D8                BNE    OIII    ;
8340: 0544 A2 00                OVI   LDXIM $00    ;
8350: 0546 8E 00 05            STX    PTR     ;
8360: 0549 4C 30 03            JMP    $0330 ;VERVOLG OORSPRONKELIJKE ROUTINE
8370:                ; TABELLEN MET OPENINGEN +
8380: 054C                OTAB1 *    $0700    ;
8390: 054C                OTABII *   $0800    ;
8400: 054C                OTABD *    $0900

```

HJCO 81 6502 ASSEMBLER V2.0 PAGE 16

```

8410:      ;
8420:      ; SUBROUTINE QUICK
8430:      ; VERSNELT REKENEN
8440:      ; KAN OVERAL WORDEN NEERGEZET, HIER ACHTER SROPEN
8450:      ;
8460:      ; PATCH IN SCHAAKPROGRAMMA +
8470:      ; 0202 JSR QUICK 20 00 05
8480:      ; 0205 NOP EA
8490:      ; 026E JMP OCKII 4C 58 05
8500:      ; 0271 NOP EA
8510:      ;
8520: 0580      ORG    $0580 ;
8530:      ;
8540: 0580 A9 C0      QUICK LDAM  #C0 ;
8550: 0582 85 08      STA    ZWRD ;
8560: 0584 A6 0C      LDX   NIVO ;
8570: 0586 A9 C8      LDAM  #C8 ;
8580: 0588 9D AC 05    STAX  TABZWD ;
8590: 058B 60          RTS ;
8600: 058C C9 41      OCKII CMPIM #41 ;
8610: 058E F0 19      BEQ   JUMP ;
8620: 0590 24 0D      BIT   CZA ;
8630: 0592 70 12      BUS  RET ;
8640: 0594 A6 0C      LDX   NIVO ;
8650: 0596 9D AC 05    STAX  TABZWD ;
8660: 0599 C9 BF      CMPIM #BF ;
8670: 059B F0 09      BEQ   RET ;
8680: 059D CA        DEX ;
8690: 059E 18        CLC ;
8700: 059F 7D AC 05    ADCAX TABZWD ;
8710: 05A2 F0 02      BEQ   RET ;
8720: 05A4 10 03      BPL  JUMP ;
8730: 05A6 4C 72 02    RET   JMP   $0272 ;
8740: 05A9 4C CE 03    JUMP  JMP   $03CE ;
8750:      ;
8760: 05AC C8        TABZWD =   $C8 ;
8770: 05AD C8        =   $C8 ;
8780: 05AE C8        =   $C8 ;
8790: 05AF C8        =   $C8 ;
8800: 05B0 C8        =   $C8 ;
8810: 05B1 C8        =   $C8 ;
8820: 05B2 C8        =   $C8 ;
8830: 05B3 C8        =   $C8 ;
8840:      ;
8850:      ; SNEL NIEUWE PARTIJ SPELEN MET CHESS IN GEHEUGEN
8860:      ; ZET DAARVOOR NA LADEN DEEL 1 EN VOOR LADEN DEEL 2
8870:      ; HET STUK 0000 - 0100 OP TAPE
8880:      ; VERVANG DAARBIJ 003B = 4C 03 00
8890:      ; NIEUWE PARTIJ SPELEN KAN NA LADEN VAN DEZE TAPE
8900:      ; EN STARTEN OP 0000

```

SYMBOL TABLE 3000 332A

BA	02BB	BB	02C5	BBA	03C5	BBB	03C9
BBC	03CB	BBD	03CE	BC	02CD	BD	02D2
BE	02EC	BF	02EE	BG	02F9	BH	02FF
BI	0307	BIA	011D	BIB	0127	BIC	012A
BID	0147	BIE	0156	BIF	0182	BIG	018A
BIH	0193	BIIZ	02A9	BII	0195	BIJ	019F
BIK	01A3	BIL	01A5	BIO	0100	BJ	00EE
BJZZ	0310	BK	0331	BL	0334	BM	0337
BN	033A	BO	0111	BOR	022B	BOC	022F
BOD	0240	BOE	0247	BOF	024B	BOG	0251
BOH	0255	BOI	0113	BOJ	0262	BOK	0265
BOL	0280	BOM	0288	BONZ	0389	BON	0291
BOO	0294	BORD	0016	BOZZ	0259	BOZ	0216
BRA	0154	BRC	00CA	BREAK	1780	BRK	0042
CALC	0200	CHESS	0000	CKAZ	0015	CODE	17D4
CZA	000D	CZET	0010	CZO	000E	DISP	17DC
DSP1	00F9	DSP11	00FA	DSP111	00FB	EPS	0013
EXCL	03EC	EXM	0110	FOUT	17D0	HFD	0092
HULP	0014	IN11	0026	IN1111	0033	INIT	0000
IN1V	0003	ITZ	0007	JUMP	05A9	LOAD	1873
MAX1	0011	MAX11	0012	MI	0096	M111	17A5
MON1	1C00	MOVE	0000	MV	17C2	MVRO	00B6
MVY	00BE	NAAR	0004	N1VO	000C	NZET	0006
O111	051C	O11	0504	O1V	053A	OTABD	0900
OTAB1	0700	OTAB11	0800	OV	0541	OVI	0544
OZET	0091	PROM	0003	PTCH	00C2	PTR	0500
PTZ	0056	PZET	000F	QCK11	058C	QUICK	0580
RC	03F0	RCT1	0084	RCT11	008A	RET	05A6
ROCO	000A	ROI	03FC	RSET	0040	SCANDS	1F1F
SROPEN	0501	STOP	003E	STUK	0009	TABZWD	05AC
TIMER	1704	TSW	005D	TZET	0064	VAN	0005
VECT	17FA	VS	052A	WIS	0090	WRDE	000B
YNIZ	0007	ZET1	0000	ZET11	0001	ZET111	0002
ZP	0000	ZWRD	0008		03C5		

## K I M S C H A A K P R O G R A M M A

- Ervaringen
- Informatie-uitwisseling
- Verdere ontwikkelingen

-----

Na de publikatie van het KIM-schaakprogramma in de KIM-Kenner nr. 11 zijn er een aantal leuke reacties gekomen. Deze reacties hebben geleid tot een aantal vragen over het programma, en een aantal uitbreidingen op het programma. Deze uitbreidingen zijn tot stand gekomen in samenwerking met Joop Tervooren, die veel van het testwerk voor zijn rekening heeft genomen, en die o.m. een vrij uitgebreide tabel met openingen voor het KIM-schaakprogramma heeft opgesteld.

1. Het schaakprogramma is opgenomen in de KIM-club programma-bibliotheek, zowel in de versie KIM speelt wit als in de versie KIM speelt zwart. Belangstellenden kunnen tijdens de KIM-club bijeenkomsten een copie maken van de ze programma's (zie regels voor het kopiëren uit de programma-bibliotheek).

2. Er is enige onduidelijkheid over welke aanpassingen moeten worden gepleegd om de KIM zwart te laten spelen. Voor alle duidelijkheid volgt hier een opsomming van de wijzigingen ten opzichte van het programma in de versie KIM speelt wit:

STY	DSP1	adres	00 10	84	F9	
JMP	CODE	"	00 12	4D	D4	17
CPY	X'3B'	"	00 C4	C0	3B	

Deze wijzigingen moeten worden aangebracht in deel 1 van het programma (dus vóórdat deel 2 is ingelezen). Hierna kan het gewijzigde deel 1 weer op de band worden gezet (adres 0000-0400); deel twee moet daar ongewijzigd achter worden geplaatst.

Wanneer men de wijzigingen vlak voor het spelen met de hand wil inbrengen kan dit nadat beide delen zijn ingelezen. (maar vóórdat de tweede keer op GO wordt gedrukt). In dat geval wel eerst de accumulator op nul zetten.

(00F3 00), voordat gestart wordt op 0000.

3. In de code is opgenomen een openingen-routine; op de volgende pagina's een tabel met openingen voor het KIM-schaakprogramma. Dit is uitgewerkt en getest door Joop Tervooren (er is wel meer dan 1 K geheugen voor nodig).

DATA /Adressen velden KIM- schaakprogramma T.K.

8	4e	4f	50	51	52	53	54	55
7	46	47	48	49	4a	4b	4c	4d
6	3e	3f	40	41	42	43	44	45
5	36	37	38	39	3a	3b	3c	3d
4	2e	2f	30	31	32	33	34	35
3	26	27	28	29	2a	2b	2c	2d
2	1e	1f	20	21	22	23	24	25
1	16	17	18	19	1a	1b	1c	1d

Veld-  
aanduiding  
schaakbord.

	A	B	C	D	E	F	G	H
1	84	83	85	86	82	85	83	84
2	80	80	80	80	80	80	80	80

stukken-code  
geplaatst op  
schaakbord::

00 = leeg veld::

7	81	C1						
8	C4	C3	C5	C6	C2	C5	C3	C4

A B C D E F G H

Stukken opzetten tijdens het spel::: Problemen???

RS- AD- veldadres - DA- stukcode - AD - en op beschreven  
manier zet invoeren.

Aanvulling schaakprogramma "Theo Kortekaas", KIM-kenner 11, voor diegene met meer dan 1K-geheugenruimte.

Subroutine die het mogelijk maakt openingen in te voeren, welke door de KIM worden aangeropen, willekeurig en/of kunnen worden aangewezen door de speler, middels een aanwijstabel die men zelf kan vullen, zie code.  
Tab1 = voor in code bepalen "afvelden" 1e vrije pagina in geheugen.  
Tab2 = " " " " "toevelden" 2e pagina=na vorige  
Tab3 = aanwijstabel voor de zetvolgorde 3e pagina= na vorige(opeenvolgend)  
settencode = veldnummering, van links onder naar rechts boven op schaakbord,  
= van 00 tem 3F, of in schaaktermen van A1 tem H8.

E2-E4, word dan OC, in tab1: en 1C in tab2: in tab3 komt dan de adresaanduiding, van tab1, waar de volgende zet staat aangegeven, in "LOW Order" aanduiden. OC dus voor e2, wil men alleen deze zet mogelijk maken, dan ipv "OC" ,4C aanduiden! In het eerste geval kan de KIM kiezen uit meerdere opeenvolgende zetten, zulke reeksen afsluiten met die ipv., of +4 links in de code.

Om KIM met zwart te laten spelen moet 66k worden gewijzigd  
0027 EA in 0027 00, alvorens te initiëren

@tab1 0700)  
 @tab2 0800) Dit alles kan overal elders in het geheugen,  
 @tab3 0900) mits aanpassingen voor de plaatsbepaling.

zetten adres TAB1 TAB2 TAB3

zetten	adres	TAB1	TAB2	TAB3
	00	00	85	01
e2-e4	1	0C	1C	05
d2-d4	2	0B	1B	60
c2-c4	3	0A	1A	98
b2-b4	4	49	19	D1
e7-e5	5	34	24	08
c7-c5	6	32	22	3C
e7-e6	7	74	2C	4e
o1-f3	8	06	15	0A
d2-d4	9	4B	1B	0B
b8-c6	A	79	2A	0C
e5xd4	B	64	1B	0E
f1-c4	C	05	1A	0F
f1-b5	D	45	21	10
d1-d4	E	43	1B	11
f8-c5	F	7D	22	12
a7-a6	10	70	28	13
b8-c6	11	79	2A	14
c2-c3	12	4A	12	15
b5-a4	13	61	18	16
d4-e3	14	5B	14	17
o8-f6	15	7E	2D	18
o8-f6	16	7E	2D	19
o8-f6	17	7E	2D	1A
d2-d4	18	4B	1B	1B
e1-o1	19	44	06	1C
b1-c3	1A	41	12	1D
e5xd4	1B	64	1B	1E
f8-e7	1C	7D	34	1F
f8-b4	1D	7D	19	20
c3xd4	1E	52	1B	21
f1-e1	1F	45	04	22
c1-d2	20	42	0B	23
c5-b4	21	62	19	24
b7-b5	22	71	21	25
e8-o8	23	7C	3E	26
b1-c3	24	41	12	27
a4-b3	25	58	11	28
e1-c1	26	44	02	29
f6xe4	27	6d	1C	2A
e8-o8	28	7C	3E	2B
f8-e8	29	7D	3C	2C
e1-o1	2A	44	06	2D
c2-c3	2B	4A	12	2E
f1-c4	2C	45	1A	2F

zetten adres TAB1 TAB2 TAB3

b4xc3	2D	59	12	30
d7-d5	2E	73	23	31
c6-a5	2F	6A	20	32
d4-d5	30	5B	23	33
e4xd5	31	5C	23	34
c4-d3	32	5A	13	35
c3-f6	33	52	2D	36
f6xd5	34	6D	23	37
d7-d5	35	73	23	38
f1-e1	36	45	04	39
f3xe5	37	55	24	3A
e4-e5	38	5C	24	3B
c6-e7	39	6A	34	00

Code tabel / Openingen

8	38	39	3A	3B	3C	3D	3E	3F
7	30	31	32	33	34	35	36	37
6	28	29	2A	2B	2C	2D	2E	2F
5	20	21	22	23	24	25	26	27
4	18	19	1A	1B	1C	1D	1E	1F
3	10	11	12	13	14	15	16	17
2	08	09	0A	0B	0C	0D	0E	0F
1	00	01	02	03	04	05	06	07
	A	B	C	D	E	F	G	H
							0	1

# 6502

# AMUSEMENT

zetten adres TAB1 TAB2 TAB3

c4xd5	A7	5A	23	A8	c1-b2	DA	42	09	DB
c6-d4	A8	6A	1B	A9	e8-o8	DB	7C	3E	DC
f3xd4	A9	55	1B	AA	f1-e2	DC	45	0C	DD
c5xd4	AA	62	1B	AB	c8-o4	DD	7A	1E	DE
e2-e3	AB	4C	1C	AC	e1-o1	DE	44	06	DF
d4-b6	AC	5B	29	AD	c7-c6	DF	72	2A	EO
o1-f3	AD	46	15	AE	a2-a4	EO	48	18	E1
o8-f6	AE	7E	2D	AF	b8-d7	E1	79	33	EO
d2-d4	AF	4B	1B	BO	c1-b2	E2	42	09	E3
c5xd4	BO	62	1B	B1	f7-f6	E3	75	2D	E4
f3xd4	B1	55	1B	B2	e2-e4	E4	4C	1C	E5
e7-e6	B2	74	2C	B3	f8xb4	E5	7D	19	E6
b1-c3	B3	41	12	B4	f1-c4	E6	45	1A	E7
f8-b4	B4	7D	19	B5	o8-e7	E7	7E	34	E8
d4-b5	B5	5B	21	B6	f2-f4	E8	4D	1D	E9
e8-o8	B6	7C	3E	B7	d7-d5	E9	73	23	EA
c1-f4	B7	42	1D	B8	e4xd5	EA	5C	23	EB
b8-c6	B8	79	2A	B9	b4-d6	EB	59	2B	EC
f4-d6	B9	5D	2B	BA	f4xe5	EC	5D	24	ED
b4xd6	BA	59	2B	BB	f6xe5	ED	6D	24	EE
b5xd6	BB	61	2B	BC	d1-15	EE	43	27	EF
f6-e8	BC	6D	3C	BD	e7-o6	EF	74	2E	FO
a2-a3	BD	48	10	BE	o1-f3	FO	46	15	F1
b7-b6	BE	71	29	BO	b8-d7	F1	79	33	F2
o1-f3	BF	46	15	CO	e1-o1	F2	44	06	F3
e5-e4	CO	64	1C	C1	e8-o8	F3	7C	3E	FO
f3-d4	C1	55	1B	C2	c1-b2	F4	42	09	F5
b8-c6	C2	79	2A	C3	e7-e6	F5	74	2C	F6
e2-e3	C3	4C	14	C4	b4-b5	F6	59	21	F7
c6xd4	C4	6A	1B	C5	a7-a6	F7	70	28	F8
e3xd4	C5	54	1B	C6	a2-a4	F8	48	18	F9
d8-f6	C6	7B	2D	C7	a6xb5	F9	68	21	FA
d4-d5	C7	5B	23	C8	a4xb5	FA	58	21	FB
f8-o5	C8	7D	22	C9	a8xa1	FB	78	00	FC
d1-e2	C9	43	0C	CA	b2xa1	FC	49	00	FD
f6-e7	CA	6D	34	CB	d7-d5	FD	73	23	FE
b1-c3	CB	41	12	CC	o1-f3	FE	46	15	FF
o8-f6	CC	7E	2D	CD	f8-e7	FF	7D	34	OO
d2-d3	CD	4B	13	CE					
e4xc3	CE	5C	13	CF					
e2xe7+	CF	4C	34	DO					
c5xe7	DO	62	34	DO					
d7-d5	D1	33	23	D4					
e7-e5	D2	34	24	E2					
o8-f6	D3	7E	2D	F4					
o1-f3	D4	46	15	D5					
o8-f6	D5	7E	2D	D6					
e2-e3	D6	4C	14	D7					
o7-o6	D7	76	2E	D8					
c2-c4	D8	4A	1A	D9					
f8-o7	D9	7D	36	DA					

zetten	adres	TAB1	TAB2	TAB3
c6xe5	3A	6A	24	00
d5-d4	3B	63	1B	00
g1-f3	3C	46	15	3D
d7-d6	3D	73	2B	3E
d2-d4	3E	4B	1B	3F
o5xd4	3F	62	1B	40
f3xd4	40	55	1B	41
o8-f6	41	7E	2D	42
b1-c3	42	41	12	43
a7-a6	43	70	28	44
o1-g5	44	42	26	45
e7-e6	45	74	2C	46
f2-f4	46	4D	1D	47
f8-e7	47	7D	34	48
d1-f3	48	43	15	49
d8-o7	49	7B	32	4A
e1-cl	4A	44	02	4B
b8-d7	4B	79	33	4C
o2-o4	4C	4E	1E	4D
b7-b5	4D	71	21	00
d2-d4	4E	4B	1B	4F
d7-d5	4F	73	23	50
b1-c3	50	41	12	51
o8-f6	51	7E	2D	52
cl-o5	52	42	26	53
f8-e7	53	7D	34	54
e4-e5	54	5C	24	55
f6-d7	55	6D	33	56
l2-l4	56	4F	1F	57
c7-c5	57	72	22	58
05xe7	58	66	34	59
e8xe7	59	7C	34	5A
f2-f4	5A	4D	1D	5B
d8-b6	5B	7B	29	5C
o1-f3	5C	46	15	5D
b8-c6	5D	79	2A	5E
o3-a4	5E	52	18	5F
b6-a5	5F	69	20	00
d7-d5	60	33	23	62
o8-f6	61	7E	2d	64
c2-c4	62	0A	1A	65
o1-f3	63	46	15	66
c2-c4	64	4A	1A	67
e7-e6	65	74	2C	68
o8-f6	66	7E	2D	69
e7-e6	67	74	2C	6A
b1-c3	68	41	12	6B
e2-e3	69	4C	14	6C
b1-c3	6A	41	12	6D
g8-f6	6B	7E	2D	7C
c8-f5	6C	7A	25	8A
f8-b4	6D	7D	19	6E
e2-e3	6E	4C	14	6F
e8-o8	6F	7C	3E	70

zetten	adres	TAB1	TAB2	TAB3
f1-d3	70	45	13	71
d7-d5	71	73	23	72
o1-f3	72	46	15	73
c7-c5	73	72	22	74
e1-o1	74	44	06	75
d5xc4	75	63	1A	76
d3xc4	76	53	1A	77
b8-d7	77	79	33	78
d1-e2	78	43	0C	79
b7-b6	79	71	29	7A
d4-d5	7A	5B	23	7B
b4xc3	7B	59	12	00
o1-f3	7C	46	15	7D
c7-c5	7D	72	22	7E
c4xd5	7E	5A	23	7F
c5xd4	7F	62	1B	80
d1xd4	80	43	1B	81
e6xd5	81	6C	23	82
e2-e4	82	4C	1C	83
b8-c6	83	79	2A	84
f1-b5	84	45	21	85
f6xe4	85	6D	1C	86
e1-gl	86	44	06	87
e4-f6	87	5C	2D	88
cl-o5	88	42	26	89
f8-e7	89	7D	34	0Q
f1-d3	8A	45	13	8B
e7-e6	8B	74	2C	8C
d3xf5	8C	53	25	8D
e6xf5	8D	6C	25	8E
d1-d3	8E	43	13	8F
d8-c8	8F	7B	3A	90
b2-b3	90	49	11	91
b8-a6	91	79	28	92
e1-o1	92	44	06	93
f8-e7	93	7D	34	94
c2-c4	94	4A	1A	95
e8-o8	95	7C	3E	96
b1-c3	96	41	12	97
c7-c6	97	72	2A	00
08-f6	98	3E	2D	9B
c7-c5	99	32	22	AD
e7-e5	9A	74	24	BF
b1-c3	9B	41	12	9C
e7-e5	9C	74	24	9D
o1-f3	9D	46	15	9E
b8-c6	9E	79	2A	9F
g2-g3	9F	4E	16	A0
f8-b4	A0	7D	19	A1
c3-d5	A1	52	23	A2
b4-c5	A2	59	22	A3
f1-o2	A3	45	0E	A4
d7-d6	A4	73	2B	A5
e1-o1	A5	44	06	A6
f6xd5	A6	6D	23	A7

01	e2-e4	01	e2-e4	01	e2-e4	01	e2-e4	01	e2-e4	02	d2-d4	02	d2-d4
05	e7-e5	05	e7-e5	05	e7-e5	06	c7-c5	07	e7-e6	60	d7-d5	60	d7-d5
08	01-f3	08	01-f3	09	d2-d4	3C	01-f3	4E	d2-d4	62	c2-c4	63	01-f3
0A	b8-c6	0A	b8-c6	0B	e5xd4	3D	d7-d6	4F	d7-d5	65	e7-e6	66	08-f6
0C	f1-c4	0D	f1-b5	0E	dlxd4	3E	d2-d4	50	b1-c3	68	b1-c3	69	e2-e3
0F	f8-c5	10	a7-a6	11	b8-c6	3F	c5xd4	51	o8-f6	6B	08-f6	6C	c8-f5
12	c2-c3	13	b5-a4	14	d4-e3	40	f3xd4	52	o1-05	7C	o1-f3	8A	f1-d3
15	08-f6	16	08-f6	17	08-f6	41	o8-f6	53	f8-e7	7D	c7-c5	8B	e7-e6
18	d2-d4	19	e1-01	1A	b1-c3	42	b1-c3	54	e4-e5	7E	c4xd5	8C	d3xf5
1B	e5xd4	1C	f8-e7	1D	f8-b4	43	a7-a6	55	f6-d7	7F	c5xd4	8D	e6xf5
1E	c3xd4	1F	f1-e1	20	o1-d2	44	c1-o5	56	h2-h4	80	dlxd4	8E	d1-d3
21	c5-b4+	22	b7-b5	23	e8-o8	45	e7-e6	57	c7-c5	81	e6xd5	8F	d8-c8
24	b1-c3	25	a4-b3	26	e1-c1	46	f2-f4	58	g5xe7	82	e2-e4	90	b2-b3
27	f6xe4	28	e8-o8	29	f8-e8	47	f8-e7	59	e8xe7	83	b8-c6	91	b8-ab
2A	e1-o1	2B	c2-c3	2C	f1-c4	48	d1-f3	5A	f2-f4	84	f1-b5	92	e1-o1
2D	b4xc3	2E	d7-d5	2F	c6-a5	49	d8-c7	5B	d8-b6	85	f6xe4	93	f8-e7
30	d4-d5	31	e4xd5	32	c1-d3	4A	e1-c1	5C	o1-f3	86	e1-o1	94	c2-c4
33	c3-f6	34	f6xd5	35	d7-d5	4B	b8-d7	5D	b8-c6	87	e4-f6	95	e8-o8
36	f1-e1	37	f3xe5	38	e4-e5	4C	g2-g4	5E	c3-a4	88	c1-o5	96	b1-c3
39	c6-e7	3A	c6xe5	3B	d5-d4	4D	b7-b5	5F	b6-a5	89	f8-e7	97	c7-c6

xx staan in de aanwijstabel

zetten staan in "CODE" in TAB1 en TAB2

02	d2-d4	03	c2-c4	03	c2-c4	03	c2-c4	04	b2-b4	04	b2-b4	04	b2-b4
61	o8-f6	98	o8-f6	99	c7-c5	9A	e7-e5	D1	d7-d5	D2	e7-e5	D3	o8-f6
64	c2-c4	9B	b1-c3	AD	o1-f3	BF	o1-f3	D4	o1-f3	E2	o1-b2	F4	c1-b2
67	e7-e6	9C	e7-e5	AM	o8-f6	CO	e5-e4	D5	o8-f6	E3	f7-f6	F5	e7-e6
6A	b1-c3	9D	o1-f3	AF	d2-d4	C1	f3-d4	D6	e2-e3	E4	e2-e4	F6	b4-b5
6D	f1-b4	9E	b8-c6	BO	c5xd4	C2	b8-c6	D7	g7-g6	E5	f8xb4	F7	a7-a6
6E	e2-e3	9F	g2-g5	B1	f3xd4	C3	e2-e3	D8	c2-c4	E6	f1-c4	F8	a2-a4
6F	e8-o8	AO	f8-b4	B2	e7-e6	C4	c6xd4	D9	f8-g7	E7	o8-e7	F9	a6xb5
70	f1-d3	A1	c3-d5	B3	b1-c3	C5	e3xd4	DA	c1-b2	E8	f2-f4	FA	a4xb5
71	d7-d5	A2	b4-c5	B4	f8-b4	C6	d8-f6	DB	e8-o8	E9	d7-d5	FB	a8xa1
72	o1-f3	A3	f1-g2	B5	d4-b5	C7	d4-d5	DC	f1-e2	EA	e4xd5	FC	b2xa1
73	c7-c5	A4	d7-d6	B6	e8-o8	C8	f8-c5	DD	c8-g4	EB	b4-d6	FD	d7-d5
74	e1-o1	A5	e1-o1	B7	c1-f4	C9	d1-e2	DE	e1-o1	EC	f4xe5	FE	o1-f3
75	d5xc4	A6	f6xd5	B8	b8-c6	CA	f6-e7	DF	c7-c6	ED	f6xe5	FF	f8-e7
76	f1xc4	A7	c4xd5	B9	f4-d6	CB	b1-c3	EO	a2-a4	EE	d1-h5+	xxxxx	
77	b8-d7	A8	c6-d4	BA	b4xd6	CC	o8-f6	E1	b8-d7	EF	e7-g6		
78	d1-e2	A9	f3xd4	BB	b5xd6	CD	d2-d3	E2	XXXXX	FO	o1-f3		
79	b7-b6	AA	c5xd4	BC	f6-e8	CE	e2xd3			F1	b8-d7		
7A	d4-d5	AB	e2-e3	BD	a2-a3	CF	d8-e7+			F2	e1-o1		
7B	b4xc3	AC	d4-b6	BE	b7-b6	DO	c5xe7			F3	e8-o8		

Zetten notatie volgens schaakprogramma T.Kortekaas.

Een 14-tal openingen om mee te beginnen, op deze wijze kan men een repertoire opbouwen, bv. per opening een aantal varianten tot de tabel vol is. Er zijn 256 regels beschikbaar, voor 127 zetten, per kleur.

# 6502 SYSTEM SOFTWARE

DISASM

KIM SOFTWARE LIBRARY

PAGE 01

```
0010: 0200      DISASM ORG    $0200  DISASSEMBLER FOR THE MCS-6502
0020:          *
0030:          * * * THIS PROGRAM IS A MODIFIED DISASSEMBLER,
0040:          *      ORIGINALLY WRITTEN TO RUN ON THE APPLE-
0050:          *      SYSTEM AND PUBLISHED IN 'INTERFACE AGE' OF
0060:          *      SEPTEMBER 1976.
0070:          *
0080:          *      IN KIM KENNER 3 OF DECEMBER 1977 WE HAVE
0090:          *      PUBLISHED THE PROGRAM, MODIFIED FOR THE
0100:          *      KIM-1, BY SIEP DE VRIES, OF WESTVRIES
0110:          *      COMPUTER CONSULTING B.V., P.O. BOX 20
0120:          *      AT OOSTZAAN (THE NETHERLANDS).
0130:          *
0140:          *      IN THE MEANTIME WE HAVE HAD SOME PATCHES TO
0150:          *      SUPPORT ALSO THE ROR INSTRUCTION IN THE VARIOUS
0160:          *      ADDRESSING MODES, FROM K. VAN NIEUWENHOVE,
0170:          *      CONSCIENCESTR. 50 4, E 3000 LEUVEN
0180:          *      (BELGIUM).
0190:          *      AND WE HAD DOZENS OF REQUESTS TO REPUBLISH
0200:          *      THIS PROGRAM.
0210:          *
0220:          *      THIS PROGRAM IS ORIGINALLY WRITTEN BY ALLEN
0230:          *      BAUM AND STEPHEN WOZNIAK, BOTH OF APPLE COMPUTER
0240:          *      INC., PALO ALTO, CA. U.S.A.
0250:          *
0260:          *      THIS PROGRAM MAY NOT BE SOLD WITHOUT PERMISSION
0270:          *      FROM THE AUTHORS.
0280:          *
0290:          *      THE PROGRAM IS A DISASSEMBLER, THAT RUNS ON
0300:          *      A STANDARD KIM-1 COMPUTER WITH A TELETYPE,
0310:          *      OR SIMILAR TERMINAL INTERFACED AS DESCRIBED IN
0320:          *      THE KIM-1 USER MANUAL, HOOKED UP TO IT.
0330:          *
0340:          *      THE PROGRAM OCCUPIES ALL OF THE LOCATIONS
0350:          *      0200-03FF PLUS PAGE ZERO 00EF-00FF. THESE
0360:          *      LOCATIONS ARE SHARED WITH THE KIM MONITOR.
0370:          *
0380:          *      THE FIRST LOCATION THAT HAS TO BE DISASSEMBLED
0390:          *      IS THE ONED WHOSE ADDRESS IS IN THE LOCATIONS
0400:          *      17F5-17F6. (LOWORDER IN 17F5) THE LAST LOCATION
0410:          *      PLUS ONE HAS TO BE IN LOCATIONS 17F7-17F8.
0420:          *      IT MAY OCCUR TO YOU, THAT THIS IS IDENTICAL TO
0430:          *      THE PROCEDURE FOR SAVING A PROGRAM ON A
0440:          *      CASSETTE.
0450:          *
0460:          *      THE STARTING ADDRESS OF THE PROGRAM IS 0200.
0470:          *      WHEN THE PROGRAM STOPS, THE KIM-DISPLAY (OR
0480:          *      TTY) SHOWS THE FIRST NON-DISASSEMBLED LOCATION.
0490:          *
0500:
```

# 6502 SYSTEM SOFTWARE

DISASM

KIM SOFTWARE LIBRARY

PAGE 02

```
0510:
0520: * THE REASONS FOR MODIFYING WERE THE FOLLOWING:
0530: * - THE APPLE MONITOR SAVES THE Y-REGISTER IN
0540: * THE PRINT-SUBROUTINE, WHILE THE KIM DOESN'T.
0550: * - THE ORIGINAL PROGRAM WAS A SET OF SUBROUTINES,
0560: * WHILE THIS VERSION IS A COMPLETE PROGRAM.
0570: * NEVERTHELESS ALL SUBROUTINES STILL EXIST,
0580: * EXCEPT FOR 'DSMEL'.
0590: * - PAGE ZERO LOCATIONS HAVE BEEN MOVED TO A MORE
0600: * CONVENIENT PLACE FOR THE KIM-1.
0610: * - FORMAT OF THE OUTPUT HAS BEEN CHANGED
0620: * SLIGHTLY.
0630:
0640: * - THE ROR INSTRUCTION IN ITS VARIOUS ADDRESSING
0650: * FORMATS HAS BEEN INCORPORATED, WHICH DID NOT
0660: * EXIST IN THE VERY EARLY VERSIONS OF THE 6502
0670: * PROCESSOR, WHEN THIS PROGRAM WAS FIRST
0680: * PUBLISHED.
0690:
0700: * IN ORDER TO OBTAIN THE PROPER TTY/SPEED, IT IS
0710: * RECOMMENDED TO START IT WITH A COMMAND FROM
0720: * THE TELETYPE, RATHER THAN A COMMAND THROUGH THE
0730: * KIM-KEYBOARD.
0740:
0750: * DEFINITIONS OF PAGE ZERO LOCATIONS:
0760: *
0770: 0200 FORMAT * $00EF
0780: 0200 LENGTH * FORMAT +01
0790: 0200 LMNEM * LENGTH +01
0800: 0200 RMNEM * LMNEM +01
0810: 0200 COUNT * RMNEM +03
0820: 0200 YSAVE * COUNT +01
0830: 0200 PCL * YSAVE +04
0840: 0200 PCH * PCL +01
0850: *
0860: * HIGH MEMORY LOCATIONS:
0870: *
0880: 0200 PLOW * $17F5
0890: 0200 PCHIG * PLOW +01
0900: 0200 PCLLST * PCHIG +01
0910: 0200 PCHLST * PCLLST +01
0920: *
0930: * SUBROUTINES IN KIM-1
0940: *
0950: 0200 MONIT * $104F
0960: 0200 CRLF * $1E2F
0970: 0200 PRTBYT * $1E3B
0980: 0200 DUTCH * $1EA0
0990:
1000:
```

# 6502 SYSTEM SOFTWARE

DISASM KIM SOFTWARE LIBRARY PAGE 03

```

1010:          *
1020:          *   BEGIN OF PROGRAM
1030:          *
1040: 0200 D8          BEGIN   CLD
1050: 0201 AD F5 17    LDA     PCLOW   ; SET PROGRAM COUNTER
1060: 0204 85 FA          STA     PCL
1070: 0206 AD F5 17    LDA     PCHIG
1080: 0209 85 FB          STA     PCH
1090: 020B 20 32 02    FURDER JSR     INSTDS   ; INTERPRET NEXT INSTRUCTION
1100: 020E 20 0F 03    JSR     PCADJ   ; ADJUST PROGRAM COUNTER
1110: 0211 85 FA          STA     PCL
1120: 0213 84 FB          STY     PCH
1130: 0215 CC F8 17    CPY     PCHLST  ; TEST IF DONE
1140: 0218 D0 F1          BNE     FURDER
1150: 021A CD F7 17    CMP     PCLLST
1160: 021D 90 EC          BCC     FURDER
1170: 021F 4C 4F 1C    ENDEM  JMP     MONIT
1180:          *
1190:          *   SUBROUTINE TO PRINT THE
1200:          *   BYTE THAT IS IN THE ACCUMULATOR
1210:          *
1220: 0222 84 F6          PRBYTE STY     YSAVE
1230: 0224 20 3B 1E          JSR     PRIBYT
1240: 0227 A4 F6          LDY     YSAVE
1250: 0229 60          RTS
1260:          *
1270:          *   SUBROUTINE TO PRINT THE ASCII
1280:          *   CHARACTER THAT IS IN THE ACCUMULATOR
1290:          *
1300: 022A 84 F6          CHAROU STY     YSAVE
1310: 022C 20 A0 1E          JSR     OUTCH
1320: 022F A4 F6          LDY     YSAVE
1330: 0231 60          RTS
1340:          *
1350:          *   DISASSEMBLE ONE INSTRUCTION
1360:          *
1370: 0232 20 F3 02    INSTDS JSR     PRPC     ; PRINT PCL/PCH
1380: 0235 A1 FA          LDAIX  PCL     ; GET OPCODE
1390: 0237 A8          TAY
1400: 0238 4A          LSRA          ; EVEN/ODD TEST
1410: 0239 90 00          BCC     IEVEN
1420: 023B 4A          LSRA          ; TEST B1
1430: 023C B0 17          BCS     ERR     ; XXXXXX11 INSTRUCTION INVALID
1440: 023E C9 22          CMPIM  #22
1450: 0240 F0 13          BEQ     ERR     ; 10001001 INSTRUCTION INVALID
1460: 0242 29 07          ANDIM  #07     ; MASK 3 BITS FOR ADDR MODE
1470: 0244 09 80          ORAIM  #80     ; ADD INDEXING OFFSET
1480: 0246 4A          IEVEN  LSRA          ; LSB INTO CARRY FOR
1490: 0247 AA          TAX
1500: 0248 BD 1E 03    LDRAAX MODE   ; INDEX INTO ADDRESSING MODE TABLE

```

# 6502 SYSTEM SOFTWARE

DISASM KIM SOFTWARE LIBRARY PAGE 04

```

1510: 024B B0 04      BCS   RTMODE      ; IF CARRY SET USE LSD FOR
1520: 024D 4A        LSRA                ; PRINT FORMAT INDEX
1530: 024E 4A        LSRA
1540: 024F 4A        LSRA                ; IF CARRY CLEAR USE MSD
1550: 0250 4A        LSRA
1560: 0251 29 0F      RTMODE ANDIM $0F   ; MASK FOR 4-BIT INDEX
1570: 0253 D0 04      BNE   GETFMT      ; 0 FOR INVALID OPCODES
1580: 0255 A0 80      ERR   LDYIM $80   ; SUBSTITUTE $80 FOR INVALID OPCD.
1590: 0257 A9 00      LDAIM $00         ; SET PRINT FORMAT INDEX TO 0
1600: 0259 AA        GETFMT TAX
1610: 025A 8D 52 07    LDAAX MODEB      ; INDEX INTO PRINT FORMAT TABLE
1620: 025D 85 EF      STA   FORMAT     ; SAVE FOR ADDR. FIELD FORMAT
1630: 025F 29 03      ANDIM $03        ; MASK 2-BIT LENGTH. 0=1-BYTE
1640: 0261 85 F0      STA   LENGTH     ; 1=2 BYTE, 2=3 BYTE
1650: 0263 99        TYA                ; OPCODE
1660: 0264 29 8F      ANDIM $8F        ; MASK IT FOR 1XXX1010 TEST
1670: 0266 AA        TAX                ; SAVE IT
1680: 0267 98        TYA                ; OPCODE AGAIN TO A
1690: 0269 A0 03      LDYIM $03
1700: 026A E0 8A      CPXIM $8A
1710: 026C F0 0B      BEQ   MNNDXC
1720: 026E 4A        MNNDXA LSRA
1730: 026F 80 00      BCC   MNNDXC     ; FORM INDEX INTO MNEMONIC TABLE
1740: 0271 4A        LSRA
1750: 0272 4A        MNNDXB LSRA     ; 1XXX1010 => 00101XXX
1760: 0273 00 20      ORAIM $20        ; XXXYYY01 => 00111XXX
1770: 0275 80        DEY                ; XXXYYY10 => 00110XXX
1780: 0276 D0 FA      BNE   MNNDXB     ; XXXYY100 => 00100XXX
1790: 0278 C8        INY                ; XXXXX000 => 000XXXXX
1800: 0279 08        MNNDXC DEY
1810: 027A D0 F2      BNE   MNNDXA
1820: 027C A8        PHA                ; SAVE MNEMONIC TABLE INDEX
1830: 027D B1 FA      PROP  LDAIY PCL
1840: 027F 20 22 02    JSR   PRBYTE
1850: 0282 A2 01      LDXIM $01
1860: 0284 20 05 03    PRDPBL JSR   PRBLB
1870: 0287 C4 F0      CPY   LENGTH     ; PRINT INSTR (1 TO 3 BYTES)
1880: 0289 C8        INY                ; IN A 12-CHARACTER FIELD
1890: 028A 90 F1      BCC   PROP
1900: 028C A2 03      LDXIM $03        ; CHAR COUNT FOR PRINTING MNEMONIC
1910: 028E C0 04      CPYIM $04
1920: 0290 90 F2      BCC   PRDPBL
1930: 0292 66        PLA                ; RECOVER MNEMONIC INDEX
1940: 0293 A8        TAY
1950: 0294 B9 70 03    LDAAY MNEML
1960: 0297 95 F1      STA   LMNEM      ; FETCH 3-CHAR MNEMONIC
1970: 0299 B9 EC 03    LDAAY MNEMR     ; PACKED IN 2 BYTES
1980: 029C 85 F2      STA   RMNEM
1990: 029E A9 00      PRMNA LDAIM $00
2000: 02A0 A0 05      LDYIM $05

```

# 6502 SYSTEM SOFTWARE

DISASM

KIM SOFTWARE LIBRARY

PAGE 05

```

2010: 02A2 06 F2      PRMNB ASL   RMNEM
2020: 02A4 26 F1      ROL   LMNEM      ; SHIFT 5 BITS OF CHAR IN ACCU
2030: 02A6 2A      ROLA      ; CLEARS CARRY
2040: 02A7 88      DEY
2050: 02A8 D0 F8      BNE   PRMNB
2060: 02AA 69 BF      ADCIM $BF      ; ADD 'X' OFFSET
2070: 02AC 20 2A 02   JSR   CHAROU   ; OUTPUT A CHAR OF MNEMONIC
2080: 02AF CA      DEX
2090: 02B0 D0 EC      BNE   PRMNA
2100: 02B2 20 04 03   JSR   PRBLNK   ; OUTPUT 1 BLANK
2110: 02B5 A2 05      LDXIM $05      ; COUNT FOR 6 PRINT-FORMAT BITS
2120: 02B7 E0 03      PRADRA CPXIM $03
2130: 02B9 D0 12      BNE   PRADRC   ; IF X=3 PRINT ADD VALUE
2140: 02BB A4 F0      LDY   LENGTH
2150: 02BD F0 0E      BEQ   PRADRC   ; NO PRINT IF LENGTH = 0
2160: 02BF A5 EF      PRADRB LDA   FORMAT
2170: 02C1 C9 E8      CMPIM $E8      ; HANDLE REL ADDRESSING MODE
2180: 02C3 B1 FA      LDAIY PCL      ; SPECIAL (PRINT TARGET ADDRESS)
2190: 02C5 B0 1C      BCS   RELADR   ; (NOT DISPLACEMENT)
2200: 02C7 20 22 02   JSR   PRBYTE   ; OUTPUT 1- OR 2-BYTE ADDRESS
2210: 02CA 86      DEY      ; MOST SIGNIFICANT BYTE FIRST
2220: 02CB D0 F2      BNE   PRADRB
2230: 02CD 06 EF      PRADRC ASL   FORMAT      ; TEST NEXT PRINT FORMAT BIT
2240: 02CF 90 0E      BCC   PRADRD   ; IF 0, DON'T PRINT
2250: 02D1 BD 6F 03      LDAAX CHARA  -01 ; CORRESPONDING CHARACTER
2260: 02D4 20 2A 02   JSR   CHAROU   ; OUTPUT 1 OR 2 CHARACTERS
2270: 02D7 BD 75 03      LDAAX CHARB  -01 ; (IF CHAR FROM CHARB IS 0)
2280: 02DA F0 03      BEQ   PRADRD   ; DON'T OUTPUT IT
2290: 02DC 20 2A 02   JSR   CHAROU
2300: 02DF CA      PRADRD DEX
2310: 02E0 D0 D5      BNE   PRADRA
2320: 02E2 60      RTS      ; RETURN IF 6 FORMAT BITS
2330: 02E3 20 11 03   RELADR JSR   PCADJB      ; PCL, PCH + DISPL + 1 TO A, Y
2340: 02E6 AA      TAX
2350: 02E7 E8      INX
2360: 02E9 D0 01      BNE   PRNTYX   ; +1 TO X, Y
2370: 02EA C8      INY
2380: 02EB 98      PRNTYX TYA
2390: 02EC 20 22 02   PRNTAX JSR   PRBYTE      ; PRINT TARGET ADDRESS OF BRANCH
2400: 02EF 8A      PRNTAX TXA      ; AND RETURN
2410: 02F0 4C 22 02   JMP   PRBYTE
2420: 02F3 A9 9D      PRPC  LDAIM $9D
2430: 02F5 20 2F 1E   JSR   CRLF     ; OUTPUT CARRIAGE RETURN
2440: 02F8 A5 FB      LDA   PCH
2450: 02FA A6 FA      LDX   PCL
2460: 02FC 20 EC 02   JSR   PRNTAX   ; OUTPUT PCH AND PCL
2470: 02FF A9 20      LDAIM $20
2480: 0301 20 2A 02   JSR   CHAROU
2490: 0304 A2 01      PRBLNK LDXIM $01      ; BLANK COUNT
2500: 0306 A9 20      PRBLB LDAIM '

```

# 6502

# SYSTEM SOFTWARE

DISASM

KIM SOFTWARE LIBRARY

PAGE 06

```

2510: 0308 20 2A 02 PRBLC JSR CHAROU ; OUTPUT A BLANK
2520: 030B CA DEX
2530: 030C D0 F8 BNE PRBLB ; LOOP UNTIL COUNT IS ZERO
2540: 030E 60 RTS
2550: 030F 9E F0 PCADJ LDA LENGTH ; 0=1 BYTE, 1=2 BYTE, 2=3 BYTE
2560: 0311 38 PCADJB SEC
2570: 0312 94 F8 PCADJD LBY PCH
2580: 0314 8A TAX ; TEST DISPLACEMENT SIGN FOR REL.
2590: 0315 10 01 BPL PCADJD ; ADDRESSING MODE
2600: 0317 88 DEY ; EXTEND NEG BY DECREMENTING PCH
2610: 0318 55 FA PCADJD ADC PCL
2620: 031A 50 01 BCC RTSA ; PCL+LENGTH (OR DISPLACEMENT) +1
2630: 031C 08 INY
2640: 031D 50 RTSR RTS
2650:
2660: 031E 40 NDBE = $40
2670: 031F 02 = $02
2680: 0320 45 = $45
2690: 0321 07 = $07
2700: 0322 D0 = $D0
2710: 0323 08 = $08
2720: 0324 40 = $40
2730: 0325 09 = $09
2740: 0326 30 = $30 ; XXXXXX10 INSTRUCTIONS
2750: 0327 22 = $22
2760: 0328 45 = $45 ; 2=0, LEFT NIBBLE
2770: 0329 33 = $33 ; 7=1, RIGHT NIBBLE
2780: 032A 00 = $00
2790: 032B 08 = $08
2800: 032C 40 = $40
2810: 032D 09 = $09
2820: 032E 40 = $40
2830: 032F 02 = $02
2840: 0330 45 = $45
2850: 0331 33 = $33
2860: 0332 D0 = $D0
2870: 0333 08 = $08
2880: 0334 40 = $40
2890: 0335 09 = $09
2900: 0336 40 = $40
2910: 0337 02 = $02
2920: 0338 45 = $45
2930: 0339 83 = $83
2940: 033A D0 = $D0
2950: 033B 08 = $08
2960: 033C 40 = $40
2970: 033D 09 = $09
2980: 033E 00 = $00
2990: 033F 22 = $22
3000: 0340 44 = $44

```

# 6502 SYSTEM SOFTWARE

DISASM KIM SOFTWARE LIBRARY PAGE 07

3010:	0341	33	=	\$33	
3020:	0342	D0	=	\$D0	
3030:	0343	8C	=	\$8C	
3040:	0344	44	=	\$44	
3050:	0345	00	=	\$00	
3060:	0346	11	=	\$11	
3070:	0347	22	=	\$22	
3080:	0348	44	=	\$44	
3090:	0349	33	=	\$33	
3100:	034A	D0	=	\$D0	
3110:	034B	8C	=	\$8C	
3120:	034C	44	=	\$44	
3130:	034D	9A	=	\$9A	
3140:	034E	10	=	\$10	
3150:	034F	22	=	\$22	
3160:	0350	44	=	\$44	
3170:	0351	33	=	\$33	
3180:	0352	D0	=	\$D0	
3190:	0353	08	=	\$08	
3200:	0354	40	=	\$40	
3210:	0355	09	=	\$09	
3220:	0356	10	=	\$10	
3230:	0357	22	=	\$22	
3240:	0358	44	=	\$44	
3250:	0359	33	=	\$33	
3260:	035A	D0	=	\$D0	
3270:	035B	08	=	\$08	
3280:	035C	40	=	\$40	
3290:	035D	09	=	\$09	
3300:	035E	62	=	\$62	
3310:	035F	13	=	\$13	; YYXXXZ01 INSTRUCTIONS
3320:	0360	78	=	\$78	
3330:	0361	A9	=	\$A9	
3340:	0362	00	=	\$00	; ERROR
3350:	0363	21	=	\$21	; IMMEDIATE
3360:	0364	81	=	\$81	; ZERO PAGE
3370:	0365	82	=	\$82	; ABSOLUTE
3380:	0366	00	=	\$00	; IMPLIED
3390:	0367	00	=	\$00	; ACCUMULATOR
3400:	0368	59	=	\$59	; INDEXED INDIRECT
3410:	0369	4D	=	\$4D	; INDIRECT INDEXED
3420:	036A	91	=	\$91	; ZEROPAGE INDEXED BY REGISTER X
3430:	036B	92	=	\$92	; ABSOLUTE INDEXED BY REGISTER X
3440:	036C	86	=	\$86	; ABSOLUTE INDEXED BY REGISTER Y
3450:	036D	4A	=	\$4A	; ABSOLUTE
3460:	036E	85	=	\$85	; ZEROPAGE INDEXED BY REGISTER Y
3470:	036F	9D	=	\$9D	; RELOCATABLE
3480:	0370	2C	=	' ,	
3490:	0371	29	=	' )	
3500:	0372	2C	=	' ,	

# 6502 SYSTEM SOFTWARE

DISASM KIM SOFTWARE LIBRARY TO 0000 PAGE 08 YP0011 00000000 KIM H00010

3510:	0373	23	=	'#					
3520:	0374	28	=	'(					
3530:	0375	24	=	'\$					
3540:	0376	59	CHARB	=	'Y				
3550:	0377	00	=	\$00					
3560:	0378	58	=	'X					
3570:	0379	24	=	'\$					
3580:	037A	24	=	'\$					
3590:	037B	00	=	\$00					
3600:	037C	10	MNEML	=	\$1C				; XXXXX000 INSTRUCTIONS
3610:	037D	8A	=	\$8A					
3620:	037E	10	=	\$1C					
3630:	037F	23	=	\$23					
3640:	0380	5D	=	\$5D					
3650:	0381	8B	=	\$8B					
3660:	0382	1B	=	\$1B					
3670:	0383	A1	=	\$A1					
3680:	0384	9D	=	\$9D					
3690:	0385	9A	=	\$9A					
3700:	0386	1D	=	\$1D					
3710:	0387	23	=	\$23					
3720:	0388	9D	=	\$9D					
3730:	0389	8B	=	\$8B					
3740:	038A	1D	=	\$1D					
3750:	038B	A1	=	\$A1					
3760:	038C	00	=	\$00					
3770:	038D	29	=	\$29					
3780:	038E	19	=	\$19					
3790:	038F	AE	=	\$AE					
3800:	0390	69	=	\$69					
3810:	0391	A8	=	\$A8					
3820:	0392	19	=	\$19					
3830:	0393	23	=	\$23					
3840:	0394	24	=	\$24					
3850:	0395	53	=	\$53					
3860:	0396	1B	=	\$1B					
3870:	0397	23	=	\$23					
3880:	0398	24	=	\$24					
3890:	0399	53	=	\$53					
3900:	039A	19	=	\$19					
3910:	039B	A1	=	\$A1					
3920:	039C	00	=	\$00					; XXXYY100 INSTRUCTIONS
3930:	039D	1A	=	\$1A					
3940:	039E	5B	=	\$5B					
3950:	039F	5B	=	\$5B					
3960:	03A0	A5	=	\$A5					
3970:	03A1	59	=	\$59					
3980:	03A2	24	=	\$24					
3990:	03A3	24	=	\$24					
4000:	03A4	AE	=	\$AE					; 1XXX1010 INSTRUCTIONS

# 6502 SYSTEM SOFTWARE

DISASM KIM SOFTWARE LIBRARY

PAGE 09

4010:	03A5	AE	=	\$AE	
4020:	03A6	A8	=	\$A8	
4030:	03A7	AD	=	\$AD	
4040:	03A8	29	=	\$29	
4050:	03A9	00	=	\$00	
4060:	03AA	7C	=	\$7C	
4070:	03AB	00	=	\$00	
4080:	03AC	15	=	\$15	; XXXYYY10 INSTRUCTIONS
4090:	03AD	9C	=	\$9C	
4100:	03AE	6D	=	\$6D	
4110:	03AF	9C	=	\$9C	
4120:	03B0	A5	=	\$A5	
4130:	03B1	69	=	\$69	
4140:	03B2	29	=	\$29	
4150:	03B3	53	=	\$53	
4160:	03B4	84	=	\$84	; XXXYYY01 INSTRUCTIONS
4170:	03B5	13	=	\$13	
4180:	03B6	34	=	\$34	
4190:	03B7	11	=	\$11	
4200:	03B8	A5	=	\$A5	
4210:	03B9	69	=	\$69	
4220:	03BA	23	=	\$23	
4230:	03BB	A0	=	\$A0	
4240:	03BC	D8	MNEMR =	\$D8	; XXXXX000 INSTRUCTIONS
4250:	03BD	62	=	\$62	
4260:	03BE	5A	=	\$5A	
4270:	03BF	48	=	\$48	
4280:	03C0	26	=	\$26	
4290:	03C1	52	=	\$52	
4300:	03C2	94	=	\$94	
4310:	03C3	88	=	\$88	
4320:	03C4	54	=	\$54	
4330:	03C5	44	=	\$44	
4340:	03C6	08	=	\$08	
4350:	03C7	54	=	\$54	
4360:	03C8	68	=	\$68	
4370:	03C9	44	=	\$44	
4380:	03CA	E8	=	\$E8	
4390:	03CB	94	=	\$94	
4400:	03CC	00	=	\$00	
4410:	03CD	B4	=	\$B4	
4420:	03CE	08	=	\$08	
4430:	03CF	84	=	\$84	
4440:	03D0	74	=	\$74	
4450:	03D1	B4	=	\$B4	
4460:	03D2	28	=	\$28	
4470:	03D3	6E	=	\$6E	
4480:	03D4	74	=	\$74	
4490:	03D5	F4	=	\$F4	
4500:	03D6	CC	=	\$CC	

# 6502 SYSTEM SOFTWARE

DISASM

KIM SOFTWARE LIBRARY

PAGE 10

4510:	03D7	4A	=	%4A	
4520:	03D8	72	=	%72	
4530:	03D9	F2	=	%F2	
4540:	03DA	A4	=	%A4	
4550:	03DB	8A	=	%8A	
4560:	03DC	00	=	%00	; XXXYY100 INSTRUCTIONS
4570:	03DD	AA	=	%AA	
4580:	03DE	A2	=	%A2	
4590:	03DF	A2	=	%A2	
4600:	03E0	74	=	%74	
4610:	03E1	74	=	%74	
4620:	03E2	74	=	%74	
4630:	03E3	72	=	%72	
4640:	03E4	44	=	%44	; 1XXX1010 INSTRUCTIONS
4650:	03E5	58	=	%58	
4660:	03E6	B2	=	%B2	
4670:	03E7	32	=	%32	
4680:	03E8	B2	=	%B2	
4690:	03E9	00	=	%00	
4700:	03EA	22	=	%22	
4710:	03EB	00	=	%00	
4720:	03EC	1A	=	%1A	; XXXYYY10 INSTRUCTIONS
4730:	03ED	1A	=	%1A	
4740:	03EE	20	=	%20	
4750:	03EF	26	=	%26	
4760:	03F0	72	=	%72	
4770:	03F1	72	=	%72	
4780:	03F2	88	=	%88	
4790:	03F3	08	=	%08	
4800:	03F4	C4	=	%C4	; XXXYYY01 INSTRUCTIONS
4810:	03F5	CA	=	%CA	
4820:	03F6	26	=	%26	
4830:	03F7	48	=	%48	
4840:	03F8	44	=	%44	
4850:	03F9	44	=	%44	
4860:	03FA	A2	=	%A2	
4870:	03FB	08	=	%08	

## PASCAL als assembler hulpmiddel

=====

H.J.C. Otten  
Ottoborrengoed 33  
3871 MJ Hoevelaken

### Inleiding

=====

Uit andere artikelen in dit nummer zult U wel een indruk hebben gekregen van gestructureerd programmeren.

Het modulair opzetten van programma's, uitwerken van programma-structuren volgens de top-down methode etc hoort daarbij.

Het eigenlijke coderen in assembler is de laatste en niet de meest ingewikkelde fase. Helaas is in de assembler listing in het algemeen niet meer terug te vinden wat de gebruikte programma-structuur is. Assembler maakt een gestructureerde layout moeilijk en losse commentaar regels helpen ook niet erg.

### Basis programma structuren

=====

PASCAL maakt het zeer fraai mogelijk programma structuren met een nette layout zichtbaar te maken. Ontwikkelen van programma's gaat ook veel meer gestructureerd in PASCAL omdat we de basis structuren al in huis hebben bij PASCAL.

De basis programma besturings structuren ( o.a. in PASCAL ) zijn :

#### 1. De opeenvolging

-----

```
statement
statement
.
.
statement
```

Dit is de basis structuur van elk programma.

#### 2. De keuze

-----

IF conditie THEN statements ELSE other statements

Afhankelijk van de conditie ( voorwaarde ) wordt een serie statements al of niet sequentieel uitgevoerd.

#### 3. De herhaling of de LOOP

+++++

```
WHILE conditie DO
BEGIN
statements
END
```

Zolang de conditie waar is wordt een aantal statements iedere keer herhald.  
Als de conditie de eerste keer al niet waar is wordt de loop nooit doorlopen.

#### Blok structuur =====

Herk op dat met statements een verzameling instructies wordt bedoeld. BASIC kenners zijn gewend aan IF .. THEN met maar een instructie. PASCAL laat verzamelingen instructies toe, blokken genoemd. Om aan te geven dat instructies bij elkaar horen worden ze omgeven door BEGIN ... END haken. Deze blok structuur is kenmerkend voor blok-gestructureerde talen zoals PASCAL, ALGOL, ADA etc.

#### Afgeleide structuren =====

Van de basis structuren zijn een paar structuren afgeleid die vaak handig zijn :

#### 4. Herhaling met eindconditie

REPEAT

statements

UNTIL conditie

Zolang aan de conditie wordt voldaan worden de statements herhaald. De loop wordt minstens eenmaal doorlopen.

#### 5. De meervoudige keuze .

CASE keuzevariabele #F

waarde1 : procedure1

waarde2 : procedure2

.

.

waardeN : procedureN

Afhankelijk van de waarde van de keuzevariabele wordt een van de procedures uitgevoerd. Een procedure is een soort subroutine.

Een ingangs- en een uitgangspunt  
=====

Al deze programma structuren hebben de eigenschap dat ze slechts een ingangspunt en slechts een uitgangspunt hebben. Elk goed programma deel voldoet aan deze eis. Het gebruik van meerdere ingangspunten

en eindpunten in een routine maken de werking uitermate onoverzichtelijk. Door het nesten van programma structuren kunnen we elke gewenste structuur bouwen met nog steeds dezelfde eis. Het is wiskundig bewezen dat elke ongestructureerde programmastructuur om te zetten is tot een gestructureerd programma dat alleen de structuren 1 t/m 3 bevat.

## Assembler en structuur =====

In assembler zijn de bovenstaande goed overzichtelijke programma structuren niet aanwezig. De enige besturing die we hebben is de jump en branch. Deze sprong ( GØTØ in vaktaal ) opdracht behoort duidelijk niet tot de basis structuren en in PASCAL hebben we de GØTØ bijna nooit nodig. Toch kunnen we in assembler de gestructureerde programma controle gebruiken met wat hulpmiddelen.

## PASCAL als documentatie =====

Allereerst moeten we de gebruikte structuur aangeven. Daartoe gebruiken we het commentaar. Een programma deel wordt in PASCAL code geschreven. Dit wordt vervolgens als leidraad voor het coderen in assembler gebruikt. Het PASCAL programma staat rechts als commentaar, de niet belangrijke ( voor de mens althans ) assembler code links. We moeten alleen de PASCAL basis structuren in assembler vertalen. De JUMP en BRANCH hebben we dus wel nodig. Het voordeel van op deze manier werken is dat het programma in PASCAL is geschreven en gedocumenteerd. Het is goed leesbaar en elke basis structuur is apart in assembler gecodeerd. Is er met een stukje programma iets mis, dan geeft de PASCAL code aan wat er in dat programma deel gebeurt. De fout kan dan makkelijk worden hersteld omdat de werking van dat stukje programma meteen duidelijk is. Door in de assembler code letterlijk de structuur van het PASCAL programma deel te volgen kunnen we uitstekend gestructureerd programmeren in assembler met een prettig leesbaar overzichtelijk resultaat. PASCAL laat ons toe de basis structuren te gebruiken en het letterlijk vertalen verhindert het ongecontroleerd rondspringen in assembler. Het ontwikkelen en debuggen van programma's verloopt veel sneller en gemakkelijker.

## Voorbeelden =====

Van elke hierboven beschreven basis structuur zal een voorbeeld worden gegeven hoe de PASCAL code te vertalen naar assembler.

### 1. De opeenvolging -----

Dit is de eenvoudigst te vormen structuur want deze structuur komt overeen met de manier waarop het programma wordt opgeschreven :  
instructie na instructie.

Een voorbeeld is niet nodig en kan ieder zelf bedenken.

## 2. De keuze

Het volgende programma deel is een voorbeeld van een IF .. THEN .. ELSE structuur.

```

CMP  #$01      ;   IF ( A = 1 ) THEN
BNE  NOTON     ;
;
INX           ;   X := X + 1
JMP  END       ;
;
ELSE
;
NOTON DEX     ;   X := X - 1
;
END  .....   ;

```

## 2. De herhaling of de LØØP

Bij de WHILE loop wordt eerst een conditie getest en alleen als de conditie waar is de loop ingegaan. Aan het eind van het loop blok wordt teruggegaan naar het testen van de conditie.

```

LDX  #$10     ;   X := 10
;
LØØP BEQ  END  ;   WHILE ( X > 0 ) DØ
;           ;   BEGIN
INX           ;   Y := Y + 1
DEX           ;   X := X - 1
;
JMP  LØØP     ;   END
;
END  ... ..   ;

```

## 4. De afgeleide LØØP : REPEAT ... UNTIL conditie

Het voorbeeld van de WHILE .. LØØP kan ook als REPEAT ... UNTIL loop worden gegeven. De conditie wordt nu pas aan het eind van de loop getest.

```

LDX  #$10     ;   X := 10
;
LØØP INY     ;   REPEAT
DEX           ;   Y := Y + 1
;           ;   X := X - 1
;
BNE  LØØP     ;   UNTIL ( X = 0 )
;
.....       ;

```

Let op de veranderde conditie in het PASCAL programma, we moeten nu wachten tot X nul wordt omdat we de loop al doorlopen hebben.

## 5. De meervoudige keuze : CASE

Het volgende programma deel is een voorbeeld van een meervoudige keuze. Het is het hoofddeel van een eenvoudige monitor, die de commando's SAVE, LOAD en GO uitvoert na intypen van respectievelijk de letters S, L en G. De subroutines SAVE, LOAD en GO handelen verder het commando af.

```

; CASE char : character of
;
CMP #'S' ; S : SAVE
BNE SL0AD ;
JSR SAVE ;
JMP ENDSSEL ;
SL0AD CMP #'L' ; L : LOAD
BNE SGO ;
JSR L0AD ;
JMP ENDSSEL ;
SGO CMP #'G' ; G : GO
BNE ENDSSEL ;
JSR GO ; END
ENDSEL ... ;

```

Pseudo-code  
=====

Vrijwel alle PASCAL statements zijn in assembler te vertalen. De voorbeelden bevatten eenvoudige variabelen en condities maar met wat fantasie zijn ingewikkelder variabelen en condities ook eenvoudig. Daarbij komt het precies statement voor statement vertalen uitstekend van pas omdat we ons kunnen concentreren op dat ene statement.

Niet altijd zullen we aan standaard PASCAL genoeg hebben. Het werken met SHIFT commando's heeft geen PASCAL equivalent en wordt toch vaak met vrucht gebruikt. Ook bitmanipulaties zijn met PASCAL niet te beschrijven. We zijn echter niet gebonden door de syntax regels van PASCAL en kunnen omschrijvingen toevoegen. Dan spreken we ook niet meer over PASCAL maar over pseudo-code. Het gebruik van pseudo-code heb ik al eens gedemonstreerd in 6502 KENNER 19 : Tape Copy. Bestudeert U eens dat programma en let U erop dat ik niet strikt de PASCAL code vertaal. Dat is veroorzaakt door het eerst later pas voorzien van pseudo-code als commentaar. Beter was geweest de volgorde om te draaien : eerst pseudo-code en daarna assembler-code.

## Voorbeelden van pseudo-code

=====

In een programma wordt vaak gerefereerd aan de inhoud van geheugenlokaties. In pseudo-code is dit te implementeren door het geheugen te zien als een array MEM, de array-elementen zijn bytes, de array index een 16 bits adres.

Het laden bijvoorbeeld van de accumulator met de inhoud van een absoluut adres kunnen we in assembler en overeenkomstige pseudo-code schrijven als :

```
LDA ADDR ; A := MEM ( ADDR )
```

ADDR is een 16 bits adres hierbij.

Het indirect lezen van data uit het geheugen kunnen we aangeven door het pointertype : MEM ( ADDR ) . We bedoelen daarmee dat we de data halen uit de geheugenlokatie waarbij ADDR het adres van die geheugenlokatie bevat. Pointer staat voor wijzer, het adres bevat een adres.

Een voorbeeld in 6502 code is :

```
LDA ( ADDR, X ) ; A := MEM ( ( ADDR + X ) ↑ )
```

De 6502 kan alleen geïndexeerd indirect adresseren, vandaar de X index. Deze adressering heet indexed indirect, wat in de pseudo-code meteen duidelijk wordt : eerst wordt de index X bij het adres ADDR opgeteld en daarna indirect de data opgehaald.

De andere indirecte adresseringsmethode van de 6502 kan als volgt worden geschreven :

```
LDA ( ADDR ), Y ; a := MEM ( ( ADDR ) ↑ + Y )
```

Eerst wordt het basis adres uit ADDR gehaald en dit basis adres wordt opgehoogd met Y om het het adres waar we de data uit ophalen te krijgen. Deze adresseringsmethode heet indirect indexed en uit de pseudo-code blijkt dit meteen : eerst wordt het indirecte adres bepaald en daar een index bij opgeteld.

De moeilijkste adresserings methoden hebben we nu al gehad en de gewone geïndexeerde adresseringsmethoden zijn eenvoudig, bijvoorbeeld :

```
LDA ADDR,X ; A := MEM ( ADDR + X )
```

De accumulator wordt hierbij gelezen met de inhoud van het adres = ADDR + X.

Mocht U geïnteresseerd geraakt zijn in het gebruik van PASCAL als ontwikkelingshulpmiddel bij assembler, dan raad ik U aan hiermee te experimenteren. Daarvoor hoeft U zelfs niet over een assembler te beschikken, op papier werkt het ook heel aardig. Misschien denkt U dat het veel tijd gaat kosten om een programma zo te ontwikkelen maar het tegendeel is in de praktijk waar gebleken. Programma's draaien sneller en betrouwbaarder en zijn veel beter te lezen en dat is toch wat we willen.

#### Literatuur

=====

Het gebruik van pseudo-code als assembler hulp wordt uitgebreid toegelicht in het overigens ook zeer aanbevelenswaardige boek :

Microcomputer programming

J.F. Wakerley

John Wiley " f 100,-

Dit engelstalige boek gaat over de 6809, 68000 en dergelijke geavanceerde processoren.

0010: SYM-BASIC VOOR DE JUNIOR

0020: =====

0030: DOOR: JARON BORENSZTAJN, SARPHATIPARK 86, A'DAM.

0040:

0050: ER ZIJN VOOR DE 6502 VERSCHILLENDE BASIC'S VERKRIJGBAAR

0060: ER IS ECHTER NOG GEEN KANT EN KLARE VERSIE VOOR DE

0070: JUNIOR. HET IS NATUURLIJK MOGELIJK EEN BESTAANDE BASIC

0080: AAN TE PASSEN. HET IS DAN OOK GELUKT MET EEN PAAR AAN-

0090: PASSINGEN DE BASIC VAN DE SYM-1 (TE KOOP BIJ BRUTECH)

0100: WERKEND TE KRIJGEN OP DE JUNIOR. DEZE BASIC NEEMT 8K IN

0110: EN WERKT BIJ MIJ IN ROM. DE BASIC MOET STAAN OP ADRES

0120: \$C000 T/M \$DFFF. OM DE BASIC TE LATEN WERKEN, MOETEN

0130: NAAST DE WIJZIGINGEN IN DE BASIC OOK EEN PAAR KLEINE

0140: ROUTINES IN HET LAGE DEEL VAN DE TM-EPROM. DIE VOLGEN

0150: HIERNA; DAARNA BESPREEK IK DE WIJZIGINGEN IN DE BASIC.

0160:

ADRES	DATA	ADRES	DATA
0170:			
0180:			
0190:	GEW. PRCHA ROUT.		
0200:	Z. BREAKTEST		
0210:	0C80 8E 60 1A	0CD0 B0 08	
0220:	0C83 8D 62 1A	0CD2 C9 41	
0230:	0C86 AD 82 1A	0CD4 B0 08	
0240:	0C89 29 FE	0CD6 C9 3A	
0250:	0C8B 8D 82 1A	0CD8 90 06	
0260:	0C8E 20 12 13	0CDA C9 30	
0270:	0C91 A2 07	0CDC 38	
0280:	0C93 4E 62 1A	0CDD 60	
0290:	0C96 90 23	0CDE E9 37	
0300:	0C98 AD 82 1A	0CE0 29 0F	
0310:	0C9B 09 01	0CE2 18	
0320:	0C9D 8D 82 1A	0CE3 60	
0330:	0CA0 20 12 13	GEW. GETCHA ROUT.	
0340:	0CA3 CA	0CE4 20 AE 12	
0350:	0CA4 D0 ED	0CE7 C9 08	
0360:	0CA6 AE 59 1A	0CE9 D0 02	
0370:	0CA9 AD 82 1A	0CEB A9 5F	
0380:	0CAC 09 01	0CED C9 10	
0390:	0CAE 8D 82 1A	0CEF D0 03	
0400:	0CB1 20 12 13	0CF1 4C 5F 10	
0410:	0CB4 CA	0CF4 60	
0420:	0CB5 D0 F2	INIT. V. BASIC	
0430:	0CB7 AE 60 1A	0CF5 A9 02	
0440:	0CBA 60	0CF7 85 0B	
0450:	HEX TO NIBBLE	0CF9 A9 00	
0460:	0CBB AD 82 1A	0CFB 85 0C	
0470:	0CBE 29 FE	0CFD 60	
0480:	0CC0 8D 82 1A	BREAKTEST ROUT.	
0490:	0CC3 4C A0 0C	0CFE 2C 80 1A	
0500:	0CC6 C9 0D	0D01 10 02	

0510:	0008	F0 19	0D03	18
0520:	00CA	C9 30	0D04	50
0530:	00CC	90 0C	0D05	2C 80 1A
0540:	00CE	C9 47	0D08	10 FB
0550:			0D0A	38
0560:			0D0B	60

0570:

0580: TOT ZOVER DE AANVULLINGEN IN DE TM-EPROM.

0590: HIERONDER VOLGEN DE WIJZIGINGEN IN DE BASIC.

0600: (DE ADRESSEN STAAN NIET OP VOLGORDE)

0610: ADRES SYM-1 JUNIOR

0620:

0630:	C38F	20 1B 8A	20 E4 0C	GETCHA ROUTINE
0640:	C678	8D 4E A6	8D 79 1A	ID-ADR. V. OP BAND
0650:	C684	8D 4C A6	8D 70 1A	SAL V. SCHR. OP BAND
0660:	C687	8C 4D A6	8C 71 1A	SAH V. SCHR. OP BAND
0670:	C68E	8D 4A A6	8D 72 1A	EAL V. SCHR. OP BAND
0680:	C691	8C 4B A6	8C 73 1A	EAH V. SCHR. OP BAND
0690:	C6B9	8D 4E A6	8D 79 1A	ID-ADRES V. VAN BAND
0700:	C6CF	A6 4A A6	A6 FA 00	EAL V. LEZEN V. BAND
0710:	C6D2	AC 4B A6	AC FB 00	EAH V. LEZEN V. BAND
0720:	C619	20 86 83	20 FE 0C	BREAKTEST ROUTINE
0730:	C98B	20 47 8A	20 80 0C	PRCHA ROUTINE
0740:	CE2B	20 75 82	20 C6 0C	CONV. ASCII-HEX NIBBLE
0750:	CE34	20 75 82	20 C6 0C	CONV. ASCII-HEX NIBBLE
0760:	DE6D	20 86 8B	20 F5 0C	INITIAL. ROUTINE
0770:	DEAD	A9 87	A9 76	SAVE ROUTINE LAAG
0780:	DEAF	A0 8E	A0 14	SAVE ROUTINE HOOG
0790:	DEB5	A9 78	A9 96	LOAD ROUTINE LAAG
0800:	DEB7	A0 8C	A0 14	LOAD ROUTINE HOOG
0810:	DEF8	A9 00	A9 00	BEGADL WERKGEBIED
0820:	DEFA	A0 02	A0 20	BEGADH WERKGEBIED
0830:	DF61	A2 00	A2 00	BEGADL WERKGEBIED
0840:	DF63	A0 02	A0 20	BEGADH WERKGEBIED
0850:	DEA1	A9 02	A9 68	GONIOMETR. FUNKTIES L
0860:	DEA3	A0 08	A0 0F	GONIOMETR. FUNKTIES H
0870:	DEA5	85 0B	EA EA	VERVALT
0880:	DEA7	84 0C	EA EA	VERVALT

0890: HET BEGIN VAN HET WERKGEBIED IS HIER \$2000. INDIEN

0900: EEN ANDER ADRES GEWENST IS, KAN DAT GEWOON VER-

0910: ANDERD WORDEN.

0920: STARTADRES BASIC: \$C000.

0930: BASIC MELDT ZICH MET: MEMORY SIZE?

0940: HIERNA MOET DEZELFDE PROCEDURE GEVOLGD WORDEN ALS

0950: BIJ DE SYM-1. ZIE VOOR VERDERE GEBRUIKSAANWIJZING

0960: DE MANUAL VAN DE SYM-1. BASIC.

0970: DE APART VERKRIJGBARE ROUTINES VOOR SINUS, COSINUS,

0980: TANGENS, EN ARCTANGENS KUNNEN BIJ DE JUNIOR IN DE

0990: TM-EPROM WORDEN GEPROGRAMMEERD, VAN ADRES \$0EC7 -

1000: \$0FFF. DE BASIC VERWIJST HIER AUTOMATISCH NAAR.

1010: SUCCES.

# 6502 SYSTEM SOFTWARE

EPROM THE JUNIOR\_PROMOTING CY (W&J) 281281 PAGE 01

```

0010:           AANVULLING OP ELEKTUUR EPROM-PROGRAMMER.
0020:           =====
0030:
0040:           HET MAANDBLAD ELEKTUUR PUBLICEERDE IN HET JANU-
0050:           ARINUMMER 1982 EEN EPROM-PROGRAMMER VOOR DE
0060:           HUISCOMPUTERS SC/MP EN DE JUNIOR, MET DE BIJ-
0070:           BEHORENDE SOFTWARE. TENEINDE MET DEZE OP EURO-
0080:           KAART UITGEVOERDE EPROM-PROGRAMMER TE BESTUREN
0090:           V.W.B. DUPLICEREN EN VERIFIEREN, IS EEN AANVUL-
0100:           LING OP DE GEPUBLICEEERDE SOFTWARE GEMAAKT.*)
0110:           EEN KLEINE WIJZIGING IN HET OORSPRONKELIJKE
0120:           PROGRAMMA, DAT VOORAF MOET GAAN AAN DE HIERNA
0130:           VOLGENDE ROUTINE, IS WEL NOODZAKELIJK, EN ZIET
0140:           ER ALS VOLGT UIT:
0150:
0160:           OP ADRESSEN 0230 T/M 0232 EN 0252 T/M 0254
0170:           STAAT 4C 1D 1C JMP RESET RETURN TO JUNIOR MON.
0180:           DAARIN MOET KOMEN 4C 78 02 JMP POORT.
0190:

```

AUTEUR: J.F. VAN SPRANG  
TULP 71  
2925 EW KRIMPEN A/D IJSSSEL.

STARTADRES : X02A0

```

0270:           ;PAGE ZERO DATA BUFFERS
0280: 0278     EPROM  ORG   X0278
0290:           00 00   SAL   *   X0000
0300:           01 00   SAH   *   X0001
0310:           02 00   EAL   *   X0002
0320:           03 00   EAH   *   X0003
0330:           04 00   MOVL  *   X0004
0340:           05 00   MOVH  *   X0005
0350:
0360:           ;EPROM-PROGRAM ADDRESSES
0370:           00 02   PROG  *   X0200
0380:           22 02   DUPL  *   X0222
0390:           33 02   VERIFY *   X0233
0400:
0410:           ;PIA-A JUNIOR
0420:           80 1A   PAD   *   X1A80
0430:           81 1A   PADD  *   X1A81
0440:           82 1A   PBD   *   X1A82
0450:           83 1A   PBDD  *   X1A83
0460:
0470:
0480:
0490:
0500:

```

\*) HIERDOOR IS HET MOGELIJK GE-  
WORDEN VANAF EEN TERMINAL TOETSEN-  
BORD TE WERKEN. HET PROGRAMMA KAN  
OOK GEBRUIKT WORDEN OM DATABLOCKS  
TE VERHUIZEN EN TE CONTROLEREN.

# 6502 SYSTEM SOFTWARE

EPROM THE JUNIOR\_PROMOTING CY (W&J) 281281 PAGE 02

```

0510:
0520: ; JUNIOR PM-ROUTINES
0530: 5F 10 LABJUN * X105F
0540: E8 11 CRLF * X11E8
0550: 6F 12 HEXNUM * X126F
0560: AE 12 RECCHA * X12AE
0570: 34 13 PRCHA * X1334
0580:
0590: ; ROUTINE VOOR HET HERSTELLEN VAN DE PIA-
0600: ; POORTEN
0610: 0278 A9 67 POORT LDAIM X67 VOOR DE POORTEN TERMINAL
0620: 027A 8D 82 1A STA PBD
0630: 027D A9 00 LDAIM X00
0640: 027E 8D 80 1A STA PAD
0650: 0282 A9 7F LDAIM X7F
0660: 0284 8D 81 1A STA PADD
0670: 0287 8D 83 1A STA PRDD
0680: 028A 20 E8 11 JSR CRLF
0690: 028D A0 9C NIEUW LDYIM X9C
0700: 028F 20 02 03 JSR TEKST NOG EEN KEER
0710: 0292 20 AE 12 JSR RECCHA
0720: 0295 C9 59 CMPJM X59 IS DIT EEN Y DAN TERUG
0730: 0297 D0 03 RNE OPE
0740: 0299 4C A0 02 JMP START
0750: 029C 4C 5F 10 OPE JMP LABJUN
0760:
0770: ; HOOFDPROGRAMMA
0780: 029F EA NOP
0790: 02A0 A0 00 START LDYIM X00
0800: 02A2 20 02 03 JSR TEKST EPROM-PROGR.+ STARTADRES
0810: 02A5 20 11 03 JSR KEY
0820: 02A8 85 01 STA SAH
0830: 02AA 20 11 03 JSR KEY
0840: 02AD 85 00 STA SAL
0850: 02AF 20 E8 11 JSR CRLF
0860: 02B2 A0 29 LDYIM X29 DATA BLOCK AND ADDRESS+01
0870: 02B4 20 02 03 JSR TEKST
0880: 02B7 20 11 03 JSR KEY
0890: 02BA 85 03 STA EAH
0900: 02BC 20 11 03 JSR KEY
0910: 02BF 85 02 STA EAL
0920: 02C1 20 E8 11 JSR CRLF
0930: 02C4 A0 3C LDYIM X3C EPROM-PROGR. STARTADRES
0940: 02C6 20 02 03 JSR TEKST
0950: 02C9 20 11 03 JSR KEY
0960: 02CC 85 05 STA MOVH
0970: 02CE 20 11 03 JSR KEY
0980: 02D1 85 04 STA MOVL
0990: 02D3 20 E8 11 JSR CRLF
1000: 02D6 A0 52 LDYIM X52 KIEZEN V. PROG/DUPL/VERIFY

```

# 6502 SYSTEM SOFTWARE

EPROM THE JUNIOR\_PROMOTING CY (W&J) 281281 PAGE 03

```

1010: 02D8 20 02 03      JSR   TEKST
1020: 02DB 20 E8 11      JSR   CRLF
1030: 02DE 20 AE 12      JSR   RECCHA
1040: 02E1 C9 50          CMPIM #50      ALS DIT EEN P IS, DAN VERDER
1050: 02E3 D0 06          BNE   AA
1060: 02E5 20 1E 03      JSR   SLEUTE
1070: 02E8 4C 00 02      JMP   PROG
1080: 02EB C9 44          AA    CMPIM #44      ALS DIT EEN D IS, DAN VERDER
1090: 02ED D0 06          BNE   BB
1100: 02EF 20 1E 03      JSR   SLEUTE
1110: 02F2 4C 22 02      JMP   DUPL
1120: 02F5 C9 56          BB    CMPIM #56      ALS DIT EEN V IS, DAN VERDER
1130: 02F7 20 E8 11      JSR   CRLF
1140: 02FA D0 03          BNE   CC
1150: 02FC 4C 33 02      JMP   VERIFY
1160: 02FF 4C 8D 02      CC    JMP   NIEUW
1170:
1180:                    ;SUBR. VOOR DE TEKST
1190: 0302 B9 3C 03      TEKST LDAAY TABEL
1200: 0305 C9 03          CMPIM #03      EINDE TEKST
1210: 0307 F0 07          BEQ   DDD
1220: 0309 20 34 13      JSR   PRCHA
1230: 030C C8            INY
1240: 030D 4C 02 03      JMP   TEKST
1250: 0310 6D            DDD   RTS
1260:
1270:                    ;VERWERKING VAN DE CIJFERS
1280: 0311 20 AE 12      KEY  JSR   RECCHA
1290: 0314 20 6F 12      JSR   HEXNUM
1300: 0317 20 AE 12      JSR   RECCHA
1310: 031A 20 6F 12      JSR   HEXNUM
1320: 031D 6D            RTS
1330:
1340:                    ;SUBR. VOOR HERNIEUWDE START
1350: 031E A0 8D          SLEUTE LDYIM #8D
1360: 0320 B9 3C 03      TA   LDAAY TABEL
1370: 0323 C9 03          CMPIM #03      EINDE TEKST
1380: 0325 F0 07          BEQ   GG
1390: 0327 20 34 13      JSR   PRCHA
1400: 032A C8            INY
1410: 032B 4C 20 03      JMP   TA
1420: 032E 20 AE 12      GG   JSR   RECCHA
1430: 0331 C9 59          CMPIM #59      ALS DIT EEN Y IS, DAN VERDER
1440: 0333 D0 04          BNE   JJJ
1450: 0335 20 E8 11      JSR   CRLF
1460: 0338 6D            RTS
1470: 0339 4C 5F 10      JJJ  JMP   LABJIN
1480:
1490:
1510:

```

# 6502

# SYSTEM SOFTWARE

EPR0M THE JUNIOR\_PROMOTING CY (W&J) 281281 PAGE 04

1520:  
1530:  
1540:

## TABEL VOOR DE TEKST.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0330:	20	20	20	20	20	20	45	50	52	4F	4D	2D	50	52	4F	47
0340:	52	41	4D	4D	45	52	0D	0A	53	54	41	52	54	2D	41	44
0350:	44	52	45	53	53	20	3A	2D	03	45	4E	44	41	44	44	52
0360:	45	53	53	20	2B	31	2D	3A	2D	03	2D	03	45	50	52	4F
0370:	4D	20	53	54	41	52	54	2D	41	44	44	52	45	53	2D	3A
0380:	2D	03	4B	49	45	53	2D	56	4F	4F	52	0D	0A	50	52	4F
0390:	47	2D	2D	2D	3D	2D	50	0D	0A	44	55	50	4C	2D	2D	2D
03AC:	3D	2D	44	0D	0A	56	45	52	49	46	59	2D	3D	2D	56	03
03BC:	2D	5A	45	54	2D	53	43	48	41	4B	45	4C	2D	32	34	56
03CC:	2D	4F	4D	2D	7B	59	2F	4E	2D	2D	03	49	4B	2D	42	
03DC:	45	4E	2D	4B	4C	41	41	52	2D	3A	2D	0D	0A	4E	4F	47
03FC:	2D	45	45	4E	2D	4B	45	45	52	2B	59	2D	26	2D	4E	2D
03FC:	2D	3A	2D	03												

SYMBOL TABLE 4000 4110 281281 PAGE 01

AA	02EB	BB	02F5	CC	02FF	CRLF	11E8
DDD	0310	DUPL	0222	EAH	0003	EAL	0002
EPR0M	0278	GG	0321	HEXNUM	126F	JJJ	0339
KEY	0311	LABJUN	1051	MOVH	0005	MOVL	0004
NIEUW	028D	OPE	029C	PADD	1A81	PAD	1A80
PBDD	1A83	PBD	1A82	POORT	0278	PRCHA	1334
PROG	0200	RECHA	12AL	SAH	0001	SAL	0000
SLEUTE	031E	START	02AD	TA	032D	TABEL	033C
TEKST	0302	VERIFY	0233				

J.P.G. VOLLERING  
 DRS. STIJKELSTRAAT 1  
 3132 BT VLAARDINGEN

AANPASSING VAN DE JUNIORPRINT TEN BEHOE-  
 VE VAN EEN 2716-EPROM (5-VOLT VERSIE).

=====

DOOR MIJN ONTEVREDENHEID MET DE VERSCHILLENDE SPANNINGEN DIE NODIG ZIJN OM MIJN JUNIOR TE VOEDEN, BEN IK OP ZOEK GEGAAN NAAR EEN VEREENVOUDIGING HIERVAN. DIT MEDE DOOR HET FEIT, DAT ER OP HET OGENBLIK EPROMS OP DE MARKT ZIJN DIE NIET DUUR MEER ZIJN EN VOOR HET UITLEZEN SLECHTS EEN ENKELE 5-VOLT VOEDINGSSPANNING NODIG HEBBEN (2716 EN DE 2732, RESP. 2K X 8 EN 4K X 8 EPROM).

BIJ HET WIJZIGEN VAN DE EPROM-AANSLUITING VAN DE 2708 (1K X 8) OP DE PRINT NAAR DE 2716 (2K X 8) STUITTE IK OP HET 'PROBLEEM', WAAR DE EXTRA 1K X 8 TE ZETTEN. DE HELFT VAN DE 2716 IS NAMELIJK GEVULD MET HET MONITORPROGRAMMA VAN DE JUNIOR DAT IN K-7 MOET STAAN, DE ANDERE HELFT VAN DE 2716 ZOU NU BIJ HET AANSLUITEN VAN ADRESBUS 10 IN K-6, 4 OF K-2 KOMEN TE STAAN. K-6 VALT REEDS AF ALS MOGELIJKHEID, DAAR DE PIA IN K-6 IS GEPLAATST. WAT NOG OPEN IS, IS K-1, 2, 3, 4 EN K-5.

WAT IS NU DE BESTE PLAATS OM DE REST VAN DE EPROM TE PLAATSEN? IK HEB GEKOZEN VOOR K-1.

EN WEL OM REDEN DAT K-2, 3, 4 EN K-5 DAN AANSLUITEND BLIJVEN (ZIE FIG.1), EN DAT BIJ K-2 EN K-4 ADRESBUS 10 LAAG IS, ZODAT OP DEZE PLAATSEN EENVOUDIG IN EEN LATER STADIUM UITWENDIG VAN DE PRINT 2 EPROMS 2716 AAN TE SLUITEN ZIJN.

DOOR ADRESBUS 11 TE GEBRUIKEN, IN PLAATS VAN ADRESBUS 10 IS HET MOGELIJK DE OVERGEBLEVEN 1K VAN DE 2716 IN K-1 TE PLAATSEN. WELNU, DE VERANDERINGEN DIE IK MOEST AANBRENGEN OP DE JUNIORPRINT WAREN ZEER EENVOUDIG EN BEPERKT:

- 1 - LEG DE PRINT VOOR U NEER MET DE COMPONENTEN NAAR BENEDEN.
- 2 - SNIJ DE PRINTBAAN DOOR DIE NAAR PIN 19 GAAT VAN DE EPROMVOET (OP ONGEVEER 2 A 3 MM VAN DE IC-VOET; VIA DEZE PRINTBAAN KOMT + 12 VOLT NAAR DE EPROM).
- 3 - SNIJ DE PRINTBAAN DOOR DIE NAAR PIN 21 GAAT VAN DE EPROMVOET (VIA DEZE PRINTBAAN KOMT DE - 5 VOLT NAAR DE EPROM).
- 4 - VERBINDT PIN 2 EN PIN 9 VAN DE 74145 (WE VERBINDEN HIERMEE K-1 MET K-7; K-7 STUURDE REEDS DE EPROM, NU STUREN K-1 EN K-7 DE EPROM).
- 5 - VERBINDT PIN 18 EN PIN 20 VAN DE EPROMVOET (EERST WERD DOOR K-7 ALLEEN CS (PIN 20) VAN DE 2708 AANGESTUURD, NU WORDT DOOR K-7 OF K-1 DE OE (PIN 20) EN DE CS (PIN 18) AANGESTUURD).

- 6 -- VERBINDT PIN 24 EN PIN 21 VAN DE EPROMVOET (HIERMEE ZETTEN WE DE + 5 VOLT VAN PIN 24 OP DE VPP VAN PIN 21).
- 7 -- VERBINDT PIN 14 VAN DE 74145 MET PIN 19 VAN DE EPROMVOET (HIERMEE ZETTEN WE ADRESBUS 11 OP DE A10-INGANG VAN DE 2716).

NADAT U DEZE WIJZIGINGEN AANGEBRACHT HEEFT, KUNT U DE 2716 (5-VOLT VERSIE), DIE HET MONITORPROGRAMMA BEVAT, IN DE EPROMVOET PLAATSEN. ALS WE DE SPANNING NU AANZETTEN ZAL DE JUNIOR, ALS U ALLES GOED GEDAAN HEEFT, WEER NORMAAL WERKEN. DE + 12 VOLT EN DE - 5 VOLT ZIJN NOG STEEDS OP DE JUNIOR AANGESLOTEN EN DEZE KUNT U NU VAN UW VOEDING VERWIJDEREN, ZODAT U EEN VOLLEDIGE + 5 VOLT JUNIOR HEEFT

FIGUUR 1. NIEUWE INDELING VAN K-0 T/M K-7 VAN DE JUNIOR.

	A12	A11	A10	
K-0	0	0	0	→ RAM
K-1	0	0	1	→ 1E GED. 2716
K-2	0	1	0	} OVER, OM EVT 2X2716 EXTERN AAN TE SLUITEN
K-3	0	1	1	
K-4	1	0	0	
K-5	1	0	1	
K-6	1	1	0	→ PIA
K-7	1	1	1	→ MONITORPROGR. IN 2E GED. VAN 2716

In de 6502-Kenner nr. 19 publiceerden wij een programma HEX-DUMP t.b.v. de KIM-1 van de auteur Frans Smeehuizen te Cappelle a.d. IJssel. Uit de listing blijkt dat binnen het programma op adres \$2A05 wordt gesprongen naar een subrou-tine op adres \$FA70, om "STARTADRES: " te printen. Die subroutine werd abusievelijk niet in de listing opgenomen. We geven u alsnog de inhoud van de adressen van bedoelde subroutine, t.w.:

```
FA70 A2 00 BD 7E FA 20 A0 1E E8 E0 0C D0 F5 60 53 54
FA80 41 52 54 41 44 52 45 53 3A 20
```

# 6502 SYSTEM SOFTWARE

KIM SOFTWARE LIBRARY

PAGE 01

```
0010: * ---- FILE 01 ----
0020: *
0030: * PATCHES OP MICRO ADE DEEL 11 (VOOR DE JUNIOR)
0040: * -----
0050: *
0060: * DOOR: A. S. HANKEL
0070: *       W. KLOOSSTRAAT 32
0080: *       7606 BB ALMELO
0090: *       Tel.: 05490 - 51151
0100: *
0110: * Het doel van deze patches, is Micro ADE - voorzien
0120: * van de patches 1 t/m 10 van Sebo Woldringh -
0130: * aan te passen voor de Junior.
0140: *
0150: * Om Micro ADE te laten draaien op de Junior, is
0160: * minimaal 8K RAM uitbreiding nodig, vanaf adres
0170: * $2000.
0180: *
0190: * Bij elke patch wordt het doel ervan omschreven.
0200: *
0210: * Micro ADE is een Copyright (C) 1977 product van
0220: * Peter Jennings, Micro-Ware Ltd., 27 Firstbrooke
0230: * Road, Toronto, Ontario, Canada M4E 2L2.
0240: *
0250: * ---- FILE 02 ----
0260: * Doel: Aanpassing OUTCH. OUTCH is een subroutine
0270: * binnen het Junior programma. OUTCH stuurt een ASCII
0280: * karakter naar het terminal.
0290: *
0300: 2E97 OUTCH *      $1334
0310: *
0320: 2EA0          ORG  $2EA0
0330: *
0340: 2EA0 4C 34 13          JMP  OUTCH          ; THE PRINTER OUTPUT ROUTINE
0350: *
0360: * ---- FILE 03 ----
0370: *
0380: * Doel: Aanpassing input routine. GETCH is, net als
0390: * de eerder genoemde OUTCH, een subroutine binnen
0400: * PM. GETCH haalt een ASCII-karakter van het keyboard.
0410: *
0420: 3000          ORG  $3000
0430: *
0440: 3000 GETCH *      $12AE
0450: *
0460: 3000 4C AE 12          JMP  GETCH          ; THE KEYBOARD INPUT ROUTINE
0470: *
0480: * ---- FILE 04 ----
0490: *
0500: * Doel: Aanpassing break-routine. Eerder genoemde OUTCH
```

# 6502 SYSTEM SOFTWARE

KIM SOFTWARE LIBRARY

PAGE 02

```
0510: * routine bevat een breaktest. Wordt de breaktoets
0520: * ingedrukt, dan volgt een sprong naar het adres gespe-
0530: * cificeerd in BRKT.
0540: *
0550: 2E19          ORG      $2E19
0560: *
0570: 2E19          MA      *      $2031
0580: 2E19          BRKT   *      $1A7C
0590: *
0600: 2E19 A9 31          LDAIM MA
0610: 2E1B 8D 7C 1A      STA   BRKT
0620: 2E1E A9 20          LDAIM MA /
0630: 2E20 8D 7D 1A      STA   BRKT +01
0640: 2E23 A6 F5          END   LDX   $00F5      ; RESTORE X-REGISTER
0650: 2E25 A4 F4          LDY   $00F4      ; RESTORE Y-REGISTER
0660: 2E27 60            RTS
0670: *
0680: * Als gevolg van het bovenstaande moet een relatieve
0690: * branch worden aangepast:
0700: *
0710: 2E3F          ORG      $2E3F
0720: *
0730: 2E3F D0 E2          BNE   END
0740: *
0750: * ---- FILE 05 ----
0760: *
0770: * Doel: Het juist zetten van de IRQ-vector. Een error
0780: * binnen MA wordt verwerkt via een break, hetgeen resul
0790: * binnen MA wordt verwerkt via een break, hetgeen
0800: * resulteert in een sprong via de IRQ-vector.
0810: *
0820: * IRQ-setting binnen de Editor:
0830: *
0840: 2039          ORG      $2039
0850: *
0860: 2039          ERRORA *      $2404
0870: 2039          IRQ     *      $1A7E
0880: *
0890: 2039 A9 04          LDAIM ERRORA
0900: 203B 8D 7E 1A      STA   IRQ
0910: 203E A9 24          LDAIM ERRORA /
0920: 2040 8D 7F 1A      STA   IRQ +01
0930: *
0940: * IRQ-setting binnen de assembler:
0950: *
0960: 2020          ORG      $      2E70
0970: *
0980: 2020          ERRORB *      $28A4
0990: *
1000: 2020 A9 A4          LDAIM ERRORB
```

# 6502 SYSTEM SOFTWARE

KIM SOFTWARE LIBRARY

PAGE 03

```
1010: 2022 8D 7E 1A          STA  IRQ
1020: 2025 A9 28          LDAIM ERRORB /
1030: 2027 8D 7F 1A          STA  IRQ  +01
1040: *
1050: * ---- FILE 06 ----
1060: *
1070: * Doel: Aanpassing backspace-karakter om bij mijn
1080: * terminal - de Elekterminal - een backspace
1090: * te genereren (code hex 08).
1100: *
1110: 2ED3          ORG  $2ED3
1120: *
1130: 2ED3 A9 08          LDAIM $08          ; ASCII BS (BACK SPACE)
1140: *
1150: * ---- FILE 07 ----
1160: *
1170: * Doel: Het plaatsen van de subroutine PACKT. Binnen
1180: * het systeem programma TM bevindt zich wel een PACKt-
1190: * achtige routine, maar deze is niet als subroutine
1200: * aan te roepen. Daarom is de routine PACKT volgens de
1210: * M.A. manual geplaatst vanaf adres 3F38.
1220: *
1230: 2E94          ORG  $2E94
1240: *
1250: 2E94          PACKT *  $3F38
1260: *
1270: 2E94 4C 38 3F          JMP  PACKT
1280: *
1290: * Voor de listing van PACKT en de overige wijzigingen
1300: * binnen M.A.: zie manual pagina 38.
1310: *
1320: * ---- FILE 08 ----
1330: *
1340: * Toelichting bij de hierna volgende files:
1350: *
1360: * De hierna volgende files hebben de aanpassing van de
1370: * cassette read/write routines als onderwerp.
1380: *
1390: * Om source-cassettes binnen de KIM-club universeel te
1400: * houden, is gekozen voor het patchen van de bestaande
1410: * routines, i.p.v. gebruik te maken van de Junior-
1420: * routines DUMP/DUMPT en READ/READT.
1430: * Bovendien blijft op deze manier een aantal patches
1440: * van Sebo Woldringh behouden, zoals een delay van 1
1450: * a 2 seconden bij een schrijfoperatie teneinde de
1460: * motor van de recorder op toeren te laten komen, en
1470: * het feature dat de ID van de gelezen files worden
1480: * zichtbaar gemaakt.
1490: *
1500: * ---- FILE 09 ----
```

# 6502 SYSTEM SOFTWARE

KIM SOFTWARE LIBRARY

PAGE 04

```

1510:      *
1520:      * Doel: Aanpassing PIA-adressen.
1530:      *
1540: 2E97      PAD      *      $1A80
1550: 2E97      PADD     *      PAD      +01
1560: 2E97      PBD      *      PADD     +01
1570: 2E97      PBDD     *      PBD      +01
1580:      *
1590:      * PAD moet worden ingevuld op de volgende adressen:
1600:      *
1610:      * $2ED3, $2EDD, $3C2A, $3C34, $3D44, $3E55, $3E5D.
1620:      *
1630:      * PADD moet worden ingevuld op de volgende adressen:
1640:      *
1650:      * $3C06, $3D3F.
1660:      *
1670:      * PBD moet worden ingevuld op de volgende adressen:
1680:      *
1690:      * $2EC7, $2FE1, $3055, $3058, $3066, $3069, $3071
1700:      * $3074, $307C, $307F, $38FF, $3902, $3BFE, $3C01,
1710:      * $3C14, $3E5A.
1720:      *
1730:      * PBDD moet worden ingevuld op de volgende adressen:
1740:      *
1750:      * $2F48, $391E.
    
```

De thans gepubliceerde patches op Micro ADE voor de Junior, worden vervolgd.

## SYMBOL TABLE D000 D04E

BRKT	1A7C	END	2E23	ERRORA	2404	ERRORB	28A4
GETCH	12AE	IRQ	1A7E	MA	2031	OUTCH	1334
PACKT	3F39	PADD	1A81	PAD	1A80	PBDD	1A83
PBD	1A82						

Frans Smeehuijzen

VOOR WAT BETREFT DE HARDWARE HEB IK VOOR MIJN KIM-1 DE VOLGENDE VOEDINGSSCHAKELING ONTWERPEN. DE OVERSPANNINGSBEVEILIGING HEB IK ER AAN TOEGEVOEGD N.A.V. DE ERVARING DIE IK HEB OPGEDAAN BIJ HET EXPRIMENTEREN MET TTL-IC WELKE HELAAS BIJ 5,5V ECHT DE GEEST GEVEN.

VOORWAARDEN VAN DE SCHAKELING MOESTEN ZIJN:

- BIJ OVERSPANNING MOET DEFINITIEF DE SPANNING VAN DE SCHAKELING WORDEN AFGEKOPPELD
- ER MAG GEEN KORTSLUITING VOOR DE SPANNINGSREGELAAR ONTSTAAN
- DE SCHAKELING MOET SNEL WERKEN

AANGEZIEN EEN RELAISSCHAKELING RELATIEF LANGZAAM IS HEB IK GEKOZEN VOOR EEN RELAIS DAT REEDS BIJ EEN "LAGE" STROOMSTERKTE BEKRACHTIGD WORDT.

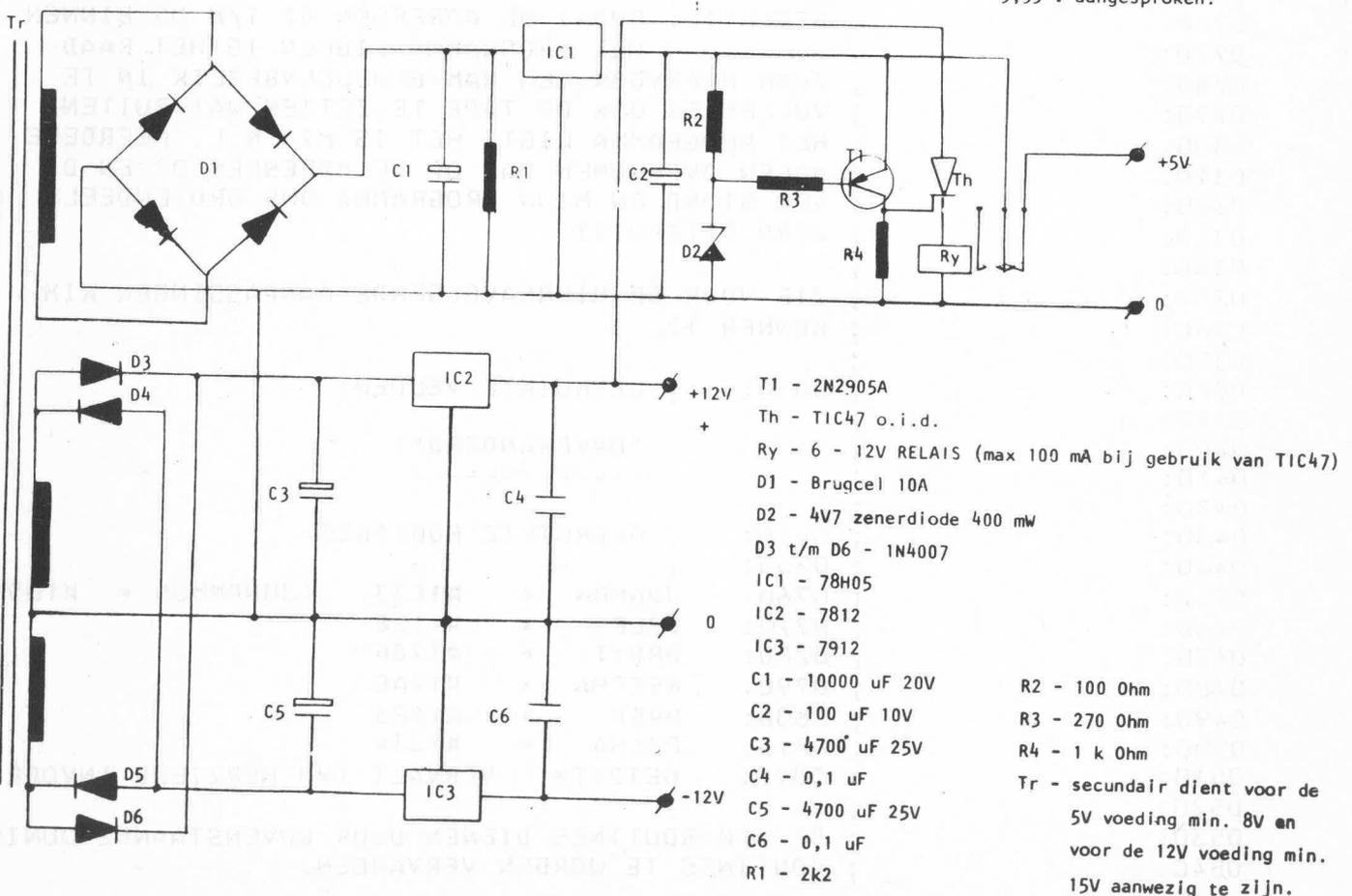
DE SCHAKELKONTAKTEN MOETEN UITERAARD DE MAXIMAAL BENODIGDE STROOM KUNNEN SCHAKELEN.

DE SCHAKELING REAGEERT OP EEN SPANNING DIE GELIJK IS AAN DE SOM VAN DE ZENERSPANNING EN DE  $V_{eb}$ .

VOOR DE 5V VOEDING IS DAT  $4,7 \text{ V} + 0,6 \text{ V}$ .

VOOR JUNIOR-GEbruikers IS HET EENVOUDIG OM M.B.V. EEN 7905 SPANNINGSREGELAAR OP DE -12V DE BENODIGDE -5V TE BETREKKEN.

Nb. De overspanningsbeveiliging wordt bij 5,35 V aangesproken.



# 6502 SYSTEEM SOFTWARE

KIMNIM THE JUNIOR\_PROMOTING CY (W&J) 261281 PAGE 01

```
0010:
0020: ; MEMORY TEST
0030: ; -----
0040: ; IN KIM KENNER 12 PUBLICEEERDE S.T. WOLDRINGH
0050: ; EEN TESTPROGRAMMA DAT OP 6 VERSCHILLENDE
0060: ; MANIEREN HET GEHEUGEN TEST, EN WAARAAN GE-
0070: ; MAKKELIJK NOG, EVT. TIJDELIJKE, TESTEN KUN-
0080: ; NEN WORDEN TOEGEVOEGD.
0090: ; DIT PROGRAMMA WERD VOOR DE JUNIOR AANGEPAST
0100: ; DOOR:
0110: ; BERT VAN TIEL
0120: ; JAN VAN BEAUMONTSTRAAT 21
0130: ; 2805 RN GOUDA.
0140: ;
0150: ; MINIMUM SYSTEEM:
0160: ; JUNIOR + INTERFACE (PM) + TTY (ELEKTERMINAL)
0170: ;
0180: ; WIJZIGING INVOER VAN HET TE TESTEN ADRESSEN-
0190: ; BEREIK EN WEL ZOALS BIJ DE DISASSEMBLER
0200: ; (KIM KENNER 3).
0210: ; 0019 A5 D3      0033 A5 D5      SAL 00D2
0220: ; 001B 20 8F 12  0035 20 8F 12  SAH 00D3
0230: ; 001E A5 D2      0038 A5 D4      EAL 00D4
0240: ; 0020 20 8F 12  003A 20 8F 12  EAH 00D5
0250: ;
0260: ; ATTENTIE: OMDAT DE ADRESSEN D2 T/M D5 BINNEN
0270: ; HET PROGRAMMA LIGGEN IS HET RAAD-
0280: ; ZAAM HIERVOOR EEN RAM-GEHEUGENBEREIK IN TE
0290: ; VULLEN EN OOK OP TAPE TE ZETTEN WAT BUITEN
0300: ; HET PROGRAMMA LIGT. HET IS MIJ N.L. MEERDERE
0310: ; MALEN OVERKOMEN DAT OP DE ADRESSEN D2 EN D3
0320: ; X00 STOND EN MIJN PROGRAMMA DUS GROTENDEELS
0330: ; WERD UITGEWIST.
0340: ;
0350: ; ZIE VOOR DE HIERNAVOLGENDE AANPASSINGEN KIM
0360: ; KENNER 12.
0370: ;
0380: ; 0610: ; GEBRUIKTE VELDEN:
0390: ;
0400: ; "ONVERANDERD"
0410: ; =====
0420: ;
0430: ; 0610: ; GEBRUIKTE ROUTINES:
0440: ; 0750: ;
0450: ; 0760: JUNMON * X1C33 (JUNPMMON * X105F)
0460: ; 0770: CRLF * X11E8
0470: ; 0780: PRBYT * X128F
0480: ; 0790: RECCHA * X12AE
0490: ; 0800: PRSP * X11F3
0500: ; 0810: PRCHA * X1334
0510: ; 0820: GETBYT* VERVALT IVM HERZIENE INVOER!!!
0520: ;
0530: ; DE KIM-ROUTINES DIENEN DOOR BOVENSTAANDE JUNIOR-
0540: ; ROUTINES TE WORDEN VERVANGEN.
```

DANNY BACKX PUBLICEEERDE IN 6502-KENNER NR. 18 EEN ARTIKEL, EPROM PROGRAMMER. AL SNEL IS VELEN, OOK HEM, DUIDELIJK GEWORDEN DAT ER IETS GOED FOUT ZAT. DANNY MAAKT VOOR DE FOUTEN ZIJN EXCUSES, EN GEEFT HIERONDER AAN WELKE WIJZIGINGEN IN ZIJN PROGRAMMA MOETEN WORDEN AANGEBRACHT OM TOCH MET SUCCES EEN EPROM TE PROGRAMMEREN. DE WIJZIGINGEN ZIJN:

0231	A9 06	MOET ZIJN	A9 16
026D	A9 14		A9 04
0277	A9 10		A9 00
0286	A9 14		A9 04
02A6	A9 06		A9 16
02AB	A9 FF		A9 00
02D8	A9 00		A9 10
02FC	A9 0A		A9 1A
0301	A9 02		A9 12
0314	00 F2		00 F4

IR. R. VONK  
ST. WILLIBRORDUSLAAN 2  
5591 BG HEEZE

HEEFT DE EPROM PROGRAMMER VAN DANNY BACKX ALS VOLGT AAN DE PRAAT GEKREGEN:

ADRES IS MOET ZIJN

0232	06	11
0245	EE	CE
0248	CE	EE
026E	14	03
0278	10	07
0287	14	03
0291	EE	CE
0294	CE	EE
02A7	06	11
02AC	FF	00
02C6	EE	CE
02C9	CE	EE
02FD	0A	19
0302	02	11

LEZEN: PB1=0  
PB2=0  
PB4=1

PROGRAMMEREN:

PB1=1  
PB4=0  
PB2  50 MSEC.

RESET: PB3=1

TELPULS: PB0 

BOVENDIEN IS HET RAADZAAM OM PBD (POOT 10 VAN DE 4040) VIA 3K3 AAN DE +5V TE LEGGEN OM ONGEWENSTE TELPULSEN TE VOORKOMEN.

REST INITIAD:

030A	00 00	CPY #00
030C	00 06	BNE #0314
030E	00 00	CPX #00
0310	00 01	BNE #0313
0312	60	RTS
0313	CA	DEX
0314	88	DEY
0315	CE 82 1A	DEC PBD
0318	EE 82 1A	INC PBD
031B	4C 0A 03	JMP #030A

GHIJSELS J  
 SIONSVEST 70  
 GIELEN AA  
 TULPENSTR 2  
 GLASBERGEN JB  
 PLATANSTR 3  
 GOETHALS J  
 KORENBLOEMDREEF 25,  
 GOUDRIAAN J  
 ROGHORST 55  
 GOUT HJ  
 HEEMSKERKSTR 38  
 GRAGT AJ VD  
 V.D. KAAVSTR 10  
 GRINSVEN H V  
 ZONNENBERG 15  
 GRINTEN GJ VD  
 BLEULANDWEG 166  
 GROOT JL D  
 LAVENDLHOF 95  
 HAAN J D  
 I DA COSTALN 265  
 HAESBROUCK F  
 GISTELSE STEENWEG 58  
 HAGEDOORN W  
 POSTBUS 94  
 HAAM H  
 BARONIEHOEVEN 34  
 HAMERSLAG GJ  
 W RIJBERTSTR 48  
 HANKEL AS  
 W. KLOOSSTR 32  
 HARTEN PR V  
 EDISONSTR 144  
 HARTHOORN F  
 PR MARGRIETLN 5  
 HARTLIEF WA  
 DR SCHAEFFMANSTR 102  
 HARTOG AW DN  
 TIENVOET 2  
 HAUBRICH PJ  
 ALPENLANDESTRASSE 12  
 HAVERKAMP LJ  
 NOORDERENK 72  
 HEK HG  
 JAC. URLUSSTR 11  
 HEMMINGA HK  
 CH LEICKERTSTR 26-III  
 HENGEVELD PH  
 V. KLEFFENSLN 207  
 HENNEKE RA  
 JULIANA STR 14A  
 HENRION G  
 KARMELIETENSTR 83  
 HERMENS R  
 LARIKSTR 35  
 HERT FJ  
 GABRIELLE PETITLN 7  
 HEUSDEN A V  
 NIJSTAD 12  
 HEYDENRIJK PH  
 KWEEKTUINSTR 8  
 HILLEN PJ  
 T DE KEYSERSTR 125  
 HOEPEL D  
 P. HULDERPLANTSOEN 115  
 HOFSTEE J  
 GROENEDAAL 11  
 HOLLANDERS HJ  
 SCHOOLSTR 169  
 HOOP R D  
 KON JULIANA STR 160  
 HOPPENBROUWER H  
 HAARLEMMEERSTR 20  
 HORNEMANN JA  
 VIOLIER 5  
 HOS JH  
 GR. SPARENLN 7  
 HOUTKAMP JJ (LAB SCHEEPSCON.)  
 MELKWEG 2  
 HUBREGTSE HC  
 ZERNIKEPLAATS 63  
 HUIJZEN C  
 KUSTVAARDERSTR 25  
 HUIJBERTS AT (DIETRONIC BV)  
 BEELANTIERSGRACHT 191  
 HUIJSMAN J  
 BILDERDIJKSTR 25-RO  
 HUIJSMANS H  
 ERASMUSSTR 26  
 HUIJTERA S  
 MEIENVAARD 158  
 HUNNELING JB  
 MELISSSTOKEZIJDE 267  
 HUNTJENS JH  
 BEATRIXSTR 13  
 HUBBINKS CC  
 ZAKSTR 12  
 HUIJZEN B  
 KLOOSTERSTR 67

B-2500 LIER  
 4818 GV BREDA  
 5671 AL NUENEN  
 B-8300 KNOKKE-HEIST  
 6708 KB WAGENINGEN  
 2518 EL DEN-HAAG  
 1815 VM ALKMAAR  
 2716 PG ZOETERMEER  
 2803 HM GOUDA  
 4907 AS OOSTERHOUT  
 2624 XZ DELFT  
 B-8200 BRUGGE  
 7890 AB KLAZIENAVEEN  
 5244 HZ ROSMALEN  
 7521 TP ENSCHEDE  
 7606 BB ALMELO  
 5621 HT EINDHOVEN  
 3956 XV LEERSUM  
 2832 GM HAARLEM  
 3261 TP OUD BEVERLAND  
 A-2380 PERCHTOLDSBORF  
 7437 VP BATHMEN  
 7558 JV HENGELD  
 1062 BD AMSTERDAM  
 4334 HG MIDDELBURG  
 8019 AW ZWOLLE  
 B-2600 BERCHEM  
 4814 HT BREDA  
 B-2128 ST JOB IN'T GOOR  
 7622 LA BORNE  
 2823 WK HAARLEM  
 7545 BD ENSCHEDE  
 1785 GS DEN HELDER  
 9722 CM GRONINGEN  
 9581 GJ MUSSELKANAAAL  
 2274 JN VOORBURG  
 2312 GA LEIDEN  
 2925 TE KRIMPEN A/D IJSSEL  
 2121 CZ BENNEBROEK  
 2628 CD DELFT  
 3068 JA ROTTERDAM  
 1034 OP AMSTERDAM  
 1015 RJ AMSTERDAM  
 2013 EH HAARLEM  
 1561 KD KROMMENIE  
 1024 ST ALKMAAR  
 2543 OR DEN HAAG  
 6241 EH BUNDE  
 6211 PS HAARSTRICHT  
 1938 AL HUIZEN

IEMHOFF J  
 BIELEN 2  
 JACOBS JC  
 MARANDE 50  
 JAGER W  
 V HASSELTLN 352  
 JANSEN D  
 REIGERSKAMP 494  
 JANSEN LJ  
 WILGENLN 272  
 JANSEN MP  
 POSTBUS 87  
 JANSSENS ??  
 GERARDSWEG 38  
 JANSSENS RA  
 LELIENSTEIN 80  
 JONG CH D  
 J.V. GOOYENKADE 3  
 JONG R D  
 V. BRUNSTR 4  
 JONGE J D  
 C. FOCKSTR 65  
 JONGH JE D  
 NIEUWENDAMLN 206  
 JONKER TR  
 EIKENLAAN 107  
 JONKERS MJ  
 NARCISSENSTR 49  
 JONKMAN FG  
 STEVN STREUVELSLN 9  
 JORIS E  
 MAAFSORTLN 116  
 JORNA JL  
 MARGRIETSTR 20  
 KAMMINGA H  
 VALLATSTEREND 21  
 KAMPMAN K  
 KLEVERPARKWEG 120  
 KARHOF B  
 KOUDEHORN 157  
 KARSTEN AC  
 LEKERMEER 8  
 KATTEVILDER PJ  
 MOZARTLN 278  
 KENHOVE F V  
 KROMVELDSTR 8  
 KESTEREN R V  
 A.V. ROYENSTR 94  
 KIKKE K  
 JONKERLAANTJE 22  
 KLEEF JP  
 BOEKET 102  
 KLEEF PV  
 DE WUP 24  
 KLEIN G  
 HOOGSTR 76  
 KLEIPOOL RF  
 ROTTERDAMSRIJWEG 127  
 KONING JH D  
 H. GORTERSTR 10  
 KOOL AJ  
 WETH KOENIGLN 2  
 KOOPMANS JJ  
 J VERMEERSTR 7  
 KOOTEN GJ V  
 DE COSE 38  
 KORTEKAAS FH  
 WESTHOFF 28  
 KORTEKAAS TP  
 KL. POELLN 26  
 KORVING RA  
 W. PIJPERSTR 6  
 KROCKE H  
 HOOFDHAL 24  
 KROMHOUT  
 KORENSCHOOF 3  
 KROON V  
 IRENESTR 35  
 KROOS CJ  
 OUDE LANSTRAAT 17  
 KRUYTS TJ VD  
 MAASEIKERWEG 210  
 KUITENS MJ  
 ANTHELNISSSTR 195  
 KULKER JG  
 DE TOL 141  
 KUYVENHOVEN HH  
 BEUKENSINGEL 10  
 LAAN HA V DR  
 OSSENWEIDE 2  
 LAAN S  
 JAANWEG 30  
 LAAT CT D  
 MEIKANTSTR 6  
 LAAT H D  
 DE ZEVENDEVEN 29  
 LAAT PJ D  
 PIETHEVENSP 250

D-4459 WIELEN  
 5641 PJ EINDHOVEN  
 2625 HZ DELFT  
 3607 JG MAAKSENBOEK  
 9742 NT GRONINGEN  
 4190 CB GELDERMALEN  
 6525 RT NIJMEGEN  
 3085 DS ROTTERDAM  
 1506 CB ZAANDAM  
 1562 BD KROMMENIE  
 2613 DD DELFT  
 2547 JN DEN HAAG  
 7271 HE BORCULO  
 6666 AX HETEREN  
 5242 GD ROSMALEN  
 B-2500 LIER  
 5502 TS VELDHOVEN  
 9203 ZB DRACHTEN  
 2023 CJ HAARLEM  
 1941 KC BEVERWIJK  
 1067 PB WOGNUM  
 2625 CM DELFT  
 B-9050 EVEREGEM  
 2202 EP NOORDWIJK A ZEE  
 1135 TL EDAM  
 1261 LW BLARICUM  
 1721 EB BROEK OP LANGEDIJK  
 6701 BX WAGENINGEN  
 3042 AS ROTTERDAM  
 3351 CM PAPENDRECHT  
 1412 GT HAARLEM  
 3351 BN PAPENDRECHT  
 7339 CM UGHELEN  
 2064 MD SPAARNDAM  
 1435 GP RIJSENHOUT  
 2551 CL DEN-HAAG  
 4207 ED GORINCHEM  
 3224 VA HELLEVOETSLUIS  
 2991 BG PAPENDRECHT  
 5041 CD TILBURG  
 6006 AD MEERT  
 2522 ZC DEN-HAAG  
 2626 EB LEIDSCHEIDAM  
 2651 TS BEPKEL EN RODENRIJS  
 1689 MR ZWAG  
 1633 GB AVENTHORN  
 5261 NH VUCHT  
 1963 SB HEEMSKERK  
 3024 RJ ROTTERDAM

059-48592  
 040-816379  
 -  
 03465-69626  
 050-778602  
 03455-2753  
 080-652082  
 -  
 -  
 -  
 -  
 -  
 05457-1816  
 08306-22565  
 04192-6146  
 031-801148  
 -  
 -  
 023-261998  
 02510-21608  
 02297-2566  
 -  
 091-539062  
 01719-15398  
 -  
 02152-58051  
 -  
 -  
 010-154025  
 078-151800  
 -  
 078-156033  
 055-416942  
 -  
 02977-21888  
 070-680148  
 -  
 01883-13435  
 01806-5814  
 013-43656  
 04950-36659  
 -  
 070-271991  
 01891-2008  
 -  
 -  
 073-563084  
 02510-40595  
 010-778967



LANGEVELD RJ L.V. MIDDEBURG 26A	2275 CC VOORBURG	-
LANKESTER HR C.V. D. LINDENLN 20	1412 BZ HAARLEM	02159-42242
LEEUWEN AJ V SPOORSINGEL 82	3134 XS VLAARDINGEN	010-341052
LEENPUT B VD PLOEGVELD 7	5261 GD VUCHT	073-567457
LEEST MA V HAREDELLN 265	2625 SH DELFT	-
LEEUW E D SCHUBERTLN 2	4837 CP BREDA	076-656416
LENDERS LJ RANE FRANKSTR 23	5912 HC VENLO	077-43123
LENTING PG KORTRYKSTR 5	4826 LX BREDA	076-712478
LIESKAMP JH MEYHORST 62-25	6537 LL NIJMEGEN	-
LOCK JP JR KOELWEG 35	6391 TG NIEUWENHAGEN	045-314085
LOENEN ET V STATIONSTR 8B	6181 AJ ELSLOO	04402-2108
LOOS P V DR ZEBRASPOOR 669	3605 HM HAARSSENBROEK	03465-70600
LORIER M JULIANAWEG 362	3523 XL UTRECHT	-
LOVEN H KAMERLINGH ONNESWEG 245	1223 JH HILVERSUM	035-834523
LUBBERS JA JOHANNAWEG 5	7555 CM HENGELO	074-420059
MAAREN CC ZWAARDVEGERSTR 6	4813 CS BREDA	076-141806
MAAS JB BAKSMEEER 225	4941 LK RAAMS DONKSVEER	01621-20906
MANDERSLOOT AG POLSTRAT 22	6942 VL DIDAM	-
MANINTVELD GJ MOSVEEN 53	2544 RV DEN HAAG	070-2987181
MARECHAL W SPOORWEGSTRAT 7	B-8211 ZEDELIGH	-
MARISSEN R KASTELENWEG 38	6136 BJ SITTARD	-
MARS H MARSHANLN 23	1422 CG UITHOORN	-
MEER F VD KANALSTR 103-2	1054 XC AMSTERDAM	-
MEER RJ VD NUMEVELD 23	5298 BG VINKEVEEN	-
MEIRWANS GJ BERNADOTTELN 13	2037 GK HAARLEM	023-342620
MEPELINK G BOEKERIJ 11	7552 KS HENGELO	074-428726
MEPSCHEN F EIKENLN 4	9636 CV ZUIDBROEK	-
MESANDER F FLANORPAD 6A	2333 AR LEIDEN	-
METZGER K THOMASLN 34	5631 GL EINDHOVEN	040-446847
MICHELIS JA BORDEAUXLN 6	5627 GY EINDHOVEN	040-413171
MILTENBURG FJ GEERTRUIDENBERGSTR 43	3006 ME ROTTERDAM	020-801389
MINGER F J SCHELPEKADE 2	2313 ZT LEIDEN	071-130401
MOOY H BLANKENSTR 33-1H	1018 RP AMSTERDAM	020-246487
MUELLER A SINJ SEMEYNSTR 78-1H	1061 GM AMSTERDAM	020-860245
NEESKENS P LAVENDELHOF 2	4907 AV OOSTERHOUT	01620-54684
NIEKERK P V KIEVITSLN 8	3233 BE OOSTVOORNE	01815-2760
NIESINK FJ WABERGSEWEG 55	7007 AA DOETINCHEM	-
NIEUWENHOVE K V CONSCIENCESTR 60-4	B-3000 LEUVEN	-
NOORLANDER C FALKLANDDREEF 72	3563 AG UTRECHT	-
OBREEN HC HOOFDSTR 115	2351 AE LEIDERDORP	071-894590
OFFEREN JR V BONIFACIUSLN 261	1216 PP HILVERSUM	-
OFFRINDA HT GR H DE RIJKELN 37	2263 GK LEIDSCHENDAM	070-27713
ONOKIEMICZ HARLSTR 234	7523 RP ENSCHEDE	053-357806
OOSTENBRUGGE RA V KAPTELSTR 5	5061 CA MANSUM	-
DOSTERLINCK H VICTORIASTR 44	B-3000 GENT	-
DOSTRUM JP DE BOSKAMP 110	3030 VV HOOGLAND	-
OP DER HEYDE GJ HEIDERSSONSTR 33	0206 NA RYSWIJK	-
OPBROEK G V EOLANTIERSTR 91	9741 BN GRONINGEN	050-716011
OVERBROEK RT LN V POOT 430	0066 BP DEN HAAG	070-230201
OVEN AH V VD BERGHSTR 25	5831 GP BOXMEER	-

OPDAM AP OUDARENSTR 50	2905 VN RIDDERKERK	-
OPREL A DILLENBURGSINGEL 200	3136 EE VLAARDINGEN	010-742449
OTTEN HJ OTTOBORRENGOED 33	3871 NJ HOOVELAKEN	03495-36551
OTTEN HR HAYDENSTR 126	7204 RW ZUTPHEN	05750-23774
OUDEM AH DM LELIESTR 78	5741 XX BEEK EN DONK	04929-2281
OUDEMETERING B VD INDUSTRIEWEG 12	2102 LH HEEMSTEDE	023-206444
PASCOE JO JHR V SYPESTEYNLN 25	1231 XL LOOSDRECHT	-
PATER P D RINDONSTRAT 10	2405 EH ALPHEN AD RYN	01720-73427
PATTJE HR NIJLANDSINGEL 10	9831 RK ADUARD	050-117056
PELT ML V J. JORDAENSTR 15	2923 CK KRIMPEN AD IJSSEL	01807-19881
PERK HK SPARRENLN 11	1272 RN HUIZEN	02159-19387
PERQUIN HH POORTERSTR 11	5663 CG GELDROF	040-864442
PETERS J PARKIETENDONK 4	5467 DR VEGHEL	04130-50612
PETERS JH KERKSTR 56	2242 HH WASSENAAR	01751-15180
PETERS MB MOLENKADE 58	3251 LM STELLENDAM	-
PEURSEN D V W. ALEXANDERSINGEL 91	1782 GP DEN HELDER	02230-13653
PICKEE D TAFELBERGSTR 38	5642 EB EINDHOVEN	-
PIJS LP ZOOMSTR 20	1784 GK DEN HELDER	-
PLESSIUS HA KOLKRIJPT 63	3828 EL HOOGLAND	033-003365
POLL JJ V BURG. VD BDINKLN 62	3237 AX VIERPOLDERS	01810-5022
PORCELIJN T C. SCHUYTSTR 4	1071 HJ AMSTERDAM	020-792634
POSTMA T HERTOGIN JOHANNA SINGEL 12	5345 AJ OSS	04120-25785
PRIKANOWSKI RH GOSELINGSTR 94	5931 HV TEGELEN	077-3289001
PUTTEN AA V P. MONDRIANSTR 11	5854 GA BERGEN	08854-2524
QUAST HA CEINTUURBAAN 37	1271 BG HUIZEN	02152-54065
RAAYMAKERS F PIUSHAVEN 30	5017 AH TILBURG	-
RANKE AC KANALSTR 2A	1741 AH SCHAGEN	02240-12818
RIBBENS JJ HAMERTEDESTR 6	4485 AT KATS	01109-386
RIJSSEL L V LEUVENLN 10	5620 WG EINDHOVEN	040-418900
RODIUS D BEKENSESTR 40	B-8010 ROESELARE	-
ROEKEL G VD PALMSTR 11-C	3135 LK VLAARDINGEN	010-351101
ROL HH PLETERING 37	1678 HL OOSTWOUD	02291-1654
ROLFF HJ DE MEENTHE 110-4H	0018 AK LEEUWARDEN	05100-66022
ROOVERS PF K. DOORWANWEG 11	3621 JY BREUKELEN	03462-2871
ROOY H V V. NUSSEN BROEKSTR 11	5223 BM S'HERTOGENBOSCH	-
RUELENS M KL. KRUISSTR 4	B-3910 HERK DE STAD	-
RYSSSEN A V BETHESDASTR 35	7909 BB HOOGEVEEN	-
SALFISCHBERGER JL J. SNEKENSTR 9	5216 HR S'HERTOGENBOSCH	073-1340021
SANDERS CG ANSBALDUS 23	5501 CV HAARLE	04904-6077
SAYN R VON BODELSCHWINGSTRASSE 23	B-3302 VELLMAR	-
SCHAPER DJ BITTERIJLN 39	1402 SM BUSSUM	-
SCHAPPERS W GIERSTR 55-RD	2011 GB HAARLEM	023-312809
SCHERPENZEEL W HOOFDSTR 146	3406 GN ALPHEN AD RIJN	-
SCHOENMAKERS H RIJNSTRAT 19	0151 HK HATTEM	-
SCHOLTEN GJ A.V. OSTRABELN 2	0066 CG BERKEL ENSCHOT	013-332317
SCHOLTEN H STRAND 9	0224 ER LELYSTAD	03200-41740
SCHONENILLE L HERINCKHVE 13	7600 BG ALMELO	05490-64877
SCHOUTEN TJ JUNOPLANTSOEN 57	0024 BN HAARLEM	023-237171
SCHREY P BARTEN LEHMENSSTR 73	B-3220 HAARSHOT	016-363064
SCHREYER UB ECHTERWACHLN 161	5625 KC EINDHOVEN	040-421821

SIBEVN FL  
K DOORHANSTR 99-3E 1055 VD AMSTERDAM 020-821029

SLOTHOUBER RM  
V. VLOTENLN 53 7412 SC DEVENTER -

SMEEMUIJZEN FJ  
LIPPEDAL 19 2904 CL CAPELLE A/D IJSEL 010-512507

SMEETS FL  
BURG. WILLEMSSTR 37 6433 CS HOEENSBRÖEK 045-211312

SMID FJ  
RUIJTEVAARTLN 104 9602 ZG HOOGEZAND -

SMIENK A  
WILSONSTR 139 2131 PH HOOFFDORP 02503-32286

SMILDE H  
MASSTR 13 8303 LP EMMELOORD 05270-8271

SMIT CD  
THORBECKESTR 400 6702 CK MAGENINGEN 08370-20648

SMIT GJ  
DE LAIRESSESTR 109 1071 NX AMSTERDAM 020-717173

SMIT R  
COEHOORNSTR 65 1222 RS HILVERSUM 035-834334

SMULDERS FJ  
HEKBOOTKADE 2 2725 AR ZOETERMEER 079-314930

SDEK R  
A. BALDUSLN 64 5581 CN WAALRE -

SPIJKERBOER JF  
ROEST 1B 5491 XX ST OEDENRODE -

SPRANG JF V  
TULP 71 2925 EW KRIMPEN AD IJSEL 01807-20569

STAAL H  
ZWIJLWENVELD 25 2727 DH ZOETERMEER 079-412906

STARRENBURG AJ  
WATERGEUSSTR 106B 3025 HW ROTTERDAM -

STEEN MA V  
VINGERHOED 11 6953 BZ DIEREN 08330-21004

STEENDAM SP JR  
MELLENSTEEG 59 9753 HM HAREN 050-345345

STELT MJ VD (KATH. HOGESCHOOL)  
HOGESCHOOLLN 225 5037 CG TILBURG 013-662377

STEVENS F  
MOZARTSTR 204 7604 GZ ALMELO 05490-21112

STOOTS RM  
HOOGSTR 22 6611 BX OVERASSEL -

STRAATHAN F  
BARRIERWEG 215 5622 CN EINDHOVEN 040-431706

STRATEN R V  
BOTERBLOEMSTR 17 2931 TA KRIMPEN AD LEK 01807-13988

SUSAN J  
MARIAPLAATS 10 2713 EC ZOETERMEER 079-165624

SUYKER LH  
SLOETSTR 26 0021 CV ARNHEM 085-5110742

SUYKERBUIJK JW  
ROSLANDERWAARD 40 2904 SM CAPELLE A/D VSEL 010-502431

SWARTS JG  
MARIENSTEIN 175 1852 SM HEILO -

TAEENZER BD  
BORAXSTR 18 9743 VP GRONINGEN 050-774895

TALLON W  
EVENDIJK 14 B-8390 KNOKKE-HEIST -

TEEKENS C  
STADH WILLEM-II LN 36 1411 ER NAARDEN 02159-40908

TERVOOREN JG  
BOSHUIZERKADE 4 2321 TS LEIDEN -

THELOSEN H  
PROVINCIALEW 13 5827 AA VORTUM-MULLEM 08855-127

THIJSSSEN TH  
BROERWEG 26 6522 JD NIJMEGEN 080-236332

TIEL HC V  
J.V. BERUONTSTR 21 2005 RN GOUDA 015-563985

TOLEDO JP  
KUBBESTR 13 3173 GD POORTUGAL 010-160414

TOTTE CT  
GR ALBRECHTWEG 4 3331 HP ZWIJNDRECHT 078-127324

TUENTE HM  
HOVENSTR 10 7091 CL DINXPERLO -

TUINHOUT AJ  
V. HASSELTLN 522 2625 JH DELFT 015-563985

UPHOFF RH  
VOORBURGPAD 10 6843 EM ARNHEM 085-816935

VALK E  
MURALTSTR 14 2461 VG TER AAR 01722-2618

VALSTER CB  
DIKKE STEENWEG 2 7421 RA HOLTEN 05483-2864

VANHOOREN D  
OLLEVIERSLN 52 B-0400 VEURNE -

VEEN A V  
RABBARDESTR 34 2563 RM DEN-HAAG 070-237708

VEEN H  
W. PYRMONTSTR 10 2741 BN WADDINKVEEN 01820-315

VEENSTRA J  
BOLKENLN 26 5363 CR MARUM -

VEIRA X  
BIEDEWEG 91-1 6706 BK MAGENINGEN 08370-22015

VELDKAMP JM  
TIJNARLOBESTR 11 9401 AA VRIES 05921-1951

VERBEEK C  
OORICHTSTR 243 2517 TN DEN-HAAG 070-464104

VERBUIJN NJ  
BETADENSTR 20 3066 EC MEESENKERK 02510-37008

VERHAAR HA  
V. SWIETENSTR 37 2334 EP LEIDEN -

VERHULST RJ  
A. DEKENLN 232 2533 SN DEN-HAAG 070-001504

VERKOOY G  
BRAKKEVELDMEG 44 1782 AH DEN-HELDER 02230-2008

VERMEY F  
MARRAISSTR 15 2021 SJ HAARLEM 023-253981

VERMINNEN JH  
V. IJSENDIJKSTR 128 1442 CS PURMEREND 02990-21739

VERRECHT B  
GR. STEENWEG 15D B-2440 GEEL -

VERSCHEUREN M  
VREDELAAAN 104 B-2610 WILRIJK -

VERZIJDEN IM  
V ALMONDEWEG 26 3231 CS BRIELLE 01810-6719

VETTER B  
TOLAKKERWEG 81 3739 JJ HOLLANSCHER RADING -

VILEYN G  
GILSONSTR 96 B-1090 BRUSSEL -

VISSER AR  
ACASIALN 38 1971 ZZ IJMUIDEN -

VISSER FJ  
TOUSSAINTSTR 7 1814 EG ALKMAAR 072-126652

VISSERS RC  
RIETSTR 47 5482 EW SCHIJNDEL -

VLEESCH-DUBOIS R  
F. NIGHTINGALESTR 212 2037 NG HAARLEM 023-330993

VONK CM  
BREDE HILLEDAK 261 3072 NK ROTTERDAM 010-845945

VONK R  
ST. WILLIBRODUSLN 2 5591 BG HEEZE -

VOOREN G V  
NACHTEGALENLN 43 B-1980 TERVUREN 02-7675139

VOS F  
BREMSTR 35 4341 JL ARNEMUIDEN 01182-2182

VOS F D  
SHAMMERDAMSTR 3 1171 XJ BADHOEVEDORP 02968-4012

VOS G  
ALTEVEER 109 7907 -G HOOGEVEEN -

VOS H D  
HOFSTADESTR 83 B-2600 BERCHEM 031-217484

VRIES H D  
BREDEROSTR 3 6824 PA ARNHEM 085-61609

VRIES JA D  
MEDERIKSTR 156 2403 CJ ALPHEN AD RIJN -

VRIES S D  
BRUGSTR 32 1906 MV LIMMEN 02205-1703

WAARD GH D  
ZERNILKELN 538 3356 XP PAPENDRECHT -

WASSEN A  
MOZARTPLAATS 21 9402 VL ASSEN 05920-4347

WEBER FM  
GEBR WIENERSTR 139 5913 XS VENLO -

WEE PP D  
PR. MARGRIETLN 9 3155 VD MARSLAND 01899-16518

WELGRAVEN JA  
NW MANDEBURTWEW 1056 6717 BK EDE -

WEPPELMAN L  
GRUITERSDREEF 213 7328 ID APELDOORN -

WESSELS A  
VARSEVELDSEWEG 77 7135 JC HARREVELD 05443-2195

WESTERBEEK TA  
DE SCHOKKER 37 1931 DB EGMOND AAN ZEE -

WESTHOFEN D (HTS-LU-ZEE)  
SCHILPLUIDENLN 20 1062 HE AMSTERDAM 020-156795

WESTPHAL CC  
RIETSCHOOT 26 1511 NK OOSTZAAN 02934-1819

WEYZE A D  
J. HARINGSTR 19 2023 KA HAARLEM -

WIETMARSCHEN H V  
CATO ELDERINKLN 3 7602 VR ALMELO -

WIGGERS BJ  
KANTERSHOF 563 1104 HE AMSTERDAM ZO 020-991261

WINKEL H TN  
RODE KRUISLN 985 1111 ZX DIEMEN -

WOLDRING ST  
KLEEVERINK 619 1104 KC AMSTERDAM ZO 020-900085

WOUDBERG RP  
LEIDUINSTR 14-II 1058 SJ AMSTERDAM 020-171866

WOUDE PL VD  
STRIJPELN 8 4047 AM TETERINGEN 076-81363

WOUDBERG JF V  
SILCHERSTRASSE 5 D-7145 MARKGROENINGEN -

WOUSTR FJ  
BEVELSTEIN 306 1182 AK AMSTERDAM ZO -

YIEK KS  
MARKT 20 6041 EL ROERMOND -

ZAGERS AA  
PRINSELN 3 4336 HH MIDDELBURG 01100-11093

ZEELEN J  
JANSTR 30 6045 EN ROERMOND -

ZHAAF AL  
WADDENDIJK 77 1025 PK AMSTERDAM -

ZHIERS G  
ARKELHOF 129 4761 NJ ZEVENBERGEN 01600-24884

KIM KENNER 1 - 12 maart 1977

- Informatie over de Kim-club
- De penningen
- Informatie over de Kim-timer
- Opnemen van Kim programma's op cassette
- De hardware en software bibliotheken

KIM KENNER 2 - 20 juli 1977

- Kim-club nieuws
- De penningen van de Kim-club
- Eenvoudige interfacing van ASCII keyboard aan Kim
- Kim-berg... klimmen of vallen???
- Microprocessor comparison chart
- Lopend display van pseudo alfanumerieke tekens
- Kim I/O routines deel 1
- One armed bandit
- Moon landing program
- Shooting stars
- Music machine 1
- Real digital clock
- Ledenlijst per 10 juli 1977

KIM KENNER 3 - 1 december 1977

- Van het bestuur:
  - . Een jaar lang kim-club
  - . Misleiding
  - . OO hebben we nodig
  - . De penningen
  - . Gezamenlijke inkoop via de Kim-club
- Nieuws:
  - . Onleesbare cassette door interval timer
  - . Single step op de Kim-1 werkt niet
  - . Fout in Kim-user's manual
  - . Attentie voor locaties F1 en F2
  - . Simple big-move
- Nagekomen mededelingen
- Boekbespreking: The first book of Kim
- Micro ade - assembler-disassembler-editor
- MCS 6502 asm/td van micro software specialists
- Vergelijking micro/computer systemen
- 8K ram board aan Kim
- Indicator voor cassette opname/weergave
- DC - DC converter 5V DC in - 12V DC 40MA uit
- Matrox video ram aan Kim
- Conversie datum naar dag van de week
- Kim I/O routines deel 2
- Phase lock loop test programma
- Program for highest speed audiodump
- Basic floating point routines
- Kim-1 debug: extension on Kim-monitor
- Disassembler for the MCS 6502
- Bugs
- Nieuwe leden

KIM KENNER 4 - 10 mei 1978

- Clubnieuws
- Bestuursverandering
- Cursusnieuws
- Samenwerking gezocht
- Kim's grote broer, De pet 2001
- De telex: een goedkope teletype
- Seriele data transmissie
- Improving Kim-1 keyboard reliability
- Kim memory test
- Teletype test programma
- Kim-pagina print programma
- Mastermind op de Kim
- Programma mover
- Papertape loader
- Memory block dump
- Evaluatie van 8K basic (microsoft)
- Bugs
- Binary to BCD conversion

KIM KENNER 5 - 30 sept 1978

- Micromcomputers. Wat doen ze ermee?
- Mass storage media
- Enige interessante trucjes met de Kim
- Kim hints
- De microsoft basic
- Waar blijft de MCS 6509?
- Eprom programmerings systeem
- Structured programming deel 1
- Modificiations and extensions to micro-ade
- Kim als besturingseenheid van een terminal
- Proglover-communicatie-programma
- De telex: een goedkope teletype deel 2
- Statuten van de Kim-club
- Financieel overzicht 1977 - begroting 1978

KIM KENNER 6 - 24 december 1978

- Clubnieuws
- KIM CLUB cursussen
- Inlezen van cassettes van een andere KIM
- Programma voor het afdrucken van memory
- Programma voor het schoonmaken van memory (RAM)
- KIM-1 als digitale 24-uurs klok
- Patches op 8K BASIC van MICROSOFT

KIM KENNER 7 - 5 mei 1979

- Taalproblemen
- Description of the 6516 microprocessor
- Using KIM as a dedicated controller
- DA and AD conversion using KIM
- Important adresses of KIM-1 and monitor
- Software routines voor TVT
- S-100 to KIM-4 bus adapter
- Patches op Micro Ade
- Ledenlijst per 270379
- Balans en ontwerpbegroting 1979

KIM KENNER 8 - 20 augustus 1979

- Target ( 1-column )
- Target ( 6-column )
- Aanvullingen op Microchess 1.0
- Patches op Micro Ade

KIM KENNER 9 - 1 december 1979

- Cassette bibliotheek
- Wordprocessing
- Single step debug programma
- Vergelijking van 3 rekenpakketten
- Microcomputers
- Datum subroutine
- Automatische hex displayer
- Tape handling programma

KIM KENNER 10 - 1 maart 1980

- Aanwijzingen bij het schrijven van een artikel door de KIM KENNER
- Een handige schakeling
- Cassette lees- en schrijfindikatie
- Overzicht van verkrijgbare hardware schema's
- Hardware fout in de MCS 6502
- Voor en nadelen van Tiny BASIC + tips
- Microsoft BASIC
- Wat doe ik met mijn KIM
- Verslag KIM club bijeenkomst dd. 171179
- Verslag KIM club bijeenkomst dd. 190180
- Cassette zoekprogramma
- Oprichting van de beleidscommissie
- Galgje
- Wat doe ik met mijn KIM?

## KIM KENNER 11 - 17 mei 1980

- Schaakprogramma
- Patches op BASIC deel 2
- I/O routines voor BASIC
- Wat doe ik met mijn KIM
- Printer voor de KIM
- Keyboard voor de KIM
- RS232 naar TTL converter
- Verslag KIM club bijeenkomst dd. 150350
- Datacommunicatie

## KIM KENNER 12 - 16 augustus 1980

- De JUNIOR
- Memory test ( een echte )
- Patches op BASIC
- Supertape
- Schaakprogramma
- aktiesnelheidsmeter
- Automatiseren van modelspoor
- Bugs

## KIM KENNER 13 - 18 oktober 1980

- Hex teller en flip flop voor de JUNIOR
- Ervaringen met de JUNIOR
- Talestudie hulp programma
- Printer output RS 232 routine
- Cassette interface
- Voeding cassette recorder
- De CBM 2001
- Modeltrein simulatie
- Goochelen met de KIM

## KIM KENNER 14 - 20 december 1980

- Junior
- One Armed Bandit (Fruitmachine) voor de Junior
- Ervaringen met de Junior
- Junior lees- en schrijfroutine voor Kansas City interface
- PET initialisator
- RAE Bootstrap for printer
- KIM Schaakprogramma
- APPLE tapes met de KIM
- Patches op MICROSOFT BASIC
- Floppy disk aanwijzingen

## De 6502 Kenner 15- 15 maart 1981

- Nauwkeurige frequentie-standaard met een CBM/PET
- Ervaringen met de JUNIOR
- Letters en cijfers op JUNIOR display
- Microline - 80 interface voor de SY4
- Uitbreidingen met Galgwoorden
- Lotto
- Chronometer
- Moonlanding
- Automatische hex displayer
- Bugs in CBM en interface

## De 6502 Kenner 16- mei 1981

- Cassette bibliotheek
- Verkeersregeling
- ASCII toetsenbord encoder
- Ervaringen met de Junior
- Automatische register uitlezing
- Shooting Stars
- Browse
- Het slangetje en de 6532
- CBM geheugen uitbreiding
- Patches op Microsoft Basic

## De 6502 Kenner 17- augustus 1981

- Music, muziekprogramma voor KIM
- Dokatimer
- Ervaringen met de Junior
- Echo van toetsenbord onderdrukken
- Patches op Q-Chess, een schaakprogramma
- BASIC Telebingo, naar het gelijknamige TV-spel
- Reset zonder memory test
- Lichtkrant
- Schaakprogramma patches
- Patches op Micro Ade
- Muziekdoos voor JUNIOR
- Informatie over The 6502 Software Exchange
- Displayer
- Informatie over BASIC MUX
- Cassette bibliotheek
- Literatuur voor 65xx gebruikers.

## De 6502 Kenner 18 - oktober 1981

- Nieuwe MOS Technology chips
- Eprom programmer
- Locate en Replace
- SC/MP cassette interface-afregeling
- Conversie van datum naar dag
- Tape copying programma
- Zenuwslag
- Belgische Lotto
- Informatie over The 6502 Software Exchange en The Computerist
- Cassette bibliotheek
- Literatuur voor 65xx gebruikers

## De 6502 Kenner 19 - december 1981

- Schrijf- en Leesroutine voor de JUNIOR (uitbreiding van de versie in KIM KENNER 14)
- Rapportcijfer
- Display op oscilloscoop
- Aanpassingen aan de Macro-Assembler van C.W. Moser
- Patches op BASIC
- Lichtshow voor de JUNIOR en de KIM
- 4-Koloms printer
- Printer routine voor H14
- Break voor JUNIOR en KIM
- Hex dump
- Cassette bibliotheek

THE JUNIOR\_PROMOTING CY {W&J} 231281 PAGE 01

0010:	ZA	03-04-82	OSI GEBRUIKERSGROEP DAG DONGEN
0020:			
0030:	ZA	15-05-82	KIM CLUB DAG
0040:			VERSCHEIJNINGSDATUM 6502-KENNER
0050:			OSI GEBRUIKERSGROEP DAG
0060:			
0070:	ZA	21-08-82	VERSCHEIJNINGSDATUM 6502-KENNER
0080:			
0090:	ZA	11-09-82	APPLE GEBRUIKERS GROEP DAG
0100:			
0110:	ZA	18-09-82	KIM CLUB DAG
0120:			OSI GEBRUIKERSGROEP DAG
0130:			
0140:	ZA	16-10-82	VERSCHEIJNINGSDATUM 6502-KENNER
0150:			
0160:	ZA	20-11-82	KIM CLUB DAG
0170:			OSI GEBRUIKERSGROEP DAG
0180:			
0190:	ZA	18-12-82	VERSCHEIJNINGSDATUM 6502-KENNER
0200:			
0210:			'S ZONDAGS VAN 18.30 TOT 19.00 UUR VIA HILV.I
0220:			HOBBYSCHOOP, VAAK MET COMPUTERPROGRAMMA'S, DIE
0230:			M.B.V. EEN CASSETTERECORDER ZIJN OP TE NEMEN.
0240:			DOOR HET GEBRUIK VAN DE 'BASICODE' IS ELK
0250:			UITGEZONDEN PROGRAMMA MET UW COMPUTER IN TE
0260:			LEZEN.

## BUGS

In de 6502 Kenner nr. 18, zo meldt ons de auteur Haijo Hemminga, in het programma locate en replace, zijn door de redactie een paar foutjes gemaakt. Op de volgende regelnummers moet U het daarachter vermelde lezen:

0700:	CMPAY LARG	COMPARE TO ARGUMENT
0760:	CMPAY LARG	
1020:	REPB LDAAY RARG	

Het vervelende van deze fout is dat we op de betreffende regels drie-bytes instructies krijgen, i.p.v. twee-bytes, waardoor telkens de rest een byte moet opschuiven, branches moeten worden aangepast etc. etc.

Voor de gepubliceerde versie is de gebruiksaanwijzing niet in overeenstemming met de listing, maar na het aanbrenge van bovengenoemde wijzigingen wel het enige dat wij juist hebben overgenomen. In 6502 Kenner nr. 21 zullen wij het programma nogmaals afdrucken, maar dan hopelijk meteen goed.

KORTGELEDEN WERD ONZE AANDACHT GEVESTIGD OP EEN DOOR DE FIRMA NAMAL ELECTRONICS UIT ENGELAND IN ELEKTUUR GEPLAATSTE ADVERTENTIE. DAARIN WERDEN RAM'S EN EPROM'S GOEDKOOP AANGEBODEN. BERT VAN TIEL NAM TELEFONISCH KONTAKT OP MET DEZE FIRMA EN ZETTE VOOR ONS ZIJN INFORMATIES OP PAPIER

NAMAL ELECTRONISC  
NO. 1 CLAYGATE ROAD  
CAMBRIDGE, U.K.,  
TEL: 0223-248257

PRIJZEN: 1. 2716 FL. 8.35 450NS EPROM  
2. 2732 FL. 18.50 450NS EPROM  
3. 2114L FL. 4.10 200NS S/RAM  
4. 6116 FL. 21.42 150NS CMOS S/RAM  
5. 4116 FL. 3.90 150/200NS DRAM

MERKEN: 1. NEC, MOTOROLA  
2. JUJITSU  
3. TOSHIBA

LEVERINGS-  
VOORWAARDEN: 1. ORDERS: BIJ VOORKEUR SCHRIFTELIJK  
I.V.M. KANS OP FOUTEN  
(ADRES BIJV.)  
2. BETALEN: MET INGESLOTEN EUROKAART  
O.I.D. (BIJV. INTERNATIO-  
NAAL GEACCEPTTEERD)  
3. KOSTEN: FL.25,= VERPAKKING +  
VERZENDING ONGEACHT  
BESTELGROOTTE.  
4. INVOER: 18% VAN DE WAARDE

VOORBEELD 1: 16K X RAM 2114 A FL. 4.10 FL. 65.60  
VRACHT+VERPAKKING FL. 25.00  
BTW 18% VAN FL. 65.60  
(TE BETALEN IN NEDERLAND) FL. 11.81

8K/S/RAM/200NS/2114 TOTAAL: FL. 102.41  
=====

VOORBEELD 2: 4 X EPROM 2732 A FL. 18.50 FL. 74.00  
VRACHT+VERPAKKING FL. 25.00  
BTW 18% VAN FL. 74.00  
(TE BETALEN IN NEDERLAND) FL. 13.32

16K/EPROM/450NS/2732 TOTAAL: FL. 112.52  
=====

BIJ GROTE HOEVEELHEDEN WORDT DE PRIJS BEPAALD DOOR DE KALE PRIJS + 18% BTW EN EEN PAAR DUBBELTJES PER STUK (VERZ. + VERP.). BIJV. 1 X 2732 KOMT DAN OP FL. 18.50 + 18% = FL. 21.83.

301181

DE **KIM** GEBRUIKERS CLUB  
NEDERLAND

EEN CLUB VAN

**6502** KENNERS