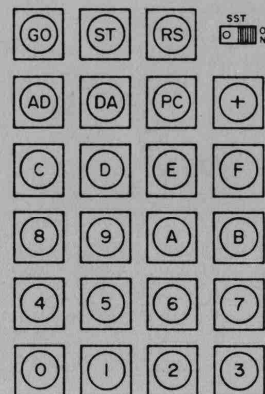


KIM GEBRUIKERS CLUB NEDERLAND
4de Jaargang Nr. 14
20 december 1980

0000 14

DE KIM KENNER 14



GEBRUIKERS CLUB NEDERLAND

Samenstelling van het bestuur:

Voorzitter	: Co Filmer Dorpsstr 1051 1566 JE ASSENDELFT Tel.: 075 - 210023
Sekretaris en ledenadministratie	: Anton Müller Sinj Semeynsstr 78 1 1061 GM AMSTERDAM Tel.: 020 - 860245
Penningmeester	: Ted Schouten Junoplnts 57 2024 RM HAARLEM Tel.: 023 - 257171 Postgirorek.nr.: 3757649
Regeling accommodatie voor KIM-club bijeenkomsten	: Bob van de Oudewetering Industriewg 12 2102 LM HEEMSTEDÉ Tel.: 023 - 286444
Technisch adviseur, cassette programma bibliotheek en propaganda KIM-club	: Uwe Schröder Echternachln 161 5625 KC EINDHOVEN Tel.: 040 - 421821
Software adviseur en regeling programma van KIM-club bijeenkomsten	: Sebo Woldringh Klieverink 619 1104 KC AMSTERDAM ZUIDOOST Tel.: 020 - 900085
Organisatie, hardware en beheer KIM-club-KIM	: Rinus Vleesch Dubois F Nightingalestr 212 2037 NG HAARLEM Tel.: 023 - 330993

DE KIM KENNER

KIM INHOUDSOPGAVE

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

Adres voor het in-
zenden van en re-
akties op artike-
len voor de KIM
KENNER:

p/a H.J.C. Otten
Dr Schaepmanstr 15
1381 BG WEESP
Tel.: 02940-13349

Redactie KIM KENNER:
Anton Muller
Hans Otten
Peter Visser

Geheel of gedeelte-
lijke overname van
de inhoud van de
KIM KENNER zonder
toestemming van
het bestuur is ver-
boden.

Toepassen van gepu-
bliceerde programma's,
hardware etc. is al-
leen voor persoonlijk
gebruik toegestaan.

© 1980 by KIM Gebrui-
kers club Nederland.

Verschijnt vijfmaal
per jaar.

Inhoudsopgave

Inhoudsopgave	Pagina:
Inhoudsopgave	1
Van het bestuur	2
Van de Redactie	3
JUNIOR H J C Otten	4
One armed bandit for the junior (F J Butterfield)	8
Ervaringen met de Junior (J van Sprang, W L van Pelt)	12
JUNIOR Leesroutine voor Kansas City cassette interface (Sebo Woldringh)	14
JUNIOR Schrijfroutine voor Kansas City cassette interface (Sebo Woldringh)	16
PET Initialisator (R Uphof)	18
System software: RAE printer bootstrap	26
KIM Schaakprogramma (T Kortekaas)	27
Apple tapes met de KIM (F Mesander en H J C Otten)	36
Patches op Microsoft BASIC (HJC Otten)	39
FLOPPY DISC AANWIJZINGEN	40

KIM VAN HET BESTUUR

KIM - GEBRUIKERSCLUB

Forum

Op bepaalde bijeenkomsten wordt in het programma ruimte gereserveerd voor een forum.

Doel van een forum

Parate beschikbare kennis en informatie mee te geven in antwoord op de door de leden gestelde vragen. De leden kunnen tijdens het FORUM-gebeuren hun problemen, t.w. hard- en/of software de vrije loop laten. Tevens kunnen ideeën en voorstellen gelanceerd worden.

Voordeel van deze directe communicatie is praktische oplossingen voor de vragensteller, zodat deze sneller vooruit komt. Wederom is gebleken dat er binnen de KIM-club veel kennis aanwezig is (ook voor andere 6502 systemen).

Een levendige discussie kwam op gang bij KIM compatible JUNIOR, waarbij forumleden Hans en Anton onze leden volstopten met informatie om Kimprogrammas op de junior te kunnen draaien.

Samenstelling forum september was:

Hans Otten - Anton Muller - Rinus Vleesch Dubois - Wim van Gelderen - Uwe Schroder - Sebo Woldringh - Co Filmer.

KIM-MARKT VOOR U BEKEKEN

Zoals gewoonlijk wordt er op elke bijeenkomst enige tijd voor de "MARKT" ingeruimd. T.a.v. de normale gang van zaken en spelregels:

1. Aanmelden dient voor de aanvang van de "MARKT" te geschieden, zodat een ieder op volgorde van aanmelding aan de beurt komt.
2. Afhankelijk van het aantal personen die iets hebben aan te bieden, wordt de beschikbare tijd verdeeld in "x" minuten per persoon.
3. Alles wat met micro's, terminals, onderdelen, boeken, tijdschriften, software, of wat hiermede in verband gebracht kan worden (in de ruimste zin van het woord), kan aangeboden worden.
4. De KIM-club cq. Bestuur kan niet verantwoordelijk of aansprakelijk gesteld worden voor o.a.:
 - . Goede werking van het aangeboden goed.
 - . Defecten en/of garantie.
 - . Plaats van herkomst en eventuele gevolgen.
 - . Afspraken betreffende leveringen, levertijden, betalingen, enz.

Globale opsomming van het gebodene tijdens "meeting september te Heemstede"

Hr. v.d. Burg	Teletype linten
Hr. Visser	Prof. 6502 systemen/printen voor videoterminals/8K ramboards op kimformaat/losse chips
Hr. Meijer	Nummerdisplays 0-9 incl latch-Kims + dokumentatie
Hr. Schroder	Boeken, OSI board, Applecomputers, OKI printers, Macro-assembler, Dynamic ramboards
Hr. Verkooy	Printers (gebruikt)

VAN DE REDACTIE

Redactioneel voorwoord bij KIM KENNER 14

H.J.C. Otten

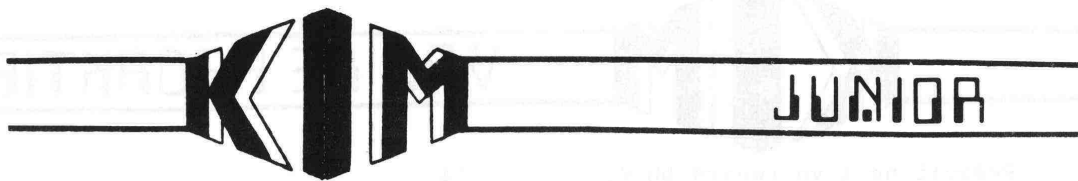
Dit is het vijfde en laatste nummer van de KIM KENNER in 1980.

Dit nummer ontvangt U in de maand vol feestdagen december , namens redactie en bestuur wens ik U prettigefeestdagen en de beste wensen voor 1980 . Als we zoals gebruikelijk in deze tijd terugkijken op het afgelopen jaar , dan is dat voor de KIM 6502 club een goed jaar geweest . In 1980 is de KIM club uitgegroeid tot een echte 6502 gebruikersclub met een flinke groei in ledental . We hebben vele nieuwe enthousiaste leden met allerlei 6502 systemen. Ook in 1981 zal de KIM club vele activiteiten kunnen ontwikkelen, waarbij ook Uw bijdrage welkom en nodig is . Een club als de KIM club wordt geheel gedragen door de leden , waarbij de bijdragen van gevorderden als beginners nodig zijn. Als we allemaal ons best doen dan kunnen we onze interesse , de computer , door gezamenlijke inspanning voor ieder inhoud geven.

Een van de belangrijkste activiteiten van de KIM club is de tweemaandelijksse bijeenkomst. Een belangrijk onderwerp van de bijeenkomst op 17 januari zal de taal FORTH en de implementatie daarvan op allerlei 6502 systemen zijn. We zijn er erg blij mee dat we een bekend deskundige , Dr. Hans Nieuwenhuyzen , secretaris van de Europese Forth Gebruikers Groep , bereid hebben gevonden daar een ongetwijfeld leerzame en interessante inleiding in de vorm van een lezing over te willen houden. Na deze voor ieder te volgen lezing zal voor de meer geïnteresseerden een workshop onder leiding van Dr. H. Nieuwenhuyzen worden gehouden. In ieder geval kan dan met Forth op Apple's met een diskdrive worden gewerkt, neem Uw Apple gerust mee daarvoor . Later kan Forth op allerlei 6502 systemen draaien .

Forth is een hogere programmeertaal met uitstekende eigenschappen voor real time controle toepassingen , en stelt daarbij bescheiden eisen aan intern en extern geheugen . Ik raad U aan van te voren Byte van augustus 1980 en met name het artikel 'What is Forth ?' te lezen en ook de Byte's van september en oktober over Forth.

Het voordeel van een onafhankelijke club zoals de KIM club is dat we samen sterk staan. Laat daarom Uw positieve of negatieve ervaringen met aankoop van hard- en software weten aan Uw medeclubleden.



Conversie van KIM naar JUNIOR en omgekeerd

H.J.C.Otten

De gelijkenis tussen de KIM en Junior hard- en software is dusdanig groot , dat het zonder gecompliceerde programma - wijzigingen mogelijk is software van de ene aan de andere computer aan te passen. Daarbij is echter documentatie nodig van die andere computer .

KIM bezitters (en ook andere 6502 gebruikers) zou ik aanraden de Junior boeken te kopen , vooral deel 2 met de junior monitor is goed leesbare en interessante literatuur over de 6502 en natuurlijk de Junior.

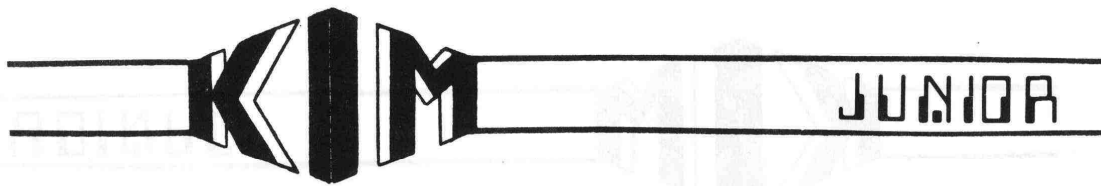
Mijn eerste indruk van de Junior was dat het een kopie van de KIM was . Wat betreft de hardware is dit ook zonder meer waar , er is één detail waarin de Junior en de KIM duidelijk verschillen en dat is de wijze van aansturen van de LED display's. Het lijkt precies op elkaar , alleen de polariteit is omgedraaid. Als we bij de KIM een één naar een LED-segment sturen , dan moet bij de Junior een nul worden gezonden . Verder is de werking identiek.

Dit verschil uit zich duidelijk in de hex- naar zevensegment tabel , de tabel in de Junior (1F0F en verder) is gelijk aan die van de KIM (1FE7 en verder) maar bit voor bit geïnverteerd !

Junior gebruikers kunnen een grote hoeveelheid programma's vinden in het "First book of KIM" , maar stuiten dan op onbekende KIM routines , waarvoor ze de KIM manuals moeten raadplegen

Om de conversie te ondersteunen zal ik een aantal overeenkomstige KIM en Junior routines aanwijzen die veel gebruikt zullen worden . Hieruit blijkt dat de software ook grote overeenkomsten vertoont, een aantal subroutines heeft dezelfde naam en coding. Er zijn echter een aantal opmerkelijke verschillen in de software .

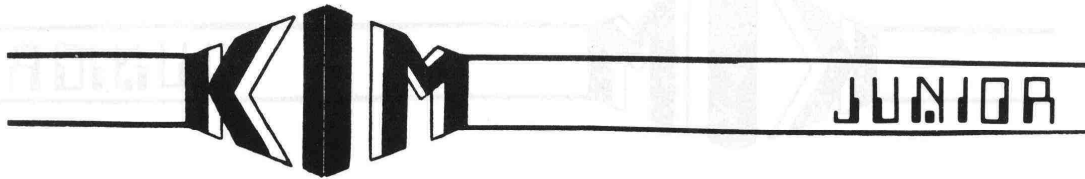
De KIM heeft meer I/O mogelijkheden zoals een teletype en een audio cassette recorder , voorzieningen die bij de Junior ongetwijfeld zullen komen. Een raadgeving : wacht op de ontwikkelingen bij Elektuur en ga niet zelf ontwikkelen of kopen , daar krijgt U spijt van. Doe zoveel mogelijk gemeenschappelijk !



De Junior heeft een erg leuk uitgevoerde monitor wat betreft het invoeren en uitproberen van programma's met het toetsenbord en de display's, wat in de KIM niet aanwezig is .

Overeenkomstige routines

KIM	JUNIOR
1. \$1C00 - SAVE Bewaar registers op zero page en start monitor	\$1C00 - SAVE Zelfde , met zelfde zero page lokaties etc
2. \$1C1D - RESET zet poorten goed en start monitor , na indrukken reset toets	\$1C22 - RESET gelijk aan KIM
3. \$1F19 - SCAND toont dat op display van FA,FB loopt over in AK	\$1D88 - SCAND gelijk KIM
4. \$1F1F - SCANDS toont inhoud van FA,FB,F9 op display als 3 x 2 hex getallen loopt over in AK	\$1D8E - SCANDS gelijk aan KIM , maar inhoud \$00F6 (BYTES) bepaalt hoeveel bytes worden getoond (1,2 of 3) , KIM toont altijd 3 bytes (= zes diplay'
5. \$1C77 - START start monitor , wachten op toetsindruk	\$1C33 - START analoog
6. \$1F6A - GETKEY haal toets van keyboard A= toetsnummer ,A>15 onjuiste toets indruk	\$1DF9 - GETKEY zelfde functie, toetsen zelfde nummering als bij KIM
7. \$1EFE - AK Bepaal of toets ingedrukt A ≠ 0 dan ingedrukt	\$1DB1 - AK gelijk



PIA lokaties

Bij de KIM hebben we twee PIA's in de twee 6530 IC's . In de Junior zit er een in de 6532 . Ze gedragen zich gelijk , die 6530 en 6532 als PIA , maar ze zitten op andere lokaties.

KIM PIA 1700 -1703 ontbreekt bij de Junior .

KIM PIA 1740 -1743 = Junior PIA 1A80-1A83 , vrijwel gelijk gebruikt.

PAD KIM is bijvoorbeeld PAD Junior : 1740 = 1A80 .

Timers

De 6530 en 6532 hebben gelijke timers , hiervoor is het beste de voor iedere serieuze 6502 gebruiker onontbeerlijke Program en HardwareManuals van de fabrikant te raadplegen .

Timerlokaties 1704 etc van de KIM ontbreken bij de Junior .

Overeenkomstige timers zijn de KIM lokaties 1744 t/m 174F en de Junior lokaties 1AF4 t/m 1AFF (bv CNTA junior 1AF4 = KIM CLK1T 1744 etc) .

De edge detect mogelijkheid van de Junior ontbreekt bij de KIM .

Vectoren

Interrupt vector is bij de KIM de RAM lokatie 17FE en 17FF , bij de Junior 1A7E en 1A7F .

Non maskable vector is bij de KIM 17FA en 17FB , bij de Junior is dit 1A7A en 1A7B .

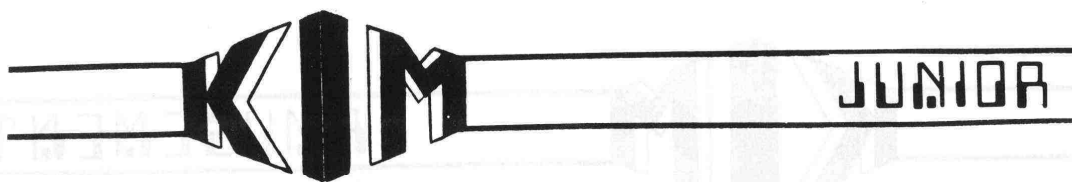
De break vector (instructie BRK) is bij de KIM de interrupt vector 17FE etc , bij de Junior 1A7E etc.

RAM lokaties

De KIM heeft RAM vanaf adres 0000 t/m 03FF evenals de Junior .

Daarnaast heeft de KIM RAM (van de 6530'S) van 1780 t/m 17FF .

De Junior heeft RAM van 1A00 t/m 1A7F , ook 128 bytes



GETBYT routine van de JUNIOR

In de Junior zit een routine ,GETBYT genaamd , die twee toets-indrukken afwacht en ze combineert tot een hex getal in de accumulator . Zijn er geen hex toetsen ingedrukt (0 .. F) dan is de N flag 0 , anders 1 . In de Junior boeken wordt dankbaar gebruik gemaakt van deze routine , onder andere in de branch berekenings routine BRANCH = 1FD5 (overigens is deze routine een bewerkte versie uit het First book of KIM !) . Voor geïnteresseerden herhaal ik deze JUNIOR GETBYT routine uit het Junior boek 2 van Elektuur : (in assembler notatie !)

```
GETBYT JSR SCANA    read high order nibble
      CMPIM $10
      BPL BYTEND    command key ?
      ASLA
      ASLA          if not save high order nibble
      ASLA
      ASLA
      STA NIBBLE    ( bv OOFÉ )
      JSR SCANA
      CMPIM $10
      BPL BYTEND    command key ?
      ORA NIBBLE    if not , compose byte
      LDXIM         set N=1
BYTEND RTS
SCANA JSR SCANDS    display
      BNE SCANA     key released ?
SCANB JSR SCANDS    display
      BEQ SCANB     any key          ?
      JSR SCANDS    display
      BEQ SCANB     any key still depressed ?
      JSR GETKEY    if yes , return with key in accu
      RTS
```

Tot zover de conversie hulp . Heeft U andere conversie problemen stuur me dan een briefje . Heeft U met succes iets omgezet , laat het dan aan de KIM KENNER weten voor Uw medegebruikers ..

KIM AMUSEMENT

```

0010. *****
0020. *
0030. *   ONE ARMED BANDIT FOR THE JUNIOR MICRO COMPUTER
0040. *   -----
0050. *
0060. *   ORIGINALLY A KIM-1 VERSION, MODIFIED BY
0070. *   ANTON MUELLER
0080. *   SINJ SEMEYNSSTR 78 1
0090. *   1061 GM AMSTERDAM
0100. *   THE NETHERLANDS
0110. *   IN ORDER TO MAKE IT SUTABLE FOR THE JUNIOR
0120. *
0130. *   THE PROGRAM INTERFACES WITH THE JUNIOR KEY-
0140. *   BOARD AND DISPLAYS TO PRODUCE A "ONE
0150. *   ARMED BANDIT" TYPE OF SLOT MACHINE ACTIVITY.
0160. *   THE RIGHT-HAND SIDE OF THE DISPLAYS SHOWS
0170. *   THE PLAYER'S BALANCE IN DUTCH FLORINS.
0180. *
0190. *   AN EARLIER VERSION OF THIS PROGRAM ALLOWS
0200. *   THE WHEELS TO SPIN UNTIL THEY ARE STOPPED,
0210. *   ONE AT A TIME, BY THE PLAYER TOUCHING A KEY.
0220. *   WHILE THE EFFECT WAS GOOD, THEY LED TO
0230. *   EXTREMELY HEAVY USEAGE OF THE PUSHBUTTONS.
0240. *   THIS VERSION CAUSES THE WHEELS TO STOP AUTO-
0250. *   MATICALLY.
0260. *
0270. *   MUCH OF THE PROGRAM IS DEVOTED TO PRODUCING
0280. *   THE "FEEL" OF THE SIMULATED MACHINE. IT
0290. *   WOULD BE VERY EASY, OF COURSE, TO HAVE THE
0300. *   WHEELS START AND STOP INSTANTANEOUSLY, AND
0310. *   THE CASH BALANCE DISPLAYED IMMEDIATELY:
0320. *   BUT THIS WOULD NOT RESEMBLE THE REAL
0330. *   MECHANICAL MACHINE.
0340. *
0350. *   THERE IS NO HOUSE PERCENTAGE, SO THE
0360. *   PLAYER CAN GO FOR CONSIDERABLE TIME ON HIS
0370. *   INITIAL 25 GULDERS. HOWEVER, HE IS NOT ALLOWED
0380. *   TO WIN A TOTAL OVER 99 GULDERS; AND THE
0390. *   WHEELS WILL NOT SPIN IF HE LOSES ALL HIS
0400. *   MONEY.
0410. *   THE PROGRAM STARTS AT LOCATION 0200.
0420. *   SPIN THE WHEELS BY TOUCHING ANY BUTTON
0430. *   BETWEEN 0 AND F.
0440. *
0450. *   AUTHOR: F. J. BUTTERFIELD
0460. *   ----- 14 BROOKLYN AVENUE
0470. *   TORONTO - ONTARIO M4M2X5
0480. *   C A N A D A
0490. *   -----
0500. *

```

KIM AMUSEMENT

```

0510:                                *      WORK AREA'S IN PAGE ZERO
0520:                                *
0530: 0200      BANDIT ORG      $0200
0540: 0200      WINDOW *      $0000
0550: 0200      AMT *          WINDOW +05
0560: 0200      ARROW *          AMT +01
0570: 0200      RWD *          ARROW +01
0580: 0200      STALL *          RWD +01
0590: 0200      TUMBLE *          STALL +01
0600:                                *
0610:                                *      REFERENCES TO JUNIOR MONITOR
0620:                                *
0630: 0200      SCDSB *          $1DAC IS KEY DEPRESSED?
0640: 0200      PADD *          $1A81
0650: 0200      PAD *          $1A80
0660: 0200      PBD *          $1A82
0670: 0200      LOOK *          $1F0F
0680:                                *
0690:                                *      MAIN PROGRAM STARTS HERE
0700:                                *
0710: 0200 A9 25      GO      LDAM #25      ASSUME THE PLAYER
0720: 0202 85 03      STA      AMT      STARTS WITH 25 BUCKS
0730: 0204 20 BA 02   JSR      CVAMT     CHANGE TO DISPLAY
0740: 0207 A9 00      LDAM #00
0750: 0209 85 06      STA      ARROW
0760:                                *      MAIN DISPLAY LOOP
0770: 020B 20 8D 02   LFA      JSR      DISPLY
0780: 020E D0 FB      BNE      LFA
0790: 0210 E6 09      ROLL     INC      TUMBLE
0800: 0212 20 8D 02   JSR      DISPLY
0810: 0215 F0 F9      BEQ      ROLL
0820: 0217 A9 03      LDAM #03      PAY 2 BUCKS
0830: 0219 85 06      STA      ARROW
0840: 021B F8          SED
0850: 021C 38          SEC
0860: 021D A5 05      LDA      AMT
0870: 021F E9 01      SECIM #01      CHARGE A BUCK
0880: 0221 85 05      STA      AMT
0890: 0223 20 BA 02   JSR      CVAMT
0900: 0226 26 09      ROL      TUMBLE
0910: 0228 20 8D 02   JSR      DISPLY
0920: 022B C6 08      DEC      STALL      WAIT
0930: 022D D0 F9      BNE      LFB
0940: 022F A6 06      LDX      ARROW
0950: 0231 A5 09      LDA      TUMBLE
0960: 0233 09 F9      ORAIM #F9
0970: 0235 29 BF      ANDIM #BF      SPIN RESULT
0980: 0237 95 01      STAAX WINDOW +01 TO DISPLAY
0990: 0239 46 09      LSR      TUMBLE
1000: 023B 46 09      LSR      TUMBLE

```

KIM AMUSEMENT

```

1010: 023D C6 06      DEC  ARROW
1020: 023F D0 E7      BNE  LFB
1030:                *  ALL WHEELS STOPPED - COMPUTE PAYOFF
1040: 0241 A0 04      LDA  WINDOW +04
1050: 0243 C3 03      CMP  WINDOW +03
1060: 0245 D0 57      BNE  NOMAT
1070: 0247 C3 02      CMP  WINDOW +02
1080: 0249 D0 53      BNE  NOMAT
1090: 024B A2 10      LDXIM #10
1100: 024D C9 BF      CMPIH #BF      PAY 15 BUCKS IF 3 BARS
1110: 024F F0 0D      BEQ  PAY
1120: 0251 A2 08      LDXIM #08
1130: 0253 C9 BD      CMPIH #BD      PAY 10 BUCKS IF 3 UPS
1140: 0255 F0 07      BEQ  PAY
1150: 0257 A2 06      LDXIM #06
1160: 0259 C9 B5      CMPIH #B5      PAY 5 BUCKS IF 3 DOWNS
1170: 025B F0 01      BEQ  PAY
1180: 025D CA      DEX
1190:                *  A WIN. PAY AMOUNT IN X
1200: 025E 86 07      PAY  STX  RWD
1210: 0260 A9 80      PAX  LDANH #80
1220: 0262 8D 08      STA  STALL
1230: 0264 20 8D 02  LFC  JSR  DISPLY
1240: 0267 C6 08      DEC  STALL
1250: 0269 D0 F9      BNE  LFC
1260: 026B C6 07      DEC  RWD
1270: 026D F0 9C      BEQ  LPA
1280: 026F 18      CLC
1290: 0270 F6      SED
1300: 0271 A0 00      LDA  AMT
1310: 0273 67 01      ADCIH #01
1320: 0275 B0 94      BCS  LPA
1330: 0277 8D 00      STA  AMT
1340: 0279 20 BA 02  LOK  JSR  CVANT
1350: 027C D0 E2      BNE  PAX
1360:                *  WHEELS NOT ALL SAME - CHECK FOR SMALL WIN.
1370: 027E A2 03      NOMAT LDXIM #03      PAY 2 BUCKS
1380: 0280 C9 B9      CMPIH #B9      IF CHERRY
1390: 0282 F0 DA      BEQ  PAY
1400: 0284 20 8D 02  LOK  JSR  DISPLY
1410: 0287 A0 00      LDA  AMT
1420: 0289 D0 80      BNE  LPA
1430: 028B F0 F7      BEQ  LOK
1440:                *  DISPLAY SUBROUTINE
1450:                *
1460: 028D A6 06      DISPLY LDX  ARROW
1470: 028F 10 02      BPL  INDIS
1480: 0291 F6 02      OVER INCAX WINDOW +02
1490: 0293 CA      INDIS DEX
1500: 0294 10 FB      BPL  OVER

```


KIM AMUSEMENT

1510:	0296	A9 7F		LDAM #7F
1520:	0298	8D 81 1A		STA PADD
1530:	029B	A0 0B		LDYIM #0B
1540:	029D	A2 04		LDXIM #04
1550:	029F	B5 00	LITE	LDAAX WINDOW
1560:	02A1	8C 82 1A		STY PBD
1570:	02A4	8D 80 1A		STA PAD
1580:	02A7	D8		CLD
1590:	02A8	A9 7F		LDAM #7F
1600:	02AA	E9 01	ZIP	SECIM #01
1610:	02AC	D0 FC		BNE ZIP
1620:	02AE	8D 82 1A		STA PBD
1630:	02B1	C8		INY
1640:	02B2	C8		INY
1650:	02B3	CA		DEX
1660:	02B4	10 E9		BPL LITE
1670:	02B6	20 AC 1D		JSR SCDSB
1680:	02B9	60		RTS
1690:			*	
1700:			*	AMOUNT CONVERSION
1710:			*	
1720:	02BA	AC 05	CVANT	LDA ANT
1730:	02BC	29 0F		ANDIM #0F
1740:	02BE	AA		TAX
1750:	02BF	BD 0F 1F		LDAAX LOOK
1760:	02C2	8C 00		STA WINDOW
1770:	02C4	AC 05		LDA ANT
1780:	02C6	4A		LSRA
1790:	02C7	4A		LSRA
1800:	02C8	4A		LSRA
1810:	02C9	4A		LSRA
1820:	02CA	AA		TAX
1830:	02CB	BD 0F 1F		LDAAX LOOK
1840:	02CE	8C 01		STA WINDOW +01
1850:	02D0	60		RTS

SYMBOL TABLE 3800 38A8							
AMT	0005	ARROW	0006	BANDIT	0200	CVANT	02BA
DISPLY	028D	GO	0200	INDIS	0293	LITE	029F
LOK	0284	LOOK	1F0F	LPA	020B	LPB	0228
LPC	0264	NGMAT	027E	OVER	0291	F	02D1
PADD	1A81	PAD	1A80	FAX	0260	FAY	025E
PBD	1A82	ROLL	0210	RWD	0007	SCDSB	1DAC
STALL	0008	TUMBLE	0009	WINDOW	0000	ZIP	02AA

KIM JUNIOR

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

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

Ervaringen met de JUNIOR

De ervaringen, wensen en verzuchtingen die in een vorige KIM KENNER zijn vermeld vormen uiteraard geen afgerond geheel. Vandaar deze voortzetting.

Bemerkt is dat door ook deze publikatie kontakten zijn ontstaan en verstevigd. Inmiddels hebben bovengenoemden elkaar gevonden in een gemeenschappelijk probleem: de behoefte om programma's op de band te zetten, zodat een hoop toetswerk eenmalig kan blijven. Vanwaar die behoefte, als er nog geen JUNIOR-programma's zijn verschenen? Dat bleek niet zo'n nood, want via de sekretaris verkregen we inzicht in het verschil tussen KIM en JUNIOR monitor routines. Niet dat daarmee alles kan worden vertaald in JUNIOR, maar een paar programma's zijn nu in orde. Ook door kontakten met andere leden elders konden over en weer gegevens worden uitgewisseld. Wij hopen dat het bestuur die verschillen, maar dan wel alle, tussen KIM en JUNIOR, in de KIM KENNER zullen publiceren, zodat een ieder daar profijt van heeft. (Is reeds gebeurd. Zie artikel in dit nummer van Hans Otten: Red.) Gezien onze behoefte werd, zodra de mogelijkheid daar was en in weerwil van de raad die ons gegeven was, een cassette-interface bij Musicprint Computer Products aangeschaft. (Deze firma is ook bekend onder de naam CHIP. Red.). Deze interface leverde wel wat stof op. Nu het uiteindelijk werkt, zeg je dat de interface op zich in orde is. Maar om het ding echt aan de praat te krijgen hebben beginners wel een helpend handje nodig in de vorm van een programma. Daar had MCP voor gezorgd, zoals ze al eerder inzag dat de concurrentie toch van je vergt dat de JUNIOR computer beter verkocht kan worden met bedrukte/gegraveerde toetsen dan met die afgrijselijke plakcijfers en -letters en zoals je beter kunt aanprijzen met gratis programma's. Niet al die programma's zijn uitgeprobeerd, maar zeker is, dat ze niet allemaal werken. Voor wat het cassetteinterface programma betreft: er is vele malen telefonisch kontakt gezocht met MCP en als je iemand kreeg met meer gezag dan de telefoniste, dan nog blijft dat we nu nog op een beter programma wachten. Wat we hebben ontvangen bij de interface bleek goede voeding voor de prullenmand. Uiteindelijk heeft de software-deskundige van de KIM club, Sebo Woldringh, ons binnen een kwartier uit de nood geholpen met nieuwe programmatuur voor de cassette in- en uitvoer. Daar moest

KIM JUNIOR

nog wel een kleinigheid aan veranderd worden, maar dat was in twee avonden rond. Dat ging dus duidelijk sneller dan bij MCP. De programma's vindt U hierbij. Nog even wat over de interface. Als U niet in het bezit bent van dit ding: over een paar maanden komt Elektuur met een cassette interface die veel meer belooft. Mocht U de MCP cassette interface toch willen kopen of al bezitten, dan zult U bemerken dat één der IC's is afgeschrapt en in de beschrijving een "Speciaal IC volgens fabrieksspecificaties" heet. Wat er voor speciaals is aan een doodgewone 74123 flip-flop is ons niet bekend. De interface werkt met TTL-IC's waarvan bekend is, althans zo heb ik begrepen, dat deze een spanning van 2,4 Volt en hoger leest als "1" en 0,8 Volt en lager als "0". Kijkt U naar de technische gegevens, dan blijkt dat de audio-input maar liefst 1,5 Watt nodig heeft. Het afregelen van de interface kon gelukkig bij Van Sprang gebeuren die een praktische ervaring heeft met electronica. Hij beschikte over een draagbare radio/cassette combinatie. Enige modificaties hieraan en klaar is Kees. Bij mij ging het wat anders. Aangemoedigd door een artikel van Hans Otten in Radio Bulletin nr. 3 van 1980, kocht ik bij Radio Service Twenthe in Den Haag de aanbiedings-inbouw-cassettedeck type Touring 108 ad f 32,50. Het ding wordt in een doos overhandigd en je gaat blij naar huis. Nu is het te hopen dat niet iedereen het overkomt dat bij thuis-komst, zoals bij mij, blijkt dat het meest essentiële onderdeel ontbreekt: de motor. Voor mij werd het door de niet geringe benzinekosten dus géén koopje. Enfin, de voeding die Hans Otten vermeldde is rond gekomen. Hier bleek ook de noodzaak van samenwerking met anderen. Dat lag niet aan het schema, maar aan gebrek aan kennis en ervaring mijnerzijds. Je bent een leek, of niet. Ook de beide versterkers zijn gebouwd. Aangezien uiteindelijk gekozen is voor de universele luidspreker/versterker uit de Elektuur-Halfgeleidergids 1980, hangt de eindversterker van Hans Otten met z'n 0,5 Watt nu aan het wachtrek. En zo ook de 1..10 maal versterker. Waarom dat? Wel, gesteld dat de gestandaardiseerde uitgangsspanning van recorders 200 mV is, zoals wij menen, dan bereiken we dus bij 10 maal zo'n 2 Volt. En dat is te weinig voor de TTL-IC's van onze interface, zoals hiervoor vermeld! De Elektuur-unit kan zo'n 200 maal versterken, dus daar waren dan ook geen problemen mee.

Nu we stad en land niet meer hoeven af te bellen, kunnen we eindelijk in deel 2 van het JUNIOR-boek duiken om daar te zien dat er inderdaad heel wat in de monitor zit. De behoefte aan uitbreiding zal na het lezen van dit deel wel glore.

N.B.: De 2½ bit tijd op locatie 0148 blijkt soms beter te werken als je die op 24 (hexadecimaal 18) zet.

KIM JUNIOR

READ

KIM SOFTWARE LIBRARY

PAGE 01

```

0010: 0100      READ   ORG   $0100
0020:
0030: *****
0040: *
0050: *      LEES ROUTINE VOOR KANSAS CITY CASSETTE      *
0060: *      INTERFACE VOOR DE JUNIOR MICRO COMPUTER    *
0070: *      FOOTJE 21 VAN DE APPLICATIE CONNECTOR      *
0080: *      DOORVERBINDEN MET DE CASSETTE INPUT        *
0090: *      START ADRES INTIKKEN OP SAL/SAH            *
0100: *      EIND ADRES + 1 INTIKKEN OP EAL/EAH          *
0110: *
0120: *      AUTEUR: SEBO WOLDRINGH                      *
0130: *      ----- KLIEVERINK 619                      *
0140: *      AMSTERDAM ZUIDOOST                          *
0150: *
0160: *      COPYRIGHT (C) BY S. T. WOLDRINGH           *
0170: *
0180: *****
0190:
0200:      CONSTANTES
0210: -----
0220: 0100      HALF   *      $0005 (HALVE BIT TIJD = 5)
0230: 0100      HEEL   *      $000A (HELE BIT TIJD = 10)
0240: 0100      TWEEH  *      $0019 (TWEEN EN EEN HALVE BIT TIJD = 25)
0250:
0260:      PAGE ZERO LOCATIES
0270: -----
0280: 0100      INL    *      $00F8
0290: 0100      INH    *      INL    +01
0300:
0310:      PIA POORT B VAN 6532
0320: -----
0330: 0100      PBD    *      $1A82 PIA B DATA REGISTER
0340: 0100      PBDD   *      PBD    +01 PIA B DATA DIRECTION REGISTER
0350:
0360:      GEBRUIKTE TIMER LOCATIE:
0370: -----
0380: 0100      CNTD   *      $1AF7 1024 TIMER
0390:
0400:      JUNIOR MONITOR ROUTINES
0410: -----
0420: 0100      RESET  *      $1C1D
0430:
0440: 0100 AD 59 01      LDA    SAL      HAAL START ADRES LOW OP
0450: 0103 85 F8        STA    INL      EN ZET DIE IN PAGE ZERO START ADR LOW
0460: 0105 AD 5A 01      LDA    SAH      HAAL START ADRES HIGH OP
0470: 0108 85 F9        STA    INH      EN ZET DIE IN PAGE ZERO START ADR HIGH
0480: 010A A9 00        LDAIM $00      ZET DATA DIRECTION REGISTER B
0490: 010C 8D 83 1A      STA    PBDD     OF INPUT
0500: 010F 20 53 01 READA JSR   READI   LEES EEN BIT

```

KIM JUNIOR

READ

KIM SOFTWARE LIBRARY

PAGE 02

```

0510: 0112 D0 FB      BNE  READA  BLIJF WACHTEN TOT ER EEN "0" KOMT
0520: 0114 A9 05      LDAIM HALF  ZET HALVE BIT TIJD
0530: 0116 20 4B 01  READB JSR  READG  EN HAAL EEN BIT OP
0540: 0119 D0 F4      BNE  READA  BLIJF WACHTEN TOT ER WAT KOMT
0550: 011B A0 08      LDYIM $08  ZET AANTAL BITS PER BYTE
0560: 011D 48          PHA          OP DE STACK
0570: 011E A9 0A      READC LDAIM HEEL  ZET HELE BIT TIJD
0580: 0120 20 4B 01  JSR  READG  EN HAAL EEN BIT OP
0590: 0123 18          CLC          DE CARRY FLAG WORDT GEZET
0600: 0124 F0 01      BEQ  READD  ALS WE EEN "1" HEBBEN
0610: 0126 38          SEC          GELEZEN.
0620: 0127 68          READD PLA          SCHUIF DE GELEZEN
0630: 0128 2A          ROLA          BITS IN DE ACCUMULATOR
0640: 0129 48          PHA          VERLAAG LOOP COUNTER
0650: 012A 88          DEY
0660: 012B D0 F1      BNE  READC
0670: 012D 68          PLA          IN ACC STAAT NU EEN CHARACTER
0680: 012E 91 F8      STAIY INL  DAT WE STOREN OP
0690: 0130 E6 F8      INC  INL  HET AANGEGEVEN
0700: 0132 D0 02      BNE  READE  GEINDEXEERDE
0710: 0134 E6 F9      INC  INH  ADRES
0720: 0136 A5 F8      READE LDA  INL  ALS HET START
0730: 0138 CD 5B 01  CMP  EAL  ADRES GELIJK IS
0740: 013B D0 0A      BNE  READF  GEWORDEN AAN
0750: 013D A5 F9      LDA  INH  HET EIND ADRES+1
0760: 013F CD 5C 01  CMP  EAH  DAN ZIJN WE KLAAR
0770: 0142 D0 03      BNE  READF  EN GAAN WE TERUG
0780: 0144 4C 1D 1C  JMP  RESET  NAAR DE RESET ROUTINE
0790: 0147 A9 19      READF LDAIM TWEEH ZET 2 1/2 BIT TIJD
0800: 0149 D0 CB      BNE  READB
0810: 014B 8D F7 1A  READG STA  CNTD  1K TIMER TIJD
0820: 014E 2C F7 1A  READH BIT  CNTD  LAAT TIMER
0830: 0151 10 FB      BPL  READH  AFTELLEN
0840: 0153 AD 82 1A  READI LDA  PBD  LEES POORT B (PB6)
0850: 0156 29 40      ANDIM $40  ZET ONGEWENSTE BITS UIT
0860: 0158 60          RTS          TERUG NAAR HOOFD ROUTINE
0870: 0159 00          SAL  =  $00  START ADRES LOW
0880: 015A 00          SAH  =  $00  START ADRES HIGH
0890: 015B 00          EAL  =  $00  EIND ADRES LOW (EIND ADRES +1)
0900: 015C 00          EAH  =  $00  EIND ADRES HIGH

```

SYMBOL TABLE 3800 388A

CNTD	1AF7	EAH	015C	EAL	015B	HALF	0005
HEEL	000A	INH	00F9	INL	00F8	PBDD	1A83
PBD	1A82	READ	0100	READA	010F	READB	0116
READC	011E	READD	0127	READE	0136	READF	0147
READG	014B	READH	014E	READI	0153	RESET	1C1D
SAH	015A	SAL	0159	TWEEH	0019		

KIM JUNIOR

WRITE

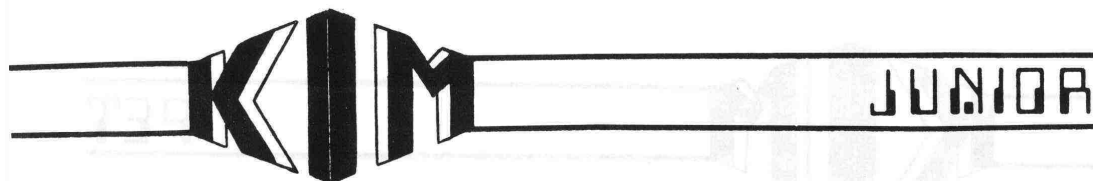
KIM SOFTWARE LIBRARY

PAGE 01

```

0010: 1A00      WRITE  ORG  $1A00
0020:
0030: *****
0040: *
0050: *      SCHRIJF ROUTINE VOOR KANSAS CITY CASSETTE
0060: *      INTERFACE VOOR DE JUNIOR MICRO COMPUTER
0070: *      POOTJE 20 VAN DE APPLICATIE CONNECTOR
0080: *      DOORVERBINDEN MET DE CASSETTE OUTPUT
0090: *      START ADRES INTIKKEN OP SAL/SAH
0100: *      EIND ADRES + 1 INTIKKEN OP EAL/EAH
0110: *
0120: *      AUTEUR: SEBO WOLDRINGH
0130: *      ----- KLIEVERINK 619
0140: *      AMSTERDAM ZUIDOOST
0150: *
0160: *      COPYRIGHT (C) BY S. T. WOLDRINGH
0170: *
0180: *****
0190:
0200:      CONSTANTES
0210: -----
0220: 1A00      HALF  *      $0005 (HALVE BIT TIJD = 5)
0230: 1A00      HEEL  *      $000A (HELE BIT TIJD = 10)
0240: 1A00      TWEH  *      $0019 (TWEEN EN EEN HALVE BIT TIJD = 25)
0250:
0260:      PAGE ZERO LOCATIES
0270: -----
0280: 1A00      INL   *      $00F8
0290: 1A00      INH   *      INL   +01
0300:
0310:      PIA POORT B VAN 6532
0320: -----
0330: 1A00      PBD   *      $1A82 PIA B DATA REGISTER
0340: 1A00      PBDD  *      PBD   +01 PIA B DATA DIRECTION REGISTER
0350:
0360:      GEBRUIKTE TIMER LOCATIE:
0370: -----
0380: 1A00      CNTD  *      $1AF7 1024 TIMER
0390:
0400:      JUNIOR MONITOR ROUTINES
0410: -----
0420: 1A00      RESET *      $1C1D
0430:
0440: 1A00 AD 56 1A      LDA  SAL      HAAL START ADRES LOW OP
0450: 1A03 85 F8          STA  INL      EN ZET DIE IN PAGE ZERO START ADR LOW
0460: 1A05 AD 57 1A      LDA  SAH      HAAL START ADRES HIGH OP
0470: 1A08 85 F9          STA  INH      EN ZET DIE IN PAGE ZERO START ADR HIGH
0480: 1A0A A7 20          LDAIM $20     ZET DATA DIRECTION REGISTER
0490: 1A0C 8D 83 1A      STA  PBDD     VOOR OUTPUT
0500: 1A0F A0 00      WRITEA LDYIM $00  ZET INDEX REGISTER Y OP NUL

```



WRITE

KIM SOFTWARE LIBRARY

PAGE 02

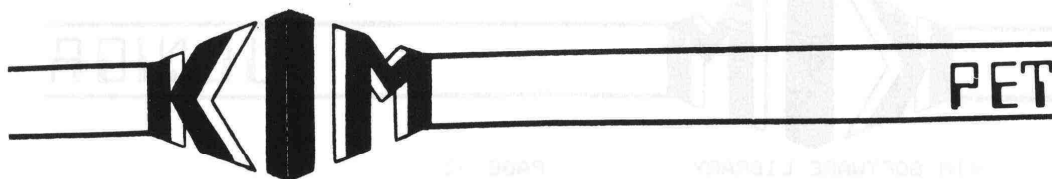
```

0510: 1A11 B1 F8          LDAIY INL    HAAL EEN BYTE OP
0520: 1A13 20 2D 1A      JSR  WRITEC EN SCHRYF HEM WEG
0530: 1A16 E6 F8          INC  INL    VERHOOG
0540: 1A18 D0 02          BNE  WRITEB START ADRES
0550: 1A1A E6 F9          INC  INH    MET 1
0560: 1A1C A5 F8          WRITEB LDA  INL    ALS HET START
0570: 1A1E CD 58 1A      CMP  EAL    ADRES GELIJK IS
0580: 1A21 D0 EC          BNE  WRITEA GEWORDEN AAN
0590: 1A23 A5 F9          LDA  INH    HET EIND ADRES +1
0600: 1A25 CD 59 1A      CMP  EAH    DAN ZIJN WE KLAAR
0610: 1A28 D0 E5          BNE  WRITEA EN GAAN WE TERUG
0620: 1A2A 4C 1D 1C      JMP  RESET  NAAR DE RESET ROUTINE
0630: 1A2D 18            WRITEC CLC    STOP DE TE SCHRYVEN
0640: 1A2E 48            PHA    BYTE OP DE STACK
0650: 1A2F 20 43 1A      JSR  WRITEE SCHRYF START BIT WEG
0660: 1A32 68            PLA    HAAL TE SCHRYVEN BYTE OP
0670: 1A33 A0 08          LDYIM $08  ZET AANTAL BITS PER BYTE
0680: 1A35 0A            WRITED ASLA  SCHUIF EEN BIT IN DE CARRY FLAG
0690: 1A36 48            PHA    ZET HET RESTANT OP DE STACK
0700: 1A37 20 43 1A      JSR  WRITEE SCHRYF EEN BIT WEG
0710: 1A3A 68            PLA    HAAL RESTANT TERUG VAN DE STACK
0720: 1A3B 88            DEY    HERHAAL DIT TOTDAT WE 8 BITS
0730: 1A3C D0 F7          BNE  WRITED HEBBEN WEGGESCHREVEN
0740: 1A3E 38            SEC    SCHRYF TWEE
0750: 1A3F 20 43 1A      JSR  WRITEE STOP BITS
0760: 1A42 38            SEC    WEG
0770: 1A43 A2 00          WRITEE LDXIM $00  SCHRYF EEN
0780: 1A45 90 01          BCC  WRITEF BIT NAAR
0790: 1A47 CA            DEX    PIA POORT B
0800: 1A48 8E 82 1A      WRITEF STX  PBD    BIT 5 (PB5)
0810: 1A4B A9 0A          LDAIM HEEL  ZET HELE BIT TIJD
0820: 1A4D 8D F7 1A      STA  CNTD  VOOR TIMER
0830: 1A50 2C F7 1A      WRITEG BIT  CNTD  EN LAAT TIMER
0840: 1A53 10 FB          BPL  WRITEG AFLOPEN
0850: 1A55 60            RTS    GA TERUG NAAR HOOFD ROUTINE
0860: 1A56 00          SAL  =  $00  START ADRES LOW
0870: 1A57 00          SAH  =  $00  START ADRES HIGH
0880: 1A58 00          EAL  =  $00  EIND ADRES LOW (EIND ADRES + 1)
0890: 1A59 00          EAH  =  $00  EIND ADRES HIGH

```

SYMBOL TABLE 3800 387E

CNTD	1AF7	EAH	1A59	EAL	1A58	HALF	0005
HEEL	000A	INH	00F9	INL	00F8	PBDD	1A83
PBD	1A82	RESET	1C1D	SAH	1A57	SAL	1A56
TWEEH	0019	WRITE	1A00	WRITEA	1A0F	WRITEB	1A1C
WRITEC	1A2D	WRITED	1A35	WRITEE	1A43	WRITEF	1A48
WRITEG	1A50						



INITIALISATOR

R. Uphoff

Dat is de naam die ik heb gegeven aan een machinetaalprogramma van 192 bytes lang. Precies, op de byte af, passende in de buffer van de niet gebruikte tweede cassette recorder. Het programma staat bijna net zoveel in rom, als we netjes de in de vorige aflevering besproken voorzieningen hebben aangebracht. Het programma gaat twee onhebbelijkheden van CBM/PET vermoorden:

1. Het ergelijke zoeken naar een file op cassette. We rusten de PET uit met "FAST FORWARD"
- 2 Een klein BASIC programma in gebruik! Zonde van de lege resterende RAM capaciteit. We zullen over meerdere programma's tegelijk in het RAM kunnen gaan beschikken. In mijn voorbeeld twee. Op dit thema mag U zelf eindeloos verder gaan hobbyen.

Allereerst is het nodig de cassette recorder geschikt te maken voor ons doel. De schakeling van fig 1 past gemakkelijk op een stukje print of montaprint van 2x4 cm.

De lijn "sense" die de CBM vertelt of een recordertoets is ingedrukt wordt onderbroken en via de dubbele transistor-inverter weer aangesloten. In plaats van te worden geaard door de recorder switch, gebeurt dat nu door het met deze schakelaar afknijpen van de eerste tor, zodat de tweede opent. Het effect is dus voor de CBM gelijk, maar we kunnen nu zonder de schakelaar te openen de sense-line hoog maken, waardoor de motor stopt, door pen c van de USERPORT door de CBM zelf nul te laten maken.

Op pen D van de USERPORT kan de CBM nu geheel onafhankelijk zien of er een recordertoets is ingedrukt.

In de +6v-leiding van de motor is een weerstand van 2,2 ohm opgenomen. Als de band is teruggespoeld en dus stil staat zal de motor, die nu het slijpende mechanisme moet aandrijven, veel meer stroom gaan opnemen zodat de BC-557 opent. Een eenvoudig filter stelt de gevoeligheid in en filtert de "troep", veroorzaakt door de draaiende motor uit. Ook voorziet het in enige vertraging, zodat niet elke stroompiek een output naar pen E van de USERPORT stuurt, op welk punt de CBM kan lezen of een bandspoeloperatie is afgelopen.

DE CBM USERPORT

De te gebruiken USER PORT bestaat uit een 8-bits databus welke is aangesloten op de data A uitgang van een 6522 VIA (Output Register A). Het genoemde ORA wordt bij een reset van de CBM "hardware" geïnitieerd, dus door het laag worden van de reset van het VIA. Het ORA heeft adres \$E84F en bevat dan \$FF. Het is rechtstreeks geschikt voor input "active low", zolang het eveneens door de reset op 0 geïnitieerde DATA DIRECTION REGISTER A (DDRA) in deze toestand blijft. Het DDRA heeft in de CBM \$E843 als adres. Door een bit van het DDRA 1 te maken zal het overeenkomstig bit van het ORA nul worden en geschikt zijn voor output (active high). Eenmaal op 1 gezet zal dat bit dan toch weer als active-low input kunnen dienen, zonder dat men iets aan de inhoud van DDRA behoeft te doen.

In het navolgende zullen we de recordermotor stop zetten door bit 0 van het DDRA op 1 te zetten, zodat bit 0 van het ORA 0 wordt. We kunnen door bit 0 van het DDRA weer nul te maken de motor weer vrij geven.

Laat U straks niet bedriegen door het feit dat dat bit ook bij gesloten DDRA nul blijkt te zijn. De oorzaak kunt U in het schema van fig 1 zien, waar pen C laag wordt gehouden door de basis-emissor overgang van de laatste BC 547. Het bit heeft dus "een input" die echter van geen betekenis is voor ons doel.

INPUTS (bit 0 is dus altijd 0) ADRES \$E84F

FE geen toets ingedrukt op recorder.

FA wel toets ingedrukt op recorder.

F8 wel toets ingedrukt op recorder én einde band bereikt.

OUTPUTS adres \$E843 01 motor stroomloos . 00 motor kan lopen.

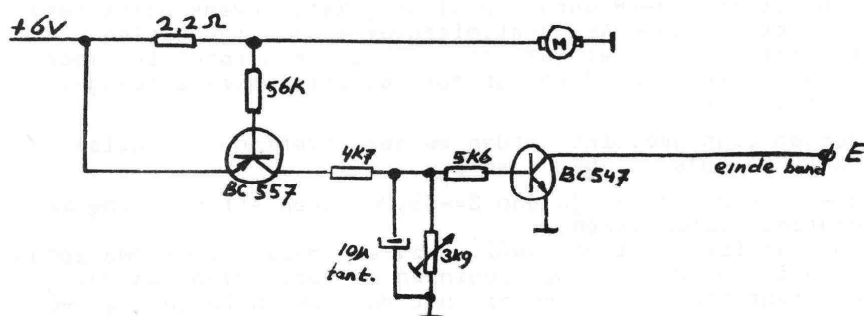
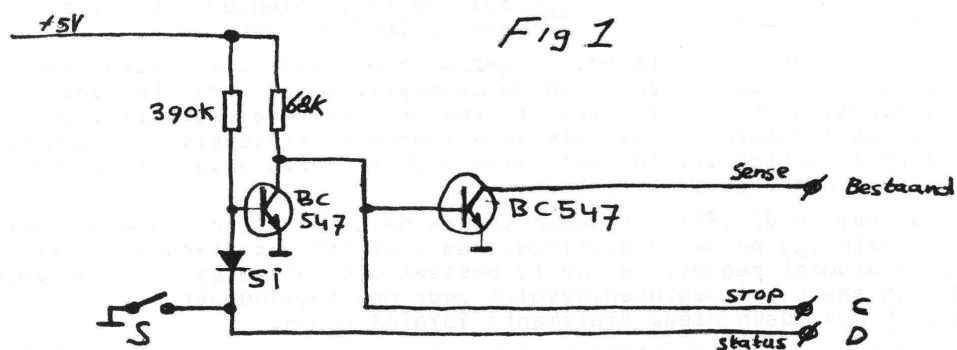


Fig 1



Een machinetaalprogramma in de 2e cassette-buffer zal straks de software gaan verzorgen. We zijn echter nog niet aan dat programma toe omdat het een "link" heeft met een tweede programma in dezelfde buffer dat ons de mogelijkheid geeft ons geheugen in twee delen van ca 4K te verdelen en dan over twee basic-programma's tegelijk te beschikken. Gelukkige bezitters van 16 of 32K machines kunnen natuurlijk nog veel verder gaan. Dan wordt het echter verplicht EPROM-programmeren want de buffer wordt dan te klein.

KIM PET

Wat we gaan doen is het operatingsystem beduvelen. Dat maakt gebruik van een 24-tal locaties op de zeropage: (PET:andere adressen!) Alle adressen zijn hexadecimaal, de dataopslag is in de bekende lo-hi volgorde, zoals in elke abs instructie van de 6502.

28 29 Startadres basictext.
2a 2b Start variabelen opslag.
2c 2d Start Array variabelen.
2e 2f Eerste vrije RAM locatie
30 31 Eerste niet meer vrije RAM locatie, gebruikt voor STRING opslag
32 33 Eerste locatie niet meer gebruikt voor string-opslag
34 35 Eerste adres dat buiten het beschikbare RAM valt.

Bij een reset initialiseert de CBM als volgt:

Geheugenlocatie 0400 is het eerste RAM adres en daar komt 00 te staan als indicatie dat daar een programma gaat beginnen. Over deze 00 straks nog een leuke tip.

Nu wordt aangenomen dat BASIC aan het begin van het RAM gaat beginnen dus de pointer 28-29 wordt op 01 04 gezet. Tevens wordt hetzelfde gedaan wat het commando NEW doet, hierover zodadelijk uitleg.

Voor een 8K machine loopt het RAM tot 1FFF dus de eerste niet meer bestaande RAM-locatie is 2000 en dus zet het initialiserings-pgm. 00 20 in locatie 34 en 35.

Zodra variabelen zijn gebruikt worden de desbetreffende locaties gezet. Nu de COMMANDO's NEW en CLR

New maakt 30-31 en 32-33 gelijk aan 34-35, dus geen stringopslag meer aan het operating system bekend.

New zet 00 op locaties 0401 en 0402. Dwz: een basic programma zonder inhoud dat eindigt op 0402 en dus beginnen de variabelen van dat oneigenlijke programma op 0403 en er komt dan ook op 2a en 2b, 03-04 te staan

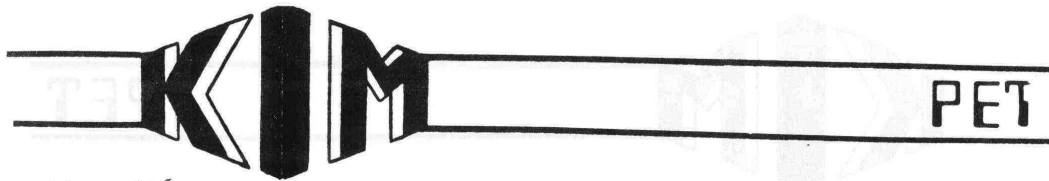
New doet verder hetzelfde als clr: wis de variabelen, dus zet alle pointers t/m 2e-2f ook op "start variabelen" dus ook 03-04.

Ik heb het bovenstaande technisch gezien niet exact beschreven maar het geeft U het juiste beeld van de pointers. Verder sprak ik over 24 bytes. Welnu 36 t/m 3F worden tijdens de programma executie voor verschillende dataopslag gebruikt. Hun inhoud interesseert ons nu niet. We moeten ze echter wel in tact laten bij de manipulaties die we ne met deze 24 bytes gaan verrichten.

Nu iets over de 00 vóór het begin van de basictext. Een statement wordt door 00 beïndigd en het operatingsystem weet dan dat daarachter het nieuwe statement begint. Dat begin bestaat uit het adres waar het volgende statement zal beginnen, gevolgt door het regelnummer. Zonder 00 dus: Geen nieuw statement: ?SYNTAX ERROR.

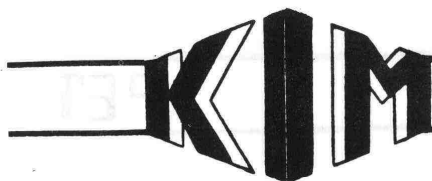
U kunt hiervan heerlijk misbruik maken door als U even weg moet POKE 1024,255 te typen. Runnen van het programma is dan onmogelijk en NEW vernietigt Uw programma niet. Met POKE1024,0 is alles weer OK.

Door allerlei manipulaties met de besproken pointers kunt U het operatingsystem helemaal naar Uw hand zetten. Alle mogelijkheden hier beschrijven is ondoenlijk. We zullen ons dus beperken tot de tweedeling van het RAM. Pagina \$11 is een mooi begin voor het tweede 4K geheugen en dus zetten we daar eerst 00



Adres \$1100 moet nu ook het einde van de eerste 4K zijn en dus moet in 34/35 dat adres gezet worden. Door nu "new" te typen hebben we ons 8K geheugen tot 4K gemaakt. Voor de tweede mogelijkheid, de laatste 4K, hebben we alle pointers weer te veranderen en dat is natuurlijk steeds een heidens karwei! Daarom creëren we een tweede 24 byte pointerset, waartoe we de laatste 24 bytes \$1fe8 - 1fff van het RAM afpikken. Ten slotte hebben we dan een machinetaalprogramma nodig dat beide pointer-sets kan omwisselen, al naar gelang we het eerste of het tweede programma willen gebruiken. Met deze gegevens moet U nu het hierna afgedrukte programma kunnen volgen. Enig commentaar zal daarbij wellicht gewenst zijn.

- \$033a Een subroutine die wacht op de juiste input op de USER-PORT "geen toets ingedrukt". Door de subroutine op \$033c aan te roepen kunnen de overige inputs van de USER PORT worden opgewacht.
- \$0342 Als bij de programmastart nog een recordertoets is ingedrukt zal het programma naar het nog te bespreken NMI programma gaan en alleen vragen de stoptoets in te drukken. Het programma moet dan opnieuw worden gestart. Vanuit basic is in direct mode met een USR-commando naar het startadres \$0342 gesprongen. Het gehele pgm is dus een subroutine van basic. De parameter van USR bevat het aantal 1/60 seconden dat we de band willen laten spoelen, in floatingpoint notatie, opgeslagen in de FAC op adres 00e5-0064. 00e5 bevat daarbij de exponent.
- \$0349 Indien geen recordertoets was ingedrukt neemt de routine aan dat de spoeloperatie mag beginnen en drukt de prompt ← af op het scherm. Dan wordt gewacht op het indrukken van de toets REW of FWD op de recorder. (besproken subroutine)
- \$0353 Als de toetsdruk is uitgevoerd wordt ter bevestiging de prompt * afgedrukt.
- \$0358 De waarde in de floatingaccumulator wordt naar een twee bytes integer op \$61 en 62 geconverteerd. (hi-lo) Vervolgens wordt de ingebouwde klok uit de CBM afgelezen. dit is de starttijd van het programma. We slaan hem op in de bytes \$63 en 64 van de FAC, gewoon omdat dat een even bruikbare zeropage locatie is. De mantissa's in de FAC zijn toch verder van geen belang meer. Vervolgens wordt de twee bytes-integer (spoeltijd) van \$61-62 gehaald en met de starttijd vermeerderd. Het resultaat komt in X en Y register, omdat dat het minste bytes kost. Dit resultaat is de stoptijd!
- \$0369 Als we niet een bepaalde tijd voorwaarts wilden spoelen, maar wilden weten hoelang het terugspoelen duurt, is het programma door ons aangeroepen met PRINT USR(0). Nu moet dus bekeken worden of de "spoeltijd" nul is en dat kan nog steeds, ondanks alle manipulaties met de FAC, door naar de exponent op \$5e te kijken. Nemen we echter eerst aan dat daar geen 0 staat. Het wachten is nu tot de CBM klok gelijk staat aan de waarde in X en Y register. Als dat zover is is de juiste tijd voorwaarts gespoeld en moet de band stil gezet gaan worden. Aan gezien dat ook moet gebeuren naar een REWIND operatie zullen we eerst het geval USR(0) gaan bezien. Daarna gaat het pgm in alle gevallen weer dezelfde weg.
- \$036d



PET

\$0377 USR was 0. Nu wordt gewacht op het einde van de rewind-operatie, waarbij zich een klein probleem voor doet: De supersnelle processor is al met dit programmadeel bezig als de aanloopstroom van de cassetemotor nog hoog is. Dit zou nu als "einde band" worden opgevat en daarom wordt ca $\frac{1}{2}$ sec. gewacht. Pas daarna (weer de subroutine) wordt op ontvangst van het einde-band signaal gewacht.

\$384 Of het einde van de FFWD-spoeltijd of eindeband nu de reden is doet er niet toe. De motor moet worden gestopt. Eerst wordt hier echter berekend hoe lang de spoel-operatie heeft geduurd. Bij REW is dat het DOEL van het pgm. Bij FFWD een leuke controle. Het resultaat van de berekening komt in Y en Accumulator en een subroutine uit de basicinterpreter plaatst het resultaat netjes in floatingpoint in de FAC.

\$390 Gevraagd wordt de stop-toets in te drukken van de recorder en:

\$0395 De motor wordt stil gezet. De enige subroutine laat weer wachten op het indrukken van de stop-toets (=geen toets gedrukt)

\$039d De motor wordt dan weer vrijgegeven en een RTS keert terug naar basic waar de inhoud van de FAC, de spoeltijd wordt afgedrukt.

Om USR te kunnen gebruiken staat in het operatingsysteem op adres \$ 0000 een constante 4C JMP. Daaraachter moet de gebruiker zelf het juiste adres invullen. Uit pure luiheid laten we dat echter een routine doen op \$03a3, door SYS 931 te typen. Dat hoeft maar één keer, na het laden van het programma te gebeuren en natuurlijk na gebruik van uw zelf aangebrachte resetschakelaar. De routine verandert ook de NMI-vector op \$0094 en 0095, teneinde NMI via ons eigen interrupt-programma naar het normale interrupt-programma op \$C389 te sturen. PET-bezitters moeten dit stukje pgm helaas weglaten omdat hun machine niet over een NMI-RAM-vector beschikt. Het NMI-pgm op 03b2 doet niets anders dan eerst even kijken of er geen cassette I/O operatie bezig is, door het lsb van de IRQ-vector te controleren. Zoja wordt NMI haastig met een RTI de laan uit gestuurd. Voorts controleert de routine, die we immers veel zullen gebruiken op ons FFWD-programma te onderbreken, of we netjes de stop-toets van de recorder hebben ingedrukt. De motor wordt ook terloops even gestopt.

Bij elk NMI-gebruik wordt ons tenslotte nog even verteld welk start-adres ons RAM-programma heeft. Waarover meer in dat deel van het tweede programma in de buffer. We zullen nu eerst de gebruiksaanwijzing behandelen, van de FFWD faciliteit.

Als U een programma naar cassette gaat schrijven bepaald u en niets of niemand anders voortaan wáár op de cassette dat programma komt. Udoet dat als volgt:

PRINTUSR(3740):SAVE"NAAM"

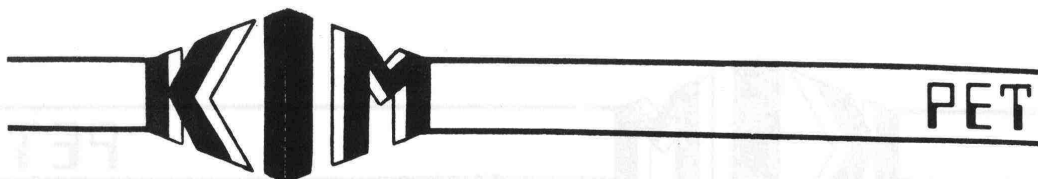
* 3740

PRESS PLAY & RECORD ON TAPE 1

OK

WRITING NAAM

READY.



Om het programma later terug te vinden typed U:

PRINTUSR(3740):LOAD

de naam mag u gerust weg laten want Uw CBM vindt daar aléén het goede programma. Er blijft nu nog een probleem: Waar mag U het volgende programma wegschrijven!

Spoel daartoe na een SAVE de recorder niet terug maar:

PRINTUSR(0)

* 1731

READY.

Zoek nu in de grafiek van fig 2 de factor V op en vraag:

?1731/V

Noteer het antwoord als plaats voor de volgende file.

U kunt voor elk type cassette het snijpunt M vinden door de totale speeltijd van de cassette te meten. De grafiek is wat onnauwkeurig omdat allerlei invloeden zoals langzamer lopende motor aan het einde van REW niet zijn verwerkt. Deze fouten moet men laten voor wat ze zijn want ze vormen toevallig nét de gewenste veiligheidsmarge.

Het tweede deel van het programma geeft U de mogelijkheid met SYS959 van RAM te verwisselen: onderste of bovenste 4K. Het programma moet na wat ik over de pointers verteld heb voor zich spreken. Vergeet niet na de eerste SYS959 het commando NEW te typen. Een volgende keer zal ik desgewinst nog een aantal mogelijkheden met deze pointers onder de loop nemen, die te maken hebben met programma OVERLAY en machinetaalroutines in het RAM naast het bijbehorend basic.

Voorts is nog een SOFTWARE blok golf generator in de maak met zeer grote frequentienauwkeurigheid.

PROGRAMMA

```
033a a9 fe      lda imm    srt cont. controleer status recorder
033c cd 4f e8    cmp abs     wacht op input user port
033f d0 fb      bne rel     bne 033c
0341 60
```

```
0342 a9 fe      lda imm     cont.stoptoets
0344 cd 4f e8    cmp abs
0347 d0 70      bne rel     bne 03b9 breek pgm af.
0349 a9 5f      lda imm     print prompt ←
034b 20 32 f2    jsr abs     jsr wrt (basic int.)
034e a9 fa      lda imm     cont. indrukken toets
0350 20 3c 03    jsr abs     jsr cont.
0353 a9 2a      lda imm     print prompt *
0355 20 32 f2    jsr abs     jsr wrt (basic int.)
0358 20 9a d0    jsr abs     jsr flp-int (basic int.)
035b a5 8f      lda zpg     lsb clock
035d 85 64      sta zpg     sta lsb clock starttijd
035f 65 62      adc zpg     lsb stoptijd
0361 a8         tay imp     sta in Y lsb stoptyd
0362 a5 8e      lda zpg     lda msb als boven voor msb
0364 85 63      sta zpg
0366 65 61      adc zpg
0368 aa         tax imp     msb sta in X
0369 a5 5e      lda zpg     cont op 0 in FAC
```

```

036b f0 0a      beq rel  indien 0 rew,zoniet ffwd.
036d e4 8e      cpx zpg  Wacht op einde spoeltyd.
036f d0 fc      bne rel  bne 036d
0371 c4 8f      cpy zpg
0373 d0 fc      bne rel  bne 0371
0375 f0 0d      beq rel  beq 0384

0377 a5 64      lda zpg  wacht op einde bandsignaal
0379 69 20      adc imm  wacht op einde danloopstroom motor
037b c5 8f      cmp zpg
037d d0 fc      bne rel  bne 037b
037f a9 f8      lda zpg
0381 20 3c 03   jsr abs  jsr cont. wacht op einde bandsignaal.

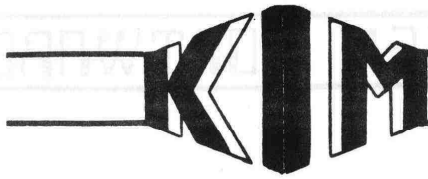
0384 a5 8f      lda zpg  bereken looptijd band
0386 e5 64      sbc zpg
0388 a8         tay imp
0389 a5 8e      lda zpg
038b e5 63      sbc zpg
038d 20 6d d2   jsr abs  jsr int-flp (basic int.)
0390 a9 a6      lda imm  print prompt
0392 20 32 f2   jsr abs  jsr wrt (basic int.)
0395 a9 01      lda imm  Blokkeer motor (stop)
0397 8d 43 e8   sta abs
039a 20 3a 03   jsr abs  jsr cont. Wacht op stopptoets
039d a9 00      lda imm  geef motor vrij
039f 8d 43 e8   sta abs
03a2 60         RTS imp  Terug naar basic/of einde srt.

03a3 a9 42      lda imm  initialiseer NMI en USR (SYS 931)
03a5 85 01      sta zpg  adres usr
03a7 a9 03      lda imm
03a9 85 02      sta zpg  page usr (vector 0001/0002 geset)
03ab 85 95      sta zpg  page nmi
03ad a9 b2      lda imm
03af 85 94      sta zpg  adres nmi ( vector 0094/0095 geset)
03b1 60         rts imp  terug naar basic

03b2 a9 2e      lda imm  "masked" nmi programma
03b4 c5 90      cmp zpg  compare irq-vector: leesoperatie gaande ?
03b6 f0 01      beq rel  beq 0309
03b8 40         RTI imp  leesoperatie: "ignore" NMI
03b9 20 90 03   jsr abs  Breek lopend spoelprogramma/ingedrukte toets-af!
03bc 4c f2 03   JMP abs  print geheugen(RAM) status

03bf ad 00 11   lda abs  memory controll program.: Controleer status.
03c2 f0 1b      beq rel  beq 03df Initialiseer niet onnodig.(fout)
03c4 a9 00      lda imm  initialiseer geheugendeling
03c6 8d 00 11   sta abs  00 pointer voor tweede RAM helft
03c9 a9 01      lda imm  creëer tweede pointer serie,
03cb 8d e8 1f   sta abs  en:
03ce a9 11      lda imm  wijzig datapointers.
03d0 85 35      sta zpg
03d2 8d e9 1f   sta abs
03d5 a9 e8      lda imm
03d7 8d f4 17   sta abs
03da a9 1f      lda imm
03dc 8d f5 1f   sta abs

```

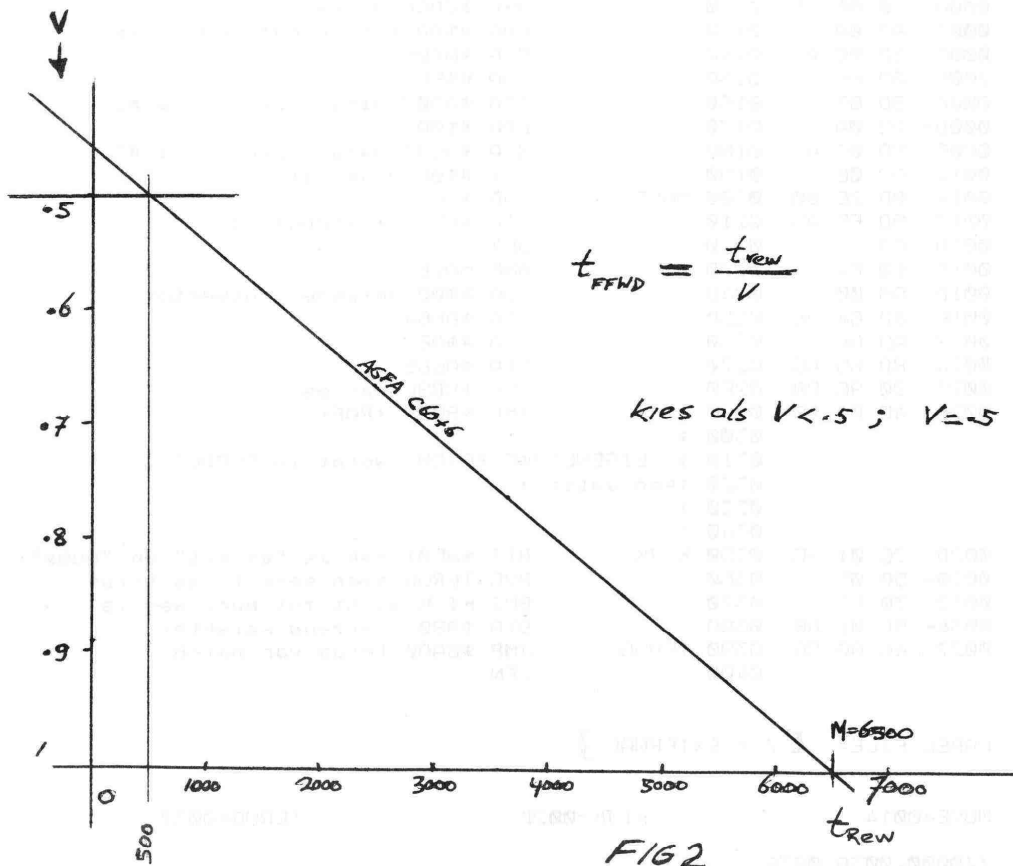


PET

```

03df a2 00 ldx imm wissel geheugen gebruik (SYS 959)
03e1 b5 28 lda zpx
03e3 48 pha imp
03e4 bd e8 lf lda abx
03e7 95 28 sta zpx
03e9 68 pla imp
03ea 9d e8 lf sta abx
03ed e8 inx imp
03ee e0 19 cpx imm
03f0 d0 ef bne rel bne 03e1
03f2 a5 29 lda zpg print geheugen keuze(status)
03f4 20 75 e7 jsr abs jsr wrob (basic int.) "type a byte"
03f7 4c 89 c3 jmp abs Warme start basic.

```



KIM SYSTEEM SOFTWARE

```

0010 ;**** RAE PRINTER BOOTSTRAP ****
0020 ;
0030 ;Laden via normale SYM L2 routine,
0040 ;vervolgens G 0 indrukken, waarna men
0050 ;direkt in de koude start van de RAE
0060 ;uitkomt.
0080 ;
0090 ;
0100 .BA 0000
0110 .OS
0000- 20 86 8B 0120 JSR $B886 acces
0003- A9 0A 0130 LDA #$0A set in/out registers
0005- 8D 0C AB 0140 STA $A80C
0008- A9 FF 0150 LDA #$FF
000A- 8D 03 AB 0160 STA $A803 datadir.reg. via #2
000D- A9 0A 0170 LDA #$0A
000F- 8D 03 AC 0180 STA $AC03 datadir.reg. via #3
0012- A2 0E 0190 LDX #$0E move patch
0014- BD 2C 00 0200 MOVE LDA $2C,X
0017- 9D FF A5 0210 STA $A5FF,X scbbuf -1
001A- CA 0220 DEX
001B- D0 F7 0230 BNE MOVE
001D- A9 00 0240 LDA #$00 verander outvector
001F- 8D 64 AE 0250 STA $A664
0022- A9 AE 0260 LDA #$AE
0024- 8D 65 AE 0270 STA $A665
0027- 20 9C 8B 0280 JSR $B89C naccs
002A- 4C 00 B0 0290 JMP $B000 *RAE*
0300 ;
0310 ; EIGENLIJKE PATCH (Wordt in SCBBUF
0320 ;geplaatst.)
0330 ;
0340 ;
002D- 2C 01 AC 0350 KIJK BIT $AC01 bekijk "select" en "busy"
0030- 50 05 0360 BVC TERUG geen select, ga terug
0032- 30 F9 0370 BMI KIJK wacht tot busy weg is
0034- 8D 01 A8 0380 STA $A801 verzend karakter
0037- 4C A0 8A 0390 TERUG JMP $8A00 terug van patch
0400 .EN

```

LABEL FILE: { / = EXTERNAL }

MOVE=0014

KIJK=002D

TERUG=0037

//0000, 003A, 003A

KIM AMUSEMENT

KIM SCHAAKPROGRAMMA

- 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 deze 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 04 17
CPY	X'3B'	"	00 C4	CO 3B

Deze wijzigingen moeten worden aangebracht in deel 1 van het programma. (Dus vóórdat deel twee 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. Op de volgende pagina's vindt U een openingen-routine en 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.)
4. Op de daaropvolgende pagina een stukje programma waarmee het schaakprogramma snel kan worden geïnitialiseerd. Door dit stukje programma in te lezen, nadat een partij is gespeeld, kan snel met een nieuwe partij worden begonnen. (Ontwikkeld door Joop Tervooren)

KIM AMUSEMENT

5. Op de pagina daaropvolgend een routine die het schaakprogramma versnelt. Ook hier is meer geheugenruimte voor nodig. Deze routine is nog niet geheel uitgetest. Wie probeert dit verder uit ?
6. Op- en aan-merkingen graag aan Theo Kortekaas tel. 02977-21888

DATA /Adressen velden KIM- schaakprogramma T.K.

8	4e	4f	50	51	52	53	54	55	Veld- aanduiding schaakbord.
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	

	A	B	C	D	E	F	G	H	
1	84	83	85	86	82	85	83	84	stukken-code geplaatst op schaakbord::
2	80	80	80	80	80	80	80	80	

00 = leeg veld:::

7	81	C1	C1	C1	C1	C1	C1	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.

KIM AMUSEMENT

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 aangeroepen, willekeurig en/of kunnen worden aangewezen door de speler, middels een aanwijstabel die men zelf kan vullen.

Tab1 = voor in code bepalen "afvelden" le vrije pagina in geheugen.

Tab2 = " " " " "toevelden" 2e pagina=na vorige

Tab3 = aanwijstabel voor de zetvolgorde 3e pagina= na vorige(opeenvolgend)

zettencode = 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.

Patches op 1798 = JSR SRØPEN

20 31 o5

PTR	DC	05 30	00(KIM speelt met zwart, dan 01)	
SRØPEN	LDX PTR	31	AE 30 05	
Ø2	LDA ØTAB1(X)	34	BD 00 07	
	STA HULP	37	85 14	
	AND X'BF'	39	29 BF	
	CMP ZET3	3B	C5 02	
	BNE Ø4	3D	DO 2B	
	LDA ØTAB2(X)	3F	BD 00 08	
	CMP ZET2	42	C5 01	
	BNE Ø4	44	DO 24	
	LDA ØTAB3(X)	46	BD 00 09	
	BEQ Ø6	49	FO 29	
	TAX	4B	AA	
Ø3	LDA ØTAB1(X)	4C	BD 00 07	
	STA HULP	4F	85 14	
	BIT HULP	51	24 14	
	BVS +5	53	70 05	
	CMP TIMER	55	CD 04 17	of ROL TIMER 2E 04 17
	BMI Ø5	58	30 17	BCS Ø5 BO 17
	AND X'BF'	5A	29 BF	
	STA ZET3	5C	85 02	
	LDA ØTAB2(X)	5E	BD 00 08	
	STA ZET2	61	85 01	
	LDA ØTAB3(X)	63	BD 00 09	
	STA PTR	66	8D 30 05	
	RTS	69	60	
Ø4	BIT HULP	6A	24 14	
	BVS Ø6	6C	70 06	
	INX	6E	E8	
	BNE Ø2	6F	DO C3	
Ø5	INX	71	E8	
	BNE Ø3	72	DO D8	
Ø6	LDX X'00'	74	A2 00	
	STX PTR	76	8E 30 05	
	JMP CALC	79	4C 30 03	(oorspronkelijk CALC op 0200)
		05 7C		

KIM AMUSEMENT

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

øtab1 0700)
øtab2 0800) Dit alles kan overal elders in het geheugen,
øtab3 0900) mits aanpassingen voor de plaatsbepaling.

zetten adres TAB1 TAB2 TAB3

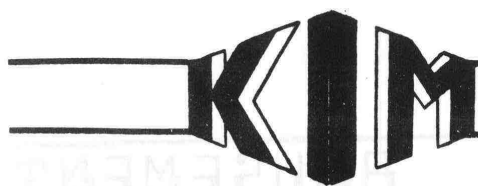
	00	00	85	01
e2-e4	1	0C	1C	05
d2-d4	2	0B	1B	60
c2-c4	3	0A	1A	98
b2-b4	4	49	19	D1
e7-e5	5	34	24	08
c7-c5	6	32	22	3C
e7-e6	7	74	2C	4e
o1-f3	8	06	15	0A
d2-d4	9	4B	1B	0B
b8-c6	A	79	2A	0C
e5xd4	B	64	1B	0E
f1-c4	C	05	1A	0F
f1-b5	D	45	21	10
d1-d4	E	43	1B	11
f8-c5	F	7D	22	12
a7-a6	10	70	28	13
b8-c6	11	79	2A	14
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 label / opening

8	30	31	3A	3B	3C	3D	3E	3F
7	30	31	32	33	34	35	36	37
6	20	21	2A	2B	2C	2D	2E	2F
5	20	21	22	23	24	25	26	27
4	10	11	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



AMUSEMENT

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
c5xd4	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
cl-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-c7	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
12-14	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
ol-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
ol-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
ol-f3	72	46	15	73
c7-c5	73	72	22	74
e1-ol	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
ol-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	00
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-ol	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
ol-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-ol	A5	44	06	A6
f6xd5	A6	6D	23	A7

KIM AMUSEMENT

zetten adres TAB1 TAB2 TAB3

c4xd5	A7	5A	23	A8
c6-d4	A8	6A	1B	A9
f3xd4	A9	55	1B	AA
c5xd4	AA	62	1B	AB
e2-e3	AB	4C	1C	AC
d4-b6	AC	5B	29	AD
ol-f3	AD	46	15	AE
o8-f6	AE	7E	2D	AF
d2-d4	AF	4B	1B	BO
c5xd4	BO	62	1B	B1
f3xd4	B1	55	1B	B2
e7-e6	B2	74	2C	B3
b1-c3	B3	41	12	B4
f8-b4	B4	7D	19	B5
d4-b5	B5	5B	21	B6
e8-o8	B6	7C	3E	B7
cl-f4	B7	42	1D	B8
b8-c6	B8	79	2A	B9
f4-d6	B9	5D	2B	BA
b4xd6	BA	59	2B	BB
b5xd6	BB	61	2B	BC
f6-e8	BC	6D	3C	BD
a2-a3	BD	48	10	BE
b7-b6	BE	71	29	OD
ol-f3	BF	46	15	CO
e5-e4	CO	64	1C	C1
f3-d4	C1	55	1B	C2
b8-c6	C2	79	2A	C3
e2-e3	C3	4C	14	C4
c6xd4	C4	6A	1B	C5
e3xd4	C5	54	1B	C6
d8-f6	C6	7B	2D	C7
d4-d5	C7	5B	23	C8
f8-c5	C8	7D	22	C9
dl-e2	C9	43	OC	CA
f6-e7	CA	6D	34	CB
b1-c3	CB	41	12	CC
o8-f6	CC	7E	2D	CD
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
ol-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

cl-b2	DA	42	09	DB
e8-o8	DB	7C	3E	DC
f1-e2	DC	45	OC	DD
c8-o4	DD	7A	1E	DE
e1-o1	DE	44	06	DF
c7-c6	DF	72	2A	EO
a2-a4	EO	48	18	E1
b8-d7	E1	79	33	OO
cl-b2	E2	42	09	E3
f7-f6	E3	75	2D	E4
e2-e4	E4	4C	1C	E5
f8xb4	E5	7D	19	E6
f1-c4	E6	45	1A	E7
o8-e7	E7	7E	34	E8
f2-f4	E8	4D	1D	E9
d7-d5	E9	73	23	EA
e4xd5	EA	5C	23	EB
b4-d6	EB	59	2B	EC
f4xe5	EC	5D	24	ED
f6xe5	ED	6D	24	EE
dl-15	EE	43	27	EF
e7-o6	EF	74	2E	FO
ol-f3	FO	46	15	F1
b8-d7	F1	79	33	F2
e1-o1	F2	44	06	F3
e8-o8	F3	7C	3E	OO
cl-b2	F4	42	09	F5
e7-e6	F5	74	2C	F6
b4-b5	F6	59	21	F7
a7-a6	F7	70	28	F8
a2-a4	F8	48	18	F9
a6xb5	F9	68	21	FA
a4xb5	FA	58	21	FB
a8xa1	FB	78	00	FC
b2xa1	FC	49	00	FD
d7-d5	FD	73	23	FE
ol-f3	FE	46	15	FF
f8-e7	FF	7D	34	OO

KIM AMUSEMENT

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	d1xd4	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	cl-o5	7C	ol-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	cl-d2	44	cl-o5	56	h2-h4	80	d1xd4	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-cl	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-ol	2B	c2-c3	2C	f1-c4	48	d1-f3	5A	f2-f4	84	f1-b5	92	e1-ol
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	cl-d3	4A	e1-cl	5C	ol-f3	86	e1-ol	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	cl-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	ol-f3	BF	ol-f3	D4	ol-f3	E2	cl-b2	F4	cl-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	ol-f3	AF	d2-d4	Cl	f3-d4	D6	e2-e3	E4	e2-e4	F6	b4-b5
6D	f1-b4	9E	b8-c6	B0	c5xd4	C2	b8-c6	D7	g7-g6	E5	f8xb4	F7	a7-a6
6E	e2-e3	9F	g2-g3	B1	f3xd4	C3	e2-e3	D8	c2-c4	E6	f1-c4	F8	a2-a4
6F	e8-o8	A0	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	cl-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	ol-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-ol	A5	e1-ol	B7	cl-f4	C9	d1-e2	DE	e1-ol	EC	f4xe5	FE	ol-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	ol-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-ol		
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.

KIM AMUSEMENT

Programma-deel, schaakprogramma TK, op de band zetten om snel een nieuwe partij te kunnen gaan spelen.

	INITIALISERING	"WIT"	KIM	"ZWART"
INI1	JMP INI2	0000	4C 26 00	
INI5	LDY X'00'	03	A0 00	
INI6	LDX X'1F'	05	A2 1F	
	STY BORD+16(X)	07	94 26	
	DEX	09	CA	
	BPL INI6	0A	10 FB	
	STY ROCCO	0C	84 0A	
	STY WRDE	0E	84 0B	
	STX WIS	10	86 90	STY DSP1 0010 84 F9
	JMP BRK	12	4C 80 17	JMP CODE 12 4C D4 17
		15	FF	
	(deel tab.BORD(	16	84 83 85 86 82 85 83 84	
		1D	80 80 80 80 80 80 80 80	
INI2	LDA X'EA'	26	A9 EA	LDA X'00' 26 A9 00
	STA INI1	28	85 00	
	LDA X'85'	2A	A9 85	
	STA INI1+1	2C	85 01	
	NOP	2E	EA	
	NOP	2F	EA	
	NOP	30	EA	
	LDX X'05'	31	A2 05	
INI4	LDA STOP(X)	33	B5 3E	
	STA VECT(X)	35	9D FA 17	
	DEX	38	CA	
	BPL INI4	39	10 F8	
	JMP INI5	3B	4C 03 00	
	(deel tab.BORD (	3E	00 00 00 00 00 00 00 00	
		46	C1 C1 C1 C1 C1 C1 C1 C1	
		4E	C4 C3 C5 C6 C2 C5 C3 C4	
		56		
PTR	SRØPEN	???? 00	(v66r "GO")	???? 01

OPZETTEN PROBLEMEN ???:

Zonodig "Rokade" uitschakelen, adres 000A FF

KIM- 1e zet laten doen, daarna probleem opzetten, zie veld adressen, tab.BORD., mogelijk maken zwart de eerste zet te doen, zodat KIM kan antwoorden voor Wit.

Indien nodig "MAX1" en "MAX2", verhogen;

LDX X'02' 1790 A2 02 (02 wordt 03)

LDX X'03' 1794 A2 03 (03 " 04)

KIM rekent daardoor meerdere zetten diep!!!
 Attentie = nog niet volledig uitgetest!!

Met dank aan de Theo Kortekaas, die dit schaakprogramma heeft ontwikkeld.

Leiden, 20 september 1980.

Jo Tervoorren.

KIM AMUSEMENT

Aanpassingen KIM-schaakprogramma T.Kortekaas.

```

*      Routine "Quick", kan overal in het geheugen
*      worden geplaatst.
*
Q1  LDA  X'C0'          A9  C0
    STA  ZWRD          85  08
    LDX  NIVO          A6  0C
    LDA  X'C8'          A9  C8
    STA  TABZWRD(X)    9D  XX  XX
    RTS               60
Q2  CMP  X'41'          C9  41
    BEQ  JMPB144        F0  19
    BIT  CZA           24  0D
    BVS  RET           70  12
    LDX  NIVO          A6  0C
    STA  TABZWRD(X)    9D  XX  XX
    CMP  X'BF'          C9  BF
    BEQ  RET           F0  09
    DEX               CA
    CLC               18
    ADC  TABZWRD(X)    7D  XX  XX
    BPL  JMPB144      10  03
RET  JMP  0272          4C  72  02
JMP14 4 JMP  B144      4C  CE  03
TABZWRD DC  X'C8'      C8
        DC  X'C8'      C8
        DC  X'C8'      C8
        DC  X'C8'      C8
        DC  X'C8'      C8
        DC  X'C8'      C8
        DC  X'C8'      C8
        DC  X'C8'      C8
        DC  X'C8'      C8
        DC  X'C8'      C8
*
*      Voor XX XX moet het adres van TABZWRD
*      worden ingevuld.
*
*      Verdere aanpassingen in het programma:
*
JSR  Q1                0202  20  Q1  Q1
NOP                      0205  EA
*
JMP  Q2                026E  4C  Q2  Q2
NOP                      0271  EA
*
*      Bij:Q1 Q1 moet het adres van Q1 worden ingevuld.
*      Bij Q2 Q2 moet het adres van Q2 worden ingevuld.
*

```

Deze routine moet bij voorkeur gebruikt worden
in samenhang met openingen-routine.

5 september 1980. T. Kortekaas,

KIM SYSTEEM SOFTWARE

Apple tape's maken met de KIM
F. Mesander en
H.J.C.Otten

Van Frank Mesander uit Leiden kregen we een programma in handen waarmee het mogelijk is om data volgens het Apple formaat te maken , op tape . De software is vrijwel letterlijk gelijk aan die van de Apple monitor , maar dan aangepast aan de KIM omgeving. Voor een andere 6502 machine is deze routine heel makkelijk aan te passen door de lokatie's \$1742 en \$1743 (respectievelijk een PIA en het bijbehorende data direction register) aan de PIA van hun machine aan te passen.

Startadres = \$00F1 (lsb) en \$00F2 (msb)

Eindadres = \$00F3 (lsb) en \$00F4 (msb)

Het signaal naar de cassette recorder is op bit 7 van PIA 1742 te vinden , de gewone audio output van de KIM .

Met dit programma is het mogelijk software van de KIM via een cassette naar een Apple over te brengen . Het omgekeerde , van de Apple naar de KIM zou ook handig zijn . Wie schrijft dit voor de club ? Andere programma's voor communicatie tussen 6502 computers zijn in de KIM club met zijn vele computer typen zeer welkom .

APPLE MICRO-WARE ASSEMBLER 65XX-1.0 PAGE 01

00100: 1000	APPLE	ORG	\$1000
00200: 1000	ADB	*	\$1742
00300: 1000	ABDD	*	\$1743
00400: 1000	OUTPUT	*	\$0330
00500: 1000	CHECKSUM	*	\$00F0
00600: 1000	AOB	*	\$00F1
00700: 1000	AOH	*	\$00F2
00800: 1000	ATL	*	\$00F3
00900: 1000	ATH	*	\$00F4
01000: 1000	TAPOUT	*	\$1742
01100: 1000	TAPEIN	*	\$1742
01200: 1000	LASTIN	*	\$00F5

KIM SYSTEM SOFTWARE

APPLE MICRO-WARE ASSEMBLER 65XX-1.0 PAGE 02

```

0130: 1000 A5 F1      NXTAO LDA    AOL
0140: 1002 C5 F3              CMP    ATL
0150: 1004 A5 F2              LDA    AOH
0160: 1006 E5 F4              SBC    ATH
0170: 1008 E6 F1              INC    AOL
0180: 100A D0 02              BNE    RTSFB
0190: 100C E6 F2              INC    AOH
0200: 100E 60              RTSFB RTS
0210: 100F A0 4B      HEADR LDYIM $4B
0220: 1011 20 21 10      JSR    ZERDLY
0230: 1014 D0 F9              BNE    HEADR
0240: 1016 69 FE      ADCIM $FE
0250: 1018 B0 F5              BCS    HEADR
0260: 101A A0 21      LDYIM $21
0270: 101C 20 21 10      WRBIT JSR    ZERDLY
0280: 101F C0              INY
0290: 1020 C8              INY
0300: 1021 88              ZERDLY DEY
0310: 1022 D0 FD              BNE    ZERDLY
0320: 1024 90 05              BCC    WRTAPE
0330: 1026 A0 02      LDYIM $32
0340: 1028 88              ONEDLY DEY
0350: 1029 00 FD              BNC    ONEDLY
0360: 102B 49      WRTAPE PHA
0370: 102C AD 42 17      LDA    PBD
0380: 102F 49 80              EORIM $80
0390: 1031 8D 42 17      STA    PBD
0400: 1034 68              PLA
0410: 1035 A0 2C      LDYIM $2C
0420: 1037 CA              DEX
0430: 1038 60              RTS
0440: 1039 A2 08      RDBYTE LDXIM $08
0450: 103B 49      RDBYTT PHA
0460: 103C 20 47 10      JSR    RDTBIT
0470: 103F 68              PLA
0480: 1040 2A              ROLA
0490: 1041 A0 3A      LDYIM $3A
0500: 1043 CA              DEX
0510: 1044 D0 F5              BNE    RDBYTT
0520: 1046 60              RTS
0530: 1047 20 4A 10      RDTBIT JSR    RDBIT
0540: 104A 89      RDBIT DEY
0550: 104B AD 42 17      LDA    TAPEIN
0560: 104E 45 F5              EOR    LASTIN
0570: 1050 10 F8              BPL    RDBIT
0580: 1052 45 F5              EOR    LASTIN
0590: 1054 85 F5              STA    LASTIN
0600: 1056 C0 80              CPYIM $80
0610: 1058 60              RTS

```

KIM SYSTEM SOFTWARE

APPLE MICRO-WARE ASSEMBLER 65XX-1.0 PAGE 03

```

0520: 1059 D8      WRITE CLD
0530: 105A A9 BF      LDAIM #BF
0540: 105C 8D 43 17    STA PSDD
0550: 105F A9 27      LDAIM #27
0560: 1061 8D 42 17    STA PSD
0570: 1064 A9 40      LDAIM #40
0580: 1066 20 0F 10    JSR HEADR
0590: 1069 A0 27      LDYIM #27
0700: 106B A2 00      WRO LDXIM #00
0710: 106D 41 F1      EORIX AOL
0720: 106F 48      PHA
0730: 1070 A1 F1      LDAIX AOL
0740: 1072 20 04 10    JSR WRBYTE
0750: 1075 20 00 10    JSR NXTAD
0760: 1078 A0 1D      LDYIM #1D
0770: 107A 60      PLA
0780: 107B 90 EE      BCC WRO
0790: 107D A0 22      LDYIM #22
0800: 107F 20 04 10    JSR WRBYTE
0810: 1082 F0 47      BEQ BELL
0820: 1084 A2 10      WRYTTE LDXIM #10
0830: 1086 00      WRYTIT ASLA
0840: 1087 10 1D 10    JSR WRBIT
0850: 1089 00 FA      BNE WRYTIT
0860: 108C 60      RTS
0870: 108D 00      READ CLD
0880: 108E 20 47 10    JSR RDBIT
0890: 1091 A0 16      LDAIM #16
0900: 1093 20 0F 10    JSR HEADR
0910: 1096 05 F0      STA CHKSUM
0920: 1098 20 47 10    JSR RDBIT
0930: 1099 A0 24      RDTWO LDYIM #24
0940: 109D 20 4A 10    JSR RDBIT
0950: 10A0 00 F9      BCS RDTWO
0960: 10A2 20 4A 10    JSR RDBIT
0970: 10A5 A0 3B      LDYIM #3B
0980: 10A7 20 39 10    RDTHR JSR RDBYTE
0990: 10A9 01 F1      STPIX AOL
1000: 10AC 45 F0      EOR CHKSUM
1010: 10AE 05 F0      STA CHKSUM
1020: 10B0 20 00 10    JSR NXTAD
1030: 10B3 A0 35      LDYIM #35
1040: 10B5 00 F0      BCC RDTHR
1050: 10B7 20 39 10    JSR RDBYTE
1060: 10BA 05 F0      CMP CHKSUM
1070: 10BC F0 0D      BEQ BELL
1080: 10BE A9 45      PRERR LDAIM #45
1090: 10C0 20 30 03    JSR OUTPUT
1100: 10C3 A9 52      LDAIM #52
1110: 10C5 20 30 03    JSR OUTPUT
1120: 10C8 20 30 03    JSR OUTPUT
1130: 10CB 60      BELL RTS

```

KIM PROGRAMMEERTALEN

Patches op MICROSOFT BASIC

H.J.C. Otten

Een gemis van Microsoft BASIC ten opzichte van Tiny BASIC (KIM versie) is het ontbreken van berekende regelnummers. Een berekende sprong is vaak erg handig en efficient , bovendien kan een subroutine met een naam worden aangeroepen wat de duidelijkheid bevordert. Die naam is dan een (integer) variabele die we van te voren een waarde geven.

Een voorbeeld :

```
10 TELOP = 1000
20 A=1 : B=2
30 GOSUB TELOP
40 PRINT C
  ⋮
1000 REM SUBROUTINE TELOP , C= A+B
1010 C = A + B
1020 RETURN
```

Het gebruiken van getallen en niet van expressies heeft het voordeel dat het snel is. De patch moet de snelheid van een GOTO etc niet aantasten liefst. De volgende patch levert het gewenste resultaat :

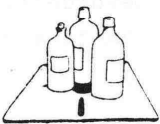
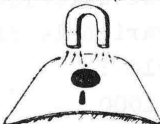
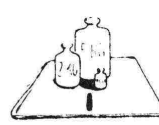
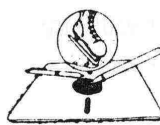
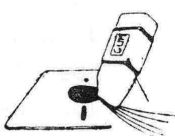
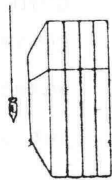
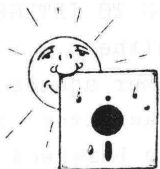
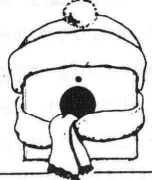
patch 27F2 in 4C 00 04 (of een andere lokatie !)

```
zet op 0400 :  B0 06  BCS REKEN  EERSTE KARAKTER EEN GETAL?
                20 B828 JSR VERZAMEL REGELNUMMER IN (11,12)
                4C F527 JMP verder met routine (normaal)
REKEN  20 342C JSR GET ADDRESS in FPAccu
                20 EF35 JSR CONVERT ADDRESS TO INTEGER IN (11,12)
                4C F527 JMP verder met routine
```

Met deze patch zal BASIC naar een regelnummer achter GOTO etc springen op de oude manier, is het eerste karakter echter geen getal (na GOTO etc) dan wordt de expressie berekend en naar het berekende regelnummer gesprongen.

KIM HARDWARE

AANWIJZINGEN VOOR GEBRUIK EN OPSLAG VAN FLOPPY DISK

1. Raak het oppervlak van de floppy niet met vingers aan. Gevaar op beschadiging, waardoor fouten ontstaan. 	2. Gebruik geen oplosmiddelen zoals alcohol of verduunners om de floppy te reinigen. 	3. Gebruik geen magnetische of statisch geladen voorwerpen vlakbij de floppy. Gegevens op de floppy kunnen verloren gaan, als de floppy aan een magnetisch veld wordt blootgesteld. 	4. De floppy niet buigen of vouwen. 
5. Plaats geen zware voorwerpen boven op de floppy. 	6. Schuif geen klemmen of paper-clips aan de floppy. 	7. Beschrijf het etiket op de floppy nooit met een potlood of ballpoint. Gebruik een viltstift. 	8. Gebruik geen gum. 
9. Plak het identificatie-etiket op de juiste plaats. Plak nooit etiketten in lagen over elkaar heen. 	10. Floppy aan bovenrand vastpakken en voorzichtig in het aandrijfmecanisme stoppen. 	11. Stop de floppy na gebruik altijd onmiddellijk terug in de bijbehorende enveloppe. 	12. Bewaar de floppy's rechtop in de bijbehorende doos. 
13. Stel de floppy nooit bloot aan overmatige warmte, verhitting of zonlicht. 	14. Omgevingstemperaturen voor gebruik: 10° C tot 50° C en vochtigheidsgraad: 20° RH tot 80° RH. 	15. Omgevingstemperaturen voor opslag: 4° C tot 53° C en vochtigheidsgraad: 8° RH tot 80° RH. 	16. Tijdens transport dient de floppy in de bijbehorende enveloppe en een beschermende doos te zijn opgeborgen. Omgevingstemp. bij vervoer: -40° C tot +53° C en vochtigheidsgraad: 8° RH tot 90° RH. 