Q-Chess adaptations by Fer Weber.

As published in the Dutch KIM Club Magazine, the KIM Kenner, Issue 17, August 1987

This document contains:

- The scans of the original article August 1987.
- The listing of the adaptations made by Fer Weber, with improvements made after the publication, checked against the binary of the Q-Chess program as distributed by Fer Weber in 1982.

Keys replace the KIM keyboard as follows:

AD -> S

DA -> R

PC -> P

+ -> Q

0..8 -> 0..8

9 -> G

0 -> H

Load at $2000, cold start at $2380

Warm start at $2500.

Note that when you let the computer do a Move, it can take long (minutes) before the computer responds!

So wait! And Wait!

Q-CHESS: van TVT-6 naar TTY-video

FER WEBER GEBR.WIENERSTR. 139 5913 XS VENLO

Sinds enige tijd is er voor de KIM-1 een bijzonder aardig schaakprogramma beschikbaar (1), geschreven door ene S.Henning. Q-CHESS 1.0 draait op een met 8K RAM uitgebreide KIM (2000...3FFF), voorzien van de legendarische TVT-6 van Don Lancaster (2).
Bij de cassette met het programma krijg je een uitvoerige handleiding, maar deze bevat geen becommentarieerde listing... waar is de tijd gebleven van uitstekende documentatie zoals bij Microchess van Peter Jennings!
Q-CHESS biedt comfortabele mogelijkheden tot het uitvoeren van de bekende "bijzondere" schaakzetten en het opzetten van andere dan de beginstellingen. Ook bevat het een aantal alternatieven betreffende de "zoekdiepte" in het algemeen en in bijzondere situaties zoals bij slagwisselingen, pion slaat stuk en matsituaties. Het is wel zaak enigszins voorzichtig te zijn met deze alternatieven: de denktijd wil nog wel eens snel oplopen tot een soort correspondentieschaak...

Er zijn twee manieren om de resultaten te tonen, duidelijk aangegeven in de handleiding:
1. door het veranderen van één spronginstructie komen de zetten op het KIM LED-display (dus kale KIM plus 8K RAM);
2. videodisplay van bord, laatste twee zetten en nog wat extra's, via de TVT-6 hard- en software in een 20 x 32 formaat.

Beide manieren gebruiken het gewone KIM-toetsenbordje. Alles OK voor mensen met een 8K KIM en evt. een TVT-6.

KIMmers met een telex-achtig geval aan hun computer, "ijzer" of "glas", met een ASCII keyboard hebben over het algemeen wel minstens een 16 x 64 formaat tot hun beschikking. Bovendien, na de eerste 8K komt meestal snel de tweede 8K om bv BASIC te kunnen draaien. De handleiding van Q-CHESS geeft een aantal aanwijzingen om het programma om te zetten; de adressen van een aantal relevante spronginstructies en wat voor routines je daarvoor zou moeten schrijven. Als je dat doet merk je snel dat het niet zo makkelijk gaat. Als je dan met bv de APPLE disassembler (omgezet naar de KIM)(3) zo hier en daar Q-CHESS omwoelt, is er echter wel uit te komen. JUNIOR-mensen zouden het op die manier ook kunnen rooien.
Wat nu volgt zijn mijn uitbreidingen op Q-CHESS. Voor een goed begrip een overzicht van mijn systeem voor zover relevant:
KIM-1 met 4K (0400...13FF) ingebouwd in een koffertje
grote kast met bufferkaart en twee 8K RAM kaarten (eigenbouw met Vero-wire)
ASCII display module, serieel (4)
daaraan parallel een eigenbouw (losse toetsjes) ASCII toetsenbord
een video monitor.
Ik heb mijn pleisters op Q-CHESS voor het gemak (tape) op adres 4000 laten beginnen; het kan ook op 0400. De linkerhelft van het videobeeld bevat het bord, aanwijzingen en de afgegeven laatste zetten; de rechterhelft wordt gebruikt om in verticale kolommen de laatste 45 zetten te bewaren. De rechterhelft kan worden schoongemaakt met toets @. De volgende ASCII-toetsen

worden gebruikt als vervanging voor KIM-toetsen: AD wordt S, DA wordt R, PC wordt P, + wordt Q, 0...8 blijft zo, 9 wordt G, 0 wordt H.

```
2370 4C 00 40        ;changes in Q-CHESS program
2BF1 20              ;

4000 A5 00    CRT    LDAZ          save Z-page
4002 48              PHA           on stack
4003 A5 01           LDAZ
4005 48              PHA
4006 A5 02           LDAZ
4008 48              PHA
4009 A5 03           LDAZ
400B 48              PHA
400C 20 00 42        JSR   MOVEMM  roll up old moves, insert 2 new moves
400F 20 70 40        JSR   CRTDIS  send all information to TTY
4012 68              PLA           restore Z-p
4013 85 03           STAZ
4015 68              PLA
4016 85 02           STAZ
4018 68              PLA
4019 85 01           STAZ
401B 68              PLA
401C 85 00           STAZ
401E 4C 00 25        JMP   WARMQ   return to Q-CHESS

2376 4C 21 40        ;changes in Q-CHESS program
389D EA EA EA        ;
38A0 EA EA EA        ;
38A3 EA EA EA        ;
38A6 EA              ;

4021 20 5A 1E KEYBOA JSR   GETCH   KIMmonitor; this subroutine converts ASCII-
4024 38              SEC           code from keyboard to KIM-keyboard code
4025 E9 30           SBC           weed out nonused keys
4027 30 F8           BMI   KEYBOA
4029 C9 24           CMP
402B 10 F4           BPL   KEYBOA
402D C9 19           CMP
402F F0 F0           BEQ   KEYBOA
4031 C9 10           CMP           is it key  ?
4033 D0 03           BNE   KEYBCO
4035 20 D6 40        JSR   FILLSP  special key, first clear MM
4038 A8       KEYBCO TAY
4039 29 0F           AND           still weeding out
403B C9 0A           CMP
403D 10 E2           BPL   KEYBOA
403F B9 43 40        LDAAY         usable key: convert
4042 60              RTS
                     ;data
4043 00 01 02 03 04 05 06 07 08 09
4054 0A 0B 0C 0D 0E 0F 09 00
4063 14 12 11 10
```

```
4070 20 2F 1E CRTDIS JSR   CRLF   KIMmonitor
4073 A2 5D           LDX          set length TABLE; TABLE contains list of
4075 20 83 40 CRTCON JSR   SUBA   addresses where videodata is stored
4078 30 08           BMI   OUT1   again end of TABLE
407A 20 C4 40        JSR   NEWLIN executes CRLF
407D EA EA           NOPs
407F 4C 75 40        JMP   CRTCON
4082 60        OUT1  RTS

4083 20 AD 40 SUBA   JSR   SUBB   this subroutine divides each video line into
4086 A0 00           LDY          three parts of different lengths of which
4088 20 BA 40 FIRST  JSR   SUBC   the starting addresses are to be found in
408B C8              INY          the TABLE: video information is scattered
408C C0 1A           CPY          27 characters?
408E D0 F8           BNE   FIRST
4090 20 AD 40        JSR   SUBB
4093 A0 00           LDY
4095 20 BA 40 SECOND JSR   SUBC
4098 C8              INY
4099 C0 06           CPY          5 characters?
409B D0 F8           BNE   SECOND
409D 20 AD 40        JSR   SUBB
40A0 30 0A           BMI   OUT2   end of TABLE?
40A2 A0 00           LDY
40A4 20 BA 40 THIRD  JSR   SUBC
40A7 C8              INY
40A8 C0 20           CPY          32 characters?
40AA D0 F8           BNE   THIRD
40AC 60        OUT2

40AD BD 00 41 SUBB   LDAAX        get TABLE pointer low and high
40B0 85 00           STAZ         and store in Z-p
40B2 CA              DEX
40B3 BD 00 41        LDAAX
40B6 85 01           STAZ
40B8 CA              DEX          ready for next pointer
40B9 60              RTS

40BA 98        SUBC  TYA          output to TTY
40BB 48              PHA
40BC B1 00           LDAIY        get character
40BE 20 A0 1E        JSR   OUTCH  KIMmonitor
40C1 68              PLA
40C2 A8              TAY
40C3 60              RTS

40C4 8A        NEWLIN TXA          output CRLF
40C5 48              PHA
40C6 20 2F 1E        JSR   CRLF   KIMmonitor
40C9 68              PLA
40CA AA              TAX
40CB 60              RTS
```

```
                        ;TABLE
4100 03 FA 03 E0 44 E0 41 7A 41 60 44 C0 41 7A 41 60
4110 44 A0 41 FA 41 E0 44 80 41 9A 41 80 44 60 41 7A
4120 41 C0 44 40 41 7A 03 A0 44 20 03 3A 03 60 44 00
4130 03 1A 03 20 43 E0 02 7A 02 E0 43 C0 02 5A 02 A0
4140 43 A0 02 3A 02 60 43 80 02 1A 02 20 43 60 01 DA
4150 01 E0 43 40 01 BA 01 A0 43 20 01 9A 41 40
                        ;DATA
4160 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
4170 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
4180 2E 2E 20 3D 20 42 4C 41 43 4B 20 53 51 55 41 52
4190 45 3B 20 2B 20 3D 20 59 4F 55 52 53 20 20 20 20
41A0 20 20 20 20 20 20 20 20 20 20 4D 59 20 53 49 44
41B0 45 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
41C0 20 20 20 20 20 20 20 20 59 4F 55 52 20 53 49 44
41D0 45 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
41E0 53 3D 53 45 54 3B 20 52 3D 52 45 56 2E 3B 20 50
41F0 3D 50 4C 41 59 3B 20 51 3D 56 41 4C 49 44 2E 20

2380 4C D0 40           ;changes in Q-CHESS program

40D0 20 D6 40 CLEANM JSR   FILLSP this subroutine clears right half of screen
40D3 4C C9 2A         JMP          return to Q-CHESS
40D6 A5 00    FILLSP LDAZ         save Z-p on stack
40D8 48                PHA
40D9 A5 01             LDAZ
40DB 48                PHA
40DC A9 00             LDA          set up pointer MM
40DE 85 00             STAZ
40E0 A9 43             LDA
40E2 85 01             STAZ
40E4 20 F0 40          JSR    FILL  go and fill MM with spaces
40E7 68                PLA          restore Z-p
40E8 85 01             STAZ
40EA 68                PLA
40EB 85 00             STAZ
40ED A9 0A             LDA          set up nonusable key for Q-CHESS program
40EF 60                RTS

40F0 A9 20    FILL   LDA          load spaces in MM
40F2 A8                TAY
40F3 91 00    SPACES STAIY
40F5 C8                INY
40F6 D0 FB             BNE    SPACES
40F8 E6 01             INCZ
40FA 91 00    AGAIN  STAIY
40FC C8                INY
40FD D0 FB             BNE    AGAIN
40FF 60                RTS

4200 A9 61    MOVEMM LDA          this subroutine adds new moves to vertical
4202 85 00             STAZ   FROML list of previous moves; field called MM
4204 A9 21             LDA          set up pointers to first column in MM
4206 85 02             STAZ   TOL
```

```
4208 A9 43           LDA
420A 85 01           STAZ  FROMH
420C 85 03           STAZ  TOH
420E 20 76 42 MM     JSR   MOVLIN bump moves two lines up
4211 18              CLC
4212 A5 00           LDAZ  FROML
4214 69 20           ADC           point to next line
4216 85 00           STAZ  FROML
4218 90 08           BCC   TOL1    pointer high to be incremented?
421A E6 01           INCZ  FROMH
421C A9 45           LDA           end of column?
421E C5 01           CMPZ  FROMH
4220 F0 1E           BEQ   FROMH1
4222 18        TOL1  CLC
4223 A5 02           LDAZ  TOL
4225 69 20           ADC           point to next line
4227 85 02           STAZ  TOL
4229 90 E3           BCC   MM      pointer high to be incremented?
422B E6 03           INCZ  TOH
422D A9 45           LDA           end of column?
422F C5 03           CMPZ  TOH
4231 D0 DB           BNE   MM
4233 C6 03           DECZ  TOH     point to next column
4235 C6 03           DECZ  TOH
4237 18              CLC
4238 A5 02           LDAZ  TOL     point to next column too
423A 69 2B           ADC
423C 85 02           STAZ  TOL
423E D0 CE           BNE   MM      branch always
4240 A9 17     FROMH1 LDA
4242 C5 00           CMPZ  FROML   last column?
4244 F0 0D           BEQ   DONE
4246 C6 01           DECZ  FROMH   point to next column
4248 C6 01           DECZ  FROMH
424A 18              CLC
424B A5 00           LDAZ  FROML   point to next column too
424D 69 2B           ADC
424F 85 00           STAZ  FROML
4251 D0 CF           BNE   TOL1    branch always
4253 18        DONE  CLC
4254 A5 02           LDAZ  TOL     point to next line
4256 69 20           ADC
4258 85 02           STAZ  TOL
425A A0 08           LDY           get first current move
425C B9 D7 03 NEWM01 LDAAY
425F 91 02           STAIY         store in MM
4261 88              DEY
4262 10 F8           BPL   NEWM01
4264 18              CLC           point to next line
4265 A5 02           LDAZ  TOL
4267 69 20           ADC
4269 85 02           STAZ  TOL
426B A0 08           LDY           get second current move
426D B9 F7 03 NEWM02 LDAAY
```

```
4270 91 02          STAIY       store in MM
4273 10 F8          BPL   NEWM02
4275 60             RTS

4276 A0 08    MOVLIN LDY
4278 B1 00    LOOP   LDAIY
427A 91 02           STAIY
427C 88              DEY
427D 10 F9           BPL   LOOP
427F 60              RTS

                     ;field MM uses memory 4320...44FF
                     ;END OF PROGRAM
```

4271 88 DEY

Door het ontbreken van een listing heb ik geen poging gedaan in O-CHESS zelf in te grijpen. Vandaar dat de gegevens voor het display zo bij elkaar gesprokkeld worden, gewoon rechttoe rechtaan geprogrammeerd. Om het iets begrijpelijker te maken nog een overzicht van adressen die gebruikt worden voor het samenstellen van het videodisplay.

karakters	1...26	27...32	33...64	field MM:		
regel 1:	41A0	019A	4320	4321	432C	4337
regel 2:	01A0	01BA	4340	4341	434C	4357
regel 3:	01E0	01DA	4360	4361	436C	4377
regel 4:	0220	021A	4380	4381	438C	4397
regel 5:	0260	023A	43A0	43A1	43AC	43B7
regel 6:	02A0	025A	43C0	43C1	43CC	43D7
regel 7:	02E0	027A	43E0	43E1	43EC	43F7
regel 8:	0320	031A	4400	4401	440C	4417
regel 9:	0360	033A	4420	4421	442C	4437
regel 10:	03A0	417A	4440	4441	444C	4457
regel 11:	41C0	417A	4460	4461	446C	4477
regel 12:	4180	419A	4480	4481	448C	4497
regel 13:	41E0	41FA	44A0	44A1	44AC	44B7
regel 14:	4160	417A	44C0	44C1	44CC	44D7
regel 15:	4160	417A	44E0	44E1	44EC	44F7
regel 16:	03E0	03FA	wordt gebruikt voor intikken van nieuwe zet			

O ja, O-CHESS speelt een aardig stuk beter dan MICROCHESS.

Noten: (1) O-CHESS cassette + handleiding verkrijgbaar bij THE 6502 PROGRAM EXCHANGE, 2920 W.MOANA, RENO, NV 89509, USA voor $20 + 15% voor verzendkosten.
(2) POPULAR ELECTRONICS, juli en augustus 1977.
(3) o.a. Dr.Dobb's Journal, vol.1 nummer 8 en vol.2 nummer 10.
(4) ASCII Display Module van Visser Assembling Electronics te Heerhugowaard.

```
0001   0000             ; Q-CHESS TVT-6 adaptations to TTY-VIDEO
0002   0000             ;
0003   0000             ; Fer Weber KIM Kenner 17
0004   0000             ;
0005   0000             ; Hans Otten
0006   0000             ; March 2025 translated to English and converted to TASM32 syntax
0007   0000             ; patches applied to original QCHESS ORIGINAL.BIN
0008   0000             ; result QCHESSTTY.BIN checked
0009   0000             ;
0010   0000             ; Q-Chess is a chess program for the KIM-1 by  S.Henning.
0011   0000             ; Q-CHESS 1.0 runs on a KIM-1 with 8K RAM at $2000 and the legendary TVT-6 by Don Lancaster.
0012   0000             ; The program comes with a good manual, with information how to paly and how to change the input and output.
0013   0000             ;
0014   0000             ; Standard Q-Chess as described in the manual: two ways to show the results on a KIM-1 with 8K and TVT-6.
0015   0000             ; 1. KIM LED display
0016   0000             ; 2. Video display of the board and last two moves and more via the TVT–6 hard– and software in a 20 x 32 formaat.
0017   0000             ; 3. Input via the KIM keyboard.
0018   0000             ;
0019   0000             ; This source modifies standard Q-Chess for TVT-6 and KIM LED/Keyboard to an ASCII TTY version.
0020   0000             ; It implements the changes indicated in the manual for a video TTY input/output.
0021   0000             ; Left side of dislay shows the board, the right side information and last 45 moves.
0022   0000             ; Clear the right side with key @.
0023   0000             ;
0024   0000             ; Patches start at $4000, $0400 can be used also.
0025   0000             ; Keys replace the KIM keybaord as follows:
0026   0000             ; AD -> S
0027   0000             ; DA -> R
0028   0000             ; PC -> P
0029   0000             ; +  -> Q
0030   0000             ; 0..8 -> 0..8
0031   0000             ; 9 -> G
0032   0000             ; 0 -> H
0033   0000             ;
0034   0000             ;Q-Chess entry points
0035   0000             ;
0036   0000             ; Zeropage pointers
0037   0000             ;
0038   0000             FROML          =            $00
0039   0000             FROMH          =            $01
0040   0000             TOL                         =            $02
0041   0000             TOH                         =            $03
0042   0000
0043   0000             COLDST         =            $2380                              ; Cold start
0044   0000             WARMQ          =            $2500                              ; Warm start of Q-Chess
0045   0000             ;
0046   0000             ; KIM entry points
0047   0000             ;
```

```
0048   0000             OUTCH          =            $$1EA0                              ; out character to TTY
0049   0000             GETCH          =            $1E5A                               ; get character from TTY
0050   0000             CRLF           =            $1E2F                               ; CRLF
0051   0000             ;
0052   0000             ; Patches to Q-Chess to new TTY routines
0053   0000             ; As described in the Q-Chess manual chapter 10 (page 9-10).
0054   0000
0055   2105                            .ORG         $2105
0056   2105 04                         .BYTE        $04
0057   2106
0058   2111                            .ORG         $2111
0059   2111 04 04 0C 08                .BYTE        $04,$04,$0C,$08,$08,$00
0059   2115 08 00
0060   2117
0061   2370                            .ORG         $2370
0062   2370 4C 00 40                   JMP                       CRT                                ; do CRT
0063   2373
0064   2380                            .ORG         $2380
0065   2380 4C D0 40                   JMP                       CLEAMM
0066   2383
0067   2BF1                            .ORG         $2BF1
0068   2BF1 20                         .BYTE        $20
0069   2BF2
0070   2376                            .ORG         $2376                    ; change in Q-Chess
0071   2376 4C 21 40                   JMP                       KEYBOA                   ; keyboard patch
0072   2379
0073   389D                            .ORG         $389D
0074   389D EA                         NOP
0075   389E EA                         NOP
0076   389F EA                         NOP
0077   38A0 EA                         NOP
0078   38A1 EA                         NOP
0079   38A2 EA                         NOP
0080   38A3 EA                         NOP
0081   38A4 EA                         NOP
0082   38A5 EA                         NOP
0083   38A6 EA                         NOP
0084   38A7
0085   3F00                            .ORG $3F00
0086   3F00
0087   3F00 F8 00 00 00   .BYTE        $F8,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00
0087   3F04 00 00 00 00
0087   3F08 00 00 00 00
0087   3F0C 00 00 00 00
0088   3F10 00 00 00 00   .BYTE        $00,$00,$00,$00,$00,$00,$00,$FF,$20,$45,$32,$20,$2D,$20,$45,$34
0088   3F14 00 00 00 FF
0088   3F18 20 45 32 20
```

```
0088   3F1C 2D 20 45 34
0089   3F20 00 00 00 00   .BYTE        $00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00
0089   3F24 00 00 00 00
0089   3F28 00 00 00 00
0089   3F2C 00 00 00 00
0090   3F30 95 36 FA 36   .BYTE        $95,$36,$FA,$36,$F4,$6F,$F4,$1F,$00,$00,$00,$00,$00,$00,$00,$00
0090   3F34 F4 6F F4 1F
0090   3F38 00 00 00 00
0090   3F3C 00 00 00 00
0091   3F40 1F 44 46 3F   .BYTE        $1F,$44,$46,$3F,$E0,$59,$DC,$9F,$01,$01,$01,$01,$01,$01,$01,$01
0091   3F44 E0 59 DC 9F
0091   3F48 01 01 01 01
0091   3F4C 01 01 01 01
0092   3F50 01 01 01 01   .BYTE        $01,$01,$01,$01,$01,$01,$01,$01,$01,$01,$01,$01,$01,$01,$01,$01
0092   3F54 01 01 01 01
0092   3F58 01 01 01 01
0092   3F5C 01 01 01 01
0093   3F60 40 22 12 12   .BYTE        $40,$22,$12,$12,$0D,$0D,$0C,$0C,$04,$04,$04,$04,$04,$04,$04,$04
0093   3F64 0D 0D 0C 0C
0093   3F68 04 04 04 04
0093   3F6C 04 04 04 04
0094   3F70 40 22 12 12   .BYTE        $40,$22,$12,$12,$0D,$0D,$0C,$0C,$04,$04,$04,$04,$04,$04,$04,$04
0094   3F74 0D 0D 0C 0C
0094   3F78 04 04 04 04
0094   3F7C 04 04 04 04
0095   3F80 03 07 05 01   .BYTE        $03,$07,$05,$01,$00,$04,$06,$02,$9F,$9F,$9F,$9F,$9F,$9F,$9F,$9F
0095   3F84 00 04 06 02
0095   3F88 9F 9F 9F 9F
0095   3F8C 9F 9F 9F 9F
0096   3F90 09 0B 0D 0E   .BYTE        $09,$0B,$0D,$0E,$9F,$0C,$0A,$08,$9F,$9F,$9F,$9F,$9F,$9F,$9F,$9F
0096   3F94 9F 0C 0A 08
0096   3F98 9F 9F 9F 9F
0096   3F9C 9F 9F 9F 9F
0097   3FA0 9F 9F 9F 9F   .BYTE        $9F,$9F,$9F,$9F,$9F,$9F,$9F,$9F,$9F,$9F,$9F,$9F,$9F,$9F,$9F,$9F
0097   3FA4 9F 9F 9F 9F
0097   3FA8 9F 9F 9F 9F
0097   3FAC 9F 9F 9F 9F
0098   3FB0 9F 9F 9F 9F   .BYTE        $9F,$9F,$9F,$9F,$0F,$9F,$9F,$9F,$9F,$9F,$9F,$9F,$9F,$9F,$9F,$9F
0098   3FB4 0F 9F 9F 9F
0098   3FB8 9F 9F 9F 9F
0098   3FBC 9F 9F 9F 9F
0099   3FC0 9F 9F 9F 9F   .BYTE        $9F,$9F,$9F,$9F,$1F,$9F,$9F,$9F,$9F,$9F,$9F,$9F,$9F,$9F,$9F,$9F
0099   3FC4 1F 9F 9F 9F
0099   3FC8 9F 9F 9F 9F
0099   3FCC 9F 9F 9F 9F
0100   3FD0 9F 9F 9F 9F   .BYTE        $9F,$9F,$9F,$9F,$9F,$9F,$9F,$9F,$9F,$9F,$9F,$9F,$9F,$9F,$9F,$9F
0100   3FD4 9F 9F 9F 9F
```

```
0100   3FD8 9F 9F 9F 9F
0100   3FDC 9F 9F 9F 9F
0101   3FE0 19 1B 1D 1E   .BYTE       $19,$1B,$1D,$1E,$9F,$1C,$1A,$18,$9F,$9F,$9F,$9F,$9F,$9F,$9F,$9F
0101   3FE4 9F 1C 1A 18
0101   3FE8 9F 9F 9F 9F
0101   3FEC 9F 9F 9F 9F
0102   3FF0 13 17 15 11   .BYTE       $13,$17,$15,$11,$10,$14,$16,$12,$9F,$9F,$9F,$9F,$9F,$9F,$9F,$03
0102   3FF4 10 14 16 12
0102   3FF8 9F 9F 9F 9F
0102   3FFC 9F 9F 9F 03
0103   4000
0104   4000            ;
0105   4000            ; New code for TTY routines
0106   4000            ;
0107   4000
0108   4000                           .ORG $4000
0109   4000            ;
0110   4000 A5 00      CRT                       LDA         FROML                   ; save zeropage
0111   4002 48                        PHA
0112   4003 A5 01                     LDA                     FROMH
0113   4005 48                        PHA
0114   4006 A5 02                     LDA                     TOL
0115   4008 48                        PHA
0116   4009 A5 03                     LDA                     TOH
0117   400B 48                        PHA
0118   400C 20 00 42                  JSR         MOVEMM                  ; roll up old moves, insert 2 new moves
0119   400F 20 70 40                  JSR         CRTDIS                  ; send all information to TTY
0120   4012 68                        PLA                                                 ; restore zeropage
0121   4013 85 03                     STA                     TOH
0122   4015 68                        PLA                                                 ; restore zeropage
0123   4016 85 02                     STA                     TOL
0124   4018 68                        PLA                                                 ; restore zeropage
0125   4019 85 01                     STA                     FROMH
0126   401B 68                        PLA                                                 ; restore zeropage
0127   401C 85 00                     STA                     FROML
0128   401E 4C 00 25                  JMP         WARMQ                   ; return to Q-CHESS
0129   4021            ;
0130   4021            ;
0131   4021 20 5A 1E   KEYBOA         JSR         GETCH                   ; this subroutine converts ASCII-code
0132   4024 38                        SEC                                                 ; from keybaord to KIM keybaord values
0133   4025 E9 30                     SBC         #$30                            ; code from keyboard to KIM-keyboard code
0134   4027 30 F8                     BMI         KEYBOA                  ; weed out non used keys
0135   4029 C9 24                     CMP                     #$24
0136   402B 10 F4                     BPL         KEYBOA
0137   402D C9 19                     CMP         #$19
0138   402F F0 F0                     BEQ         KEYBOA
0139   4031 C9 10                     CMP                     #$10                    ; is it key ?
```

```
0140   4033 D0 03                    BNE          KEYBCO
0141   4035 20 D6 40                 JSR          FILLSP                  ; special key, first clear MM
0142   4038 A8          KEYBCO       TAY
0143   4039 29 0F                    AND          #$0F                    ; still weeding out
0144   403B C9 0A                    CMP                    #$0A
0145   403D 10 E2                    BPL          KEYBOA
0146   403F B9 43 40                 LDA                    KEYTAB,Y     ; usable key: convert
0147   4042 60                       RTS
0148   4043             ;
0149   4043             ; keyboard code convert table
0150   4043 00 01 02 03 KEYTAB       .BYTE        $00,$01,$02,$03,$04,$05,$06,$07,$08,$09
0150   4047 04 05 06 07
0150   404B 08 09
0151   404D 2E 1B E2 4B              .BYTE        $2E,$1B,$E2,$4B,$0A,$0B,$0A
0151   4051 0A 0B 0A
0152   4054 0A 0B 0C 0D              .BYTE        $0A,$0B,$0C,$0D,$0E,$0F,$09,$00
0152   4058 0E 0F 09 00
0153   405C 2B AE 8B 02              .BYTE        $2B,$AE,$8B,$02,$14,$12,$11
0153   4060 14 12 11
0154   4063 14 12 11 10              .BYTE        $14,$12,$11,$10
0155   4067 87 6B 97 6B              .BYTE        $87,$6B, $97,$6B,$5F,$9B,$57,$9B,$0F
0155   406B 5F 9B 57 9B
0155   406F 0F
0156   4070             ;
0157   4070             ; Display on TTY
0158   4070             ;
0159   4070 20 2F 1E    CRTDIS       JSR                    CRLF                    ; to be sure
0160   4073 A2 5D                    LDX                    #$5D                    ; set lenght TABLE, contains list of addresses
0161   4075 20 83 40    CRTCON       JSR          SUBA                    ; where videodata is stored
0162   4078 30 08                    BMI                    OUT1                    ; again end of table
0163   407A 20 C4 40                 JSR          NEWLIN                  ; execute CRLF
0164   407D EA                       NOP
0165   407E EA                       NOP
0166   407F 4C 75 40                 JMP          CRTCON
0167   4082 60          OUT1         RTS
0168   4083             ;
0169   4083             ; Divides each videoline into three parts of different length of which starting
0170   4083             ; addresses are to be found in the TABLE, video information is scattered
0171   4083             ;
0172   4083
0173   4083 20 AD 40    SUBA         JSR          SUBB                                    ;
0174   4086 A0 00                    LDY                    #$00
0175   4088 20 BA 40    FIRST        JSR          SUBC
0176   408B C8                       INY
0177   408C C0 1A                    CPY                    #26                                             ; 27 characters?
0178   408E D0 F8                    BNE          FIRST
0179   4090 20 AD 40                 JSR          SUBB
```

```
0180   4093 A0 00                   LDY                  #$00
0181   4095 20 BA 40    SECOND      JSR          SUBC
0182   4098 C8                      INY
0183   4099 C0 06                   CPY                  #6                                    ; 5 characters?
0184   409B D0 F8                   BNE          SECOND
0185   409D 20 AD 40                JSR          SUBB
0186   40A0 30 0A                   BMI                  OUT2                          ; end of TABLE?
0187   40A2 A0 00                   LDY          #$00
0188   40A4 20 BA 40    THIRD       JSR                  SUBC
0189   40A7 C8                      INY
0190   40A8 C0 20                   CPY                  #32                                   ; 32 characters?
0191   40AA D0 F8                   BNE                  THIRD
0192   40AC 60          OUT2        RTS
0193   40AD
0194   40AD BD 00 41    SUBB        LDA          TABLE,X
0195   40B0 85 00                   STA                  FROML
0196   40B2 CA                      DEX
0197   40B3 BD 00 41                LDA                  TABLE,X
0198   40B6 85 01                   STA          FROMH
0199   40B8 CA                      DEX
0200   40B9 60                      RTS
0201   40BA
0202   40BA 98          SUBC        TYA                                                        ; output to TTY
0203   40BB 48                      PHA
0204   40BC B1 00                   LDA                  (FROML),Y               ; Get character
0205   40BE 20 A0 1E                JSR                  OUTCH                         ; out to TTY
0206   40C1 68                      PLA
0207   40C2 A8                      TAY
0208   40C3 60                      RTS
0209   40C4             ;
0210   40C4             ; Output CRLF
0211   40C4             ;
0212   40C4 8A          NEWLIN      TXA
0213   40C5 48                      PHA
0214   40C6 20 2F 1E                JSR                  CRLF
0215   40C9 68                      PLA
0216   40CA AA                      TAX
0217   40CB 60                      RTS
0218   40CC             ;
0219   40CC             ; filler
0220   40CC             ;
0221   40CC DB AE AB 02             .BYTE        $DB,$AE,$AB,$02
0222   40D0             ;
0223   40D0             ; Clears right half of screen
0224   40D0             ;
0225   40D0 20 D6 40    CLEAMM      JSR                  FILLSP
0226   40D3 4C C9 2A                JMP                  $2AC9                         ; return to Q-chess
```

```
0227   40D6
0228   40D6 A5 00       FILLSP         LDA                    FROML                              ; save zeropage
0229   40D8 48                         PHA
0230   40D9 A5 01                      LDA                    FROMH
0231   40DB 48                         PHA
0232   40DC A9 00                      LDA                    #$00                               ; set up pointer
0233   40DE 85 00                      STA                    FROML
0234   40E0 A9 43                      LDA                    #$43
0235   40E2 85 01                      STA                    FROMH
0236   40E4 20 F0 40                   JSR          FILL                               ; go and fill MM with spaces
0237   40E7 68                         PLA                                                                    ; restore zeropage
0238   40E8 85 01                      STA                    FROMH
0239   40EA 68                         PLA
0240   40EB 85 00                      STA                    FROML
0241   40ED A9 0A                      LDA                    #$0A                               ; setup nousable key for Q-Chess
0242   40EF 60                         RTS
0243   40F0             ;
0244   40F0             ; Load spaces in MM
0245   40F0             ;
0246   40F0 A9 20       FILL           LDA                    #' '                               ; space
0247   40F2 A8                         TAY
0248   40F3 91 00       SPACES         STA                    (FROML),Y
0249   40F5 C8                         INY
0250   40F6 D0 FB                      BNE          SPACES
0251   40F8 E6 01                      INC                    $01
0252   40FA 91 00       AGAIN          STA                    (FROML),Y
0253   40FC C8                         INY
0254   40FD D0 FB                      BNE                    AGAIN
0255   40FF 60                         RTS
0256   4100             ;
0257   4100             ; TABLE
0258   4100             ;
0259   4100             TABLE
0260   4100 03 FA 03 E0                .BYTE        $03,$FA,$03,$E0,$44,$E0,$41,$7A,$41,$60,$44,$C0,$41,$7A,$41,$60
0260   4104 44 E0 41 7A
0260   4108 41 60 44 C0
0260   410C 41 7A 41 60
0261   4110 44 A0 41 FA                .BYTE        $44,$A0,$41,$FA,$41,$E0,$44,$80,$41,$9A,$41,$80,$44,$60,$41,$7A
0261   4114 41 E0 44 80
0261   4118 41 9A 41 80
0261   411C 44 60 41 7A
0262   4120 41 C0 44 40                .BYTE        $41,$C0,$44,$40,$41,$7A,$03,$A0,$44,$20,$03,$3A,$03,$60,$44,$00
0262   4124 41 7A 03 A0
0262   4128 44 20 03 3A
0262   412C 03 60 44 00
0263   4130 03 1A 03 20                .BYTE        $03,$1A,$03,$20,$43,$E0,$02,$7A,$02,$E0,$43,$C0,$02,$5A,$02,$A0
0263   4134 43 E0 02 7A
```

```
0263   4138 02 E0 43 C0
0263   413C 02 5A 02 A0
0264   4140 43 A0 02 3A               .BYTE       $43,$A0,$02,$3A,$02,$60,$43,$80,$02,$1A,$02,$20,$43,$60,$01,$DA
0264   4144 02 60 43 80
0264   4148 02 1A 02 20
0264   414C 43 60 01 DA
0265   4150 01 E0 43 40               .BYTE       $01,$E0,$43,$40,$01,$BA,$01,$A0,$43,$20,$01,$9A,$41,$A0,$88,$D0
0265   4154 01 BA 01 A0
0265   4158 43 20 01 9A
0265   415C 41 A0 88 D0
0266   4160
0267   4160            ; 4160: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
0268   4160            ; 4170: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
0269   4160            ; 4180: 2E 2E 20 3D 20 42 4C 41 43 4B 20 53 51 55 41 52  .. = BLACK SQUAR
0270   4160            ; 4190: 45 3B 20 2B 20 3D 20 59 4F 55 52 53 20 20 20 20  E; + = YOURS
0271   4160            ; 41A0: 20 20 20 20 20 20 20 20 20 4D 59 20 53 49 44            MY SID
0272   4160            ; 41B0: 45 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  E
0273   4160            ; 41C0: 20 20 20 20 20 20 20 20 59 4F 55 52 20 53 49 44         YOUR SID
0274   4160            ; 41D0: 45 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  E
0275   4160            ; 41E0: 53 3D 53 45 54 3B 20 52 3D 52 45 56 2E 3B 20 50  S=SET; R=REV.; P
0276   4160            ; 41F0: 3D 50 4C 41 59 3B 20 51 3D 56 41 4C 49 44 2E 20
0277   4160            DATA
0278   4160 20 20 20 20               .BYTE       $20,$20,$20,$20,$20,$20,$20,$20,$20,$20,$20,$20,$20,$20,$20,$20
0278   4164 20 20 20 20
0278   4168 20 20 20 20
0278   416C 20 20 20 20
0279   4170 20 20 20 20               .BYTE       $20,$20,$20,$20,$20,$20,$20,$20,$20,$20,$20,$20,$20,$20,$20,$20
0279   4174 20 20 20 20
0279   4178 20 20 20 20
0279   417C 20 20 20 20
0280   4180 2E 2E 20 3D               .BYTE       $2E,$2E,$20,$3D,$20,$42,$4C,$41,$43,$4B,$20,$53,$51,$55,$41,$52
0280   4184 20 42 4C 41
0280   4188 43 4B 20 53
0280   418C 51 55 41 52
0281   4190 45 3B 20 2B               .BYTE       $45,$3B,$20,$2B,$20,$3D,$20,$59,$4F,$55,$52,$53,$20,$20,$20,$20
0281   4194 20 3D 20 59
0281   4198 4F 55 52 53
0281   419C 20 20 20 20
0282   41A0 20 20 20 20               .BYTE       $20,$20,$20,$20,$20,$20,$20,$20,$20,$20,$4D,$59,$20,$53,$49,$44
0282   41A4 20 20 20 20
0282   41A8 20 20 4D 59
0282   41AC 20 53 49 44
0283   41B0 45 20 20 20               .BYTE       $45,$20,$20,$20,$20,$20,$20,$20,$20,$20,$20,$20,$20,$20,$20,$20
0283   41B4 20 20 20 20
0283   41B8 20 20 20 20
0283   41BC 20 20 20 20
0284   41C0 20 20 20 20               .BYTE       $20,$20,$20,$20,$20,$20,$20,$20,$59,$4F,$55,$52,$20,$53,$49,$44
```

```
0284   41C4 20 20 20 20
0284   41C8 59 4F 55 52
0284   41CC 20 53 49 44
0285   41D0 45 20 20 20              .BYTE       $45,$20,$20,$20,$20,$20,$20,$20,$20,$20,$20,$20,$20,$20,$20,$20
0285   41D4 20 20 20 20
0285   41D8 20 20 20 20
0285   41DC 20 20 20 20
0286   41E0 53 3D 53 45              .BYTE       $53,$3D,$53,$45,$54,$3B,$20,$52,$3D,$52,$45,$56,$2E,$3B,$20,$50
0286   41E4 54 3B 20 52
0286   41E8 3D 52 45 56
0286   41EC 2E 3B 20 50
0287   41F0 3D 50 4C 41              .BYTE       $3D,$50,$4C,$41,$59,$3B,$20,$51,$3D,$56,$41,$4C,$49,$44,$2E,$20
0287   41F4 59 3B 20 51
0287   41F8 3D 56 41 4C
0287   41FC 49 44 2E 20
0288   4200            ;
0289   4200            ; add new new moves to vertical list of previous moves MM
0290   4200            ;
0291   4200 A9 61      MOVEMM        LDA                   #$61                            ; setup pointers for first column in MM
0292   4202 85 00                    STA         FROML
0293   4204 A9 21                    LDA                   #$21
0294   4206 85 02                    STA                   TOL
0295   4208 A9 43                    LDA                   #$43                            ; MM buffer is at $4300
0296   420A 85 01                    STA                   FROMH
0297   420C 85 03                    STA                   TOH
0298   420E 20 76 42   MM            JSR                   MOVLIN
0299   4211 18                       CLC
0300   4212 A5 00                    LDA                   FROML
0301   4214 69 20                    ADC                   #$20                            ; point to next line
0302   4216 85 00                    STA                   FROML
0303   4218 90 08                    BCC                   TOL1                            ; pointer high to be incremented?
0304   421A E6 01                    INC                   FROMH
0305   421C A9 45                    LDA                   #$45                            ; end of column?
0306   421E C5 01                    CMP                   FROMH
0307   4220 F0 1E                    BEQ                   FROMH1
0308   4222 18         TOL1          CLC
0309   4223 A5 02                    LDA                   TOL
0310   4225 69 20                    ADC                   #$20                            ; point to next line
0311   4227 85 02                    STA                   TOL
0312   4229 90 E3                    BCC                   MM                                        ; pointer high to be incremented?
0313   422B E6 03                    INC                   TOH
0314   422D A9 45                    LDA                   #$45                            ; end of column?
0315   422F C5 03                    CMP                   TOH
0316   4231 D0 DB                    BNE         MM
0317   4233 C6 03                    DEC                   TOH                                       ; point to next column
0318   4235 C6 03                    DEC                   TOH
0319   4237 18                       CLC
```

```
0320   4238 A5 02                 LDA                 TOL                                     ; point to next column too
0321   423A 69 2B                 ADC                 #$2B
0322   423C 85 02                 STA                 TOL
0323   423E D0 CE                 BNE                 MM                                      ; branch always
0324   4240 A9 17       FROMH1    LDA                 #$17
0325   4242 C5 00                 CMP                 FROML                           ; last column?
0326   4244 F0 0D                 BEQ                 DONE
0327   4246 C6 01                 DEC                 FROMH                           ; point to next column
0328   4248 C6 01                 DEC                 FROMH
0329   424A 18                    CLC
0330   424B A5 00                 LDA                 FROML
0331   424D 69 2B                 ADC                 #$2B                            ; point to next column too
0332   424F 85 00                 STA                 FROML
0333   4251 D0 CF                 BNE                 TOL1                                    ; branch always
0334   4253 18          DONE      CLC
0335   4254 A5 02                 LDA                 TOL                                     ; point to next column too
0336   4256 69 20                 ADC                 #$20
0337   4258 85 02                 STA                 TOL
0338   425A A0 08                 LDY                 #$08                            ; get first current move
0339   425C B9 D7 03    NEWM01    LDA                 $03D7,Y
0340   425F 91 02                 STA                 (TOL),Y                         ; store in MM buffer
0341   4261 88                    DEY
0342   4262 10 F8                 BPL                 NEWM01
0343   4264 18                    CLC                                                 ; point to next line
0344   4265 A5 02                 LDA                 TOL
0345   4267 69 20                 ADC                 #$20
0346   4269 85 02                 STA                 TOL
0347   426B A0 08                 LDY                 #$08                            ; get second move
0348   426D B9 F7 03    NEWM02    LDA                 $03F7,y
0349   4270 91 02                 STA                 (TOL),Y
0350   4272 88                    DEY
0351   4273 10 F8                 BPL                 NEWM02
0352   4275 60                    RTS
0353   4276
0354   4276 A0 08       MOVLIN    LDY                 #$08
0355   4278 B1 00       LOOP      LDA                 (FROML),Y
0356   427A 91 02                 STA                 (TOL),Y
0357   427C 88                    DEY
0358   427D 10 F9                 BPL                 LOOP
0359   427F 60                    RTS
0360   4280
0361   4320             MMBUF     .ORG        $4320                           ; to $44FF
0362   4320
0363   4320             ; Overview of addresses to construct video display
0364   4320             ;
0365   4320             ; Characters 1 .. 26        27 ..  32    33 .. 64     Field MM
0366   4320             ;
```

```
0367   4320            ; Line  1:      41A0                    019A                    4320                    4321        4320        4337

0368   4320            ; Line  2:      01A0                    01BA                    4340                    4341        4340        4357

0369   4320            ; Line  3:      01E0                    01DA                    4360                    4361        4360          .

0370   4320            ; Line  4:      0220                    021A                    4380                    4381        4380     .

0371   4320            ; Line  5:      0260                    023A                    43A0                    43A1          .

0372   4320            ; Line  6:      02A0                    025A                    43C0                    43C1          +20         +20

0373   4320            ; Line  7:      02E0                    027A                    43C0                    43E1          .                       .

0374   4320            ; Line  8:      0320                    031A                    4400                    4401

0375   4320            ; Line  9:      0360                    033A                    4420                    4421

0376   4320            ; Line 10:      03A0                    417A                    4440                    4441

0377   4320            ; Line 11:      41C0                    417A                    4460                    4461

0378   4320            ; Line 12:      4180                    419A                    4480                    4481

0379   4320            ; Line 13:      41E0                    41FA                    44A0                    44A1

0380   4320            ; Line 14:      4160                    417A                    44C0                    44C1

0381   4320            ; Line 15:      4160                    417A                    44E0                    44E1        44E0        44F7

0382   4320            ; Line 16:      03E0                    03FA                    Used for entering a new move

0383   4320

0384   4320

0385   4320              .ENDtasm: Number of errors = 0
```