```
;                                          RCSL:   44-RT 983
;                                          AUTHOR: JESPER JOHANSEN
;                                          EDITED: 75.05.05
```

```
;                    RC 3600 CHECKERBOARD III
```

```
;          BINARY TAPE  : 44-RT 985
;          BINARY CARDS : 44-RT 986
;          ASCII TAPE   : 44-RT 984

; KEYWORDS: RC 3600, CPU 705, TESTPROGRAM
```

↑↑↑

```
;*****************************************************************
;
;
;
; DESCRIPTION: RC3600 CHECKERBOARD III
;
;
; REVISION HISTORY:
;
;       REV.                DATE
;
;       00                  05.01.75
;
;
; COPYRIGHT (C) A/S REGNECENTRALEN,1975
; ALL RIGHTSB RESERVED.
;*****************************************************************
```

```
↑↑↑
;                RC3600 MEMORY CHECKERBOARD III

;1.      ABSTRACT
;                CHECKERBOARD III IS A MAINTENCE PROGRAM DESIGNE
;                TO PRODUCE WORST CASE NOISE CONDITION ON THE
;                SENSE/INHIBIT WIRES. THE PROGRAM SHOULD BE RUN
;                TO INSURE PROPER OPERATION OF SENSE AMPS,INHIBI
;                DRIVERS, AND MEMORY CURRENTS.


;2.      MACHINE REQUIREMENTS
;2.1               ANY RC 3601D PROCESSOR
;                4K READ/WRITE MEMORY


;3.      SWITCH SETTINGS
;3.1             STARTING ADDRESS        =000002
;3.2             SWITCH 0(1)             =1024 READ/WRITE DISTURB
;3.3             SWITCH 15(1)            =INHIBIT HALT ON ERROR
;4.      OPERATING PROCEDURE
;4.1             LOAD THE PROGRAM VIA THE BINARY LOADER
;4.2             SET SWITCHES TO 000002
;4.3             PRESS START
;4.3.1           THE PROGRAM WILL PRINT THE HIGHEST
;                LOCATION THE PATTERN IS TO USE.
;4.4             IF THE FAILURES ARE MARGINAL, SETTING SWITCH
;                 0 MAY AID IN INDUCING A FAILURE TO OCCURE.
;4.5             WHEN SCOPING OR ADJUSTING CURRENT, SETTING
;                SWITCH 15 WILL ENABLE THE ERROR HALT. THE
;                 BELL WILL STILL BE RUNG.
;4.6             PROGRAM MODIFICATIONS
;4.6.1           C(3)=ADR          THE STARTING PATTERN ADD
;4.6.2           C(5)=INHIBIT     INHIBIT THE CHECKERBOARD
;                                 PATTERN ON CLEARED BITS.
```

```
↑↑↑
;5.     PROGRAM OUTPUT/ERROR DISCRIPTION
;5.1        AT EACH OCCURANCE OF ERROR, IF THE TELETYPE IS NOT
;           BUSY THE BELL WILL BE RUNG, IF SWITCH (15) IS ZERO
;           THE PROGRAM WILL HALT AT LOCATION"ER".
;5.2        WHEN A ERROR HALT OCCURES:
;                   C(1)=THE ERROR WORD
;                   C(2)=THE ERROR ADDRESS
;                   C(3)=ADDRESS OF PROGRAM
;5.3        SET SWITCH (15) IF SCOPING, PRESS CONTINUE.
;5.4        SYNC PULSES
;                           A "S" PULSE (A52) CHECK ONES PATTERN
;                           A "C" PULSE (A50) CHECK ZEROS PATTERN

;6.     PROGRAM DISCRIPTION
;6.1        STORE THE CHECKERBOARD PATTERN
;6.2        IF SWITCH0(1) DISTURBS THE CONTENTS OF MEMORY BY
;           REFERANCING LOCATION 0101,0202,0303,ETC. 512
;           TIMES. THIS PRODUCES 1024 READ/WRITE DISTURBS.
;6.3        CHECK THE PATTERN WORD
;6.4        COMPLEMENT AND CHECK THE WORD
;6.5        RESTORE THE WORD
;6.6        WHEN THE END OF THE PATTERN IS REACHED THE
;           PROGRAM COMPLEMENTS THE PATTERN WORD AND RE-
;           TURNS TO STEP 6.1

;7.     LIMITATIONS
;           NONE
```

↑↑↑

```
       000002  .LOC 2
00002  000164          JMP MSIZ

00003  000620  ADR:    620                ;PATTERN STARTING ADDRESS
00004  007577  FINAL:  7577               ;PATTERN FINAL ADDRESS
00005  177777  INH:     -1                ;MASK FOR INHIBITED BITS
00006  000000  PATT:    0
00007  000000  ERET:    0

       000040  .LOC 40
00040  000017  C17:     17
00041  000400  C400:    400
00042  000077  C77:     77
00043  007777  C7777:   7777
00044  000207  C207:    207
00045  000101  C101:    101
00046  060200  CNIOC:   NIOC 0
00047  060100  CNIOS:   NIOS 0
00050  070000  C070000:070000
00051  000000  MODUAL:  0
00052  000000  EDIST:   0
00053  001000  K1000:   1000
00054  177577  M201:    -201
00055  000226  CMA:     MMA2
00056  177601  PSIZE:   BEGIN-CEND
00057  000000  PLOC:    0
00060  000000  BPROG:   0
00061  000000  EPROG:   0
00062  012345  RANDOM:  12345
00063  177760  M20:    -20
00064  000020  C20:     20
00065  000432  CBEG:    BEGIN
00066  000000  RETURN:  0
00067  004126  SEND:    JSR .RAND                  ;GET A RANDOM N
00070  000062           RANDOM
00071  024004           LDA 1,FINAL
00072  030056           LDA 2,PSIZE
00073  147000           ADD 2,1
00074  101220           MOVZR 0,0
00075  122422           SUBZ 1,0,SZC
00076  000075           JMP .-1
00077  123000           ADD 1,0              ;AC0=# MODULO C(FINAL)
00100  024063          LDA 1,M20
00101  123400           AND 1,0
00102  024003           LDA 1,ADR
00103  122433          SUBZ# 1,0,SNC         ;IF # TOO SMALL USE
00104  121000           MOV 1,0              ;C(ADR) FOR STARTER
00105  040057           STA 0,PLOC
00106  034063          LDA 3,M20
00107  117000           ADD 0,3
00110  054060           STA 3,BPROG
00111  142400           SUB 2,0
00112  040061           STA 0,EPROG
```

```
↑↑↑
  00113 145000   MOVE:   MOV 2,1
  00114 030065           LDA 2,CBEG
  00115 021000           LDA 0,0,2          ;MOVE A COPY OF
  00116 041420           STA 0,20,3         ;CHECKERBOARD TO
  00117 175400           INC 3,3
  00120 151400           INC 2,2
  00121 125404           INC 1,1,SZR        ;TEST FOR LAST REG
  00122 000115           JMP MOVE+2         ;TO BE MOVED.
  00123 034057           LDA 3,PLOC
  00124 005401           JSR 1,3            ;EXIT TO PROG.
  00125 000212           JMP MMA
  00126 054160   .RAND:  STA 3,.UD03        ;GENERATE A RANDOM #
  00127 010160           ISZ .UD03
  00130 044156           STA 1,.UD01
  00131 050157           STA 2,.UD02
  00132 031400           LDA 2,0,3
  00133 021000           LDA 0,0,2
  00134 004143           JSR .UD50
  00135 034162           LDA 3,.UD20
  00136 163000           ADD 3,0
  00137 041000           STA 0,0,2
  00140 024156           LDA 1,.UD01
  00141 030157           LDA 2,.UD02
  00142 002160           JMP @.UD03
  00143 024163   .UD50:  LDA 1,.UD21
  00144 044161           STA 1,.UD10
  00145 105120           MOVZL 0,1
  00146 125120           MOVZL 1,1
  00147 014161           DSZ .UD10
  00150 000146           JMP .-2
  00151 107000           ADD 0,1
  00152 125120           MOVZL 1,1
  00153 125120           MOVZL 1,1
  00154 123000           ADD 1,0
  00155 001400           JMP 0,3
  00156 000000   .UD01:  0
  00157 000000   .UD02:  0
  00160 000000   .UD03:  0
  00161 000000   .UD10:  0
  00162 033031   .UD20:  33031
  00163 000010   .UD21:  10
```

```
↑↑↑
00164 020053  MSIZ:   LDA 0,K1000          ;SIZE THE MEMORY
00165 115000          MOV 0,3
00166 031400  MSIZ1:  LDA 2,0,3            ;SAVE C(MEM)
00167 055400          STA 3,0,3
00170 025400          LDA 1,0,3
00171 051400          STA 2,0,3            ;RESTORE MEMORY

00172 125014          MOV# 1,1,SZR
00173 124015          COM# 1,1,SNR
00174 000202          JMP MSIZ2            ;END OF MEMORY
00175 136414          SUB# 1,3,SZR         ;AC1=BAD, AC3=GOOD
00176 063077          HALT                 ;MEMORY FAILED.
00177 117000          ADD 0,3
00200 175113          MOVL# 3,3,SNC        ;INCREMENT MEMORY ADDRESS.
00201 000166          JMP MSIZ1            ;TEST FOR 32K

00202 020054  MSIZ2:  LDA 0,M201
00203 163000          ADD 3,0
00204 040004          STA 0,FINAL
00205 004367          JSR CRLF
00206 004244          JSR MESS
00207 000236          MESIZE
00210 024004          LDA 1,FINAL
00211 004263          JSR POCT

00212 063511  MMA:    SKPBZ TTO
00213 000212          JMP .-1              ;WAIT FOR TTO DONE.
00214 063535          SKPBZ 35
00215 000214          JMP .-1
00216 020055          LDA 0,CMA
00217 040001          STA 0, 1             ;SET INTERRUPT RETURN
00220 152520          SUBZL 2,2            ;THIS PROGRAM TEST FOR INTERRUPT
00221 102000  MMA1:   ADC 0,0              ;ABILITY TO CLEAR MA.
00222 040000          STA 0,0
00223 025000          LDA 1,0,2
00224 060177          NIOS CPU             ;ENABLE INTERRUPT
00225 005000          JSR 0,2              ;SET BIT INTO MA
00226 045000  MMA2:   STA 1,0,2
00227 020000          LDA 0,0
00230 112414          SUB# 0,2,SZR         ;AC0=PC STORED
00231 063077          HALT
00232 151120          MOVZL 2,2
00233 151113          MOVL# 2,2,SNC
00234 000221          JMP MMA1
00235 000067          JMP SEND
00236 046114  MESIZE: .TXT ILL
00237 052040   T
00240 051505  ES
00241 042524  TE
00242 020104  D
00243 000000  I
```

↑↑↑

```
                ;TTO NON INTERRUPT PACKAGE
                ;"MESS" PRINTS ASCII MESSAGES AS SPECIFIED BY ASSEMBLER
                ;"CHAR" PRINTS ASCII CHARACTOR, C(0)R,C(0)L MUST BE 0
                ;WILL RETURN +2 IF C(0)R=0,CORRECTS THE PARITY,33 SIMULATE
                ;"TYPE" PRINTS C(0)R. MUST HAVE PROPER PARITY. RETURN IS
                ;TO CALL+1.REPLACE THIS ROUTINE WITH INTERRUPT TYPE IF DESIRED.
                ;"CRLF" PRINTS A CARRIAGE RETURN
                ;"POCT" PRINTS C(1) IN OCTAL FOLLOWED BY A TAB
                ;"POEC" PRINTS C(1) IN DECIMAL,LEADING ZEROS SUPPRESSED,
                ;FOLLOWED BY A TAB.

00244 054546  MESS:   STA 3,MESSR             ;PRINT A TEXT MESSAGE
00245 010545          ISZ MESSR
00246 031400          LDA 2,0,3
00247 024542          LDA 1,C377        ;A B BIT MASK
00250 021000  MES.1:  LDA 0,0,2         ;C(2)=DATA WORD
00251 125112          MOVL# 1,1,SZC
00252 123701          ANDS 1,0,SKP
00253 123401          AND 1,0,SKP       ;C(0)=DATA CHARACTOR RIGHT
00254 151400          INC 2,2           ;INC TO NEXT WORD
00255 124000          COM 1,1           ;FLIP MASK
00256 004340          JSR CHAR.         ;PRINT
00257 000250          JMP MESS+4        ;ANOTHER
00260 002532          JMP @MESSR        ;LAST
00261 020525  ZOCT:   LDA 0,CH240
00262 101001          MOV 0,0,SKP
00263 020525  POCT:   LDA 0,C60
00264 030317          LDA 2,OCTAB       ;PRINT C(1) IN OCTAL
00265 000270          JMP .+3
00266 030327  PDEC:   LDA 2,DECTB       ;PRINT C(1) IN DECIMAL
00267 020517          LDA 0,CH240       ;SUPPRESS LEADING ZEROS
00270 054337          STA 3,RADRET      ;BOTH ENTRYS PRINT NUMBER
00271 040336  TNUM:   STA 0,ZSUPP       ;THEN TAB TO NEXT POSITION
00272 050273          STA 2,.+1
00273 000000  DECOCT: 0                 ;A"LDA 2,TABLE" INSTRUCTION
00274 010273          ISZ .-1
00275 034337          LDA 3,RADRET      ;SETUP "TAB" AT END
00276 020503          LDA 0,CHTAB
00277 151005          MOV 2,2,SNR       ;IF TABLE ENTRY=0
00300 000340          JMP CHAR.         ;EXIT WITH TAB
00301 034336          LDA 3,ZSUPP       ;ZEROS SUPPRESS STUF
00302 102400          SUB 0,0
00303 146512  DECOT:  SUBL# 2,1,SZC
00304 000311          JMP DECP
00305 146400          SUB 2,1           ;FORM THE DIGIT
00306 034502          LDA 3,C60
00307 101400          INC 0,0
00310 000303          JMP DECOT
00311 151235  DECP:   MOVZR# 2,2,SNR
00312 034476          LDA 3,C60
00313 054336          STA 3,ZSUPP       ;C(0)=DIGIT
00314 163000          ADD 3,0           ;MAKE ASCII
00315 004340          JSR CHAR.         ;PRINT
00316 000273          JMP DECOCT        ;GET NEXT DIGIT
```

↑↑↑

```
00317 030320  OCTAB:  LDA 2,.+1
00320 100000          100000
00321 010000          10000
00322 001000          1000
00323 000100          100
00324 000010          10
00325 000001          1
00326 000000          0

00327 030330  DECTB:  LDA 2,.+1
      000012  .RDX 10
00330 023420          10000
00331 001750          1000
00332 000144          100
00333 000012          10
00334 000001          1
00335 000000          0
      000010  .RDX 8

00336 000000  ZSUPP:  0
00337 000000  RADRET:0
00340 054442  CHAR.:  STA 3,CHRET      ;PRINT C(0) RIGHT
00341 101325          MOVZS 0,0 SNR   ;RETURN +2 IF NULL
00342 001401          JMP 1,3
00343 040440          STA 0,CHSAV
00344 176000          ADC 3,3         ;COMPUTE THE PARITY
00345 117000          ADD 0,3
00346 163404          AND 3,0,SZR
00347 000344          JMP .-3
00350 176660          SUBCR 3,3       ;COMBINE PARITY WITH CHAR
00351 020432          LDA 0,CHSAV
00352 163300          ADDS 3,0

00353 034426  CHAR1:  LDA 3,CHTAB     ;IS THIS A TAB
00354 116405          SUB 0,3,SNR
00355 000360          JMP .+3         ;YES
00356 004435          JSR TYPE        ;NO PRINT IT
00357 002423          JMP @CHRET      ;EXIT
00360 020424          LDA 0,CHORZ     ;SIMULATE A TAB
00361 034424          LDA 3,CHAR7     ;VIA 1 TO 8 SPACES
00362 117405          AND 0,3,SNR
00363 002417          JMP @CHRET
00364 020422          LDA 0,CH240
00365 004426          JSR TYPE
00366 000360          JMP .-6
```

```
↑↑↑
  00367 054420  CRLF:   STA 3,CRLFR      ;SAVE RETURN
  00370 020410          LDA 0,C215
  00371 004340          JSR CHAR.       ;PRINT CARRIAGE AND LF
  00372 020377          LDA 0,C212
  00373 004340          JSR CHAR.
  00374 102400          SUB 0,0
  00375 040407          STA 0,CHORZ     ;CLEAR HORZ POSISTION
  00376 002411          JMP @CRLFR      ;EXIT
  00377 000212  C212:   212
  00400 000215  C215:   215
  00401 000011  CHTAB:  11
  00402 000000  CHRET:  0
  00403 000000  CHSAV:  0
  00404 000000  CHORZ:  0
  00405 000007  CHAR7:  7
  00406 000240  CH240:  240
  00407 000000  CRLFR:  0
  00410 000060  C60:    60
  00411 000377  C377:   377
  00412 000000  MESSR:  0
  00413 054415  TYPE:   STA 3,TYPRET    ;TYPE THE C(0)R IF
  00414 010770          ISZ CHORZ
  00415 074477          READS 3         ;SWITCH 1(0).
  00416 175100          MOVL 3,3
  00417 175102          MOVL 3,3,SZC
  00420 002410          JMP @TYPRET     ;INHIBIT TYPE EXIT.
  00421 063511          SKPBZ TTO
  00422 000777          JMP .-1
  00423 063535          SKPBZ 35
  00424 000777          JMP .-1
  00425 061135          DOAS 0,35
  00426 061111          DOAS 0,TTO
  00427 002401          JMP @TYPRET
  00430 000000  TYPRET: 0
```

```
↑↑↑
00431 063077            HALT                ;OPERATOR ERROR FIX C(ADR)
00432 004401  BEGIN:    JSR .+1
00433 054066            STA 3,RETURN
00434 034003            LDA 3,ADR
00435 030050            LDA 2,C070000
00436 020004            LDA 0,FINAL
00437 143400            AND 2,0
00440 040052            STA 0,EDIST
00441 173400            AND 3,2
00442 050051            STA 2,MODUAL        ;THE MEMORY MODUAL
00443 030003  IPAT:     LDA 2,ADR
00444 024041            LDA 1,C400
00445 020006            LDA 0,PATT          ;PRESET PATTERN
00446 147404            AND 2,1,SZR
00447 100000  IPAT1:    COM 0,0
00450 024005            LDA 1,INH
00451 123400            AND 1,0             ;MASK INHIBITED BITS
00452 024040            LDA 1,C17
00453 060300  FILL:     NIOP 0              ;SYNC AT A74
00454 034060            LDA 3,BPROG         ;CODE TO AVOIDE OVERWRITE
00455 156436            SUBZ# 2,3,SEZ       ;OF PROGRAM
00456 000403            JMP .+3             ;PATT . PROGRAM BEGIN
00457 034061            LDA 3,EPROG
00460 172436            SUBZ# 3,2,SEZ       ;DONT SKIP IF > PROG END.
00461 041000            STA 0,0,2
00462 034042            LDA 3,C77
00463 151400            INC 2,2             ;PATTERN
00464 133414            AND# 1,2,SZR        ;SKIP EVERY 16 TIMES
00465 000766            JMP FILL
00466 157414            AND# 2,3,SZR        ;SKIP EVERY 64 TIMES
00467 000760            JMP IPAT1
00470 020004            LDA 0,FINAL         ;TEST FOR FINAL ADDRESS
00471 142432            SUBZ# 2,0,SZC       ;EVERY 64 LOC. 4K
00472 000752            JMP IPAT+1          ;FILL TIME=100MS
```

```
↑↑↑
00473 030051  DISTUR: LDA 2,MODUAL     ;DISTURB MODULE SELECT
00474 020043          LDA 0,C7777      ;DISTURB AT LOCATION
00475 024052          LDA 1,EDIST      ;0101,0202,0303,ETC.
00476 123000          ADD 1,0
00477 024045          LDA 1,C101       ;EVERY OTHER CORE IN MEMORY
00500 133000          ADD 1,2          ;IS DISTURBED AT LEAST
00501 074477          READS 3          ;1024 TIMES+INHIBIT DISTURBS,
00502 175112          MOVL# 3,3,SZC    ;BUT ONLY IF SWITCH 0
00503 142433          SUBZ# 2,0,SNC    ;IS SET TO A ONE.
00504 000406          JMP ICHECK       ;END OF DISTURB
00505 176400          SUB 3,3
00506 025000          LDA 1,0,2        ;REFFERENCE MEMORY
00507 175704          INCS 3,3,SZR
00510 000776          JMP .-2
00511 000766          JMP DISTURB+4

00512 030003  ICHECK: LDA 2,ADR        ;INITIALIZE CHECK CYCLE
00513 024041          LDA 1,C400
00514 020006          LDA 0,PATT
00515 133414          AND# 1,2,SZR
00516 100000  ICK:    COM 0,0
00517 034005          LDA 3,INH
00520 163400          AND 3,0          ;MASK INHIBITED BITS
00521 024047          LDA 1,CNIOS      ;"S" PULSE
00522 114044          COMO 0,3,SZR     ;"C" PULSE
00523 024046          LDA 1,CNIOC      ;ON 1/0 DISTURB SIGNALS
00524 044412          STA 1,CHECK
00525 024060          LDA 1,BPROG      ;CODE TO PREVENT PATTERN
00526 146436          SUBZ# 2,1,SEZ
00527 000407          JMP CHECK        ;PATT.PROGRAM BEGIN
00530 024061          LDA 1,EPROG
00531 132436          SUBZ# 1,2,SEZ
00532 000404          JMP CHECK        ;PATT>PROGRAM END
00533 024064          LDA 1,C20
00534 133000          ADD 1,2
00535 000416          JMP ECHECK
00536 000000  CHECK:  0                ;A SYNC PULSE ISSUED
00537 025000          LDA 1,0,2        ;SIGNALS RWV7,RWV1
00540 106414          SUB# 0,1,SZR
00541 004427          JSR ERR1
00542 055000          STA 3,0,2
00543 025000          LDA 1,0,2
00544 136414          SUB# 1,3,SZR
00545 004424          JSR ERR2
00546 041000          STA 0,0,2
00547 151400          INC 2,2
00550 024040          LDA 1,C17        ;COUNT 16 TIMES
00551 147414          AND# 2,1,SZR
00552 000764          JMP CHECK
```

```
↑↑↑
00553 034042  ECHECK: LDA 3,C77
00554 157414         AND# 2,3,SZR      ;CHECK FOR END OF
00555 000741         JMP ICK           ;LINE
00556 024004         LDA 1,FINAL       ;EVERY 64 TIMES
00557 146432         SUBZ# 2,1,SZC      ;CHECK FOR END OF CORE
00560 000733         JMP ICHECK+1
00561 020006         LDA 0,PATT        ;COMP THE
00562 100000         COM 0,0           ;PATTERN
00563 040006         STA 0,PATT
00564 101004         MOV 0,0,SZR
00565 000647         JMP BEGIN+2
00566 034066         LDA 3,RETURN
00567 001400         JMP 0,3
00570 101020  ERR1:  MOVZ 0,0          ;DISTURB ENTRY
00571 054007  ERR2:  STA 3,ERET        ;UNDISTURB ENTRY
00572 034044         LDA 3,C207        ;C(1)=ERROR WORD
00573 063411         SKPBN TTO         ;C(2)=ERROR ADDRESS
00574 075111         DOAS 3,TTO        ;SET SWITCH 1 TO
00575 054413         STA 3,SUSAV
00576 004413         JSR ALARM
00577 034411         LDA 3,SUSAV
00600 074477         READS  3          ;INHIBIT HALT
00601 175200         MOVR 3,3
00602 034057         LDA 3,PLOC
00603 054066         STA 3,RETURN
00604 101003         MOV 0,0,SNC
00605 063077  ER:    HALT              ;IF TTY NOT BUSY
00606 114040         COMO 0,3          ;TURN OF TTY IF
00607 002007         JMP @ERET         ;TO NOISEY.
00610 000000  SUSAV: 0
      000032         .DUSR ALM=32
00611 054414  ALARM: STA 3,.HERR
00612 034414         LDA 3,BIT4
00613 075032         DOA 3,ALM
00614 014413         DSZ .CNT
00615 000404         JMP .RTUR
00616 176420         SUBZ 3,3
00617 075032         DOA 3,ALM
00620 000402         JMP .+2
00621 034407  .RTUR: LDA 3,C1
00622 054405         STA 3,.CNT
00623 034402         LDA 3,.HERR
00624 001400         JMP 0,3
00625 000000  .HERR: 0
00626 004000  BIT4:  1B4
00627 000001  .CNT:  1
00630 000001  C1:    1
00631 000631  CEND:  .
00632 047503         .TXT !CO
00633 054520  PY
00634 044522  RI
00635 044107  GH
00636 020124  T
00637 041450  (C
00640 020051  )
00641 043504  DG
00642 026103  C,
00643 034461  19
00644 030067  70
```

```
00645 033454   ,7
00646 026061   1,
00647 032067   74
00650 046101   AL
00651 020114   L
00652 044522   RI
00653 044107   GH
00654 051524   TS
00655 051040    R
00656 051505   ES
00657 051105   ER
00660 042526   VE
00661 000104   DI
                        .END
```