

```
.PABS
.PHEX
.IDENT CONVERT
.slist
; 300 TO 2000 BAUD LOADER PROGRAM
; (C) 1981 Jay Fenton for Astrovision, Inc.
;
; This program loads audio tapes generated by the old
; version of BALLY BASIC into the New and Improved version
; of BALLY BASIC.
;
; Operatins Instructions:
;
; 1) Load the LOADER PROGRAM into memory using the :RUN
; command. When loaded the screen background
; brightness will change continuously.
; 2) Cue up the 300 baud tape and PLAY it.
; 3) Press the GO key to release the LOADER PROGRAM
; 4) Observe the program statements accumulating in
; the buffer just below the program zone. The screen
; will reveal the scratchpad during this phase.
; 5) Program loading will stop on any of the following
; conditions:
; a) :RETURN is found in an unnumbered line.
; b) RUN is found in an unnumbered line.
; c) the HALT key is pressed, in which case we
; proceed to end loading
; d) any other key in the column beneath HALT
; is pressed.
; Cases a), b), and d) cause the program to go
; back to step 2)
; If no more segments are to be loaded, press
; HALT to proceed to final loading.
; 6) After HALT is pressed the program will convert the
; statements in the buffer into the representation
; used within BALLY BASIC. This is done by moving
; the buffer to the highest possible address and
; using special I/O routines which feed characters
; from this buffer to the BALLY BASIC input routines.
; This will be seen on the screen as a race between
; the input buffer and the program storage area.
; 7) The LOADER PROGRAM will then exit to BALLY BASIC.
; The variables A-Z will be zeroed out, and the screen
; will be cleared. The program may now be executed or
; written out at 2000 baud.
;
; NOTES:
; During the readin phase, lines without line numbers,
; rubout characters, and spaces appearing before and
; immediately after line numbers are filtered out.
; It is permissible to load tapes which contain line
; numbers out of sequence or which delete previous
; lines.
```

.SLIST

```

; These equates establish addresses within BALLY BASIC
2531      INIT0=2531H      ; HALT ROUTINE ENTRY
4E98      CHKIO=4E98H     ; INPUT CHARACTER HOOK VECTOR
4E9B      OUTCH=4E9BH     ; OUTPUT CHARACTER HOOK VECTOR
4E20      TXTUNF=4E20H   ; PROGRAM STORAGE AREA POINTER
4E22      VARBGN=4E22H   ; VARIABLE AREA START ADDR
4280      BUFSTART=4280H ; 300 BAUD BUFFER START
4E55      LASTBUF=4E55H  ; LAST BYTE WHICH BUFFER CAN USE
4E92      HKVECT=4E92H   ; HOOK VECTOR AREA START
218B      HOOKER=218BH   ; INITIAL HOOK VECTOR VALUES
4F2A      TARGET=4F2AH   ; ADDR TO RELOCATE I/O ROUTINE TO
4F26      SAVDAT=4F26H   ; REMEMBERS DATA BENEATH TXTUNF
4F28      POINTER=4F28H  ; 300 BAUD INPUT BUFFER POINTER
4FEA      STACK = 4FEAH  ; TOP OF BASIC STACK ZONE
2C68      GL2=2C68H     ; RET ADDR FOR OUTCH TO SKIP
2C70      GL2A=2C70H    ; ADDR IN INPUT LINE ROUTINE TO SKIP TO
; BIT BANGER GOODIES FOLLOW:
3C00      BANGIN = 3C00H ; BIT BANGER READ PORT
3800      BANG1  = 3800H ; BIT BANG CODE TO WRITE A ONE
3C00      BANGO  = 3C00H ; BIT BANG CODE TO WRITE A ZERO
;
; PARAMETERS FOR 300 BAUD BIT DETECTOR
; THESE SPECIFY THE MINIMUM AND MAXIMUM NUMBER
; OF CONSECUTIVE CYCLES OF TONE REQUIRED TO
; RECOGNIZE A ZERO OR ONE BIT
0003      MIN0=3
0004      MAX0=4
0006      MIN1=6
0008      MAX1=8
;
; SOME CHARACTER EQUATES
001F      RUBOUT=1FH    ; BASIC 'ERASE' CHARACTER
000D      NL=0DH       ; ASCII NEW LINE
;
0015      HLTPORT=15H   ; KEYPAD COLUMN FOR HALT KEY
;
4000      .LDC         4000H
; :RUN EXECUTION BEGINS HERE
4000      BEGIN:
4000      STARTUP:
4000      F3          DI          ; NIX INTERRUPTS
4001      AF          XRA         A          ; USE OTHER COLOR REGISTERS
4002      D309        OUT        HORCB
4004      3E02        MVI        A,2        ; GREY SCALE FOREGROUND
4006      D301        OUT        COL1R
4008      07          RLC
4009      D302        OUT        COL2R
400B      3E07        MVI        A,7
400D      D303        OUT        COL3R
400F      3ECC        MVI        A,204
4011      D30A        OUT        VERBL
4013      21 4280     LXI        H,BUFSTART ; RESET 300 BAUD

```

```

4016      22 4F28          SHLD   POINTER ; BUFFER POINTER
          ; WAIT FOR USER TO SWITCH TAPES
          ; WHEN DONE USING ORGANISM WILL PRESS 'GO' KEY
4019      DB17          REREAD:
          ..CUEW: IN      17H      ; SENSE 'GO' KEY COL.
401B      E63F          ANI      3FH      ; IS GO KEY DOWN?
401D      2011          JRNZ     ..GOTO   ; YES - WE GOT ONE
401F      DB15          IN      HLTPORT ; READ HLT COLUMN
4021      A7            ANA      A        ; CHECK HALT COLUMN
4022      C2 410D       JNZ      TAF      ; IF SO ENDIT
4025      03            INX      B        ; SWEEP BC TO SHOW
4026      78            MOV      A,B     ; WE ARE WAITING
4027      0F            RRC          ; FOR CUE
4028      0F            RRC
4029      0F            RRC
402A      E607          ANI      7        ; SWEEP ONLY
402C      D300          OUT      COLOR   ; GREY SCALE
402E      18E9          JMPR     ..CUEW
          ; WE GOT THE CUE
4030      AF            ..GOTO: XRA     A        ; BACK TO
4031      D300          OUT      COLOR   ; BLACK
          ; NOW WE WAIT FOR 10 CONSECUTIVE 1 BITS TO APPEAR
          ; THIS WILL IGNORE ANY GARBAGE APPEARING BEFORE THE
          ; PROGRAM DATA
4033      2E0A          ..WLDR: MVI     L,10    ; (RE)SET BIT CTR
4035      CD 40C6       ..WLL:  CALL    GETBIT  ; GET BIT FROM TAPE
4038      28F9          JRZ      ..WLDR  ; ZERO-START OVER
403A      2D            DCR      L        ; ONE-COUNT IT
403B      20F8          JRNZ     ..WLL   ; AWAIT QUOTA
          ; WE NOW FALL INTO THE ...
          ; AWAIT LINE NUMBER STATE-IGNORE EVERYTHING UNTIL A LINE
          ; NUMBER APPEARS - THIS WILL DISREGARD ANY UNNUMBERED
          ; COMMANDS
          ; IF AN UNNUMBERED 'RUN' OR
          ; :RETURN COMMAND IS FOUND WE ENTER
          ; THE AWAIT NEXT SEGMENT CUE STATE
403D      CD 408E       ..WLN:  CALL    GETBYTE ; GET BYTE FROM TAPE
4040      FE3A          ..NRET: CPI     ':'     ; HAVE WE GOT A COLON?
4042      281A          JRZ      ..COLN  ; YEP, TRY FOR :RETURN
4044      FE6A          CPI      106    ; 'RUN' COMMAND?
4046      28D1          JRZ      REREAD  ; YEP-THEN QUIT
4048      CD 407C       CALL    TSTNUM ; IS SHE A NUMBA?
404B      30F0          JRNC     ..WLN  ; NO-IGNORE IT
404D      CD 4085       ..STUF: CALL   STUFBUF ; STORE HER AWAY
          ; WE ARE NOW ON A NUMBER
          ; NOW WE COPY NUMBERS, AND IGNORE SPACES UNTIL A NONSPACE,
          ; NONNUMERIC IS DETECTED
4050      CD 408E       ..IGNS: CALL   GETBYTE ; FETCH ANOTHER VICTIM
4053      CD 407C       CALL    TSTNUM ; TEST FOR NUMERIC
4056      38F5          JRC     ..STUF ; YEP-KEEP LOOPING
4058      FE20          CPI      ' '    ; A SPACE PERHAPS?
405A      28F4          JRZ      ..IGNS ; IGNORE THE BLANK
405C      1811          JMPR     ..LIN1 ; NONBLANK - LEAVE STATE

```

```

; TRY FOR :RETURN STATE - WE CAME HERE AFTER
; READING IN A COLON, SO IF WE GET A 'RETURN'
; WE CAN CEASE LOADING THIS SEGMENT
405E   CD 408E   ..COLN: CALL   GETBYTE ; READ ANOTHER BYTE
4061   FE20     CPI     ' ' ; ALLOW
4063   28F9     JRZ     ..COLN ; INTERMIXED SPACES
4065   FE70     CPI     112  ; :RETURN TOKEN
4067   20D7     JRNZ    ..NRET ; IF NOT GO BACK
4069   C3 4019  JMP     REREAD ; YEP-END THIS SEG

; READ LINE STATE
; WE WILL HANG OUT HERE UNTIL A NEW LINE COMES IN
; IN WHICH CASE WE WILL GO BACK TO 'AWAIT LINE NUMBER'
406C   CD 408E   ..LINL: CALL   GETBYTE ; FETCH NEXT BYTE
406F   FE1F     ..LIN1: CPI     RUBOUT ; IF RUBOUT IGNORE IT
4071   28F9     JRZ     ..LINL
4073   CD 4085   CALL   STUFBUF ; STUFFER IN THE BUFFER
4076   FE0D     CPI     NL     ; IF NEW LINE
4078   28C3     JRZ     ..WLN  ; WAIT FOR NEXT LINE #
407A   18F0     JMPR   ..LINL  ; ELSE GOFOR NEXT CHAR

; ROUTINE TO TEST FOR CHARACTER BEING NUMERIC
407C   FE30     TSTNUM: CPI     '0'   ; LESS THAN ZERO?
407E   3803     JRC     ..NN    ; NO - JUMP
4080   FE3A     CPI     '9'+1 ; GREATER THAN NINE?
4082   D8       RC      ; RETURN CY IF NOT
4083   A7       ..NN:  ANA     A     ; CLEAR CY BIT
4084   C9       RET     ; TO SAY NOT NUMBER

; ROUTINE TO STORE A CHARACTER INTO THE BUFFER
STUFBUF:
4085   2A 4F28   LHLD   POINTER ; GET BUFFER POINTER
4088   77       MOV     M,A   ; STUFF CHAR WITHIN
4089   23       INX     H     ; BUMP POINTER
408A   22 4F28   SHLD   POINTER ; AND SAVE
408D   C9       RET     ; AND GO HOME

; GET A BYTE FROM AUDIO TAPE ROUTINE
GETBYTE:
408E   CD 40A0   CALL   GETO    ; GET A START BIT
4091   20FB     JRNZ    GETBYTE ; WAIT FOR IT
4093   21 0800   LXI     H,800H ; INIT BIT CTR AND ACCUM
4096   CD 40C6   ..BYTL: CALL   GETBIT ; FETCH A BIT
4099   B5       ORA     L     ; OR IN DATA SO FAR
409A   0F       RRC     ; SHIFT OVER
409B   6F       MOV     L,A   ; AND SAVE
409C   25       DCR     H     ; DCR BIT COUNTER
409D   20F7     JRNZ    ..BYTL ; FILL OUT BYTE
409F   C9       RET

; GET BIT ROUTINE
;
; THIS ROUTINE INPUTS A BIT IN 300 BAUD FORMAT
; USING THE BUILT IN AUDIO INTERFACE CIRCUIT
; IT DOES THIS BY MEASURING THE FREQUENCY OF
; OF THE INPUT SQUARE WAVES.
; AS 8 CONSECUTIVE SHORT SQUARE WAVES DENOTES A ONE BIT
; AND 4 CONSECUTIVE LONG SQUARE WAVES DENOTES A ZERO BIT

```

```

; WE MAINTAIN COUNTERS FOR EACH TYPE, AND RETURN
; ONE OR ZERO WHEN ENOUGH CYCLES ARE COUNTED
;
; THE ENTRY 'GETO' IS ENTERED WHEN A START BIT IS
; SOUGHT. 'GETO' INSISTS ON RECIEVING A FULL LENGTH
; ZERO WAVE
40A0 0E00 GETO: MVI C,0 ; SYNC ON START BIT
40A2 1E00 ..NEXW: MVI E,0 ; RESET BIT LENGTH TIMER
40A4 DB15 IN HLTPORT ; SENSE HALT COLUMN
40A6 A7 ANA A ; IF ANY DOWN
40A7 C2 410D JNZ TAF ; KICKOUT
40AA 3A 3C00 LDA BANGIN ; GET TAPE DATA
40AD 57 MOV D,A ; STUFF IN TRACKER
40AE 3A 3C00 ..GOL: LDA BANGIN ; SENSE IT AGAIN
40B1 AA XRA D ; ANY TRANSITION?
40B2 0F RRC
40B3 3805 JRC ..GOTT ; JUMP IF SO
40B5 1C INR E ; COUNT TIME
40B6 20F6 JRNZ ..GOL ; TIMEOUT COND?
40B8 18E6 JMPR GETO ; YEP - START OVER
40BA 7A ..GOTT: MOV A,D ; LAST TIME TO ACC
40BB 53 MOV D,E ; SET NEW LAST TIME
40BC 83 ADD E ; SUM LAST AND NOW
40BD FE0F CPI OFH ; GOOD ENOUGH?
40BF 38E1 JRC ..NEXW ; NO-KEEP LOOKIN
40C1 01 0001 LXI B,1 ; YEP-COUNT THAT ZERO
40C4 1809 JMPR GBNEXT ; AND JOIN GETBIT
; THIS ENTRY IMPOSES NO SPECIAL CRITERIA
; AND IS ENTERED FROM LEADER WAIT AND TO GET
; CHARACTER DATA BITS
; B=ONE COUNTER, C=ZERO COUNTER
40C6 GETBIT:
40C6 01 0000 LXI B,0 ; RESET BIT COUNTERS
40C9 DB15 IN HLTPORT ; ANOTHER HALT COL
40CB A7 ANA A ; CHECK
40CC C2 410D JNZ TAF
40CF GBNEXT:
40CF 1E00 MVI E,0 ; RESET BIT TIMER
40D1 3A 3C00 LDA BANGIN ; SAMPLE TAPE DATA
40D4 57 MOV D,A ; SAVE IN TRACKER
40D5 3A 3C00 ..TIME: LDA BANGIN ; SAMPLE AGAIN
40D8 AA XRA D ; REVEAL TRANSITION
40D9 0F RRC ; IF WE GOT ONE
40DA 3805 JRC ..TRAN ; JUMP OUT
40DC 1C INR E ; COUNT TIME
40DD 20F6 JRNZ ..TIME ; TIMEOUT CHECK
40DF 18E5 JMPR GETBIT ; TIMEOUT - START OVER
; WE HAVE ONE HALF CYCLE-SO WE GO FOR THE OTHER HALF
; SAME LOGIC AS ABOVE CODE
40E1 ..TRAN:
40E1 3A 3C00 LDA BANGIN
40E4 AA XRA D
40E5 0F RRC

```

```

40E6      3005          JRNC      ..GOTB
40E8      1C           INR        E
40E9      20F6        JRNZ      ..TRAN
40EB      ..TIMO:
40EB      18D9        JMPR      GETBIT ; WE HAVE TIMED OUT
; WE HAVE A FULL WAVE TIME - WE WILL
; CLASSIFY IT AND INCREMENT THE APPROPRIATE BIT
; COUNTER IF THE 'OTHER' BIT COUNTER MEETS
; MINIMUM CRITERION WE WILL RETURN THE 'OTHER' BIT
; VALUE, ELSE IF THE 'NOW' BIT COUNTER IS HIGH
; ENOUGH WE WILL RETURN THE 'NOW' BIT
; IF NEITHER ARE LARGE ENOUGH WE WILL KEEP SAMPLING
40ED      7B          ..GOTB: MOV      A,E
40EE      FE0C        CPI        12      ; WHICH HAVE WE?
40F0      380D        JRC       ..TRY1 ; JUMP IF ONE
40F2      0C          ..TRY0: INR      C      ; COUNT THE ZERO
40F3      78          MOV       A,B      ; ENOUGH ONES?
40F4      FE06        CPI        MIN1
40F6      3012        JRNC      ..RET1 ; YES - RETURN 1 BIT
40F8      79          MOV       A,C      ; AT MAX ZERDS?
40F9      FE04        CPI        MAX0
40FB      38D2        JRC       GBNEXT ; NO - GET MORE CYCLES
40FD      AF          ..RETO: XRA     A      ; RETURN A ZERO BIT
40FE      C9          RET
40FF      04          ..TRY1: INR      B      ; COUNT A ONE BIT
4100      79          MOV       A,C      ; ENOUGH ZERO BITS?
4101      FE03        CPI        MIN0
4103      30F8        JRNC      ..RETO ; YEP - GO TO ZERO RETURN
4105      78          MOV       A,B      ; MAX ONE BITS?
4106      FE08        CPI        MAX1
4108      38C5        JRC       GBNEXT ; YEP
410A      AF          ..RET1: XRA     A
410B      3C          INR       A
410C      C9          RET

```

```

; THIS SEQUENCE IS ENTERED AFTER THE USING ORGANISM
; PRESSES A KEY IN THE HALT COLUMN
; IF THE KEY WAS NOT THE HALT KEY WE GO BACK FOR AN
; ADDITIONAL SEGMENT. IF IT WAS HALT WE CLOSE THE BUFFER
; AND THEN RELOCATE THE BUFFER TO THE HIGHEST POSSIBLE
; ADDRESS, ESTABLISH THE SPECIAL I/O ROUTINES IN
; THE BASIC STACK ZONE, AND MODIFY THE HOOK VECTORS
; POINT AT THEM.
; WE THEN CALL BASIC TO CONVERT THE BUFFER

```

```

410D      TAF:
410D      31 4FEA      LXI      SP,STACK ; RESET THE STACK POINTER
4110      0F          RRC       ; WAS THAT A HALT?
4111      D2 4019      JNC      REREAD  ; NO - GO FOR NEXT SEGMENT
4114      AF          XRA      A      ; DENOTE OUT OF HALT
4115      D300        OUT      COLOR
4117      21 4166      LXI      H,LASTLINE ; CLOSE THE BUFFER WITH
411A      ED5B 4F28    LDED     POINTER ; NL,PRINT,NL,OFFH
411E      01 0005      LXI      B,EOLL-LASTLINE+1

```

TDL Z80 CP/M DISK ASSEMBLER VERSION 2.21
CONVER -

```

4121  EDB0          LDIR
4123  EB           XCHG          ; HL= POINTER
; MOVE BUFFER UP TO HIGHEST POSSIBLE ADDRESSES
4124  11 4E56      LXI          D, LASTBUF+1
4127  2B           ..CPYL: DCX    H          ; DECREMENT POINTERS
4128  1B           DCX          D
4129  7E           MOV          A,M        ; FETCH SOURCE
412A  3600        MVI          M,0        ; CLEAR SOURCE
412C  12           STAX         D          ; STORE AT DEST
412D  7C           MOV          A,H        ; BACK AT START?
412E  FE42        CPI          BUFSTART>8
4130  20F5        JRNZ         ..CPYL    ; LOOP BACK
4132  7D           MOV          A,L
4133  FE80        CPI          BUFSTART&OFFH
4135  20F0        JRNZ         ..CPYL    ; IF NOT YET
4137  ED53 4F28   SDED         POINTER ; SAVE CORRECTED PTR
413B  2A 4E20     LHLD         TXTUNF  ; REMEMBER DATA BENEATH
413E  22 4F26     SHLD        SAVDAT  ; TXTUNF
4141  21 5555     LXI          H,5555H ; INITIALIZE PROGRAM
4144  22 4002     SHLD        4002H  ; STORAGE AREA
4147  21 A004     LXI          H,0A004H ; AND TXTUNF
414A  22 4E20     SHLD        TXTUNF
414D  21 4F2A     LXI          H,TARGET ; PLUG HOOK VECTORS
4150  22 4E99     SHLD        CHKIO+1 ; WITH SPECIAL I/O
4153  23           INX          H          ; ROUTINES
4154  23           INX          H
4155  22 4E9C     SHLD        OUTCH+1
; MOVE THE SPECIAL I/O ROUTINES
; INTO BASIC STACK AREA
4158  21 416A     LXI          H,CODS
415B  11 4F2A     LXI          D,TARGET
415E  01 0056     LXI          B,END-CODS
4161  EDB0          LDIR
4163  C3 2531     JMP          INITO   ; ENTER BASIC
; THIS INFORMATION IS APPENDED TO THE TAPE BUFFER
; TO CLOSE IT OFF. IT WILL CLEAR THE SCREEN AFTER
; THE CONVERSION PROCESS IS COMPLETE
4166  0D           LASTLINE: .BYTE NL
4167  69           .BYTE    105      ; TOKEN CODE FOR 'CLEAR'
4168  0D           .BYTE    NL
4169  FF           .BYTE    OFFH     ; LIST TERMINATOR
416A           EOLL:
416A           CODS:
416A  1817         JMPR         LCHKIO
; SPECIAL OUTPUT CHARACTER ROUTINE
; IT REFUSES TO DO ANY PRINTING
; IT DOES CHECK TO SEE WHERE IT WAS CALLED FROM
; IF IT WOULD GO BACK TO GL2 IN
; GLED WE WILL SKIP SEVERAL UGLY INSTRUCTIONS
; WHICH WOULD PUT UP A CURSOR
416C  E3           XTHL         ; HL=RETURN ADDR
416D  F5           PUSH         PSW   ; SAVE A
416E  7C           MOV          A,H   ; CHECK FOR CALL

```

TDL Z80 CP/M DISK ASSEMBLER VERSION 2.21
 CONVER -

```

416F    FE2C          CPI      GL2>8    ; FROM GL2
4171    200D          JRNZ     ..OK
4173    7D           MOV      A,L
4174    FE68          CPI      GL2&OFFH
4176    2008          JRNZ     ..OK
; WE WERE CALLED FROM GL2, SO SHUFFLE STACK AND
; GO BACK AT GL2A
4178    F1           POP      PSW
4179    E1           POP      H
417A    C5           PUSH     B
417B    E5           PUSH     H
417C    D5           PUSH     D
417D    C3 2C70      JMP      GL2A
; CALL FROM ELSEWHERE - JUST GO HOME
4180    F1           ..OK: POP      PSW
4181    E3           XTHL
4182    C9           RET
; GET CHARACTER ROUTINE - FETCHS CHARACTER
; FROM 300 BAUD BUFFER
; WE RESTORE THE DATA BENEATH TXTUNF JUST IN
; CASE OUR POINTER POINTS THERE
4183    2A 4E20      LCHKIO: LHL  TXTUNF ; SAVE CURRENT TXTUNF
4186    E5           PUSH     H
4187    2A 4F26      LHL  SAVDAT ; GET BUFFER DATA
418A    22 4E20      SHLD  TXTUNF ; AND FUDGE
418D    2A 4F28      LHL  POINTER ; GET BUFFER PTR
4190    7E           MOV     A,M ; AND CHARACTER
4191    3600         MVI     M,0 ; CLEAR CHARACTER
4193    23           INX     H ; BUMP PTR
4194    22 4F28      SHLD  POINTER ; SAVE PTR
4197    2A 4E20      LHL  TXTUNF ; SAVE BUFFER DATA
419A    22 4F26      SHLD  SAVDAT ; AS MAY BE ZEROED
419D    E1           POP      H ; RESTORE TXTUNF
419E    22 4E20      SHLD  TXTUNF
41A1    3C           INR     A ; END OF LIST?
41A2    2802         JRZ     ..GEOF ; YEP JUMP
41A4    3D           DCR     A ; NO - FIXIT
41A5    C9           RET ; AND GO BACK
; WE ARE DONE - RESET VARIABLES TO ZERO
; ZERO A-Z
41A6    FF           ..GEOF: FILL 1[INTP%[.IFE .INTP.,[RST 7]]
41A7    1B           +.BYTE 26+1]
41A8    4E22         .WORD  VARBGN
41AA    0034         .WORD  52
41AC    00           .BYTE  0
; RESET HOOK VECTOR
41AD    FF           MOVE  1[INTP%[.IFE .INTP.,[RST 7]]
41AE    5F           +.BYTE 94+1]
41AF    4E92         .WORD  HKVECT
41B1    000E         .WORD  14
41B3    218B         .WORD  HOOKER
41B5    3EB0         MVI     A,176 ; RESET VERTICAL
41B7    D30A         OUT     VERBL ; BLANK PORT

```

```
41B9      3E2C          MVI     A,44      ; AND HORIZONTAL
41BB      D309          OUT     HORCB     ; COLOR BOUNDARY
41BD      C3 2531       JMP     INITO     ; HALT BASIC
41C0                                END:
                                .END
```