cosmos

SCREEN MIXER

cosmos™

WORLD WIDE COMPUTER SUPER SHOP

ASTAR INTERNATIONAL COMPANY

5676 FRANCIS AVENUE
CHINO, CA91710

ASTAR INTERNATIONAL CO.
5676 FRANCIS AVE.,CHINO
CA 91710 714-627-9887

[SCREEN MIXER MANUAL/ MAR.01,82]

1:PREFACE

1-1.CAPABILITIES

The SCREEN MIXER is a set of three modules which replace three ICs on the Apple-II mother board. Without taking up any slots, the SCREEN MIXER provides more capabilities for your Apple's screen, such as:

Mixing of any two screens from HGR1,HGR2,GR1,GR2,TEXT1 and TEXT2 without software.

Completely new character modes that even Hi-Res Text Generator Software could not generate, found only on a few of the most sophisticated CRT terminals.

Generating Double Density High Resolution Graphics with 560 dots in one line.

80 character display without using an expensive 80-character adapter.

1-2.HARDWARE REQUIREMENTS

The Screen mixer works with any kind of Apple-II including the Apple-II PLUS.

While Screen Mixer works with any kind of CRT display, the Green CRT monitor provides the best results.

When you use the Character Enhance Program (CEP), you must have Applesoft in ROM or in a Language Card, 48K of memory and one disk drive.

The program is stored in DOS 3.2 format and can be converted to DOS 3.3 format using the MUFFIN program on the DOS 3.3 system diskette. There is no protection on the diskette.

80 character display interfaces such as Videoterm are not affected by Screen Mixer. However, the Screen Mixer will not

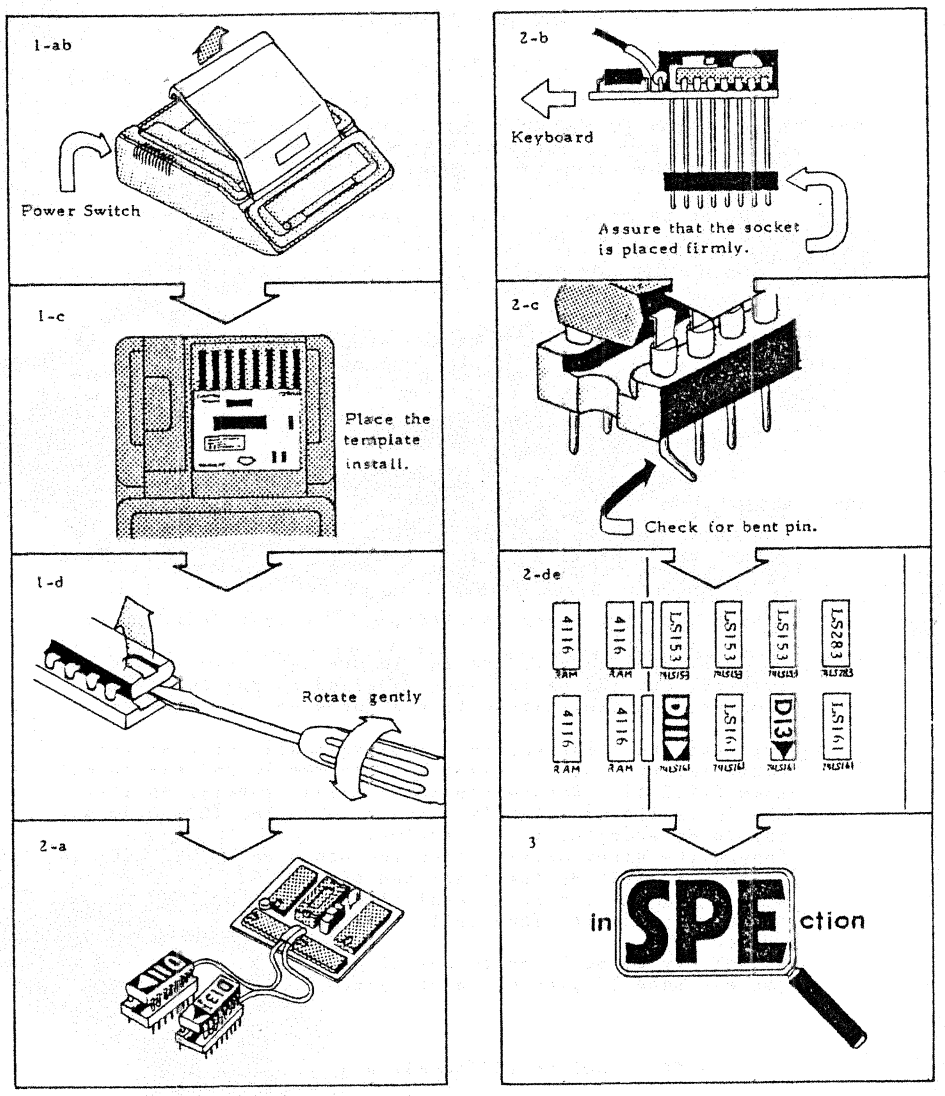


Fig. 2-1 **Installation**

2:HOW TO GET STARTED

2-1.INSTALLATION

NOTE: THIS SECTION IS THE MOST IMPORTANT PART OF THE MANUAL. FAILURE TO FOLLOW THESE INSTRUCTIONS MAY DAMAGE THE SCREEN MIXER OR YOUR APPLE-II OR BOTH!

However, do not be afraid , read the instructions carefuly, and then follow them. Do NOT try to do anything you do not understand.
If you have any difficulty, ask your dealer to install the SCREEN MIXER.

The installation consists of three steps. First, you have to locate three ICs on the Apple mother board, then pull them out. Second , three modules must be placed correctly in the vacant sockets specified. Finaly, all the installation must be inspected.

[STEP 1: IC REMOVAL]

(1-a) Turn OFF the Apple-II and any peripherals connected to the Apple. Assure that the Apple is OFF whenever you are going to install anything in or remove anything from the Apple, or the Apple may be destroyed.

(1-b) Remove the top cover by lifting the rear edge first.

(1-c) Use the supplied template to locate the three ICs to be removed. Put the marker dots supplied on the ICs to be removed. See the instructions printed on the template.

(1-d) Pull out the three marked ICs using small screw driver. First insert a screw driver between the IC and the socket (see FIG.2-1) . Rotate the screw driver gently, so that the IC is lifted slightly. Do not force the IC and do not try to do everything at one time.
Insert the screw driver under the other end of the IC and rotate it, lifting the other end of the IC.
Repeat this procedure several times until the IC comes out of the socket. Then pull the IC out with your fingers.
Do NOT try to pull the IC out if you feel resistance, or you may bend IC pins or injure yourself.

[STEP 2: MODULE INSTALLATION]

(2-a) Find the three SCREEN MIXER modules in their package. There are two small modules and one big module in the kit. Each of three modules has its own location and direction. Failure to install them correctly may damage the computer.

(2-b) Plug the biggest module into socket F-14. See 2-B of fig.2-1 for the direction of the module in its socket. Do NOT force the pins of the module into the socket. First, put all the pins on the holes of the socket, assuring that none of them are bent or out of their holes, press the module down gently.

(2-c) If you feel any resistance, remove the module to check the pins. You may find a bent pin. Use small pliers or your finger nails to fix the bent pin. Again, do not force them. Apply pressure gently.

(2-d) Plug the D-11 module into D-11 socket as described in step (2-b).

(2-e) Plug the D-13 module into D-13 socket as described in step (2-b).

[STEP 3: FINAL CHECK]

Check all the connections. Check carefuly whether the modules are placed correctly.

The Color Trim capacitor near F-14 socket must be visible. if it is under the biggest module board, the module is in the wrong direction.

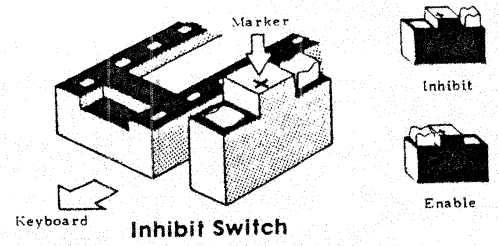
The Small modules (D-11,D-13) have wires leading to the big module. Those wires should be on the back end of the ICs, toward the back of the Apple.

2-2.INHIBIT SWITCH

The Inhibit switch is a small DIP switch on the biggest module. The switch 'kills' all the functions of the SCREEN MIXER. You can not use the SCREEN MIXER if the switch is turned to INHIBIT. Then you have an ordinary Apple-II (Fig.2-2)

Inhibit switch is furnished to prevent conflicts caused by programs which do not support the Screen Mixer. You could have an unexpectedly mixed screen when a program uses screen control commands such as HGR and GR. Almost all game software uses the HGR screens. The Screen Mixer does not recognize whether a program needs mixing or not unless the program tells the Screen Mixer.

Therefore, turn the Screen Mixer off to get the original screen when you have an unexpected mixed screen.



Inhibit Switch

Fig. 2-2

Some software has another conflicts with the Screen Mixer. If a program uses [AN] outputs in the game I/O connector, the screen will be messed up. This is because the Screen Mixer uses these outputs to determine the mode of the Holder-B. However only a few programs use these AN outputs.

Use the inhibit switch to select between the Screen Mixing and normal modes.

2-3.AUTO START ROM AND OLD ROM

There are three versions of Apple-II ROM Firmwares in the market (theoricaly four), those are:

(1) Old Monitor ROM and Integer BASIC
(2) Auto Start ROM and Integer BASIC
(3) Auto Start ROM and Applesoft Floating Point BASIC

Both BASICs work well with the SCREEN MIXER.
However there is a deference between Auto Start ROM and Old ROM. When the Apple-II is turned on:

Auto Start ROM initializes the SCREEN MIXER for text page-1. This is the default mode and the screen of Apple-II appeares as ordinary Apple.

Old ROM does not initialize the output ports of Apple-II with the same manner of Auto Start ROM. as a result the mixed screen may appear.

The user of Old ROM must type

C05A (RETURN)
C05F (RETURN)

just after the power is applied to your Apple-II. This must be done only after the power up. Reset or re-boot does not affect the setting.

3:THE SCREEN MIXER

3-1.THEORY OF OPERATION

The SCREEN MIXER mixes screens as its name implies, but how does it work?

Suppose there are several slides on a table. Pick up any two of them and lay one on another. Now you can see both pictures mixed together but each picture does not affect the other one. You can replace one picture with another one without affecting the remaining slide.

The operation of the SCREEN MIXER is somewhat similar. Apple-II has three kinds of screens. Those are Text(TEXT), Low Resolution Graphics(LGR), and High Resolution Graphics (HGR) screens.Each screen has two pages and in this manual we will call these screens TEXT-1,TEXT-2,LGR-1, LGR-2, HGR-1 and HGR-2.

The excellent design of Apple-II made it possible to include four text lines on the bottom of graphics screen. But this is not the screen mixing we are discussing here because you cannot mix screens at the same position of the display. Thus you cannot print characters on the HGR screen nor mix two graphics screens. The SCREEN MIXER makes it possible, using an imaginary holder of screen.

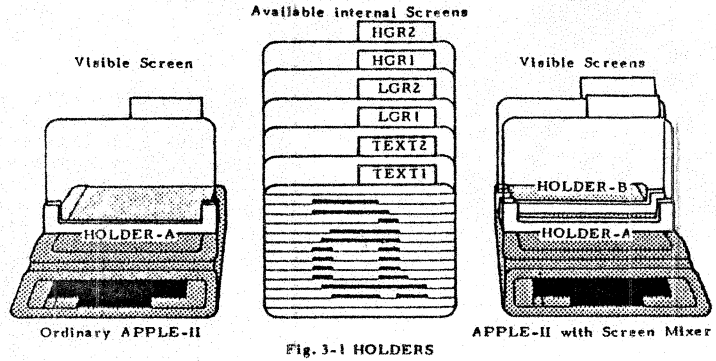


Fig. 3-1 HOLDERS

(3-1-1) HOLDERS

Here we introduce an imadinary holder of the screens for the explanation. The holder can hold any of Apple's screen. While the Apple-II has four pages of the screens, only the screen which is set in the holder is visible and the remaining screens are hidden in the memory. Some of the BASIC commands such as TEXT,HGR or GR changes the contents of the holder.

The ordinary Apple-II has only one (imaginary) holder. Here we call it HOLDER-A. The Screen Mixer adds the second holder in your Apple-II. We call this new holder HOLDER-B (Fig.3-1)

HOLDER-A is built in to the ordinary Apple-II. Several BASIC commands such as TEXT,GR and and HGR set this holder to the corresponding modes.
It is also possible to set them to the informal modes using PEEK or POKE procedures to the corresponding addresses. These procedures are described in page 13 of the Apple-II Reference Manual.

HOLDER-B is a new display made by the SCREEN MIXER. You can set it for any screens just like HOLDER-A. The only difference is the corresponding addresses to be PEEKed or POKEd. All modes including Text and Graphics Mix mode can be set.

You can see the mixed screen when you set two different screens for these holders.
You may set the same screen to both holders and you will get ordinary screen.

(3-1-2) THE MIXING OF TWO TEXT SCREENS

Text screen mixing is useful for text processing. You can have new character modes such as Halftone,Highlight and Over-Type. The difficulty is that the Apple-II monitor ROM does not support the second page of the text (TEXT-2), while we need both of TEXT-1 and TEXT-2.
The problem is more serious with Applesoft BASIC. Applesoft BASIC uses the memory area for TEXT-2 for an important part of the user program and Applesoft command including HIMEM: and LOMEM: do not solve the problem.

Our Character Enhance Program solves the problem.
It moves the pointer for Applesoft upwords and makes room for TEXT-2. It also accepts several control codes to define the character mode. More details will be provided later.

MIXING OF HGR AND TEXT

This is one of the most important functions of the Screen Mixer.

There are Several High Resolution Character Generator Softwares available in the market. They are excellent when you need new character sets such as lower case or foreign characters. However there is trade off with print out speed. Also it is difficult to handle HGR and TEXT separate since a character is just a graphics pattern. For example, if you want to scroll TEXT but not HGR, the program must replace original HGR screen without TEXT, and then put the scrolled characters on it again. Thus scrolling is even slower than normal.

The SCREEN MIXER does not create any new character set. Instead you may use the usual TEXT print procedures such as the PRINT command of BASIC or COUT1 routine of the monitor program. Also you may HPLOT or DRAW on the HGR screen and both results will be mixed on the screen. The text is scrolled at full speed while HGR remains unchanged.

You need no special software as long as you use HGR-1,HGR-2 and TEXT-1. You should have a machine language routine (not included in this kit) if you want to use TEXT-2, as explained above.

The HGR command mixes HGR-1 and TEXT-1.
While the bottom four lines display just TEXT, you may use:

```
POKE -16302,0
or  A=PEEK -16302
```

to get complete HGR and TEXT mixing.

The HGR2 command mixes HGR-2 and TEXT-1.
This is similar to HGR, however the whole screen is mixed.

MIXING OF LGR AND TEXT

You must be careful when you choose the combination of TEXT and LGR. This is because TEXT and LGR use the same memory area. TEXT-1 and LGR-1 occupy memory from $0400 through $07FF and TEXT-2 and LGR-2 occupy from $0800 through $0BFF.

The Apple ROM Monitor and Applesoft use only TEXT-1 for text output and LGR-1 for Lo-Res Graphics. TEXT2 and LGR2 are invalid commands in Applesoft.

Thus without handler software, this mode has no value. To utilize this mode, first you have to move the pointer of Applesoft upward, set HOLDER-B for LGR-2 and then POKE data to the LGR-2 page which is located $0800 through $0BFF.

MIXING OF HGR AND LGR

This is similar to TEXT+HGR mode.
First you have to use HGR (or HGR2) to set Applesoft to recognise HPLOT command. Then use GR command.

In this case you can HPLOT on HGR screen and PLOT on LGR screen without switching from one mode to the other one. Since Applesoft does not supprt LGR2, you can have only two combinations those are LGR1+HGR1 and LGR1+HGR2.

MIXING OF HGR1 AND HGR2

In this mode, you can plot up to 560 dots in one line or you can have 280 of half tone, black or white dots in one line.

(Half tone display)

The reduced intensity is made by alternating between two screens.

Black dot
If there is no dot on either of the two HGR screens, it will appear as a black dot or 'no dot'.

Half Intensity
If a dot is on one screen and not on the other screen the dot displays 30 times a second and appears to have reduced intensity.

Full Intensity
If there are two dots in the same position in both screens, they forms a Full Intensity dot or 'white dot'.

In this case we assume that you HPLOT only white dots. You will have ordinary 280 dots resolution.

(3-1-3) HOLDER CONTROL ADDRESSES

Holders can be set by PEEKing or POKEing the corresponding addresses. The addresses of HOLDER-A are described on page 13 of the Apple Reference Manual.

HOLDER-A			HOLDER-B	
Hex	Decimal	Function	Decimal	Hex
$C050	49232	½ GRAPHICS mode	½ 49246	$C05E
$C051	49233	½ TEXT mode	½ 49247	$C05F
$C052	49234	½ All TEXT or GRAPHICS	½ 49244	$C05C
$C053	49235	½ Mix 4 lines TEXT w/GRAPHICS	½ 49245	$C05D
$C054	49236	½ Page-1	½ 49242	$C05A
$C055	49237	½ Page-2	½ 49243	$C05B
$C056	49238	½ LGR	½ 49240	$C058
$C057	49239	½ HGR	½ 49241	$C059

49234

4:CHARACTER ENHANCE PROGRAM (CEP)

Character Enhance Progam (CEP) adds a new dimension to the Apple screen. Besides ordinary NORMAL and INVERSE characters, you can use Half Intensity character and Highlighted characters. Also overwritten characters may be used.

table 4-1 shows the whole set of the character mode available.

CEP is written for Applesoft in ROM or Applesoft on the Language Card. It does not work with Integer BASIC or Applesoft in RAM. A machine language program may use CEP through COUT routine, however care must be taken not to use the memory areas which are used by CEP (see memory map).

Please note that CEP does NOT generate any new charater set such as lower case characters or graphics patterns.

The hardware lower case adapters work with the CEP as long as the adapter generates lower case by character generator ROM.

You cannot use CEP with 80 character video interfaces such as the Videoterm since their video signals are generated on their own boards and are beyond the control of Screen Mixer.

4-1.LOADING

Boot the utility disk to load CEP, this procedure erases all programs in the memory.

4-2.CONTROL CODE

Some control codes are used to change the character mode. Those codes are listed in Table 4-1. They may apear in strings with the PRINT command or may be used with PRINT CHR$ command. There is no limitation about the number of these control codes in one string. Thus you may change the mode character by character in the same string. Please note that Apple ignores control codes in a string when it copies the string with a right arrow key.

Direct use of the control code changes the mode, however a syntax error message will be displayed. To avoid this, type the conrol code followed by control-X.

NORMAL

CODE	CHARACTER	BACK GROUND
N	Full Intensity	Black
V	Half Intensity	Black
W	Highlight	Full Intensity	Half Intensity box
O	Overwrite	Half Intensity	Black*

INVERSE

N	Full Intensity	Black	Full Intensity box
V	Half Intensity	Black	Half Intensity box
W	Highlight	Half Intensity	Full Intensity box
O	Overwrite	Black	Half Intensity box*

Actual codes are used with 'CTRL' key.

* There can be another character.

TABLE 4-1 CONTROL CODE LIST

COMMAND LIST

HOME	Clears HOLDER-B screen. HOLDER-A remains unchanged.
&HOME	Clears TEXT screen.
TEXT	Sets TEXT mode and disables CEP.
&TEXT	Sets TEXT mode and enables CEP.
HGR	Mixes HGR-1 and TEXT-1.
HGR2	Mixes HGR-2 and TEXT-1.
RESET	RESET KEY resets to &TEXT mode.(this is not a command))

TABLE 4-2 COMMAND LIST

ADDITIONAL COMMAND FOR APPLESOFT

There are several additional commands for Applesoft to ease the use of CEP.

HOME: does not work in the same manner as on ordinary Apple. HOME erases only the text in HOLDER-B and the text in HOLDER-A remains unchanged. As a result, all the characters on the screen will be Half Intensity while the cursor goes to the home position.

&HOME: is the substitute of the ordinary HOME command. It clears all the text on the screens and puts the cursor at the home position.

TEXT: works normally. However the TEXT command disconnects CEP from Applesoft, so none of the functions of the CEP will not work after a TEXT command.
When you get unexpectedly mixed screens, use this command to avoid them and use &TEXT to resume CEP mode.

&TEXT: works the same way as TEXT except it re-connects CEP with Applesoft. We suggest you to use &TEXT as the substitute for tthe TEXT command to eliminate unnecessary disconnection of the CEP.
This command is also useful to use with the TEXT command to enable and disable the CEP function in your program.

PR#n: Use this command with the TEXT and the &TEXT commands as follows.

```
      :
      :
      10 TEXT            : disconnects the CEP.
      20 PR#n            : activates the slot.
      :
      :   (processes)
      :
     100 PR#0            : disconnects the slot.
     110 &TEXT           : activates the CEP.
      :
      :
```

As shown above, disconnect the CEP before the PR#n command and resume the CEP after the PR#0 command. The PR#n command without associated the TEXT command causes the scroll of only one screen while the other screen remains same position (try it!).

HGR : mixes HGR-1 and TEXT-1 screens. HOLDER-A holds HGR-1 and HOLDER-B holds TEXT-1. Thus both HOME and &HOME clear text on the screen while the HGR screen remains unchanged.

HGR2: is identical to HGR except that it mixes HGR-2 with TEXT-1.

RESET:key works normally and enables the CEP.

If you need only an HGR screen and do not want a mixed screen, use the combination of TEXT and HGR commands.

TEXT:HGR

This set of commands disable the mixing screen and display HGR screen only. Then you have a choice of TEXT or &TEXT to return to text mode. As you already know, TEXT does not connect the CEP to Applesoft while &TEXT does.

4-4.CEP CONFIGURATION PROGRAM (CONFIG CEP)

The Configuration program for the CEP (CONFIG CEP) changes the control codes and the memory map of the CEP.

You must have the CEP in memory to configure it. So boot the disk, and then RUN CONFIG CEP. If you want to keep the new configuration, prepare a new formatted disk.

Follow the instructions on the screen to configure.

(4-4-1) CODE CONFIGURATION

You may change the character mode control code to any other control code. However some control codes are already defined for some functions. For example, control-G generates a bell sound and must not be used as the character control code.
You must not press 'CTRL' key for new code, instead just type an alphabet key. The program automaticaly converts the alphabetic key to the corresponding control code.
The change of the control code becomes valid right after you enter the new code and press 'RETURN'. You can also save CEP with new configuration with CREATE NEW DISK command which is described later.

(4-4-2) MEMORY MAP

Section 4-5 shows the memory map of a 48k Apple with Applesoft in ROM ,DOS and CEP.

The Lowest 4k bytes are used for the CEP , a work area, and two pages of TEXT or LGR screens. The highest portion of memory is occupied by DOS.

Thus there remains plenty of memory between $1000 and $9600 for your BASIC program and variables.

However, when you use HGR screens, you have to keep $2000-3FFF for HGR-1 and $4000-$5FFF for HGR-2. And you can use only 4K of memory from $1000 to $1FFF for Applesoft since it requires continuous memory. And more than 13K of memory from $6000 to $95FF remains unused.

To solve this problem, CEP changes the memory map. As a result you have more than 13K of memory for Applesoft programs and 4K of memory ($1000-$1FFF) remains unused.

This is the default memory map forthe CEP. However if you need more memory for your Applesoft program and not HGR-2, you may change the pointer to $4000 to get another 8K of memory. Even more memory? If you need neither HGR-1 nor HGR-2 then you may move the pointer to $1000 to use all available memory.

Just follow the instructions on the screen to change the memory map. Please note, unlike configuration of control code the memory map configuration is not valid until the new diskette is created and booted.
NEW command does not change the actual memory map and the memory map is remains unchanged even new memory locations are entered.
To change the memory map, you must create a new diskette with CREATE NEW FILE command of the configuration program. And then boot the new diskette to get the new configuration. The CONFIG CEP program alters only the parameter for the cold start initialize routine of the CEP and does not alter the current parameter for the CEP. This is done to avoid system hang up.

(4-4-3) CREATE NEW FILE

A System Diskette can be made with the 'Create New File' command of the CEP. With a System Diskette, you have the CEP in a new configuration whenever you boot the diskette.

First you must have the formatted diskette with the HELLO program listed below:

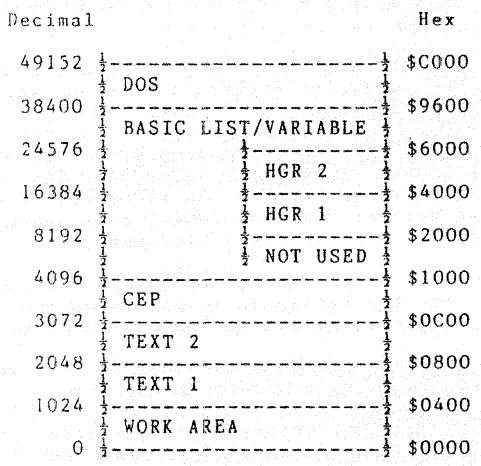
```
10 D$=CHR$(4) : REM CONTROL-D
20 HOME : VTAB 10 : HTAB 10
30 INVERSE : PRINT "CONFIGURED CEP LOADING" : NORMAL
40 PRINT D$; "BRUN CODE.CEP"
```

You must not configure the original diskette or accidental clash may result.

Follow the instructions from the computer.

After you have created the disk, turn the Apple off. Power it up. Boot the diskette, and then check it.

4-5.MEMORY MAP



5:DOUBLE DENSITY HIGH RESOLUTION GRAPHICS

560 dots per line are generated by the mixing of two HGR screens. In this case one screen is moved half a dot to right while the other screen stays. Thus two screen images are superimposed (Fig.5-1)

To move the screen half a dot, you must set the Most Significant Bit of all of the refresh memory in HGR. You have to reset the MSB of all the memory in the other page.

This sounds somewhat technical and hard to understand, however you only have to chose the value of HCOLOR for each of HGR screens.

First, you have to set the mixing mode of the two HGRs.

```
HGR           : Sets HOLDER-A for :HGR-1
POKE 49234,0  :                   :all graphics
POKE 49246,0  : Sets HOLDER-B for :graphics
POKE 49244,0  :                   :all graphics
POKE 49243,0  :                   :page-2
POKE 49241,0  :                   :high resolution
```

For best results, you must not use color dots. You may use only Black and White. And there are two sets of HCOLOR for Black and White combinations.

```
SET-A  Black:HCOLOR=0    this is original position
       White:HCOLOR=3    (MSB is reset)

SET-B  Black:HCOLOR=4    moves half a dot to right
       White:HCOLOR=7    (MSB is set)
```

You must use set-A for HGR-1 and set-B for HGR-2, and must not mix them up.

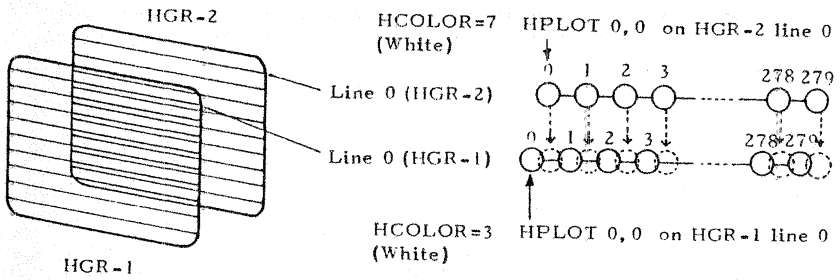


Fig.5-1 DOTS' POSITIONS

You also must switch between the two HGR screens. However, whenever you use the HGR and HGR2 commands , the screen will be erased and cannot be kept. Instead of these commands, you have to POKE the page number of HGR to the corresponding addres.

POKE 230,32 : Sets for HGR-1 plotting.

POKE 230,64 : Sets for HGR-2 plotting.

POKEd value must be 32 or 64. The Apple may be hung up if other values are POKEd.
It changes only the page of HGR screen to be plotted by HPLOT and other drawing commands. It does NOT changes the screen mode.

6:APPENDIX

6-1.FLICKER

An ordinary Apple-II CRT displays 60 screens each second like a TV. SCREEN MIXER also generates 60 screens or 60 frames each second too. However the Screen Mixer scans alternate lines each pass. Thus each screen is displayed only 30 frames every second and there is flicker.

The actual degree of flicker may vary from one CRT monitor to another. Generaly speaking, Black and White Monitors make less flicker than Color Monitors and Green Monitors make even less flicker.

The Green Filter on the front of the Green Monitor reduces flicker even more.

A darker screen makes less flicker than a brighter screen. So adjust bright control as dark as possible for the best result.

A screen with flicker makes your eyes tired. If you intend to use SCREEN MIXER for long periods, we suggest that you design your program to reduce the actual use of the mixed screen. You can select the ordinary screen and the mixed screen with small a program (with PEEK or POKE).

While you use the mixed screen you can reduce the flicker with well planned screen design. The flicker occurs only when the dot is on one screen and not on the other screen. When you put two dots on the same position on the different Holders, no flicker occurs on the position.

When you use the Character Enhance Program use the full intensity character as the normal character, and use the half intensity character as the special effect. Thus most of the screen will be flicker free.

If you have a CRT Monitor specialy designed for character display, you need not worry about flicker at all.

```
LINE# LOC CODE   LINE

0002 0000                        ; *************************
0003 0000                        ; * SCREEN MIXER CONTROLER *
0004 0000                        ; *    DEC.01,1981 S.SATO  *
0005 0000                        ; *************************
0006 0000                        ;
0007 0000                        ; Utility program for SCREEN MIXER.
0008 0000                        ; This program provides the capability
0009 0000                        ; of display Half Intensity, Hilighted
0010 0000                        ; and Over-Written characters besides
0011 0000                        ; ordinary Noramal and Inverse characte
0012 0000                        ;
0013 0000                        ; Memory Map
0014 0000                        ;    $0000-03FF Same as ordinary Apple.
0015 0000                        ;    $0400-07FF Text page-1
0016 0000                        ;    $0800-0BFF Text page-2
0017 0000                        ;    $0C00-0FFF Screen Mixer Program
0018 0000                        ;    $1000-1FFF Not used
0019 0000                        ;    $2000-3FFF Reserved for HGR page-1
0020 0000                        ;    $4000-5FFF Reserved for HGR page-2
0021 0000                        ;    $6000-95FF Program listing & stack
0022 0000                        ;    $9600-BFFF DOS
0023 0000                        ;
0024 0000                        ;    Pointer for Applesoft ($67-68) may
0025 0000                        ;    be lowered if HGR-2 or both of HGRs
0026 0000                        ;    are not used.
0027 0000                        ;
0028 0000                        ; [ ZERO PAGE USAGE ]
0029 0000                        ;
0030 0000        TEMP1=$00
0031 0000        WNDLFT=$20
0032 0000        WNDWDT=$21
0033 0000        WNDTOP=$22
0034 0000        WNDBTM=$23
0035 0000        CH=$24          ; H-CURSOR POSITION
0036 0000        CV=$25          ; V-CURSOR POSITION
0037 0000        BASL=$28
0038 0000        BASH=$29
0039 0000        BAS2L=$2A
0040 0000        BAS2H=$2B
0041 0000        INVFLG=$32
0042 0000        YSAV1=$35
0043 0000        CSWL=$36        ; CHARACTER OUTPUT
0044 0000        CSWH=$37
0045 0000        KSWL=$38        ; CHARACTER INPUT
0046 0000        KSWH=$39
0047 0000                        ;
0048 0000                        ; [ VECTORS ]
0049 0000                        ;
0050 0000        SOFTEV=$03F2    ; RESET VECTOR (WARM)
0051 0000        PWREDU=$03F4    ; PWREDU=SOFTEV+1 EOR #$A5
0052 0000        AMPSND=$03F5    ; AMPERSAND VECTOR
0053 0000                        ;
0054 0000                        ; [ APPLESOFT ]
0055 0000                        ;     <ZERO PAGE>
0056 0000        TXTTAB=$0067    ; START OF BASIC ATATEMENTS
```



```
LINE# LOC CODE   LINE

0057 0000                        ;      <RAM ENTRY POINT>
0058 0000        CHRGET=$00B1    ; ENTRY POINT FOR CHARCTER GET
0059 0000        TXTPTR=$00B8    ; POINTER
0060 0000                        ;      <ROM ENTRY POINT>
0061 0000        SCRTCH=$D64B    ; 'NEW' SUB ROUTINE
0062 0000                        ;
0063 0000                        ; [ MONITOR ROM ENTRY POINTS ]
0064 0000                        ;
0065 0000        BELL=$FBD9
0066 0000        COUT=$FDED      ; TO CSWL
0067 0000        COUT1=$FDF0     ; TO CRT
0068 0000        ERROR=$D412     ; ERROR HANDLING
0069 0000        KEYIN=$FD1B
0070 0000        OHOME=$FC58     ; ORIGINAL HOME (-937)
0071 0000        RDKEY=$FD0C
0072 0000                        ;
0073 0000                        ; [ I/O LOCATIONS ]
0074 0000                        ;
0075 0000        KBD=$C000
0076 0000        KBDSTR=$C010
0077 0000                        ;
0078 0000                        ; [ CONTROL ADRESS/BASE MODE ]
0079 0000                        ;
0080 0000        TXTCLR=$C050    ; SET GRAPHICS MODE
0081 0000        TXTSET=$C051    ;     TEXT MODE
0082 0000        MIXCLR=$C052    ; FULL SCREEN FOR GRAPHICS
0083 0000        MIXSET=$C053    ; GRAPHICS+TEXT
0084 0000        LOWSCR=$C054    ; PAGE 1
0085 0000        HISCR=$C055     ;      2
0086 0000        LORES=$C056     ; RESOLUTION: LO
0087 0000        HIRES=$C057     ;             HIGH
0088 0000                        ;
0089 0000                        ; [ CONTROL ADRESS/EXTENSION ]
0090 0000                        ;
0091 0000        TXTCL2=$C05E
0092 0000        TXTSE2=$C05F
0093 0000        MIXCL2=$C05C
0094 0000        MIXSE2=$C05D
0095 0000        LOWSC2=$C05A
0096 0000        HISCR2=$C05B
0097 0000        LORES2=$C058
0098 0000        HIRES2=$C059
0099 0000                        ;
0100 0000              *=$0C00   ; RIGHT AFTER THE TEXT PAGE-2
0101 0C00                        ;
0102 0C00                        ;-[ RESERVATION / PERMISSION REQUIRED
0103 0C00                        ;
0104 0C00                        ; These adresses must not be changed.
0105 0C00                        ; They are considered unchanged from
0106 0C00                        ; to version and the contents of them
0107 0C00                        ; referenced or changed by outer soft
0108 0C00                        ;
0109 0C00 4C2E0C START  JMP INITC ; COLD START (Don't call twice!)
0110 0C03 4C400C SWARM  JMP INITS ; WARM with screen & I/O set.
0111 0C06 4C430C SETIO  JMP INITI ; WARM with I/O set.
```

```
LINE# LOC CODE   LINE

0112 0C09 4C090C AUX1    JMP AUX1        ; For extention use these tables
0113 0C0C 4C0C0C AUX2    JMP AUX2        ; to connect user supplied
0114 0C0F 4C0F0C AUX3    JMP AUX3        ; routines.
0115 0C12 4C120C AUX4    JMP AUX4
0116 0C15 4C150C AUX5    JMP AUX5
0117 0C18 4C180C AUX6    JMP AUX6
0118 0C1B 4C1B0C AUX7    JMP AUX7
0119 0C1E 4C1E0C AUX8    JMP AUX8
0120 0C21 4C210C AUX9    JMP AUX9
0121 0C24 4C240C AUX10   JMP AUX10
0122 0C27                               ;
0123 0C27                               ; [ MODE SELECT CODE TABLE ]
0124 0C27                               ;
0125 0C27 8E      CODENO .BYT $8E        ; CTRL-N  NORMAL
0126 0C28 8F      CODEOW .BYT $8F        ; CTRL-O  OVER WRITE
0127 0C29 96      CODEHI .BYT $96        ; CTRL-V  HALF INTENSITY
0128 0C2A 97      CODEHL .BYT $97        ; CTRL-W  HIGHLIGHT
0129 0C2B                               ;
0130 0C2B 60      BSTART .BYT $60        ; DEFAULT OF LOWEST MEMORY
0131 0C2C                               ;
0132 0C2C 00      WRESET .BYT $00,$00    ; WARM RESET VECTOR
0132 0C2D 00
0133 0C2E                               ;
0134 0C2E                               ;-[ RESERVATION ENDS HERE ]----------
0135 0C2E                               ;
0136 0C2E 205A0C INITC   JSR SETRV      ; SET RESET VECTOR
0137 0C31 20860C         JSR AMPSET     ; SET AMPERSAND VECTOR
0138 0C34 20760C         JSR ASSET      ; MOVE BASIC POINTER
0139 0C37 204BD6         JSR SCRTCH     ; NEW
0140 0C3A 2058FC         JSR OHOME      ; CLEAR TEXT PAGE-1
0141 0C3D 20F60C         JSR CLRSC2     ; CLEAR TEXT PAGE-2
0142 0C40 20100D INITS   JSR SCRSET     ; SET SCREEN MODE
0143 0C43 20490C INITI   JSR INITIO     ; CONNECT I/O
0144 0C46 6C2C0C         JMP (WRESET)
0145 0C49                               ;
0146 0C49 A90C   INITIO  LDA #>SMKEY    ; CONNECT INPUT ROUTINE
0147 0C4B 8539           STA KSWH
0148 0C4D A9BA           LDA #<SMKEY
0149 0C4F 8538           STA KSWL
0150 0C51 A90D           LDA #>POUT     ; CONNECT OUTPUT ROUTINE
0151 0C53 8537           STA CSWH
0152 0C55 A929           LDA #<POUT
0153 0C57 8536           STA CSWL
0154 0C59 60             RTS
0155 0C5A                               ;
0156 0C5A ADF203 SETRV   LDA SOFTEV     ; CHANGE RESET VECTOR
0157 0C5D 8D2C0C         STA WRESET
0158 0C60 ADF303         LDA SOFTEV+1
0159 0C63 8D2D0C         STA WRESET+1
0160 0C66 A90C           LDA #>SWARM
0161 0C68 8DF303         STA SOFTEV+1
0162 0C6B 49A5           EOR #$A5       ; MAKE FUNNY COMPLEMENT
0163 0C6D 8DF403         STA PWREDU
0164 0C70 A903           LDA #<SWARM
0165 0C72 8DF203         STA SOFTEV
```



```
LINE# LOC CODE   LINE

0166 0C75 60             RTS
0167 0C76                               ;
0168 0C76 AD2B0C ASSET   LDA BSTART     ; SET APPLESOFT START ADR.
0169 0C79 8568           STA TXTTAB+1
0170 0C7B A000           LDY #$00
0171 0C7D A900           LDA #$00
0172 0C7F 8567           STA TXTTAB
0173 0C81 9167           STA (TXTTAB),Y
0174 0C83 E667           INC TXTTAB
0175 0C85 60             RTS
0176 0C86                               ;
0177 0C86 A94C   AMPSET  LDA #$4C       ; SET AMPERSAND VECTOR
0178 0C88 8DF503         STA AMPSND
0179 0C8B A996           LDA #<AMPROC
0180 0C8D 8DF603         STA AMPSND+1
0181 0C90 A90C           LDA #>AMPROC
0182 0C92 8DF703         STA AMPSND+2
0183 0C95 60             RTS
0184 0C96                               ;
0185 0C96 A000   AMPROC  LDY #$00
0186 0C98 B1B8           LDA (TXTPTR),Y
0187 0C9A C997           CMP #$97       ; IS IT HOME TOKEN?
0188 0C9C F009           BEQ XHOME
0189 0C9E C989           CMP #$89       ; IS IT TEXT?
0190 0CA0 F00E           BEQ XTEXT
0191 0CA2 A210           LDX #$10
0192 0CA4 4C12D4         JMP ERROR
0193 0CA7                               ;
0194 0CA7 20F60C XHOME   JSR CLRSC2     ; & HOME FUNCTION
0195 0CAA 2058FC         JSR OHOME      ; CLEARS BOTH SCREEN
0196 0CAD 4CB100         JMP CHRGET
0197 0CB0                               ;
0198 0CB0 A900   XTEXT   LDA #$0        ; & TEXT FUNCTION
0199 0CB2 8522           STA WNDTOP     ; SET TEXT MODE AND RECONNECT
0200 0CB4 20400C         JSR INITS      ; SCREEN MIXER I/O ROUTINE.
0201 0CB7 4CB100         JMP CHRGET
0202 0CBA                               ;
0203 0CBA 20E10C SMKEY   JSR SET2P      ; SET CURSOR ON PAGE-2
0204 0CBD 48             PHA
0205 0CBE B128           LDA (BASL),Y
0206 0CC0 8DAE0E         STA TEMPC
0207 0CC3 293F           AND #$3F       ; FLASHING
0208 0CC5 0940           ORA #$40
0209 0CC7 9128           STA (BASL),Y
0210 0CC9                               ;
0211 0CC9 20EB0C         JSR SET1P      ; RESET TO PAGE-1
0212 0CCC 68             PLA            ; RECOVER SCREEN CHARACTER P-1
0213 0CCD 201BFD         JSR KEYIN
0214 0CD0 20E10C         JSR SET2P      ; ERASE CURSOR P-2
0215 0CD3 ADAE0E         LDA TEMPC
0216 0CD6 9128           STA (BASL),Y
0217 0CD8 20EB0C         JSR SET1P
0218 0CDB AD00C0         LDA KBD        ; RECOVER KEYBOARD INPUT
0219 0CDE 0980           ORA #$80
0220 0CE0                               ;
```

```
LINE# LOC  CODE    LINE

0221 0CE0 60              RTS
0222 0CE1                                ;
0223 0CE1 18       SET2P  CLC
0224 0CE2 48              PHA
0225 0CE3 A529            LDA BASH
0226 0CE5 6904            ADC #$04
0227 0CE7 8529            STA BASH
0228 0CE9 68              PLA
0229 0CEA 60              RTS
0230 0CEB                                ;
0231 0CEB 18       SET1P  CLC
0232 0CEC 48              PHA
0233 0CED A529            LDA BASH
0234 0CEF 38              SEC
0235 0CF0 E904            SBC #$04
0236 0CF2 8529            STA BASH
0237 0CF4 68              PLA
0238 0CF5 60              RTS
0239 0CF6                                ;
0240 0CF6 A900     CLRSC2 LDA #$00
0241 0CF8 8500            STA TEMP1
0242 0CFA A908            LDA #$08
0243 0CFC 8501            STA TEMP1+1
0244 0CFE A20C            LDX #$0C
0245 0D00 A9A0     CLRS2  LDA #$A0
0246 0D02 A000            LDY #$00
0247 0D04 9100     CLRS1  STA (TEMP1),Y
0248 0D06 C8              INY
0249 0D07 D0FB            BNE CLRS1
0250 0D09 E601            INC TEMP1+1
0251 0D0B E401            CPX TEMP1+1
0252 0D0D D0F1            BNE CLRS2
0253 0D0F 60              RTS
0254 0D10                                ;
0255 0D10 8D51C0 SCRSET   STA TXTSET     ; SET BOTH MODE FOR:
0256 0D13 8D5FC0          STA TXTSE2     ;    TEXT,MIX,LORES
0257 0D16 8D53C0          STA MIXSET     ; SET BASE MODE FOR PAGE-1
0258 0D19 8D5DC0          STA MIXSE2     ;     EXTENSION FOR PAGE-2
0259 0D1C 8D55C0          STA HISCR
0260 0D1F 8D5AC0          STA LOWSC2
0261 0D22 8D56C0          STA LORES
0262 0D25 8D58C0          STA LORES2
0263 0D28 60              RTS
0264 0D29                                ;

0265 0D29          .FILE 'COUT'
0266 0D29                                ;
0267 0D29                                ; [ ALTERNATE OF COUT1 ]
0268 0D29                                ;     BASE PAGE CHANGED TO 2 ($800-)
0269 0D29                                ;     BY THE ALTERNATION OF CONTENTS OF
0270 0D29                                ;     $FBC6 FROM $04 TO $08
0271 0D29                                ;
0272 0D29 48       POUT   PHA
0273 0D2A 208C0E          JSR SAVEP1
0274 0D2D ADAF0E          LDA LISTOP     ; STOP FLAG=1?
0275 0D30 F00B            BEQ CONT       ; NO, CONTINUE LISTING
```



```
LINE# LOC  CODE    LINE

0276 0D32 CEAF0E          DEC LISTOP     ; RESET FLAG
0277 0D35 2C00C0 HOLD     BIT KBD         ; WAIT TILL NEXT KEY PRESS
0278 0D38 10FB            BPL HOLD
0279 0D3A 8D10C0          STA KBDSTR
0280 0D3D 18       CONT   CLC
0281 0D3E A904            LDA #$04
0282 0D40 6529            ADC BASH
0283 0D42 8529            STA BASH
0284 0D44 2CB20E          BIT PFLAG2     ; WHICH SCREEN?
0285 0D47 B021            BCS CONT1      ; SKIP SCREEN-2
0286 0D49 68              PLA
0287 0D4A 48              PHA
0288 0D4B 206F0D          JSR POUT1
0289 0D4E 209D0E          JSR LOADP1
0290 0D51 AD00C0          LDA KBD        ; LIST CONTROL?
0291 0D54 1014            BPL CONT1
0292 0D56 C983            CMP #$83       ; IF CTRL-C THEN PASS TO COUT
0293 0D58 F010            BEQ CONT1
0294 0D5A 8D10C0          STA KBDSTR
0295 0D5D C993            CMP #$93       ; IF CTRL-S THEN SET LISTOP FLAG
0296 0D5F D009            BNE CONT1
0297 0D61 EEAF0E          INC LISTOP
0298 0D64 2CB10E          BIT PFLAG1     ; TO SCREEN-1?
0299 0D67 B001            BCS CONT1
0300 0D69 60              RTS            ; NO OUTPUT TO SCREEN-1
0301 0D6A 68       CONT1  PLA
0302 0D6B 18              CLC
0303 0D6C 4CF0FD          JMP COUT1
0304 0D6F                                ;
0305 0D6F                                ;
0306 0D6F                                ;
0307 0D6F C9A0     POUT1  CMP #$A0
0308 0D71 9002            BCC COUTZ
0309 0D73 2532            AND INVFLG
0310 0D75 8435     COUTZ  STY YSAVI
0311 0D77 48              PHA
0312 0D78 20CD0D          JSR VIDOUT
0313 0D7B 68              PLA
0314 0D7C A435            LDY YSAVI
0315 0D7E 60              RTS
0316 0D7F                                ;
0317 0D7F                                ; [ BASCALC ROUTINE ]
0318 0D7F                                ;     BASE ADRESS FOR TEXT PAGE-2 ($8
0319 0D7F                                ;
0320 0D7F 48       BASCAL PHA             ; CALC BASE ADRESS IN BASL,H
0321 0D80 4A              LSR A          ;   FOR GIVEN LINE NO.
0322 0D81 2903            AND #$03
0323 0D83 0908            ORA #$08       ; *** PAGE-2 ***
0324 0D85 8529            STA BASH
0325 0D87 68              PLA
0326 0D88 2918            AND #$18
0327 0D8A 9002            BCC BSCLC2
0328 0D8C 697F            ADC #$7F
0329 0D8E 8528     BSCLC2 STA BASL
0330 0D90 0A              ASL A
```

```
LINE# LOC CODE   LINE

0331 0D91 0A             ASL A
0332 0D92 0528           ORA BASL
0333 0D94 8528           STA BASL
0334 0D96 60             RTS
0335 0D97                              ;
0336 0D97                              ; [ VIDOUT ROUTINE ]
0337 0D97                              ;     THIS CODING IS INDENTICAL TO THE
0338 0D97                              ;     ORIGINAL, HOWEVER RE-CODING IS
0339 0D97                              ;     DONE SINCE 'BASCALC' IS CALLED
0340 0D97                              ;     FROM MANY ROUTINES.
0341 0D97                              ;
0342 0D97 4CD9FB BELL1   JMP BELL
0343 0D9A                              ;
0344 0D9A A424   STOADV  LDY CH          ;CURSER H INDEX TO Y-REG
0345 0D9C 2CB00E         BIT PFLAGO      ; TEST OUTPUT MODE
0346 0D9F 3004           BMI EXT         ; AND CHANGE THE CHARACTER
0347 0DA1 500A           BVC NORMAL      ;   NORMAL MODE
0348 0DA3 700A           BVS ADVANC      ;   OVER WRITE MODE
0349 0DA5 5004   EXT     BVC HALF        ;    HALF INTENSITY
0350 0DA7 A920           LDA #$20        ;   HIGH-LIGHTING
0351 0DA9 D002           BNE NORMAL
0352 0DAB A9A0   HALF    LDA #$A0
0353 0DAD 9128   NORMAL  STA (BASL),Y
0354 0DAF E624   ADVANC  INC CH
0355 0DB1 A524           LDA CH
0356 0DB3 C521           CMP WNDWDT      ; BEYOND WINDOW WIDTH?
0357 0DB5 9003           BCC RTS3        ; YES, CR TO NEXT LINE
0358 0DB7 4C460E         JMP CR
0359 0DBA 60     RTS3    RTS              ; NO, RETURN
0360 0DBB                              ;
0361 0DBB A900   SETNOR  LDA #$00        ; MODE SETTING
0362 0DBD F00A           BEQ SETEND
0363 0DBF A940   SETOW   LDA #$40
0364 0DC1 D006           BNE SETEND
0365 0DC3 A980   SETHI   LDA #$80
0366 0DC5 D002           BNE SETEND
0367 0DC7 A9C0   SETHL   LDA #$C0
0368 0DC9 8DB00E SETEND  STA PFLAGO
0369 0DCC 60             RTS
0370 0DCD                              ;
0371 0DCD C9A0   VIDOUT  CMP #$A0        ; CONTROL CHAR?
0372 0DCF B0C9           BCS STOADV      ;  NO, OUTPUT IT
0373 0DD1 A8             TAY             ; INVERSE VIDEO?
0374 0DD2 10C6           BPL STOADV      ;  YES, OUTPUT IT.
0375 0DD4 CD270C         CMP CODENO      ; NORMAL? CTRL-N
0376 0DD7 F0E2           BEQ SETNOR
0377 0DD9 CD280C         CMP CODEOW      ; OVER WRITE? CTRL-O
0378 0DDC F0E1           BEQ SETOW
0379 0DDE CD290C         CMP CODEHI      ; HALF INTENSITY
0380 0DE1 F0E0           BEQ SETHI
0381 0DE3 CD2A0C         CMP CODEHL      ; HIGH LIGHTING
0382 0DE6 F0DF           BEQ SETHL
0383 0DE8 C98D           CMP #$8D        ; CR?
0384 0DEA F05A           BEQ CR          ;  YES.
0385 0DEC C98A           CMP #$8A        ; LF?
```



```
LINE# LOC CODE   LINE

0386 0DEE F05A           BEQ LF          ;  YES.
0387 0DF0 C988           CMP #$88        ; BS?
0388 0DF2 D0A3           BNE BELL1       ;  NO, CHECK FOR BELL.
0389 0DF4                              ;
0390 0DF4 C624   BS      DEC CH          ; DECREMENT CURSOR H INDEX
0391 0DF6 10C2           BPL RTS3        ; IF POS.OH ELSE MOVE UP.
0392 0DF8 A521           LDA WNDWDT      ; SET CH TO WNDWDT-1
0393 0DFA 8524           STA CH
0394 0DFC C624           DEC CH
0395 0DFE A522   UP       LDA WNDTOP      ; CURSOR V INDEX
0396 0E00 C525           CMP CV
0397 0E02 B00B           BCS RTS4        ; IF TOP LINE THEN RETURN
0398 0E04 C625           DEC CV
0399 0E06 A525   VTAB    LDA CV
0400 0E08 207F0D VTABZ   JSR BASCAL
0401 0E0B 6520           ADC WNDLFT      ; ADD WINDOW LEFT INDEX
0402 0E0D 8528           STA BASL        ; TO BASL
0403 0E0F 60     RTS4    RTS
0404 0E10                              ;
0405 0E10 49C0   ESC1    EOR #$C0        ; ESC?
0406 0E12 F028           BEQ HOME  IF SO, DO HOME AND CLEAR
0407 0E14 69FD           ADC #$FD ESC-A OR B CHECK
0408 0E16 9097           BCC ADVANC      ;   A:ADVANCE
0409 0E18 F0DA           BEQ BS          ;   B:BACK SPACE
0410 0E1A 69FD           ADC #$FD        ; ESC-C OR D CHECK
0411 0E1C 902C           BCC LF          ;   C:DOWN
0412 0E1E F0DE           BEQ UP          ;   D:GO UP
0413 0E20 69FD           ADC #$FD        ; ESC-E OR F CHECK
0414 0E22 905C           BCC CLREOL      ;   E:CLEAR TO END OF LINE
0415 0E24 D0E9           BNE RTS4        ;   NOT F, RETURN
0416 0E26 A424   CLREOP  LDY CH          ; CURSOR H TO Y INDEX
0417 0E28 A525           LDA CV          ; CURSOR V TO A-REG.
0418 0E2A 48     CLEOP1  PHA              ; SAVE CURRENT LINE TO STACK
0419 0E2B 20080E         JSR VTABZ       ; CALC BASE ADRESS
0420 0E2E 20820E         JSR CLEOLZ      ; CLEAR TO EOL,SET CARRY
0421 0E31 A000           LDY #$00        ; CLEAR FROM H INDEX=0 FOR REST
0422 0E33 68             PLA             ; INCREMENT CURRENT LINE
0423 0E34 6900           ADC #$00        ; (CARRY IS SET)
0424 0E36 C523           CMP WNDBTM      ; DONE TO BOTTOM OF WINDOW?
0425 0E38 90F0           BCC CLEOP1      ;   NO, KEEP CLEARING LINES
0426 0E3A B0CA           BCS VTAB        ;   YES, TAB TO CURRENT LINE
0427 0E3C A522   HOME    LDA WNDTOP      ; INIT CURSOR V
0428 0E3E 8525           STA CV          ;   AND H-INDICES
0429 0E40 A000           LDY #$00
0430 0E42 8424           STY CH          ; THEN CLEAR TO END OF PAGE
0431 0E44 F0E4           BEQ CLEOP1
0432 0E46 A900   CR       LDA #$00        ; CURSOR TO LEFT OF INDEX
0433 0E48 8524           STA CH
0434 0E4A E625   LF       INC CV          ; INCREMENT CURSOR V
0435 0E4C A525           LDA CV
0436 0E4E C523           CMP WNDBTM      ; OFF SCREEN?
0437 0E50 90B6           BCC VTABZ       ;   NO, SET BASE ADRESS
0438 0E52 C625           DEC CV          ; DECR CURSOR (BACK TO BOTTOM)
0439 0E54 A522   SCROLL  LDA WNDTOP      ; START AT TOP
0440 0E56 48             PHA
```

```
LINE# LOC CODE   LINE

0441 0E57 20080E        JSR VTABZ       ; GENERATE BASE ADRESS
0442 0E5A A528   SCRL1  LDA BASL        ; COPY BASL,H
0443 0E5C 852A          STA BAS2L       ; TO BAS2L,H
0444 0E5E A529          LDA BASH
0445 0E60 852B          STA BAS2H
0446 0E62 A421          LDY WNDWDT      ; INIT Y TO RIGHT MOST INDEX
0447 0E64 88            DEY             ;  OF SCROLLING WINDOW
0448 0E65 68            PLA
0449 0E66 6901          ADC #$01        ; INCR LINE NUMBER
0450 0E68 C523          CMP WNDBTM      ; DONE?
0451 0E6A B00D          BCS SCRL3       ;   YES, FINISH
0452 0E6C 48            PHA
0453 0E6D 20080E        JSR VTABZ       ; FORM BASL,H (BASE ADRESS)
0454 0E70 B128   SCRL2  LDA (BASL),Y    ; MOVE A CHR UP ONE LINE
0455 0E72 912A          STA (BAS2L),Y
0456 0E74 88            DEY
0457 0E75 10F9          BPL SCRL2
0458 0E77 30E1          BMI SCRL1       ; NEXT LINE
0459 0E79 A000   SCRL3  LDY #$00        ; CLEAR BOTTOM LINE
0460 0E7B 20820E        JSR CLEOLZ      ; GET BASE ADDR FOR BOTTOM
0461 0E7E B086          BCS VTAB        ; CARRY IS SET
0462 0E80 A424   CLREOL LDY CH          ; CURSOR H INDEX
0463 0E82 A9A0   CLEOLZ LDA #$A0
0464 0E84 9128   CLEOL2 STA (BASL),Y    ; STORE BLANKS
0465 0E86 C8            INY             ; HERE TO EOL
0466 0E87 C421          CPY WNDWDT
0467 0E89 90F9          BCC CLEOL2
0468 0E8B 60            RTS
0469 0E8C                               ;
0470 0E8C 8A     SAVEP1 TXA
0471 0E8D 48            PHA
0472 0E8E A200          LDX #$00
0473 0E90 B524   SA1    LDA CH,X
0474 0E92 9DB30E        STA PSAVE,X
0475 0E95 E8            INX
0476 0E96 E009          CPX #$09
0477 0E98 D0F6          BNE SA1
0478 0E9A 68            PLA
0479 0E9B AA            TAX
0480 0E9C 60            RTS
0481 0E9D                               ;
0482 0E9D 8A     LOADP1 TXA
0483 0E9E 48            PHA
0484 0E9F A200          LDX #$00
0485 0EA1 BDB30E LO1    LDA PSAVE,X
0486 0EA4 9524          STA CH,X
0487 0EA6 E8            INX
0488 0EA7 E009          CPX #$09
0489 0EA9 D0F6          BNE LO1
0490 0EAB 68            PLA
0491 0EAC AA            TAX
0492 0EAD 60            RTS
0493 0EAE                               ;
0494 0EAE 00     TEMPC  .BYT 0           ; TEMPORARY REGISTER
0495 0EAF 00     LISTOP .BYT 0           ; LIST STOP FLAG
```



```
LINE# LOC CODE   LINE

0496 0EB0                               ;
0497 0EB0 00     PFLAGO .BYT 0          ; OUTPUT MODE FLAGS
0498 0EB1                               ;-----------------------
0499 0EB1                               ;!B7 B6                 !
0500 0EB1                               ;! 0  0  NORMAL         !
0501 0EB1                               ;! 0  1  OVER WRITE     !
0502 0EB1                               ;! 1  0  HALF INTENSITY !
0503 0EB1                               ;! 1  1  HIGH-LIGHT     !
0504 0EB1                               ;-----------------------
0505 0EB1                               ;
0506 0EB1 00     PFLAG1 .BYT 0          ; TEXT PAGE-1 CONTROL
0507 0EB2                               ; B7=H TO SKIP PAGE-1
0508 0EB2                               ;
0509 0EB2 00     PFLAG2 .BYT 0          ; TEXT PAGE-2 CONTROL
0510 0EB3                               ; B7=H TO SKIP PAGE-2
0511 0EB3                               ;
0512 0EB3 00     PSAVE  .BYT 0,0,0,0,0,0,0,0 ; 8 BYTES STACK
0512 0EB4 00
0512 0EB5 00
0512 0EB6 00
0512 0EB7 00
0512 0EB8 00
0512 0EB9 00
0512 0EBA 00
0513 0EBB                               ;
0514 0EBB              .END

ERRORS = 0000 <0000>
```