HGR6
HGR6
HGR6
HGR6
HGR6
HGR6
HGR6
HGR6
HGR6
HGR6
HGR6
HGR6
HGR6

ALF

CONTENTS

A L F Products Inc.
Part Number: 11-1-10A

INTRODUCTION

DOUBLE-RES GRAPHICS

Your Apple //e personal computer, and its Applesoft graphics commands, let you make exciting color displays with ease. Besides the obvious entertainment value of vivid color images, practical applications abound. Detailed color charts and graphics bring complex data to life for instant comprehension. Unique character sets and special symbols allow technical subjects to be handled as easily on the display as they are on a blackboard. For business or pleasure, your Apple's graphics features take you beyond the limited world of letters and numbers displayed in simple columns, to a full universe of visual possibilities in shape, motion, and color.

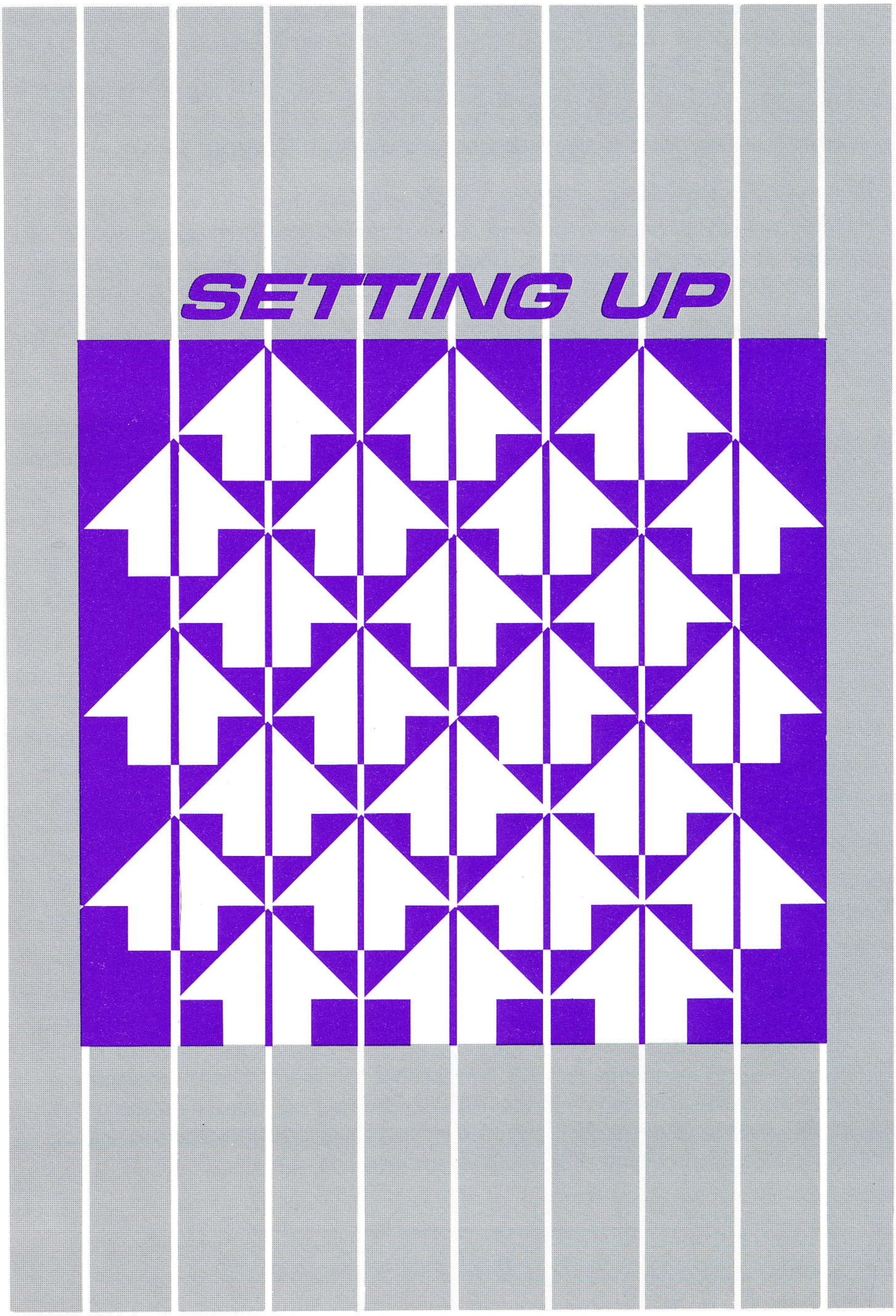
With the addition of Apple's Extended 80-Column Card (#A2B2023), this universe is expanded from the usual 280x192 hi-res display to the new 560x192 "double hi-res" display. The original hi-res colors: black, white, purple, light green, medium blue, and orange are augmented with the new double hi-res colors, adding: magenta, dark blue, dark green, gray, light blue, brown, grey, pink, yellow, and aquamarine. While the card itself adds the possibility of using these new features, the Applesoft graphics functions have no means of doing so.

ALF's "HGR6 Double-Res Graphics Package" supplies the missing software link. With the HGR6 package, you can use the new functions of the Extended 80-Column Card with your own graphics programs or by modifying existing Applesoft graphics programs. The HGR6 package works so smoothly with Applesoft that using it is a breeze. It's almost as if double-res was written into Applesoft from the start. Just skim through the explanation of the new features, and you can begin writing double-res graphics programs as easily as you used to write regular hi-res programs!

HGR6 also comes with a "slide show" program that lets you display images you've created. You can show hi-res and double-res images in any combination. And HGR6 comes with a "compress" function that reduces the amount of disk storage a stored image uses. The slide show program automatically "expands" compressed images so they can be viewed normally.

Your purchase of ALF's HGR6 package shows you have very discriminating taste in software. ALF has been in the computer business since 1975, and our goal of producing only the finest quality hardware and software is backed by our reputation for top quality, reliable products. The HGR6 package lets you access previously unavailable functions of the Apple to create anything from great works of art which will inspire awe in your friends and family, to sophisticated graphic reports which enligten business associates with their instant revelation of detailed information. Truly, your graphics experience is just beginning.

SETTING UP



JUMPER BLOCK

Double hi-res and lo-res are enabled with a jumper block (Apple #517-1230) plugged into your Extended 80-Column Card. To use the HGR6 package, you must have this jumper plugged into the pair of pins on the card (as described in the sheet supplied with the card).

The manual supplied with your Extended 80-Column Card discusses other requirements for using double hi-res, such as turning on "Annunciator 3". When you use the HGR6 package, you don't need to worry about any of this. If you have the jumper block plugged in, HGR6 will do the rest.

BRUN DOUBLERES

The HGR6 package is set up when you type BRUN DOUBLERES. This program copies the Applesoft language from ROM to the equivalent area of RAM. That is, it copies Applesoft from memory that can't be changed to memory that can be. Once Applesoft is in the (changeable) RAM area, various graphics routines are replaced with new routines that can handle double graphics. This changed Applesoft works pretty much like the usual Applesoft, but some new features are added to let you take advantage of double hi-res and double lo-res graphics. DOUBLERES can be used with either DOS 3.3 or ProDOS.

DOS 3.3

Once set up by running DOUBLERES, the HGR6 package can be used until it is erased or deselected. Booting a standard DOS 3.3 disk will erase the HGR6 pacakge, because DOS 3.3 erases the "language card" area (used by HGR6) while booting up. Typing INT will deselect HGR6, returning you to normal Applesoft (after LANGUAGE NOT AVAILABLE is printed); typing FP will select the HGR6 modified Applesoft again.

When you boot the DOS 3.3-version disk supplied, DOUBLERES is run before the menu is presented; so there's no need for you to run it again. The bootup program also runs FAST LOAD, which is a program that changes DOS 3.3 so LOAD and BLOAD commands will operate faster than they do in normal DOS 3.3. Note that to obtain enough memory to add the faster loading routines, FAST LOAD erases the INIT command. If you wish to use the INIT command, you must boot any standard DOS 3.3 disk to remove the FAST LOAD changes.

ProDOS

Once set up by running DOUBLERES, the HGR6 package can be used until you exit BASIC and enter another system. When you boot the

ProDOS-version disk supplied, DOUBLERES is run before the menu is presented; so there's no need for you to run it again. Note that since double resolution graphics uses the alternate 64K memory, the ramdisk "/RAM" disappears when you run DOUBLERES.

DOUBLE HI-RES

HGR3-HGR6

The HGR6 package is very easy to use. Using simple statements, such as HGR3, double resolution images can be made as easily as regular hi-res images. Here's a quick chart showing the various new HGR statements:

HGR	MIXED TEXT	PAGE	RESOLUTION	# OF COLORS	CLEARS TO	40/80 COLUMN
HGR	Yes	1	0-279	6	0 (black)	40 or 80
HGR2	No	2	0-279	6	0 (black)	40
HGR3	Yes	1	0-559	16	HCOLOR	80
HGR4	No	2	0-559	16	HCOLOR	40
HGR5	Yes	1	0-279	16	0 (black)	80
HGR6	No	2	0-279	16	0 (black)	40

HGR functions normally. Plotting is done on page 1, in normal hi-res, with 4 text lines at the bottom. HGR can be used from either 40- or 80-column mode.

HGR2 also functions normally. Plotting is done on page 2, in normal hi-res, with no text lines. HGR 2 should be used only from 40-column mode.

HGR3 gives you double resolution hi-res: X coordinates can range from 0 to 559 rather than from 0 to 279 as in normal hi-res. 16 colors can be used rather than the usual 6. Plotting is done on page 1, with 4 text lines at the bottom. HGR3 should be used only from 80-column text mode (that is, use PR#3 before using HGR3). Unlike HGR or HGR2, HGR3 clears the screen to the color currently selected by HCOLOR.

HGR4 gives you double resolution hi-res, the same as HGR3. Plotting is done on page 2, with no text lines. HGR4 should be used only from 40-column text mode.

HGR5 is a mixture of normal and double hi-res. Plotting is actually done using double resolution, so you can use 16 colors instead of the usual 6, but it is done twice as wide, reducing the X coordinate range down to the usual 0 to 279. This allows existing programs to use the new colors, without having to re-scale X coordinates, simply by changing HGR to HGR5 and using new HCOLOR values. Since the text cannot also be made twice as wide, some reformatting may be necessary if the 4 text lines are used. Plotting is done on page 1, with 4 text lines. HGR5 should be used only from 80-column text mode (use PR#3).

HGR6 is the same as HGR5, except it uses page 2 and has no text lines. HGR6 should be used only from 40-column text mode. Adapting programs which used HGR2 to 16-color plotting is easy, since there are no text lines to modify. Just change all HGR2's to HGR6's and change HCOLOR values where you wish.

Before using HGR3 or HGR5, you must be in 80-column mode. This is done with the statement:

PRINT CHR$(4)"PR#3"

Before using HGR2, HGR4, or HGR6, you must be in 40-column mode. Pressing control-RESET will activate 40-column mode. If you've used PR#3 since the last time you pressed control-RESET, you should add this statement to go back to 40-column mode:

PRINT CHR$(21)

HCOLOR

HCOLOR can be set to a variety of new values:

COLOR	HGR/HGR2	HGR3-HGR6	LO-RES (shown for comparison only)
Black	0 or 4	8, 0, or 4	0
Magenta		9	1
Dark Blue		10	2
Purple	2	11 or 2	3
Dark Green		12	4
Gray		13	5
Medium Blue	6	14 or 6	6
Light Blue		15	7
Brown		16	8
Orange	5	17 or 5	9
Grey		18	10
Pink		19	11
Light Green	1	20 or 1	12
Yellow		21	13
Aquamarine		22	14
White	3 or 7	23, 3, or 7	15

As you can see, colors like the ones available in lo-res can now be used in hi-res. Just add 8 to the usual lo-res color numbers when setting HCOLOR. (Eight must be added in order to skip over values 0-7, which are needed for compatibility with single resolution hi-res.)

When using double-res, you have more freedom in choosing colors. With HGR and HGR2, colors close to each other must often be in the HCOLOR range of 0-3 or 4-7. If you tried to plot HCOLOR 1 next to HCOLOR 5, you might get the same color. That's because the 0-279 X range of hi-res is divided into 40 blocks, each 7 dots wide, and each block can have colors 0-3 or 4-7 but not both. When using double-res with HGR3 through HGR6, this problem doesn't occur, and you can choose colors however you like. Another color restriction with HGR and HGR2 is that it really takes 2 dots to create a color, so you can't necessarily get the color you want where you want it. Double-res has this same restriction, except it takes up to 4 dots to create a color.

The HGR6 package works with all the plotting functions: HPLOT, DRAW, and XDRAW.

AUTOMATIC SET-UP

As mentioned in the previous section, before you can use the new features of the HGR6 package, you must type BRUN DOUBLERES. If you wish, your program can automatically set up double-res with the statement:

```
IF PEEK(62444) < >44 THEN PRINT CHR$(4)“BRUN
   DOUBLERES,A8192”
```

8192 is hi-res screen 1. If your program is so large that it extends into hi-res screen 1, a value larger than 8192 must be used or DOUBLERES will erase your program. Another common choice is 16384, which is hi-res screen 2. Any address where at least 3300 bytes of memory are available can be used, and can be erased after the program has run.

If you'll be using DOS 3.3 and doing any LOADing or BLOADing, you might rather use:

```
IF PEEK(62444) < > 44 THEN PRINT CHR$(4)“BRUN FAST
   LOAD,A8192”CHR$(13);CHR$(4)“BRUN DOUBLERES,A8192”
```

so you'll get FAST LOAD also (remember, FAST LOAD removes the INIT command). Like DOUBLE-RES, with FAST LOAD the 8192 address can be replaced with any address where at least 500 bytes of memory are available.

When using ProDOS, a simpler procedure than BRUN can be used: you can create a new BASIC.SYSTEM which includes DOUBLERES. Here's how:

```
BLOAD BASIC.SYSTEM,TSYS,A$2ØØØ
BLOAD DOUBLERES,A$48ØØ
POKE 8294,36
POKE 8295,72
UNLOCK BASIC.SYSTEM
BSAVE BASIC.SYSTEM,TSYS,A$2ØØØ,L$35ØØ
LOCK BASIC.SYSTEM
```

Using Apple's FILER, you can move this new BASIC.SYSTEM to any ProDOS disk. If it's the first system file in the catalog, DOUBLERES will automatically be set-up when you boot the disk. (Note that this

BASIC.SYSTEM won't boot on an Apple that doesn't have the Extended 80-Column Card.) If you accidentally BRUN DOUBLERES when you're using the BASIC.SYSTEM that includes DOUBLERES, there won't be any problem.

SAVING DOUBLE-RES IMAGES

You may wish to save a double-res image once you've created it. Obviously, you can't just use BSAVE and BLOAD as usual, since half the image is in main memory and the other half is in the Extended 80-Column Card's auxiliary memory (see your 80-Column Card manual). Once you've BRUN DOUBLERES to set up the HGR6 package, there are some handy routines to help you load and save:

TO MAIN is used to move 8K of memory from the auxiliary memory to main memory. Put the high byte of the auxiliary memory address into location 6, the high byte of the main memory address into location 7, then CALL 61142. This is useful for saving screens. To save page 1, you might use:

PRINT CHR$(4)"BSAVE FLOWERS.1,A8192,L8188" to save the main memory part

POKE 6,32 : POKE 7,32 : CALL 61142 to move the auxiliary memory part into main memory

PRINT CHR$(4)"BSAVE FLOWERS.2,A8192,L8188" to save the auxiliary part

(Adding .1 and .2 to the name help identify which part is which, and also allow the image to be loaded with the Slide Show program.) To save page 2 instead, you would change the 32's in the two POKE statements to 64's, and the two A8192's to A16384's (leave the two L8188's the same).

TO AUX is used to move 8K of memory from main memory to the auxiliary memory. Put the high byte of the main memory address into location 6, the high byte of the auxiliary memory address into location 7, then CALL 61140. This is useful for loading screens. To load page 1, you might use:

PRINT CHR$(4)"BLOAD FLOWERS.2,A8192" to load the auxiliary memory part into main memory

POKE 6,32 : POKE 7,32 : CALL 6114Ø to move it to auxiliary memory

PRINT CHR$(4)"BLOAD FLOWERS.1,A8192" to load the main memory part

As above, change the 32's to 64's and A8192's to A16384's to load onto page 2.

COMPRESS is used to squeeze a double hi-res image down in size, so it will consume less disk space. You put the high byte of the main memory address of 8K to be compressed (32 for page 1, or 64 for page 2) into location 6, and the high byte of the main memory address where the compressed image will be put in location 7, and CALL 63334. Usually, you just put 32 for page 1 or 64 for page 2 in both locations 6 and 7, thus compressing the image "in place". If you don't set locations 6 and 7 the same, you should have them different by 32 or more. After the compression is done, the new length is in locations 8 (low byte) and 9 (high byte). To save a compressed image, you use pretty much the same procedure as described above:

```
POKE 6,32 : POKE 7,32 : CALL 63334 : PRINT CHR$(4)"BSAVE
   FLOWERS.1,A8192,L" PEEK(8) + 256*PEEK(9) :
   CALL 61142 : CALL 63334 : PRINT CHR$(4)"BSAVE
   FLOWERS.2,A8192,L" PEEK (8) + 256*PEEK(9)
```

(Again, change the two 32's to 64's and the two A8192's to A16384's for page 2.) There's no need to set locations 6 and 7 between each call, as they are not changed by these routines.

EXPAND is used to undo the squishing done by COMPRESS. Put the high byte of the main memory address of the compressed image into location 7, the high byte of the main memory address of 8K where the original image will be reconstructed into location 6, and CALL 63203. To load a compressed image, use:

```
POKE 6,32 : POKE 7,32 : PRINT CHR$(4)"BLOAD
   FLOWERS.2,A8192" : CALL 632Ø3 : CALL 6114Ø : PRINT
   CHR$(4)"BLOAD FLOWERS.1,A8192" : CALL 632Ø3
```

As usual, if you wish to load into page 2 rather than page 1, replace the two 32's with 64's, and the two A8192's with A16384's.

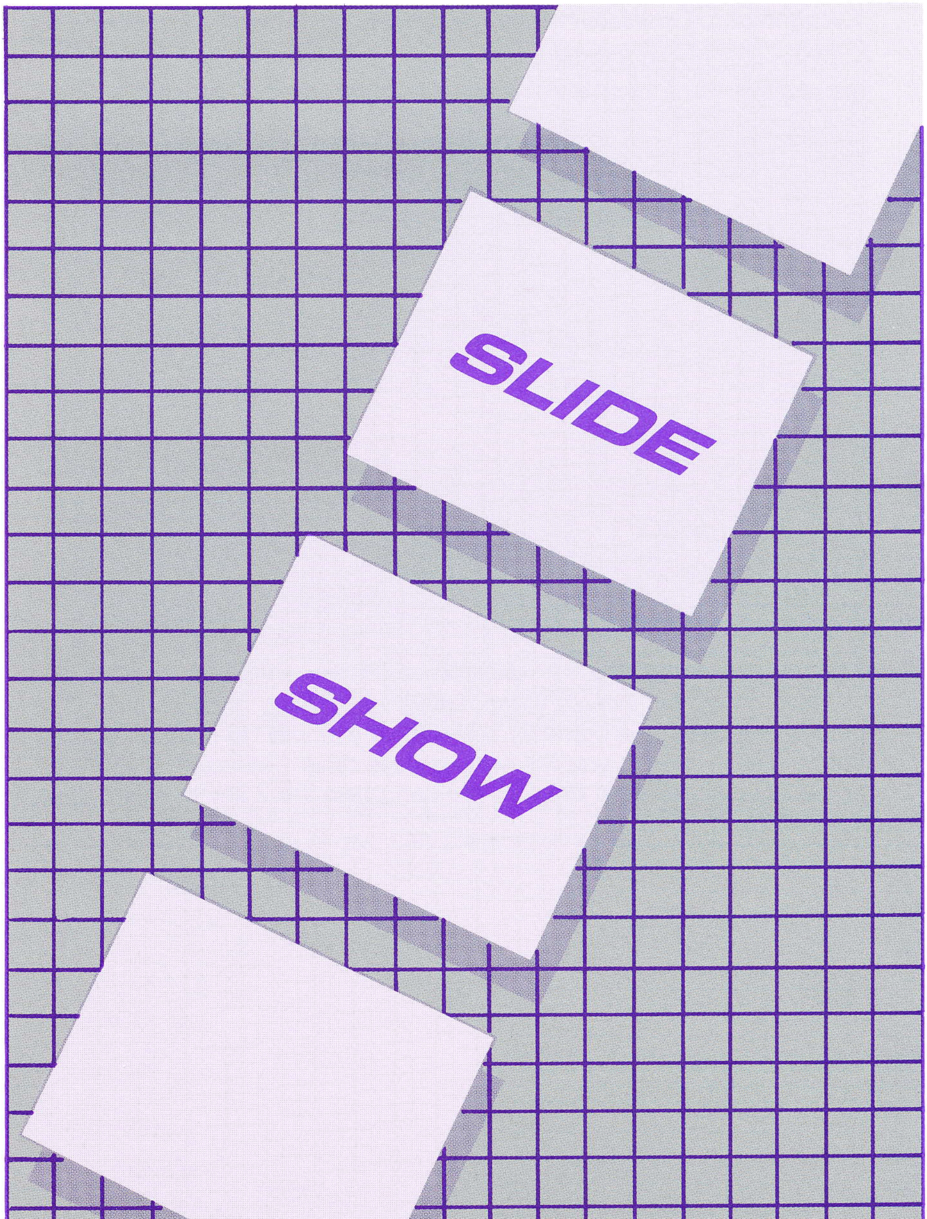
When an image is compressed and expanded, the most significant bit of each byte will be cleared. This doesn't matter when using double-res, since it doesn't use the most significant bit. If for some reason you wish to use COMPRESS and EXPAND on a normal hi-res image (which will not be very efficient, since they are designed for double hi-res), keep in mind that HCOLORs 4-7 will be converted to 0-3.

You can also use the program ***DOUBLESAVE*** to save a compressed double hi-res image. Once your image is on page 1 or page 2, type RUN DOUBLESAVE. It will ask "Screen name?". Type the name you want the image saved as (it will add .1 and .2 to the end of the name). Do not use ,D2 or other drive specifications. The program will then ask "Page 1 or 2?", and you should type the appropriate page number. Your image will then be compressed and saved. To view the image, you can use the Slide Show program.

If you wish to compress or expand an image, but you don't need the complete graphics package (that is, you don't want to BRUN

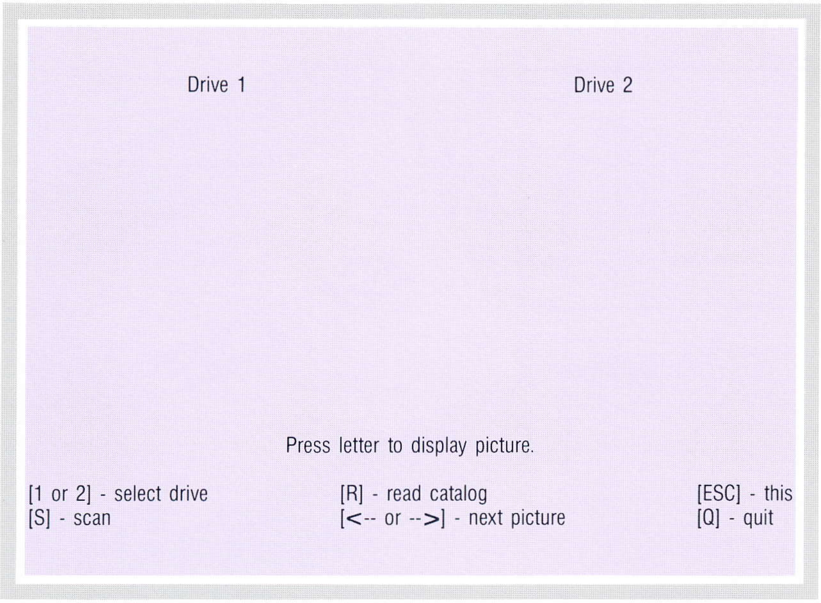
DOUBLERES), you can use the COMPRESS and EXPAND programs. Just set locations 6 and 7 as described above, and BRUN the appropriate program. You can use any address where 200 bytes are available for the "A" parameter in BRUN. Both programs use locations $DØ-$D5 for temporary storage.

If you wish to write your own program to expand a compressed image, you'll need to know the format of a compressed image. First of all, the last byte is always Ø. This identifies the file as being compressed with this algorithm, so values from 1 to 255 can identify other algorithms. The other bytes can be interpreted as follows: if the msb of a byte is Ø (that is, the byte is < 128), store that byte. (The first byte of the file would be stored at $2ØØØ, or $4ØØØ, if its msb is Ø.) If the msb is 1 (the byte is > 127), then think of the bits as: 1RRRABCD, and store RRR + 2 bytes: first store ØBCDABCD, then ØDABCDAB, then back to ØBCDABCD, and so forth. These repeated bytes (msb = 1) will not go across a screen "hole" (a hole is any address where bits 3 through 6 are all 1's). Process all bytes, skipping over "holes", until 768Ø bytes have been stored.


SLIDE
SHOW

USING SLIDESHOW

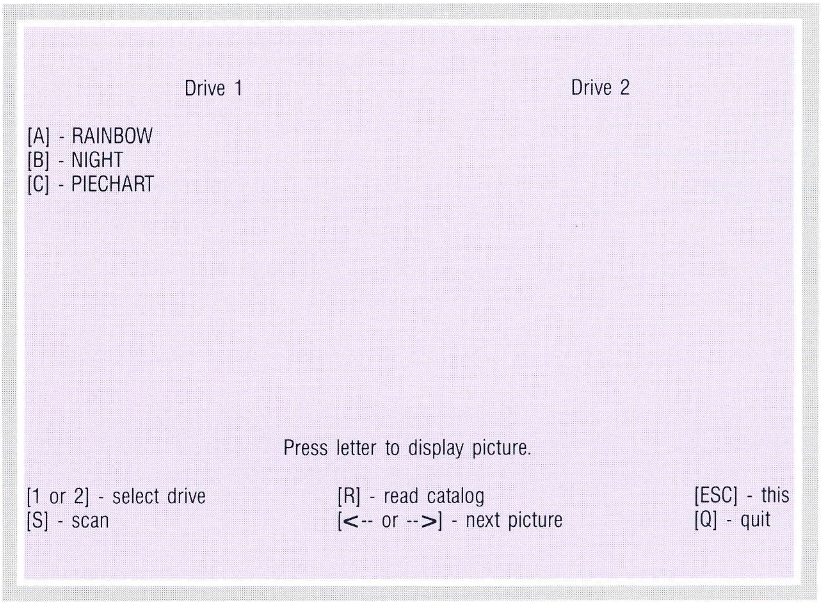
When you run SLIDESHOW, the screen shows:



```
                 Drive 1                                  Drive 2




                               Press letter to display picture.

[1 or 2] - select drive        [R] - read catalog                [ESC] - this
[S] - scan                     [<-- or -->] - next picture       [Q] - quit
```

"Drive 1" will be inverse (black letters on a white background). This means that drive 1 is currently selected. If you wish to display pictures from a disk in drive 1, you will want to leave drive 1 selected. If you wish to display pictures from drive 2, you can press the 2 key to select drive 2 instead.

Press the R key to read the catalog. The screen will show something like:

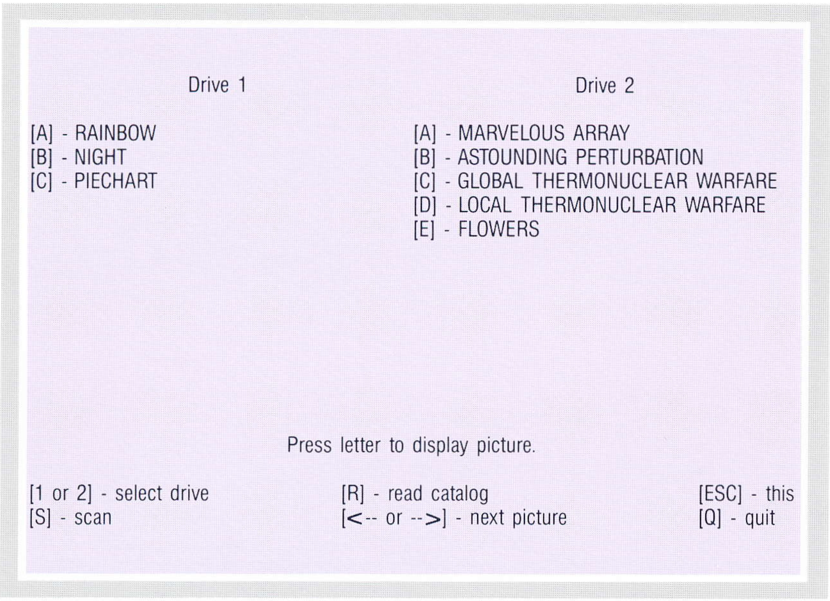


```
                 Drive 1                                      Drive 2

[A] - RAINBOW
[B] - NIGHT
[C] - PIECHART







                                Press letter to display picture.

[1 or 2] - select drive               [R] - read catalog                 [ESC] - this
[S] - scan                            [<-- or -->] - next picture        [Q] - quit
```

As you've probably guessed, pressing the A key will show the picture named RAINBOW, pressing the B key shows NIGHT, and so forth. The A through P keys are used to display one of up to 16 pictures stored on a disk.

When you're through viewing a picture, you can press the ESC key to go back to the menu, or you can press another A through P key to see another picture.

If you have another disk in drive 2, you could press the 2 key (to select drive 2) then the R key (to read drive 2's catalog). The screen might show:



```
                Drive 1                                      Drive 2

[A] - RAINBOW                              [A] - MARVELOUS ARRAY
[B] - NIGHT                                [B] - ASTOUNDING PERTURBATION
[C] - PIECHART                             [C] - GLOBAL THERMONUCLEAR WARFARE
                                           [D] - LOCAL THERMONUCLEAR WARFARE
                                           [E] - FLOWERS









                           Press letter to display picture.

[1 or 2] - select drive        [R] - read catalog                [ESC] - this
[S] - scan                     [<-- or -->] - next picture       [Q] - quit
```

Now pressing the A key will show MARVELOUS ARRAY instead of RAINBOW. To display RAINBOW, you would press 1 (to select drive 1) and then A.

FUNCTIONS

Here's a summary of all the functions:

[1 or 2] - select drive: pressing the 1 or 2 key selects the pictures from drive 1 or 2 for display. The drive selected will be indicated by Drive 1 or Drive 2 in inverse. If no titles appear below the inverse word, you must use the R function (below) before you can view the pictures.

[R] - read catalog: pressing the R key causes the catalog to be read. The titles of pictures available for display will appear on the screen. Each time you put in a new disk, you must press the R key to read in that disk's catalog.

[ESC] - this: press the ESC key when you're viewing a picture and wish to see the menu again.

[S] - scan: press the S key when you wish to scan through all pictures on a disk, viewing them as rapidly as possible. If one of the A through P letters on the screen is in inverse, viewing will begin with that picture (otherwise the scan begins with picture A). When the last picture is shown, scanning continues starting with picture A. To stop scanning, press ESC (or any other function key).

[<-- or -->] - next picture: press the right arrow (-- >) key when you wish to see the next picture. If one of the A through P letters on the screen is in inverse, pressing the -- > key will show the picture after the one indicated by the inverse letter (otherwise, picture A will be shown). When you press -- >, the SLIDESHOW program will also load in the next picture but not show it until you press -- > again. Pressing the solid Apple key, or paddle button 1, is the same as pressing -- >. The left arrow key (< --) has the same function as -- >, except it shows the previous picture. Pressing open Apple, or paddle button 0, is the same as pressing < -- .

[Q] - quit: press the Q key when you wish to stop using the SLIDESHOW program. The SLIDESHOW program stops and the Applesoft prompt appears. If you wish to go back to the boot-up menu, RUN STARTUP (return).

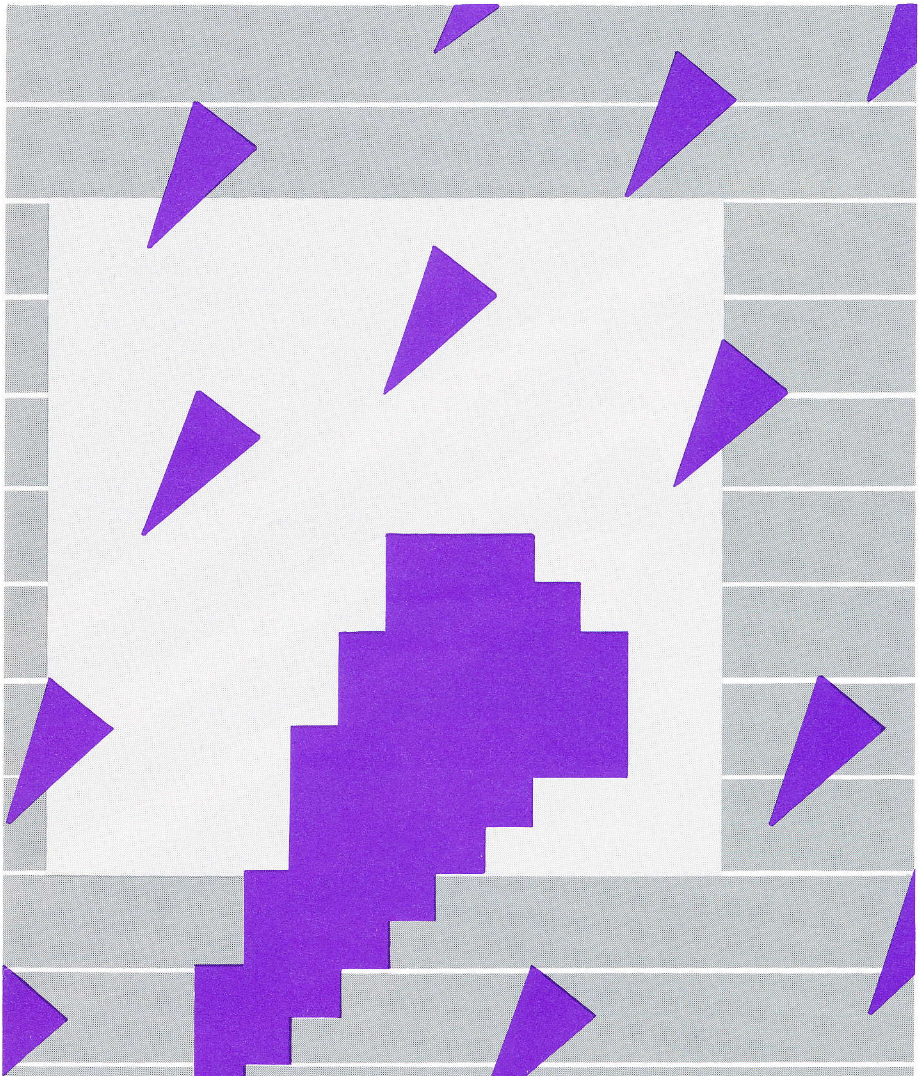
NAMING YOUR PICTURES

The SLIDESHOW program displays regular hi-res pictures and double hi-res pictures interchangeably, and automatically "expands" any pictures that were stored "compressed".

When you save pictures you wish to view with SLIDESHOW, you must give them special names so the SLIDESHOW program will recognize them as pictures.

TYPE OF PICTURE	ADD TO NAME:	SAMPLE NAME
Hi-res	.Ø	FLOWERS.Ø
Double-res main memory part	.1	FLOWERS.1
Double-res auxiliary part	.2	FLOWERS.2

SLIDESHOW automatically determines whether or not the picture is compressed. Non-compressed pictures must have a length greater than 7935 bytes (use L8188 when BSAVEing), and compressed pictures, of course, will have a smaller length.



DOUBLE LO-RES

DOUBLE LO-RES

The HGR6 Double-Res Graphics Package also lets you use lo-res graphics with a 80 x 48 display rather than the usual 40 x 48. All the usual functions (PLOT, HLIN, VLIN, and SCRN) can be used.

To use double lo-res, simply replace the usual GR statement with:

```
PRINT CHR$(4)"PR#3" : GR2
```

The text area will be 4 lines of 80 columns each (rather than the usual 4 lines of 40 columns). If you wish to have an 80 x 48 display, with no text lines, add:

```
POKE 49234,Ø
```

as usual. The lower portion of the screen (40 to 47) will not be cleared to black, just like it isn't when you use GR : POKE 49234,Ø.

The COLOR= statement is used in the same fashion as with GR.

SPEED

SPEED

Plotting normal hi-res images requires the manipulation of 7680 bytes of memory, which control the visible part of the screen. Double hi-res images use 15,360 bytes in the visible part. Obviously, when your program is working with twice as many bytes, it will run twice as slow.

Fortunately, most Applesoft programs are written in a straightforward fashion, and some speed can be gained by rewriting them in obscure fashions. For example, often-called routines (which are usually written last and thus placed at the end of the program) can be moved to the beginning of the program. Variables which are often used can be defined early in the program, too. These techniques allow the routines and variables to be found faster when Applesoft "searches" for them. By experimenting, you can find other ways to speed up programs.

If you don't want to rewrite your programs, you can buy ALF's AD8088 Processor Card. The Processor Card comes with a program called FTL (Formula Transfer Link) which allows most math functions in your Applesoft program to be computed using the Processor Card's fast 8088 processor. Just like DOUBLERES replaces Applesoft's normal graphics routines with routines that can handle double hi-res, FTL replaces math routines with calls to 8088 math functions. Most math functions in your program will run faster when you use FTL and the Processor Card.

For complete information on the Processor Card and FTL, see your local computer dealer, or write to ALF.

CREDITS

HGR6 Double-Res Graphics Package by John Ridges
except . . .

Demo programs by Bill Fickas
Compress/expand programs by John Ives
Manual written by Phil Tubb
Manual graphics by Karen Kokjer
Index by Greg Bloom

INDEX

A L F Products Inc.
1315F Nelson St.
Denver, CO 80215

[303] 234-0871
Telex: 4991824