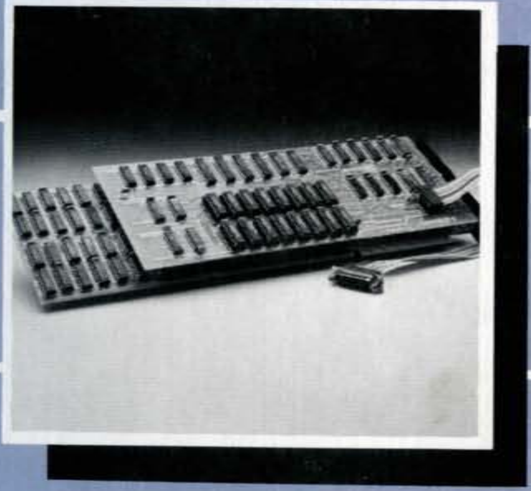
QUADRAM
CORPORATION
An Intelligent Systems Company

4355 International Blvd. / Norcross, Ga. 30093 / (404) 923-6666
TWX 810-766-4915 (QUADRAM NCRS)

PART NO. 9530-1

Quadcolor™ II Operation Manual



QUADRAM
CORPORATION
An Intelligent Systems Company

Quadcolor™ II Operation Manual

Quadcolor II Operations Manual

This operations manual applies to Quadram's Quadcolor II Color Graphics Adapter Enhancement card (version ±1).

Before attempting to install and use Quadram's Quadcolor II, read this operations manual thoroughly. If for any reason you should experience difficulty in the operation of this board contact your local authorized Quadram dealer for assistance.

CONTENTS Page

WARRANTY PROVISIONS

The Quadram Corporation warrants this product to be in good working order for a period of one year from the date of purchase as a new product. Shoulo this product fail to perform properly any time within that one year period. Quadram Corporation will, at its option, repair or replace this product at no cost except as set forth in this warranty. Replacement parts or products will be furnished on an exchange basis only. Replaced parts and/or products become the property of Quadram Corporation. No warranty is expressed or implied for products damaged by accident, abuse, misuse, natural or personal disaster, or unauthorized modification.

Warranty service described herein may be obtained by delivering this product, during the one year warranty period, to any authorized Quadram Dealer or to Quadram's Factory Service Center located at 4355 Intl. Blvd., Norcross, Georgia, 30093. Whether delivering to a Dealer or to the Factory, product must be accompanied by proof of purchase date. If shipped by U.S. Mail or any common carrier, owner agrees to insure and accept all liability for loss or damage to this product, to prepay all shipping charges, and to use shipping container equivalent to original packaging.

WARRANTY LIMITATIONS

All warranties for this product expressed or implied, including merchantability and fitness for a purpose, are limited to a one year duration from date of purchase, and no warranties, expressed or implied, will apply after that period.

If this product does not perform as warranted herein, owner's sole remedy shall be repair or replacement as provided above. In no event will Quadram Corporation be liable to any purchaser for damages, lost revenue, lost wages, lost savings, or any other incidental or consequential damages arising from purchase, use or inability to use this product, even if Quadram has been advised of the possibility of such damages.

Periodic updates or changes to this material will be made in new editions of this publication.

Requests for dealer information, comments or other inquiries may be addressed to: Quadram Corporation 4355 Intl. Blvd., Norcross, Georgia 30093. Quadram Corporation has no obligation to return or hold confidential any material or information you supply unless arrangements are made pursuant to Quadram's receipt of said information.

INTRODUCTION

NOTE: Quadram's Quadcolor II requires a PC system operating with at least 96K of RAM. If your system does not contain this minimum amount you will need to upgrade your computer. You can perform this upgrade yourself or you can contact your local computer retailer for assistance. Also, Quadcolor II software operates only under DOS 2.0

Quadcolor II

Quadram's Quadcolor II is an enhancement board for the Quadcolor I, Quadram's color graphics adapter board. Quadram's Quadcolor II combines with Quadcolor I to give your PC system two additional screen modes as well as extended color capabilities. And Quadcolor software allows you to use these enhancements in all your graphics programming work.

With Quadcolor II you are able to put 136 different colors on the screen at once in medium resolution mode (320 x 200) and up to 16 different colors at once in high resolution mode (640 x 200). Quadcolor II comes with complete software to enhance IBM BASICA so that the common graphics commands adapt to the wider parameters offered by Quadcolor II. Quadcolor II also contains an IBM compatible game port for your use.

The Quadcolor II Package

When you bring Quadcolor II home from the store, the first thing you will want to do is to make sure all the proper pieces to the package are there. The Quadcolor II package contains three items:

1. The Quadcolor II circuit board with a game port cable attached,
2. a floppy disk containing Quadcolor Software, and
3. this operations manual.

Make sure all these pieces are present. If anything is missing contact your local authorized Quadram dealer.

Backing up Quadcolor Software

Before you do anything else with your Quadcolor II package, it is advisable to make a back-up copy of the Quadcolor II software.This will ensure that no mishap — accidental erasure, disk failure, etc. — robs you of your use of the Quadcolor programs.

To make a copy of the Quadcolor software, procede as follows:

NOTE: The following is written for a one drive system. Two drive systems only require that the disk be inserted into the B drive whenever the computer calls for it.

---

1. Boot your PC system with DOS 2.0 by putting the DOS disk in Drive A and turning on the computer. (Quadcolor II is designed to work only under DOS 2.0.)
2. When the DOS A> appears type:

   DISKCOPY<RETURN>

3. The computer will prompt you for a disk change now. Take the DOS disk out of Drive A and put in the Quadcolor software disk.
4. Press any key on the keyboard. When you do so the computer will copy the Quadcolor software into memory.
5. When the drive stops, take out the Quadcolor disk and insert a blank disk in Drive A. (The disk can be put in Drive B for two drive systems.) The computer will call this the TARGET disk.
6. Press any key.

The computer will begin copying the Quadcolor information onto the blank disk, formatting it as it goes. Depending on the type of disk drive your system uses, the computer may prompt you to repeat steps 3 and 4. The computer will let you know when the copying process is complete.

With the Quadcolor software safely copied, you can put away the Original disk in a safe place, taking it out only when you need to make additional copies of the disk information.

INSTALLATION

Hardware Installation

With the Quadcolor software safely copied you can now prepare to install the Quadcolor II into your Personal Computer system.

If You Already Have Quadcolor I In Your PC System

If your PC system is already operating with Quadcolor I color graphics card, you will have to remove it from your computer before Quadcolor II can be installed. Quadcolor II must be snapped onto Quadcolor I and this cannot be done with the Quadcolor I in one of the expansion slots. For information on removing Quadcolor I from your IBM PC consult Quadram's "Quadcolor I Operations Manual," or the IBM "Guide to Operations" Options section.

If You Have Purchased Quadcolor II with Quadcolor I

If your purchased Quadcolor II and Quadcolor I together you will, of course, not need to get into the PC yet. But before you begin to use Quadcolor II Quadram recommends you first familarize yourself with the operation of Quadcolor I. Only after you appreciate the capabilities and functions of this card can you begin to fully enjoy the use of the Quadcolor II enhancement.

Joining Quadcolor II to Quadcolor I

When you are ready to use Quadcolor II you will need to attach it to the Quadcolor I card. This is done by matching two mating connectors on each board.

Quadcolor I Connectors

If you look at the top side — the component side — of the Quadcolor I board you will notice two male connectors each consisting of two rows of pins The first connector consists of two rows of thirteen pins. Adjacent to these pins should be four additional pins with four jumper shunts placed over them. These shunts ground the IRGB input (the color signals) coming from the Quadcolor II and must be removed before Quadcolor II can be installed.

Remove the jumper shunts on Quadcolor I now.

The second male connector consists of two rows of 17 pins. This connector runs perpendicular to the board's end plate and is located near the nine-pin video output connector.

Quadcolor II Connectors

On the back of the Quadcolor II board are the mating connectors for those found on the Quadcolor I. Again, one is a 26 pin female connector and the other is a 34 pin female connector.

To attach Quadcolor II to Quadcolor I, mate the connectors in the following manner:

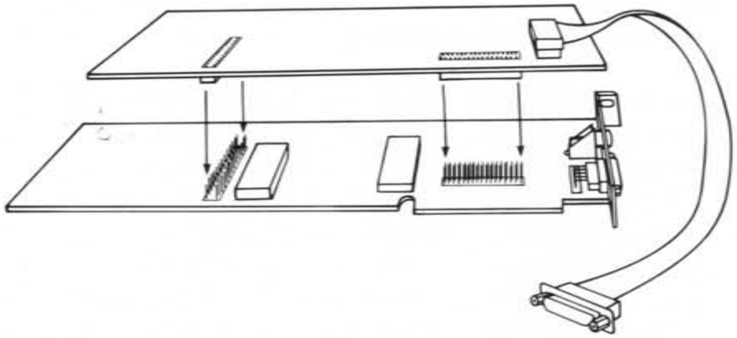


Fig. 1: Joining Quadcolor II to Quadcolor I

Again, make sure the jumper shunts have been removed. (For the proper location of these jumpers on the Quadcolor I board, see the Quadcolor I Operations Manual.)

Once the boards have been properly linked, Quadcolor can be installed inside your IBM PC. For installation instructions for installing Quadram's Quadcolor inside your PC, consult your Quadcolor I Operations Manual .

Software Installation

A special software disk accompanies your Quadcolor II package. This disk contains two important files that allow you to take advantage of Quadcolor's many capabilities in your programming work. These two files are listed on the disk as follows:

INSTALL.COM
QCI.STD

INSTALL.COM copies the Quadcolor driver software to your DOS disk and enables automatic installation of the driver on future boot-ups. It also creates an enhanced version of BASIC called BASICQ. BASICQ contains a sophisticated version of the BASIC graphic commands, a version which can take advantage of the added screen modes, the increased resolution, and the expanded color selection provided by Quadcolor.

QCI.STD is the standard Quadcolor driver software. INSTALL.COM copies this driver onto your DOS disk under the name QCI.COM. Before BASICQ can be run, you must first execute QCI.COM. If you copy QCI to your disk using INSTALL, then INSTALL automatically creates the QCI command in your AUTOEXEC.BAT file (if you have one on your disk) so the QCI runs whenever you boot-up.

Getting Into BASICQ

NOTE: The following is performed only when you wish to create BASICQ on disk for the first time. After this operation is completed, you will not need to repeat the process.

When the Quadcolor boards are properly installed in your PC you can install BASICQ in your system. To do this, follow the steps below. (NOTE: The computer performs this software installation through the use of the default drive, so the following steps apply for one, two, or more drive systems.)

1. Boot your system with a DOS 2.0 disk so that the DOS A> prompt appears on the screen.

2. Take the IBM DOS disk out of Drive A and insert the Quadcolor software disk.

3. Type:

   ```
   INSTALL<RETURN>
   ```

4. The disk in the drive will spin and the screen will display the following options menu.

5. Select option A by typing A:

   ```
   A
   ```

6. The screen will then prompt you to insert the distribution disk. This is already in the drive so all you need do is press the space bar.

7. When the disk stops spinning the computer will prompt you to insert a system disk into Drive A.

   This system disk should be the disk you will use on a day-to-day basis to boot Quadcolor. It will contain the Quadcolor driver software and, ultimately, BASICQ. Put your system disk in Drive A and press the space bar.

8. The computer will return you to the main menu to select the next option.

   Type B:

   ```
   B
   ```

9. At this, the computer will ask you to insert the disk containing BASICA.COM. Put this disk — a DOS 2.0 Master disk will do — into Drive A and press the space bar.

   When the disk in the drive stops spinning, BASICQ will have been created in system memory.

10. Next the computer will ask you insert the disk to receive BASICQ. Put your system disk in Drive A and press the space bar.

11. When the disk stops spinning, you'll be returned to the main menu. Now choose option C by typing in a C:

    C

12. Now the computer will ask you insert the disk you wish to boot. In all probability this will be the disk you already have in the drive: your system disk, the disk containing the driver software and BASICQ. So leave this in Drive A and just press the space bar.

13. The A>will appear. When it does, type:

    BASICQ<RETURN>

    The screen will scroll to what appears to be the normal BASIC screen. Only now the ability to take full advantage of Quadcolor is at your disposal.

    From now on, whenever you boot your system disk you'll only have to type in BASICQ<RETURN> after the A> (Because QCI automatically installs itself in an AUTOEXEC.BAT file it puts on the disk).

FEATURES AND COMMANDS

Working with Quadcolor II

Quadcolor II gives you two additional screen modes to use in your graphics programming work: screens 3 and 4. Screen 3 has a resolution of 320 horizontal pixels by 200 vertical pixels. In this mode there are 136 different colors which can be placed on the screen at one time. Ther are no palettes to choose from to restrict your color selection.

Screen 4 has a resolution of 640 horizontal pixels by 200 vertical pixels. You can place sixteen different colors on the screen at once in this mode. Again, there are no palettes to restrict color selection.

Using the SCREEN Statement

Getting into the two Quadcolor II screen modes is easy. All you have to do is use the standard SCREEN statement used in regular BASICA. Now however you can enter:

```
SCREEN 3 or
SCREEN 4
```

The screen syntax remains the same, except that WIDTH in modes 3 and 4 is set at 80 and cannot be changed. You should remember also that, as in IBM BASICA, when you change screen modes using the SCREEN statement you will erase any screen image you leave as you enter a new screen mode. For instance, if you're operating in SCREEN 0 and issue a SCREEN 4 command the text stored in SCREEN 0 will automatically be erased.

BASICQ also implements two other SCREEN statements: SCREEN TO and SCREEN ON/OFF. Both these commands are possible due to the hardware configuration of the Quadcolor boards.

Quadcolor I gives you three standard screen modes: SCREEN 0, 1, and 2. The images for each of these screens is stored in the Quadcolor I's display memory, 32K of RAM. Quadcolor II gives you two additional screen modes: SCREEN 3 and 4. The images for these two screens is stored in the Quadcolor II's display memory, 64K of RAM.

Because these memories do not overlap, you can use these two enhanced SCREEN statements to switch back and forth between the two banks to see what each contains or to write to them. Also, because each generates a display independently, the images from both can be displayed on the screen simultaneously, allowing the capability of graphic overlaying for sophisticated display tecniques.

Using the ",1" Option in the Screen Statement

The ",1" option is a feature that gives you a programming advantage when you are using Quadcolor I and II in your programming work. When you issue a SCREEN statement for modes 3 or 4, you have the option of adding a ",1" after the screen mode number, as in

```
SCREEN 3,1<RETURN>
        or
SCREEN 4,1<RETURN>
```

(Issuing SCREEN 3,0 or SCREEN 4,0 is the same as issuing a SCREEN 3 or SCREEN 4.)

The ",1" option tells the system that mode 3 or 4 (whichever was specified) will be used for all graphics work and the last previously selected mode — whichever it may have been — will be used for all text. This will come in handy in your work when you want to switch back and forth between your program code and program results.

EXAMPLE

```
    Screen 0 'Select normal text mode
    Screen 4,1 'Select graphics to go on
                 Quadcolor II
                 'mode 4 but text to go on
                   Quadcolor I
                 'mode 0
```

The convenience is this: all subsequent PRINT statements will automatically put the text on Quadcolor I Screen 0 (in this example) and all subsequent graphics statements will automatically put the data on Quadcolor II Screen 4 (in this example). This keeps you from having to continually select the Quadcolor display with the SCREEN Statement.

SCREEN TO

The SCREEN TO statement is used when you wish to move back and forth between the Quadcolor screen modes. The syntax for this statement looks like:

```
SCREEN TO n
```

with n being the screen value you wish to switch to.

For example, if you are working in SCREEN 1 and you wish to work in SCREEN 3, you should enter the following:

```
SCREEN TO 3<RETURN>
```

When you press the RETURN the screen cursor will vanish and the system will now write to the undisplayed SCREEN 3 residing in Quadcolor II's memory bank. SCREEN 1 will remain intact on the display screen and in Quadcolor I's display memory. You can return the cursor and the ability to write to SCREEN 1 by entering:

```
SCREEN TO 1<RETURN>
```

(But this command will not appear on the screen as you type it.)

By using the SCREEN TO statement, you can switch back and forth between the Quadcolor screen modes as long as you wish without clearing the display memory of either board. However, you should keep in mind that when you begin to switch back and forth between Quadcolor I display modes (SCREENS 0, 1, and 2) you risk losing information.

As an example, if you work in SCREEN 1 for a while then issue a SCREEN TO 0 statement the display will contain a good deal of "garbage". The only way to clear this is to use the BASICQ

```
CLS
```

comn.and. This will clear the Quadcolor I memory.

The same is true if you're working in SCREEN 2 and issue a SCREEN TO 1 command. CLS will have to be used to clear the screen for work.

In the Quadcolor II display bank this restriction does not exist. Using SCREEN TO, you may switch back and forth between modes 3 and 4 without the risk of losing any information.

For instance, if you begin writing to SCREEN 3, then issue a SCREEN TO 4 statement the SCREEN 3 image will not be affected. Rather it will appear on the SCREEN 4 display and you can edit this image under the restrictions of SCREEN 4 (640 x 200; 16 color capability, etc.)

The ",1" option can also be used in the SCREEN TO statement. For instance SCREEN TO 3,1 indicates that SCREEN 3 will be used for all graphics work while the mode you are leaving will be used for all text (unless you happen to be in SCREEN 3, then the last previous mode will be used for text.)

You will wish to practice with the SCREEN TO command to get used to its function.

SCREEN ON/OFF

The following commands can be used in much the same way as SCREEN ON/OFF:

<ALT 1> toggles Quadcolor I on and off;
<ALT 2> toggles Quadcolor II on and off.

We stated earlier that the Quadcolor II screen mode images are stored in display memory distinct from the display memory holding the Quadcolor I screen mode images. This means two independent screen images co-exist in the Quadcolor boards. Couple this with the fact that each Quadcolor board has its own IRGB outputs and you end up with the ability to overlay screen images.

The commands SCREEN ON and SCREEN OFF are used to overlay images from Quadcolor I and II (such as text from SCREEN 0 over a graph from SCREEN 3). These two commands can only be used to overlay images stored in different banks. They cannot overlay images stored in the same bank. For instance you cannot overlay a SCREEN 0 image with a SCREEN 2 image.

The syntax for SCREEN ON and SCREEN OFF is:

```
SCREEN ON n<RETURN>

SCREEN OFF n<RETURN>
```

with n being a 1 or a 2. The 1 represents the display from Quadcolor I and the 2 represents the display from Quadcolor II. For example, when you are working in modes 0,1, or 2, SCREEN ON 2<RETURN> executes the overlay of Quadcolor I's display with Quadcolor IIs display: SCREEN OFF 2<RETURN> turns off the overlay without erasing anything.

For example, if you were working in SCREEN 1 (Quadcolor I's display bank) and you issued a

```
SCREEN ON 2<RETURN>
```

you could then issue a

```
SCREEN OFF 1<RETURN>
```

The system would be left ready to write to SCREEN 1, but the Quadcolor II's display image would remain display on the monitor screen.

To write to this display you must enter either:

SCREEN TO 3<RETURN>

or

SCREEN TO 4<RETURN>

If not the system will continue writing to the undisplayed SCREEN 1 image.

Using SCREEN ON/OFF and SCREEN TO Together

As you begin to work with Quadcolor II and BASICQ you will notice that SCREEN ON/OFF and SCREEN TO are often used together to allow you to select the visual screen mode you wish to work with. The two commands complement each other in that SCREEN ON/OFF controls which display bank appears on the monitor screen but does not effect the ability of the system to write to that screen, while SCREEN TO controls which screen the system writes to but has no effect on which Quadcolor memory bank is displayed.

The following example will help you better understand the manner in which these two screen statements work together.

1. *With BASICQ loaded get into Screen 1 by typing SCREEN 1 <RETURN>.*

2. *Enter this short program to draw a white box on the screen:*

   ```
   10Line (100,50)–(220,150),7,B
   ```

3. *RUN the program. A box will be drawn with white lines.*

4. *Type SCREEN TO 4<RETURN>*

   The cursor vanishes as the computer is now ready to write to the undisplayed Screen 4.

5. *Though you can't see it on the screen, type the following short program:*

   ```
   10 Locate 12,39
   20 Print "Box
   ```

6. *RUN the program*

7. *Now type SCREEN ON 2<RETURN>*

   The word box will appear in the middle of the graphic box as Screen 4 is overlayed with Screen 1.

8. *Now type SCREEN OFF 1<RETURN>*

The graphic box will vanish (along with the Screen 1 text) since Quadcolor I's memory bank is turned off. The computer displays subsequent text to the displayed Screen 4. We can again write to Screen 1 by typing: SCREEN TO 1<RETURN>.

NOTE: Do not type SCREEN TO 0 or SCREEN TO 2. This will result in a screen full of garbage and potentially lost information when the display is activated. Always remember to return to the Quadcolor I screen mode you previously left when leaving a Quadcolor II screen mode.

Using The Extended Color Selection

When you combine BASICQ with Quadcolor II you are able to program graphics that contain a very wide color selection. In screen mode 3 (320 x 200) there are 136 colors to choose from. If you wish you can put all these on the screen at once. There are no palettes or color restrictions to worry about.

In screen mode 4 (640 x 200) there are 16 colors available to you. Again you can put all 16 on the screen at once if you so choose. The following two tables detail the exact colors which Quadcolor II can create. It also shows you the hex notation for each color value, the decimal notation, and the color components of each color value.

Refer to these charts when you wish to specify certain colors in your graphics work.

SCREEN 4 COLOR VALUES (640 x 200)

Hex	Decimal	Color
0	0	Black
1	1	Blue
2	2	Green
3	3	Cyan
4	4	Red
5	5	Magenta
6	6	Brown
7	7	White
8	8	Gray
9	9	Light Blue
A	10	Light Green
B	11	Light Cyan
C	12	Light Red
D	13	Light Magenta
E	14	Yellow
F	15	High Intensity

SCREEN 3 COLOR VALUES (320 x 200)

LEAST SIGNIFICANT

MOST SIGNIFICANT	Black	Blue	Green	Cyan	Red	Magenta	Brown	White	Gray	Light Blue	Light Green	Light Cyan	Light Red	Light Magenta	Yellow	H.I. White
Black	0 0H	1 1H	2 2H	3 3H	4 4H	5 5H	6 6H	7 7H	8 8H	9 9H	10 AH	11 BH	12 CH	13 DH	14 EH	15 FH
Blue	16 10H	17 11H	18 12H	19 13H	20 14H	21 15H	22 16H	23 17H	24 18H	25 19H	26 1AH	27 1BH	28 1CH	29 1DH	30 1EH	31 1FH
Green	12 20H	33 21H	34 22H	35 23H	36 24H	37 25H	38 26H	39 27H	40 28H	41 29H	42 2AH	43 2BH	44 2CH	45 2DH	46 2EH	47 2FH
Cyan	48 30H	49 31H	50 32H	51 33H	52 34H	53 35H	54 36H	55 37H	56 38H	57 39H	58 3AH	59 3BH	60 3CH	61 3DH	62 3EH	63 3FH
Red	64 40H	65 41H	66 42H	67 43H	68 44H	69 45H	70 46H	71 47H	72 48H	73 49H	74 4AH	75 4BH	76 4CH	77 4DH	78 4EH	79 4FH
Magenta	80 50H	81 51H	82 52H	83 53H	84 54H	85 55H	86 56H	87 57H	88 58H	89 59H	90 5AH	91 5BH	92 5CH	93 5DH	94 5EH	95 5FH
Brown	96 60H	97 61H	98 62H	99 63H	100 64H	101 65H	102 66H	103 67H	104 68H	105 69H	106 70H	107 6BH	108 6CH	109 6DH	110 6EH	111 6FH
White	112 70H	113 71H	114 72H	115 73H	116 74H	117 75H	118 76H	119 77H	120 78H	121 79H	122 7AH	123 7BH	124 7CH	125 7DH	126 7EH	127 7FH
Gray	128 80H	129 81H	130 82H	131 83H	132 84H	133 85H	134 86H	135 87H	136 88H	137 89H	138 8AH	139 8BH	140 8CH	141 8DH	142 8EH	143 8FH
Light Blue	144 90H	145 91H	146 92H	147 93H	148 94H	149 95H	150 96H	151 97H	152 98H	153 99H	154 9AH	155 9BH	156 9CH	157 9DH	158 9EH	159 9FH
Light Green	160 A0H	161 A1H	162 A2H	163 A3H	164 A4H	165 A5H	166 A6H	167 A7H	168 A8H	169 A9H	170 AAH	171 ABH	172 ACH	173 ADH	174 AEH	175 AFH
Light Cyan	176 B0H	177 B1H	178 B2H	179 B3H	180 B4H	181 B5H	182 B6H	183 B7H	184 B8H	185 B9H	186 BAH	187 BBH	188 BCH	189 BDH	190 BEH	191 BFH
Light Red	192 C0H	193 C1H	194 C2H	195 C3H	196 C4H	197 C5H	198 C6H	199 C7H	200 C8H	201 C9H	202 CAH	203 CBH	204 CCH	205 CDH	206 CEH	207 CFH
Light Magenta	208 D0H	209 D1H	210 D2H	211 D3H	212 D4H	213 D5H	214 D6H	215 D7H	216 D8H	217 D9H	218 DAH	216 DBH	219 DCH	230 DDH	230 DEH	223 DFH
Yellow	224 E0H	225 E1H	226 E2H	227 E3H	228 E4H	229 E5H	230 E6H	231 E7H	232 E8H	233 E9H	234 EAH	235 EBH	236 ECH	237 EDH	238 EEH	239 EFH
H.I. White	240 F0H	241 F1H	242 F2H	243 F3H	244 F4H	245 F5H	246 F6H	247 F7H	248 F8H	249 F9H	250 FAH	251 FBH	252 FCH	253 FDH	254 FEH	255 FFH

Color Selection In A BASICQ Program

When you are programming using BASICQ you can use various BASICA graphics commands to put color on the screen. The values for the colors you desire can be indicated using hex or decimal notation. When you use decimal notation in a program line you need only enter the specific number, as in

```
10 COLOR 10
```

But when you are using hex notation you must remember to precede the hex color value by an &H, as in

```
10 COLOR &HA
```

The color value can also be contained as in operating parameter in a specific graphics command, such as CIRCLE:

```
10 CIRCLE (21,21),15,245,,,5/6
```

or

```
10 CIRCLE (21,21),15,&HCD,,,5/6
```

BASICQ Graphics Command Changes

In order to take full advantage of the many enhancements provided by Quadcolor II (additional screen modes, wider color selection, etc.) IBM's BASICA is modified to create BASICQ, which will run any program written to run under BASICA while providing support for the extended capabilities of Quadcolor II. This support is provided in the form of extensions to the standard BASIC graphics comands. Thus you do not have to learn new commands; the parameter limits for the old graphics commands have simply been expanded.

The following section explains the graphics commands and how they operate in the two Quadcolor II screen modes. For a full and complete description of each commanc, see IBM's "BASIC" Manual.

Out of Range Color Values

When you are specifying color values in either Screen 3 or Screen 4, make sure you enter only the values allowed in BASICQ. In either mode, if you specify a value larger than 255 (FFH), the computer will send you back an ILLEGAL FUNCTION CALL or an OUT OF RANGE VALUE message. In Screen 4, if you specify a value greater than 15 (FH) but less than 255 (FFH), BASICQ will convert the value to one that is acceptable by ignoring the most significant nibble value and accepting only the least significant nibble value.

For example, in Screen 4 if you specify a color value of 33 — a combination of blue (least significant) and green (most significant) the program will only place blue on the screen. The green will be automatically eliminated.

BASICQ STATEMENTS

CIRCLE

Format: CIRCLE (x,y),r[,color[,start,end[,aspect]]]

Screen 3: In Screen 3 (320 x 200) you can specify any of 136 distinct colors for the color of the ellipse. The color is specified inserting a *color* value in the command syntax. This value can be specified in decimal form as any number from 0 to 255 (due to repeats in the color, only 136 distinct hues are represented in the 256 combinations), or in hex form (preceded by &H) as any symbol from OO to FF. (For more information on this manner of color selection see the section titled COLOR SELECTION.)

In Screen 3, the aspect ratio defaults to 5/6.

Screen 4: In Screen 4 (640 x 200) you can specify any of sixteen colors for the color of the ellipse. The color is specified by inserting the *color* value in the command syntax. This value can be specified in decimal form as any number from 0 to 15, or in hex form (preceded by &H) as any symbol from 0 to F.

In Screen 4, the aspect ratio defaults to 5/12.

EXAMPLE

```
10 Screen 3
20 Cls
30 Circle (200,100),20,&H74
40 Circle (200,100),50,17
50 End
```

This example draws in Screen 3 a pink (&H74:white/red) circle inside a blue circle (17:blue/blue).

```
10 Screen 4
20 Cls
30 Circle (160,100),40,&HEE,,,5/6
40 Circle (240,100),40,68
50 End
```

This example draws in Screen 4 a yellow (&HEE:yellow/yellow) oval (the oval shaped coming from the changed aspect ratio: 5/6) next to a red (68:red/red) circle.

CLS

Format: CLS

Screen 3 and 4: CLS, Clear Screen, works the same way in either Screen 3 or Screen 4. When the CLS command is issued the currently active display bank is erased. The nonactive bank, however, remains intact. For instance, if you were working in Screen 3 and you issued a CLS command, Quadcolor II's display memory would be erased, wiping out the images stored in Screen 3 and 4. The display memory in Quadcolor I, however, would remain intact.

EXAMPLE

```
10 Screen 1
20 CLS
30 Circle (160,50),15,15
40 Screen to 4
50 CLS
60 Screen on 2
70 Circle (320,50),60,4
80 Screen to 1*
90 CLS
100 Screen to 4
110 Screen off 1
120 End
```

This program begins in Screen 1 by erasing the Quadcolor I display bank (Screens 0,1, and z). Then a white circle is drawn on the screen (line 30). After this, the programs switches to write to Screen 4 (line 40) where it begins by clearing the Quadcolor II bank (line 50). The white circle in Screen 1 remains unaffected. Next an overlay of Quadcolor I with Quadcolor II is initiated (line 60) and a red circle is drawn on Screen 4 (line 70). A red circle around a white circle is now dispayed due to the overlay of the screens. Next, the computer switches to write to Screen 1 again (line 80) and the white circle is erased (line 90).The computer switches again to write to Screen 4 (line 100) and the overlay is then turned off (line 110) resulting in the Screen 4 image — a red circle — remaining on screen.

**Remember, after working in a Quadcolor I screen mode then switching to a Quadcolor II screen mode, you must always return to the Quadcolor I mode you left when you write to Quadcolor I again, or you will risk losing the Quadcolor I information.*

COLOR

Format: COLOR (foreground) (,background)

Screen 3 and 4: The BASICA COLOR statement is used chiefly for placing colored text on the screen. A *foreground* and a *background* color can be selected. The following restrictions apply: In Screens 3 and 4, the *foreground* color can range in value from 0 to 15 (decimal) or O to F (Hex). The *background* color can range in value from 0 to 7 (decimal) or 0 to 7 (Hex).

EXAMPLE

```
10 Screen 3
20 CLS
30 Color 15,6
40 Print "This is text with white foreground and
brown background"
50 End
```

This sets the background color of the screen to brown and the foreground, the text characters, to high intensity white.

DRAW

Format: DRAW string

Screen 3: In Screen 3 the DRAW subcommands C and P can accept the same color values as in the CIRCLE command above. That is, when the DRAW command is used the option Cn can set a color that ranges, in decimal form, from 0 to 255. In hex form, this range is from 00 to FF. When the C option is used, the value given colors the image drawn to the specific color value selected. The default color is the current foreground color for the screen mode you are working in.

The P option syntax in Screen 3 is:

P paint, boundary

This option lets you paint the inside of the image drawn.

The image color is set to the specified *paint* value: in hex, an expression from 00 to FF; in decimal, any number, any number from 0 to 255. The boundary color is set by the value of *boundary*: in hex, any expression from 00 to FF; in decimal, any number form 0 to 255.

Separate the *paint* and *boundary* values with comma.

Screen 4: In Screen 4 the DRAW subcommands C and P work the same way as in Screen 3 with one exception. The color value range changes. In this mode, the color values for C can range from 0 to 15 in decimal notation, and O to F in hex notation. The range is the same for the paint and boundary parameters in subcommand P.

For more detail on the Pn option, see the PAINT command.

EXAMPLE

```
10 Screen 3
20 CLS
30 Draw "C174"
40 Draw "U80 R80 D80 L80"
50 End
```

This program draws a yellow/light green (C174) box on the screen.

NOTE: Always place the Cn option before the box parameters.

```
10 Screen 4
20 CLS
30 Draw "C10"
40 Draw "U50 R100 D50 L100"
50 Draw "BU3 BR3"
60 Draw "P15,10"
```

This program draws a light green (C10) box on the screen then paints the inside of the box high intensity white ("P15,10").

GET

Format: GET (xl,yl)-(x2,y2), arrayname

Screen 3: The GET statement works the same way in Screen 3 as it does in Screens 1 and 2. Only now the bits-per-pixel size is changed. Quadcolor II and BASICQ use 8 bits per pixel in Screen 3. This requires a larger array size to hold the larger bit/pixel sizes.

Screen 4: In Screen 4, GET works just like in Screen 3, but now the bits-per-pixel is 4.

EXAMPLE

To determine the size in bytes of the array you wish to GET, IBM supplies the following formula:

4 + INT ((x* bitsperpixel + 7)/8) *y

In Screen 3 if you wanted to GET an image that occupied 100 horizontal dots (the x value) and 100 vertical dots (the y value) the formula would be interpreted in this manner:

4 + INT((100* 8 + 7)/8) * 100 or
4 + INT((807)/8) * 100 or
4 + 100 * 100 or
10,004 bytes

LINE

Format: LINE [(xl,yl)]-(x2,y2) [,[color] [,B[F]] [,style]]

Screen 3: in Screen 3 the color of the line can be specified as any of 136 different colors. In decimal notation, this value can range form 0 to 255. In hex notation this value can range from 00 to FF. The *color* of line will always default to the current character foreground color. Plus, a distinct last endpoint value will be kept for each of the four graphics modes.

Screen 4: LINE can be used in Screen 4 the same way it is used in Screen 3, except the color range is restricted to 16 colors. In decimal notation this value can range form 0 to 15. In hex notation this value can range from 0 to F.

EXAMPLE

```
10 Screen 3
20 CLS
30 Line (70,80)-(120,150),124,BF
40 Line (100,100)-(150,150),&HC1
50 End
```

This program draws a pink (124: light red/white) box in the upper portion screen and a purple (&HC1: blue/light red) diagonal line across the lower portion of the screen.

```
10 Screen 4
20 CLS
30 Line (320,10)-(520,80),4,B
40 Line (320,100)-(520,180),2,BF
50 End
```

This program draws a red (color 4) box in the upper right corner of the screen and solid green box (color 2) in the lower right corner of the screen.

LOCATE

Format: LOCATE [row] [,[column]]

Screen 3 and 4: LOCATE positions the cursor in Screen 3 and 4 the same as it does in the other screen modes. Its operation does not change. But the Quadcolor driver software will keep a separate row and column for the Quadcolor I and Quadcolor II displays so that switching screen modes will not cause the computer to lose track of the corresponding cursor position.

EXAMPLE

```
10 Screen 3
20 Cls
30 locate 12,28
40 Print "The Middle of the Screen";
50 locate 25,1
60 Print "The Lower Left Corner of the Screen"
70 End
```

The example places the cursor in the center of the screen to write: "The Middle of the Screen"; then places the cursor at the lower left corner of the screen to write: "The Lower Left Corner of the Screen."

PAINT

Format: PAINT (x,y) [[,paint] [,boundary] [, background]]

Screen 3: In Screen 3, the *paint* and *boundary* parameters of the PAINT command can accept the same color values as in the CIRCLE command. PAINT can be a numeric expression or a string expression. When you define PAINT with a numeric expression you can specify a color value in either hex or decimal notation. In hex, your values can range from OO to FF; in decimal, from 0 to 255.

If PAINT is a string expression then tiling is performed, as stated in the IBM BASIC Manual. In Screen 3, tiling is supported with 8 bits per pixel.

Boundary is a numeric expression for the color of the edges of the image to be painted. The value for the color of *boundary* can be in hex or decimal notation. In hex notation, the value can range from OO to FF. In decimal, the value can range from 0 to 255.

Screen 4: In Screen 4, *paint* can also be a numeric expression or a string expression. If *paint* is a string expression, tiling is performed as it is in Screen 3, but there are now 4 bits per pixel instead of 8.

If *paint* is a numeric expression, the color value can be specified in either hex or decimal notation. In hex, the values 0 through F are allowed. In decimal, the values 0 through 15 are allowed.

Boundary, in Screen 4, can also be specified as either a hex or a decimal value. If hex is used, the expressions 0 through F are acceptable. If decimal is used,the numbers 0 through 15 are allowed.

EXAMPLE

```
10 Screen 3
20 CLS
30 Circle (100,100),40,&HAE
40 Paint (100,100), 121,&HAE
50 End
```

This program paints the inside of a yellow/light green (&HAE) circle with white/light blue (121).

```
10 Screen 4
20 CLS
30 Line (100,50)-(200,75),3,B
40 Line (100,50)-(200,75),3,
50 Paint (110,70),2,3
60 End
```

This programs draws a cyan (3) rectangle, bisects the rectangle with a cyan line, then paints the lower half of the rectangle green (2).

```
10 Screen 3
20 CLS
30 Line (100,130)-(200,170),15,B
40 Line (133,130)-(133,170),15
50 Line (166,130)-(166,170),15
60 Circle (104,134),2,15
70 Circle (104,166),2,15
80 Paint (110,135),&H4E,15
90 Paint (135,135),119,15
100 Paint (168,135),34,15
110 End
```

This program produces a flag by drawing a rectangle (line 30), dividing the rectangle into three sections (lines 40-50), and painting the sections orange (&H4E: yellow/red; line 80), white (119: white; line 90), and green (34: green; line 100).

Running this program in Screen 4 produces results with some major differences. The same coordinates produce a much more squarish rectangle due to the doubled horizontal resolution. Additionally the program now takes only the least significant nibble from the color values specified in lines 80, 90 and 100. Naturally this affects the appearance of the specified colors. But when the program is run on Screen 4 you will notice that colors 34 and 119 appear unchanged from Screen 3. This is because they are "pure" values: combinations of the same colors (white/white and green/green). When the most significant nibble value is ignored in Screen 4, white stays white and green stays green. But the orange created on Screen 3 is not pure, rather it is a blend of yellow (least significant) and red (most significnat). In the change to Screen 4 the most significant nibble value is lost and the color appears simple as yellow.

POINT

Format: v=POINT (x,y)
v=POINT (n)

Screen 3 and 4: In screen modes 3 and 4, POINT functions exactly as it does in the other graphics modes: it returns the color of a specific point on the screen. In mode 3, any of 136 colors can be returned as a value from 0 to 255. In mode 4, any of 16 colors can be returned.

PSET and PRESET

Format: PSET (x,y) [,color]
PRESET (x,y) [,color]

Screen 3: In Screen 3, the color value for both PSET and PRESET can be specified in either hex or decimal notation. If hex is used, the acceptable range of values is OO through FF. If decimal is used, the acceptable range is 0 through 255.

Screen 4: In Screen 4, the color value for both PSET and PRESET can be specified in either hex or decimal notation. If hex is used, the acceptable range of values is O through F. If decimal is used, the acceptable range is 0 through 15.

EXAMPLE

```
10 Screen 3
20 Cls
30 For I =0 to 100
40 PSET (1,1),234
50 Next I
60 End
```

This program draws a diagonal line beginning in the upper left corner of the screen. The line is color yellow/light green.

PUT

Format: PUT (x,y) ,array [,action]

Screen 3: The PUT statement works the same way in screen mode 3 as it does in screen modes 1 and 2. Only now the bits-per-pixel size is changed. Quadcolor II ad BASICQ support 8 bits per pixel in screen mode 3. This requires a larger array size to hold the larger bit/pixel size.

Screen 4: In screen mode 4, PUT works just as in screen mode 3, but now BASICQ supports 4 bits-per-pixel.

For more information on this statement, see GET.

VIEW

Format: VIEW [[SCREEN] [(xl,yl)-(xy,y2) [,[color][,[boundary]]]]]

Screen 3: The color parameter of VIEW in Screen 3 can be specified as any of 136 colors. Hex or decimal notation can be used. If hex is used the acceptable color range is OO through FF. If decimal is used the acceptable range is 0 through 255.

Screen 4: The color parameter of VIEW in Screen 4 can be specified as any of 16 colors. Hex or decimal notation can be used. If hex is used the acceptable color range is O through F. If decimal is used the acceptable range is 0 through 15.

EXAMPLE

```
10 Screen 4
20 Cls
30 View Screen (100,30)-(200,80),4,1
40 End
```

This example establishes a view port in the upperleft screen. The view port is red (color 4) with a blue (color 1) boundary.

WIDTH

Format: Width size

Screen 3 and 4: The WIDTH value in Screens 3 and 4 is set a 80. It cannot be changed form this in these modes. However, under BASICQ, WIDTH will store a value for the Quadcolor I text mode (Screen 0,) so that it may return to this screen with the preselected WIDTH value intact whenever SCREEN TO is used.

SPECIFICATIONS

Quadcolor II is a bit-mapped graphics board that, when combined with Quadcolor I, increases the display resolution of the IBM Personal Computer to 640 x 200 x 16 colors. Quadcolor II also provides you with an IBM compatible game port for connecting joysitcks or a pair of game paddles. By mounting Quadcolor II piggyback fashion onto Quadcolor I no additional I/O slots are needed for the expansion.

Provided with Quadcolor II is a software package used to modify IBM BASICA to BASICQ. BASICQ is an enhanced version of BASICA which allows you to take full advantage of the extended graphic capabilities of Quadcolor II while using standard BASICA graphics commands.

This section provides technical information for those of you who wish to write programs for Quadcolor II using Assembley language or some high level language other than BASICA or BASICQ.

Hardware

Quadcolor II can be used only in conjunction with Quadram's Quadcolor I color graphics adapter board. Quadcolor II is attached to Quadcolor I via two on-board connectors: a 34-pin and a 26-pin connector.

The horizontal 34 pin connector carries standard IBM bus signals (address,data, control, clocks) while the vertical 26 pin connector carries the timing signals generated and used by Quadcolor I.

Quadcolor II contains 64K bytes of memory with each pixel stored in a nibble. One I/O address is used to disable and enable Quadcolor II's video output. The game port is accessed through one I/0 address. It will accept four resistive inputs and four switch outputs.

Display Memory

Quadcolor II's display memory consists of 64K bytes of dualported, bit-mapped RAM. The CPU has access to the memory at all times with one, two, or three Wait states (210, 420, or 630 nanoseconds) occuring with each access.

As mentioned earlier, each pixel occupies one nibble in Quadcolor II memory. Since Quadcolor II's resolution is 640 x 200 pixels, the total number of pixels displayed is 128,000. This translates to 64,000 bytes of display memory, leaving 1,536 bytes (65,536 minus 64,000) of undisplayed memory in the 64K byte address space. Quadcolor II's display memory occupies the space from D0000H to DFFFFH. It can also occupy the 64K block from E0000H to EFFFFH. This isn't an additional 64K, just another 64K block at which memory can be accessed. For the sake of simplicity, the remainder of this section will discuss only the 64K block starting at D0000H.

The byte at D0000H contains the two pixels that are displayed in the upper left corner of the screen. There are two nibbles in each byte: most significant and least significant. The most significant nibble of any byte is diaplayed to the left of the least significant nibble. Therefore, the pixel in the upper left corner of the screen is controlled by the most significant nibble in memory location D0000H.

Mapping of pixels to display memory is done in a linear fashion. The first display line (640 pixels) is contained in the first 320_{10} (140H) bytes of memory while the second display line is located in the next 320 byte block. The absolute byte address for any pixel may be found with the following formula:

$$\text{Pixel's byte address} = \text{DOOOOH} + (\text{row\#} \times 320_{10}) + (\text{column\#}/2)$$

where and $O \leq \text{row\#} \leq 199$ and
$O \leq \text{column\#} \leq 639$

If the column number is even the pixel is the most significant nibble; if the column number is odd, it is the least significant nibble.

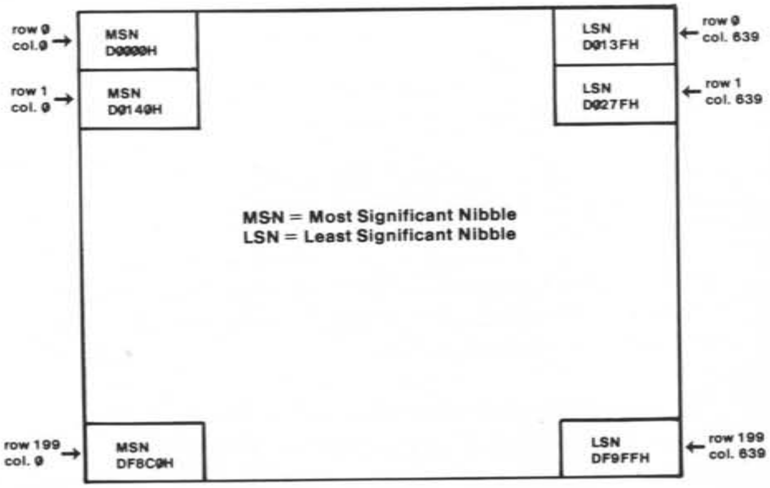
The last display pixel — occuring in the bottom right-hand corner of the screen — is the least significant nibble of memory address DF9FF. This leaves DFA)) to DFFFF as undisplayed memory space. (You may use this for ordinary memory storage if you wish.)

Figure 2 shows some display pixels and their respective addresses. The byte and nibble organization is shown below:

D7	*D6*	*D5*	*D4*	*D3*	*D2*	*D1*	*D0*
I	*R*	*G*	*B*	*I*	*R*	*G*	*B*
Pixel A				*Pixel B*			

Fig. 2
Pixel A appears to the left of pixel B. Each bit of a nibble controls a different color or intensity. The four bits are Intensity, Red, Green, and Blue. When a bit is on (1) that color is on; when a bit is off =0) that color is off. The Intensity bit will intensify, or brighten, the color selected by the R, G, and B bits.

Fig. 3 Quadcolor Display Memory/Map



Display I/O

Quadcolor II uses only one bit at one I/O location to enable or disable the video output. Bit D4 of I/O location 3DEH, when set to 1, will enable Quadcolor II's video output. Setting this bit to 0 will disable the video output. A short discussion of the Quadcolor I/II video enabling and disabling features is in order at this point.

The ultimate output of Quadcolor II is four lines carrying the I, R, G, and B data to the Quadcolor I board. An intermediate output of the Quadcolor I board (before the video buffers and composite video circuitry) also consists of four lines carrying Quadcolor I IRGB data. These two sets of IRGB lines are logically ORed together to create on IRGB set of lines that becomes the output of Quadcolor I.

Like outputs of the two boards are ORed with Quadcolor II Blue; Quadcolor I Red is ORed with Quadcolor II Red; and so on. The ORing of the two sets of video signals is called overlaying. With overlaying, a high resolution graphics image, created in Quadcolor II's memroy, can be superimposed over a text or graphics image created in Quadcolor I's memory. Your program may be listed and scrolled off the screen while the image, with which you are having a problem, remains (on screen **and** in Quadcolor II memory).

Forcing the IRGB outputs of either board to all Os has the effect of disabling that board's output and letting the other board's output be the only video displayed. Thus the purpose of the I/O bit (D4 of 3DEH) discussed at the beginning of this section is to allow you a simple means of disabling and enabling Quadcolor II's video output.

Disabling Quadcolor I's video while leaving Quadcolor II's video enabled is accomplished in a slightly different manner. Two steps must be taken to disable Quadcolor I's video output. They are:

1. A 0H must be written to the least significant nibble of I/O port 3DDH (the Quadcolor I register as described in the Quadcolor I Operations Manual). This register powers up with a 00H and most users will not ordinarily change this value. If a nonzero has been stored in the least significant nibble, the user must remember this value so that it may be rewritten if and when Quadcolor I's video is reenabled.

2. If Quadcolor I is currently operating in 80 column text mode (SCREEN 0) a 13H must be stored in the Mode Control Register at I/O location 3D8H. For any other Quadcolor I mode, a 12H must be stored here. Again, since this is a write-only register, the user must "save" the last value written here or at least be able to determine what needs to be written there in order to return to his current mode of operation when and if Quadcolor I's video is reenabled.

Game Control Port

The Game Control port is operationally identical to IBM's. The port accepts four resistive inputs and four switch closure inputs. This port could also be used as a four channel A/D converter. Each resistive input is used as part of an RC timing circuit to change the length of a timing pulse. Software measures the time of this pulse to convert the resistive (joystick position) to a digital value. The Game Control interfaces to the outside world through a 15 pin D-type connector cable. The other end of this cable is a 16 pin DIP type connector that plugs into Quadcolor II. The Game Control adapter occupies one I/O location at 201H.

Game Control Adapter

Writing to I/O location 201H will cause the resistive timing networks to "reset". Immediately after the write operation occurs, the four resistive inputs (read in bits D3-D0 of I/O location 201H and explained later) will all be true (i.e., equal to 1). The time out period (the time at which the digital input representing each resistive input goes low) will be a function of the joystick position or resistive input. Thus, software must time the period for which each resistive input channel's digital input is high.

Reading I/O location 201H reads the states of the four switch inputs and the states of the digital outputs of the four resistive input RC networks. Normally, you would have two joysticks, A and B (or two game paddles) with two buttons on each.

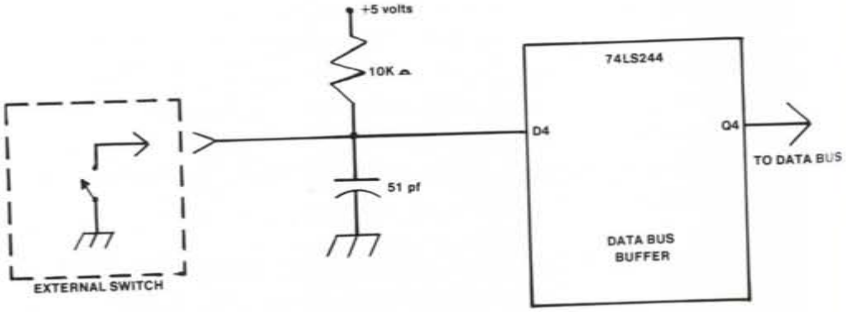
Joystick A's position in the X direction would be referred to as A-X, while its Y direction would be referred to as A-Y. The two buttons would be A1 and A2. The bit position assignments when read I/O location 201H are:

D7	D6	D5	D4	D3	D2	D1	D0
B2	B1	A2	A1	B-Y	B-X	A-Y	A-X
Switch Inputs				Resistive Inputs			

The switch input lines are pulled up through 10K ohm pullup resistors to +5v. When a switch is open (or button not pressed) a "1" will be read for that input. When a switch is closed (or button pressed) a "0" will be read for that position. The switch inputs are not debounced and software debounce should be employed.

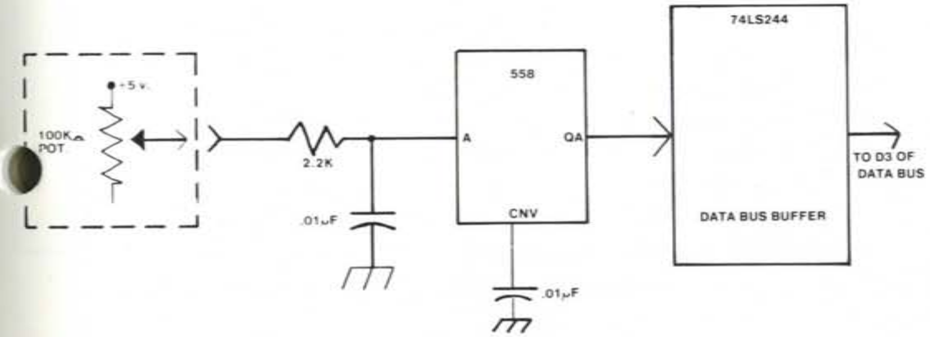
The switch circuitry for one switch is shown below.

Fig. 4 Switch circuitry



The resistive inputs, as discussed earlier, are used in an RC network to create a digital pulse whose duration is proportional to the resistive input value. The range of resistive values is from 0-100K ohms. The RC network connects to a 558 (four channel timing IC) whih is used essentially to convert the charge on the timing capacitor (C) to a digital pulse. The digital pulses of the four resistive channels are read on the lower four bits of the I/O location 201H. The circuitry for one resistive channel is shown below.

Fig. 5 Resistive channel circuitry



The duration of a pulse (time it is equal to 1) may be calculated from the following formula:

Time = 24.2microsec + (0.011 x R)microsec

When the wiper area of the potentiometer is at one extreme and connected to +5v (R=0) the time period is 24.2 microseconds; when at the other extreme (R= 100K ohms) the time period is 1.124 miliseconds. These times will be aproximate due to component tolerances. The user must do a write (data doesn't matter) to I/O location 201H to reset the four timers before measuring the periods of the pulses. The connections at the 15 pin D-type connector are:

16 pin DIP type	15 pin D-type	Signal
16	1	+5v
1	2	Button A1
15	3	Resistive Input AX
2	4	GND
14	5	GND
3	6	Resistive Input AY
13	7	Button A2
4	8	+5v
12	9	+5v
5	10	Button B1
11	11	Resistive Input BX
6	12	GND
10	13	Resistive Input BY
7	14	Button B2
9	15	+5v
8	-	Not Used

Software

The driver software (QCI) supplied with Quadcolor II is completely compatible with the IBM video I/O BIOS driver. All function calls for Interrupt 10h remain unchanged. There are, however, two additional function calls designed to provide access to Quadcolor I and II's capabilities. These extra Interrupt 10H function call are described below.

AH = 16

Select Mode Without Erasing

AL = desired mode number
0-7, normal modes as defined for AH=0.
8, select Quadcolor II. This is also used, when AH=0, to select Quadcolor II with erasing.)

This call selects a new screen mode without erasing the screen. It should be noted that when this function is performed it should be immediatley followed by a call to set the cursor position.

AH = 17

Switch Quadcolor I or II On/Off

AL=on/off control of Quadcolor I and II
0, turn Quadcolor I off
1, turn Quadcolor II off
2, turn Quadcolor I on
3, turn Quadcolor II on

Since Quadcolor I and II operate independently, both can be turned on at the same time and turned off individually.

NOTES

NOTES

NOTES

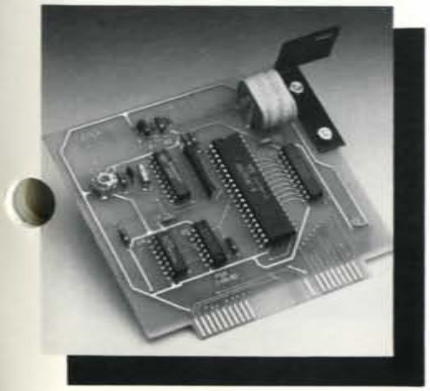
Get more out of your Personal Computer system with these other fine Quadram Products

Quadboard

Quadram's Quadboard is a six-function expansion board for the IBM Personal Computer. It is designed to greatly increase the PC's capabilities for minimum cost while using only one internal expansion slot. The functions provided on this one board include: Random Access Memory (RAM) expansion from 64K to 256K bytes in 64K increments, an EIA RS-232C Serial Interface for asynchronous communications, a Centronics compatible parallel interface for printer driving, and a battery-powered Quartz clock/calendar to keep your PC's internal clock always on time.

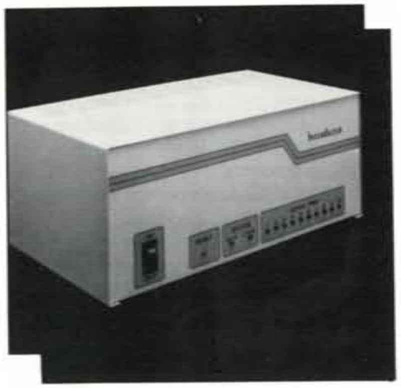
Memory Expansion Board

Perfectly compatible with other peripheral cards, Quadram's Memory Expansion Board is designed to increase the random access memory capacity of the IBM Personal Computer in 64K byte increments up to 192K. The Memory Expansion board has been manufactured with performance and reliability in mind, and has been constructed of only high-quality material and components. Each board has been burned-in and thoroughly tested to insure many years of dependable service.



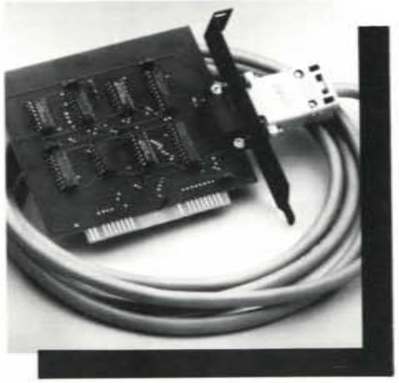
Clock/Calendar

The Clock/Calendar Card from Quadram Corporation is a small plug-in unit designed for the IBM Personal Computer. It provides an accurate crystal-controlled date/time keeping function which automatically sets the IBM DOS internal clock/calendar whenever DOS is booted from a diskette. The card is powered by a self-contained battery automatically recharged by the computer, and provides continuous time keeping even when the computer is off for long periods of time.



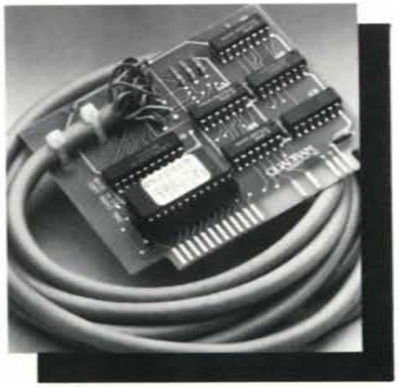
InterFazer

The InterFazer from the Quadram Corporation is an intelligent peripheral controller/buffer dsigned to provide interfacing and priority control for up to eight output devices feeding and sharing one or two input devices. The inputs and outputs may be in any combination of serial or parallel. The InterFazer's state-of-the-art design allows it to function as a multi-user printer controller, an incompatible device interface, a computer I/O expander, a peripheral multiplexer, a data transfer rate converter, and an additional peripheral buffer.



IPIC

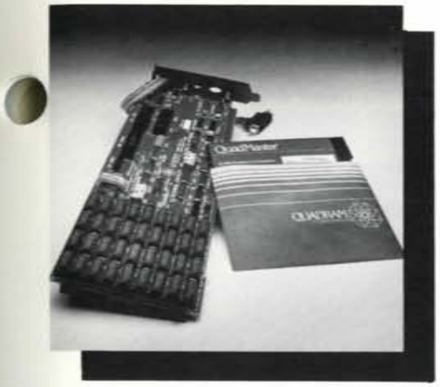
Quadram's IPIC (IBM Parallel Interface Card) is a general purpose Centronics parallel interface card complete with cable for the IBM Personal Computer. It is specifically designed to drive printers, but may be used as an I/O driver for any parallel device that matches its input/output capabilities. The IPIC works exactly like IBM's Parallel Printer Adapter and supports the graphics function on many common graphics printers including IBM, Epson, Centronics, and others.



APIC Graphics

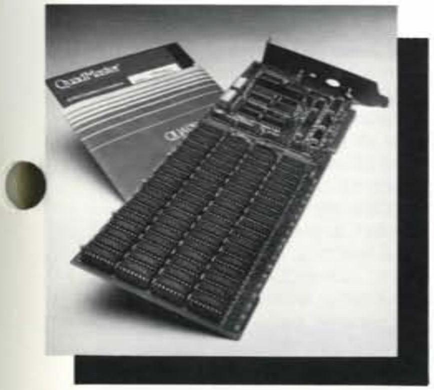
The Apple Parallel Interface Card is manufactured by the Quadram Corporation as a Centronics compatible parallel interface for the Apple II and Apple II+ computers. The card contains a graphics option. Both Apple Graphics pages can be printed from the screen with several simple commands.

APIC-G has many text features, including adjustable margins, page length, line length, and others. APIC-G is available for several types of printers.



Quadboard II

Quadram's Quadboard II is an all-on-one board which combines six IBM PC functions in one. This multifunction board combines two serial ports, chronograph, memory expansion, RAM disk and spooler — all compatible with IBM PC hardware. The two RC-232 async ports can be used for modems, printers, and other serial devices. Memory expansion is socketed and fully expandable in 64K increments up to 256K. Full parity generation and checking are standard.



Quad 512+

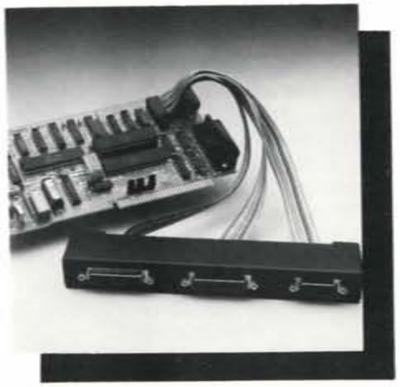
Now IBM PC owners can enjoy faster computing times and more capabilities with the Quad 512+. This new board from Quadram combines memory expansion of 512K RAM in increments of 64K, 256K, or 512K; a serial port; RAM disk; and spooler. The RS-232 async serial port uses the same chip as the IBM serial board and is used for connecting modems, printers, and other serial devices. QuadMaster software with QuadRAM Drive and spooler is included.

Microfazer

The Quadram Microfazer is a univeral print buffer with 8K to 512K RAM — the first with a half-megabyte! The Microfazer receives information in its memory, then sends it to the printer at an appropriate speed. The Microfazer comes with a copy feature allowing additional copies of the buffered information. The unit is sized to stack with popular modems and other peripherals; and some even plug directly into the printer!

Quadscreen

Quadscreen is a high-resolution (960 x 512), bit-mapped, "big-screen" monitor for the IBM Personal Computer. The large monochrome screen can display 10,240 characters — more than five times the screen capacity of IBM's PC monitor. Dot addressability, super spreadsheet, and split-screen functions are some of the distinctive elements of Quadscreen that maximize graphics capabilities and increase word processing efficiency. Quadscreen comes with monitor, connector cable, and a video card with 128K bytes of available RAM.



Quad I/O

Quadram's Quad I/O combines up to five of the most-needed IBM PC input/output devices onto one card. This multifunction board contains a parallel port designed to operate most printers and other parallel devices; a battery-powered chronograph — an extremely accurate real-time clock/calendar; a game port for game paddles, fire buttons, or joysticks; a serial port for connecting modems, printers, and other serial devices; plus an optional second serial port which can be ordered with the board or added at a later date.

Quadchrome

This is the perfect monitor for any data processing environment — Quadchrome by Quadram. Quadchrome is a high-resolution (690 x 240) RGB color monitor designed with an NEC .31 dot pitch cathode ray tube. The high resolution makes Quadchrome perfect for word processing, spreadsheet programs, and other common business applications. But Quadchrome also has amazing color capabilities. It's able to produce up to 16 different colors on the screen at once. This makes Quadchrome perfect for color graphics work, too.

Quadlink

Quadram introduces Quadlink, the revolutionary enhancement board that turns your IBM Personal Computer into an Apple-compatible system. Quadlink is an Apple-emulator that plugs into one of the PC's expansion slots. The board uses the same disk drive and I/O devices as the PC, working in complete harmony with the computer system. With Quadlink, no diskette reformatting is necessary. Just put the Apple diskette in the drive and watch it go.

Quadlink comes complete with printed circuit board and all the necessary cables for integrating the board into the system. Plus there's software for booting the Quadlink and running the Apple DOS 3.3 and thorough documentation explaining the installation and operation of the board.

Quadcolor I and II

Quadram presents Quadcolor, the most powerful color graphics adapter package available today. Quadcolor is perfectly compatible with the Personal Computer and XT and comes in two versions.

Quadcolor I has 32K of on-board RAM for up to 16 active video pages in text mode and two complete pages in graphics mode. Plus, you get a choice of two true colors in high-resolution mode with Quadcolor I.

Snap Quadcolor II onto Quadcolor I for 96K of RAM and choose from 136 different colors for ultra high-resolution bit-mapped graphics. Quadcolor II also comes with BASICQ, an enhanced graphics software package that's powerful and easy to use.

The Quadram Qlique (pronounced clique)

You're special to us . . .

As a Quadram product owner, you're very special to us. You've become a preferred customer. And as a preferred customer we want to keep you updated on what's happening at Quadram. That's why we want you in the Qlique (pronounced clique). The Quadram Qlique is a special membership of all our preferred customers. As a member in the Qlique you'll receive information on new product releases, general information about Quadram's complete line of microcomputer enhancements, plus free periodic software updates. It's a great way to "stay tuned" to what's happening at Quadram!

Joining the Qlique is easy . .

It's free! All you have to do is complete the attached card and mail it back to us. We'll put you on our membership list and make sure you stay up-to-date on the exciting things going on at Quadram.

C'mon. Get in tune. Join the Qlique by sending in the card today.

NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 397 NORCROSS, GA 30093

ATTN: QUADRAM QLIQUE
QUADRAM CORPORATION
4355 INTERNATIONAL BOULEVARD
NORCROSS, GEORGIA 30093

TAPE PLEASE DO NOT STAPLE TAPE