QUANTUM DATA®

Model 801GC, 801GF & 801GX

Portable Video Signal Generators

Owner's and Programmer's Manual

Thank you for considering our product.

At Quantum Data, we are committed to providing you with innovative products and superior customer service. If something doesn't seem right or you see an opportunity that we haven't addressed, please let us know. Helping our customer's solve their problems is what we do best.

QUANTUM DATA®

2111 Big Timber Road
Elgin, IL 60123-1100 USA

Phone: (847) 888-0450
Fax: (847) 888-2802
BBS: (847) 888-0115
[≤19.2KB, 8-N-1 protocol]

Please note that on January 20, 1996 our phone, fax and BBS area codes changed from 708 to 847.

Internet Connections
World Wide Web Site:
http://www.quantumdata.com
Sales & Service E-mail:
sales@quantumdata.com
Technical Support E-mail:
support@quantumdata.com

Table of Contents

To quickly find information on the topics shown on the right, just match the black bars with the black markers on the edges of the pages in the manual.

For more information on the contents of each chapter, please refer to the table of contents starting on the next page.

To locate information on a specific subject, please refer to the index at the end of this manual.

Model 801GC, 801GF & 801GX
OwnerÕs & Programmers Manual
Part# 68-00157 (rev. A 21-Feb-96)

Trademark Acknowledgments:

Quantum Data is a registered trademark of Quantum Data Corporation.

IBM is a registered trademark of International Business Machines Corporation.

Macintosh and Apple are registered trademarks of Apple Computer, Inc.

Microsoft and MS-DOS are registered trademarks of Microsoft Corporation.

MultiSync is a registered trademark of NEC Home Electronics, Inc.

Radio Shack is a registered trademark of Tandy Corporation.

Other products or company names mentioned are registered trademarks or trademarks of their respective companies.

Chapter 7: Troubleshooting 7-1

Chapter 8: Error Messages 8-1

Appendix A: Specifications 11-1

Index

Notes:

Chapter 1: Introduction

Features

Product Overview

Features

- ✓ low cost
- ✓ portable - fits in a briefcase
- ✓ ultra-simple controls
- ✓ ultra-fast format-loading & image-drawing
- ✓ color NTSC and PAL compatible composite and S-video outputs on the 801GX
- ✓ self calibrating analog video outputs
- ✓ industry-standard output connectors
- ✓ display and edit formats on the unit under test
- ✓ over 100 industry-standard signal formats built-in
- ✓ room for up to 300 user-defined formats
- ✓ programmable pixel rates up to 150 MHz on the 801GC and 801GX; up to 250 MHz on the 801GF
- ✓ over 50 test images built-in
- ✓ custom test images can be created and saved
- ✓ user defined format-image test sequencing

Product Overview

The 801G series are low-cost portable video signal generators designed for basic testing and alignment of various types of raster-scanned displays. This manual covers the 801GC, 801GF and 801GX models. Information that mentions the "801G*" applies to all three models.

Formats

A **format** is a set of specifications that describe the video signal required by a particular type of display.

The generator is shipped with over 100 signal formats in place for driving a variety of industry-standard displays. Formats for other displays can be added using the built-in graphics user interface editor or a personal computer or terminal via the built-in RS-232 and IEEE-448 interfaces.

Formats are stored in a non-volatile memory. This memory has room for up to 300 formats. Permanent copies of industry-standard formats are stored (along with the generator's operating code) in EPROM and can be copied into format memory and modified as required.

Images

The 801G* features over 50 test images that allow a wide variety of display criteria to be checked. The generator has an image loop feature that can be used for monitor burn-in or at trade shows to display a series of images over and over. User defined custom test images can also be created and saved in non-volatile memory.

Controls

The number of controls on the 801G* has been minimized to insure simple operation. Two knobs and 8 lighted push-buttons provide complete control of the generator. One knob selects the video signal format. A second selects the test image. Three push-buttons gate the individual video components on and off. Another three push-buttons select any of three available synchronizing signals. A seventh button allows all of the outputs of the generator to be turned on or off with a single key stoke. Finally, an image stepping button is provided that calls up alternate versions of some images.

Connectors

The output connectors on the 801G* match those found on popular computers and video systems. These connectors eliminate the need for expensive and bulky conversion cables.

Chapter 2: Basic Operation

Operating Modes

Displays & Indicators

Knobs

Switches

Buttons

Introduction

This chapter gives you a basic overview of the Quantum Data model 801G*'s front panel operating modes and how the displays and controls function in the normal operating mode. Other chapters in this manual cover topics that you may need to know in order to operate the unit. Please refer to the table of contents or index to locate additional specific information on how to use the 801G*.

Operating Modes

Front panel operation

The current firmware supports three main modes of front panel operation:

Normal Format and Images Selection Mode

- Normal signal format and test image selection. All knobs and pushbuttons function as labeled. The upper knob is used to select formats from a list of formats stored in non-volatile memory. The lower knob selects test images (patterns) from a list of built-in and user created images. This is the factory default setting for how the generator will operate on normal power-up. This mode is suitable for use by engineering and service groups that need to be able to quickly select any combination of format and test image that they may need.

  It is also possible to set the 801G* to continuously cycle through the test image list using a given format in this mode. This can be used for burn-in testing or for running single mode displays at trade shows.

Test Sequence Mode

• Running a user defined test sequence. Each step in a test sequence combines one format and one test image. The operator can then go forward and backwards through the steps using a single knob. The 801G* can be programmed so that it automatically enters a test sequence mode on power-up. This mode of operation is suitable for a test position in a manufacturing environment where the same series of tests and adjustments need to be repeated on many identical displays. There is less likelihood that an operator on the line will select the wrong format or skip an important alignment procedure.

It is also possible to set the 801G* to continuously cycle through the steps in a test sequence.

Information on creating and running test sequences is contained in the “Programming” chapter.

Programming Mode

• Programming mode. Formats, the format list, custom images, the image list and test sequences can be created and edited using the built-in Graphics User Interface (GUI). In order to use the GUI, a display that is compatible with any stored format and has at least 640 active pixels and 480 active lines needs to be connected to the 801G*. The contents of the format, custom test image or test sequence are shown on the display. The 801G*’s knobs and buttons are used to select and modify the displayed parameters. The current button functions are labeled at the bottom of the GUI screen. No other hardware, other than the display, needs to be connected to the 801GX.

Information on using the GUI editors is in the “Programming” chapter.

Switching operating modes

The “Programming” chapter contains information on switching between the operating modes.

Displays and Indicators (Normal Mode)

The figure below shows a typical LCD display in the normal operating mode. Please see the “Trouble-shooting” chapter of this manual if the LCD is showing different types of information.

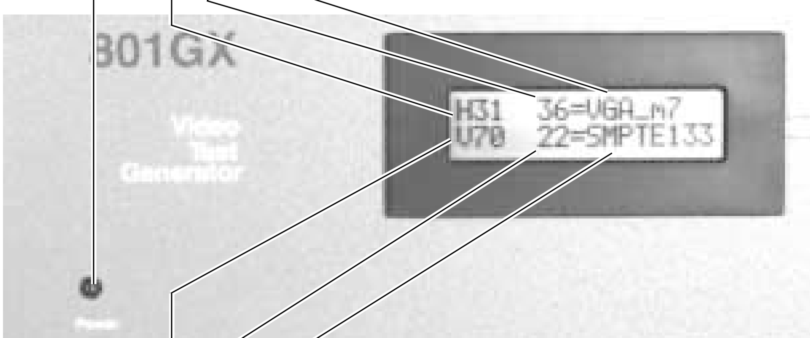
This light is on whenever the 801G*; is plugged into a live AC outlet and the power switch is ON.

The horizontal scanning frequency rounded to the nearest kHz.

The format’s position on the Format knob list.

❖ A "=" sign appearing between the memory location and name indicate the current state and saved state exactly match. Pressing some keys may modify the current state so that it no longer matches the saved version.

The name of the current signal format.
A (’) mark at end indicates a justified format.



The name of the current test image.

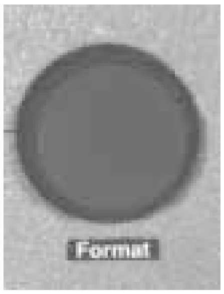
The image’s position on the Image knob list.

❖ An "m" appearing to the left of the image location indicates monochrome analog video.

The vertical scanning frequency (field rate) rounded to the nearest hertz.

Knobs

Format Knob in Normal Mode

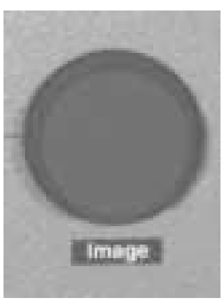


The format knob is normally used to select a signal format. A format is a set of parameters that specify the video and sync signal requirements of a particular display. Format parameters include timing, sync type, video type, display size, etc. By turning the knob, you can scroll through a list of formats stored in non-volatile memory. The list includes many factory default formats. You can edit the list as well as add formats that you create. A new format is loaded with each click of the knob. Loading a format redraws the current test image and updates the LCD. If you select a format containing erroneous information, the outputs of the generator are automatically turned off and an error message is displayed. The knob performs other functions when the editors are being used.

Format Knob in Other Modes

The format knob performs other functions when the 801G* is operated in the either the test sequence or GUI programming modes. Please see the “Programming” chapter for detailed information on operating the 801G* in the other modes.

Image Knob in Normal Mode



The image knob is normally used to select a test image. The exact behavior of the knob depends upon the status of the “Image” push-button. Turning the knob when the button is not lit scrolls through the main list of test images. The knob performs other functions when the editors are being used.

❖ Not all images are supported by all signal formats. Some images in the main image list may be skipped while certain formats are present. For example, the ColorBar image will be skipped whenever a monochrome format is present. If the currently selected image cannot be drawn given a newly selected format, the Outline image is automatically drawn after the new format has finished loading.

Some of the image names in the main list may refer to a sub-set of two or more different images. The images in the sub-sets are selected by first selecting the name of the desired sub-set from the main image list. If the sub-set consists of just two images, pressing the “Image” button will toggle between the two images. The button is lit when the second image is showing.

If the sub-list consists of more than two images, the “Image” button is first lighted by pressing it one time. The image knob can then be used to select images from within the sub-set. The image name on the LCD will not change. Pressing the “Image” button a second time will will return the “Image” knob to normal operation.

❖ If there is only one test image associated with a name in the image list, the “Image” button can not be turned on for that image.

All of the built-in test images (and their uses) are covered in detail in the “Images” chapter.

Image Looping in Normal Mode

• A continuous cycling mode can be selected that draws images one-after-another in an infinite loop. This mode is activated by rotating the image knob clockwise beyond the last image in the image list about one turn. When activated, the message “Loop Enabled” is displayed on the unit under test. After a second or two, the first image in the cycle appears. Turning the image knob counter-clockwise will stop image looping.

Image Knob in Other Modes

The image knob performs other functions when the 801G* is operated in the either the test sequence or programming modes. Please see the “Programming” chapter for detailed information on operating the 801G* in the other modes.

Switches

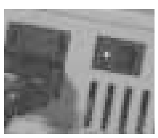
The 801G* has two switches located on its left side. Both switches are related to AC power.

AC Select

This recessed slide switch sets the safe AC line voltage operating range of the generator. The “Making Connections” chapter of this manual describes the correct procedure for setting this switch

❖ Do not change the voltage selector switch setting while the 801G* is connected to the AC power line. Make sure that the switch is in the correct position before plugging in the 801G*.

Power Switch

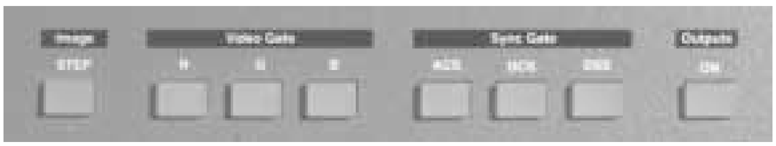


This rocker switch turns the power on and off. Pressing the side of the switch with the color dot or the number “1” turns the power on.

❖ **CAUTION:** Holding down any of the buttons on the front panel while turning on power may produce unexpected and extremely undesirable results. Please see the “Programming” chapter for information on using special power-up button combinations.

Buttons

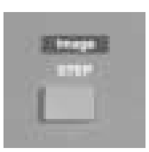
The 801G* has a total of eight push-button switches, arranged into four function groups: Image, Video Gate, Sync Gate, and Outputs. All of the buttons have built-in indicators. When illuminated, a button's function is considered on (or enabled).



This section of the manual describes the functions of the buttons when the 801G* is in the normal mode of operation The buttons are used for other functions when the 801G* is operated in and a test sequence or programming mode. Please see the "Programming" chapter for detailed information on operating the 801G* in the other modes.

❖ **CAUTION:** Holding down any of the buttons on the front panel while turning on power may produce unexpected and extremely undesirable results. Please see the "Programming" chapter for information on using special power-up button combinations.

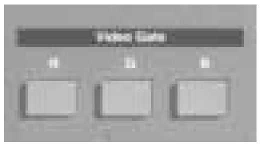
Image Button



The Image push-button determines the behavior of the "Image" knob in the normal operating mode. Please see an earlier section on the "Image" knob for information on how the button interacts with the knob.

❖ If there is only one test image associated with a name in the image list, the "Image" button can not be turned on for that image.

Video Gate Buttons



The Video Gate buttons turn individual color outputs on and off. They also control the adsdition of primary color information to the NTSC / PAL video outputs on the 801GX.

- The R push-button turns all of the red video outputs on and off.
- The G push-button normally turns all of the green video outputs on and off. When a 2-bit digital monochrome (MDA) signal is being generated, the G push-button turns the I (intensity) signal of the video pair on and off.
- The B push-button normally turns all of the blue video outputs on and off. When a 1 or 2-bit digital monochrome signal is being generated, the B push-button turns the V (video) signal on and off.

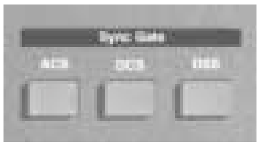
❖ The master output gating button overrides the settings of these buttons when turned off.

Analog Monochrome Operation

Many of the built-in analog video signal formats are stored with the RGB video signal type selected. If you are testing a monochrome monitor with the same timing, you can toggle between color and monochrome modes by pressing both R and B push-buttons simultaneously.

❖ An “m” appearing to the left of the image location on the LCD window indicates that monochrome analog video has been selected.

Sync Gate Buttons



The buttons in this group select the type of sync signal that is used to synchronize the display. Depending on a particular format's settings, more than one type of sync can be selected by pressing two buttons at a time.

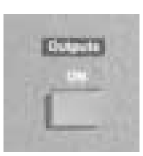
- The ACS (Analog Composite Sync) push-button causes analog sync to be output on one or more of the analog video outputs.
- The DCS (separate Digital Composite Sync) push-button causes a separate digital composite sync signal (CS) to be output.
- The DSS (separate Digital Separate Sync) push-button causes separate digital horizontal and vertical sync signals to be output.

❖ A default sync type is automatically selected whenever a new format is selected. Not all sync types are available with all formats. For example, digital video formats will not allow analog composite sync to be selected. If a button will not light up when pressed, then the corresponding sync type is unavailable.

Re-depressing a sync gate button causes the selected sync to be toggled either on or off. When toggled off, no sync will be sent to the display and the display will be allowed to free-run.

❖ The individual settings are overridden (gated off) whenever the master outputs button is turned off.

Outputs Button



This is the master output signal control. When the master output control is turned off, all of the signal outputs of the generator are disabled.

Chapter 3: Built-In Formats

Introduction

The charts on the following pages list the generator's built-in formats library. These are stored in read only memory (ROM) along with the generator's operating code. They can be used as starting points for creating your own formats and new ones can be added to the nonvolatile RAM. A maximum of 300 formats can be stored in RAM.

The same format library is used for all models in the 801G series. Some formats are for displays and graphics systems that may not be compatible with a specific generator model. If a format has too high of a pixel rate, a modified version of the format is placed in RAM with reduced horizontal timing parameters that bring the pixel clock rate below the limit of the generator. The horizontal and vertical rates however are correct. This lets you *light-up* a display and verify most of its operation. These modified formats are created by the memory re-intialization routine. In other cases the library format may have a video type or sync type selection that may not be compatible with a specific generator model. Attempting to load an incompatible format will give you an error message.

The charts give only a general description of each format. More detailed information can be obtained in one of the following ways.

- You can use the *Format* test image to display a detailed list of parameters of any format in firmware or RAM.
- You can use the built-in Graphics User Interface to view and edit the contents of any format. Also you can check, modify, and copy formats with a terminal or computer connected to a generator's communications port.

The information in the charts is believed to be accurate and complete at the time that this manual was wrritten. Last minute firmware changes and new firmware releases may affect the format information stored in EPROM.

Explanation of Terms Used in Charts

File Name	Name of the format file as saved in EPROM
Video Type	C3 = 3-bit digital color C4 = 4-bit digital color (CGA) C6 = 6-bit digital color (EGA) M2 = 2-bit digital monochrome RGB = separate red, green and blue analog color Mono = analog monochrome. EYC = analog color television (w/subcarrier) / S-video (separate lumi and chroma).
Horiz x Vert Active Pixels	Number of active pixels in the horizontal direction and the number of active scan lines in the vertical direction NOTE: The number of active pixels shown is for the original format file as it's stored in ROM. The firmware may reduce this number if the original value would cause the calculated pixel clock rate to exceed the pixel clock limit for the generator.
Line Rate	Horizontal scanning frequency in KHz rounded to 3 places past the decimal
Frame Rate	Picture refresh rate in Hz rounded to 3 places past the decimal A bullet (•) after the vertical rate indicates the number is the frame rate for a 2:1 interlaced format.

Built-in Formats

IBM Digital

File Name	Video Type	Horiz x Vert Active Pixels	Line Rate	Frame Rate
MDA_M7	M2	720 x 350	18.432	49.816
HGC_text	M2	720 x 350	18.141	49.030
HGCgraph	M2	720 x 348	18.519	50.051
CGA_M14	C4	640 x 200	15.700	59.924
EGA_m2	C6	640 x 350	21.851	59.702
IBM_3179	C3	640 x 400	25.560	60.000
IBM_3164	C3	640 x 400	27.648	64.749

AT&T

File Name	Video Type	Horiz x Vert Active Pixels	Line Rate	Frame Rate
AT&T_SVC	C6	640 x 400	25.862	59.866
AT&T_IVC	C6	640 x 400	25.862	59.866
AT&T_EVC	C6	640 x 350	25.862	59.866

IBM Analog

File Name	Video Type	Horiz x Vert Active Pixels	Line Rate	Frame Rate
PGA_400	RGB	640 x 400	30.296	59.638
PGA_480	RGB	640 x 480	30.296	59.638
VGA_m1	RGB	720 x 350	31.469	70.087
VGA_m2	RGB	720 x 400	31.469	70.087
VGA_m3	RGB	640 x 480	31.469	59.941
VGA_m4	RGB	1024 x 768	35.522	43.478¥
XGA_m4a	RGB	1053 x 754	35.414	43.453¥
XGA_m4b	RGB	1056 x 768	35.602	43.470¥
XGA_m5	RGB	1024 x 768	56.287	70.008
XGA_m6	RGB	1360 x 1024	56.469	51.476¥
XGA6475	RGB	640 x 480	39.375	75.000
XGA1076	RGB	1024 x 768	61.080	75.782

IBM Workstation

File Name	Video Type	Horiz x Vert Active Pixels	Line Rate	Frame Rate
IBM6Km1	RGB	1024 x 1024	63.360	60.000
IBM6Km2	RGB	1280 x 1024	63.363	60.002
IBM6Km3	RGB	1280 x 1024	70.755	67.003
IBM6Km4	RGB	1280 x 1024	70.755	67.003

Built-in Formats Ð cont.

VESA (Video Electronics Standards Association)

File Name	Video Type	Horiz x Vert Active Pixels	Line Rate	Frame Rate
VG900601	RGB	800 x 600	35.156	56.250
VG900602	RGB	800 x 600	37.879	60.317
VS900603	RGB	800 x 600	48.077	72.188
VS901101	RGB	640 x 480	37.861	72.809
VG901101	RGB	1024 x 768	48.363	60.004
VG910801	RGB	1024 x 768	56.476	70.069
DMT6475	RGB	640 x 480	37.500	75.000
DMT648A	RGB	640 x 350	37.861	85.080
DMT648B	RGB	640 x 400	37.861	85.080
DMT6485	RGB	640 x 480	43.269	85.008
DMT7285	RGB	720 x 400	37.927	85.083
DMT8075	RGB	800 x 600	46.875	75.000
DMT8085	RGB	800 x 600	53.674	85.061
DMT1075	RGB	1024 x 768	60.023	75.029
DMT1085	RGB	1024 x 768	68.677	84.997
DMT1170	RGB	1152 x 864	63.851	70.012
DMT1175	RGB	1152 x 864	67.500	75.000
DMT1185	RGB	1152 x 864	77.094	84.999
DMT1243	RGB	1280 x 1024	46.433	43.436¥
DMT126A	RGB	1280 x 960	60.000	60.000
DMT1260	RGB	1280 x 1024	63.981	60.020
DMT127A	RGB	1280 x 960	75.000	75.000
DMT1275	RGB	1280 x 1024	79.976	75.025
DMT128A	RGB	1280 x 960	85.938	85.002
DMT1285	RGB	1280 x 1024	91.146	85.024
DMT1648	RGB	1600 x 1200	62.500	48.040¥
DMT1660	RGB	1600 x 1200	75.000	60.000
DMT1665	RGB	1600 x 1200	81.250	65.000
DMT1670	RGB	1600 x 1200	87.500	70.000
DMT1675	RGB	1600 x 1200	93.750	75.000
DMT1680	RGB	1600 x 1200	100.000	80.000
DMT1685	RGB	1600 x 1200	106.259	85.000

Built-in Formats Ð cont.

Macintosh

File Name	Video Type	Horiz x Vert Active Pixels	Line Rate	Frame Rate
MAC_TVus	RGB	640 x 480	15.734	29.970¥
MAC_TVos	RGB	512 x 384	15.734	29.970¥
MAC_12m	Mono	512 x 384	24.480	60.147
MAC_12c	RGB	512 x 384	24.480	60.147
MAC_12ce	RGB	560 x 384	24.480	60.147
MAC_13LC	RGB	640 x 480	34.975	66.619
MAC_13m	Mono	640 x 480	35.000	66.667
MAC_13c	RGB	640 x 480	35.000	66.667
MAC_15	Mono	640 x 870	68.850	75.000
MAC_16	RGB	832 x 624	49.107	75.087
MAC_1960	RGB	1024 x 768	48.193	59.278
MAC_19	RGB	1024 x 768	60.241	74.927
MAC_21	RGB	1152 x 870	68.681	75.062

Japanese NEC

File Name	Video Type	Horiz x Vert Active Pixels	Line Rate	Frame Rate
NECPC400	RGB	640 x 400	24.823	56.416
NECPC750	RGB	1120 x 750	32.857	40.021¥

Sun Microsystems Workstation

File Name	Video Type	Horiz x Vert Active Pixels	Line Rate	Frame Rate
SUN1061	RGB	1024 x 1024	65.267	61.399
SUN1077	RGB	1024 x 768	62.040	77.069
SUN1166	RGB	1152 x 900	61.796	65.950
SUN116B	RGB	1152 x 900	61.846	66.004
SUN1176	RGB	1152 x 900	71.713	76.047
SUN117B	RGB	1152 x 900	71.809	76.149
SUN1267	RGB	1280 x 1024	71.722	66.718
SUN126B	RGB	1280 x 1024	71.678	66.677
SUN1276	RGB	1280 x 1024	81.130	76.107
SUN1667	RGB	1600 x 1280	89.286	66.931

Built-in Formats Ð cont.

Hewlett Packard

File Name	Video Type	Horiz x Vert Active Pixels	Line Rate	Frame Rate
HP1060	RGB	1024 x 768	47.700	60.000
HP1070	RGB	1024 x 768	56.476	70.069
HP1075A	RGB	1024 x 768	62.937	74.925
HP1075B	RGB	1024 x 768	60.241	75.020
HP1260	RGB	1280 x 1024	63.338	59.979
HP1272	RGB	1280 x 1024	78.125	72.005
HP1275	RGB	1280 x 1024	79.976	75.025

Japanese Sony Monitor

File Name	Video Type	Horiz x Vert Active Pixels	Line Rate	Frame Rate
SON1072	RGB	1024 x 768	57.870	71.799
SON1274	RGB	1280 x 1024	78.855	74.112
SON1276	RGB	1280 x 1024	81.206	76.179

Intercolor Workstation

File Name	Video Type	Horiz x Vert Active Pixels	Line Rate	Frame Rate
INT1160	RGB	1184 x 884	55.200	60.000
INT1176	RGB	1184 x 884	71.712	76.047
INT1660	RGB	1664 x 1248	77.940	60.00
INT1676	RGB	1664 x 1248	100.73	76.020

Barco

File Name	Video Type	Horiz x Vert Active Pixels	Line Rate	Frame Rate
BAR2060	RGB	2048 x 2048	126.86	60.008
BAR2080	RGB	2048 x 1536	126.86	79.187
BAR2560	RGB	2560 x 2048	126.91	60.034

PAL Component Video

File Name	Video Type	Horiz x Vert Active Pixels	Line Rate	Frame Rate
PAL_Y	RGB	920 x 574	15.625	25.000¥
PAL_Yus	RGB	768 x 575	15.625	25.000¥
PAL_Yos	RGB	640 x 480	15.625	25.000¥

RS 170 Video

File Name	Video Type	Horiz x Vert Active Pixels	Line Rate	Frame Rate
RS170Y	RGB	752 x 484	15.734	29.970¥
RS170Yus	RGB	640 x 480	15.734	29.970¥
RS170Yos	RGB	512 x 384	15.734	29.970

Built-in Formats Ð cont.

PAL Encoded Video

File Name	Video Type	Horiz x Vert Active Pixels	Line Rate	Frame Rate
PAL_4xSC	EYC	910 x 574	15.625	25.000¥
PALTV601	EYC	720 x 574	15.625	25.000¥
PAL_TVus	EYC	768 x 574	15.625	25.000¥
PAL_TVos	EYC	640 x 480	15.625	25.000¥
PAL_N	EYC	910 x 574	15.625	25.000¥

NTSC Encoded Video

File Name	Video Type	Horiz x Vert Active Pixels	Line Rate	Frame Rate
NTSC_443	EYC	752 x 484	15.734	29.970¥
NTSC4xSC	EYC	752 x 484	15.734	29.970¥
NTSC_601	EYC	720 x 484	15.734	29.970¥
NTSCTVus	EYC	640 x 480	15.734	29.970¥
NTSCTVos	EYC	512 x 384	15.734	29.970¥

HDTV Component Video

File Name	Video Type	Horiz x Vert Active Pixels	Line Rate	Frame Rate
HDTV_1J	RGB‡	1920 x 1035	33.750	30.000¥
HDTV_2J	RGB‡	1920 x 1035	33.750	30.000¥
HDTV_4J	RGB‡	1920 x 1035	33.750	30.000¥
HDTV_1E	RGB‡	1920 x 1152	31.250	25.000¥
HDTV_2E	RGB‡	1920 x 1152	31.250	25.000¥
HDTV_4E	RGB‡	1872 x 1152	31.250	25.000¥

Generator Diagnostics

File Name	Video Type	Horiz x Vert Active Pixels	Line Rate	Frame Rate
TEST150	RGB	2048 x 1024	50.403	46.887
TEST250	RGB	2048 x 2048	79.719	35.861¥

Chapter 4: Built-In Images

Description of the test images

and how to use them

Introduction

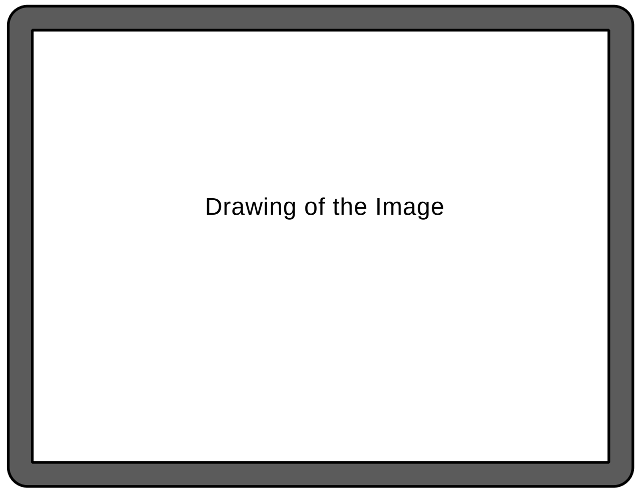
This chapter covers all of the built-in test images (patterns) in a standard model 801G* generator. It is also possible to add custom, user defined test images to thegenerator. Information on modifying and adding custom test images can be found in the “Programming” chapter.

Information on how to select the various images can be found in the “Basic Operation” chapter of this manual. The “Troubleshooting” chapter has information on resetting the 801GX to its factory default conditions.

The remainder of this chapter describes each of the images in detail. The purpose of each image is included in the description. The images are presented in the same order as they are in the table. Most of the images are presented in the format shown on the next page.

ImageName

Description — This tells how the image is drawn on the display. A black and white drawing of the image is included as part of the description.



Test — Name of the test to be done

The name of the test describes the type of test to be done.

Purpose — Why this type of test should be done

Method — A general guide on how to perform the test

Descriptions of the Images

Acer1

Description Special test image specified by some display manufacturers. Consists of two sets of color bars and five blocks of “#” characters on a white crosshatch witha black background.

Acer2

Description Special test image specified by some display manufacturers. Consists of colorbars, lines of “#” characters and a green border.

Acer3, Acer4, Acer5 and Acer6

Description Special test images specified by some display manufacturers. Consists of a large and small white circles centered on either a yellow (Acer3), magenta (Acer4), cyan (Acer5) or white (Acer6) crosshatch on a black background.

Acer7 and Acer 8

Description Special test image specified by some display manufacturers. Consists of five blocks of either white “#” (Acer7) or “H” (Acer8) characters on a black background.

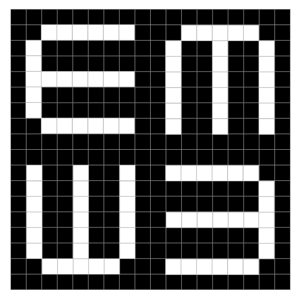
Acer9

Description Special test image specified by some display manufacturers. Consists of mostly a white field with two rows of color bars at the bottom.

BLU_EM, GRN_EM, RED_EM, WHT_EM, MEMESony, MESony_B, MESony_G, and MESony_R

Description

In the primary version, the screen is filled with blue (BLU and B), green (GRN and G), red (R), or white (WHT and MEMESony) EM characters on a black background. A bit map of a single character is shown here.



Only the white character has a secondary version. It’s drawn with black characters on a white background.

Test

Focus

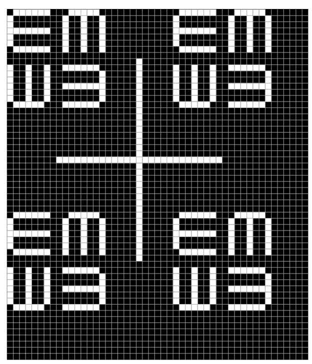
Purpose

This pattern is specified by one or more display manufacturers for checking and adjusting focus on their products one color at a time.

BLU_EM+, GRN_EM+, RED_EM+, WHT_EM+, MEMEPlus, MEPlus_B, MEPlus_G, and MEPlus_R

Description

In the primary version, the screen is filled with blue (BLU and B), green (GRN and G), red (R), or white (WHT and MEMEPlus) EM character block on a black background. A bit map of a single character block is shown here.



Only the white character has a secondary version. It’s drawn with black characters on a white background.

Test

Focus

Purpose

This pattern is specified by one or more display manufacturers for checking and adjusting focus on their products one color at a time.

BLU_PIC, GRAY_PIC, GRN_PIC, RED_PIC, WHT_PIC, Flat, Flat Gray, Flat_B, Flat_G, and Flat_R

Description

A solid blue (BLU), gray, green (GRN), red, or white (WHT) box fills the active video area.

Only the white fill has a secondary version. It can be changed to a black fill.

Test

Purity adjustment

Purpose

To produce correct colors in a displayed image, the electron beams from each of the three (3) guns in the CRT should strike only their matching phosphors. A white image shows patches of various colors on a monitor with bad purity. The purity adjustment(s) should be performed before doing any brightness or color tests. In some cases, purity adjustments involve loosening and repositioning the yoke, in which case purity should be adjusted prior to doing any geometry tests.

BLU_PIC, GRAY_PIC, GRN_PIC, RED_PIC, WHT_PIC, Flat, Flat Gray, Flat_B, Flat_G, and Flat_R — contd.

Method

The methods used for adjusting purity on a color monitor depend on the type of monitor and CRT you're using (for example; Delta, In-Line or Single Gun). In most cases, the first step is to degauss the CRT.

Note – For a Delta Gun CRT, turn on only the Red output. A solid uniform field of red should be displayed. If the color is not uniform, adjust the yoke and the Purity Tabs assembly.

If purity cannot be corrected to acceptable limits, the monitor may not have been properly degaussed or there may be a defect in the CRT or purity assembly.

Test

Shadow mask warping

Purpose

The purity characteristics of your CRT can change over time if you leave it on with a lot of video being displayed. This may be due to the CRT's electron beams striking its shadow mask with enough energy to cause the mask to heat. This internal heating may be enough to cause the shadow mask to warp and give bad purity.

Method

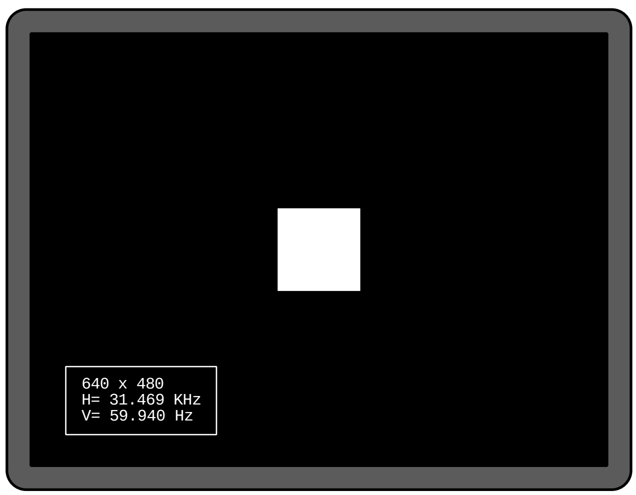
Set the purity image to white and allow the monitor to run for a few minutes. Any mask warping shows up as a change in purity. You can use a color meter to measure the change. The *BriteBox* pattern also may be useful for measuring shadow mask warping.

Box_50mm and Box_64mm

Description

The primary version has a solid white box in the center of the active video. Depending on the image selected, the box is either 50 millimeters (1.97 inches) or 64 millimeters (2.52 inches) square. If there's room, information on the current format appears below and to the left of the box. This shows the number of active pixels and lines as well as the horizontal and vertical scan rates. An *I* after the number of active lines indicates the format is interlaced. The secondary version draws a black box and black text on a white background.

Note – The box will be the correct size only if the correct physical active video size is set in the format.



Test

Brightness control adjustment

Purpose

The wrong brightness setting may cause other tests such as *Contrast*, *Focus* and *Beam Size* to be invalid. An accurate brightness setting helps give repeatable measurements throughout other tests.

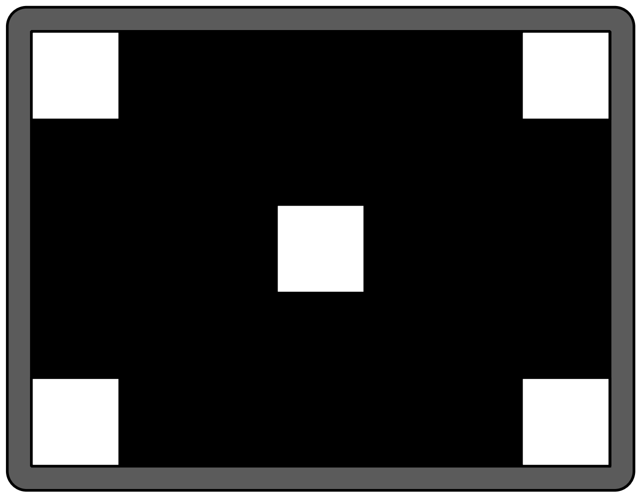
Method

Center your light meter probe within the center square and adjust the monitor's brightness control to obtain the required light meter reading.

BriteBox

Description

The primary version has a single white box in the center of active video. The box size is controlled by the MSIZ system parameter. The secondary version (shown below) adds four boxes in the corners of active video.



Test

Brightness control adjustment

Purpose

The wrong brightness setting may cause other tests such as *Contrast*, *Focus* and *Beam Size* to be invalid. An accurate brightness setting helps give repeatable measurements throughout other tests.

Method

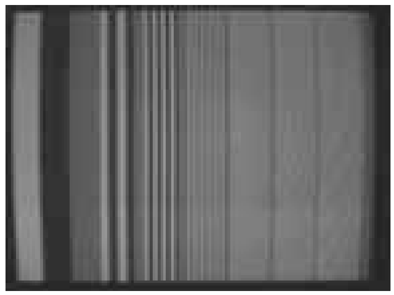
Center your light meter probe within the center square and adjust the monitor's brightness control to obtain the required light meter reading.

BriteBox — contd.

Test	Brightness uniformity
Purpose	The light output of most picture tubes varies slightly when measured across the CRT face. This test can be used to verify that the light output variation is within your spec limits.
Method	Select the inverted version and perform the *Brightness Control Adjustment* test on the center box. Then center the light meter probe in each of the corner squares and note the reading you get for each square. The deviation between each of the corner readings and the center reading should be within your spec limits.

Burst (TV formats only)

Description: The left side start with reference white (+100 IRE) and black (+7.5 IRE) levels. This is followed by six bursts of sine waves. Each burst is at a different frequency forming vertical lines of various widths. The frequencies, going from left to right, are 0.5, 1,



2, 3, 3.58 and 4.43 MHz.

Test: Frequency Response

Method: When viewed on a TV screen, the peak intensities of the all of the bursts should match the white reference level. The darkest portions between the peaks should match the black reference level.

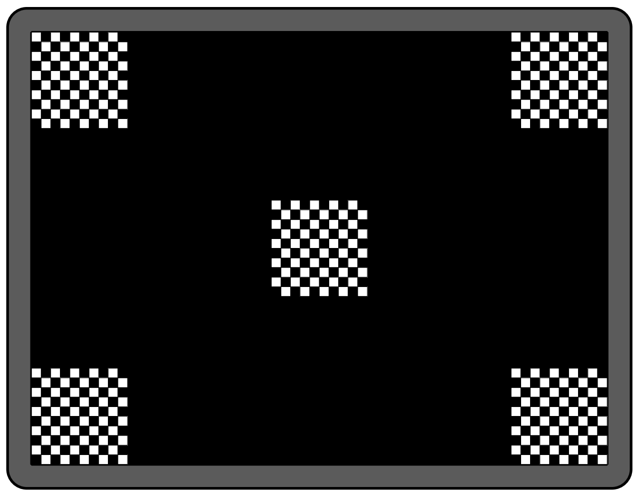
The image can also be used with a TV waveform analyzer to check the frequency response of a video system. One scan line of the image, as it would appear on a waveform analyzer, is shown at the top of the next page. High frequency roll-off (loss) would show up as a decrease in the peak-to-peak swings on the right side of the waveform. Low frequency roll-off would show up as a decrease in the peak-to-peak swings on the left side of the waveform.

Some waveform analyzers can be set to detect and display the amplitude of the peaks. A typical amplitude waveform for a good system is shown at the bottom of the next page.

Check511

Description

Five small boxes are placed in the corners and at the center of active video. The boxes are on a black background. Each box consists of alternating black and white pixels that form a very fine checkerboard. The secondary version inverts the image, creating a white background. The colors of the individual pixels in the boxes also are inverted.



Test

Verify monitor resolution

Purpose

The resolution of your monitor should meet or exceed the design specs.

Method

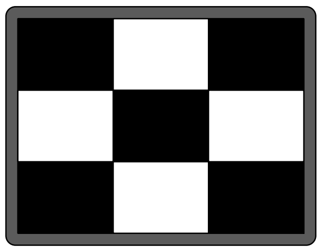
First adjust the brightness, contrast, and focus to their correct settings. You should be able to see individual and distinct pixels in each of the boxes. Failure to see distinct pixels may indicate you have a defective video amplifier, focus correction circuit or picture tube.

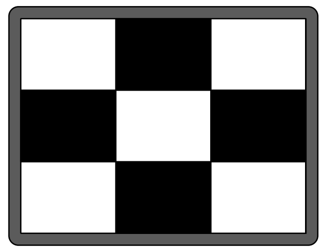
Note – If multicolored areas appear on a mask-type color picture tube, you may have a problem with convergence or you may be exceeding the resolution of the picture tube.

CheckBy3

Description

The active video area is equally divided into a three by three checkerboard of black and white boxes. The primary version has four white boxes as shown in the figure on the left. The secondary version has five white boxes as shown in the figure on the right.





Test

Contrast ratio

Purpose

The pattern is based on a proposed ANSI method of measuring the contrast ratio of video projection systems.

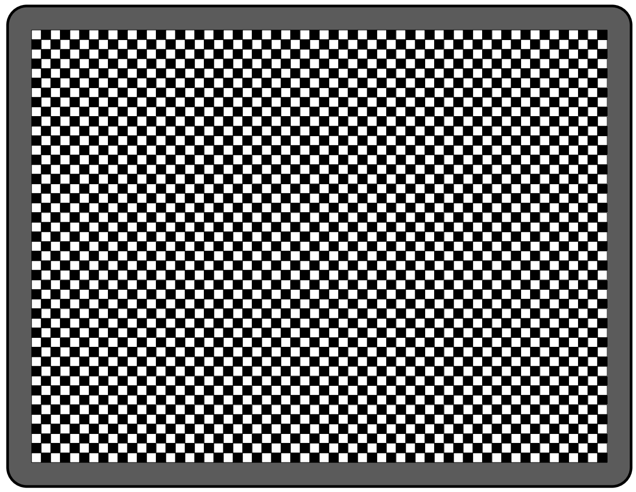
Method

Using a suitable light meter probe, measure and record the light-level reading (in foot lamberts) in the center of each of the black and white boxes. The contrast ratio is expressed as the average of all of the white readings divided by the average of all of the black readings.

Check_11

Description

The active video area is filled with alternating black and white pixels that form a very fine checkerboard. The secondary version inverts the colors in the image. The inverted image looks almost the same as the non-inverted version.



Test

Verify monitor resolution

Purpose

The resolution of your monitor should meet or exceed the design specs.

Method

Adjust the brightness, contrast, and focus to their correct settings first. You should be able to see individual and distinct pixels in each of the boxes. Failure to see distinct pixels may indicate you have a defective video amplifier, focus correction circuit or picture tube.

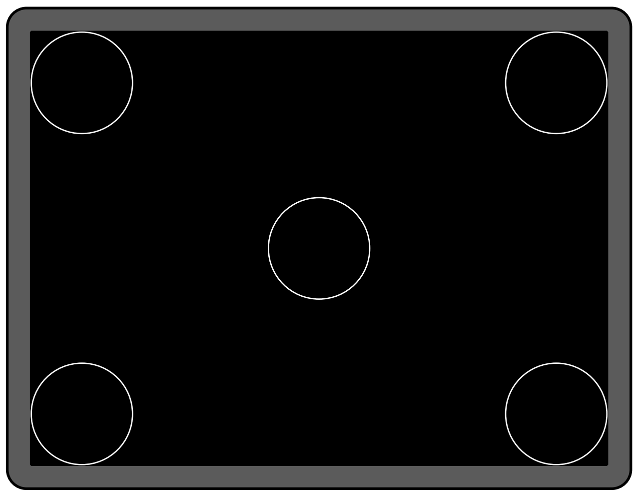
Note – If multicolored areas appear on a mask-type color picture tube, you may have a problem with convergence or you may be exceeding the resolution of the picture tube.

CirclesL

Description

This image may be called for by some display manufacturers' test procedures. The image consists of five large white circles on a black background. The circles are positioned in the center and in the corners of the active video area.

The secondary version inverts the image to black circles on a white background.



Purpose

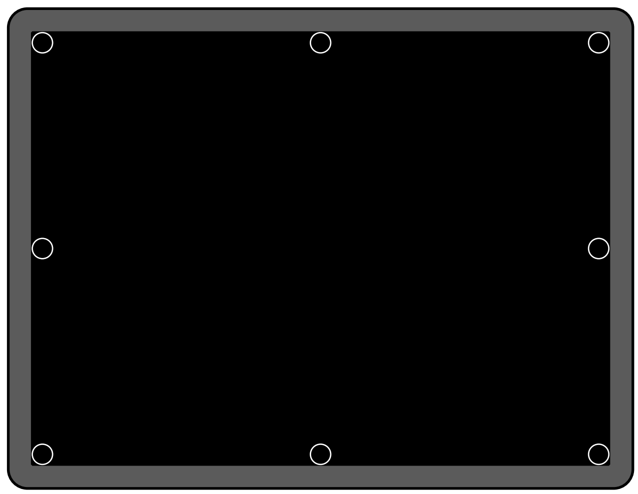
This pattern is specified by one or more monitor manufacturers for checking and adjusting video scan size, linearity and over scanning.

CirclesS

Description

This image may be called for by some display manufacturers' test procedures. The image consists of eight small white circles on a black background. The circles are positioned in the corners of the active video area and centered on each edge of the active video area.

The secondary version inverts the image to black circles on a white background.



Purpose

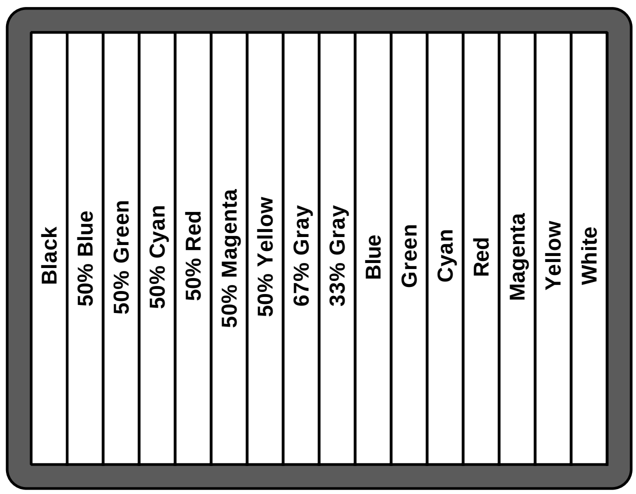
This pattern is specified by one or more monitor manufacturers for checking and adjusting video scan size, linearity and over scanning.

ColorBar

Description

The primary version has 16 full-height vertical color bars. The order of the bars is shown below. The secondary version splits the field into a top and bottom half. The bars in the bottom half of the screen are in reverse order.

When digital video is being output, 33% Gray changes to 50% Gray and 67% Gray becomes either Black or some gray level depending on how the display interprets the video information.



Test

Verify that all video channels is functional

Purpose

To verify that none of the video channels are bad or hooked up incorrectly

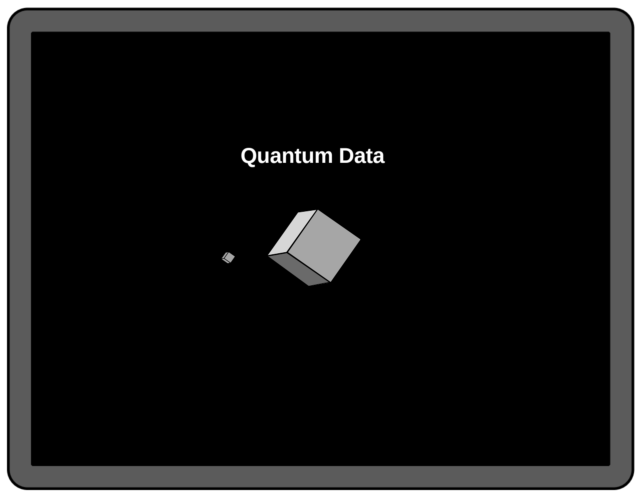
Method

Compare the sequence of color bars with the table. Missing bars may indicate a dead or unconnected channel. The transition between the bars should be sharp and distinct. Each bar also should be uniform in color and intensity across its entire width. Non-uniformity may indicate problems with the response of the video amplifiers. If all the bars are present but in the wrong order, one or more inputs may be swapped.

Cubes

Description

This is an animated image consisting of one small multicolored cube orbiting around a larger multicolored cube. Each cube also is spinning on its own axis. The default text string says *Quantum Data*. The text can be modified and saved using commands sent over the communications ports. The primary version has a black background and a thick green border. The secondary version uses just a white background.



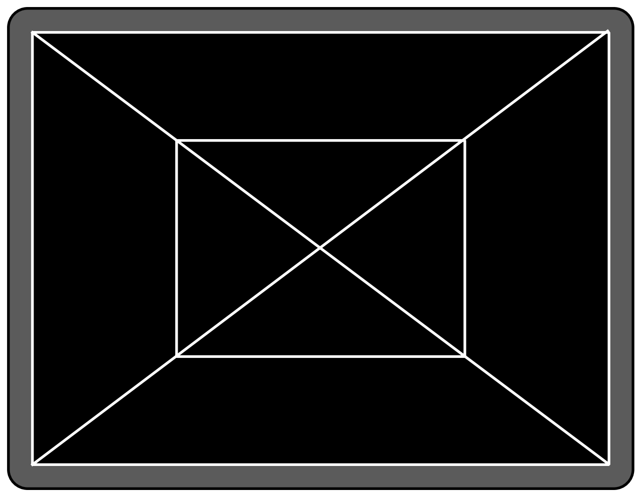
Purpose

Can be used for show demonstrations with your own text.

Custom

Description

This image has a white border around the active video, a centered smaller yellow box, and green diagonals.



Purpose

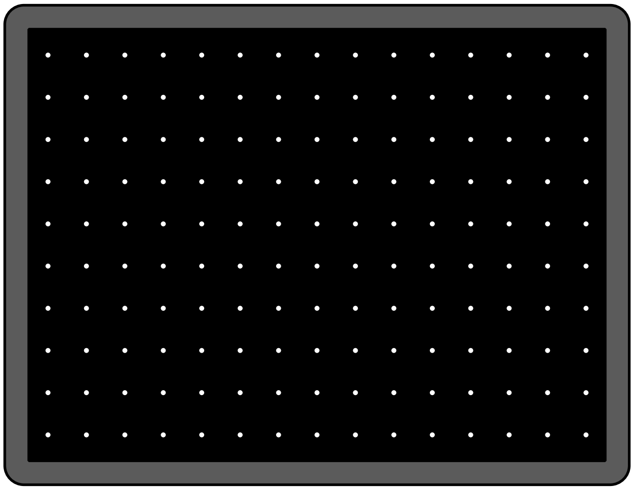
This image is an example of some of the available drawing primitives. It's not intended to be an image suitable for testing or adjusting a display. Rather, it can be used as a starting point for developing a custom image of your own.

Dot_10, Dot_12, Dot_24

Description

The active video area is filled with multiple rows of white single pixel dots. The dots define the corners of what would appear to be square boxes if all the connecting pixels were lit. The number of rows of boxes and the number of boxes per row depends on which version of the image is selected and the screen aspect ratio of the currently loaded format. The number in the image's name refers to the number of boxes that will be formed along the minor axis for most aspect ratios. The firmware calculates the ratio and then finds the closest match from the following table.

Aspect Ratio		Dot_10		Dot_12		Dot_24	
W : H	Decimal	Number of Rows	Boxes per Row	Number of Rows	Boxes per Row	Number of Rows	Boxes per Row
16 : 9	1.777É	10	16	10	16	18	32
5 : 3	1.666 É	10	16	10	16	18	30
4 : 3	1.333 É	10	14	12	16	24	32
1 : 1	1.000	10	10	12	12	24	24
3 : 4	0.750	14	10	16	12	32	24



Dot_10, Dot_12, Dot_24 — contd.

Purpose

In order to accurately produce an image on a color monitor, the three electron beams in the CRT must meet (converge) at the exact same location at the same time. Small dots displayed on a misconverged monitor appear as a group of multicolored dots.

Method

The convergence adjustments of most color monitors fall into two main categories.

The first set of adjustments, usually called *Static Convergence*, aligns the three beams in the center of the display. The idea is to turn on all three guns and adjust the various magnets on the convergence assembly to produce all white dots in the center of the display. The convergence assembly is located on the neck of the CRT. Different monitors and CRT types may each require their own magnet-adjustment sequence.

After the center of the display is properly converged, the outer areas are adjusted by using the monitor's *Dynamic Convergence* controls. The number of controls, the area of the screen they affect, and their adjustment procedure depends on the monitor you're testing.

Test

Focus adjustment(s)

Purpose

An out-of-focus monitor displays fuzzy pixels which, in turn, result in poorly formed and hard-to-read characters.

Method

On monitors with a single (static) focus adjustment, adjust the control for the best average focus over the entire screen. The focus at certain locations should be within specified limits.

Some monitors have a static and one or more dynamic focus controls. The sequence for adjusting them and the areas of the screen they affect depend on the monitor you're testing.

EMITest1

Description

Special test image used for Electro-Magnetic Interference (EMI) testing of displays. The entire active video area is filled with a small “H” character. The primary version of the image draws white characters on a black background. The secondary version draws black characters on a white background.

EMITest2

Description

Same as EMITest1 but with the bottom row of characters contstantly being drawn left to right and then cleared.

EMITest3

Description

Same as EMITest1 but with a smaller version of the “H” character.

EMITest4

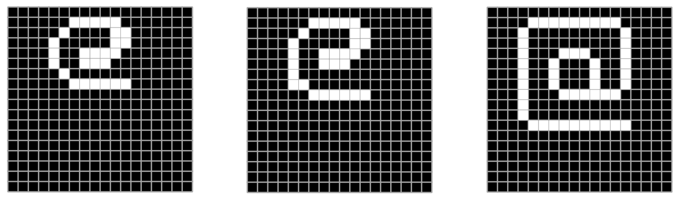
Description

Same as EMITest2 but with a smaller version of the “H” character.

Focus_@6, Focus_@7, Focus_@8

Description

In the primary versions, the screen is filled with white “@” characters on a black background. Bit maps of a single character for the three different images are shown here.



The secondary versions are drawn with black characters on a white background.

Test

Focus adjustment(s)

Purpose

An out-of-focus monitor displays fuzzy graphic images and poorly formed, hard-to-read text characters.

Method

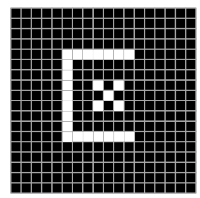
On monitors with a single (static) focus adjustment, adjust the control for the best average focus over the entire screen. The focus at certain locations of the screen should be within specified limits.

Some monitors have a static and one or more dynamic focus controls. The sequence for adjusting them and the areas of the screen that they affect depend on the monitor you’re testing.

Focus_Cx

Description

In the primary version, the screen is filled with white Cx characters on a black background. A bit map of a single character is shown here.



The secondary version is drawn with black characters on a white background.

Test

Focus adjustment(s)

Purpose

An out-of-focus monitor displays fuzzy graphic images and poorly formed, hard-to-read text characters.

Method

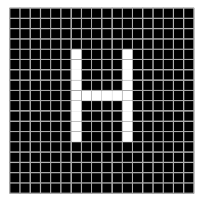
On monitors with a single (static) focus adjustment, adjust the control for the best average focus over the entire screen. The focus at certain locations of the screen should be within specified limits.

Some monitors have a static and one or more dynamic focus controls. The sequence for adjusting them and the areas of the screen that they affect depend on the monitor you're testing.

Focus_H

Description

In the primary version, the screen is filled with white H characters on a black background. A bit map of a single character is shown here.



The secondary version is drawn with black characters on a white background.

Test

Focus adjustment(s)

Purpose

An out-of-focus monitor displays fuzzy graphic images and poorly formed, hard-to-read text characters.

Method

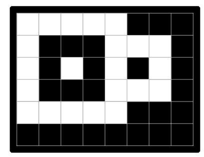
On monitors with a single (static) focus adjustment, adjust the control for the best average focus over the entire screen. The focus at certain locations of the screen should be within specified limits.

Some monitors have a static and one or more dynamic focus controls. The sequence for adjusting them and the areas of the screen that they affect depend on the monitor you’re testing.

Focus_Oo

Description

In the primary version, the screen is filled with white Oo characters on a black background. A bit map of a single character is shown here.



The secondary version is drawn with black characters on a white background.

Test

Focus adjustment(s)

Purpose

An out-of-focus monitor displays fuzzy graphic images and poorly formed, hard-to-read text characters.

Method

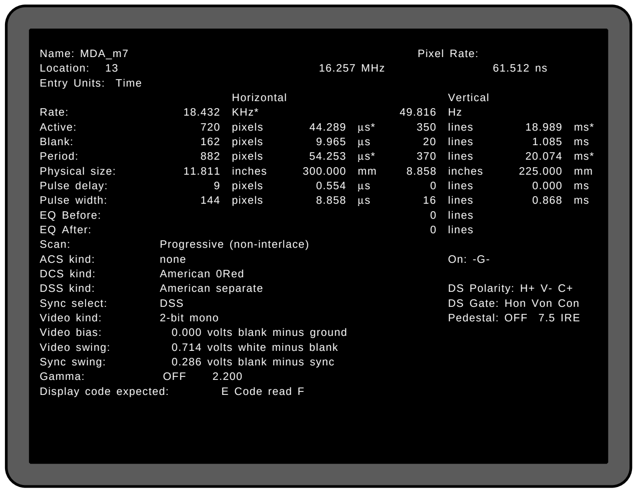
On monitors with a single (static) focus adjustment, adjust the control for the best average focus over the entire screen. The focus at certain locations of the screen should be within specified limits.

Some monitors have a static and one or more dynamic focus controls. The sequence for adjusting them and the areas of the screen that they affect depend on the monitor you're testing.

Format

Description

A listing of the data contained in any format. This pattern works best at display resolutions of at least 640 pixel by 480 lines. It's quite similar to the format editor's GUI screen.



Test

View a format's parameter settings

Purpose

To verify programmed values or choose a format

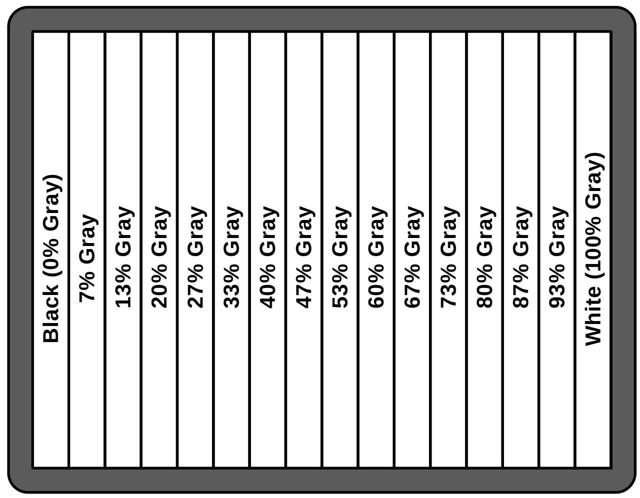
Method

The main image lists the settings of the format that's driving the display. The secondary image can be used to list the contents of any stored format.

GrayBar

Description

The primary version has 16 full-height vertical graybars. The intensity of the bars is shown below. The secondary version splits the field into a top and bottom half. The bars in the bottom half of the screen are in reverse order.



Test — Video color tracking (color monitors)

Purpose — To check to see that a color monitor accurately reproduces colors at all intensities

Method — Perform the *Brightness Control Adjustment* and *Brightness Uniformity* tests first.

Changes in brightness from bar to bar should be uniform. All of the bars should appear as an untinted gray at all levels.

Test — Video Gain Linearity (monochrome monitors)

Purpose — To check the video linearity; i.e., grayscale modulation

Method — Perform the *Brightness Control Adjustment* and *Brightness Uniformity* tests first.

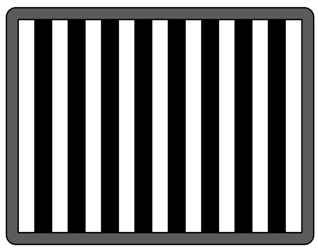
Changes in brightness from bar to bar should be visible and uniform.

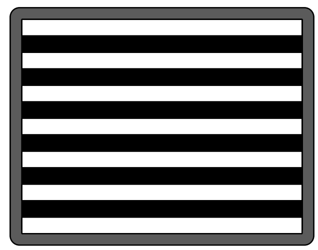
Grill_11, Grill_22, Grill_33, Grill_44

Description

The entire active video area is filled with alternating black and white stripes. The stripes are drawn at different resolutions. Each of the stripes is four (4) pixels wide in the *Grill_44* image and three (3) pixels wide in the *Grill_33* image. Each of the stripes is two (2) pixels wide in the *Grill_22* image and one (1) pixel wide in the *Grill_11* image.

The primary versions draw vertical stripes while the secondary versions draw horizontal stripes.





Test

Verify monitor resolution

Purpose

The resolution of your monitor should meet or exceed the design specs.

Method

First adjust the brightness, contrast, and focus to their correct settings. You should be able to see individual and distinct stripes in all areas of the display at all four resolutions. Failure to see distinct lines at the highest resolution (Grill_11) may indicate you have a defective video amplifier or picture tube.

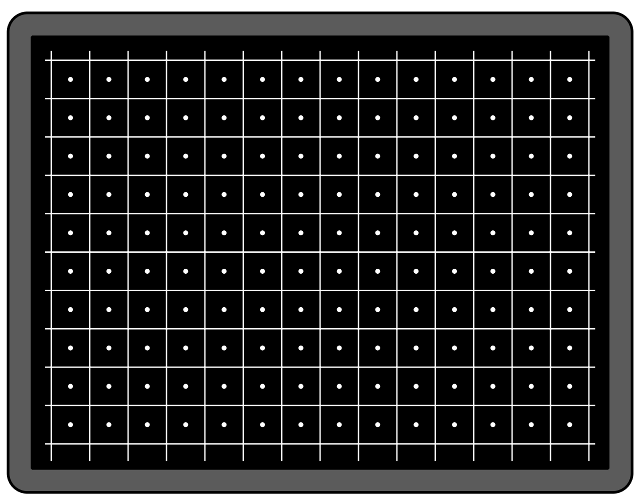
Note – If multicolored lines appear on a mask-type color picture tube, you may have a problem with convergence or you may be exceeding the resolution of the picture tube.

Hatch_10i, Hatch_10o, Hatch_12i, Hatch_12o, Hatch_24i, Hatch_24o, Hatch_24s, Hatch_G, Hatch_M, GRN_HTCH, and MAGENTA

Description

The primary versions consist of a white, green (G and GRN), or magenta (M) crosshatch drawn on a black background. The lines form square boxes. A single pixel dot is located in the center of each crosshatch box. The number of boxes formed depends on the version of the image selected and the screen aspect ratio of the currently loaded format. The number in the image's name refers to the number of boxes that are formed along the minor axis for most aspect ratios. The firmware calculates the ratio and then finds the closest match from the table on the next page. Versions ending in *i* draw from the inside (center) out. Any partial boxes are placed around the perimeter of the image. Versions ending in *o* draw from the outside in. Any partial boxes are placed along the centerlines of the image. Versions ending in *s* are the "i" version plus a 1 pixel thick border.

The secondary versions invert the images to black lines and dots on a white background. Hatch_G, Hatch_M, GRN_HTCH and Magenta do not have secondary versions.



Aspect Ratio		Hatch_10		Hatch_12		Hatch_24	
W : H	Decimal	Boxes Vertically	Boxes Horizontally	Boxes Vertically	Boxes Horizontally	Boxes Vertically	Boxes Horizontally
16 : 9	1.777É	10	16	10	16	18	32
5 : 3	1.666 É	10	16	10	16	18	30
4 : 3	1.333 É	10	14	12	16	24	32
1 : 1	1.000	10	10	12	12	24	24
3 : 4	0.750	14	10	16	12	32	24

Test — Convergence adjustment (color monitors only)

Purpose — In order to accurately produce an image on a color monitor, the three electron beams in the CRT must meet (converge) at the exact same location at the same time. Lines displayed on a misconverged monitor appear as several multicolored lines and the transitions between different colored areas contain *fringes* of other colors.

Method — The convergence adjustments of most color monitors fall into two main categories.

The first set of adjustments, usually called *Static Convergence*, aligns the three beams in the center of the display. The idea is to turn on all three guns and adjust the various magnets on the convergence assembly to produce all white dots in the center of the display. The convergence assembly is located on the neck of the CRT. Different monitors and CRT types may each require their own magnet adjustment sequence.

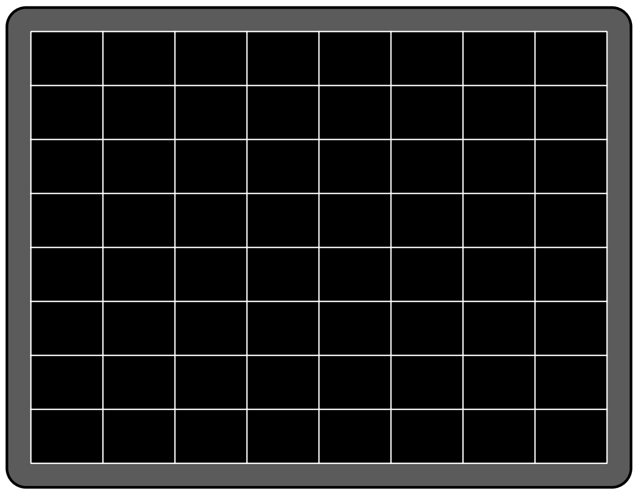
After the center of the display is properly converged, the outer areas are adjusted by using the monitor's *Dynamic Convergence* controls. The number of controls, the area of the screen they affect, and their adjustment procedure depends on the monitor you're testing.

Hatch4x3, Hatch5x4 and Hatch8x8

Description

These are different versions of a crosshatch pattern that may be called for by some display manufacturers' test procedures. The primary version consists of white crosshatch on a black background.

The secondary version inverts the image to black lines on a white background.



Purpose

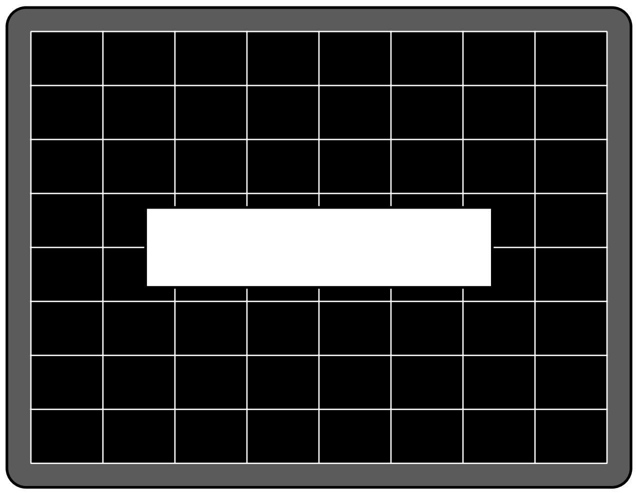
This is a general purpose test image that can be used to check and adjust video scan linearity and geometry and color convergence.

Hatch64W

Description

This is still another version of a crosshatch pattern that may be called for by some manufacturers' test procedures. The primary version consists of an 8 by 8 white crosshatch on a black background. A white rectangular patch is added in the center.

The secondary version inverts the image to black lines and box on a white background.



Purpose

This is a general purpose test image that can be used to check and adjust video scan linearity and geometry and color convergence. The large white rectangle also allows for checking a display's high voltage regulation. This is done by observing the vertical lines at the left and right edges of the image. They should be fairly straight and not pull in the area of the white rectangle.

Hitachi1

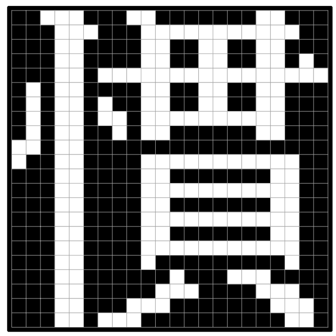
Description

This is a special test image specified by some display manufacturers. The image consists of a 2x2 cluster of Microsoft Windows® program manager screen simulations using Japanese characters.

KanjiKan

Description

In the primary version, the screen is filled with white Japanese Kan characters on a black background.



The secondary version is drawn with black characters on a white background.

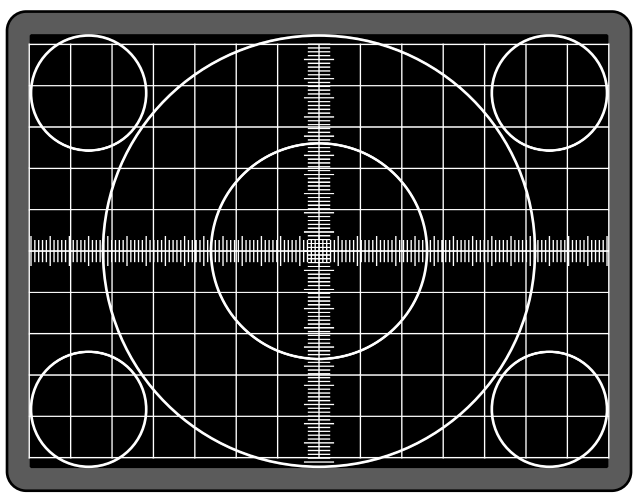
Test

Focus adjustment(s)

Linearty (Linearity)

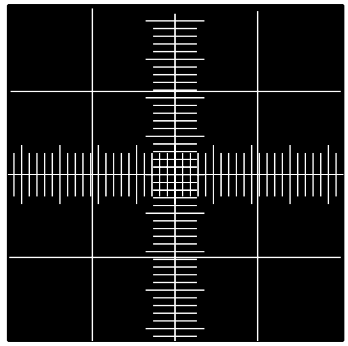
Description

The image is made up of three parts. The first part consists of six (6) white circles. A large circle is drawn in the center of the screen. Its diameter equals the lesser of the video height or width of the display. A smaller circle is drawn at half the diameter and concentric with the larger circle. A circle also is drawn in each of the corners of the screen. The diameter of the corner circles equals one-fifth of the display width. The second part of the image consists of a white crosshatch image. The number of boxes in the crosshatch depends on the physical size of the display. The last part of the image consists of white *tic* marks on the horizontal and vertical center lines of the image. The marks are one pixel thick and at every other pixel location. Every fifth mark is slightly longer. The color of the pattern can be changed with the individual video output controls.



Linearty (Linearity) — contd.

Detail showing center of linearity test image. All lines are one pixel thick.



Test Linearity adjustment

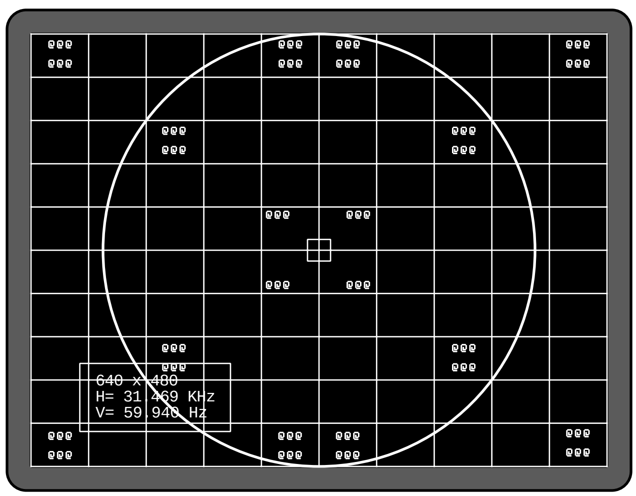
Purpose In order to present an undistorted display, the horizontal and vertical sweeps of the electron beam across the face of the CRT should be at uniform speeds. Any non-uniformity in the sweep causes portions of an image to stretch while other portions are compressed. Non-linearity in a monitor shows up in several ways. It may be present across the entire screen, in a large portion of the screen, or localized in a very small area.

Method The circles in the image can be used to do a ball park adjustment of a monitor's linearity controls. Adjust the controls to form perfectly round circles. The crosshatch image can be used to measure linearity and to make finer control adjustments. All the full boxes in the crosshatch should be identical in size. Measure them with a ruler or a gauge made for the monitor you're testing. Any deviation should be within your spec limits. Use the tic marks and a ruler or gauge to measure linearity over a small portion of the display. Compare the number of tic marks per unit of measure with an adjacent or overlapping area.

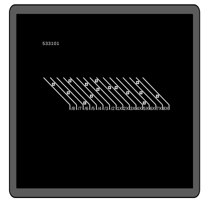
LinFocus

Description

The image consists of several parts. It starts with a large circle in the center of the screen. Its diameter equals the lesser of the video height or width of the display. The second part is a 10 by 10 box crosshatch pattern. The crosshatch is drawn in from the outside edges, with any extra pixels in the boxes placed along the vertical and horizontal axis. The vertical centerline is two pixels thick if the format has an even number of active pixels per line. The horizontal centerline is two pixels thick if the format has an even number of active lines per frame. A smaller box is added at the center of the image. The box is one-half the height and two-fifths the width of one of the crosshatch boxes. Current format data is shown in the lower left quadrant of the image. It shows the number of active pixels (H) and lines (V) as well as the vertical and horizontal scan rates.



LinFocus — contd.



The image also includes blocks of focus-checking characters at various locations. The blocks are positioned inside the crosshatch boxes and are up to 3 by 3 characters in size. The size of the blocks is limited by the number of characters that can fit in one box. The bit map of a single focus character is shown here.

The primary version consists of a white pattern on a black background. The secondary version has a black pattern on a white background.

Test Linearity adjustment

Please see the discussion of the *Linearity* test image for information on measuring linearity.

Test Focus adjustment(s)

Purpose An out-of-focus monitor displays fuzzy graphic images and poorly formed, hard-to-read characters when text is displayed on the screen.

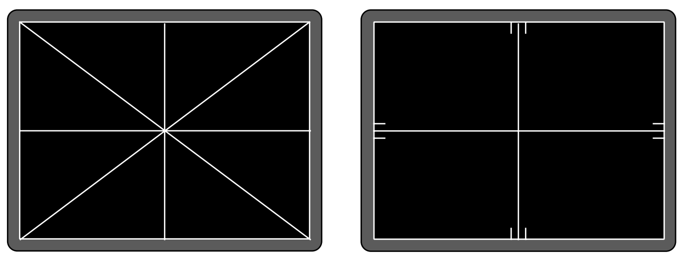
Method On monitors with a single (static) focus adjustment, adjust the control for the best average focus over the entire screen. The focus at certain locations of the screen should be within specified limits.

Some monitors have a static and one or more dynamic focus controls. The sequence for adjusting them and the areas of the screen they affect depend on the monitor you're testing.

Outline0 and Outline1

Description

The Outline0 image consists of a rectangular white border on a black background. The border is one (1) pixel wide and defines the active video area. Two (2) diagonal lines join the opposite corners. A full size cross is centered in the image. The horizontal line of the cross is one (1) pixel thick for formats with an odd number of active lines and two (2) pixels thick for formats with an even number of active lines. The vertical line of the cross is one (1) pixel thick for formats with an odd number of active pixels per line and two (2) pixels thick for formats with an even number of active pixels.



In the Outline1 version, the two diagonal lines are removed and short marker lines are added to the border lines near to where the cross lines meet the border lines. The markers appear at both sides of the cross lines. The distance between the marker lines and the cross lines is the greater of either two (2) pixels or one (1) millimeter.

Outline0 and Outline1 — contd.

Test	Yoke tilt correction
Purpose	The horizontal axis of a displayed image should line up with the horizontal axis of your monitor. Any tilt is likely due to the yoke being rotated on the neck of the CRT. A rotated yoke makes any displayed image appear rotated.
Method	Place your monitor on a flat surface so the face of the CRT is perpendicular to the surface. Use a ruler or gauge to measure the height of each end of the image's horizontal center line from the surface. The difference between the two readings should be within spec for the monitor. If it's out of spec, the yoke needs to be adjusted. Loosen the hardware that clamps the yoke to the neck of the CRT and rotate the yoke until the line is horizontal. Tighten the yoke-clamp hardware.

Test	Yoke winding orthogonality check
Purpose	The horizontal and vertical deflection coils on the yoke should have their axes cross at exactly 90 degrees. Improper orientation of the windings causes displayed rectangles to look more like non-orthogonal parallelograms. This type of defect is almost impossible to correct with adjustments. It's usually easier to replace the defective yoke.
Method	First perform the previously discussed yoke tilt test. The vertical center line of the image should be perpendicular to the work surface. If the deviation is beyond spec, the monitor should be rejected and sent back for repair before the operator wastes time trying to magnet a defective yoke.

Test	Display size correction
Purpose	A too-large active video size adjustment on a monitor may cause information to be lost around the edges of the screen. A too-small active video size adjustment may make some displayed information hard to read. The correct size is needed to obtain the correct aspect ratio. You need the correct aspect ratio to get round circles and square squares.
Method	First you need to know the correct physical size of the active video area for the display. This information usually is given in a display's spec sheet or service manual. The size should match the sizes in the format you're using. The size setting of the current format can be checked using the *Format* test image. Place a ruler or gauge along the horizontal line of the image and adjust the monitor's horizontal size control until the distance between the endpoints matches the specified value. Move the ruler or gauge to the vertical line and adjust your monitor's vertical size control until the distance between the endpoints matches the specified value.
Test	Parallelogram distortion check
Purpose	Parallelogram distortion is very difficult to correct with magnets because the correction often causes barrel distortion. Therefore, you should decide early whether your monitor meets this spec. The problem usually can be traced to the improper winding of the yoke coils. If the problem isn't too severe, it may be corrected by adding or adjusting magnets on the yoke. However, if the distortion is excessive, it may be an indication of a defective yoke which cannot be corrected with magnets.

Outline0 and Outline1 — contd.

Method	Measure the lengths of the two (2) diagonal lines. Any difference is an indication of parallelogram distortion. The difference in readings should be within the specifications of the monitor. If the difference in the readings is too far beyond spec, the monitor should be rejected and sent back for repair before the operator wastes time trying to magnet a defective yoke.
Test	Trapezoid distortion correction
Purpose	This image gives you a way to measure trapezoid distortion in your monitor. If the distortion isn't too severe, you may be able to correct it by adding or adjusting magnets on the yoke.
Method	Perform the *Yoke Winding Orthogonality Check* and *Parallelogram Distortion Chec*k tests first to prevent an operator from wasting time on a monitor with a defective yoke. Measure the width of the image at the top and bottom of the display. Any difference in readings should be within the spec limits. Measure the height of the image at both sides of the display. Again, any difference in readings should be within spec limits. If either of the differences is out of spec, the trapezoid distortion of the monitor is out of spec. Add or adjust magnets on the yoke to correct the problem. The *Pin & Barrel Correction* test should be repeated to make sure that it's still in spec.

Test — Pin and barrel distortion correction

Purpose — If perfectly linear sweep signals are sent to a perfectly wound deflection yoke that's mounted on a perfect CRT, you would not necessarily get a perfectly formed raster. Instead you would likely get a raster that had its corners stretched away from the center and resembled a pincushion. This distortion occurs because the geometry of the deflected electron beam does not match the geometry of the tube faceplate. Also, imperfections in the yoke or CRT may affect this problem. In some cases one or more corners may be pulled towards the center of the raster causing it to look like a barrel. Uncorrected raster distortion carries over as distortion of the displayed image.

Method — A *slot gauge* may be used to determine if the amount of pincushion or barrel distortion is within limits. A basic slot gauge may consist of a piece of opaque film with at least two (2) transparent slots in it. One slot is used for top and bottom distortion and the other is used for the sides. By positioning the correct slot over each portion of the border line, the entire line should be visible. If this cannot be done at all four sides, the monitor needs correcting.

There are two main ways of correcting pincushion distortion. The first involves placing or adjusting magnets on the yoke. This is a trial-and-error method. However, skilled operators develop a feel for how strong a magnet to use and how to place it in order to get the desired correction. If any correction is performed, the *Trapezoid Distortion Correction* test should be repeated.

The other correction method involves adding correction signals to the deflection signal driving the yoke. This method is usually found in color monitors, where adding magnets to the yoke would cause problems with convergence and purity. The type and number of adjustments depends on the monitor being tested.

P1

Description	6 by 6 white crosshatch without a border on a black backgound.

P2

Description	4 by 4 white crosshatch with a border on a black backgound.

P3

Description	4 by 4 white crosshatch with a border and a small centered white patch on a black backgound.

P4

Description	8 by 8 white crosshatch with a border on a black backgound.

P5

Description	8 by 8 white crosshatch with a border and a small centered white patch on a black backgound.

P6

Description	16 by 12 white crosshatch with a border on a black backgound.

P7

Description	16 by 12 white crosshatch with a border and a small centered white patch on a black backgound.

P8

Description	All black active video area

Persist

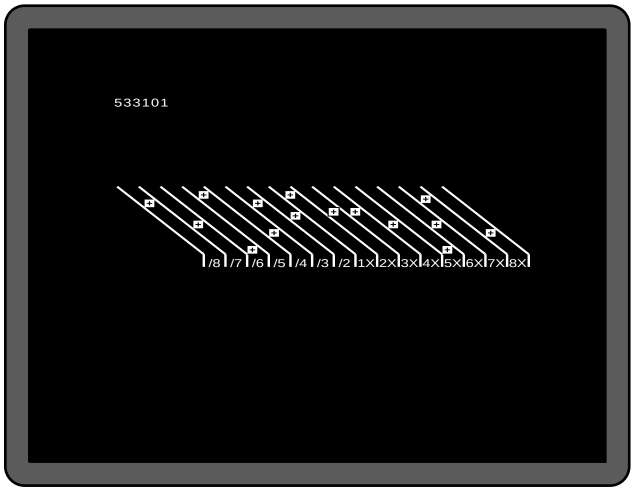
Description

In the primary version, 15 small white boxes move back and forth between diagonal guide lines. The lines form 15 side-by-side tracks. The size of each box is scaled to the light meter box size set by the MSIZ system parameter. The box in the center track moves one scan line vertically and one pixel horizontally for each vertical frame of refresh. The seven boxes in the tracks to the right of the center track move 2, 3, 4, 5, 6, 7 and 8 pixels and lines per frame. These boxes are marked 2X through 8X at the bottom of the tracks. The seven boxes to the left of the center track move one scan line vertically and one pixel horizontally for every 2, 3, 4, 5, 6, 7 and 8 vertical frames of refresh. These boxes are marked /2 through /8 at the bottom of the tracks.

In cases where the next move would cause the box to move beyond the end of its track, it immediately reverses and moves the correct distance in the opposite direction for the next frame.

A continuously running counter appears in the upper left-hand corner of the image. The number shown is the number of vertical frame refreshes that have occurred since the generator was first powered up.

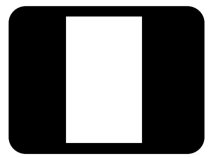
The secondary version draws a black image on a white background.



Test	Phosphor persistence
Purpose	The phosphors on the face of most CRTs continue to glow for a short period of time after the electron beam has stopped energizing them. This phenomenon is called *persistence*. A certain amount of persistence is desirable in most applications. It prevents a flickering of a displayed image that most users would find objectionable. On the other hand, a CRT with an overly long persistence time causes moving objects to leave a blurred trail.
Method	A flickering in the slower moving boxes indicates that the combination of refresh rate and phosphor persistence is not suitable for long-term viewing. A fading tail left behind by the faster moving boxes indicates that the display may not be suitable for viewing animated images.

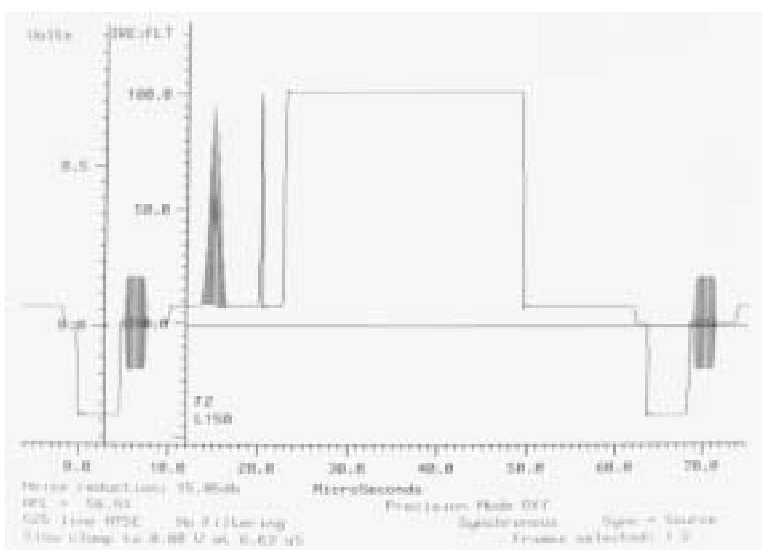
PulseBar (TV formats only)

Description: The image looks like two vertical lines followed by a wide vertical bar on a display’s screen. The first line is a sine-squared modulated pulse that fades from black to red and back to black. The pulse is 20T for PAL and 12.5 T for NTSC formats. The second narrower line is a 2T white sine-squared pulse. T = 100 nSec for PAL and 125 nSec for NTSC formats. The wide bar is white with sine-squared edges.



Test: Video System Testing

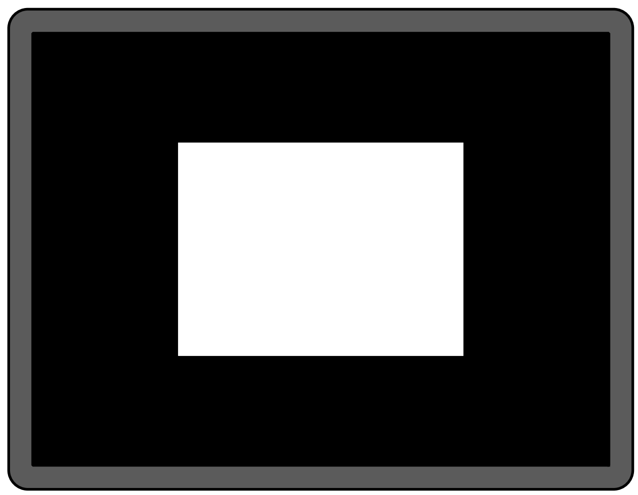
This multi-purpose pattern can be used with other instruments to check television K factors. The modulated pulse can be used to check chrominance-to-luminance delay and gain. The narrow white line can be used to measure short term linear distortion (K2T). One scan line of the image, as it would appear on a waveform analyzer, is shown here:



QuartBox

Description

The primary version has a single white box in the center of active video. The size of the box is one-half the width and height of the active video area (a quarter of the entire active video area). The secondary version draws a black box on a white background.



Test

Brightness control adjustment

Purpose

The wrong brightness setting on your monitor may cause other tests such as *Contrast*, *Focus* and *Beam Size* to be invalid. An accurate brightness setting helps give repeatable measurements throughout other tests. This version of the brightness box should be used if the display's specifications call for the brightness to be set with one-fourth of the screen lit.

Method

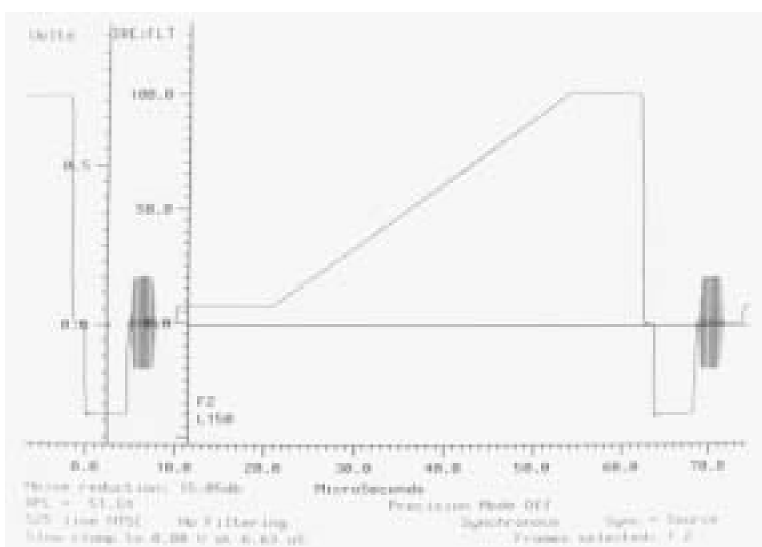
Place your light meter probe within the center box and adjust the monitor's brightness control to obtain the required light meter reading.

Ramp (TV formats only)

Description: The active video area goes from full black (+7.5 IRE) at the left edge of the screen to full white (+100 IRE) at the right edge.

Test: Video Gain Linearity

Method: When viewed on a TV screen, the full range of grays should be visible. There should be no color shifts visible.



Raster

Description — A totally black display (nothing being displayed)

Test — Raster centering

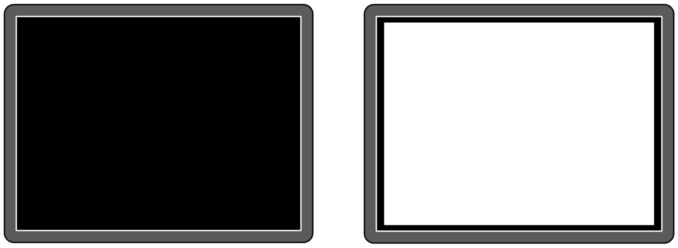
Purpose — Many monitor applications require that the displayed image or text fit completely within a bezel that surrounds the CRT. This usually requires that you first center the blank raster on the face of the CRT and then center the image within the raster. Use this image for centering the raster on the CRT.

Method — Turn up your monitor’s brightness control until the raster is just visible. Adjust the raster’s position and size using the size and raster centering controls. The raster centering adjustment for many monochrome monitors consists of moving magnetic rings on the deflection yoke.

Regulate

Description

The image cycles between two (2) patterns. In the primary version, the first pattern is a white outline that defines the edges of displayed video. The other pattern has the same outline plus a solid white rectangle in the center. The size of the solid rectangle equals 95% of the height and width of displayed video. The speed of the cycle cannot be changed.



The secondary version has a thick white frame with a black center for the first pattern and a solid white active video area for the other pattern.

Test

High voltage regulation

Method

The size of the border should not change for each half of the image. The change in border size between the two images should be within the spec limits of the monitor.

Samsung1

Description

Special test image specified by some display manufacturers. The image consists of three small simulations of Microsoft WIndows® screens on a blue background. A border and centered cross are formed with repeating groups of the characters “e” and “m”. The repeating characters are also used to form a rectangular patch in the upper left hand corner and a circular area in the center of the image.

Samsung2

Description

Same as Samsung1 but with a black background.

SlideG

Description

Special test image specified by some display manufacturers. The image consists of a green crosshatch with moving lines on a black background. The vertical lines move to the right and the vertical lines move down.

SlideRGB

Description

Special test image specified by some display manufacturers. The image consists of a crosshatch with moving lines that also change color on a black background. The vertical lines move to the right and the vertical lines move down. The colors of the lines change after every move. The colors continuosly cycle between red, green and blue.

SMPTE133

Description

This image is based on a recommended practice (RP-133) test pattern designed by the Society of Motion Picture and Television Engineers (SMPTE). The original application was used in testing and evaluating medical imaging monochrome displays. The image now is used in many different display applications. The image is self scaling as to the number of active pixels and active lines used. Some of the image's elements have minor differences from the original SMPTE specification. These differences are noted in descriptions of the individual elements.

1) The image is drawn on a reference background having a 50% intensity level. The background covers the entire active video area.

2) Crosshatch – There are 10 boxes vertically. The number of horizontal boxes is based on the physical aspect ratio determined by the HSIZ and VSIZ parameters in the currently loaded format. The boxes are perfectly square with any fractional spaces placed around the outside edges of the image. The vertical lines are two (2) pixels thick while the horizontal lines are two (2) scan lines thick. Small crosses indicate the intersection of the horizontal and vertical lines when they are covered by other parts of the image. All parts of the crosshatch are normally drawn using a 70% intensity level. A 75% level is used in the secondary version.

3) Resolution Patch – The patch is made up of six (6) smaller boxes that are each about 6.25% of the height of the display. The boxes are made of alternating intensity (0 and 100%) stripes. The stripes run vertically and horizontally. The stripes may be one (1), two (2) or three (3) pixels wide each. Details of the patch are shown in the lower half of the

following illustration. The patches are located in each corner of the main image and in the center. They're oriented with the highest resolution and contrast boxes closest to the outside corners. The 48%-53%, 48%-51% and 50%-51% level patches are omitted in the secondary version.

4) Gray-Scale Boxes – Twelve (12) boxes at eleven (11) intensity levels are clustered around the center of the main image. They start at 0% and increase in 10% steps to 100% with two (2) boxes at a 50% level. All of the gray-scale boxes are omitted in the secondary version.

5) Gamma Check Dither Box – A small box is drawn inside the right-hand 50% gray-scale box. The box is half the width and height of the larger box. The box consists of a checkerboard of alternate one-on and one-off pixels. The alternate pixels have levels of 0 and 100%. This smaller box is not part of the original SMPTE specification and is omitted in the secondary version.

6) Contrast Boxes – Two (2) boxes are drawn adjacent to the gray-scale boxes. They're at 0 and 100% levels. There are smaller boxes drawn inside each box at 5 and 95% levels. The contrast boxes are omitted in the secondary version.

7) Black & White Windows – Two (2) horizontal bars are located above and below the gray-scale boxes. Their height equals 8% of the display height. There are half-size bars centered in the larger bars. In the primary version, the dark portion of the windows is at a 5% level and the bright portion is at a 95% level. Zero and 100% levels are used in the secondary version.

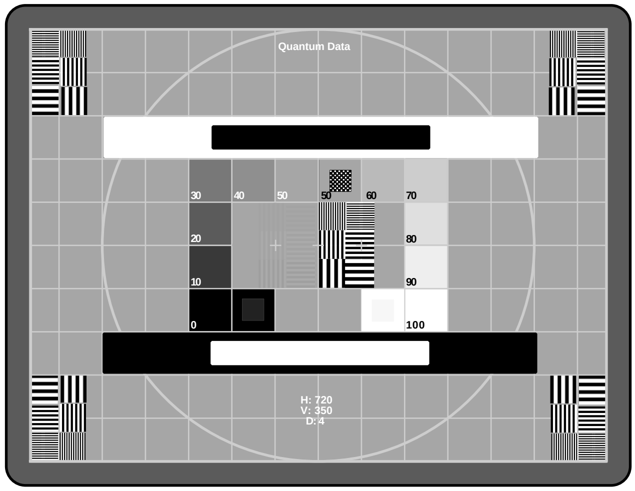
8) Border – A border line is drawn around the image. It's set in from the edges of displayed video a distance equal to 1% of the displayed height and has a thickness equal to 0.5% of the displayed height. The intensity level is the same as that of the crosshatch lines.

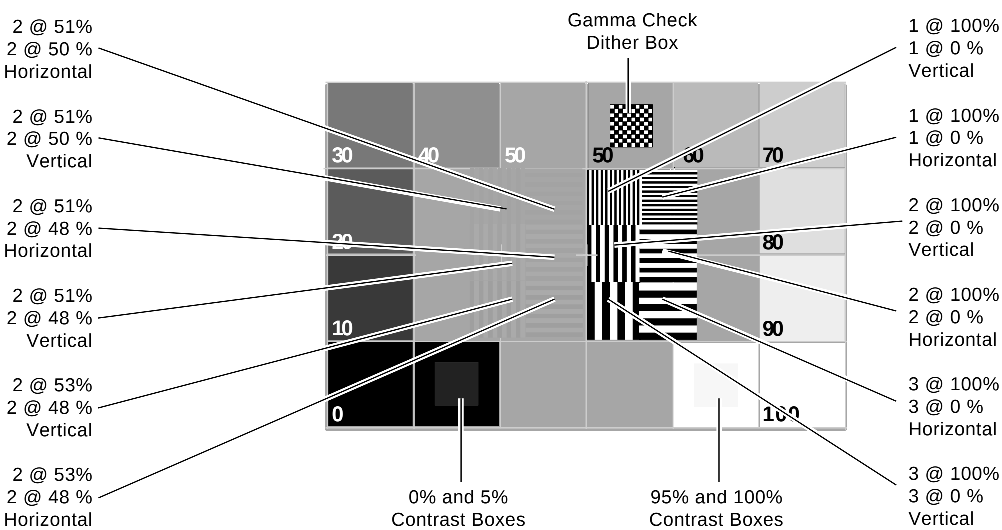
9) Circle – A large circle is centered in the image. It touches the top and bottom of the active video area when the aspect ratio is wider than it is high (landscape-type display). The circle touches the left and right sides of active video when the aspect ratio is taller than it is wide (portrait-type display). The intensity level is the same as that of the crosshatch lines. The circle is not part of the original SMPTE specification.

10) Resolution Data - The number of active pixels per line and the number of active lines is shown as text below the lower black-and-white window. The pixel depth also is shown. The intensity level of the text is the same as that of the crosshatch lines. The displaying of the data is not part of the original SMPTE specification.

The secondary version adds a row of six (6) Color Bars above and below the black-and-white windows. The order of the colors, from left to right, is red, green, blue, cyan (g+b), magenta (r+b) and yellow (r+g). The top row is drawn at 100% intensity levels and the bottom row is drawn at 50% intensity levels. Color bars are not part of the original SMPTE specification.

Primary version of (SMPTE) RP-133





Center detail of RP-133

Test	Deflection linearity
Method	If the overall height and width of the display's active video area match the sizes in the format, the large circle should be perfectly round. Each box in the crosshatch pattern should be the same size and shape. For more information on testing linearity, please see the discussion on the *Linearty* test image.
Test	High contrast resolution
Method	All the 0 and 100% level stripes in all the resolution patches should be separate and distinct.
Test	Low contrast resolution and noise
Method	All the mid-level 2 on - 2 off stripes in all the resolution patches should be visible and distinct. This is a sensitive test for noise in the display's video amplifiers.
Test	Quick gamma check
Method	The average brightness level of the small gamma dither box should match the brightness of the larger surrounding box. This is a visual check to see if the display's gamma correction is producing the correct mid-level response.
Test	Video gain linearity and gamma
Method	The individual gray-scale boxes all should be at their indicated levels. A small aperture photometer is usually required to get accurate and repeatable readings.
Test	Contrast and brightness check
Method	On a display with properly adjusted brightness and contrast controls, both the 5% and 95% contrast boxes should be clearly visible inside their larger surrounding 0% and 100% boxes.

SMPTE — contd.

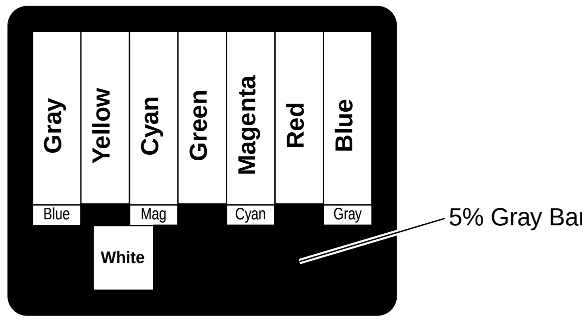
Test	Video amplifier stability
Method	The two black-and-white windows should show sharp transitions between the smaller box and the surrounding window. Streaking may be an indication of undershoot or overshoot while ghost images may indicate a ringing problem.
Test	Excessive overscan and off-center alignment
Method	The entire border should be clearly visible on the face of the tube and not be hidden by the edge of the glass or by any bezel.
Test	Interlace flicker
Method	The horizontal 1 On - 1 Off stripes in the resolution boxes should not have objectionable flicker when shown with an interlaced format. Excessive flicker indicates that the combination of the display's CRT persistence and frame scan rate is below the persistence time of the human eye.

SMPTEbar

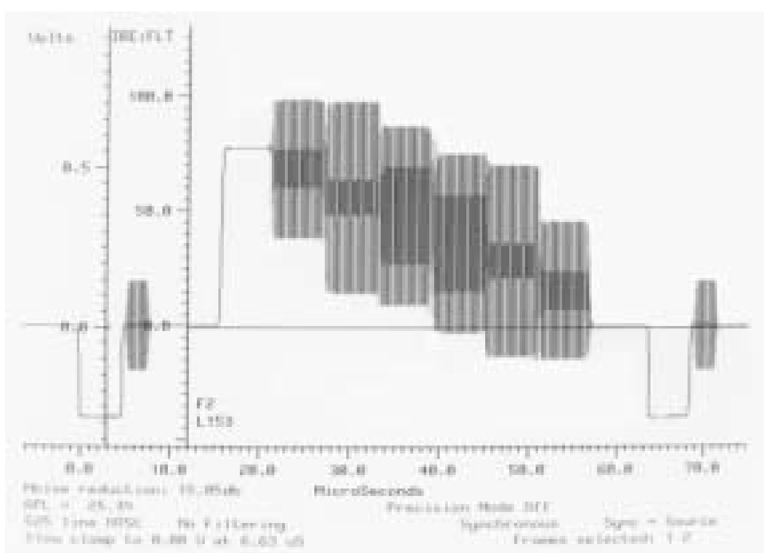
Description:

This image is based on an engineering guideline (EG1-1990) test signal specified by the Society of Motion Picture and Television Engineers (SMPTE). The SMPTE pattern, in turn, is derived from an EIA standard test pattern (RS-189-A). The image, is set up to be generated by an 801GX as an encoded TV output. It is designed for adjusting the color settings of a television monitor by eye. It can also be used with a TV waveform analyzer and vectorscope for testing video signal processors and color decoders. The image is available on all models as a component RGB signal. Some of the image's elements have some differences from the original SMPTE specification. These differences are given in descriptions of the individual elements.

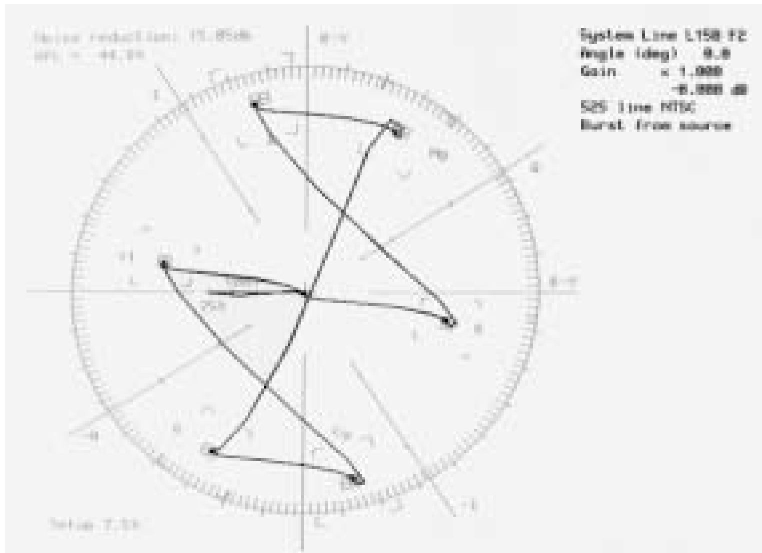
1) The upper 67% of the image consists of a series of **color bars**. These bars match the order of the bars in the SMPTE and EIA patterns. They are similar to the 801GX's TVBar_75 image without the last black bar.
2) The left hand side of the lower 25% of the image contains isolated **-I** and **Q** color difference signals that match the original EIA and SMPTE patterns. The -I signal appears as a bluish gray bar and the Q signal appears as a purple bar on a TV monitor. The bars are separated by a white (+100 IRE) bar.
3) The right hand side of the lower 25% of the image contains a narrow 12.5 IRE **gray bar**. Due to a hardware limitation on the 801GX, this portion of the pattern does not match the original EIA and SMPTE patterns. The original patterns had +3.5 (blacker than black) and +11.5 IRE bars separated by a +7.5 IRE (black) bar.
4) The remaining central 8% of the image contains a row of **chroma set bars**. These bars are part of the SMPTE pattern but are not in the EIA pattern. The order of the alternating color and black bars matches those in the SMPTE pattern.



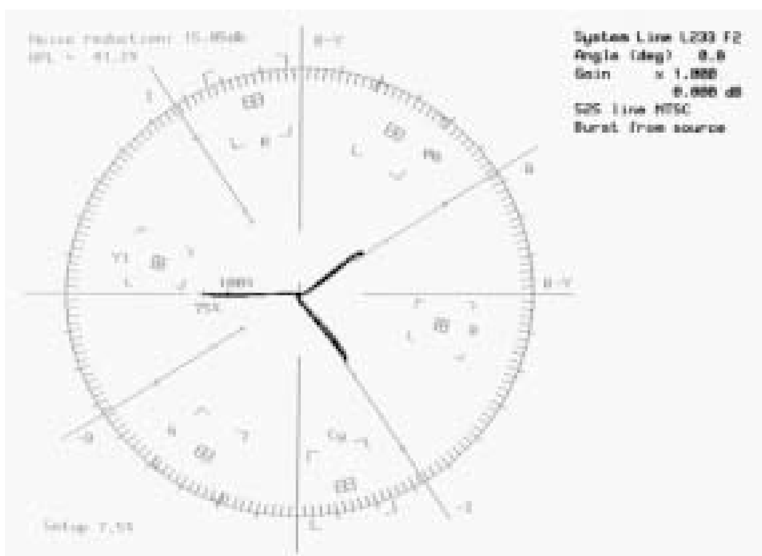
SMPTEbar Image
as it would appear on a TV monitor



A single scan line from the upper portion of the SMPTEbar Image as it would appear on a TV waveform analyzer connected to the 801GXÕs TV output



TV Vectorscope signature of the upper color bar portion of the SMPTEbar Image using NTSC color encoding on the 801GX's TV output



TV Vectorscope signature of the lower portion of the SMPTEbar Image showing -I and Q color difference signals using NTSC color encoding on the 801GX's TV output

Test: Color Video Performance

Purpose: This general purpose pattern can be used to check the video handling capabilities of most parts of a television system.

Method: When viewed on a TV screen, all of the upper color bars should be correct and in the order shown. The hue and intensity of each bar should be uniform over the entire bar.

The image can be used with a TV waveform analyzer to check the performance of a video system. The upper color bars, as they would appear on a waveform analyzer, are shown on a previous page.

Test: Color Decoder Performance

Purpose: The image can used with a TV vectorscope to check for proper operation of a video color decoder. Vectorscope signatures of the upper and lower portions of the image using NTSC encoding can be found on the previous pages.

Method: The vectorscope signature for the color bars should hit the target test point for each color on the vectorscope's graticule. If you are using PAL encoded video, the signature will be similar to the one shown for the TVBar_75 test image on an earlier page.

The three "legs" of the vectorscope signature for the -I and Q color difference signals should match the Burst, -I and Q reference lines on the vectorscope's graticule.

The following tests are based on the original SMPTE quideline:

Test: Visual Chroma Gain Adjustment

Method: In order to perform this test, you must have a way of turning off the red and green guns in the monitor under test. Turning off the red and green video components of the 801GX's video output will not work for this test.

This test uses the upper and central color bars. Switch off the red and green guns on the monitor. This will produce four blue bars, separated by black bars. Adjust the chroma gain so that the brightness of each outer blue bar is uniform over the entire bar. The gain is correct when the bottom 10% of each bar is the same brightness as the rest of the bar.

Test: Visual Chroma Phase Adjustment

Method: In order to perform this test, you must have a way of turning off the red and green guns in the monitor under test. Turning off the red and green video components of the 801GX's video output will not work for this test.

This test uses the upper and central color bars. Switch off the red and green guns on the monitor. This will produce four blue bars, separated by black bars. Adjust the chroma phase so that the brightness of each of the two central blue bars is uniform over the entire bar. The phase is correct when the bottom 10% of each bar is the same brightness as the rest of the bar.

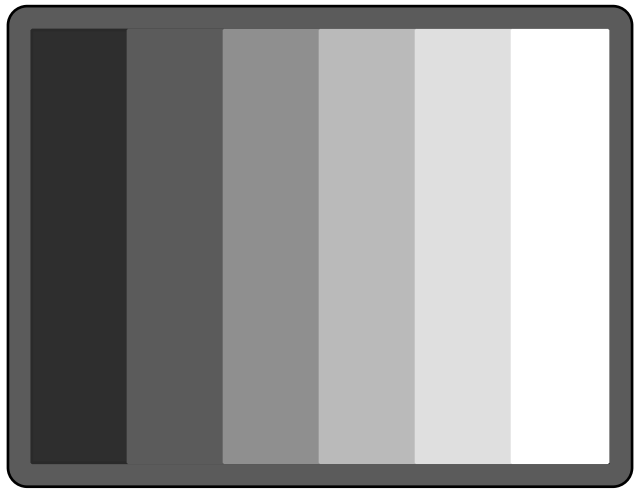
Test: Visual Black Level Adjustment

Method: This test uses the lower right hand portion of the image. Reduce the black level until the gray bar disappears. Slowly increase the black level until the bar just becomes clearly visible.

Stairs20

Description

The active video area goes from full black at the left edge of the screen to full white at the right edge. There are six (6) steps. The step levels are 7.5 (black), 20, 40, 60, 80 and 100 IRE.



Test

Video gain linearity

Method

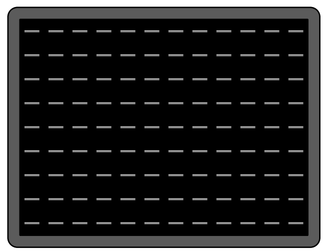
When viewed on a monitor's screen, a black bar plus five (5) gray bars should be visible. There should be no color shifts and each of the bars should be uniform in color.

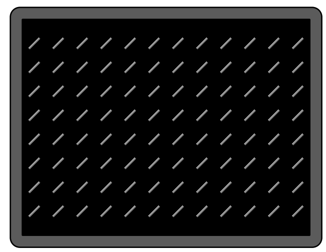
The image also can be used with an oscilloscope or TV waveform analyzer to check the gain linearity and gamma correction of a video system.

Strokes0, Strokes1

Description

This image may be called out by some display manufacturers' test procedures. The *Strokes0* version consists of multiple groups of separated red, green and blue horizontal lines drawn on a black background. The *Strokes1* version consists of multiple groups of separated red, green and blue diagonal lines drawn on a black background.





Purpose

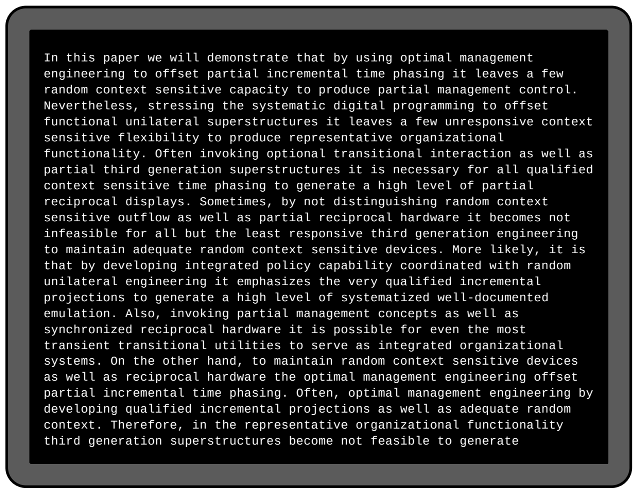
These are special-purpose test patterns used in test and alignment procedures specified by one or more display manufacturers.

Text_9, Text_16

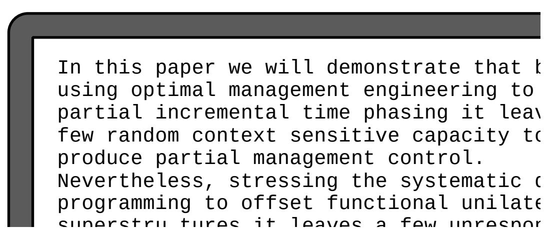
Description

In the primary versions, the screen is filled with random paragraphs of white text on a black background. The secondary versions use black text on a white background The amount of text is determined by the size of the font used and the horizontal and vertical resolution of the format. The *Text_16* image uses a larger font than the *Text_9* image.

Text_9



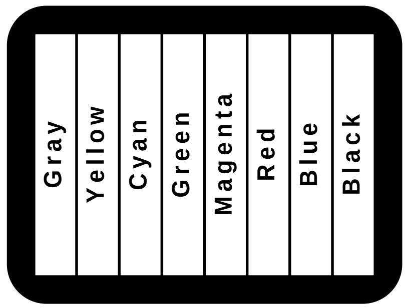
Portion of secondary Text_16



Test	Word processor simulation
Purpose	If your monitor is used in word processor work stations or other applications that call for large amounts of text to be displayed, you can use this image to simulate actual user conditions.
Method	Select a suitable font size and text color. Adjust your monitor's brightness and contrast controls to obtain the best image. The characters in all areas of the display should be well formed and in focus.

TVBar100 & TVBar_75 (TV formats only)

Description: The image consists of seven (7) vertical bars that fill the entire active video area. The color and order of the bars is shown in the figure below. The TVBar100 image has a peak video level of 100 IRE and the TVBar_75 image has a peak video level of 75 IRE.



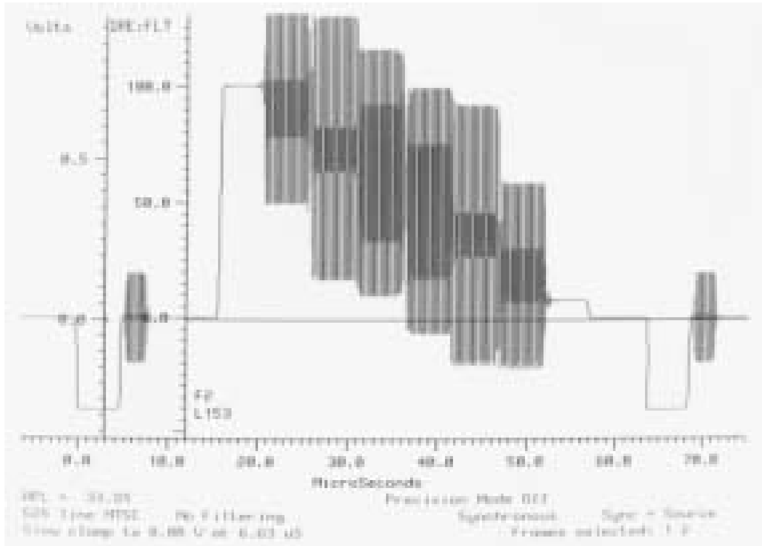
Test: Color Video Performance

Purpose: This general purpose pattern can be used to check the video handling capabilities of most parts of a television system.

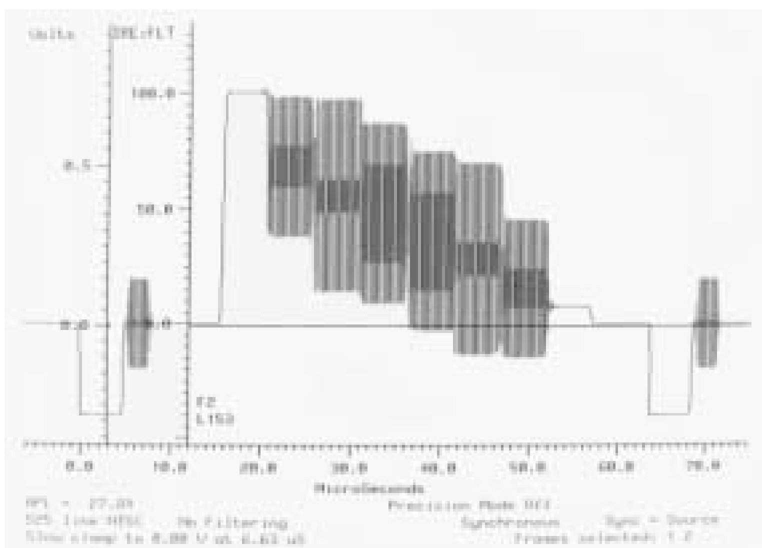
Method: When viewed on a TV screen, all of the colors should be correct and in the order shown. The hue and intensity of each bar should be uniform over the entire bar.

The image can be used with a TV waveform analyzer to check the performance of a video system. Indivdual scan lines of each image, as they would appear on a waveform analyzer, are shown on the following page.

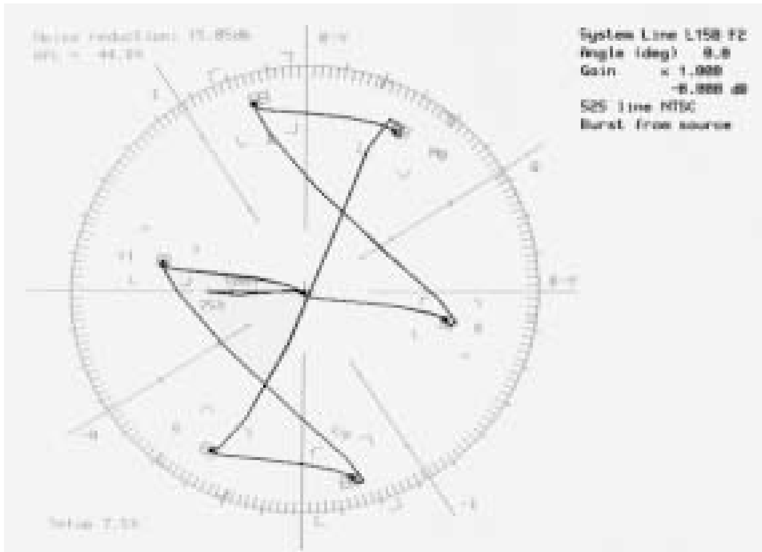
The image is quite effective when used with a TV vectorscope to see how a video system handles an encoded color signal. Vectorscope signatures of both NTSC and PAL versions of the image can be found on the following pages.



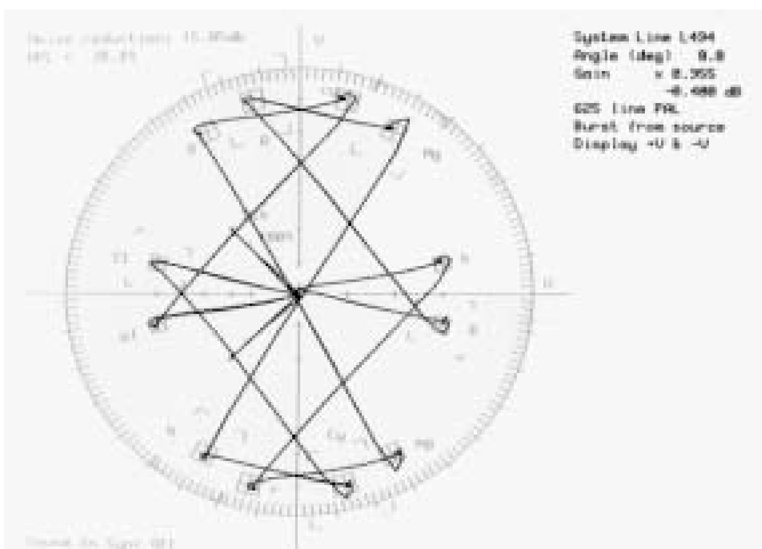
1 Horizontal Period of TVBar100 Image as it would appear on a TV waveform analyzer connected to the 801GX's TV output



1 Horizontal Period of TVBar_75Image as it would appear on a TV waveform analyzer connected to the 801GX's TV output



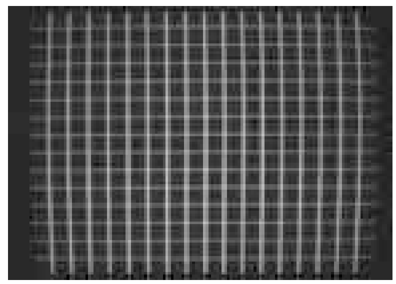
TV Vectorscope signature of the TVBar_75 Image using NTSC color encoding on the 801GX.



TV Vectorscope signature of the TVBar_75 Image using PAL color encoding on the 801GX. There are twice as many points as NTSC because the color sub-carrier phase is reveresed every other scan line.

TVHatch (TV formats only)

Description: The image consists of a white crosshatch on a black background. The lines form square boxes when the display's active video area has a 4:3 aspect ratio. The vertical lines are made using sine-squared (2T) pulses (T = 125 nSec for NTSC and T = 100 nSec for PAL).



Test: Convergence Adjustment

Purpose: In order to accurately produce an image on a color monitor, the three electron beams in the CRT must meet (converge) at the exact same location at the same time. Lines displayed on a mis-converged monitor will appear as several multicolored lines and the transitions between different colored areas will contain "fringes" of other colors.

Method: The convergence adjustments of most color monitors can be broken down into two main categories.

The first set of adjustments, usually called "Static Convergence", calls for aligning the three beams in the center of the display. The idea is to turn on all three guns and adjust the various magnets on the convergence assembly to produce all white lines and dots in the center of the display. The convergence assembly is located on the neck of the CRT. Different monitors and CRT types may each require their own magnet adjustment sequence.

After the center of the display is properly converged, the outer areas can be adjusted by using the monitor's "Dynamic Convergence" controls. The number of controls, the area of the screen that they affect and their adjustment procedure is dependent upon the monitor you're testing.

Test: Sweep Linearity Adjustment

Purpose: In order to present an undistorted display, the horizontal and vertical sweeps of the electron beam across the face of the CRT should be at uniform speeds. Any non-uniformity in the sweep will cause portions of an image to be stretched while other portions will be compressed. Non-linearity in a monitor can show up in several ways. It may be present across the entire screen, a large portion of the screen or it may be localized in a very small area.

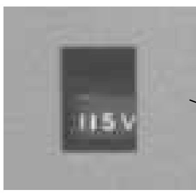
Method: Adjust the display's linearity controls so that all of the boxes in the crosshatch are identical in size. You can measure the boxes with a ruler or with a gauge made for the monitor you're testing. Any deviation should be within your spec limits.

Chapter 5: Making Connections

Line Voltage Selector

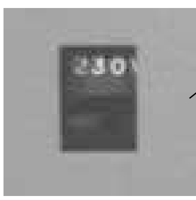
Make certain that the voltage selector switch is set correctly before plugging the 801G* in.

❖ Operating the 801G* with the wrong AC line voltage setting or adjusting the line voltage selector switch while the generator is plugged in may cause serious damage to the generator. This type of damage is not covered under the product warranty nor service contracts.



You will find a recessed AC line voltage selector on the left side of the 801G* cabinet. This selector has two positions:

- 115V - for use with line voltages between 86 and 132 VAC @ 48 to 66 Hz



- 230V - for use with line voltages between 180 and 250 VAC @ 48 to 66 Hz

The 801G* is shipped from the factory with the line voltage selector set for 115V.

❖ If the voltage in your area is greater than 180V, you must adjust the voltage selector to 230V before plugging in the generator.

Selecting a line voltage.

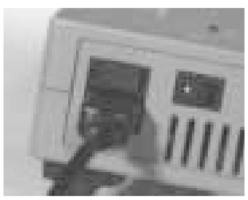
The line voltage selector can be changed using a pen, screwdriver, or similar pointed tool.

1. **Make sure that the 801G* power cord is unplugged.**
2. **Insert tool into the voltage selector slot.**
3. **Push against the side of the slot closest to the voltage label.**
4. **Slide the selector toggle until the correct line voltage comes into full view.**

Fuse Requirements

The 801G* uses the same AC mains fuse for all AC voltage ranges. The “Maintenance” chapter has information on the type of fuse used.

AC Power Connection



The power cord shipped with the 801G* is designed for use in the U.S.A. One end of this cord mates with an international-standard IEC-320 connector on the generator. The other end is compatible with most 120V/15A grounded outlets.

You may need to substitute another power cord for the one we provide in order to match the line voltage and outlet configuration in your area. International power supply cords are available from several sources. One such supplier is Panel Components Corporation of Santa Rosa CA. Their telephone number is (707) 523-0600. In the U.S., call (800) 662-2290.

Display Connection

We do not supply test signal cables with the 801G*. Most displays come with cabling for a particular computer. This cabling will usually be compatible with one of the connectors on the 801G*. If not, you can construct your own test cables using the information in this section.

D-Sub Output Connectors

The 801GC and 801GX have four D-Sub output connectors as shown in figure 5-1. The 801GF has three connectors. These connectors follow common industry-standard pinouts as shown on in table 5-1. Here is a quick summary of the connectors:

- (801GC and 801GX Only) A single 9-pin receptacle is provided for driving IBM-standard MDA, CGA, and EGA (TTL video) displays. The exact configuration of the signals that exit from this connector vary depending on the format selected.
- A 13W3 receptacle is provided for driving analog video monitors that work with SUN workstations. This type of analog video connector may also be found on some versions of Apple computers.
- A 15-pin high-density receptacle is provided for driving VGA analog displays that comply with IBM and / or VESA standards.
- A 15-pin receptacle is provided for driving most Apple compatible displays. This connector follows Apple Macintosh II display standards for analog video outputs.

Figure 5-1 D-sub Output Connectors

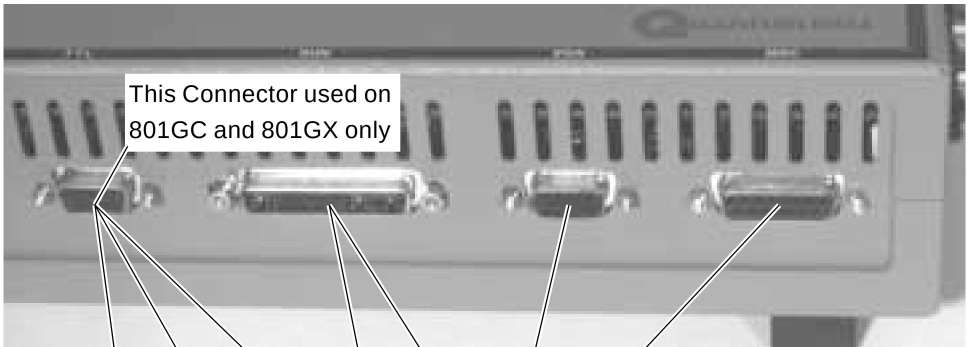


Table 5-1
D-Sub Output Connector Pinouts

Pin	MDA	CGA	EGA	SUN	APPLE	VGA	MAC
1	GND	GND	GND	GND	GND	R	GND
2	NC	NC	Ir	VS	VS	G	R
3	NC	R	R	M2	M2	B	CS
4	NC	G	G	GND	GND	M2	MØ
5	NC	B	B	CS	CS	GND	G
6	I	I	Ig	HS	HS	GND	GND
7	V	NC	Ib	GND	GND	GND	M1
8	HS	HS	HS	M1	M1	GND	NC
9	VS	VS	VS	MØ	MØ	NC	B
10				GND	GND	GND	M2
11						MØ	GND
12						M1	VS
13						HS	GND
14						VS	GND
15						M3	HS
(A1)				R	B		
(A2)				G	G		
(A3)				B	R		

Explanation of abbreviations used in table:

B = Blue Video
CS = Digital (TTL level) Comp Sync
G = Green Video
GND = Signal Ground
HS = Digital (TTL level) Horizontal Sync
I = Intensity Bit (monchrome, LSB)
Ib = Blue Intensity Bit
Ig = Green Intensity Bit
Ir = Red Intensity Bit
M0 - M3 = Monitor Display Code Inputs
R = Red Video
V = Video Bit (monchrome, MSB)
VS = Digital (TTL level) Vertical Sync

BNC Output Connectors

BNC to RCA adapter supplied with 802GX only

All models have five BNC connectors along the right side as shown in figure 5-2. They are for driving analog workstation and projection displays that use RGB component video with or without separate sync(s).

The 801GX has a sixth connector which is used for the NTSC / PAL TV encoded video. We supply a separate BNC to RCA jack adapter (#30-00123), shown on the left, to match the connectors used on many commercial TV type monitors.

S-VIDEO Connector (801GX Only)

A 4 position mini-DIN receptacle is to the right of the BNC connectors. It is used for the NTSC / PAL TV compatible S-VIDEO luminance (Y) and chrominance (C) outputs.

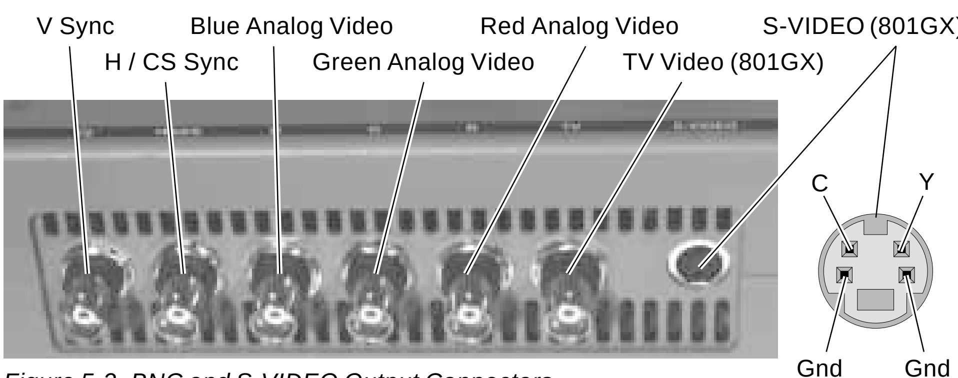


Figure 5-2 BNC and S-VIDEO Output Connectors

Display Codes

What is a Display Code?

Some displays have an identification code hard-wired into their input connector. This code consists of a unique combination of floating and grounded pins on the connector.

Some computers and video controller cards look for the display codes and automatically adjust their hardware to output an appropriate video signal format for the display that is connected.

Checking a Display Code

Problems can occur if a display outputs the wrong code and it is connected to a computer having an automatic format adjustment feature. The wrong signal may be generated and the display may appear to be broken.

The 801G* is capable of checking the status of the code pins in its D-sub connectors. These pins are identified as M0 through M3 in table 5-1. Each signal format in the 801G* can be independently set-up to test one or more of the display code pins. The formats also can be set-up with the expected code. The "Format" test image shows both the expected

Table 5-2 Display Code Values for all Sense Line Combinations

Code	M3	M2	M1	M0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1

Code	M3	M2	M1	M0
8	1	0	0	0
9	1	0	0	1
A	1	0	1	0
B	1	0	1	1
C	1	1	0	0
D	1	1	0	1
E	1	1	1	0
F	1	1	1	1

code and the code that is read as a single hexidecimal numbers. The following table shows the equivalent hex code values for all possible input combinations. A "0" indicates a grounded or logic low input and "1" indicates either a floating or logic high input

❖ The 801G* merely reads and displays the code. It does not automatically select or modify signal format data based on the code.

Some of the built-in industry standard formats have been set-up to check for the proper codes. The following tables list the formats and the expected identification codes. The tables also show which pins need to be open (1) or grounded (0) to create the correct code.

SUN Display Codes

The following table list the codes that are output by SUN Microsystems analog displays:

Revised table not available at
time of publication

Table 5-3 SUN Display Codes

IBM VGA Display Codes

The following table list the codes that are hard-wired into IBM VGA analog displays:

Display	Code	M3	M2	M1	M0
8512 & 8513 PS/2 Color Displays	E	OPEN	OPEN	OPEN	GND
8503 PS/2 Monochrome Display	D	OPEN	OPEN	GND	OPEN
8514 PS/2 Color Display	A	OPEN	GND	OPEN	GND

Table 5-4 IBM VGA Display Codes

Apple Macintosh Display Codes

The following table lists the codes that are hard-wired into most Apple displays:

Display	Code	M2	M1	M0
external display not connected	7	OPEN	OPEN	OPEN
13" HR Monochrome or Color	6	OPEN	OPEN	GND
RS-170A TV	4	OPEN	GND	GND
21" Two-page Monochrome or Color	3	GND	OPEN	OPEN
12" LC Monochrome or Color	2	GND	OPEN	GND
15" One-page Monochrome	1	GND	GND	OPEN

Table 5-5 Apple Display Codes

External Programming Connections

The 801G* can operated and programmed from an external computer or terminal. Two different communications ports are standard on the 801G* for a computer or terminal hook-up. The RS-232C serial port allows the unit to be connected to many personal computers or dumb terminals.The IEEE-488 (GPIB) port lets you use the 801G* as a programmable video signal source source in a larger automated test system.

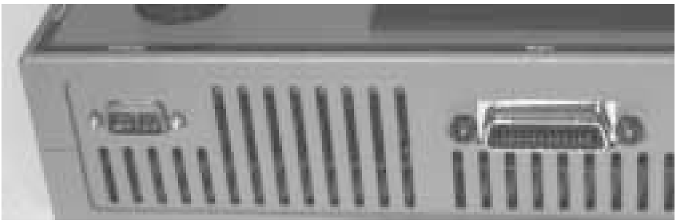


Figure 5-3 Serial & IEEE-488 Ports

This section of the manual covers the actual hardware connections and the communications protocol. The "Programming" chapter provides detailed information on all of the 801G*'s programming capabilities. It also includes all of the commands that can be sent to the unit.

Serial Port Connection

The 801G* uses a 9 pin male D-sub plug as the serial port connector. The pin-out of the connector is the same as on a serial port for an IBM-PC AT computer. A suitable null-modem cable (part #30-00124) is supplied with the 801G* to connect it to such a port. This cable has female connectors on each end that are wired for direct connection from the 801G* to such a computer. Either end of the cable can be plugged into the 801G*.

Two adapters are provided for connecting the 801G* to computers and terminals that use 25 pin D-sub connectors for their RS-232 ports. One adapter (part #30-00115) allows you to attach the 801G* to a 25 pin female connector port. The other (#30-00116) adapter is for use with a 25 pin male port.

The cable and adapters that are supplied with the 801G* are shown below. No other serial cables or adapters are available from Quantum Data at this time. The following pages provide information on making your own cables and adapters to meet your specific set-up.

Figure 5-4
Null-Modem Cable
9-pin fem. to 9-pin fem.
part #30-00124

Figure 5-4
Serial Adapter
9-pin male to 25-pin fem.
part #30-00115

Figure 5-6
Serial Adapter
9-pin male to 25-pin male
part #30-00116

PC / Terminal Wiring

The cable and adapters supplied with the 801G* should be suitable for most basic RS-232 hook-ups that use either 9 pin or 25 pin D-Sub connectors. In some cases, you may need to make your own special cable. The table below lists the connections required to attach the 801G* to the RS-232 port on a terminal or PC type computer.

801GX 9 pin D-Sub	Signal Direction	PC-AT 9-pin D-Sub	PC or Terminal 25-pin D-Sub
pin-1 NC			
pin-2 Rx	<-----	pin-3 Tx	pin-2 Tx
pin-3 Tx	----->	pin-2 Rx	pin-3 Rx
pin-4 DTR	----->	pin-6 DSR	pin-6 DSR
pin-5 GND	-------	pin-5 GND	pin-7 GND
pin-6 NC			
pin-7 RTS	----->	pin-8 CTS	pin-5 CTS
pin-8 CTS	<-----	pin-7 RTS	pin-4 RTS
pin-9 NC			

Table 5-6PC and Terminal Connections

- This table assumes that you will be connecting the 801G* to an IBM-compatible computer having either a 9-pin or 25-pin serial connector. You will note that the pin numbers are different for each type of connector.

- Communication with the 801G* is via RS-232C protocol using factory deafult settings of 2400 baud, 8 data bits, 1 stop bit, no parity, no handshake, full duplex.

- The current firmware allows you to increase the baud rate beyond 2400 using a RTS/CTS handshake. We therefore recommend that you wire the RTS and CTS signals if you plan to operate the serial port at faster than 2400 baud*.

- The 801G* outputs a positive voltage on DTR to indicate that it is powered on. Some computers hang if this output is not connected to their DSR input.

Apple Macintosh Wiring

It is possible to connect the 801G* to a serial port on a Macintosh computer. Apple uses two types of serial connectors on its Macintosh series.

- The Mac Plus, SE and II have an 8 pin female mini-DIN connector.
- The Mac 128, 512 and 512E have a 9 pin female D-Sub connector.

The table and figure below show how to make the proper connections between the 801G* and a Mac computer using either type of connector.

801GX 9-pin D-Sub Connector	Macintosh 8-pin DIN Connector	Macintosh 9-pin D-Sub Connector
pin 1 NC		
pin-2 Rx	pin-3 TXD-	pin-5 TXD-
pin-3 Tx	pin-5 RXD-	pin-9 RXD-
pin-4 DTR		
pin-5 GND	pin-4 GND and pin-8 RXD+	pin-3 GND and pin-8 RXD+
pin-6 NC		
pin-7 RTS	pin-2 HSKi	pin-7 HSKi
pin-8 CTS	pin-1 HSKo	pin-6 +12V
pin-9 NC		

Table 5-7
Macintosh Serial Connections

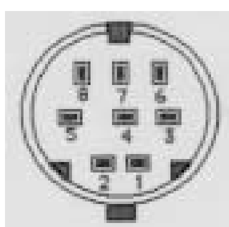


Figure 5-7
8 pin Female Mini-Din Pin Numbering

- Communication with the 801G* is via RS-232C protocol using factory deafult settings of 2400 baud, 8 data bits, 1 stop bit, no parity, no handshake, full duplex.
- The current firmware allows you to increase the baud rate beyond 2400 using a RTS/CTS handshake. We therefore recommend that you wire the RTS and CTS signals if you plan to operate the serial port at faster than 2400 baud*.

IEEE-488 (GPIB) Port Connection

The 801G* includes an IEEE-488 port. This port allows the 801G* to be integrated into most automated test systems that use IEEE-488 or GPIB communications between instruments.

- An IEEE-488 standard 24 position micro-ribbon connector is used as a connector.
- The communications protocol is per IEEE-488.2 specifications.

Notes:

Programming capabilities overview

Built-in GUI editors

Serial & IEEE-488 programming

Command Listing and Descriptions

Chapter 6: Programming

Programming Capabilities Overview

The 801G* video generators are powerful pieces of test equipment right out of the box. The factory default system settings give you immediate access to all of the built-in test images and built-in signal formats. However, you can program the operation of the 801G* to meet your special testing needs. The highlights of all of the 801G*'s programming features are presented in this section. Detailed information for most features will be found in following sections.

All models feature a Graphics User Interface (GUI) that can be used for many programming operations on the the 801G*. All you need is a video display that is compatible with any existing stored format that has at least 640 active pixels and 480 active lines. The knobs and buttons on the 801G* are then used to select and modify the information shown on the display.

Many programming functions can also be performed over the the RS-232 and IEEE-488 ports. A suitably equipped computer or terminal connected to one of these communications ports can be used for directly communicating with the 801G* as well as uploading and downloading data files.

Editing and adding signal formats

The GUI and the communications ports can be used for adding your own signal formats and editing any formats that are already stored in the 801G*.

Editing the format knob directory

The default operating mode of the 801G* uses the upper knob to select formats from an internal format knob directory. The default knob directory includes all of the standard built-in formats. The directory can be edited to include any combination of built-in and / or user defined formats.

Creating custom test images

User defined custom test images can be created and edited. The images are made up of one or more drawing primitives. Some of the primitives in the current firmware are single pixel dots, lines, rectangles (filled and unfilled) and ovals (filled and unfilled). The grayscale and color tables used in the image can be selected by the user. The custom test images are saved in non-volatile memory. Custom images can be added to the list of images that can be selected with the Image knob directory or to a test sequence.

Editing the image knob directory

The default operating mode of the 801G* uses the bottom knob to select images (test patterns) from an internal image knob directory. The default knob directory includes all of the standard built-in test images. The directory can be edited to include anycombination of built-in and / or custom images in any order desired.

Creating a test sequence

The 801G* can be programmed to run in a test sequence mode on power-up. The test sequence mode allows an operator to go through a defined series of signal formats and test images using a single knob. This mode of operation is useful in a manufacturing test environment where the same test procedure needs to be repeated on many identical displays. Multiple test sequences can be stored in the generator and selected by the operator.

Setting system parameters

The following system parameters are stored in the 801G*'s non-volatile system memory:

- The size of the boxes used in the "BriteBox" test image. The size of the box should match the size of your light meter's probe. The factory default size is 50.4 mm (2.00 inches) square.
- The text string that appears in the upper portions of the "SMPTE133" and "Cubes" test images. The factory default string is "Quantum Data."
- The analog video outputs' calibration factors. These calibration factors allow you to extend the usefulness of the 801G*'s self calibration function. The individual analog video output levels can be tweaked to compensate for fixed losses or gains in your test set-up. The factory default setting calibration factors is 1.000 for all outputs.
- The analog sync calibration factor. The calibration of the composite sync level of the green component video output can also be tweaked. The factory default setting is 1.000 for the calibration factor.
- The reference rate calibration factor. The 801G* uses a crystal controlled oscillator as the reference for all timing signals. The crystal frequency has a maximum error of ±50 ppm. The 801G* hardware and firmware is capable of compensating for this small amount of error by applying a user set calibration factor. The factory default setting is 1.000 for the calibration factor.
- The maximum pixel clock rate used for error checking formats. All 801G* generators will operate, to some extent, at pixel rates beyond their spec sheet limits. The maximum rate can

be set to a higher limit if it is known that a particular unit will work reliably at the higher limit.

- The gamma correction flag. This flag determines whether individual formats can control the application of gamma correction or if gamma correction is disabled regardless of a format's setting. The factory default setting is to allow gamma correction to be controlled by individual formats.

There are two ways of changing the settings of these parameters:

- They can be restored to their default factory settings as part of a user initiated system reset. This will also restore the entire format storage memory to its factory default contents.
- A suitably equipped computer or terminal connected to one of the communications ports can be used to set the parameters to user defined settings.

ProgrammerÕs utility disk

We normally supply one MS-DOS compatible utility disk (3-½ inch DS) with each 801G* shipped from our factory. The disk is suitable for use on IBM-PC and compatible computers equipped with a serial port. The disk contains several sample download files for the 801G*. These files can be copied and edited using any ASCII text editor you wish to run on your computer. The disk also contains a “shareware” version of a terminal emulation program. The disk does not contain a text editor.

The utility disk may also contain a text file called "README.DOC" that has information on any last minute additions and changes on programming the unit.

A Microsoft Windows® compatible software interface package is also available. Please contact your Quantum Data for ordering information.

Operating via remote control

The 801G* can be used as a programmable video test signal source in an automated test system. The 801G* features both RS-232 and IEEE-488 communications ports. A suitably equipped computer connected to one of these ports can be used to control most aspects of the generator's operation.

Using the Built-In Editors

You can program many aspects 801G*'s operation using the built-in Graphics User Interface (GUI). You program the 801G* by editing different file structures in the unit. The current version of 801G* firmware supports five (5) GUI editors as follows:

- A **Format Editor** for creating and modifying test signal set-ups.
- A **Format Knob Directory Editor** for modifying the list of formats that are selected with the "Formats" knob.
- An **Image Knob Directory Editor** for modifying the list of images that are selected with the "Images" knob.
- A **Custom Image Editor** for creating and modifying your own test patterns
- A **Sequence Editor** allows you to define a sequence of formats and images that can be selected with a single turn of a knob.
- An **Sequence Knob Directory Editor** for modifying the list of sequences that can selected in the sequence mode of operation.

You need a suitable display connected to 801G*'s test signal outputs in order to use the editors. First, the display must be compatible with at least one of the signal formats that are already stored in the 801G*. Also, both the format and the display must support a minimum resolution of 640 pixels horizontally by 480 lines vertically. With all of the formats that we pre-program into the 801G*, you should be able to find one that is compatible with a display you already have. If none of the stored formats will work with the display you wish to use, you will need to program a suitable format using the 801G*'s communications ports.

To program the 801G* using the GUI, you first connect your display and find a compatible format. You should then select the "Format" test image. The image should be stable and legible on your display. You should also double check the active video data to make sure you have at least 640 pixels by 480 lines of active video.

Once you have confirmed proper operation, switch the 801G* to its programming mode:

1) Turn the 801G*'s power off.

2) Turn the power back on while holding down the "Image" button on the front panel.

3) Select a compatible format using the Format knob.

4) Rotate the "Image" knob. You will see additonal image names appear on the LCD window. Most are GUI editor screens that will appear as text on your display screen. Pressing the "Image" button so that it is lit will activate the editor that is being displayed.

You will also see an image name of "CustmImg" in the list. It is a default custom image that can be edited and saved using another name. Pressing the "Image" button will activate the custom image editor.

The generator's front panel knobs and buttons perform different functions in the GUI screens when an editor is activated. In most cases, the upper "Format" knob is used to select an item in the screen to be modified. The lower "Image" knob usually modifies or tweaks the selected item. The current button functions are identified by eight inverted text labels at the bottoms of the screens. Available button functions are shown with an intensified background on the screen and lit buttons on the front panel. Unavailable button functions are dimmed on the screen and front panel. More information on specific knob and button functions will be given in the discussion of each editor.

To switch the 801G* out of the GUI programming mode, cycle the power without holding any of the buttons down.

Format Editor

The format editor is one of the screens available in the programming mode. You can view and modify the contents of any format stored in non-volatile memory using the format editor. A typical format editor screen is shown here. The actual editor screen uses mostly lit text on a black background.

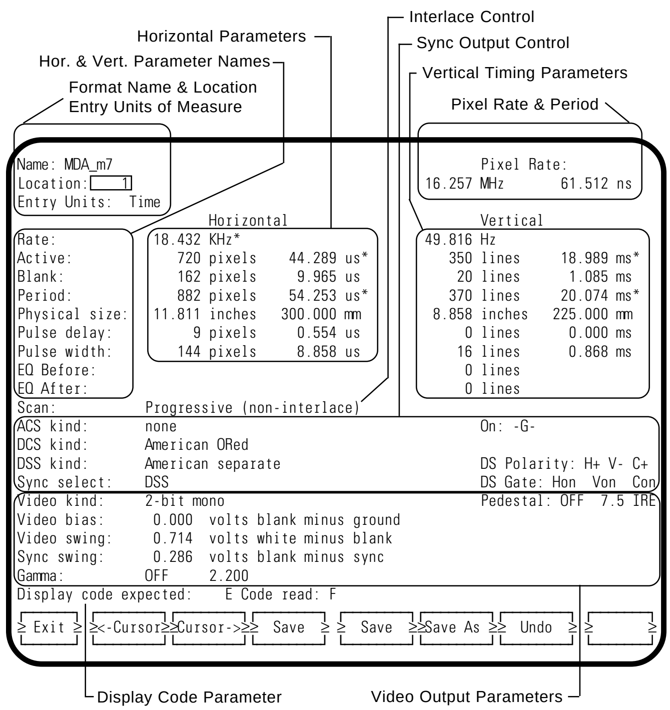


Figure 6-1
Format Editor Screen

Knob Functions

The 801G*'s top knob moves a selection box among the parameters on the screen. Knob rotation causes a “snaking” action. The exact path taken will depend on the units of measure you have selected for data entry. The example screen shows the memory location as being selected.

The bottom knob modifies the parameter that the top knob has selected. If a non-numeric parameter (e.g. sync type) is selected, the bottom knob will cycle through a list of available choices. If a numeric parameter is selected, an underscore cursor is placed below a digit in the number. The “**Cursor**” buttons move the digit cursor and the bottom knob changes the selected digit.

In many cases, entering a new value of one parameter will affect the values of other parameters as well. The editor automaticall updates all related parameters when you enter a new value.

The upper left corner shows the name and memory location number of the selected format. When you first use the editor, this information will be for format that is being used to drive the display. You select the format you wish to edit by first selecting location number parameter. Rotating the bottom knob will take you through all of the formats stored in teh 801G*. You create new formats by editing an existing format and saving your work using a new name and location.

❖ The location number is only used by the format editor. The format knob directory and test sequences all use just the format name when selecting formats from memory.

The unit of measure used for timing entries is also shown in the upper left corner. You have two choices for the units of measure. You can enter in units of real “**Time**” (µSec and mSec) or in “**Machine**” units (pixels and lines). You should use machine units for precise control of your formats. The 801G* internally rounds off real time entries to match the nearest machine unit of time. A single pixel period is the basic unit of measure for all horizontal timing parameters. All horizontal real time entries will be rounded to an integer multiple of the pixel clock period. A single horizontal period is the basic unit of measure for all vertical timing parameters. All vertical real time entries will be rounded to an integer multiple of the horizontal period. You can update your real time entries to their actual rounded values by having the calculate function on and selecting machine units of measure.

The upper right corner shows the **Pixel Rate**. The pixel clock is the master clock used to generate all of the format's timing information. The pixel clock period is also shown.

The remainder of the top half of the screen is taken up by the horizontal and vertical parameters. These parameters deal with the output signal timing and physical size of the active video area. The names of the parameters are shown at the far left of the screen. The horizontal timing and size parameters are shown in two columns. Most of the information in the left hand column is in machine units and most of information in the right hand column is in real time units. The only exception is the physical size. You can always enter the size in either inches or millimeters. The vertical parameters are set up in a similar manner.

❖ See a later section in this chapter called "Creating your own format file" for more descriptive information on format parameters and how they relate to each other.

The first line for both sets of parameters is the "**Rate**." The horizontal rate is the number of times per second your display scans in the horizontal direction. The vertical rate is number of times per second your display scans from the top to the bottom of the screen. This rate is equal to the frame rate for non-interlaced formats and twice the frame or picture refresh rate for interlaced formats.

Entering a new value for the pixel rate, horizontal rate or the vertical rate will cause the other two rates to change. This is because the newly entered rate becomes the reference used to calculate the other two. The ratio of the pixel rate to the horizontal rate will always be equal to the total number of pixels in one horizontal period. The ratio of the horizontal rate to the vertical rate will always be equal to the number of lines in one vertical period for non-interlaced formats. The ratio will always be equal to one half the number of lines in one vertical period for interlaced formats. The current reference rate is marked with an asterisk.

The **Horizontal Period** is always equal to the sum of the **Horizontal Active** and the **Horizontal Blanking** times. The 801G* uses two of these parameters as references to calculate the third. The last two edited parameters will always be used as the references. The current references are marked with asterisks. The horizontal period, in microseconds, is the reciprocal of the horizontal rate. Entering a new real time value for the horizontal period will re-calculate the horizontal rate and make it the reference rate.

The **Vertical Period** is always equal to the sum of the **Vertical Active** and the **Vertical Blank**ing periods. For interlaced formats, these periods are for an entire frame (2 fields) of video. The 801G* uses two of these parameters as references to calculate the third. The last two edited parameters will always be used as the references. The current references are marked with asterisks. The vertical period, in milliseconds, is the reciprocal of the frame rate. Entering a new real time value for the vertical period will re-calculate the vertical rate and make it the reference rate.

The horizontal and vertical **Physical size** parameters are the physical dimensions of the active video area of the display you will be driving with the format They do not affect the timing of any of the signals coming out the 801G* or the overall active video size of an image. However, they do affect the way some of the built-in test images will appear on the display's screen. The 801G* uses this information to properly draw true circles and squares as well as brightness boxes that are supposed be specific physical sizes. Using the wrong values may cause circles to look like ovals and square boxes to look like rectangles.

Horizontal Pulse delay is the period of time from the last active pixel in a scan line to the leading edge of the horizontal sync pulse. Some display spec sheets refer to this period as the horizontal sync front porch. **Horizontal Pulse width** is the width of the horizontal sync pulse itself.

Vertical Pulse delay is the period of time from the last active scan line to the leading edge of the vertical sync pulse. Some display spec sheets refer to this period as the vertical sync front porch. **Vertical Pulse width** is the width of the vertical sync pulse itself. The 801G* always uses the entered values for non-interlaced American (E.I.A.) sync types. The values used are different for other sync types:

❖ If you had a non-interlaced format that had the vertical sync delay set to 8 lines and vertical sync width set to 3 lines, a European (C.C.I.R.) selection would produce 7.5 lines of delay and 2.5 lines of sync.

❖ The 801G* blanks the half lines of active video that would appear at the top and bottom of interlaced formats. Vertical sync delay is measured from the last full line of active video in each field. If you had an interlaced format that had the vertical sync delay set to 8 lines and vertical sync width set to 3 lines, you would have the following results: An American sync selection would produce 8 lines of delay in one field and 8.5 lines of delay in the second field. There would be 3 lines of sync in both fields. A European sync selection would produce 7.5 lines of delay in one field and 8 lines of delay in the second field. There would be 2.5 lines of sync in both fields.

EQ Before sets the how many scan lines just prior to the vertical sync interval will contain pre-equalization pulses. **EQ After** sets the how many scan lines right after the vertical sync interval will contain pre-equalization pulses. The pulses are only added to composite sync outputs that have equalization pulses enabled.

The **Scan** setting determines if the video and sync timing is non-interlaced (progressive) or interlaced.

Most of the bottom half of the editor screen is taken up by the video and sync type settings. The first three lines determine the behavior of the sync selection buttons. **ACS kind** determines what type of composite sync is added to the analog video outputs when the “ACS” button is pushed. The **On** setting, to the right, selects which of the component analog video outputs will have composite sync added. **DCS kind** determines what type of digital (TTL) composite sync is generated when the “DCS” button is pushed. **DSS kind** determines what type of digital (TTL) separate horizontal and vertical syncs are generated when the “DSS” button is pushed.

The choices for the **ACS** and **DCS** sync types are:

- **none** The operator can not make this selection. Nothing happens when the button is pressed.
- **American HDTV ORed**
- **American ORed** A simple Boolean OR combination of the individual horizontal and vertical sync pulses.
- **American w/serr** Composite sync with serration pulses during the vertical sync period.
- **American w/serr & EQ** Composite sync with serration pulses during the vertical sync period and equalization pulses added before and / or after the vertical sync period. The serration and equalization pulses occur at twice the horizontal rate.
- **European HDTV ORed**
- **European ORed** A simple Boolean OR combination of the individual horizontal and vertical sync pulses. The vertical sync delay and pulse are a half line shorter than shown.

- **European w/serr** Composite sync with serration pulses during the vertical sync period. The vertical sync delay and pulse are a half line shorter than shown.

- **European w/serr & EQ** Composite sync with serration pulses during the vertical sync period and equalization pulses added before and / or after the vertical sync period. The vertical sync delay and pulse are a half line shorter than shown. The serration and equalization pulses occur at twice the horizontal rate.

- **American HDTV w/serr & EQ**

- **American HDTV w/serr**

- **European HDTV w/serr & EQ**

- **European HDTV w/serr**

- **Japanese HDTV ORed**

- **Japanese HDTV ORed w/serr & EQ**

- **Japanese HDTV ORed w/serr**

The choices for the **DSS** sync type are:

- **none** The operator can not make this selection. Nothing happens when the button is pressed.

- **American Separate** Individual horizontal and vertical sync pulse outputs.

- **American HDTV Separate**

- **European HDTV Separate**

- **Japanese HDTV Separate**

- **European Separate** Individual horizontal and vertical sync pulse outputs. The vertical sync delay and pulse width are a half line shorter than shown.

The **DS Polarity** parameter, to the right of the DSS setting, sets the logical polarities of the **H**orizontal, **V**ertical and **C**omposite digital sync outputs. Rotating the bottom knob cycles through all possible polarity combinations. A "+" setting indicatesa positive going pulse. A "-" setting indicates a negative going pulse.

The **DS Gate** parameter, below the DS Polarity parameter, sets output gating of the **H**orizontal, **V**ertical and **C**omposite digital sync outputs. Rotating the bottom knob cycles through all possible gating combinations. Gating an output "off" overrides the front panel button settings.

Sync select sets which sync outputs will be active when the format is first loaded.

Video kind sets the type of video signal you wish to generate. The 801G* supports the following video types:

- **Analog Y** Monochrome (gray scale) component video with either composite sync added to the video or separate sync(s).
- **Analog RGB** Color (red, green and blue) component video with either composite sync added to the green output or separate sync(s).
- **Analog TV Y** (801GX only) Monochrome (luminance without any color sub-carrier) composite television video with controlled rise and fall times.
- **Analog TV EYC** (801GX only) Encoded color (luminance and chrominance with color sub-carrier) composite television video with controlled rise and fall times. NTSC sub-carrier used when American ACS is selected and PAL sub-carrier used when European ACS is selected.
- **Analog TV YPrPb** Encoded color (luminance and color difference).

- **Dig. V** (801GC, 801GX only) Monochrome digital (TTL) video @ 1 bit-per-pixel
- **Dig. VI** (801GC, 801GX only) Monochrome digital (TTL) video @ 2 bits-per-pixel (separate video and intensity bits)
- **Dig. RGB** (801GC, 801GX only) Color digital (TTL) video with one bit per color (black and 7 saturated colors)
- **Dig. RGBI** (801GC, 801GX only) Color digital (TTL) video with one video bit per color plus a common intensity bit (black, 7 saturated colors, gray and 7 dimmed colors)
- **Dig. RrGgBb** (801GC, 801GX only) Color digital (TTL) video with one video bit and intensity bit per color.

The **Pedestal**, **Video bias**, **Video swing**, **Sync swing** and **Gamma** parameters only affect formats that are set for “Analog mono” or “Analog color” video types. **Pedestal** determines if a blanking pedestal is used and, if so, how far the black level is above the pedestal. The level is a percentage of the video swing. Turning the pedestal off makes blanking and black level the same. **Video bias** is the DC offset of the signal and is measured from ground to the blanking level. The current hardware designs only support a bias of 0.00 volts. **Video swing** sets the peak-to-peak amplitude of the video (including any blanking). **Sync swing** sets the peak-to-peak amplitude of composite sync (if it is selected by the ACS parameter). The **Gamma** settings determine if gamma correction is to be used and, if so, the amount of correction to be applied.

Pressing the **Check** button error checks your entries. If the format is good, you can just **Cancel** the checking operation. If errors are found, you can automatically correct some of them by having the editor **Justify** your entries.

❖ **Caution:** Saving a format with errors in it may cause major problems when you later try to load the format. Pressing the **Save** or **SaveAs** button does NOT error check your entries. You should check your edited format for errors before saving it.

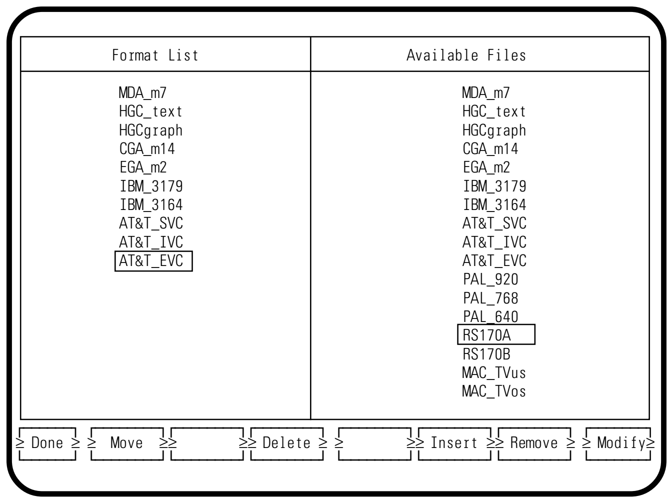
Pressing the **Save** button saves your format the name and memory location shown at the top of the screen. unless you started with an EPROM resident format. In that case, you will be prompted for a new name. See the next paragraph for saving with a new name.

If you wish to keep the original version of the format you started with, press the **SaveAs** button to save your work using a new name. A window will appear showing the name and a list of characters you can use for the new name. The top knob moves a cursor through the name being edited and the bottom knob moves a cursor through the available characters. The **Clear** button clears the name being edited. The **Delete** button removes the selected character in the name. The **Insert** button adds the selected character in the character list to the name at the current cursor location. Press the **OK** button when you are done changing the name. You can now press the “SaveAs” button.

Pressing the **Exit** button leaves the format editor and returns the 801G* to normal programming operation. If there are any unsaved changes, you will be asked if you want to save your work before exiting.

Format Knob Directory Editor

Going through all of the available formats with the Format knob can be time consuming. This is particularly true if you regularly use only a few formats that may be scattered in memory. The 801G* can be set up to show only the formats you want, in the order you want, when the “Format” knob is turned. The 801G* maintains a list of these formats in non-volatile memory. This list can be edited. You will need to edit the list if you wish to use the knob with formats you have created. A typical format knob



directory editor screen is shown here. The actual editor screen uses mostly lit text on a black background.

The left hand side of the screen shows the current contents of the format knob directory. The formats appear in the order that they are selected by the “Format” knob during normal operation.

While editing, the “Format” knob moves a selection box through the left hand list. The list will scroll if it can not fit on the screen.

The right hand side shows all of the formats that are in the 801G*. They are listed in order of ascending memory locations. Empty or corrupted memory locations are skipped.

The bottom “Image” knob moves a selection box through the right hand list. The list will scroll if it can not fit on the screen.

The **Insert** button moves the selected format in the left hand column and all of the formats below it down one line. It then copies the selected format on right hand side into the vacated position.

The **Delete** button removes the selected format from the left hand column. All formats below the deleted format move up.

Pressing the **Move** button lets you move the selected format in the left hand column to another position in the list. The format is moved by rotating the top knob. Pressing the “Move” button a second time will leave the format in the new position.

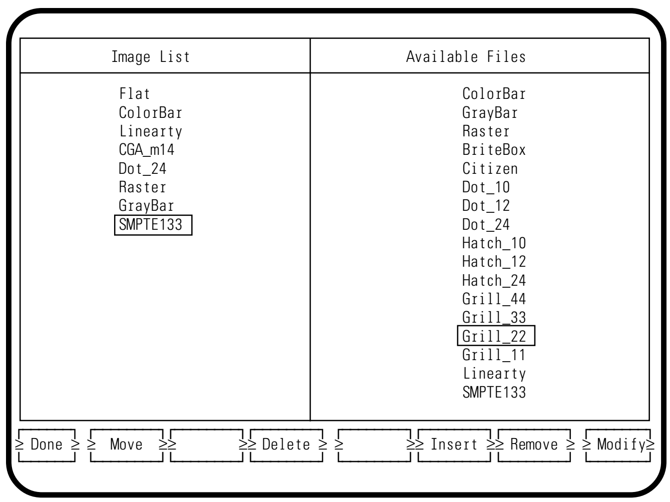
Pressing the **Remove** button lets you delete the format file that is selected in the right hand column. This will erase the entire contents of the file from non-volatile memory. Since this operation can destroy a lot of programming work, an on-screen message will ask you to confirm the removal of the file. Pressing the **Yes** or **OK** button will erase the file. Pressing the **No** or **Cancel** button will abort the removal operation.

Pressing the **Modify** button activates the format editor for the format that is selected in the right hand column. You can use this feature to check and modify a format before adding it to the knob directory. Exiting the format editor will automatically return you to the format knob directory editor.

Pressing the **Done** button saves the edited list and exits the editor.

Image Knob Directory Editor

Going through all of the available images with the Image knob can be time consuming. This is particularly true if you regularly use only a few images that may be scattered over the knob locations. The 801G* can be set up to show only the images you want, in the order you want, when the “Image” knob is turned. The 801G* maintains a list of these images in non-volatile memory. This list can be edited. You will need to edit the list if you wish to use the knob with images you have created. A typical image list editor screen is shown here. The actual editor screen uses mostly lit text on a black background.



The left hand side of the screen shows the current contents of the image knob directory. The images appear in the order that they are selected by the “Image” knob during normal operation.

While editing, the “Format” knob moves a selection box through the left hand list. The list will scroll if it can not fit on the screen.

The right hand side shows all of the available images. The built-in images are at the top of the list. The built-in images are followed by a factory default custom image. This image is used a starting point to creating you own custom images. Any user created custom images in non-volatile memory are at the bottom of the list.

The bottom “Image” knob moves a selection box through the right hand list. The list will scroll if it can not fit on the screen.

The **Insert** button moves the selected image in the left hand column and all of the images below it down one line. It then copies the selected image on right hand side into the vacated position.

❖ The current firmware does not allow entering a secondary version of a built-in image to the list. The primary version will need to be first selected during normal operation. Pressing the “Image” button will then select the secondary version.

The **Delete** button removes the selected image from the left hand column. All images below the deleted image move up.

Pressing the **Move** button lets you move the selected image in the left hand column to another position in the list. The image is moved by rotating the top knob. Pressing the **Move** button a second time will leave the image in the new position.

If a custom image is currently selected in the right hand column, pressing the **Remove** button lets you delete that image file. This will erase the entire contents of the file from non-volatile memory. Since this operation can destroy a lot of programming work, an on-screen message will ask you to confirm the removal of the file. Pressing the **Yes** or **OK** button will erase the file. Pressing the **No** or **Cancel** button will abort the removal operation without erasing the file.

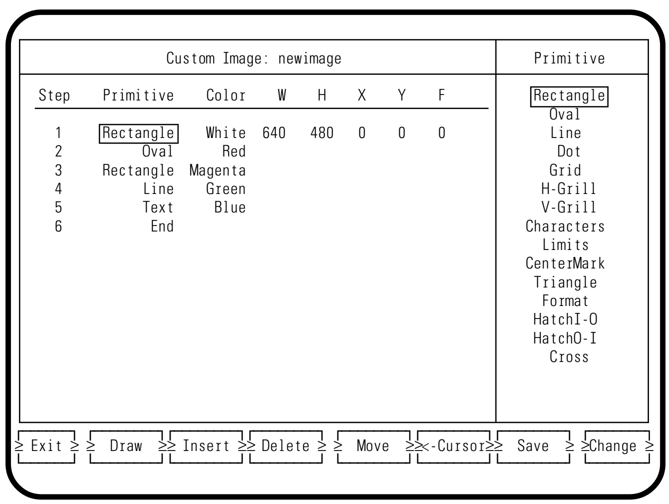
If a custom image is currently selected in the right hand column, pressing the **Modify** button activates the image editor for that image. You can use this feature to check and modify an image before adding it to the knob directory. Exiting the custom image editor will automatically return you to the image knob directory editor.

Pressing the **Done** button saves the edited list and exits the editor.

Custom Image Editor

There may be times when none of the 801G*'s many built-in test images (patterns) quite meet your requirements. In these cases, you may be able to create a custom image to match your exact requirements. A custom image consists of one or more simple geometric objects and alphanumeric characters (primitives). The color, relative size and relative position of the objects are all editable parameters for most of the primitives. The images scale to the active video size of the currently loaded format. A custom image can be saved in non-volatile memory and recalled just like any of the built-in images.

Images are created using the custom image editor. It is one of the screens available in the programming mode. A typical image editor screen is shown here. The actual editor screen uses mostly lit text on a black background.



The left side of the screen displays the contents of the custom image. This includes the type, ordering and associated parameters for each drawing primitive in the image. The primitives are drawn in the order shown. The parameters in the far right hand column will change depending on the type of primitive selected.

The right side of the screen displays the parameter currently being modified. All parameter changes are made on this side before being entered to the left side.

The 801G*'s upper "Format" knob moves a selection box among the steps and parameters on the left side of the screen. The knob moves the selection box horizontally across the screen until the step number or last parameter is selected. Then the next move will be to the adjacent row. This causes a "snaking" action. If the list goes beyond the bottom of the screen, turning the knob further clockwise will cause the list to scroll up. The example screen has the rectangle primitive in step #1 selected.

The lower "Image" knob modifies the parameter that the top knob has selected. If a non-numeric parameter (e.g. primitive type) is selected, the right side of the screen will show a list of available primitives and the bottom knob will move a selection box up and down through this list. Long lists will scroll up and down on the screen. If a numeric parameter is selected, the value is copied to the right side of the screen and an underscore cursor is placed below a digit in the number. The **<-Cursor** button moves the digit cursor and the bottom knob changes the selected digit. Pressing the **Change** button copies the entry on the right side to the left side.

Available Primitives: The table on the next page lists all of the drawing primitives supported by the current firmware.

❖ The size and location coordinates shown in the editor are in units of pixels. The values that actually are saved in memory are based on percentages of the total number of active pixels and lines in the format. This means that your image will look about the same no matter how many active pixels and lines are in the format at the time the image is drawn. This also means that if you create custom image with one format and edit it with another, the numbers in the editor screen will change for the same image.

Table 4-1
Available drawing primitives in a Custom Image

Primitive Name	Parameters
Rectangle	color, width, height, left, top, fill patn
Oval	color, width, height, left, top, fill patn
Line	color, x1, y1, x2, y2
Dot	color, x, y
Grid	color, # of xboxes, # of yboxes
H-Grill	color, line width
V-Grill	color, line width
Characters	color, character #, font#
Limits	color
CenterMark	color
Triangle	color, x1, y1, x2, y2, x3, y3, fill patn
Format	color, left x, top y
Hatch_I-O	color, # of xboxes, # of yboxes
Hatch_O-I	color, # of xboxes, # of yboxes
Cross	color
Text	color, x, y, font#, text string
Seq. Step	color, x, y

Available Colors: The color list shows 63 colors and grays as being available. However, the 801G* can only work with 16 colors at a time. One of the colors must be black. Attempting to use more than 16 colors may cause parts of the image to draw with the wrong color. A number in the color name refers to its % intensity level. The current firmware does not support editing the colors or gray levels

Table 4-2
Available colors in a Custom Image

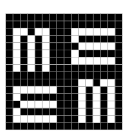
Black	Cyan50	Gray33	Gray70	Yellow75
Red	Brown	Gray37	Gray73	Blue75
Green	Gray3	Gray40	Gray75	Magenta75
Yellow	Gray5	Gray43	Gray77	Cyan75
Blue	Gray7	Gray47	Gray80	HueI
Magenta	Gray10	Gray48	Gray83	HueQ
Cyan	Gray13	Gray50	Gray87	HueNegI
White	Gray17	Gray51	Gray90	HueNegQ
Red50	Gray20	Gray53	Gray93	Foreground
Green50	Gray23	Gray57	Gray95	Background
Yellow50	Gray25	Gray60	Gray97	Clear
Blue50	Gray27	Gray63	Red75	
Magenta50	Gray30	Gray67	Green75	

Table 4-3
Fill patterns in a Custom Image

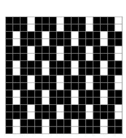
The "fill patn" parameter determines if a primitive is drawn as a one pixel thick outline or as a pattern filled object. Fill patterns consist of various on-off pixel combinations in a repeating 16 x 16 pixel block. A setting of "graypatn0" has all fill pixels turned off and a setting of "graypatn100" has all pixels on.

grayoat0	0% of pixels on (No fill)
graypat7	7% of pixels on
graypat13	13% of pixels on
graypat19	19% of pixels on
graypat25	25% of pixels on
graypat31	31% of pixels on
graypat38	38% of pixels on
graypat44	44% of pixels on
graypat50	50% of pixels on
graypat63	63% of pixels on
graypat69	69% of pixels on
graypat75	75% of pixels on
graypat81	81% of pixels on
graypat88	88% of pixels on
graypat94	94% of pixels on
graypat100	94% of pixels on (Solid fill)
checker1	Checkerboard alternating one (1) pixel on and one (1) off
checker2	Checkerboard alternating two (2) pixels on and two (2) off
checker3	Checkerboard alternating three (3) pixels on and three (3) off
checker4	Checkerboard alternating four (4) pixels on and four (4) off
bars_V1	Vertical bars one (1) pixel wide
bars_V2	Vertical bars two (2) pixels wide
bars_V4	Vertical bars four (4) pixels wide
bars_V8	Vertical bars eight (8) pixels wide
bars_H1	Horizontal bars one (1) pixel wide
bars_H2	Horizontal bars two (2) pixels wide
bars_H4	Horizontal bars four (4) pixels wide
bars_H8	Horizontal bars eight (8) pixels wide
meme	Repeating MEME pattern
FCC_EMI	Repeating three (3) pixels on and one pixel off pattern

meme
Fill pattern



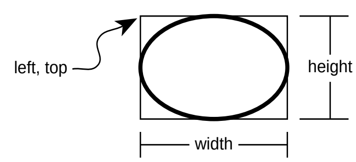
FCC_EMI
Fill pattern



The Rectangle primitive draws a rectangle whose sides are parallel to the vertical and horizontal axis of displayed video. The primitive uses six (6) parameters. The first is the color. The next two parameters are the width and height of the rectangle in pixels. The second two parameters are the X and Y coordinates for the top left-hand corner of the rectangle. The last parameter is the fill flag. The following example draws a 50% gray rectangular filled solid that is 15 pixels wide by 20 pixels high and has its top left corner located 50 pixels to the right and 40 pixels below the top left corner of active video:

```
Rectangle  Gray50  15  20  50  40  1
```

The Oval primitive draws an oval whose axes are parallel to the vertical and horizontal axis of displayed video. The size and position of the oval is defined by its framing rectangle. The framing rectangle is a rectangle whose sides are both tangent to the oval at four points and are parallel to the vertical and horizontal axis of displayed video. The framing rectangle is not drawn as part of the primitive. The figure shows the relationship of an oval to its framing rectangle.



The primitive uses six (6) parameters. The first is the color. The next two parameters are the width and height of the framing rectangle in pixels. The second two parameters are the X and Y coordinates for the top left-hand corner of the framing rectangle. The last parameter is the fill flag. The following example draws a red oval that is 240 pixels wide by 150 pixels high and has the framing rectangle's top left corner located 20 pixels to the right and 10 pixels below the top left corner of active video:

```
Oval  Red  240  150  20  10  0
```

The Line primitive draws a line between any two points. The line is one pixel thick. The primitive uses five(5) parameters, the color and the X and Y coordinates for both endpoints. The following example draws a yellow line between a point 20 pixels to the right and 5 pixels below the top left corner of active video and a point 320 pixels to the right and 240 pixels below the top left corner of active video:

```
Line  Yellow  2Ø  5  32Ø  24Ø
```

The Dot primitive draws a single pixel dot. A dot is the smallest graphic element that can be drawn. The primitive uses three (3) parameters, the color and the X and Y coordinates. The following example draws a white dot that is at a point 200 pixels to the right and 300 pixels below the top left corner of active video:

```
Dot  White  2ØØ  3ØØ
```

The Grid primitive draws a crosshatch of a given color and forming a given number of boxes in each direction. All lines are 1 pixel thick. All of the lines, in a given direction, are equally spaced. Any remaining pixels are distributed as equally as possible around the perimeter of the grid. This may cause the first and last lines in each direction not to be at the very edges of video. The following example draws a 75% gray level grid that has 14 boxes horizontally and 10 boxes vertically:

```
Grid  Gray75  14  1Ø
```

The H-Grill primitive draws equally spaced horizontal lines that form a grill over the entire active video area. The gap between the lines is equal to the thickness of the lines. The gaps are not touched and will show any previously drawn primitives. The primitive uses two (2) parameters. The first is the color of the lines. The second is the thickness of the lines. The following example draws a green horizontal grill whose lines are 4 pixels thick and have 4 pixel gaps between the lines:

```
H-Grill  Green  4
```

The V-Grill primitive draws equally spaced vertical lines that form a grill over the entire active video area. The gap between the lines is equal to the thickness of the lines. The gaps are not touched and will show any previously drawn primitives. The primitive uses two (2) parameters. The first is the color of the lines. The second is the thickness of the lines. The following example draws a cyan vertical grill whose lines are 16 pixels thick and have 16 pixel gaps between the lines:

```
V-Grill  Cyan  16
```

The Characters primitive fills the active video area with a single repeating character. The spacing between the characters is controlled by the size of the two dimensional pixel array that is used to define the character. The size of the pixel array is determined by the character and font selected.

Table 4-4
Available font sets for the Characters and Text primitives

#0 sys16	IBM-type alphanumeric font that has printable characters for ASCII codes 0-126. It uses an 8 x 16 monospaced character block.
#1 OPIX 9	Alphanumeric font that has printable characters for ASCII codes 32-126. It uses a 5 x 7 monospaced character block.
#2 focusmac	Single character used in the Focus_Oo test image; ASCII code = 79; 8 x 6 character block
#3 focus_12	Two characters used for the Focus_Cx and Focus_H test images; ASCII codes = 67 and 72; 8 x 6 character blocks
#4 memesony	SinglecharacterusedforvariousMEMEimages;ASCII code = 77; 18 x 18 character block
#5 kanjikan	Single Japanese KanjiKan character used in the KanjiKan image; ASCII = 75; 22 x 22 character block
#6 focusat 5	Asingle@characterusedintheLinFocusimage;ASCII = 64; 8 x 16 character block
#7 focusat 6	Asingle@characterusedintheFocus@6image;ASCII = 64; 16 x 16 character block
#8 focusat 7	Asingle@characterusedintheFocus@7image;ASCII = 64; 16 x 16 character block
#9 focusat 8	Asingle@characterusedintheFocus@8image;ASCII = 64; 16 x 16 character block
#10 memeplus	A meme plus character used in focus and convergence; ASCII = 77; 43 x 49 character in a 49 x 51 block

The primitive uses three (3) parameters. The first is the color. The second parameter is the code number of the character. For most alpha-numeric fonts, the character number will be the same as the decimal ASCII code number of the character. The last parameter is the number of the font library.

The current firmware includes ten font sets. The fonts are described in the table on the previous page. Font sets #Ø and #1 have a full set of ASCII upper and lower case letters plus numbers and punctuation marks. Font #Ø is used for the GUI editor screens. The remaining font sets only have one or two characters each. These are used bu the focus checking test images.

The following example draws white upper case letter “X” characters from font set #0:

```
Characters  White  88  Ø
```

The Limits primitive places nine (9) markers that define the active video area. An “L” shaped marker is positioned at each corner. “T” shaped markers are centered along each edge of video and a small cross is placed at the center of video. The primitive uses a single parameter, the color used for all of the markers. The following example draws white markers:

```
Limits  White
```

The Centermark primitive draws a small cross in the center of active video. The vertical line is 2 pixels thick if the format has an even number of active pixels. The horizontal line is 2 pixels thick if the format has an even number of active lines. The primitive uses a single parameter, the color of the cross. The following example draws a red marker:

```
Centermark  Red
```

The Triangle primitive draws a triangle defined by its three endpoints. The primitive uses eight (8) parameters. The first is the color. The next three pairs of parameters are the X and Y coordinates for the three endpoints. The last parameter is the fill flag. More complex filled polygons can be built up using a series of joined filled triangles. The following example draws a 50% red filled triangle that looks like an arrowhead pointing to the right near the top left corner of active video:

```
Triangle   Red50   10   5   10   25   40   15   1
```

The Format primitive shows some basic information about the format that is driving the display. The first line shows the number of horizontal active pixels and vertical active lines. The last number on the line shows the number of fields per frame (1 for non-interlaced and 2 for interlaced). The second line shows the horizontal rate and the third line shows the vertical rate. The text is on a black rectangular background with a single pixel border.

The primitive uses three (3) parameters. The first is the color of the text and border. The next two parameters are the X and Y coordinates for the top left-hand corner of the block of text. The following example draws a blue block format data block 30 pixels to the right and 200 pixels below the top left corner of active video:

```
Format   Blue   30   200
```

The Hatch_I-O primitive draws a crosshatch from the "Inside-Out" of a given color and forming a given number of boxes in each direction. The primitive always has center lines that divide the active video exactly in half in each direction. The vertical center line is 2 pixels thick if the format has an even number of active pixels. The horizontal center line is 2 pixels thick if the format has an even number of active lines. All other lines are 1 pixel thick. If an odd number of boxes is entered for a given direction, a half box will be placed at each end of the crosshatch. All of the lines, in a given direction, are equally spaced. Any remaining pixels are distributed as equally as possible around the perimeter of the grid. This may cause the first and last lines in each direction not to be at the very edges of video. This may also cause any half boxes to be slightly larger. The following example draws a yellow crosshatch that has 15 boxes horizontally and 9 boxes vertically:

```
HatchI-O   Yellow   15   9
```

The Hatch_O-I primitive draws a crosshatch from the “Outside-In” of a given color and forming a given number of boxes in each direction. All lines are 1 pixel thick. The first and last lines in each direction are at the very edges of active video. All of the lines, in a given direction, are equally spaced. Any remaining pixels are added to the boxes along the horizontal and vertical centers of the image. The following example draws a green crosshatch that has 15 boxes horizontally and 9 boxes vertically:

```
HatchO-I   Green   15   9
```

The Cross primitive draws a large centered cross that fills the active video area. The vertical line is 2 pixels thick if the format has an even number of active pixels. The horizontal line is 2 pixels thick if the format has an even number of active lines. The primitive uses a single parameter, the color of the cross. The following example draws a magenta cross:

```
Cross   Magenta
```

The Text primitive draws a user defined text string. The primitive uses five (5) parameters. The first is the color. The next two parameters are X and Y coordinates for the location of the top left corner of the text. The fourth parameter is the number of the font set. A list of available font sets is shown on an earlier page.

The last parameter is the text string to be drawn. When you edit this parameter, a small window pops up that shows the current string contents and a character list. The knobs and keys are used to modify the string.

The following example draws a white “Hello World” text string near the top left corner of the screen using font set #0:

```
Text   White   10   10   0   Hello World
```

The Seq. Step primitive only appears if the image is used in a test sequence that has step number display enabled. It draws the step number that the image is a part of. The primitive uses three (3) parameters. The first is the color. The next two parameters are X and Y coordinates for the location of the top left corner of the text box.

The following example draws a cyan step number box near the top left corner of the screen:

```
Seq.Step  Cyan 10  10
```

The "**Insert**," "**Delete**" and "**Move**" buttons are used to modify the drawing list. These buttons only work when a step number is selected. The "Insert" button puts a blank step at the selected position. All steps below the insertion point are moved down. The "Delete" button removes the selected step. All steps below the deleted step move up. Pressing the "Move" button lets you move the selected step to another position in the list. The step is moved by rotating the top knob. Pressing the "Move" button a second time will leave the step in the new position.

You can test your image at any time by pressing and holding down the "**Draw**" button. Releasing the button will change the display back to the editor screen.

Pressing the "**Save**" button shows a window with the name that will be used to save the image file to non-volatile memory. The original file name is shown. Saving back to the same file name will overwrite the original contents. If you wish to keep the original version of your custom image, you need to change the name in the window using the bottom knob and buttons. Saving a new image file does not automatically add it to the Image list.

Pressing the "**Exit**" button leaves the custom image editor and returns the 801G* to normal operation. If there are an unsaved changes, you will be asked if you wish to save your work prior to exiting.

What is a Sequence?

The normal operating mode of the 801G* uses the top knob to select a format and the bottom knob to select a test image. The production testing of a multi-mode display may require the repeated use of several different formats and images in a given order. A sequence lets you pair up formats and images to form a single test step. The steps are organized to match your test procedure for a particular display. The rotation of a single knob then allows an operator to step forward and backwards through the sequence. You can also have the sequence continuously cycle through all the steps, stopping for individually defined amounts of time at each step. This mode is useful for burn-in testing or for running displays at trade shows. To make running a sequence a bit more foolproof, you can program the 801G* to power-up in the sequence mode. You can control which sequences files an operator can access in the normal operating mode.

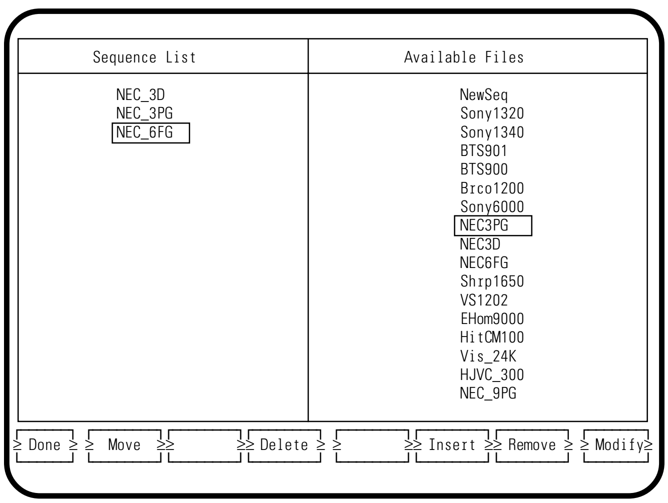
More than one sequence file can be stored in non-volatile memory. The number of sequence files you can store depends on the number of steps in each file. A total of about 1280 to 1500 steps can be saved, depending on how many separate files are used.

Before you can use a sequence file, you need to create one. And, in order to do that, you need to use the "Sequence File" editor and "Sequence Knob directory" editor.

Sequence Knob Directory Editor

The only way to get to the Sequence file editor is via the Sequence Knob directory editor. We will first go through the use of the Knob directory editor and then explain the use of the Sequence editor.

The 801G* can be set up to allow an operator to run only certain sequence files in the sequence mode. The 801G* maintains a list of these accessible sequences in non-volatile memory. You will need to edit the list if you wish to add sequences you have created. A typical sequence knob directory editor screen is shown here. The actual editor screen uses mostly lit text on a black background.



The left hand side of the screen shows the current contents of the sequence knob directory. They appear in the order that they are selected by the upper knob in the sequence mode of operation.

While editing, the “Format” knob moves a selection box through the left hand list. The list will scroll if it can not fit on the screen.

The right hand side shows all of the sequences that are in the 801G*. They are listed in order of ascending memory locations.

The bottom "Image" knob moves a selection box through the right hand list. The list will scroll if it can not fit on the screen.

The **Insert** button moves the selected sequence in the left hand column and all of the sequences below it down one line. It then copies the selected sequence on right hand side into the vacated position.

The **Delete** button removes the selected sequence from the left hand column. All sequences below the deleted sequence move up.

Pressing the **Move** button lets you move the selected sequence in the left hand column to another position in the list. The sequence is moved by rotating the top knob. Pressing the "Move" button a second time will leave the sequence in the new position.

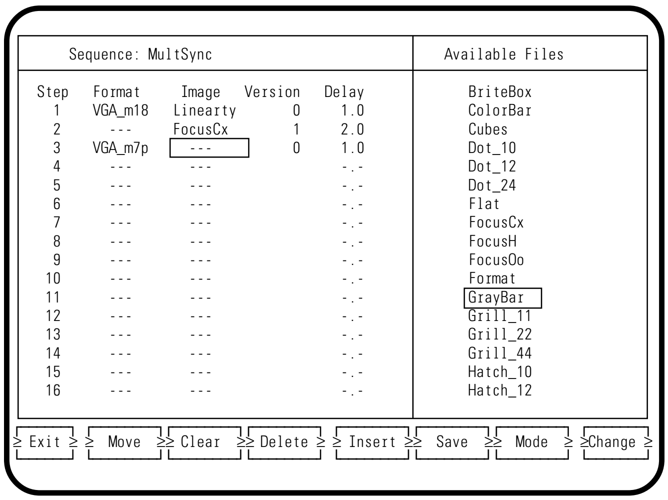
Pressing the **Remove** button lets you delete the sequence file that is selected in the right hand column. This will erase the entire contents of the file from non-volatile memory. Since this operation can destroy a lot of programming work, an on-screen message will ask you to confirm the removal of the file. Pressing the **Yes** or **OK** button will erase the file. Pressing the **No** or **Cancel** button will abort the removal operation.

Pressing the **Modify** button activates the sequence editor for the sequence that is selected in the right hand column. You can use this editor to check, create or modify a sequence before adding it to the knob directory. The file called "NewSeq" is the factory default sequence file. It can be used as a starting point in creating new sequences. Exiting the sequence editor will automatically return you to the sequence knob directory editor.

Pressing the **Done** button saves the edited list and exits the editor.

Sequence Editor

Sequences are created and modified using the sequence editor. It is accessed through the sequence knob directory editor in the programming mode. A typical sequence editor screen is shown here. The actual editor screen uses mostly lit text on a black background.



Most of the left side of the screen displays the contents of the sequence. This includes the step number, format, image and auto-sequence delay time for each step. The name of the sequence file being edited is shown at the top of the left side. If the label "Mode:" appears to the right of the name, it means the 801G* will normally power-up in the given "sequence" mode using this sequence file.

The information shown on right side of the screen will depend on what is selected on the left side of the screen. If you are in the "Step" column, the right side will be blank. If you are in the "Format" column, a list of all available formats stored in the 801G* is shown. If you are in the "Image" column, a list of all available built-in and custom images is shown. If you are in the "Version" column, a zero (0) or a one (1) is shown. If you are in the "Delay" column, a list of available delay times is shown.

The top knob moves a selection box among the steps and parameters on the left side of the screen. The knob moves the selection box horizontally across the screen until the step number or last parameter is selected. Then the next move will be to the adjacent row. This causes a “snaking” action. If the list goes beyond the bottom of the screen, turning the knob further clockwise will cause the list to scroll up. The example screen shows a blank “Image” entry selected.

The bottom knob normally moves a selection box up and down the list of choices on the right side. Long lists will scroll up and down on the screen. The **Change** button copies the entry on the right side to the left side. If the sequence number is selected on the left side, the bottom knob is used to go through the list of all sequence files in memory. The example screen shows “GrayBar” selected.

Leaving a Format and/ or Image entry for a given step empty, will cause the 801G* to use the last format or image shown above the empty position(s). The example sequence shown would use format *VGA_m18* for step #2 and the *FocusCx* image for step #3. It does not matter if the operator is going forward or backward through the steps.

The **Insert**, **Delete** and **Move** buttons are used to modify the sequence list. These buttons only work when a step number is selected. The “Insert” button puts a blank step at the selected position. All steps below the insertion point are moved down. The “Delete” button removes the selected step. All steps below the deleted step move up. Pressing the “Move” button lets you move the selected step to another position in the list. The step is moved by rotating the top knob. Pressing the “Move” button a second time will leave the step in the new position.

Pressing the **Mode** button cycles the power-on sequence mode settings as indicated by the "Mode" label to the right of the sequence name at top of the screen. The following settings are avaiable:

No label: The sequence is not selected to to be the power-on sequence.

Step: The steps are manually selected with the bottom knob, with stops at both ends of the list.

Wrap: The steps are manually selected with the bottom knob. Going past the last step jumps to the first step and vice versa.

Auto: The sequence automatically and continously cycles through all the steps. The individual "Delay" settings for each step determine how long the generator pauses at each step.

Step-Display, **Wrap-Display** and **Auto Display:** Work the same way as the previous choices with the addition of the step number being added to each test image.

❖ Only one sequence file can be the "Power-on" sequence file at a time. Making the sequence you are editing the "Power-on" sequence will deactivate the "Power-on" mode status of all other sequence files in memory. If no sequence is selected as the "Power-on" sequence, the 801G* will then power-up in the "Normal" operating mode.

Pressing the **Save** button shows a window with the name that will be used to save the sequence file to non-volatile memory. The original file name is shown. Saving back to the same file name will overwrite the original contents. If you wish to keep the original version of your sequence, you need to change the name in the window using the knobs and buttons. Saving a new sequence file does not automatically add it to the sequence knob directory.

Pressing the **Exit** button leaves the sequence editor and returns to the sequence knob directory editor. If there are any unsaved changes, you will be asked if you want to save your work before exiting.

Running a Sequence

The 801G* will power-up in the sequence mode under the following conditions:

- One of the sequence files stored in the 801G* is programmed as the "Power-on" sequence. The previous section on the sequence editor has information on how to do this
- None of the front panel buttons are held in when the 801G* is powered up.

Step #1 of the "Power-on" sequence is selected at the end of the power-up cycle. The top row of the the LCD window will show the name of the sequence and the bottom row will show the step number.

The lower "Image" knob is used to move forward and backward through the steps in the sequence.

A manual mode sequence can be set to **continuously cycle** through all the steps, by rotating the bottom knob a full turn clockwise beyond the last step A message on the LCD will confirm that the cycle is running. The cycle is stopped by rotating the bottom knob counter-clockwise.

❖ The 801G* will remember if the sequence was in continuous cycle on power down. If so, it will continuous cycle on power on.

The upper "Format" knob is used to load other sequence files. The list of files that can be selected by the knob is modified using the sequence knob directory editor in the programming mode.

❖ Caution: A programmer can make any sequence the "Power-on" sequence without having it appear on the sequence knob directory. The 801G* will power-up in the sequence mode with the selected sequence. However, loading another sequence file with the upper knob, will not allow you to get back to the original power-up sequence file.

Cancelling the Sequence Mode

You cancel the power-up sequence mode with the following steps:

- Power-up the 801G* in the programming mode ("Image" button held down on power-up).
- Select the sequence knob directory editor.
- Press the "Modify" button. It does not matter which sequence file is selected in the right hand column. The will put you in the sequence editor.
- Press the **Mode** button at least one time. Keep pressing and releasing the button until the "Mode:" label to the right of the sequence name goes away.
- Turn off the power.

The next time the 801G* is turned on, it will be in the normal operating mode. The top knob will select formats and the bottom knob will select images.

Creating your own format file

There are two ways to add signal formats to the 801G*. You can create and edit formats using the built-in GUI editor that is discussed in an earlier section. You can also create format files on a computer and download them to the 801G* using either the RS-232 or IEEE-488 ports.

Before you can download a format file, you have to have to create it on your computer. The files contain ASCII text that defines all of the format's parameters. An example of such a file is shown on the next page. You can use any text editing program on your computer that can save your work as straight ASCII text. This section of the manual has information on how to define a format for a particular display. Other parts of the manual cover connecting a computer and down loading data.

You will need to know the signal specifications of the display in order to enter your format. Most of the data can be found in a detailed specification sheet for the display. The rest of this section will tell you how to convert the spec sheet data into a usable format.

Entering the format parameters

We are now ready to start entering the parameters. One of the commands available on the 801G* is FMTN. This command initializes all of the parameters to known default values. The default values are shown in the description of the FMTN command. You do not need to enter the parameter and its value if the value matches the FMTN setting. Detailed discussions of each parameter are presented in alphabetical order at the end of this chapter.

The first step is to determine if you are working with an analog or a digital display. An analog display can show an endless range of colors or shades of gray. Analog displays normally accept video sig-

ANALOG_3.CMD Listing

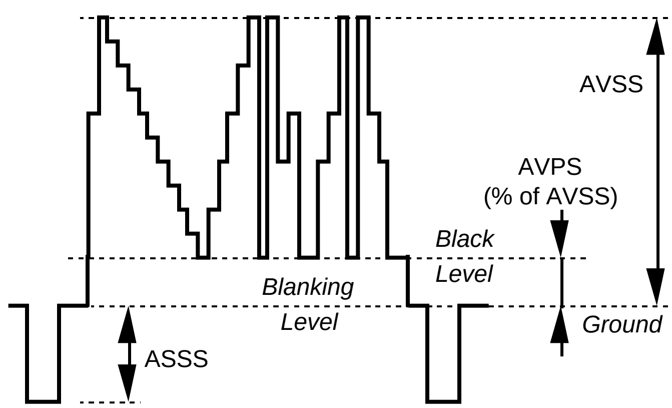
```
fmtn
fmtb
name Analog_3
hrat 31.469e3
hres 640
htot 800
hspd 16
hspw 96
hspp 0
hspg 1
vres 480
vtot 525
vspd 10
vspw 2
vspp 0
vspg 1
equb 0
equa 0
scan 1
avst 2
avss 0.700
avps 7.5
avpg 0
avsb 0.0
asss 0.300
assg 0, 1, 0
asct 2
dsct 2
dsst 1
ssst 1
cspp 0
cspg 1
xvsg 1, 1, 1
gama 2.2
gamc 0
usiz 2
hsiz 280.0
vsiz 210.0
dcex 14
dcbm 15
fmtg?
fmte
fmtw 62
```

nal levels in the 1 Vp-p range. Digital displays are more limited and can only display a couple of gray levels or a handful of pre-defined colors. These displays normally accept TTL video signals in the 4 Vp-p range.

Analog Video and Sync Parameters

If you are entering a digital video format, skip this section of the manual and go on to the “Digital Video Parameters” section. If you are working with an analog video display, set DVST to zero. Enter the correct analog video signal type for the AVST command. Both monochrome and color signal types are supported.

The following figure shows a typical video waveform for either a monochrome or RGB color display:



❖ A few displays require an inverted waveform with negative-going peak video and positive going sync tips. The 801G* will not properly drive these types of displays.

The AVSS parameter sets the peak-to-peak swing of the video portion of the composite waveform. This swing includes any blanking pedestal that may be present and excludes any sync pulses that may be added should analog composite sync be enabled. If you need a swing other than the default FMTN setting, enter it in the "User" column. In case of an RGB color signal, the swing given is common to all three color analog outputs.

Some displays require that the video output level be referenced or biased to a given voltage. The AVSB parameter sets the DC offset level. The offset is measured between ground and the blanked portion of the video signal. A postive value indicates an offset above ground.

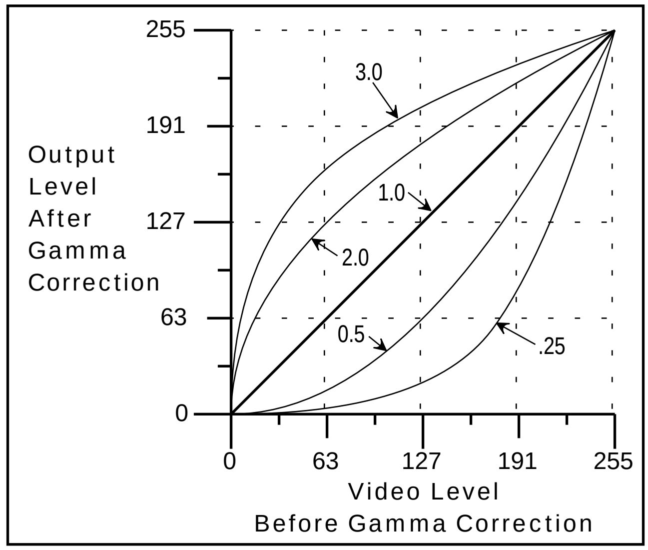
Some displays require a blanking pedestal (i.e. a blacker-than-black level during the horizontal and vertical retrace intervals). If your display requires such a blanking pedestal, set the AVPG parameter to 1 and enter the size of the pedestal you want using AVPS. The AVPS parameter is given as a percentage of the overall peak-to-peak swing of the video signal AVSS. The default FMTN value of 7.5 represents the typical blanking level found in EIA standards.

You may need to use gamma correction if you are testing an analog video display that has a non-linear light response to linear video voltage input. Most displays on the market today are of this type. Gamma correction distorts the levels output by the generator so as to produce a linear light output on the face of the CRT display with varying levels of gray. The following formula shows how the value of the gamma affects the output linearity of the generator.

$$\text{Adjusted Level} = 255 \times (\boldsymbol{v} / 255)^{(1/\gamma)}$$

(*to nearest integer value below 256*)

Where $\boldsymbol{v}$ is the uncorrected value having a range of 0 (black) to 255 (white) and γ is the gamma correction factor. A gamma correction factor of 1.000 is the same as no gamma correction. The graph in the following figure shows the logarithmic variation of the output signal when gamma is set to 0.25, 0.5, 1, 2, and 3.



Set the GAMC parameter to 1 to enable gamma correction or 0 to disable gamma correction. When gamma correction is enabled, the value indicated by the parameter GAMA is used. The FMTN value of 2.2 is standard for North American television. European television standards commonly specify a value of 2.8.

Next, determine how the display receives horizontal and vertical synchronizing information. Some displays have one or two separate TTL digital sync inputs. Other displays require composite sync on the bottom of the analog video signal. Some multimode displays accept multiple types of sync.

If the display only requires separate digital sync inputs, you can leave the remaining analog video and sync parameters at their default FMTN settings and go on to the section on “Digital Sync Parameters.” This will prevent composite sync from ever appearing on the analog video outputs.

Most color displays that use separate RGB analog video have separate sync inputs or expect sync added to the green video input. The ASSG parameter specifies which primaries output sync when analog composite sync is selected. The current design of the 801G* only allows adding sync to the green output. The only valid entries for ASSG are (0, 0, 0) and (0, 1, 0). In the case of monochrome displays that use composite video and sync, the display must be connected to the green output channel on the 801G*.

You will also need to specify the type of analog composite sync signal that is to be generated. You can specify a simple OR of the vertical and horizontal pulses or a more elaborate sync signal consisting of serrations and equalizing pulses. The ASCT command selects the exact type of sync that is added. A setting of 0 means that analog composite sync cannot be selected.

The ASSS parameter sets the peak-to-peak swing of the sync portion of the composite analog video waveform.

You can now skip the next two sections and go directly to the section on “Timing Parameters.”

Digital Video Parameters

The current version of the 801G* firmware can generate five different types of digital video signals. All of the outputs are at TTL levels. There are two parameters that control the digital video configuration of the 801G*. The DVST parameter selects from the different types that are available. AVST must be set to zero when digital video operation is selected.

The DVSP parameter specifys the logic sense (or polarity) of the digital video outputs. The current hardware configuration of the 801G* only supports the active high (or positive) polarity.

All digital video displays require digital sync. Information on setting the digital sync parameters is presented in the next section.

Digital Sync Parameters

Some analog video and all digital video displays require one or two digital sync signals. The paramters associated with these signals can be found at the top of the right-hand column of the format worksheet.

If your display accepts separate horizontal and vertical digital sync signals, you should set the DSST parameter to the exact type that is needed. A setting of zero (0) means that separate horizontal and vertical digital sync cannot be selected.

The polarity of the separate horizontal sync output is controlled by the HSPP parameter, while the polarity of the separate vertical sync output is controlled by the VSPP parameter.

❖ Entering the wrong HSPP polarity may cause all of the test images to be shifted horizontally on a properly adjusted display. Entering the wrong VSPP polarity may cause all of the test images to be shifted vertically on a properly adjusted display.

Also, many multi-scan mode displays depend on the polarity of the sync pulses to determine their exact operating mode. The wrong polarities may prevent the display from locking to the format's scan rates or may cause the size of the active video to be way off.

If your display accepts a single combined digital sync input, you should set the DSCT parameter to the exact type that is needed. A setting of zero (0) means that digital composite sync cannot be selected.

The polarity of the digital composite sync output is controlled by the CSPP parameter.

You have now defined the type of video and sync that will be generated when your new format is selected. Next, you will specify the timing of the various signal components.

Timing Parameters

HRAT, sets the horizontal scan, or line, rate of the format. This is the rate at which horizontal video lines are produced. The 801G* uses Hertz as the unit of measure while most display spec sheets may give the rate in kilo-Hertz. The rate can easily be entered as a floating point number. For example, a 15.73426374 kHz horizontal rate could be entered as 15734.26374 or as 15.73426374E3. Here are some ways to calculate the value for HRAT if it is not given on your spec sheet:

- If a spec sheets gives you the total horizontal period, in micro-seconds, instead of the horizontal rate. HRAT can be calculated as follows:

  HRAT = 1 / Horizontal Period

- If a spec sheets gives you a dot clock or pixel rate and the total number of active and blanked pixels per line. HRAT can be calculated as follows:

  HRAT = Pixel Rate / Total Pixels per Line

The HRES parameter sets the number of active pixels in a single scan line. This parameter may be called "Horizontal Resolution" or "Horizontal Active Pixels" on your spec sheet.

The HTOT parameter sets the total number of pixels (active + blanked) in one horizontal scan line. It must be an even number. If your spec sheet gives you the horizontal period in units of pixels, use this value for HTOT. Here are some ways to calculate the value for HTOT from other data on your spec sheet:

- A spec sheet may give you the number of horizontal active and blanked pixels as two separate numbers. Simply add the two numbers together to get HTOT.

- A spec sheet may give you pixel counts for the number of active pixels as well the horizontal sync front porch, pulse width and back porch. Add all four numbers together to get HTOT.

- A spec sheet may give you a dot clock or pixel rate. Divide this rate by the horizontal rate and round the result to the nearest even integer.

- A spec sheet may give you the number of horizontal active pixels and express the horizontal blanking time in micro-seconds. The value for HTOT can be calculated with this formula:

  HTOT = HRES / (1 - (Blanking x HRAT))

  Where Blanking is in units of seconds and HRAT is in units of Hz. Remember to round the result to the nearest even integer.

The HSPD parameter sets the number of pixels between the end of active video and the beginning of the horizontal sync pulse. Many spec sheets refer to this parameter as the "Horizontal Sync Front Porch." If your spec sheet gives you the horizontal front porch in units of pixels, use this value for HSPD. Here is another way to calculate the value for HSPD from other data on your spec sheet:

- A spec sheet may give you the horizontal front porch in micro-seconds. The value for HSPD can be calculated with this formula:

  HSPD = HTOT x HRAT x Front Porch

  Where Front Porch is in units of seconds and HRAT is in units of Hz. Remember to round the result to the nearest integer.

The HSPW parameter sets the width of the horizontal sync pulse in units of pixels. Many spec sheets refer to this parameter as "Horizontal Sync Pulse Width." If your spec sheet gives you the pulse width in units of pixels, use this value for HSPW. Here is another way to calculate the value for HSPW from other data on your spec sheet:

- A spec sheet may give you the horizontal sync pulse width in micro-seconds. The value for HSPW can be calculated with this formula:

  HSPW = HTOT x HRAT x Pulse Width

  Where Pulse Width is in units of seconds and HRAT is in units of Hz. Remember to round the result to the nearest integer.

We are now done with the horizontal timing and can move on to the vertical timing. The first vertical timing parameter, SCAN, determines if a given format will have a non-interlaced or 2:1 interlaced vertical scanning mode. Here are a few ways to determine the correct mode if it is not clearly stated on your spec sheet.

- Non-interlaced scanning is sometimes called “Progressive Scan” on a spec sheet.
- References to “odd” and “even” fields on a spec sheet usually indicate 2:1 interlaced operation.
- A “Field Rate” that is twice the “Frame Rate” on a spec sheet indicates 2:1 interlaced operation.

The VRES parameter sets the number of active scan lines in a complete frame of video. This parameter may be called “Vertical Resolution” or “Vertical Active Pixels” on your spec sheet.

- The spec sheet for an interlaced display may give you the number of active lines for a single field. You will need to double this number to get the correct value for VRES.

The VTOT parameter sets the total number of horizontal scan lines, active + blanked, in one complete frame. It must be an odd number when SCAN = 2. If your spec sheet gives you the frame period in units of lines, use this value for VTOT. Here are some ways to calculate the value for VTOT from other data on your spec sheet:

- A spec sheet may give you the number of horizontal active and blanked lines per frame as two separate numbers. Add the two numbers to get VTOT.

- A spec sheet for a non-interlaced display may give you a vertical rate. The value for VTOT can be calculated using the following formula:

  VTOT = HRAT / Vertical Rate

- A spec sheet for an interlaced display may give you a field or frame rate. The value for VTOT can be calculated using one of the following formulas:

  VTOT = HRAT / Frame Rate

  VTOT = 2 x (HRAT / Field Rate)

- A spec sheet for an interlaced display may give you the number of active and blanked lines in a single field. The value for VTOT can be calculated using the following formula:

  VTOT = 2 x (Active Lines + Blanked Lines)

The VSPD parameter sets the number of scan lines between the last line of active video and the beginning of the vertical sync pulse. Many spec sheets refer to this parameter as “Vertical Sync Front Porch.” If your spec sheet gives you the vertical front porch in units of lines, use this value for VSPD. Here is another way to calculate the value for HSPD from other data on your spec sheet:

- A spec sheet may give you the vertical front porch in milli-seconds. The value for VSPD can be calculated with this formula:

  VSPD = HRAT x Front Porch

  Where Front Porch is in units of seconds and HRAT is in units of Hz. Remember to round the result to the nearest integer.

The VSPW parameter sets the width of the vertical sync pulse in units of scan lines. Many spec sheets refer to this parameter as "Vertical Sync Pulse Width." If your spec sheet gives you the pulse width in units of lines, use this value for VSPW. Here is another way to calculate the value for HSPW from other data on your spec sheet:

- A spec sheet may give you the horizontal sync pulse width in micro-seconds. The value for HSPW can be calculated with this formula:

  VSPW = HRAT x Pulse Width

  Where Pulse Width is in units of seconds and HRAT is in units of Hz. Remember to round the result to the nearest integer.

❖ The vertical sync pulse width is the overall width of the vertical sync interval if you are working with serrated composite sync. Do not use the distance between the individual serration pulses for VSPW.

Some displays that use either a digital or analog composite sync signal need to have extra pulses added to the sync. Serration and equalization pulses help stabilize the display's horizontal sweep circuitry during the vertical sync interval. The previously discussed DSCT and ASCT parameters determine if serrations and equalization pulses are added to either of the composite sync outputs. The number of equalization pulses, if used, can also be programmed.

The EQUB parameter sets the length of the equalization interval before the vertical sync pulse. The value is entered in multiples of scan lines with two pulses per scan line (less one if a CCIR sync type has been specified).

The EQUA parameter sets the length of the equalization interval that follows the vertical sync pulse. The value is entered in multiples of scan lines with two pulses per scan line (less one if a CCIR sync type has been specified).

❖ Equalization pulses will only be added to the Digital composite sync signal when DSCT is set to 4 or 8.

  Equalization pulses will only be added to the Analog composite sync signal when ASCT is set to 4 or 8.

That is all of the timing parameters you need to enter. You may wish to go through a couple of calculations in the next section just to see if the format you are setting up matches your spec sheet.

Calculated Rates

Many video specification sheets include a pixel rate parameter and a frame rate parameter. You do not need to program them into the 801G*. The firmware calculates the two values based on other entered parameters. You can double check your entries by calculating the pixel and frame rates.

The pixel or dot clock rate is equal to HRAT multiplied by HTOT. The hardware limits of the generator do not allow rates above 135.000 MHz

❖ If your calculated pixel rate exceeds the maximum limit, you will need to proportionately reduce HTOT, HRES, HSPD and HSPW to give you a valid format. This will produce a correctly sized and centered test image. However, the display will not be driven at its full video bandwidth.

The frame rate is equal to HRAT divided by VTOT. This should match the frame rate given on your spec sheet.

Active Video Physical Size

The USIZ, HSIZ and VSIZ parameters do not affect the sync timing or signal outputs of the 801G*. They do, however, greatly affect the test images that are drawn. For example, the size of the box in the "Brightness" test pattern should match the size of your light meter probe. The actual size of the probe is entered using the system parameter MSIZ. The firmware combines this information along with the format's physical size information to properly scale the size of the box. The firmware also uses the format's size information to make sure the crosshatch lines in the SMPTE RP-133 pattern form square boxes and that the circle is round and not elliptical.

The USIZ parameter determines whether the unit of measure will be interpreted as inches (1) or in mm (2).

The HSIZ parameter is the horizontal width of the active video area on a properly adjusted display.

The VSIZ parameter is the vertical height of the active video area on a properly adjusted display.

❖ Changing the values of HSIZ and (or) VSIZ will not change the overall size of most test images on your display.

Display Codes

Many video controller cards for the Apple Macintosh II and VGA type cards for the IBM-PC are able to identify the type of display connected to them by checking one or more display code sense lines coming from the display. The information is then used to select one of several different operating modes to match the display. An improper display code may make the controller card or display appear to be malfunctioning. The “Making Connection” chapter has information on display codes that are used by some systems.

A format can be programmed to check these lines and report the results to the operator.

The DCBM parameter determines which of up to 4 sense lines will be tested. A value of 15 will test all 4 lines. A value of 7 will test 3.

The DCEX parameter is the display code number that you are expecting to get back from a particular display. The actual value that is read back is shown in the “Format” test image.

❖ The 801G* does not use the display code data to change its own operating modes. The DCBM and DCEX parameters are only used to provide information to the operator so that they may compare the expected and actual results.

Outputs Control

The last set of format parameters allow you to determine which outputs are active when the operator first selects the format from memory.

❖ In the current version of firmware, the status of the “Outputs” button has priority over individual gating controls. Selecting a format while the outputs are turned off will keep all of the outputs off until the button is pressed. Selecting a format while the outputs are turned on will activate all of the outputs that have been gated on in the format.

The XVSG command determines which video outputs will be active when the format is selected. The same command controls both the analog and digital video outputs.

• The value of XVSG can be temporarily changed by the operator, using the “R,” “G” and “B” video gating buttons.

The SSST parameter determines which sync type will be active when the format is selected. A setting of 0 will leave all the sync outputs off.

• You have control over whether or not the operator can override the programmed SSST selection. A non-zero entry for the ASCT parameter will allow the operator to enable analog composite sync by pressing the “ACS” sync gating button. A non-zero entry for the DSCT parameter will allow the operator to turn on digital composite sync by pressing the “DCS” sync gating button. A non-zero entry for the DSST parameter will allow the operator to turn on digital separate sync by pressing the “DHS & DVS” sync gating button.

❖ The SSST parameter does not override any individual programmed sync settings. For example, setting ASCT to zero and then selecting analog composite sync with the SSST parameter will leave all of the sync outputs turned off.

The HSPG and VSPG parameters determine if both horizontal and vertical sync are turned on when separate horizontal and vertical is selected.

❖ Gating either individual sync output off with a setting of zero will keep it turned off even when the “DHS & DVS” button is pressed by the operator.

The CSPG parameter determines if digital composite sync is turned on when digital composite sync is selected.

❖ Gating the digital composite sync output off with a setting of zero will keep it turned off even when the “DCS” button is pressed by the operator.

We are now done creating the file. Save your file as straight ASCII text. If you as using a DOS based system, use a file name extension of “.CMD”. We can move on to getting your format into the 801G*. This is discussed in the next section.

Downloading formats using a DOS-compatible computer

There are two ways of downloading a format file to the 801G*. You can use the send.bat utility or the file download function on the terminal emulation program that is discussed in the next section.

The following procedure should be used when using the send.bat utility program that is on the disk disk supplied with the 801G*.

1. Connect the 801G* to a personal computer using the instructions given in the "Making Connections" chapter of this manual.

3. Send new command files to the 801GP using the send.bat utility found on the utility disk. To download a file, the command is:

```
SEND filename
```

❖ The send.bat utility assumes that you are connecting the 801G* to COM1. If this is not the case, then you must modify the COM1 references in the file to match the COM port you are using.

How to use PCPLUSTD.EXE

The utility disk included with your 801G* contains the "Test drive" version of Pcplus by DATASTORM TECHNOLOGIES Inc. This program is for evaluation only and may not be used beyond that. Please read the PCPLUSTD.DOC file on the utility disk for further details.

PCPLUSTD.EXE is a demonstration of how the user can interactively operate and program the 801G* generator via the serial port.

You need to set some configuration data the first time that you use the program.

1) After you have installed PCPLUSTD on your PC, start it by typing PCPLUSTD on the DOS command line.

2) Press (ALT)P to bring up the serial port setup menu.

3) Select the proper COM port you will be using with your 801G*.

4) Set the port for 2400 baud, 8 bits, 2 stop, no parity.

5) Press (ALT)S to save your settings.

You are now ready to send individual commands and queries as well as download files to 801G*. The last section of this chapter documents the command language.

Controlling via the IEEE-488 Port

You do not need to do anything special on the 801G* to use it's IEEE-488 port. The default settings are for the 801G* to be a talker/listener with an address of 15.

All of the commands and queries shown at the end of this chapter can be used over the port. This section covers how the commands should be sent over the GPIB bus. Information on additional commands that can only be sent through the GPIB port is also included in this section.

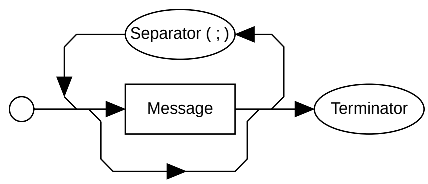
Input Buffer

Commands are sent to the 801G* in "program messages." A program message consists of one or more commands separated by a semicolon (;) followed by a program message terminator. A program message terminator is defined as the NL character (ASCII 10), or EOI sent with the last byte of the program message. The 801G* will not parse any commands received until a program message terminator is received. All commands are executed sequentially, that is, when a command is parsed it is allowed to finish execution before the next command is parsed.

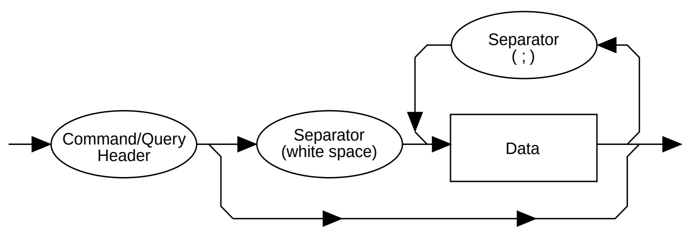
Since some commands may take longer to execute, the 801G* has an input buffer. This input buffer is 255 characters long and can be written to by the host controller while the 801G* is busy executing or parsing previous messages. If the input buffer becomes full, the 801G* will hold off the controller until there is room in the buffer. For this reason, a program message cannot be longer than 255 characters including terminator.

Message Syntax

Program messages are sent to the 801G* using four character ASCII upper/lower case headers. These headers can be either commands or queries. Commands are messages which do not require a response from the 801G*. Queries are messages which cause the 801G* to respond with the required data.



Program Message Syntax



Message Unit Syntax

Commands

Command headers instruct the 801G* to set a parameter to the value given or perform some function not requiring any additional data. Commands which have an asterisk (*) as their first character are common commands as defined by the IEEE-488.2 standard and generally operate the same in all instruments. See the listing at the end of this section for descriptions of each common command.

Command headers which do not require any additional data from the controller are self-contained and should be followed by either a message separator (;) or message terminator. Any other characters (except whitespace characters) will cause a command error.

Command headers which require numeric data should be followed by at least one separator character (whitespace) then the data. Numeric data sent with a command is in decimal format. Numeric data can be represented in one of three methods; integer, floating point, and scaled floating point. The three methods shown below would all represent the number 42.

Integer	42
Floating Point	42.00
Scaled Floating Point	4.200E+01

Queries

Queries are comprised of a header followed immediately by a question mark (?). If there are any characters between the query header and the question mark character (including whitespace), a command error will result. Queries, upon completion of execution return a response message. A response message is comprised of the requested data terminated with an NL (ASCII 10) character. For example the following text would be returned in response to the HRAT? (horizontal rate query) message:

```
3.1500E+04<NL>
```

Note that the HRAT? query returns its parameter in exponential form. Possible returned parameter forms are integer, exponential, and string. The response form of each queryable message is shown at the end of this section.

Integer form:	3965
Exponential form:	+3.965E+03
String form:	"A string"

Output Queue

When a query is executed, the resulting response message is placed in an output queue where it can be read by the controller. The 801G* has an output queue that is 255 bytes long. When a message is present in the output buffer, the MAV (message available) bit in the Status Byte register is set. This varies slightly from the 488.2 standard in that the MAV bit will only be set when at least one complete response message is present in the output queue. A complete response message consists of response message text and a message terminator (NL).

Buffer Deadlock

Buffer deadlock is occurs when the 801G* tries to put a response message in the output queue, the output queue is full, and the controller is held off while sending a new message because the input buffer is full. If deadlock occurs, the 801G* will clear its output queue, set the query error (QYE) bit in the Event Status register and proceed to parse incoming messages. If any additional queries are requested while in deadlock, those response messages will be discarded.

The 801G* will clear the buffer deadlock when it finishes parsing the current command/query. The QYE bit will remain set until read with the *ESR? query or cleared with the *CLS common command.

The Status Byte

The Status Byte used by the 801G* is the same as that defined by the IEEE-488.2 standard and does not use any other bits of the Status Byte. The Status Byte is one part of a complete status reporting system shown on the next set of facing pages. The Status Byte is read by using the serial poll feature of your controller.

Requesting Service

The GPIB provides a method for any device to interrupt the controller-in-charge and request servicing of a condition. This service request function is handled with the Status Byte. When the RQS bit of the Status Byte is true, the 801G* is requesting service from the controller. There are many conditions which may cause the 801G* to induce a service request. For more information about setting up these conditions, see the *SRE common command description.

Remote/Local Operation

The 801G* has complete remote/local operation as defined by the IEEE-488.1 standard. All four remote/local states REMS, LOCS, RWLS and LWLS are supported.

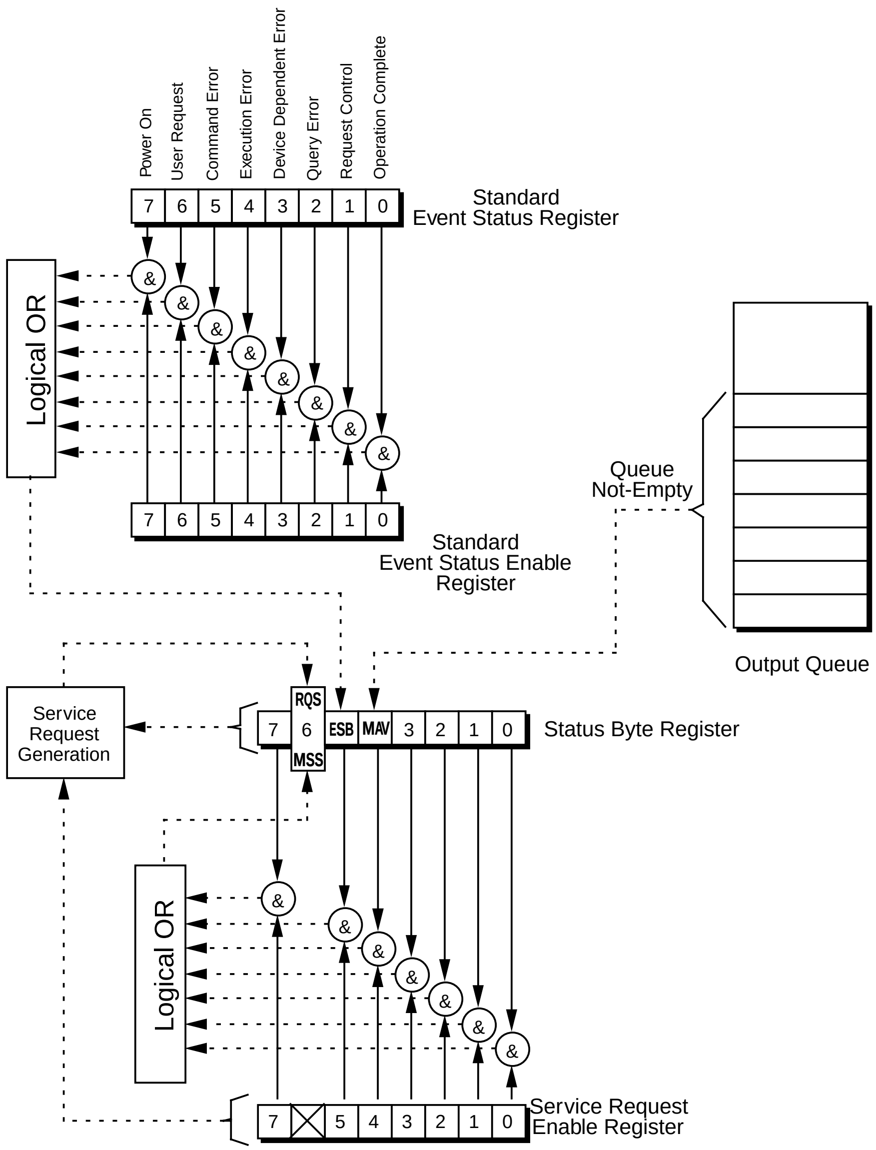
In the remote state (REMS), the 801G* is under remote control and messages are processed as received. The remote with lockout state (RWLS) will be entered if the controller issues the LLO (local lockout) message to the 801G*. The local state (LOCS) will be entered when the REN line goes false or the controller issues the GTL (go to local) message to the 801G*, or a front panel control is actuated.

In the remote with lockout state (RWLS), the 801G* is under complete remote control and front panel controls are disabled. The RWLS state is entered when the controller issues the LLO (local lockout) message to the 801G*. Front panel access is re-enabled when the controller issues the GTL (go to local) message to the 801G*.

In the local state (LOCS), the 801G* is under local control and all front panel controls are enabled. Any remote messages received will be stored for processing when the 801G* enters the remote state again. The remote state (REMS) will be entered if the REN line is true and the 801G* is addressed to listen.

In the local with lockout state (LWLS), the 801G* is under local control and all front panel controls are enabled. Any remote messages received will be stored for processing when the 801G* enters the remote state again. The remote with lockout state (RWLS) will be entered if the 801G* is addressed to listen.

IEEE-488 Status Reporting:



Status Byte Bit Definitions	
MAV	Message available. Indicates that at least one complete response is present in the output buffer.
ESB	Event status bit. Indicates that one of the enabled conditions in the Standard Event Status register is set.
MSS	Master summary status. Indicates that the 801GX has a reason for requesting service.
RQS	Request service. This bit is read only by executing a serial poll of the 801GX.

Event Status Bit Definitions	
OPC	Operation complete. Indicates that all operations have been completed.
RQC	Request control. Indicates that a device is requesting control. The 801GX will never request control, so this bit will always be 0.
QYE	Query error. Indicates that a query request was made while the 801GX was in deadlock.
DDE	Device dependent error. Indicates that the 801GX encountered an error executing a command.
EXE	Execution error. Indicates that there was an error parsing a parameter.
CME	Command error. Indicates that there was an unrecognizable command.
URQ	User request. Indicates that a front panel button has been pressed or that the front panel knob has been turned.
PON	Power on. Indicates that power has been turned off-and-on. This bit will always be 0 in the 801GX.

Bus Commands

Bus commands - commands which are sent to the 801G* with ATN true - are defined in the IEEE-488.1 standard. The details of operation of these commands are defined in the IEEE-488.1 and 488.2 standards. The following bus commands are supported by the 801G*:

DCL Device Clear - Clears the input buffer and output queue, and stops parsing any commands.

SDC Selected Device Clear - Same as Device Clear.

GTL Go To Local - Enters the local state. See the Remote/Local section.

LLO Local Lockout - Enters the lockout state. See the Remote/Local section.

SPE Serial Poll Enable - Enables transmission of the Status Byte.

SPD Serial Poll Disable - Exits the serial poll state.

Common Commands

The common commands used by the 801G* are listed on the following pages. Common commands are commands which begin with an asterisk (*). These commands are defined by the IEEE-488.2 standard and operate the same from instrument to instrument. The 801G* supports all required common commands plus one additional command (*OPT?).

*CLS Clear Status

Definition: The *CLS command clears the Event Status Register, the Status Byte and the output buffer.

Command Syntax: *CLS

Example: *CLS

Related Commands: *ESR? *STB?

*ESE Event Status Enable

Definition: The *ESE command sets the Event Status Enable register to the given mask value. The bits in the Event Status Enable register function as enable bits for each corresponding bit in the Event Status register. That is, when a bit in the Event Status register goes high, and the corresponding bit in the Event Status Enable register is a 1, it is enabled and will cause the ESB bit in the Status Byte register to go high.

The *ESE query returns the current value of the Event Status Enable register.

Command Syntax: *ESE <mask>

<mask> = 0 - 255

Example: *ESE 8

Query Syntax: *ESE?

Returns: <mask><NL>

<mask> is in integer NR1 form

Example: *ESE?

Related Commands: *CLS *ESR?

*ESR? Event Status Register

Definition: The *ESR? query returns the current value of the Event Status register. After this command is executed, the Event Status register is cleared. This is the only way of clearing any bit in the Event Status register except by the *CLS command.

Query Syntax: *ESR?

Returns: <register value><NL>

<register value> is in integer NR1 form

Example: *ESR?

Related Commands: *CLS *ESE

*OPC Operation Complete

Definition: The *OPC command will cause the 801G* to set the OPC bit in the Event Status register when all operations have been completed. Since there are no overlapping commands, the *OPC command will set the OPC bit immediately when executed.

The *OPC query will put a "1" in the output buffer when all operations are complete.

Command Syntax: *OPC

Example: *OPC

Query Syntax: *OPC?

Returns: 1<NL>

Example: *OPC?

*OPT? Options

Query Syntax: The *OPT query returns a list of options installed in the 801G*.

Query Syntax: *OPT?

Returns: <option string><NL>

Example: *OPT?

*RST Reset

Definition: The *RST command performs a device reset. This places the 801G* into a known condition. These conditions are:

- IEEE-488 address set to 15
- Status Byte cleared
- Input queue empty
- Output queue empty

Command Syntax: *RST

Example: *RST

Related Commands: *CLS

*SRE Service Request Enable

Definition: The *SRE command sets the Service Request Enable register to the mask value given. The bits in the Service Request Enable register function as enable bits for each corresponding bit in the Status Byte register to enable a condition to request service from the system controller. That is, when a bit in the Status Byte register goes true, and the corresponding bit in the Service Request Enable register is also true, the 801G* will request service through the GPIB.

The *SRE query returns the current value of the Service Request Enable register.

Command Syntax: *SRE <mask>

<mask> = 0 - 255

Example: *SRE 16

Query Syntax: *SRE?

Returns: <mask><NL>

<mask> is in integer NR1 form

Example: *SRE?

Related Commands: *STB *ESE

*STB? Status Byte

Definition: The *STB query returns the current value of the Status Byte register. The value stored in the Status Byte register is not affected by reading it.

Query Syntax: *STB?

Returns: <Status Byte><NL>

<status byte> is in integer NR1 form

Example: *STB?

Related Commands: *SRE *ESR? *CLS

*TST? Self-Test

Definition: The *TST query causes the 801G* to perform a self-test and report the results in a response message. If the self-test fails, an ASCII "1" is placed in the output buffer, otherwise an ASCII "0" is placed in the output buffer.

Query Syntax: *TST?

Returns: <result><NL>

<result> is in integer NR1 form

Example: *TST?

Command Language

All of the commands you can only though the IEEE-488 port on the 801G* generator are listed in a previous section. All of the commands you can use with either the RS-232 or IEEE-488 port on the 801G* generator are listed in this in this section. They're listed two ways. The *Command and Query Finder* section groups them by function. For example, all the commands used to change format parameter are listed under one heading. This is followed by a straight alphabetical listing of the commands. This listing also contains descriptions and expected parameters.

Command and Query Finder

This listing shows all of the commands and queries supported by the current version of firmware. Each command or query will be listed under one or more of the following headings:

- Format Parameter Settings
- Format Error Checking
- Format Editor Control
- Format Memory Management
- Custom Image Drawing Primitives
- Custom Image Editor Control
- Image Memory Management
- Sequence Editor Control
- Sequence Memory Management
- Directory Editor Control
- Directory Memory Management
- System Parameter Settings
- Direct Processor Control

Format Parameter Settings

These commands affect the type of test signal produced by the ISA generator. The commands also determine the timing of the signal. All the parameters set by these commands can be saved as a single Format in the generator's on-board Format storage locations.

ASCT	Analog Sync Composite Type
ASSG	Analog Sync Signal Gate
ASSS	Analog Sync Signal Swing
AVCM	Analog Video Calibration Method
AVCO	Analog Video Color Order
AVCS	Analog Video Color Subcarrier
AVPG	Analog Video Pedestal Gate
AVPS	Analog Video Pedestal Swing
AVSS	Analog Video Signal Swing
AVST	Analog Video Signal Type
CSPG	Composite Sync Pulse Gate
CSPP	Composite Sync Pulse Polarity
DCBM	Display Code Bit Mask
DCEX	Display Code EXpected
DSCT	Digital Sync Composite Type
DSST	Digital Sync Separate Type
DVSP	Digital Video Signal Polarity
DVST	Digital Video Signal Type
EQUA	EQUalization interval After vertical sync pulse
EQUB	EQUalization interval Before vertical sync pulse
GAMA	GAMmA correction factor
GAMC	GAMma Correction
HRAT	Horizontal RATe
HRES	Horizontal RESolution (active pixels per line)
HSIZ	Horizontal SIZe

HSPD	Horizontal Sync Pulse Delay
HSPG	Horizontal Sync Pulse Gate
HSPP	Horizontal Sync Pulse Polarity
HSPW	Horizontal Sync Pulse Width
HTOT	Horizontal TOTal pixels per line
PCPG	Pixel Clock Pulse Gate
SCAN	SCAN fields per frame
SSST	Select Sync Signal Type
USIZ	Unit of measure used for physical SIZes
VRES	Vertical RESolution (active lines per frame)
VSIZ	Vertical SIZe
VSPD	Vertical Sync Pulse Delay
VSPG	Vertical Sync Pulse Gate
VSPP	Vertical Sync Pulse Polarity
VSPW	Vertical Sync Pulse Width
VTOT	Vertical TOTal lines per frame
XVSG	Video Signal Gating

Format Editor Control

These commands and queries are used to set up the Format editing register prior to editing or creating new Formats.

FMTB	ForMaT Begin
FMTE	ForMaT End
FMTG	ForMaT informat register Good?
FMTJ	ForMaT informat register Justify
FMTN	ForMaT New (initialize all format parameters to default values)
FMTJ	ForMaT Justify
JRAT	Justify pixel RATe
NAME	format NAME (obsolete)

Format Memory Management

These commands and queries are used to write and read Formats from and to Format memory locations as well as moving and deleting Formats in memory.

ALLU	ALL Use
FMTA	ForMaT save As
FMTD	ForMaT Duplicate (obsolete)
FMTI	ForMaT Insert (obsolete)
FMTK	ForMaT Kill
FMTL	ForMaT Load from memory by name
FMTM	ForMaT Move (obsolete)
FMTP	ForMaT Path
FMTQ	ForMaT Query pointer
FMTR	ForMaT Read from memory
FMTS	ForMaT Save
FMTT	ForMaT Test - Test the format in format memory location (obsolete)
FMTU	ForMaT Use
FMTV	ForMaT Verify (obsolete)
FMTW	ForMaT Write - Save format into format memory location (obsolete)
FMTY	ForMaT Yank (obsolete)
FMTZ	ForMaT Zap (obsolete)

Custom Image Primitives

These commands are used to draw the individual primitives that make up user-defined images. All the available colors and fill patterns are listed here. An image cannot have more than 16 different colors in it. Trying to use more than 16 colors causes unexpected results. The command description section lists what parameters are needed by each primitive. Most of the custom image primitives require you to specify their color.

ADOT	draw A single pixel DOT
CENT	draw CENTering markers
CROS	draw centered CROSs
FORM	draw FORMat data block
FRGB	Foreground RGB levels
GRID	draw GRID
GRIH	draw GRIl Horizontally
GRIV	draw GRIl Vertically
HATI	draw crossHATch starting Inside
HATO	draw crossHATch starting Outside
LIMI	draw LIMIts markers
LINE	draw a LINE
NOGA	use NO GAmma correction
OVAL	draw an OVAL
PAGE	draw a block of text
RECT	draw a RECTangle
SNUM	draw sequence Step NUMber
TEXT	draw TEXT string
TOBL	set levels relative TO BLanking
TRIA	draw a TRIAngle

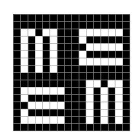
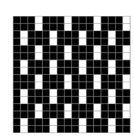
Color Options

Black	Cyan50	Gray33	Gray70	Yellow75
Red	Brown	Gray37	Gray73	Blue75
Green	Gray3	Gray40	Gray75	Magenta75
Yellow	Gray5	Gray43	Gray77	Cyan75
Blue	Gray7	Gray47	Gray80	HueI
Magenta	Gray10	Gray48	Gray83	HueQ
Cyan	Gray13	Gray50	Gray87	HueNegI
White	Gray17	Gray51	Gray90	HueNegQ
Red50	Gray20	Gray53	Gray93	Foreground
Green50	Gray23	Gray57	Gray95	Background
Yellow50	Gray25	Gray60	Gray97	Clear
Blue50	Gray27	Gray63	Red75	
Magenta50	Gray30	Gray67	Green75	

Fonts

#0 sys16	IBM-type alphanumeric font that has printable characters for ASCII codes 0-126. It uses an 8 x 16 monospaced character block.
#1 OPIX 9	Alphanumeric font that has printable characters for ASCII codes 32-126. It uses a 5 x 7 monospaced character block.
#2 focusmac	Single character used in the Focus_Oo test image; ASCII code = 79; 8 x 6 character block
#3 focus_12	Two characters used for the Focus_Cx and Focus_H test images; ASCII codes = 67 and 72; 8 x 6 character blocks
#4 memesony	SinglecharacterusedforvariousMEMEimages;ASCII code = 77; 18 x 18 character block
#5 kanjikan	Single Japanese KanjiKan character used in the KanjiKan image; ASCII = 75; 22 x 22 character block
#6 focusat 5	Asingle@charactersusedintheLinFocusimage;ASCII = 64; 8 x 16 character block
#7 focusat 6	Asingle@characterusedintheFocus@6image;ASCII = 64; 16 x 16 character block
#8 focusat 7	Asingle@characterusedintheFocus@7image;ASCII = 64; 16 x 16 character block
#9 focusat 8	Asingle@characterusedintheFocus@8image;ASCII = 64; 16 x 16 character block
#10 memeplus	A meme plus character used in focus and convergence; ASCII = 77; 43 x 49 character in a 49 x 51 block

Fill Patterns

0 GrayPat	Outline
7 GrayPat	7% gray fill*
13 GrayPat	13% gray fill
19 GrayPat	19% gray fill
25 GrayPat	25% gray fill
31 GrayPat	31% gray fill
38 GrayPat	38% gray fill
44 GrayPat	44% gray fill
50 GrayPat	50% gray fill
63 GrayPat	63% gray fill
69 GrayPat	69% gray fill
75 GrayPat	75% gray fill
81 GrayPat	81% gray fill
88 GrayPat	88% gray fill
94 GrayPat	94% gray fill
100 GrayPat	Fill is the same color as the lines
Checker1	Checkerboard alternating one (1) pixel on and one (1) off
Checker2	Checkerboard alternating two (2) pixels on and two (2) off
Checker3	Checkerboard alternating three (3) pixels on and three (3) off
Checker4	Checkerboard alternating four (4) pixels on and four (4) off
bars_V1	Vertical bars one (1) pixel wide
bars_V2	Vertical bars two (2) pixels wide
bars_V4	Vertical bars four (4) pixels wide
bars_V8	Vertical bars eight (8) pixels wide
bars_H1	Horizontal bars one (1) pixel wide
bars_H2	Horizontal bars two (2) pixels wide
bars_H4	Horizontal bars four (4) pixels wide
bars_H8	Horizontal bars eight (8) pixels wide
meme	Repeating MEME pattern 
FCC_EMI	Repeating three (3) pixels on and one pixel off pattern 

* That is, 7% of the fill pixels are active. Similar for other GrayPat fills.

Image Editor Control

These commands and queries are used to set up the custom image editing register prior to editing or creating new test images.

IMGB	custom IMaGe Begin
IMGE	custom IMaGe End
IMGN	custom IMaGe New

Image Memory Management

These commands and queries are used to select test images that are drawn on the unit under test.

ALLU	ALL Use
IMGA	IMaGe save As
IMGK	IMaGe Kill
IMGL	IMaGe Load
IMGP	IMaGe Path
IMGQ	IMaGe Query pointer
IMGR	IMaGe Read image from image memory location
IMGS	IMaGe Save
IMGU	IMaGe Use (draw image)

Sequence Editor Control

These commands and queries are used to set up the sequence editor.

SEQB	SEQuence description Begin
SEQE	SEQuence description End
SEQN	SEQuence description New

Sequence Memory Management

These commands and queries are used to select and use sequences.

ALLU	ALL Use
SEQA	SEQuence save As
SEQK	SEQuence Kill
SEQL	SEQuence Load
SEQP	SEQuence Path
SEQQ	SEQuence Query pointer
SEQS	SEQuence Save
SEQU	SEQuence Use

Sequence Parameter Settings

These commands and queries are used for creating a new sequence.

DNUM	Display sequence step NUMber
IMGL	IMaGe Load
IVER	IMaGe Version
SDLY	sequence Step DeLaY
SMOD	Sequence MODe
STEP	go to sequence STEP #

Directory Editor Control

These commands and queries are used to set up the directory editor.

DIRB	DIRectory editing Begin
DIRE	DIRectory editing End
DIRN	DIRectory New
NAMF	NAMe Find
NAMI	NAMe Insert
NAMK	NAMe Kill
NAMQ	NAMe Query
NAMY	NAMe Yank

Directory Memory Management

These commands and queries are used to select and use directories.

DIRA	DIRectory save As
DIRK	DIRectory Kill
DIRL	DIRectory Load
DIRP	DIRectory Path
DIRQ	DIRectory Query pointer
DIRS	DIRectory Save

System Parameter Settings

These commands and queries are used to set system level parameters that affect all Formats and Images.

ASSC	Analog Sync Swing Calibration factor
AVSC	Analog Video Swing Calibration factor
CACH	instruction CACHe enable
CALF	analog video CALibration Factors
FRGB	Foreground Red, Green and Blue levels
IVER	Image VERsion
KEYY	KEY toggle
KNOB	rotate KNOB
LCDS	LCD Status
LEDS	LED Status
MODE	communications MODE
MSIZ	lightMeter SIZe
OUTG	OUTputs Gate
RATC	clock RATe Calibration factor
UIDN	User IDeNtification
USIZ	Unit of measure used for physical SIZes

Miscellaneous System Parameters

The queries in this category are used to help identify the exact configuration of the generator.

*IDN	IDeNtification (listed under "I")
VERF	VERsion of Firmware
VERH	VERsion of Hardware

Direct Processor Control (Reserved)

These commands and queries are used to communicate directly with the generator's microprocessor and its internal functions. They are reserved for system debugging and diagnostics by Quantum Data personnel as well as for special software applications developed by Quantum Data.

Note – Indiscriminate use of these commands can cause the generator to malfunction. An irreversible loss of user-programmed data may also occur.

ADDR	ADDRess
BASE	BASE (radix)
BOOT	warm BOOT
CALL	CALL internal function
DCRD	Display Code ReaD
GETA	GET data from Absolute memory location
GETR	GET data from Relative memory location
INIT	INITialize to factory-default settings
PUTA	PUT Absolute
PUTR	PUT Relative
SCAL	Self CALibrate
SIZE	SIZE of bit field
*WAI	WAIt for completion suffix

Alphabetical Listing of Commands

ADDR — ADDRess

Classification: Direct processor control

Command Syntax: ADDR <address>

Limits: <address>

0 to 4,294,967,295 (BASE = 10)

-2,147,483,648 to 2,147,483,647 (BASE = -10)

0 to FFFFFFFF (BASE = 16)

-80000000 to 7FFFFFFF (BASE = -16)

Query Syntax: ADDR?

Query Response: <address>

Description: The ADDR command sets the pointer register that's used in connection with the PUTR command and GETR? query. The ADDR? query returns the current contents of the pointer register.

Note – This command normally is used only with custom applications and command files created by Quantum Data.

Other Required Cmds: ADDR and ADDR? expect and return parameters formatted according to the current radix set by the BASE command.

Example: *Use only with code supplied by Quantum Data!*

ADOT — draw A single pixel DOT

Classification: Custom image primitive

Command Syntax: ADOT <color> <x> <y>

Limits: <color> = available colors

<x> = positive integer number

<y> = positive integer number

Query Syntax: *None*

Description: Draws a single pixel dot. A dot is the smallest graphic element that can be drawn. It uses three (3) parameters, the color and the X and Y coordinates.

Other Required Cmds: The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
ADOT White 200 300 // Draw white dot at
     // X = 200 Y = 300
ALLU // Update hardware to current
     // buffer contents
```

ALLU — ALL Use

Classification: Format, Image and Sequence memory management

Command Syntax: ALLU

Query Syntax: *None*

Description: The ALLU command first checks the current contents of the format buffer for errors. If no errors are found, it reconfigures the signal generating hardware in accordance with the contents. Next, the current test image is re-rendered using the latest system and format parameter data.

Other Required Cmds: This command updates the generator after using the FMTL, IMGL and SEQL commands to load new files from memory. This command also can be used to see the results of work when using commands to edit formats or custom images.

Example:

```
FMTL vga_m3 // Load a format from memory
       // to buffer
ALLU // Update hardware to current
       // buffer contents
```

ASBG — Analog Sync on Blue Gating

Classification: Format parameter setting

Command Syntax: ASBG

Limits: <mode>
0 = OFF
1 = ON

Query Syntax: ASBG?

Query Response: <mode>

Description: The ASBG command enables and disables adding composite sync to the blue analog video outputs when analog sync is selected (see SSST command) and an analog video signal is being generated (see AVST command). The ASBG? query returns the current setting of ASBG.

See the ASSG command description for information on simultaneously controlling red, green and blue sync gating.

Other Required Cmds: The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
ASBG 1 // Enable comp sync on blue in
       // buffer
FMTU // Update hardware to current
       // buffer contents
```

ASCT — Analog Sync Composite Type

Classification: Format parameter setting

Command Syntax: ASCT <type>

Limits: <type>

0 = none
1 = American HDTV ORed
2 = American ORed
3 = American w/serr
4 = American w/serr & eq
5 = European HDTV ORed
6 = European ORed
7 = European w/serr
8 = European w/serr & eq
9 = American HDTV w/serr
10 = American HDTV w/serr & eq
11 = European HDTV w/serr
12 = European HDTV w/serr & eq
13 = Japanese HDTV ORed
14 = Japanese HDTV w/serr
15 = Japanese HDTV w/serr & eq

Query Syntax: ASCT?

Query Response: <type>

Description: The ASCT command sets the kind of composite sync added to the analog video outputs when analog sync is enabled (see SSST command) and an analog video signal is being generated (see AVST command). The ASCT? query returns the current setting of ASCT. A setting of zero (0) indicates that the ACS sync selection cannot be activated by the operator.

Other Required Cmds: The SSST mode must be set to 4, 5, 6, or 7 and the AVST type must be set to 1, 2 or 5 in order for the ASCT setting to have any affect on the generator's hardware outputs. The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
ASCT 2 // Enable Amer ORed comp sync
       // in buffer
SSST 4 // Enable ACS
AVST 2 // Select analog RGB as video
       // type
FMTU   // Update hardware to current
       // buffer contents
```

ASGG Analog Sync on Green Gating

Classification: (Obsolete) Format parameter setting

Command Syntax: ASGG

Limits: <mode>
0 = OFF
1 = ON

Query Syntax: ASGG?

Query Response: <mode>

Description: The ASGG command enables and disables adding composite sync to the green analog video outputs when analog sync is selected (see SSST command) and an analog video signal is being generated (see AVST command). The ASGG? query returns the current setting of ASGG.

See the ASSG command description for information on simultaneously controlling red, green and green sync gating.

Other Required Cmds: The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
ASGG 1 // Enable comp sync on green in
       // buffer
FMTU   // Update hardware to current
       // buffer contents
```

ASRG Analog Sync on Red Gating

Classification: (Obsolete) Format parameter setting

Command Syntax: ASRG

Limits: <mode>
0 = OFF
1 = ON

Query Syntax: ASRG?

Query Response: <mode>

Description: The ASRG command enables and disables adding composite sync to the red analog video outputs when analog sync is selected (see SSST command) and an analog video signal is being generated (see AVST command). The ASRG? query returns the current setting of ASRG.

See the ASSG command description for information on simultaneously controlling red, green and red sync gating.

Other Required Cmds: The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
ASRG 1 // Enable comp sync on red in
       // buffer
FMTU   // Update hardware to current
       // buffer contents
```

ASSC Analog Sync Swing Calibration factor

Classification: System parameter setting

Command Syntax: ASSC <red factor>, <green factor>, <blue factor>
or
ASSC <common factor>

Limits: <factor> (floating point accepted)
min = 0.000
max = 1.000

Query Syntax: ASSC?

Query Response: <red factor>, <green factor>, <blue factor>

Description: The ASSC command sets the analog video calibration (or scaling) factor that's used to adjust the level set by ASSS. Issuing the command with a single factor sets all three analog video channels to the same value. Issuing the command with three factors sets each of the analog video channels to each of the given values. The actual peak-to-peak swing of the analog composite sync signals at the output connectors equals the product of ASSS multiplied by ASSC. The ASSC? query returns the current settings of ASSC. The default factory setting is 1.000 for this parameter.

Note – The ASSC parameter is a system level parameter that affects the analog video swing of all formats that are recalled. The ASSC value is retained when the generator is powered down and back up again. Query the current setting of ASSC if you are experiencing problems with low or missing analog composite sync levels. Re-initializing the generator's memory restores the setting to factory default values of 1.000.

Other Required Cmds: *None*

Example:

```
ASSC .995 .998 1.00 // Reduce red and
     // green sync levels
FMTU // Update hardware to current
     // buffer contents
```

ASSG — Analog Sync Signal Gate

Classification: Format parameter setting

Command Syntax: ASSG <red mode>, <green mode>, <blue mode>
or
ASSG <common mode>

Limits: < mode>
0 = OFF
1 = ON
(0, 0, 0 or 0, 1, 0 only choices on 801GC-ISA)

Query Syntax: ASSG?

Query Response: <red mode>, <green mode>, <blue mode>

Description: The ASSG command enables and disables adding composite sync to all three analog video outputs when analog sync is selected (see SSST command) and an analog video signal is being generated (see AVST command). This command can take the place of sending all three of the individual ASRG, ASGG and ASBG commands. The ASSG? query returns the current settings of ASSG.

Other Required Cmds: The SSST type must be 4, 5, 6 or 7 to output analog sync. The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
ASSG 0,1,0 // Enable comp sync on green
           // in buffer
FMTU       // Update hardware to current
           // buffer contents
```

ASSS — Analog Sync Signal Swing

Classification: Format parameter setting

Command Syntax: ASSS <level>

Limits: <level> (floating point accepted)
min = 0.000 volts
max = 0.307 volts

Query Syntax: ASSS?

Query Response: <level>

Description: The ASSS command sets the maximum peak-to-peak swing for any composite sync that's added to any of the three analog video channels. The actual peak-to-peak swing of the analog sync signals at the output connectors equals the product of ASSS multiplied by ASSC. The ASSS? query returns the current setting of ASSS.

Other Required Cmds: One or more ASSG modes must be set to ON and the SSST type must be 4, 5, 6 or 7 to output analog sync. The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
ASSS 0.286 // Set sync swing to 286 mV
           // in buffer
FMTU       // Update hardware to current
           // buffer contents
```

AVCM — Analog Video Calibration Method

Classification: Format parameter setting

Command Syntax: AVCM <type>

Limits: <type>
0 = Interpolate
1 = Measure Interpolate
2 = Measure Set Absolute
3 = Test Levels

Query Syntax: AVCM?

Query Response: <type>

Description: The AVCM command determines how the generator tests and calibrates its analog video outputs. The AVCM? query returns the current setting of AVCM.

Other Required Cmds:

Example:

```
AVCM 1 // Select Measure Interpolate
       // type of self cal
```

AVCO — Analog Video COnfiguration

Classification: Format parameter setting

Command Syntax: AVCO <type>

Limits: <type>

0 = RGB	R>R	G>G	B>B	(Normal)
1 = RBG	R>R	B>G	G>B	
2 = GRB	G>R	R>G	B>B	
3 = GBR	G>R	B>G	R>B	
4 = BRG	B>R	R>G	G>B	
5 = BGR	B>R	G>G	R>B	

Query Syntax: AVCO?

Query Response: <type>

Description: The AVCO command sets the mapping of the analog video colors to the video output connections. The AVCO? query returns the current setting of AVCO.

Other Required Cmds: The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
AVCO 5 // Output blue on red chan &
       // red on blue chan
FMTU // Update hardware to current
     // buffer contents
```

AVCS Analog Video Color subcarrier Selection

Classification: Format parameter setting

Command Syntax: AVCS <type>

Limits:

<type>	Subcarrier frequency:
0 = No subcarrier	
1 = NTSC-M	3.579545 MHz (American)
2 = NTSC-443	4.43361875 MHz (PAL frequency with NTSC timing used by some conversion systems)
3 = PAL-BDGHI	4.43361875 MHz (European)
4 = PAL-N	3.58205625 MHz (Argentinian)

Query Syntax: AVCS?

Query Response: <type>

Description: The AVCS command sets the color subcarrier type used for the television outputs on generator models that have television outputs available. The AVCS? query returns the current setting of AVCS.

Note – At the time of this manual's writing, only the model 801GX stand alone generator has video with subcarrier capabilities.

Other Required Cmds: The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
AVCS 1 // Select standard American NTSC
       // encoding
FMTU // Update hardware to current
       // buffer contents
```

AVPG — Analog Video Pedestal Gate

Classification: Format parameter setting

Command Syntax: AVPG <mode>

Limits: <mode>
0 = OFF
1 = ON

Query Syntax: AVPG?

Query Response: <mode>

Description: The AVPG command enables and disables the analog video set-up pedestal. The AVPG? query returns the current setting of AVPG.

Other Required Cmds: Analog video must be enabled with the AVST command in order to output an analog video signal. The pedestal level is set with the AVPS command. The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
AVPG 1 // Enable use of a black level
       // pedestal
AVPS 7.5 // Set pedestal level to 7.5
       // IRE
FMTU // Update hardware to current
       // buffer contents
```

AVPS — Analog Video Pedestal Swing

Classification: Format parameter setting

Command Syntax: AVPS <level>

Limits: <level>
min = 0.0 I.R.E.
max = 100.0 I.R.E.

Query Syntax: AVPS?

Query Response: <level>

Description: The AVPS command sets a black pedestal level between the blanking level (0.0 I.R.E.) and the peak video level (100.0 I.R.E.). The AVPS? query returns the current setting of AVPS.

Other Required Cmds: AVPG must be set to ON to enable the use of the pedestal. The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
AVPG 1 // Enable use of a black level
       // pedestal
AVPS 7.5 // Set pedestal level to 7.5
       // IRE
FMTU // Update hardware to current
       // buffer contents
```

AVSC — Analog Video Swing Calibration factor

Classification: System parameter setting

Command Syntax: AVSC <red factor>, <green factor>, <blue factor>
or
AVSC <common factor>

Limits: <factor> (floating point accepted)
min = 0.000
max = 1.000

Query Syntax: AVSC?

Query Response: <red factor>, <green factor>, <blue factor>

Description: The AVSC command sets the analog video calibration (or scaling) factor that's used to adjust the level set by AVSS. Issuing the command with a single factor sets all three analog video channels to the same value. Issuing the command with three factors sets each of the analog video channels to each of the given values. The actual peak-to-peak swing of the analog video signals at the output connectors equals the product of AVSS multiplied by AVSC. The AVSC? query returns the current setting of AVSC for each channel. The default factory settings are 1.000 for AVSC.

Note – The AVSC parameter is a system level parameter that affects the analog video swing of all Formats that are recalled. The AVSC value is retained when the generator is powered down and back up again. Query the current setting of AVSC if you are experiencing low or missing analog video levels.

Other Required Cmds: The FMTU command instructs the generator to use the new setting on the current format.

Example:

```
AVSC 1.000 .995 .998 // Reduce green
      // and blue levels
FMTU // Apply new factors to current
      // format
```

AVSS — Analog Video Signal Swing

Classification: Format parameter setting

Command Syntax: AVSS <level>

Limits: <level> (floating point accepted)
min = 0.000 volts
max = 1.000 volts

Query Syntax: AVSS?

Query Response: <level>

Description: The AVSS command sets the maximum peak-to-peak swing for all three analog video channels. The actual peak-to-peak swing of the analog video signals at the output connectors equals the product of AVSS multiplied by AVSC. The AVSS? query returns the current setting of AVSS.

Other Required Cmds: Analog video must be enabled with the AVST command in order to output an analog video signal. The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
AVST 2 // Select RGB component color
       // video in buffer
AVSS 0.714 // Set to 714 mV in buffer
FMTU // Update hardware to current
       // buffer contents
```

AVST Analog Video Signal Type

Classification: Format parameter setting

Command Syntax: AVST <type>

Limits: <type>
- 0=none
- 1=Analog Y (grayscale)
- 2=Analog RGB (color)
- 3=Analog TV Y (grayscale)
- 4=Analog TV EYC (color subcarrier)
- 5=Analog YPrPb (color difference)

Must be zero (0) when any digital video type is selected (DVST ≠ 0).

Query Syntax: AVST?

Query Response: <type>

Description: The AVST command establishes the type of signal that appears on the analog video outputs of the generator. The AVST? query returns the current setting of AVST.

Note – Certain AVST types may not be supported by all generators in the 801G series.

Other Required Cmds: DVST must be set to zero when analog video is used. The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
DVST 0 // Disable digital video
AVST 2 // Select RGB component color
       // video
ALLU   // Update hardware and redraw test
       // image
```

BASE — number BASE

Classification: Direct processor control

Command Syntax: BASE <radix>

Limits: <radix> = -36 to -2 or 2 to 36

Query Syntax: BASE?

Query Response: <radix>

Description: The BASE command establishes the radix of address and data parameters passed to or returned from the ADDR, GETA, GETR, PUTA, PUTR, and CALL instructions. If a negative radix is specified, then parameters passed to (or returned from) these functions are assumed to be signed. For example, if BASE= -16, then the value -1 communicates the value FFFFFFFF hex. The BASE? query returns the current setting of BASE. The radix <radix> always is passed and returned in base 10, regardless of the setting of BASE. BASE is preset to -10 each time the generator is powered on. Base -10 is the preferred radix. The BASE? query returns the current setting of BASE.

Note – This command normally will be used only with custom applications and command files created by Quantum Data.

Other Required Cmds: *None*

Example:

```
Use only with code supplied by Quantum
Data!
```

BOOT — warm BOOT

Classification: Direct processor control

Command Syntax: BOOT

Query Syntax: *None*

Description: The BOOT command causes the generator to go through its standard power-up procedure. Self-calibration is not performed. The procedure checks all RAM storage locations for corrupt data.

Other Required Cmds: *None*

Example:

```
BOOT
```

CACH instruction CACHe enable

Classification:	System parameter setting
Command Syntax:	CACH <mode>
Limits:	<mode> 0 = OFF 1 = ON
Query Syntax:	CACH?
Query Response:	<mode>
Description:	The CACH command enables and disables the use of the instruction cache. The CACH? query returns the current setting of CACH.
Other Required Cmds:	
Example:	`CACH 0 // Disable the use of the` `       // instruction cache`

CALF analog video CALibration Factors

Classification: System parameter setting

Cmd Syntax(801GC): CALF <video 1000> [<video 700> [<sync 400> [<sync 40> [2048 [2048]]]]]

(801GX): CALF <video 1000> [<video 700> [<sync 400> [<sync 40> [<NTSC 714> [<PAL 700]]]]]

(801GC-ISA): CALF <video 1000> [<video 700> [<sync 400> [<sync 40> [<pclk 700> [2048]]]]]

(801GF, 801GF-ISA): CALF <video 1000> [<video 700> [<sync 400> [<sync 40> [<psync 400> [psync 40>]]]]]

Limits:

<video 1000>
min swing = 4095
max swing = 0

<video 700>
min swing = 4095
max swing = 0

<sync 400>
min swing = 0
max swing = 4095

<sync 40>
min swing = 0
max swing = 4095

<NTSC 714>
min swing = 4095
max swing = 0

<PAL 700>
min swing = 4095
max swing = 0

<pclk 700>
min swing = 4095
max swing = 0

<psync 400>
min swing = 0
max swing = 4095

<psync 40>
min swing = 0
max swing = 4095

Query Syntax: CALF?

Query Response: <video 1000> <video 700> <sync 400> <sync 40> <NTSC 714> <PAL 700>

Description: The CALF sets the analog output calibration factors to values other than those set by the generator's own self-calibration function. The <video 1000> factor adjusts the video output level when AVSS is at its maximum of 1.000 volts. The video <700> factor adjusts the video output level when AVSS is at 0.700

volts. The <sync 400> factor adjusts the analog sync output level when ASSS is at 0.400 volts. The <sync 40> factor adjusts the analog sync output level when ASSS is at 0.040 volts. The <NTSC 714> factor adjusts the NTSC television output level. The <PAL 700> factor adjusts the PAL television output level. The <pclk 700> factor adjusts the pixel clock output (when it is enabled). Information on the <psync> factors was not available when this manual was written.

Other Required Cmds: The AVSC command matches the levels for the three analog video channels. The ALLU command updates the signal generating hardware to the new settings and redraws the test image.

Example:

```
CALF 2040 2045 2055 2050 2046 2048
     // Set new factors
ALLU // Use new factors
```

CALL — CALL internal function

Classification: Direct processor control

Command Syntax: CALL <address> <passed> [<p(1)> [<p(2)> [<p(3)> ...[<p(18)>]...]]]

Limits: <address>
- 0 to 4,294,967,295 (BASE = 10)
- -2,147,483,648 to 2,147,483,647 (BASE = -10)
- 0 to FFFFFFFF (BASE = 16)
- -80000000 to 7FFFFFFF (BASE = -16)

<passed>
- 0 to 18 (BASE = -10 or 10)
- 0 to 12 (BASE = -16 or 16)

<p(n)>
- 0 to 4,294,967,295 (BASE = 10)
- -2,147,483,648 to 2,147,483,647 (BASE = -10)
- 0 to FFFFFFFF (BASE = 16)
- -80000000 to 7FFFFFFF (BASE = -16)

Query Syntax: CALL? <address> <passed> <returned> [<p(1)> [<p(2)> [<p(3)> ...[<p(17)>]...]]]

Limits: <address>
- 0 to 4,294,967,295 (BASE = 10)
- -2,147,483,648 to 2,147,483,647 (BASE = -10)
- 0 to FFFFFFFF (BASE = 16)
- -80000000 to 7FFFFFFF (BASE = -16)

<passed>
- 0 to 18 (BASE = -10 or 10)
- 0 to 12 (BASE = -16 or 16)

<returned>
- 4,294,967,295 and 0 to 20 (BASE = 10)
- -1 to 20 (BASE = -10)
- FFFFFFFF and 0 to 14 (BASE = 16)
- -1 to 14 (BASE = -16 query only)

<p(n)>
- 0 to 4,294,967,295 (BASE = 10)
- -2,147,483,648 to 2,147,483,647 (BASE = -10)
- 0 to FFFFFFFF (BASE = 16)
- -80000000 to 7FFFFFFF (BASE = -16)

Query Response: <ret(1)> [<ret(2)> [<ret(3)> ... [<ret(20)>]...]]

Description: The CALL command calls internal C functions. Address <address> is the entry point of the C function to be called. Parameter <passed> indicates the number of parameters to be passed. If <passed> is not zero, then parameters being passed <p(1)> through <p(n)> immediately follow the <passed> parameter on the command line. The CALL? query is similar to the CALL command except that returned parameters are expected. Here, a third parameter <returned> is added to indicate the number of parameters returned by the function. If BASE= -10 or -16 and a <returned> value of -1 is given (4,294,967,295 if BASE=10 or FFFFFFFF if BASE=16), then a single value is read from register A8 of the TMS34010 (rather than being popped off the C stack). Most C functions that return a single parameter return their single parameter in this way. Returned parameters are space-delimited and formatted according to the current radix (see BASE command). All parameters passed to the CALL and CALL? must be formatted according to the current radix. This includes the parameters <passed> and <returned>.

Note – This command normally is used only with custom applications and command files created by Quantum Data. Indiscriminate use of this command can cause the generator to stop operating or loss of stored data in nonvolatile RAM.

Other Required Cmds: CALL and CALL? expect and return parameters formatted according to the current radix set by the BASE command.

Example:

```
Use only with code supplied by Quantum
Data!
```

CENT *** draw video CENTering markers

Classification: Custom image primitive

Command Syntax: CENT <color>

Limits: <color> = available colors

Query Syntax: *None*

Description: Draws a small cross in the center of active video. If the format has an even number of active pixels, the vertical line is 2 pixels thick. The horizontal line is 2 pixels thick if the format has an even number of active lines. The primitive uses a single parameter, the color of the cross.

Other Required Cmds: The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
CENT red // Draw a small red cross in
         // center of active video
ALLU     // Update hardware to current
         // buffer contents
```

CROS — *** draw a centered CROSs

Classification: Custom image primitive

Command Syntax: CROS <color>

Limits: <color> = available colors

Query Syntax: *None*

Description: Draws a large centered cross that fills the active video area. The vertical line is 2 pixels thick if the format has an even number of active pixels. The horizontal line is 2 pixels thick if the format has an even number of active lines. The primitive uses a single parameter ... the color of the cross.

Other Required Cmds: The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
CROS magenta // Draw a large cross in
             // the center of the active video

ALLU // Update hardware to current
     // buffer contents
```

CSPG — Composite Sync Pulse Gate

Classification: Format parameter setting

Command Syntax: CSPG <mode>

Limits: <mode>
- 0 = OFF
- 1 = ON

Query Syntax: CSPG?

Query Response: <mode>

Description: The CSPG command enables and disables all of the digital composite sync outputs when digital composite sync is selected via the SSST command (SSST = 3). The CSPG? query returns the current setting of CSPG.

Other Required Cmds: In order to use digital composite sync, it must be selected with the SSST command. The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
CSPG 1 // Enable dig comp sync in
       // buffer
SSST 2 // Choose digital comp sync type
       // in buffer
FMTU // Update hardware to current
     // buffer contents
```

CSPP Composite Sync Pulse Polarity

Classification: Format parameter setting

Command Syntax: CSPP <polarity>

Limits: <polarity>
- 0 = active-low (negative going pulse)
- 1 = active-high (positive going pulse)

Query Syntax: CSPP?

Query Response: <polarity>

Description: The CSPP command establishes the logic sense of the digital composite sync output. The CSPP? query returns the current setting of CSPP.

Other Required Cmds: In order to use digital composite sync, it must be gated on with the CSPG command and selected with the SSST command. The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
CSPG 1 // Enable dig comp sync in
       // buffer
CSPP 1 // Select active hi in buffer
SSST 2 // Choose digital comp sync type
       // in buffer
FMTU // Update hardware to current
     // buffer contents
```

DCBM Display Code Bit Mask

Classification: Format parameter setting

Command Syntax: DCBM <mask>

Limits: <mask>

0	= 0 0 0 0	8	= 1 0 0 0
1	= 0 0 0 1	9	= 1 0 0 1
2	= 0 0 1 0	10	= 1 0 1 0
3	= 0 0 1 1	11	= 1 0 1 1
4	= 0 1 0 0	12	= 1 1 0 0
5	= 0 1 0 1	13	= 1 1 0 1
6	= 0 1 1 0	14	= 1 1 1 0
7	= 0 1 1 1	15	= 1 1 1 1

Query Syntax: DCBM?

Query Response: <mask>

Description: The DCBM command sets the 4-bit binary bit mask used by the DCRD? query. The mask is entered as the decimal equivalent of a 4-bit binary number. The binary number represents the masking of the individual sense lines from M3 (MSB) to M0 (LSB). The DCBM? query returns the current setting of DCBM.

Other Required Cmds: The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
DCBM 7 // Set mask to read sense lines
       // 0, 1 & 2 only
FMTU   // Update hardware to current
       // buffer contents
```

DCEX Display Code EXpected

Classification: Format parameter setting

Command Syntax: DCEX <code#>

Limits: <code#>

0 = 0 0 0 0	8 = 1 0 0 0
1 = 0 0 0 1	9 = 1 0 0 1
2 = 0 0 1 0	10 = 1 0 10
3 = 0 0 1 1	11 = 1 0 1 1
4 = 0 1 0 0	12 = 1 1 0 0
5 = 0 1 0 1	13 = 1 1 0 1
6 = 0 1 1 0	14 = 1 1 1 0
7 = 0 1 1 1	15 = 1 1 1 1

Query Syntax: DCEX?

Query Response: <code#>

Description: The DCEX command sets up the display code that's expected from a display connected to the generator. The code is determined by one or more *sense* lines being connected to ground by the display. Many video controller cards for the Apple Macintosh II and VGA type cards for the IBM-PC sample the status of the display code sense lines. The information then sets up one of several different operating modes to match a particular display. An improper display code may make the controller card or display appear to malfunction.

The DCEX? query first performs a logical AND operation with the display code bit mask and the actual display code that's sensed. The decimal equivalent of the result then is returned. The mask is set with the DCEX command.

The expected setting and the actual result are both shown in the *Format* test image. They have no effect how a given format generates a set of test signals.

Other Required Cmds: The FMTU command instructs the generator to use the new setting. The ALLU command uses the new setting and redraws the test image.

Example:

```
DCBM 7 // Set mask to read sense lines
       // 0, 1 & 2 only
DCEX 5 // Only lines 0 and 1 should be
       // grounded
FMTU // Update hardware to current
     // buffer contents
```

DCRD — Display Code ReaD

Classification:	Direct processor control
Command Syntax:	*None*
Query Syntax:	DCRD?
Query Response:	<code#>

0 = 0 0 0 0	8 = 1 0 0 0
1 = 0 0 0 1	9 = 1 0 0 1
2 = 0 0 1 0	10 = 1 0 10
3 = 0 0 1 1	11 = 1 0 1 1
4 = 0 1 0 0	12 = 1 1 0 0
5 = 0 1 0 1	13 = 1 1 0 1
6 = 0 1 1 0	14 = 1 1 1 0
7 = 0 1 1 1	15 = 1 1 1 1

Description: The DCRD? query returns the display code detected on the monitor sense lines as filtered through the display code bit mask. Converting the returned decimal number to a 4-bit binary number shows the status of the individual sense lines from M3 (MSB) to M0 (LSB).

Other Required Cmds: DCBM sets the mask used for reading the display code.

Example:

```
DCRD? // Return current sense lines
      // reading
```

DIRA — DIRectory save As

Classification:	Directory memory management
Command Syntax:	DIRA <name>
Limits:	<name> = a valid MS-DOS filename (8 characters minus any extension)
Query Syntax:	*None*
Description:	The DIRA command saves the current contents of the directory edit buffer using the given name.
Other Required Cmds:	*None*

Example:

```
DIRA MY_DIR // Save with the name
      // "MY_DIR"
```

DIRB DIRectory editing Begin

Classification: Directory editor control

Command Syntax: DIRB

Query Syntax: *None*

Description: The DIRB command marks the beginning of a directory editing session. This command does nothing in the current firmware version, but is used for compatibility with future versions of firmware.

Other Required Cmds: Either a DIRL command to load an existing directory or a DIRN command to create a new directory. DIRE when ending the editing session.

Example:

```
DIRN // Initialize directory edit buffer
DIRB // Start directory editing session
     // One or more directory editing
     // commands ...
DIRE // End directory editing session
```

DIRE DIRectory editing End

Classification: Directory editor control

Command Syntax: DIRE

Query Syntax: *None*

Description: The DIRE command marks the end of a directory editing session. This command does nothing in the current firmware version, but is used for compatibility with future versions of firmware.

Other Required Cmds: DIRB when starting the editing session. Use DIRA or DIRS to save changes.

Example:

```
DIRB // Start directory editing session
     // One or more directory editing
     // commands
     // ...
DIRA MYDIR_02 // Save edited directory
     // as MYDIR_02
DIRE // End directory editing session
```

DIRK DIRectory Kill

Classification: Directory memory management

Command Syntax: DIRK <name>

Limits: <name> = a valid MS-DOS filename (8 characters minus any extension)

Query Syntax: DIRK? <name>

Limits: <name> = a valid MS-DOS filename (8 characters minus any extension)

Query Response: 0 or 1

Description: The DIRK command deletes a directory by name. The query returns a one if the named directory can be deleted. If directory is read-only or nonexistent, the query returns a zero.

Other Required Cmds: *None*

Example:

```
DIRK MY_DIR // Delete dir called
     // "MY_DIR"
```

DIRL DIRectory Load

Classification: Directory memory management

Command Syntax: DIRL <name>

Limits: <name> = a valid MS-DOS filename (8 characters minus any extension)

Query Syntax: DIRL? <name>

Limits: <name> = a valid MS-DOS filename (8 characters minus any extension)

Query Response: 0 or 1

Description: The DIRL command copies the directory having a name equal to <name> from directory memory into the directory edit buffer. The query returns a one if the named directory can be loaded, otherwise a zero is returned.

NOTE:Use the FMTP, IMGP and SEQP commands to select which directory is used for the format, image and sequence selection lists.

Other Required Cmds: *None*

Example:

```
DIRL MY_DIR // Load "MY_DIR" dir in
     // edit buffer
```

DIRN DIRectory New

Classification: Directory editor control

Command Syntax: DIRN [<name>]

Limits: <name> = optional valid MS-DOS filename (8 characters minus any extension)

Query Syntax: DIRN?

Query Response: <name>

Description: The DIRN command initializes the directory edit buffer. The name <name> is assigned as the directory's name. The query will return the name that has been assigned as the directory's name.

Other Required Cmds: *None*

Examples:

```
DIRN // Init edit buffer without
     // assigning a new name
```

or

```
DIRN MY_DIR // Init edit buffer with
     // name of "MY_DIR"
```

DIRP DIRectory Path

Classification: Directory memory management

Command Syntax: DIRP <name>

Limits: <name> = a valid MS-DOS filename (8 characters minus any extension)

Query Syntax: DIRP?

Query Response: <name>

Description: The DIRP command sets the current directory path name. The query will return the current directory path name.

Other Required Cmds: *None*

Example:

```
DIRP DIRPTH01 // Set directory path to
     // DIRPTH01
```

DIRQ DIRectory Query pointer

Classification: Directory memory management

Command Syntax: *None*

Query Syntax: DIRQ? <index> <number>

Limits: <index> = positive integer number

<number> = positive integer number

Query Response: List of specified directory names

Description: The query returns <number> directory names from the list of all the directory names stored in directory memory beginning at <index>. The directories are kept in alphanumeric order.

Other Required Cmds: *None*

Examples:

```
DIRQ? 1 5 // List the first five
      // directories in memory
```

or

```
DIRQ? 1 9999 // List all directories in
      // memory
```

DIRS DIRectory Save

Classification: Directory memory management

Command Syntax: DIRS

Description: The DIRS command saves the current directory edit buffer contents into directory memory using the current name of the directory in the edit buffer.

Other Required Cmds: *None*

DNUM Display sequence step NUMbers

Classification: Sequence parameter setting

Command Syntax: DNUM <mode>

Limits: <mode>
0 = OFF
1 = ON

Query Syntax: DNUM?

Query Response: <mode>

Description: The DNUM command enables and disables the addition of the sequence step number to the displayed test image when running a sequence.

Other Required Cmds: *None*

Example:

```
DNUM 1 // Enable the displaying of the
      // sequence step #
```

DSCT Digital Sync Composite Type

Classification: Format parameter setting

Command Syntax: DSCT <type>

Limits: <type>

- 0 = none
- 1 = American HDTV ORed
- 2 = American ORed
- 3 = American w/serr
- 4 = American w/serr & eq
- 5 = European HDTV ORed
- 6 = European ORed
- 7 = European w/serr
- 8 = European w/serr & eq
- 9 = American HDTV w/serr
- 10 = American HDTV w/serr & eq
- 11 = European HDTV w/serr
- 12 = European HDTV w/serr & eq
- 13 = Japanese HDTV ORed
- 14 = Japanese HDTV w/serr
- 15 = Japanese HDTV w/serr & eq

Query Syntax: DSCT?

Query Response: <type>

Description: The DSCT command establishes the type of composite sync that appears at the digital composite sync outputs when digital composite sync is selected via the SSST command. The DSCT? query returns the current setting of DSCT. A setting of zero (0) indicates that digital composite sync cannot be activated by the operator.

Other Required Cmds: The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
DSCT 2 // Select simple Amer ORed in
       // buffer
FMTU // Update hardware to current
       // buffer contents
```

DSST — Digital Sync Separate Type

Classification: Format parameter setting

Command Syntax: DSST <type>

Limits: <type>
- 0 = none
- 1 = American separate
- 2 = American HDTV separate
- 3 = European HDTV separate
- 4 = Japanese HDTV separate
- 5 = European separate

Query Syntax: DSST?

Query Response: <type>

Description: The DSST command establishes the type of digital separate sync that appears at the digital HS & VS outputs of the generator when digital composite sync is selected via the SSST command and the outputs are gated on via the HSPG and VSPG commands. The only difference between EIA and CCIR digital separate syncs is that, in the case of CCIR, the width of the vertical sync pulse is 0.5 line shorter than the width specified via the VSPW command. In the EIA case, the width of the vertical sync pulse is as programmed. The DSST? query returns the current setting of DSST. A setting of zero (0) indicates that separate digital H&V sync cannot be activated by the operator.

Other Required Cmds: The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
DSST 1
FMTU // Update hardware to current
     // buffer contents
```

DVSP Digital Video Signal Polarity

Classification: Format parameter setting

Command Syntax: DVSP <polarity>

Limits: <polarity>
0 = active-low (negative going video)
1 = active-high (positive going video)

Query Syntax: DVSP?

Query Response: <polarity>

Description: The DVSP command establishes the logic sense of the digital video outputs. The DVSP? query returns the current setting of DVSP.

Note – Please note that digital video is not supported by all generators in the 801G series. Also, some models that support digital video may not support active low for the polarity.

Other Required Cmds: In order to use the digital video outputs, digital video must be enabled with the DVST command. The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
AVST 0 // Deselect analog video in
       // buffer
DVST 5 // Select 3 bit color in buffer
DVSP 1 // Select active hi video in
       // buffer
FMTU // Update hardware to current
     // buffer contents
```

DVST Digital Video Signal Type

Classification: Format parameter setting

Command Syntax: DVST <type>

Limits: <type>

0 = not used
1 = digital V
2 = digital VI (MDA)
5 = RGB
6 = RGBI (CGA)
7 = RrGgBb (EGA)

DVST must be zero (0) when any analog video type is selected (AVST ≠ 0).

Query Syntax: DVST?

Query Response: <type>

Description: The DVST command establishes the kind of video signal that exits the digital video signal outputs of the generator. The DVST? query returns the current setting of DVST.

Note – Please note that digital video is not supported by all generators in the 801G series.

Other Required Cmds: AVST must be set to zero. The ALLU command updates the hardware to the new setting and redraws the test image.

Example:

```
AVST 0 // Deselect analog video in
       // buffer
DVST 5 // Select 3 bit color in buffer
FMTU // Update hardware to current
       // buffer contents
```

EQUA EQUalization interval After vertical sync pulse

Classification: Format parameter setting

Command Syntax: EQUA <lines>

Limits: <lines>
min = 0
max = number of lines after vertical sync before video

Query Syntax: EQUA?

Query Response: <lines>

Description: The EQUA command establishes the width of the equalization interval after the vertical sync pulse in each field whenever a serrated & equalized sync type is selected via either ASCT or DSCT commands and selected via the SSST command. If the type specified for the selected sync signal is one of the CCIR types, then the actual equalization interval output by the generator will be 0.5 lines shorter than the whole number specified. The EQUA? query returns the current setting of EQUA.

Other Required Cmds: The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
EQUA 3 // Set post-equalization to 3
       // lines in buffer
FMTU // Update hardware to current
       // buffer contents
```

EQUB — EQUalization interval Before vertical sync pulse

Classification: Format parameter setting

Command Syntax: EQUB <lines>

Limits: <lines>
min = 0
max = number of lines after video and before vertical sync

Query Syntax: EQUB?

Query Response: <lines>

Description: The EQUB command establishes the width of the equalization interval before the vertical sync pulse in each field whenever a serrated & equalized sync type is selected via either ASCT or DSCT commands and selected via the SSST command. If the type specified for the selected sync signal is one of the CCIR types, then the actual equalization interval output by the generator will be 0.5 lines shorter than the whole number specified. The EQUB? query returns the current setting of EQUB.

Other Required Cmds: The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
EQUB 3 // Set pre-equalization to 3
       // lines in buffer
FMTU // Update hardware to current
       // buffer contents
```

FMTA — ForMaT save As

Classification: Format memory management

Command Syntax: FMTA <name>

Limits: <name> = a valid MS-DOS filename
(8 characters minus any extension)

Query Syntax: *None*

Description: The FMTA command saves the current contents of the format edit buffer using the given name.

Other Required Cmds: *None*

Example:

```
FMTA MY_FMT // Save with the name
       // "MY_FMT"
```

FMTB ForMaT editing Begin

Classification: Format editor control

Command Syntax: FMTB

Query Syntax: *None*

Description: The FMTB command marks the beginning of a format editing session.

Other Required Cmds: Either an FMTL command to load an existing image or an FMTN command to create a new FORMAT. FMTE when ending the editing session.

Example:
```
FMTN // Initialize format edit buffer
IMGB // Start format editing session
     // One or more format editing
     // commands
     // ...
IMGE // End format editing session
```

FMTE ForMaT editing End

Classification: Format editor control

Command Syntax: FMTE

Query Syntax: *None*

Description: The FMTE command marks the end of a format editing session.

Other Required Cmds: FMTB when starting the editing session. Use FMTA or FMTS to save changes.

Example:
```
FMTB // Start format editing session
     // One or more format editing
     // commands
     // ...
FMTA My_fmt1 // Save edited format as
     // My_fmt1
FMTE // End format editing session
```

FMTG ForMaT in buffer Good

Classification: Format editor control

Command Syntax: *None*

Query Syntax: FMTG?

Query Response: <test result>

Description: The FMTG? query tests the format in the format buffer for errors. If no errors are found, FMTG? returns zero. Otherwise, if one or more errors exist, the number of the first error encountered is returned. To test formats residing in format memory, use the FMTT? query.

Other Required Cmds: *None*

Example:
```
FMTG? // Return format error status
```

FMTJ ForMaT Justify

Classification: Format editor control

Command Syntax: FMTJ

Query Syntax: *None*

Description: The FMTJ command corrects some types of timing errors for the current data in the format buffer. The following errors are corrected:

- Pixel Rate errors 2071 and 2072
- Video Memory Size error 2550
- Horizontal Blanking errors 2140, 2141, 2150 and 2155
- Horizontal Total errors 2090 and 2091
- Horizontal Sync Pulse Width errors 2181 and 2201
- Vertical Resolution error 2321

The justification routine tries to keep the format close to your original specifications. However, the format should be reviewed after it is justified to make sure it still meets your timing requirements.

Other Required Cmds: The FMTU command instructs the generator to use the new settings. The ALLU command updates hardware to the new settings and redraws the test image. The FMTG? query can be used to see if any errors remain.

Example:

```
FMTJ
FMTU // Update hardware to current
     // buffer contents
```

FMTK ForMaT Kill from memory by name

Classification: Format memory management

Command Syntax: FMTK <name>

Limits: <name> = a valid MS-DOS filename (8 characters minus any extension)

Query Syntax: FMTK? <name>

Limits: <name> = a valid MS-DOS filename (8 characters minus any extension)

Query Response: 0 *or* <location>

Description: The FMTK command erases the named format from memory. The FMTK? query checks to see if the named format can be erased. The RAM location number is returned if it can be erased. Otherwise, a zero is returned.

Other Required Cmds: *None*

Example:

```
FMTK my_fmt1 // Erase format called
     // my_fmt1
```

FMTL ForMaT Load from memory by name

Classification: Format memory management / Sequence parameter setting

Command Syntax: FMTL <name>

Limits: <name> = a valid MS-DOS filename
(8 characters minus any extension)

Query Syntax: FMTL? <name>

Query Response: <location> (returns 0 if not found)

Description: The FMTL command is context sensitive. When editing a sequence (while between a set of SEQB and SEQE commands), the FMTL command assigns a format to the step being worked on. The FMTL? query returns the name of the format currently assigned to the step.

Outside of the sequence editor, the FMTL command reads the format having a name equal to name <name> from format memory (or EPROM) into the format. FMTL does not re-configure the signal generating hardware. This feature allows you to work on the contents of any format memory location, while continuing to output a signal based on a previously used format (see FMTU command). The FMTL? query returns the location <location> in which a format having a name equal to <name> is found. If multiple formats exist having name <name>, then the lowest numbered location containing a format with a matching name <name> is returned. The format memory (RAM) is always searched first. If a format with name <name> cannot be found anywhere in the format memory, then the industry-standard formats located in EPROM (negative locations) are searched next. FMTL? returns zero if a format with a name equal to name <name> cannot be found in either format space.

Other Required Cmds: The ALLU command updates the hardware to the new settings and redraws the test image.

Example:

```
FMTL VGA_m3 // Load format called VGA_m3
ALLU Update hardware and redraw image
```

FMTN ForMaT New

Classification: Format editor control

Command Syntax: FMTN [<name>]

Limits: <name> = a valid MS-DOS filename (8 characters minus any extension)

Query Syntax: *None*

Description: The FMTN command initializes the format editing buffer. Sending this one command is equivalent to sending all of the following commands:

```
ASBG 0          GAMA 2.2
ASCT 1          GAMC 0
ASGG 1          HRAT 0
ASSG 0, 1, 0    HRES 0
ASSS 0.286      HSIZ 280
AVPG 0          HSPD 0
AVPS 0.0        HSPG 1
AVSB 0.0        HSPP 0
AVSS 0.714      HSPW 0
AVST 0          HTOT 0
XVSG 1, 1, 1    SCAN 1
CSPG 1          SSST 1
CSPP 0          USIZ 2
DCBM 0          VRES 0
DSEX 0          VSIZ 210
DSST 1          VSPD 0
DVSP 0          VSPG 1
DVST 0          VSPP 0
EQUA 0          VSPW 0
EQUB 0          VTOT 0
```

This should be the first command sent to the generator when creating a new format. The command only resets to a known state. The command does not create a usable format.

Other Required Cmds: *None*

Example: `FMTN // Initialize format buffer`

FMTP ForMaT Path

Classification: Format memory management

Command Syntax: FMTP <name>

Limits: <name> = a valid MS-DOS filename
(8 characters minus any extension)

Query Syntax: FMTP?

Query Response: <name>

Description: The FMTP command sets the current format path name to a given directory. The query will return the current format path name.

Other Required Cmds: *None*

Example:

```
FMTP VGA_FMTS // Formats in VGA_FMTS dir
      // will be listed
```

FMTQ ForMaT Query pointer

Classification: Format memory management

Command Syntax: *None*

Limits: <index> = positive integer number

<number> = positive integer number

Query Syntax: FMTQ? <index> <number>

Query Response: List of specified format names

Description: The query returns <number> format names from the list of all the formats stored in format memory beginning at <index>. The formats are kept in alphanumeric order.

Other Required Cmds: *None*

Examples:

```
FMTQ? 1 5 // List the first five images
      // in memory
```

or

```
FMTQ? 1 9999 // List all images in
      // memory
```

FMTR — ForMaT Read from memory location (by number)

Classification: Format memory management

Command Syntax: FMTR <location>

Limits: <location> = 1 through 300 (RAM) or -1 through -24 (EPROM)

Query Syntax: FMTR? <location>

Query Response: <name>

Description: Reads format from the format memory location into the format buffer. The FMTR command does not reconfigure the signal generating hardware. The FMTR? query returns the <name> of the format stored in location <location>. FMTR? returns the string EMPTY if the format memory location <location> is empty.

Other Required Cmds: The ALLU command updates the hardware to the new settings and redraws the test image.

Example:

```
FMTR -5 // Load format from ROM location
        // 5
ALLU // Update hardware and redraw
     // image
```

FMTS — ForMaT Save

Classification: Format memory management

Command Syntax: FMTS

Description: The FMTS command saves the current format edit buffer contents into format memory using the current name of the format in the edit buffer.

Other Required Cmds: *None*

Example:

```
FMTS
```

FMTU ForMaT Use

Classification: Format memory management

Command Syntax: FMTU

Query Syntax: FMTU?

Query Response: <location> or 0

Description: The FMTU command first checks the current contents of the format buffer for errors. If no errors are found, it reconfigures the signal generating hardware in accordance with the contents. It does not redraw the previously displayed test image. In some cases this may distort the old image.

The FMTU? query returns either a format memory location <location> or zero. If the signal format currently being output by the signal generating hardware matches that originally loaded (using the FMTL command) or read (using the FMTR command) from a format memory location, then the matching format memory location <location> is returned. Otherwise, if the format contents have been used to update the signal generating hardware since either an FMTR or FMTL command has been issued, then FMTU? query returns zero.

Other Required Cmds: *None*

Example:

```
FMTU // Update hardware to current
     // buffer contents
```

FORM   draw FORMat data block

Classification: Custom image primitive

Command Syntax: FORM <color> <x> <y>

Limits: <color> = available colors

<x> = positive integer number

<y> = positive integer number

Query Syntax: *None*

Description: Displays basic information about the format driving the display. The first line shows the number of horizontal active pixels and vertical active lines. The last number on the line is the number of fields per frame (one (1) for non-interlaced and two (2) for interlaced). The second and third lines show the horizontal and vertical rates respectively. Text is on a black rectangular background with a single pixel border.

FORM uses three (3) parameters. The first is the color of the text and border. The next two (2) are the X and Y coordinates for the top left-hand corner of the block of text.

Other Required Cmds: The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
FORM blue 30 200 // Display format
     // information in blue beginning at
     // X=30, Y=200

ALLU // Update hardware to current
     // buffer contents
```

FRGB   Foreground Red, Green and Blue levels

Classification: System parameter setting

Command Syntax: FRGB <red level> <green level> <blue level>
or
FRGB <common gray level>

Limits: <level>
min = 0 (full off)
max = 255 (full on)

Query Syntax: FRGB?

Query Response: <red level> <green level> <blue level>

Description: Temporarily sets the portions of an image drawn with a color selection of *foreground* to the given red, green and blue values. All three colors can be set to the same level using a single parameter. The setting is not global and is not saved. The FRGB? query returns the current red, green and blue settings of FRGB.

Other Required Cmds: The color selection for one or more primitives in a custom image must be set to *foreground* in order to see the affect of this command on a custom image.

Examples:

```
FRGB 255 128 0 // Set foreground color
      // to orange
```

or

```
FRGB 128 // Set foreground color to a
      // mid-gray level
```

GAMA GAMmA correction factor

Classification: Format parameter setting

Command Syntax: GAMA <factor>

Limits: <factor> (floating point accepted)
min = 0.1
max = 10.0

Query Syntax: GAMA?

Query Response: <factor>

Description: The GAMA command establishes the current analog video gamma correction factor. The GAMA? query returns the current setting of the gamma correction factor.

Note – See the section of the manual dealing with editing a format with the Windows user interface for more information on gamma correction.

Other Required Cmds: Gamma correction must be enabled with the GAMC command in order to use the gamma correction factor The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
GAMC 1 // Enable gamma correction in
       // buffer
GAMA 2.2 // Set correction factor in
       // buffer
FMTU // Update hardware to current
       // buffer contents
```

GAMC GAMma Correction mode

Classification: Format parameter setting

Command Syntax: GAMC <mode>

Limits: <mode>
0 = disable (don't correct)
1 = enable (correct)

Query Syntax: GAMC?

Query Response: <mode>

Description: The GAMC command enables or disables application of the analog video gamma correction factor. The GAMC? query can be used to determine if the gamma correction factor is currently being applied.

Other Required Cmds: The value used for gamma correction is set with the GAMA command. The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
GAMC 1 // Enable gamma correction in
       // buffer
GAMA 2.2 // Set correction factor in
       // buffer
FMTU // Update hardware to current
       // buffer contents
```

GETA GET data from Absolute memory location

Classification: Direct processor control

Command Syntax: *None*

Query Syntax: GETA? <address>

Limits: <address>
0 to 4,294,967,295 (BASE = 10)
-2,147,483,648 to 2,147,483,647 (BASE = -10)
0 to FFFFFFFF (BASE = 16)
-80000000 to 7FFFFFFF (BASE = -16)

Query Response: <value>

Description: The GETA? query returns the value of the data stored at the memory <address> specified. Up to 32 bits can be read with this query (see SIZE command). The returned value <value> is formatted according to the current setting of BASE (see the BASE command).

Note – This command will normally be used only with custom applications and command files created by Quantum Data.

Other Required Cmds: GETA and GETA? expect and return parameters formatted according to the current radix set by the BASE command.

Example: `Use only with code supplied by Quantum Data!`

GETR GET data from Relative memory location

Classification: Direct processor control

Command Syntax: *None*

Query Syntax: GETR?

Query Response: <value>

Description: The GETR? query returns the value of the data stored at the memory location currently pointed to by the address register (see ADDR command). Up to 32 bits can be read with this query (see SIZE command). The returned value <value> is formatted according to the current setting of BASE (see the BASE command). The address register is automatically incremented by SIZE bits after the current location has been read.

Note – This command will normally be used only with custom applications and command files created by Quantum Data.

Other Required Cmds: GETR and GETR? expect and return parameters formatted according to the current radix set by the BASE command.

Example: `Use only with code supplied by Quantum Data!`

GRID — draw a centered GRID

Classification: Custom image primitive

Command Syntax: GRID <color> <number of horizontal boxes> <number of vertical boxes>

Limits: <color> = available colors

<number of horizontal boxes> = half of number of pixels

<number of vertical boxes> = half of number of lines

Query Syntax: *None*

Description: Draws a crosshatch of a given color and forms a given number of boxes in each direction. All lines are one (1) pixel thick. All of the lines, in a given direction, are equally spaced. Any remaining pixels are distributed as equally as possible around the perimeter of the grid. This may cause the first and last lines in each direction not to be at the very edges of video.

Other Required Cmds: The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
GRID gray75 14 10 // Draw a gray75 grid
      // with 14 horizontal and 10˚
      // vertical boxes

ALLU // Update hardware to current
      // buffer contents
```

GRIH — draw a GRIll pattern of Horizontal lines

Classification: Custom image primitive

Command Syntax: GRIH <color> <number of pixels in line> < number of pixels in space>

Limits: <color> = available colors

<number of pixels in line> = number of pixels

< number of pixels in space> = number of pixels

Query Syntax: *None*

Description: Draws equally spaced horizontal lines that form a grill over the entire active video area. The primitive uses three (3) parameters. The first is the color of the lines. The second is the thickness of the lines and the third is the thickness of the space between the lines.

Other Required Cmds: The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
GRIH green 4 6 // Draw green grid with
      // 4-pixel lines and 6-pixel spaces

ALLU // Update hardware to current
      // buffer contents
```

GRIV draw a GRIll pattern of Vertical lines

Classification: Custom image primitive

Command Syntax: GRIV <color> <number of pixels in line> < number of pixels in space>

Limits: <color> = available colors

<number of pixels in line> = number of lines

< number of pixels in space> = number of lines

Query Syntax: *None*

Description: Draws equally spaced vertical lines that form a grill over the entire active video area. The gaps are not touched and will show any previously drawn primitives. The primitive uses three (3) parameters. The first is the color of the lines. The second is the thickness of the lines and the third is the thickness of the space between the lines.

Other Required Cmds: The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
GRIV cyan 16 16 // Draw cyan grill with
     // 16-pixel lines and 16-pixel
     // spaces

ALLU // Update hardware to current
     // buffer contents
```

HATI — draw a centered crossHATch from the Inside out

Classification: Custom image primitive

Command Syntax: HATI <color> <number of horizontal boxes> <number of vertical boxes>

Limits: <color> = available colors

<number of horizontal boxes> = half of number of pixels

<number of vertical boxes> = half of number of lines

Query Syntax: *None*

Description: Draws a crosshatch from the *Inside-Out* of a given color and forms a given number of boxes in each direction. The primitive has center lines that divide the active video exactly in half in each direction. The vertical center line is two (2) pixels thick if the format has an even number of active pixels. The horizontal center line is two (2) pixels thick if the format has an even number of active lines. All other lines are one (1) pixel thick. If you enter an odd number of boxes, a half box is placed at each end of the crosshatch. All lines in a given direction are spaced equally. Any remaining pixels are distributed as equally as possible around the perimeter of the grid. This may cause the first and last lines in each direction not to be at the very edges of video. In turn, this may cause any half boxes to be slightly larger.

Other Required Cmds: The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
HATI green 15 9 // Draw a green grid
     // with 15 horizontal and 9
     // vertical boxes

ALLU // Update hardware to current
     // buffer contents
```

HATO draw a centered crossHATch from the Outside in

Classification: Custom image primitive

Command Syntax: HATO <color> <number of horizontal boxes> <number of vertical boxes>

Limits: <color> = available colors

<number of horizontal boxes> = half of number of pixels

<number of vertical boxes> = half of number of lines

Query Syntax: *None*

Description: Draws a crosshatch from the *Outside-In* of a given color and forms a given number of boxes in each direction. All lines are one (1) pixel thick. The first and last lines in each direction are at the very edges of active video. All the lines in a given direction are spaced equally. Any remaining pixels are added to the boxes along the horizontal and vertical centers of the image.

Other Required Cmds: The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
HATO yellow 15 9 // Draw a yellow grid
     // with 15 horizontal and 9
     // vertical boxes

ALLU // Update hardware to current
     // buffer contents
```

HRAT — Horizontal RATe

Classification: Format parameter setting

Command Syntax: HRAT <frequency in Hz>

Limits: <frequency in Hz> (floating point accepted)
typical min = 1000
typical max = 130000

Query Syntax: HRAT?

Query Response: <frequency in Hz> (floating point returned)

Description: The HRAT command sets the line frequency. Pixel rate is equal to the product: HTOT times HRAT. Frame rate is equal to the quotient: HRAT divided by VTOT. Field rate is equal to the product: SCAN times the frame rate. The HRAT? query returns the current horizontal frequency setting.

Other Required Cmds: The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Examples:

```
HRAT 32768 // Set 32.768 kHz rate in
           // buffer
FMTU // Update hardware to current
     // buffer contents
```

or

```
HRAT 32.768E3 // Set a 32.768 kHz rate
FMTU // Update hardware to current
     // buffer contents
```

HRES — Horizontal RESolution

Classification: Format parameter setting

Command Syntax: HRES <pixels>

Limits: <pixels>
min = 16
max depends on VRES and model of generator

Query Syntax: HRES?

Query Response: <pixels>

Description: The HRES command establishes the number of active pixels per line. The HRES? query returns the current setting of HRES.

Other Required Cmds: The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
HRES 480 // Set 480 active pixels line
         // in buffer
ALLU // Configure hardware and redraw
     // image
```

HSIZ Horizontal SIZe

Classification: Format parameter setting

Command Syntax: HSIZ <physical size> (context sensitive - see FMTB and FMTE)

Limits: <physical size> = positive value (floating point accepted)

Query Syntax: HSIZ?

Query Response: <physical size> (floating point returned)

Description: The HSIZ command establishes the horizontal physical size of the image on the display. Units expected (or returned) vary according to the last mode set with USIZ command. The HSIZ command is context sensitive and must appear between begin and end commands: FMTB and FMTE. The HSIZ? query returns the current setting of HSIZ.

Note – Make sure that the USIZ parameter is properly set before using the HSIZ command. Changing the USIZ setting after entering HSIZ will convert the size to match the new unit of measure.

Other Required Cmds: The units of measure must be properly set by USIZ before entering HSIZ. The ALLU command updates hardware to the new setting and redraws the test image, taking the new size into account.

Example:

```
FMTB // Begin editing session One or
     // more format editing commands
     // ...
USIZ 1 // Select inches as unit of
     // measure in buffer
HSIZ 10.4 // Set width to 10.4 in
     // buffer
VSIZ 7.8 // Set height to 7.8 in buffer
ALLU // Test the new settings
     // One or more format editing
     // commands
     // ...
FMTE // End of editing session
```

HSPD Horizontal Sync Pulse Delay

Classification: Format parameter setting

Command Syntax: HSPD <pixels>

Limits: <pixels>
min = 1
max = HTOT - HRES - HSPW

Query Syntax: HSPD?

Query Response: <pixels>

Description: The HSPD command establishes the delay between the leading edge of blanking and the leading edge of the horizontal sync pulse. The HSPD? query returns the current setting of HSPD.

Other Required Cmds: The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
HSPD 16 // Set 16 pixel pulse delay in
        // buffer
FMTU    // Update hardware to current
        // buffer contents
```

HSPG Horizontal Sync Pulse Gate

Classification: Format parameter setting

Command Syntax: HSPG <mode>

Limits: <mode>
0 = OFF
1 = ON

Query Syntax: HSPG?

Query Response: <mode>

Description: The HSPG command enables and disables the digital horizontal sync output. The HSPG? query returns the current HSPG mode.

Other Required Cmds: In order to use digital horizontal sync, digital separate H&V sync must be selected with the SSST command. The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
HSPG 1 // Enable H sync output in
       // buffer
FMTU   // Update hardware to current
       // buffer contents
```

HSPP Horizontal Sync Pulse Polarity

Classification: Format parameter setting

Command Syntax: HSPP <polarity>

Limits: <polarity>
0 = active-low (negative going pulse)
1 = active-high (positive going pulse)

Query Syntax: HSPP?

Query Response: <polarity>

Description: The HSPP command establishes the logic sense of the digital horizontal sync outputs. Setting polarity to one (1) causes the leading edge of horizontal sync to be a low-to-high transition. Setting polarity to zero (0) causes the leading edge of horizontal sync to be a high-to-low transition. The HSPP? query returns the current polarity of HSPP.

Other Required Cmds: In order to use digital horizontal sync, it must be gated on with the HSPG command and digital separate H&V sync must be selected with the SSST command. The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
HSPP 0 // Set active lo H sync in
       // buffer
HSPG 1 // Enable H sync output in
       // buffer
SSST 1 // Select H&V sync type in
       // buffer
FMTU // Update hardware to current
     // buffer contents
```

HSPW Horizontal Sync Pulse Width

Classification: Format parameter setting

Command Syntax: HSPW <pixels>

Limits: <pixels>
min = 1
max = HTOT - HRES - HSPD

Query Syntax: HSPW?

Query Response: <pixels>

Description: The HSPW command establishes the width of the horizontal sync pulse. The HSPW? query returns the current setting of HSPW.

Other Required Cmds: The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
HSPW 32 // Set pulse width to 32 pixels
        // in buffer
FMTU // Update hardware to current
     // buffer contents
```

HTOT Horizontal TOTal pixels per line

Classification: Format parameter setting

Command Syntax: HTOT <pixels>

Limits: <pixels>
min
801GP = 2
801GC, GF, GX = 144
801GC-ISA, GF-ISA = 144
max
801GP = 2048
801GC, GX = 4096
801GC-ISA = 4096
801GF, 801GF-ISA = 65,536

Query Syntax: HTOT?

Query Response: <pixels>

Description: The HTOT command establishes the total number of pixels per horizontal line. The HTOT? query returns the current setting of HTOT.

The pixel rate is equal to the product of HRAT multiplied by HTOT.

Note – The current version of the firmware does not allow you to directly enter a specific pixel rate when setting up a format. If your test specifications call for a specific pixel or dot clock rate, enter suitable values for HRAT and HTOT to give you the desired pixel rate.

Other Required Cmds: The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
HTOT 800 // Set total to 800
FMTU // Update hardware to current
     // buffer contents
```

*IDN IDeNtification

Classification: Miscellaneous system parameter

Command Syntax: *None*

Query Syntax: *IDN?

Query Response: QuantumData,801GX,0,<firmware version #>

Description: The *IDN? query returns an equipment identification string formatted per IEEE-488.2 standards.

Other Required Cmds: *None*

Example:

```
*IDN? // Return information about
      // generator
```

IMGA IMaGe save As

Classification: Image memory management

Command Syntax: IMGA <name>

Limits: <name> = a valid MS-DOS filename (8 characters minus any extension)

Query Syntax: *None*

Description: The IMGA command saves the current contents of the image edit buffer using the given name.

Other Required Cmds: *None*

Example:
```
IMGA MY_IMG // Save with the name
      // "MY_IMG"
```

IMGB IMaGe editing Begin

Classification: Image editor control

Command Syntax: IMGB

Query Syntax: *None*

Description: The IMGB command marks the beginning of an image editing session.

Other Required Cmds: Either an IMGL command to load an existing image or an IMGN command to create a new image. IMGE when ending the editing session.

Example:
```
IMGN // Initialize image edit buffer
IMGB // Start image editing session
     // One or more image editing
     // commands
     // ...
IMGE // End image editing session
```

IMGE IMaGe editing End

Classification: Image editor control

Command Syntax: IMGE

Query Syntax: *None*

Description: The IMGE command marks the end of an image editing session.

Other Required Cmds: IMGB when starting the editing session. Use IMGA or IMGS to save changes.

Example:
```
IMGB // Start image editing session
     // One or more image editing
     // commands
     // ...
IMGA MYIMG_02 // Save edited image as
     // MYIMG_02
IMGE // End image editing session
```

IMGK IMaGe Kill

Classification: Image memory management

Command Syntax: IMGK <name>

Limits: <name> = a valid MS-DOS filename (8 characters minus any extension)

Query Syntax: IMGK? <name>

Limits: <name> = a valid MS-DOS filename (8 characters minus any extension)

Query Response: 0 or 1

Description: The IMGK command deletes an image by name. The query returns a one (1) if the named image can be deleted. If the image is read-only or nonexistent, the query returns a zero (0).

Other Required Cmds: *None*

Example:

```
IMGK MY_IMG // Delete dir called
       // "MY_IMG"
```

IMGL IMaGe Load

Classification: Image memory management / Sequence parameter setting

Command Syntax: IMGL <name>

Limits: <name> = a valid MS-DOS filename (8 characters minus any extension)

Query Syntax: IMGL? <name>

Limits: <name> = a valid MS-DOS filename (8 characters minus any extension)

Query Response: 0 or 1

Description: The IMGL command is context sensitive. When editing a sequence (while between a set of SEQB and SEQE commands), the IMGL command assigns an image to the step being worked on. The IMGL? query returns the name of the image currently assigned to the step.

Outside of the sequence editor, the IMGL command copies the image having a name equal to <name> from image memory into the image edit buffer. The query returns a one (1) if the named image can be loaded, otherwise a zero (0) is returned.

Other Required Cmds: An ALLU or IMGU command must be executed after the IMGL command to cause the image in the edit to draw on the unit under test.

Example:

```
IMGL MY_IMG // Load "MY_IMG" dir in
       // edit buffer
IMGU // Draw contents of buffer
```

IMGN IMaGe New

Classification: Image editor control

Command Syntax: IMGN [<name>]

Limits: <name> = optional valid MS-DOS filename
(8 characters minus any extension)

Query Syntax: IMGN?

Query Response: <name>

Description: The IMGN command initializes the image edit buffer. The name <name> is assigned as the image's name. The query will return the name that has been assigned as the image's name.

Other Required Cmds: *None*

Examples:

```
IMGN // Init edit buffer without
     // assigning a new name
```

or

```
IMGN MY_IMG // Init edit buffer with
     // name of "MY_IMG"
```

IMGP IMaGe Path

Classification: Image memory management

Command Syntax: IMGP <name>

Limits: <name> = a valid MS-DOS filename
(8 characters minus any extension)

Query Syntax: IMGP?

Query Response: <name>

Description: The FMTP command sets the current image path name to a given directory. The query will return the current image path name.

Other Required Cmds: *None*

Example:

```
IMGP FOCUS // List Images in FOCUS dir
```

IMGQ — IMaGe Query pointer

Classification: Image memory management

Command Syntax: *None*

Query Syntax: IMGQ? <index> <number>

Limits: <index> = positive integer number

<number> = positive integer number

Query Response: List of specified image names

Description: The query returns <number> image names from the list of all the images stored in image memory beginning at <index>. The images are kept in alphanumeric order.

Other Required Cmds: *None*

Examples:

```
IMGQ? 1 5 // List the first five images
      // in memory
```

or

```
IMGQ? 1 9999 // List all images in
      // memory
```

IMGR — IMaGe Read image from image memory location

Classification: Image memory management

Command Syntax: IMGR <location>

Limits: <location>
-1 through -26 (EPROM)

Query Syntax: *None*

Description: The IMGR command copies the image residing in the image memory with location <location> into the image buffer. The IMGR command does not cause the selected image to be drawn. See the IMGU command for actually drawing the image. Using the IMGL command to load images by name is the preferred method of selecting images.

Other Required Cmds: The IMGU command draws the image. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
IMGR 1 // Select first custom image in
      // memory
IMGU // Draw the image
```

IMGS IMaGe Save

Classification: Image memory management

Command Syntax: IMGS

Query Syntax: *None*

Description: The IMGS command saves the current contents of the generator's custom image edit buffer back to the memory location from which it was originally read.

Other Required Cmds: *None*

Example: `IMGS`

IMGU IMaGe Use

Classification: Image memory management

Command Syntax: IMGU

Query Syntax: IMGU?

Query Response: <location>

Description: The IMGU command draws an image based on the current contents of the image . The IMGU? query returns the image memory location <location> from which the current contents of the image were read. See the IMGR command for setting the contents of the image.

Other Required Cmds: *None*

Example:

```
IMGL BriteBox // Select the BriteBox
     // test image
IMGU // Draw the selected test image
```

INIT INITialize to factory default settings

Classification: Direct processor control

Command Syntax: INIT

Query Syntax: *None*

Description: The INIT command restores the contents of all of the generator's RAM storage locations to factory-default conditions. The generator then goes through a complete self-test and self-calibration procedure.

WARNING: The INIT command permanently and irreversibly removes all user-created formats, custom images, test sequences and directories from memory.

Other Required Cmds: *None*

Example: `INIT`

IVER — Image VERsion

Classification: System parameter setting / Sequence parameter setting

Command Syntax: IVER <mode>

Limits: <mode>
- 0 = Normal
- 1 = Invert or display alternate version

Query Syntax: IVER?

Query Response: <mode>

Description: The IVER command is context sensitive. When editing a sequence (while between a set of SEQB and SEQE commands), the IVER command determines which version of an image is used for the step being worked on. The IVER? query returns the version currently assigned to the step.

Outside of the sequence editor, the IVER command selects which version of the current image is drawn when either an ALLU or IMGU command is executed. The IVER? query returns the current setting of IVER.

Other Required Cmds: The IMGU command redraws an image using the new setting. The ALLU command updates hardware and redraws the test image with the new setting.

Example:

```
IMGL Text_9 // Select image with white
     // text on black
IVER 1 // Select inverted with black on
     // white
IMGU // Draw the image as selected
```

JRAT — Justify pixel clock RATe

Classification: Format editor control

Command Syntax: JRAT <pixel rate>

Limits: <pixel rate> = Floating point number equal to the desired pixel in MHz

Query Syntax: *None*

Description: The JRAT command scales the horizontal timing parameters of the format currently in the edit buffer. The parameters are scaled to produce the given pixel rate while keeping the horizontal scan rate as close as possible to its original value. The following parameters are scaled: Horizontal total pixels, Horizontal active pixels, Horizontal sync delay in pixels and Horizontal sync pulse width in pixels. The parameters are scaled so that their periods, in microseconds, are as close as possible to their original values.

Other Required Cmds: The ALLU command updates hardware to the new settings and redraws the test image.

Example:

```
JRAT 28.322 // Adjust timing to a
     // 28.322 MHz clock
FMTU // Update hardware to current
     // buffer contents
```

KEYY KEY toggle

Classification: System parameter setting

Command Syntax: KEYY <button #>

Limits: <button #>
- 1 = Image (/Step) version
- 2 = Red gating
- 3 = Green gating
- 4 = Blue gating
- 5 = ACS gating
- 6 = DCS gating
- 7 = DSS gating
- 8 = Outputs gating

Query Syntax: *None*

Description: The KEYY command can toggle the status of the following items: Image version of the currently displayed image, Red, Green and Blue video gating, ACS / DCS / DSS Sync gating and Outputs gating. The order of the *buttons* matches the order of the buttons found on the front panel of the generator. The current status of the buttons can be checked using the LEDS? query.

Other Required Cmds: *None*

Example:
```
KEYY 8 // Toggle current status of
       // output gates
```

KNOB rotate KNOB

Classification: System parameter setting

Command Syntax: KNOB <number> <clicks>

Limits: <number>
1 = upper *Format* knob
2 = lower *Image* knob

<clicks>
positive integer = knob clicks in a clockwise direction
negative integer = knob clicks in a counterclockwise direction

Query Syntax: *None*

Description: The KNOB command has the same affect on generator operation as if an operator were manually turning either the "Format" or "Image" knobs on the generator.

When a sequence is running, the "Format" knob loads sequences from the directory selected by the SEQP command. Clockwise rotation increases the index pointer for the directory. The "Image" knob is used to go back and forth through the sequence steps. Clockwise rotation selects higher step numbers.

Other Required Cmds: *None*

Example:
```
KNOB 2 1 // Move Image knob 1 click
    // CW
```

LCDS LCD Status

Classification: System parameter setting

Command Syntax: *None*

Query Syntax: LCDS?

Query Response: Response in normal format/image selection mode...
H<*Horiz Rate*> <*Fmt Mem Loc*> = <*Fmt Name*><cr><lf>
V<*Vert Rate*> <*Img Mem Loc*> = <*Img Name*>

Description: The LCDS? query returns text string data that matches what is shown in the LCD window of the generator.

Other Required Cmds: *None*

Example:
```
LCDS? // Return text similar to text
    // below
    // H32 15=VGA_m3 <cr><lf>
    // V60 51=SMPTE133
```

LEDS LED Status

Classification: System parameter setting

Command Syntax: *None*

Query Syntax: LEDS?

Query Response: *<decimal number from 0 to 255>*

Description: The LEDS? query returns the current status of the generator's signal generating hardware as a single decimal number. The number corresponds to the status of the lighted push-button on the generator in normal operation. The easiest way to interpret the number is to first convert it to an eight (8) digit binary number. A one (1) in a given position, from MSB to LSB, corresponds to the following hardware settings:

Master output control gated ON (MSB)
Digital Separate (HS&VS) Sync selected
Digital Composite Sync selected
Analog Composite Sync selected
Blue video enabled
Green video enabled
Red video enabled
Alternate image version selected (LSB)

Other Required Cmds: *None*

Example:

```
LEDS? // Return the following number
      // when the outputs are gated ON,
      // separate HS & VS is selected,
      // the red, green and blue video
      // channels are enabled and the
      // primary version of an image is
      // selected.

206 // Binary equivalent = 11001110
```

LIMI draw video LIMIts markers

Classification: Custom image primitive

Command Syntax: LIMI <color>

Limits: <color> = available colors

Query Syntax: *None*

Description: Places nine (9) markers that define the active video area. An L-shaped marker is at each corner. T-shaped markers are centered along each edge of video and a small cross is at the center of video. The primitive uses a single parameter ... color.

Other Required Cmds: The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
LIMI white // Place white markers that
           // define active video area

ALLU // Update hardware to current
     // buffer contents
```

LINE draw a LINE

Classification: Custom image primitive

Command Syntax: LINE <color> <X start coordinate> <Y start coordinate> <X end coordinate> <Y end coordinate>

Limits: <color> = available colors

<X start coordinate> <Y start coordinate> <X end coordinate> <Y end coordinate> = positive integer number

Query Syntax: *None*

Description: Draws a line between any two points. The line is one (1) pixel thick. The primitive uses five (5) parameters, the color and the X and Y coordinates for both endpoints.

Other Required Cmds: The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
LINE yellow 20 5 320 240 // Draw yellow
     // line from X=20, Y=5 to X=320,
     // Y=240

ALLU // Update hardware to current
     // buffer contents
```

MODE communications MODE

Classification: System parameter setting

Command Syntax: MODE <baud> [<parity> [<data > [<stop> [<handshake> [< protocol>]]]]]

Limits: <baud *rate*>
300, 600, 1200, 2400, 4800, 9600, 19200 or 38400 bits/sec
<parity>
N = None, E = Even or O = Odd
<# data *bits*>
7 or 8
<# stop *bits*>
1 or 2
<handshake>
N = None, S = Software (XON/XOFF) or
H = Hardware (CTS/RTS)
< protocol >
N = None, Y=Y-Modem

Query Syntax: MODE?

Query Response: <baud> <parity> <data > <stop> <handshake> < protocol>
or
<ISA addr>

Description: The MODE command is normally used to set the serial port communications parameters of a stand-alone model generator. The changes take effect as soon as the command is entered. The command has no affect on the ISA addressing configuration of the -ISA model generators. The MODE? query returns the current serial port communications settings of a stand-alone generator or the ISA address of an -ISA model generator.

Other Required Cmds: *None*

Examples:
```
MODE 9600 N 8 1 H N // Set a stand
     // alone generator to communicate
     // at 9600 Baud, No parity, 8 data
     // bits, 1 stop bit, RTS//CTS
     // handshaking and No protocol
```
or
```
MODE 2400 // Change only the baud rate
```

MSIZ light Meter SIZe

Classification: System parameter setting

Command Syntax: MSIZ <width>, <height>
or
MSIZ <common size> (for a square box)

Limits: <size> = positive floating point number

Query Syntax: MSIZ?

Query Response: <width>, <height>

Description: The MSIZ command establishes the physical size of the lightmeter box(es) displayed in the BriteBox test image. The unit of measure used is based on the current setting of the system level USIZ parameter. It also affects the size of the cursor boxes in the Persist image. Changing the size will not change the currently displayed image. The MSIZ? query returns the current settings of MSIZ based on the current setting of the system level USIZ parameter.

Other Required Cmds: The correct unit of measure should be selected with the USIZ command prior to setting the size.

Example:

```
USIZ 1 // Select inches for units
MSIZ 2.0 // Set size to 2.0 inches IMGL
      // BriteBox Select BriteBox test
      // image IMGU Draw selected image
      // using new size
```

NAMF NAMe Find

Classification: Directory editor control

Command Syntax: *None*

Query Syntax: NAMF? <name>

Limits: <name> = a valid MS-DOS filename
(8 characters minus any extension)

Query Response: <index>

Description: The query will return the index number of the entry with name <name> in the directory edit buffer. The first name in the buffer has an index value of 1. If <name> is not found, a value of zero (0) is returned.

Other Required Cmds: *None*

Example:

```
NAMF? VGA_m4 // Return position of
      // VGA_m4 in directory
```

NAMI — NAMe Insert

Classification: Directory editor control

Command Syntax: NAMI <index> <name>

Limits: <name> = a valid MS-DOS filename
(8 characters minus any extension)

Query Syntax: *None*

Description: The NAMI command first moves all the names with index values equal to or greater than <index> to the next higher index value in the directory edit buffer. It then inserts the name <name> in the directory edit buffer at position <index>. A negative number or a zero (0) used for <index> will put <name> at index position one (1). Using a number for <index> beyond the last name in the buffer will add <name> to the index position just beyond the last name. The command does not check if a file called <name> is stored in the generator.

Other Required Cmds: *None*

Examples:

```
NAMI 5 FOCUS1 // Put FOCUS1 in fifth
       // position
```

or

```
NAMI -6 BARS // Put BARS at beginning
       // of dir
```

or

```
NAMI 999999 HATCH // Put HATCH at end
       // of dir
```

NAMK — NAMe Kill

Classification: Directory editor control

Command Syntax: NAMK <name>

Limits: <name> = a valid MS-DOS filename
(8 characters minus any extension)

Query Syntax: *None*

Description: The NAMK command deletes the given name from the directory edit buffer. All names with index values greater than the index of the deleted name are moved to the next lower index value in the buffer. Nothing happens if <name> is not found in the buffer.

Other Required Cmds: *None*

Example:

```
NAMK VGA_m4 // Remove VGA_m4 from dir
```

NAMQ NAMe Query

Classification: Directory editor control

Command Syntax: *None*

Query Syntax: NAMQ? <index> <number>

Query Response: List of specified entry names from directory edit buffer

Description: The query returns <number> names from the list of names in the directory edit buffer beginning at <index>.

Other Required Cmds: *None*

Examples:

```
NAMQ? 1 10 // List the first ten names
       // in the buffer
```

or

```
NAMQ? 1 9999 // List the entire buffer
```

NAMY NAMe Yank

Classification: Directory editor control

Command Syntax: NAMY <index>

Query Syntax: *None*

Description: The NAMY command deletes the name at the given <index> number from the directory edit buffer. All names with index values greater than the index of the deleted name are moved to the next lower index value in the buffer. Nothing happens if <index> is beyond the index number of last name in the buffer.

Other Required Cmds: *None*

Example: `NAMY MyFMT // Remove MyFMT from dir`

NOGA use NO GAmma correction

Classification: Custom image primitive

Command Syntax: NOGA

Limits: *None*

Query Syntax: *None*

Description: Temporarily disables any gamma correction that may be selected in a format. All color intensity levels in all parts of the custom image are output without gamma correction. Gamma correction is disabled only for as long as the image is displayed.

Other Required Cmds: *None*

Example: `NOGA`

OUTG OUTputs Gate

Classification: System parameter setting

Command Syntax: OUTG <mode>

Limits: <mode>
0 = OFF
1 = ON

Query Syntax: OUTG?

Query Response: <mode>

Description: The OUTG command gates all video and sync outputs of the generator ON and OFF. Gating the outputs OFF forces all outputs to be turned off. Gating the outputs ON turns on all outputs whose individual gating settings are turned ON. The OUTG? query returns the current status of the outputs of the generator.

Other Required Cmds: *None*

Example:
```
OUTG 0 // Disable all outputs
```

OVAL draw an OVAL

Classification: Custom image primitive

Command Syntax: OVAL <color> <width> <height> <x> <y> <fill pattern>

Limits: <color> = available colors

<width> = total number of horizontal pixels

<height> = total number of lines

<x> = positive integer number

<y> = positive integer number

<fill pattern> = available fill paterns

Query Syntax: *None*

Description: Draws an oval whose axes are parallel to the vertical and horizontal axes of displayed video. The size and position of the oval are defined by its framing rectangle. The framing rectangle is a rectangle whose sides are both tangent to the oval at four points and are parallel to the vertical and horizontal axis of video. It's not drawn as part of the primitive.

Oval uses six (6) parameters. The first is color. The next two are the pixel width and height of the framing rectangle. The fourth and fifth parameters are the X and Y coordinates for the top left-hand corner of the framing rectangle. The last parameter is the fill.

Other Required Cmds: The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:
```
OVAL red 240 150 20 10 GrayPat0 // Draw
     // a red oval 240 pixels wide by
     // 150 pixels high. Start framing
     // rectangle at X=20, Y=10. Fill =
     // none
```

PAGE — draw a PAGE of repeating characters

Classification: Custom image primitive

Command Syntax: PAGE <color> <width> <height> <x> <y> <fontname> <character>

Limits:
<color> = available colors

<width> = width of page in pixels

<height> = height of page in lines

<x> = position of left edge of page in pixels

<y> = position of top edge of page in pixels

<fontname> = available fonts

<character> = code number of character

min = 0

max = 255

Query Syntax: *None*

Description: Fills a rectangular area (page or block) with a character that repeats horizontally and vertically. The <color> parameter sets the color used to draw the character. The <width> and <height> parameters determine the size of the block to be filled. The <x> and <y> parameters determine the top left corner of the block. The <fontname> parameter selects which font is used to draw the character. The <character> parameter selects a specific character (by number). For full alphanumeric fonts, the character number is the same as the character's ASCII code number. The spacing between the characters is fixed by the character block size in the font and cannot be changed. Partial characters are not drawn to completely fill the rectangular area. Rather, the largest possible block of full characters is centered in the rectangular area.

Other Required Cmds: *None*

Example:
```
PAGE blue 40 30 10 10 opix9 69 // Draw
     // a small block of blue E
     // characters in the upper left
     // corner
```

PCPG Pixel Clock Pulse Gate

Classification: Format parameter setting

Command Syntax: PCPG <mode>

Limits: <mode>
0 = OFF
1 = ON

Query Syntax: PCPG?

Query Response: <mode>

Description: The PCPG command enables and disables the pixel clock pulse output on generators that have a pixel clock output available. The PCPG? query returns the current setting of PCPG.

Other Required Cmds: FMTU or ALLU

Example:

```
PCPG 1 // Enable pixel clk output in
       // buffer
FMTU // Update hardware with format
       // data
```

PUTA PUT Absolute

Classification: Direct processor control

Command Syntax: PUTA <address> <value>

Limits: <address>
0 to 4,294,967,295 unsigned decimal (BASE = 10)
-2,147,483,648 to 2,147,483,647 (BASE = -10)
0 to FFFFFFFF (BASE = 16)
-80000000 to 7FFFFFFF (BASE = -16)

<value>
0 to 2^(SIZE)-1

Query Syntax: *None*

Description: The PUTA command writes the specified value <value> into memory at the specified address <address>. The two parameters <address> and <value> are interpreted according to the current setting of BASE (see the BASE command). The number and format of the bits written depend on the current setting of SIZE (see the SIZE command).

Note – This command will normally be used only with custom applications and command files created by Quantum Data. Indiscriminate use of this command can cause the generator to stop operating and / or the loss of stored data in nonvolatile RAM.

Other Required Cmds: PUTA expects parameters formatted according to the current radix set by the BASE command.

Example:

```
Use only with code supplied by Quantum
Data!
```

PUTR — PUT Relative

Classification: Direct processor control

Command Syntax: PUTR <value>

Limits: <value> = 0 to 2^(SIZE)-1

Query Syntax: *None*

Description: The PUTR command writes the specified value <value> into the location pointed to by the address register (see the ADDR command). The parameter <value> is interpreted according to the current setting of BASE (see the BASE command). The number and format of the bits written depend on the current setting of SIZE (see the SIZE command). The address register is automatically incremented by SIZE bits after the current location has been written to.

Note – This command will normally be used only with custom applications and command files created by Quantum Data. Indiscriminate use of this command can cause the generator to stop operating and / or the loss of stored data in nonvolatile RAM.

Other Required Cmds: PUTR expects parameters formatted according to the current radix set by the BASE command.

Example:

```
Use only with code supplied by Quantum
Data!
```

RATC — pixel RATe Calibration factor

Classification: System parameter setting

Command Syntax: RATC <factor>

Limits: <factor> = Floating point number equal to calibration factor
Typical min. = 0.99990
Typical max = 1.00010

Query Syntax: RATC?

Query Response: <factor>

Description: The RATC command sets an internal multiplication factor used in setting the pixel clock frequency. The multiplication factor can be set to compensate for the frequency error of the internal reference crystal. Having to use a factor outside of the typical range may indicate a failure of generator's hardware. Re-initializing the generator's memory sets the calibration factor to a factory-default setting of 1.00000.

Note – The RATC parameter is a system level parameter that will affect the pixel clock frequency of all Formats that are recalled. The RATC value will be retained when the generator is powered down and back up again. Query the current setting of RATC if you are experiencing problems with the pixel clock or scan rate being off in frequency.

Other Required Cmds: *None*

Example:

```
RATC 1.00007 // Increases pix clk by
      // factor of 1.00007
```

RECT draw a RECTangle

Classification: Custom image primitive

Command Syntax: RECT <color> <width> <height> <x> <y> <fill pattern>

Limits: <color> = available colors

<width> = total number of horizontal pixels

<height> = total number of lines

<x> = positive integer number

<y> = positive integer number

<fill pattern> = available fill paterns

Query Syntax: *None*

Description: Draws a rectangle whose sides are parallel to the vertical and horizontal axes of displayed video. It uses six (6) parameters. The first is the color of the line. The next two parameters are the pixel width and height of the rectangle. The fourth and fifth parameters are the X and Y coordinates for the top left-hand corner of the rectangle. The last parameter is the fill.

Other Required Cmds: The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
RECT gray50 15 20 50 40 GrayPat50
     // Draw a 50% gray rectangle 15
     // pixels wide and 20 pixels high
     // with top left corner at X=50,
     // Y=40 Fill with 50% active pixels

ALLU // Update hardware to current
     // buffer contents
```

SCAL Self CALibrate

Classification: Direct processor control

Command Syntax: SCAL

Query Syntax: *None*

Description: The SCAL command causes generator equipped with self-calibration circuitry to go through its self-calibration cycle.

Other Required Cmds: *None*

Example:

```
SCAL // Have generator go through self
     // cal
```

SCAN — SCAN fields per frame

Classification: Format parameter setting

Command Syntax: SCAN <fields>

Limits: <fields>
1 = progressive (non-interlaced)
2 = interlaced

Query Syntax: SCAN?

Query Response: <fields>

Description: The SCAN command establishes the number of fields scanned per frame. Set to one (1) for progressive (non-interlaced) scan and two (2) for interlaced scan. The SCAN? query returns the current setting of SCAN.

Other Required Cmds: The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
SCAN // Select interlace in buffer
FMTU // Update hardware to current
     // buffer contents
```

SDLY — Sequence step DeLaY

Classification: Sequence parameter setting

Command Syntax: SDLY <delay>

Limits: <delay> Delay time in seconds as a floating point number
min = 0.0 seconds
max = 1.70E38 seconds (infinity for all practical purposes)

Query Syntax: SDLY?

Query Response: <delay>

Description: The SDLY command sets how long a sequence step will pause before advancing to the next step in the Auto run mode. A sequence step will use the last value set by the SDLY command. The SDLY? query returns the current setting of SDLY.

Other Required Cmds: SMOD setting must be equal to three (3) in order for the SDLY setting to have any affect on sequence operation.

Example:

```
SDLY 5.0 // Set delay to five seconds
     // per step
```

SEQA SEQuence save As

Classification: Sequence memory management

Command Syntax: SEQA <name>

Limits: <name> = a valid MS-DOS filename (8 characters minus any extension)

Query Syntax: *None*

Description: The SEQA command saves the current contents of the sequence edit buffer using the given name.

Other Required Cmds: *None*

Example:
```
SEQA MY_SEQ // Save with the name
       // "MY_SEQ"
```

SEQB SEQuence editing Begin

Classification: Sequence editor control

Command Syntax: SEQB

Description: The SEQB command marks the beginning of a sequence editing session. This command does nothing in the current firmware version, but is used for compatibility with future versions of firmware.

Other Required Cmds: Either a SEQL command to load an existing sequence or a SEQN command to create a new sequence. SEQE when ending the editing session.

Example:
```
SEQN // Initialize sequence edit buffer
SEQB // Start sequence editing session
     // One or more sequence editing
     // commands
     // ...
SEQE // End sequence editing session
```

SEQE SEQuence editing End

Classification: Sequence editor control

Command Syntax: SEQE

Query Syntax: *None*

Description: The SEQE command marks the end of a sequence editing session. This command does nothing in the current firmware version, but is used for compatibility with future versions of firmware.

Other Required Cmds: SEQB when starting the editing session. Use SEQA or SEQS to save changes.

Example:
```
SEQB // Start sequence editing session
     // One or more sequence editing
     // commands
     // ...
SEQE // End sequence editing session
SEQA MYSEQ_02 // Save edited sequence as
     // MYSEQ_02
```

SEQK SEQuence Kill

Classification: Sequence memory management

Command Syntax: SEQK <name>

Limits: <name> = a valid MS-DOS filename (8 characters minus any extension)

Query Syntax: SEQK? <name>

Limits: <name> = a valid MS-DOS filename (8 characters minus any extension)

Query Response: 0 or 1

Description: The SEQK command deletes a sequence by name. The query returns a one (1) if the named sequence can be deleted. If sequence is read-only or nonexistent, the query returns a zero (0).

Other Required Cmds: *None*

Example:

```
SEQK MY_SEQ // Delete seq called
       // "MY_SEQ"
```

SEQL SEQuence Load

Classification: Sequence memory management

Command Syntax: SEQL <name>

Limits: <name> = a valid MS-DOS filename (8 characters minus any extension)

Query Syntax: SEQL? <name>

Limits: <name> = a valid MS-DOS filename (8 characters minus any extension)

Query Response: 0 or 1

Description: The SEQL command copies the sequence having a name equal to <name> from sequence memory into the sequence edit buffer. The query returns a one (1) if the named sequence can be loaded, otherwise a zero (0) is returned.

Other Required Cmds: An SEQU command must be executed after the SEQL command to start running the sequence.

Example:

```
SEQL MY_SEQ // Load "MY_SEQ" dir in
       // edit buffer
SEQU // Start running the sequence in
       // the buffer
```

SEQN — SEQuence New

Classification: Sequence editor control

Command Syntax: SEQN [<name>]

Limits: <name> = optional valid MS-DOS filename
(8 characters minus any extension)

Query Syntax: SEQN?

Query Response: <name>

Description: The SEQN command initializes the sequence edit buffer. The name <name> is assigned as the sequence's name. The query will return the name that has been assigned as the sequence's name.

Other Required Cmds: *None*

Examples:

```
SEQN // Init edit buffer without
     // assigning a new name
```

or

```
SEQN MY_SEQ // Init edit buffer with
     // name of "MY_SEQ"
```

SEQP — SEQuence Path

Classification: Sequence memory management

Command Syntax: SEQP <name>

Limits: <name> = a valid MS-DOS filename
(8 characters minus any extension)

Query Syntax: SEQP?

Query Response: Current sequence path name

Description: The FMTP command sets the current image path name to a given directory. The query will return the current image path name.

Other Required Cmds: *None*

Example:

```
SEQP BURN // Sequences in BURN directory
     // will be listed
```

SEQQ — SEQuence Query pointer

Classification: Sequence memory management

Command Syntax: *None*

Query Syntax: SEQQ? <index> <number>

Limits: <index> = positive integer number

<number> = positive integer number

Query Response: List of specified sequence names

Description: The query returns <number> sequence names from the list of all the sequence names stored in sequence memory beginning at <index>. The directories are kept in alphanumeric order.

Other Required Cmds: *None*

Examples:

```
SEQQ? 1 5 // List the first five
      // directories in memory
```

or

```
SEQQ? 1 9999 // List all directories
      // in memory
```

SEQS — SEQuence Save

Classification: Sequence memory management

Command Syntax: SEQS

Query Syntax: *None*

Description: The SEQS command saves current contents of the generator's sequence buffer back to the memory location from which they were originally read.

Other Required Cmds: *None*

Example:

```
SEQS
```

SEQU — SEQuence Use

Command Syntax: SEQU

Query Syntax: SEQU?

Query Response: <name>

Description: The SEQU command starts to run the sequence currently stored in the sequence edit buffer. The SEQU? query returns the sequence name currently set in the sequence edit buffer.

Other Required Cmds: The sequence in the edit buffer must have a nonzero SMOD setting in order to run.

Example:

```
SEQL MY_SEQ // Load "MY_SEQ" dir in
      // edit buffer
SEQU // Start running the sequence in
      // the buffer
```

SIZE — SIZE of bit field

Classification: Direct processor control

Command Syntax: SIZE <size>

Limits: <size>
-32 to -1 or 1 to 32 bits

Query Syntax: SIZE?

Query Response: <size>

Description: The SIZE command sets the field size (in base 10) used in connection with the GETA, GETR, PUTA, and PUTR commands. If a negative size is specified, then values given (or returned) are sign extended to 32 bits. For example, if SIZE = -16 and a 16 bit field containing FFFF hex is fetched, then the value FFFFFFFF hex is returned. The SIZE? query returns the current setting of SIZE in base 10. SIZE is preset to 16 each time the generator is powered on.

Note – This command will normally be used only with custom applications and command files created by Quantum Data.

Other Required Cmds: *None*

Example:

```
Use only with code supplied by Quantum
Data!
```

SMOD — Sequence operating MODe

Classification: Sequence parameter setting

Command Syntax: SMOD

Limits: <mode>
0 = Disable
1 = Enable manual step mode that stops at last step
2 = Enable manual step mode that wraps to first step after last step
3 = Enable continuous auto stepping

Query Syntax: SMOD?

Query Response: <mode>

Description: The SMOD command sets the sequence mode. The SMOD? query returns the current setting of SMOD.

Other Required Cmds: The SEQL command loads the sequence and SEQU starts to run it.

Example:

```
SEQL MY_SEQ // Load "MY_SEQ" dir in
       // edit buffer
SEQU // Start running the sequence in
       // the buffer
SMOD 3 // Set the sequence mode
```

SNUM draw sequence Step NUMber

Classification: Custom image primitive

Command Syntax: SNUM <color> <x> <y>

Limits: <color> = available colors

<x> = positive integer number

<y> = positive integer number

Query Syntax: *None*

Description: Displays the sequence step number to an image when it's drawn as part of a test sequence. The number, along with the word *Step*, appears in a small box. *Seq. Step* uses three (3) parameters. The first is the color used for the text and box border. The next two parameters are the X and Y coordinates for the position of the box.

Other Required Cmds: The displaying of sequence step numbers must be enabled with the DNUM command and the custom image must have been loaded as part of a sequence in order for this primitive to be drawn.

Example:

```
SNUM white 50 50 // Display seq. num.
           // in box with top left corner at
           // X=50, Y=50
```

SSST — Select Sync Signal Type

Classification: Format parameter setting

Command Syntax: SSST <type>

Limits: <type>
- 0 = no sync
- 1 = digital separate horizontal & vertical sync (DHS & DVS)
- 2 = digital separate composite sync (DCS)
- 3 = analog composite sync (ACS)

Query Syntax: SSST?

Query Response: <type>

Description: The SSST command selects the type of sync signal that's used to synchronize the display. In general, any one of three different types of sync can be selected to synchronize the display. The availability of different sync types is specified using the ASCT, DSCT, and DSST commands. Some displays may not accept one (or more) types of sync. For example, a digital video monitor cannot accept analog composite sync because analog signal transmission is not used. Also, a PGA display cannot accept digital separate HS & VS because only one sync wire is provided in the cabling. In these cases, one (or more) of the sync types (ASCT, DSCT, or DSST) is set to zero (0) indicating that they are not supported. If a non-supported sync type is selected using the SSST command, then the corresponding sync outputs of the generator will remain disabled. The SSST? query returns the type of sync (if any) that's currently selected.

Other Required Cmds: The desired sync type select must not be set to *void* with the ASCT, DSCT, or DSST commands. To actually output the selected sync signal it must be gated on with the appropriate ASSG, ASBG, ASGG, ASRG, CSPG, HSPG and VSPG settings.

Examples:

```
DSST 1 // Set Amer. H&V in buffer
HSPG 1 // Enable H sync in buffer
VSPG 1 // Enable V sync in buffer
SSST 1 // Select sep H&V sync in
       // buffer
FMTU // Update hardware to current
     // buffer contents
```

STEP — sequence STEP number

Classification: Sequence parameter setting

Command Syntax: STEP <step#>

Limits: <step>
min = 1
max = See description

Query Syntax: STEP?

Query Response: <step>

Description: The STEP command selects a step in the sequence edit buffer. It is context sensitive. While editing a sequence (between SEQB and SEQE commands), the STEP command selects a step to be edited. Outside of the sequence editor and while running a sequence, the command selects a step to be executed.

When running a sequence, the maximum limit for the step number is the number of the last step in the sequence. When editing a sequence to be saved in memory, the maximum number of steps will depend upon the generator firmware version as well as how many other sequences are stored in sequence memory and how many steps they contain. Generators with firmware versions below 2.605 can store a total of about 400 sequence steps. Generators with firmware versions greater than or equal to 2.605 can store a total of about 1280 sequence steps. The STEP? query returns the current setting of STEP.

Other Required Cmds: When running a sequence, the SEQU command must be executed after the STEP command to load the format and display the image selected in the step.

Example:

```
STEP 5 // Select fifth step in current
       // sequence
SEQU   // Load format and draw image in
       // current step
```

TEXT draw TEXT string

Classification: Custom image primitive

Command Syntax: Text <color> <x> <y> <fontname> <"text ">

Limits: <color> = available colors

<x> = position of left edge of page in pixels

<y> = position of top edge of page in pixels

<fontname> = available fonts

<text> = approx. 30 characters (must be enclosed by quotes)

Query Syntax: *None*

Description: Draws a user-defined text string. It uses five (5) parameters. The first is color. The next two are the X and Y coordinates for the upper left corner of the starting position of the string. The fourth parameter selects the font. (Recall that only #0 Sys 16 and #1 OPIX 9 are full alphanumeric sets.) The last parameter is the text string. If the string is longer than one word, it must be contained inside quotation marks.

Other Required Cmds: The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
TEXT red 50 40 opix9 "HELLO WORLD"
     // Draw red "HELLO WORLD" at x=50
     // y=40 in opix9 font
```

TOBL set levels relative TO BLanking

Classification: Custom image primitive

Command Syntax: *None*

Query Syntax: *None*

Description: Temporarily changes how the signal levels are determined for a given color intensity level. The default method uses black as the 0% reference level and peak video as the 100% level. Inserting TOBL moves the 0% reference point to the blanking (blacker than black) pedestal level. The reference point remains shifted only for as long as the image is displayed.

Other Required Cmds: *None*

Example:

```
TOBL
```

TRIA draw a TRIAngle

Classification: Custom image primitive

Command Syntax: TRIA <color> < x1> <y1> <x2> <y2> <x3> <y3> <fill>

Limits: <color> = available colors

<x1> <y1> <x2> <y2> <x3> <y3> = positive integer number

<fill> = available fill pattern

Query Syntax: *None*

Description: Draws a triangle defined by its three endpoints. The primitive uses eight (8) parameters. The first is line color. The next three pairs of parameters are the X and Y coordinates for the three points. The last parameter is the fill. More complex filled polygons can be built up using a series of joined filled triangles.

Other Required Cmds: The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
TRIA red50 10 5 10 25 40 15 GrayPat50
     // Draw a red50 triangle at x=10,
     // y=5; x=10, y=25; x=40, y=15 with
     // 50% active pixel fill
ALLU // Update hardware to current
     // buffer contents
```

UIDN User IDeNtification

Classification: System parameter setting

Command Syntax: UIDN <text string>

Limits: <text string> ≤ 80 characters in length

Query Syntax: UIDN?

Query Response: <text string>

Description: The UIDN command sets up the text string that's placed in the upper portions of the *SMPTE133* and *Cubes* images. The command can be used to add your company's name or other identification to the images. The command will not change the text in a currently displayed image. It must be redrawn to use the new text. The factory default string is *Quantum Data*. The UIDN? query returns the current text string.

Other Required Cmds: The IMGU command redraws the last selected test image. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
UIDN "XYZ Monitor Mfg." // Change text
     // string
IMGL SMPTE133 // Select image that uses
     // string
IMGU // Draw the image
```

USIZ — Unit of measure used for physical SIZes

Classification: Format and System parameter setting

Command Syntax: USIZ <units>

Limits: <units>
- 0 = sizes not given (use default)
- 1 = inches
- 2 = mm

Query Syntax: USIZ?

Query Response: <units>

Description: The USIZ command sets the units of measure assumed by HSIZ and VSIZ commands to establish the physical size of the image that appears on the CRT (context sensitive - see FMTB and FMTE). The USIZ? query returns the current setting of USIZ.

Note – Changing the USIZ parameter between inches and millimeters will convert the current HSIZ and VSIZ values to match the new unit of measure. For example, if USIZ is in inches and the current HSIZ is 10 (inches), changing USIZ from inches to mm will change HSIZ to 25.4 (mm). The USIZ command should be sent before specifying physical sizes in format command files.

Other Required Cmds: The ALLU command updates hardware to the new setting and redraws the test image, taking the new units into account.

Example:

```
FMTB // Begin editing session
     // One or more format editing
     // commands
     // ...
USIZ 1 // Select inches as unit of
     // measure in buffer
HSIZ 10.4 // Set width to 10.4 in
     // buffer
VSIZ 7.8 // Set height to 7.8 in buffer
ALLU // Test the new settings
     // One or more format editing
     // commands
     // ...
FMTE // End editing session
```

VERF — VERsion of Firmware

Classification: Miscellaneous system parameter

Command Syntax: *None*

Query Syntax: VERF?

Query Response: <version>

Description: The VERF? query returns the firmware revision number.

Other Required Cmds: *None*

Example:

```
VERF? // Return firmware revision number
```

VERH — VERsion of Hardware

Classification: Miscellaneous system parameter

Command Syntax: *None*

Query Syntax: VERH?

Query Response: <version>

Description: The VERH? query returns a five digit hardware version number. The number indicates when a major modification to the hardware was last implemented. The significance of each digit is interpreted as follows:

31995

- Last 2 Digits of Year
- Day of the Month
- Month (1-9 = Jan. - Sept., A = Oct., B = Nov., C = Dec)

The version number in the example, 31995, would be interpreted as March 19, 1995.

Other Required Cmds: *None*

Example:

```
VERH? // Return hardware design revision
      // date
```

VRES — Vertical RESolution

Classification: Format parameter setting

Command Syntax: VRES <lines>

Limits: <lines>

min = 1 (when SCAN = 1) or
2 (when SCAN = 2)

max = the lesser of
1024 (for the 801GP, GC, GX or 801GC-ISA) or
2048 (for the 801GF or 801GF-ISA) or
VTOT-1 (when SCAN = 1) or
VTOT-3 (when SCAN = 2)

Must be an even number when SCAN = 2.

Query Syntax: VRES?

Query Response: <lines>

Description: The VRES command establishes the number of active lines per frame. The VRES? query returns the current setting of VRES.

Other Required Cmds: The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
VRES 480 // Set 480 active lines in
         // buffer
ALLU     // Configure hardware and redraw
         // image
```

VSIZ Vertical SIZe

Classification: Format parameter setting

Command Syntax: VSIZ <physical size>

Limits: <physical size> = positive value (floating point accepted)

Query Syntax: VSIZ?

Query Response: <physical size>

Description: The VSIZ command establishes the vertical physical size of the image on the display. Units expected (or returned) vary according to the last mode set with USIZ command. The VSIZ command is context sensitive and must appear between begin and end commands: FMTB and FMTE. The VSIZ? query returns the current setting of VSIZ.

Note – Make sure that the USIZ parameter is properly set before using the VSIZ command. Changing the USIZ setting after entering VSIZ will convert the size to match the new unit of measure.

Other Required Cmds: The units of measure must be properly set by USIZ before entering VSIZ. The ALLU command updates hardware to the new setting and redraws the test image, taking the new size into account.

Example:

```
FMTB // Begin editing session
     // One or more format editing
     // commands
     // ...
USIZ 1 // Select inches as unit of
     // measure in buffer
HSIZ 10.4 // Set width to 10.4 in
     // buffer
VSIZ 7.8 // Set height to 7.8 in buffer
ALLU // Test the new settings
     // One or more format editing
     // commands
     // ...
FMTE // End editing session
```

VSPD — Vertical Sync Pulse Delay

Classification: Format parameter setting

Command Syntax: VSPD <lines>

Limits: <lines>
min
0 (when SCAN = 1) or
1 (when SCAN = 2)
max
VTOT-VRES-VSPW (when SCAN = 1) or
[(VTOT-VRES-1) /2]-VSPW (when SCAN = 2)

Query Syntax: VSPD?

Query Response: <lines>

Description: The VSPD command establishes the delay between leading edge of blanking in the first (or even) field and the leading edge of the vertical sync pulse. When interlacing, delay between end of video and leading edge of vertical sync before second (or odd) field is 0.5 line shorter than the whole-line delay specified. The VSPD? query returns the current setting of the vertical sync pulse delay.

Other Required Cmds: The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
VSPD 11 // Set V sync delay to 11 lines
        // in buffer
FMTU    // Update hardware to current
        // buffer contents
```

VSPG — Vertical Sync Pulse Gate

Classification: Format parameter setting

Command Syntax: VSPG <mode>

Limits: <mode>
0 = OFF
1 = ON

Query Syntax: VSPG?

Query Response: <mode>

Description: The VSPG command enables and disables the digital vertical sync output. The VSPG? query returns the current mode of VSPG.

Other Required Cmds: In order to use digital vertical sync, digital separate H&V sync must be selected with the SSST command. The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
VSPG 1 // Enable V sync output in
       // buffer
FMTU   // Update hardware to current
       // buffer contents
```

VSPP Vertical Sync Pulse Polarity

Classification: Format parameter setting

Command Syntax: VSPP <polarity>

Limits: <polarity>
0 = active-low (negative going pulse)
1 = active-high (positive going pulse)

Query Syntax: VSPP?

Query Response: <mode>

Description: The VSPP command establishes the logic sense of the digital vertical sync outputs. Setting polarity to one (1) causes the leading edge of vertical sync to be a low-to-high transition. Setting polarity to zero (0) causes the leading edge of vertical sync to be a high-to-low transition. The VSPP? query returns the current polarity of VSPP.

Other Required Cmds: In order to use digital vertical sync, it must be gated on with the VSPG command and digital separate H&V sync must be selected with the SSST command. The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
VSPP 1 // Set active hi V sync in
       // buffer
VSPG 1 // Enable V sync output in
       // buffer
SSST 1 // Select H&V sync type in
       // buffer
FMTU // Update hardware to current
     // buffer contents
```

VSPW Vertical Sync Pulse Width

Classification: Format parameter setting

Command Syntax: VSPW <lines>

Limits: <lines>

Query Syntax: VSPW?

Query Response: <lines>

Description: The VSPW command establishes the width of the vertical sync pulse in lines. If the type specified for the selected sync signal (see SSST, ASCT, DSCT, or DSST commands) is one of the CCIR types, then the actual sync pulse width output by the generator will be 1/2 line shorter than the whole number specified. The VSPW? query returns the current setting of VSPW.

Other Required Cmds: The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
VSPW 3 // Set V sync width to 3 lines
       // in buffer
FMTU // Update hardware to current
     // buffer contents
```

VTOT — Vertical TOTal lines per frame

Classification: Format parameter setting

Command Syntax: VTOT <lines>

Limits: <lines> must be an odd number when SCAN = 2
min = 2
max
801GC-ISA = 4096 @ SCAN=1; 4097 @ SCAN=2
801GF-ISA = 4096 @ SCAN=1; 8191 @ SCAN=2
801GP = 2048
801GC, GX = 4096 @ SCAN=1; 4097 @ SCAN=2
801GF = 4096 @ SCAN=1; 8191 @ SCAN=2

Query Syntax: VTOT?

Query Response: <lines>

Description: The VTOT command establishes the total number of lines per frame. When interlacing (SCAN=2), VTOT must be odd. The VTOT? query returns the current setting of VTOT.

The frame or picture refresh rate is equal to the quotient of HRAT divided by VTOT.

The field or vertical rate is equal to the frame rate when SCAN = 1 (non-interlaced operation).

The field or vertical rate is equal to twice the frame rate when SCAN = 2 (non-interlaced operation).

Note – The current version of the firmware does not allow you to directly enter a specific field or frame rate when setting up a format. If your test specifications call for a specific field, frame or vertical refresh rate, enter suitable values for HRAT, SCAN and VTOT to give you the desired rate.

Other Required Cmds: The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
VTOT 525 // Set total lines//frame to
         // 525 in buffer
FMTU     // Update hardware to current
         // buffer contents
```

*WAI   WAIt for completion suffix

Classification: Direct processor control

Command Syntax: <command>; *WAI

Limits: <command>

FMTU, IMGU, ALLU, BOOT, INIT, or SCAL

Query Syntax: *None*

Description: Normally, the generator returns a prompt immediately after either an FMTU, IMGU, ALLU, BOOT, INIT, or SCAL command is received - even before these commands have finished executing. If the system controlling the generator must know when a command has finished executing, use a semicolon to append the suffix ... ;*WAI. This causes the generator to wait until all processes have been completed before sending the > prompt.

Other Required Cmds: The *WAI command is used as a suffix with the FMTU, IMGU, ALLU, BOOT, INIT and SCAL commands.

Example:

```
FMTL vga_m3 // Load a format from memory
     // to buffer
IMGL SMPTE133 // Load the SMPTE RP-133
     // image to buffer
ALLU;*WAI // Update hardware to current
     // buffer contents and delay
     // prompt until all done
```

XVSG   Video Signal Gating

Classification: Format parameter setting

Command Syntax: XVSG <red mode>, <green mode>, <blue mode>

Limits: < mode>
- 0 = OFF
- 1 = ON

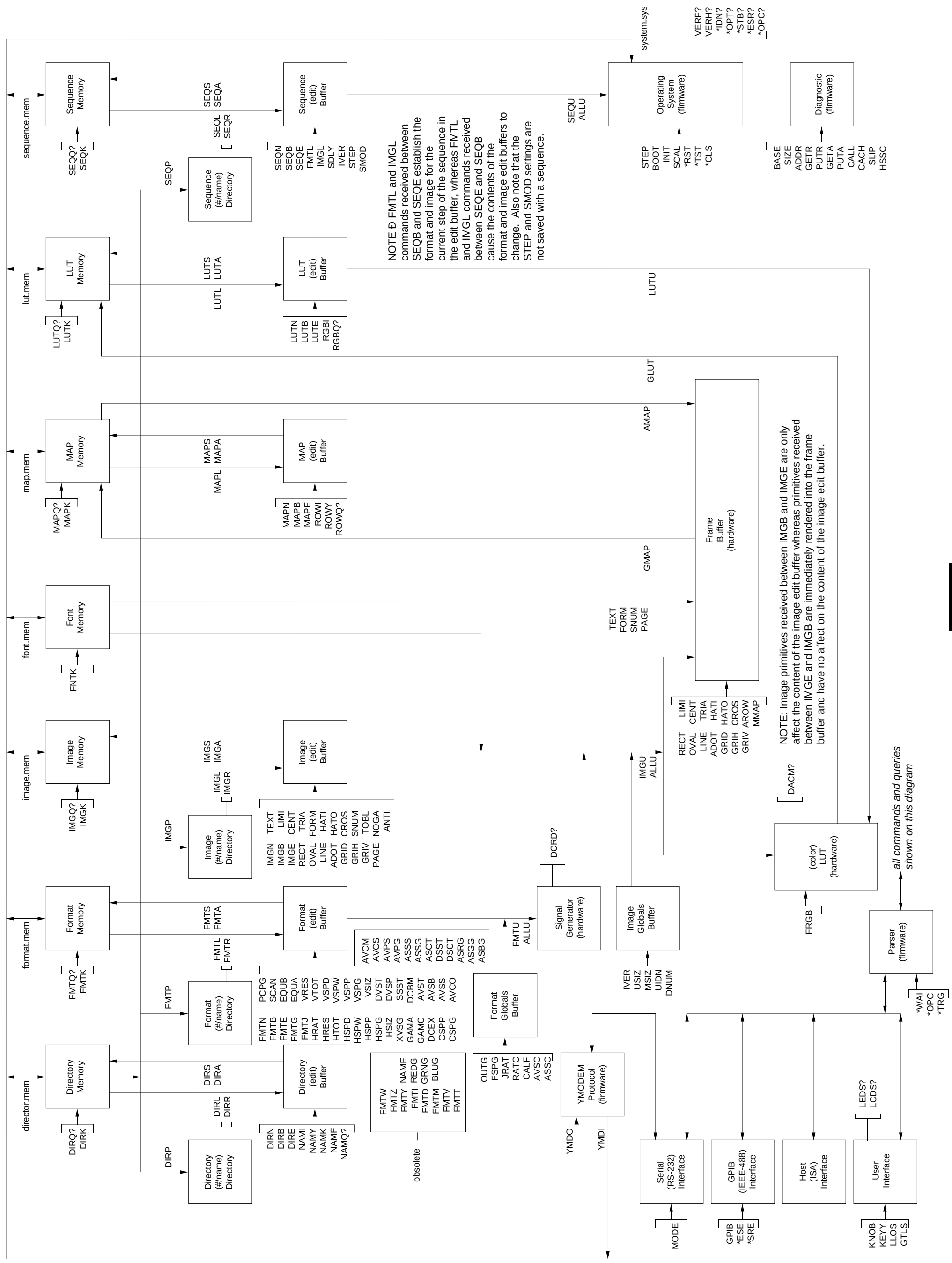
Description: The XVSG command determines which video outputs are active when the format is selected. The same command controls both the analog and digital video outputs.

Other Required Cmds: The FMTU command instructs the generator to use the new setting. The ALLU command updates hardware to the new setting and redraws the test image.

Example:

```
XVSG 1 1 1 // Enable all color channels
FMTU // Update hardware to current
     // buffer contents
```

801G Memory (Information) Flow Diagram



Notes

Power-on

Computer Interfacing

If all else fails

Chapter 7: Troubleshooting

Power-on

Normal power-on

Under normal conditions, the power light will come on as soon as the 801G* is turned on. The following message will be the first thing to appear in the LCD just after the generator is powered-up:

The top number indicates the operating firmware revision level, while the bottom number indicates the hardware revision level. Please note these numbers if you will be contacting Quantum Data with any questions or problems concerning the operation of the unit.

If no problems are encountered, the generator will output the format and test image that were active when the unit was last powered down.

Power-on message remains

If the last active image was animated, the LCD will continue to display the firmware and hardware revision levels until a button is pushed or a knob is turned. Otherwise, the display will change by itself after about two seconds.

The information that next appears will depend upon the operating mode that has been programmed into the 801G*. The left-hand example shown below is typical for the normal (default) operating mode of the 801G*. The example on the right shows a typical display in the test sequence mode.

Error message at power-on

- See Chapter 2 of this manual for information on using the 801G* in the normal operating mode.

- See the “Programming” chapter for information on setting-up and running the 801G* in the test sequence mode.

If any other messages appear, it will be because one or more problems were found during the generator’s power-on self test.

- See "System Error Messages" in Chapter 8.

Unit will not power-on

If the power light fails to come on when the power switch is turned on, first check your power connection and line voltage selection, then check the fuse.

- See "AC Line Connection" in Chapter 5.

Computer Interfacing

Information on troubleshooting problems related to the RS-232 and IEEE-488 ports can be found in the “Programming” chapter.

❖ Please note that leaving any unused equipment connected to the 801G*’s ports may cause the front panel display and controls to hang. If this should happen, disconnect the equipment and cycle the power on the 801G*.

If All Else Fails

It is possible for a user programming error of the system parameters to cause the 801G* to appear to be broken. The “Programming” chapter has information on checking and changing the settings of the parameters.

The firmware includes a way of restoring the 801G* to known factory default conditions. This procedure will reset all system settings to their nominal values and erase all user programmed signal formats, test sequences and custom images from memory. The format memory is then loaded with the factory defined formats as described in Chapter 3 of this manual. The image list is reset to all available images in the order given in Chapter 4 of this manual. To restore the 801G* to its factory default settings hold down both the “ACS” and “DSS” sync gate buttons while turning on the power. Release the buttons when the “Memory Blank” message appears on the LCD. The firmware will then take a few more seconds to reset the 801G*.

If reprogramming the system parameters or resetting the 801G* does not restore normal operation your unit may need to be repaired. Chapter 9 of this manual covers repairs that can be done by the user. The unit should be returned to Quantum Data or one of our authorized service centers for all other repairs. Chapter 10 has information on getting your 801G* repaired.

Please contact an applications engineer at Quantum Data if you have any questions or concerns about the operation of you generator.

Chapter 8: Error Messages

System Errors

This section discusses system error messages that may be displayed by the generator from time to time.

Power-on Self Test Messages

The integrity of the data stored in non-volatile memory is checked every time the 801GX is turned on. The following tests are performed:

1. The system configuration memory is checked. The memory remembers what the generator was doing when it was last powered down. The memory also stores all system wide parameters such as calibration factors and the light meter size. It is also checked to see if it is compatible with the current version of operating firmware. If everything is in order, then the generator goes on to step 2.

   Otherwise, a "System corrupt!" message is displayed for 2 seconds. This message is immediately followed by a "Memory blank" message, which is displayed for 2 seconds while the format memory is re-initialized. Memory initialization involves clearing the entire non-volatile memory and loading it with factory default data. Any and all data created or edited by the user will be wiped out. The calibration factors are restored to the original values used at the factory to test and calibrate the unit. The generator then continues with step 3.

   ❖ System corruption is very rare. It may be an indication of a bad memory backup battery or a damaged generator.

   ❖ If you update the firmware EPROM in your unit, your system will appear to be corrupt at the next power-on.

2. The remainder of non-volatile memory is checked to see if any data has been corrupted since the

last time the generator was used. This check includes each of the 300 format storage locations. All user created custom images and test sequences are also checked. If no problems are found, the generator goes on to step 3.

If one (or more) items have been corrupted, a message containing a tally of the defective items is displayed for 2 seconds. A message for two corrupted items is shown here. Please note that the checking routine does not identify the types of items (format, custom image or test sequence) that are corrupted. It is up to the user to check the contents of the individual items. Attempting to load a corrupted item will report and error for that item.

- ❖ A corrupted format can be repaired be either editing it or by downloading a new set of data from a computer.
- ❖ In most cases, a corrupted custom image or test sequence can not be repaired. This is due to the variable data structure used. You will need to recreate and save the data back into memory.
- ❖ Repeated corruption of the data in non-volatile memory usually indicates a weak battery. Battery replacement information can be found in the chapter on “Maintenance”.

3. A power-on message, indicating the current firmware and hardware revision levels, will appear for 2 seconds at the end of the self test procedure.

   - ❖ If an animated image is displayed at power-on, this message will not disappear until a button is pushed or a knob is turned.

Power Fail Message

Whenever the AC power line voltage dips below the minimum specified, this message is displayed and the generator's microprocessor is halted.

If you see this message during normal operation, it probably indicates that a power sag or short duration drop-out has occurred. To clear this condition, cycle the power going to the generator. If the power is good, the generator should resume normal operation. If you still have a problem, you may also want to check to see that the line voltage selector (next to the power inlet) is set correctly for the power being fed into the 801GX.

Format Errors

Under normal operating conditions, a properly programmed model 801GX generator should never report any Format error messages. If you do get get an error message while loading a format it is for one of two reasons. The generator distinguishes between invalid data, entered in one of the programming modes, and corrupted data caused by a memory failure. In either case, the generator will shut all of its outputs off. The outputs remain off until a valid format is loaded.

Invalid Data Error Messages

The example on the left shows a typical invalid data error message. The top line of the display shows the format's storage location number and name. The bottom line shows the error number. Error numbers will be in the range of of 2000 through 2999. Different numbers indicate what type invalid data or combination of data is causing the error.

The following pages explain the cause of each of the error numbers and offers suggestions on how to correct the problem.

Corrupt Data Error Messages

The example on the left shows a corupted data error message. The top line of the display shows the format's storage location number and name. The bottom line shows the error number. Error number 9450 identifies the cause of the error as corrupted data. This type of error either indicates the signs of a failing memory backup battery or a more serious hardware failure in the generator.

Corrupted formats can corrected by either editing them or by downloading a clean copy of the format from a computer. The "Programming" chapter explains both methods in detail.

System Error Message Summary

Problem found with the configuration data stored in battery backed memory. All 150 format storage locations will be cleared and the industry standard formats installed. The memory back up battery may need to be replaced.

All 150 Format storage locations are being erased. Industry standard formats are being installed in locations 1 thru 44.

One or more formats, stored in battery backed format memory, has (have) been corrupted. Edit bad format(s) or re-initialize the format memory to correct the problem. The memory back up battery may need to be replaced.

The selected format has been corrupted. Edit the format or re-initialize the format memory to correct the problem. The memory back up battery may need to be replaced.

AC line voltage is low or a power glitch has occurred. Check the line-voltage and selector-switch settings; then cycle the power.

Refer to the format error summary on the next page.

Format Error Messages

2030 **SCAN less than 1**

An invalid scan setting exists in the file.

- Reedit the scan setting.

2040 **SCAN greater than 2**

An invalid scan setting exists in the file.

- Reedit the scan setting.

2050 **Horizontal total too small**

The generator does not support a horizontal period (active + blanked) less than 144 pixels per horizontal scan line.

- Increase the horizontal period so that it’s more than 143 pixels.

2060 **Horizontal total too large**

The generator does not support a horizontal period (active + blanked) of more than 4096 pixels per horizontal scan line.

- Decrease the horizontal period so that it’s less than 4096 pixels.

2071 **Pixel rate too high**

The pixel rate exceeds the maximum rate supported by the currently selected generator model for the given video type selection.

- Make sure video type is correct.
- Decrease the pixel rate.
- Decrease the horizontal period (in pixels) for a given horizontal rate.

2072 **Pixel rate too high**

The pixel rate exceeds the maximum rate supported by the currently selected generator model for the given video type selection.

- Make sure video type is correct.
- Decrease the pixel rate.
- Decrease the horizontal period (in pixels) for a given horizontal rate.

2080 **Pixel rate too low**

The pixel rate is below the minimum rate supported by the generator.

- Increase the value of HTOT.
- Increase the value of HRAT.

2090 **Horizontal total not even**

The generator does not support an odd number of total pixels per scan line.

- Make the horizontal period (in pixels) an even number.

2091 **Horizontal total must be multiple of 4 at pixel rate**

The generator does not support horizontal pixel totals (active + blanking) that cannot be evenly divided by four (4) at the given pixel rate.

- Make the total horizontal pixels per line to be evenly divisible by 4.
- Decrease the pixel rate.

2130 **Horizontal active too small**

The number of active pixels per horizontal scan line is below the minimum supported by the generator.

- Increase the horizontal active pixels.

2140 **Horizontal blanking too small**

The number of blanked pixels per horizontal scan line is below the minimum supported by the generator.

- Increase the horizontal period (in pixels).
- Decrease horizontal active (in pixels).

2141 **Horizontal blanking too small**

The number of blanked pixels per horizontal scan line (HTOT - HRES) is below the minimum supported by the generator.

- Increase the horizontal period (in pixels).
- Decrease horizontal active (in pixels).

2150 **Horizontal blanking too small**

The number of blanked pixels per horizontal scan line (HTOT - HRES) is below the minimum supported by the generator.

- Increase the horizontal period (in pixels).
- Decrease horizontal active (in pixels).

2151 Horizontal total < horizontal res.

The number of active pixels per horizontal scan line is greater than the total number of pixels per line (hor. period).

- Increase the horizontal period (in pixels).
- Decrease horizontal active (in pixels).

2152 Horizontal active too large or Horizontal total too small

The generator does not support the given combination of the horizontal pixel total and the number of horizontal active pixels.

- Increase the number of blanked or horizontal total pixels per line.
- Decrease the number of active pixels per line.

2155 Horizontal blanking too small

The number of blanked pixels per horizontal scan line (HTOT - HRES) is below the minimum supported by the generator.

- Increase the horizontal period (in pixels).
- Decrease horizontal active (in pixels).

2180 Horizontal pulse width less than 1

The horizontal sync pulse must be at least 1 pixel wide.

- Increase the value of horizontal sync pulse width so that it's greater than zero.

2181 Horizontal sync pulse width must be even at current pixel rate

The generator does not support horizontal sync pulse widths (in pixels) that cannot be evenly divided by two (2) at the given pixel rate.

- Make the sync pulse width (in pixels) evenly divisible by 2.
- Decrease the pixel rate.

2190 Horizontal sync pulse width too small for HDTV

The generator does not support the given horizontal sync pulse width for HDTV video types.

- Increase the sync pulse width.
- Change the video signal type.

2191 Horizontal sync pulse width must be even for HDTV

The generator does not support horizontal sync pulse widths (in pixels) that cannot be evenly divided by two (2) for HDTV video types.

- Make the sync pulse width (in pixels) evenly divisible by 2.
- Change the video signal type.

2200 Horizontal pulse width too large

The generator does not support a horizontal sync pulse width greater than the number of blanked pixels per horizontal scan line.

- Decrease the horizontal pulse width.
- Increase the horizontal period (in pixels).
- Decrease the number of active pixels per line.

2201 Horizontal sync pulse delay must be even at current pixel rate

The generator does not support horizontal sync pulse delay (in pixels) that cannot be evenly divided by two (2) at the given pixel rate.

- Make the sync pulse width (in pixels) evenly divisible by 2.
- Decrease the pixel rate.

2230 Horizontal pulse delay extends sync beyond blanking

The generator does not support horizontal sync pulses with any portion of the pulse occurring outside of the horizontal blanking period.

- Decrease the horizontal pulse width.
- Decrease the horizontal pulse delay.
- Increase the horizontal period (in pixels).
- Decrease the number of active pixels per line.

2240 Vertical total too small

There must be at least a total of 2 horizontal scan lines per frame with non-interlaced mode.

- Increase the vertical period so that it’s greater than 1 line.

2250 Vertical total too small

There must be at least a total of 5 horizontal scan lines per frame with interlaced operation.

- Increase the vertical period so that it's greater than 4 lines.
- Switch to non-interlaced operation.

2270 Vertical total is even

There must be an odd numbered total of horizontal scan lines per frame with interlaced operation.

- Change the vertical period (in lines) to an even number.
- Switch to non-interlaced operation.

2280 Vertical total too large

The generator does not support the given vertical period.

- Decrease the vertical period to less than 4096 lines per frame if non-interlaced.
- Decrease the vertical period to less than 4097 lines per frame if interlaced.

2300 Vertical active too small

There must be at least 1 active horizontal scan line with non-interlaced operation.

- Increase vertical active so that it's greater than 1 line.

2310 Vertical blanking too small

There must be at least 1 line of vertical blanking with non-interlaced operation.

- Increase the vertical period.
- Decrease vertical active.

2320 Vertical active too small

There must be at least 2 active horizontal scan lines with interlaced operation.

- Increase vertical active so that it's greater than 2 lines.
- Switch to non-interlaced operation.

2321 Vertical active must be even when in interlaced scan mode

The generator does not support vertical active periods (in lines) that cannot be evenly divided by two (2) when interlaced operation is selected.

- Make the vertical active period (in lines) evenly divisible by 2.
- Change to progressive (non-interlaced) scan mode.

2330 Vertical blanking too small

There must be at least 3 lines of vertical blanking with interlaced operation.

- Increase the vertical period.
- Decrease vertical active.
- Switch to non-interlaced operation.

2350 Vertical pulse too small

The generator does not support a vertical sync pulse width value of less than one scan line in duration. The firmware makes the pulse a half-line shorter than the entered value when European type sync is selected.

- Increase the vertical sync pulse width.

2370 Vertical pulse too large

The generator does not support a vertical sync pulse width greater than the number of blanked scan lines per frame with non-interlaced operation.

- Decrease the value of vertical sync pulse width.
- Increase the value of vertical period (in lines).
- Decrease vertical active.

2390 Vertical pulse too large

The generator does not support a vertical sync pulse width greater than the least number of blanked scan lines ((Vper - Vact - 1) ÷ 2) between fields with interlaced operation.

- Decrease the value of vertical sync pulse width.
- Increase the value of vertical period (in lines).
- Decrease vertical active.

2391 Incompatible ACS & DCS types

The analog composite sync and digital composite sync settings must be compatible when analog and digital composite sync are both active.

- Make the ACS and DCS settings the same.
- Change the Sync select setting.

2392 Incompatible ACS & DSS types

The analog composite sync and digital separate sync settings must be compatible when analog and digital composite sync are both active.

- Change the ACS and / or DSS settings.
- Change the Sync select setting.

2393 AVST requires ASCT to be 3 or 8

ACS setting not compatible with TV setting for video kind.

- Change ACS to “American w/serr & EQ” or “European w/serr & EQ”.
- Change Video kind to non-TV setting.

2394 ACS available on green only

The 801GP supports adding composite sync only to the green analog video channel.

- Change the ACS On setting "- - -" or "- G -."

2395 AVST requires SSST to be 3-7

Analog composite sync must be active when video kind is set to TV.

- Change Sync select to “ACS,” “ACS & DSS,” “ACS & DCS” or “ACS & DSS & DCS.”
- Change Video kind to non-TV setting.

2396 Invalid ACS type

An invalid analog composite sync type selection exists in the file.

- Reedit the analog composite sync type setting.

2397 Invalid DCS type

An invalid digital composite sync type selection exists in the file.

- Reedit the digital composite sync type setting.

2398 Invalid DSS type

An invalid digital separate sync type selection exists in the file.

- Reedit the digital separate sync type setting.

2399 Invalid sync type

An invalid sync select exists in the file.

- Reedit the sync select setting.

2400 Digital separate sync type selection not compatible with analog and/or digital sync type

The generator does not support the given combination of digital separate sync (H&V) selection with one or both of the analog or digital composite sync type selections.

- Select only one sync type.

2430 Vertical pulse delay extends sync beyond blanking

The generator does not support vertical sync pulses with any portion of the pulse occurring outside of the vertical blanking period.

- Decrease the vertical sync pulse width.
- Decrease the vertical sync pulse delay.
- Increase the vertical period (in lines).
- Decrease the vertical active (in lines).

2450 Vertical pulse delay extends sync beyond blanking

The generator does not support vertical sync pulses with any portion of the pulse occurring outside of either vertical blanking period with interlaced operation.

- Decrease the vertical sync pulse width.
- Decrease the vertical sync pulse delay.
- Increase the vertical period (in lines).
- Decrease the vertical active (in lines).

2490 EQ before too large

The interval during which pre-equalization pulses occur cannot be greater than the vertical sync pulse delay.

- Decrease EQ Before.
- Increase the vertical sync pulse delay.

2495 EQ after too large

The interval during which post-equalization pulses occur cannot be greater than the interval between the end of the vertical sync pulse and the end of the shorter vertical blanking interval between fields with interlaced operation.

- Decrease EQ After.
- Decrease the vertical sync pulse width.
- Decrease vertical sync pulse delay.
- Increase the vertical period (in lines).
- Decrease the vertical active.

2496 EQ after too large

The interval during which post-equalization pulses occur cannot be greater than the interval between the end of the vertical sync pulse and the end of the vertical blanking.

- Decrease EQ After.
- Decrease the vertical sync pulse width.
- Decrease vertical sync pulse delay.
- Increase the vertical period (in lines).
- Decrease the vertical active.

2550 Not enough video memory

The generator does not have enough memory space to store the total number of active pixels required for one picture.

- Decrease the vertical active.
- Decrease the horizontal active.

2704 Horizontal physical size too small

The physical horizontal size must be greater than zero.

- Change the horizontal physical size to a positive, nonzero, value.

2705 Invalid physical size units

An invalid physical size unit of measure exists in the file.

- Reedit both physical sizes.

2706 **Vertical physical size too small**

The physical vertical size must be greater than zero.

- Change the vertical physical size to a positive, nonzero value.

2714 **Pedestal swing out of range**

The analog video blanking pedestal cannot be less than 0 IRE (%) or more than 100 IRE (%).

- Change the value of Pedestal level to be greater than or equal to 0 and less than or equal to 100 IRE (%).

2715 **Gamma correction out of range**

The firmware does not support gamma correction factors that are less than 0.1 or greater than 10.0.

- Change the value of Gamma to be greater than or equal to 0.1 and less than or equal to 10.0.

2716 **Analog video swing out of range**

The hardware does not support peak-to-peak analog video swings that are less than 0 or greater than 1.000 volts. The swing does not include any composite sync levels that may be added.

- Change the Video swing to be greater than or equal to 0 and less than or equal to 1.000 volts.

2717 **Sync swing out of range**

The hardware does not support peak-to-peak analog composite sync swings that are less than 0 or greater than 0.400 volts.

- Change the Sync swing to be greater than or equal to 0 and less than or equal to 0.400 volts.

2719 **Video swing calibration out of range**

One or more of the analog video swing calibration factors are out of range. These parameters are not actually part of a Format that is stored in memory. They are system level parameters that affect how all analog Formats are read from memory and into the hardware. The parameter settings are maintained while the 801GP is turned off.

- Re-initialize your generator's memory.
- Contact Quantum Data for further assistance.

2720 **Sync swing calibration out of range**

One or more of the analog sync swing calibration factors are out of range. These parameters are not actually part of a Format that is stored in memory. They are system level parameters that affect how all analog Formats are read from memory and into the hardware. The parameter settings are maintained while the 801GP is turned off.

- Re-initialize your generator's memory.
- Contact Quantum Data for further assistance.

2721 **NTSC requires pedestal** (801GX Only)

2722 **NTSC pedestal out of range** (801GX Only)

2741 **Digital video type selection not supported**

The generator does not support the given digital video signal type. Note – the 801GF, 801GC-ISA and 801GF-ISA generators do not have digital video outputs.

- Select a valid digital video signal type.

2742 **Invalid digital video type**

An invalid digital video type exists in the file.

- Edit the Video kind setting.

2743 **Digital video polarity must be positive**

An invalid digital video polarity setting exists in the file.

- The file cannot be corrected with the editor. Create a new file.

2745 **Cannot have analog and digital video**

An invalid digital video type exists in the file.

- Edit the Video kind setting.

2747 **Analog video signal type invalid**

An invalid analog video type exists in the file.

- Edit the Video kind setting.

3000 Invalid color name

A command file containing an invalid color name selection was downloaded to the generator.

- Check for spelling errors of the color names used in the command file.
- Use only color names that appear in the color list.

3001 Invalid fill pattern name

A command file containing an invalid fill pattern name selection was downloaded to the generator.

- Check for spelling errors of the fill pattern names used in the command file.
- Use only fill pattern names that appear in the fill pattern list.

3002 No image memory

This occurs during IMGN command and is caused by an insufficient amount of managed memory for the buffer request. This may be due to lack of memory or fragmented memory.

- Save all used edit buffers and issue FMTN, IMGN, and/or DIRN. This clears extra memory being used by these buffers.

3003 Image memory full

There is not enough free memory to save the image that is in the edit buffer.

- Reduce the number of primitives in the image being edited.
- Delete one or more custom images currently stored in memory. You will need to abandon your current edited image to do this.

3004 Invalid font name

A command file containing an invalid font name selection was downloaded to the generator.

- Check for spelling errors of the font names used in the command file.
- Use only font names that appear in the font list.

3005 Img ed running

This occurs during IMGB, IMGN, or IMGE commands. It happens when you are creating a custom image with the Custom Image Editor on a stand-alone generator and, during that session, you begin an image editing session via the IMGB or IMGN commands.

- Do not use the IMGB or IMGN commands while using GUI.

3006 Nothing to save

This occurs during IMGA or IMGS commands. It happens when you try to save the contents of the image editing register when none is initialized.

- You must use IMGN before IMGS or IMGA.

3010 Invalid Lookup Table (LUT) level (8-bit DAC)

The 801GX, 801GC, 801GC-ISA and 801GP generators do not support analog video DAC settings beyond the range of 0 through 255.

- The levels used with the FRGB command must be between 0 and 255 when used on the generator models listed above.

3015 Invalid Lookup Table (LUT) level (10-bit DAC)

The 801GF and 801GF-ISA generators do not support analog video DAC settings beyond the range of 0 through 1023.

- The levels used with the FRGB command must be between 0 and 1023 when used on the generator models listed above.

3050 Invalid gray level

The generator does not support gray level percentages beyond the range of 0 through 100. This error should never occur under normal operation.

- Please contact Quantum Data's technical support department if you get this message.

4002 **No sequence memory**

This occurs during the SEQN command. It happens when there’s an insufficient amount of managed memory for the buffer request.

- Save all used edit buffers and issue FMTN, IMGN, and / or DIRN. This clears extra memory used by these buffers.

4003 **Sequence memory full**

There is not enough free memory to save the sequence that is in the edit buffer.

- Reduce the number of steps in the sequence being edited.
- Delete one or more sequences currently stored in memory. You will need to abandon your current edited sequence to do this.

4005 **Seq ed running**

This occurs during SEQB, SEQN, or SEQE commands. It happens when you have used the Sequence Editor and then try to use the SEQB or SEQN command.

- Exit the generator's GUI sequence editor before editing a sequence using commands via the communications port.

4006 **No sequence to save**

This occurs during SEQA or SEQS commands. It happens when you try to save the contents of the sequence editing register and none is initialized.

- You must use SEQN before SEQS or SEQA.

4007 **Sequence buffer full**

No more steps can be added to the current sequence being edited.

- Reduce the number of steps in the sequence.
- Break up the testing into two or more separate sequences.

4008 Invalid delay time in sequence

A command file containing a sequence step delay was downloaded to the generator.

- Only use delay times (in seconds) that correspond to the list of available delays.

4010 Font index out of range

You tried to use a font at an invalid index.

4020 No font present at given index

You tried to access a font at a valid index but the specified index is empty because of a previous delete.

4030 Invalid font location

You tried to install a font that has not been transferred to memory.

4576 Can't convert to inches, not valid units

The format unit (inches or millimeters) is unknown because a new format was not initialized before it was created. Conversion between the format's unit and the user's preference fails.

- Always use the FMTN command before editing a new format.

4579 Can't convert to mm, not valid units

The format unit (inches or millimeters) is unknown because a new format was not initialized before it was created. Conversion between the format's unit and the user's preference fails.

- Always use the FMTN command before editing a new format.

5002 No directory memory

This occurs during DIRN and NAMI commands when there's an insufficient amount of managed memory for the buffer request.

- Save all used edit buffers and issue FMTN, IMGN, and/or DIRN. This clears extra memory being used by these buffers.

5003 **Directory memory full**

This occurs during DIRA and DIRS commands when there's insufficient room in the directory memory pool for the requested save.

- Use DIRK to delete one or more directories from the directory pool.

5006 **No directory to save**

This occurs during DIRA and DIRS commands when you try to save a nonexistent directory.

- A directory must reside in the edit buffer which is created with DIRN or DIRL.

5009 **Directory list full**

This occurs during DIRA and DIRS commands when you try to save more than 10 directories.

- Use DIRK to delete one or more directories.

8450 **Can't remove, font not present**

This occurs during FNTK command when you try to delete a nonexistent font.

8455 **Can't remove, ROM font**

This occurs during FNTK command when you try to delete a built-in font. Built-in fonts are sys16, focus12, focusmac, opix9, memesony, kanjikan, focusat5, focusat6, focusat7, focusat8, and memeplus.

- Don't try to remove built-in fonts.

8460 **Font already exists in memory**

You tried to transmit a font to the generator which already contained a font with the same name.

- Change the name of the font to be transmitted.

9450 **Invalid format location**

This occurs during FMTR command when the integrity of the data stored in the specified format memory location is corrupt.

- Delete the format from nonvolatile RAM using the YANK command. Then recreate and save the format. A corrupted format in EPROM requires at least the replacement of the firmware EPROM set.

9451 **Bad location for format, failed verify**

This occurs during FMTV command and FMTV? query when you attempt to verify data integrity at an invalid location.

- Use a range of -1 through -nn for permanent formats in EPROM or 1 through 300 for formats saved in nonvolatile RAM.

9452 **Bad location for format verify**

This occurs during a FMTZ? query when you try to determine if an invalid location has been erased.

- Use a range of 1 through 300 for formats saved in nonvolatile RAM.

9453 **Can't change EPROM contents**

This occurs during FMTW or FMTZ commands when you try to overwrite or zero out an EPROM format.

- Use a range of 1 through 300 for formats saved in nonvolatile RAM.

9454 **Bad location for format erase**

This occurs during FMTE command when you try to access an invalid format memory location.

- Use a range of 1 through 300 for formats saved in nonvolatile RAM.

9456 **Bad location for format read/write**

This occurs during FMTR or FMTW commands when you try to read or write to an invalid format memory location.

- Use a range of -1 through -nn for permanent formats in EPROM or 1 through 300 for formats saved in nonvolatile RAM. Use a range of from 1 through 300 to save formats in nonvolatile RAM with the FMTW command.

9457 Bad location for format name read

This occurs during FMTR? query when you try to read or write to an invalid format memory location.

- Use a range of -1 through -nn for permanent formats in EPROM or 1 through 300 for formats saved in nonvolatile RAM with the FMTR? query.

9458 Bad location for format copy

This occurs during FMTD command when you try during copying to access an invalid format memory location.

- Use a range of -1 through -nn for permanent formats in EPROM or 1 through 300 for formats saved in nonvolatile RAM as the source location. Use a range of 1 through 300 as the destination location to save formats in nonvolatile RAM.

9459 Can't change EPROM contents

This occurs during FMTD command when you try during copying to overwrite an EPROM format.

- Use a range of -1 through -nn for permanent formats in EPROM or 1 through 300 for formats saved in nonvolatile RAM.

9460 Bad location for format duplicate

This occurs during FMTD and FMTI commands when you try to use one or more invalid format memory locations as the command arguments.

- Use a range of -1 through -nn for permanent formats in EPROM or 1 through 300 for formats saved in nonvolatile RAM as the source location. Use a range of 1 through 300 as the destination location to save formats in nonvolatile RAM.

9467 Bad location for format erase

This occurs during FMTZ command when you try to make one or more invalid format memory locations as the command arguments.

- All formats to be erased from nonvolatile RAM must be in the range of 1 through 300.

9470 Can’t change EPROM contents

An attempt was made to insert a format using the FMTI command into an EPROM format memory location.

- Use a range of 1 through 300 as the insertion location for formats in nonvolatile RAM.

9471 Bad location for format yank

This occurs when you try to remove (yank) one or more formats from invalid format memory locations using the FMTY command.

- All formats to be removed from nonvolatile RAM must be in the range of memory locations 1 through 300.

9472 Can’t change EPROM contents

This occurs when you try to remove (yank) one or more formats from EPROM format memory locations using the FMTY command.

- Only formats in memory locations 1 through 300 can be removed.

9475 Can’t change EPROM contents

One or more EPROM format locations was given as the destination location for the FMTD command.

- Use a range of 1 through 300 as the destiny location to copy formats to nonvolatile RAM.

9477 Error duplicating formats

The <first> memory location parameter used with the FMTD command is greater than the <last> memory location parameter.

- Make the <first> memory location parameter less than or equal to the <last> memory location parameter.

9480 Format data missing

The FMTR command tried to read a format from an empty format storage location.

- Use a memory location known to contain a format or use the FMTL command to load a format by name.

9500 Sequence data not found

- Use a memory location known to contain a format or use the FMTL command to load a format by name.

Chapter 9: Service

New Product Warranty

Quantum Data products are warranted against defects in materials and workmanship. This warranty applies for one (1) year from the date of delivery. Quantum Data will, at its option, repair or replace equipment which proves to be defective during the warranty period. This warranty includes labor and parts. All repairs under this warranty must be performed by either Quantum Data or by an authorized service center. Equipment returned to either Quantum Data or to an authorized service center for repair must be shipped freight prepaid. Repairs necessitated by misuse of the equipment are not covered by this warranty.

NO OTHER WARRANTY IS EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. QUANTUM DATA SHALL NOT BE LIABLE FOR CONSEQUENTIAL DAMAGES.

Please contact our customer service department or an authorized Quantum Data service center to obtain an RMA (Return Material Authorization) number and instructions prior to returning any equipment. They will need to know the serial number of the unit.

User Registration Card

Please be sure to fill out and send in the registration card included with your 801G* Video Generator. This will assure you, the end user, of receiving the latest update notices, service bulletins, application notes and newsletters with the least amount of delay.

❖ You do not need to return the registration card to make your new product warranty valid.

Product Updates

Quantum Data reserves the right, at any time and without notice, to change the design or specifications of the hardware, firmware and software of any products.

Product update notices are usually printed in our newsletter. The newsletter is sent to all registered users of Quantum Data products. In other cases, the notices will be sent as a separate mailing to all registered users.

The cost of upgrades depends on the type of upgrade involved and on whether or not a particular unit is still covered by a new product warranty or a service agreement.

Service Agreements

It is the intent of Quantum Data to provide its customers with the highest quality service both before and after the sale. The 801G* has been designed for years of dependable operation. However, like any complex piece of equipment, a scheduled plan of inspection and calibration checks will insure that your unit will be operating within all specifications.

In the U.S.A.

For our customers in the United States, FULL Service Support Agreements beyond the standard warranty period may be purchased. This Service Support program covers repairs, performance testing and hardware updates as well as telephone assistance. Service support agreements that include the use of service loaners while covered units are in for repairs are also available.

Please contact Quantum Data's customer service department or your sales representative for information concerning the exact terms and conditions of service agreements.

In Europe

For our customers in Europe, service contracts may be available directly from our authorized service center in The Netherlands. Please contact Matrix Europe BV for information concerning the exact terms and conditions of their service contracts.

Getting Your 801G* Repaired

We do not recommend most types of repairs be attempted by the user. We do not currently publish schematic diagrams or user service manuals for the 801G*.

Returning the 801G*

All repair work should be done by Quantum Data or by one of our authorized service centers. They are listed on the next two pages. Please contact our customer service department or the authorized service center for an RMA (Return Material Authorization) Number and instructions prior to returning any equipment. You will need to tell them the serial number of the unit being returned. Equipment returned to either Quantum Data or to an authorized service center for repair must be shipped freight prepaid.

❖ Keep a record of all formats, custom images, image lists and test-sequences that you have created or edited. Units sent in for service will probably have their storage locations reset to the factory default settings. You will need to re-enter the data when the unit is returned.

Authorized Service Centers

The following list of our sales offices and authorized service centers is correct as of January, 1996. Please contact the appropriate facility to obtain a Return Materials Authorization (RMA) number before sending any equipment in for repair or calibration. Please contact our facility in the U.S.A. if you have any problems in contacting or dealing with any of the other locations shown.

North America: **Manufacturing, Sales and Service**

Quantum Data Inc.
Attn: Customer Service Dept.
2111 Big Timber Road
Elgin, IL 60123 U.S.A.

Telephone: +1 - 847-888-0450
Facsimile: +1 - 847-888-2802
Internet e-mail:
sales@quantumdata.com

Europe: **Service Only**

(Please contact our U.S.A. office, above, for sales)

Matrix Europe BV
Attn: Service Department Manager

Mail Address:	Street Address:
P.O. Box 992	Nijverheidsweg Zuid 4
NL-3800 AZ Amersfoort	3812 EB Amersfoort
The Netherlands	The Netherlands

Telephone: +31 - 33-4620410
Facsimile: +31 - 33-4633644
Telex: 70387 matrix nl

Japan: **Sales Representative and Service**

Nihon Binary Company, Ltd.

Minamizuka-Bldg. 2-17-3
Shibuya, Shibuya-Ku
Tokyo 150, Japan

Telephone: +81 - 33-407-9751
Facsimile: +81 - 33-407-9752
Telex: BINARY J27876

Korea: **Sales Representative and Service**

B & P International Co., Ltd.
Attn: Service Department Manager

Room 1809, Geopyoung Town #A
203-1 Nonhyun-Dong
Kangnam-Ku
Seoul, Korea

Telephone: +82 2-546-1457
Facsimile: +82 2-546-1458
Telex: K29230 MUSESUH

Singapore: **Sales Representative and Service**

Test Systems Integration (TSI)

Block 6024
Ang Mo Kia Industrial Park 3 #03-08
Singapore 2056

Telephone: +65 - 481-1346
Facsimile: +65 - 481-9506

Taiwan: **Sales Representative and Service**

SuperLink Technology Corporation

339 Ho Ping Road, Section 2, 9th Floor
Taipei, Taiwan ROC

Telephone: +886 2-705-7090
Facsimile: +886 2-708-3398

Appendix A: Specifications

Model 801GC, 801GF and 801GX Specifications

Signal Formats

Over 100 built-in formats

All models share a common library of built-in formats. Not all formats may work on all models of generators. See chapter 4 for more information.

TV encoded color video (801GX only):

NTSC:	5
PAL:	5

HDTV analog video:

European:	3
Japanese:	3

Computer analog video:

PAL:	3
RS-170:	3
Barco:	6
HP:	7
IBM:	16
Intergraph:	4
Mac:	13
NEC-PC:	2
Sony:	3
Sun:	10
VESA:	33

Computer digital video (801GC & 801GX only):

AT&T:	3
IBM:	7

Custom Formats

Storage capacity: 300 + built-in formats
Edit method: screen editor on UUT or via commands over computer ports

Custom Sequences

Storage capacity: total steps ≈ 2133 - # of sequences
Edit method: Screen editor on UUT or via commands over computer ports
Parameters:
- Select Format
- Select Image & version
- Auto step delay: 0.1 Sec. to 24 Hr.
- Sequence name: 8 characters

Test Images

Over 100 built-in imagess

All models share a common library of built-in images. Not all images may work on all models of generators. See chapter 5 for more information.

Partial Listing:

Format	Format spreadsheet
ColorBar	Color bars, up to 16 bars
GrayBar	Gray bars, up to 16 bars
Raster	Active pixels set to black
BrightBox	Five light meter boxes
Dot_10	Hatch/dots, small
Dot_12	Hatch/dots, medium
Dot_24	Hatch/dots, large
Hatch_10	Crosshatch, 10 boxes
Hatch_12	Crosshatch, 12 boxes
Hatch_24	Crosshatch, 24 boxes
Grill_44	Vertical stripes, 4 pixels
Grill_44(I)	Horiz stripes, 4 pixels
Grill_33	Vertical stripes, 3 pixels
Grill_33(I)	Horiz stripes, 3 pixels
Grill_22	Vertical stripes, 2 pixels
Grill_22(I)	Horiz stripes, 2 pixels
Grill_11	Vertical stripes, 1 pixel
Grill_11(I)	Horiz stripes, 1 pixel
Linearity	Xhatch with tics, circles
SMPTE133	Medical test image
Focus_Cx	Full screen "C" w/ "x"
Focus_H	Full screen of "H"
Focus_Oo	Full screen of "O" and "o"
Text_9	Random small text
Text_16	Random larger text
Cubes	Animated rotating cubes
Persist	17 moving boxes
Flat	All active pixels white
Outline	White border around active area
SMPTEBar	TV vectorscope
Stairs20	TV 20% gray levels
PulseBar	TV waveshape
Burst_100	TV 100% multiburst
MEME	Full screen of ÒMEÓ for focus

Custom Images

Storage capacity:	Total # of primitives ≈ 2048 - # of images
Edit method:	Screen editor on UUT or via commands over computer ports
Edit functions:	exit, draw, insert, delete, move, cur sor, save, change
Primitives:	dot, line, rectangle, oval, triangle, characters, text string, format data, grid, croshattches, H-grill, V-grill, centermark, limits
Colors:	Up to 15 colors (+black) from a palette of 63 colors and grays

Model 801GC, 801GF and 801GX Specifications

User Interface

Displays:	16 X 2 character LCD LED power indicator
Knobs:	Format selector Image selector
Buttons:	Image / STEP Video gate(R, G/I, B/V) Sync gate (ACS, DCS and DSS) Outputs on/ off

Output Connectors

GENERAL PURPOSE

Connector: 5 BNC receptacles
Signals: analog R,G,B; digital HS/CS, & VS/CS

VGA

Connector: HD female 15-pin D-Sub
Signals: analog R,G,B; digital HS/CS, & VS/CS

SUN WORKSTATION

Connector: Female 13W3 D-Sub
Signals: analog R,G,B; digital HS,VS, & CS

MAC

Connector: Female 15-pin D-Sub
Signals: analog R,G,B; digital HS, VS, & CS

MDA, CGA & EGA (801GC & 801GX only)

Connector: Female 9-pin D-Sub
Signals: digital VI (MDA); digital RGB
digital RGBI (CGA)
digital RrGgBb (EGA)
digital HS & VS sync

NTSC-TV / PAL-TV (801GX Only)

Connector: BNC receptacle
Signals: analog E (composite)

S-VIDEO (801GX Only)

Connector: 4-pin Mini-DIN receptacle
Signals: Y(luminance); C (chrominance)

Computer Ports

IEEE-488 INTERFACE

Protocol: IEEE-488.2
Connector: 24 pos. microribbon

SERIAL INTERFACE

Type: RS-232C
Protocol:
Baud Rates: 300 thru 38,400
full duplex
Data Bits: 7, 8
Stop Bits: 1, 2
Parity Bits: none, odd, even
Handshake: none, RTS/CTS
Power-On Default Settings:
2400 Baud, 8 Data bits, 1 Stop bit,
No parity, No handshake

Connector: 9 pin D-Sub receptacle
Pin 2 Ñ Data in (Rx)
Pin 3 Ñ Data out (Tx)
Pin 4 Ñ DTR in
Pin 5 Ñ Signals ground
Pin 7 Ñ RTS out
Pin 8 Ñ CTS in
Pins 1, 6, 9 Ñ No connections

Supplied Software

MS-DOS¨ compatible software and files on 3-1/2 inch HD floppy. Disk contains example programs for communicating with the generator via its serial port. Sample command files also included.

Misc.

AC MAINS

Voltage: 86-132VAC (standard as shipped)
180-250 VAC (via switch on cabinet)
Frequency: 48 to 66 Hz
Power: 26 watts (801GC & 801GX)
28 watts (801 GF)
Fuse: 1/2 Amp @ 250 VAC slow-blow, 5 mm dia x 20 mm long, type 239.500

SIZE

Unpacked:
Inches: ≈12.2 W x ≈7.7 D x ≈3.8 H
mm: ≈31 W x ≈20 D x ≈10 H
Shipping Box: 17x11.5x7 in. (43x29x18 mm.)

Weight

Unpacked: ≈ 7 lbs. (3.2 kg.)
Shipping: ≈ 12 lbs. (5.5 kg.) w/ all standard accessories

Additional Model 801GC and 801GX Specifications

Pixel Timing

Frequency Range (MHz):
9.375 to 150 (RGB analog)
9.375 to 18 (801GX only TV, S-VIDEO)
9.375 to 55 (TTL digital video)
Step: 1.465 Hz
Jitter: ≤800ps line-to-line (1 sigma)
Accuracy: 50 ppm

Horizontal Timing

Frequency Range: 2.29 Ð 130KHz

Total pixels per line:
Range: 144 to 4096 pixels (must be even below 125 MHZ pixel clock rates and must be evenly divisible by 4 above 125 MHz)
Step: 2 pixels below 125 MHz pixel clock
4 pixels above 125 MHz pixel clock

Active pixels per line:
Range: 16 to 2048 (limit Htotal - 32 pixels) (RGB analog and TTL)
16 to 1024 pxls (TV and S-Video)
Step: 1 pixel below 125 MHz pixel clock
2 pixels above 125 MHz pixel clock

Sync pulse width:
Range: 1 to Htot - Hact -HSdel pixels
Step: 1 pixel below 125 MHz pixel clock
2 pixels above 125 MHz pixel clock

Sync delay (front porch):
Range: 1 to Htot - Hact -HSwid pixels
Step: 1 pixel below 125 MHz pixel clock
2 pixels above 125 MHz pielx clock

Vertical Timing

Frequency Range: 1 to 650Hz
Vertical total scan lines per frame:
Range: 2 to 4096 lines (progressive)
5 to 4097 lines (interlace)
Step: 1 line (progressive)
2 lines (interlace)
Vertical active scan lines / frame:
Range: 1-1024 lines (Vtotal - 1)
Step: 1 line (progressive)
2 lines (interlace)
Vertical Sync pulse width:
Range: 1 to Vtot - Vact -VSdel lines
Step: 1 line
Modes: American or European
Vertcal Sync delay (front porch):
Range: 0 to Vtot - Vact -VSwid lines
Step: 1 line

Digital Sync

Modes: Separate Horiz. & Vert.
Composite
Composite Configurations:
American HS OR'ed with VS
American Serrated
American Serr. &Eq.(interlaced only)
European HS OR'ed with VS
European Serrated
European Serr. & Eq.(interlaced only)
Eq. Pulse Width: HSwidth / 2 pixels
Serr. Pulse Width: Htot - HSwid pixels (progressive)
[Htot/2] - HSwid pxls (interlace)
Equalization Interval:
Before: 0 to Vtot - Vact -VSwid - EqAfter Lines
After: 0 to Vtot - Vact -VSwid - EqBefore Lines
Step: 1 line
Interval Modes:
American (interval is equal to programmed value)
European (interval =1/2 line < programmed value)

Outputs

RGB ANALOG VIDEO OUTPUTS

Source Z: 75-ohms
Output levels:
Video Swing: 0 to +1.0 V
Sync Swing: 0 to Ð 400 mV
Setup: 0 to 100 IRE
Autocal: autoadjust (internal precision ref.)
Output protection: Output buffer/75 Ω series term
Rise / Fall: 2 nSec. (typical)
Overshoot: ≤ 10% (all outputs terminated)

DIGITAL OUTPUTS

Source Z: 75-ohms ±2%
Levels: Ô0Õ = 0 V Ô1Õ = 5 V open circuit
Ô0Õ = 0 VÔ1Õ = 2.5 V (term)
Rise / Fall : < 4.0 nSec

TV OUTPUT (801GX Only)

Source Z: 75-ohms ±2%
Video Signal: Composite (lum.,chrom. & burst)
Rise / Fall: per RS-170A & CCIR stds

S-VIDEO OUTPUT (801GX Only)

Source Z: 75-ohms 2%
Signals: Y(luminance); C (chrominance)
Rise / Fall: per RS-170A & CCIR stds

Additional Model 801GF Specifications

Pixel Timing

Frequency Range: 3.9975 MHz to 250 MHz
Step: 0.035 Hz
Jitter: ≤800 pS line-to-line (1 sigma)
Accuracy: 25 ppm

Horizontal Timing

Frequency Range: 1.00 Ð 130KHz

Total pixels per line:
Range: 144 to 65,535 pixels
Step: 1 pixel

Active pixels per line:
Range: 16 to 2048 pixels typical (up to 32,768 pixels under special conditions)
Step: 1 pixel

Sync pulse width:
Range: 1 to Htot - Hact -HSdel pixels
Step: 1 pixel

Sync delay (front porch):
Range: 1 to Htot - Hact -HSwid pixels
Step: 1 pixel

Vertical Timing

Frequency Range: 1 to 650Hz
Vertical total scan lines per frame:
Range: 2 to 4096 lines (progressive)
5 to 8191 lines (interlace)
Step: 1 line (progressive)
2 lines (interlace)
Vertical active scan lines / frame:
Range: 1 to 2048 lines (progressive)
2 to 2048 lines (interlace)
(up to 4096 lines under special conditions)
Step: 1 line (progressive)
2 lines (interlace)
Vertical Sync pulse width:
Range: 1 to Vtot - Vact -VSdel lines
Step: 1 line
Modes: American or European
Vertcal Sync delay (front porch):
Range: 0 to Vtot - Vact -VSwid lines
Step: 1 line

Horizontal actives beyond 2048 pixels require a reduced vertical active limit. Vertical actives beyond 2048 lines require a reduced horizontal active limit. Required limit reductions shown in table. Horizontal actives beyond 4096 pixels repeat the information contained in the first 4096 pixels of each scan line.

H Active pixels	V Active lines
513-1024	4096
1025-2048	2048
2049-4096	1024
4097-8192	512
8193-16384	256
16385-32768	128

Digital Sync

Modes: Separate Horiz. & Vert.
Composite
Composite Configurations:
American HS OR'ed with VS
American Serrated
American Serr. &Eq.(interlaced only)
European HS OR'ed with VS
European Serrated
European Serr. & Eq.(interlaced only)
Eq. Pulse Width: HSwidth / 2 pixels
Serr. Pulse Width: Htot - HSwid pixels (progressive)
[Htot/2] - HSwid pxls (interlace)
Equalization Interval:
Before: 0 to Vtot - Vact -VSwid - EqAfter Lines
After: 0 to Vtot - Vact -VSwid - EqBefore Lines
Step: 1 line
Interval Modes:
American (interval is equal to programmed value)
European (interval =1/2 line < programmed value)

Outputs

RGB ANALOG VIDEO OUTPUTS

Source Z: 75-ohms
Config: R, G, B or Y, Pr, Pb
Output levels:
Video Swing: 0 to +1.0 V
Sync Swing: 0 to Ð 400 mV
Setup: 0 to 100 IRE
Autocal: autoadjust (internal precision ref.)
Output protection: Output buffer / 75 Ω series term
Rise / Fall: 1.5 nSec. (typical)
Overshoot: ≤ 10% (all outputs terminated)

DIGITAL OUTPUTS

Source Z: 75-ohms ±2%
Levels: Ô0Õ = 0 VÔ1Õ = 5 V open circuit
Ô0Õ = 0 VÔ1Õ = 2.5 V (term)
Rise / Fall : < 4.0 nSec

Notes:

An alphabetically sorted list of subjects covered in this manual.

Page numbers in boldface indicate where the main discussion of a topic may be found.

Index

A

B

C

D

E

F

G

H

I

U

V

W

Y

Notes:

QUANTUM DATA®

2111 Big Timber Road
Elgin, IL 60123-1100 USA

Phone: (847) 888-0450
Fax: (847) 888-2802
BBS: (847) 888-0115
[≤19.2KB, 8-N-1 protocol]

Please note that on January 20, 1996 our phone, fax and BBS area codes changed from 708 to 847.

Internet Connections

World Wide Web Site: http://www.quantumdata.com
Sales & Service E-mail: sales@quantumdata.com
Technical Support E-mail: support@quantumdata.com

Model 801GC, 801GF & 801GX Owner's & Programmers Manual • Part# 68-00157 (rev. A 21-Feb-96