# POLYCOM®

Installed Voice Business Group

# RS-232 Command Set:

# Vortex EF2280 Programming Guide

Copyright ©  2003  Polycom, Inc.

Polycom ® and the Polycom logo are registered trademarks of Polycom, Inc.

ASPI, Vortex, VS4000, Viewstation, and iPower are
registered trademarks of Polycom, Inc.

All other brand names, product names, and trademarks are the
sole property of their respective owners

# Vortex EF2280 Programming Guide

## Table of Contents

# 1. Introduction

This document describes the command protocol that is used to communicate with the Vortex EF2280 via its RS-232 port.

# 2. RS-232 Hardware

The communication between the Vortex and a host controller is conducted via the RS-232 port on the back panel of the Vortex. The Vortex's RS-232 port operates at the following settings:

- Bit Rate (bps): 9600 (default), 19200, or 38400
- Data Format: 8 data bits, no parity, 1 stop bit (8N1)
- Flow Control: none (default) or hardware (RTS, CTS)

Note that although the flow control setting is user selectable to none or hardware, hardware flow control must be used when updating the firmware of the Vortex. This means that while a 3-wire RS-232 cable (RX, TX, GND) is acceptable for control of the Vortex, the minimum cable for updating firmware should contain 5 wires (RX, TX, RTS, CTS, GND).

# 3. Programming Tips

## 3.1. Initialization

During power up of the Vortex device or initialization of the host program, electrical fluctuations on the RS-232 lines may cause the Vortex to receive spurious data. After initialization, the host program should send a few carriage return characters (^M, ASCII 13) to the connected Vortex to flush out any spurious characters it may have received. Failure to do this could result in the Vortex ignoring the first command received after initialization.

As part of the host program initialization, the host will want to synchronize the status of its controls with the current status of the Vortex devices that are being controlled. The host could initialize its controls to default

values and then send commands to the Vortexes to set them to the same state. This approach has significant disadvantages since initializing the host program could undo settings made by other host programs or by the front panel controls of one of the linked Vortex devices. A better approach is to query the linked Vortex devices for their status and set the host program controls based on the return values.

## 3.2. Wildcard Characters

The use of the wildcard character, '*', can make programming the host controller much easier. Be careful when using wildcards, however, since they can generate a lot of traffic on the digital bus.

## 3.3. Using Acknowledgements

It is a good idea for the host program or control system to make sure that all connected Vortex devices have acknowledgment mode enabled (see the ACKMOD command). When acknowledgment mode is on, a Vortex device will send an acknowledgment for each command it receives. Proper use of acknowledgments makes the host program more robust and makes supporting multiple hosts effortless. The rest of this section describes how to use acknowledgments to achieve these goals.

As a convenient example, let us imagine a host program with a graphical user interface (GUI). The user presses buttons on the GUI to enable or disable features of various linked Vortex devices. The buttons on the GUI reflect the current status of the corresponding feature.

When the user presses a button on the GUI to enable or disable a feature, the host program should send the corresponding command to the selected Vortex device. It may be tempting to update the status of the GUI button at this point, but this can cause problems if there are transmission errors or if there are multiple host controllers. The proper way to handle this is to only update the GUI controls based on acknowledgments received from the Vortex device.

To implement this, organize your code so that the functions that send commands are totally separate from the functions that receive responses from the Vortex devices. This also enables your host program to support the presence of multiple host controllers. For example, consider the following sequence of events.

1. Another host sends a command to a Vortex device.
2. The Vortex device responds with an acknowledgment that is broadcast to all of the other hosts.
3. Your host program receives the acknowledgment and updates the status of the corresponding control.

The result of this programming model is that all hosts and linked Vortex devices will always be synchronized.

For simple on/off features, your host program can make use of the toggle arguments to some commands (e.g., 'SSEN2'). By sending a toggle command when a button is pressed, and updating the button based on acknowledgments, your host program will not have to keep track of the status of the button.

In a similar fashion, many of the integer commands (such as gains) can be controlled by incrementing or decrementing them by a specific amount. For example, the command 'GAINIA>1 ' increments the input gain on channel A by 1 dB. The acknowledgment for this command will return the actual value that the gain was incremented to. Thus, to implement a volume control, your control program can send a command to increment the gain by 1 dB when the "up" button is pressed and decrement the gain by 1 dB when the "down" button is pressed. The control can update its level indicator based on the acknowledgment that is received.

## 3.4. Macros and Presets

Although macros and presets are similar, there are times when using one is better than the other. Presets store

the absolute values of all of the non-global settings of the device. This includes, but is not limited to, input and output gain settings, matrix settings, algorithm settings, parametric EQ settings, and automixer settings. See Section 7 for a list showing all the commands and which are saved to presets.

Presets should be used when you really want to change all the settings in a device. One example would be when you want to have one unit be able to control different rooms. In this case, having a preset for each different room is the easiest solution.

Macros are like mini-presets. You can define them to change only the settings you are interested in. One advantage of macros over presets is that macros can make relative changes in addition to absolute changes. An absolute change is something like "set the input gain to -3 dB". A relative change is something like "raise the input gain by 3 dB". One example of using macros for a relative change is stereo volume ramping. If you have two outputs setup to have left and right program audio, then you could build a macro that contains two commands: one to increment the left channel by 1 dB and the other to increment the right channel by 1 dB. Then, by calling that macro, you can ramp the stereo outputs. A similar thing can be done with decreasing the volume.

Another thing to consider when using macros and presets is to use the MACROQ and PRESETQ commands instead of the MACROX and PRESETX commands. Both the Q and X versions execute the macro or preset, but the X versions produce acknowledgements for the settings that change, while the Q versions don't. If your control system updates its status by looking at the acknowledgements that come back, then you'll probably want to use the X versions. Another option would be to use the Q versions and then manually query the values you're interested in. If your control system does not need use acknowledgements, or if you are manually querying the values you're interested in, using the Q versions is better since it doesn't generate acknowledgements and thus reduces RS-232 traffic.

## 3.5. Automixer Dependencies

The commands AMASGN, AMCHAIR, AMCHNUM, AMLMM, AMLMN, and MGATE have dependencies on each other and can cause errors (ERROR#040 through ERROR#045) if an assignment attempts to break one of these dependencies. See the descriptions of the above commands and the ERROR command for more information on these dependencies.

The dependencies in these commands can cause a problem when trying to build macros. For example, your macro may use the above commands to set the automixer to a certain configuration. The problem is that although your commands would put the automixer in a valid configuration, one of the intermediate configurations might be invalid. If this happens, the invalid command(s) will not execute and the automixer will not be in the configuration that you intended.

For example, assume that we start with all inputs assigned to automixer group "none" (AMASGN), chairman mode (AMCHAIR) is disabled for both automixers, chairman mic (AMCHNUM) is set to 1 for both automixers, last mic mode (AMLMM) is set to off for both automixers, and last mic number (AMLMN) is set to 1 for both automixers. Now, suppose your macro executes the following commands in the order shown.

```
AMASGN*àààààååå
```
(assign inputs 1-4 to automixer 1 and inputs 5-8 to automixer 2)
```
AMCHAIR11
AMCHAIR21
AMCHNUM11
AMCHNUM25
AMLMM11
```

```
AMLMM21
AMLMN25
```

In this case, the `AMCHAIR21` command and the `AMLMM21` would not get executed. They would produce `ERROR#044` and `ERROR#042`, respectively. The `AMCHAIR21` command causes an error because we try to turn on chairman mode for automixer 2, but automixer 2's chairman mic is set to 1, whic belongs to automixer 1. Similarly, the `AMLMM21` command causes an error because we try to set last mic on mode to manual for automixer 2, but automixer 2's last mic number is set to 1, which belongs to automixer 1.

There are many other ways that these dependencies can cause problems. Fortunately, there is a way to avoid them. In you command sequences and macros, follow the following sequence when dealing with the automixer paramters.

- turn off chairman mode (<u>AMCHAIR</u>) for both automixers
- set last mic mode (<u>AMLMM</u>) off for both automixers
- ungate all matrix crosspoints (<u>MGATE</u>) that correspond to inputs that you will be changing
- Now, execute your automixer commands in the following order:

# 4. Command Structure

A Vortex command consists of a series of ASCII characters with the following structure.

| Description | Number of Characters | Range of Values |
|---|---|---|
| Device Type | 1 | `0-9, A-Z, *` |
| Device ID | 2 | `0-9, *` |
| Command Name | 1-7 | `0-9, A-Z` |
| Command Data | 0-64 | ASCII characters |
| Command Terminator | 1 | `^M` (ASCII 13) |

## 4.1. Device Type

A single alphanumeric character is used to indicate the device type. The devices in Polycom's EchoFree family have the following device types.

| Device | Device Type |
|---|---|
| EF200 | `A` |
| EF1210 | `C` |
| EF2280 | `F` |
| EF2241 | `B` |
| EF2211 | `S` |
| EF2210 | `Q` |
| EF2201 | `T` |

Device type '*' can be used to send a command to all device types simultaneously.

## 4.2. Device ID

Two numeric characters are used to indicate the device ID. The Vortex can be configured for device IDs from '00' to '07'. Note that even though the device ID is less than 10, the leading '0' must be included. Device ID '**' can be used to send a command to all device IDs simultaneously. Some examples of using wildcard characters are given below.

- 'F**' broadcasts to all EF2280 devices that are linked together.
- '*07' broadcasts to all devices with device ID 07 (this format is not commonly used).
- '***' broadcasts to all devices that are linked together.

## 4.3. Command Name

The command name can be from 1 to 7 characters long. Command names will be specific to device types. In other words, the EF2280 has its own command set, which is different from the EF2241's, which is different from the EF1210's, etc. There are some commands, such as 'PING', that are common among all the various command sets.

## 4.4. Command Data

The command data is a series of 0-64 characters containing payload data for the command. Obviously, the command data will be specific to the command type. note that the maximum number of payload characters for the EF200 and EF1210 is 32, but the Vortex devices support up to 64 characters in the payload. This increase was necessary to accommodate the matrix gain and macro/preset commands.

## 4.5. Command Terminator

The command terminator is a single character indicating the end of a command. ASCII 13 (^M) was chosen as the terminator to allow manually typing commands using a simple text terminal.

## 4.6. Examples

In the following examples, Vortex commands are enclosed in single quotes, 'like this'. Also, the terminator character is not explicitly shown, but its presence is implied.

Consider the command '***PING'. The device type and ID for this command are wildcards, thus the command will be sent to all devices. The command name in this case is 'PING', and there are no data characters (payload). Note that the 'PING' command is supported by all of the Vortex devices, thus broadcasting the command to all devices makes sense.

Consider the command 'F**GAINIA10'. The device type for this command is 'F' and the device ID is a wildcard, thus the command will be broadcast to all EF2280 devices linked together. The command name in this case is 'GAINIA' and the command data (payload) is '10'. This command sets the gain on input A of all connected EF2280's to 10 dB.

| Command | Effects |
|---|---|
| ***PING | Requests PONG response from all linked Vortex devices. |
| F**GAINIA10 | Sets the gain on input A of all connected EF2280 devices to 10 dB. |

# 5. Status Messages

The Vortex sends status messages via RS-232 and EF Bus any time one of its internal parameters changes. This means that the host program does not need to continually poll the Vortex in order to detect status changes. Status messages are in the same format as the commands used to set the corresponding parameter.

For example, suppose you send the command 'F**GAINIA10' and there are two EF2280's linked together with device IDs 3 and 7. The EF2280's will respond with 'F03GAINIA10' and 'F07GAINIA10', respectively. Now, someone uses the front panel of the EF2280 set to ID 7 to decrease input A's gain by 1 dB. When this happens, the EF2280 will respond with 'F07GAINIA9'. This example illustrates that status messages can be sent as the result of an RS-232 command or as the result of some other change in the Vortex device such as front panel adjustments, logic inputs, etc.

Status messages can be turned off via the ACKMOD command. ACKMOD refers to acknowledgement mode since the term status message and acknowledgement are synonymous for our purposes.

# 6. Command Types

Many of the Vortex commands have similar formats. The main formats are described here in order to provide a better understanding of the command set.

## 6.1. Boolean Commands

Boolean commands take one of the three following arguments.

- '0' indicates that the parameter should be turned off.
- '1' indicates that the parameter should be turned on.
- '2' indicates that the parameter should be toggled (i.e., '0' becomes '1' and '1' becomes '0').

Parameters associated with boolean commands can be queried using the '?' character. For example, if input A is muted, and you send 'F04MUTEIA? ', the EF2280 will respond with a status message of 'F04MUTEIA1 '. When a status message is generated for a boolean command, the command data will either be a '0' or '1', since '2' is obviously not a valid state.

## 6.2. Integer Commands

Integer commands can take one of two types of arguments. The first argument type is absolute, meaning that the parameter will be set to the specified number. For example, 'F04GAINIA10 ' means that the gain on input A will be set to 10 dB. In this case, the device will respond with a status message of 'F04GAINIA10 '.

The second argument type is relative, meaning that the parameter will be incremented or decremented by the specified amount. The increment character is '>' and the decrement character is '<'. For example, 'F04GAINIA>3 ' increments the gain on input A by 3 dB. If the input's gain was previously set to 6 dB, then it would now be set to 9 dB. In this case the device would respond with a status message of 'F04GAINIA9 '.

The numeric part of both the absolute and relative arguments can contain a '+' or '-' to indicate the algebraic sign of the argument. If no sign is given, '+' is assumed.

The parameters associated with integer commands have maximum and minimum values associated with them. If you try to set a parameter above its maximum or below its minimum, the parameter will be set to its maximum

or minimum value, respectively.

Parameters associated with integer commands can be queried using the '?' character. For example, if input A 's gain is set to 12, and you send 'F04GAINIA? ', the device will respond with a status message of 'F04GAINIA12 '.

# 6.3. Channel Commands

A command can be a channel command in addition to being one of the other types of commands (integer or boolean). A channel command means that the command applies to a specific input or output channel. The channel is specified by a single character (e.g., '1', '2', 'A', 'B', etc.) occurring before any other payload data.

An example of a boolean channel command is the 'AGC' (Automatic Gain Control) command. 'F04AGC30 ' disables the AEC on input channel 3 . After sending this command, the device will respond with a status message of 'F04AGC30 '.

An example of an integer channel command is the 'GAINI' command, which adjusts the gain on the input channels. 'F04GAINIA12 ' sets the input gain of channel A to 12 dB. After sending this command, the device will respond with a status message of 'F04GAINIA12 '.

A wildcard character ('*') can be used as the channel specifier for many of the channel commands. If this is the case, there are two options for specifying the values for the channels. The first method is to specify a single value that will be applied to all the channels.

Take the MUTEI command for example: 'F04MUTEI*1 ' mutes input channels 1-8 and A-D After sending this command, the device will respond with a status message of 'F04MUTEI*111111111111 '. In this status message, the device reports the status of all the channels. Since the MUTEI command applies to channels 1-8 and A-D the status of all 12 channels. The first (left most) value corresponds to channel 1 and the last (right most) value corresponds to channel D .

This leads us to the second method of using a wildcard character: specifying the values for each of the channels. For example, 'F04MUTEI*101010101010 ' mutes channels 1, 3, 5, 7, A, C and unmutes channels 2, 4, 6, 8, B, D . In this case, the device will respond with a status message of 'F04MUTEI*101010101010 '.

As an interesting example, consider sending 'F04MUTEI*2 ' after the above example. The device will respond with a status message of 'F04MUTEI*010101010101 '. Notice that all the states have been toggled.

Queries using the '?' character are straightforward. For example, 'F04MUTEI1? ' might return 'F04MUTEI11 ', while 'F04MUTEI*? ' might return 'F04MUTEI*111111110000 '.

Using the wildcard character with integer channel commands is similar to using it with boolean commands, but there are some differences. If we sent 'F04GAINI*10 ', the input gains on channels 1-8 and A-D will all be set to 10 dB. The device will respond with a status message of 'F04GAINI*ÄÄÄÄÄÄÄÄÄÄÄÄ '. Those weird characters are the main difference between using wildcards with integer channel commands versus boolean channel commands. When reporting multiple integer values, the Vortex uses a binary format with one byte per integer value. This allows for more compact commands and reduces RS-232 and bus traffic. To convert from an integer value to a byte value, we add 132 to the integer value. In the above example, where the gain is set to 10 dB, we have: 10 + 132 = 142 = 0x8E (hex) = Ä (ASCII). The reason for adding 132 is to allow us to conveniently represent negative numbers as well as avoid the use of special characters that are normally used in RS-232 and EF Bus communications.

When using a wildcard character to specify separate values for each channel, you must also use the binary

format. For example, 'F04GAINI*ÄÄÄÄÄÄÄÄääää ' sets the input gain of channels 1-8 to 10 dB and the input gain of channels A-D to 0 dB. (10 + 132 = 142 = 0x8E (hex) = Ä (ASCII), 0 + 132 = 132 = 0x84 (hex) = ä (ASCII))

As an interesting example, consider sending 'F04GAINI*>3 ' after the above example. This will result in all the input gains being incremented by 3 dB so that channels 1-8 are at 13 dB and channels A-D are at 3 dB. The Vortex will respond with a status message of 'F04GAINI*ææææææææçççç '. (13 + 132 = 145 = 0x91 (hex) = æ (ASCII), 3 + 132 = 135 = 0x87 (hex) = ç (ASCII))

Queries using the '?' character are straightforward. For example, 'F04GAINI1? ' might return 'F04GAINI110 ', while 'F04GAINI*? ' might return 'F04GAINI*ÄÄÄÄÄÄÄÄääää '.

## 6.4. Matrix Commands

Matrix commands are used for controlling parameters that exist at the crosspoints of the mixing matrices. Typical parameters include gating (for automixed signals), gain and mute. Before describing the matrix commands, it is necessary to give a description of the matrices involved. During the following discussion, it will be helpful to refer to the figure below, which shows all the matrices with their input and output labels.

XM1
XM2

YB0 YB1 YB2 YB3 YB4 YB5 YB6 YB7

YM0

YM1

YM2

ZB0 ZB1 ZB2 ZB3 ZB4 ZB5 ZB6 ZB7

ZM0

ZM1

ZM2
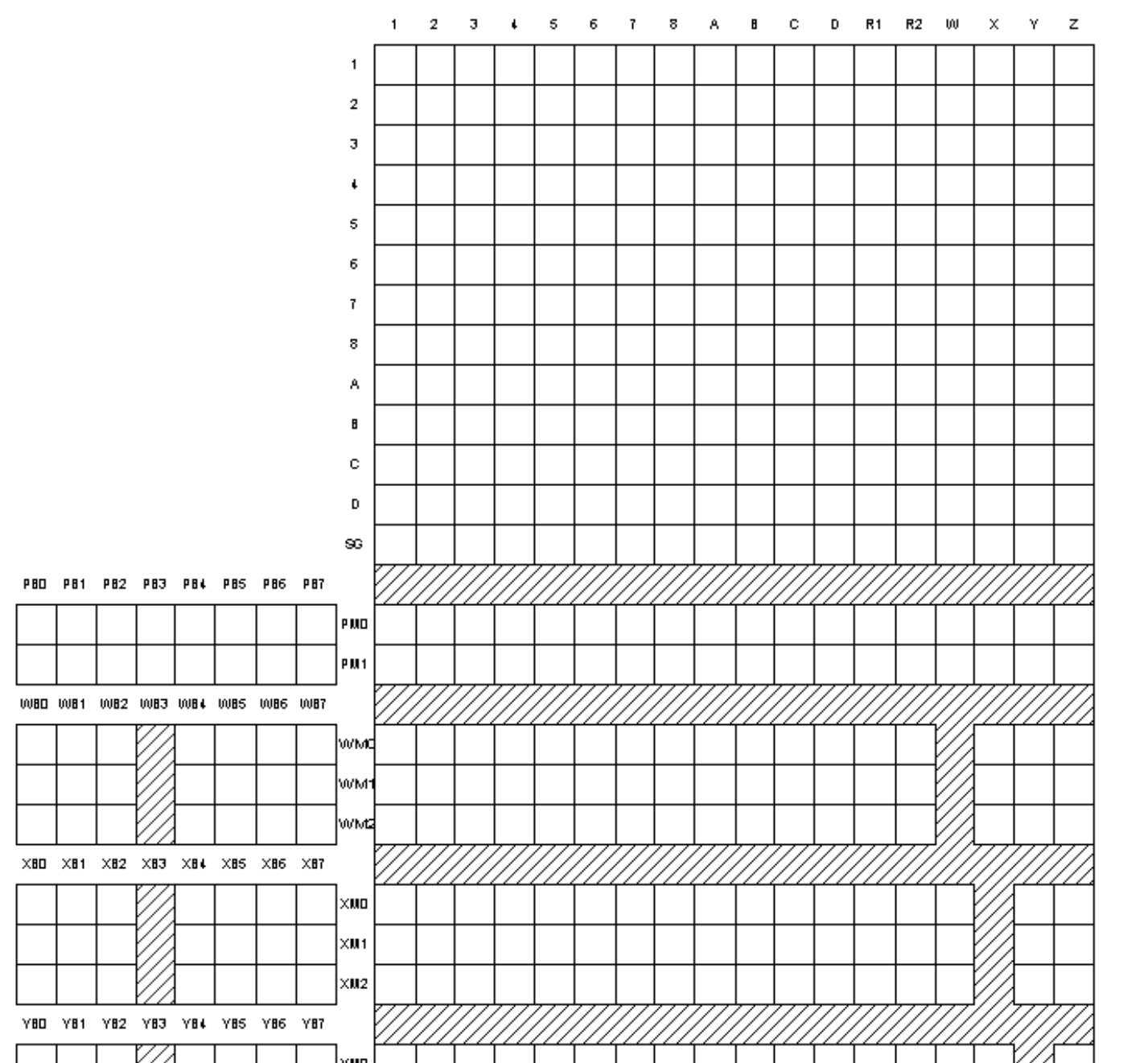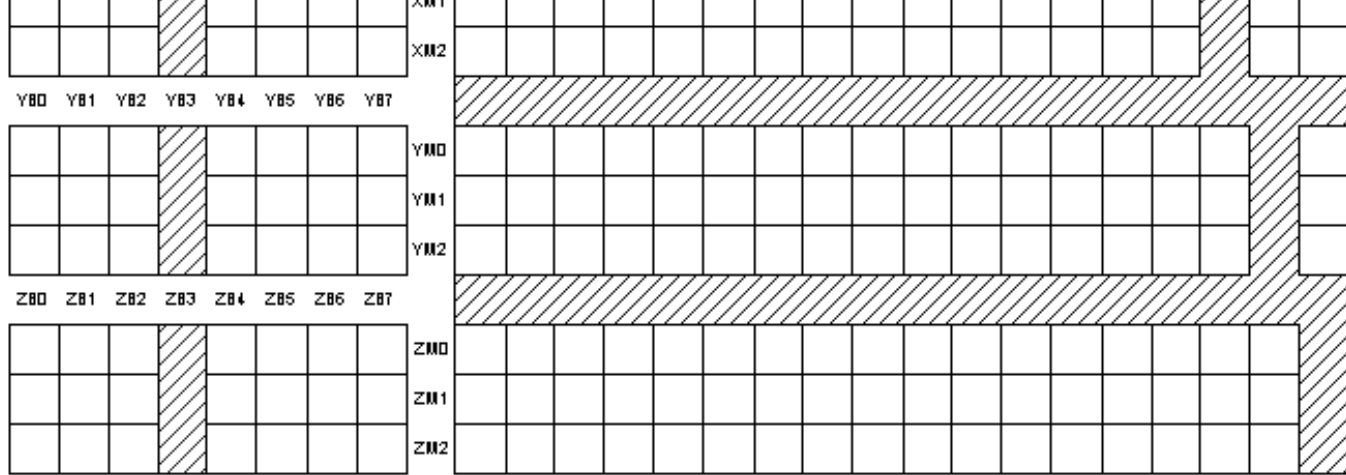
The EF2280 has twelve analog outputs labeled 1-8 and A-D. These outputs are all at line level.

The EF2280 has twelve analog inputs labeled 1-8 and A-D. Inputs 1-8 are mic/line selectable, and inputs A-D are line level only. Inputs 1-8 can also have phantom power enabled and contain channel processing, which includes the follwoing DSP algorithms: Acoustic Echo Cancellation, Noise Cancellation, AGC, and Automatic Microphone Mixing.

Vortex devices can be linked together so that they can share control information and digital audio signals. The audio signals are shared on four digital busses labeled P, W, X, Y, and Z. All Vortex devices can receive signals from all of these busses. Only certain devices can transmit on the busses. This information is given in the following table.

| Device | Transmit on P Bus | Transmit on W, X, Y, Z Busses | Receive P Bus | Receive W, X, Y, Z Busses |
| --- | --- | --- | --- | --- |
| EF2280 | No | Yes | Yes | Yes |
| EF2241 | Yes | Yes | Yes | Yes |
| EF2211 | Yes | Yes | Yes | Yes |
| EF2210 | No | Yes | Yes | Yes |
| EF2201 | Yes | No | Yes | Yes |

The P bus is meant for routing telephone audio between the devices. The W, X, Y, and Z busses are meant for routing microphone and auxiliary audio between the devices. The W, X, Y, and Z busses also carry NOM (Number of Open Microphones) information from the automixer so that outputs created from these busses can be appropriately attenuated for the number of open microphones.

The digital inputs consist of all of the signals placed on the EF Bus by the other connected Vortex devices. Each P, W, X, Y, and Z bus can carry channels from up to eight other devices, so we have the following digital inputs to each Vortex device: PB0-PB7, WB0-WB7, XB0-XB7, YB0-YB7, and ZB0-ZB7. The inputs are designated by three characters: the bus letter (P, W, X, Y, or Z), a B indicating that it is a bus input, and a number between 0 and 7 indicating the channel of the bus.

There is also an internal signal generator, labeled SG, that is capable of producing white or pink noise. this signal is fed into the matrix so that it can be routed to the appropriate outputs for calibration or testing.

The mixing capabilities of the Vortex devices can be broken down into two parts: the EF Bus submatrices and the main matrix.

For each of the W, X, Y, and Z signal busses, there is a 7 x 3 matrix that allows the user to define up to three mixes of each of the four signal busses. The reason the matrix is 7 x 3 instead of 8 x 3 is that since we can transmit on the W, X, Y, and Z busses, we do not need to mix our own channels in these matrices. The inputs and outputs for the 7 x 3 matrices are as follows.

### W Submatrix

- Inputs: WB0-WB7 (with one invalid)
- Outputs: WM0-WM2

### X Submatrix

- Inputs: XB0-XB7 (with one invalid)
- Outputs: XM0-XM2

### Y Submatrix

- Inputs: YB0-YB7 (with one invalid)
- Outputs: YM0-YM2

### Z Submatrix

- Inputs: ZB0-ZB7 (with one invalid)
- Outputs: ZM0-ZM2

For the P signal bus, there is an 8 x 2 matrix that allows the user to define up to two mixes of the P signal bus. This matrix has a full 8 inputs since the EF2280 does not transmit on the P bus. The inputs and outputs for the 8 x 2 P bus matrix are as follows.

### P Submatrix

- Inputs: PB0-PB7
- Outputs: PM0-PM1

The crosspoint gains on all outputs are user adjustable. The "M" in the output labels of the submatrices indicates that the signals are being fed into the Main Matrix.

The main matrix consists of the following inputs: analog inputs 1-8 and A-D, the signal generator (SG), the outputs of the EF Bus submatrices PM0-PM1, WM0-WM2, XM0-XM2, YM0-YM2, and ZM0-ZM2. This is a total of 27 inputs.

The main matrix consists of the following outputs: analog outputs 1-8 and A-D, AEC reference signals R1 and R2, and the EF Bus outputs W, X, Y, and Z. There is no P output since the EF2280 does not transmit on the P bus. This is a total of 18 outputs.

The matrix commands can adjust two types of parameters: integer and boolean. We will introduce the matrix commands by using the MGAIN command as an example. This is an integer matrix command that is used to set the gain (in dB) at any of the crosspoints in the main matrix or EF Bus submatrices.

Matrix commands are similar to channel commands except that instead of specifying a single channel, it is necessary to specify a crosspoint (or range of crosspoints). In order to specify a single crosspoint, you use the input and output labels discussed in this section. The first label always specifies the input to the matrix and the second label always specifies the output of the matrix. For example, to set the gain of the crosspoint (1, A) to -3 dB, you would send `F04MGAIN1,A,-3 ` which sets the gain at the crosspoint to -3 dB. In this case, a status message will be generated similar to `F04MGAIN1,A,-3 `.

It is also possible to use the wildcard character ('`*`') to specify ranges of crosspoints with the matrix commands. The only restriction is that you can only use a wildcard to specify the input or output, but not both

simultaneously. Thus you could specify all the inputs going to a specific output (one column) or the value of an input to all of the outputs (one row), but not the entire matrix. One example of using a wildcard for an integer matrix command would be `'F04MGAINSG,*,0 '`. This will set all the crosspoints in the signal generator row of the main matrix to 0 dB. Thus, the signal generator will be added to all of the outputs of the main matrix with a gain of 0 dB. In this case a status message will be generated that looks like `'F04MGAINSG,*,ääääääääääääääää '`. The binary representation used here is the same method described in [Section 6.3](#).

You can also use the wildcard character to set the crosspoints of a row or column individually. For example, `'F04MGAIN1,*,ääääääääzzzzxxxxxx'` sets the crosspoints of input 1 to 0 dB for outputs 1-8, -10 dB for outputs A-D, and -12 dB for outputs R1, R2, and W-Z. In this case, the EF2280 will respond with a status message of `'F04MGAINSG,*,ääääääääzzzzxxxxxx'`.

Queries using the `'?'` character work in the usual manner. For example, `'F04MGAIN3,A,? '` might return `'F04MGAIN3,A,-6 '`, while `'F04MGAIN2,*,? '` might return `'F04MGAIN2,*,ääääääääääääääää '`.

Boolean matrix commands work as you would expect. They use the characters `'0'`, `'1'`, and `'2'` as described in [Section 6.1](#). Here are some examples.

- `'F04MMUTE2,3,1 '` mutes crosspoint (2, 3) of the main matrix. In other words, the signal at input 2 will not be heard on output 3 A status message will be generated of the form `'F04MMUTE2,3,1 '`.
- `'F04MMUTEA,*,111111110000000000'` sets the mutes for input A of the main matrix. The signal path from input A to outputs 1-8 is muted, while the signal path from input A to outputs A-D, R1, R2, and W-Z is unmuted. The EF2280 will respond with a status message of `'F04MMUTEA,*,111111110000000000'`.
- `'F04MMUTEA,*,2'` toggles the mutes for input A of the main matrix. If this command follows after the command in the above example, the EF2280 will respond with a status message of `'F04MMUTEA,*,000000001111111111'`.
- `'F04MMUTEA,*,?'` queries the status of the mutes for input A of the main matrix. If this command follows after the commands in the above examples, the EF2280 will respond with a status message of `'F04MMUTEA,*,000000001111111111'`.

It should be noted that in EF2280 firmware versions earlier than 2.x, the P-bus was not implemented. Thus, there were two less inputs to the main matrix (PM0 and PM1) and one less output to the main matrix (P). This means that matrix commands for earlier versions of the firmware had different requirements for the number of characters in a row or column of the main matrix. To preserve backward compatibility and ease migration to the new firmware, the 2.x firmware supports both formats of matrix commands. If the P-bus entries are left out of a matrix command, the command will still execute correctly and the P-bus crosspoints will not be changed.

## 6.5. Miscellaneous Commands

Miscellaneous commands are those that don't fall under any of the other categories. See the description of a given command for specific details on how it operates.

# 7. Command List

The following table is a list of the commands recognized by the EF2280 Detailed descriptions of each command are given in the next section.

The Storage column contains one of the following values indicating when and where the parameter is stored.

- "Global"
- "Preset"

- "-" = not stored or not applicable

Globally stored parameters are not changed when a preset is executed. Only one copy of a global parameter is stored. Global parameters are written to non-volatile memory each time they are changed. Globally stored parameters retain their values when the power is cycled.

Parameters stored in presets are changed each time a new preset is restored/executed. Preset parameters are not saved in non-volatile memory until a PRESETW command is executed. Parameters stored in the power-on preset (see PRESETP) are restored when the power is cycled.

| Command | Storage | Description |
|---------|---------|-------------|
| ACKMOD | Global | Enable or Disable Acknowledgment Mode |
| AEC | Preset | Enable or Disable Acoustic Echo Cancellation |
| AECMODE | Preset | Set Amount of Double Talk Suppression used in the AEC. |
| AGC | Preset | Enable or Disable Mic/Line Input Automatic Gain Control |
| AGCMAX | Preset | Set Maximum Allowed Mic/Line Input AGC Gain |
| AGCMIN | Preset | Set Minimum Allowed Mic/Line Input AGC Gain |
| AGCRATE | Preset | Set Ramp Rate of Mic/Line Input AGC |
| AMASGN | Preset | Assign Inputs to an Automixer |
| AMAUTO | Preset | Select Automatic or Manual Gating for each Automixer Input |
| AMBUSID | Preset | Set Automixer Groupings for EF Bus |
| AMCHAIR | Preset | Enable Chairman Mode for Specified Automixer |
| AMCHNUM | Preset | Set Chairman Mic for Specified Automixer |
| AMDECAY | Preset | Set Decay Time for Automixers |
| AMGATEC | Preset | Set Automixer Gating Control Mode |
| AMGATER | Preset | Set Automixer Gate Ratio |
| AMGATET | Preset | Set Automixer Gate Threshold |
| AMGNOM | Preset | Set Global Maximum Number of Open Mics for Bus Automixer |
| AMHOLD | Preset | Set Automixer Hold Time |
| AMLMM | Preset | Set Last Mic On Mode for Specified Automixer |
| AMLMN | Preset | Set Microphone That Will Remain On in Manual Last Mic On Mode |
| AMNOM | Preset | Set Local Maximum Number of Open Mics for Automixer |
| AMNOMAT | Preset | Select NOM Attenuation on Each Output |
| AMOFFAT | Preset | Set Off Attenuation for the Specified Automixer |
| AMPRIOR | Preset | Set Gating Priority for the Specified Mic |
| AMREFB | Preset | Set Automixer Reference Bias for the Specified Automixer |
| AMREFE | Preset | Enable Automixer Reference for Specified Automixer |
| BAUD | Global | Set Baud Rate for RS-232 Port |
| BLAUTO | Preset | Enable Automatic BLDATA Messages |
| BLDATA | - | Request Level Information |
| BLINFO | Preset | Select Information to be Reported in BLDATA |

| | | |
|---|---|---|
| BROAD2 | - | Broadcast Arbitrary Command Strings to RS-232 Port |
| BROADA | - | Broadcast Commands to Other Connected Devices |
| BUSREF | Preset | Set Which AEC Reference is Placed on EF Bus |
| CGATE | - | Query Camera Gating Status Information |
| CGATEEN | Preset | Enable Automatic Camera Gating Messages |
| CGATET | Preset | Set Camera Gating Hold Time |
| DELAYO | Preset | Set Output Delay |
| DELAYOE | Preset | Enable Output Delay |
| DSPAUTO | - | Enable Automatic DSPLOAD Status Messages |
| DSPLOAD | - | Query Percentage of Variable DSP Resources Used |
| ERROR | Global | Enable or Disable Error Messages |
| FADERGIL | Preset | Set Fader Gain of Line Inputs as a Group |
| FADERGIM | Preset | Set Gain of Microphone Inputs as a Group |
| FADERI | Preset | Set Input Gain Fader |
| FLOW | Global | Set Flow Control Mode for RS-232 Port |
| FPLOCK | Global | Lock/Unlock Front Panel |
| FPPSWD | - | Change Front Panel Password |
| GAINGIL | Preset | Set Gain of Line Inputs as a Group |
| GAINGIM | Preset | Set Gain of Microphone Inputs as a Group |
| GAINI | Preset | Set Input Gain |
| GAINO | Preset | Set Output Gain |
| GATE | - | Query Gating Status Information |
| GATEEN | Preset | Enable Automatic Gating Messages |
| GMUTEO | Preset | Mute All Outputs |
| ID | Global | Set Device ID |
| LABEL | Global | Set or Query one of the Device Labels |
| LAGC | Preset | Enable or Disable Line Input Automatic Gain Control |
| LAGCLINKAB | Preset | Enable or Disable Stereo AGC Linking on Inputs A and B |
| LAGCLINKCD | Preset | Enable or Disable Stereo AGC Linking on Inputs C and D |
| LAGCMAX | Preset | Set Maximum Allowed Line Input AGC Gain |
| LAGCMIN | Preset | Set Minimum Allowed Line Input AGC Gain |
| LAGCRATE | Preset | Set Ramp Rate of Line Input AGC |
| LI | Global | Query State of Logic Inputs |
| LIA | Global | Assign Action for when Logic Input is Activated |
| LID | Global | Assign Action for when Logic Input is Deactivated |
| LIH | Global | Assign Action for when Logic Input is Held |
| LIEN | Preset | Enable Automatic Logic Input Status Messages |
| LIG | Global | Configure Logic Input Pins Into a Group |

| | | |
|---|---|---|
| LIK | Global | Delete One or All Logic Input Pin Commands |
| LIM | Preset | Mask Logic Input Pins |
| LIN | Global | Assign Command to Logic Input Group |
| LIP | Global | Set Polarity for Logic Inputs |
| LO | - | Query or Set Status of Logic Output Pins |
| LOA | Global | Define Behavior for Logic Output Activated State |
| LOD | Global | Define Behavior for Logic Output Deactivated Status |
| LOEN | Preset | Enable Automatic Logic Output Status Messages |
| LOK | Global | Delete One or All Logic Output Pin Commands |
| LOM | Preset | Mask Logic Output Pins |
| LOP | Global | Set Polarity for Logic Outputs |
| MACROA | - | Add Command to Current Macro |
| MACROK | Global | Delete One or All Macros |
| MACROL | - | List All Commmands in a Macro |
| MACROQ | - | Execute Macro Quietly |
| MACROS | - | Start a New Macro |
| MACROW | Global | Write Macro to Non-Volatile Memory |
| MACROX | - | Execute Macro |
| METER | Preset | Select which Signal is Displayed on the Front Panel LED Meter |
| MGAIN | Preset | Set Crosspoint Gains in Main Matrix or Submatrix |
| MGATE | Preset | Select Gated or Ungated Microphone Signal in Matrix |
| MIC | Preset | Enable Microphone Gain Stage on Inputs 1-8 |
| MINI | Global | Enable Modem Initialization String |
| MINISTR | Global | Set Modem Initialization String |
| MMUTE | Preset | Mute Crosspoint in Main Matrix or Submatrix |
| MUTEGIL | Preset | Set Mute Status of Line Inputs as a Group |
| MUTEGIM | Preset | Set Mute Status of Microphone Inputs as a Group |
| MUTEI | Preset | Mute One or More Inputs |
| MUTEO | Preset | Mute One or More Outputs |
| NC | Preset | Enable Noise Cancellation |
| NCL | Preset | Set Noise Cancellation Attenuation |
| NVINIT | - | Reinitialize Non-Volatile Memory |
| NVLOCK | Global | Lock/Unlock Non-Volatile Memory |
| NVPSWD | - | Change Non-Volatile Memory Password |
| PEQIA | Preset | Set All Parameters for Specified Parametric EQ Input Stage |
| PEQIB | Preset | Set Bandwidth Parameter for Specified Parametric EQ Input Stage |
| PEQIE | Preset | Set Enabled Parameter for Specified Parametric EQ Input Stage |
| PEQIF | Preset | Set Frequency Parameter for Specified Parametric EQ Input Stage |

| | | |
|---|---|---|
| PEQIG | Preset | Set Gain Parameter for Specified Parametric EQ Input Stage |
| PEQIS | Preset | Set Slope Parameter for Specified Parametric EQ Input Stage |
| PEQIT | Preset | Set Type Parameter for Specified Parametric EQ Input Stage |
| PEQOA | Preset | Set All Parameters for Specified Parametric EQ Output Stage |
| PEQOB | Preset | Set Bandwidth Parameter for Specified Parametric EQ Output Stage |
| PEQOE | Preset | Set Enabled Parameter for Specified Parametric EQ Output Stage |
| PEQOF | Preset | Set Frequency Parameter for Specified Parametric EQ Output Stage |
| PEQOG | Preset | Set Gain Parameter for Specified Parametric EQ Output Stage |
| PEQOS | Preset | Set Slope Parameter for Specified Parametric EQ Output Stage |
| PEQOT | Preset | Set Type Parameter for Specified Parametric EQ Output Stage |
| PHANTOM | Preset | Enable Phantom Power on Inputs 1-8 |
| PING | - | See Which Devices Are Present |
| PRESETK | Global | Delete One or All Presets |
| PRESETL | - | List All Commmands in a Preset |
| PRESETP | Global | Set Which Preset Will Be Activated At Power-Up |
| PRESETQ | - | Execute a Preset Quietly |
| PRESETW | Global | Save a Preset |
| PRESETX | - | Execute a Preset |
| REFASGN | Preset | Assign AEC Reference to Input Channel |
| REFGAIN | Preset | Set Reference Output Gain |
| SGGAIN | Preset | Set Gain of Signal Generator |
| SGMUTE | Preset | Mute Signal Generator |
| SGTYPE | Preset | Set Type of Signal Produced by Signal Generator |
| SSDELAY | Preset | Set Delay Between Screen Saver Screens |
| SSEN | Preset | Enable or Disable Screen Saver |
| SSSTART | Preset | Set Idle Time Required for Screen Saver to Start |
| SSTEXT | Preset | Set Text to be Displayed by Screen Saver |
| SWRESET | - | Perform Soft Reset of System |
| SWVER | - | Query Software Version |
| VTXMODI | Preset | Enable VTX Mode on Specified Inputs |
| VTXMODO | Preset | Enable VTX Mode on Specified Inputs |

# 8. Command Reference

## 8.1. ACKMOD -- Enable or Disable Acknowledgment Mode

This command controls whether or not status messages are sent. See Section 5 for more information on status messages. This parameter is enabled by default, and it is rarely turned off by the host controller.

This command is a boolean command. See [Section 6.1](#) for more information on this type of command.

This command is saved to global non-volatile memory and is not part of a preset. Its value is saved each time it is changed. It will retain its value after power-down. Since this command writes to non-volatile memory, there will be a delay before an acknowledgment is returned.

| Example | Description | Status Message |
|---|---|---|
| F01ACKMOD1 | Enable acknowledgement mode. | F01ACKMOD1 |
| F01ACKMOD0 | Disable acknowledgement mode. | F01ACKMOD0 |
| F01ACKMOD2 | Toggle acknowledgement mode. | F01ACKMODx , where x is 0 or 1 depending on the current state of acknowledgement mode. |
| F01ACKMOD? | Query acknowledgement mode. | F01ACKMODx , where x is 0 or 1 depending on the current state of acknowledgement mode. |

## 8.2. `AEC` -- Enable or Disable Acoustic Echo Cancellation

This command sets or queries the status of the Acoustic Echo Cancellation (AEC) algorithm on input channels 1-8 .

This command is a channel boolean command. See [Section 6.3](#) and [Section 6.1](#) for more information on this type of command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| F01AEC31 | Enable AEC on input channel 3 . | F01AEC31 |
| F01AEC30 | Disable AEC on input channel 3 . | F01AEC30 |
| F01AEC32 | Toggle AEC state on input channel 3 . | F01AEC3x , where x is 0 or 1 depending on the current state of the AEC on input channel 3 . |
| F01AEC3? | Query AEC state on input channel 3 . | F01AEC3x , where x is 0 or 1 depending on the current state of the AEC on input channel 3 . |
| F01AEC*1 | Enable AEC on input channels 1-8. | F01AEC*11111111 |
| F01AEC*0 | Disable AEC on input channels 1-8. | F01AEC*00000000 |
| F01AEC*2 | Toggle AEC state on input channels 1-8. | F01AEC*abcdefgh, where a-h are each 0 or 1 depending on the current state of the AEC for each of the eight input channels. |
| F01AEC*? | Query AEC state on input channels 1-8. | F01AEC*abcdefgh, where a-h are each 0 or 1 depending on the current state of the AEC for each of the eight input channels. |

## 8.3. `AECMODE` -- Set Amount of Double Talk Suppression used in the AEC.

This command sets the amount of double talk suppression used in the AEC on input channels 1-8 . The values correspond to the following settings.

- `1` = No Suppression
- `2` = Light Suppression
- `3` = Heavy Suppression
- `4` = Half Duplex

This command is a channel integer command. See [Section 6.3](#) and [Section 6.2](#) for more information on this type of command. The minimum and maximum values for this command are 1 and 4, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| `F01AECMODE13` | Set AEC on input channel 1 to Heavy Suppression. | `F01AECMODE13` |
| `F01AECMODE1?` | Query AEC suppression level on input channel 1. | `F01AECMODE1x` , where `x` is `1`, `2`, `3`, or `4` depending on the current setting of the AEC suppression level on input channel 1. |
| `F01AECMODE*1` | Set AEC on input channels 1-8 to No Suppression. | `F01AECMODE*àààààààà` |
| `F01AECMODE*ààààêêêê` | Set AEC on input channels 1-4 to No Suppression and AEC on input channels 5-8 to Half Duplex. | `F01AECMODE*ààààêêêê` |
| `F01AECMODE*?` | Query AEC suppression level on input channels 1-8. | `F01AECMODE*abcdefgh`, where `a-h` are each à, å, ç, or ê depending on the the current setting of the AEC suppression levels on input channels 1-8. |

# 8.4. `AGC` -- Enable or Disable Mic/Line Input Automatic Gain Control

This command sets or queries the status of the Automatic Gain Control (AGC) algorithm on input channels 1-8 .

This command is a channel boolean command. See [Section 6.3](#) and [Section 6.1](#) for more information on this type of command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| `F01AGC31` | Enable AGC on input channel 3 . | `F01AGC31` |
| `F01AGC10` | Disable AGC on input channel 1 . | `F01AGC10` |
| `F01AGC12` | Toggle AGC state on input channel 1 . | `F01AGC1x` , where `x` is `0` or `1` depending on the current state of the AGC on input channel 1 . |
| `F01AGC2?` | Query AGC state on input channel 2 . | `F01AGC2x` , where `x` is `0` or `1` depending on the current state of the AGC on input channel 2 . |
| `F01AGC*1` | Enable AGC on input channels 1-8. | `F01AGC*11111111` |
| `F01AGC*0` | Disable AGC on input channels 1-8. | `F01AGC*00000000` |

| | | |
|---|---|---|
| `F01AGC*2` | Toggle AGC state on input channels 1-8. | `F01AGC*abcdefgh`, where `a-h` are each `0` or `1` depending on the current state of the AGC for each of the eight input channels. |
| `F01AGC*?` | Query AGC state on input channels 1-8. | `F01AGC*abcdefgh`, where `a-h` are each `0` or `1` depending on the current state of the AGC for each of the eight input channels. |

## 8.5. `AGCMAX` -- Set Maximum Allowed Mic/Line Input AGC Gain

This command sets the maximum gain that the AGC can apply on input channels 1-8 . For example, if `AGCMAX` is set to `10`, then the AGC for that channel can apply a maximum of 10 dB of gain to the input signal.

This command is a channel integer command. See [Section 6.3](#) and [Section 6.2](#) for more information on this type of command. The minimum and maximum values for this command are 0 and 15, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| `F01AGCMAX13` | Set AGC maximum gain on input channel 1 to 3 dB. | `F01AGCMAX13` |
| `F01AGCMAX1?` | Query the AGC maximum gain on input channel 1 . | `F01AGCMAX1x` where `x` is a number between 0 and 15, depending on the current setting of the AGC maximum gain on input channel 1 . |
| `F01AGCMAX*6` | Set AGC maximum gain on input channels 1-8 to 6 dB. | `F01AGCMAX*èèèèèèèè` |
| `F01AGCMAX*ääääìììì` | Set AGC maximum gain on input channels 1-4 to 0 dB and AGC maximum gain on input channels 5-8 to 9 dB. | `F01AGCMAX*ääääìììì` |
| `F01AGCMAX*?` | Query AGC maximum gain on input channels 1-8. | `F01AGCMAX*abcdefgh`, where `a-h` are each between `ä` and `É`, depending on the current setting of the AGC maximum gain for each of the eight input channels. |

## 8.6. `AGCMIN` -- Set Minimum Allowed Mic/Line Input AGC Gain

This command sets the minimum gain that the AGC can apply on input channels 1-8 . For example, if `AGCMIN` is set to `-10`, then the AGC for that channel can apply a minimum of -10 dB of gain to the input signal.

This command is a channel integer command. See [Section 6.3](#) and [Section 6.2](#) for more information on this type of command. The minimum and maximum values for this command are -15 and 0, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| `F01AGCMIN1-3` | Set AGC minimum gain on input channel 1 to -3 dB. | `F01AGCMIN1-3` |

| | | |
|---|---|---|
| `F01AGCMIN1?` | Query the AGC minimum gain on input channel 1 . | `F01AGCMIN1x` where `x` is a number between -15 and 0, depending on the current setting of the AGC minimum gain on input channel 1 . |
| `F01AGCMIN*-6` | Set AGC minimum gain on input channels 1-8 to -6 dB. | `F01AGCMIN*~~~~~~~~` |
| `F01AGCMIN*ääääüüüü` | Set AGC minimum gain on input channels 1-4 to 0 dB and AGC minimum gain on input channels 5-8 to -3 dB. | `F01AGCMIN*ääääüüüü` |
| `F01AGCMIN*?` | Query AGC minimum gain on input channels 1-8. | `F01AGCMIN*abcdefgh`, where `a-h` are each between `u` and `ä`, depending on the current setting of the AGC minimum gain for each of the eight input channels. |

## 8.7. `AGCRATE` -- Set Ramp Rate of Mic/Line Input AGC

This command sets or queries the maximum rate at which the AGC can increase or decrease the gain of the signals on input channels 1-8 . The ramp rate is expressed in dB/sec.

This command is a channel integer command. See [Section 6.3](#) and [Section 6.2](#) for more information on this type of command. The minimum and maximum values for this command are 1 and 5, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| `F01AGCRATE13` | Set AGC ramp rate on input channel 1 to 3 dB/sec. | `F01AGCRATE13` |
| `F01AGCRATE1?` | Query the AGC ramp rate on input channel 1 . | `F01AGCRATE1x` where `x` is a number between 1 and 5, depending on the current setting of the AGC ramp rate on input channel 1 . |
| `F01AGCRATE*5` | Set AGC ramp rate on input channels 1-8 to 5 dB/sec. | `F01AGCRATE*ëëëëëëëë` |
| `F01AGCRATE*àààâêêêê` | Set AGC ramp rate on input channels 1-4 to 1 dB/sec and AGC ramp rate on input channels 5-8 to 4 dB/sec. | `F01AGCRATE*àààâêêêê` |
| `F01AGCRATE*?` | Query AGC ramp rate on input channels 1-8. | `F01AGCRATE*abcdefgh`, where `a-h` are each between `à` and `ë`, depending on the current setting of the AGC ramp rate for each of the eight input channels. |

## 8.8. `AMASGN` -- Assign Inputs to an Automixer

This command is used to assign one of the mic/line inputs (1-8) to an internal automixers. Setting `AMASGN` to `0` for a given input channel corresponds to no automixer, `1` corresponds to Automixer #1 , and `2` corresponds to automixer #2.

An `AMASGN` command usually removes a microphone from one automixer and adds it to another. If the microphone is removed from an automixer where it was assigned as the "Last Mic On" and list mic mode

([AMLMM](#)) is set to manual for that automixer, then the `AMASGN` command will fail and an `ERROR#040` message will be generated. If the microphone is removed from an automixer where it was a ssigned as the "Chairman Mic" and chairman mode ([AMCHAIR](#)) is enabled for that automixer, then an `ERROR#041` message will be generated. See the [ERROR](#) command for more information on error messages. See [Section 3.5](#) for more information on dependencies within the automixer commands.

Although this command is a channel integer command, the increment and decrement operators (> and <) are not supported for this particular command.

This command is a channel integer command. See [Section 6.3](#) and [Section 6.2](#) for more information on this type of command. The minimum and maximum values for this command are 0 and 2, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| `F01AMASGN31` | Assign input 3 to automixer #1. | `F01AMASGN31` |
| `F01AMASGN3?` | Query current automixer assignment for channel 3 . | `F01AMASGN3x` , where `x` is `0`, `1`, or `2` depending on the current automixer assignment for channel 3 . If this command is issued after the example above, then the status message will be `F01AMASGN31` . |
| `F01AMASGN*0` | Assign all mic inputs to no automixer. | `F01AMASGN*äääääää` |
| `F01AMASGN*àààààååå` | Assign inputs 1-4 to automixer #1 and inputs 5-8 to automixer #2. | `F01AMASGN*àààààååå` |
| `F01AMASGN*?` | Query automixer assignment for all mic/line input channels. | `F01AMASGN*abcdefgh` , where each of the letters (`a`, `b`, etc.) is either `ä`, `à`, or `å` depending on the current automixer assignment for each channel. If this command is issued after the example above, then the status message will be `F01AMASGN*àààààååå` . |

# 8.9. `AMAUTO` -- Select Automatic or Manual Gating for each Automixer Input

This command selects or queries the state of automatic or manual automixer gating thresholds for the specified input channel. Automatic thresholds mean that the automixer adaptively determines the gating thresholds based on current speech and noise levels using the gating ratio specified by the [AMGATER](#) command. Manual thresholds mean that the automixer uses the absolute threshold set via the [AMGATET](#) command.

This command is a channel boolean command. See [Section 6.3](#) and [Section 6.1](#) for more information on this type of command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|

| | | |
|---|---|---|
| F01AMAUTO31 | Select automatic thresholds for automixer gating on input channel 3 . | F01AMAUTO31 |
| F01AMAUTO30 | Select manual thresholds for automixer gating on input channel 3 . | F01AMAUTO30 |
| F01AMAUTO32 | Toggle between automatic and manual thresholds for automixer gating on input channel 3 . | F01AMAUTO3x , where x is 0 or 1 depending on whether input channel 3 is currently set for manual or automatic thresholds. |
| F01AMAUTO3? | Query AMAUTO state on input channel 3 . | F01AMAUTO3x , where x is 0 or 1 depending on whether input channel 3 is currently set for manual or automatic thresholds. |
| F01AMAUTO*1 | Select automatic thresholds for automixer gating on input channels 1-8. | F01AMAUTO*11111111 |
| F01AMAUTO*0 | Select manual thresholds for automixer gating on input channels 1-8. | F01AMAUTO*00000000 |
| F01AMAUTO*2 | Toggle between manual and automatic thresholds for automixer gating input channels 1-8. | F01AMAUTO*abcdefgh, where a-h are each 0 or 1 depending on whether each input channel is currently set for manual or automatic thresholds. |
| F01AMAUTO*? | Query AMAUTO state on input channels 1-8. | F01AMAUTO*abcdefgh, where a-h are each 0 or 1 depending on whether each input channel is currently set for manual or automatic thresholds. |

# 8.10. AMBUSID -- Set Automixer Groupings for EF Bus

This command is used to assign one of the two internal automixers to one of the EF Bus automixer groups. For example, consider three EF2280's, each of which has four microphones assigned to Automixer 1 and four microphones assigned to Automixer 2. Now, if each of these EF2280's sets their Automixer 1 to have Bus ID 5, then the three automixers (one from each EF2280) will work as a single automixer containing 12 (3 x 4) microphones. Setting AMBUSID to 0 means that the specified automixer is not grouped on the bus.

The first argument in the AMBUSID command is the automixer number (1 or 2) and the second argument is the Bus ID (0 for none, or 1 through 8). Although this command is a channel integer command, use of the wildcard character for the automixer number is not supported.

This command is a channel integer command. See [Section 6.3](#) and [Section 6.2](#) for more information on this type of command. The minimum and maximum values for this command are 0 and 8, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| F01AMBUSID12 | Configure Automixer 1 to be part of the Bus Automixer having Bus ID 2. | F01AMBUSID12 |

| | | |
|---|---|---|
| `F01AMBUSID20` | Configure Automixer 2 to be part of the Bus Automixer having Bus D 0. This means that the automixer is not part of any Bus Automixer. | `F01AMBUSID20` |
| `F01AMBUSID1?` | Query the current Bus ID of Automixer 1 . | `F01AMBUSID1x` , where `x` is a number between 0 and 8 indicating the current Bus ID of Automixer 1 . |

## 8.11. `AMCHAIR` -- Enable Chairman Mode for Specified Automixer

This command enables, disables, or queries the chairman mode feature for the specified automixer. The first argument in the command specifies the automixer number (1 or 2) and the second argument specifies whether chairman mode should be enabled, disabled, toggled, or queried.

If an `AMCHAIR` command requests that chairman mode be enabled, but the chairman microphone (`AMCHNUM`) does not belong to the specified automixer, the `AMCHAIR` command will fail and return `ERROR#044`. See the `ERROR` command for more information on error messages. See Section 3.5 for more information on dependencies within the automixer commands.

Even though this is a channel boolean command, use of the wildcard character for the automixer number is not supported.

This command is a channel boolean command. See Section 6.3 and Section 6.1 for more information on this type of command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| `F01AMCHAIR10` | Disable chairman mode for Automixer 1 . | `F01AMCHAIR10` |
| `F01AMCHAIR21` | Enable chairman mode for Automixer 2 . | `F01AMCHAIR21` |
| `F01AMCHAIR2?` | Query current setting of chairman mode for Automixer 2 . | `F01AMCHAIR2x` , where `x` is 0 or 1 depending on the current setting of chairman mode for Automixer 2 . |

## 8.12. `AMCHNUM` -- Set Chairman Mic for Specified Automixer

This command sets the chairman microphone for the specified automixer. The first argument of the command specifies the automixer number (1 or 2). The second argument specifies which microphone should be the chairman microphone (1-8) for the automixer.

If chairman mode is enabled (`AMCHAIR`) and the `AMCHNUM` command tries to set a microphone number that does not belong to the specified automixer, the command will fail and generate an `ERROR#045`. See the `ERROR` command for more information on error messages. See Section 3.5 for more information on dependencies within the automixer commands.

Even though this is a channel integer command, use of the wildcard character for the automixer number is not supported.

This command is a channel integer command. See Section 6.3 and Section 6.2 for more information on this type

of command. The minimum and maximum values for this command are 1 and 8, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01AMCHNUM14 | Set microphone 4 as the chairman mic for automixer 1. | F01AMCHNUM14 |
| F01AMCHNUM21 | Set microphone 1 as the chairman mic for automixer 2. | F01AMCHNUM21 |
| F01AMCHNUM1? | Query the current chairman mic for automixer 1. | F01AMCHNUM1x, where x is between 1 and 8 depending on the current chairman mic setting for automixer 1. |

## 8.13. `AMDECAY` -- Set Decay Time for Automixers

This command sets or queries the decay time (in milliseconds) for both automixers. Note that the decay time is set globally for both automixers.

This command is an integer command. See [Section 6.2](#) for more information on this type of command. The minimum and maximum values for this command are 0 and 5000, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01AMDECAY500 | Set automixer decay time to 500 ms. | F01AMDECAY500 |
| F01AMDECAY>500 | Increase automixer decay time by 500 ms. | F01AMDECAYx , where x is between 0 and 5000 depending on the current AMDECAY setting. If this command is issued after the above example, then the status message will be F01AMDECAY1000 |
| F01AMDECAY? | Query automixer decay time. | F01AMDECAYx , where x is between 0 and 5000 depending on the current AMDECAY setting. If this command is issued after the above example, then the status message will be F01AMDECAY1000 |

## 8.14. `AMGATEC` -- Set Automixer Gating Control Mode

This command sets the automixer gating control mode for the specified input channel. The possible modes are:

- `0` - normal gating
- `1` - microphone forced on
- `2` - microphone forced off

This command is a channel integer command. See [Section 6.3](#) and [Section 6.2](#) for more information on this type of command. The minimum and maximum values for this command are 0 and 2, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| `F01AMGATEC10` | Configure microphone input 1 for normal gating. | `F01AMGATEC10` |
| `F01AMGATEC21` | Configure microphone input 2 to be forced on. | `F01AMGATEC21` |
| `F01AMGATEC32` | Configure microphone input 3 to be forced off. | `F01AMGATEC32` |
| `F01AMGATEC3?` | Query current gating control mode for microphone input 3 . | `F01AMGATEC3x` , where `x` is `0`, `1`, or `2` depending on the current gating control mode setting for microphone input 3 . |
| `F01AMGATEC*0` | Configure microphone inputs 1-8 for normal gating. | `F01AMGATEC*ääääääää` |
| `F01AMGATEC*ääääàààå` | Configure microphone inputs 1-3 for normal gating, microphone inputs 4-6 to be forced on, and microphone inputs 7-8 to be forced off. | `F01AMGATEC*ääääàààå` |
| `F01AMGATEC*?` | Query gating control mode for all microphone inputs. | `F01AMGATEC*abcdefgh` , where `a-h` are each `ä`, `à`, or `å` depending on the current setting of the gating control mode for each channel. If this command is issued after the example above, then the status message will be `F01AMGATEC*ääääàààå` . |

## 8.15. `AMGATER` -- Set Automixer Gate Ratio

This command sets the automixer gate ratio (in dB) for the specified input channel. The gate ratio is the ratio of the speech power to noise power required to gate the microphone on. This value is only used if the input is set to automatic gating via the [AMAUTO](#).

This command is a channel integer command. See [Section 6.3](#) and [Section 6.2](#) for more information on this type of command. The minimum and maximum values for this command are 0 and 100, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| `F01AMGATER312` | Set gate ratio for automatic gating threshold to 12 dB for input channel 3 . | `F01AMGATER312` |
| `F01AMGATER3>3` | Increase gate ratio for automatic gating threshold by 3 dB for input channel 3 . | `F01AMGATER3x` , where `x` is between `0` and `100` depending on the current setting of the gate ratio for input channel 3 . If this command is issued after the example above, then the status message will be `F01AMGATER315` . |

| | | |
|---|---|---|
| `F01AMGATER3?` | Query gate ratio for automatic gating threshold for input channel 3 . | `F01AMGATER3x` , where `x` is between `0` and `100` depending on the current setting of the gate ratio for input channel 3 . If this command is issued after the example above, then the status message will be `F01AMGATER315` . |
| `F01AMGATER*12` | Set gate ratio for automatic gating threshold to 12 dB for all input channels. | `F01AMGATER*ÉÉÉÉÉÉÉÉ` |
| `F01AMGATER*ÉÉÉÉôôôô` | Set gate ratio for automatic gating threshold to 12 dB for inputs 1-4 and 15 dB for inputs 5-8. | `F01AMGATER*ÉÉÉÉôôôô` |
| `F01AMGATER*<3` | Decrease gate ratio for automatic gating threshold by 3 dB for all input channels. | `F01AMGATER*abcdefgh` , where `a-h` will each be between `ä` and `Φ` depending on the setting of the gate ratio for each input channel. If this command is issued after the example above, the status message will be `F01AMGATER*ìììììÉÉÉÉ` . |
| `F01AMGATER*?` | Query gate ratio for automatic gating threshold for all input channels. | `F01AMGATER*abcdefgh` , where `a-h` will each be between `ä` and `Φ` depending on the setting of the gate ratio for each input channel. If this command is issued after the example above, the status message will be `F01AMGATER*ìììììÉÉÉÉ` . |

## 8.16. `AMGATET` -- Set Automixer Gate Threshold

This command sets the automixer gate threshold (in dB) for the specified input channel. The gate threshold is the level that the input signal must reach in order to gate the microphone on. This value us only used if the input is set to manual gating via the [AMAUTO](#) command.

This command is a channel integer command. See [Section 6.3](#) and [Section 6.2](#) for more information on this type of command. The minimum and maximum values for this command are 0 and 100, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| `F01AMGATET312` | Set gate threshold for manual gating threshold to 12 dB for input channel 3 . | `F01AMGATET312` |
| `F01AMGATET3>3` | Increase gate threshold for manual gating threshold by 3 dB for input channel 3 . | `F01AMGATET3x` , where `x` is between `0` and `100` depending on the current setting of the gate threshold for input channel 3 . If this command is issued after the example above, then the status message will be `F01AMGATET315` . |
| `F01AMGATET3?` | Query gate threshold for manual gating threshold for input channel 3 . | `F01AMGATET3x` , where `x` is between `0` and `100` depending on the current setting of the gate threshold for input channel 3 . If this command is issued after the example above, then the status message will be `F01AMGATET315` . |

| F01AMGATET*12 | Set gate threshold for manual gating threshold to 12 dB for all input channels. | F01AMGATET*ÉÉÉÉÉÉÉÉ |
|---|---|---|
| F01AMGATET*ÉÉÉÉôôôô | Set gate threshold for manual gating threshold to 12 dB for inputs 1-4 and 15 dB for inputs 5-8. | F01AMGATET*ÉÉÉÉôôôô |
| F01AMGATET*<3 | Decrease gate threshold for manual gating threshold by 3 dB for all input channels. | F01AMGATET*abcdefgh , where a-h will each be between ä and Φ depending on the setting of the gate threshold for each input channel. If this command is issued after the example above, the status message will be F01AMGATET*ìììììÉÉÉÉ . |
| F01AMGATET*? | Query gate threshold for manual gating threshold for all input channels. | F01AMGATET*abcdefgh , where a-h will each be between ä and Φ depending on the setting of the gate threshold for each input channel. If this command is issued after the example above, the status message will be F01AMGATET*ìììììÉÉÉÉ . |

## 8.17. AMGNOM -- Set Global Maximum Number of Open Mics for Bus Automixer

This command sets the global maximum number of open mics (NOM) allowed for the specified bus automixer. The NOM limit is a global limit, meaning that this applies to all bus automixers with the same AMBUSID. In contrast, the AMNOM command is a local limit that applies to the two local automixers in the EF2280.

The first argument of this command specifies the automixer number (1-2) to adjust. The second argument specifies the NOM limit (1-64). Even though this is a channel integer command, use of the wildcard for specifying the automixer number is not supported.

This command is a channel integer command. See Section 6.3 and Section 6.2 for more information on this type of command. The minimum and maximum values for this command are 1 and 64, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| F01AMGNOM13 | Set global NOM for automixer 1 to a maximum of 3 mics. | F01AMGNOM13 |
| F01AMGNOM2? | Query current global NOM limit for automixer 2 . | F01AMGNOM2x , where x is between 1 and 64 depending on the current setting of the global NOM limit for automixer 2 . |

## 8.18. AMHOLD -- Set Automixer Hold Time

This command sets or queries the hold time (in milliseconds) for both automixers. Note that the hold time is set globally for both automixers.

This command is an integer command. See Section 6.2 for more information on this type of command. The minimum and maximum values for this command are 0 and 5000, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| `F01AMHOLD500` | Set automixer hold time to 500 ms. | `F01AMHOLD500` |
| `F01AMHOLD>500` | Increase automixer hold time by 500 ms. | `F01AMHOLDx` , where `x` is between 0 and 5000 depending on the current `AMHOLD` setting. If this command is issued after the above example, then the status message will be `F01AMHOLD1000` |
| `F01AMHOLD?` | Query automixer hold time. | `F01AMHOLDx` , where `x` is between 0 and 5000 depending on the current `AMHOLD` setting. If this command is issued after the above example, then the status message will be `F01AMHOLD1000` |

## 8.19. `AMLMM` -- Set Last Mic On Mode for Specified Automixer

This command sets "last mic on" mode for the specified automixer. The first argument specifies which automixer (1-2) The second parameter specifies the operation of "last mic on" mode and can be one of the following:

- `0` - last mic mode is off
- `1` - manual (use a specific mic)
- `2` - automatic (the last gated mic remains on)

If the `AMLMM` command sets "last mic on" mode to manual, but the last mic number (`AMLMN`) is set to a microphone that does not belong to the specified automixer, then the `AMLMM` command will fail and `ERROR#042` will be generated. See the `ERROR` command for more information on error messages. See Section 3.5 for more information on dependencies within the automixer commands.

Even though this is a channel integer command, use of the wildcard character for specifying the automixer number is not supported.

This command is a channel integer command. See Section 6.3 and Section 6.2 for more information on this type of command. The minimum and maximum values for this command are 0 and 2, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| `F01AMLMM10` | Disable "last mic on" mode for automixer 1. | `F01AMLMM10` |
| `F01AMLMM21` | Set "last mic on" mode to manual for automixer 2 . | `F01AMLMM21` |
| `F01AMLMM12` | Set "last mic on" mode to automatic for automixer 1. | `F01AMLMM12` |
| `F01AMLMM1?` | Query the current setting of "last mic on" mode for automixer 1. | `F01AMLMM1x` , where `x` is 0, 1, or 2 depending on the current setting of "last mic on" mode for automixer 1. If this command is issued after the example above, then the status message will be `F01AMLMM12` . |

## 8.20. `AMLMN` -- Set Microphone That Will Remain On in Manual Last Mic On Mode

This command sets the microphone number that will remain on when "last mic on" mode is set to manual (see the `AMLMM` command). The first argument to this command is the number of the automixer (1-2) that will be adjusted. The second argument is the microphone number (1-8) that should be gated on if no other mics are gated on and `AMLMM` is set to manual. The value of the `AMLMN` command is only valid when `AMLMM` is set to manual.

If "last mic on" moce is set to manual for the specified automixer and the `AMLMN` command attempts to specify a microphone that does not belong to the automixer, then the `AMLMN` command will fail and an `ERROR#043` will be generated. See the `ERROR` command for more information on error messages. See Section 3.5 for more information on dependencies within the automixer commands.

Even though this is a channel integer command, use of the wildcard character for the automixer number is not supported.

This command is a channel integer command. See Section 6.3 and Section 6.2 for more information on this type of command. The minimum and maximum values for this command are 1 and 8, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01AMLMN14 | Set microphone 4 as the manual last mic on for automixer 1. | F01AMLMN14 |
| F01AMLMN21 | Set microphone 1 as the manual last mic on for automixer 2. | F01AMLMN21 |
| F01AMLMN1? | Query the current manual last mic on number mic for automixer 1. | F01AMLMN1x , where x is between 1 and 8 depending on the AMLMN setting for automixer 1. |

## 8.21. `AMNOM` -- Set Local Maximum Number of Open Mics for Automixer

This command sets the local maximum number of open mics (NOM) allowed for the specified automixer. The NOM limit is a local limit, meaning that this limit applies only to the specific Vortex that is is set on. In contrast, the `AMGNOM` command is a global limit that applies to all linked Vortex automixers with the same `AMBUSID`.

The first argument of this command specifies the automixer number (1-2) to adjust. The second argument specifies the NOM limit (1-8) . Even though this is a channel integer command, use of the wildcard for specifying the automixer number is not supported.

This command is a channel integer command. See Section 6.3 and Section 6.2 for more information on this type of command. The minimum and maximum values for this command are 1 and 64, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| F01AMNOM13 | Set local NOM for automixer 1 to a maximum of 3 mics. | F01AMNOM13 |
| F01AMNOM2? | Query current local NOM limit for automixer 2 . | F01AMNOM2x , where x is between 1 and 8 depending on the current setting of the local NOM limit for automixer 2. |

## 8.22. `AMNOMAT` -- Select NOM Attenuation on Each Output

This command enables, disables, or queries NOM attenuation for the specified output (1-8, A-D) . . NOM attenuation is calculated as 10*log(Number of Open Microphones).

This command is a channel boolean command. See [Section 6.3](#) and [Section 6.1](#) for more information on this type of command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| F01AMNOMAT41 | Enable NOM attenuation on output 4 . | F01AMNOMAT41 |
| F01AMNOMATA0 | Disable NOM attenuation on output A. | F01AMNOMATA0 |
| F01AMNOMAT72 | Toggle NOM attenuation status on output 7 . | F01AMNOMAT7x , where x is either 0 or 1 depending on the current mute status of NOM attenuation on the ouptut. |
| F01AMNOMATB? | Query NOM attenuation status of output B. | F01AMNOMATBx , where x is either 0 or 1 depending on the current NOM attenuation status of the output. |
| F01AMNOMAT*1 | Enable NOM attenuation on all outputs (1-8 and A-D) . | F01AMNOMAT*111111111111 |
| F01AMNOMAT*0 | Disable NOM attenuation on all outputs (1-8 and A-D) . | F01AMNOMAT*000000000000 |
| F01AMNOMAT*111111110000 | Enable NOM attenuation on outputs 1-8 and disable NOM attenuation on outputs A-D. | F01AMNOMAT*111111110000 |
| F01AMNOMAT*2 | Toggle status of NOM attenuation on all outputs. | F01AMNOMAT*abcdefghijkl , where each of the letters (a, b, etc.) is either 0 or 1 depending of the current status of NOM attenuation on the corresponding output. If this command was sent after the example above, then the status message would be F01AMNOMAT*000000001111 . |

| | | |
|---|---|---|
| `F01AMNOMAT*?` | Query NOM attenuation status of all outputs. | `F01AMNOMAT*abcdefghijkl`, where each of the letters (a, b, etc.) is either `0` or `1` depending of the current status of the NOM attenuation on the corresponding output. If this command was sent after the example above, then the status message would be `F01AMNOMAT*000000001111`. |

## 8.23. `AMOFFAT` -- Set Off Attenuation for the Specified Automixer

This command sets the off attenuation (in dB) for the specified automixer. Setting this value to 18 would result in the microphone signals being attenuated by 18 dB when gated off. This value is set independently for each of the automixers. The first argument of this command specifies the automixer number (1-2) to adjust. The second argument specifies the off attenuation.

Even though this is a channel integer command, use of the wildcard for specifying the automixer number is not supported.

This command is a channel integer command. See Section 6.3 and Section 6.2 for more information on this type of command. The minimum and maximum values for this command are 1 and 100, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| `F01AMOFFAT118` | Set off attenuation for automixer 1 to 18 dB. | `F01AMOFFAT118` |
| `F01AMOFFAT2?` | Query current off attenuation for automixer 2 . | `F01AMOFFAT2x`, where `x` is between `1` and `100` depending on the current setting of the off attenuation for automixer 2 . |

## 8.24. `AMPRIOR` -- Set Gating Priority for the Specified Mic

This command sets the automixer gating priority for the specified input channel. Priority levels of 1-4 are allowed with 1 being the highest priority and 4 being the lowest.

This command is a channel integer command. See Section 6.3 and Section 6.2 for more information on this type of command. The minimum and maximum values for this command are 1 and 4, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| `F01AMPRIOR11` | Set microphone input 1 to gating priority 1. | `F01AMPRIOR11` |
| `F01AMPRIOR24` | Set microphone input 2 to gating priority 4. | `F01AMPRIOR24` |
| `F01AMPRIOR3?` | Query current gating priority for microphone input 3. . | `F01AMPRIOR3x`, where `x` is between `1` and `4` depending on the current gating priority setting for microphone input 3. . |

| F01AMPRIOR*2 | Set microphone inputs 1-8 to gating priority 2. | F01AMPRIOR*àààààààà |
|---|---|---|
| F01AMPRIOR*àààåççêê | Set gating priority of inputs 1-2 to 1, inputs 3-4 to 2, inputs 5-6 to 3, and 7-8 to 4. | F01AMPRIOR*àààåççêê |
| F01AMPRIOR*? | Query gating priorities for all microphone inputs. | F01AMPRIOR*abcdefgh , where a-h are each between à and ê depending on the current setting of the gating priority for each channel. If this command is issued after the example above, then the status message will be F01AMPRIOR*àààåççêê |

## 8.25. AMREFB -- Set Automixer Reference Bias for the Specified Automixer

This command sets or queries the reference bias for the automixer reference mode feature of the specified automixer. When enabled, the automixer reference feature uses the AEC reference to prevent local microphones from gating on audio from the remote side. The AMREFE command controls the enabled status of this feature. The AMREFB command can be used to bias the AEC reference signal to make the automixer even less likely to gate on remote audio.

The first argument in this command specifies the automixer number (1-2) and the second argument specifies the reference bias in dB.

Even though this is a channel integer command, use of the wildcard character for the automixer number is not supported.

This command is a channel integer command. See Section 6.3 and Section 6.2 for more information on this type of command. The minimum and maximum values for this command are 0 and 20, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| F01AMREFB16 | Set reference bias for automixer 1 to 6 dB. | F01AMREFB16 |
| F01AMREFB2? | Query current reference bias for automixer 2 . | F01AMREFB2x , where x is between 0 and 20 depending on the current setting of the reference bias for automixer 2 . |

## 8.26. AMREFE -- Enable Automixer Reference for Specified Automixer

This command enables, disables, or queries the automixer referece feature for the specified automixer. When enabled, the automixer reference feature uses the AEC reference to prevent local microphones from gating on audio from the remote side. The AMREFE command controls the enabled status of this feature. The AMREFB command can be used to bias the AEC reference signal to make the automixer even less likely to gate on remote audio.

The first argument in the command specifies the automixer number (1-2) and the second argument specifies whether automixer reference mode should be enabled, disabled, toggled, or queried.

Even though this is a channel boolean command, use of the wildcard character for the automixer number is not supported.

This command is a channel boolean command. See Section 6.3 and Section 6.1 for more information on this type of command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01AMREFE10 | Disable automixer reference mode for Automixer 1. | F01AMREFE10 |
| F01AMREFE21 | Enable automixer reference mode for Automixer 2 . | F01AMREFE21 |
| F01AMREFE2? | Query current setting of automixer reference mode for Automixer 2 . | F01AMREFE2x , where x is 0 or 1 depending on the current setting of automixer reference mode for Automixer 2 . |

## 8.27. BAUD -- Set Baud Rate for RS-232 Port

This command sets the baud rate for the rear panel RS-232 port. The baud rate is specified in bits per second (bps). Valid baud rates are 9600, 19200, and 38400. Although, this command returns an acknowledgement, it is likely that you will not receive it, since it is sent at the new baud rate. If you setup your control system to quickly change its RS-232 baud rate after you send this command, then you can probably receive the acknowledgement at the new baud rate.

This command is saved to global non-volatile memory and is not part of a preset. Its value is saved each time it is changed. It will retain its value after power-down. Since this command writes to non-volatile memory, there will be a delay before an acknowledgment is returned.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01BAUD38400 | Set baud rate of rear panel RS-232 port to 38400 bps. | F01BAUD38400 |
| F01BAUD? | Query current baud rate of rear panel RS-232 port. | F01BAUDx , where x is 9600, 19200, or 38400 depending on the current baud rate setting. |

## 8.28. BLAUTO -- Enable Automatic BLDATA Messages

This command sets whether or not BLDATA messages are automatically generated by the Vortex. See the BLDATA command for more information.

This command is a boolean command. See Section 6.1 for more information on this type of command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| F01BLAUTO0 | Disable automatic BLDATA messages. | F01BLAUTO0 |
| F01BLAUTO1 | Enable automatic BLDATA messages. | F01BLAUTO1 |
| F01BLAUTO2 | Toggle enabled status of automatic BLDATA messages. | F01BLAUTOx , where x is 0 or 1 depending on the current setting of BLAUTO. |
| F01BLAUTO? | Query enabled status of automatic BLDATA messages. | F01BLAUTOx , where x is 0 or 1 depending on the current setting of BLAUTO. |

## 8.29. BLDATA -- Request Level Information

This command is used to request "blinking light" data from the EF2280. Blinking light data consists of signal levels for the 12 inputs (1-8, A-D), 12 outputs (1-8, A-D), and 2 AEC references (R1, R2) as well as room gain levels for inputs 1-8, AGC gain levels for inputs 1-8, and AEC state information for inputs 1-8.

The blinking light data can be received in two ways. The first way is by polling the Vortex by sending a BLDATA? command. The second way is to have the Vortex automatically send BLDATA responses via the BLAUTO command.

The most general format of the BLDATA responses generated by the Vortex is as follows.

```
F01BLDATAiiiiiiiiiiiioooooooooooorrggggggggaaaaaaaassssssss
```

Each of the i, o, r, g, a and s characters above represents a single byte of data as shown in the table below.

| BLDATA Bytes | Meaning |
|---|---|
| iiiiiiiiiiii | input levels 1-8, A-D |
| oooooooooooo | output levels 1-8, A-D |
| rr | AEC reference levels R1, R2 |
| gggggggg | room gain 1-8 |
| aaaaaaaa | AGC gain 1-8 |
| ssssssss | AEC state 1-8 |

Each of the i bytes correspond to signal levels on the 12 inputs. The first i byte indicates the signal level on input 1 and the last i byte indicates the signal level on input D. The o bytes work the same way. The first o byte indicates the signal level on output 1 and the last o byte indicates the signal level output D. The first and second r bytes correspond to the signal levels for AEC references R1 and R2, respectively. The first g byte indicates the room gain for input 1 and the last g byte indicates the room gain for input 8. The first a byte indicates the AGC gain for input 1 and the last a byte indicates the AGC gain for input 8. The first s byte indicates the AEC state for input 1 and the last s byte indicates the AEC state for input 8.

The formats of the i, o, r, g, and a bytes are the same as for the binary gain commands (binary values offset by 132). See Section 6.3 and Section 6.4 for more information on the binary format.

The range for the i and o bytes is -100 dB to +20 dB, which corresponds to byte values of 32 to 152. For reference, the LEDs on the front panel signal meter correspond to teh following dB levels (left to right): -20, -12, -7, -3, 0, 3, 9, 20.

The format of the s bytes are also the same as the binary gain commands. The bytes are integers (offset by 132) that represent the current AEC state. The following table shows how the byte values translate to AEC states.

| Byte Value | AEC State Value (Byte Value - 132) | AEC State Description |
|---|---|---|
| 132 | 0 | Idle |
| 133 | 1 | Transmit |
| 134 | 2 | Receive |
| 135 | 3 | Double Talk |

Here's an example BLDATA message to clear things up.

F01BLDATApppppppppèpppppppppppçpppppppppppppppppèêèêèêèêèääàäàåçç

The first twelve bytes (pppppppppèppp) consist of only two values.

p = 0x70 = 112 → 112 - 132 = -20
è = 0x8A = 138 → 138 - 132 = 6

So, the signal at input A is at 6 dB, while the signals at the rest of the inputs are at -20 dB.

The next twelve bytes (pppppppçppppp) consist of only two values.

p = 0x70 = 112 → 112 - 132 = -20
ç = 0x87 = 135 → 135 - 132 = 3

So, the signal at output 7 is at 3 dB, while the signals at the rest of the outputs are at -20 dB.

The next two bytes (pp) consist of only one value.

p = 0x70 = 112 → 112 - 132 = -20

So, the signal level of both AEC references is -20 dB.

The next eight bytes (pppppppp) consist of only one value.

p = 0x70 = 112 → 112 - 132 = -20

So, we know that the room gain for inputs 1-8 is -20 dB.

The next eight bytes (èèèèèèèè) consist of only one value.

è = 0x8A = 138 → 138 - 132 = 6

So, we know that the AGC gain for inputs 1-8 is 6 dB.

The last eight bytes (äääàäàçç) consist of four different values.

ä = 0x84 = 132 → 132 - 132 = 0 = Idle

à = 0x85 = 133 → 133 - 132 = 1 = Transmit

å = 0x86 = 134 → 134 - 132 = 2 = Receive

ç = 0x87 = 135 → 135 - 132 = 3 = Double Talk

So, we know that the AEC on inputs 1-2 are in idle, inputs 3-4 are in transmit, inputs 5-6 are in receive, and inputs 7-8 are in double talk.

In the above description, the data in the `BLDATA` command consists of 50 bytes (12 input levels + 12 output levels + 2 AEC reference levels + 8 room gain levels + 8 AGC gain levels + 8 AEC states). It is possible to tell the EF2280 to only send a subset of this information. There are two reasons you might want to do this. First, it makes parsing the data easier since you can ask for only the data that you are interested in. Second, when you decrease the amount of data being sent, the EF2280 can send the messages faster. This means that if you are trying to implement a signal level meter, you can get faster refresh rates by asking the EF2280 to only send the data you're interested in.

Suppose you were only interested in signal levels on inputs A-D. You could send the following command.

```
F01BLINFO000000000111100000000000000000000000000000000000000000
```

In the above `BLINFO` command, there are 1's in the positions of the data that we want and 0's in the positions of the data that we don't want. Now, when we send a `BLDATA?` command or enable `BLAUTO`, the Vortex sends messages of the form:

```
F01BLDATAiiii
```

Where the `iiii` bytes are the signal levels for inputs A-D. Looking at the `BLDATA` response above, we see that it consists of 14 bytes (`F01BLDATA` = 9 bytes, `iiii` = 4 bytes, carriage return = 1 byte). If all of the `BLINFO` bits were set to 1, the command would take a total of 60 bytes (`F01BLDATA` = 9 bytes, data = 50 bytes, carriage return = 1 byte). In automatic mode (`BLAUTO`), the Vortex sends `BLDATA` commands at a constant bitrate, so this truncated command would be sent 60/14 = 4.29 times more often than the full version. This results in a much faster refresh rate.

## 8.30. `BLINFO` -- Select Information to be Reported in `BLDATA`

This command controls which bytes are sent in the `BLDATA` command. See the description of the `BLDATA` command for more information on the `BLINFO` command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

## 8.31. `BROAD2` -- Broadcast Arbitrary Command Strings to RS-232 Port

This command is used to broadcast arbitrary commands to the RS-232 port for controlling other devices (not necessarily other Vortex devices). For example, `'F01BROAD2:ATDT4048921180'''''` would cause the Vortex

to send out `ATDT4048921180` via its RS-232 port. If a modem was connected to the RS-232 port, this command would cause the modem to dial the Polycom Installed Voice Business Group in Atlanta.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01BROAD2:ATDT4048921180 | Sends the string `ATDT4048921180` out the RS-232 port. | F01BROAD2 |

## 8.32. `BROADA` -- Broadcast Commands to Other Connected Devices

This command is used to broadcast commands to other connected Polycom devices via the EF Bus and/or ASPI Bus. For example, the command `'F01BROADA:B02PHONE1 '` would cause device ID 1 to send a command to the EF2241 at device ID 2 to tell it to take its phone off-hook. The EF2241 at device ID 2 would respond by taking its device off-hook and sending an acknowledgement.

This command is usually used in macros when one needs to have a macro on one device trigger an action on another device.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01BROADA:B02PHONE1 | Sends the command `B02PHONE2` over the digital bus. | F01BROADA<br><br>B02PHONE1, this second status message is from the EF2241 at device ID 2 when it takes its phone off hook. |

## 8.33. `BUSREF` -- Set Which AEC Reference is Placed on EF Bus

This command sets or queries whic AEC reference (if any) is placed on the EF Bus. Setting `BUSREF` to '0' means that no reference is placed on the EF Bus. Setting `BUSREF` to '1' means that AEC reference 1 (R1) is placed on the bus. Setting `BUSREF` to '2' means that AEC reference 2 (R2) is placed on the bus.

The AEC references are created in the main matrix via outputs R1 and R2.

Only one Vortex can put a reference on the EF Bus at a time. If more than one Vortex places its reference on the EF Bus, then an `'ERROR#093'` will occur.

This command is an integer command. See [Section 6.2](#) for more information on this type of command. The minimum and maximum values for this command are 0 and 2, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01BUSREF2 | Put AEC Reference 2 (R2) on the EF Bus. | F01BUSREF2 |
| F01BUSREF0 | Set Vortex to not place any AEC reference on the EF Bus. | F01BUSREF0 |
| F01BUSREF? | Query which AEC reference this Vortex is placing on the EF Bus. | F01BUSREFx , where x is 0, 1, or 2 depending on which AEC reference (if any) is currently being placed on EF Bus. |

## 8.34. `CGATE` -- Query Camera Gating Status Information

This command is used to query the gating status of the microphone inputs . It is identical to the <u>GATE</u> command except that any given microphone must be gated on for a specified hold time in order for it to be considered "on" by the `CGATE` command. The command can be used to query the status of individual microphones, or the wildcard character can be used to query the status of all the microphone inputs. A typical use for this command is to control camera pointing based on microphone activity. The added hold time prevents the camera from jumping too quickly between positions when there are short amounts of signal present.

The <u>CGATET</u> specifies the hold time used for the gating decisions in this command. The <u>CGATEEN</u> can be used to have gating information sent automatically instead of having to poll this command.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01CGATE3? | Query camera gating status of microphone input 3 . | F01CGATE3x , , where x is 0 or 1 depending on whether the input is gated off or gated on, respectively. |
| F01CGATE*? | Query camera gating status of all microphone inputs. | F01CGATE*abcdefgh , where each letter (a, b, etc.) is either 0 or 1 depending on whether the corresponding microphone input is gated off or gated on, respectively. |

## 8.35. `CGATEEN` -- Enable Automatic Camera Gating Messages

This command controls whether or not camera gating information messages are sent automatically. If `CGATEEN` is enabled, then a `CGATE*?` query is performed each time the camera gating status of any microphone changes.

For example, if no microphones are currently gated on and `CGATEEN` is enabled, then if microphone input 3 gates on, the following status message will be automatically generated:

F01CGATE*00100000

See the <u>CGATE</u> command for more information.

This command is a boolean command. See [Section 6.1](#) for more information on this type of command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01CGATEEN0 | Disable automatic camera gating messages. | F01CGATEEN0 |
| F01CGATEEN1 | Enable automatic camera gating messages. | F01CGATEEN1 |
| F01CGATEEN2 | Toggle enabled state of automatic camera gating messages. | F01CGATEENx , where x is either 0 or 1 depending on whether automatic camera gating messages are currently disabled or enabled. |
| F01CGATEEN? | Query enabled state of automatic camera gating messages. | F01CGATEENx , where x is either 0 or 1 depending on whether automatic camera gating messages are currently disabled or enabled. |

## 8.36. `CGATET` -- Set Camera Gating Hold Time

This command sets the hold time (in milliseconds) for the camera gating (CGATE) feature.

This command is an integer command. See Section 6.2 for more information on this type of command. The minimum and maximum values for this command are 100 and 5000, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01CGATET500 | Set camera gating hold time to 500 ms. | F01CGATET500 |
| F01CGATET>500 | Increase camera gating hold time by 500 ms. | F01CGATETx , where x is between 100 and 5000 depending on the current setting of CGATET. If this command is issued after the example above, the status message will be F01CGATET1000 . |
| F01CGATET? | Query camera gating hold. | F01CGATETx , where x is between 100 and 5000 depending on the current setting of CGATET. If this command is issued after the example above, the status message will be F01CGATET1000 . |

# 8.37. DELAYO -- Set Output Delay

This command sets the amount of output delay (in tenths of milliseconds) on each of the twelve output channels (1-8, A-D) . Use of the wildcard character (*) for specifying the channel is not supported for this command. The output delay can be separately enabled or disabled via the DELAYOE command.

The output delay feature is only available on Rev F and later EF2280's. All Rev F and later EF2280's have the Polycom logo on the front panel. EF2280 revisions earlier than Rev F have the ASPI Digital logo on the front panel. For units where this command is not supported, it will always return a status message indicating that the output delay is set to 0.

This command is an integer command. See Section 6.2 for more information on this type of command. The minimum and maximum values for this command are 0 and 3400, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01DELAYO21500 | Set output delay on output channel 2 to 150 ms (1500 tenths of milliseconds). | F01DELAYO21500 |
| F01DELAYO2>1700 | Increase output delay on output channel 2 by 170 ms (1700 tenths of milliseconds.) | F01DELAYO2x , where x is between 0 and 3400 depending on the current output delay setting for output channel 2 . If this command is issued after the above example, then the status message will be F01DELAYO23200 . |

| | | F01DELAYO2x , where x is between 0 and 3400 depending on the current output delay setting for output channel 2. If this command is issued after the above example, then the status message will be F01DELAYO23200 . |
|---|---|---|
| F01DELAYO2? | Query current output delay on output channel 2 . | |

## 8.38. DELAYOE -- Enable Output Delay

This command sets or queries the enabled state of the output delay for each of the twelve output channels (1-8, A-D) .

This command is a channel boolean command. See [Section 6.3](#) and [Section 6.1](#) for more information on this type of command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| F01DELAYOE41 | Enable output delay on output 4 . | F01DELAYOE41 |
| F01DELAYOEA0 | Disable output delay on output A . | F01DELAYOEA0 |
| F01DELAYOE72 | Toggle enabled status of output delay on output 7 . | F01DELAYOE7x , where x is either 0 or 1 depending on the current enabled status of the output delay on the output. |
| F01DELAYOEC? | Query enabled status of output delay on output C . | F01DELAYOECx , where x is either 0 or 1 depending on the current enabled status of the output delay on the output. |
| F01DELAYOE*1 | Enable output delay on all outputs (1-8, A-D) . | F01DELAYOE*111111111111 |
| F01DELAYOE*0 | Disable output delay on all outputs (1-8, A-D) . | F01DELAYOE*000000000000 |
| F01DELAYOE*111111110000 | Enable output delay on outputs 1-8 and disable output delay on outputs A-D. | F01DELAYOE*111111110000 |
| F01DELAYOE*2 | Toggle enabled status of output delay on all outputs. | F01DELAYOE*abcdefghijkl , where each of the letters (a, b, etc.) is either 0 or 1 depending of the current enabled status of the output delay on the corresponding output. If this command was sent after the example above, then the status message would be F01DELAYOE*000000001111 . |
| F01DELAYOE*? | Query enabled status of output delay on all outputs. | F01DELAYOE*abcdefghijkl , where each of the letters (a, b, etc.) is either 0 or 1 depending of the current enabled status of the output delay on the corresponding output. If this command was sent after the example above, then the status message would be F01DELAYOE*000000001111 . |

## 8.39. `DSPAUTO` -- Enable Automatic `DSPLOAD` Status Messages

This command sets or queries whether or not <u>DSPLOAD</u> messages will be automatically generated whenever the DSP utilization changes.

This command is a boolean command. See <u>Section 6.1</u> for more information on this type of command.

| Example | Description | Status Message |
|---|---|---|
| F01DSPAUTO1 | Enable automatic DSPLOAD messages. | F01DSPAUTO1 |
| F01DSPAUTO0 | Disable automatic DSPLOAD messages. | F01DSPAUTO0 |
| F01DSPAUTO2 | Toggle automatic DSPLOAD messages. | F01DSPAUTOx , where x is 0 or 1 depending on the current setting of the DSPAUTO paramter. |
| F01DSPAUTO? | Query the value of the DSPAUTO parameter. | F01DSPAUTOx , where x is 0 or 1 depending on the current setting of the DSPAUTO paramter. |

## 8.40. `DSPLOAD` -- Query Percentage of Variable DSP Resources Used

This command queries the current percentage used of the available variable DSP resources. When this number reaches 100 percent, no more variable DSP veatures may ben enabled. An 'ERROR#060' will be generated if a command attempts to exceend 100 percent utilization.

If the <u>DSPAUTO</u> feature is enabled, then a DSPLOAD status message will be generated automatically any time the percent utilization changes.

| Example | Description | Status Message |
|---|---|---|
| F01DSPLOAD? | Query percentage of variable DSP resources used. | F01DSPLOADx , where x is between 0 and 99 and indicates the percentage of the variable DSP resources being used. |

## 8.41. `ERROR` -- Enable or Disable Error Messages

This command sets or queries whether or not error messages for non-fatal errors are reported via RS-232 and the digital bus.

If an error is generated and error messages are enabled, a status message will be automatically generated of the form 'F01ERROR#xxx ', where xxx is a three digit number indicating the error code. The following table lists the non-fatal error messages that can be generated by the Vortex.

| Error Number | Description |
|---|---|
| ERROR#001 | Unrecognized command. |
| ERROR#002 | Syntax error in command. |
| ERROR#004 | Attempt to change parameter that is locked via <u>NVLOCK</u> or <u>FPLOCK</u>. |
| ERROR#005 | Attempt to unlock <u>NVLOCK</u> or <u>FPLOCK</u>, but invalid password given. |

| | |
|---|---|
| ERROR#040 | This error occurs as the result of an AMASGN command. It happens when the command would remove a microphone from an automixer where it is assigned as the "last mic on" and the automixer is set to manual "last mic on" mode. |
| ERROR#041 | This error occurs as the result of an AMASGN command. It happens when the command would remove a microphone from an automixer where it assigned as the chairman imcrophone and the automixer has chairman mode enabled. |
| ERROR#042 | This error occurs as the result of an AMLMM command. It happens when the command attempts to set "last mic on" mode to manual, but the last mic number is set to a microphone that does not belong to the specified automixer. |
| ERROR#043 | This error occurs as the result of an AMLMN command. It happens when the command attempts to set a "last mic on" number that does not belong to the specified automixer while "last mic on" mode is set to manual. |
| ERROR#044 | This error occurs as the result of an AMCHAIR command. It happens when the command tries to enable chairman mode, but the chairman microphone is set to a microphone that does not belong to the specified automixer. |
| ERROR#045 | This error occurs as the result of an AMCHAIR command. It happens when the command tries to set the chairman microphone to a microphone that does not belong to the specified automixer while chairman mode is enabled. |
| ERROR#060 | This error occurs when the user has attempted to utilize more variable DSP resources than are available. This may occur if too many matrix crosspoints are unmuted or too many parametric EQ filters are enabled. The DSPLOAD command can query the current percentage utilization of DSP resources. |
| ERROR#070 | This error occurs as a result of a macro or preset execution command (PRESETX, PRESETQ, MACROX, MACROQ). It indicates that the macro or preset requested was empty. |
| ERROR#071 | This error occurs as a result of a PRESETW or PRESETK command. In the case of PRESETW it indicates that an attempt was made to write to a factory preset, which is not allowed. In the case of PRESETK, it indicates that an attempt was made to delete a factory preset, which is not allowed. |
| ERROR#072 | This error occurs as a result of a PRESETQ command. It indicates that one or more commands in the preset had errors during execution. |
| ERROR#073 | This error occurs as a result of a MACROA command when there are already too many commands in the macro. The limit is 256 commands per macro. |
| ERROR#074 | This error occurs in response to a MACROA or a logic pin (LIA, LIH, etc.) command. If the command written to the macro or logic pin is illegal, then this error occurs. |
| ERROR#075 | This error occurs when a command attempts to write non-volatile memory, but non-volatile memory is password protected via NVLOCK. |
| ERROR#076 | This error occurs when attempting to add a macro command (either during an upload or while writing a single macro) and the Vortex runs out of room in non-volatile memory. As a result, the entire macro is discarded. If this occurs during an upload, all previous macros (before the one that caused the error) are written successfully. |
| ERROR#077 | This error occurs when attempting to add a command to a macro, but the Vortex is not in the process of writing a macro. This may happen if you forget to do a MACROS or UMACROS command, or if you had an ERROR#076 and continued trying to add commands. |
| ERROR#090 | This error indicates an EF Bus hardware handshaking error. This could happen if bussed units are not powered on simultaneously. Powering the units up simultaneously should remove the error. |

| | |
|---|---|
| `ERROR#091` | This error indicates an EF Bus software handshaking error. This could happen if bussed units are not powered on simultaneously. Powering the units up simultaneously should remove the error. |
| `ERROR#092` | This error indicates an EF Bus ID conflict. See the Vortex user manual for information on valid device IDs. |
| `ERROR#093` | This error indicates an EF Bus reference conflict. This error occurs when more than one Vortex is placing its AEC reference on the bus. Use the `BUSREF` command to remove the extra reference(s) in order to resolve the conflict. |

This command is a boolean command. See Section 6.1 for more information on this type of command.

This command is saved to global non-volatile memory and is not part of a preset. Its value is saved each time it is changed. It will retain its value after power-down. Since this command writes to non-volatile memory, there will be a delay before an acknowledgment is returned.

| Example | Description | Status Message |
|---|---|---|
| `F01ERROR1` | Enable error messages. | `F01ERROR1` |
| `F01ERROR0` | Disable error messages mode. | `F01ERROR0` |
| `F01ERROR2` | Toggle error message mode. | `F01ERRORx` , where `x` is `0` or `1` depending on the current state of error mode. |
| `F01ERROR?` | Query the state of error mode. | `F01ERRORx` , where `x` is `0` or `1` depending on the current state of error mode. |

# 8.42. `FADERGIL` -- Set Fader Gain of Line Inputs as a Group

This command sets the fader gains of all the line inputs simultaneously. This is different from using a wildcard, `*` with the `FADERI` command because that command sets all the mic and line inputs together, while the `FADERGIL` sets only the line inputs.

This command will generate acknowledgements from each of the line inputs separately.

This command is an integer command. See Section 6.2 for more information on this type of command. The minimum and maximum values for this command are -100 and 20, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| `F01FADERGIL7` | Set fader gain of all line inputs to 7 dB. | `F01FADERIA7`<br>`F01FADERIB7`<br>`F01FADERIC7`<br>`F01FADERID7` |

| F01FADERGIL>3 | Increase fader gains on all line inputs by 3 dB. | F01FADERIAa<br>F01FADERIBb<br>F01FADERICc<br>F01FADERIDd ,<br>where a, b, c, and d are the new values of the fader gains for each of the line inputs. If this command was issued after the example above, then the status messages would be:<br>F01FADERIA10<br>F01FADERIB10<br>F01FADERIC10<br>F01FADERID10 . |
|---|---|---|
| F01FADERGIL? | Query fader gains on all line inputs. | F01FADERIAa<br>F01FADERIBb<br>F01FADERICc<br>F01FADERIDd ,<br>where a, b, c, and d are the new values of the fader gains for each of the line inputs. If this command was issued after the example above, then the status messages would be:<br>F01FADERIA10<br>F01FADERIB10<br>F01FADERIC10<br>F01FADERID10 . |

## 8.43. FADERGIM -- Set Gain of Microphone Inputs as a Group

This command sets the fader gains of all the microphone inputs simultaneously. This is different from using a wildcard, * with the FADERI command because that command sets all the mic and line inputs together, while the FADERGIM sets only the mic inputs.

This command will generate acknowledgements from each of the mic inputs separately.

This command is an integer command. See Section 6.2 for more information on this type of command. The minimum and maximum values for this command are -100 and 20, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| F01FADERGIM7 | Set fader gain of all mic inputs to 7 dB. | F01FADERI17<br>F01FADERI27<br>F01FADERI37<br>F01FADERI47<br>F01FADERI57<br>F01FADERI67<br>F01FADERI77<br>F01FADERI87 |

| | | |
|---|---|---|
| `F01FADERGIM>3` | Increase fader gain on all mic inputs by 3 dB. | `F01FADERI1a`<br>`F01FADERI2b`<br>`F01FADERI3c`<br>`F01FADERI4d`<br>`F01FADERI5e`<br>`F01FADERI6f`<br>`F01FADERI7g`<br>`F01FADERI8h` , where `a`, `b`, `c`, `d`, `e`, `f`, `g`, and `h` are the new values of the fader gains for each of the mic inputs. If this command was issued after the example above, then the status messages would be:<br>`F01FADERI110`<br>`F01FADERI210`<br>`F01FADERI310`<br>`F01FADERI410`<br>`F01FADERI510`<br>`F01FADERI610`<br>`F01FADERI710`<br>`F01FADERI810` . |
| `F01FADERGIM?` | Query fader gains on all mic inputs. | `F01FADERI1a`<br>`F01FADERI2b`<br>`F01FADERI3c`<br>`F01FADERI4d`<br>`F01FADERI5e`<br>`F01FADERI6f`<br>`F01FADERI7g`<br>`F01FADERI8h` , where `a`, `b`, `c`, `d`, `e`, `f`, `g`, and `h` are the new values of the fader gains for each of the mic inputs. If this command was issued after the example above, then the status messages would be:<br>`F01FADERI110`<br>`F01FADERI210`<br>`F01FADERI310`<br>`F01FADERI410`<br>`F01FADERI510`<br>`F01FADERI610`<br>`F01FADERI710`<br>`F01FADERI810` . |

## 8.44. `FADERI` -- Set Input Gain Fader

The input channels of the Vortex pass through an analog gain stage before reaching the analog to digital converter. The gain of this stage is adjustable via a the GAINI command. Once in the digital domain, there is an additional gain stage, referred to as a fader. The fader gain is adjustable by the `FADERI` command. The GAINI setting should be used for calilbration in order to maximize the resolution and quality of the signal at the analog to digital converter. The `FADERI` command is provided as a way to do volume control on the inputs without affecting the calibration.

This command was introduced in firmware version 2.5.0.

This command is a channel integer command. See [Section 6.3](#) and [Section 6.2](#) for more information on this type of command. The minimum and maximum values for this command are -100 and 20, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01FADERI210 | Set fader gain on input 2 to 10 dB. | F01FADERI210 |
| F01FADERI2? | Query the fader gain on input 2 . | F01FADERI2x , where x is a number between 0 and 20 , depending on the current setting of the fader gain on input channel 2 . |
| F01FADERI*6 | Set fader gain on all inputs (1-8, A-D) to 6 dB. | F01FADERI*èèèèèèèèèèèè |
| F01FADERI*ääääääääìììì | Set fader gain on input channels 1-8 to 0 dB and gain on input channels A-D to 9 dB. | F01FADERI*ääääääääìììì |
| F01FADERI*? | Query fader gain on all inputs (1-8, A-D) . | F01FADERI*abcdefghijkl, where a-l are each between ä and ÿ, depending on the current setting of the fader gains for each of the twelve input channels. |

## 8.45. `FLOW` -- Set Flow Control Mode for RS-232 Port

This command sets the flow control mode for the rear panel RS-232 port. The valid settings are:

| Command Value | Description |
|---------------|-------------|
| 0 | No flow control. |
| 1 | Hardware flow control. |
| 2 | Auto-detect hardware flow control or no flow control. |

We recommend using hardware flow control whenever possible, especially when using higher baud rates.

This command is saved to global non-volatile memory and is not part of a preset. Its value is saved each time it is changed. It will retain its value after power-down. Since this command writes to non-volatile memory, there will be a delay before an acknowledgment is returned.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01FLOW1 | Set rear panel RS-232 port to use hardware flow control. | F01FLOW1 |
| F01FLOW? | Query flow control setting of rear panel RS-232 port. | F01FLOWx , where x is 0, 1, or 2 depending on the current flow control setting. |

## 8.46. `FPLOCK` -- Lock/Unlock Front Panel

This command controls the front panel lock feature. When `FPLOCK` is enabled, the user may not make any

changes to the system via the front panel. The system settings will still be viewable on the LCD, but the user will get an error message if he tries to change them. When `FPLOCK` is disabled, the user has full access to the system settings via the front panel.

Usage of this command is similar to other boolean commands (see [Section 6.1](#)) except that when disabling this feature, the password must be supplied for the command to work. If an incorrect password is supplied, `'ERROR#005'` will be generated. The examples below illustrate the correct usage. The examples assume that the password has been set to `'aspi'` (the default). The front panel password can be changed via the [FPPSWD](#) command.

This command is saved to global non-volatile memory and is not part of a preset. Its value is saved each time it is changed. It will retain its value after power-down. Since this command writes to non-volatile memory, there will be a delay before an acknowledgment is returned.

| Example | Description | Status Message |
|---------|-------------|----------------|
| `F01FPLOCK1` | Lock the front panel to prevent users from changing the settings. | `F01FPLOCK1` |
| `F01FPLOCK0,yumyum` | Attempt to unlock the front panel by using an invalid password (yumyum). | `F01ERROR#005` -- since an invalid password was given, an error message was generated. If error messages have been disabled (via the [ERROR](#) command), then no status message will be generated. |
| `F01FPLOCK0,aspi` | Unlock the front panel by using the correct password. | `F01FPLOCK0` |
| `F01FPLOCK?` | Query the locked status of the front panel. | `F01FPLOCKx` , where `x` is `0` or `1` depending on whether the front panel is unlocked or locked, respectively. |

# 8.47. `FPPSWD` -- Change Front Panel Password

This command sets or queries the front panel password. This password is used in conjunction with the [FPLOCK](#) command. The front panel must be unlocked (`'FPLOCK0'`) in order to use this command to set or query the front panel password. If the front panel is locked, then this command will result in `'ERROR#004'`. The examples below assume that the front panel is unlocked.

| Example | Description | Status Message |
|---------|-------------|----------------|
| `F01FPPSWDmonkey` | Set front panel password to `'monkey'`. | `F01FPPSWDmonkey` |
| `F01FPPSWD?` | Query the current front panel password. | `F01FPPSWDmonkey` |

# 8.48. `GAINGIL` -- Set Gain of Line Inputs as a Group

This command sets the gains of all the line inputs simultaneously. This is different from using a wildcard, `*` with the [GAINI](#) command because that command sets all the mic and line inputs together, while the `GAINGIL` sets only the line inputs.

This command will generate acknowledgements from each of the line inputs separately.

This command is an integer command. See [Section 6.2](#) for more information on this type of command. The

minimum and maximum values for this command are 0 and 20, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01GAINGIL7 | Set gain of all line inputs to 7 dB. | F01GAINIA7<br>F01GAINIB7<br>F01GAINIC7<br>F01GAINID7 |
| F01GAINGIL>3 | Increase incoming gain on all line inputs by 3 dB. | F01GAINIAa<br>F01GAINIBb<br>F01GAINICc<br>F01GAINIDd ,<br>where a, b, c, and d are the new values of each of the line inputs. If this command was issued after the example above, then the status messages would be:<br>F01GAINIA10<br>F01GAINIB10<br>F01GAINIC10<br>F01GAINID10 . |
| F01GAINGIL? | Query gains on all line inputs. | F01GAINIAa<br>F01GAINIBb<br>F01GAINICc<br>F01GAINIDd ,<br>where a, b, c, and d are the new values of each of the line inputs. If this command was issued after the example above, then the status messages would be:<br>F01GAINIA10<br>F01GAINIB10<br>F01GAINIC10<br>F01GAINID10 . |

## 8.49. GAINGIM -- Set Gain of Microphone Inputs as a Group

This command sets the gains of all the microphone inputs simultaneously. This is different from using a wildcard, * with the GAINI command because that command sets all the mic and line inputs together, while the GAINGIM sets only the mic inputs.

This command will generate acknowledgements from each of the mic inputs separately.

This command is an integer command. See Section 6.2 for more information on this type of command. The minimum and maximum values for this command are 0 and 30, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| | | |

| | | |
|---|---|---|
| F01GAINGIM7 | Set gain of all mic inputs to 7 dB. | F01GAINI17<br>F01GAINI27<br>F01GAINI37<br>F01GAINI47<br>F01GAINI57<br>F01GAINI67<br>F01GAINI77<br>F01GAINI87 |
| F01GAINGIM>3 | Increase incoming gain on all mic inputs by 3 dB. | F01GAINI1a<br>F01GAINI2b<br>F01GAINI3c<br>F01GAINI4d<br>F01GAINI5e<br>F01GAINI6f<br>F01GAINI7g<br>F01GAINI8h ,<br>where a, b, c, d, e, f, g, and h are the new values of each of the mic inputs. If this command was issued after the example above, then the status messages would be:<br>F01GAINI110<br>F01GAINI210<br>F01GAINI310<br>F01GAINI410<br>F01GAINI510<br>F01GAINI610<br>F01GAINI710<br>F01GAINI810 . |
| F01GAINGIM? | Query gains on all mic inputs. | F01GAINI1a<br>F01GAINI2b<br>F01GAINI3c<br>F01GAINI4d<br>F01GAINI5e<br>F01GAINI6f<br>F01GAINI7g<br>F01GAINI8h ,<br>where a, b, c, d, e, f, g, and h are the new values of each of the mic inputs. If this command was issued after the example above, then the status messages would be:<br>F01GAINI110<br>F01GAINI210<br>F01GAINI310<br>F01GAINI410<br>F01GAINI510<br>F01GAINI610<br>F01GAINI710<br>F01GAINI810 . |

# 8.50. GAINI -- Set Input Gain

The input channels of the Vortex pass through an analog gain stage before reaching the analog to digital converter. The gain of this stage is adjustable via a digitally controlled analog trim pot. It is important that this

gain is set correctly in order to maximize the resolution and quality of the signal at the analog to digital converter. This command lets you adjust the gain of these input gain stages. For volume control, the `FADERI` command can be used to adjust the signal level in the digital domain without affecting the calibration.

The following table shows the relationship between the input gain settings (via `GAINI` and `MIC`) and the nominal level expected at each of the inputs.

| Input Channel | `MIC` Setting | `GAINI` Setting (dB) | Expected level at Input (dBu) |
|---|---|---|---|
| 1-8 | 1 (mic level) | 0 to 30 | 0 to -30 |
| 1-8 | 0 (line level) | 0 to 30 | 33 to 3 |
| A-D | N/A (line level) | 0 to 20 | 0 to -20 |

This command is a channel integer command. See Section 6.3 and Section 6.2 for more information on this type of command. The minimum and maximum values for this command are 0 and 20 or 30, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| `F01GAINI210` | Set gain on input 2 to 10 dB. | `F01GAINI210` |
| `F01GAINI2?` | Query the gain on input 2 . | `F01GAINI2xB01GAINITxS01GAINIAxQ01GAINIAxT01GAINITx`, where `x` is a number between 0 and 20 , depending on the current setting of the gain on input channel 2 . |
| `F01GAINI*6` | Set gain on all inputs (1-8, A-D) to 6 dB. | `F01GAINI*èèèèèèèèèèèè` |
| `F01GAINI*äääääääääìììì` | Set gain on input channels 1-8 to 0 dB and gain on input channels A-D to 9 dB. | `F01GAINI*äääääääääìììì` |
| `F01GAINI*?` | Query gain on all inputs (1-8, A-D) . | `F01GAINI*abcdefghijkl`, where `a-l` are each between ä and ÿ, depending on the current setting of the gains for each of the twelve input channels. |

## 8.51. `GAINO` -- Set Output Gain

This command sets or queries the gain (in dB) of the twelve output channels (1-8, A-D) .

This command is a channel integer command. See Section 6.3 and Section 6.2 for more information on this type of command. The minimum and maximum values for this command are -100 and 20, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be

restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01GAINO210 | Set gain on output channel 2 to 10 dB. | F01GAINO210 |
| F01GAINO2? | Query the gain on output channel 2 . | F01GAINO2x , , where x is a number between -100 and 20, depending on the current setting of the gain on output channel 2 . |
| F01GAINO*6 | Set the gain on all output channels (1-8, A-D) to 6 dB. | F01GAINO*èèèèèèèèèèèè |
| F01GAINO*ääääääääìììì | Set gain on output channels 1-8 to 0 dB and gain on output channels A-D to 9 dB. | F01GAINO*äääääääìììì |
| F01GAINO*? | Query gain settings on all output channels (1-8, A-D) . | F01GAINO*abcdefghijkl, where a-l are each between <space> and ÿ, depending on the current setting of the gains for each of the twelve output channels (1-8, A-D). |

## 8.52. GATE -- Query Gating Status Information

This command is used to query the gating status of the microphone inputs. The command can be used to query the status of individual microphones, or the wildcard character can be used to query the status of all the microphone inputs.

The GATEEN can be used to have gating information sent automatically instead of having to poll this command.

Also see the CGATE command, which only reports microphones that have been gated on for a specified hold time.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01GATE3? | Query gating status of microphone input 3 . | F01GATE3x , where x is 0 or 1 depending on whether the input is gated off or gated on, respectively. |
| F01GATE*? | Query gating status of all microphone inputs. | F01GATE*abcdefgh , where each letter (a, b, etc.) is either 0 or 1 depending on whether the corresponding microphone input is gated off or gated on, respectively. |

## 8.53. GATEEN -- Enable Automatic Gating Messages

This command controls whether or not gating information messages are sent automatically. If GATEEN is enabled, then a GATE*? query is performed each time the gating status of any microphone changes. For example, if no microphones are currently gated on and GATEEN is enabled, then if microphone input 3 gates on, the following status message will be automatically generated:

F01GATE*00100000

See the GATE command for more information.

This command is a boolean command. See [Section 6.1](#) for more information on this type of command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01GATEEN0 | Disable automatic gating messages. | F01GATEEN0 |
| F01GATEEN1 | Enable automatic gating messages. | F01GATEEN1 |
| F01GATEEN2 | Toggle enabled state of automatic gating messages. | F01GATEENx , where x is either 0 or 1 depending on whether automatic gating messages are currently disabled or enabled. |
| F01GATEEN? | Query enabled state of automatic gating messages. | F01GATEENx , where x is either 0 or 1 depending on whether automatic gating messages are currently disabled or enabled. |

## 8.54. `GMUTEO` -- Mute All Outputs

The name of this command is an abbreviation of "Global Mute Outputs." Enabling this option causes all of the physical outputs to be muted (outputs 1-8, A-D) . This muting is independent of the normal output mute command, [MUTEO](#). This command is provided so that the control program can implement a "saftey mute" feature. The idea is that if the user accidentally misconfigures the Vortex and feedback begins to occur, they can press the saftey mute to instantly mute all outputs before they damage their equipment or ears.

This command is a boolean command. See [Section 6.1](#) for more information on this type of command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01GMUTEO1 | Enable global output mute. | F01GMUTEO1 |
| F01GMUTEO0 | Disable global output mute. | F01GMUTEO0 |
| F01GMUTEO2 | Toggle global output mute. | F01GMUTEOx , where x is 0 or 1 depending on the current state of the global output mute. |
| F01GMUTEO? | Query acknowledgement mode. | F01GMUTEOx , where x is 0 or 1 depending on the current state of the global output mute. |

## 8.55. `ID` -- Set Device ID

This command sets or queries the Device ID for the Vortex. Typically, the device ID is set from the front panel of the Vortex, so this command is not normally used. There are restrictions involving having devices with the same ID linked on EF Bus -- see the Vortex Reference Manual for more details.

This command is an integer command. See [Section 6.2](#) for more information on this type of command. The minimum and maximum values for this command are 0 and 7, respectively.

This command is saved to global non-volatile memory and is not part of a preset. Its value is saved each time it is changed. It will retain its value after power-down. Since this command writes to non-volatile memory, there will be a delay before an acknowledgment is returned.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01ID3 | Change the device ID from 1 to 3 | F03ID3 |
| F01ID? | Query current device ID. | F01ID1 |

## 8.56. `LABEL` -- Set or Query one of the Device Labels

This command sets or queries one of the many labels stored in the device. The `LABEL` command has the following syntax:

`iLABELx,y`

where `i` is the device type and ID, `x` is the label specifier, and `y` is the label text (to set the label) or a `?` character (to query the label). The label specifiers are as follows:

| Label Specifier | Description |
|-----------------|-------------|
| D | A label for the device itself. |
| SG | A label for the signal generator. |
| I1-I8, IA-ID | Labels for the input channels. |
| O1-O8, OA-OD | Labels for the output channels. |
| W, X, Y, Z | Labels for matrix outputs to EF Bus. |
| PB0-PB7, WB0-WB7, XB0-XB7, YB0-YB7, ZB0-ZB7 | Labels for inputs from EF Bus to submatrices. |
| PM0-PM1, WM0-WM2, XM0-XM2, YM0-YM2, ZM0-ZM2 | Labels for outputs from EF Bus submatrices to main matrix. |
| R1, R2 | Labels for AEC references. |
| G0-G3 | Labels for logic input groups. |
| LI1-LI24 | Labels for logic input pins |
| LO1-LO20 | Labels for logic output pins |
| P0-P47 | Labels for user presets. |
| M0-M255 | Labels for user macros. |

The text of each label can be up to 16 characters long. Labels for presets 0-15 are read-only.

This command is saved to global non-volatile memory and is not part of a preset. Its value is saved each time it is changed. It will retain its value after power-down. Since this command writes to non-volatile memory, there will be a delay before an acknowledgment is returned.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01LABELOB,Zone 2 Speaker | Set label of output B to "Zone 2 Speaker". | F01LABELOB,Zone 2 Speaker |
| F01LABELOB,? | Query label for output B . | F01LABELOB,<string> , where <string> is the current label assigned to output B . |

## 8.57. `LAGC` -- Enable or Disable Line Input Automatic Gain Control

This command sets or queries the status of the Automatic Gain Control (AGC) algorithm on input channels A-D .

This command was introduced in firmware version 2.5.0.

This command is a channel boolean command. See [Section 6.3](#) and [Section 6.1](#) for more information on this type of command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01LAGCB1 | Enable AGC on input channel B. | F01LAGCB1 |
| F01LAGCA0 | Disable AGC on input channel A. | F01LAGCA0 |
| F01LAGCC2 | Toggle AGC state on input channel C . | F01LAGCCx , where x is 0 or 1 depending on the current state of the AGC on input channel C . |
| F01LAGCD? | Query AGC state on input channel D . | F01LAGCDx , where x is 0 or 1 depending on the current state of the AGC on input channel D . |
| F01LAGC*1 | Enable AGC on input channels A-D. | F01LAGC*1111 |
| F01LAGC*0 | Disable AGC on input channels A-D. | F01LAGC*0000 |
| F01LAGC*2 | Toggle AGC state on input channels A-D. | F01LAGC*abcd , where a-d are each 0 or 1 depending on the current state of the AGC for each of the four line input channels. |
| F01LAGC*? | Query AGC state on input channels A-D. | F01LAGC*abcd , where a-d are each 0 or 1 depending on the current state of the AGC for each of the four line input channels. |

## 8.58. `LAGCLINKAB` -- Enable or Disable Stereo AGC Linking on Inputs A and B

This command links the line input AGC of inputs A and B into a stereo pair. The overall volume level of the stereo signal will be analyzed, and the same gain will be applied to both channels.

This command has some side effects on the way the other line input AGC commands work. When `LAGCLINKAB` is first enabled, the settings for [LAGC](#), [LAGCMAX](#), [LAGCMIN](#), and [LAGCRATE](#) on channel A will be copied to channel B. The original settings for channel B will not be restored after `LAGCLINKAB` is disabled. Acknowledgements will be sent for any changes to channel B. From then on, any changes for these commands for either channel A or B will be set for both channels, and acknowledgements will be sent for both channels.

This command is a boolean command. See [Section 6.1](#) for more information on this type of command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01LAGCLINKAB1 | Enable stereo AGC linking on inputs A and B. | F01LAGCLINKAB1 |

| | | |
|---|---|---|
| `F01LAGCLINKAB0` | Disable stereo AGC linking on inputs A and B. | `F01LAGCLINKAB0` |
| `F01LAGCLINKAB2` | Toggle stereo AGC linking on inputs A and B. | `F01LAGCLINKABx` , where `x` is `0` or `1` depending on the current state of stereo AGC linking on inputs A and B. |
| `F01LAGCLINKAB?` | Query status of stereo AGC linking on inputs A and B. | `F01LAGCLINKABx` , where `x` is `0` or `1` depending on the current state of stereo AGC linking on inputs A and B. |

## 8.59. `LAGCLINKCD` -- Enable or Disable Stereo AGC Linking on Inputs C and D

This command links the line input AGC of inputs C and D into a stereo pair. The overall volume level of the stereo signal will be analyzed, and the same gain will be applied to both channels.

This command has some side effects on the way the other line input AGC commands work. When `LAGCLINKCD` is first enabled, the settings for [LAGC](#), [LAGCMAX](#), [LAGCMIN](#), and [LAGCRATE](#) on channel C will be copied to channel D. The original settings for channel D will not be restored after `LAGCLINKCD` is disabled. Acknowledgements will be sent for any changes to channel D. From then on, any changes for these commands for either channel C or D will be set for both channels, and acknowledgements will be sent for both channels.

This command is a boolean command. See [Section 6.1](#) for more information on this type of command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| `F01LAGCLINKCD1` | Enable stereo AGC linking on inputs C and D. | `F01LAGCLINKCD1` |
| `F01LAGCLINKCD0` | Disable stereo AGC linking on inputs C and D. | `F01LAGCLINKCD0` |
| `F01LAGCLINKCD2` | Toggle stereo AGC linking on inputs C and D. | `F01LAGCLINKCDx` , where `x` is `0` or `1` depending on the current state of stereo AGC linking on inputs C and D. |
| `F01LAGCLINKCD?` | Query status of stereo AGC linking on inputs C and D. | `F01LAGCLINKCDx` , where `x` is `0` or `1` depending on the current state of stereo AGC linking on inputs C and D. |

## 8.60. `LAGCMAX` -- Set Maximum Allowed Line Input AGC Gain

This command sets the maximum gain that the AGC can apply on input channels A-D . For example, if `LAGCMAX` is set to `10`, then the AGC for that channel can apply a maximum of 10 dB of gain to the input signal.

This command was introduced in firmware version 2.5.0.

This command is a channel integer command. See [Section 6.3](#) and [Section 6.2](#) for more information on this type of command. The minimum and maximum values for this command are 0 and 15, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| F01LAGCMAXA3 | Set AGC maximum gain on input channel A to 3 dB. | F01LAGCMAXA3 |
| F01LAGCMAXB? | Query the AGC maximum gain on input channel B. | F01LAGCMAXBx , where x is a number between 0 and 15, depending on the current setting of the AGC maximum gain on input channel B. |
| F01LAGCMAX*6 | Set AGC maximum gain on input channels A-D to 6 dB. | F01LAGCMAX*èèèè |
| F01LAGCMAX*ääìì | Set AGC maximum gain on input channels A-B to 0 dB and AGC maximum gain on input channels C-D to 9 dB. | F01LAGCMAX*ääìì |
| F01LAGCMAX*? | Query AGC maximum gain on input channels A-D. | F01LAGCMAX*abcd , where a-d are each between ä and É, depending on the current setting of the AGC maximum gain for each of the four line input channels. |

## 8.61. LAGCMIN -- Set Minimum Allowed Line Input AGC Gain

This command sets the minimum gain that the AGC can apply on input channels A-D . For example, if LAGCMIN is set to -10, then the AGC for that channel can apply a minimum of -10 dB of gain to the input signal.

This command was introduced in firmware version 2.5.0.

This command is a channel integer command. See Section 6.3 and Section 6.2 for more information on this type of command. The minimum and maximum values for this command are -15 and 0, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| F01LAGCMINA-3 | Set AGC minimum gain on input channel A to -3 dB. | F01LAGCMINA-3 |
| F01LAGCMINB? | Query the AGC minimum gain on input channel B. | F01LAGCMINBx , where x is a number between -15 and 0, depending on the current setting of the AGC minimum gain on input channel B. |
| F01LAGCMIN*-6 | Set AGC minimum gain in input channels A-D to -6 dB. | F01LAGCMIN*~~~~ |
| F01LAGCMIN*ääüü | Set AGC minimum gain on input channels A-B to 0 dB and AGC minimum gain on input channels C-D to -3 dB. | F01LAGCMIN*ääüü |
| F01LAGCMIN*? | Query AGC minimum gain on input channels A-D. | F01LAGCMIN*abcd , where a-d are each between u and ä, depending on the current setting of the AGC minimum gain for each of the line input channels. |

## 8.62. LAGCRATE -- Set Ramp Rate of Line Input AGC

This command sets or queries the maximum rate at which the AGC can increase or decrease the gain of the signals on input channels A-D . The ramp rate is expressed in dB/sec.

This command was introduced in firmware version 2.5.0.

This command is a channel integer command. See Section 6.3 and Section 6.2 for more information on this type of command. The minimum and maximum values for this command are 1 and 5, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| F01LAGCRATEA3 | Set AGC ramp rate on input channel A to 3 dB/sec. | F01LAGCRATEA3 |
| F01LAGCRATEB? | Query the AGC ramp rate on input channel B. | F01LAGCRATEBx , where x is a number between 1 and 5, depending on the current setting of the AGC ramp rate on input channel B. |
| F01LAGCRATE*5 | Set AGC ramp rate on input channels A-D to 5 dB/sec. | F01LAGCRATE*;ëëëë |
| F01LAGCRATE*ààêê | Set AGC ramp rate on input channels A-B to 1 dB/sec and AGC ramp rate on input channels C-D to 4 dB/sec. | F01LAGCRATE*ààêê |
| F01LAGCRATE*? | Query AGC ramp rate on input channels A-D. | F01LAGCRATE*abcd , where a-d are each between à and ë, depending on the current setting of the AGC ramp rate for each of the line input channels. |

## 8.63. LI -- Query State of Logic Inputs

This command returns the current state of the logic inputs. There are 24 logic inputs, so an array of 24 boolean values is returned with the first value indicating the state of the first logic input, the second value indicating the state of the second logic input, and so on.

This command is saved to global non-volatile memory and is not part of a preset. Its value is saved each time it is changed. It will retain its value after power-down. Since this command writes to non-volatile memory, there will be a delay before an acknowledgment is returned.

| Example | Description | Status Message |
|---|---|---|
| F01LI*? | Query current state of logic inputs. | F01LI*111010100010111100001110 , this is an exapmle response, the actual values will depend on the actual states on the logic inputs. |

## 8.64. LIA -- Assign Action for when Logic Input is Activated

This function assigns a single command to be executed when a given logic input changes from the inactive state to the active state. Typically, the inactive state is logic high ('1') and the active state is logic low ('0'). This is commonly referred to as active low. Acitve low is considered normal because a closed switch would ground the

input and a closed switch would normally be considered active. This polarity setting can be changed via the LIP command.

A single command can be assigned to each of three conditions occurring on the logic pin: a change to the active state, a change to the inactive state, and a repeating command when the pin is held in the active state. The LIA command assigns a single command to the logic pin that is executed when the logic pin changes to the active state.

Although only one command can be assigned to the state change, the command may be a MACROX or MACROQ command. Since up to 256 commands can be stored in each macro, this gives the effect of having up to 256 commands execute when the logic pin changes state. The command associated with the state change can also be a PRESETX or PRESETQ. This makes it easy to reconfigure the device for different rooms based on external logic settings.

The BROADA is also useful in logic pin assignments. The BROADA command transmits a command to another device on the EF Bus, so you can effectively make a logic pin state change on one device cause an action to occur on another device.

ERROR#074 will be generated if the assigned command is one that writes non-volatile memory, and the pin will not actually be assigned.

This command is saved to global non-volatile memory and is not part of a preset. Its value is saved each time it is changed. It will retain its value after power-down. Since this command writes to non-volatile memory, there will be a delay before an acknowledgment is returned.

| Example | Description | Status Message |
|---|---|---|
| F01LIA4,MUTEI11 | Assign the command MUTEI11 to occur when logic input pin 4 changes from the inactive to active state. The MUTEI11 mutes input channel 1 | F01LIA4,MUTEI11 |
| F01LIA5,MACROX23 | Assign the command MACROX23 to occur when logic input pin 5 changes from the inactive state to the active state. The MACROX23 command executes macro 23, which can contain up to 256 other commands. | F01LIA5,MACROX23 |
| F01LIA6,BROADA:F02MUTEI11 | Assign the command BROADA:F02MUTEI11 to occur when logic input pin 6 changes from the inactive state to the active state. This command causes the EF2280 at ID 2 to mute its input channel 1 when logic pin 6 on the EF2280 at ID 1 changes from the inactive to active state. | F01LIA6,BROADA:F02MUTEI11 |

| | | |
|---|---|---|
| `F01LIA7,` | Assign a blank command to occur when logic input pin 7 changes from the inactive state to the active state. Assigning a blank command means no action will take place as a result of this state change. | `F01LIA7,` |
| `F01LIA4,?` | Query the command associated with the active state of logic pin 4. | `F01LIA4,<string>`, where `<string>` is the command string that will be executed when logic input 4 is activated. If this query was given after the example for logic input 4 above, the response would be `F01LIA4,MUTEI11`. |

## 8.65. `LID` -- Assign Action for when Logic Input is Deactivated

This function assigns a single command to be executed when a given logic input changes from the active state to the inactive state. Typically, the inactive state is logic high ('1') and the active state is logic low ('0'). This is commonly referred to as active low. Acitve low is considered normal because a closed switch would ground the input and a closed switch would normally be considered active. This polarity setting can be changed via the LIP command.

A single command can be assigned to each of three conditions occurring on the logic pin: a change to the active state, a change to the inactive state, and a repeating command when the pin is held in the active state. The `LID` command assigns a single command to the logic pin that is executed when the logic pin changes to the inactive state.

Although only one command can be assigned to the state change, the command may be a MACROX or MACROQ command. Since up to 256 commands can be stored in each macro, this gives the effect of having up to 256 commands execute when the logic pin changes state. The command associated with the state change can also be a PRESETX or PRESETQ. This makes it easy to reconfigure the device for different rooms based on external logic settings.

The BROADA is also useful in logic pin assignments. The BROADA command transmits a command to another device on the EF Bus, so you can effectively make a logic pin state change on one device cause an action to occur on another device.

`ERROR#074` will be generated if the assigned command is one that writes non-volatile memory, and the pin will not actually be assigned.

See the description of the LIA for examples on how to use the `LID` command. The syntax for this command is exactly the same.

This command is saved to global non-volatile memory and is not part of a preset. Its value is saved each time it is changed. It will retain its value after power-down. Since this command writes to non-volatile memory, there will be a delay before an acknowledgment is returned.

## 8.66. `LIH` -- Assign Action for when Logic Input is Held

This function assigns a single command to be executed repeatedly when a given logic input is held in the active

state. Typically, the inactive state is logic high ('1') and the active state is logic low ('0'). This is commonly referred to as active low. Acitve low is considered normal because a closed switch would ground the input and a closed switch would normally be considered active. This polarity setting can be changed via the LIP command.

A single command can be assigned to each of three conditions occurring on the logic pin: a change to the active state, a change to the inactive state, and a repeating command when the pin is held in the active state. The LIH command assigns a single command to the logic pin that is executed repeatedly when the logic pin is held in the active state. This command is useful for volume ramping controls, especially when combined with the relative operators (< and >) of integer commands (see Section 6.2).

Although only one command can be assigned to the state change, the command may be a MACROX or MACROQ command. Since up to 256 commands can be stored in each macro, this gives the effect of having up to 256 commands execute when the logic pin changes state. The command associated with the state change can also be a PRESETX or PRESETQ. This makes it easy to reconfigure the device for different rooms based on external logic settings.

The BROADA is also useful in logic pin assignments. The BROADA command transmits a command to another device on the EF Bus, so you can effectively make a logic pin state change on one device cause an action to occur on another device.

ERROR#074 will be generated if the assigned command is one that writes non-volatile memory, and the pin will not actually be assigned.

See the description of the LIA for examples on how to use the LIH command. The syntax for this command is exactly the same. One additional example is given here: using the LIH command for volume control.

This command is saved to global non-volatile memory and is not part of a preset. Its value is saved each time it is changed. It will retain its value after power-down. Since this command writes to non-volatile memory, there will be a delay before an acknowledgment is returned.

| Example | Description | Status Message |
|---|---|---|
| F01LIH8,GAINIA>3 | Assign the command GAINIA>3 to occur repeatedly when logic pin 8 is held in the active state. The GAINIA>3 command causes the gain on input channel A to increase by 3 dB each time the command is executed. | F01LIH8,GAINIA>3 |

## 8.67. LIEN -- Enable Automatic Logic Input Status Messages

This command controls whether or not logic input status messages are sent automatically. If LIEN is enabled, then a logic input query (LI*?) is performed any time any of the states change on the logic input pins. For example, if all logic inputs are currently in the low state ('0') and LIEN is enabled, then if logic input 12 changes state to '1', the following status message will be generated:

F01LI*000000000001000000000000

A status message is only generated if there is a state change. This helps keep data traffic to a minimum.

This command is a boolean command. See Section 6.1 for more information on this type of command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01LIEN1 | Enable automatic logic input status messages. | F01LIEN1 |
| F01LIEN0 | Disable automatic logic input status messages. | F01LIEN0 |
| F01LIEN2 | Toggle automatic logic input status messages. | F01LIENx , where x is 0 or 1 depending on the current setting of LIEN. |
| F01LIEN? | Query automatic logic input status messages. | F01LIENx , where x is 0 or 1 depending on the current setting of LIEN. |

## 8.68. LIG -- Configure Logic Input Pins Into a Group

This conmmand configures which logic input pins are in a group. Logic groups allow events to happen when certain combinations of inputs are present on the logic pins. For example, if a logic group was defined that consisted of three logic input pins (pin 1, pin 2, and pin 3), then a total of eight combinations ($2^3$) are possible, and a command or macro can be assigned to each combination. This feature is particularly useful in room-combining applications. It can also be used to make controls based on rotary switches that output binary, BCD, or Gray codes.

Commands can be assigned to the configurations of a logic group via the LIN command.

This command is saved to global non-volatile memory and is not part of a preset. Its value is saved each time it is changed. It will retain its value after power-down. Since this command writes to non-volatile memory, there will be a delay before an acknowledgment is returned.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01LIG2,000011110000000000000000 | Configure Logic Group 2 to contain logic input pins 5-8. | F01LIG2,000011110000000000000000 |
| F01LIG2,011010010000010000000000 | Configure Logic Group 2 to contain logic input pins 2, 3, 5, 8, and 13. Note that the pins in a logic group do not have to be contiguous. | F01LIG2,011010010000010000000000 |
| F01LIG2,? | Query which pins are in logic group 2. | F01LIG2,abcdefghijklmnopqrstuvwx , where a-x are each 0 or 1 depending on which logic input pins are assigned to logic group 2. |

## 8.69. LIK -- Delete One or All Logic Input Pin Commands

This command "kills" or deletes all commands for a given logic input pin. In other words, the commands associated with LIA, LID, and LIH will be deleted for the specified logic input. A wildcard character can also be specified for the logic input pin, in which case the commands for all logic input pins will be deleted.

Using this command is more efficient than deleting one command or pin at a time since this command deletes them all at once and requires fewer writes to non-volatile memory.

This command is saved to global non-volatile memory and is not part of a preset. Its value is saved each time it is changed. It will retain its value after power-down. Since this command writes to non-volatile memory, there will be a delay before an acknowledgment is returned.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01LIK5 | Delete all the commands associated with logic input pin 5. This one command is equivalent to executing the commands 'F01LIA5, ', 'F01LID5, ', and 'F01LIH5, ' except that this command is more efficient. | F01LIK5 |
| F01LIK* | Delete all the commands associated with all logic input pins. This one command is equivalent to executing the commands 'F01LIA1, 'F01LID1, ', and 'F01LIH1, ' for each logic input, except that this command is much more efficient. | F01LIK* |

# 8.70. `LIM` -- Mask Logic Input Pins

This command is used to mask (i.e., disable) logic input pins. The mask consists of 24 fields (one for each logic input) that can each be set to `0` or `1`. If the mask bit for a given logic input is set to `1`, then the logic input pin works normally. If the mask bit for a given logic input is set to `0`, then the logic input is disabled. When a logic input is disabled, any commands assigned to the logic pin via `LIA`, `LID`, and `LIH` will not be executed.

If a logic pin is disabled and then re-enabled at some later time and the logic input has changed state from when it was disabled, the command associated with the state change will be executed. For example, if a logic input pin is high when it is disabled, and it is low when it is re-enabled, the command defined via `LID` (or `LIA` if the polarity is changed via `LIP`) will be executed. However, if the pin changes state multiple times while it is disabled, multiple commands will not be executed; only the difference in state when the logic input is re-enabled is considered.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01LIM100101101111011111111111 | Mask (disable) logic inputs 2, 3, 5, 8, and 13. | F01LIM100101101111011111111111 |
| F01LIM? | Query current logic input mask. | F01LIMabcdefghijklmnopqrstuvwx where a-x are each 0 or 1 depending on the current state of the logic input mask. |

# 8.71. `LIN` -- Assign Command to Logic Input Group

This command assigns a command to be executed when a group of logic input pins is in a certain configuration. Logic groups are defined via the `LIG` command.

As an example, assume that the command F01LIG2,000001111100000000000000 has been sent to the device. This command configures logic input pins 6-10 to be part of logic group 2. Now, we send the command F01LIN2,10,MACROX25. This command configures logic input group 2 to execute the command MACROX25 when logic inputs 7 and 9 are active ('1') and logic inputs 6, 8, and 10 are inactive ('0'). The first number in the

`LIN` command specifies which logic group to be affected. The second number specifies the configuraion. In this case, the number '10' translates to '01010' in binary. There are zeros in the positions for the first, third, and last bits -- these correspond to logic inputs 6, 8, and 10 for our case. Similarly, the ones in the second and fourth positions correspond to bits 7 and 9. The last part of the command specifies the command to be executed with the pins enter this configuration.

This command is saved to global non-volatile memory and is not part of a preset. Its value is saved each time it is changed. It will retain its value after power-down. Since this command writes to non-volatile memory, there will be a delay before an acknowledgment is returned.

| Example | Description | Status Message |
|---|---|---|
| `F01LIN2,10,MACROX25` | Configure logic input group 2 to execute `MACROX25` when the pins in logic input group 2 correspond to the decimal value 10. | `F01LIN2,10,MACROX25` |
| `F01LIN1,7,?` | Query the command associated with decimal configuration 7 on logic input group 1. | `F01LIN1,7,<string>` , where <string> corresponds to the command that is associated with decimal configuration 7 on logic input group 1. |

## 8.72. `LIP` -- Set Polarity for Logic Inputs

This command sets the polarity for the 24 logic input pins. Setting the polarity for a given pin to `0` indicates that the polarity should be normal (active low). Setting the polarity for a given pin to `1` indicates that the polarity should be inverted (active high). Active low is considered normal because a closed switch would ground the input, and a closed switch would normally be considered active.

This command affects the operation of the `LIA`, `LID`, and `LIH` commands. If a logic input pin's polarity is reversed, the operation of these commands is reversed.

By default, the polarity for all pins is set to normal (active low).

This command is saved to global non-volatile memory and is not part of a preset. Its value is saved each time it is changed. It will retain its value after power-down. Since this command writes to non-volatile memory, there will be a delay before an acknowledgment is returned.

| Example | Description | Status Message |
|---|---|---|
| `F01LIP000000000000000000001111` | Set logic inputs 1-20 to normal (active low) and logic inputs 21-24 to inverted (active high). | `F01LIP000000000000000000001111` |

## 8.73. `LO` -- Query or Set Status of Logic Output Pins

This command sets or queries the current state of the logic outputs. There are 20 logic outputs, so an array of 20 boolean values is required or returned, with the first value indicating the state of the first logic output, the second value indicating the state of the second logic output, and so on.

If a logic output pin has conditions driving it (via the `LOA` and `LOD` commands), those conditions will override any

settings imposed by the `LO` command.

If a logic pin is masked via the `LOM` command, its status will still be affected by the `LO` command.

| Example | Description | Status Message |
|---|---|---|
| `F01LO*000000000000000000000` | Set all logic outputs low. | `F01LO*000000000000000000000` |
| `F01LO*10101010101010101010` | Set even numbered logic output pins low and odd numbered logic output pins high. | `F01LO*10101010101010101010` |
| `F01LO*?` | Query current state of logic outputs | `F01LO*abcdefghijklmnopqrst`, where `a-t` are each `0` or `1` depending on the current state of the corresponding logic output. |

## 8.74. `LOA` -- Define Behavior for Logic Output Activated State

This command is used to define the conditions under which a given logic output pin goes into the active state. By default the logic output pins are active high. Active high is the default because it would light an LED connected to the output when the output was in the active state. The polarity of the logic output pins can be changed with the `LOP` command.

The syntax of this command allows one to use one of the boolean channel commands to determine the state of the logic output pin. The commands that are valid for use with `LOA` are `AEC`, `AGC`, `AMNOMAT`, `CGATE`, `GATE`, `MUTEO`, `MUTEI`, `NC`, `MIC`, and `PHANTOM`. Each command must be specified using the wildcard character, `*`, for the channel number.

Each character in the array string is compared to the status of the channels in the Vortex. A `1` or a `0` in the string means that the status for the corresponding channel must match that character for the condition to be true. A period character, `.`, in the string is a don't care, meaning that the condition will be true no matter what the status for that channel is. A `+` or `-` character works as an OR function, with `+` corresponding to a `1` in the channel status and `-` corresponding to a `0`. If any `+` or `-` characters are in the string, at least one channel's status must match for the condition to be true.

Consider the following example. In this example, we also use the `LOD` command, which is just like the `LOA` command except that it sets the condition for the logic output to be in the deactivated state. Normally, you must set both the `LOA` and `LOD` commands in order to make use of a logic output pin. In our example, we issue the following commands:

```
F01LOA10,MUTEI*1100++--....
F01LOD10,MUTEI*00......00++
```

These two commands set the conditions for activation and deactivation of logic output 10. Under these conditions, logic output 10 will be activated if:

- (inputs 1 and 2 are muted) AND
- (inputs 3 and 4 are unmuted) AND
- (input 5 is muted OR input 6 is muted OR input 7 is unmuted OR input 8 is unmuted)

Logic output 10 will be deactivated if:

- (input 1, input 2, input A, and input B are unmuted) AND
- (input C OR input D is muted)

An error condition of `ERROR#074` will be generated if the assigned command is invalid.

This command is saved to global non-volatile memory and is not part of a preset. Its value is saved each time it is changed. It will retain its value after power-down. Since this command writes to non-volatile memory, there will be a delay before an acknowledgment is returned.

| Example | Description | Status Message |
|---------|-------------|----------------|
| `F01LOA10,MUTEI*1100++--....` | Configure the conditions for activation of logic output 10 as described in the example above. | `F01LOA10,MUTEI*1100++--....` |
| `F01LOA7,` | Delete conditions for activation of logic output 7. | `F01LOA7,` |

# 8.75. `LOD` -- Define Behavior for Logic Output Deactivated Status

This command is used to define the conditions under which a given logic output pin goes into the deactive state. The syntax and behavior of this command is identical to that of the `LOA` command. Please refer to the description of the `LOA` command for more information, including examples.

This command is saved to global non-volatile memory and is not part of a preset. Its value is saved each time it is changed. It will retain its value after power-down. Since this command writes to non-volatile memory, there will be a delay before an acknowledgment is returned.

# 8.76. `LOEN` -- Enable Automatic Logic Output Status Messages

This command controls whether or not logic output status messages are sent automatically. If `LOEN` is enabled, then a logic output query (`LO*?`) is performed any time any of the states change on the logic output pins. For example, if all logic outputs are currently in the low state ('0') and `LOEN` is enabled, then if logic output 12 changes state to '1', the following status message will be generated:

`F01LO*00000000000100000000`

A status message is only generated if there is a state change. This helps keep data traffic to a minimum.

This command is a boolean command. See [Section 6.1](#) for more information on this type of command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| `F01LOEN1` | Enable automatic logic output status messages. | `F01LOEN1` |
| `F01LOEN0` | Disable automatic logic output status messages. | `F01LOEN0` |
| `F01LOEN2` | Toggle automatic logic output status messages. | `F01LOENx` , where `x` is `0` or `1` depending on the current setting of `LOEN`. |

| | | |
|---|---|---|
| `F01LOEN?` | Query automatic logic output status messages. | `F01LOENx` , where `x` is `0` or `1` depending on the current setting of `LOEN`. |

## 8.77. `LOK` -- Delete One or All Logic Output Pin Commands

This command "kills" or deletes all commands for a given logic output pin. In other words, the commands associated with `LOA` and `LOD` will be deleted for the specified logic output. A wildcard character can also be specified for the logic output pin, in which case the commands for all logic output pins will be deleted.

Using this command is more efficient than deleting one command or pin at a time since this command deletes them all at once and requires fewer writes to non-volatile memory.

This command is saved to global non-volatile memory and is not part of a preset. Its value is saved each time it is changed. It will retain its value after power-down. Since this command writes to non-volatile memory, there will be a delay before an acknowledgment is returned.

| Example | Description | Status Message |
|---|---|---|
| `F01LOK5` | Delete all the commands associated with logic output pin 5. This one command is equivalent to executing the commands '`F01LOA5,` ' and '`F01LOD5,` ' except that this command is more efficient. | `F01LOK5` |
| `F01LOK*` | Delete all the commands associated with all logic output pins. This one command is equivalent to executing the commands '`F01LOA1,` ' and '`F01LOD1,` ' for each logic output, except that this command is much more efficient. | `F01LOK*` |

## 8.78. `LOM` -- Mask Logic Output Pins

This command is used to mask (i.e., disable) logic output pins. The mask consists of 20 fields (one for each logic output) that can each be set to `0` or `1`. If the mask bit for a given logic output is set to `1`, then the logic output pin works normally. If the mask bit for a given logic output is set to `0`, then the logic output is disabled. When a a logic output is disabled, it is "frozen" in whatever state it is in when it is disabled. The output pin's state will not change when the condition that drives it (via `LOA` and `LOD`) changes. However, any changes to that pin caused by setting the values with an `LO` command will affect the output pin value. For instance, you could disable (mask) a pin and then explicitly set its value to low with the `LO` command.

When a logic output pin is re-enabled, it will be updated to the current status of whatever condition drives it (via (`LOA` and `LOD`).

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| `F01LOM10010110111101111111` | Mask (disable) logic outputs 2, 3, 5, 8, and 13. | `F01LOM10010110111101111111` |
| `F01LOM?` | Query current logic output mask. | `F01LOMabcdefghijklmnopqrst` , where `a-t` are each `0` or `1` depending on the current state of the logic output mask. |

## 8.79. `LOP` -- Set Polarity for Logic Outputs

This command sets the polarity for the 20 logic input pins. Setting the polarity for a given pin to `1` indicates that the polarity should be normal (active high). Setting the polarity for a given pin to `0` indicates that the polarity should be inverted (active low). Active high is considered normal because it would light an LED connected to an output pin if the output pin was in the active state.

This command affects the operation of the <u>LOA</u> and <u>LOD</u> commands. If a logic input pin's polarity is reversed, the operation of these commands is reversed.

By default, the polarity for all pins is set to normal (active high).

This command is saved to global non-volatile memory and is not part of a preset. Its value is saved each time it is changed. It will retain its value after power-down. Since this command writes to non-volatile memory, there will be a delay before an acknowledgment is returned.

| Example | Description | Status Message |
|---------|-------------|----------------|
| `F01LOP11111111111111110000` | Set logic outputs 1-16 to normal (active high) and logic outputs 17-20 to inverted (active low). | `F01LOP11111111111111110000` |
| `F01LOP?` | Query current polarity settings for logic outputs. | `F01LOPabcdefghijklmnopqrst` , where `a-t` are each `0` or `1` depending on the current state of the logic output polarity for the corresponding logic output pin. |

## 8.80. `MACROA` -- Add Command to Current Macro

This command adds a command to the macro currently in progress. This is used in conjunction with the <u>MACROS</u> and <u>MACROW</u> commands to create a new macro.

The syntax of this command specifies a macro number, a comma, and then the command to be added to the macro. The command to be added to the macro can be any valid command except the following:

- any `PRESET` command
- any `MACRO` command
- any logic input or output command

If an attempt to assign an invalid command to a macro is made (via <u>MACROA</u>), then the error condition `ERROR#074` will be generated.

An error will be generated if the specified command is not a valid command. However, an error will not be generated if the specified command is valid, but its data is invalid. For example:

`F01MACROA25,LAYDOWNTHEBOOGIE`

Would return an error because it does not contain a valid command. However:

```
F01MACROA62,MUTEOMYMOTHERINLAW
```

Would not return an error because it contains a [MUTEO](#) command. An error would be returned when the macro is executed with a [MACROX](#) or [MACROQ](#) command and it tries to execute the [MUTEO](#) command with strange data.

The proper sequence for creating a macro (macro number 112 in this example) is as follows.

```
F01MACROS112
F01MACROA112,MUTEI*0
F01MACROA112,GAINI*0
F01MACROA112,MUTEO*0
F01MACROW112
```

This defines macro number 112 to unmute all inputs, set all input gains to 0 dB, and unmute all outputs. It is acceptable if another command is sent in between these commands (such as `F01NC*?` ) as long as it isn't another non-volatile memory command which could interfere with the storage of the macro.

## 8.81. `MACROK` -- Delete One or All Macros

This command kills (deletes) the specified macro. If the wildcard character (*) is used the specify the macro number, then all macros are deleted. Deleting all macros via `MACROK*` is more efficient than deleting them all individually since it requires less writes to non-volatile memory.

This command is saved to global non-volatile memory and is not part of a preset. Its value is saved each time it is changed. It will retain its value after power-down. Since this command writes to non-volatile memory, there will be a delay before an acknowledgment is returned.

| Example | Description | Status Message |
|---|---|---|
| F01MACROK25 | Delete all the commands associated with macro number 25. | F01MACROK25 |
| F01MACROK* | Delete all the commands associated with all 255 macros. | F01MACROK* |

## 8.82. `MACROL` -- List All Commmands in a Macro

This command lists all the commands in a given macro. For example, assume that a macro has been defined via the following command sequence.

```
F01MACROS112
F01MACROA112,MUTEI*0
F01MACROA112,GAINI*0
F01MACROA112,MUTEO*0
F01MACROW112
```

Once this macro has been defined, issuing the command:

```
F01MACROL112?
```

Will result in the following status messages:

```
F01MACROL112,MUTEI*0
F01MACROL112,GAINI*0
F01MACROL112,MUTEO*0
F01MACROL112
```

## 8.83. `MACROQ` -- Execute Macro Quietly

This command executes the macro corresponding to the specified number. The macro must have been previously defined and stored in non-volatile memory via the MACROS, MACROA, and MACROW commands. If the specified macro is empty, then an error condition of `ERROR#070` will be generated.

When the `MACROQ` command executes, status messages for all of the commands in the macro will not be generated. The `MACROQ` is provided so that a control system can execute a macro and not generate heavy data traffic from the resulting status messages. Of course, this assumes that the control system is not interested in any of the acknowledgements. The MACROX command can be used to execute the macro without suppressing status messages.

| Example | Description | Status Message |
|---|---|---|
| F01MACROQ125 | Execute macro number 125 without generating any status messages. | F01MACROQ125 |

## 8.84. `MACROS` -- Start a New Macro

This command is used to start writing a new macro with a specified macro number. This command is used in conjunction with the MACROA and MACROW commands to create a new macro. If this command is sent while another macro is in the process of being written (before the MACROW command is sent), then the macro in progress will be deleted and the new one will be started.

See the description of the MACROA command for detailed information on creating macros.

## 8.85. `MACROW` -- Write Macro to Non-Volatile Memory

This command writes the macro that is currently being defined to non-volatile memory. This command is used in conjunction with MACROA and MACROS to create a new macro. If this command is sent when no macro is in progress (i.e., if a MACROS command has not been sent) or if the macro number sent with this command does not match the number of the macro in progress, then an error will be returned.

See the description of the MACROA command for detailed information on creating macros.

This command is saved to global non-volatile memory and is not part of a preset. Its value is saved each time it is changed. It will retain its value after power-down. Since this command writes to non-volatile memory, there will be a delay before an acknowledgment is returned.

## 8.86. `MACROX` -- Execute Macro

This command executes the macro corresponding to the specified number. The macro must have been previously defined and stored in non-volatile memory via the MACROS, MACROA, and MACROW commands. If the specified macro is empty, then an error condition of `ERROR#070` will be generated.

When the `MACROX` executes, status messages for all of the commands in the macro will be generated (unless [acknowledgement mode](#) is turned off). The status messages can be suppressed by using the [MACROQ](#) command, which is exactly the same as `MACROX` except that status messages are suppressed.

| Example | Description | Status Message |
|---------|-------------|----------------|
| `F01MACROX125` | Execute macro number 125. | Status messages for all the commands in the macro will be generated first, followed by the message: `F01MACROX125` |

## 8.87. `METER` -- Select which Signal is Displayed on the Front Panel LED Meter

This command selects which signal is displayed on the front panel LED meter. The options correspond to any of the 12 inputs, 12 outputs, or two AEC references. The inputs are specified by the labels `I1` - `I8` and `IA` - `ID`. The outputs are specified by the labels `O1` - `O8` and `OA` - `OD`. The AEC references are specified by the labels `R1` and `R2`.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| `F01METERI1` | Set front panel LED meter to display the signal on input 1 . | `F01METERI1` |
| `F01METER?` | Query which signal the front panel LED meter is currently set to monitor. | `F01METERx` , where `x` is the label corresponding to the signal currently being monitored. If this command was sent after the first example given above, the response would be `F01METERI1` . |

## 8.88. `MGAIN` -- Set Crosspoint Gains in Main Matrix or Submatrix

This command sets or queries one or more crosspoint gains in either the main matrix or one of the EF Bus submatrices.

This command is a matrix integer command. See [Section 6.4](#) and [Section 6.2](#) for more information on this type of command. The minimum and maximum values for this command are -100 and 20, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| `F01MGAINB,R1,-3` | Set gain from input B to AEC reference 1 to -3 dB. | `F01MGAINB,R1,-3` |
| `F01MGAIN1,B,>6` | Increase gain from input 1 to output B by 6 dB. | `F01MGAIN1,B,x` where `x` is the new value of the crosspoint gain. If the crosspoint gain was set to -3 dB before this command, then the status message would be `F01MGAIN1,B,3` . |

| | | |
|---|---|---|
| `F01MGAINA,W,?` | Query gain from input A to W Bus output. | `F01MGAINA,W,x` , where `x` is the current value of the crosspoint gain. If the crosspoint gain was set to -12 dB before this command, then the status message would be `F01MGAINA,W,-12` . |
| `F01MGAINWB0,WM0,-3` | Set crosspoint gain in EF Bus submatrix on crosspoint that routes the W bus signal from the device at ID0 (WB0) to W submatrix output 0 (WM0). | `F01MGAINWB0,WM0,-3` |
| `F01MGAIN1,*, ääääääää{{{{ääçççç` | Set all crosspoint gains for input channel 1. Set the gain to outputs 1-8 to 0 dB (ä), the gain to outputs A-D to -9 dB ({), the gain to AEC reference 1 and 2 to 0 dB (ä), and the gain to bus outputs W, X, Y, and Z to -3 dB (ç). | `F01MGAIN1,*, ääääääää{{{{ääçççç` |
| `F01MGAINB,*,-3` | Set all crosspoint gains for input channel B to -3 dB. | `F01MGAINB,*,çççççççççççççççççç` |
| `F01MGAINB,*,>12` | Increase the gain for all crosspoints of input channel B by 12 dB. | `F01MGAINB,*,abcdefghijklmnopqr` , where the value of each of letters (a, b, etc.) depends on the current state of each of the crosspoint gains. If this command was issued after the example above, then the status message would be `F01MGAINB,*,èèèèèèèèèèèèèèèèèè` |
| `F01MGAINB,*,?` | Query the gains for all crosspoints of input channel B . | `F01MGAINB,*,abcdefghijklmnopqr` , where the value of each of letters (a, b, etc.) depends on the current state of each of the crosspoint gains. If this command was issued after the example above, then the status message would be `F01MGAINB,*,èèèèèèèèèèèèèèèèèè` |

## 8.89. `MGATE` -- Select Gated or Ungated Microphone Signal in Matrix

For each crosspoint of the microphone input channels of the main matrix, the microphone signals can be taken

before the automixer (ungated) or after the automixer (gated). This command selects the gated or ungated microphone signal. If MGATE is set to 1 for a given crosspoint, then the signal will be gated (taken after the automixer). If MGATE is set to 0 for a given crosspoint, then the signal will be ungated (taken before the automixer).

This command is a matrix boolean command. See [Section 6.4](#) and [Section 6.1](#) for more information on this type of command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| F01MGATE3,A,1 | Select the gated version of microphone input 3 to be sent to output A. | F01MGATE3,A,1 |
| F01MGATE3,A,0 | Select the ungated version of microphone input 3 to be sent to output A. | F01MGATE3,A,0 |
| F01MGATE3,A,2 | Toggle the gated status of the crosspoint that routes microphone input 3 to output A. | F01MGATE3,A,x , where x is 0 or 1 depending on the current gated status of the crosspoint. |
| F01MGATE3,A,? | Query the gated status of the crosspoint that routes microphone input 3 to output A. | F01MGATE3,A,x , where x is 0 or 1 depending on the current gated status of the crosspoint. |
| F01MGATE*,A,1 | Select the gated version of all microphone inputs to be sent to output A. | F01MGATE*,A,11111111 -- note that there are only eight values returned, since only the eight microphone inputs can be gated. The other inputs do not pass through the automixer. |
| F01MGATE*,A,0 | Select the ungated version of all microphone inputs to be sent to output A. | F01MGATE*,A,00000000 -- note that there are only eight values returned, since only the eight microphone inputs can be gated. The other inputs do not pass through the automixer. |
| F01MGATE*,A,11110000 | Select the gated version of microphone inputs 1-4 to be sent to output A and the ungated version of microphone inputs 5-8 to be sent to output A. | F01MGATE*,A,11110000 |
| F01MGATE*,A,? | Query the gated status of all microphone inputs routed to output A. | F01MGATE*,A,abcdefgh, where a-h are each 0 or 1 depending on the current gated status of each crosspoint. |

## 8.90. MIC -- Enable Microphone Gain Stage on Inputs 1-8

This command sets or queries the enabled state of the microphone gain stages on each of the microphone inputs . Enabling the gain stage adds an extra 33 dB of gain in the input signal path. See the description of the [GAINI](#) command for information on how the MIC and [GAINI](#) commands relate to nominal input levels.

This command is a channel boolean command. See [Section 6.3](#) and [Section 6.1](#) for more information on this type of command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01MIC21 | Enable microphone gain stage for input channel 2 . | F01MIC21 |
| F01MIC20 | Disable microphone gain stage for input channel 2 . | F01MIC20 |
| F01MIC22 | Toggle microphone gain stage for input channel 2 . | F01MIC2x , where x is 0 or 1 depending on the current enabled state of the microphone gain stage on input channel 2 . |
| F01MIC2? | Query enabled status of microphone gain stage for input channel 2 . | F01MIC2x , where x is 0 or 1 depending on the current enabled state of the microphone gain stage on input channel 2 . |
| F01MIC*0 | Disable microphone gain stage for input channels 1-8. | F01MIC*00000000 |
| F01MIC*1 | Enable microphone gain stage for input channels 1-8. | F01MIC*11111111 |
| F01MIC*2 | Toggle enabled status of microphone gain stage for input channels 1-8. | F01MIC*abcdefgh where a-h are each 0 or 1 depending on the current enabled state of the microphone gain stage for the corresponding input channel. |
| F01MIC*? | Query enabled status of microphone gain stage for input channels 1-8. | F01MIC*abcdefgh where a-h are each 0 or 1 depending on the current enabled state of the microphone gain stage for the corresponding input channel. |
| F01MIC*11110000 | Enable microphone gain stages for inputs 1-4 and disable microphone gain stages for inputs 5-8. | F01MIC*11110000 |

## 8.91. `MINI` -- Enable Modem Initialization String

This command controls whether or not the modem initialization string is sent at power-up. If `MINI` is `0`, then the modem initialization string is not sent. If `MINI` is `1`, then the modem initialization string is sent. The text of the modem initialization string is set via the [MINISTR](#) command.

This command is a boolean command. See [Section 6.1](#) for more information on this type of command.

This command is saved to global non-volatile memory and is not part of a preset. Its value is saved each time it is changed. It will retain its value after power-down. Since this command writes to non-volatile memory, there will be a delay before an acknowledgment is returned.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01MINI1 | Enable modem initialization string. | F01MINI1 |
| F01MINI0 | Disable modem initialization string. | F01MINI0 |

| | | |
|---|---|---|
| `F01MINI2` | Toggle enabled state of modem initialization string. | `F01MINIx`, where `x` is `0` or `1` depending on the current enabled state of the modem initialization string. |
| `F01MINI?` | Query enabled state of modem initialization string. | `F01MINIx`, where `x` is `0` or `1` depending on the current enabled state of the modem initialization string. |

## 8.92. `MINISTR` -- Set Modem Initialization String

This command is used to set or query the modem initialization string. The string can be a maximum of 32 characters long. The recommended modem initialization string for the Vortex is:

`ATF1E0&B1S0=2`

This can be set via the following command.

`F01MINISTRATF1E0&B1S0=2`

Whether or not the modem initialization string is sent at power-up is controlled via the <u>`MINI`</u> command.

This command is saved to global non-volatile memory and is not part of a preset. Its value is saved each time it is changed. It will retain its value after power-down. Since this command writes to non-volatile memory, there will be a delay before an acknowledgment is returned.

| Example | Description | Status Message |
|---|---|---|
| `F01MINISTRATF1E0&B1S0=2` | Set modem initialization string to `ATF1E0&B1S0=2`. | `F01MINISTRATF1E0&B1S0=2` |
| `F01MINISTR?` | Query current modem initialization string. | `F01MINISTR<string>`, where `<string>` is the current modem initialization string. If this command was sent after the above example, then the status message would be `F01MINISTRATF1E0&B1S0=2`. |

## 8.93. `MMUTE` -- Mute Crosspoint in Main Matrix or Submatrix

This command sets or queries the mute status of one or more crosspoints in either the main matrix or one of the EF Bus submatrices.

This command is a matrix boolean command. See [Section 6.4](#) and [Section 6.1](#) for more information on this type of command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| | | |

| | | |
|---|---|---|
| `F01MMUTEB,R1,1` | Mute signal path (in main matrix) from input B to AEC reference 1 . | `F01MMUTEB,R1,1` |
| `F01MMUTEWB0,WM0,0` | Mute signal path (in EF Bus W submatrix) from W bus signal at device ID0 (WB0) to W submatrix output 0 (WM0). | `F01MMUTEWB0,WM0,0` |
| `F01MMUTE1,B,2` | Toggle mute status of signal path (in main matrix) from input 1 to output B . | `F01MMUTE1,B,x` , where `x` is either `0` or `1` depending on the current mute status of the crosspoint. |
| `F01MMUTEA,W,?` | Query mute status of signal path (in main matrix) from input A to W-Bus output . | `F01MMUTEA,W,x` , where `x` is either `0` or `1` depending on the current mute status of the crosspoint. |
| `F01MMUTE1,*,111111110000000000` | Set all crosspoint mutes for input channel 1. Mute the paths from input 1 to outputs 1-8 and unmute the paths from input 1 to outputs A-D, R1, R2, W, X, Y, and Z. | `F01MMUTE1,*,111111110000000000` |
| `F01MMUTE2,*,0` | Unmute all crosspoints for input channel 2 . | `F01MMUTE2,*,000000000000000000` |
| `F01MMUTE2,*,2` | Toggle all crosspoint mutes for input channel 2 . | `F01MMUTE2,*,abcdefghijklmnopqr` , where the value of each of letters (a, b, etc.) is either `0` or `1` depending on the current state of each of the crosspoint mutes. If this command was issued after the example above, then the status message would be `F01MMUTE2,*,000000000000000000` |
| `F01MMUTE2,*,?` | Query all crosspoint mutes for input channel 2 . | `F01MMUTE2,*,abcdefghijklmnopqr` , where the value of each of letters (a, b, etc.) is either `0` or `1` depending on the current state of each of the crosspoint mutes. If this command was issued after the example above, then the status message would be `F01MMUTE2,*,000000000000000000` |

## 8.94. MUTEGIL -- Set Mute Status of Line Inputs as a Group

This command sets the mute status of all the line inputs simultaneously. This is different from using a wildcard, *
with the MUTEI command because that command sets all the mic and line inputs together, while the MUTEGIL
sets only the line inputs.

This command will generate acknowledgements from each of the line inputs separately.

This command is a boolean command. See Section 6.1 for more information on this type of command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be
restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| F01MUTEGIL1 | Mute all line inputs. | F01MUTEIA1<br>F01MUTEIB1<br>F01MUTEIC1<br>F01MUTEID1 |
| F01MUTEGIL2 | Toggle mute status of all line inputs. | F01MUTEIAa<br>F01MUTEIBb<br>F01MUTEICc<br>F01MUTEIDd ,<br>where a, b, c, and d reflect the mute status (0 or 1) of each of the line inputs. If this command was issued after the example above, then the status messages would be:<br>F01MUTEIA0<br>F01MUTEIB0<br>F01MUTEIC0<br>F01MUTEID0 . |
| F01MUTEGIL? | Query the mute status of all the line inputs. | F01MUTEIAa<br>F01MUTEIBb<br>F01MUTEICc<br>F01MUTEIDd ,<br>where a, b, c, and d reflect the mute status (0 or 1) of each of the line inputs. If this command was issued after the example above, then the status messages would be:<br>F01MUTEIA0<br>F01MUTEIB0<br>F01MUTEIC0<br>F01MUTEID0 . |

## 8.95. MUTEGIM -- Set Mute Status of Microphone Inputs as a Group

This command sets the mute status of all the microphone inputs simultaneously. This is different from using a
wildcard, * with the MUTEI command because that command sets all the mic and line inputs together, while the
MUTEGIM sets only the mic inputs.

This command will generate acknowledgements from each of the mic inputs separately.

This command is a boolean command. See Section 6.1 for more information on this type of command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| F01MUTEGIM1 | Mute all mic inputs. | F01MUTEI11<br>F01MUTEI21<br>F01MUTEI31<br>F01MUTEI41<br>F01MUTEI51<br>F01MUTEI61<br>F01MUTEI71<br>F01MUTEI81 |
| F01MUTEGIM2 | Toggle mute status of all mic inputs. | F01MUTEI1a<br>F01MUTEI2b<br>F01MUTEI3c<br>F01MUTEI4d<br>F01MUTEI5e<br>F01MUTEI6f<br>F01MUTEI7g<br>F01MUTEI8h ,<br>where a, b, c, d, e, f, g, and h reflect the mute status (0 or 1) of each of the mic inputs. If this command was issued after the example above, then the status messages would be:<br>F01MUTEI10<br>F01MUTEI20<br>F01MUTEI30<br>F01MUTEI40<br>F01MUTEI50<br>F01MUTEI60<br>F01MUTEI70<br>F01MUTEI80 . |
| F01MUTEGIM? | Query mute status of all mic inputs. | F01MUTEI1a<br>F01MUTEI2b<br>F01MUTEI3c<br>F01MUTEI4d<br>F01MUTEI5e<br>F01MUTEI6f<br>F01MUTEI7g<br>F01MUTEI8h ,<br>where a, b, c, d, e, f, g, and h reflect the mute status (0 or 1) of each of the mic inputs. If this command was issued after the example above, then the status messages would be:<br>F01MUTEI10<br>F01MUTEI20<br>F01MUTEI30<br>F01MUTEI40<br>F01MUTEI50<br>F01MUTEI60 |

| | | F01MUTEI70 |
| | | F01MUTEI80 . |

## 8.96. `MUTEI` -- Mute One or More Inputs

This command sets or queries the mute status of the analog inputs (1-8, A-D) .

This command is a channel boolean command. See [Section 6.3](#) and [Section 6.1](#) for more information on this type of command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01MUTEI41 | Mute input 4 . | F01MUTEI41 |
| F01MUTEIA0 | Unmute input A . | F01MUTEIA0 |
| F01MUTEI72 | Toggle mute status of input 7 . | F01MUTEI7x , where x is either 0 or 1 depending on the current mute status of the input. |
| F01MUTEIC? | Query mute status of input C . | F01MUTEICx , where x is either 0 or 1 depending on the current mute status of the input. |
| F01MUTEI*1 | Mute all inputs (1-8, A-D) . | F01MUTEI*111111111111 |
| F01MUTEI*0 | Unmute all inputs (1-8, A-D) . | F01MUTEI*000000000000 |
| F01MUTEI*111111110000 | Mute inputs 1-8 and unmute inputs A-D. | F01MUTEI*111111110000 |
| F01MUTEI*2 | Toggle mute status of all inputs. | F01MUTEI*abcdefghijkl , where each of the letters (a, b, etc.) is either 0 or 1 depending of the current status of the mute on the corresponding input. If this command was sent after the example above, then the status message would be F01MUTEI*111111110000 . |
| F01MUTEI*? | Query mute status of all inputs. | F01MUTEI*abcdefghijkl where each of the letters (a, b, etc.) is either 0 or 1 depending of the current status of the mute on the corresponding input. If this command was sent after the example above, then the status message would be F01MUTEI*111111110000 . |

## 8.97. `MUTEO` -- Mute One or More Outputs

This command sets or queries the mute status of the analog outputs (1-8, A-D) .

This command is a channel boolean command. See [Section 6.3](#) and [Section 6.1](#) for more information on this type of command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01MUTEO41 | Mute output 4 . | F01MUTEO41 |
| F01MUTEOA0 | Unmute output A . | F01MUTEOA0 |
| F01MUTEO72 | Toggle mute status of output 7 . | F01MUTEO7x , where x is either 0 or 1 depending on the current mute status of the output. |
| F01MUTEOC? | Query mute status of output C . | F01MUTEOCx , where x is either 0 or 1 depending on the current mute status of the output. |
| F01MUTEO*1 | Mute all outputs (1-8, A-D) . | F01MUTEO*111111111111 |
| F01MUTEO*0 | Unmute all outputs (1-8, A-D) . | F01MUTEO*000000000000 |
| F01MUTEO*111111110000 | Mute outputs 1-8 and unmute outputs A-D. | F01MUTEO*111111110000 |
| F01MUTEO*2 | Toggle mute status of all outputs. | F01MUTEO*abcdefghijkl where each of the letters (a, b, etc.) is either 0 or 1 depending of the current status of the mute on the corresponding output. If this command was sent after the example above, then the status message would be F01MUTEO*000000001111 . |
| F01MUTEO*? | Query mute status of all outputs. | F01MUTEO*abcdefghijkl , where each of the letters (a, b, etc.) is either 0 or 1 depending of the current status of the mute on the corresponding output. If this command was sent after the example above, then the status message would be F01MUTEO*000000001111 |

## 8.98. NC -- Enable Noise Cancellation

This command sets or queries the status of the Noise Cancellation (NC) algorithm on input channels 1-8 .

This command is a channel boolean command. See [Section 6.3](#) and [Section 6.1](#) for more information on this type of command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01NC31 | Enable NC on input channel 3 . | F01NC31 |
| F01NC10 | Disable NC on input channel 1 . | F01NC10 |
| F01NC12 | Toggle NC state on input channel 1 . | F01NC1x , where x is 0 or 1 depending on the current state of NC on input channel 1 . |
| F01NC2? | Query NC state on input channel 2 . | F01NC2x , where x is 0 or 1 depending on the current state of NC on the input channel. |
| F01NC*1 | Enable NC on input channels 1-8 . | F01NC*11111111 |
| F01NC*0 | Disable NC on all microphone input channels (1-8) . | F01NC*00000000 |

| | | |
|---|---|---|
| `F01NC*2` | Toggle NC state on all microphone input channels (1-8) . | `F01NC*abcdefgh` , where the letters (`a`, `b`, etc.) are each `0` or `1` depending on the current state of NC for each of the corresponding input. |
| `F01NC*?` | Query NC state on all microphone input channels (1-8) . | `F01NC*abcdefgh` , where the letters (`a`, `b`, etc.) are each `0` or `1` depending on the current state of NC for each of the corresponding input. |

## 8.99. `NCL` -- Set Noise Cancellation Attenuation

This command sets or queries the attenuation level of the Noise Cancellation algorithm on input channels 1-8 . For example, if `NCL` is set to `10`, then the Noise Cancellation for that channel will cancel 10 dB of noise. Higher numbers mean more cancellation will be applied, but may result in slight artifacts depending on the characteristics of the noise. Typical settings are 10 dB for normal cancellation and 6 dB for light cancellation.

This command is a channel integer command. See [Section 6.3](#) and [Section 6.2](#) for more information on this type of command. The minimum and maximum values for this command are 0 and 20, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| `F01NCL13` | Set NC attenuation level on input channel 1 to 3 dB. | `F01NCL13` |
| `F01NCL1?` | Query the NC attenuation level on input channel 1 . | `F01NCL1x` , where `x` is a number between 0 and 20, depending on the current setting of the NC attenuation level on input channel 1 . |
| `F01NCL*6` | Set NC attenuation level on all microphone input channels (1-8) to 6 dB. | `F01NCL*èèèèèèèè` |
| `F01NCL*ääääìììì` | Set NC attenuation level on input channels 1-4 to 0 dB and NC attenuation level on input channels 5-8 to 9 dB. | `F01NCL*ääääìììì` |
| `F01NCL*?` | Query NC attenuation level on all microphone input channels (1-8) . | `F01NCL*abcdefgh`, where `a-h` are each between ä and ÿ, depending on the current setting of the NC attenuation level for each of the eight input channels. |

## 8.100. `NVINIT` -- Reinitialize Non-Volatile Memory

This command reinitializes (or formats) all non-volatile memory settings. This will effectively erase settings for global settings, user presets, macros, labels, and logic I/O configurations. This will basically reset the unit to a "fresh-out-of-the-box" state.

When this command is issued, it writes to non-volatile memory in order to format the contents back to the factory default state. The writing process results in a short delay. After the non-volatile memory has been reformatted, the Vortex will perform a software reset, which is equivalent to a power cycle.

| Example | Description | Status Message |
|---|---|---|
| | | |

| | | |
|---|---|---|
| `F01NVINIT` | Reinitialize non-volatile memory and perform a software reset. | No status message will be generated. There will be a short delay and then the Vortex will reset. |

# 8.101. `NVLOCK` -- Lock/Unlock Non-Volatile Memory

This command controls the non-volatile memory lock feature. When `NVLOCK` is enabled, the user may not save any system settings to non-volatile memory. This includes global parameters, presets, macros, labels, and logic assignments. The lock applies whether the user tries to make the changes via RS-232, front panel, logic inputs, or any other method. The user will still be able to query all the features of the device, but will get an error message if an attempt is made to change them. When `NVLOCK` is disabled, the user has full access to the system settings.

Usage of this command is similar to other boolean commands (see [Section 6.1](#)) except that when disabling this feature, the password must be supplied for the command to work. If an incorrect password is supplied, `'ERROR#005'` will be generated. The examples below illustrate the correct usage. The examples assume that the password has been set to `'aspi'` (the default). The `NVLOCK` password can be changed via the [NVPSWD](#) command.

This command is saved to global non-volatile memory and is not part of a preset. Its value is saved each time it is changed. It will retain its value after power-down. Since this command writes to non-volatile memory, there will be a delay before an acknowledgment is returned.

| Example | Description | Status Message |
|---|---|---|
| `F01NVLOCK1` | Lock the non-volatile memory to prevent users from changing the settings. | `F01NVLOCK1` |
| `F01NVLOCK0,yumyum` | Attempt to unlock the non-volatile memory by using an invalid password. | `F01ERROR#005` -- since an invalid password was given, an error message was generated. If error messages have been disabled (via the [ERROR](#) command), then no status message will be generated. |
| `F01NVLOCK0,aspi` | Unlock the non-volatile memory by using the correct password. | `F01NVLOCK0` |
| `F01NVLOCK?` | Query the locked status of the non-volatile memory. | `F01NVLOCKx` , where `x` is `0` or `1` depending on whether the non-volatile memory is unlocked or locked, respectively. |

# 8.102. `NVPSWD` -- Change Non-Volatile Memory Password

This command sets or queries the non-volatile memory password. This password is used in conjunction with the [NVLOCK](#) command. The non-volatile memory must be unlocked (`'NVLOCK0'`) in order to use this command to set or query the non-volatile memory password. If the non-volatile memory is locked, then this command will result in `'ERROR#004'`. The examples below assume that the non-volatile memory is unlocked.

| Example | Description | Status Message |
|---|---|---|
| `F01NVPSWDlemur` | Set non-volatile memory password to `'lemur'`. | `F01NVPSWDlemur` |
| `F01NVPSWD?` | Query the current non-volatile memory password. | `F01NVPSWDlemur` |

# 8.103. `PEQIA` -- Set All Parameters for Specified Parametric EQ Input Stage

This command sets or queries all of the parameters for the parametric equalizer (EQ) filters on input channels 1-8, and A-D . Each channel has five bands of parametric EQ that can be independently controlled.

The input parametric EQ filter parameters can be set individually via the `PEQIT`, `PEQIF`, `PEQIB`, `PEQIG`, `PEQIS`, and `PEQIE` commands. The output parametric EQ filter parameters can be set simultaneously via the `PEQOA` command or individually with the `PEQOT`, `PEQOF`, `PEQOB`, `PEQOG`, `PEQOS`, and `PEQOE` commands.

When using this command, you must specify all of the following parameters.

| Description | Value Range | Units |
|---|---|---|
| Channel | 1-8, A-D | N/A |
| Band | 1-5 | N/A |
| Filter Type | 1-5 | 1 = parametric filter<br>2 = low shelf<br>3 = high shelf<br>4 = lowpass<br>5 = highpass<br>6 = Linkwitz-Riley lowpass<br>7 = Linkwitz-Riley highpass |
| Frequency | 20 - 20000 | Hz |
| Bandwidth | 5 - 200 | 1/100th octave |
| Gain | -20 - 20 | dB |
| Slope | 1 - 1.2 * Gain for shelving filters, 12 or 24 for Linkwitz-Riley filters | dB per octave |
| Enable | 0 - 1 | 0 = filter disabled<br>1 = filter enabled |

The paramters are specified in the order shown and are separated by commas. The wildcard character, *, may not be used for any of the parameters listed above. If an attempt is made to set one of the parameters outside the valid range, the command will fail and return an error message of `ERROR#002`.

Not all of the parameters are valid for each of the filter types. The following table shows which parameters are used in each of the filter types.

| Filter Type | Bandwidth | Frequency | Gain | Slope |
|---|---|---|---|---|
| Parametric (1) | Yes | Yes | Yes | No |
| Low Shelf (2) | No | Yes | Yes | Yes |
| High Shelf (3) | No | Yes | Yes | Yes |
| Lowpass (4) | No | Yes | No | No |
| Highpass (5) | No | Yes | No | No |
| Linkwitz-Riley Lowpass (6) | No | Yes | No | Yes |
| Linkwitz-Riley Highpass (7) | No | Yes | No | Yes |

When a parameter is invalid for a given filter type, the parametric EQ commands will still set that parameter,

however its value will not be used for filter computations. This valid/invalid parameter information above is provided mainly as guidelines for programmers writing control software for the Vortex. If a parameter is not valid for a specific filter type, then the control for editing that parameter should be disabled or removed when that filter type is selected.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| `F01PEQIA3,2,1,1250,40,-20,1,1` | Set the parametric EQ on input 3 , band 2 to the following parameters: type = parametric filter, frequency = 1250 Hz, bandwidth = 40 1/100th octaves, gain = -20 dB, enabled = yes, slope = 1 dB per octave. The value of the slope parameter is irrelevant for this type of filter, we could have set it to anything. | `F01PEQIA3,2,1,1250,40,-20,1,1` |
| `F01PEQIAB,1,?` | Query the current parameter settings for the parametric eq filter on input B , band 1. | `F01PEQIAB,1,t,f,b,g,s,e` , where the parameters `t`, `f`, `b`, `g`, `s`, and `e` correspond to the current settings of the type, frequency, bandwidth, gain, slope, and enabled parameters, respectively. If this filter was set to the same parameters as given in the above example, the status message would be `F01PEQIAB,1,1,1250,40,-20,1,1` . |

# 8.104. `PEQIB` -- Set Bandwidth Parameter for Specified Parametric EQ Input Stage

This command sets or queries the bandwidth parameter (in 1/100th octaves) for the parametric equalizer (EQ) filters on input channels 1-8 and A-D . Each channel has five bands of parametric EQ that can be independently controlled.

The minimum and maximum values for the bandwidth parameter are 5 and 200, respectively.

This command has the same format and restrictions as the `PEQIA` command except that only one parameter, the bandwidth, is specified instead of all the parameters. See the `PEQIA` command for more information. To set the bandwidth parameter for an output parametric EQ filter, use the `PEQOB` command. To set all the parameters for an output parametric EQ filter, use the `PEQOA` command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01PEQIB3,2,40 | Set bandwidth of parametric EQ on input 3 , band 2 to 40 1/100th octaves. | F01PEQIB3,2,40 |
| F01PEQIBB,1,? | Query current bandwidth setting of parametric EQ on input B , band 1. | F01PEQIBB,1,b , where b is the current setting of the bandwidth parameter for the parametric EQ on input B , band 1. If the bandwidth parameter of this filter is set to the same value given in the example above, then the status message will be F01PEQIBB,1,40 . |

## 8.105. PEQIE -- Set Enabled Parameter for Specified Parametric EQ Input Stage

This command sets or queries the enabled parameter (0 = filter is disabled, 1 = filter is enabled) for the parametric equalizer (EQ) filters on input channels 1-8 and A-D . Each channel has five bands of parametric EQ that can be independently controlled.

This command has the same format and restrictions as the PEQIA command except that only one parameter, the enabled status, is specified instead of all the parameters. See the PEQIA command for more information. To set the enabled parameter for an output parametric EQ filter, use the PEQOE command. To set all the parameters for an output parametric EQ filter, use the PEQOA command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01PEQIE3,2,1 | Enable parametric EQ on input 3 , band 2. | F01PEQIE3,2,1 |
| F01PEQIEB,1,? | Query enabled status of parametric EQ on input B , band 1. | F01PEQIEB,1,e , where e is the current setting of the enabled parameter for the parametric EQ on input B , band 1. If the enabled parameter of this filter is set to the same value given in the example above, then the status message will be F01PEQIEB,1,1 . |

## 8.106. PEQIF -- Set Frequency Parameter for Specified Parametric EQ Input Stage

This command sets or queries the frequency parameter (in Hz) for the parametric equalizer (EQ) filters on input channels 1-8 and A-D . Each channel has five bands of parametric EQ that can be independently controlled.

The minimum and maximum values for the frequency parameter are 20 and 20000, respectively.

For filter types that have a center frequency (e.g., parametric), this parameter specifies the center frequency. For filter types that have a cutoff frequency (e.g, highpass, lowpass, etc.), this parameter specifies the cutoff frequency.

This command has the same format and restrictions as the PEQIA command except that only one parameter, the frequency, is specified instead of all the parameters. See the PEQIA command for more information. To set the frequency parameter for an output parametric EQ filter, use the PEQOF command. To set all the parameters for an output parametric EQ filter, use the PEQOA command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| `F01PEQIF3,2,1250` | Set frequency of parametric EQ on input 3 , band 2 to 1250 Hz. | `F01PEQIF3,2,1250` |
| `F01PEQIFB,1,?` | Query current frequency setting of parametric EQ on input B , band 1. | `F01PEQIFB,1,f` , where `f` is the current setting of the frequency parameter for the parametric EQ on input B , band 1. If the frequency parameter of this filter is set to the same value given in the example above, then the status message will be `F01PEQIFB,1,1250` . |

## 8.107. `PEQIG` -- Set Gain Parameter for Specified Parametric EQ Input Stage

This command sets or queries the gain parameter (in dB) for the parametric equalizer (EQ) filters on input channels 1-8 and A-D . Each channel has five bands of parametric EQ that can be independently controlled.

The minimum and maximum values for the gain parameter are -20 and 20, respectively.

This command has the same format and restrictions as the PEQIA command except that only one parameter, the gain, is specified instead of all the parameters. See the PEQIA command for more information. To set the gain parameter for an output parametric EQ filter, use the PEQOG command. To set all the parameters for an output parametric EQ filter, use the PEQOA command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| `F01PEQIG3,2,-20` | Set the gain of the parametric EQ on input 3 , band 2 to -20 dB. | `F01PEQIG3,2,-20` |
| `F01PEQIGB,1,?` | Query current gain setting of parametric EQ on input B , band 1. | `F01PEQIGB,1,g` , where `g` is the current setting of the gain parameter for the parametric EQ on input B , band 1. If the gain parameter of this filter is set to the same value given in the example above, then the status message will be `F01PEQIGB,1,-20` . |

## 8.108. `PEQIS` -- Set Slope Parameter for Specified Parametric EQ Input Stage

This command sets or queries the slope parameter (in dB per octave) for the parametric equalizer (EQ) filters on

input channels 1-8 and A-D . Each channel has five bands of parametric EQ that can be independently controlled.

For the Linkwitz-Riley filters, the slope can either be 12 or 24 (dB/Octave).For the low shelf and high shelf filters, the minimum value for the slope parameter is 1, and the maximum value is 1.2 times the current value of the gain parameter. See the PEQIG command for information on the gain parameter.

This command has the same format and restrictions as the PEQIA command except that only one parameter, the slope, is specified instead of all the parameters. See the PEQIA command for more information. To set the slope parameter for an output parametric EQ filter, use the PEQOS command. To set all the parameters for an output parametric EQ filter, use the PEQOA command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01PEQIS3,2,1 | Set the slope of the parametric EQ on input 3 , band 2 to 1 dB per octave. | F01PEQIS3,2,1 |
| F01PEQISB,1,? | Query current slope setting of parametric EQ on input B , band 1. | F01PEQISB,1,s , where s is the current setting of the slope parameter for the parametric EQ on input B , band 1. If the slope parameter of this filter is set to the same value given in the example above, then the status message will be F01PEQISB,1,1 . |

# 8.109. `PEQIT` -- Set Type Parameter for Specified Parametric EQ Input Stage

This command sets or queries the type parameter for the parametric equalizer (EQ) filters on input channels 1-8 and A-D . Each channel has five bands of parametric EQ that can be independently controlled.

The type parameter specifies what type of filter is executed for the specified band of the specified input channel. The different filter types and their corresponding values are shown in the table below.

| Filter Type | Value of Type Parameter |
|-------------|-------------------------|
| Parametric | 1 |
| Low Shelf | 2 |
| High Shelf | 3 |
| Lowpass | 4 |
| Highpass | 5 |
| Linkwitz-Riley Lowpass | 6 |
| Linkwitz-Riley Highpass | 7 |

This command has the same format and restrictions as the PEQIA command except that only one parameter, the filter type, is specified instead of all the parameters. See the PEQIA command for more information. To set the slope parameter for an output parametric EQ filter, use the PEQOT command. To set all the parameters for an output parametric EQ filter, use the PEQOA command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be

restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| `F01PEQIT3,2,1` | Set the type of the parametric EQ on input 3 , band 2 to parametric. | `F01PEQIT3,2,1` |
| `F01PEQITB,1,?` | Query current type setting of parametric EQ on input B , band 1. | `F01PEQITB,1,t` , where `t` is the current setting of the type parameter for the parametric EQ on input B , band 1. If the type parameter of this filter is set to the same value given in the example above, then the status message will be `F01PEQITB,1,1` . |

# 8.110. `PEQOA` -- Set All Parameters for Specified Parametric EQ Output Stage

This command sets or queries all of the parameters for the parametric equalizer (EQ) filters on output channels 1-8, and A-D . Each channel has five bands of parametric EQ that can be independently controlled.

The output parametric EQ filter parameters can be set individually via the [PEQOT](), [PEQOF](), [PEQOB](), [PEQOG](), [PEQOS](), and [PEQOE]() commands. The input parametric EQ filter parameters can be set simultaneously via the [PEQIA]() command or individually with the [PEQIT](), [PEQIF](), [PEQIB](), [PEQIG](), [PEQIS](), and [PEQIE]() commands.

When using this command, you must specify all of the following parameters.

| Description | Value Range | Units |
|-------------|-------------|-------|
| Channel | 1-8, A-D | N/A |
| Band | 1-5 | N/A |
| Filter Type | 1-5 | 1 = parametric filter<br>2 = low shelf<br>3 = high shelf<br>4 = lowpass<br>5 = highpass<br>6 = Linkwitz-Riley lowpass<br>7 = Linkwitz-Riley highpass |
| Frequency | 20 - 20000 | Hz |
| Bandwidth | 5 - 200 | 1/100th octave |
| Gain | -20 - 20 | dB |
| Slope | 1 - 1.2 * Gain for shelving filters, 12 or 24 for Linkwitz-Riley filters | dB per octave |
| Enable | 0 - 1 | 0 = filter disabled<br>1 = filter enabled |

The paramters are specified in the order shown and are separated by commas. The wildcard character, *, may not be used for any of the parameters listed above. If an attempt is made to set one of the parameters outside the valid range, the command will fail and return an error message of `ERROR#002`.

Not all of the parameters are valid for each of the filter types. The following table shows which parameters are used in each of the filter types.

| Filter Type | Bandwidth | Frequency | Gain | Slope |
|---|---|---|---|---|
| Parametric (1) | Yes | Yes | Yes | No |
| Low Shelf (2) | No | Yes | Yes | Yes |
| High Shelf (3) | No | Yes | Yes | Yes |
| Lowpass (4) | No | Yes | No | No |
| Highpass (5) | No | Yes | No | No |
| Linkwitz-Riley Lowpass (6) | No | Yes | No | Yes |
| Linkwitz-Riley Highpass (7) | No | Yes | No | Yes |

When a parameter is invalid for a given filter type, the parametric EQ commands will still set that parameter, however its value will not be used for filter computations. This valid/invalid parameter information above is provided mainly as guidelines for programmers writing control software for the Vortex. If a parameter is not valid for a specific filter type, then the control for editing that parameter should be disabled or removed when that filter type is selected.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| `F01PEQOA3,2,1,1250,40,-20,1,1` | Set the parametric EQ on output 3 , band 2 to the following parameters: type = parametric filter, frequency = 1250 Hz, bandwidth = 40 1/100th octaves, gain = -20 dB, enabled = yes, slope = 1 dB per octave. The value of the slope parameter is irrelevant for this type of filter, we could have set it to anything. | `F01PEQOA3,2,1,1250,40,-20,1,1` |
| `F01PEQOAB,1,?` | Query the current parameter settings for the parametric eq filter on output B , band 1. | `F01PEQOAB,1,t,f,b,g,s,e` , where the parameters `t`, `f`, `b`, `g`, `s`, and `e` correspond to the current settings of the type, frequency, bandwidth, gain, slope, and enabled parameters, respectively. If this filter was set to the same parameters as given in the above example, the status message would be `F01PEQOAB,1,1,1250,40,-20,1,1` . |

## 8.111. `PEQOB` -- Set Bandwidth Parameter for Specified Parametric EQ Output Stage

This command sets or queries the bandwidth parameter (in 1/100th octaves) for the parametric equalizer (EQ) filters on output channels 1-8 and A-D . Each channel has five bands of parametric EQ that can be independently controlled.

The minimum and maximum values for the bandwidth parameter are 5 and 200, respectively.

This command has the same format and restrictions as the PEQOA command except that only one parameter, the bandwidth, is specified instead of all the parameters. See the PEQOA command for more information. To set the bandwidth parameter for an input parametric EQ filter, use the PEQIB command. To set all the parameters for an input parametric EQ filter, use the PEQIA command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| F01PEQOB3,2,40 | Set bandwidth of parametric EQ on output 3 , band 2 to 40 1/100th octaves. | F01PEQOB3,2,40 |
| F01PEQOBB,1,? | Query current bandwidth setting of parametric EQ on output B , band 1. | F01PEQOBB,1,b , where b is the current setting of the bandwidth parameter for the parametric EQ on output B , band 1. If the bandwidth parameter of this filter is set to the same value given in the example above, then the status message will be F01PEQOBB,1,40 . |

## 8.112. PEQOE -- Set Enabled Parameter for Specified Parametric EQ Output Stage

This command sets or queries the enabled parameter (0 = filter is disabled, 1 = filter is enabled) for the parametric equalizer (EQ) filters on output channels 1-8 and A-D . Each channel has five bands of parametric EQ that can be independently controlled.

This command has the same format and restrictions as the PEQOA command except that only one parameter, the enabled status, is specified instead of all the parameters. See the PEQOA command for more information. To set the enabled parameter for an input parametric EQ filter, use the PEQIE command. To set all the parameters for an intput parametric EQ filter, use the PEQIA command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| F01PEQOE3,2,1 | Enable parametric EQ on output 3 , band 2. | F01PEQOE3,2,1 |
| F01PEQOEB,1,? | Query enabled status of parametric EQ on output B , band 1. | F01PEQOEB,1,e , where e is the current setting of the enabled parameter for the parametric EQ on output B , band 1. If the enabled parameter of this filter is set to the same value given in the example above, then the status message will be F01PEQOEB,1,1 . |

## 8.113. `PEQOF` -- Set Frequency Parameter for Specified Parametric EQ Output Stage

This command sets or queries the frequency parameter (in Hz) for the parametric equalizer (EQ) filters on output channels 1-8 and A-D . Each channel has five bands of parametric EQ that can be independently controlled.

The minimum and maximum values for the frequency parameter are 20 and 20000, respectively.

For filter types that have a center frequency (e.g., parametric), this parameter specifies the center frequency. For filter types that have a cutoff frequency (e.g, highpass, lowpass, etc.), this parameter specifies the cutoff frequency.

This command has the same format and restrictions as the `PEQOA` command except that only one parameter, the frequency, is specified instead of all the parameters. See the `PEQOA` command for more information. To set the frequency parameter for an input parametric EQ filter, use the `PEQIF` command. To set all the parameters for an input parametric EQ filter, use the `PEQIA` command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| `F01PEQOF3,2,1250` | Set frequency of parametric EQ on output 3 , band 2 to 1250 Hz. | `F01PEQOF3,2,1250` |
| `F01PEQOFB,1,?` | Query current frequency setting of parametric EQ on output B , band 1. | `F01PEQOFB,1,f` , where `f` is the current setting of the frequency parameter for the parametric EQ on output B , band 1. If the frequency parameter of this filter is set to the same value given in the example above, then the status message will be `F01PEQOFB,1,1250` . |

## 8.114. `PEQOG` -- Set Gain Parameter for Specified Parametric EQ Output Stage

This command sets or queries the gain parameter (in dB) for the parametric equalizer (EQ) filters on output channels 1-8 and A-D . Each channel has five bands of parametric EQ that can be independently controlled.

The minimum and maximum values for the gain parameter are -20 and 20, respectively.

This command has the same format and restrictions as the `PEQOA` command except that only one parameter, the gain, is specified instead of all the parameters. See the `PEQOA` command for more information. To set the gain parameter for an input parametric EQ filter, use the `PEQIG` command. To set all the parameters for an input parametric EQ filter, use the `PEQIA` command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|

| | | |
|---|---|---|
| `F01PEQOG3,2,-20` | Set the gain of the parametric EQ on output 3 , band 2 to -20 dB. | `F01PEQOG3,2,-20` |
| `F01PEQOGB,1,?` | Query current gain setting of parametric EQ on output B , band 1. | `F01PEQOGB,1,g` , where `g` is the current setting of the gain parameter for the parametric EQ on output B , band 1. If the gain parameter of this filter is set to the same value given in the example above, then the status message will be `F01PEQOGB,1,-20` . |

## 8.115. `PEQOS` -- Set Slope Parameter for Specified Parametric EQ Output Stage

This command sets or queries the slope parameter (in dB per octave) for the parametric equalizer (EQ) filters on output channels 1-8 and A-D . Each channel has five bands of parametric EQ that can be independently controlled.

For the Linkwitz-Riley filters, the slope can either be 12 or 24 (dB/Octave).For the low shelf and high shelf filters, the minimum value for the slope parameter is 1, and the maximum value is 1.2 times the current value of the gain parameter. See the PEQIG command for information on the gain parameter.

This command has the same format and restrictions as the PEQOA command except that only one parameter, the slope, is specified instead of all the parameters. See the PEQOA command for more information. To set the slope parameter for an input parametric EQ filter, use the PEQIS command. To set all the parameters for an input parametric EQ filter, use the PEQIA command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| `F01PEQOS3,2,1` | Set the slope of the parametric EQ on output 3 , band 2 to 1 dB per octave. | `F01PEQOS3,2,1` |
| `F01PEQOSB,1,?` | Query current slope setting of parametric EQ on output B , band 1. | `F01PEQOSB,1,s` , where `s` is the current setting of the slope parameter for the parametric EQ on output B , band 1. If the slope parameter of this filter is set to the same value given in the example above, then the status message will be `F01PEQOSB,1,1` . |

## 8.116. `PEQOT` -- Set Type Parameter for Specified Parametric EQ Output Stage

This command sets or queries the type parameter for the parametric equalizer (EQ) filters on output channels 1-8 and A-D . Each channel has five bands of parametric EQ that can be independently controlled.

The type parameter specifies what type of filter is executed for the specified band of the specified output channel. The different filter types and their corresponding values are shown in the table below.

| Filter Type | Value of Type Parameter |
|---|---|
| Parametric | 1 |

| Low Shelf | 2 |
|---|---|
| High Shelf | 3 |
| Lowpass | 4 |
| Highpass | 5 |
| Linkwitz-Riley Lowpass | 6 |
| Linkwitz-Riley Highpass | 7 |

This command has the same format and restrictions as the `PEQOA` command except that only one parameter, the filter type, is specified instead of all the parameters. See the `PEQOA` command for more information. To set the slope parameter for an input parametric EQ filter, use the `PEQIT` command. To set all the parameters for an input parametric EQ filter, use the `PEQIA` command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| `F01PEQOT3,2,1` | Set the type of the parametric EQ on output 3 , band 2 to parametric. | `F01PEQOT3,2,1` |
| `F01PEQOTB,1,?` | Query current type setting of parametric EQ on output B , band 1. | `F01PEQOTB,1,t` , where `t` is the current setting of the type parameter for the parametric EQ on output B , band 1. If the type parameter of this filter is set to the same value given in the example above, then the status message will be `F01PEQOTB,1,1` . |

## 8.117. `PHANTOM` -- Enable Phantom Power on Inputs 1-8

This command sets or queries the status of phantom power on each of the microphone inputs.

This command is a channel boolean command. See Section 6.3 and Section 6.1 for more information on this type of command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| `F01PHANTOM21` | Enable phantom power for input channel 2 . | `F01PHANTOM21` |
| `F01PHANTOM20` | Disable phantom power for input channel 2 . | `F01PHANTOM20` |
| `F01PHANTOM22` | Toggle phantom power state for input channel 2 . | `F01PHANTOM2x` , where `x` is `0` or `1` depending on the current state of phantom power on input channel 2 . |
| `F01PHANTOM2?` | Query state of phantom power for input channel 2 . | `F01PHANTOM2x` , where `x` is `0` or `1` depending on the current state of phantom power on input channel 2 . |
| `F01PHANTOM*0` | Disable phantom power for input channels 1-8. | `F01PHANTOM*00000000` |
| `F01PHANTOM*1` | Enable phantom power for input channels 1-8. | `F01PHANTOM*11111111` |

| F01PHANTOM*2 | Toggle phantom power for input channels 1-8. | F01PHANTOM*abcdefgh where a-h are each 0 or 1 depending on the current state of phantom power for the corresponding input channel. |
|---|---|---|
| F01PHANTOM*? | Query status of phantom power for input channels 1-8. | F01PHANTOM*abcdefgh where a-h are each 0 or 1 depending on the current state of phantom power for the corresponding input channel. |
| F01PHANTOM*11110000 | Enable phantom power for inputs 1-4 and disable phantom power for inputs 5-8. | F01PHANTOM*11110000 |

# 8.118. PING -- See Which Devices Are Present

When any Vortex device receives this command, it responds with a PONG status message. This is typically used by the host program to determine the types and IDs of all linked devices. When used in this manner, wildcard characters are usually given for the device type and ID so that all connected devices will respond.

In the following example, it is assumed that there are two EF2280's (device IDs 0 and 1), two EF2241's (device IDs 2 and 3), two EF2211's (device IDs 4 and 5), two EF2210's (device ID's 6 and 7), and two EF2201's (device ID's 0 and 1) linked together via EF Bus.

| Example | Description | Status Message |
|---|---|---|
| ***PING | Ping all connected device to determine the number, type, and ID of the connected devices. | F00PONG<br>F01PONG<br>B02PONG<br>B03PONG<br>S04PONG<br>S05PONG<br>Q06PONG<br>Q07PONG<br>T00PONG<br>T01PONG |

# 8.119. PRESETK -- Delete One or All Presets

This command kills (deletes) the specified preset. If a wildcard character (*) is used to specify the preset, then all presets are deleted.

If an attempt is made to delete one of the factory presets (in the range 0-15), the PRESETK command will fail and generate an ERROR#071. This is because the factory presets are read-only, thus the cannot be changed or deleted.

This command is saved to global non-volatile memory and is not part of a preset. Its value is saved each time it is changed. It will retain its value after power-down. Since this command writes to non-volatile memory, there will be a delay before an acknowledgment is returned.

| Example | Description | Status Message |
|---|---|---|

| | | |
|---|---|---|
| `F01PRESETK18` | Delete preset number 18. | `F01PRESETK18` |
| `F01PRESETK*` | Delete all presets. | `F01PRESETK*` |

## 8.120. `PRESETL` -- List All Commmands in a Preset

This command lists all the commands in a given preset. For example, assume that user preset 18 has been defined previously by the following command.

`F01PRESETW18`

Now, issuing the command:

`F01PRESETL18?`

Will result in a long sequence of status mesages reflecting all the data stored in the preset. At the end of the status messages, a final message of

`F01PRESETL18`

will be displayed.

## 8.121. `PRESETP` -- Set Which Preset Will Be Activated At Power-Up

This command sets or queries the power-up preset, which is executed each time the device powers up. The power-up preset must be one of the factory presets (0-15) or one of the user presets (16-47).

This command is saved to global non-volatile memory and is not part of a preset. Its value is saved each time it is changed. It will retain its value after power-down. Since this command writes to non-volatile memory, there will be a delay before an acknowledgment is returned.

| Example | Description | Status Message |
|---|---|---|
| `F01PRESETP16` | Set the power-up preset to be user preset 16. | `F01PRESETP16` |
| `F01PRESETP?` | Query the current power-up preset. | `F01PRESETPx` , where `x` is between `0` and `47` depending on which preset is currently set to be the power-on preset. |

## 8.122. `PRESETQ` -- Execute a Preset Quietly

This command executes the preset corresponding to the specified number, which must be between 0 and 47. The preset must either be a pre-defined factory preset (0 - 15) or a user preset (16 - 47) that has been previously defined via the <u>PRESETW</u> command. If the specified preset is empty, then an error condition of `ERROR#070` will be returned. If one or more of the commands in the preset generated errors, then an error condition of `ERROR#072` will be returned.

When the `PRESETQ` command executes, status messages for all of the commands in the preset will not be

generated. The `PRESETQ` command is provided so that a control system can execute a preset and not generated heavy data traffic from the resulting status messages. Of course, this assumes that the control system is not interested in any of the acknowledgements. The PRESETX command can be used to execute the preset without suppressing the status messages.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01PRESETQ18 | Execute preset number 18 without generating any status messages. | F01PRESETQ18 |

## 8.123. `PRESETW` -- Save a Preset

This command writes the current settings of the device into the user preset with the specified number. Since the factory presets (0 - 15) are read-only, an `ERROR#071` will be generated if they are specified in the `PRESETW` command.

Once a preset it saved, it can be recalled via the PRESETX and PRESETQ commands. The preset can also be set to be executed automatically at power-up via the PRESETP command.

This command is saved to global non-volatile memory and is not part of a preset. Its value is saved each time it is changed. It will retain its value after power-down. Since this command writes to non-volatile memory, there will be a delay before an acknowledgment is returned.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01PRESETW18 | Save the current device settings as preset 18. | F01PRESETW18 |

## 8.124. `PRESETX` -- Execute a Preset

This command executes the preset corresponding to the specified number, which must be between 0 and 47. The preset must either be a pre-defined factory preset (0 - 15) or a user preset (16 - 47) that has been previously defined via the PRESETW command. If the specified preset is empty, then an error condition of `ERROR#070` will be returned. If one or more of the commands in the preset generated errors, then an error condition of `ERROR#072` will be returned.

When the `PRESETX` command executes, status messages for all of the commands in the preset will be generated (unless acknowledgement mode is turned off). The status messages can be suppressed by using the PRESETQ command, which is exactly the same as the `PRESETX` command except that status messages are suppressed.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01PRESETX18 | Execute preset number 18. | Status messages for all the commands in the preset will be generated first, followed by the message: F01PRESETX18 |

## 8.125. `REFASGN` -- Assign AEC Reference to Input Channel

This command assigns an AEC reference signal (R1, R2, or EF Bus) to an input channel. Only inputs 1-8 have echo cancellation, so the command is only valid for those channels. Assigning 1 to an input channel selects AEC Reference 1 (R1). Assigning 2 to an input channel selects AEC Reference 2 (R2). Assigning 3 to an input channel selects the EF Bus Reference. The EF Bus reference is an AEC reference that is placed on the EF Bus by one (and only one) of the linked Vortexes. See the BUSREF command for more information about the EF Bus reference.

Although this command is a channel integer command, the increment and decrement operators (> and <) are not supported for this particular command.

This command is a channel integer command. See [Section 6.3](#) and [Section 6.2](#) for more information on this type of command. The minimum and maximum values for this command are 1 and 3, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| `F01REFASGN31` | Configure the AEC on input channel 3 to use AEC Reference 1 (R1). | `F01REFASGN31` |
| `F01REFASGN2?` | Query which AEC reference is currently being used for the AEC on input channel 2 . | `F01REFASGN2x` , where `x` is either `1`, `2`, or `3` depending on which AEC reference is currently being used. |
| `F01REFASGN*3` | Configure the AEC on all input channels (1-8) to use the EF Bus reference. | `F01REFASGN*ççççççç` |
| `F01REFASGN*?` | Query which AEC reference is currently being used for the AEC on each of the input channels. | `F01REFASGN*abcdefgh`, where `a-h` are each either à, å, or `ccedil`; depending on which AEC reference is currently being used on each of the input channels. |

## 8.126. `REFGAIN` -- Set Reference Output Gain

This command sets the effective output gain of the corresponding AEC reference. When doing volume control of room speakers, it is a good idea to adjust this reference gain along with the speaker output gain so that the AEC is aware of any volume changes. This will help prevent short echoes when volume changes are made.

There are two AEC references on the EF2280, EF2241, and EF2241, so there are two separate commands, `REFGAINR1` and `REFGAINR2`, for controlling the effective output gain of R1 and R2, respectively.

There is only one AEC reference on the EF2211 and EF2210, so there is only one command, `REFGAINR1` for controlling the effective output gain of R1.

This command was introduced in firmware version 2.5.0.

This command is an integer command. See [Section 6.2](#) for more information on this type of command. The minimum and maximum values for this command are -100 and 20, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| `F01REFGAINR16` | Set effective output gain of AEC reference 1 (R1) to 6 dB. | `F01REFGAINR16` |
| | Increase phone output gain by 3 dB. | , where `x` is the new value of the phone output gain. If this command was issued after the example above, then the status message would be . |

| | Query phone output gain. | , where x is the new value of the phone output gain. If this command was issued after the example above, then the status message would be . |

## 8.127. SGGAIN -- Set Gain of Signal Generator

This command sets the gain of the internal signal generator. The value of SGGAIN specifies the gain in dB applied to signal generator. A gain of 0 dB produces a signal at nominal level (-20 dB full scale).

This command is an integer command. See [Section 6.2](#) for more information on this type of command. The minimum and maximum values for this command are -100 and 20, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01SGGAIN6 | Set signal generator gain to 6 dB. | F01SGGAIN6 |
| F01SGGAIN>3 | Increase signal generator gain by 3 dB. | F01SGGAINx , where x is the new value of the signal generator gain. If this command was issued after the example above, then the status message would be F01SGGAIN9 . |
| F01SGGAIN? | Query signal generator gain. | F01SGGAINx , where x is the new value of the signal generator gain. If this command was issued after the example above, then the status message would be F01SGGAIN9 . |

## 8.128. SGMUTE -- Mute Signal Generator

This command controls the muting of the signal generator.

This command is a boolean command. See [Section 6.1](#) for more information on this type of command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01SGMUTE1 | Mute signal generator. | F01SGMUTE1 |
| F01SGMUTE0 | Unmute signal generator. | F01SGMUTE0 |
| F01SGMUTE2 | Toggle mute on signal generator. | F01SGMUTEx , where x is 0 or 1 depending on the current state of the signal generator mute. |
| F01SGMUTE? | Query signal generator mute. | F01SGMUTEx , where x is 0 or 1 depending on the current state of the signal generator mute. |

## 8.129. SGTYPE -- Set Type of Signal Produced by Signal Generator

This command sets the type of signal produced by the internal signal generator. Setting the type to `0` produces white noise, while setting the type to `1` produces pink noise.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| `F01SGTYPE0` | Configure signal generator to produce white noise. | `F01SGTYPE0` |
| `F01SGTYPE1` | Configure signal generator to produce pink noise. | `F01SGTYPE1` |
| `F01SGTYPE?` | Query current signal generator type. | `F01SGTYPEx` , where `x` is either `0` or `1` depending on whether the signal generator is currently configured to generate white or pink noise, respectively. |

## 8.130. `SSDELAY` -- Set Delay Between Screen Saver Screens

This command sets or queries the amount of time (in milliseconds) between each new screen of the LCD screen saver.

This command is an integer command. See [Section 6.2](#) for more information on this type of command. The minimum and maximum values for this command are 500 and 600000, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| `F01SSDELAY2000` | Set screen saver delay to 2 seconds (2000 ms). | `F01SSDELAY2000` |
| `F01SSDELAY>500` | Increase screen saver delay by 0.5 seconds (500 ms). | `F01SSDELAYx` , where `x` is the new value of the screen saver delay. If this command was issued after the example above, then the status message would be `F01SSDELAY2500` . |
| `F01SSDELAY?` | Query screen saver delay. | `F01SSDELAYx` , where `x` is the new value of the screen saver delay. If this command was issued after the example above, then the status message would be `F01SSDELAY2500` . |

## 8.131. `SSEN` -- Enable or Disable Screen Saver

This command controls whether or not the LCD screen saver is enabled.

This command is a boolean command. See [Section 6.1](#) for more information on this type of command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01SSEN1 | Enable screen saver. | F01SSEN1 |
| F01SSEN0 | Disable screen saver. | F01SSEN0 |
| F01SSEN2 | Toggle screen saver enabled state. | F01SSENx , where x is 0 or 1 depending on the current enabled state of the scrren saver. |
| F01SSEN? | Query screen saver enabled state. | F01SSENx , where x is 0 or 1 depending on the current enabled state of the scrren saver. |

## 8.132. SSSTART -- Set Idle Time Required for Screen Saver to Start

This command sets or queries the amount of time (in milliseconds) that the front panel buttons must be idle before the screen saver starts.

This command is an integer command. See Section 6.2 for more information on this type of command. The minimum and maximum values for this command are 500 and 600000, respectively.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|
| F01SSSTART2000 | Set screen saver start time to 2 seconds (2000 ms). | F01SSSTART2000 |
| F01SSSTART>500 | Increase screen saver start time by 0.5 seconds (500 ms). | F01SSSTARTx , where x is the new value of the screen saver start time. If this command was issued after the example above, then the status message would be F01SSSTART2500 . |
| F01SSSTART? | Query screen saver start time. | F01SSSTARTx , where x is the new value of the screen saver start time. If this command was issued after the example above, then the status message would be F01SSSTART2500 . |

## 8.133. SSTEXT -- Set Text to be Displayed by Screen Saver

This command sets or queries the text messages displayed by the screen saver. The arguments of this command are the screen number, the line number, and the text to be displayed. The screen number must be between 0 and 3. It indicates which message screen we are trying to set or query. Each message screen consists of two lines of text that can be up to 16 characters long. The second number indicates which line of the message we are tring to set or query. The line number must be 0 or 1 corresponding to the first and second line, respectively. Finally, the text argument is the actual text that will be displayed on the LCD. If the text is '?', then a query will be performed. If the text empty, then the message will be cleared. If both message lines are cleared for a given screen, then that screen will not be displayed by the screen saver. If all screens have been cleared, then the screen saver will never start (although it is easier just to use the SSEN command for this).

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---------|-------------|----------------|

| | | |
|---|---|---|
| `F01SSTEXT0,0,Play that` | Set the first line of the first screen to read "Play that". | `F01SSTEXT0,0,Play that` |
| `F01SSTEXT0,1,funky music` | Set the second line of the first screen to read "funky music". | `F01SSTEXT0,1,funky music` |
| `F01SSTEXT0,1,?` | Query the text of the second line of the first screen. | `F01SSTEXT0,1,<string>`, where `<string>` is the text of the second line of the first screen. If this command was issued after the example above, then the status message would be `F01SSTEXT0,1,funky music`. |

## 8.134. `SWRESET` -- Perform Soft Reset of System

Executing this command causes the Vortex to perform a software reset. The effect of the software reset is similar to cycling the power.

| Example | Description | Status Message |
|---|---|---|
| `F01SWRESET` | Perform a software reset. | No status message will be received. The Vortex will reset within a few seconds. |

## 8.135. `SWVER` -- Query Software Version

This command is used to query the firmware version of the Vortex.

| Example | Description | Status Message |
|---|---|---|
| `F01SWVER?` | Query current software version. | `F01SWVER<string>`, where `<string>` is the current sofware version. If the software version was 2.5.0, then the status message would be `F01SWVER2.5.0`. |

## 8.136. `VTXMODI` -- Enable VTX Mode on Specified Inputs

Vortex devices can be connected to a Polycom VTX 1000 in order to use the wideband capabilities of that device. When connecting to a VTX 1000, special proccessing is done on the Vortex in order to guarantee compatibility with the VTX 1000. This command enables processing for the VTX 1000 on the specified line input or inputs. Note that VTX 1000 mode can only be enabled on the line inputs (A-D) .

This command is a channel boolean command. See Section 6.3 and Section 6.1 for more information on this type of command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| `F01VTXMODIA1` | Enable VTX mode on input A. | `F01VTXMODIA1` |
| `F01VTXMODIB0` | Disable VTX mode on input B. | `F01VTXMODIB0` |

| | | |
|---|---|---|
| `F01VTXMODIB2` | Toggle status VTX mode on input B. | `F01VTXMODIBx` , where `x` is either `0` or `1` depending on the current status of VTX mode on input B. |
| `F01VTXMODIA?` | Query status of VTX mode on input A. | `F01VTXMODIAx` , where `x` is either `0` or `1` depending on the current status of VTX mode on input A. |
| `F01VTXMODI*1` | Enable VTX mode on inputs A-D . | `F01VTXMODI*1111` |
| `F01VTXMODI*0` | Disable VTX mode on inputs A-D . | `F01VTXMODI*0000` |
| `F01VTXMODI*1100` | Enable VTX mode on inputs A-B and disable VTX mode on inputs C-D . | `F01VTXMODI*1100` |
| `F01VTXMODI*2` | Toggle status of VTX mode on inputs A-D . | `F01VTXMODI*abcd` , where each of the letters (a, b, etc.) is either `0` or `1` depending of the current status VTX mode on the corresponding input. If this command was sent after the example above, then the status message would be `F01VTXMODI*0011` . |
| `F01VTXMODI*?` | Query status of VTX mode on inputs A-D . | `F01VTXMODI*abcd` , where each of the letters (a, b, etc.) is either `0` or `1` depending of the current status VTX mode on the corresponding input. If this command was sent after the example above, then the status message would be `F01VTXMODI*0011` . |

# 8.137. `VTXMODO` -- Enable VTX Mode on Specified Inputs

Vortex devices can be connected to a Polycom VTX 1000 in order to use the wideband capabilities of that device. When connecting to a VTX 1000, special proccessing is done on the Vortex in order to guarantee compatibility with the VTX 1000. This command enables processing for the VTX 1000 on the specified line output or outputs.

This command is a channel boolean command. See [Section 6.3](#) and [Section 6.1](#) for more information on this type of command.

This command is saved to non-volatile memory only as part of a preset. The state of this command will be restored after power-up only if a preset is saved and that preset is set to be the power-on preset.

| Example | Description | Status Message |
|---|---|---|
| `F01VTXMODO11` | Enable VTX mode on output 1. | `F01VTXMODO11` |
| `F01VTXMODOA0` | Disable VTX mode on output A. | `F01VTXMODOA0` |
| `F01VTXMODO12` | Toggle status VTX mode on output 1. | `F01VTXMODO1x` , where `x` is either `0` or `1` depending on the current status of VTX mode on output 1. |
| `F01VTXMODOA?` | Query status of VTX mode on output A. | `F01VTXMODOAx` , where `x` is either `0` or `1` depending on the current status of VTX mode on output A. |
| `F01VTXMODO*1` | Enable VTX mode on outputs 1-8, A-D . | `F01VTXMODO*111111111111` |
| `F01VTXMODO*0` | Disable VTX mode on outputs 1-8, A-D . | `F01VTXMODO*000000000000` |

| | | |
|---|---|---|
| `F01VTXMODO*111111110000` | **Enable VTX mode on outputs 1-8 and disable VTX mode on outputs A-D .** | `F01VTXMODO*111111110000` |
| `F01VTXMODO*2` | **Toggle status of VTX mode on outputs 1-8, A-D .** | `F01VTXMODO*abcdefghijkl` , where each of the letters (a, b, etc.) is either `0` or `1` depending of the current status VTX mode on the corresponding output. If this command was sent after the example above, then the status message would be `F01VTXMODO*000000001111` . |
| `F01VTXMODO*?` | **Query status of VTX mode on outputs 1-8, A-D .** | `F01VTXMODO*abcdefghijkl` , where each of the letters (a, b, etc.) is either `0` or `1` depending of the current status VTX mode on the corresponding output. If this command was sent after the example above, then the status message would be `F01VTXMODO*000000001111` . |