AXON DBU

AVC Extension Over Network
Dante™, Bluetooth®, USB Audio
Connectivity Interface

3rd Party Control API

1675 MacArthur Boulevard · Costa Mesa, CA 92626
Ph: 800/854-4079 or 714/957-7100 · Fax: 714/754-6174

www.qsc.com

Axon DBU Control Overview

The Attero Tech Axon DBU utilizes a JSON (JavaScript Object Notation) formatted control API. Full device control includes two parts, both defined by JSON schemas:

- Command protocol: This defines the JSON format of commands
- Device data model: This defines the values that can be set or retrieved

Transport

Commands are sent as UDP messages to destination port 49494 and IP address of the Ultimo processor.

Command protocol

A command consists of a JSON object containing one and only one of these keys:

- "get": reads data from the device
- "set": writes data to the device

"get" and "set" commands use JSON pointers (https://tools.ietf.org/html/rfc6901) to identify the target parameter or parameters within the data model. The pointers are strings similar to folder paths with each object key or array indices separated by a / character. The pointers are case sensitive and can point to anything from a specific key to data for a complete object.

"get" Command

A "get" command retrieves data from the device.
The target pointer can refer to any value including objects and arrays, though may return an error if the response would be too large for a single UDP packet.

Command Example:

```
{"get":"TARGET_PTR"}
```

"TARGET_PTR" is a JSON pointer to the desired parameter or set of parameters. There are no additional parameters.

Response Example:

```
{"result":"OK","cmd":"get","param":"TARGET_PTR","value":"PARAM_VALUE"}
```

"TARGET_PTR" is the JSON pointer from the original command.

"PARAM_VALUE" is the desired object requested

"set" Command

A "set" command writes data to the device. "set" commands can refer to any key including objects or arrays. However, if writing multiple objects, the entire "set" message must fit into a single UDP packet.

"set" operations have three mutually exclusive forms: by value, by adjustment, and by adjustment with wrapping.

- `"value"` takes the literal value to set the parameter to
- `"adjust"` takes an increment (positive or negative) to adjust the value by. Adjusting past the minimum or maximum values sets to those values.
- `"adjust_wrap"` acts the same as "adjust", but wraps rather than saturating at a limit.

The `"adjust"` and `"adjust_wrap"` variants are only defined for boolean, integer, number, and enumerated types. Enumerated types always take an integer increment, positive values moving to subsequent entries in the list and negative values moving to previous entries.

For `"adjust"` and `"adjust_wrap"`, boolean values are treated as enumerated values: `{"adjust": 1}` behaves like `{"set": true}`, `{"adjust": -1}` behaves like `{"set": false}`, `{"adjust_wrap": 1}` toggles the parameter.

***Note:** A set command can operate on a read-only object that contains writable values, provided only the writable fields are touched. Set commands that attempt to operate on read-only values will return an error. The read-only check is only performed on scalar (non array/object) values.*

Command Example:

```
{"set": "TARGET_PTR", "value": "SET_VALUE"}

{"set": "TARGET_PTR", "adjust": "ADJUST_VALUE"}

{"set": "TARGET_PTR", "adjust_wrap": "ADJUST_WRAP_VALUE"}
```

"TARGET_PTR" is the JSON pointer from the original command.

"SET_VALAUE", "ADJUST_VALE" and "ADJUST_WRAP" values are complete structures describing the data to be written.

Response Example:

The response to a set command includes the parameter pointer and the values that were applied. Note that these values may differ slightly from those in the set command due to rounding.

```
{"result"=>"OK", "cmd"=>"set", "param"=>TARGET_PTR, "value": "PARAM_VALUE"}

{"set": "/preset/dsp/dante_rx/0/gain", "value": -12}  ...Dante Rx Ch. 1 DSP gain is set to -12...

{"set": "/preset/dsp/dante_rx/0/gain", "adjust": 2}   ...Dante Rx Ch. 1 DSP Gain is adjusted by 2...
```

Tokens

All commands may optionally include a `"token"` value of up to 32 characters which, if supplied, is simply returned in the response. The sender may supply a unique token to serve a similar purpose as a sequence number and/or allow tracking of the state of a sequence of operations.

Get Command with Token

Command Example:

```
{"get": TARGET_PTR, "token": TOKEN}
```

Response Example:

```
{"result":"OK", "token":"TOKEN", "cmd":"get", "param":"TARGET_PTR", "value":"PARAM_VALUE"}
```

`TARGET_PTR` is a JSON pointer to the desired parameter. There are no additional parameters.

Set Command with Token

Command Example:

```
{"set":"TARGET_PTR","value": SET_VALUE,"token":"TOKEN"}

{"set":"TARGET_PTR","adjust":"ADJUST_VALUE","token":"TOKEN"}

{"set":"TARGET_PTR","adjust_wrap":"ADJUST_VALUE","token":"TOKEN"}
```

Response Example:

```
{"result":"OK","token":"TOKEN","cmd":"set","param":"TARGET_PTR","value": PARAM_VALUE}
```

Errors

Successful commands return a result of "OK", unsuccessful commands return a result of "ERROR", with the token, command ("set", "get"), and the parameter passed to the command (the JSON key).

Command Examples

Command: Get Ultimo MAC

```
{"get":"/info/ultimo_mac"}
```

Response:

```
{"result":"OK","cmd":"get","param":"/info/ultimo_mac","value":"00:1d:c1:00:00:00"}
```

Command: Mute Dante Rx Channel 1 (Index 0)

```
{"set":"/preset/dsp/dante_rx/0/mute","value":true}
```

Response:

```
{"result":"OK","cmd":"set","param":"/preset/dsp/dante_rx/0/mute","value":true}
```

Configure multiple DSP parameters for Dante Rx Channel 1 (Index 0):

```
{"set":"/preset/dsp/dante_rx/0","value":{"gain":0.0,”mute":false}}
```

Response:

```
{"result":"OK","cmd":"set","param":"/preset/dsp/dante_rx/0","value":{"gain":0.0,”mute":false}}
```

Axon DBU JSON Control Data Model

At the top level, the Axon DBU device data model is divided into "info", "status", "system", "preset", and "control" sections.

- `/info`: Read-only static device information, such as vendor ID or MAC addresses. This section is the same for all devices.
- `/status`: Read-only runtime device status information: uptime, IP addresses, etc.
- `/system`: System-wide configuration parameters. These parameters are automatically saved to flash.
- `/preset`: Configuration parameters that can be stored or restored from presets. These are loaded from preset 0 at startup and are not automatically saved. These can be saved to or loaded from the default preset by writes to `/control/save_preset` and `/control/load_preset`.
- `/control`: Non-persistent settings and special action-trigger fields.

/info

Static read-only device information

Pointer	Type	Format	Description
/info	object		
/info/product_name	string	Max 32 characters	Plain text product name (“DBU”)
/info/product_id	string	16 hexadecimal digits (0-9, a-f)	Numeric product ID (0000000000000022)
/info/vendor_id	string	Max 32 characters	Plain text vendor ID (“AtteroT”)
/info/protocol_version	string	Version format: x.y	Numeric protocol version
/info/data_model	string		Data model name/version. Example: "product_data_model_1.0.json"
/info/bootloader_version	string	Version format: x.y	Numeric control processor bootloader version
/info/firmware_version	string	Version format: x.y	Numeric control processor firmware version
/info/mcu_mac	string	6 colon-separated octets: "xx:xx:xx:xx:xx:xx"	MAC Address of the control processor
/info/bt_mac	string	6 colon-separated octets: "xx:xx:xx:xx:xx:xx"	MAC Address of the Bluetooth® interface
/info/ultimo_mac	string	6 colon-separated octets: "xx:xx:xx:xx:xx:xx"	MAC address of Dante Ultimo processor.
/info/capabilities	string	4 byte hex string, 0x prefixed	Manufacturer defined Dante capabilities field, pulled from Ultimo processor

The `/info/data_model` field specifies a versioned file name for the data model schema. The data model schema provides sufficient information for a program to simulate control of a device with minor manual coding to implement any special cases or needed side effects.

/status

Real time, read-only device status

Pointer	Type	Format	Description
/status/state	enum	"running", "updating", "error"	Application status
/status/post	integer	Range: -2147483648 to 2147483647	Power-on-self-test error codes. A value of "0" indicates no errors. See "POST Error Defintion" in the "Definitions" section
/status/uptime	integer	Range: 0 to 2147483647	Time since boot in seconds
/status/dante/device_name	string	Max 32 characters	The device's Dante name.
/status/dante/device_lock	enum	"unknown", "not_supported", "unlocked", "locked"	The device's Dante device lock status.
/status/dante/rx_status	array	Length: 4	Array of Dante receive channel status objects
/status/dante/rx_status/[0..3]	object		Dante receive channel status object
/status/dante/rx_status/[0..3]/status	integer	Range: 0 to 65535	Status of the given Dante receive channel. See "Dante RX Status Codes" in the "Definitions" section
/status/dante/rx_status/[0..3]/available	integer	Range: 0 to 255	Any non-zero value indicates the channel is available to receive audio.
/status/dante/rx_status/[0..3]/active	integer	Range: 0 to 255	Any non-zero value indicates channel activity.
/status/dante/channel_labels/	object		Object of receive and transmit channel labels
/status/dante/channel_labels/rx	array	Length: 4	Array of receive channel labels
/status/dante/channel_labels/rx/[0..3]	string	Max 32 Characters	The label of the given Dante receive channel
/status/dante/channel_labels/tx	array	Length: 4	Array of transmit channel labels
/status/dante/channel_labels/tx/[0..3]	string	Max 32 Characters	The label of the given Dante transmit channel
/status/usb	object		USB interface status
/status/usb/state	enum	"disconnected", "connected", "priority"	State of the USB interface
/status/usb/playback	object		USB playback interface status
/status/usb/playback/gain	number	Range-100 to 0	USB playback volume (dB)
/status/usb/playback/mute	boolean	true, false	USB playback mute
/status/usb/record	object		USB record interface status
/status/usb/record/gain	number	Range -100 to 0	USB record volume (dB)
/status/usb/record/mute	boolean	true, false	USB record mute
/status/voice_cue	array	Length: 4	Array of voice cue status objects. See "Voice Cue Indexes" in the "Definitions" section
/status/voice_cue/[0..3]	object		Voice cue status object
/status/voice_cue/[0..3]/valid	boolean	true, false	Indicates whether the stored audio file for the given voice cue is valid
/status/bluetooth/state	enum	"idle", "discoverable", "connected", "conflict", "error"	Bluetooth interface state
/status/bluetooth/list_size	integer	Range 0 to 8	Size of the Bluetooth® Pairing List
/status/bluetooth/list	array	Length: 8	Array of Bluetooth® pairing list device information objects
/status/bluetooth/list/[0..7]	object		Pairing list remote device information
/status/bluetooth/list/[0..7]/mac	string	6 colon-separated octets: "xx:xx:xx:xx:xx:xx"	MAC address for the given remote Bluetooth® device in pairing list

/status/bluetooth/list/[0..7]/name	string	Max 32 Characters	Friendly name of the remote Bluetooth® device in pairing list
/status/bluetooth/list/[0..7]/keep	boolean	true, false	Read/write value which indicates whether the remote Bluetooth® device should remain in the pairing list when the list exceeds the maximum length of 8 devices
/status/bluetooth/device	object		Information about the currently connected Bluetooth® device
/status/bluetooth/device/mac	string	6 colon-separated octets: "xx:xx:xx:xx:xx:xx" or “”	MAC address for the currently connected Bluetooth® device.
/status/bluetooth/device/name	string	Max 32 Characters	Friendly name of the currently connected remote Bluetooth® device
/status/bluetooth/device/rssi	integer	Range: -100 to 0	Received Signal Stregnth Indicator (dBm) of the currently connected remote Bluetooth® device. Nominal range of [-90..-30]. An RSSI value of -80 or below indicates a poor Bluetooth® connection.
/status/bluetooth/device/song	string	Max 64 Characters	Name of the selected song on the connected remote Bluetooth® device. Note: The DBU must have the Bluetooth® “media” profile enabled and the remote Bluetooth® device must support the AVRCP Bluetooth® profile.
/status/bluetooth/device/album	string	Max 64 Characters	Name of the selected album on the connected remote Bluetooth® device. Note: The DBU must have the Bluetooth® “media” profile enabled and the remote Bluetooth® device must support the AVRCP Bluetooth® profile.
/status/bluetooth/device/artist	string	Max 64 Characters	Name of the selected artist on the connected remote Bluetooth® device. Note: The DBU must have the Bluetooth® “media” profile enabled and the remote Bluetooth® device must support the AVRCP Bluetooth® profile.

/system

Pointer	Type	Format	Description
/system/notifications	object		Asynchronous notification settings
/system/notifications/enable	boolean	true, false	Enable/disable asynchronous notifications
/system/notifications/ip	IP string	xxx.yyy.aaa.bbb where x, y, a, b are 0 to 255	IP address to which the DBU will send asynchronous notifications.
/system/notifications/port	Integer	Range: 0 to 65535	IP port to which the DBU will send asynchronous notifications.
/system/metering	object		Metering message settings
/system/metering/ip	IP string	xxx.yyy.aaa.bbb where x, y, a, b are 0 to 255	IP address to which the DBU will send metering. messages
/system/metering/port	Integer	Range: 0 to 65535	IP port to which the DBU will send metering.messages
/system/function_gen	object		Function generator settings
/system/function_gen/enable	boolean	true, false	Enable/disable the function generator
/system/function_gen/frequency	integer	Range: 10 to 20000	Function generator frequency in Hz
/system/function_gen/amplitude	integer	Range -100 to 0	Function generator amplitude in dB
/system/usb	object		USB audio interface configuration
/system/usb/mode	enum	"speakerphone_no_ec", "speakerphone_ec", "speakerphone_telephony_no_ec", "speakerphone_telephony_ec", "2x2"	USB audio interface mode
/system/usb/priority	boolean	true, false	Enable/disable USB audio priority mode
/system/voice_cue	array		Array of voice cue configuration objects. See "Voice Cue Indexes" in the "Definitions" section
/system/voice_cue/[0..3]	object		Voice cue configuration object.
/system/voice_cue/[0..3]/enable	boolean	true, false	Enable/disable the given voice cue
/system/bluetooth/	object		Bluetooth® audio interface configuration
/system/bluetooth/enable	boolean	true, false	Enable/disable Bluetooth® interface
/system/bluetooth/name	string	Max 31 Characters	Friendly name of the Bluetooth® interface
/system/bluetooth/profile	enum	"media", "hfp", "both"	Bluetooth® profile(s) configuration
/system/bluetooth/connection_mode	enum	"basic", "reconnect", "exclusive"	Bluetooth® connection mode configuration
/system/bluetooth/button	boolean	true, false	Enable/disable the Bluetooth® pair/disconnect button on the front panel

/preset

Pointer	Type	Format	Description
/preset/dsp	object		DSP configuration
/preset/dsp/dante_rx	array	Length: 4	Array of Dante RX Channel DSP configuration objects
/preset/dsp/dante_rx/[0..3]	object		DSP configuration for given Dante RX channel
/preset/dsp/dante_rx/[0..3]/gain	number	Range: -100.0 to 20.0	Gain setting for given Dante RX channel in dB
/preset/dsp/dante_rx/[0..3]/mute	boolean	true, false	Mute setting for given Dante RX channel
/preset/dsp/dante_tx	array	Length: 4	Array of Dante TX Channel DSP configuration objects
/preset/dsp/dante_tx/[0..3]	object		DSP configuration for given Dante TX channel
/preset/dsp/dante_tx/[0..3]/gain	number	Range: -100.0 to 20.0	Gain setting for given Dante TX channel in dB
/preset/dsp/dante_tx/[0..3]/mute	boolean	true, false	Mute setting for given Dante TX channel
/preset/dsp/usb_playback	array	Length: 2	Array of USB Playback DSP configuration objects. Note: Index 0 is the left channel and Index 1 is the right channel.
/preset/dsp/usb_playback/[0..1]	object		DSP configuration for given USB playback channel
/preset/dsp/usb_playback/[0..1]/gain	number	Range: -100.0 to 20.0	Gain setting for given USB playback channel in dB
/preset/dsp/usb_playback/[0..1]/mute	boolean	true, false	Mute setting for given USB playback channel
/preset/dsp/usb_record	array	Length: 2	Array of USB Record DSP configuration objects. Note: Index 0 is the left channel and Index 1 is the right channel.
/preset/dsp/usb_record/[0..1]	object		DSP configuration for given USB playback channel
/preset/dsp/usb_record/[0..1]/gain	number	Range: -100.0 to 20.0	Gain setting for given USB record channel in dB
/preset/dsp/usb_record/[0..1]/mute	boolean	true, false	Mute setting for given USB record channel
/preset/dsp/bt_sink	array	Length: 2	Array of Bluetooth® Sink (RX) DSP configuration objects. Note: Index 0 is the left channel and Index 1 is the right channel.
/preset/dsp/bt_sink/[0..1]	object		DSP configuration for given Bluetooth® sink (RX) channel
/preset/dsp/bt_sink/[0..1]/gain	number	Range: -100.0 to 20.0	Gain setting for given Bluetooth® sink (RX) channel in dB
/preset/dsp/bt_sink/[0..1]/mute	boolean	true, false	Mute setting for given Bluetooth® sink (RX) channel
/preset/dsp/bt_source	array	Length: 1	Array of Bluetooth® Source DSP configuration objects. Note: the Bluetooth® source (TX) is mono so this array only has one element.
/preset/dsp/bt_source/0	object		DSP configuration for the Blueooth® source (TX) channel
/preset/dsp/bt_source/0/gain	number	Range: -100.0 to 20.0	Gain setting for the Bluetooth® source (TX) channel in dB
/preset/dsp/bt_source/0/mute	boolean	true, false	Mute setting for the Bluetooth® source (TX) channel

/preset/mixer

The Axon DBU features a 10x7 matrix mixer. Each input and output is assigned an index. This forms a two-dimensional array of cross-point values. Cross-points are indexed by output first and then input. The indexes are listed in the table below.

Pointer	Type	Format	Description
/preset/dsp/mixer/[Output Index]	array		Array of cross-points for each mixer input for the given mixer output
/preset/dsp/mixer/[Output index]/[Input Index]	number	Range: -100.0 to 20.0	Cross-point for the given output/input pair

Index	DSP Output
0	Dante Tx 1
1	Dante Tx 2
2	Dante Tx 3
3	Dante Tx 4
4	USB Record Left
5	USB Record Right
6	BT Source (Mono)

Index	DSP Input
0	Dante Rx 1
1	Dante Rx 2
2	Dante Rx 3
3	Dante Rx 4
4	USB Playback Left
5	USB Playback Right
6	BT Sink Left
7	BT Sink Right
8	Voice Cue
9	Function Generator

Command Examples:

Example 1:

Set the cross-point of Output 0 (Dante Tx 1) and Input 9 (Function Generator) to 4.0 dB

```
  {"set":"/preset/dsp/mixer/0/9","value": 4.0}
```

Example 2:

Set cross-points for Inputs 4 and 5 (USB Playback Left, USB Playback Right) to 0.0 dB and all other cross-points to -100.0 dB for Output 3 (Dante Tx 4)

```
  {"set":"/preset/dsp/mixer/3","value": [-100.0,-100.0,-100.0,-100.0,0.0,0.0,-100.0,-100.0,-100.0,-
100.0]}
```

/control

Pointer	Type	Format	Description
/control/reboot	boolean	true, false	Setting to any value initiates a reboot.
/control/defaults	boolean	true, false	Setting to any value resets all device parameters (system and all presets) to defaults.
/control/identify	boolean	true, false	Controls the identify functionality (typically blinking the power LED).
/control/fwdl	Reserved		
/control/fwupdate	Reserved		
/control/mfg_cmd	Reserved		
/control/req_metering	integer	Range: 0 to 2147483647	Request N metering updates. Updates are sent at 10 Hz. This field can be read to determine how many updates remain to be sent.
/control/load_preset	integer	0	Load the default preset. This preset is also loaded at device startup
/control/save_preset	integer	0	Save the default preset. This preset is loaded at device startup.
/control/hid/hook_switch	boolean	true, false	Issue a USB telephony hook-switch command to the connected USB device (Host)
/control/hid/phone_mute	boolean	true, false	Issue a USB telephony phone mute command to the connected USB device (Host)
/control/hid/output_volume	number	Range: -100.0 to 0.0	Issue a USB HID output volume command to the connected USB device (Host)
/control/hid/output_mute	boolean	true, false	Issue USB HID output volume command to the connected USB device (Host)
/control/voice_cue/[0..3]/play	boolean	true, false	Initiate playback of a voice cue audio file (if the voice cue audio file is valid)
/control/bluetooth/clear_list	boolean	true, false	Clear the Bluetooth® pairing list. If in exclusive mode, this will also clear the exclusive Bluetooth® device, allowing a new Bluetooth® device to be configured as exclusive.
/control/bluetooth/disconnect	boolean	true, false	Disconnect from the connected remote Bluetooth® device. This command has no effect if a Bluetooth® device is not currently connected.
/control/bluetooth/pair	boolean	true, false	Enables Bluetooth® pair/connect mode. This command has no effect if a Bluetooth® device is currently connected.
/control/bluetooth/play	boolean	true, false	Issues a "Play" command to the remote Bluetooth® device. Note: The DBU must have the Bluetooth® "media" profile enabled and the remote Bluetooth® device must support the AVRCP Bluetooth® profile.
/control/bluetooth/pause	boolean	true, false	Issues a "Pause" command to the remote Bluetooth® device. Note: The DBU must have the Bluetooth® "media" profile enabled and the remote Bluetooth® device must support the AVRCP Bluetooth® profile.
/control/bluetooth/next	boolean	true, false	Issues a "Next" command to the remote Bluetooth® device. Note: The DBU must have the Bluetooth® "media" profile enabled and the remote Bluetooth® device must support the AVRCP Bluetooth® profile.
/control/bluetooth/prev	boolean	true, false	Issues a "Previous" command to the remote Bluetooth® device. Note: The DBU must have the Bluetooth® "media" profile enabled and the remote Bluetooth® device must support the AVRCP Bluetooth® profile.
/control/bluetooth/vol_up	boolean	true, false	Issues a "Volume Up" command to the remote Bluetooth® device. Note: The DBU must have the Bluetooth® "media" profile enabled and the remote Bluetooth® device must support the AVRCP Bluetooth® profile.
/control/bluetooth/vol_down	boolean	true, false	Issues a "Volume Down" command to the remote Bluetooth® device. Note: The DBU must have the Bluetooth® "media" profile enabled and the remote Bluetooth® device must support the AVRCP Bluetooth® profile.

Asynchronous Notifications

Asynchronous notifications are sent to the IP address and port configured at "/system/notifications". Asynchronous notifications formatted as JSON messages in the following format. The parameter value is formatted as described in the above data model

```
{"param":"TARGET_PTR","value":"PARAM_VALUE"}
```

Pointer	Trigger
/status/usb/state	Triggers when a USB device connects or disconnects while USB priority mode is enabled
/status/bluetooth/state	Triggers when Bluetooth® state changes a. DBU enters Bluetooth® pairing/connect mode b. DBU exits Bluetooth® pairing/connect mode c. Remote Bluetooth® device successfully connects to DBU d. Remote Bluetooth® device fails to connect to DBU
/status/bluetooth/device/song	Triggers when the song on the remote Bluetooth® device changes. Note: The DBU must have the Bluetooth® "media" profile enabled and the remote Bluetooth® device must support the AVRCP Bluetooth® profile.
/status/bluetooth/device/album	Triggers when the album on the remote Bluetooth® device changes. Note: The DBU must have the Bluetooth® "media" profile enabled and the remote Bluetooth® device must support the AVRCP Bluetooth® profile.
/status/bluetooth/device/artist	Triggers when the artist on the remote Bluetooth® device changes. Note: The DBU must have the Bluetooth® "media" profile enabled and the remote Bluetooth® device must support the AVRCP Bluetooth® profile.
/control/req_metering	Triggers when metering request value is updated (from any source)

Metering Messages

Metering messages are sent to the IP address and port configured at "/system/metering" formatted in the binary format defined below. Metering updates are sent at a rate of 10 Hz (10 messages per second). The DBU will not send metering messages unless "/control/req_metering" is set to a non-zero value. The value of "/control/req_metering" is decremented each time a metering message is emitted. When "/control/req_metering" reaches zero, the DBU will cease to emit metering messages until the value is again set to non-zero value. In order to continuously receive metering information, the value "control/req_metering" must be periodically set.

```
0x00: uint8_t - Metering Message (0x81)
0x01: uint8_t - Protocol Version (0x01)
0x02: uint8_t - Number of Metering Samples (0x10 = 16)
0x03: int8_t - Dante RX 1 Metering Sample (-100 to 20)
0x04: int8_t - Dante RX 2 Metering Sample (-100 to 20)
0x05: int8_t - Dante RX 3 Metering Sample (-100 to 20)
0x06: int8_t - Dante RX 4 Metering Sample (-100 to 20)
0x07: int8_t - USB Playback Left Metering Sample (-100 to 20)
0x08: int8_t - USB Playback Right Metering Sample (-100 to 20)
0x09: int8_t - BT Sink Left Metering Sample (-100 to 20)
0x0A: int8_t - BT Sink Right Metering Sample (-100 to 20)
0x0B: int8_t - Voice Cue Metering Sample (-100 to 20)
0x0C: int8_t - Dante TX 1 Metering Sample (-100 to 20)
0x0D: int8_t - Dante TX 2 Metering Sample (-100 to 20)
0x0E: int8_t - Dante TX 3 Metering Sample (-100 to 20)
0x0F: int8_t - Dante TX 4 Metering Sample (-100 to 20)
0x10: int8_t - USB Record Left Metering Sample (-100 to 20)
0x11: int8_t - USB Record Right Metering Sample (-100 to 20)
0x12: int8_t - BT Source Metering Sample (-100 to 20)
```

Definitions

Power-On-Self-Test (POST) Error Definition

Bit Index	0	1	2	3	4	5
Error	Ultimo	DSP	Bluetooth	External Flash	Ethernet Switch	External Oscillator

Voice Cue Indexes

Index	Voice Cue
0	Bluetooth® Pairing/Connect Initiated
1	Bluetooth® Pairing/Connect Ended
2	Bluetooth® Pairing/Connect Successful
3	Bluetooth® Pairing/Connect Failed

Dante Rx Status Codes

Note: These status codes are provided by the Ultimo processor. Future updates to the Ultimo firmware may add additional status code definitions or deprecate current definitions.

Status Code	Definition
0	RX Channel is not subscribed
1	Remote transmitter/TX channel is not yet resolved
2	Remote transmitter/TX channel has been resolved, waiting to be processed
3	Error: Failed to resolve remote transmitter/TX channel
4	RX Channel is subscribed to one of the device’s own TX Channels (Feature not available on Axon DBU)
7	RX Channel is idle
8	RX Channel’s flow is currently being processed
9	RX Channel is subscribed via automatic configuration
10	RX Channel is subscribed via manual configuration
14	RX Channel’s flow has been manually configured
15	Error: The local device could not communicate with the remote transmitter
16	Error: The local device’s channel format does not match remote transmitter’s channel format
17	Error: The local device’s flow formats do not match remote transmitter’s flows
18	Error: The local device is out of resources (No more flows)
19	Error: The local device failed to configure the flow
20	Error: The remote transmitter is out of resources (No more flows)
21	Error: The remote transmitter failed to configure the flow
22	Error: The local device received a QoS failure
23	Error: The remote transmitter received a QoS failure
24	Error: The remote transmitter rejected the local device’s RX address
25	Error: The remote transmitter rejected the flow request as invalid
26	Error: The remote TX channel latency is too high
27	Error: The RX channel and TX channel are in different clock subdomains
28	Error: Attempt to use an unsupported feature
29	Error: All receive links are down
30	Error: All transmit links are down
31	Error: Cannot find suitable protocol for automatic configuration
32	Error: Invalid channel
33	Error: Transmitter scheduler failure
64	Error: Template device name mismatch
65	Error: Template flow/channel format mismatch
66	Error: Template multicast flow does not contain channel
67	Error: Template configuration mismatch error
68	Error: Unicast template is full
96	Error: Transmitter access control denied the request. Transmitter is locked
97	Transmitter access control request is in process
255	Error: Unexpected system failure