

STcontroller API UDP Examples

All commands and messages used between STcontroller and Studio Technologies' products can be recreated and transmitted with User Datagram Protocol (UDP) transmitter software. Studio Technologies has taken the time to test and outline the usage with a number of their products. The construction of the UDP message will also be discussed.

This document contains lists of tested settings controlled by third-party UDP software for the following products:

Model 207 eSports Console
Model 209 Talent Console
Models 232/234/236 Announcer's Consoles
Model 391 Dante Alerting Unit
Model 392 Visual Indicator Unit
Model 5205 Mic/Line to Dante Interface
Model 5364 Headset Interface
Model 5401A Dante Leader Clock
Model 5402 Dante Leader Clock with GNSS Synchronization
ZEVO audio dovery

Copyright © 2022 by Studio Technologies, Inc., all rights reserved
studio-tech.com

Table of Contents

[Usage](#)

[Testing](#)

[Model 207 eSports Console](#)

[Model 209 Talent Console](#)

[Model 232/234/236 Announcer's Consoles](#)

[Model 391 Dante Alerting Unit](#)

[Model 392 Visual Indicator Unit](#)

[Model 5205 Mic/Line to Dante Interface](#)

[Model 5364 Headset Interface](#)

[Device Specific Settings](#)

[Mic Pre Bus Settings](#)

[Channel Settings](#)

[Model 5364 Headphone Settings](#)

[Model 5401A Dante Leader Clock](#)

[Model 5402 Dante Leader Clock with GNSS Synchronization](#)

[ZEVO audio dobery](#)

[Appendix A—Using Elgato Stream Deck and Bitfocus Companion](#)

Usage

STcontroller communicates with our Dante® products using Audinate's Packet Bridge protocol which allows an OEM's CPU to receive UDP datagrams via the corresponding Dante interface. A reliable implementation of Packet Bridge requires the use and licensure of Dante API, however UDP datagrams sent to the appropriate address will suffice in this case. In order to construct a UDP message a 24-byte header must be concatenated with data specific to the device being transmitted to. If a packet sniffing tool is used to analyze messages sent to a device from STcontroller the header will be similar to the example below, but the example header can also be used in your own application. The example header is as follows:

```
0xFF 0xFF 0x00 <msg_len> 0x07 0xE1 0x00 0x00 0x90 0xB1 0x1C 0x5B 0xD2
0x85 0x00 0x00 0x53 0x74 0x75 0x64 0x69 0x6F 0x2D 0x54 [data]
```

msg_len is the combined length of the header and data and is the only modifiable value in the example header.

Following the header is the unique device data. It is indicated with the Studio Technologies' start byte 0x5A. It is typically followed by the specific command ID (cmd_id), its data length (cmd_data_len), setting ID (setting_id) and value (setting_val), and finally a crc (crc8). Here is the typical structure:

```
0x5A <cmd_id> <cmd_data_len> [<setting_id>, <setting_val>, ...] <crc8>
```

Note that multiple settings can be set at the same time if desired. crc8 is calculated as CRC-8/DVB-S2 and uses the Studio Technologies' start byte through the command data in its calculation.

The example command below is for toggling the Talkback button on the Model 209 Talent Console when the Talkback button is set for Latching. The setting ID and value can be found in Table 2.

```
0x5A 0x0D 0x02 0x0A 0x01 0x7C
```

If combined with the necessary header the complete message to be sent to the Model 209 is:

```
0xFF 0xFF 0x00 0x1E 0x07 0xE1 0x00 0x00 0x90 0xB1 0x1C 0x5B 0xD2 0x85
0x00 0x00 0x53 0x74 0x75 0x64 0x69 0x6F 0x2D 0x54 0x5A 0x0D 0x02 0x0A
0x01 0x7C
```

Note that msg_len is 0x1E, or 30 decimal, is the entire length of the message.

The message must be sent to the device's Dante IP address on port 8700. This can be found using Dante Controller. It is suggested that only one device should be transmitted to at one time and that there should be at least 200ms between each transmitted message to allow for ample processing time.

This approach is slightly different from STcontroller which creates a subscription to the device in order to transmit the message more reliably. The device will always acknowledge a received message, however this is to a multicast address. One can also confirm that setting messages have been received by periodically polling a device for its settings. In the case of the Model 209 the following command can be used:

```
0x5A 0x0C 0x96
```

The entire message will be:

```
0xFF 0xFF 0x00 0x1B 0x07 0xE1 0x00 0x00 0x90 0xB1 0x1C 0x5B 0xD2 0x85  
0x00 0x00 0x53 0x74 0x75 0x64 0x69 0x6F 0x2D 0x54 0x5A 0x0C 0x96
```

The response to this message will also be sent to a multicast address, but may be easier to identify using packet-sniffing software.

Testing

Using packet-sniffing software to identify and parse commands from STcontroller will help to test UDP transmitter solutions with Studio Technologies' STcontroller-compatible products. Contact support if you have questions on how to proceed or for example messages for your products.

Model 207 eSports Console

Supported in Application Firmware version 1.1 and later.

Table 1. Model 207 Remote Settings

Setting ID	Setting Name	Setting Values
-	Talkback Output State*	0x00 - Off 0x01 - Toggle 0x02 - On

* The implementation used with the Model 207 has depreciated and will be updated.

Model 209 Talent Console

Supported in Application Firmware version 1.03 and later.

Table 2. Model 209 Remote Settings

Setting ID	Setting Name	Setting Values
0x02	Remote Control Inputs 1 & 2	0x00 - Input 1 Off/Input 2 Off 0x01 - Input 1 On/Input 2 Off 0x02 - Input 1 Off/Input 2 On 0x03 - Input 1 On/Input 2 On
0x03	Tally Out	0x00 - Tally 1 and 2 Off 0x01 - Tally 1 On 0x02 - Tally 2 On
0x04	Encoder Position	0x00 - 0x1F
0x05	Encoder State	0x00 - Off 0x01 - On OR 0x02 - Pushed
0x06	Encoder Color*	RGB value sent (0 - 0xFF) <n8_red>, <n8_green>, <n8_blue>
0x07	Rx Ch1 Dim (steps - 1.5dB per step)	0x00 - 0x0A (0 - 15dB)
0x08	Rx Ch2 Dim (steps - 1.5dB per step)	0x00 - 0x0A (0 - 15dB)
0x09	Save Rx Ch Dim Values	0x00 - False 0x01 - True
0x0A	Main Output State	0x00 - Off 0x01 - Toggle 0x02 - On

*Does not have any impact on the current product.

Command Structure (without UDP header):

```
<start> <cmd_dev_spec> <settings_len> <setting_id> <setting_val>  
[<setting_id>, <setting_val>...] <crc8>
```

In this case, the command structure for setting all Rx Ch1 Dim to 15dB is:

```
0x5A 0x0D 0x02 0x07 0x0A 0x52
```

Model 232/234/236 Announcer's Consoles

Supported in Main Firmware version 1.00 and later.

Table 3. Model 232/234/236 Remote Settings

Setting ID	Setting Name	Setting Values
0x01	Microphone Preamplifier Gain	0x14-0x41 (20-65dB)
0x02	Microphone P48 Phantom Power	0x00 - Off 0x30 - On (48V)

These are settings within the Mic Pre Bus command sent to the Model 232/234/236 from STcontroller.

Command Structure (without UDP header):

```
<start> <cmd_mic_pre_bus> <bus_chan> <settings_len> <setting_id>  
<setting_val> <crc8>
```

In this case, the command structure for setting the Mic Pre Gain to 36dB is:

```
0x5A 0x12 0x00 0x02 0x01 0x24 0x3C
```

Model 391 Dante Alerting Unit

Supported in Application Firmware version 2.1 and later.

Table 4. Model 391 Remote Settings

Setting ID	Setting Name	Setting Values
0x19	Alerting State	0x00 - Off 0x01 - On

Command Structure (without UDP header):

```
<start> <cmd_system> <settings_len> <setting_id> <setting_val>  
[<setting_id>, <setting_val>...] <crc8>
```

In this case, the command structure for setting Alerting State to Off is:

```
0x5A 0x09 0x02 0x19 0x00 0xC5
```


Model 392 Visual Indicator Unit

Supported in Application Firmware version 1.00 and later.

Table 5. Model 392 Remote Settings

Setting ID	Setting Name	Setting Values
0x19	Active State On/Off	0x00 - Off 0x01 - On

Command Structure (without UDP header):

```
<start> <cmd_system> <settings_len> <setting_id> <setting_val>  
[<setting_id>, <setting_val>...] <crc8>
```

In this case, the command structure for setting Active State to On is:

```
0x5A 0x09 0x02 0x19 0x01 0x10
```

Model 5205 Mic/Line to Dante Interface

Supported in Application Firmware version 2.1 and later.

The Model 5205 has two microphone preamplifier channels. A channel number, microphone preamplifier gain, and/or P48 on/off status value can be specified in the command sent to the unit (zero indexed).

Table 6. Model 5205 Remote Settings

Setting ID	Setting Name	Setting Values
0x01	Microphone Preamplifier Gain	0x00 - 0dB 0x14 - 20dB 0x1E - 30dB 0x28 - 40dB 0x32 - 50dB 0x3C - 60dB
0x02	Microphone P48 Phantom Power	0x00 - Off 0x30 - On (48V)

These are settings within the Mic Pre Bus command sent to the Model 5205 from STcontroller.

Command Structure (without UDP header):

```
<start> <cmd_mic_pre_bus> <bus_chan> <settings_len> <setting_id>  
<setting_val> <crc8>
```

In this case, the command structure for setting the Ch 1 Mic Pre Gain to 40dB is:

```
0x5A 0x12 0x00 0x02 0x01 0x28 0x8D
```

Model 5364 Headset Interface

Supported in Application Firmware version 1.00 and later.

Like many Studio Technologies' STcontroller-enabled products, the Model 5364 supports a number of command types outlined in the STcontroller protocol. Typically, the Device Specific commands are the ones users are looking to access and listed in this document. However, since the Model 5364 is intended for remote control only, all the applicable commands are listed in this document including Device Specific, Microphone Bus, Headphone, and Channel Settings. Device Specific Settings Multiple settings can be sent in one message.

Table 7. Model 5364 Device Specific Remote Settings

Setting ID	Setting Name	Setting Values
0x00	Tx Ch1 Output State	0x00 - Off 0x01 - On 0x02 - Toggle
0x01	Tx Ch2 Output State	0x00 - Off 0x01 - On 0x02 - Toggle
0x02	Rx Ch1 Mix Level	0x00 - 0x1F
0x03	Rx Ch2 Mix Level	0x00 - 0x1F
0x04	Rx Ch3 Mix Level	0x00 - 0x1F
0x05	Rx Ch4 Mix Level	0x00 - 0x1F
0x06	Overall Level	0x00 - 0x1F

Command Structure (without UDP header):

```
<start> <cmd_dev_spec> <settings_len> <setting_id> <setting_val>  
[<setting_id>, <setting_val>...] <crc8>
```

In this case, the command structure for setting all Rx Mix Levels to Max is:

```
0x5A 0x0D 0x08 0x02 0x1F 0x03 0x1F 0x04 0x1F 0x05 0x1F 0x6C
```

Mic Pre Bus Settings

These settings are bus specific, however there is only one mic pre bus on the Model 5364. In this case, the channel index will be 0. Note only one setting can be sent per packet.

Table 8. Model 5364 Mic Pre Bus Remote Settings

Setting ID	Setting Name	Setting Values
0x0D	Headset Gain	0x12 - 18dB 0x18 - 24dB 0x1E - 30dB 0x24 - 36dB 0x2A - 42dB
0x0E	Headset Electret Power	0x00 - Off 0x05 - On (+5V)

Command Structure (without UDP header):

```
<start> <cmd_mic_pre_bus> <bus_chan> <settings_len> <setting_id>  
<setting_val> <crc8>
```

In this case, the command structure for setting the Mic Pre Gain to 36dB is:

```
0x5A 0x12 0x00 0x02 0x0D 0x24 0x2E
```

Channel Settings

There are four (4) channels in the Model 5364. They are indexed zero (0) to three (3).

Table 9. Model 5364 Channel Remote Settings

Setting ID	Setting Name	Setting Values
0x01	Receive Audio Headphone Routing	0x00 - Off 0x01 - Left 0x02 - Right 0x03 - Left and Right

Command Structure (without UDP header):

```
<start> <cmd_channel> <channel> <settings_len> <setting_id>  
<setting_val> <crc8>
```

In this case, the command structure for setting Channel 3 Headphone Routing to Left and Right is:

```
0x5A 0x14 0x02 0x02 0x01 0x03 0x3E
```

Headphone Settings

Table 10. Model 5364 Headphone Remote Settings

Setting ID	Setting Name	Setting Values
0x15	Sidetone Level	0x00 - Off 0x01 - Low 0x02 - Medium-Low 0x03 - Medium 0x04 - Medium-High 0x05 - High

Command Structure (without UDP header):

<start> <cmd_headphone> <settings_len> <setting_id> <setting_val> <crc8>

In this case, the command structure for setting the Sidetone Level to Medium-High is:

0x5A 0x05 0x02 0x15 0x04 0xD9

Model 5401A Dante Leader Clock

Supported in Application Firmware version 1.07 and later.

Model 5401A System Settings are read-only and can be polled periodically. More extensive status information and device configuration can be found in the web interface of the product.

Table 11. Model 5401A System Settings

Setting ID	Setting Name	Setting Values
0x1E	Set Device Status	0x0E - Current Clock Source 0x0F - Sync Input Status 0x10 - Sync Input Type 0x11 - Primary PTPv1 0x12 - Primary PTPv2 0x13 - Secondary PTPv1 0x14 - Secondary PTPv2
0x1F	Set Device Information	0x05 - Capabilities 0x06 - Management IP Address

Table 12. Model 5401A Device Status

Setting ID	Setting Name	Setting Values
0x0E	Current Clock Source	0x00 - Internal 0x01 - Sync Input 0x02 - Dante 0x03 - GNSS 0x80 - Internal (Failover Active) 0x81 - Sync Input (Failover Active) 0x82 - Dante (Failover Active) 0x83 - GNSS (Failover Active)
0x0F	Sync Input Status	0x00 - Unlocked 0x01 - Locked (Active) 0x02 - Idle 0x03 - Locked (Standby)
0x10	Sync Input Type	0x00 - Word Clock 0x01 - Video 0x02 - 10 MHz
0x11	Primary PTPv1	0x04 - Disabled 0x07 - Leader 0x0A - Follower 0x0C - Link Down 0x0D - Error
0x12	Primary PTPv2	See Primary PTPv1 (0x11)
0x13	Secondary PTPv1	See Primary PTPv1 (0x11)
0x14	Secondary PTPv2	See Primary PTPv1 (0x11)

Table 13. Model 5401A Device Information

Setting ID	Setting Name	Setting Values
0x05	Capabilities	Not Applicable (4 bytes)
0x06	Management IP Address	IP Address Octet Values (0x00 - 0xFF) <n8_oct3>, <n8_oct2>, <n8_oct1>, <n8_oct0>

In order to poll the System Settings of the Model 5401A, the Get Device Settings command should be sent to the device with the following Command Structure (without UDP header):

```
0x5A 0x0A 0x17
```

The following Response Structure (shown without UDP header) will be returned to a non-specific multicast address (see [Usage](#) section for more information):

```
0x5A 0x8A 0x00 0x10 0x1E* 0x0E 0x0E^ <src> 0x0F^ <sis> 0x10^ <sit> 0x11^  
<pp1> 0x12^ <pp2> 0x13^ <sp1> 0x14^ <sp2> 0x0C 0x1F* 0x0A 0x05^ <cap1>  
<cap2> <cap3> <cap4> 0x06^ <oct3> <oct2> <oct1> <oct0> <crc>
```

* Denotes System Setting ID

^ Denotes Sub-System Setting ID

Model 5402 Dante Leader Clock with GNSS Synchronization

Supported in Application Firmware version 1.04 and later.

Model 5402 System Settings are read-only and can be polled periodically. More extensive status information and device configuration can be found in the web interface of the product.

Table 14. Model 5402 System Settings

Setting ID	Setting Name	Setting Values
0x1E	Set Device Status	0x0E - Current Clock Source 0x0F - Sync Input Status 0x10 - Sync Input Type 0x11 - Primary PTPv1 0x12 - Primary PTPv2 0x13 - Secondary PTPv1 0x14 - Secondary PTPv2 0x15 - GNSS Status
0x1F	Set Device Information	0x05 - Capabilities 0x06 - Management IP Address

Table 15. Model 5402 Device Status

Setting ID	Setting Name	Setting Values
0x0E	Current Clock Source	0x00 - Internal 0x01 - Sync Input 0x02 - Dante 0x03 - GNSS 0x80 - Internal (Failover Active) 0x81 - Sync Input (Failover Active) 0x82 - Dante (Failover Active) 0x83 - GNSS (Failover Active)
0x0F	Sync Input Status	0x00 - Unlocked 0x01 - Locked (Active) 0x02 - Idle 0x03 - Locked (Standby)
0x10	Sync Input Type	0x00 - Word Clock 0x01 - Video 0x02 - 10 MHz
0x11	Primary PTPv1	0x04 - Disabled 0x07 - Leader 0x0A - Follower 0x0C - Link Down 0x0D - Error
0x12	Primary PTPv2	See Primary PTPv1 (0x11)
0x13	Secondary PTPv1	See Primary PTPv1 (0x11)

Table 15. Model 5402 Device Status - continued

Setting ID	Setting Name	Setting Values
0x14	Secondary PTPv2	See Primary PTPv1 (0x11)
0x15	GNSS Status	0x00 - Unlocked 0x01 - Locked (Active) 0x02 - Locking 0x03 - Idle 0x04 - Locked (Standby) 0x05 - Antenna Disconnected 0x06 - Antenna Error

Table 16. Model 5402 Device Information

Setting ID	Setting Name	Setting Values
0x05	Capabilities	Not Applicable (4 bytes)
0x06	Management IP Address	IP Address Octet Values (0x00 - 0xFF) <n8_oct3>, <n8_oct2>, <n8_oct1>, <n8_oct0>

In order to poll the System Settings of the Model 5402, the Get Device Settings command should be sent to the device with the following Command Structure (without UDP header):

```
0x5A 0x0A 0x17
```

The following Response Structure (shown without UDP header) will be returned to a non-specific multicast address (see [Usage](#) section for more information):

```
0x5A 0x8A 0x00 0x12 0x1E* 0x10 0x0E^ <src> 0x0F^ <sis> 0x10^ <sit> 0x11^
<pp1> 0x12^ <pp2> 0x13^ <sp1> 0x14^ <sp2> 0x15^ <gnss> 0x0C 0x1F* 0x0A
0x05^ <cap1> <cap2> <cap3> <cap4> 0x06^ <oct3> <oct2> <oct1> <oct0>
<crc>
```

* Denotes System Setting ID

^ Denotes Sub-System Setting ID

ZEVO audio dobry

Supported in MCU Firmware version 0.35 and later.

Table 17. dobry Remote Settings

Setting ID	Setting Name	Setting Values
0x00	Overall Level	0x00 - 0x1F
0x01	Sidetone Level	0x80 - 0x14
0x02	Remote Control Input	0x00 - Off 0x01 - On
0x03	Tally Out	0x00 - Tally 1 and 2 Off 0x01 - Tally 1 On 0x02 - Tally 2 On
0x04	Encoder Position	0x00 - 0x1F
0x05	Encoder State	0x00 - Off 0x01 - On OR 0x02 - Pushed
0x06	Encoder Color*	RGB value sent (0 - 0xFF) <n8_red>, <n8_green>, <n8_blue>

*Does not have any impact on the current product.

Appendix A—Using Elgato Stream Deck and Bitfocus Companion

The Elgato Stream Deck should be used in conjunction with Bitfocus Companion. This will give the device access to a number of modules, specifically the Generic TCP and UDP module. This will allow a Stream Deck button push to transmit UDP packets to the Studio Technologies' device.

When configuring the module, the following settings are required:

Target IP: the Studio Technologies' device's Dante IP address

Target Port: 8700

Connect with TCP/UDP: UDP

When configuring a button choose a KEY UP/OFF ACTION. This will ensure the command is only sent once and not overloading the device. The selected action should be [UDP module name]: Send Command. The configuration should be:

Delay: 0

Command: Hexadecimal values for the desired UDP packet (see above, must include header and data sections) without '0x', and each preceded by '%'; for example, if the packet is to be 0x00 0x01 0x02, the field should have %00%01%02

Command End Character: None