Digital Signage Device RS-232 Control Standard

Request for Comments - Revision 1.0, June. 15, 2006

Editor-In-Chief
Jeff Collard (Omnivex Corporation) jcollard@omnivex.com

Editors
Preferably 2-3 Display Manufacturers

POPAI Digital Signage Standards Committee

Abstract

This document is initiated by the POPAI Digital Signage Standards Committee and collaborated among many digital signage industry leaders for the purpose of forming a standard for remote management of display devices.

Effective digital signage installations require remote management capabilities of all hardware within the network. Most commercial display devices provide some form of remote management capability for either operating the device or to query the device status. This is one of several characteristics that differentiate these products from consumer units. Many of these devices support the use of RF control for local management of individual display devices but this does not lend itself to a networked system. Some devices also support network commands through the video input cable or an IP addressable port. The most common remote management method is via ASCII commands over a RS-232 port. Connectors vary between manufacturers and cable connections can be either a serial (straight) connection or a null (cross-over) connection. Some devices support daisy chaining of devices and have a facility to allow the user to address individual panels within a chain.

RS-232 commands are usually in the form of a desired action (simple on/off commands to more complex tiling commands) or a status queries about the state of a device. Not all commands may be accessible by the RS-232 command set.

Please send any comments on the standards to jcollard@omnivex.com *Any inquiry of the Digital Signage Standards Committee, please send to matt.nelson@avocent.com. Any inquiry of the POPAI membership, please contact jfinizio@popai.com,*

This document is based on an Internal Omnivex document that has been contributed to the POPAI Digital Standards Committee to establish a general standard for RS-232 control and is not based on any specific one type of hardware or any particular hardware manufacturer.

Document History

Revision	Date	Comment
1.0	07-15-06	Initial draft distributed for comments by members of the POPAI standards committee

Contents

1. Background

Most commercial hardware used in Digital Signage networks supports remote management through RS-232 control although few installations take full advantage of this capability. As these networks expand the need to properly manage systems becomes more important. Unfortunately, RS-232 capabilities vary dramatically between products and there is no common protocol to control hardware. Many digital signage users often purchase consumer based units with no RS-232 control capability with no appreciation of the problems that they will later encounter when they try to expand their networks. POPAI Digital Signage Standards committee has identified this requirement as a critical area to be addressed in order to for the industry to grow.

The reference system calls for each device to communicate with a central management console via an existing PC based Player or other network based hardware. This document defines the committee's recommended format for RS-232 communications.

A display device can be one of many display technologies (plasma display, LCD panel, projector, LED panel, video cube) but for the purposes of this standard, it is a device capable of displaying a PC VGA signal or television signal. A digital signage network may consist of any number or combination of these devices. RS-232 commands are hexadecimal communications to these devices instructed them to operate in specific manner or requesting information from the device on its current status. Other communication methods exist and have been adopted by some manufacturers but RS-232 is the most common method.

The standards provided here are intended to be used as a guideline highlighting the important aspects of central hardware management. It is versioned in a way that would allow future revision as the industry evolves over time. Like any other industry standard, it is created with the following requirements in mind:

- *Adaptability* – it can cover broadly what people care about and adapt to most of the existing implementation with little change
- *Manageability* – it can be used as the key information in managing a digital signage network
- *Extensibility* – it allows future expansion for various digital signage applications
- *Scalability* – its existence does not provide a hardship when the network scales large
- *Easy to implement* – the effort to generate such log and any value-added services (e.g. reporting, analytics) are reasonable

2. Introduction

As digital signage networks expand, the requirement for central management of hardware becomes more important. Digital signage networks may be comprised of various display technologies that needs to be managed under a common method. The RS-232 control command system recommended by POPAI Digital Signage Standards committee covers a wide-range of usage possibilities. It does not preclude the use of alternate control methods such as IR, IP Socket, VGA or other connection method. This standard is not intended to dictate specific strings for specific commands or to dictate what commands are required. It does attempt to establish guidelines or best practices on how certain commands are implemented in order to support remote operation of hardware and provide continuity for users.

3. Definitions

Command
A string of characters that are sent from a computer to a device to perform some function.

Device
The actual piece of hardware. Each device is of a particular model.

Model
A particular style or design of device. A user may have many devices, all of the same model. A manufacturer may make several models of plasma screens, for example, each with different features.

Parameter
The variable in a command or response that contains a property value.

Property
A feature of a device that can be changed such as power, brightness, or input source.

Query
A special type of command that requests feedback from a device.

Response
A set of characters that are sent from a device to a computer to report a status.

Value
The setting of a property such as on or off for power, a number such as 50 for brightness, or Video 1 for input source.

4. Communication Guidelines

The POPAI Digital Signage standards committee recognizes that different device models are targeted at different markets and therefore have different capabilities. Not all models will support a full contingent of properties. To address this we have compiled a list of general points and suggestions that can help avoid common pitfalls. These suggestions are implemented in the screen control serial protocol proposed in Appendix A.

1. All commands should set one property or perform one function and have no more than one parameter.
2. All values should be expressed in bytes.
3. All values should be positive numbers.
4. Toggle commands should not be used.
5. Every command should return a meaningful response.
6. Every property should support querying to return the current value.
7. Devices should support diagnostic queries.
8. Devices should allow bulk queries.
9. Devices should support auto notification.
10. Devices should have a serial exclusive mode.
11. No command should ever damage the device.
12. Suggested commands.

4.1. All commands should set one property or perform one function and have no more than one parameter.

Each property should have its own command. If one command controls more than one property, the user may affect a property that could already be set correctly. For example, if one command sets both the blue bias and blue gain properties and the user only wants to change the blue bias, they are forced to set the value for blue gain at the same time. This can cause undesired effects. It requires them to know what the value of blue gain property is in order to change the blue bias.

In order to provide full flexibility and optimal control, each property should be set by one unique command.

A partial exception to this rule is the reset command, which will restore several properties to their factory defaults at once. However, all properties that are affected by the reset should still have their own commands to support individual control. The reset command might have zero parameters.

4.2. All values should be expressed in bytes.

There are many different ways to encode values in a serial string. Some of these include expressing a number using ASCII characters (e.g. the number 50 represented as two characters: Hex 35 and Hex 30 which represent the characters '5' and '0'). This is inefficient however as each value requires several characters for transmission.

The best way to encode a value from the point of view of brevity is to express it in bytes (e.g. the number 50 represented in one character: Hex 32 which is Decimal 50). Values for on and off or selection between a number of inputs can usually be represented by numbers between 0 and 255 and would therefore only require one byte.

ON should always be represented by the value 01 and OFF by 00. These two values should never be reversed.

If a value needs more than one byte then more than one byte is used (e.g. the number 500 as two bytes: Hex 01 and Hex F4). Multi-byte values should always have the high order byte first, followed by the low order byte. (e.g. The number 500 should always be represented as 01 F4 and not F4 01).

A value should be a fixed size. This means that a value that might need two bytes should always use two bytes.

4.3. All values should be positive numbers.

Just as there are different ways of encoding values in a string, there are different ways of indicating a negative number. Rather than dealing with these, it is much simpler to only deal with positive values. Instead of having a value range between –32 and +32 for example, it could just as easily range between 0 and 64 with 32 as the midpoint.

4.4. Toggle commands should not be used.

A toggle command is one that changes the value of a property based on the current value of that property. An example is a sound mute toggle that turns the mute on if it is off, and off if it is on. Another example is a volume increment or decrement. The problem with these commands is that there may be no way of knowing the current value of the property. Since many devices support an IR control and front panel switches in addition to the serial commands, the software will not know if it should increase or decrease the volume if a customer wants it set at midpoint before switching inputs. In addition, it is much more straightforward to set the volume to a specific value in a single command rather than sending a number of increment or decrement commands.

A preferable way of implementing these same commands would be to have a command that sets the sound mute property with a parameter to deliver a value representing either on or off. Another command would set the volume and its value would be an absolute setting in the supported range.

4.5. Every command should return a meaningful response.

When a command is executed successfully or if the command fails, the software should be able to inform the user of the status. Therefore, there should be some feedback from the device after every command.

If a command is correct and understood by the device and the device was able to execute it, the device should send an acknowledgement (ACK).

On the other hand, when a command is not understood or is not executed successfully, the device should return a negative acknowledgement (NAK). The NAK should contain an error code to indicate why the command was unsuccessful. These error codes as a minimum should include the following results:

- invalid data received
- checksum invalid
- property identifier out of range
- parameter length invalid
- property value out of range
- wrong mode
- can't execute due to hardware problem

A NAK should not be returned from a command that sets the value of a property to the current value. For example, if a command tries to set the volume to 50 when it is already 50, an ACK should be returned rather than a NAK, as the command is valid, does not cause any harm, and can be executed.

It is not recommended to incorporate a heartbeat response. Some devices automatically send a string of characters every few seconds to indicate that the device is still connected. ACK and NAK responses provide the same functionality and do not add unnecessary network traffic for customers controlling their devices remotely. Similarly, the device should not send any unexpected serial data that must be filtered out by the software.

4.6. Every property should support querying to return the current value.

It should be possible for the software to find the value of any property in order to provide feedback to users. One way to find the value is for the software to request it. The device should support a query command that passes as its parameter a value giving the property identifier. The device should return a response with the current value.

4.7. Devices should support diagnostic queries.

In order to provide effective unattended management of the device, the software should be able to retrieve results from internal diagnostic tests of the device such as internal temperature, power supply output, fan rotation, etc. Although these states cannot be set by commands, they should be considered properties for consistency and the values should be available by query.

If a command is sent to the device that attempts to set the value of a diagnostic property, the device should return a NAK.

4.8. Devices should allow bulk queries.

To cut down on data traffic and provide quicker feedback, it should be possible to query many properties simultaneously.

Depending on the number of properties supported by the device, this may be implemented with one query command or several. In the event that there are too many properties to reasonably return them all at once, the properties should be grouped together logically; for example, all audio properties, all video properties, all diagnostic properties, etc.

4.9. Devices should support auto notification.

In order to provide immediate, accurate feedback on the status of the device, the device should support an auto notification mode. When this mode is turned on, the device should send a notification every time a property is changed using either the front panel controls or the IR remote control. This should include diagnostic properties such as fan failure.

This mode should support being turned on or off. Some users may not need this information and it could cause unnecessary network traffic in those installations where devices are being controlled remotely as well as locally.

4.10. Devices should have a serial exclusive mode.

In cases where a device is placed in an unattended location, it would be very advantageous for customers to be able to lock out the front panel controls and the IR remote control once the device is set up properly. This prevents passersby from changing the device and the device can be left unsupervised in confidence.

When in this mode, the device should turn off all onscreen indications such as "Video 1" when the input source changes or a gauge when setting brightness or any type of menu.

This mode could be turned off by a serial command, a recessed reset button, or possibly by cycling the power to the device.

4.11. No command should ever damage the device.

There should never be any way for a user to send serial data to the device that will cause permanent damage or harm to the device. At a minimum the device communications module should guard against receiving invalid data, series of commands in any order, and commands sent in rapid succession possibly before previous commands have been executed. For example, some devices will fail if they are rapidly powered on and off. Others can lock up and require a hard reset if invalid data is received.

4.12. Suggested Commands.

In addition to commands used to operate a device, the following commands make it easier to manage a network of display devices:

- Request manufacturer and model information.
- Request device serial number.
- Set and request a custom device name.
- A simple “are you there?” command that can be used to make sure a device is present and responding that works in all modes of operation and doesn’t have any side effects.

5 Device Addressing

Some devices support daisy-chained connections. This is a wiring scheme where the cable runs from a PC (may also be a Com Port Extender or IP addressable RS-232 Port) to the first device, then from that device to the next device and so on. Using this type of wiring it is possible to connect many devices to one serial port.

There are pros and cons to this type of arrangement. The biggest advantage is ease of cabling in most installations. The biggest disadvantage is single point of failure – if the cable from the PC to the first device is cut or disconnected, none of the devices will receive data.

Daisy-chaining is generally a good option to offer the customer. The customer can weigh the benefits at their location and decide whether to daisy-chain or not.

If a device supports daisy-chaining, it must provide a mechanism for the software to send commands to only one device out of the many connected to the single serial port. This is usually done through device addressing. The command would have an address parameter and only the device with the address in the command would execute the command. All the other devices would ignore the command.

At the time of installation, the user must be able to set the address of each device to ensure that each device in the chain has a unique identifier. These addresses are usually numeric and can be set a number of ways.

Some devices store their address in volatile memory that is lost when the device is unplugged or during a power failure. This can cause many problems. Therefore, device addresses should always be stored in non-volatile memory. This could be handled through electronic storage such as EEPROM or Flash memory set by front panel controls or IR control, or by a bank of DIP switches.

It is recommended that the device addresses be set by some means other than through software. Setting device addresses by software can cause problems if a customer later inserts or removes a device in the middle of the chain or changes cabling and tries to reassign the addresses.

In order to control all devices in the chain simultaneously, for example turning them all on at once, the device should support a global address that will apply to all devices in the chain. It is recommended that address zero be reserved for this purpose. Therefore, sending a command to address zero will have the same effect as sending that command to every device in the chain at the same time.

Aside from the address, there should be no difference between commands that are meant for a specific device and global commands meant for all devices.

6 Standard Commands

One of the goals of the POPAI Digital Signage Standards Committee is to provide a standard list of command names and properties so that customers can easily access and control a wide range of devices.

Digital Signage control systems attempt to use standard terms to provide a consistent user interface across different models. Having different terms in manufacturer's documentation can cause confusion. Along with each property name we have listed prospective values that a control system might present to the user.

Since not all device models will support every command, the list has been broken down into two tiers. We recommend that if a manufacturer chooses not to implement all commands, they should at least provide support for the tier 1 commands.

Tier 1 Commands – High Utility

- Power
  - On
  - Off
- Input Source

  The committee recommends using the following labels to identify the different inputs. Not all the labels need to be used.
  - Video 1, Video 2, Video 3, Video 4, Video 5
  - RGB 1, RGB 2, RGB 3, RGB 4, RGB 5
  - HDTV 1, HDTV 2, HDTV 3, HDTV 4, HDTV 5
- Screen Sizing

  Note that only a 4:3 aspect ratio input source will be affected by the different screen modes. A 16:9 aspect ratio input source will appear the same in both modes.
  - Normal (4:3 image shown as 4:3 with blank areas on sides, 16:9 shown full screen)
  - Full Screen (4:3 image stretched to 16:9 display, 16:9 shown full screen)
- Brightness
  - 0% to 100%
- Contrast
  - 0% to 100%
- Tint
  - 0% to 100%
- Sharpness
  - 0% to 100%
- Volume (if supported)
  - 0% to 100%
- Volume Mute (if supported)
  - On
  - Off
- Auto Notification
  - On
  - Off
- Serial Exclusive Mode
  - On
  - Off
- Query
  - Property Identifier
- Bulk Query
  - Property Group Identifier
- Reset

- Ping

Tier 2 Commands – Moderate Utility

- Treble
    - 0% to 100%
- Bass
    - 0% to 100%
- Sound Balance
    - 0% to 100%
- Horizontal Position
    - 0% to 100%
- Horizontal Size
    - 0% to 100%
- Vertical Position
    - 0% to 100%
- Vertical Size
    - 0% to 100%
- Clock Frequency
    - 0% to 100%
- Clock Phase
    - 0% to 100%
- Color System
    - Auto
    - NTSC (may be denoted by number i.e. 3 or 4)
    - PAL (may be denoted by number or letter i.e. 60 or M)
    - SECAM
- Color
    - 0% to 100%
- Color Temperature
    - Cool
    - Medium
    - Warm
- Red Bias
    - 0% to 100%
- Green Bias
    - 0% to 100%
- Blue Bias
    - 0% to 100%
- Red Gain
    - 0% to 100%
- Green Gain
    - 0% to 100%
- Blue Gain
    - 0% to 100%
- Gamma
    - 0% to 100%

Tier 3 Commands – Low Utility

- Tiling
    - Width of tiled wall in number of screens
    - Height of tiled wall in number of screens
    - Location of this screen in tiled wall horizontally.
    - Location of this screen in tiled wall vertically.

- Picture in Picture
  - On/Off
- Orbiting
  - On/Off
- Power Save
  - On/Off
- On Screen Menu
  - On/Off
- Language
  - English, French, German, Spanish, Italian, Japanese, Chinese, etc.

7 Connectivity and Cabling

For serial connection from a computer to a display screen, the POPAI Digital Signage Standards committee recommends manufacturers use RS-232 communications over a straight through serial cable with a 9 pin female DB9 connector and a DCE pin out on the display device.

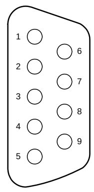
The main reason for this is that a display screen conforms to the definition of a DCE (data communications equipment) device more closely than a DTE (data terminal equipment) device. Once this distinction has been made, the connector, the pin out, and the cable are all defined.

The DB9 connector has replaced the DB25 as the de facto standard for serial connections. The standard DB9 connector on a DTE device (the computer) is a male connector and for a DCE device (the plasma) it is a female connector.

The wiring between a DTE device and a DCE device is a straight through cable. A 9 pin female to 9 pin male straight through cable is a very standard part and readily available in many lengths. This will make connection easy for the customer.

The only pins that need to be connected in order to communicate between the computer and the screen are Transmit, Receive, and Ground. The others are not needed given the relatively slow connection speed and small amounts of data that will be transmitted.

Female DB9 DCE pin numbering and definitions:



Pin number	Name
2	Transmit
3	Receive
5	Ground

The POPAI Digital Signage Committee recommended communication rates are:

- 9600 Baud
- No Parity
- 8 Data bits
- 1 Stop bit

Several manufacturers have adopted a null modem or cross over connection in which the Transmit and Receive connections are reversed. Although this connection provides assurance that the communications port on the device is used for its intended purpose, it requires special cables to be fabricated.

APPENDIX A. Sample Command Set and Strings

There are many different ways to build a protocol for communicating with a display screen. The committee offers the following protocol as a suggestion of one possible method. This command set takes all the points outlined in the Communications Guidelines into account and provides a completely defined system of communication. It also provides extensibility and room for growth to add new commands and properties for a wide range of models.

In this protocol the range used for parameter values such as brightness or volume are from 0 to 100. Using a range from 0 to 255 is certainly possible and does provide a finer adjustment of these parameters. However, from a user's point of view 0 to 100% is a much more natural and easily understood range than 0 to 255 and that is why we have used it.

Note

- All numbers shown are in the protocol description are in hexadecimal unless indicated otherwise. Percentages are shown in decimal.

General Command Form

- **STX**. The first character is the standard Start of Text (STX) character 02.
- **Address**. The second and third characters are the device address. If a device does not support addressing, these should be 00 and 00 to act as the global address.
- **Type**. The fourth character is the command type. There are four types of commands:
  - Command 01
  - Ack 02
  - Nak 03
  - Notification 04
- **Property ID**. The fifth and sixth characters are the property identifier. By using two bytes we have 65,536 different identifiers. Although this may seem excessive, the 256 different identifiers available in one byte are insufficient for some models.
- **Data Length**. The seventh character is the number of characters in the parameter data section.
- **Data**. Following the seventh character is the parameter data section that is as many characters long as the seventh character indicates. If the seventh character is 00 (as with the Reset command for example) then there are no characters in the parameter section. This parameter section contains the value for the property.
- **ETX**. The last character is a the standard End of Text (ETX) character 03. Note that there is no checksum in this protocol. Several checksum methods were considered before being abandoned. A checksum is typically used to detect changed bits due to line noise in the transmission. However, in a digital signage application, the cable run from the PC to the display screen is usually very short (less than 6 feet). Since not a lot of data is being sent, the low 9600 baud rate is used to further reduce the effects of interference. The big advantage of not having a checksum is that users can more easily hand code their own commands if necessary.

Sample:

STX	Address		Type	Property ID		Length	Data	ETX
02	00	00	01	00	01	01	01	03

In this example (Power On to everything), the Address characters are both 00, which is the global address for all devices connected to the serial port. The Type character is 01 to indicate the transmission is a command.

The Property ID characters indicate a property identifier of 1, which in this protocol is the value for power.

The Length character is a 1, which indicates that there is one character of data associated with this command.

That Data has a value of 1, which is the value for ON.

The last character is the ETX.

A1. Command Strings

In the following samples, the global device address of 00 00 is used. If the command was being sent to a device with an address, these characters would change along with the checksum.

The numbers in brackets are the hexadecimal value that the committee has associated with the property or the property value. Note that these values are used in the command string.

For the properties that support ranges such as brightness and volume, examples are provided for low, midrange, and high. Any intervening values can be deduced from these.

This section only covers the commands sent from the computer to the device. For responses from the device, see sections A2 – ACKs/Notifications and A3 – NAKs.

Tier 1 Commands

Power	(00 01)	
On	(01)	02 00 00 01 00 01 01 01 03
Off	(00)	02 00 00 01 00 01 01 00 03
Input Source	(00 02)	
Video 1	(01)	02 00 00 01 00 02 01 01 03
Video 2	(02)	02 00 00 01 00 02 01 02 03
Video 3	(03)	02 00 00 01 00 02 01 03 03
Video 4	(04)	02 00 00 01 00 02 01 04 03
Video 5	(05)	02 00 00 01 00 02 01 05 03
PC 1	(11)	02 00 00 01 00 02 01 11 03
PC 2	(12)	02 00 00 01 00 02 01 12 03
PC 3	(13)	02 00 00 01 00 02 01 13 03
PC 4	(14)	02 00 00 01 00 02 01 14 03
PC 5	(15)	02 00 00 01 00 02 01 15 03
HDTV 1	(21)	02 00 00 01 00 02 01 21 03
HDTV 2	(22)	02 00 00 01 00 02 01 22 03
HDTV 3	(23)	02 00 00 01 00 02 01 23 03
HDTV 4	(24)	02 00 00 01 00 02 01 24 03
HDTV 5	(25)	02 00 00 01 00 02 01 25 03
Screen Sizing	(00 03)	
Normal	(01)	02 00 00 01 00 03 01 01 03
Full Screen	(02)	02 00 00 01 00 03 01 02 03
Brightness	(00 04)	
Value = 0%	(00)	02 00 00 01 00 04 01 00 03
Value = 50%	(32)	02 00 00 01 00 04 01 32 03
Value = 100%	(64)	02 00 00 01 00 04 01 64 03
Contrast	(00 05)	
Value = 0%	(00)	02 00 00 01 00 05 01 00 03
Value = 50%	(32)	02 00 00 01 00 05 01 32 03
Value = 100%	(64)	02 00 00 01 00 05 01 64 03
Tint	(00 06)	
Value = 0%	(00)	02 00 00 01 00 06 01 00 03
Value = 50%	(32)	02 00 00 01 00 06 01 32 03
Value = 100%	(64)	02 00 00 01 00 06 01 64 03
Sharpness	(00 07)	
Value = 0%	(00)	02 00 00 01 00 07 01 00 03

Value = 50%	(32)	02 00 00 01 00 07 01 32 03
Value = 100%	(64)	02 00 00 01 00 07 01 64 03

Volume	(00 08)	
Value = 0%	(00)	02 00 00 01 00 08 01 00 03
Value = 50%	(32)	02 00 00 01 00 08 01 32 03
Value = 100%	(64)	02 00 00 01 00 08 01 64 03

Volume Mute	(00 09)	
On	(01)	02 00 00 01 00 09 01 01 03
Off	(00)	02 00 00 01 00 09 01 00 03

Auto Notification	(00 0A)	
On	(01)	02 00 00 01 00 0A 01 01 03
Off	(00)	02 00 00 01 00 0A 01 00 03

Serial Exclusive Mode	(00 0B)	
On	(01)	02 00 00 01 00 0B 01 01 03
Off	(00)	02 00 00 01 00 0B 01 00 03

Reset	(FF 00)	
		02 00 00 01 FF 00 00 03

This protocol reserves the 256 values that begin with FF for resets and other macro commands. Note that in the general reset there are no parameters and therefore the length character is 00 and the total command only has eight characters.

Ping	(FF 01)	
		02 00 00 01 FF 01 00 03

The Ping command is simply a way for the program to test the communication to the device. It doesn't change any property values. If the device is operating properly it will respond with an ACK.

Query	(FE 00)	
Power	(00 01)	02 00 00 01 FE 00 02 00 01 03
Input Source	(00 02)	02 00 00 01 FE 00 02 00 02 03
Brightness	(00 04)	02 00 00 01 FE 00 02 00 04 03

This protocol reserves the 256 values that begin with FE for queries. Note that in the property query there are two parameter characters for the property identifier and therefore the length character is 02 and the total command has nine characters. Only a few properties are shown as examples here, but all valid property identifier values should be queriable. The response to a query command back from the device will be a Notification.

Bulk Query	(FE 01)	
All	(00)	02 00 00 01 FE 01 01 00 03
Video	(01)	02 00 00 01 FE 01 01 01 03
Audio	(02)	02 00 00 01 FE 01 01 02 03
Other	(03)	02 00 00 01 FE 01 01 03 03
Diagnostic	(0F)	02 00 00 01 FE 01 01 0F 03

The response from the device to a bulk query is a series of Notification transmissions, one for each property defined in that group.

Tier 2 Commands

Treble	(00 21)	
Value = 0%	(00)	02 00 00 01 00 21 01 00 03
Value = 50%	(32)	02 00 00 01 00 21 01 32 03

Value = 100%	(64)	02 00 00 01 00 21 01 64 03
Bass	(00 22)	
Value = 0%	(00)	02 00 00 01 00 22 01 00 03
Value = 50%	(32)	02 00 00 01 00 22 01 32 03
Value = 100%	(64)	02 00 00 01 00 22 01 64 03
Sound Balance	(00 23)	
Value = 0%	(00)	02 00 00 01 00 23 01 00 03
Value = 50%	(32)	02 00 00 01 00 23 01 32 03
Value = 100%	(64)	02 00 00 01 00 23 01 64 03
Horizontal Position	(00 24)	
Value = 0%	(00)	02 00 00 01 00 24 01 00 03
Value = 50%	(32)	02 00 00 01 00 24 01 32 03
Value = 100%	(64)	02 00 00 01 00 24 01 64 03
Horizontal Size	(00 25)	
Value = 0%	(00)	02 00 00 01 00 25 01 00 03
Value = 50%	(32)	02 00 00 01 00 25 01 32 03
Value = 100%	(64)	02 00 00 01 00 25 01 64 03
Vertical Position	(00 26)	
Value = 0%	(00)	02 00 00 01 00 26 01 00 03
Value = 50%	(32)	02 00 00 01 00 26 01 32 03
Value = 100%	(64)	02 00 00 01 00 26 01 64 03
Vertical Size	(00 27)	
Value = 0%	(00)	02 00 00 01 00 27 01 00 03
Value = 50%	(32)	02 00 00 01 00 27 01 32 03
Value = 100%	(64)	02 00 00 01 00 27 01 64 03
Clock Frequency	(00 28)	
Value = 0%	(00)	02 00 00 01 00 28 01 00 03
Value = 50%	(32)	02 00 00 01 00 28 01 32 03
Value = 100%	(64)	02 00 00 01 00 28 01 64 03
Clock Phase	(00 29)	
Value = 0%	(00)	02 00 00 01 00 29 01 00 03
Value = 50%	(32)	02 00 00 01 00 29 01 32 03
Value = 100%	(64)	02 00 00 01 00 29 01 64 03
Color System	(00 2A)	
Value = Auto	(00)	02 00 00 01 00 2A 00 00 03
Value = NTSC	(01)	02 00 00 01 00 2A 00 01 03
Value = NTSC 3	(02)	02 00 00 01 00 2A 00 02 03
Value = NTSC 4	(03)	02 00 00 01 00 2A 00 03 03
Value = PAL	(04)	02 00 00 01 00 2A 00 04 03
Value = PAL 60	(05)	02 00 00 01 00 2A 00 05 03
Value = SECAM	(06)	02 00 00 01 00 2A 00 06 03
Color	(00 2B)	
Value = 0%	(00)	02 00 00 01 00 2B 01 00 03
Value = 50%	(32)	02 00 00 01 00 2B 01 32 03
Value = 100%	(64)	02 00 00 01 00 2B 01 64 03
Color Temperature	(00 2C)	
Value = Cool	(00)	02 00 00 01 00 2C 01 00 03
Value = Medium	(01)	02 00 00 01 00 2C 01 01 03
Value = Warm	(02)	02 00 00 01 00 2C 01 02 03
Red Bias	(00 2D)	

Value = 0%	(00)	02 00 00 01 00 2D 01 00 03
Value = 50%	(32)	02 00 00 01 00 2D 01 32 03
Value = 100%	(64)	02 00 00 01 00 2D 01 64 03
Green Bias	(00 2E)	
Value = 0%	(00)	02 00 00 01 00 2E 01 00 03
Value = 50%	(32)	02 00 00 01 00 2E 01 32 03
Value = 100%	(64)	02 00 00 01 00 2E 01 64 03
Blue Bias	(00 2F)	
Value = 0%	(00)	02 00 00 01 00 2F 01 00 03
Value = 50%	(32)	02 00 00 01 00 2F 01 32 03
Value = 100%	(64)	02 00 00 01 00 2F 01 64 03
Red Gain	(00 30)	
Value = 0%	(00)	02 00 00 01 00 30 01 00 03
Value = 50%	(32)	02 00 00 01 00 30 01 32 03
Value = 100%	(64)	02 00 00 01 00 30 01 64 03
Green Gain	(00 31)	
Value = 0%	(00)	02 00 00 01 00 31 01 00 03
Value = 50%	(32)	02 00 00 01 00 31 01 32 03
Value = 100%	(64)	02 00 00 01 00 31 01 64 03
Blue Gain	(00 32)	
Value = 0%	(00)	02 00 00 01 00 32 01 00 03
Value = 50%	(32)	02 00 00 01 00 32 01 32 03
Value = 100%	(64)	02 00 00 01 00 32 01 64 03
Gamma	(00 33)	
Value = 0%	(00)	02 00 00 01 00 30 01 00 03
Value = 50%	(32)	02 00 00 01 00 30 01 32 03
Value = 100%	(64)	02 00 00 01 00 30 01 64 03

Tier 3 Commands

Tiling – Width of Tiled Wall	(00 41)	
Value e.g. 2	(02)	02 00 00 01 00 41 01 02 03
Tiling – Height of Tiled Wall	(00 42)	
Value e.g. 2	(02)	02 00 00 01 00 42 01 02 03

In the example above, this would specify a 2x2 tiled matrix wall. To specify a single screen without any tiling, a 1x1 wall would be specified. Values outside the supported range would return a NAK.

Tiling – Horizontal Location	(00 43)	
Value e.g. 2	(02)	02 00 00 01 00 43 01 02 03
Tiling – Horizontal Location	(00 44)	
Value e.g. 1	(01)	02 00 00 01 00 44 01 01 03

In the example above, this would specify the top right screen in a 2x2 tiled matrix wall. Horizontal values start at 1 and increase from left to right. Vertical values start at 1 and increase from top to bottom. Values outside the defined tiled wall size would return a NAK.

Picture in Picture	(00 45)	
On	(01)	02 00 00 01 00 45 01 01 03
Off	(00)	02 00 00 01 00 45 01 00 03

Orbiting	(00 46)	
On	(01)	02 00 00 01 00 46 01 01 03
Off	(00)	02 00 00 01 00 46 01 00 03
Power Save	(00 47)	
On	(01)	02 00 00 01 00 47 01 01 03
Off	(00)	02 00 00 01 00 47 01 00 03
On Screen Menu	(00 48)	
On	(01)	02 00 00 01 00 48 01 01 03
Off	(00)	02 00 00 01 00 48 01 00 03
Language	(00 49)	
Value = English	(00)	02 00 00 01 00 49 01 00 03
Value = French	(01)	02 00 00 01 00 49 01 01 03
Value = German	(02)	02 00 00 01 00 49 01 02 03
Value = Spanish	(03)	02 00 00 01 00 49 01 03 03
Value = Italian	(04)	02 00 00 01 00 49 01 04 03
Value = Japanese	(05)	02 00 00 01 00 49 01 05 03
Value = Chinese	(06)	02 00 00 01 00 49 01 06 03

A2. ACKs/Notifications

ACKs

The device should return an acknowledgement for every valid command that is received and processed. It acts as a confirmation that the command was transmitted and received properly and the device property was set to the desired value.

The structure of an ACK is just like a command. The only thing that changes is the Type character which is 02 for an ACK. The ACK includes the address of the device or the global address, the property ID, data length, and property value the same as the command.

A sample is provided of the ACK string from a successful Power command:

Power	(00 01)	
On	(01)	02 00 00 02 00 01 01 01 03
Off	(00)	02 00 00 02 00 01 01 00 03

Query Responses

A response to a query is also an ACK. The difference is that the query response will contain more data than the query command did since it must include the property value. Therefore, the length character indicating the number of data characters will be higher and the value of the property will immediately follow the property identifier.

Here is a sample of a query command sent to the device asking for the input source and the response that the input source is set to HDTV 1:

Sent to device:	02 00 00 01 FE 00 02 00 02 03
Response:	02 00 00 02 FE 00 03 00 02 21 03

Here is another example of a query asking for the brightness and the response returning a value of 75% (decimal).

Sent to device:	02 00 00 01 FE 00 02 00 04 03
Response:	02 00 00 01 FE 00 03 00 04 4B 03

The response to a bulk query command is a number of query responses appended together, one for each of the properties in the requested bulk, each starting with STX 02 and ending with ETX 03.

Notifications

When Auto Notification is turned on, a notification should be sent every time a property value is changed by a method other than a serial command. This could be via the IR remote, front panel controls, or any other method. When a serial command is sent, only an ACK is received back. If Auto Notification is turned on and a user, for example, turned the power off using the IR remote, the device would transmit a notification to the computer to indicate the power property had changed value.

The structure of a notification is just like an ACK or a command. The only thing that changes is the Type character which is 04 for a notification.

A sample is provided of the notification string from a user turning the power on and off using front panel controls or the IR remote:

Power	(00 01)	
On	(01)	02 00 00 04 00 01 01 01 03

Off	(00)	02 00 00 04 00 01 01 00 03

A3. NAKs

The device should return a negative acknowledgement whenever it receives data it does not understand or cannot execute. The NAK should contain information about why the command could not be executed. This way the sending program can decide whether to retry the command or display an error to the user.

Note that sending a command to an invalid device ID will not cause anything to happen as none of the devices will be aware of the identifiers of other devices on the chain. It will therefore be up to the sending program to wait for a time out period and deduce what has gone wrong. This time out period may vary depending on whether the commands are being sent directly from a PC or if they are traveling over a network.

NAKs have the same basic format as a command with a couple of changes.

General NAK Form

- **STX**. Each NAK starts with the STX character 02.
- **Address**. This is the same as the command.
- **Type**. A NAK is indicated by type 03.
- **Error ID**. This is a single character indicating the type of error.
- **Data Length**. Same as in a command.
- **Data**. Depending on the Error ID, this may be nothing or may contain more information about the failure.
- **ETX**. Each NAK ends with the ETX character 03.

This protocol supports five types of errors that the device can return.

Invalid data received (00) 02 00 00 03 00 00 03

When some unknown characters are received by the device, or if only part of a command is sent, the device should respond with this NAK. It tells the sending program to retry the transmission. If the device only receives the first few characters of a command, it should time out after 2 seconds and send this NAK. Note that in a daisy-chain configuration, this NAK may be sent by every device in the chain since the device identifier may not be apparent.

Property identifier out of range (01) 02 00 00 03 01 02 AB 01 03

If the command received by the device is properly formed, is the correct length, and ends with an ETX, then the device can assume it has received a command that was transmitted correctly. However, if the property identifier is not one that is recognized by the device it should return this NAK. It makes sense to return the command to the sending program so that the program or user can see where the problem lies. The faulty property ID is sent as the data in the NAK. The error in the example command above is that the command attempted to set a property with an identifier of AB 01, which under this protocol has not yet been defined.

Property value out of range (02) 02 00 00 03 02 03 00 01 02 03

Similar to the previous NAK, this NAK is sent when the command is properly formed but there is an error in the command. In this case, the value that the command is attempting to assign to a property is not supported. The data includes both the property ID and the value the command was attempting to set. In the example above, the command is attempting to set the power property to a value of 2, which under this protocol is undefined.

Wrong mode (03) 02 00 00 03 03 00 05 32 03

This NAK is sent when the command is properly formed but the command is trying to do something that under some situations may be valid, but under the current mode of the device the command is invalid. Some devices have extended screen modes that are only valid when *the device is set to a particular input source. In the example above, if the device is in powered down standby mode, the command to change the contrast will be unsupported.*

Can't execute due to hardware problem (04) 02 00 00 03 04 03 00 01 01 03

If the command checksum is correct and the entire command is valid, but due to some device hardware problem the command cannot be executed, the device should return this NAK. In the example above, a valid power on command could not be executed due to a problem with the device. The software could then perform a diagnostic query to attempt to determine the fault.