NST Simple Control Protocol

Summary

This document describes the simple control protocol which can be used for 3rd party control of NST devices over ethernet networks. This protocol can be used by AMX®, Crestron® and similar systems and to implement a simple control interface or integrate with other control systems.

For any queries on information not contained in this document, please contact support@nstaudio.com

Table of Contents

Requirements

This document applies to the following NST devices and firmware versions :

D48 / ID48 firmware v0.7.0.3xx and above (also any OEM products using a DX1 DSP card)

D48S / ID48S firmware v0.2.0.1xx and above (also any OEM products using a DX2 DSP card)

VMX88 firmware v0.2.0.95 and above (including all FFA G3 amplifiers)

VMO16 firmware v0.2.0.37 and above

DM88 firmware v1.2.0.898 and above

Ethernet Configuration

IP Configuration

The NST device will allocate itself an IP address using a DHCP server if there is one available on the network, otherwise it will automatically allocate itself an IP address in the 169.254.xxx.xxx range.

If a fixed IP address is required for the device, this should be configured in the DHCP server on the network. The device MAC address is shown either on the System page of the device control panel in the NST D-Net software, or in the System menu on the front panel of a D48/D48S.

UDP Configuration

All messages should be sent to the device using UDP. Messages should be sent to the NST device on port 7090 and replies will be returned on the port that the request was sent on.

Message Format

Messages to NST devices are to be sent in binary format. Each message has a fixed header structure used to process the subsequent data in the message.

Data Types

The following data types are used in the messages :

Data Type	Number Bytes	Description
char	1	Single byte to be used as an unsigned integer (0..255) or an ASCII character
int	4	Four byte signed integer in little endian format (-5 = 0xFB, 0xFF, 0xFF, 0xFF)
uint	4	Four byte unsigned integer in little endian format (270 = 0x0E, 0x01, 0x00, 0x00)

Message Header

Messages to the device should always start with a 20 byte header of the following format :

Data Type	Label	Description
uint	MessageType	The ID of the message
uint	MessageSize	Size of the data which follows this header (max = 900)
uint	MessageCounter	A unique number sent from the host to uniquely identify this message (not compulsory, can be set to zero). This value will be returned with the ACK to identify the message being ACKed if multiple messages of the same type are sent in quick succession (eg. Gain changes).
char	MessageDirection	0x01 = Command, 0x02 = ACK success, 0x03 = ACK failure
char	N/A	Reserved – set to 0
char	N/A	Reserved – set to 0
char	N/A	Reserved – set to 0
char	N/A	Reserved – set to 0
char	N/A	Reserved – set to 0
char	N/A	Reserved – set to 0
char	N/A	Reserved – set to 0

Message Responses

All commands to an NST device will return an acknowledge message when complete. For “set” messages the ACK will be a message with the same MessageType and MessageCounter values, a MessageSize of zero and MessageDirection = 0x02. For “get” messages the return messages are described in the following command descriptions.

Command Reference

The commands below are described in with their MessageType uint number followed by a description of the command data format, then the reply data format. Commands with a MessageType below 1000 are “get” messages which are header only, then a description of the reply data format. Commands with a MessageType above 1000 are “set” messages which acknowledge success or failure of the command in the MessageDirection of the reply header.

1 – Get Device Information

Command Format :
Header only, no data.

Reply Format :
uint – Device type (see appendix)
uint – Number of input channels
uint – Number of output channels
char[50] – ASCII character string terminated with a '\0' character.

Example Hex Bytes:

01	00	00	00	00	00	00	00	12	34	56	11	01	00	00	00	00	00	00	00
MessageType				MessageSize				MessageCounter				Dir	Reserved						
Message Header																			

2 – Get Channel Names

Command Format :
Header only, no data.

Reply Format :
uint – Number of channels X (input + output)
uint – Number of characters Y per channel name
char[X][Y] – Array of X channels of Y ASCII character strings terminated with a '\0' character.

Example Hex Bytes:

02	00	00	00	00	00	00	00	12	34	56	11	01	00	00	00	00	00	00	00
MessageType				MessageSize				MessageCounter				Dir	Reserved						
Message Header																			

3 – Get Channel Gain Values

Command Format :
Header only, no data.

Reply Format :
uint – Total number of channels X (input, then output)
int[X] – Array of signed integers indicating gain value in 100ths of a dB (eg. 12.3dB = 0x000004CE)

Example Hex Bytes:

03	00	00	00	00	00	00	00	12	34	56	11	01	00	00	00	00	00	00	00
MessageType				MessageSize				MessageCounter				Dir	Reserved						
Message Header																			

4 – Get Channel Mute Values

Command Format :
Header only, no data.

Reply Format :
uint – Total number of channels X (input, then output)
char[X] – Array of char values indicating mute status (0x00 = OFF, 0x01 = ON)

Example Hex Bytes:

04	00	00	00	00	00	00	00	12	34	56	11	01	00	00	00	00	00	00	00
MessageType				MessageSize				MessageCounter				Dir	Reserved						
Message Header																			

5 – Get Matrix Gain Values

Command Format :
Header only, no data.

Reply Format :
uint – Total number of output channels X
uint – Total number of input channels Y
int[Y][X] – Array of signed integers indicating gain value in 100ths of a dB (eg. 12.3dB = 0x000004CE)

So gain order for 4 inputs, 8 outputs = Out1/In1, Out1/In2, Out1/In3, Out1/In4, Out2/In1, Out2/In2....

Example Hex Bytes:

05	00	00	00	00	00	00	00	12	34	56	11	01	00	00	00	00	00	00	00
MessageType				MessageSize				MessageCounter				Dir	Reserved						
Message Header																			

6 – Get Matrix Mute Values

Command Format :
Header only, no data.

Reply Format :
uint – Total number of output channels X
uint – Total number of input channels Y
char[Y][X] – Array of char values indicating mute status (0x00 = OFF, 0x01 = ON)

So mute order for 4 inputs, 8 outputs = Out1/In1, Out1/In2, Out1/In3, Out1/In4, Out2/In1, Out2/In2....

Example Hex Bytes:

06	00	00	00	00	00	00	00	12	34	56	11	01	00	00	00	00	00	00	00
MessageType				MessageSize				MessageCounter				Dir	Reserved						
Message Header																			

7 – Get Preset Status

Command Format :
Header only, no data.

Reply Format :
uint = Number of presets available X
char[X] – Array of usage flags (0x00 = unused, 0x01 = used)

Example Hex Bytes:

07	00	00	00	00	00	00	00	12	34	56	11	01	00	00	00	00	00	00	00
MessageType				MessageSize				MessageCounter				Dir	Reserved						
Message Header																			

8 – Get Preset Name

Command Format :
uint – Preset Index (zero based index so preset 1 = 0, preset 2 = 1 etc.)

Reply Format :
uint – Preset Index (zero based index so preset 1 = 0, preset 2 = 1 etc.)
char[64] – ASCII character string terminated with a '\0' character.

Example Hex Bytes:

08	00	00	00	04	00	00	00	12	34	56	11	01	00	00	00	00	00	00	00	02	00	00	00
MessageType				MessageSize				MessageCounter				Dir	Reserved							Preset Number			
Message Header																				Message Data			

9 – Get Dante Network Matrix Gain Values

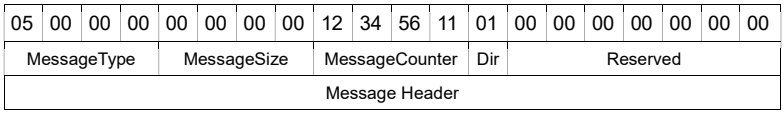
N.B. This command is only for devices with Dante network inputs (VMX88 and VMO16)

Command Format :
Header only, no data.

Reply Format :
uint – Total number of output channels X
uint – Total number of input channels Y
int[Y][X] – Array of signed integers indicating gain value in 100ths of a dB (eg. 12.3dB = 0x000004CE)

So gain order for 4 inputs, 8 outputs = Out1/In1, Out1/In2, Out1/In3, Out1/In4, Out2/In1, Out2/In2....

Example Hex Bytes:



10 – Get Dante Network Matrix Mute Values

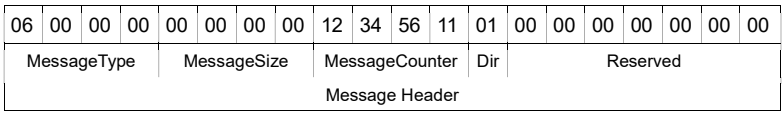
N.B. This command is only for devices with Dante network inputs (VMX88 and VMO16)

Command Format :
Header only, no data.

Reply Format :
uint – Total number of output channels X
uint – Total number of input channels Y
char[Y][X] – Array of char values indicating mute status (0x00 = OFF, 0x01 = ON)

So mute order for 4 inputs, 8 outputs = Out1/In1, Out1/In2, Out1/In3, Out1/In4, Out2/In1, Out2/In2....

Example Hex Bytes:



11 – Get DM Series Source Gain Values

N.B. This command is only for DM series devices

Command Format :
Header only, no data.

Reply Format :
uint – Total number of input channels X
uint – Total number of input source channels Y
uint – Total number of output channels J
uint – Total number of output source channels K
int[X][Y] – Array of signed integers (inputs) indicating gain value in 100ths of a dB
(eg. 12.3dB = 0x000004CE)
int[J][K] – Array of signed integers (outputs) indicating gain value in 100ths of a dB
(eg. 12.3dB = 0x000004CE)

So gain order for 8 inputs, 8 outputs, both with 4 source channels each = In1/src1, In1/src2, In1/src3, In1/src4, In2/src1, In2/src2....

Example Hex Bytes:

0B	00	00	00	00	00	00	00	12	34	56	11	01	00	00	00	00	00	00	00
MessageType				MessageSize				MessageCounter				Dir	Reserved						
Message Header																			

12 – Get DM Series Source Mute Values

N.B. This command is only for DM series devices

Command Format :
Header only, no data.

Reply Format :
uint – Total number of input channels X
uint – Total number of input source channels Y
uint – Total number of output channels J
uint – Total number of output source channels K
int[X][Y] – Array of char values (inputs) indicating mute status (0x00 = OFF, 0x01 = ON)
int[J][K] – Array of char values (outputs) indicating mute status (0x00 = OFF, 0x01 = ON)

So mute order for 8 inputs, 8 outputs, both with 4 source channels each = In1/src1, In1/src2, In1/src3, In1/src4, In2/src1, In2/src2....

Example Hex Bytes:

0C	00	00	00	00	00	00	00	12	34	56	11	01	00	00	00	00	00	00	00
MessageType				MessageSize				MessageCounter				Dir	Reserved						
Message Header																			

1001 – Recall Preset

Command Format :

uint – Index of preset to recall (zero based index so preset 1 = 0, preset 2 = 1 etc.)

Reply Format :

Header only, no data. MessageDirection value indicates success or failure. Failure usually indicates that there is no preset stored at that memory location

Example Hex Bytes:

E9	03	00	00	04	00	00	00	12	34	56	11	01	00	00	00	00	00	00	00	01	00	00	00
MessageType				MessageSize				MessageCounter				Dir	Reserved							Preset Number			
Message Header																				Message Data			

1002 – Set Gain Value

Command Format :

uint – X, the number of gains commands in this message

Then a sequence of X multiples of :

uint – Channel index, zero based

int – Gain value signed integer in 100ths of a dB (eg. 12.3dB = 0x000004CE). Max = +15dB. Min = -30dB.

The ability to send multiple gain commands in the same single message is intended to reduce the number of messages needed to carry out more complex operations and/or allow simple channel linking in the host.

Reply Format :

Header only, no data. MessageDirection value indicates success or failure. Failure will be indicated if any channel or gain value is out of range, even if other sections of the message have been applied successfully.

Example Hex Bytes:

EA	03	00	00	0C	00	00	00	12	34	56	11	01	00	00	00	00	00	00	00	01	00	00	00	04	00	00	00	CE	04	00	00
MessageType				MessageSize				MessageCounter				Dir	Reserved							Number of Gains				Channel Number				Gain Value			
Message Header																				Message Data											

1003 – Set Mute Value

Command Format :

uint – X, the number of mute commands in this message X

Then a sequence of X multiples of :

uint – Channel index, zero based

char - Mute status (0x00 = OFF, 0x01 = ON)

The ability to send multiple mute commands in the same single message is intended to reduce the number of messages needed to carry out more complex operations and/or allow simple channel linking in the host.

Reply Format :
Header only, no data. MessageDirection value indicates success or failure

Example Hex Bytes:

EB	03	00	00	09	00	00	00	12	34	56	11	01	00	00	00	00	00	00	00	01	00	00	00	05	00	00	00	01
MessageType				MessageSize				MessageCounter				Dir	Reserved							Number of Mutes				Channel Number				On
Message Header																				Message Data								

1004 – Set Matrix Gain Value

Command Format :
uint – X, the number of matrix gain commands in this message
Then a sequence of X multiples of :
uint – Output channel index, zero based
uint – Matrix input channel index, zero based
int – Gain value signed integer in 100ths of a dB (eg. 12.3dB = 0x0000300C). Max = 0dB. Min = -30dB.

The ability to send multiple matrix gain commands in the same single message is intended to reduce the number of messages needed to carry out more complex operations and/or allow simple channel linking in the host.

Reply Format :
Header only, no data. MessageDirection value indicates success or failure. Failure will be indicated if any channel or gain value is out of range, even if other sections of the message have been applied successfully.

Example Hex Bytes:

EC	03	00	00	10	00	00	00	12	34	56	11	01	00	00	00	00	00	00	00	01	00	00	00	05	00	00	00	01	00	00	00	00	00	00	00
MessageType				MessageSize				Message Counter				Dir	Reserved							Number of Gains				Output Channel				Matrix Input Channel				Gain Value			
Message Header																				Message Data															

1005 – Set Matrix Mute Value

Command Format :
uint – Number of mute commands in this message X
Then a sequence of X :multiples of :
uint – Output channel index, zero based
uint – Matrix input channel index, zero based
char - Mute status (0x00 = OFF, 0x01 = ON)

The ability to send multiple matrix mute commands in the same single message is intended to reduce the number of messages needed to carry out more complex operations and/or allow simple channel linking in the host.

Example Hex Bytes:

ED	03	00	00	0D	00	00	00	12	34	56	11	01	00	00	00	00	00	00	00	01	00	00	00	05	00	00	00	01	00	00	00	01
MessageType				MessageSize				MessageCounter				Dir	Reserved							Number of Mutes				Output Channel				Matrix Input Channel				On
Message Header																				Message Data												

Turns output channel 5, mute from input channel B on.

1006 – Global Mute

Command Format :

char - Mute status (0x00 = OFF, 0x01 = ON)

Reply Format :

Header only, no data. MessageDirection value indicates success or failure.

Example Hex Bytes:

EE	03	00	00	01	00	00	00	12	34	56	11	01	00	00	00	00	00	00	00	01	--	--	--
MessageType				MessageSize				MessageCounter				Dir	Reserved							Global Mute State			
Message Header																				Message Data			

1007 – Set Dante Network Matrix Gain Value

Command Format :

uint – X, the number of matrix gain commands in this message

Then a sequence of X multiples of :

uint – Output channel index, zero based

uint – Matrix input channel index, zero based

int – Gain value signed integer in 100ths of a dB (eg. 12.3dB = 0x0000300C). Max = 0dB. Min = -30dB.

The ability to send multiple matrix gain commands in the same single message is intended to reduce the number of messages needed to carry out more complex operations and/or allow simple channel linking in the host.

Reply Format :

Header only, no data. MessageDirection value indicates success or failure. Failure will be indicated if any channel or gain value is out of range, even if other sections of the message have been applied successfully.

Example Hex Bytes:

EF	03	00	00	10	00	00	00	12	34	56	11	01	00	00	00	00	00	00	00	01	00	00	00	05	00	00	00	01	00	00	00	00	00	00	00
MessageType				MessageSize				Message Counter				Dir	Reserved							Number of Gains				Output Channel				Matrix Input Channel				Gain Value			
Message Header																				Message Data															

1008 – Set Dante Network Matrix Mute Value

Command Format :

uint – Number of mute commands in this message X

Then a sequence of X :multiples of :

uint – Output channel index, zero based

uint – Matrix input channel index, zero based

char - Mute status (0x00 = OFF, 0x01 = ON)

The ability to send multiple matrix mute commands in the same single message is intended to reduce the number of messages needed to carry out more complex operations and/or allow simple channel linking in the host.

Example Hex Bytes:

F0	03	00	00	0D	00	00	00	12	34	56	11	01	00	00	00	00	00	00	00	01	00	00	00	05	00	00	00	01	00	00	00	01
MessageType				MessageSize				MessageCounter				Dir	Reserved							Number of Mutes				Output Channel				Matrix Input Channel				On
Message Header																				Message Data												

Turns output channel 5, mute from input channel B on.

1009 – Set DM Series Source Gain Value

Command Format :

uint – X, the number of source gain commands in this message

Then a sequence of X multiples of :

uint – Channel index, zero based

uint – Source index, zero based

int – Gain value signed integer in 100ths of a dB (eg. 12.3dB = 0x0000300C). Max = 0dB. Min = -30dB.

The ability to send multiple source gain commands in the same single message is intended to reduce the number of messages needed to carry out more complex operations and/or allow simple channel linking in the host.

Reply Format :

Header only, no data. MessageDirection value indicates success or failure. Failure will be indicated if any channel or gain value is out of range, even if other sections of the message have been applied successfully.

Example Hex Bytes:

F1	03	00	00	10	00	00	00	12	34	56	11	01	00	00	00	00	00	00	00	01	00	00	00	05	00	00	00	01	00	00	00	00	00	00	00
MessageType				MessageSize				Message Counter				Dir	Reserved							Number of Gains				Processing Channel				Source Channel				Gain Value			
Message Header																				Message Data															

1010 – Set DM Series Source Mute Value

Command Format :

uint – X, the number of source mute commands in this message

Then a sequence of X multiples of :

uint – Channel index, zero based

uint – Source index, zero based

char - Mute status (0x00 = OFF, 0x01 = ON)

The ability to send multiple source mute commands in the same single message is intended to reduce the number of messages needed to carry out more complex operations and/or allow simple channel linking in the host.

Reply Format :

Header only, no data. MessageDirection value indicates success or failure. Failure will be indicated if any channel or gain value is out of range, even if other sections of the message have been applied successfully.

Example Hex Bytes:

F2	03	00	00	0D	00	00	00	12	34	56	11	01	00	00	00	00	00	00	00	01	00	00	00	05	00	00	00	01	00	00	00	01
MessageType				MessageSize				MessageCounter				Dir	Reserved							Number of Mutes				Processing Channel				Source Channel				On
Message Header																				Message Data												

Appendix

Device Types

The standard device types returned by the “Get Device Information” message are :

201 – NST D48
202 – FFA Matrix DSP
204 – Funktion One F1-ID48
206 – FFA 4004DSP
207 – FFA 6004DSP
209 – NST ID48
210 – Funktion One F40QD
211 – Funktion One F60QD
225 – FFA 2004DSP
228 – FFA 8004DSP
230 – Funktion One F20QD
240 – DAS ID48S
231 – Funktion One F80QD
301 – NST D48S
302 – NST ID48S
311 – NST D48X
312 – NST ID48X
313 – FFA F48X
314 – FFA IF48X
501 – NST VMX88
502 – NST VMO16
505 – TPI Director
506 – TPI Producer
507 – TPI Director S
508 – NST VMX88L
521 – FFA 2004DSP G3D
524 – FFA 8004DSP G3D
525 – TPI PMX
600 – NST DM88
611 – FFA TOUR1