

Automixer Protocol Description

**SDXI-U2 (Serial ASCII Data Exchange Interface)
Version 1.18**

Date: 08.06.2015

Document version

version	date	Name	Remark
1.14	30.11.2009 - 23.12.2009	R. Wuppinger	- initial version: after the SDXI-U2 documentation, Command set requires Automixer (SW Version 1.24 or higher)
1.15	22.01.2010	R. Wuppinger	- command "U2DMxDS=..." changed to "U2DMxDST=..." - added command "U2DMxDRI=..." to enable and disable the remote interface (SW Version 1.24 or higher)
1.16	10.04.2013	R. Wuppinger	- fixes and corrections in the manual text - changes supporting the new functions of the Automixer devices - document renamed from "SDXI U2 Protocol Description Vx.yy.doc" to "SDXI U2 Automixer Protocol Description Vx.yy.doc" - removed "Virtual Talk Station" commands - removed "Request Device Status" commands
1.17			(skipped)
1.18	08.06.2015		- added (again) "Request Device Status" commands - added Preset Commands - added Routing Commands - added Network Parameter Commands

Available Documents

Name	Remark
SDXI General Description	Describes the fundamental protocol parameters and the data elements of the SDXI protocol. Also contains the description of these telegrams, which are supported by all SDXI devices, e.g. software and hardware version.
SDXI AXENS + US1	Describes all telegrams used for the AXENS Central Unit (AXCU), the AXENS talking station (AXTS) and the Salve Clock System US1.
SDXI ASx	Describes all telegrams used to control the digital automatic mixer family.
SDXI-U2 Automixer <...>	Describes all ASCII based telegrams provided by the digital automatic mixer family

Distributed by:

© Stemin GmbH

Hauptstraße 25

D – 82549 Königsdorf

Phone: 0049 – (0)8179 – 931 10

FAX: 0049 – (0)8179 – 931 99

Email: info@stemin.comInternet: <http://www.Stemin.com>

Contents

This page is empty.....	4
1 General description.....	5
1.1 Protocol SDXI.....	5
1.2 Protocol Extension SDXI-U2 (ASCII Mode).....	5
1.3 Serial Parameters.....	5
1.4 Availability of Telegrams.....	5
1.5 Important: Enabling Remote Control first.....	5
1.6 Telegram Specification.....	6
2 Commands.....	7
2.1 Definition of General Command Parameter.....	7
2.2 Version Commands.....	9
2.2.1 Software Version („U2DM<d>SW?<CR>“).....	9
2.2.2 Hardware Version („U2DM<d>HW?<CR>“).....	9
2.3 Device Configuration.....	10
2.3.1 Set Baudrate („U2DM<d>COM<c>BD=<bd>“).....	10
2.3.2 Set Remote Interface („U2DM<d>DRI=<s>“).....	10
2.4 Encoder Commands.....	11
2.4.1 Encoder Request Value („U2DM<d>EN<e>?<CR>“).....	11
2.4.2 Encoder Rotate Left („U2DM<d>EN<e>RL<CR>“).....	11
2.4.3 Encoder Rotate Right („U2DM<d>EN<e>RR<CR>“).....	12
2.4.4 Encoder Set Value („U2DM<d>EN<e>=<ev><CR>“).....	12
2.5 Status Commands.....	13
2.5.1 Request Channel Status („U2DM<d>CS<c>?<CR>“).....	13
2.5.2 Set Channel Status Remote Update Time („U2DM<d>CS<c>=<t><CR>“).....	13
2.5.3 Channel Status Set Mute („U2DM<d>EN<c>MU=<mut><CR>“).....	14
2.6 Preset Commands.....	16
2.6.1 Preset Status („U2DM<d>PRE<p>?<CR>“).....	16
2.6.2 Preset Config („U2DM<d>PRE<p>=<sel>,<cfg>,<def><CR>“).....	16
2.7 Routing Commands.....	18
2.7.1 Get Routing Status („U2DM<d>RTO<c>?<CR>“).....	18
2.7.2 Set Routing of Channel („U2DM<d>RTO<c>=<out_rec_usb>,<dante><CR>“).....	19
2.8 Network Parameter Commands.....	20
2.8.1 Network Parameter Get („U2DM<d>NWP<para>?<CR>“).....	20
2.8.2 Network Parameter Get („U2DM<d>NWP<para>=<Oct1>,<Oct2>,<Oct3>,<Oct4>“).....	21

This page is empty.

1 General description

1.1 Protocol SDXI

The SDXI-U2 (**S**erial **D**ata **E**xchange **I**nterface) should allow interacting, controlling and maintaining a variety of different devices. The serial interface was designed without any higher protocol function like a handshake or something equal. This interface allows the communication of embedded devices with a PC and/or among each other. Every device, a PC or the embedded device is responsible for the correct receiving and interpreting of the telegrams. This refers also to the timing-behaviour, this means that the devices must be able to react and process to the telegrams during runtime, without validating the timing conditions, which could lead to the loss of telegrams. A loss of a telegram can also cause a loss of data because for some telegram a repetition of the telegram is not possible. If the external device is fast enough and is controlled by an interrupt driven interface there should be no problem to handle the telegrams.

1.2 Protocol Extension SDXI-U2 (ASCII Mode)

The SDXI protocol includes a second, higher level protocol interface called SDXI-U2 protocol, which allows the user to communicate and control the device with a simple terminal program using an ASCII-based interface language.

This SDXI-U2 protocol provides a subset of the regular SDXI telegrams respectively SDXI commands, to make the basic functions available for an easy access via various media control devices like AMX, Crestron or something similar.

1.3 Serial Parameters

To select a user specific baud rate, you can select a different baud rate using the according SDXI-U2 telegram. For detailed description, please refer to the manual.

Default parameters:

```
BAUDRATE: 9600
PARITY:    NONE (fixed)
STOPBIT:   1    (fixed)
DATABITS:  8    (fixed)
```

1.4 Availability of Telegrams

The availability of each telegram depends on optional and device specific software modules as well.

1.5 Important: Enabling Remote Control first

To **enable** the **remote control functionality**, it is first necessary to enable the remote interface of the device by sending a “**Set Remote Interface**” command to the device!

1.6 Telegram Specification

The following table shows the structure of the SDXI-U2 telegrams:

Byte Index	Byte Quantity	Meaning	value / example
0	1	start byte	'U' (0x55H): Start of Telegram
1	1	telegram version	'2' (0x32H): signals that a SDXI-U2 telegram follows (ASCII mode)
2	2 ... 3	Device Code	"DM" (Digital Mixer)
variable	1 ... 3	Device Address	"1", ..., "999"
variable	2 ... 3	Object Code	"EN": Encoder "COM": ComPort
variable	1 ... 3	Object Address	"0", ..., "255"
variable	2 ... 3	Command Code / Object Property	Command Codes: "RR": encoder turn right "RL": encoder turn left "MU": mute channel "BD": Baudrate
variable	1	Operator	'=' set operator '?' request operator
variable	1 ... 6	Parameter 1 (optional)	"-32767"..."32767"
variable	1	Seperator (optional)	','
variable	1 ... 6	Parameter 2 (optional)	"-32767"..."32767"
variable	1	Seperator (optional)	','
variable	1 ... 6	Parameter 3 (optional)	"-32767"..."32767"
variable	1	Seperator (optional)	','
variable	1 ... 6	Parameter 4 (optional)	"-32767"..."32767"
variabel	1	end byte	0x0d (HEX) End of Telegram CR

Example:

ASCII	"U2DM1EN2=35<CR>"	sets encoder 2 to value 35 (level)
HEX	55 32 44 4d 31 45 4e 32 3d 33 35 0d	

2 Commands

2.1 Definition of General Command Parameter

Parameter	Symbol	Range	Description
transmission to the device	➔		data will be send/transmitted to the device (device receives data)
transmission from the device	➔		data will be send/transmitted from the device (device transmits data)
Telegram definition	📄		Shows the telegram definition including abstract parameters as variables.
Example of the Telegram	📄		Shows an example telegram with real values sent to the device
Example of the Telegram	📄		Shows an example telegram with real values sent from the device
optional parameter	[<px>]		Parameters placed in square brackets are optional.
device address	<d>	0 ... 999	0 = reserved for broadcast address 1...999 = device addresses
encoder number	<e>	1 ... 16	<p><u>6 input channels:</u></p> <p>Enc 1 = In 1 Enc 3 = In 2 Enc 5 = In 3 Enc 7 = In 4 Enc 9 = In 5 Enc 11 = In 6 Enc 13 = Sys Ctrl Enc 15 = Out1 Enc 16 = Headphone</p> <p><u>12 input channels:</u></p> <p>Enc 1 = In 1 Enc 2 = In 2 Enc 3 = In 3 Enc 4 = In 4 Enc 5 = In 5 Enc 6 = In 6 Enc 7 = In 7 Enc 8 = In 8 Enc 9 = In 9 Enc 10 = In 10 Enc 11 = In 11 Enc 12 = In 12 Enc 13 = Sys Ctrl Enc 14 = Out1 Enc 15 = Out2 Enc 16 = Headphone</p>
encoder value	<ev>	0 ... 255	encoder value/setting according to the actual system control selection

Parameter	Symbol	Range	Description
system control selection	<scs>		<p>value (selection) of the system control encoder:</p> <p>6 input channels:</p> <p>1: = LEVEL 2: = TREBLE 3: = BASS 4: = LOW CUT 5: = LIMITER 6: = COMPRESSOR 7: = AUTOMIXING 8: = PRIORITY 9: = PAN / BALANCE 10: = DELAY 11: = ROUTING TO REC 12: = ROUTING TO USB 13: = EQUALIZER LOW 14: = EQUALIZER HIGH 15: = PRESET</p> <p>12 input channels:</p> <p>1: = LEVEL 2: = TREBLE 3: = BASS 4: = LOW CUT 5: = LIMITER 6: = COMPRESSOR 7: = AUTOMIXING 8: = PRIORITY 9: = PAN / BALANCE 10: = Delay 11: = ROUTING TO REC 12: = ROUTING TO USB 13: = EQUALIZER 14: = PRESET</p>
Mute State	<mut>	0, 1	Mute state of the according encoder/channel: 0: encoder/channel not muted 1: encoder/channel muted
"ON"-LED State	<onl>	0, 1	"ON"-LED state of the according encoder: 0: LED is Off 1: LED is On
"PEAK"-LED State	<pkl>	0, 1	"PEAK"-LED state of the according encoder: 0: LED is Off 1: LED is On
channel number	<c>	0 ... 16	1...16 = automixer 6 input channels 1...16 = automixer 12 input channels
channel level	<cl>	0 ... 45	Represents the input level at the according channel
channel status	<csx>	00000 ... 45111	<p>Represents the input level, the ON and the Peak LED state of channel x in the following format:</p> <p><cl><mut><onl><pkl></p> <p><cl>xxx : first two number represent <cl> xx<mutl>xx : third number represents <mut> xxx<onl>x : fourth number represents <onl> xxxx<pkl>: fifth number represents <pkl></p>

2.2 Version Commands

2.2.1 Software Version („U2DM<d>SW?<CR>“)





Dir		Telegram	Description
➔	📄	U2DM<d>SW?<CR>	This command requests the actual software version/info of the device.
➔	🖨️	U2DM1SW?<CR>	
➡	📄	U2DM<d>SW=<p1> , <p2> , <p3><CR>	Device returns the software version: <p1>: SW-Version: (e.g.: 102 = 1.02) <p2>: <u>Application Type</u> 6: Digital Automixer 12: Digital Automixer <p3>: OEM Version (reserved)
➡	⚖️	U2DM1SW=121 , 6 , 0<CR>	<p1>: SW-Version: 1.21 <p2>: Application: Digital Automixer <p3>: OEM Version: 0 (reserved)

2.2.2 Hardware Version („U2DM<d>HW?<CR>“)






Dir		Telegram	Description
➔	📄	U2DM<d>HW?<CR>	This command requests the actual hardware version/info of the device.
➔	🖨️	U2DM1HW?<CR>	
➡	📄	U2DM<d>HW=<p1> , <p2> , <p3> , <p4> , <p5><CR>	Device returns the hardware version/info: <p1>: HW-Version <p2>: FPGA Version VHDL <p3>: FPGA Version MLAB <p4>: S-<card nr> (e.g.: 810 = S810) <p5>: Processor type (reserved)
➡	⚖️	U2DM1HW=1 , 1 , 2 , 830 , 0<CR>	<p1>: HW-Version = 1 <p2>: FPGA Version VHDL = 1 <p3>: FPGA Version MLAB = 2 <p4>: S<card nr> = S830 <p5>: Processor type = 0 (reserved)

2.3 Device Configuration

2.3.1 Set Baudrate („U2DM<d>COM<c>BD=<bd><CR>“)

Dir		Telegram	Description
➔		U2DM<d>COM<c>BD=<bd><CR>	<p>This command sets the baudrate of the specified COM port of the device:</p> <p><c> = <u>COM port</u> 0: COM port actually connected to the device 1: COM 1 of the device 2: COM 2 of the device</p> <p><bd>= <u>Baudrate</u> 24 (=2400 Baud) 48 (=4800 Baud) 96 (=9600 Baud) 192 (=19200 Baud) 384 (=38400 Baud) 576 (=57600 Baud) 1152 (=115200 Baud)</p> <p>Note: these settings can not be changed, they are fix: Parity: no Databits: 8 Stopbit(s): 1</p>
➔		U2DM1COM0BD=1152<CR>	<p1>: Baudrate = 1152 (the new device baudrate is set to 115200 Baud)
←		---	no device response
←		---	no device response

2.3.2 Set Remote Interface („U2DM<d>DRI=<s><CR>“)

Dir		Telegram	Description
➔		U2DM<d>DRI=<s><CR>	<p>This command enables or disables the remote interface of the device. For a remote control of the device, it is necessary to send this command first to enable the complete remote control functionality:</p> <p><s>: 1 = enable the remote interface 0 = disable the remote interface</p>
➔		U2DM1DRI?<CR>	Request the state of the remote interface
←		U2DM1DRI=0<CR>	returns the state of the remote interface: <s>: 0 = remote interface disabled
➔		U2DM1DRI=1<CR>	Enables the remote interface <s>: 1 = enables the interface
←		U2DM1DRI=1<CR>	Returns the state of the remote interface: <s>: 1 = remote interface is enabled

2.4 Encoder Commands

2.4.1 Encoder Request Value („U2DM<d>EN<e>?<CR>“)

Dir		Telegram	Description
➔	📄	U2DM<d>EN<e>?<CR>	This command requests the actual value of the selected encoder. <e>: selected encoder
➔	🖨️	U2DM1EN1?<CR>	Request the value of encoder 1 <e>: encoder nr = 1
⬅️	📄	U2DM<d>EN<e>=<scs> , <ev><CR>	returns the settings of the selected encoder: <scs>: system control selection <ev>: encoder setting according to the actual system control selection <mut>: Mute State <onl>: “ON”-LED State <pkl>: “PEAK”-LED State
⬅️	🖨️	U2DM1EN1=1 , 26<CR>	returns the settings of the encoder 1: <scs>: System Control Setting Level (=1) <ev>: encoder value = 26 <mut>: encoder/channel not muted <onl>: “ON”-LED is Off <pkl>: “Peak”-LED is Off

2.4.2 Encoder Rotate Left („U2DM<d>EN<e>RL<CR>“)

Dir		Telegram	Description
➔	📄	U2DM<d>EN<e>RL<CR>	This command executes a rotate left action at the selected encoder. <e>: selected encoder
➔	🖨️	U2DM1EN1RL<CR>	Executes a rotate left action at encoder 1
⬅️	📄	U2DM<d>EN<e>=<scs> , <ev> , <mut> , <onl> , <pkl><CR>	returns the settings of the selected encoder: (see. “Encoder Request Value” command)
⬅️	🖨️	U2DM1EN1=1 , 26 , 0 , 0 , 0<CR>	returns the settings of the encoder 1: <ev> = 26
➔	🖨️	U2DM1EN1RL<CR>	Executes a rotate left action at encoder 1
⬅️	🖨️	U2DM1EN1=1 , 27 , 0 , 0 , 0<CR>	returns the settings of the encoder 1: <ev> = 27
➔	🖨️	U2DM1EN1RL<CR>	Executes a rotate left action at encoder 1
⬅️	🖨️	U2DM1EN1=1 , 28 , 0 , 0 , 0<CR>	returns the settings of the encoder 1: <ev> = 28

2.4.3 Encoder Rotate Right („U2DM<d>EN<e>RR<CR>“)

Dir		Telegram	Description
➔	📄	U2DM<d>EN<e>RR<CR>	This command executes a rotate right action at the selected encoder. <e>: selected encoder
➔	🔒	U2DM1EN1RR<CR>	Executes a rotate right action at encoder 1
⬅	📄	U2DM<d>EN<e>=<scs>,<ev>,<mut>,<on1>,<pk1><CR>	returns the settings of the selected encoder: (see. “Encoder Request Value” command)
⬅	🔒	U2DM1EN1=1,27,0,0,0<CR>	returns the settings of the encoder 1: <ev> = 27
➔	🔒	U2DM1EN1RL<CR>	Executes a rotate left action at encoder 1
⬅	🔒	U2DM1EN1=1,26,0,0,0<CR>	returns the settings of the encoder 1: <ev> = 26
➔	🔒	U2DM1EN1RL<CR>	Executes a rotate left action at encoder 1
⬅	🔒	U2DM1EN1=1,25,0,0,0<CR>	returns the settings of the encoder 1: <ev> = 25

2.4.4 Encoder Set Value („U2DM<d>EN<e>=<ev><CR>“)

Dir		Telegram	Description
➔	📄	U2DM<d>EN<e>=<ev><CR>	This command sets directly a new value to the encoder <e>: selected encoder <ev>: new encoder value
➔	🔒	U2DM1EN1=32<CR>	sets encoder 1 to value of 32: <ev> = 32
⬅	📄	U2DM<d>EN<e>=<scs>,<ev>,<mut>,<on1>,<pk1><CR>	returns the settings of the selected encoder: (see. “Encoder Request Value” command)
⬅	🔒	U2DM1EN1=1,32,0,0,0<CR>	returns the settings of the encoder 1: <ev> = 32
➔	🔒	U2DM1EN13=3<CR>	sets encoder 13 (=system control) to value 3: <ev> = 3 (Bass)
⬅	🔒	U2DM1EN1=2,-4,0,0,0<CR> U2DM1EN2=2,0,0,0,0<CR> U2DM1EN3=2,0,0,0,0<CR> U2DM1EN4=2,0,0,0,0<CR> U2DM1EN5=2,0,0,0,0<CR> U2DM1EN6=2,0,0,0,0<CR> U2DM1EN7=2,2,0,0,0<CR> U2DM1EN8=2,0,0,0,0<CR>	returns the bass settings of all encoder/channels:

2.5 Status Commands

2.5.1 Request Channel Status („U2DM<d>CS<c>?<CR>“)

Dir		Telegram	Description
➔	📄	U2DM<d>CS<c>?<CR>	This command requests the status of the selected channel from the device <c>: selected channel
➔	📄	U2DM1CS1?<CR>	Requests the channel status of channel 1
➡	📄	U2DM<d>CS<c>=<cl>,<mut>,<onl>,<pk1><CR>	returns the status data of the selected channel <cl>: channel input level <mut>: Mute State <onl>: “ON”-LED State <pk1>: “PEAK”-LED State:
➡	📄	U2DM1CS1=6,0,1,0<CR>	Device returns the status of channel 1: <cl>: = 6 <mut>: = 0 <onl>: = 1 <pk1>: = 0

2.5.2 Set Channel Status Remote Update Time („U2DM<d>CS<c>=<t><CR>“)

Dir		Telegram	Description
➔	📄	U2DM<d>CS<c>=<t><CR>	With this command you can set the time period (= t*100 ms) the device automatically sends a single channel status telegram for each channel: <c>: channel (has no effect) <t>: time period t*100 ms 0 = disables the automatic update 1 ... 15
➔	📄	U2DM1CS1=5<CR>	Set the automatically update period for the channel status us set to 5*100 ms (= 500 ms)
➡	📄	U2DM<d>CS<c>=<cl>,<mut>,<onl>,<pk1><CR>	returns the status of the selected channel: (see. “Request Channel Status” command)
➡	📄	U2DM1CS1=13,0,0,0<CR> <Delay t*100ms> U2DM1CS2=18,0,1,0<CR> <Delay t*100ms> U2DM1CS3=35,0,1,0<CR> <Delay t*100ms> U2DM1CS4=43,0,1,1<CR> <Delay t*100ms> U2DM1CS5=30,0,1,0<CR> <Delay t*100ms> U2DM1CS6=33,0,1,0<CR> <Delay t*100ms> U2DM1CS7=0,0,0,0<CR> <Delay t*100ms> U2DM1CS8=13,0,1,0<CR> <Delay t*100ms> U2DM1CS9=13,0,1,0<CR> <Delay t*100ms> ...	The device sends automatically the actual status of all channels. After telegram the device inserts a delay of t*100ms, in that example the delay is 500 ms. After transmitting the last channel the device starts again with channel number 1. Note: channel number 13 represents the system control, which status is always zero: “U2DM1CS13=0,0,0,0<CR>”





2.5.3 Channel Status Set Mute („U2DM<d>EN<c>MU=<mut><CR>“)

Dir		Telegram	Description
➔	📄	U2DM<d>EN<e>MU=<mut><CR>	This command sets the mute state of the encoder <e>: selected encoder <mut>: Mute State
➔	📄	U2DM1EN1MU=1<CR>	Mutes encoder/channel 1
⬅	📄	U2DM<d>EN<e>=<scs>,<ev>,<mut>, <on1>,<pk1><CR>	returns the settings of the selected channel: (see. “Encoder Request Value” command)
⬅	📄	U2DM1EN1=1,32,1,1,0<CR>	Device returns the setting of encoder 1: <mut>: encoder is muted
➔	📄	U2DM1EN1MU=0<CR>	Clears the mute state of encoder/channel 1
⬅	📄	U2DM1EN1=1,32,0,1,0<CR>	Device returns the setting of encoder1: <mut>: encoder is not muted

2.5.4 Request Device Status („U2DM<d>DS?<CR>“)

Dir		Telegram	Description
📄➔		U2DM<d>DS?<CR>	This command requests the status of all channels from the device
📄➔		U2DM1DS?<CR>	Requests the device status
⬅	📄	U2DM<d>DS=<csx>,<csx>,..., <csx>,<csx><CR>	returns the status of all channels in the csx format (see. “Encoder Request Value” command)
⬅	📄	U2DM1DS=00010,04010,00010,00010,00010,00010,00000,02010,...,00000,00000<CR>	Device returns the status of all channels. The channel number depends on the device type.

2.5.5 Set Device Status Remote Update Time („U2DM<d>DST=<t><CR>“)

Dir	Telegram	Description
	U2DM<d>DST=<t><CR>	With this command you can set the time period (= t*100 ms) the device automatically sends a the device status telegram containing the status of all channels within one telegram: <t>: time period t*100 ms 0 = disables the automatic update 1 ... 15
	U2DM1DST=10<CR>	Set the automatically update period for the device status to 10*100 ms (= 1000 ms)
	U2DM<d>DST=<csx>,<csx>,...,<csx>,<csx><CR>	returns the device status of all channels: (see. "Request Device Status" command)
	U2DM1DST=00010,04010,00010,00010,00010,00010,00000,02010,00000,00000<CR> <Delay t*100ms> U2DM1DST=00010,04010,00010,00010,00010,00010,00000,02010,00000,00000<CR> <Delay t*100ms> ... U2DM1DST=00010,04010,00010,00010,00010,00010,00000,02010,00000,00000<CR>	The device periodically transmits the status of all channels.













2.6 Preset Commands

2.6.1 Preset Status („U2DM<d>PRE<p>?<CR>“)

Dir		Telegram	Description
➔	📄	U2DM<d>PRE<p>?<CR>	This command requests the status of a certain preset from the device <p>: preset number (0...12) 0: requests the active preset nr. 1...12: returns the status of the selected preset.
➔	🔒	U2DM1PS0?<CR>	Requests the number and status of the active preset
⬅	📄	U2DM<d>PRE<p>=<sel>,<cfg><CR>	returns the status data of the selected channel <p>: preset number (0...12) 0: Error invalid preset nr. <sel>: preset is selected (=active) 0: preset not active 1: preset is active <cfg>: config is assigned to Preset 0: no configuration assigned to preset 1: valid configuration assigned to preset
⬅	🔒	U2DM1PRE1=1,1<CR>	Device returns the status of preset 1: <sel>: = 1 (preset is selected and active) <cfg>: = 1 (configuration is assigned)
➔	🔒	U2DM1PRE3?<CR>	Requests the status of preset 3
⬅	🔒	U2DM1PRE3=0,1<CR>	Device returns the status of preset 3: <sel>: = 0 (preset not active) <cfg>: = 1 (configuration is assigned)
➔	🔒	U2DM1PRE7?<CR>	Requests the status of preset 7
⬅	🔒	U2DM1PRE7=0,0<CR>	Device returns the status of preset 7: <sel>: = 0 (preset not active) <cfg>: = 0 (no configuration assigned)







2.6.2 Preset Config („U2DM<d>PRE<p>=<sel>,<cfg>,<def><CR>“)

Dir		Telegram	Description
📄➔		U2DM<d>PRE<p>=<sel>{,<cfg>}{,<def>}<CR>	sets the status data of the selected channel <p>: preset number (1...12) <sel>: 1 = select preset 0 = (no function, selection unchanged) <cfg>: 1 = programs actual configuration to the selected preset, which is then active 0 = delete configuration of preset <def>: 1 = assign default configuration to preset and select preset 0 = (no function)
⬅	🔒	U2DM<d>PRE<p>=<sel><CR>	returns the status data of the selected channel <p>: preset number (1...12) <sel>: 1 = select preset 0 = (no function and selection unchanged)






Dir	Telegram	Description
	U2DM1PRE1=1<CR>	Sets the preset number 1 to device: <sel>: = 1 (preset is selected and active)
	U2DM1PRE1=1,1<CR>	Device returns the status of preset 1: <sel>: = 1 (preset is selected and active) <cfg>: = 1 (configuration is assigned)
	U2DM<d>PRE<p>=<sel>,<cfg><CR>	returns the status data of the selected channel <p>: preset number (1...12) <sel>: 1 = select preset 0 = (no function and selection unchanged) <cfg>: 1 = program actual configuration to preset and select preset 0 = delete configuration of preset
	U2DM1PRE3=1,1<CR>	program actual configuration to preset nr. 3: <sel>: = 1 (preset is selected and active) <cfg>: = 1 program actual configuration to preset and select preset
	U2DM1PRE3=1,1<CR>	Device returns the status of preset 3: <sel>: = 1 (preset is selected and active) <cfg>: = 1 (configuration is assigned)
	U2DM1PRE3=1,0<CR>	clear configuration of preset nr. 3: <sel>: = x (don't care) <cfg>: = 0 (clear configuration)
	U2DM1PRE3=0,0<CR>	Device returns the status of preset 3: <sel>: = 0 (preset is not active) <cfg>: = 0 (no configuration is assigned)
	U2DM<d>PRE<p>=<sel> ,<cfg> ,<def><CR>	returns the status data of the selected channel <p>: preset number (1...12) <sel>: 1 = select preset 0 = (no function and selection unchanged) <cfg>: 1 = program actual configuration to preset and select preset 0 = delete configuration of preset <def>: 1 = assign default configuration to preset and select preset 0 = (no function)
	U2DM1PRE3=1,1<CR>	Set preset number 1 to device: <sel>: = 1 (preset is selected and active)
	U2DM1PRE3=0,1<CR>	Device returns the status of channel 1: <sel>: = 0 (preset not active) <cfg>: = 1 (configuration is assigned)
	U2DM1PRE7?<CR>	Requests the status of preset 7
	U2DM1PRE7=0,0<CR>	Device returns the status of channel 1: <sel>: = 0 (preset not active) <cfg>: = 0 (no configuration assigned)

2.7 Routing Commands

2.7.1 Get Routing Status („U2DM<d>RTO<c>?<CR>“)










Dir	Telegram	Description
	U2DM<d>RTO<c>?<CR>	This command requests the routing status of a certain channel from the device <c>: channel number
	U2DM<d>RTO<c>=<out_rec_usb>,<dante><CR>	Returns the routing status of the selected channel <c>: source channel number <out_rec_usb>: routing deststatus for OUT 1/2, REC and USB 0: no routing ..._out_... bit 0: route to OUT 1 (0x01) bit 1: route to OUT 2 (0x02) ..._rec_...(for future use) bit 2: route to REC (0x04) ..._usb_...(for future use) bit 3: route to USB L (0x08) bit 4: route to USB R (0x10) Optional <dante>: routing status for DANTE 0: no routing bit 0: route to DANTE OUT 1 (0x01) bit 1: route to DANTE OUT 2 (0x02) bit 2: route to DANTE OUT 3 (0x04) bit 3: route to DANTE OUT 4 (0x08) bit 4: route to DANTE OUT 5 (0x10) bit 5: route to DANTE OUT 6 (0x20) bit 6: route to DANTE OUT 7 (0x40) bit 7: route to DANTE OUT 8 (0x80) nn: (bit combination of the channels)
	U2DM1RTO1?<CR>	Requests the routing status of channel nr. 1
	U2DM1RTO1=2,0<CR>	Device returns the new/actual routing status of channel 1: <out_rec_usb>: = 2 (routing to OUT 2) <dante> : = 0 (no routing to DANTE)
	U2DM1RTO4=3,0<CR>	Sets the routing of channel 4 to OUT 1 and OUT 2
	U2DM1RTO4=3,0<CR>	Device returns the new/actual routing status of channel 4: <out_rec_usb>: = 3 (routing to OUT 1 and OUT 2) <dante> : = 0 (no routing to DANTE)

2.7.2 Set Routing of Channel („U2DM<d>RTO<c>=<out_rec_usb>, <dante><CR>“)

Dir	Telegram	Description
	U2DM<d>RTO<c>=<out_rec_usb>, <dante><CR>	Returns the routing status of the selected channel <c>: source channel number <out_rec_usb>: routing deststatus for OUT 1/2, REC and USB 0: no routing ..._out_... bit 0: route to OUT 1 (0x01) bit 1: route to OUT 2 (0x02) ..._rec_...(for future use) bit 2: route to REC (0x04) ..._usb_...(for future use) bit 3: route to USB L (0x08) bit 4: route to USB R (0x10) Optional <dante>: routing status for DANTE 0: no routing bit 0: route to DANTE OUT 1 (0x01) bit 1: route to DANTE OUT 2 (0x02) bit 2: route to DANTE OUT 3 (0x04) bit 3: route to DANTE OUT 4 (0x08) bit 4: route to DANTE OUT 5 (0x10) bit 5: route to DANTE OUT 6 (0x20) bit 6: route to DANTE OUT 7 (0x40) bit 7: route to DANTE OUT 8 (0x80) nn: (bit combination of the channels)
	U2DM1RTO1=2,0<CR>	Sets the routing of channel 1 to OUT 2
	U2DM1RTO1=2,0<CR>	Device returns the new/actual routing status of channel 1: <out_rec_usb>: = 2 (routing to OUT 2) <dante> : = 0 (no routing to DANTE)
	U2DM1RTO4=3,0<CR>	Sets the routing of channel 4 to OUT 1 and OUT 2
	U2DM1RTO4=3,0<CR>	Device returns the new/actual routing status of channel 4: <out_rec_usb>: = 3 (routing to OUT 1 and OUT 2) <dante> : = 0 (no routing to DANTE)







2.8 Network Parameter Commands

2.8.1 Network Parameter Get („U2DM<d>NWP<para>?<CR>“)

Dir	Telegram	Description
	U2DM<d>NWP<para>?<CR>	This command requests the status of a certain network parameter from the device <para>: Network Parameter index (1...4) 1: IP Address 2: MAC Address 3: Subnet Mask Address 4: Gateway Address
	U2DM1NWP1?<CR>	Requests the IP Address
	U2DM<d>NWP<para>=<Oct1>.<Oct2>.<Oct3>.<Oct4><CR>	Returns the Network Parameter of the selected Parameter <para>: Network Parameter index 1: IP Address 3: Subnet Mask Address 4: Gateway Address
	U2DM<d>NWP2=<Oct1>:<Oct2>:<Oct3>:<Oct4>:<Oct5>:<Oct6><CR>	Returns the Network Parameter of the selected Parameter <para>: Network Parameter index 2: MAC Address
	U2DM1NWP1=192.168.100.1<CR>	Device returns the IP Address
	U2DM1NWP2?<CR>	Requests the the MAC Address
	U2DM1NWP2=00:1e:c0:dd:0b:01<CR>	Device returns the MAC Address
	U2DM1NWP3?<CR>	Requests the Subnet Mask Address
	U2DM1NWP1=255.255.255.0<CR>	Device returns the Subnet Mask Address

2.8.2 Network Parameter Get („U2DM<d>NWP<para>=<Oct1>,<Oct2>, <Oct3>, <Oct4>“)

Note: It is recommended to restart the device after network parameter were modified!

Dir	Telegram	Description
	U2DM<d>NWP<para>=<Oct1>, <Oct2>, <Oct3>, <Oct4><CR>	Sets the Network Parameter of the selected parameter <para>: Network Parameter 1: IP Address 3: Subnet Mask Address 4: Gateway Address
	U2DM<d>NWP<para>=<Oct1>.<Oct2>.<Oct3>.<Oct4><CR>	Returns the Network Parameter of the selected Parameter <para>: Network Parameter index 1: IP Address 3: Subnet Mask Address 4: Gateway Address
	U2DM1NWP1=192,168,100,12<CR>	Sets the IP Address to the value “192.168.100.12”
	U2DM1NWP1=192.168.100.12<CR>	Device returns the new IP Address
	U2DM1NWP3=255.255.255.128<CR>	Sets the Subnet Mask Address to the value “255.255.255.128”
	U2DM1NWP3=255.255.255.128<CR>	Device returns the Subnet new Mask Address