

Nice

DMBM

CE

Integration Protocol

Nice

SERIAL / ETHERNET

The standard interfaces (Serial / Ethernet) allow Nice products to be operated and controlled on the “BusT4” channel.

Serial configuration

The device can be connected to a PC running a terminal emulator. The RS232 port must be configured as follows:

- Baudrate: 115200
- Parity: None
- Data bits: 8
- Stop bits: 1
- Flow Control: None

ATTENTION: the strings must all end with the string terminator \$0d (CR).

IP connection

The serial interface is emulated via TCP/IP. The default settings are:

- Ip-address: 192.168.0.1
- Subnet-mask: 255.255.255.0
- Default gateway: 192.168.0.254
- Tcp-port: 23
- The IP configuration can be modified via a serial connection RS232, NICE CONFIGURATOR TOOL

COMMUNICATIONS PROTOCOL

The serial interface and tcp/ip connection allow you to control the device with the following commands.

The protocol syntax is as follows:

<COMMAND> <Flag> <Parameters>

<COMMAND>	Identifies the command to be executed or associated information	
<Flag>	Identifies the action associated with the command. The protocol provides for the following values:	
	'<'	command sent by the PC; indicates a request for information (GET) from the automation or interface, as specified in the <Parameters> field
	'>'	command sent by the PC; indicates a request to change a setting/ state (SET) on the automation, as specified in the <Parameters> field
	'#'	command sent by the interface; indicates a response (RSP) to the simple reception of a command, not to its execution
	'**'	command sent by the interface; indicates an event (EVT) on one of the automations connected to the “BusT4” or the interface itself (for example, for commands/ queries relating to its 4 inputs and outputs). The event may have been tripped by a request from the PC itself or any other system interacting directly with the automation, such as a photocell, remote control, programmable timer, etc, of which the interface may know/not know the details
	'!'	command sent by the interface; indicates communications or syntax error (ERR) in the command sent by the PC to the interface or by the interface to the final device for which the command in question is intended
<Parametri>	Composed of one or more strings whose syntax depends on the command in question	

For more specification about this syntax, see the TTPCI user manual.

1. “BusT4” commands: POS

Associating the POS command with the FLAG allows you to query or change the percentage position of an automation connected to the BusT4. The protocol is as follows (caution! parameter [yy] is not required in the commands; it is sent by the interface solely when returning an error message, such as ERR):

“pos [flag] [gg] [ee] [pppp] [qqqq] [x] [yy]”

<Flag>	In combination with POS, defines the type of message sent:	
	'>'	SET command, sends a request to vary the percentage position of an automation
	'<'	GET command, queries the percentage position of an automation
	'#'	response of the interface (RSP) confirming reception of the GET / SET command
	'**'	response of the interface (EVT), sending the requested information
	'!'	response of the interface (ERR) indicating a syntax error in the GET / SET command
<gg>	Address of the automation	
<ee>	Endpoint of the automation	

The meaning of the following parameters depends on the [flag].

A. FOR THE SET / GET COMMANDS AND FOR RSP / EVT RESPONSES:

<pppp>	indicates the target percentage position of the motor (always present), in thousandths. The value runs from 0000 to 1000. The value FFFF indicates that the position of the motor is not to be changed. The value 0000 indicates fully closed, 1000 fully open	
<qqqq>	always FFFF	
<x>	a parameter used to indicate a series of additional behaviours to the automation. The following configuration parameters are available:	
	<0>	step-by-step mode operation of the automation. Not implemented
	<1>	locks the automation once the manoeuvre specified in parameters <pppp> and <qqqq> has terminated
	<2>	releases the automation before the manoeuvre specified in parameters <pppp> and <qqqq> starts
	<3>	unlock before manoeuvre and lock after manoeuvre (i.e. the manoeuvre defined by parameters <pppp> and <qqqq>)
	<4> - <e>	reserved for future implementations
	<f>	no additional information

B. For ERR responses:

<pppp>	not applicable in case of ERR message	
<qqqq>	not applicable in case of ERR message	
<x>	not applicable in case of ERR message	
<yy>	tells the operator which error has been detected, according to the following table:	
	<00>	no error
	<01>	command syntax error
	<02>	command timeout
	<03>	collision on "BusT4"
	<04>	device not found
	<05> - <0f>	not used; available for future developments
	<10>	command execution error
	<11>	protocol data error
	<12>	encoder data error
	<13>	position data error
	<14>	"Bluebus" error. Not implemented
	<15>	memory error
	<16> - <ef>	not used; available for future developments
	<f0>	"BusT4" not operating
	<f1>	not used; available for future developments

C. For EVT responses:

<x>	reports the status of the automation. May take the following values:	
	<1>	automation locked
	<2>	automation unlocked
	<3>	automation moving
	<f>	no additional information

Example – request to single motor automation, address 3 and endpoint 5, to move to the fully open position:

"POS > 03 05 1000 FFFF F"

"POS # 03 05 1000 FFFF F"

"POS * 03 05 1000 FFFF F"

2. "BusT4" commands: CMD

Associating the CMD command with the FLAG allows you to send a command to an automation connected to the BusT4. The protocol is as follows (caution! parameter [yy] is not required in the commands; it is sent by the interface solely when returning an error message, such as ERR):

"cmd [flag] [gg] [ee] [xx] [yy]"

<Flag>	In combination with CMD, defines the type of message sent:	
	'>'	SET command, sends a request to vary the percentage position of an automation
	'<'	GET command, queries the percentage position of an automation
	'#'	response of the interface (RSP) confirming reception of the GET / SET command
	'*'	response of the interface (EVT), sending the requested information
	'!'	response of the interface (ERR) indicating a syntax error in the GET / SET command
[gg]	address of the automation	
<ee>	endpoint of the automation	

The meaning of the following parameters depends on the [flag].

A. For the SET / GET commands and for RSP / EVT responses:

[xx]	a parameter used to indicate the command that has to be executed. The following parameters are available in the following table:	
	<00> - <01>	not used; available for future developments
	<02>	stop command
	<03>	up command
	<04>	down command
	<05>	intermediate position 1
	<06>	intermediate position 2
	<07>	intermediate position 3
	<08> - <0A>	not used; available for future developments
	<0B>	intermediate position 4
	<0C>	intermediate position 5
	<0D>	intermediate position 6
	<0A> - <FF>	not used; available for future developments

B. For ERR responses:

<xx>	not applicable in case of ERR message	
<yy>	tells the operator which error has been detected, according to the following table:	
	<00>	no error
	<01>	command syntax error
	<02>	command timeout
	<03>	collision on "BusT4"
	<04>	device not found
	<05> - <0f>	not used; available for future developments
	<10>	command execution error
	<11>	protocol data error
	<12>	encoder data error
	<13>	position data error
	<14>	"Bluebus" error Not implemented
	<15>	memory error
	<16> - <ef>	not used; available for future developments
	<f0>	"BusT4" not operating
	<f1>	not used; available for future developments

C. For EVT responses:

<x>	reports the status of the automation. May take the following values:	
	<1>	automation locked
	<2>	automation unlocked
	<3>	automation moving
	<4>	automation opening
	<5>	automation closing
	<6>	no limits set
	<7>	automation malfunction/error
	<f>	no additional information

4. “BusT4” commands: INP

The INP command allows you to query the interface for the status of a given DMBM device input. The protocol is as follows (parameter [yy] is not obligatory; it is sent solely by ERR responses):

“inp [flag] [g] [v] [yy]”

[flag]	in combination with INP, defines the type of message sent:	
	<	GET command, requests the status of a given input
	#	response of the interface (RSP) indicating that the GET command has been received
	!	response of the interface (ERR) indicating a GET command syntax error
	*	response of the interface (EVT), sending the requested information
[g]	Address of the input (0 - F). To enable the inputs use the Nice Screen Configuration Tool	
[v]	The meaning of the parameter depends on the [flag]	
	<i>For GET commands:</i>	
	<0>	fixed parameter
	<i>For an ERR response, the parameter has no meaning</i>	
	<i>For an EVT response, the parameter indicates the state of the input in question</i>	
	<0>	input not active
	<1>	input active
[yy]	<i>Tells the operator which error has been detected, according to the following table:</i>	
	<00>	no error
	<01>	command syntax error
	<02>	command timeout
	<03>	collision on “BusT4”
	<04>	device not found
	<05> - <0f>	not used; available for future developments
	<10>	command execution error
	<11>	protocol data error
	<12>	encoder data error
	<13>	position data error
	<14>	“Bluebus” error. Not implemented
	<15>	memory error
	<16> - <ef>	not used; available for future developments
	<f0>	“BusT4” not operating
	<f1>	not used; available for future developments

Example – a request to the automation as to whether input 1 is active, and the interface’s confirmation:

“INP < 0 0”

“INP # 0 0”

“INP * 0 1”

5. “BusT4” commands: INF

“inf [flag] [gg] [ee] [ww] [xx] [zz][yy]”

INF commands return information about the devices on the “BusT4” line. The protocol is as follows:

[flag]	in combination with INF, defines the type of message sent:	
	<	GET command, used to request the list of devices on the “BusT4” line
	#	response of the interface (RSP) confirming reception of the GET / SET command
	!	response of the interface (ERR) indicating a syntax error in the GET / SET command
	*	response of the interface (EVT), sending the requested information
[gg]	address of the automation	
	<i>Indicates the filter to be applied to the list of devices in the GET:</i>	
	00 ≤ gg ≤ fe	filter set to <gg>; only devices whose address is <gg> are listed
	<ff>	no filter; devices with any address will be listed. For the RSP, ERR and EVT responses, it assumes the value passed by the previous SET / GET
[ee]	Endpoint of automation	
	<i>Indicates the filter on the endpoint to apply to the list of devices in the GET:</i>	
	00 ≤ ee ≤ fe	filter set to <ee>; only devices whose endpoint is <ee> are listed
	<ff>	no filter; devices with any endpoint will be listed. For the RSP and EVT responses, it assumes the value passed by the previous SET / GET
[ww]	The meaning of the parameter depends on the [flag]. Its default value is <ff>	
	<i>Indicates the profile of the automation for the EVT response or the GET filter:</i>	
	<00>	standard
	<01>	display. Not implemented
	<02>	PC. Not implemented
	<03>	not used; available for future expansions
	<04>	motor control unit. Not implemented
	<05>	not used; available for future expansions
	<06>	screen control unit
	<07> - <09>	not used; available for future expansions
	<0a>	OxI. Not implemented
	<0b>	not used; available for future expansions
	For the RSP and ERR responses, it assumes the value passed by the previous GET	
[xx]	The meaning of the parameter depends on the [flag]. Its default value is <ff>	
	<i>Indicates the type of automation for the RSP response or the GET filter:</i>	
	<00>	not used; available for future expansions
	<01>	sliding. Not implemented
	<02>	sectional. Not implemented
	<03>	swing. Not implemented
	<04>	barrier. Not implemented
	<05>	up-and-over. Not implemented
	<06> - <69>	not used; available for future expansions
	<70>	shutters
	<71>	rolling blinds
	<72>	sun shades
	<73>	projection screens
	<74>	dampers
	<75>	mosquito nets
	<76>	venetian blinds
	<77>	orientable shutters
	<78>	patio/pergola
	<79>	curtains for interiors
	<7A>	roman blinds
	<7B> - <fe>	not used; available for future expansions
	<ff>	no filter active.
	For the RSP and ERR responses, it assumes the value passed by the previous GET	

[zz]	The meaning of the parameter depends on the [flag]. Its default value is <ff>	
	<i>Indicates the status of the device for the RSP response or the GET filter:</i>	
	<00>	automation closed
	<01>	automation open
	<02>	automation moving
	<03>	automation malfunction/error
	<04>	no limits set
	<05>	automation opening
	<06>	automation closing
	<07> - <fe>	reserved for future implementations
	<ff>	no filter active
	For the RSP and ERR responses, it takes the value passed by the previous GET	
[yy]	Reports the error, according to the following table:	
	<00>	no error
	<01>	command syntax error
	<02>	command timeout
	<03>	collision on "BusT4"
	<04>	device not found
	<05> - <0f>	not used; available for future developments
	<10>	command execution error
	<11>	protocol data error
	<12>	encoder data error
	<13>	position data error
	<14>	"Bluebus" error. Not implemented
	<15>	memory error
	<16> - <ef>	not used; available for future developments
	<f0>	"BusT4" not operating
	<f1> - <ff>	not used; available for future developments

6. "BusT4" commands: LST

LST commands relate to the handling/acquiring of information in the list of automations on the "BusT4" line.

The protocol is as follows:

"lst [flag] [gg] [ee] [ww] [xx] [zz] [xx] [yy]"

[flag]	In combination with LST, defines the type of message sent:	
	<	GET command, used to request the list of devices on the "BusT4" line
	#	response of the interface (RSP) confirming reception of the GET / SET command
	!	response of the interface (ERR) indicating a syntax error in the GET / SET command
	*	response of the interface (EVT), sending the requested information
[gg]	Address of the automation	
	<i>Indicates the filter to be applied to the list of devices in the GET:</i>	
	00 ≤ gg ≤ fe	filter set to <gg>; only devices whose address is <gg> are listed
	<ff>	no filter; devices with any address will be listed. For the RSP, ERR and EVT responses, it assumes the value passed by the previous SET / GET
[ee]	Endpoint of automation	
	<i>Indicates the filter on the endpoint to apply to the list of devices in the GET:</i>	
	00 ≤ ee ≤ fe	filter set to <ee>; only devices whose endpoint is <ee> are listed
	<ff>	no filter; devices with any endpoint will be listed. For the RSP and EVT responses, it assumes the value passed by the previous SET / GET

[ww]	The meaning of the parameter depends on the [flag]. Its default value is <ff>																																								
	<p data-bbox="518 107 1313 136"><i>Indicates the profile of the automation for the EVT response or the GET filter:</i></p> <table border="1" data-bbox="518 136 1533 521"> <tr><td data-bbox="518 136 671 174"><00></td><td data-bbox="671 136 1533 174">standard</td></tr> <tr><td data-bbox="518 174 671 212"><01></td><td data-bbox="671 174 1533 212">display. Not implemented</td></tr> <tr><td data-bbox="518 212 671 250"><02>.</td><td data-bbox="671 212 1533 250">PC. Not implemented</td></tr> <tr><td data-bbox="518 250 671 288"><03></td><td data-bbox="671 250 1533 288">not used; available for future expansions</td></tr> <tr><td data-bbox="518 288 671 327"><04></td><td data-bbox="671 288 1533 327">motor control unit. Not implemented</td></tr> <tr><td data-bbox="518 327 671 365"><05></td><td data-bbox="671 327 1533 365">not used; available for future expansions</td></tr> <tr><td data-bbox="518 365 671 403"><06></td><td data-bbox="671 365 1533 403">screen control unit</td></tr> <tr><td data-bbox="518 403 671 441"><07> - <09>.</td><td data-bbox="671 403 1533 441">not used; available for future expansions</td></tr> <tr><td data-bbox="518 441 671 479"><0a></td><td data-bbox="671 441 1533 479">OXI. Not implemented</td></tr> <tr><td data-bbox="518 479 671 517"><0b></td><td data-bbox="671 479 1533 517">not used; available for future expansions</td></tr> </table> <p data-bbox="518 521 1390 551"><i>For the RSP and ERR responses, it assumes the value passed by the previous GET</i></p>	<00>	standard	<01>	display. Not implemented	<02>.	PC. Not implemented	<03>	not used; available for future expansions	<04>	motor control unit. Not implemented	<05>	not used; available for future expansions	<06>	screen control unit	<07> - <09>.	not used; available for future expansions	<0a>	OXI. Not implemented	<0b>	not used; available for future expansions																				
<00>	standard																																								
<01>	display. Not implemented																																								
<02>.	PC. Not implemented																																								
<03>	not used; available for future expansions																																								
<04>	motor control unit. Not implemented																																								
<05>	not used; available for future expansions																																								
<06>	screen control unit																																								
<07> - <09>.	not used; available for future expansions																																								
<0a>	OXI. Not implemented																																								
<0b>	not used; available for future expansions																																								
[xx]	The meaning of the parameter depends on the [flag]. Its default value is <ff>																																								
	<p data-bbox="518 604 1262 633"><i>Indicates the type of automation for the RSP response or the GET filter:</i></p> <table border="1" data-bbox="518 633 1533 1406"> <tr><td data-bbox="518 633 671 672"><00></td><td data-bbox="671 633 1533 672">not used; available for future expansions</td></tr> <tr><td data-bbox="518 672 671 710"><01></td><td data-bbox="671 672 1533 710">sliding. Not implemented</td></tr> <tr><td data-bbox="518 710 671 748"><02></td><td data-bbox="671 710 1533 748">sectional. Not implemented</td></tr> <tr><td data-bbox="518 748 671 786"><03>.</td><td data-bbox="671 748 1533 786">swing. Not implemented</td></tr> <tr><td data-bbox="518 786 671 824"><04></td><td data-bbox="671 786 1533 824">barrier. Not implemented</td></tr> <tr><td data-bbox="518 824 671 862"><05></td><td data-bbox="671 824 1533 862">up-and-over. Not implemented</td></tr> <tr><td data-bbox="518 862 671 900"><06> - <69></td><td data-bbox="671 862 1533 900">not used; available for future expansions</td></tr> <tr><td data-bbox="518 900 671 938"><70></td><td data-bbox="671 900 1533 938">shutters</td></tr> <tr><td data-bbox="518 938 671 976"><71></td><td data-bbox="671 938 1533 976">rolling blinds</td></tr> <tr><td data-bbox="518 976 671 1014"><72></td><td data-bbox="671 976 1533 1014">sun shades</td></tr> <tr><td data-bbox="518 1014 671 1052"><73></td><td data-bbox="671 1014 1533 1052">projection screens</td></tr> <tr><td data-bbox="518 1052 671 1090"><74></td><td data-bbox="671 1052 1533 1090">dampers</td></tr> <tr><td data-bbox="518 1090 671 1128"><75></td><td data-bbox="671 1090 1533 1128">mosquito nets</td></tr> <tr><td data-bbox="518 1128 671 1167"><76></td><td data-bbox="671 1128 1533 1167">venetian blinds</td></tr> <tr><td data-bbox="518 1167 671 1205"><77></td><td data-bbox="671 1167 1533 1205">orientable shutters</td></tr> <tr><td data-bbox="518 1205 671 1243"><78></td><td data-bbox="671 1205 1533 1243">patio/pergola</td></tr> <tr><td data-bbox="518 1243 671 1281"><79></td><td data-bbox="671 1243 1533 1281">curtains for interiors</td></tr> <tr><td data-bbox="518 1281 671 1319"><7A></td><td data-bbox="671 1281 1533 1319">roman blinds</td></tr> <tr><td data-bbox="518 1319 671 1357"><7B> - <fe></td><td data-bbox="671 1319 1533 1357">not used; available for future expansions</td></tr> <tr><td data-bbox="518 1357 671 1395"><ff></td><td data-bbox="671 1357 1533 1395">no filter active</td></tr> </table> <p data-bbox="518 1406 1390 1435"><i>For the RSP and ERR responses, it assumes the value passed by the previous GET</i></p>	<00>	not used; available for future expansions	<01>	sliding. Not implemented	<02>	sectional. Not implemented	<03>.	swing. Not implemented	<04>	barrier. Not implemented	<05>	up-and-over. Not implemented	<06> - <69>	not used; available for future expansions	<70>	shutters	<71>	rolling blinds	<72>	sun shades	<73>	projection screens	<74>	dampers	<75>	mosquito nets	<76>	venetian blinds	<77>	orientable shutters	<78>	patio/pergola	<79>	curtains for interiors	<7A>	roman blinds	<7B> - <fe>	not used; available for future expansions	<ff>	no filter active
<00>	not used; available for future expansions																																								
<01>	sliding. Not implemented																																								
<02>	sectional. Not implemented																																								
<03>.	swing. Not implemented																																								
<04>	barrier. Not implemented																																								
<05>	up-and-over. Not implemented																																								
<06> - <69>	not used; available for future expansions																																								
<70>	shutters																																								
<71>	rolling blinds																																								
<72>	sun shades																																								
<73>	projection screens																																								
<74>	dampers																																								
<75>	mosquito nets																																								
<76>	venetian blinds																																								
<77>	orientable shutters																																								
<78>	patio/pergola																																								
<79>	curtains for interiors																																								
<7A>	roman blinds																																								
<7B> - <fe>	not used; available for future expansions																																								
<ff>	no filter active																																								
[zz]	The meaning of the parameter depends on the [flag]. Its default value is <ff>																																								
	<p data-bbox="518 1489 1270 1518"><i>Indicates the status of the device for the RSP response or the GET filter:</i></p> <table border="1" data-bbox="518 1518 1533 1861"> <tr><td data-bbox="518 1518 671 1556"><00></td><td data-bbox="671 1518 1533 1556">automation closed</td></tr> <tr><td data-bbox="518 1556 671 1594"><01></td><td data-bbox="671 1556 1533 1594">automation open</td></tr> <tr><td data-bbox="518 1594 671 1632"><02></td><td data-bbox="671 1594 1533 1632">automation moving</td></tr> <tr><td data-bbox="518 1632 671 1671"><03></td><td data-bbox="671 1632 1533 1671">automation malfunction/error</td></tr> <tr><td data-bbox="518 1671 671 1709"><04></td><td data-bbox="671 1671 1533 1709">no limits set</td></tr> <tr><td data-bbox="518 1709 671 1747"><05></td><td data-bbox="671 1709 1533 1747">automation opening</td></tr> <tr><td data-bbox="518 1747 671 1785"><06></td><td data-bbox="671 1747 1533 1785">automation closing</td></tr> <tr><td data-bbox="518 1785 671 1823"><07> - <fe></td><td data-bbox="671 1785 1533 1823">reserved for future implementations</td></tr> <tr><td data-bbox="518 1823 671 1861"><ff></td><td data-bbox="671 1823 1533 1861">no filter active</td></tr> </table> <p data-bbox="518 1868 1353 1897"><i>For the RSP and ERR responses, it takes the value passed by the previous GET</i></p>	<00>	automation closed	<01>	automation open	<02>	automation moving	<03>	automation malfunction/error	<04>	no limits set	<05>	automation opening	<06>	automation closing	<07> - <fe>	reserved for future implementations	<ff>	no filter active																						
<00>	automation closed																																								
<01>	automation open																																								
<02>	automation moving																																								
<03>	automation malfunction/error																																								
<04>	no limits set																																								
<05>	automation opening																																								
<06>	automation closing																																								
<07> - <fe>	reserved for future implementations																																								
<ff>	no filter active																																								
[xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx]	The meaning of the parameter depends on the [flag]. Its default value is <ff....fff>.																																								
	Indicates the description of the scenario for the RSP response or the GET filter. 32chars																																								
[yy]	Reports the error, according to the following table:																																								
	<table border="1" data-bbox="518 1991 1533 2157"> <tr><td data-bbox="518 1991 671 2029"><00></td><td data-bbox="671 1991 1533 2029">no error</td></tr> <tr><td data-bbox="518 2029 671 2067"><01></td><td data-bbox="671 2029 1533 2067">command syntax error</td></tr> <tr><td data-bbox="518 2067 671 2105"><02></td><td data-bbox="671 2067 1533 2105">command timeout</td></tr> <tr><td data-bbox="518 2105 671 2143"><03></td><td data-bbox="671 2105 1533 2143">collision on "BusT4"</td></tr> </table>	<00>	no error	<01>	command syntax error	<02>	command timeout	<03>	collision on "BusT4"																																
<00>	no error																																								
<01>	command syntax error																																								
<02>	command timeout																																								
<03>	collision on "BusT4"																																								

<04>	device not found
<05> - <0f>	not used; available for future developments
<10>	command execution error
<11>	protocol data error
<12>	encoder data error
<13>	position data error
<14>	“Bluebus” error. Not implemented
<15>	memory error
<16> - <ef>	not used; available for future developments
<f0>	“BusT4” not operating
<f1> - <ff>	not used; available for future developments

7. “BusT4” commands: SCN

Associating the SCN command with the FLAG allows you to activate a scenario of automations connected to the BusT4. To create the groups it is necessary to use the Nice Screen Configuration Tool. The protocol is as follows (caution! parameter [yy] is not required in the commands; it is sent by the interface solely when returning an error message, such as ERR):

“scn [flag] [gg] [v] [yy]”

[flag]	In combination with SCN, defines the type of message sent:	
	'>	SET command, request a scenario
	#	response of the interface (RSP) indicating that the GET command has been received
	!	response of the interface (ERR) indicating a GET command syntax error
	*	response of the interface (EVT), sending the requested information
[gg]	Scenario ID (0 - FF). To enable the Scenes use the Nice Screen Configuration Tool	
[v]	The meaning of the parameter depends on the [flag]	
	<i>For GET commands:</i>	
	<0>	fixed parameter
	<i>For an ERR response, the parameter has no meaning:</i>	
	<1>	input active
[yy]	<i>Tells the operator which error has been detected, according to the following table:</i>	
	<00>	no error
	<01>	command syntax error
	<02>	command timeout
	<03>	collision on “BusT4”
	<04>	device not found
	<05> - <0f>	not used; available for future developments
	<10>	command execution error
	<11>	protocol data error
	<12>	encoder data error
	<13>	position data error
	<14>	“Bluebus” error. Not implemented
	<15>	memory error
	<16> - <ef>	not used; available for future developments
	<f0>	“BusT4” not operating
	<f1>	not used; available for future developments

8. “BusT4” commands: GRC

Associating the GRC command with the FLAG allows you to send a command to a group of automation connected to the BusT4. To create the groups it is necessary to use the Nice Screen Configuration Tool. The protocol is as follows (caution! parameter [yy] is not required in the commands; it is sent by the interface solely when returning an error message, such as ERR):

“grc [flag] [gg] [xx] [yy]”

[flag]	In combination with GRC, defines the type of message sent:	
	<'>	SET command, request a group command
	#	response of the interface (RSP) indicating that the GET command has been received
	!	response of the interface (ERR) indicating a GET command syntax error
[gg]	Group ID (0 - FF). To enable the Scenes use the Nice Screen Configuration Tool	

The meaning of the following parameters depends on the [flag].

A. For the SET / GET commands and for RSP / EVT responses:		
[xx]	a parameter used to indicate the command that has to be executed. The following parameters are available in the following table:	
	<00> - <01>	not used; available for future developments
	<02>	stop command
	<03>	up command
	<04>	down command
	<05>	intermediate position 1
	<06>	intermediate position 2
	<07>	intermediate position 3
	<08> - <0A>	not used; available for future developments
	<0B>	intermediate position 4
	<0C>	intermediate position 5
	<0D>	intermediate position 6
	<0E> - <2A>	not used; available for future developments
	<2B>	go to % position
	<2C> - <FF>	not used; available for future developments
[ff]	interpreted only if only if <xx> is 0x2b, Position in from 0 to 100 where 0 is the low limit and 100 is the high limit	

B. For ERR responses:		
<xx>	not applicable in case of ERR message	
<yy>	Tells the operator which error has been detected, according to the following table:	
	<00>	no error
	<01>	command syntax error
	<02>	command timeout
	<03>	collision on “BusT4”
	<04>	device not found
	<05> - <0f>	not used; available for future developments
	<10>	command execution error
	<11>	protocol data error
	<12>	encoder data error
	<13>	position data error
	<14>	“Bluebus” error. Not implemented
	<15>	memory error
	<16> - <ef>	not used; available for future developments
	<f0>	“BusT4” not operating
	<f1>	not used; available for future developments

9. "BusT4" commands: LSC

LSC commands relate to the handling/acquiring of information in the list of Scenes on the "BusT4" line.
The protocol is as follows:

"lsc [flag] [gg] [ww] [xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx] [yy]"

[flag]	In combination with LSC, defines the type of message sent:	
	<	GET command, used to request the list of devices on the "BusT4" line
	#	response of the interface (RSP) confirming reception of the GET / SET command
	!	response of the interface (ERR) indicating a syntax error in the GET / SET command
	*	response of the interface (EVT), sending the requested information
[gg]	Scenario ID (0 - FF). To enable the Scenes use the Nice Screen Configuration Tool	
	<i>Indicates the filter to be applied to the list of devices in the GET:</i>	
	00 ≤ gg ≤ fe	filter set to <gg>; only devices whose address is <gg> are listed
	<ff>	no filter; devices with any address will be listed. For the RSP, ERR and EVT responses, it assumes the value passed by the previous SET / GET
[ww]	The meaning of the parameter depends on the [flag]. Its default value is <ff>	
	<i>Indicates the profile of the automation for the EVT response or the GET filter:</i>	
	<00>	standard
	<01>	display. Not implemented
	<02>	PC. Not implemented
	<03>	not used; available for future expansions
	<04>	motor control unit. Not implemented
	<05>	not used; available for future expansions
	<06>	screen control unit
	<07> - <09>	not used; available for future expansions
	<0a>	OXI. Not implemented
	<0b>	not used; available for future expansions
	<i>For the RSP and ERR responses, it assumes the value passed by the previous GET</i>	
[xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx]	The meaning of the parameter depends on the [flag]. Its default value is <ff...fff>. Indicates the description of the scenario for the RSP response or the GET filter. 32chars	
[yy]	Reports the error, according to the following table:	
	<00>	no error
	<01>	command syntax error
	<02>	command timeout.
	<03>	collision on "BusT4"
	<04>	device not found
	<05> - <0f>	not used; available for future developments
	<10>	command execution error
	<11>	protocol data error
	<12>	encoder data error
	<13>	position data error
	<14>	"Bluebus" error. Not implemented
	<15>	memory error
	<16> - <ef>	not used; available for future developments
	<f0>	"BusT4" not operating
	<f1> - <ff>	not used; available for future developments

10. "BusT4" commands: LGR

LGR commands relate to the handling/acquiring of information in the list of Groups on the "BusT4" line.

The protocol is as follows:

"lgr [flag] [gg] [ww] [xx] [yy]"

[flag]	In combination with LGR, defines the type of message sent:	
	<	GET command, used to request the list of devices on the "BusT4" line
	#	response of the interface (RSP) confirming reception of the GET / SET command
	!	response of the interface (ERR) indicating a syntax error in the GET / SET command
	*	response of the interface (EVT), sending the requested information
[gg]	Group ID (0 - FF). To enable the Groups use the Nice Screen Configuration Tool	
	<i>Indicates the filter to be applied to the list of devices in the GET:</i>	
	00 ≤ gg ≤ fe	filter set to <gg>; only devices whose address is <gg> are listed
	<ff>	no filter; devices with any address will be listed. For the RSP, ERR and EVT responses, it assumes the value passed by the previous SET / GET
[ww]	The meaning of the parameter depends on the [flag]. Its default value is <ff>	
	<i>Indicates the profile of the automation for the EVT response or the GET filter:</i>	
	<00>	standard
	<01>	display. Not implemented
	<02>	PC. Not implemented
	<03>	not used; available for future expansions
	<04>	motor control unit. Not implemented
	<05>	not used; available for future expansions
	<06>	screen control unit
	<07> - <09>	not used; available for future expansions
	<0a>	OXI. Not implemented
	<0b>	not used; available for future expansions
	For the RSP and ERR responses, it assumes the value passed by the previous GET	
[xx]	The meaning of the parameter depends on the [flag]. Its default value is <ff...fff>. Indicates the description of the group for the RSP response or the GET filter. 32chars	
[yy]	Reports the error, according to the following table:	
	<00>	no error
	<01>	command syntax error
	<02>	command timeout
	<03>	collision on "BusT4"
	<04>	device not found
	<05> - <0f>	not used; available for future developments
	<10>	command execution error
	<11>	protocol data error
	<12>	encoder data error
	<13>	position data error
	<14>	"Bluebus" error. Not implemented
	<15>	memory error
	<16> - <ef>	not used; available for future developments
	<f0>	"BusT4" not operating
	<f1> - <ff>	not used; available for future developments

11. "BusT4" commands: CGR

Associating the CGR command with the FLAG allows you to send a command to a "category group" of automation connected to the BusT4. To create the "category group" it is necessary to use the Nice Screen Configuration Tool. We have defined 4 "category group":

- i. Application type
- ii. Floor number
- iii. Room name
- iv. Orientation

The protocol is as follows (caution! parameter [yy] is not required in the commands; it is sent by the interface solely when returning an error message, such as ERR):

"cgr [flag] [gg] [xx] [ff] [tt] [nn] [rr] [oo] [yy]"

**Type
Flor
Room
Orientation**

[flag]	In combination with GRC, defines the type of message sent:	
	<'>	SET command, request a group command
	#	response of the interface (RSP) indicating that the GET command has been received
	!	response of the interface (ERR) indicating a GET command syntax error
[gg]	Group ID (0 - FF). To enable the Scenes use the Nice Screen Configuration Tool	

The meaning of the following parameters depends on the [flag].

C. For the SET / GET commands and for RSP / EVT responses:		
[xx]	a parameter used to indicate the command that has to be executed. The following parameters are available in the following table:	
	<00> - <01>	not used; available for future developments
	<02>	stop command
	<03>	up command
	<04>	down command
	<05>	intermediate position 1
	<06>	intermediate position 2
	<07>	intermediate position 3
	<08> - <0A>	not used; available for future developments
	<0B>	intermediate position 4
	<0C>	intermediate position 5
	<0D>	intermediate position 6
	<0E> - <2A>	not used; available for future developments
	<2B>	go to % position
	<2C> - <FF>	not used; available for future developments
[ff]	Interpreted only if only if <xx> is 0x2b, Position in from 0 to 100 where 0 is the low limit and 100 is the high limit	
[tt]	Type of automation that has to execute the command:	
	<00>	not used; available for future expansions
	<01>	sliding. Not implemented
	<02>	sectional. Not implemented
	<03>	swing. Not implemented
	<04>	barrier. Not implemented
	<05>	up-and-over. Not implemented
	<06> - <69>	not used; available for future expansions
	<70>	shutters
	<71>	rolling blinds

<72>	sun shades
<73>	projection screens
<74>	umbrella
<75>	mosquito nets
<76>	venetian blinds
<77>	orientable shutters
<78>	patio/ pergola
<79>	curtains for interiors
<7A>	roman blinds
<7B> - <fe>	not used; available for future expansions
<ff>	no filter active
[nn]	Flor of automation that has to execute the command:
<00> - <100>	flor 0 – flor 100
<ff>	every flor (no filter active)
[rr]	Room of automation that has to execute the command:
<00> - <100>	room 0 – room 100
<ff>	every room (no filter active)
[oo]	Orientation of automation that has to execute the command:
<00>	North
<01>	North - East
<02>	East
<03>	South
<04>	South - East
<05>	South - West
<06>	West
<07>	North- West
<ff>	no filter active

D. For ERR responses:

<xx>	Not applicable in case of ERR message	
<yy>	Tells the operator which error has been detected, according to the following table:	
<00>	no error	
<01>	command syntax error	
<02>	command timeout	
<03>	collision on “BusT4”	
<04>	device not found	
<05> - <0f>	not used; available for future developments	
<10>	command execution error	
<11>	protocol data error	
<12>	encoder data error	
<13>	position data error	
<14>	“Bluebus” error. Not implemented	
<15>	memory error	
<16> - <ef>	not used; available for future developments	
<f0>	“BusT4” not operating	
<f1>	not used; available for future developments	

12. "BusT4" commands: VER

The VER commands reads the DMBM interface's firmware version.

The protocol is as follows:

[Flag]	In combination with VER, defines the type of message sent:	
	'<	SET command, sends a request to vary the percentage position of an automation
	'#'	response of the interface (RSP) confirming reception of the GET / SET command

Example – reading the device's firmware version:

```
ver <
```

```
ver # 1.00.05
```

HTTP COMMANDS

Commands syntax

In the following sections there are many symbols with the following meaning;

- **IP_ADDRESS**: it's the IP address of the Gateway used as an interface to communicate with the motors (ex. IP_ADDRESS = 192.168.0.1). If you want to change IP address, subnet mask and gateway use the Nice Screen Configuration tool.
- **ADR**: it's the address of the NICE network where is the blind that you want to control. It can be a value from 1 to 63 (from 1 to 3F). This value has to be in HEX. If the destination is an integration module on the DIN-BAR, this value is 0 (adr=0), if the destination is a Smart Motor this value is 1 (adr=1).
- **EPT**: it's the address of the Nice motor belonging to the network ADR. It can be a value from 1 to 127. This value has to be in HEX. To check the EPT value use the Nice Screen Configuration Tool.
- **CMD**: it's the command that you want to send to the destination (ADR, EPT). The value depend in what you want to do.
- **PRF**: profile set command.
- **FNC**: it's the function that you want to send to the destination (ADR, EPT).
- **EVT**: it's the event to be triggered to destination (ADR, EPT).

•All the fields has to be written in Upper Case.

In the following sections, there is the description of the following commands:

1. Commands

To send basic commands to a specific device destination.

2. Events Commands

To send commands to a specific device destination; these commands are a bit complicated than "commands" but allows you to send more elaborate commands.

3. Group Commands type 1

To send commands to a group of devices; this group is created directly with this command.

All devices belonging to the group will receive the command at the same time. All the device belonging to the group has to have the same ADR. In this command, you can also enter a delay after which will activate the command sent.

4. Group Commands type 2

To send commands to a group of devices; this group is created by "Nice Screen Configuration Tool".

All devices belonging to the group will receive the command at the same time.

5. Scenario Activation

To activate a Scenario; the scenario is created by "Nice Screen Configuration Tool".

6. Set functions

To Set functions to a specific device.

7. Get functions

A. States or information: To Get the states of the functions e/o information from a specific device.

B. Diagnostic: Specific get functions, which show diagnostics of a specific device.

8. Lists

A. List of Device

B. List of Group

C. List of Scenarios

1. Commands **CMD_UP**, **CMD_DOWN**, **CMD_STOP**, **GO_TO_INTERMEDIATE_POSITION_x** (smart motor only)

It is possible to control movements of the devices (or in general Rolling Shutter, awning, Nice tubular motors in general...) in particular it is possible to send commands to our systems commands like UP, DOWN and STOP.

http://IP_ADDRESS/cgi/devcmd.xml?adr=ADR&ept=EPT&cmd=CMD

Where:

- **CMD**: it's the command that you want to send to the motor (ADR, EPT). It can be one of the following.

COMMAND	CMD
UP	0x03
DOWN	0x04
STOP	0x02
INTERMEDIATE POSITION 1	0x05
INTERMEDIATE POSITION 2	0x06
INTERMEDIATE POSITION 3	0x07
INTERMEDIATE POSITION 4	0x0B
INTERMEDIATE POSITION 5	0x0C
INTERMEDIATE POSITION 6	0x0D

Examples:

UP COMMAND: <http://192.168.0.1/cgi/devcmd.xml?adr=1&ept=2&cmd=3>
DOWN COMMAND: <http://192.168.0.1/cgi/devcmd.xml?adr=1&ept=2&cmd=4>
STOP COMMAND: <http://192.168.0.1/cgi/devcmd.xml?adr=1&ept=3&cmd=2>
GO TO INTERMEDIATE POSITION: <http://192.168.0.1/cgi/devcmd.xml?adr=1&ept=2&cmd=5>

2. Events Commands CMD_GO_TO_POSITION (smart motor only)

To send commands to a specific device destination; these commands are a bit complicated than “commands” but allows you to send more elaborate commands.

[http:// IP_ADDRESS /cgi/devevt.xml?adr=ADR&ept=EPT&evt=EVT&dat= DAT](http://IP_ADDRESS/cgi/devevt.xml?adr=ADR&ept=EPT&evt=EVT&dat=DAT)

COMMAND	EVT	DAT	DESCRIPTION
GO TO POSITION	C0	0 TO 64	Command a device to go to a specific position where 0 is the lower limit switch and 100 (64 in Hex) is high limit.

Examples:

cmd_gotoposition(50%)
<http://192.168.0.1/cgi/devevt.xml?adr=01&ept=02&evt=C0&dat=32>
 Dat is 32 because you have to insert the HEX value.

cmd_gotoposition(100%)
<http://192.168.0.1/cgi/devevt.xml?adr=01&ept=02&evt=C0&dat=64>
 Dat is 64 because you have to insert the HEX value.

3. Group Commands type 1

It is possible to send commands (to our systems commands like UP, DOWN and STOP) to group of actuators; each actuator will receive the same command. In this command, you can also enter a delay after which will activate the command sent.

http://IP_ADDRESS/cgi/devevt.xml?adr=ADR&ept=FF&evt=EVT&dat=WWYYZZZ ZXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Specifications:

- **EPT:** this is a broadcast command in the ADR network so this field has to be always 0xFF.
- **EVT:** this field for group command type 1 has to be 0xC2.
- **DAT:** it's the data transmitted. It is composed of 4 subfields:
 - o **WW:** it's the command that you want to send
 - o **YY:** is data for the command
 - o **ZZZ:** delay after which will activate the command sent in milliseconds
 - o **XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX:** xxx represents the recipients of the command all belonging to ADR.
 Bit mask to activate the command to the respective motor: you have to follow the following formula for each motor where EPT is the address of the motor:

$$\text{mskToSend}[(\text{EPT} / 8)] = (0x01 \ll ((\text{EPT} \& 7)))$$

COMMAND	WW	YY
UP	0x03	00
DOWN	0x04	00
STOP	0x02	00
INTERMEDIATE POSITION 1	0x05	00
INTERMEDIATE POSITION 2	0x06	00
INTERMEDIATE POSITION 3	0x07	00
INTERMEDIATE POSITION 4	0x0B	00
INTERMEDIATE POSITION 5	0x0C	00
INTERMEDIATE POSITION 6	0x0D	00
GO TO POSITION	0x2B	from 00 to 64 (0% to 100%)

Note: cmd_gotoposition

The engine in this case will be commanded to go from one position between 0 and 100 where 0 is the lower limit and 100 is high limit.

o **BitMask:**

$$\text{mskToSend}[(\text{EPT} / 8)] = (0x01 \ll ((\text{EPT} \& 7)))$$

o **XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX**: are 16byte that equals to 128 bits, every bit represent a destination. If you want to address a command to a specific address the corresponding bit as to change from 0 to 1.

Example:

EPT n	mskToSend [HEX]															
EPT 0	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15
	01	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
EPT 1	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15
	02	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
EPT 2	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15
	04	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
EPT 3	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15
	08	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
EPT 4	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15
	10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
EPT 5	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15
	20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
...	...															
EPT 8	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15
	00	01	00	00	00	00	00	00	00	00	00	00	00	00	00	00
...	...															
EPT 127	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15
	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	10

Examples 1: UP to the motor 1 using the Group Command type 1

<http://192.168.0.1/cgi/devevt.xml?adr=01&ept=FF&evt=C2&dat=03000000020000000000000000000000000000000000>

- **IP_ADDRESS:** 192.168.0.1
- **ADR:** 01
- **EPT:** FF
- **EVT:** C2 (group command type 1)
- **DAT:**
 - o **WW:** 03, UP CMD
 - o **YY:** 00, NO DATA
 - o **ZZZZ:** 0000, NO DELAY
 - o **XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX:** 0200 (activation on EPT 1)

6. Sets functions

Messages to set in the engine special features:

http:// IP_ADDRESS /cgi/devset.xml?adr=ADR&ept=EPT&prf=PRF&fnc=FNC&dat=DAT

Where:

- **PRF:** for this application has to be 6.
- **FNC, DAT:** It can be one of the following:

COMMAND	FNC	DESCRIPTION (DAT)
"wink"	0x35	If DAT is 1 the command is to ENABLE the WINK If DAT is 0 the command is to DISABLE the WINK
"motor lock"	0x2a	If DAT is 1 the command is to Lock the motor If DAT is 0 the command is to UnLock the motor

Examples:

Enable wink feature: <http://192.168.0.1/cgi/devset.xml?adr=01&ept=02&prf=06&fnc=35&dat=01>

Disable wink feature: <http://192.168.0.1/cgi/devset.xml?adr=01&ept=02&prf=06&fnc=35&dat=00>

Lock the Motor: <http://192.168.0.1/cgi/devset.xml?adr=01&ept=02&prf=06&fnc=2A&dat=01>

UnLock the Motor: <http://192.168.0.1/cgi/devset.xml?adr=01&ept=02&prf=06&fnc=2A&dat=00>

7. Get functions

Request messages:

http:// IP_ADDRESS /cgi/devget.xml?adr=ADR&ept=EPT&prf=PRF&fnc=FNC&dat=

Answer of the system:

```
<nice version="1.0">
<adr/>
<ept/>
<prf/>
<fnc/>
<val>XXXXXXXXXXXX</val>
```

- **PRF:** for this application has to be 06.
- **FNC:** the function to get.

COMMAND / REQUEST	FNC	ANSWER DESCRIPTION										
"Read State & position"	0x01	<p>After the request message, you receive two bytes, the first containing the status of the motors and the second position in percentage.</p> <ul style="list-style-type: none"> • First byte: <table border="1"> <thead> <tr> <th>State</th> <th>Answer</th> </tr> </thead> <tbody> <tr> <td>Motor in closing</td> <td>0x03</td> </tr> <tr> <td>Motor in opening</td> <td>0x02</td> </tr> <tr> <td>Motor stopped</td> <td>0x01</td> </tr> <tr> <td>Unknown state</td> <td>0x00</td> </tr> </tbody> </table> • Second byte: <p>A value between 0 and 100. In particular, 0 corresponds to the low limit switch and 100 corresponds to the high limit switch.</p> <p><i>Examples:</i> http://192.168.0.1/cgi/devget.xml?adr=00&ept=03&prf=06&fnc=1&dat=</p> 	State	Answer	Motor in closing	0x03	Motor in opening	0x02	Motor stopped	0x01	Unknown state	0x00
State	Answer											
Motor in closing	0x03											
Motor in opening	0x02											
Motor stopped	0x01											
Unknown state	0x00											
Wink state	0x35	<p>After the request message, you receive one byte. If the byte received is 1 the Wink is enabled, if the byte received is 0 the Wink is disabled.</p>										
Lock state	0x2a	<p>After the request message, you receive one byte. If the byte received is 1 the "lock motor" function, if the byte received is 0 the "lock motor" function is disabled.</p>										

DIAGNOSTIC	FNC	ANSWER DESCRIPTION
"NUMBER OF MOVEMENTS" (smart motor only)	D4	Number of movements (6 digit in hex).
"TOTAL MOVEMENTS TIME" (smart motor only)	D5	Total time of movements in seconds (6 digit in hex).
"ERRORS MEX" (smart motor only)	DD	BITMASK of the errors (4 digit in hex).

Meaning	MSK
Internal problem 1	0b0000000000000001
Voltage to low for the electronic	0b0000000000000010
Voltage to low for the motor	0b0000000000000100
Voltage to low at start up	0b0000000000001000
Hight temp 1	0b0000000000010000
Hight temp 2	0b000000000100000
Hight temp 3	0b000000001000000
Internal problem 3	0b000000010000000
Hight temp 4	0b000000100000000
Last move impatc	0b0000001000000000
Internal problem 4	0b0000010000000000
Internal problem 5	0b0000100000000000
Internal problem 6	0b0001000000000000
Internal problem 7	0b0010000000000000
Internal problem 8	0b0100000000000000

Examples 1: number of movements:

Request: `http://192.168.0.1/cgi/devget.xml?adr=01&ept=02&prf=06&fnc=D4&dat=`
 Answer: `<val>003892</val>`
 The Number of movements is 14482 (003892 in HEX is 14482 in DEC)

Examples 2: total movements time:

Request: `http://192.168.0.1/cgi/devget.xml?adr=01&ept=02&prf=06&fnc=D5&dat=`
 Answer: `<val>00BB83</val>`
 The movements time is 48003 seconds (00BB83 in HEX is 48003 in DEC)

Examples 3: errors mex:

Request: `http://192.168.0.1/cgi/devget.xml?adr=01&ept=02&prf=06&fnc=DD&dat=`
 Answer: `<val>0000</val>`
 The interpretation is: 0000 in Hex is 0000000000000000 in binary, so there is none error.

Answer2: `<val>0006</val>`

The interpretation is: 0006 in Hex is 0000000000000110 in binary, so there are two errors: Voltage to low for the electronic and Voltage to low for the motor.

8. Lists

These features display the list of installed devices, the list of groups and scenarios configured with the Nice Screen Configuration Tool.

A. List of Device

Request messages: `http:// IP_ADDRESS /cgi/devlst.xml`

B. List of Groups

Request messages: `http:// IP_ADDRESS /cgi/grplst.xml`

C. List of Scenarios

Request messages: `http:// IP_ADDRESS /cgi/scnlst.xml`



Nice

Nice S.p.A.
Via Pezza Alta, 13
31046 Oderzo TV Italy
info@niceforyou.com

www.niceforyou.com