Overview

This document is a reference for the local control protocol using the device TCP port. This protocol supports the following connection method:

- Local WiFi network communication using HTTP/IP GET requests

Before using the local control communication, set up the controller and blinds using our “Neo Smart Blinds” app found at Google Play, Itunes or web Chrome version. Follow the in app instruction for adding the controller(s) and blind(s).

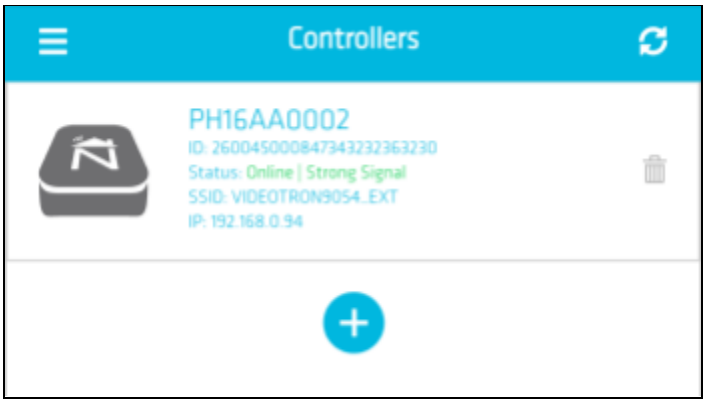
Blinds can be individually or group/room controlled (max 15 per group).

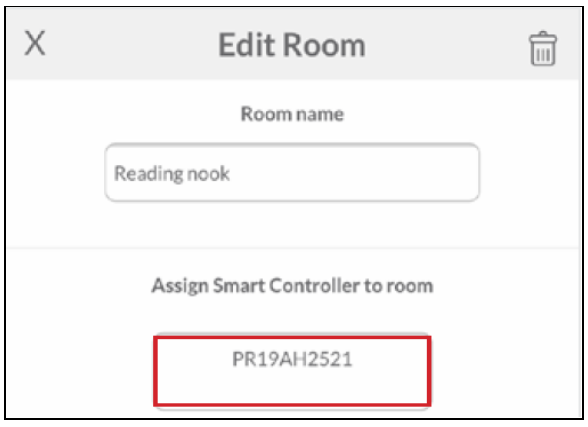
1. Protocol Conventions

Commands are case-insensitive. All commands and their responses are terminated with a carriage return/line feed pair (CRLF).

Local connection requires previous knowledge of IP and port addresses.

- Controller local IP can be found using the Neo Smart Blinds app, at “Your Controllers”. For accounts with more than one controller make sure to use the IP address from the controller assigned to the room where the respective blind is located.
- Use port number 8838 (for http/ip communication).
- The controller’s channel and ids can be found on each blinds’ advanced page using the Neo Smart Blinds app.





Each new request/command should open a new connection between the client (app or other devices) and the server (controller). The server will close the connection after every single request/command.

2. Script implementation

When implementing this protocol to an automated script some considerations need to be taken into account.

- The TCP server can only handle as many as 1-3 simultaneous client connections at maximum. However, it's highly recommended to only try one connection at a time.
- Each new client connection is closed by the server as soon as the message is received and replied. Which means that before the command is validated and finally transmitted. That explains why only an echo message is returned to the client.
- Each received command takes at least 500ms to be verified, processed and transmitted by the controller/hub. [OPTIONAL] Add a delay of at least 100-500ms between each TCP connection and command transmission if many commands must be sent at once.

3. HTTP "transmit" function

3.1. HTTP Request

Syntax: *GET /neo/v1/transmit{?params}*

Params:

command:	"command"	
id:	"controller_id"	
hash:	"unique_hash#"	**[optional]**

command: - It's a valid expression containing group_id, channel_id and action syntax is fully described below.

id: - It's the valid controller_id number, composed of 20+ characters. You can find it using the app in "Your Controller"

hash: - It's 7 digits representing an integer number, unique, used to avoid sending more than once a single command when it comes from different sources (e.g. Cloud request). A simple implementation is taking the last 7 digits from time.now() in milliseconds.

Example:

GET /neo/v1/transmit?command=001.001-00-mu!no&id=380030001047343339383037&hash=2941795

3.2. HTTP Responses:

Syntax:

```
HTTP/1.1 409 Conflict\r\n
Content-Type: text/html\r\n
Content-Length: 0\r\n
Access-Control-Allow-Origin: *\r\n
```

Response codes	Conditions
200 OK	Message received and transmitted by the Controller
400 Bad Request	Command not found or not valid
401 Unauthorized	Device_ID was not found or not valid
404 Not Found	"/neo/v1/transmit" not found or not valid
409 Conflict	hash # found and format is valid, but # was already used

3.3. Command protocol

length : 16 characters
syntax : "**ID1.ID2-CHANNEL-ACTION!MOTORCODE"**

"ID1" : controller byte 1, from integer "000" to "255"
"." : address separator
"ID2" : controller byte 2, from integer "000" to "255"
"-" : channel separator
"CHANNEL" : channel, individual channel from integer "01" to "14", use channel "15" for a group
"-" : command separator
"ACTION" : action, as follows
- "up" : blinds move UP
- "dn" : blinds move DOWN
- "sp" : blinds STOP moving
- "gp" : go to favourite position
- "mu" : micro-step up
- "md" : micro-step down
- "XX" : go to a specific position from 01 to 99. ***ONLY** available for use with motors which **motor code is "no"**.*

Options only available when using motor code “tb”, TopDown-BottomUp blinds.

"op” : open shades, move both rails UP
"cl” : close shades, move top rail UP and lower rail down
"u3” : move both rails UP
"u2” : move lower rail UP
"u3” : move both rails UP
“d2” : blinds lower rail DOWN
“d3” : blinds both rails DOWN
“o2” : micro-step lower rail UP
“c2” : micro-step lower rail DOWN

“!” : exclamation symbol, command separator preceding the motor code

“MOTORCODE”: code designation to indicate the motor protocol to be transmitted

“bf” : standard Bofu motor
“vb” : vertical Bofu motor
“tb” : top-down/bottom-up Bofu motor
“no” : NEO motor
“rx” : Raex motor
“nc” : Nice motor
“k1” : standard A-OK motor
“k2” : version 2 A-OK motor (louvolite exclusive)
“dy” : standard unidirectional Dooya motor
“by” : special code compatible with both Bofu and Dooya standard motors
“wt” : standard Wistar and Alpha motor

Exemples:

- 123.123-05-up!no
- 012.005-15-dn!bf (for controlling a group of blinds)
- 129.021-02-sp!dy
- 129.021-06-87!no

3.4. Finding Controller ID and Channels from APP

Using the app, go to each blind's advanced page to find the necessary ID numbers. See the image below.

All blinds inside the same ROOM have the same unique combination of Controller ID codes. Use CHANNEL = 15 to control all blinds as a group.

To control individual blinds, use the indicated Motor Channel.

That's the same behaviour found using physical remote controls.