

Table of contents

1. General	1
2. Product overview	1
2.1 Hardware interfaces	2
2.2 Control buttons	3
2.3 Status LEDs	4
2.4 Typical setups	4
3. USB Serial configuration	6
4. UDP configuration	7
4.1 Enable or Disable UDP communication	7
4.2 Identify NEO controllers on your network	8
4.3 Configure UDP and Network settings	9
4.4 Adjust the controller behaviour	11
4.5 Diagnostic tools	13
4.6 Create start-up commands	14
5. Functionality and API commands	15
5.1 Measure power parameters	16
5.2 Control power outputs	20
5.3 Scheduling	22
5.4 Monitoring	23
5.5 Device settings	29
5.6 Resets	30
5.7 X-talk sensors	31
6. Sensmi configuration	32
6.1 Enable or Disable Sensmi connection	33
6.2 Provisioning	34
6.3 Dashboard	35

1. General

NEO power controllers can measure and control their on-board power outputs. Typically this is used to measure the power consumption of each component in an installation, remotely control start-up and shut-down sequences, and reduce energy consumption.

2. Product overview

NEO power controllers are available in 6 different models:

	NEO 320	NEO 520	NEO 620	NEO 340	NEO 540	NEO 640
Outputs <i>individually controllable</i>	2 outputs C13			4 outputs C13		
Measure parameters	V voltage	A current	W power	V voltage	A current	W power
Control <i>output switching</i>	-	✓	✓	-	✓	✓
USB <i>API interface</i>	✓	✓	✓	✓	✓	✓
Network <i>connectivity</i>	-	-	✓	-	-	✓
Cloud <i>dashboard</i>	-	-	✓	-	-	✓
X-talk <i>channels</i>	4 X-talk channels for sensor interfacing			4 X-talk channels for sensor interfacing		
Input <i>voltage range</i>	100 - 230 VAC			100 - 230 VAC		
Power <i>max rating</i>	10A			10A		

This manual covers the functionalities of the NEO620 and NEO640.

2.1 - Hardware interfaces

FRONT

NEO620 | NEO640



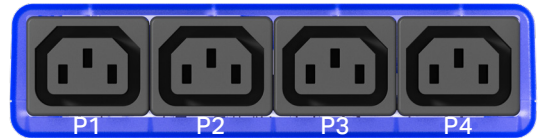
1	X-talk interface 001
2	X-talk interface 002
3	X-talk interface 003
4	X-talk interface 004
U	USB-C API interface
N	RJ45 Network connector
IN	C14 Power input

BACK

NEO620



NEO640



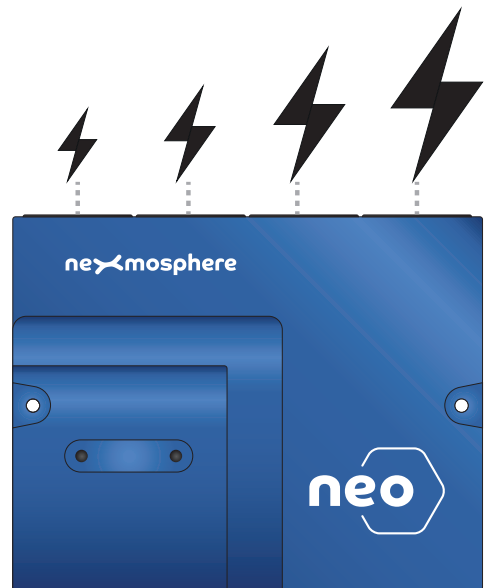
P1	C13 Power output 1
P2	C13 Power output 2
P3	C13 Power output 3
P4	C13 Power output 4

Power output connection order

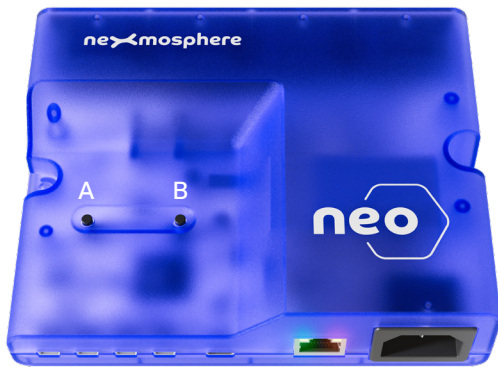
The NEO controller performs highly precise, detailed power measurements for which (small) cross talk can already cause fluctuation in the power measurement data.

To make sure the NEO controller measurements are as precise as possible, **connect the device with the highest power consumption to power output 1**, the device with the second-highest power consumption to output 2, and so on.

Furthermore, **unused or unconnected power outputs must be switched off** (see section 5.2 Control Power Outputs).



2.2 - Control buttons

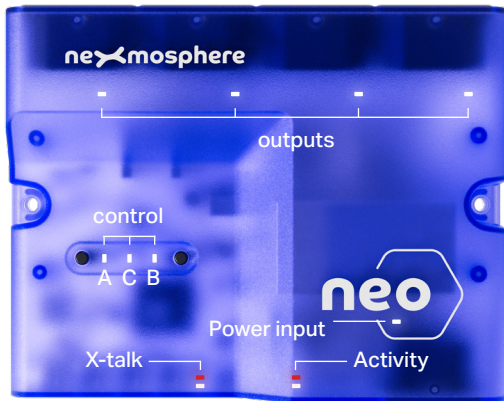


A	Control button A
B	Control button B

The control buttons provide options for manually adjusting the mode when you have physical access to the NEO device. These are explained in the chapter of each corresponding functionality. Below is a summary of the available options.

Initiate start-up sequence	Press button A for 5 seconds <i>API message when start-up sequence is initiated</i> P000B[BUTTON=STARTUPSEQ]
Initiate shut-down sequence	Press button B for 5 seconds <i>API message when shut-down sequence is initiated</i> P000B[BUTTON=SHUTDOWNSEQ]
Enable or Disable UDP communication	Press button A for 10 seconds <i>API message when UDP is enabled</i> <i>API message when UDP is disabled</i> S000B[UDP=ENABLED] S000B[UDP=DISABLED]
Send UDP identification message	Press button A for 1 second <i>API message when UDP identification message is initiated</i> N000B[IP=xxx.xxx.xxx.xxx] N000B[PORTIN=xxxx]
Activate Sensmi Provisioning mode	Press button A+B for 1 second <i>API message when Sensmi Provisioning mode is activated</i> SENSMI[PROVISIONING ACTIVE]
Set the NEO to "Simple Socket" mode	Push button A+B for 10 seconds <i>API message when Simple Socket mode is initiated</i> P000B[BUTTON=SIMPLESOCKET]
Set the NEO back to "Smart Socket" mode	Push button A+B for 10 seconds <i>API message when Smart Socket mode is initiated</i> P000B[BUTTON=SMARTSOCKET]
Trigger "button is pressed" API message	Press button A or press button B <i>API message for button A</i> <i>API message for button B</i> P000B[BUTTONA=PRESSED] P000B[BUTTONB=PRESSED]
Trigger "button is released" API message	Release button A or release button B <i>API message for button A</i> <i>API message for button B</i> P000B[BUTTONA=RELEASED] P000B[BUTTONB=RELEASED]
Factory reset	Press button A+B for 20 seconds <i>API message when factory reset is initiated</i> S000B[FACTORYRESET]

2.3 - Status LEDs



Outputs

Each power output has a dedicated status LED which is on when the output is switched ON, and off when the output is switched OFF.

Power input

When the AC power input is above 80V, the status LED will be on. If the AC power input is below 80V, the status LED will be off.

X-talk

The X-talk status LEDs show the same behaviour as Nexmosphere controllers with USB API interface. The white LED blinks when a sensor is triggered, or an X-talk command is received. The red LED will blink when an invalid X-talk command is received.

Activity

The white LED blinks when a command is sent or received, either via USB or Ethernet (UTP). The red LED blinks when an invalid command is received, or, in case of a controller error.

Control

When button A is pressed, control LED A starts to blink slowly. If the button is held for 5 seconds, it will continue to blink at medium speed. If the button is held for 10 seconds, it will continue to blink fast. If the button is held for 20 seconds, the LED will go off. The same applies to button B and control LED B. Control LED C is used to identify units via the N000B[BLINK!] command.

2.4 - Typical setups

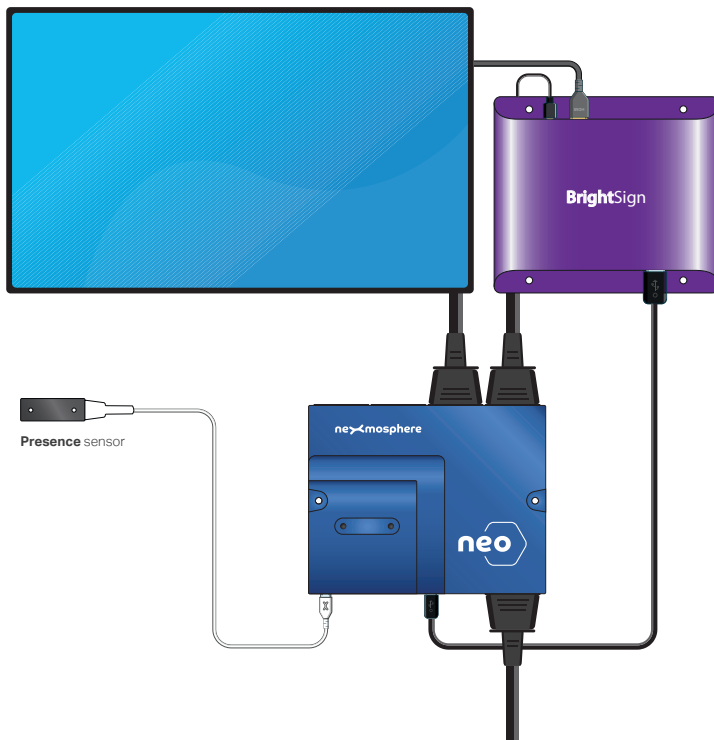


Example 1: USB-C connection to player

The NEO controller is connected to a PC or digital signage player via USB-C.

The digital signage player utilizes the USB API to control the power outputs on the NEO device and obtain power-measurement data.

Both the digital signage player and the screen are powered via the NEO controller.

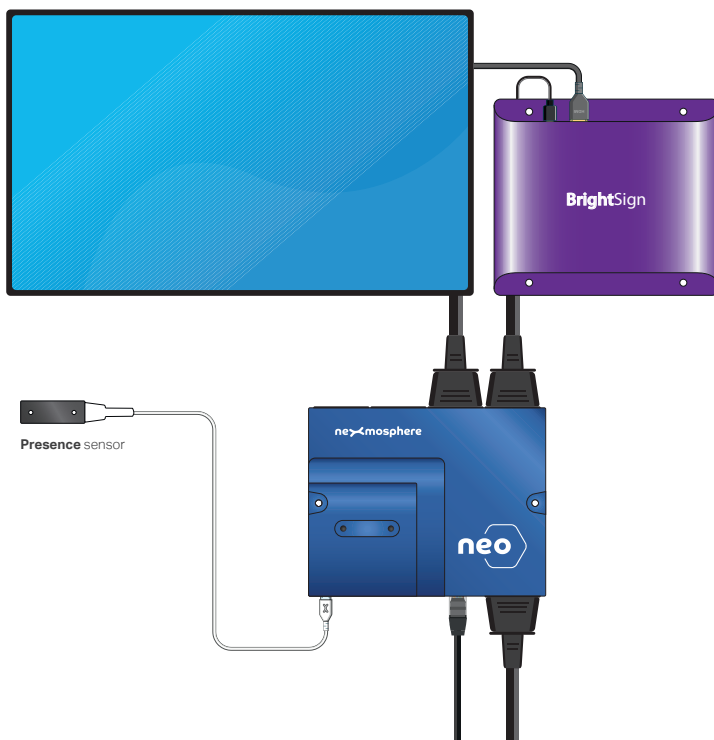


Example 2: USB-C connection, with sensor

This example is similar to Example 1. In this case, a presence sensor is added, the output of which will also be sent to the digital signage player.

Based on the sensor status, the NEO controller can be set to switch ON or OFF power outputs, for example to switch on a screen when a person is detected by the presence sensor.

Other examples include connecting an ambient light sensor, and using the ambient light value as input to adjust the screen brightness to save power.



Example 3: Network connection

Instead of connecting the NEO controller via USB, NEO 600 series can connect to your network via its UTP connector.

The UDP API can be used to control the power outputs on the NEO device and obtain power-measurement data.

Pre-built online dashboards powered by Sensmi provide real-time power measurement data and power controls.

Both UDP communication and Sensmi connectivity can be disabled or enabled at any time.

3.1 USB Serial configuration

When connecting the NEO controller via its USB-C connector to a PC or player, a virtual COM port is created on the connected device. The Virtual COM Port chip used in the NEO controller is the Prolific PL2303GL. The driver for this chip is typically already embedded in the OS (for example BrightSign, Windows, Android, Linux, Tizen). The latest driver can be downloaded on Nexmosphere's Integrator Portal. <https://www.nexmosphere.com/technicalresources>

The virtual COM port facilitates serial communication between the NEO controller and the connected device. Each API command explained in this manual is an ASCII command sent or received via the virtual COM port. The following settings must be applied on the serial port.

Baudrate	115200	Flow Control	None
Parity	None	EOL	CR+LF
Data	Bits 8	Protocol	ASCII
Stop	Bits 1		

Software setup for USB testing (Terminal)

Typically, the NEO controller is connected to a 3rd party device, such as a Digital Signage Player, on which CMS software is installed that has built-in functionality for sending and receiving Serial Events. However, if you want to do a first test on a PC, follow the instructions below:

1. Download a terminal program. For example [Termite](#) or [Hercules](#).
2. Open the Terminal program and go to settings. Choose the COM port on which the NEO controller enumerated.
In most cases, this is the highest available number in the COM port drop-down setting.
3. Set the COM port settings to the values indicated above
4. Set the COM port to "Open". **The NEO controller is now ready for use via USB.**

4. UDP configuration

All NEO 600 series controllers have an on-board UTP socket to connect the controller to your local network via a UTP cable. This allows the API messages from the NEO, for example power data or sensor triggers, to be sent over your local network to cover large distances (for example to a server room) or create a one-to-many connection. Vice versa, API commands can also be send to the NEO controller via UDP commands, for example to control the power outputs. The factory default Network/UDP settings are:

- IP Mode = **DHCP**
- Default output mode = **Broadcast**
- Incoming port = **5000**
- Outgoing port = **5000**

4.1 Enable or Disable UDP communication

Per factory default, UDP communication is disabled for security reasons. To enable UDP communication, the following API command can be send via the USB serial connection:

```
S000B [UDP=ENABLE]
```

Enable UDP communication

If UDP communication is disabled, it can also be enabled via the physical control buttons on the NEO device itself, by **pressing button A for 10 seconds**. For more information, please see the section 2.2 "Control buttons". Both when UDP is enabled via a UDP API command, or via a physical button press, the reply from the NEO controller will be as follows:

```
S000B[UDP=ENABLED]
```

After UDP is enabled it will immediately try to connect to your router with its current network settings. If the connection is made successfully, the NEO controller will automatically send out UDP identification messages on your local network. For more information, please see the section 4.2 "Identify NEO controllers on your network".

Enabling UDP communication only needs to be done once. It will stay enabled also after power-cycling the device.

UDP communication can be disabled at any time. To disable UDP communication, the following API command can be send either via UDP or USB:

```
S000B [UDP=DISABLE]
```

Disable UDP communication

If UDP communication is enabled, it can also be disabled via the physical control buttons on the NEO device itself, by **pressing button A for 10 seconds**. For more information, please see the section 2.2 "Control buttons". Both when UDP is disabled via a UDP API command, or via a physical button press, the reply from the NEO controller will be as follows:

```
S000B[UDP=DISABLED]
```

Disabling UDP communication only needs to be done once. It will stay disabled also after power-cycling the device.

4.2 - Identify NEO controllers on your network

When there is physical access to the NEO controller, you can identify it on your local network and obtain its IP address, by following these steps:

1. Connect the NEO controller to your router and power it. Wait for 20 seconds.
2. Have a UDP terminal program on your PC/laptop **listen to port 5000**.
3. On the NEO device, **press button A** for 1 second
4. The controller will broadcast **2 identification messages** over the local network on port 5000.

FROMID=***:N000B[IP=XXX.XXX.XXX.XXX]**
FROMID=***:N000B[PORTIN=PPPPP]**

XXX.XXX.XXX.XXX = IP Address of the NEO device
PPPPP = Incoming (receiver) port of the NEO device

****** = Controller ID*
Length: 1-24 ASCII characters
Default: MAC address of the controller

Example messages:
FROMID=00:08:DC:75:79:8E:N000B[IP=192.168.1.171]
FROMID=00:08:DC:75:79:8E:N000B[PORTIN=5000]

Alternatively to pressing button A, a set of API commands is available to identify NEO controllers via UDP on your local network. As the IP address of the controller is usually not yet known when sending an identification command, these commands should be **broadcast** (ip 255.255.255.255) on your network. The default incoming and outgoing port of the NEO controller is **5000**. There are 3 identification commands available:

N000B [IDENTIFY=ALL]	Identify all NEO devices
N000B [IDENTIFY=DEFAULT]	Only identify unconfigured NEO devices <i>(with factory default UDP settings)</i>
N000B [IDENTIFY=CONFIGURED]	Only identify configured NEO devices <i>(with configured UDP settings)</i>

For all 3 options, the reply is identical to the identification messages which are sent when pressing button A. Per default, a NEO device will also automatically send an identification once, each time it is booted.

Please note that for configured controllers with non-default settings, the network settings and identification messages might be different:

- The Incoming or Outgoing port might be different than 5000
- The Output mode might be set to a specific destination IP, instead of Broadcasting
- The Controller ID is a customized ID, for example "LOBBY", and is depicted instead of the MAC address

Example message: **FROMID=LOBBY:N000B[IP=192.168.1.171]**
FROMID=LOBBY:N000B[PORTIN=5000]

See section 4.4 "Adjust the NEO controller behaviour" for more information on how to change the Controller ID

- The FROMID prefix is disabled and therefore not included in the identification messages

Example message: **N000B[IP=192.168.1.171]**

- The Index prefix is enabled and included in the identification messages

Example message: **#00001:FROMID=LOBBY:N000B[IP=192.168.1.171]**
#00002:FROMID=LOBBY:N000B[PORTIN=5000]

4.3 Configure UDP and Network settings

In this section, the commands to configure the UDP communication and Network settings are listed.

Important! To initiate or save new settings, one of the following commands must always be sent *after* sending one or multiple configuration commands:

N000B [INITIATE!]	Initiate the new settings, but don't save them <i>after a reboot, it will go back to its previous settings</i>
N000B [SAVE!]	Initiate the new settings, and save them <i>also after a reboot, the new settings will apply</i>

IP Mode - Configure how the NEO controller obtains its IP

N000B [MODE=DHCP]	Set the IP mode to DHCP <i>to obtain an available IP address from your router</i>
N000B [MODE=FIXEDIP]	Set the IP mode to have a fixed IP <i>to make sure the IP address of the controller will not change</i>
N000B [FIXEDIP=XXX . XXX . XXX . XXX]	Set the fixed IP XXX.XXX.XXX.XXX = IP address Default 192.168.0.100 Example command: N000B[FIXEDIP=192.168.1.172]
N000B [SUBNETMASK=XXX . XXX . XXX . XXX]	Set the subnet mask XXX.XXX.XXX.XXX = subnet mask Default - 255.255.255.0 Example command: N000B[SUBNETMASK=255.255.255.0]

When setting the IP mode to DHCP, please be aware that the IP might be different after a reboot of the controller. In case of setting the mode to fixed IP, please make sure this IP is available and not assigned to another device on your network. Secondly, when setting the mode to fixed IP, also make sure the subnetmask matches that of your router.

Ports - Configure Incoming and Outgoing port of the UDP communication

N000B [PORTIN=XXXXX]	Set the Incoming (receiver) port of the NEO controller <i>this should match the Outgoing (destination) port of your player</i> XXXXX= Incoming port between 1 - 65535 Default = 5000 Example command: N000B[PORTIN=4999]
N000B [PORTOUT=XXXXX]	Set the Outgoing (destination) port of the NEO controller <i>this should match the Incoming (receiver) port of your player</i> XXXXX= Outgoing port between 1 - 65535 Default = 5000 Example command: N000B[PORTOUT=5001]

Per default, the incoming and outgoing port are both set to 5000. For field operation, it is recommended to set the incoming and outgoing port to different values, to avoid undesired echoes and cross communication between multiple NEOs on the same network.

Output mode - Configure to what IP the NEO controller sends its messages

N000B [DESTIP=BROADCAST] Set the NEO device to broadcast its messages *Default*
to all devices on your local network (on the same subnet)

N000B [DESTIP=XXX.XXX.XXX.XXX] Set a specific destination IP to which the API messages are sent
typically this is the IP of your player
XXX.XXX.XXX.XXX = IP address
Example command:
N000B[DESTIP=192.168.1.175]

Per factory default, the NEO unit will broadcast all outgoing messages. Broadcasting is also convenient if the power data or sensor triggers should be sent to multiple devices on your network. For most point-to-point applications, setting a fixed destination IP is the recommended option, as this has the least overhead on your network.

When using multiple NEO devices in the same network that broadcast, using an ID prefix for incoming commands is strongly recommended to avoid a broadcast storm or ACK implosion. For more information on the ID prefix, please see the section 4.4 "Adjust the controller behaviour".

Message termination - Configure the EOL termination for the UDP messages which the NEO controller sends

N000B [TERMINATION=] No message termination

N000B [TERMINATION=00] Set termination to 0x00

N000B [TERMINATION=CR] Set termination to CR (Carriage Return)

N000B [TERMINATION=LF] Set termination to LF (Line Feed)

N000B [TERMINATION=CRLF] Set termination to CR + LF *Default*

N000B [TERMINATION=X] Set termination to custom character *x = ASCII character*
Example command to terminate with @ character:
N000B[TERMINATION=@]

N000B [TERMINATION=X|Y] Set termination to double character *x-y= ASCII characters*
Example command to terminate with @> characters:
N000B[TERMINATION=@|>]

The message termination should match the required termination of your player. For example, BrightSign players require the termination of UDP messages to be 0x00.

4.4 Adjust the NEO controller behaviour

The NEO controller offers several options for adjusting its behaviour and UDP command structure. The available API commands are as follows:

Outgoing message prefix - Configure the prefix of the outgoing messages

N000B [PREFIX=ID]	Set the Prefix to be the NEO controller ID <i>Per default, the controller ID is the MAC address</i>	<i>Default</i>
N000B [PREFIX=NO]	Disable prefix of outgoing messages <i>This will make the outgoing messages significantly shorter</i>	

Having a prefix in the outgoing messages ensures that you can filter from which controller a specific message was sent, in case of multiple controllers broadcasting on the same network. In case of having only one UDP controller on the same local network, it is recommended to disable the prefix, as this simplifies the communication.

Incoming command prefix - Set the requirement for a prefix on incoming commands

N000B [REQUIREID=NO]	Set the controller to not require the Controller ID as a prefix for all incoming commands	<i>Default</i>
N000B [REQUIREID=YES]	Set the controller to require the controller ID as a prefix for all incoming commands	

When the NEO device is set to require the controller ID as prefix, the following prefix must be added to *all* commands which are sent to the NEO controller.

TOID=***** :	<i>***** = Controller ID</i> <i>Length: 1-24 ASCII characters</i> <i>Default: MAC address of the controller</i> <i>Example command to a controller with Controller ID "ABC"</i> TOID=ABC:N000B[BLINK!]
---------------------	---

This prefix is recommended in scenarios where multiple NEO devices are broadcasting on the same network, to avoid a broadcast storm or ACK implosion. For all other scenarios it is recommended to leave the incoming message prefix disabled, which is the default factory setting.

Controller ID - adjust the Controller ID

N000B [ID=*****]	<i>***** = Controller ID</i> <i>Length: 1-24 ASCII characters</i> <i>Default: MAC address of the controller</i> <i>Example command to set the Controller ID to "SEC1"</i> N000B[ID=SEC1]
---------------------------	---

The Controller ID provides the option to identify a NEO controller other than via its IP address. The controller ID is user settable and can be any word/number/combination. Please consider that when the controller ID is used as prefix for outgoing and/or incoming messages, the length of the controller ID also affects the length of the incoming / outgoing messages. A long controller ID will result in a long prefix which results in long messages.

Acknowledgement message - to confirm receipt of incoming Network messages

N000B [ACK=ON]

NEO will reply on incoming commands with **N000B[ACK]**

N000B [ACK=ECHO]

NEO will echo the received commands *Default*
The controller will transmit an exact copy of the received message

N000B [ACK=OFF]

NEO will not send any acknowledgement messages

Acknowledgement messages facilitate an additional check in the communication which confirms that a message sent to the NEO controller was received. If ACK messages are enabled (ON or ECHO) and an incorrect message is received, the NEO device will reply with **N000B[NACK]**

Startup message - to identify NEO controllers at startup

N000B [STARTUPMESSAGE=ID]

The NEO unit will send an Identification message *Default*
automatically at startup

Example identification message:
FROMID=00:08:DC:75:79:8E:N000B[IP=192.168.1.171]
FROMID=00:08:DC:75:79:8E:N000B[PORTIN=5000]

N000B [STARTUPMESSAGE=NO]

The NEO unit will not send an Identification message at startup
The controller will transmit an exact copy of the received message

An Identification message can also be triggered by **pressing button A** for 1 second.

Index prefix - to check the communication sequence

N000B [INDEXNR=ON]

The NEO controller will add an upward-counting index number
at the start of each outgoing message

Example messages (presence sensor) with index number and ID prefix :
#00011:FROMID=00:08:DC:75:79:8E:X001B[Dz=05]
#00012:FROMID=00:08:DC:75:79:8E:X001B[Dz=XX]

Example messages (presence sensor) with index number and no prefix:
#00011:X001B[Dz=05]
#00012:X001B[Dz=XX]

N000B [INDEXNR=OFF]

The NEO Controller will not add an index number *Default*
to outgoing messages

Index numbers facilitate the implementation of a mechanism that checks if outgoing UDP messages from the NEO controller got lost. Packet loss on properly working networks is very low (<0.1%). Therefore it is recommended to only enable index numbers to check message loss when the local network can be busy and congested.

4.5 Diagnostic tools

The NEO controller offers several commands for Network diagnostics and debugging.

Diagnostic network commands

N000B [CONFIGURED?]

Check if the NEO controller is configured or not (factory default)

The controller will reply indicating whether it is configured or not

Reply options (without ID prefix):

N000B[CONFIGURED=YES]

N000B[CONFIGURED=NO]

N000B [MAC?]

Request the MAC address of the NEO controller

The controller will reply with its MAC address

Example reply (without ID prefix):

N000B[MAC=34:C9:08:ED:90:2A]

N000B [BLINK!]

Control LED C will start to blink continuously

It will stop blinking when the NEO device receives a new command

Network reset commands

N000B [NETWORKRESET!]

Reset all network and communication settings to factory default.

These settings include:

IP mode

Fixed IP address and Subnet mask

Destination IP

Incoming port and Outgoing port

Outgoing message prefix and Incoming command prefix

Startup message

Index prefix settings

Acknowledgement settings

Termination settings

N000B [FACTORYRESET!]

Reset all network and configuration settings to factory default

*Additionally to resetting all network and communication settings, this command will also reset the **controller ID**. After a factory reset, **the controller will report as "default" (not configured)**.*

After a Network reset or Factory reset, the controller will reboot and reconnect to your network. Depending on the network settings this process can take anywhere between 20 to 90 seconds.

The Factory reset described above will only factory-reset the settings and configurations listed in section 4. For a full Factory reset of the NEO device, please see section 5.6 "Resets".

4.6 Create start-up commands

The NEO controller can be configured via UDP to automatically execute specific commands at start-up, for example to adjust sensor settings. These commands can be recorded and stored, after which they will be executed after every power cycle. The available API commands are as follows:

Recording start-up commands

N000B [STARTRECORD!]

Start recording automated start-up commands
After this command is received, any valid command which is received is recorded and stored as an automated start-up command

N000B [STOPRECORD!]

Stop recording automated start-up commands

The following command types are available to be recorded as an automated start-up message:

XXXXXB [*...***]**

X-talk message (long) - to configure a sensor
Example command to configure an Activation Zone on a lidar sensor
X001B[ZONE01=-200,+100,100,100]

XXXXXA [*]

X-talk message (short) - to control outputs on Elements
Example command to set button LEDs to soft blink
X001A[170]

XXXXXS [* : *]

X-talk message (setting) - to adjust sensor settings
Example command to set a Presence sensor to Absolute distance mode
X001S[4:2]

PXXXXB [*...***]**

Power control message - to control power outputs
Example command to set power output 1 ON
P000B[OUTPUT1=ON]

NXXXXB [*...***]**

Network configuration message - to configure UDP
Example command to adjust the Destination IP
N000B[DESTIP=192.168.1.175]

SXXXXB [*...***]**

Settings message - to adjust a NEO controller setting
Example command to set the time
S000B[TIME=12.40.52-01/04/2025]

More examples are available in the Product manuals of the sensors and throughout this manual.

Verifying start-up commands

N000B [ECHORECORD!]

Echo all recorded commands
The controller will reply with a list of all recorded commands

N000B [SENDRECORD!]

Execute all recorded commands
The controller will execute all recorded commands

N000B [RECORDSTATE!]

Request the current state of the record memory
The controller will reply with the memory status

Clear start-up commands

N000B [CLEARRECORD!]

Clear all recorded commands
The controller will delete all stored start-up commands

5. Functionality and API commands

In this section, the core power management functionalities and API commands of the NEO controller are provided. These commands apply both for USB and UDP communication.

In case of **USB communication**, the commands and examples listed in this section are complete and can be copy/pasted during testing or development.

In case of **UDP communication**, the commands and examples listed in this section might not be complete. Depending on your UDP communication settings, the commands in this section might get additional prefixes.

- Per default, all messages that the NEO controller sends over UDP have a prefix with the controller ID (default=MAC).
Example in this section: `P000B[INPUTPOWER=1201.917]`
Example of UDP message: `FROMID=00:08:DC:75:79:8E:P000B[INPUTPOWER=1201.917]`
- In case the incoming command prefix setting is set to N000B[REQUIREDID=YES], all commands sent to the NEO controller must contain a prefix with the controller ID.
Example in this section: `P000B[OUTPUT1=ON]`
Example of UDP command: `TOID=00:08:DC:75:79:8E:P000B[OUTPUT1=ON]`
- In case the Index prefix is enabled, all messages from the NEO controller will contain a numbered prefix ID.
Example in this section: `P000B[INPUTPOWER=1201.917]`
Example of UDP message with Index and ID prefix: `#00001: FROMID=00:08:DC:75:79:8E: P000B[INPUTPOWER=1201.917]`
Example of UDP message with Index prefix only: `#00001:P000B[INPUTPOWER=1201.917]`

For detailed information on the UDP communication settings, please see section 4.4 "Adjust the NEO controller behaviour".

5.1 Measure power parameters

All NEO models can measure the real-time power parameters of each individual output. In this section, the commands to obtain the power parameters via the USB API interface are provided.

Voltage

P000B [INPUTVOLTAGE?] Request the real-time voltage level on the AC input

The reply from the NEO controller will have the following format

P000B[INPUTVOLTAGE=VVVVVVV] VVV.VVV=Voltage level **000.000 - 240.000** (Volts)

Example reply:

P000B[INPUTVOLTAGE=219.314]

The voltage level on the power outputs is the same as the input voltage level. Therefore, there is no "output voltage" request.

Output current

P000B [OUTPUTXCURRENT?] Request the real-time current level for an individual output

X = output nr 1 - 4

Example request for output 2:

P000B[OUTPUT2CURRENT?]

The reply from the NEO controller will have the following format

P000B[OUTPUTXCURRENT=CC.CCC] X = output nr 1 - 4 CC.CCC=Amount of current **00.000 - 16.000** (Ampere)

Example reply for output 2:

P000B[OUTPUT2CURRENT=03.314]

Output power

P000B [OUTPUTXPOWER?] Request the real-time power level for an individual output

X = output nr 1 - 4

Example request for output 3:

P000B[OUTPUT3POWER?]

The reply from the NEO controller will have the following format

P000B[OUTPUTXPOWER=PPPP.PPP] X = output nr 1 - 4 PPPP.PPP=Amount of power **0000.000 - 2300.000** (Watt)

Example reply for output 3:

P000B[OUTPUT3POWER=1201.917]

Output usage

P000B [OUTPUTXUSAGE?] Request the usage for an individual output

X = output nr 1 - 4

Example request for output 3:

P000B[OUTPUT3USAGE?]

The reply from the NEO controller will have the following format

P000B[OUTPUTXUSAGE=UUUUU.UUU] X = output nr 1 - 4 UUUUU.UUU= Usage **00000.000 - 99999.999** (kWh)

Example reply for output 3:

P000B[OUTPUT3USAGE=03410.917]

The usage value can be reset to 0 via the following API command

P000B [OUTPUTX=USAGERESET] Reset the usage for an individual output
X = output nr 1 - 4 or S for all outputs simultaneously

Example command to reset the usage of output 4 :

P000B[OUTPUT4=USAGERESET]

Example command to reset the usage of all outputs :

P000B[OUTPUTS=USAGERESET]

Output status

P000B [OUTPUTXSTATUS?]

Request the current status of a specific output

X = output nr 1 - 4

Example request for output 2:

P000B[OUTPUT2STATUS?]

The reply from the NEO controller will have the following format

P000B[OUTPUTX=*]**

X = output nr 1 - 4 *** = output status **ON** or **OFF**

Example reply:

P000B[OUTPUT2=ON]

Power input measurements

Next to measuring the power outputs, the parameters of the AC power input can also be requested:

Input voltage

P000B [INPUTVOLTAGE?]

Request the real-time voltage level on the AC input

The reply from the NEO controller will have the following format

P000B[INPUTVOLTAGE=VVV.VVV]

VVV.VVV=Voltage level **000.000 - 240.000** (Volts)

Example reply:

P000B[INPUTVOLTAGE=219.314]

Input current

P000B [INPUTCURRENT?]

Request the real-time current level of the AC input

The reply from the NEO controller will have the following format

P000B[INPUTCURRENT=CC.CCC]

CC.CCC=Amount of current **00.000 - 16.000** (Ampere)

Example reply:

P000B[INPUTCURRENT=007.404]

Input power

P000B [INPUTPOWER?]

Request the real-time power level of the AC input

The reply from the NEO controller will have the following format

P000B[INPUTPOWER=PPPP.PPP]

PPPP.PPP=Amount of power **0000.000 - 2300.000** (Watt)

Example reply:

P000B[INPUTPOWER=1201.917]

Input usage

P000B [INPUTUSAGE?]

Request the usage for the AC input

The reply from the NEO controller will have the following format

P000B[INPUTUSAGE=UUUUU.UUU]

UUUUU.UUU= Usage **00000.000 - 99999.999** (kWh)

Example reply:

P000B[INPUTUSAGE=05823.446]

The usage value can be reset to 0 via the following API command

P000B [INPUT=USAGERESET]

Reset the usage level of the AC input.

CAUTION: resetting the usage level of the input, also resets the usage level of all outputs and of the NEO device itself.

Input status

P000B [INPUTSTATUS?]

Request the current status of the AC power input

The reply from the NEO controller will have the following format

P000B[INPUT=*]**

***= input status **ON** or **OFF**

Example reply:

P000B[INPUT=ON]

The OFF reply can only be given by NEO320 and NEO340 controllers, as their internal ICs are powered via the USB-C input, and not via the AC input. All other controllers will not reply when there is no AC power input, as the controller itself will have no power and therefore will not be able to reply. The ON reply will be given on all NEO controllers

NEO device measurements

The power parameters of the NEO device itself can also be requested. In case sensors are connected to the X-talk interfaces of the NEO controller, their current, power and usage are added to the device power parameters.

Device current

P000B [DEVICECURRENT?]

Request the real-time current level of the NEO device

The reply from the NEO controller will have the following format

P000B[DEVICECURRENT=CC.CCC]

CC.CCC=Amount of current **00.000 - 16.000** (Ampere)

Example reply:

P000B[DEVICECURRENT=000.016]

Device power

P000B [DEVICEPOWER?]

Request the real-time power level of the NEO device

The reply from the NEO controller will have the following format

P000B[DEVICEPOWER=PPPP.PPP]

PPPP.PPP=Amount of power **0000.000 - 2300.000** (Watt)

Example reply:

P000B[DEVICEPOWER=0001.580]

Device usage

P000B [DEVICEUSAGE?]

Request the usage of the NEO device

The reply from the NEO controller will have the following format

P000B[DEVICEUSAGE=UUUUU.UUU]

UUUUU.UUU= Usage **00000.000 - 99999.999** (kWh)

Example reply:

P000B[DEVICEUSAGE=00002.351]

The usage value can be reset to 0 via the following API command

P000B [DEVICE=USAGERESET]

Reset the usage level of the NEO device

When high loads are connected to the output of the NEO device, it is not possible to accurately measure the device usage, and therefore the device usage reply parameter will be 0.

Autosend power measurements

Alternatively to requesting the real-time parameters of the power measurements, the NEO controller can also be configured to send data automatically at a set interval.

Configure autosend

P000B [AUTOSEND=OUTPUTX:*.***:####]**

Autosend specific measurement parameters

X = output nr 1 - 4 or S for all outputs simultaneously

. = power parameter **VOLTAGE, CURRENT, POWER, USAGE**
or **ALL**

= interval **0001 - 3600** seconds or **OFF**

Example configuration to autosend the real-time voltage level of output1 every 15 seconds:

P000B[AUTOSEND=OUTPUT1:VOLTAGE:0015]

Example configuration to autosend the real-time current level of output 2 every 30 seconds:

P000B[AUTOSEND=OUTPUT2:CURRENT:0030]

Example configuration to autosend the real-time power level of output 3 every 60 seconds:

P000B[AUTOSEND=OUTPUT3:POWER:0060]

Example configuration to autosend the usage of output 4 every hour (3600 seconds):

P000B[AUTOSEND=OUTPUT4:USAGE:3600]

Example configuration to autosend all parameters of all outputs every 15 minutes (900 seconds):

P000B[AUTOSEND=OUTPUTS:ALL:0900]

Example configuration to disable autosend of the usage for output 1

P000B[AUTOSEND=OUTPUT1:USAGE:OFF]

The API messages that the autosend configuration triggers are identical to the reply commands of the data requests listed in section 5.1 "Measure power parameters". The autosend configuration can also be applied on the power measurements of the AC input, as well as on the power measurements of the NEO device itself.

P000B [AUTOSEND=INPUT:*.***:####]**

Autosend specific measurement parameters of the AC input

. = power parameter **VOLTAGE, CURRENT, POWER, USAGE**
ALL

= interval **0001 - 3600** seconds or **OFF**

P000B [AUTOSEND=DEVICE:*.***:####]**

Autosend specific measurements parameter of the NEO device itself

. = power parameter **VOLTAGE, CURRENT, POWER, USAGE**
ALL or **OFF**

= interval **0001 - 3600** seconds or **OFF**

5.2 Control power outputs

The NEO 500 Series and NEO 600 series offer the option to switch the power outputs ON or OFF. In this section, the available commands for output control via the USB API interface are provided.

Switch power outputs - ON or OFF

P000B [OUTPUTX=ON]

Switch a power output ON

X = output nr 1 - 4 or S for all outputs simultaneously

Example command for output 1:

P000B[OUTPUT1=ON]

Example command for all outputs:

P000B[OUTPUTS=ON]

P000B [OUTPUTX=OFF]

Switch a power output OFF

X = output nr 1 - 4 or S for all outputs simultaneously

Example command for output 4:

P000B[OUTPUT4=OFF]

Example command for all outputs:

P000B[OUTPUTS=OFF]

When a power output is ON, the status LED of the corresponding output will be ON. When a power output is off, the status LED of the corresponding output will be OFF. **When an output does not have a device connected to it, the output needs to be switched OFF.**

Toggle outputs - switch to the opposite state

P000B [OUTPUTX=TOGGLE]

Toggle a power output

X = output nr 1 - 4 or S for all outputs simultaneously

Example command for output 3:

P000B[OUTPUT3=TOGGLE]

Example command for all outputs:

P000B[OUTPUTS=TOGGLE]

When a power output receives a toggle command, it will switch to the opposite state. When a power output is ON, a toggle command will switch the power output OFF. When a power output is OFF, a toggle command will switch the power output ON. After a toggle command, the new status of the output is indicated via an API message, for example **P000B[OUTPUT3=ON]**.

Cycle outputs - ON or OFF

P000B [OUTPUTX=CYCLEON]

Cycle an output to ON. The output will first go off.

X = output nr 1 - 4 or S for all outputs simultaneously

Example command for output 2:

P000B[OUTPUT2=CYCLEON]

Example command for all outputs:

P000B[OUTPUTS=CYCLEON]

P000B [OUTPUTX=CYCLEOFF]

Cycle an output to OFF. The output will first go on.

X = output nr 1 - 4 or S for all outputs simultaneously

Example command for output 1:

P000B[OUTPUT1=CYCLEOFF]

Example command for all outputs:

P000B[OUTPUTS=CYCLEOFF]

P000B [CYCLETIMEXX=##]

Set the cyletime for a specific output

XX = output nr 01 - 04 or AO for all outputs simultaneously

= cyletime 01 - 60 seconds. Default = 10 seconds.

Example to set output 1 to 10s:

P000B[CYCLETIME01=10]

Example to set all outputs to 20s:

P000B[CYCLETIMEAO=20]

The cycle time sets how long an output will be in the opposite state. For example, when the cycletime is 10 seconds, and a CYCLEON command is sent, the sequence is: Output switches OFF - waits 10 seconds - then switches ON.

Power control sequence

Next to individually switching the power outputs, the NEO controller can also execute a power sequence command. A power sequence command allows you to switch all 4 outputs ON or OFF one by one, in a timed sequence. This facilitates controlled booting of systems in which there is a particular order in which the components must be powered. For example, when a screen must be powered up before the media player. The API commands for power control sequences are:

P000B [STARTUPSEQ=T1 : T2 : T3 : T4]

Set the power sequence for switching outputs ON

T1 = time in 000-999 seconds to switch output 1 ON, or, leave it OFF. Def = 000

T2 = time in 000-999 seconds to switch output 2 ON, or, leave it OFF. Def = 000

T3 = time in 000-999 seconds to switch output 3 ON, or, leave it OFF. Def = 000

T4 = time in 000-999 seconds to switch output 4 ON, or, leave it OFF. Def = 000

Example command for a power sequence to switch output 1 on immediately, output 2 after 5 seconds, output 3 after 20 seconds, and leave output 4 OFF:

P000B[STARTUPSEQ=000:005:020:OFF]

P000B [SHUTDOWNSEQ=T1 : T2 : T3 : T4]

Set the power sequence for switching outputs OFF

T1 = time in 000-999 seconds to switch output 1. Default = 000

T2 = time in 000-999 seconds to switch output 2. Default = 000

T3 = time in 000-999 seconds to switch output 3. Default = 000

T4 = time in 000-999 seconds to switch output 4. Default = 000

Example command for a power sequence to switch output 1 off immediately, output 2 after 3 seconds, output after 60 seconds, and output 4 after 90s:

P000B[SHUTDOWNSEQ=000:003:060:090]

P000B [POWERONSEQ=T1 : T2 : T3 : T4]

Set the power sequence to automate the behavior of the power outputs when the NEO controller is powered on (AC input)

T1 = time in 000-999 seconds to switch output 1 ON, or, leave it OFF. Def = 000

T2 = time in 000-999 seconds to switch output 2 ON, or, leave it OFF. Def = 000

T3 = time in 000-999 seconds to switch output 3 ON, or, leave it OFF. Def = 000

T4 = time in 000-999 seconds to switch output 4 ON, or, leave it OFF. Def = 000

Example command for a power sequence to automatically switch output 1 on after 7 seconds of NEO powerup, output 2 after 11 seconds, output 3 after 40 seconds, and output 4 after 120 seconds

P000B[POWERONSEQ=007:011:040:120]

The timings listed above are all measured absolute from the moment the command is sent (or the NEO device is powered). For example, if for a STARTUP command T1 = 3 seconds, and T2 = 5 seconds, the difference between output 1 and output 2 powering on is 2 seconds.

To initiate a startup or shutdown sequence, use the following API commands:

P000B [STARTUP]

Initiate startup sequence

P000B [SHUTDOWN]

Initiate shutdown sequence

The STARTUP and SHUTDOWN sequence can also be initiated via the physical control buttons on the NEO device itself. For more information, please see the section 2.2 "Control buttons".

5.3 Scheduling

All NEO devices have an on-board Real Time Clock that allows you to schedule power control for specific times and days. In this section, the available commands for creating schedules via the USB API interface are provided.

Schedule power control

P000B [SCHED ## = ** . . ** : TIME : DAY]

Create a power schedule

= schedule nr 01 - 16

.. = operation 01ON, 02ON, 03ON, 04ON,

01OFF, 02OFF, 03OFF, 04OFF,

ALLON, ALLOFF, STARTUP, SHUTDOWN

Time = time 00.00 - 23.59

Day = day MON, TUE, WED, THU, FRI, SAT, SUN

ALL, WEE (weekdays), WND (weekends),

Example command for schedule 01 to switch all outputs ON at 08.00 on all days of the week:

P000B[SCHED01=ALLON:08.00:ALL]

Example command for schedule 02 to switch all outputs OFF at 18.00 on all days of the week:

P000B[SCHED02=ALLOFF:18.00:ALL]

Example command for schedule 03 to switch output 1 ON at 07.00 on weekdays (Mon-Fri):

P000B[SCHED03=01ON:07.00:WEE]

Example command for schedule 04 to switch output 2 OFF at 23.00 on weekends (Sat-Sun):

P000B[SCHED04=02OFF:23.00:WND]

Example command for schedule 15 to execute the startup sequence (see page 9) at 06.00 on Mondays:

P000B[SCHED15=STARTUP:06.00:MON]

Example command for schedule 16 to execute the shutdown sequence (see page 9) at 19.00 on Fridays:

P000B[SCHED16=SHUTDOWN:19.00:FRI]

After a schedule API command has been sent to the NEO controller, it will become active after the first full hour.

For example, a command send at 13.10, will become active after 14.00. In order for schedules to work accurately, please check whether the time currently set on the Real Time Clock is correct for your time zone (see section 5.5 "Device settings").

Manage power schedules

P000B [SCHED ## ?]

Request the currently set parameters for a specific schedule

= schedule nr 01 - 16

P000B [SCHED ## CLEAR]

Clear a specific schedule

= schedule nr 01 - 16

P000B [CLEARALLSCHEDULES]

Request the currently set parameters for all schedules

The reply to a schedule request is identical to the commands used to create the schedule. For example:

P000B[SCHED01=ALLON:0800:ALL]

5.4 Monitoring

All NEO devices have a range of smart monitoring features to track status and health of your system. Additionally, several safeguard functions can be used to increase the robustness of your installation. This section is divided into the following topics:

Watchdogs	page 23-24
Soft fuses	page 25
Min/Max warning	page 25
Diagnostic counters	page 26
Run time tracking	page 27
Last event logging	page 28

Watchdogs

The watchdog feature in NEO is a timer that toggles a specific power output when it expires. It can be reset by sending an API command from the connected device to the NEO controller. This feature is typically used to detect whether a connected device—such as a media player—has frozen, and if so, to automatically power-cycle the device.

P000B [WATCHDOGXTIMER=####]

Configure a watchdog timer

X = output nr 1 - 4

= timer duration 001 - 3600 seconds 0000 = disable watchdog (default)

Example command to configure the watchdog for output 1 to 360 seconds.

P000B [WATCHDOG1TIMER=0360]

P000B [WATCHDOGX]

Call a watchdog to reset its running timer

X = output nr 1 - 4

Example command to reset the timer of watchdog 2

P000B [WATCHDOG2]

When a watchdog is triggered, the corresponding output is toggled with the set cycletime of the output (see section 5.2). After the output is toggled, the watchdog will wait for 30 seconds before it is activated again. This gives the connected device time to reboot and start resetting the watchdog timer.

When a watchdog is activated, the following API message is sent:

P000B [WATCHDOGXSTARTED]

X = output nr 1 - 4

Example reply:

P000B [WATCHDOG1STARTED]

When a watchdog is triggered and the corresponding output is toggled, the following API message is sent:

P000B [WATCHDOGXTRIGGERED]

X = output nr 1 - 4

Example reply:

P000B [WATCHDOG1TRIGGERED]

To avoid an endless loop of power cycles being caused by a watchdog (in case a device does not recover), the number of times that a watchdog will attempt to recover the device can be configured.

P000B [WATCHDOG~~X~~MAXTRY=##]

Set the number of consecutive attempts for which a watchdog will trigger a power cycle

X = output nr 1 - 4

= number of attempts 01 - 10 00 = endless Default = 02

Example command to set the max attempts of the watchdog for output 4 to 3 times

P000B [WATCHDOG4MAXTRY=03]

When a watchdog is deactivated due to the maximum attempts being reached, the following API message is sent:

P000B [WATCHDOG~~X~~STOPPED]

X = output nr 1 - 4

Example reply:

P000B [WATCHDOG1STOPPED]

A watchdog can be re-activated by calling the watchdog via the **P000B [WATCHDOG~~X~~]** command.

The current set value of the watchdog timer, as well as the current status of the watchdog can be requested at any time via the following API commands:

P000B [WATCHDOG~~X~~TIMER?]]

Request the current value of a watchdog timer

X = output nr 1 - 4

The reply from the NEO controller will have the following format

P000B [WATCHDOG~~X~~TIMER=####]

X = output nr 1 - 4

= timer duration 001 - 3600 seconds

Example reply:

P000B [WATCHDOG1TIMER=0060]

P000B [WATCHDOG~~X~~STATUS?]]

Request the current status of a watchdog

X = output nr 1 - 4

The reply from the NEO controller will have the following format

P000B [WATCHDOG~~X~~STATUS=*]**

X = output nr 1 - 4

**** = status ON, OFF or MAXTRY*

Example reply:

P000B [WATCHDOG1STATUS=ON]

Soft fuse

For each power output, a soft fuse can be set. When the measured current of an output exceeds the soft fuse value, the corresponding power output will automatically switch off.

P000B [OUTPUTXSOFTFUSE=CC.CCC]

Set the trigger level of the soft fuse

X = output nr 1 - 4

CC.CCC = Amount of current **00.000 - 09.999** (Ampere) Default =09.500

Example command to set the soft fuse for output 1 to 2.5 Amp.

P000B[OUTPUT1SOFTFUSE=02.500]

When a soft fuse is triggered, the NEO device automatically sends the following API message:

P000B[OUTPUTXSOFTFUSE]

X = output nr 1 - 4

Example reply:

P000B[OUTPUT1SOFTFUSE]

Min and Max warnings

In addition to the soft fuse, the NEO device can also be configured to send warning messages when a power output goes above or below a certain level. These levels can be set via the following API commands:

P000B [OUTPUTXMINWARNING=PPPP.PPP]

Set the trigger level for a minimum power warning

X = output nr 1 - 4

PPPP.PPP=Amount of power **0000.000 - 2300.000** (Watt) Default =0000.000

Example command to set the min warning level for output 2 to 14 Watt.

P000B[OUTPUT2MINWARNING=0014.000]

P000B [OUTPUTXMAXWARNING=PPPP.PPP]

Set the trigger level for a maximum power warning

X = output nr 1 - 4

PPPP.PPP=Amount of power **0000.000 - 2300.000** (Watt) Default =2300.000

Example command to set the max warning level for output 3 to 450 Watt.

P000B[OUTPUT3MAXWARNING=0450.000]

When the power level of an output goes below the set minimum power level, the following API message is sent:

P000B[OUTPUTXMINWARNING]

X = output nr 1 - 4

Example reply:

P000B[OUTPUT2MINWARNING]

When the power level of an output goes above the set maximum power level, the following API message is sent:

P000B[OUTPUTXMAXWARNING]

X = output nr 1 - 4

Example reply:

P000B[OUTPUT3MAXWARNING]

Diagnostic counters

The NEO controller has several automated counters to track the number of times that specific power-related events occurred. This data can for example be used to analyze unexpected scenarios and find the root cause of malfunctioning or broken devices in your setup.

P000B [OUTPUTXSWITCHEDCOUNT?]

Request the number of times an output was switched from OFF to ON

X = output nr 1 - 4

Example command for output 1

P000B[OUTPUT1SWITCHEDCOUNT?]

P000B [OUTPUTXSWITCHEDOFFUNDERLOAD?]

Request the number of times an output was switched off under load (>5W), typically meaning the connected device was powered on, and not switched off

X = output nr 1 - 4

Example command for output 2

P000B[OUTPUT2SWITCHEDOFFUNDERLOAD?]

P000B [REGULARPOWEROFF?]

Request the number of times the **input** power of the NEO device was cut off when all **outputs** were switched off

P000B [IRREGULARPOWEROFF?]

Request the number of times the **input** power of the NEO device was cut off under load (>5W), typically meaning a connected device was powered on.

The replies of the NEO device will be as follows:

P000B[OUTPUTXSWITCHEDCOUNT=#####]

X = output nr 1 - 4 ##### = count nr 0 - 999999

P000B[OUTPUTXSWITCHEDUNDERLOAD=#####]

P000B[REGULARPOWEROFF=#####]

Example reply for the number of times output 2 was switched on
P000B[OUTPUT2SWITCHEDCOUNT=000012]

P000B[IRREGULARPOWEROFF=#####]

The counters can be reset to 0 by sending the following API commands to the NEO device:

P000B [OUTPUTXSWITCHEDCOUNT=RESET]

Reset the number of times an output was switched from OFF to ON AND reset the number of times an output was switched off under load

X = output nr 1 - 4

Example command for output 3

P000B[OUTPUT3SWITCHEDCOUNT=RESET]

P000B [REGULARPOWEROFF=RESET]

Reset the number of times the **input** power of the NEO device was cut off when all **outputs** were switched off

P000B [IRREGULARPOWEROFF=RESET]

Reset the number of times the **input** power of the NEO device was cut off when there was still at least one **output** switched on

Run time tracking

The NEO controller tracks the run time of the NEO device itself, as well as each individual output. This data can for example be used to schedule maintenance or check on warranty and lifetime requirements of the connected devices.

P000B [OUTPUTXRUNTIME?]

Request how long an output currently has been switched on consecutively

X = output nr 1 - 4

Example command for output 4

P000B[OUTPUT4RUNTIME?]

P000B [RUNTIME?]

Request how long the NEO device has been powered on consecutively

The reply of the NEO device to both requests will be as follows:

P000B[RUNTIME=####D-&&H]

= nr of days 0 - 9999 && = nr of hours 0 - 23

Operation time tracking

The NEO controller tracks the operation time of the NEO device itself, as well as each individual output.

P000B [OUTPUTXOPERATIONTIME?]

Request how long an output has been switched on cumulatively

X = output nr 1 - 4

Example command for output 1

P000B[OUTPUT1OPERATIONTIME?]

P000B [OPERATIONTIME?]

Request how long the NEO device has been switched on cumulatively

The reply of the NEO device to both requests will be as follows:

P000B[OPERATIONTIME=####D-&&H]

= nr of days 0 - 9999 && = nr of hours 0 - 24

Last event logging

The NEO controller tracks and stores the date and time of specific power-related events. These can be used for logging purposes but also to analyze unexpected scenarios.

P000B [OUTPUTXLASTTIMESWITCHEDON?]

Request the last time an output was switched on since the NEO device was last powered

X = output nr 1 - 4

Example command for output 4

P000B[OUTPUT4LASTTIMESWITCHEDON?]

P000B [OUTPUTXLASTTIMESWITCHEDOFF?]

Request the last time an output was switched off since the NEO device was last powered

X = output nr 1 - 4

Example command for output 2

P000B[OUTPUT2LASTTIMESWITCHEDOFF?]

P000B [LASTTIMEPOWERON?]

Request the last time the NEO device was powered on

P000B [LASTTIMEPOWEROFF?]

Request the last time the NEO device was powered off

The reply of the NEO device will be as follows:

P000B[TIME=HH.MM.SS-DD/MM/YYYY]

X = output nr 1 - 4

HH = hours 00 - 23 MM = minutes 00 - 59 SS = seconds 00 - 59

DD = day 01 - 31 MM = month 01 - 12 YYYY = year 0000 - 9999

Example reply:

P000B[TIME=21.30.22-14/02/2025]

When a NEO device loses power on its AC power input, it will automatically send the following API message:

P000B[VOLTAGELOST!]

The NEO device has lost power on AC input

5.5 Device settings

In this section, the available commands for adjusting and accessing general device settings via the USB or UDP API interface are provided.

Set time

S000B [TIME=HH.MM.SS-DD/MM/YYYY]

Set the time for the Real Time Clock

HH = hours 00 - 23 MM = minutes 00 - 59 SS = seconds 00 - 59

DD = day 01 - 31 MM = month 01 - 12 YYYY = year 0000 - 9999

Example command to set the time to 12:40:52

and the date to 01-04-2025 is:

S000B[TIME=12.40.52-01/04/2025]

The Real Time Clock automatically calculates the weekday (Mon - Sun) based on the given date. When the NEO controller is unplugged from the power input, the time and date will stay accurate for ~3 months. The current time can be requested at any given moment via the following command:

S000B [TIME?]

Request the current time

The reply to a time request is identical to the commands used to set the time. For example:

S000B[TIME=12.40.52-01/04/2025]

The model name of the NEO device can be requested by sending the following API command:

S000B [MODEL?]

Request the model name of the NEO device

The reply from the NEO controller will have the following format

S000B[MODEL=***]**

***** = model name *NEO640, NEO620, NEO540, NEO520, NEO340 or NEO320*

Example reply for NEO-640:

S000B[MODEL=NEO640]

The current firmware version of the NEO device can be requested by sending the following API command:

S000B [FWVERSION?]

Request the current firmware version

The reply from the NEO controller will have the following format

S000B[FWVERSION=##.##.##]

##.##.## = firmware version *00.00.00 - 99.99.99*

Example reply:

S000B[FWVERSION=01.01.01]

5.6 Resets

In this section, the available commands for resetting the NEO controller are provided

Factory reset

S000B [FACTORYRESET!]

All settings and configuration are set to factory default

This includes all NEO power settings (e.g. schedules, power sequences), UDP communication settings and Sensmi configurations. A Factory reset can also be initiated by pressing button A+B for 20 seconds. After successfully completing the Factory reset, the NEO device will reply as follows:

S000B[FACTORYRESET]

If the UDP communication settings on your control device (e.g. PC or player) do not match the default UDP settings of the NEO controller, you will not receive the confirmation message via UDP.

Controller reset

S000B [CONTROLLERRESET!]

Reset all NEO power settings and configurations

This includes all NEO power settings (e.g. schedules, power sequences), but excludes UDP communication settings and Sensmi configurations. After successfully completing the Controller reset, the NEO device will reply as follows:

S000B[CONTROLLERRESET]

UDP configuration reset

N000B [NETWORKRESET!]

Reset network and communication settings to factory default.
This resets all UDP configurations, except for the Controller ID.

N000B [FACTORYRESET!]

Reset all network and communication settings to factory default.
This resets all UDP configurations, including the Controller ID.

None of the NEO power settings (e.g. schedules, power sequences), nor the Sensmi configuration, will be affected.

Sensmi provisioning reset

SENSMI [RESETCONFIG!]

Reset the Sensmi credentials provisioned on this controller.

None of the NEO power settings (e.g. schedules, power sequences), nor the UDP communication, will be affected.

After successfully completing the Sensmi configuration reset, the NEO device will reply as follows:

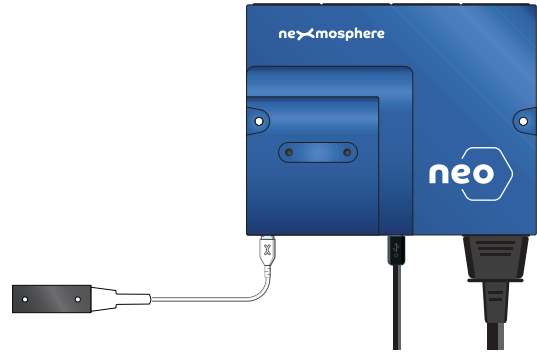
SENSMI[CONFIG RESET]

5.7 X-talk sensors

Sensor output

Nexmosphere controllers and sensors by default work trigger-based, meaning that an API message is automatically sent when a sensor detects a new event. The API messages indicating the sensor status are identical on all Nexmosphere controllers, including the NEO controllers. This means that the output of each sensor is exactly as indicated in the sensor Product Manuals (available on nexmosphere.com/support), is sent via the USB Serial API, or UDP API.

API message of a **presence sensor** connected to **X-talk channel 004** that detected a person in **distance zone 3**
X004B[Dz=03]

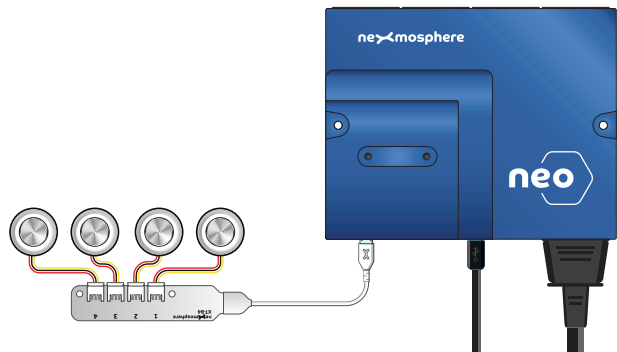


Control input

Nexmosphere offers various Elements of which the output can be controlled, for example button LEDs and I/O interfaces. Next to this, some sensors require configuration, such as the lidar sensor. This is done by sending control commands via the USB serial API or UDP API to the NEO controller.

The API commands are identical on all Nexmosphere controllers, including the NEO controllers. This means that the commands to control an output or to configure a sensor are exactly as indicated in the sensor Product Manuals (available on our support page) or Controller Quick Start Guide.

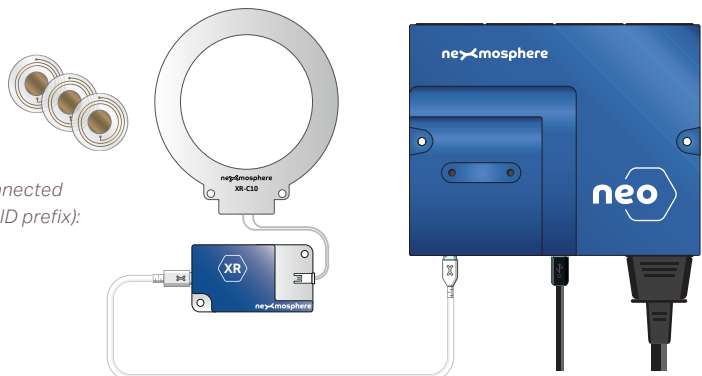
API command to set all **button LEDs** connected to **X-talk channel 003** to a **slow blinking output**
X003A[170]



Sensor settings

All Nexmosphere Elements have settings to adjust the behaviour and output of the sensors. This is done by sending Setting commands via the USB serial API to the NEO controller. The API commands for this are identical on all Nexmosphere controllers, including the NEO controllers. This means that the commands to adjust the settings of a sensor are exactly as indicated in the sensor Product Manuals (available on our support page).

API command to increase the gain **setting** of a RFID Driver connected to **X-talk channel 003** (Default controller settings, without TOID prefix):
X003S[4-4]



Power control of X-talk

The NEO 500 series and NEO 600 series offer the option to switch the power of the X-talk interfaces ON or OFF.

P000B [XTALK=ON]	Switch all X-talk interfaces ON
P000B [XTALK=OFF]	Switch all X-talk interfaces OFF
P000B [XTALK=TOGGLE]	Toggle all X-talk interfaces
P000B [XTALK=CYCLEON]	Cycle all X-talk interfaces to ON. They will first go off.
P000B [XTALK=CYCLEOFF]	Cycle all X-talk interfaces to OFF. They will first go ON.
P000B [CYCLETIMEXT=##]	Set the cycletime for all X-talk interfaces <i>## = cycletime 01 - 60 seconds. Default = 05 seconds.</i> Example to set power cycle time for the X-talk interfaces to 8 seconds: P000B[CYCLETIMEXT=08]

When the X-talk interfaces receive a toggle command, they will switch to the opposite state. When they are ON, a toggle command will switch the X-talk interfaces OFF. When they are OFF, a toggle command will switch the X-talk interfaces ON.

The cycle time sets how long X-talk interfaces will be in the opposite state. For example, when the cycletime is 8 seconds, and a CYCLEON command is sent, the sequence is: Output switches OFF - waits 08 seconds - then switches ON.

When the X-talk interfaces are switched off, the entire X-talk section of NEO is cut from power, including the 2 X-talk status LEDs.

6. Sensmi configuration

All NEO 600 series controllers have an on-board UTP socket to connect the controller to your network via a UTP cable. When connected, pre-built online dashboards, powered by Sensmi, provide real-time power measurement data and power controls. This can be used in parallel with USB and UDP API communication.

6.1 Enable or Disable Sensmi connection

Per factory default, the Sensmi connection is enabled. To disable the Sensmi connection, the following API command can be sent via USB or UDP:

S000B [SENSMI=DISABLE]	Disable Sensmi connection
---------------------------------	---------------------------

The reply from the NEO controller will be as follows:

S000B[SENSMI=DISABLED]

Once the Sensmi connection is disabled, the NEO unit will no longer send any data to the Sensmi cloud, nor will it try to make a connection with it of any kind.

Disabling the Sensmi connection only needs to be done once. It will stay disabled also after power-cycling the device.

The Sensmi connection can be enabled at any time. To enable the Sensmi connection, the following API command can be sent either via UDP or USB:

S000B [SENSMI=ENABLE]	Enable Sensmi connection
--------------------------------	--------------------------

The reply from the NEO controller will be as follows:

S000B[SENSMI=ENABLED]

After the Sensmi connection is enabled, the NEO unit needs to be provisioned with the Sensmi credentials. This is explained in section 6.2 "Provisioning of Sensmi credentials".

Enabling the Sensmi connection only needs to be done once. It will stay enabled also after power-cycling the device.

6.2 Provisioning of Sensmi credentials

The NEO controller must be provisioned with your Sensmi credentials to assign it to your Sensmi account and access the online dashboard. In this section, the API commands for provisioning the controller are provided.

If you do not yet have Sensmi credentials, please send an e-mail to support@sensmi.eu to request customer registration.

The only required details are your **Company name** and registered **email address**.

You will receive a confirmation e-mail with your **CUID** (Customer ID)

Required provisioning commands, in order:

SENSMI [PROV=ON]	Enter provisioning mode <i>Sensmi reply:</i> SENSMI[ENTERING PROVISIONING]	<i>Sensmi reply when device is already in provisioning mode:</i> SENSMI[ALREADY IN PROVISIONING]
SENSMI [DEVICENAME=XXX . . .XXX]	Enter the <i>unique</i> name under which you want the NEO device to appear in your Sensmi account and dashboard <i>XXX...XXX = device name 32 characters max</i> <i>Example command to set the device name to "SETUP ABC 123"</i> SENSMI[DEVICENAME=SETUP ABC 123] <i>Sensmi reply:</i> SENSMI[SETTING RECEIVED]	
SENSMI [CUID=XXX . . .XXX]	Enter your <i>fixed</i> Sensmi customer ID <i>XXX...XXX = device name 32 characters max</i> <i>Example command in case your Sensmi CUID is "NEXMOSPHERE"</i> SENSMI[CUID=NEXMOSPHERE] <i>Sensmi reply:</i> SENSMI[SETTING RECEIVED]	
SENSMI [PROV=SAVE]	Save the entered provisioning data <i>Sensmi reply:</i> SENSMI[SETTING RECEIVED]	
SENSMI [PROV=OFF]	Exit provisioning mode <i>Sensmi reply:</i> SENSMI[EXITED PROVISIONING]	

Sensmi Provisioning mode can also be activated by **pressing button A+B for 1 second**. For more information, please see the section 2.2 "Control buttons".

Optional provisioning commands, to be sent after entering the Customer ID are:

SENSMI [COUNTRY=XXX . . .XXX]	Enter the country in which the NEO device is installed <i>Example command in case the country is "SPAIN"</i> SENSMI[COUNTRY=SPAIN] <i>Sensmi reply:</i> SENSMI[SETTING RECEIVED]	
SENSMI [AREA=XXX . . .XXX]	Enter the area in which the NEO device is installed <i>Example command in case the area is "CATALUNYA"</i> SENSMI[AREA=CATALUNYA] <i>Sensmi reply:</i> SENSMI[SETTING RECEIVED]	
SENSMI [CITY=XXX . . .XXX]	Enter the city in which the NEO device is installed <i>Example command in case the city is "BARCELONA"</i> SENSMI[CITY=BARCELONA] <i>Sensmi reply:</i> SENSMI[SETTING RECEIVED]	

6.3 Dashboard

To view your NEO dashboard, log in to the Sensmi portal with your credentials. <https://portal.sensmi.eu/>

If it is the first time you are using your NEO with Sensmi, **install the NEO Dashboard**:

1. Click on Dashboard marketplace (left menu)
2. Search for NEO
3. Click install
4. Name the dashboard

After the NEO unit is provisioned and connected to Sensmi, **enable data transmission**.

This must be done for each new NEO unit:

1. Click on Device manager (left menu)
2. Click on NEO
3. Search for your NEO unit (device name used in provisioning)
4. Select your desired data interval (e.g. 5 minutes)
5. Click on Save

To **access the Dashboard**:

1. Click on Dashboard (left menu)
2. Select your NEO dashboard

