Table of content

1. General

NEO power controllers can measure and control it's onboard power outputs. Typically this is used to measure the power consumption of each component in an installation, and, reduce the energy consumption. This document offers explanation and instructions on the available functionalities of the **NEO Evaluation samples**. The final production release will contain more functionalities.

The information in this document is created for users who are familiar with the Nexmosphere API and are able to control a basic setup with a Nexmosphere API controller. If this is not the case yet, please read the general documentation on the Nexmosphere serial API first.

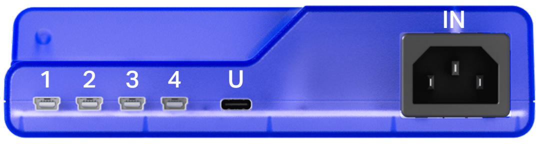
2. Product overview

	NEO **320**	NEO **520**	NEO **620**	NEO **340**	NEO **540**	NEO **640**
Outputs *individually controllable*	**2** outputs **C13**			**4** outputs **C13**		
Measure *parameters*	**V** voltage **A** current **W** power **KwH** usage			**V** voltage **A** current **W** power **KwH** usage		
Control *output switching*	-	✔	✔	-	✔	✔
USB *API interface*	✔	✔	✔	✔	✔	✔
Network *connectivity*	-	-	✔	-	-	✔
Cloud *dashboard*	-	-	✔	-	-	✔
X-talk *channels*	**4** X-talk channels for sensor interfacing			**4** X-talk channels for sensor interfacing		
Input *voltage range*	**100** - **230** VAC			**100** - **230** VAC		
Power *max rating*	**10A** CE **15A** UL			**10A** CE **15A** UL		

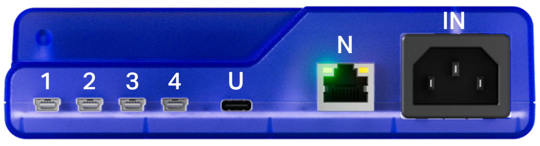
2.1 - Hardware interfaces

FRONT

NEO320 | NEO340 | NEO520 | NEO540

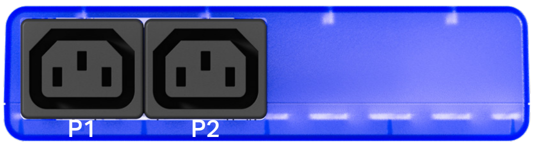


NEO620 | NEO640

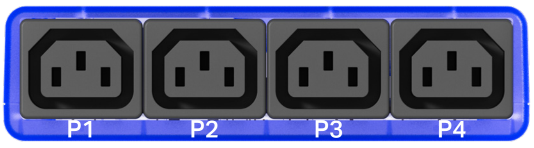


BACK

NEO320 | NEO520 | NEO620



NEO340 | NEO540 | NEO640

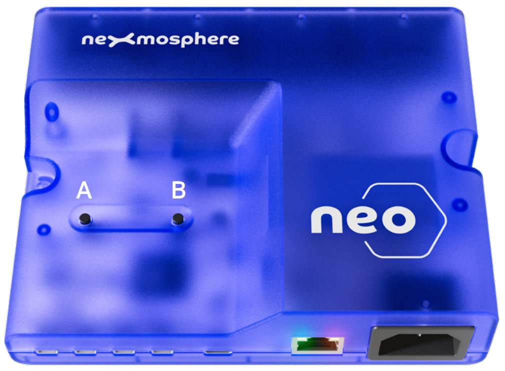


1	X-talk interface **001**
2	X-talk interface **002**
3	X-talk interface **003**
4	X-talk interface **004**
U	USB-C **API interface**
N	RJ45 **Network connector**
IN	C14 **Power input**

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

The NEO540 Evaluation Sample will have the hardware interfaces of a NEO640. However the RJ45 network connector is non-functioning, effectively making it a NEO540.

2.2 - Control buttons



A	Control **button A**
B	Control **button B**

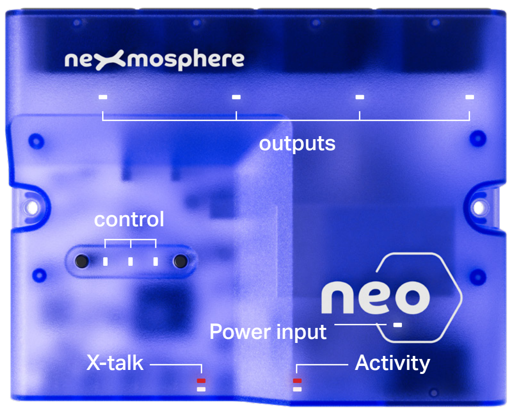
In the Evaluation Samples, the control buttons are disabled.

In the final production models, these buttons will offer dedicated functionalities such as Manual power control, Network provisioning and Settings adjustments.

Nexmosphere
Le Havre 136
5627 SW Eindhoven • The Netherlands

T +31 40 240 7070
E support@nexmosphere.com

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 .

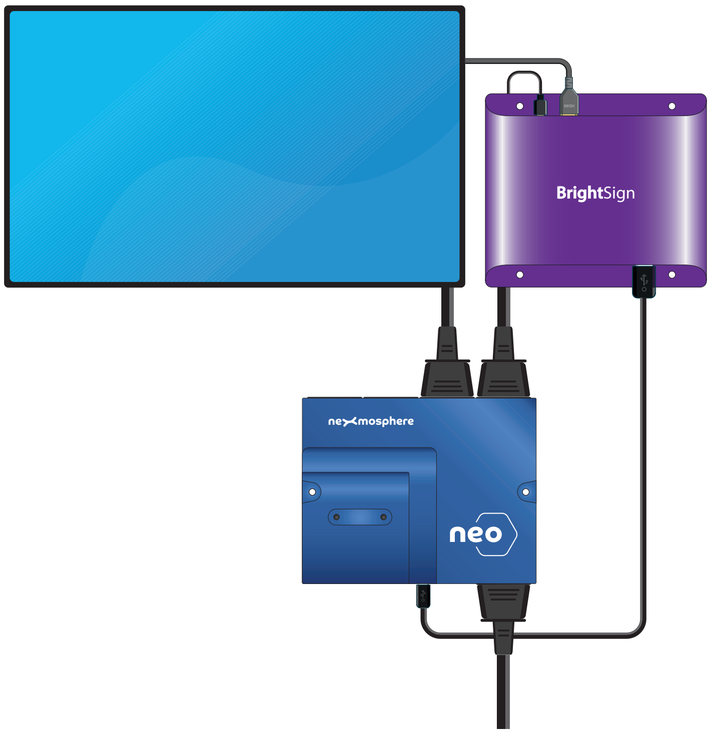
Power input
When the AC power input is above 80V, the status LED will be on. If the AC power input is below 80V, the status LED will be off.

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 blinks when a invalid X-talk command is received.

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

Control
The control LEDs are disabled (continuous on) in the engineering samples. In the final production models, these status LEDs will offer dedicated feedback for functionalities such as Manual power control, Network provisioning and Settings adjustments.

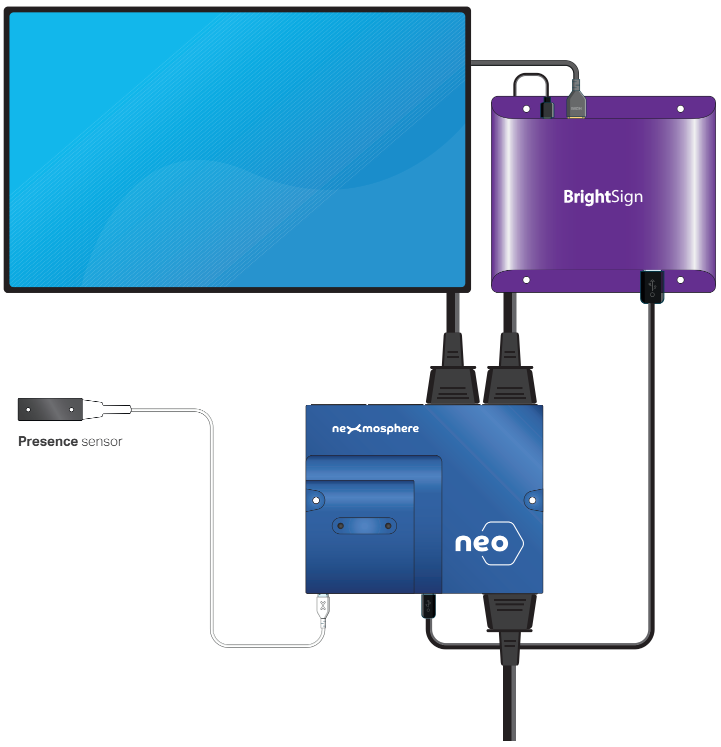
2.4 - Typical setups



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.

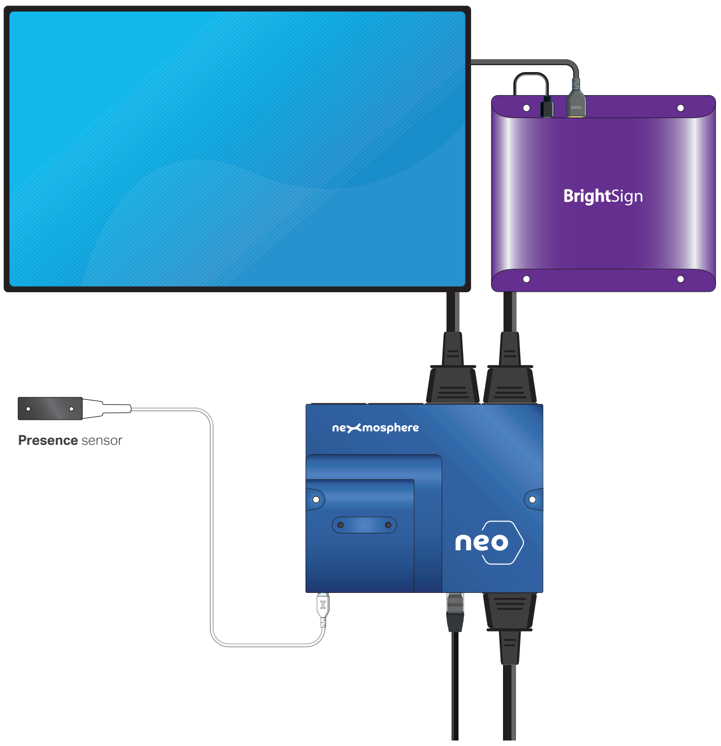
Both the digital signage player and the screen are powered via the NEO controller.



Example 2: USB-C connection, with sensor
This example is similar to Example 1. In this case, a presence sensor is added of which the output will also be send 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 use 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. NEO6xx models can connect to your network via its UTP connector. When connected, pre-built online dashboards powered by Sensmi provide real time power measurement data and power controls.

Nexmosphere
Le Havre 136
5627 SW Eindhoven • The Netherlands

T +31 40 240 7070
E support@nexmosphere.com

3.1 USB Serial settings

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 the following page, under the tab resources https://prolificusa.com/product/pl2303gl-8-pin-usb-uart-bridge-controller/.

The virtual com port facilitates serial communication between the NEO controller and the connected device. Each USB API command explained in this manual is an ASCII command send 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 controller is now ready for use.**

**In case the NEO controller is not recognized as a COM port by the 3rd party device, a driver (Prolific PL2303) can be downloaded here.*

3.2 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=VVV.VVV**]**	*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[OUTPUTX**CURRENT=**CC.CCC**]**	*X = output nr* ***1 - 4*** *CC.CCC=Amount of current* ***00.000 - 16.000*** *(Ampere)* ***Example reply for output 2:*** ***P000B[OUTPUT2CURRENT=003.314]***

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[OUTPUTX**POWER=**PPPP.PPP**]**	*X = output nr* ***1 - 4*** *PPPP.PPP=Amount of current* ***0000.000 - 2300.000*** *(Watt)* ***Example reply for output 3:*** ***P000B[OUTPUT3POWER=1201.917]***

Output usage

`P000B[OUTPUTXUSAGE?]`	Request the real-time usage 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[OUTPUTX**USAGE=**UUUUU.UUU**]**	*X = output nr* ***1 - 4*** *UUUUU.UUU= Usage* ***00000.000 - 99999.999*** *(kWh)* ***Example reply for output 1:*** ***P000B[OUTPUT3USAGE=03410.917]***

In the Evaluation samples, the Usage is cumulative, and reset every time the NEO controller is repowered. In the final production version, the Usage can be set to cumulative or interval-based. It will be possible to manually reset the usage value.

Output status

`P000B[OUTPUTXSTATUS?]`	Request the current status of a specific output *X = output nr* ***1 - 4*** ***Example request for output 2:*** *P000B[OUTPUT1STATUS?*

The reply from the NEO controller will have the following format

P000B[OUTPUTX**STATUS=***]**	*X = output nr* ***1 - 4*** ****= output status* ***ON*** *or* ***OFF*** ***Example reply:*** ***P000B[OUTPUT2STATUS=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 for an individual output

The reply from the NEO controller will have the following format

P000B[INPUTPOWER=PPPP.PPP**]**	*PPPP.PPP=Amount of current* ***0000.000 - 2300.000*** *(Watt)* ***Example reply:*** ***P000B[INPUTPOWER=1201.917]***

Input usage

`P000B[INPUTUSAGE?]`	Request the real-time 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]***

In the Evaluation samples, the Usage is cumulative, and reset every time the NEO controller is repowered. In the final production version, the Usage can be set to cumulative or interval-based. It will be possible to manually reset the usage value.

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 measurements 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 on page 6 and 7. The autosend configuration can also be applied on the power measurements of the AC input:

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

Autosend specific measurements parameters

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

*#### = interval **0001 - 3600** seconds*

In the Evaluation samples, the Usage is cumulative, and reset every time the NEO controller is repowered. In the final production version, the Usage can be set to cumulative or interval-based. It will be possible to manually reset the usage value.

3.3 Control power outputs

The NEO 5xx Series and NEO 6xx 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:	***Example command for all outputs:***
`P000B[OUTPUT1=ON]`	`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:	***Example command for all outputs:***
`P000B[OUTPUT4=OFF]`	`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.

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:	***Example command for all outputs:***
`P000B[OUTPUT3=TOGGLE]`	`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.

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:	***Example command for all outputs:***
`P000B[OUTPUT2=CYCLEON]`	`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:	***Example command for all outputs:***
`P000B[OUTPUT1=CYCLEOFF]`	`P000B[OUTPUTS=CYCLEOFF]`

`P000B[CYCLETIMEXX=##]`

Set the cycletime for a specific output

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

*## = cycletime **01** - **60** seconds. Default = 05 seconds.*

Example to set output 1 to 10s:	***Example to set all outputs to 20s:***
`P000B[CYCLETIME01=10]`	`P000B[CYCLETIMEAO=20]`

The cyletime sets how long an output will be in the opposite state. For example, when the cycletime is 10 seconds, and an CYCLEON command is send, the sequence is: Output switches OFF - waits 20 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 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 mediaplayer. 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 them OFF. Def = 000*

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

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

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

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

`P000B[STARTUP=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 immediatly, output 2 after 3 seconds, output after 60 seconds, and output 4 after 90s:

`P000B[SHUTDOWN=000:003:060:90]`

`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 them OFF. Def = 000*

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

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

*T4 = time in **000-999** seconds to switch output 4 ON, or, leave them 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[POWERON=007:011:040:120]`

The timings listed above are all measured absolute from the moment the command is send (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

In the final production models, more advanced functionalities to create smart sequences for automation will be available,

3.4 Scheduling

All NEO devices have an on-board Real Time Clock that allows 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]
```

In order for schedules to work accurately, please check if the time currently set on the Real Time Clock is correct for your timezone (see page 12).

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]

Nexmosphere
Le Havre 136
5627 SW Eindhoven • The Netherlands
T +31 40 240 7070
E support@nexmosphere.com

3.5 Device settings

In this section, the available commands for adjusting general device settings via the USB API interface are provided. In the final production models, more settings (e.g. for automated power/error monitoring) will be available.

Set time

`P000B[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 to12:40:52 and the date to 01-04-2025 is:*** *P000B[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:

`P000B[TIME?]`	Request the current time

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

P000B[TIME=12:40:52-01/04/2025]

3.4 X-talk sensors

Sensor output

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

*API message of a **presence sensor** connected to **X-talk channel 001** that detected a person in **distance zone 3***
X001B[Dz=03]

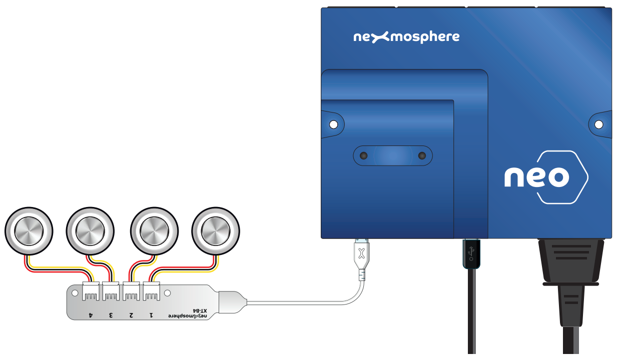
nexmosphere
neo

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 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 the configure a sensor is 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 002** to a slow blinking output*
X002A[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).

nexmosphere
XR-C10
XR
nexmosphere
neo

*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]

Nexmosphere
Le Havre 136
5627 SW Eindhoven • The Netherlands

T +31 40 240 7070
E support@nexmosphere.com

4. Network connectivity and Sensmi Dashboard

NEO6xx models can connect to your network via its UTP connector. When connected, pre-built online dashboards, powered by Sensmi, provide real time power measurement data and power controls.

4.1 Provisioning of controller

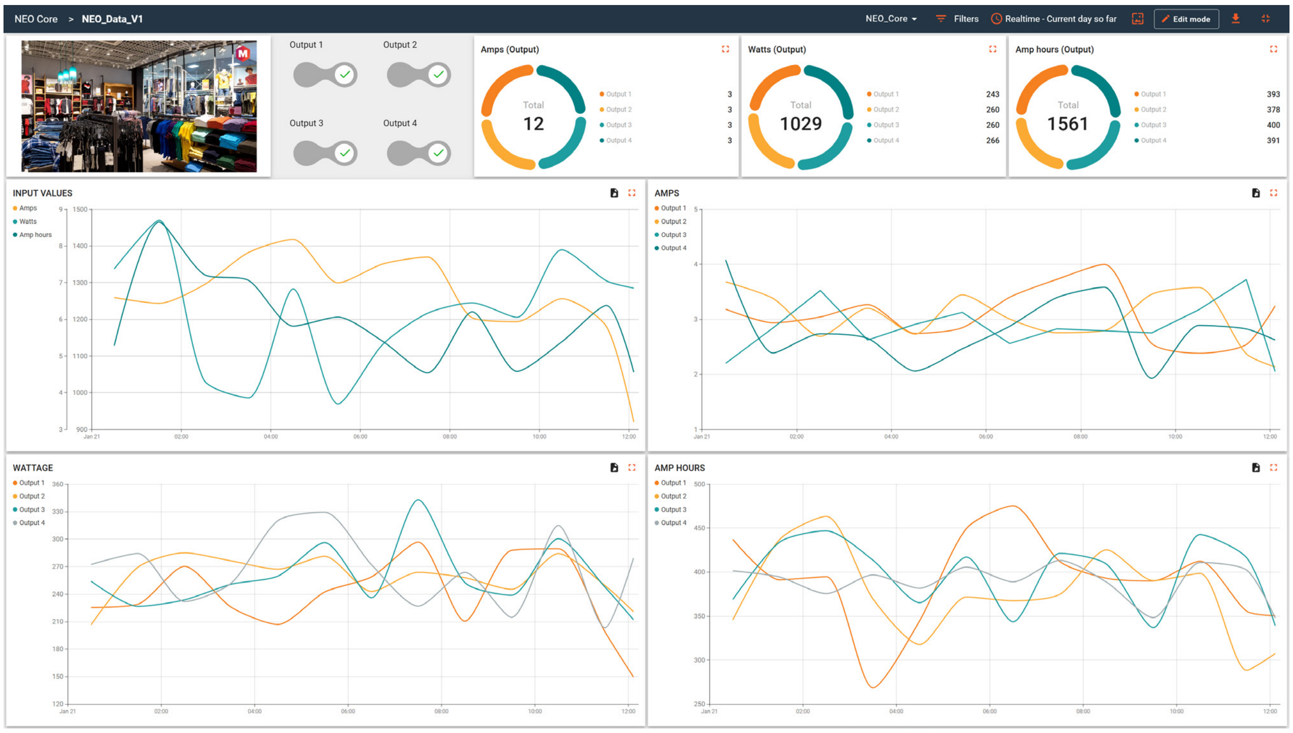
Evaluation samples come pre-provisioned, meaning that once they are connected to the internet, they will be visible on your Sensmi dashboard.

In the final production models, various provisioning methods will be available to provision your NEO controller and assign it to your Sensmi account.

4.2 Dashboard

To view your NEO dashboard, log-in to the Sensmi portal with the credentials that are provided to you by Nexmosphere. https://portal.sensmi.eu/

Once logged in, a dashboard similar to the screenshot below will be visible, to see real-time power consumption and remotely switch the power outputs ON or OFF.



Amps (Output)		Watts (Output)		Amp hours (Output)	
Output 1	3	Output 1	243	Output 1	393
Output 2	3	Output 2	260	Output 2	378
Output 3	3	Output 3	260	Output 3	400
Output 4	3	Output 4	266	Output 4	391