## Table of content



## 1. General

Nexmosphere UDP controllers can send and receive API messages over a local network via UDP commands. This document offers explanation and instructions on the available functionalities and settings for the UDP communication. For model-specific functionalities, please see the Quick Start Guide of your controller.

*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

UDP for API communication is available on the following controllers:

**XC-101U(P) / XC-102U**
X-talk interfaces

**XC-741U**
X-talk & LightMark

**XC-841U**
X-talk & RGBW LED

**XC-931U / XC-941U**
X-talk & Audio switches

All UDP 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 sensors to be send 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 controller via UDP commands, for example to adjust settings or control LEDs. The factory default Network/UDP settings are:

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



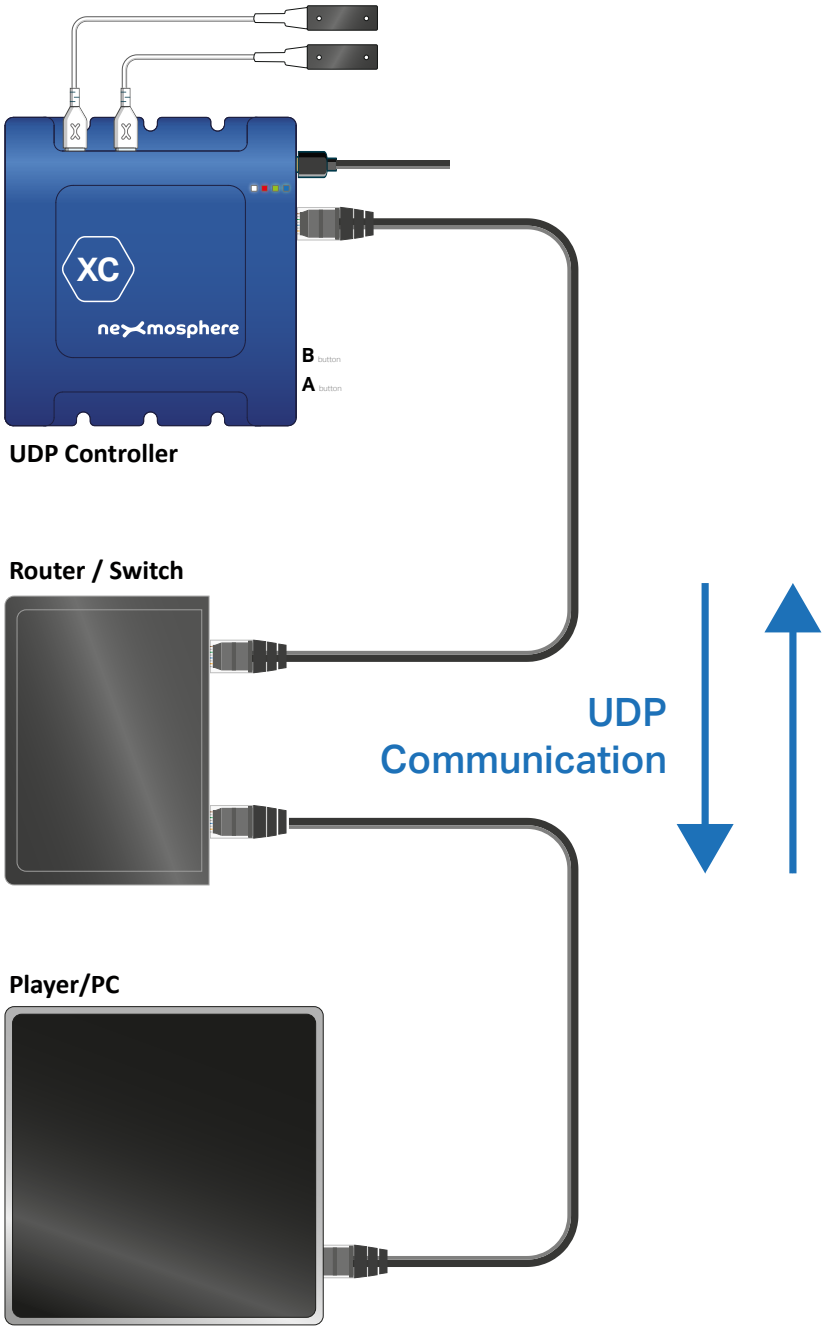
**Nexmosphere**
Le Havre 136
5627 SW Eindhoven • The Netherlands
**T** +31 40 240 7070
**E** support@nexmosphere.com

### 3. Controller configuration

Before the UDP controller can be deployed, it must be configured. All UDP controllers provide the following configuration options:

1. **Identify UDP controllers** - that are connected to your local network
2. **Configure UDP and Network settings** - to set up the connection with your player
3. **Adjust the controller behaviour** - and UDP command structures
4. **Create start-up commands** - that are automatically send to connected sensors or LED strips after each reboot
5. **Diagnostic tools** - for troubleshooting

The following sections will cover each of these functionalities in detail and provide specific examples for each functionality.






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

### 3.1 - Identify UDP controllers

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

1. Connect the UDP controller to your router and power the controller. Wait untill the blue status LED stops blinking.
2. Have a UDP terminal program on your PC/laptop **listen to port 5000.**
3. On the controller, **press button A** (closest to the corner) for 1 second, until the green LED blinks.
4. The controller will broadcast **2 identification messages** over the local network on port 5000.

| | |
|---|---|
| **FROMID=**************************:**N000B[IP=**XXX.XXX.XXX.XXX**]** | *XXX.XXX.XXX.XXX = IP Address of the controller* |
| **FROMID=**************************:**N000B[PORTIN=**PPPPP**]** | *PPPPP = Incoming (receiver) port of the controller* |
| | ************************ *= Controller ID*<br>***Length:** 1-24 ASCII characters*<br>***Default:** MAC address of the controller* |
| | ***Example messages:***<br>***FROMID=00:08:DC:75:79:8E:N000B[IP=192.168.1.171]***<br>***FROMID=00:08:DC:75:79:8E:N000B[PORTIN=5000]*** |

Alternatively to pressing button A, a set of API commands is available to identify UDP controllers 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 **broadcasted** on your network. The default incoming and outgoing port of the UDP controller is **5000**. There are 3 identification commands available:

| | |
|---|---|
| `N000B[IDENTIFY=ALL]` | Identify all UDP controllers |
| `N000B[IDENTIFY=DEFAULT]` | Only identify unconfigured UDP controllers (with factory defaults) |
| `N000B[IDENTIFY=CONFIGURED]` | Only identify configured UDP controllers |

For all 3 options, the reply is identical to the identification messages which are send when pressing button A. Per default, a controller will also automatically send an identification once, each time it is booted (see page 7 for more information).

**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 then 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 adress

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

  See page 6 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]***



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

### 3.2 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 send *after* sending one or mulltiple configuration commands:**

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

**IP Mode** - Configure how the controller obtains its IP

| | |
|---|---|
| `N000B[MODE=DHCP]` | Set the IP mode to DHCP<br>*to obtain an available IP address from your router* |
| `N000B[MODE=FIXEDIP]` | Set the IP mode to have a fixed IP<br>*to make sure the IP address of the controller will not change* |
| `N000B[FIXEDIP=XXX.XXX.XXX.XXX]` | Set the fixed IP<br>*XXX.XXX.XXX.XXX = IP address* *Default 192.168.0.100*<br>***Example command:***<br>`N000B[FIXEDIP=192.168.1.172]` |
| `N000B[SUBNETMASK=XXX.XXX.XXX.XXX]` | Set the subnet mask<br>*XXX.XXX.XXX.XXX = subnet mask* *Default - 255.255.255.0*<br>***Example command:***<br>`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 controller<br>*this should match the Outgoing (destination) port of your player*<br>*XXXXX= Incoming port* *between* **1 - 65535** *Default = 5000*<br>***Example command:***<br>`N000B[PORTIN=4999]` |
| `N000B[PORTOUT=XXXXX]` | Set the Outgoing (destination) port of the controller<br>*this should match the Incoming (receiver) port of your player*<br>*XXXXX= Outgoing port* *between* **1 - 65535** *Default = 5000*<br>***Example command:***<br>`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 controllers on the same network.



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

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

| | |
|---|---|
| `N000B[DESTIP=BROADCAST]` | Set the controller to broadcast its messages *Default*<br>*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 send<br>*typically this is the IP of your player*<br>*XXX.XXX.XXX.XXX = IP address*<br>***Example command:***<br>`N000B[DESTIP=192.168.1.175]` |

Per factory default, the controller will broadcast all outgoing messages. Broadcasting is also convenient if the sensor data should be send 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.

**Message termination** - Configure the EOL termination for the UDP messages which the controllers 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*<br>***Example command to terminate with @ character:***<br>`N000B[TERMINATION=@]` |
| `N000B[TERMINATION=X\|Y]` | Set termination to double characer *x-y= ASCII characters*<br>***Example command to terminate with @> characters:***<br>`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.



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

### 3.3 Adjust the controller behaviour

The 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 controller ID<br>*Per default, the controller ID is the MAC address* | ***Default*** |
| `N000B[PREFIX=NO]` | Disable prefix of outgoing messages<br>*This will make the outgoing messages significantly shorter* | |

Having a prefix in the outgoing messages ensures that in case of multiple controller broadcasting on the same network, you can filter from which controller a specific message was send. 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 | ***Default*** |
| `N000B[REQUIREID=YES]` | Set the controller to require the controller ID as a prefix for all incoming commands | |

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

`TOID=**********:`

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

***Example command to a controller with Controller ID "ABC"***
`TOID=ABC:N000B[BLINK!]`

This prefix is only recommended in scenarios where a player is broadcasting UDP messages to multiple controllers on the network, and each controller must be receive individual commands. For all other scenarios it is recommend to leave the incoming message prefix disabled, which is the default factory setting.

**Controller ID** - adjust the controller ID

`N000B[ID=**********]`

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

***Example command to set the Controller ID to "SEC1"***
`N000B[ID=SEC1]`

The Controller ID provides the option to identify a 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.



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

**Acknowledgement message** - to confirm receipt of incoming Network messages

| | |
|---|---|
| `N000B[ACK=ON]` | Controller will reply on incoming commands with **N000B[ACK]** |
| `N000B[ACK=ECHO]` | Controller will echo the received commands *Default*<br>*The controller will transmit an exact copy of the received message* |
| `N000B[ACK=OFF]` | Controller will not send any acknowledegement messages |

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

**Startup message** - to identifiy controllers at startup

| | |
|---|---|
| `N000B[STARTUPMESSAGE=ID]` | Controller will send an Identification message automatically at startup *Default*<br>***Example identification message:***<br>***FROMID=00:08:DC:75:79:8E:N000B[IP=192.168.1.171]***<br>***FROMID=00:08:DC:75:79:8E:N000B[PORTIN=5000]*** |
| `N000B[STARTUPMESSAGE=NO]` | Controller will not send an identification message at startup |

An Identification message can also be triggered by **pressing button A** (closest to the corner) for 1 second, until the green LED blinks, or by sending an Identification request (see page 2)

**Index prefix** - to check the communication sequence

| | |
|---|---|
| `N000B[INDEXNR=ON]` | Controller will add an upwards counting index number *at the start of each outgoing message*<br>***Example messages (presence sensor) with index number and ID prefix :***<br>***#00011:FROMID=00:08:DC:75:79:8E:X001B[Dz=05]***<br>***#00012:FROMID=00:08:DC:75:79:8E:X001B[Dz=XX]***<br>***Example messages (presence sensor) with index number and no prefix:***<br>***#00011:X001B[Dz=05]***<br>***#00012:X001B[Dz=XX]*** |
| `N000B[INDEXNR=OFF]` | Controller will not add an index number *to outgoing messages* *Default* |

Index numbers facilitate the implementation of a mechanism that checks if an outgoing messages from the 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.



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

### 3.4 Create start-up commands

The controller can be configured to automatically execute specific command at start-up, for example to adjust sensor settings or set LED outputs. 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<br>*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:

| | |
|---|---|
| `XXXXB[***...***]` | X-talk message (long) - *to configure a sensor*<br>***Example command to configure an Activation Zone on a lidar sensor***<br>`X001B[ZONE01=-200,+100,100,100]` |
| `XXXXA[*]` | X-talk message (short) -*to control outputs on Elements*<br>***Example command to set button LEDs to soft blink***<br>`X001A[170]` |
| `XXXXS[*:*]` | X-talk message (setting) - *to adjust sensor settings*<br>***Example command to set a Presence sensor to Absolute distance mode***<br>`X001S[4:2]` |
| `GXXXB[***...***]` | Generic message (long) - *to control LED outputs*<br>***Example command to set all RGBW LED outputs to 100% white***<br>`G111B[X 0 100 3]` |
| `GXXXA[*]` | Generic message (short) - *to control Audio switches*<br>***Example command to activate audio switch 111***<br>`G111A[1]` |

More examples are available in the Product Manuals of the sensors and Quick Start Guides of the controllers.

#### Verifying start-up commands

| | |
|---|---|
| `N000B[ECHORECORD!]` | Echo all recorded commands<br>*The controller will reply with a list of all recorded commands* |
| `N000B[SENDRECORD!]` | Execute all recorded commands<br>*The controller will execute all recorded commands* |
| `N000B[RECORDSTATE!]` | Request the current state of the record memory<br>*The controller will reply with the memory status* |

#### Clear start-up commands

| | |
|---|---|
| `N000B[CLEARRECORD!]` | Clear all recorded commands<br>*The controller will delete all stored start-up commands* |



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

### 3.5 Diagnostic tools

The controller offers several commands for diagnostics and debugging.

#### Diagnostic commands

| | |
|---|---|
| `N000B[RUNTIME?]` | Request the current runtime of the controller since the last reboot<br>*The controller will reply with the current runtime*<br>***Example reply (without ID prefix):***<br>***N000B[RUNTIME=00168HOUR]*** |
| `N000B[CONFIGURED?]` | Check if the controller is configured or not (factory default)<br>*The controller will reply indicating whether it is configured or not*<br>***Reply options (without ID prefix):***<br>***N000B[CONFIGURED=YES]***<br>***N000B[CONFIGURED=NO]*** |
| `N000B[MAC?]` | Request the MAC address of the controller<br>*The controller will reply with its MAC address*<br>***Example reply (without ID prefix):***<br>***N000B[MAC=34:C9:08:ED:90:2A]*** |
| `N000B[BLINK!]` | Controller will start to continuously blink its blue status LED<br>*The blue LED will stop blinking when the controller receives a new command* |

#### Reset commands

| | |
|---|---|
| `N000B[NETWORKRESET!]` | Reset all network and communication settings to factory default.<br>*These settings include:*<br>***IP mode***<br>***Fixed IP address*** *and* ***Subnet mask***<br>***Destination IP***<br>***Incoming port*** *and* ***Outgoing port***<br>***Outgoing message prefix*** *and* ***Incoming command prefix***<br>***Startup message***<br>***Index prefix settings***<br>***Acknowledgement settings***<br>***Termination settings*** |
| `N000B[FACTORYRESET!]` | Reset all settings and configuration back to factory default<br>*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 45 to 90 seconds. A factory reset can also be initiated by **pressing button A** (closest to the corner) for **more then 5 seconds**.



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

**4. Normal operation**

After the network configuration and communication settings are finalized, the controller is ready for normal operation. Typically, sensors are connected to the controller with the intent of triggering content on a screen, or to gather data. Other use cases include remotely controlling LED strips or outputs such as relay switches. The functionalities and behaviour for normal operation are classified in the following categories:

**1. Start-up behaviour** - what to expect when the controller is powered on
**2. Sensor output** - the output messages of connected Elements
**3. Control input** - how to control LEDs and connected Elements
**4. Sensor settings** - adjust the behaviour of connected sensors
**5. Status LEDs** - indicating communication and controller status

The following sections will cover each of these functionalities in detail and provide specific examples for each functionality.

### 4.1 - Start-up behaviour

The UDP controller will automatically boot when power is applied. For most models this requires connecting 12V or 24V DC power supply. For units with PoE, connecting a UTP cable with PoE will suffice. There is no ON/OFF button. The start-up procedure is as follows:

1. **Connect the sensors** to the UDP controller before applying power
2. **Connect the controller** to your local network with a UTP cable
2. **Connect the power supply** (or UTP cable with PoE) to the controller
3. The white, green and blue status LED will go on for 1 second
4. The **white LED** will start blinking, indicating that the connected Elements are being discovered and mounted. When done, the white LED will be lit continuously.

*When IP mode = DHCP (factory default)*

5. Simultaneously, the **blue LED** wil start blinking rapidly, indicating that the Controller is connecting to your local network and try to obtain an IP address from your router's DHCP Pool. Depending on the network settings this process can take anywhere between 45 to 90 seconds.
6. When the connection is established, per default, an Identification message is send out from the controller (see page 2 and 6). The green LED will briefly blink when the Identification message is send. When the IP address is obtained, the blue LED will be lit continuously.

   *In case no IP is provided from your router's DHCP pool, the Controller will automatically do 3 reattempts, at which each time the green LED will go on for 1 second. In case after 3 re-attempts no IP address is obtained by the controller, it will automatically assign IP address **169.254.123.45** to itself. This is an IP address reserved for **APIPA use cases**. It allows you to **directly connect** the controller to the UTP port of your **PC** and communicate via a UDP terminal program. When the IP address is obtained, the blue LED will be lit continuously.*

*When IP mode = fixed IP*

5. Simultaneously, the **blue LED** wil start blinking rapidly, indicating that the Controller is connecting to your local network with a fixed IP address. This process can take anywhere between 5 to 15 seconds. When the controller is ready, the blue LED will be lit continuously.

   *When in fixed IP mode, the controller will not recognize IP conflicts on your network. Meaning, that also in the case of another device on your network having the same local IP, the controller will still indicate that it is "ready" by the blue lit LED.*



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

### 4.2 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 UDP. Depending on the settings of the UDP controller, the API messages might have an additional prefix. Below are some examples for common scenarios.

*API message of a **presence sensor** connected to **X-talk channel 001** (**Default controller settings**, without ID prefix and Index prefix):*
***X001B[Dz=03]***

*API message of a **presence sensor** connected to **X-talk channel 001** (With **ID prefix "UNITA"** and without Index prefix):*
***FROMID=UNIT1:X001B[Dz=03]***

*API message of a **presence sensor** connected to **X-talk channel 001** (Without ID prefix and with **Index prefix**):*
***#00941:X001B[Dz=03]***

*API message of a **presence sensor** connected to **X-talk channel 001** (With **ID prefix "UNITA"** and with **Index prefix**):*
***#00941:FROMID=UNIT1:X001B[Dz=03]***

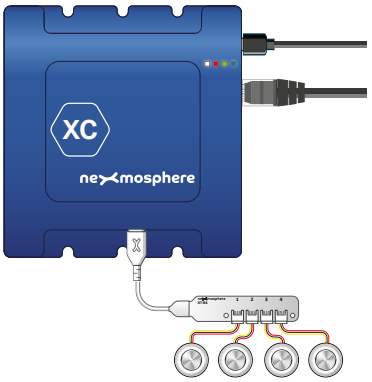

### 4.2 Control input

Nexmosphere offers various Elements of which the output can be controlled, for example button LEDs, I/O interfaces and RGBW LED strips. Next to this, some sensors require configuration, such as the lidar sensor. This is done by sending control commands via UDP to the IP address of the controller with the correct incoming port (see page 2).

The API commands are identical on all Nexmosphere controllers, including the UDP 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. Depending on the settings of the UDP controller, the API commands might have an additional prefix. Below are some examples for common scenarios.

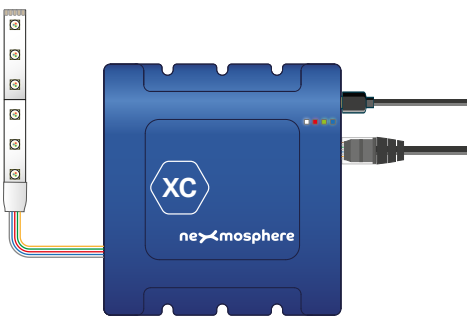
*API command to control **button LEDs** connected to **X-talk channel 002** (**Default controller settings**, without TOID prefix):*
***X002A[170]***

*API command to control **button LEDs** connected to **X-talk channel 002** (With **TOID prefix "UNITB"**):*
***TOID=UNITB:X002A[170]***




*API command to control **RGBW LEDs** connected to **RGBW output A** (**Default controller settings**, without TOID prefix):*
***G111B[A 0 100 3]***

*API command to control **RGBW LEDs** connected to **RGBW output A** (With **TOID prefix "LEDCONTROLLER"**):*
***TOID=LEDCONTROLLER:G111B[A 0 100 3]***






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

### 4.4 Sensor settings

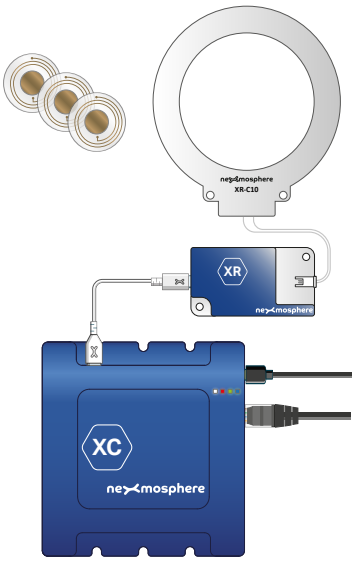
All Nexmosphere Elements have settings to adjust the behaviour and output of the sensors. This is done by sending Setting commands via UDP to the IP address of the controller with the correct incoming port (see page 2). The API commands for this are identical on all Nexmosphere controllers, including the UDP 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). Depending on the settings of the UDP controller, the API commands might have an additional prefix. Below are some examples for common scenarios.

*API command to increase the gain **setting** of a RFID Driver connected to **X-talk channel 005** (**Default controller settings**, without TOID prefix):*

***X005S[4:4]***

*API command to increase the gain **setting** of a RFID Driver connected to **X-talk channel 005** (With **TOID prefix "CTLR5"**):*

***TOID=CTLR5: X005S[4:4]***




### 4.5 Status LEDs

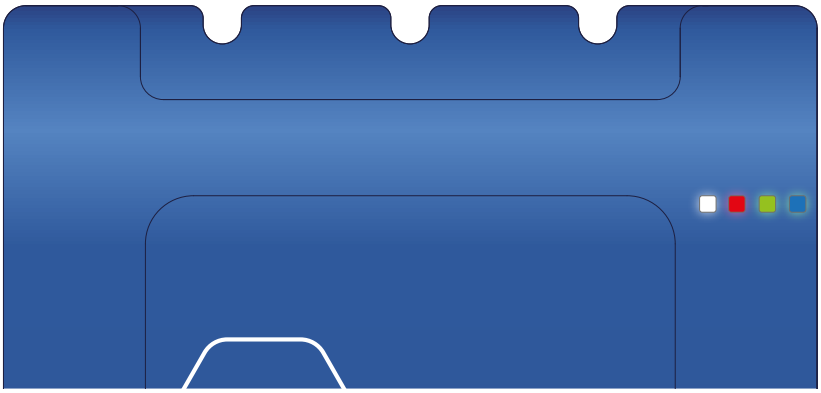
Each UDP controller has status LEDs. During normal operation, they indicate the following:

**White status LED** - X-talk communication
Default on. Blinks when sensor is triggered, or, when an API command is forwarded to a sensor.

**Red status LED** - X-talk error
Default off. Blinks when an incorrect X-talk command is received. For example when a message is send to an X-talk port with no sensor connected, or, when an X-talk command doesn't have the correct format.

**Green status LED** - UDP Transmit
Default off. Blinks when a API message is send from the controller via UDP.

**Blue status LED** - UDP Receive
Default on. Blinks when an API command is received on the controller via UDP.





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

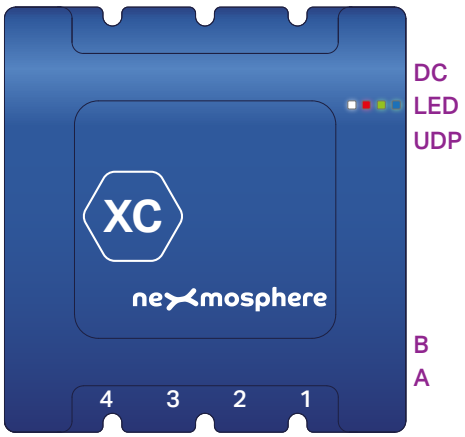
### 5. Models

In this section, an overview is provided of all UDP controllers and their interfaces.

**XC-101U** - 8x X-talk interface

**1.** X-talk interface 001
**2.** X-talk interface 002
**3.** X-talk interface 003
**4.** X-talk interface 004
**5.** X-talk interface 005
**6.** X-talk interface 006
**7.** X-talk interface 007
**8.** X-talk interface 008

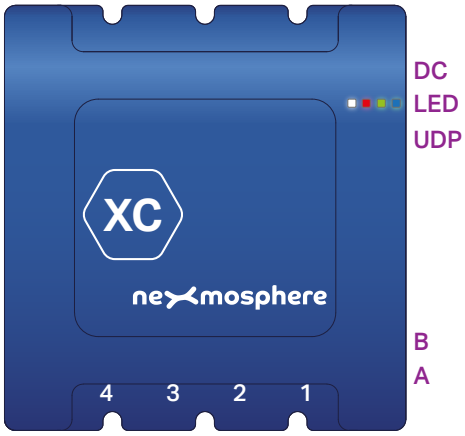
**UDP** API interface (UTP connector)
**LED.** Status LEDs
**DC.** 12-24VDC Power input (2.1mm DC socket)
**A/B.** Button A, Buttton B




**XC-101UP** - 8x X-talk interface, PoE

**1.** X-talk interface 001
**2.** X-talk interface 002
**3.** X-talk interface 003
**4.** X-talk interface 004
**5.** X-talk interface 005
**6.** X-talk interface 006
**7.** X-talk interface 007
**8.** X-talk interface 008

**UDP** API interface (UTP connector), with PoE
**LED.** Status LEDs
**DC.** 2.1mm DC socket, not used
**A/B.** Button A, Buttton B

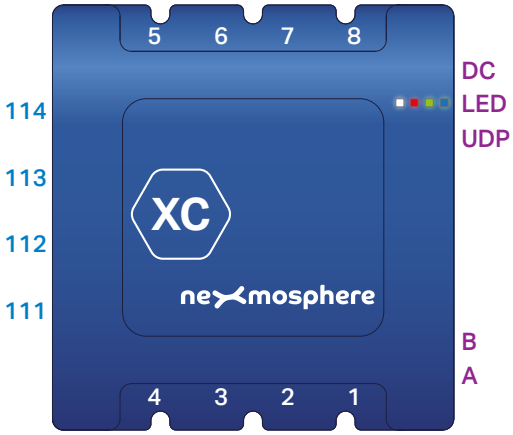



**XC-102U** - 12x X-talk interface

**1.** X-talk interface 001
**2.** X-talk interface 002
**3.** X-talk interface 003
**4.** X-talk interface 004
**5.** X-talk interface 005
**6.** X-talk interface 006
**7.** X-talk interface 007
**8.** X-talk interface 008

**111.** X-talk interface 111
**112.** X-talk interface 112
**113.** X-talk interface 113
**114.** X-talk interface 114

**UDP** API interface (UTP connector)
**LED.** Status LEDs
**DC.** 12-24VDC Power input (2.1mm DC socket)
**A/B.** Button A, Buttton B





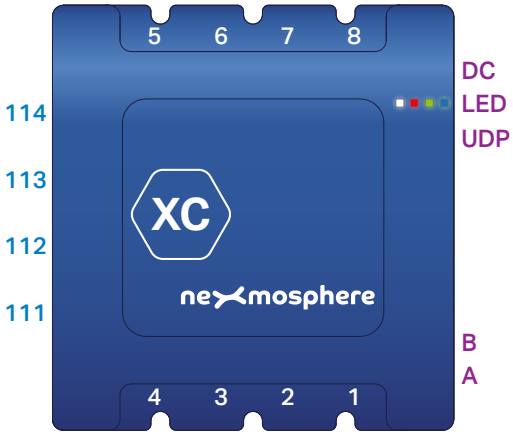

**Nexmosphere**
Le Havre 136
5627 SW Eindhoven • The Netherlands
**T** +31 40 240 7070
**E** support@nexmosphere.com

**XC-741U** - 8x X-talk interface, 4x LightMark interface

**1.** X-talk interface 001
**2.** X-talk interface 002
**3.** X-talk interface 003
**4.** X-talk interface 004
**5.** X-talk interface 005
**6.** X-talk interface 006
**7.** X-talk interface 007
**8.** X-talk interface 008

**111.** LightMark interface, API address 111
**112.** LightMark interface, API address 112
**113.** LightMark interface, API address 113
**114.** LightMark interface, API address 114

**UDP** API interface (UTP connector)
**LED.** Status LEDs
**DC.** 12-24VDC Power input (2.1mm DC socket)
**A/B.** Button A, Buttton B

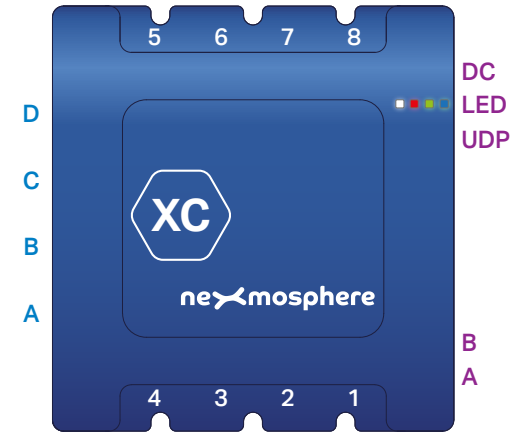



**XC-841U** - 8x X-talk interface, 4x RGBW output

**1.** X-talk interface 001
**2.** X-talk interface 002
**3.** X-talk interface 003
**4.** X-talk interface 004
**5.** X-talk interface 005
**6.** X-talk interface 006
**7.** X-talk interface 007
**8.** X-talk interface 008

**A.** RGBW LED output A (Generic API address 111, RGBW channel A)
**B.** RGBW LED output B (Generic API address 111, RGBW channel B)
**C.** RGBW LED output C (Generic API address 111, RGBW channel C)
**D.** RGBW LED output D (Generic API address 111, RGBW channel D)

**UDP** API interface (UTP connector)
**LED.** Status LEDs
**DC.** 12-24VDC Power input (2.1mm DC socket)
**A/B.** Button A, Buttton B

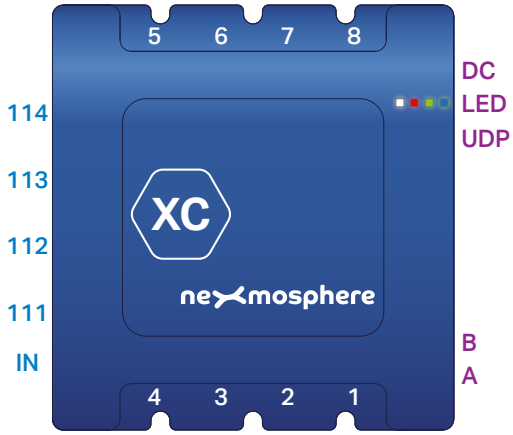



**XC-931U / XC-941U** - 8x X-talk interface, 4x Audio switch

**1.** X-talk interface 001
**2.** X-talk interface 002
**3.** X-talk interface 003
**4.** X-talk interface 004
**5.** X-talk interface 005
**6.** X-talk interface 006
**7.** X-talk interface 007
**8.** X-talk interface 008

**111.** Audio switch 1
**112.** Audio switch 2
**113.** Audio switch 3
**114.** Audio switch 4
**IN.** Audio input

**UDP** API interface (UTP connector)
**LED.** Status LEDs
**DC.** 12-24VDC Power input (2.1mm DC socket)
**A/B.** Button A, Buttton B






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