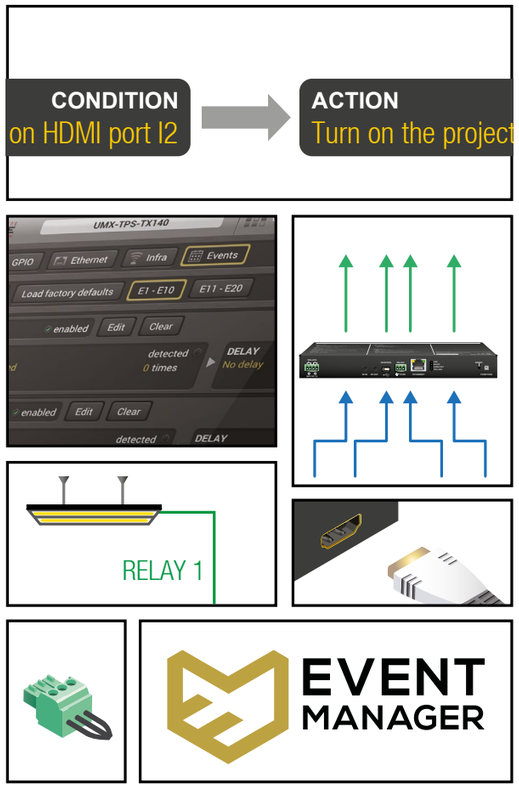
Application Notes

Event Manager



v1.4 07-03-2025

Table of Contents

1. INTRODUCTION 3
1.1. Description 4
1.2. Definition 4
1.3. Supported Devices 4
1.4. Typical Applications 5
2. EVENT MANAGER – WHERE IS IT? 6
2.1. The Events Tab 7
2.2. Adding an Event – the Event Editor 8
2.3. Special Tools and Accessories 13
3. THE CONDITION – EXPRESSIONS AND PARAMETERS 15
3.1. General Category 16
3.2. Video Category 16
3.3. Audio Category 17
3.4. Infra Category 17
3.5. GPIO Category 17
4. THE ACTION - EXPRESSIONS AND PARAMETERS 18
4.1. General Category 19
4.2. Video Category 19
4.3. Audio Category 20
4.4. RS-232 Category 21
4.5. Infra Category 21
4.6. Ethernet Category 21
4.7. GPIO Category 22
4.8. EDID Category 22
4.9. Relay Category 22
4.10. Sending a Message via an RS-232 Port 23
4.11. Sending a Message via a TCP/IP Port 23
4.12. Sending a Message via a UDP Port 24
5. TYPICAL CONNECTED DEVICES 25
5.1. Condition: Using a Pushbutton 26
5.2. Condition: Using a Motion Sensor 26
5.3. Condition: IR Signal Detection 27
5.4. Action: Controlling a Projector/Monitor 27
5.5. Action: Switching a Power Relay 28
5.6. Action: Supplying a LED Directly 29
5.7. Action: Controlling a Motorized Screen (Part #1) 29
5.8. Action: Controlling a Motorized Screen (Part #2) 30
6. TYPICAL APPLICATIONS – EXAMPLE DESCRIPTIONS 31
6.1. Details of Example 1 32
6.2. Details of Example 2 33

Document Information

Document revision: **v1.4**

Release date: **07-03-2025**

Editor: Tamas Forgacs

Contact Us

sales@lightware.com
+36 1 255 3800

support@lightware.com
+36 1 255 3810

Lightware Visual Engineering PLC.
Gizella 51-57, Budapest H-1143, Hungary
www.lightware.com

©2025 Lightware Visual Engineering. All rights reserved. All trademarks mentioned are the property of their respective owners. Specifications subject to change without notice.

1

Introduction

The Event Manager is a smart, built-in feature in the Lightware HDBaseT™ [1] compatible TPS extender family, the MODEX line and in certain matrix switchers like the MMX6x2-HT200 series. The feature is available through the Lightware Device Controller (LDC) software. In the first chapter, we would like to introduce the most important features:

- Description
- Definition
- Supported Devices
- Typical Applications

[1] HDBaseT™ and the HDBaseT Alliance logo are trademarks of the HDBaseT Alliance.

1.1. Description

The Event Manager was developed to handle tasks from the most simple to expert ones, like controlling the rolling shutter, the air conditioning system or the lights, based on any condition changes on the media ports, such as a new source being connected or removed.

Event Manager application is continuously updated with additional features via firmware upgrades: a delay can be added between the condition and the action and more actions can be triggered by a single condition change. With the help of the 'condition count' and 'action test' features, the predefined settings can be tested before going live. The system can recognize infrared commands, which can also be set as conditions, and commands can also be sent via Ethernet.

Event Manager saves time, cost and even installation space, which makes Lightware equipment the optimal choice in a number of different configurations.

1.2. Definition

The Event Manager reacts to internal status changes or user interactions without any external control system. The detected event is called **Condition**, the response is called **Action**.

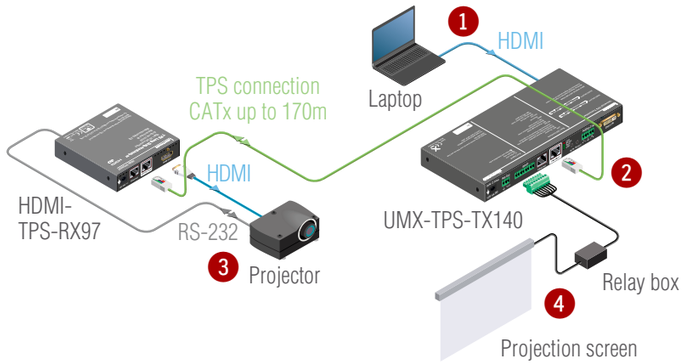
1.3. Supported Devices

As of the latest publication of this document, the following products include Event Manager:

Device Type(s)	Supported Nr. of Events	Macros
DP-TPS-TX210/TX220	20	-
DVI-HDCP-TPS-TX210/TX220	20	-
FP-UMX-TPS-TX120/130	20	-
HDMI-3D-OPT-RX150RA	20	-
HDMI-3D-OPT-TX210A/TX210RAK	20	-
HDMI-TPS-RX110AY	100	-
HDMI-TPS-RX110AY-Plus	100	✔
HDMI-TPS-TX210/TX220/TX226	20	-
HDMI-TPS-RX220AK	100	✔
HDMI20-OPTC-TX220/RX220 series	20	-
MMX4x2-HDMI/HT200	100	✔
MMX4x2-HDMI-USB20-L	100	✔
MMX6x2-HT200/210/220	100	-
MMX8x4-HT420M/HT400MC	100	✔
MMX8x8-HDMI-4K-A/HDMI-4K-A-USB20	100	✔
MODEX	32	-
RAP-B511-EU/UK/US	300	-
SW4-OPT-TX240RAK	20	-
SW4-TPS-TX240	20	-
SW4-TPS-TX240-Plus	100	✔
UMX-HDMI-140	20	-
UMX-HDMI-140-Plus	100	✔
UMX-TPS-TX120/130/140	20	-
UMX-TPS-TX140-Plus/TX140K	100	✔
WP-UMX-TPS-TX120-US/130-US	20	-
WP-UMX-TPS-TX130-Plus-US	100	✔

1.4. Typical Applications

Example 1



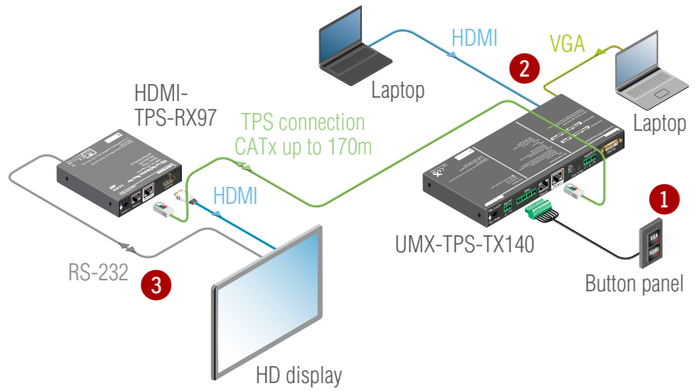
In the first example, if a signal is detected on the HDMI input port of the UMX-TPS-TX140, the listed actions are launched automatically:

- The HDMI input will be selected to transmit.
- The projector will be switched on.
- The projection screen will be rolled down.

Condition	Action	
1 Signal detected (HDMI input port)		2 Switch HDMI to transmit
		3 Switch on the projector
		4 Roll down the screen

The detailed description of this example can be found in the Details of Example 1 section.

Example 2



In the second example, if a button is pressed on the panel, the following actions are launched by the transmitter automatically:

- The related input is going to be selected to transmit.
- The HD display is going to be switched on.

Condition	Action	
1 Button is pressed		2 Select the proper input to transmit
		3 Switch on the HD display

The detailed description of this example can be found in the Details of Example 2 section.

2

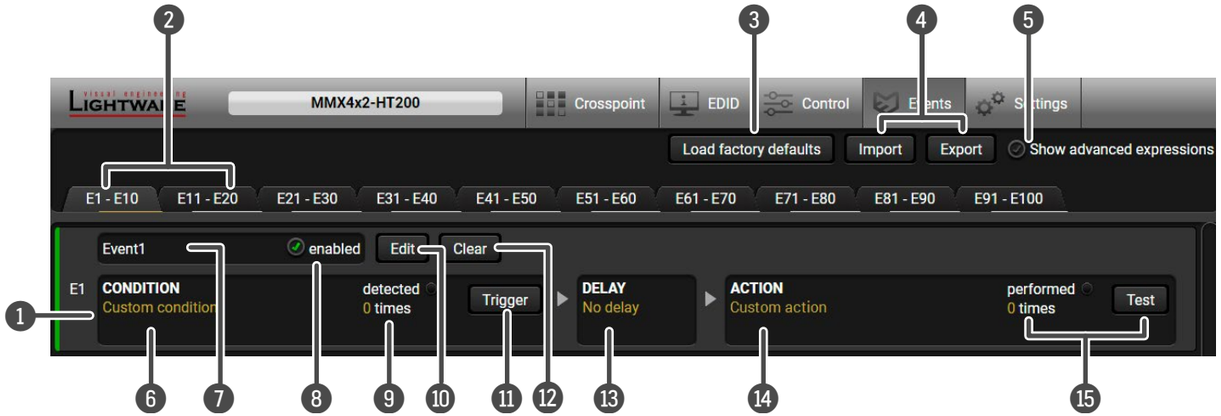
Event Manager – Where is it?

This feature is the part of the Lightware 3 protocol, therefore all settings can be arranged in the protocol tree or set by sending LW3 commands. To provide a user-friendly method for setting the necessary parameters, the feature is implemented in the Lightware Device Controller software with numerous useful features.

- The Events Tab
- Adding an Event – the Event Editor
- Useful Tools
- Further Features

2.1. The Events Tab

The location is the same in all cases in the software: navigate to the **Events** submenu next to the **Control** submenu.

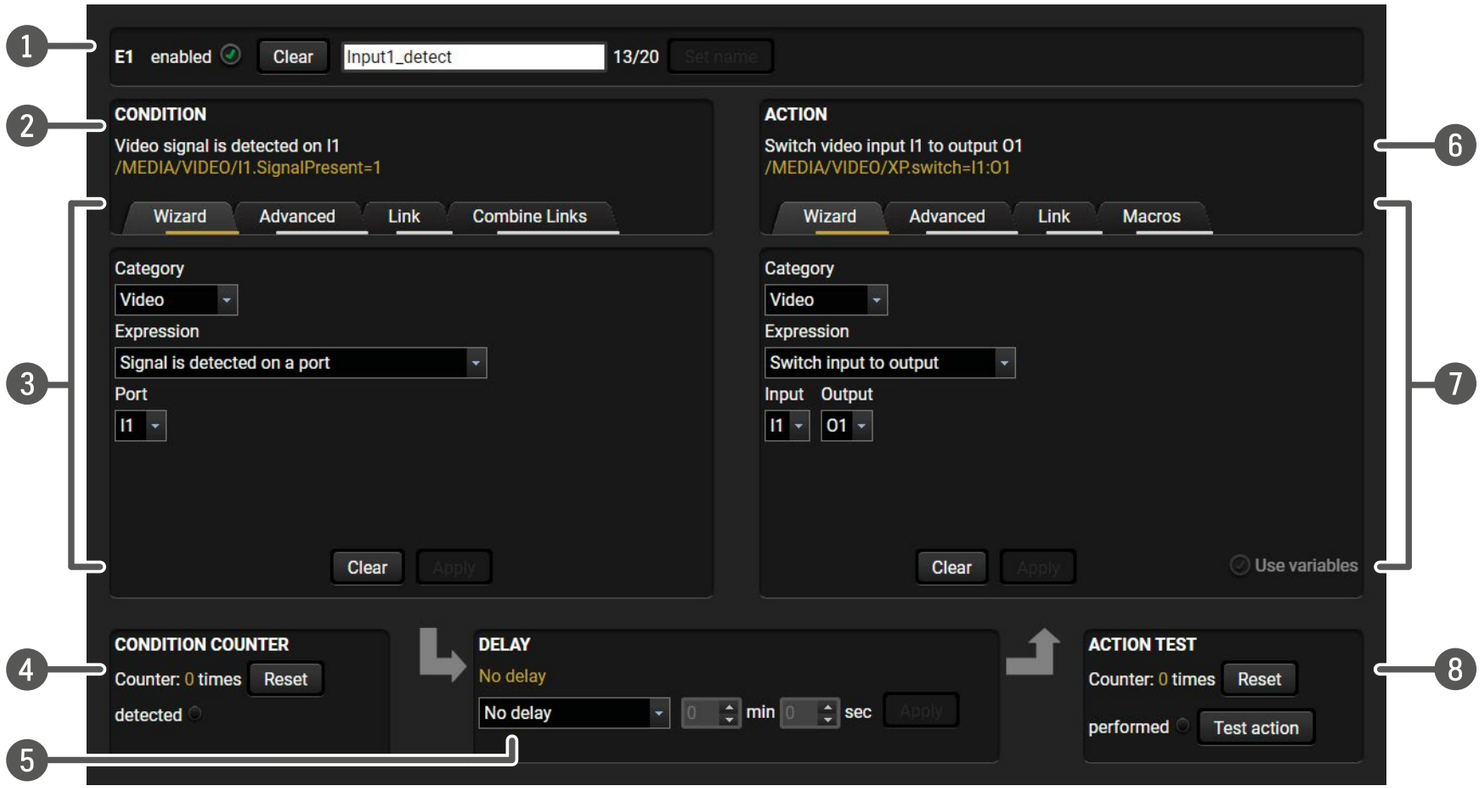


1. **Event Lines** — Each line means an **Event**: a **Condition** and an **Action**. The green line means the **Event** is enabled and both the **Condition** and the **Action** are set properly.
2. **Event Pages** — 10 **Events** can be listed at once.
3. **Factory Defaults** — All the **Events** can be deleted by pressing this button (a confirmation window will pop up before the deleting).
4. **Export and Import Buttons** — The **Events** (with all their settings) can be saved into a file and can be imported. See more details in the Clear One or More Event(s) section.
5. **Show Advanced Expressions** — Toggle the display mode of the **Conditions** and **Actions** shown in the list (see below).
6. **Condition** — Displays the expression shown in **Wizard** mode or the exact LW3 path and node.
7. **The Name of the Event** — It can be edited by the user in the **Event Editor.**
8. **Switch** — The **Event** can be enabled or disabled.
9. **Condition Test** — If the **Condition** is detected, the green indicator is lit for three seconds and the counter is increased. See also the Testing the Condition section.
10. **Edit Button** — Press the button to open the **Event Editor** and set all the parameters.
11. **Trigger Button** — In complex control systems, where several **Actions** may be based on the same **Condition**, the number of commands can be reduced by setting a "key" **Condition** that can start the process of performing the relevant **Actions**.
12. **Clear Button** — Delete the settings of the given **Event**. See also the Clear One or More Event(s) section.
13. **Delay Settings** — The **Action** can be scheduled to follow the **Condition** after the set time value. See also the Delaying the Action section.
14. **Action** — Displays the expression shown in **Wizard** mode or the exact LW3 path and node.
15. **Action Test** — The counter works the way same as with the **Condition** test, but in this case the **Action** can be tested by pressing the **Test** button See also the Testing the Action section..



2.2. Adding an Event – the Event Editor

Press the **Edit** button in the desired **Event** line to open the **Event Editor** window.

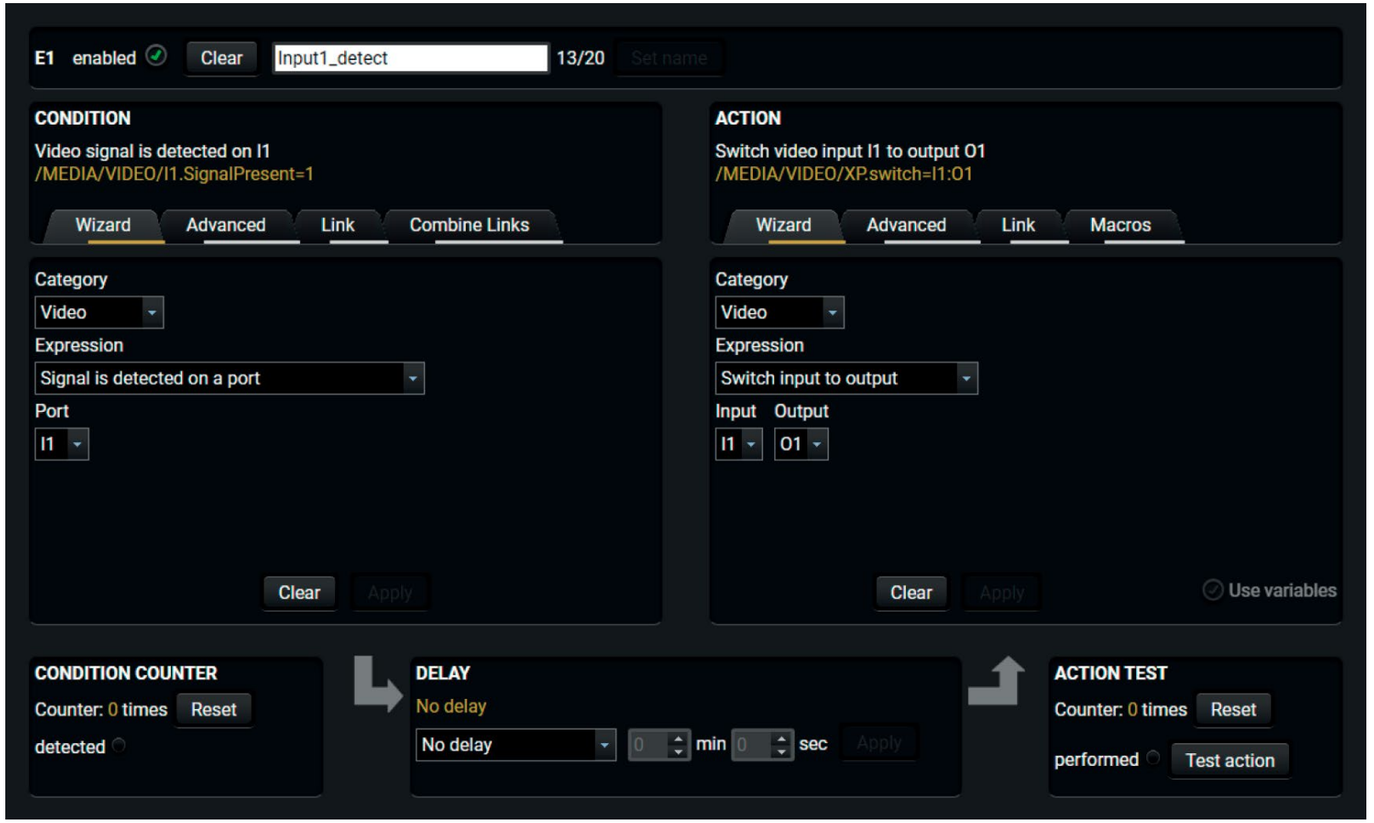


INFO: The **Delay** and **Action test** features are described in the next sections.

1	**Event Header**	The name of the **Event** is displayed. Type the desired name and press the **Set name** button. The Event can be cleared by the **Clear** button. Add/remove the tick mark to enable/disable the **Event**.
2	**Condition Header**	If the **Condition** is set, the description (white colored text) and the exact LW3 protocol expression (yellow colored text) can be seen. **Custom Condition** means the entry is not available in **Wizard** mode, only in **Advanced** mode.
3	**Condition Panel**	Different tools are available at the tabs to set the desired **Condition**. The parameters and settings are displayed below the buttons.
4	**Condition Counter**	If the set **Condition** is detected (or triggered), the indicator turns green for two seconds, and the counter is increased. The counter is reset at boot.
5	**Delay Settings**	The **Action** can be scheduled to follow the **Condition** after the set time value.
6	**Action Header**	If the **Action** is set, the description (white colored text) and the exact LW3 protocol expression (yellow colored text) can be seen. If the **Advanced** mode was used, the description is "**Custom Action**".
7	**Action Panel**	Different tools are available at the tabs to set the **Action**. The parameters and settings are displayed below the buttons.
8	**Action Test**	The set **Action** can be tested to see the working method in the practice.

2.2.1. The Wizard Interface

The most often used method to arrange the settings of an **Event** is to use the **Wizard** mode.



Setting the Condition

Step 1. Open the **Event Editor** by pressing the **Edit** button in the desired **Event** line. The default tab is the **Wizard** mode.

Step 2. Select the desired **Category** from the left panel (**Condition** section).

Step 3. Select the desired **Expression**.

Step 4. In most cases further parameters have to be set (e.g. port number), which are displayed in a new line.

Step 5. Check the entered parameters and press the **Apply** button to store the **Condition** settings.

Step 6. Check the **Enabled** option in the top line of the **Event Editor**.

ATTENTION! Do not forget to press the Apply button when the Condition is arranged.

Setting the Action

Step 1. Open the **Event Editor** by pressing the **Edit** button in the desired **Event** line. The default tab is the **Wizard** mode.

Step 2. Select the desired **Category** from the right panel (**Action** section).

Step 3. Select the desired **Expression**.

Step 4. In most cases further parameters have to be set (e.g. port number), which are displayed in a new line.

Step 5. Check the entered parameters and press the **Apply** button to store the **Condition** settings.

Step 6. Check the **Enabled** option in the top line of the **Event Editor**.

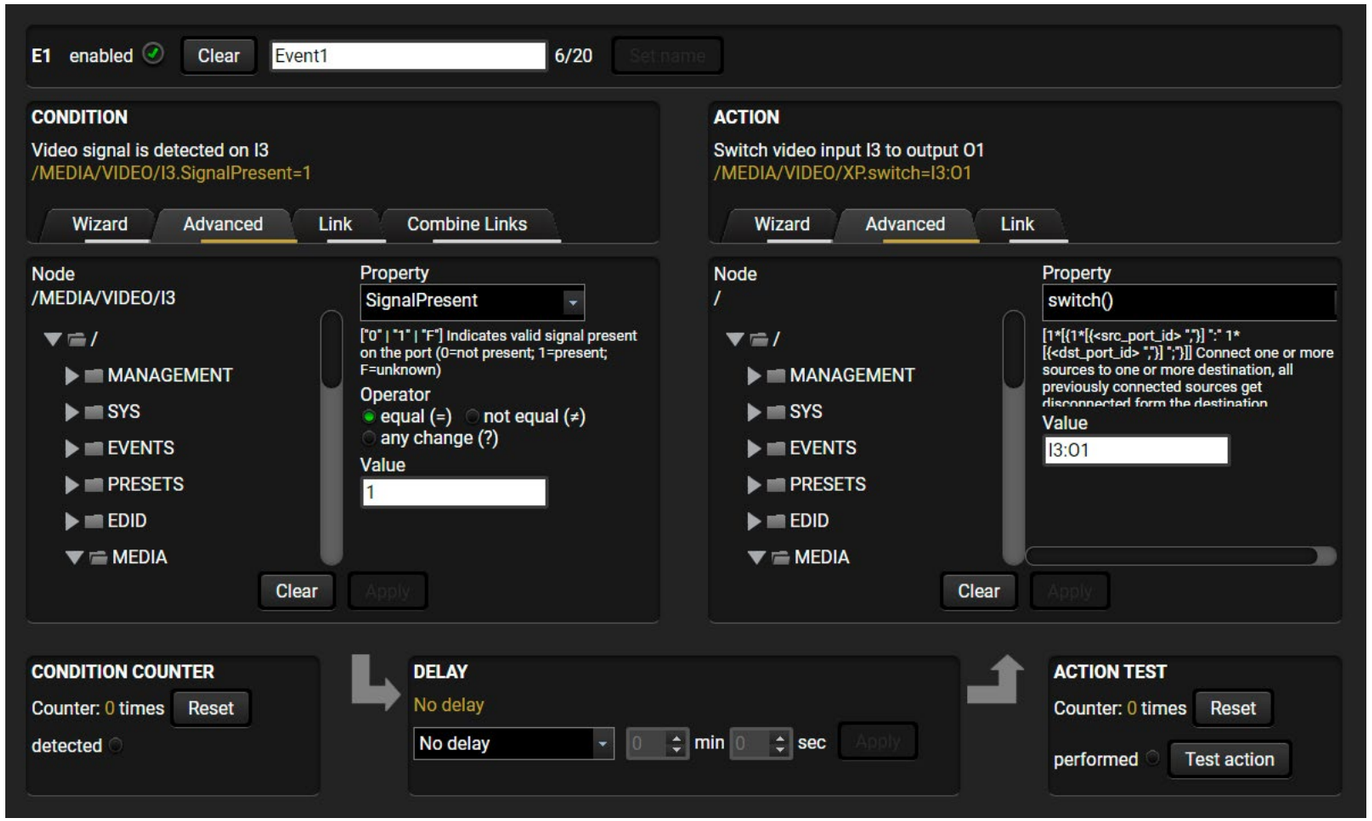
ATTENTION! Do not forget to press the Apply button when the Action is arranged.

TIPS AND TRICKS: You do not have to set the **Condition/Action** again if it is the same as at another **Condition/Action,** since it can be linked from a previously set **Event** on the **Link** tab.

INFO: The **Use variables** option is relevant only if the device defines variables (devices supported by the Advanced Control Pack v3) For details about whether your device is capable of defining variables, please see its User's Manual.

2.2.2. The Advanced Interface

The most often used **Conditions** and **Actions** are available in **Wizard** mode. Sometimes a special parameter/method is needed to observe/call/set, in these cases the **Advanced** mode is the solution. This mode allows you to set any node of the LW3 tree as a **Condition/Action** expression, so it is recommended only for expert users.



INFO: The **Wizard** and the **Advanced** modes can be used simultaneously; if the **Condition** is set in **Wizard** mode, the **Action** can be set both in **Wizard** or **Advanced** mode and vice versa.

Setting the Condition

Step 1. Open the **Event Editor** by pressing the **Edit** button in the desired **Event** line. Select the **Advanced** tab on the left panel (**Condition** section). The **LW3 protocol tree** can be browsed on the left side and the currently selected **Node** is displayed above the tree. If a **Node** is opened, all its child nodes are loaded in the **Property** drop-down menu.

Step 2. Navigate to the desired **Node** and select the **Property**. The corresponding node manual (descriptor) is also displayed. In most cases further parameters have to be set, which will be displayed under the **Property** drop-down menu.

Step 3. Check the entered parameters and press the **Apply** button to store the **Condition** settings.

ATTENTION! Do not forget to press the **Apply** button when the **Condition** is arranged.

Setting the Action

Step 1. Open the **Event Editor** by pressing the **Edit** button in the desired **Event** line. Select the **Advanced** tab on the right panel (**Action** section). The LW3 protocol tree can be browsed on the left side and the currently selected **Node** is displayed above the tree. If a **Node** is opened, all its child nodes are loaded in the **Property** drop-down menu.

Step 2. Navigate to the desired **Node** and select the **Property**. The corresponding node manual (descriptor) is also displayed. In most cases further parameters have to be set, which will be displayed under the **Property** drop-down menu.

Step 3. Check the entered parameters and press the **Apply** button to store the **Action** settings.

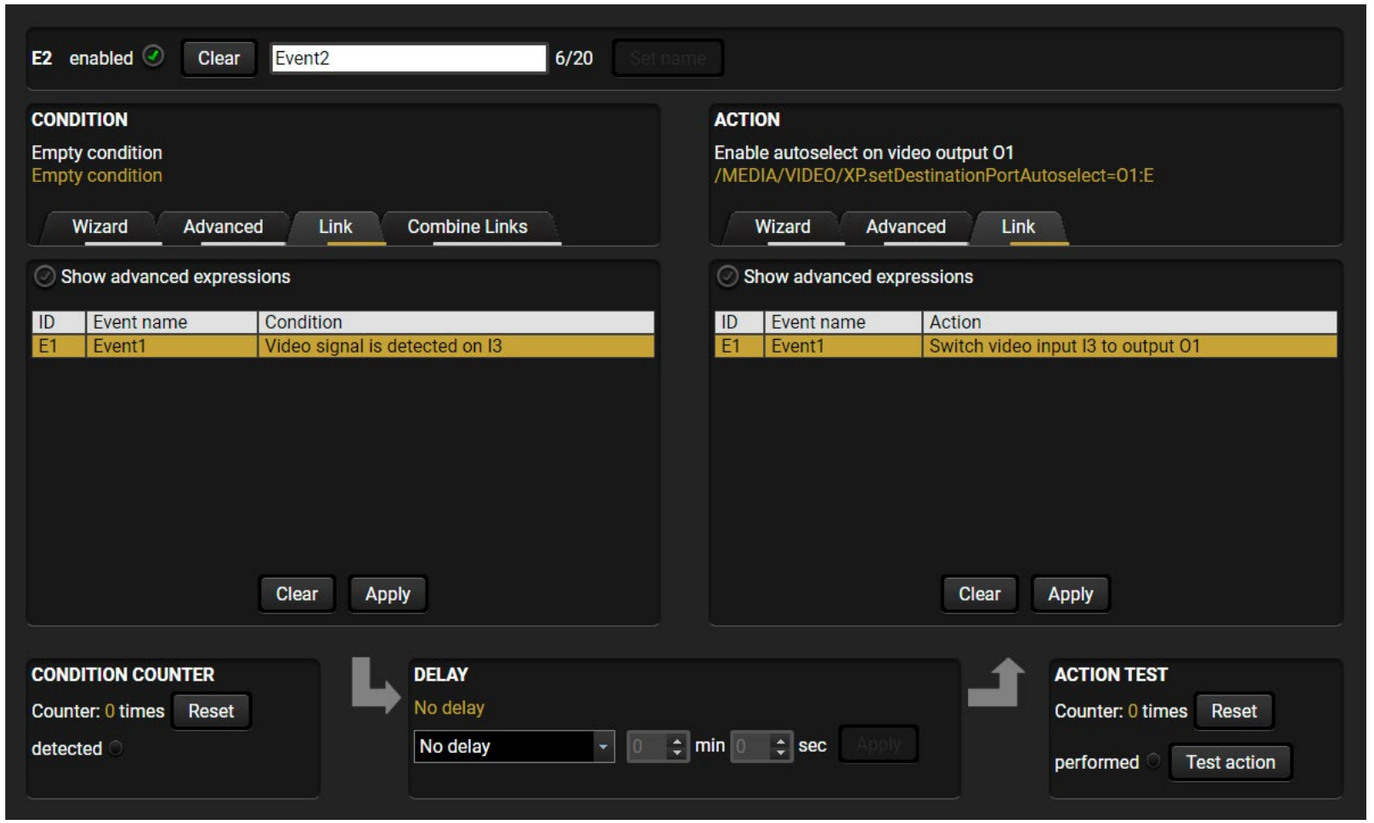
ATTENTION! Do not forget to press the **Apply** button when the **Action** is arranged.

TIPS AND TRICKS: You do not have to set the **Condition/Action** again if it is the same as at another **Condition/Action,** since it can be linked from a previously set **Event** on the **Link** tab.

2.2.3. The Link Interface

The creation of this feature represents Lightware's development processes: we received much feedback from the first users of the **Event Manager** about the missing and desired functions. The **Link** tool is a comfortable way to set **Events** faster by linking a **Condition** or **Action** that was defined at another **Event** previously, therefore:

- A **Condition** could trigger more **Actions**, and
- Different **Conditions** could launch the same **Action**.



This helps a lot when the **Action** or **Condition** that is used multiple times needs to be changed. Thanks to the linking, only the original **Condition** or **Action** has to be changed, and all linked ones will be updated automatically.

Linking a Condition or an Action

Step 1. Open the **Event Editor** by pressing the **Edit** button in the desired **Event** line. Select the **Link** tab on the desired panel (**Condition** or **Action**). All the **Conditions/Actions** will be listed which were set previously at other **Events**.

Step 2. Select the desired **Condition/Action**.

Step 3. Check the entered parameters and press the **Apply** button to store the settings.

> **ATTENTION!** Do not forget to press the **Apply** button when the linked **Condition/Action** is arranged.

2.2.4. The Combining Links Interface (Condition only)

The first generation of the Event Manager is able to sense a change (one condition) but in some cases that is not enough. The practical experience has shown that there is a need to examine several conditions as follows: one condition exists, and the following is a change that occurs. Up to four Conditions can be set. The **Combine Links** tool allows launching an Action if the Conditions are present at the same time. Follow the steps below:

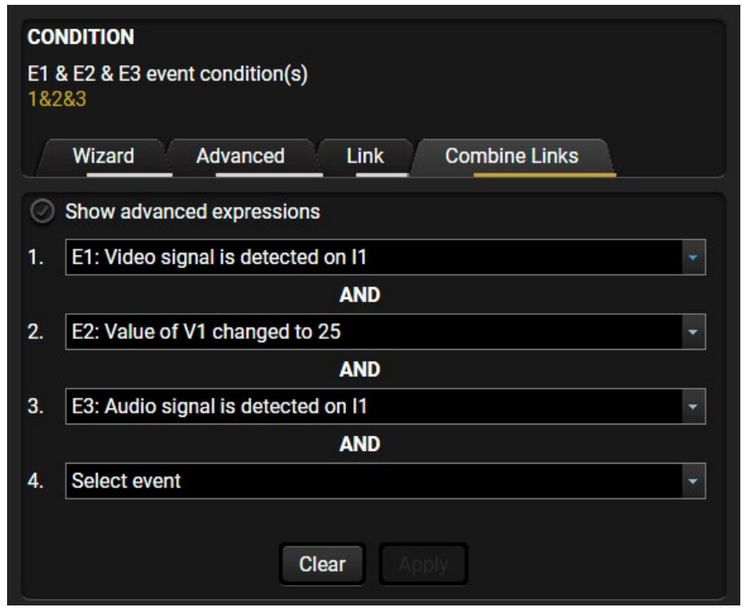
Step 1. Create the **Conditions** in separate Events.

Step 2. Create a new Event and navigate to the **Combine Links** tab in the Editor.

Step 3. Select the **Conditions** in the **left** panel.

Step 4. Set the desired **Action** (and do not forget to **enable** the Events).

INFO: This mode is available for setting the **Condition only**.



2.2.5. The Macros Interface (Action only)

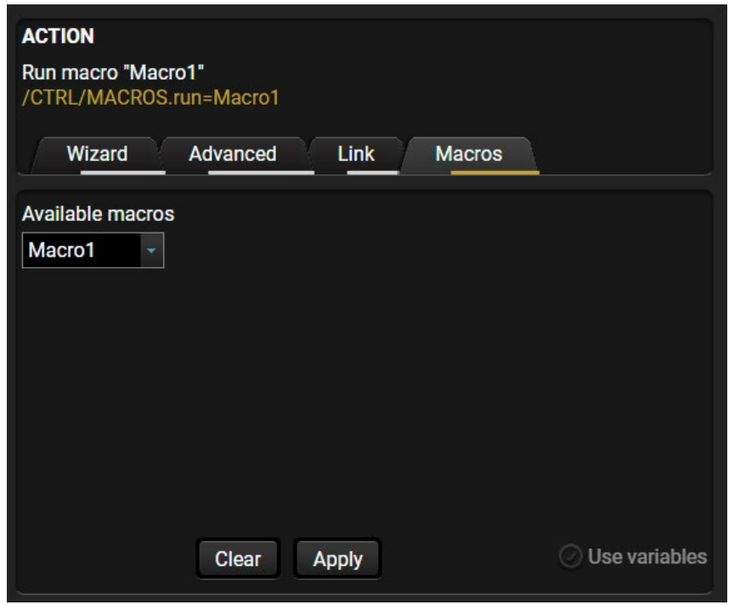
ATTENTION! Macros are only available for certain devices. See the list of devices that support macros in the Supported Devices table.

Macros stored in the device can be run as Actions. Thus, a lot of commands can be run by using only one Event. The commands will be processed one-by-one after each other. #macro

Step 1. Click on the **Edit** button of the desired Event; the **Event editor** is displayed.

Step 2. The wizard mode is the default, press the **Macros** button in the **Action** panel.

Step 3. Select the desired **Macro** from the list, then press the **Apply** button to store settings.



2.3. Special Tools and Accessories

The Name of the Event

The name of an Event can be set by typing the new name and clicking on the **Set** button. The name can be 20 characters long at most, and the following characters are allowed:

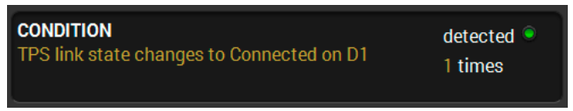
Letters (A-Z) and (a-z), numbers (0-9), special characters: hyphen (-), underscore (_), and space ( ).

Enable or Disable an Event

The set Event can be enabled and disabled in the Event list, or directly in the Event editor window by adding/removing a tick mark beside the name.

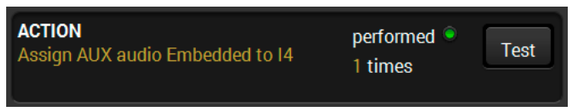
Testing the Condition

When the desired **Condition** is arranged, the setting can be tested. The Event list and the Event editor contains a small panel that shows an indicator if the set **Condition** is detected (or triggered). The detected mark turns green for two seconds and the **Counter** is increased. The **Counter** is reset at boot or can be reset manually via the button in Event editor.



Testing the Action

The counter works the same as at the **Condition** test, but in this case the **Action** itself can be triggered manually by pressing the **Test** button.



TIPS AND TRICKS: The **Test** button is also placed on the Action panel in the Event list. Thus, you can check the Actions without opening the Event editor.

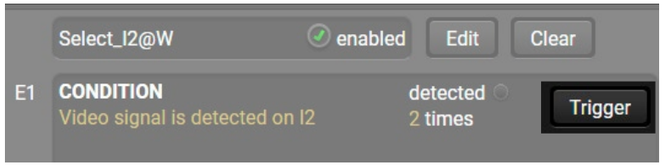
Incorrect Values

If you typed an incorrect value, you will get a notice in the form of an exclamation mark.



Condition Triggering

This improvement works as if a condition is detected. When a complex control system is built, a **Condition** may trigger numerous **Actions**. A typical example is when a system is powered on and the 'ready-to-use' state has to be loaded. In this case, there could be many **Actions** that are based on the same **Condition**. In order to reduce the number of the commands, you can trigger one 'key' condition, which could start the whole process.

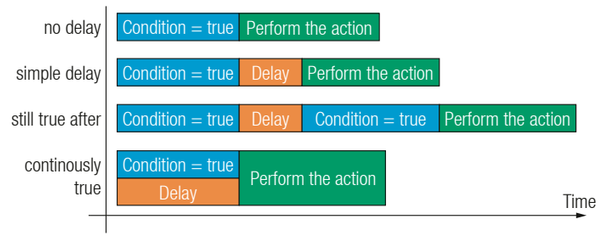


INFO: Please note that Condition triggering is only available for certain devices. For details on whether your device contains this feature, plese see its User's Manual.

Delaying the Action

In most cases, the **Action** is performed immediately after the **Condition** is detected. However, sometimes a delay is necessary between the **Condition** and the **Action**. Therefore, the **Event Manager** contains the **Delay** panel with the below settings:

- **No delay**: when the Condition is detected, the Action is launched.
- **Simple delay:** when the Condition is detected, the Action is launched after the set time interval.
- **Still true after:** After the Condition was detected and the Delay time is over, the Condition is checked again. If it still exists the Action is launched.
- **Continuously true:** After the Condition was detected, the Condition is checked continuously throughout the Delay time. If it is continuously existing, the Action is launched.



Variables

> **ATTENTION!** Please note that this feature is only available in certain devices. To see if your device supports this feature, check the corresponding User's Manual, which can be downloaded from www.lightware.com.

The variables and the tools (shown in the Control/Variables tab) can be used in the Event Manager.

Condition Wizard *#variables #new*

The desired variable can be checked if its value is changed to a specific value.

Action Wizard

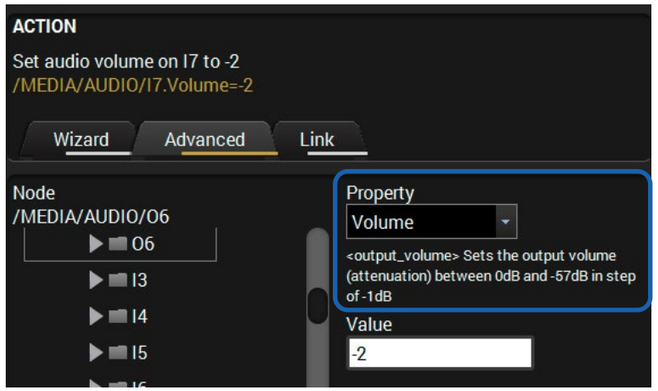
The value of a variable can be set/changed by the same methods as in the Control/Variables tab:

- Setting the exact value
- Increasing the value with/without limits
- Converting the value (e.g. interval change)
- Scanning and storing, or reformatting a property/parameter

If the **Use variables** option is enabled, you can link a variable by the **$** character. If you do so (e.g. $1), the device will handle it as the V1 variable. You can send the value of the variable e.g. as a TCP message.

Displaying the Manual of a Property

When using the **Advanced** tab in **Event Editor,** the **Manual** (Short information) of the currently selected **Property** is displayed. That can be used when setting exact parameters (e.g. volume level or Autoselect mode):



Clear One or More Event(s)

Clear an Event

Press the **Clear** button in the Event list or in the header section in the Event editor.

Clear all Events

When all the Events must be cleared, press the **Load factory defaults** button above the Event list. You will be prompted to confirm the process.

> **ATTENTION!** Clicking on the **Load factory defaults** button means the configuration of the buttons are also deleted, because the conditions and the actions of the buttons are saved as events.

Import / Export

The list of the **Events** can be easily imported and saved to a file. The feature allows creating different lists for different circumstances and applications, but it is also suitable for creating backups.

> **ATTENTION!** The structure of the devices are different, which is also true for the software and firmware components. Therefore, the exported/imported list of **Events** is guaranteed to work only in the same type of device and running the same Firmware version. Certain **Events** may be applied in different types of devices, but this is not guaranteed.

Export all Events

Step 1. Press the **Export** button above the Event list.

Step 2. The Save as dialog box will appear. Set the desired folder and file name, then press the **Save** button.

The generated file is a simple text file, which contains LW3 protocol commands. The file can be viewed by a simple text editor, e.g. Notepad.

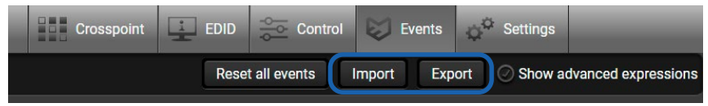
> **ATTENTION!** Editing the file is recommended only for expert users.

Import all Events

Step 1. Press the **Import** button above the Event list.

Step 2. The Open dialog box will appear. Select the desired folder and file, then press the **Open** button.

The import/export feature is available under the **Events** menu:



3

The Condition – Expressions and Parameters

This chapter describes the Conditions that are listed in Wizard mode under the following groups:

- General Category
- Video Category
- Audio Category
- Infra Category
- GPIO Category

Basic Rules

1. The port numbering can be different in the devices. To set the proper input/output port, please see the exact port numbers in the User's Manual of your device.
2. It may take a few seconds to load all the parameters. Please wait until the **Loading** animation disappears from the bottom of the main screen before selecting a port/parameter.



3. The easiest way to set the parameters is to use the **Lightware Device Controller** software:
   a) Start **LDC** and connect to the device.
   b) Navigate to the **Control / Events** tab.
   c) Press the **Edit** button in the desired **Event** line to open the **Event Editor**.

ATTENTION! The following sections include an aggregated list of all the **Conditions**. Some of the **Conditions** are available only at certain devices.

3.1. General Category

1.) TPS link state changes to Connected on a port

Parameters: - Port number (TPS input or TPS output)

Notes: - The **Condition** may be triggered by plugging in the TPS cable, restarting/powering on the connected device, etc.

2.) TPS link state changes to Disconnected on a port

Parameters: - Port number (TPS input or TPS output)

Notes: - The above **Condition** may be triggered by unplugging the TPS cable, restarting/powering off the connected device, etc.

3.) OPT link state changes to Connected on a port

Parameters: - Port number (OPT input or OPT output port)

Notes: - The **Condition** may be triggered by plugging in the fiber optical cable, restarting/powering on the connected device, etc.

4.) OPT link state changes to Disconnected on a port

Parameters: - Port number (OPT input or OPT output port)

Notes: - The above **Condition** may be triggered by unplugging the fiber optical cable, restarting/powering off the connected device, etc.

5.) Show me button pressed

Parameters: N/A

Notes: - This is the **Show me** button on the front panel (not the same as the **Function** button).

6.) Function button pressed

Parameters: N/A

Notes: - This is the **Function** button on the front panel (not the same as the **Show me** button).

3.2. Video Category

1.) Signal is detected on a port

Parameters: - Port number (input or output)

Notes: - Any port type can be selected that can carry a video signal (e.g. HDMI, DVI, TPS, OPT, etc.).
- This **Condition** is about checking the video signal of a port, the audio is not checked. If you want to check the audio presence, see the related sections under **Audio** category.

2.) Signal is not detected on a port

Parameters: - Port number (input or output)

Notes: - Any port type can be selected that can carry a video signal (e.g. HDMI, DVI, TPS, OPT, etc.).
- This **Condition** is about checking the video signal of a port, the audio is not checked. If you want to check the audio presence, see the related sections under **Audio** category.

3.) Signal type changes to DVI

Parameters: - Port number (output)

Notes: N/A

4.) Signal type changes to HDMI

Parameters: - Port number (output)

Notes: N/A

5.) Signal type changes to Undefined (no signal)

Parameters: - Port number (output)

Notes: N/A

3.3. Audio Category

Signal Detection – Important Note

Due to the structure of the audio ports, the signal detection works as follows:

- **Analog audio port (Phoenix):** shows that signal is always present.
- **Analog audio port (3.5mm Jack):** the signal is present if the plug is connected.
- **Digital ports (HDMI, TPS, S/PDIF, etc...):** audio stream presence is detected.

1.) Signal is detected on a port

PARAMETERS: - Port number (input or output)
NOTES: N/A

2.) Signal is not detected on a port

PARAMETERS: - Port number (input or output)
NOTE: N/A

3.) Signal type changes to PCM

PARAMETER: - Port number (output)
NOTES: N/A

4.) Signal type changes to Compressed

PARAMETERS: - Port number (output)
NOTES: - Check the desired port specifications about the supported audio signal types.

5.) Signal type changes to HBR

PARAMETERS: - Port number (output)
NOTES: - Check the desired port specifications about the supported audio signal types.

6.) Signal type changes to Undefined (no signal)

PARAMETERS: - Port number (output)
NOTES: N/A

3.4. Infra Category

1.) Infra code recognized

PARAMETERS: - Port number (IR input)
- IR code

NOTES: - The port can be a local IR port and/or a TPS port (which transmits the IR signal).
- In order to select a value from the code list, the infra code has to be defined (learned) in advance in the **Control** menu, **Infra / IR codes** tab.

3.5. GPIO Category

1.) State changes to 'High'

PARAMETERS: - Pin number (GPIO port)

NOTES: - To sense the status of a GPIO port, it has to be set as **Input**. However, the state is also shown when it is configured as output.
- Because of the internal pull-up on the ports, when there is nothing connected to the **GPIO input**, the state is **High**.
- When detecting an external button press, the state will change to **Low**.

2.) State changes to 'Low'

PARAMETERS: - Pin number (GPIO port)

NOTES: - To sense the status of a GPIO port, it has to be set as **Input**. However, the state is also shown when it is configured as output.
- When there is nothing connected to the GPIO input, the default state is **High**.
- When detecting an external button press, the state will change to **Low**.

4

The Action - Expressions and Parameters

This chapter describes the Actions that are listed in Wizard mode under the following groups:

- General Category
- Video Category
- Audio Category
- RS-232 Category
- Infra Category
- Ethernet Category
- GPIO Category
- EDID Category
- Relay Category
- Sending a Message via an RS-232 Port
- Sending a Message via a TCP/IP Port
- Sending a Message via a UDP Port

ATTENTION! The following sections include an aggregated list of all the **Actions**. Some of the **Actions** are available only at certain devices.

4.1. General Category

1.) Turn off LCD backlight

PARAMETERS: N/A

NOTES: - This setting is stored, so when the device is rebooted, the backlight will still be off.

2.) Turn on LCD backlight

PARAMETERS: N/A

NOTES: N/A

3.) Blink LCD backlight

PARAMETERS: N/A

NOTES: - This setting is stored, so when the device is rebooted, the backlight will still be blinking.

4.) Show text on LCD

PARAMETERS: - Message text line 1
- Message text line 2
- Duration time (x10 ms); e.g. 200 means 2 seconds.

NOTES: - The accepted characters:
a-z, A-Z, 0-9, and ' " + ! % / = ( ) , . - : < > _ * + |
- Two lines can be displayed, no text scrolling

5.) Beep

PARAMETERS: - Duration time (x10 ms)

NOTES: N/A

4.2. Video Category

1.) Switch input to output

PARAMETERS: - Port number (input)
- Port number (output)

NOTES: - Any port type can be selected that can carry a video signal (e.g. HDMI, DVI, TPS, OPT, etc.)
- The Switching Action disables the Autoselect automatically if it was enabled previously.

2.) Switch next input to output

PARAMETERS: N/A

NOTES: - The order is based on the port numbering (I1 → I2 → ... → In). E.g. if I3 is on the output and a condition triggers this action, I4 will be on the output.
- The Switching Action disables the Autoselect automatically if it was enabled previously.

3.) Enable autoselect on output

PARAMETERS: - Port number (output)

NOTES: N/A

4.) Disable autoselect on output

PARAMETERS: - Port number (output)

NOTES: N/A

5.) Load crosspoint preset

PARAMETERS: - Preset name

NOTES: - Define the preset in advance for it to appear in the drop-down menu.
- The Switching Action disables the Autoselect automatically if it was enabled previously.

6.) Mute output

PARAMETERS: - Port number (output)

NOTES: N/A

7.) Unmute output

PARAMETERS: - Port number (output)

NOTES: N/A

8.) Mute input

PARAMETERS: - Port number (input)

NOTES: N/A

9.) Unmute input

PARAMETERS: - Port number (input)

NOTES: N/A

4.3. Audio Category

1.) Set volume

Parameters: - Port number (analog audio input or output)
- Volume

Notess: - The accepted value range is device-dependent. The supported interval can be found in the User's manual of the device, but in most cases, a workaround could also help, see the Displaying the Manual of a Property section.
- Volume adjustment is only available at the analog audio ports.

2.) Increase volume

Parameters: - Port number (output)
- Percent value

Notes: - Volume adjustment is only available at the analog audio ports.

3.) Decrease volume

Parameters: - Port number (output)
- Percent value

Notes: - Volume adjustment is only available at the analog audio ports.

4.) Mute output

Parameters: - Port number (output)

Notes: - Muting the audio at a video port will remove the audio stream from the HDMI signal, while the video will still be present.

5.) Unmute output

Parameters: - Port number (output)

Notes: N/A

6.) Mute input

Parameters: - Port number (input)

Notes: - Muting the audio at a video port will remove the audio stream from the HDMI signal, while the video will still be present.

7.) Unmute input

Parameters: - Port number (input)

Notes: N/A

8.) Mute analog output

Parameters: - Port number (output)

Notes: N/A

9.) Unmute analog output

Parameters: - Port number (output)

Notes: N/A

10.) Mute HDMI output embedded audio

Parameters: - Port number (output)

Notes: - Muting the audio at a video port will remove the audio stream from the HDMI signal, while the video will still be present.

11.) Unmute HDMI output embedded audio

Parameters: - Port number (output)

Notes: N/A

12.) Change audio assignment

Parameters: - Port number (analog audio input port) or Embedded audio
- Destination port (input or output)

Notes: N/A

13.) Switch input to output

Parameters: - Port numbers (audio input and output port)

Notes: - The Switching Action disables the Autoselect automatically if it was enabled previously.

14.) Switch next input to output

Parameters: N/A

Notes: - The order is based on the port numbering (I1 → I2 → ... → In). If a condition triggers this action the following allowed input port will be switched to the output.
- The Switching Action disables the Autoselect automatically if it was enabled previously.

15.) Enable autoselect on output

Parameters: - Port number (output)

Notes: N/A

16.) Disable autoselect on output

Parameters: - Port number (output)

Notes: N/A

4.4. RS-232 Category

1.) Send RS-232 message

PARAMETERS: - Port number
- Message text

NOTES: - Any port type can be selected that can carry an RS-232 signal (e.g. TPS, OPT, etc.).
- Control characters can be inserted into the defined text since the escaping is allowed in this method (the control characters are interpreted).
- The detailed description of this action can be found in the Sending a Message via an RS-232 Port section.

2.) Switch input to output

PARAMETERS: - Port numbers (input and output port)

NOTES: - Any port type can be selected that can carry an RS-232 signal (e.g. TPS, OPT, etc.).

3.) Mute output

PARAMETERS: - Port number

NOTES: - Any port type can be selected that can carry an RS-232 signal (e.g. TPS, OPT, etc.).

4.) Unmute output

PARAMETERS: - Port number

NOTES: - Any port type can be selected that can carry an RS-232 signal (e.g. TPS, OPT, etc.).

4.5. Infra Category

1.) Switch input to output

PARAMETERS: - Port numbers (input and output port)

NOTES: - Any port type can be selected that can carry an IR signal (e.g. TPS, OPT, local IR port, etc.)

2.) Mute output

PARAMETERS: - Port number

NOTE: - Any port type can be selected that can carry an IR signal (e.g. TPS, OPT, local IR port, etc.)

3.) Unmute output

PARAMETER: - Port number

NOTES: - Any port type can be selected that can carry an IR signal (e.g. TPS, OPT, local IR port, etc.)

4.6. Ethernet Category

1.) Send TCP command

PARAMETERS: - Message
- IP address (destination device)
- Port number (destination device)

NOTES: - Control characters can be inserted in the defined text since the escaping is allowed in this method (the control characters are interpreted).
- See more information in the Sending a Message via a TCP/IP Port section.

2.) Send UDP command

PARAMETERS: - Message
- IP address (destination device)
- Port number (destination device)

NOTES: - See more information in the Sending a Message via a UDP Port section.

3.) Send Wake-on LAN

PARAMETERS: - Target MAC address

NOTES: - The MAC address is separated by double-dots as such: aa:bb:cc:dd:ee:ff. Any other format results in an **Invalid MAC address** error message.

4.7. GPIO Category

1.) Set output state to 'High'

Parameters: - Pin number (GPIO port)

Notes: - To set the status of a GPIO port, it has to be set as **Output**.
- Always check the voltage level and the supported maximum current.
- The default direction is **input** and the level is **High**.

2.) Toggle output state

Parameters: - GPIO pin number

Notes: - To set the status of a GPIO port, it has to be set as **Output**.
- Always check the voltage level and the supported maximum current.
- The default direction is **input** and the level is **High**.

3.) Set output state to 'Low'

Parameters: - GPIO pin number

Notes: - To set the status of a GPIO port, it has to be set as **Output**.
- Always check the voltage level and the supported maximum current.
- The default direction is **input** and the level is **High**.

4.8. EDID Category

1.) Switch EDID

Parameters: - Source EDID (Factory, User, or Dynamic EDID memory)
- Destination EDID (Emulated EDID memory of the input port)

Notes: N/A

4.9. Relay Category

1.) Open contact on relay 1/2

Parameters: N/A

Notes: N/A

2.) Close contact on relay 1/2

Parameters: N/A

Notes: N/A

3.) Toggle contact on relay 1/2

Parameters: N/A

Notes: N/A

4.10. Sending a Message via an RS-232 Port

ATTENTION! The commands mentioned below can be used to send messages, but responses are not received and not processed.

Sending a Message

Path: /MEDIA/UART/<Pn>.sendMessage(<message>)

This is the command listed in **Wizard mode** for sending a message. The command is for sending a text message in ASCII-format, with an option for escaping control charaters (e.g. <CR><LF>).

Escaping in the Message

DEFINITION: Escaping means signaling **Control characters** to be interpreted not as part of the command line, but as fulfilling their own unique function. As such, it can also be used to send many commands at once, like the following:

<command1><\x0d\x0a><command2><\x0d\x0a>

Legend: '\x' = escaping; '0d' = carriage return; '0a' = line feed.

Example

- CALL /MEDIA/UART/P1.sendMessage(PWR0\x0d\x0a)

Sending a Text Message

Path: /MEDIA/UART/<P1>.sendText(<message>)

The command is available in **Advanced mode** and was created for sending a text message in ASCII-format.

Example

- CALL /MEDIA/UART/P1.sendText(PWR0)

ATTENTION! Control characters (e.g. <CR><LF>) are not processed, escaping is not working in this case.

Sending a Binary Message

Path: /MEDIA/UART/<Pn>.sendBinaryMessage(<message>)

The command is available in **Advanced mode** and was created for sending a binary message in HEX format.

ATTENTION! Control characters (e.g. <CR><LF>) are not processed, escaping is not working in this case.

- CALL /MEDIA/UART/P1.sendBinaryMessage(0100000061620000cdcc2c40)

4.11. Sending a Message via a TCP/IP Port

ATTENTION! The commands mentioned below can be used to send messages, but responses are not received and not processed.

Sending a Message

Path:
/MEDIA/ETHERNET.tcpMessage(<IP_address>:<port_nr>=<message>)

This is the command listed in **Wizard mode** for sending a message. The command is for sending a text message in ASCII-format, with an option for escaping control charaters (e.g. <CR><LF>).

Escaping in the Message

DEFINITION: Escaping means signaling **Control characters** to be interpreted not as part of the command line, but as fulfilling their own unique function. As such, it can also be used to send many commands at once, like the following:

<command1><\x0d\x0a><command2><\x0d\x0a>

Legend: '\x' = escaping; '0d' = carriage return; '0a' = line feed.

Example

- CALL /MEDIA/ETHERNET.tcpMessage(192.168.0.20:5555=PWR0\x0d\x0a)

Sending a Text Message

Path:

/MEDIA/ETHERNET.tcpText(<IP_address>:<port_nr>=<message>)

The command is available in **Advanced mode** and was created for sending a text message in ASCII-format.

Example

- CALL /MEDIA/ETHERNET.tcpText(192.168.0.20:5555=pwr_on)

ATTENTION! Escaping is not working in this case.

Sending a Binary Message

Path:
/MEDIA/ETHERNET.tcpBinary(<IP_address>:<port_nr>=<message>)

The command is available in **Advanced mode** and was created for sending a binary message in HEX format.

Example

- CALL /MEDIA/ETHERNET.tcpBinary(192.168.0.20:5555=0100000061620000cdcc2c40)

ATTENTION! Escaping is not working in this case.

4.12. Sending a Message via a UDP Port

> **ATTENTION!** The commands mentioned below can be used to send messages, but responses are not received and not processed.

Sending a Message

Path:
/MEDIA/ETHERNET.udpMessage(<IP_address>:<port_nr>=<message>)

This is the command listed in **Wizard mode** for sending a message. The command is for sending a text message in ASCII-format, with an option for escaping control charaters (e.g. <CR><LF>).

Escaping in the Message

> **DEFINITION: Escaping** means signaling **Control characters** to be interpreted not as part of the command line, but as fulfilling their own unique function. As such, it can also be used to send many commands at once, like the following:

<command1><\x0d\x0a><command2><\x0d\x0a>

Legend: '\x' = escaping; '0d' = carriage return; '0a' = line feed.

Example

- CALL /MEDIA/ETHERNET.udpMessage(192.168.0.20:5555=PWR0\x0d\x0a)

Sending a Text Message

Path:

/MEDIA/ETHERNET.udpText(<IP_address>:<port_nr>=<message>)

The command is available in **Advanced mode** and was created for sending a text message in ASCII-format.

Example

- CALL /MEDIA/ETHERNET.udpText(192.168.0.20:5555=pwr_on)

> **ATTENTION!** Escaping is not working in this case.

Sending a Binary Message

Path:
/MEDIA/ETHERNET.udpBinary(<IP_address>:<port_nr>=<message>)

The command is available in **Advanced mode** and was created for sending a binary message in HEX format.

Example

- CALL /MEDIA/ETHERNET.udpBinary(192.168.0.20:5555=0100000061620000cdcc2c40)

> **ATTENTION!** Escaping is not working in this case.

5

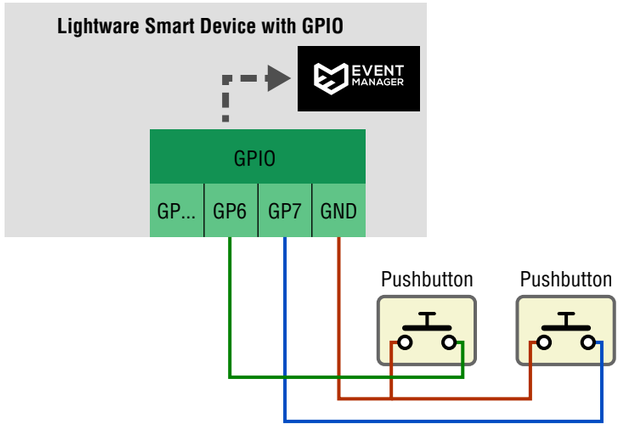
Typical Connected Devices

> **WARNING! The following sections contain third-party devices connected to the control ports of the Lightware devices. Please always check the technical parameters of the device (Voltage/Current) and the port of the Lightware device before connecting. Overloading a port could damage the device.**

- Condition: Using a Pushbutton
- Condition: Using a Motion Sensor
- Condition: IR Signal Detection
- Action: Controlling a Projector/Monitor
- Action: Switching a Power Relay
- Action: Supplying a LED Directly
- Action: Controlling a Motorized Screen (Part #1)
- Action: Controlling a Motorized Screen (Part #2)

5.1. Condition: Using a Pushbutton

DEFINITION: The simple pushbutton below is a dry contact with only one function: closing a circuit.



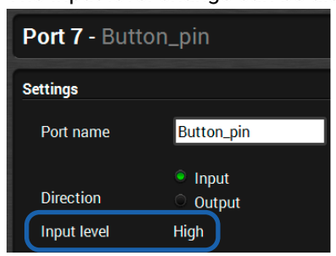
Step 1. Connect the wires to the button panel and the **GPIO port** as seen in the figure. The brown line is the ground, which is the same in both buttons.

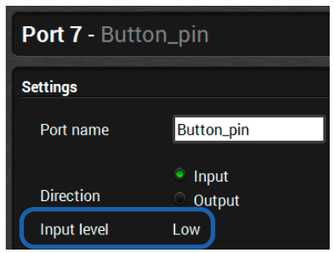
Step 2. Set the **Direction** of the two GPIO pins to **Input**.

Step 3. The default **Input level** of the GPIO pins is **High**. When a button is pressed, the circuit is closed and the given pin's state becomes **Low**. That change can be used as a **Condition** in the **Event Manager**.

Step 4. When the button is released, the level of the pin is changed to **High** again.

The Input level change can be also seen in LDC:



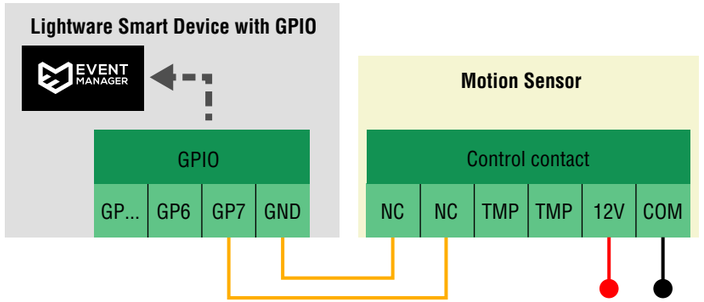


The path of the node that shows the input level:
(L=low, H=high)

- /MEDIA/GPIO/P7.Input

5.2. Condition: Using a Motion Sensor

DEFINITION: The Motion sensor below is a device that keeps the connected circuit closed in default idle state. When the sensor gets activated (Alarm), the circuit is opened.



NC: normally closed; **TMP**: tamper contact

The working method is similar to that of the pushbutton, as the motion sensor can be used to change the **Input level** of a **GPIO pin** (and use it as a **Condition**).

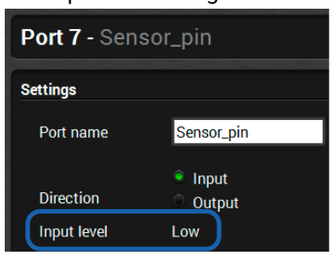
Step 1. Connect the wires to the motion sensor and the **GPIO port** as seen in the figure.

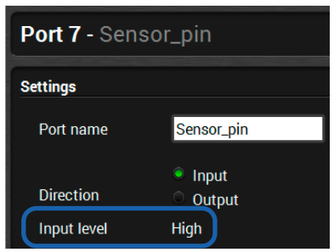
Step 2. Set the **Direction** of the GPIO pin to **Input**.

Step 3. The default **Input level** of the GPIO pins is **High**. The circuit (towards the GPIO port) is closed by the motion sensor by default, so the pin is in **Low** state. When the Motion sensor gets activated (Alarm), the circuit will be opened and the GPIO pin level changes to **High**. That change can be used as a **Condition**.

Step 4. When the circuit is closed by the motion sensor, the level of the pin changes to **Low** again.

The Input level change can be also seen in LDC:



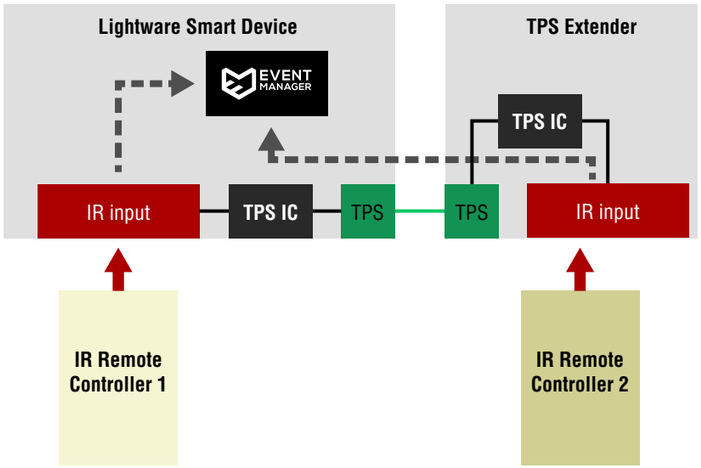


The path of the node that shows the input level: (L=low, H=high)

- /MEDIA/GPIO/P7.Input

5.3. Condition: IR Signal Detection

The example below describes two ways of applying an incoming IR signal as a **Condition**. In both cases the **Condition** is processed in the smart TPS device.



Step 1. Set up the system as seen in the figure.

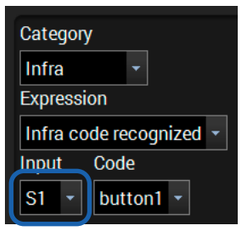
Step 2. Make the **Smart TPS device** learn the desired IR code (navigate to Control/Infra tab in LDC).

Step 3. Set the **Repeat timeout** and a **Name** for the code.

Step 4. The saved code can be used as a **Condition** (the name of the IR code is listed in the list of the parameters at the **Event Manager**).

Local IR Port – TPS IR Port

Since the TPS connection allows transmitting IR signals, the **Condition** can be set to sense the **local** or the **TPS IR port**. In the example above, the **RC1** sends IR signal to the **local port** of the **Smart TPS device**, and the RC2 sends IR signal to the IR receiver connected to the IR input port of the **TPS extender**. In this case the IR signal of the **RC2** is transmitted from the **TPS extender** via the TPS connection to the **Smart TPS device**. When the **Condition** is set in the **Event Manager**, the available IR ports are listed, see the attached figure.



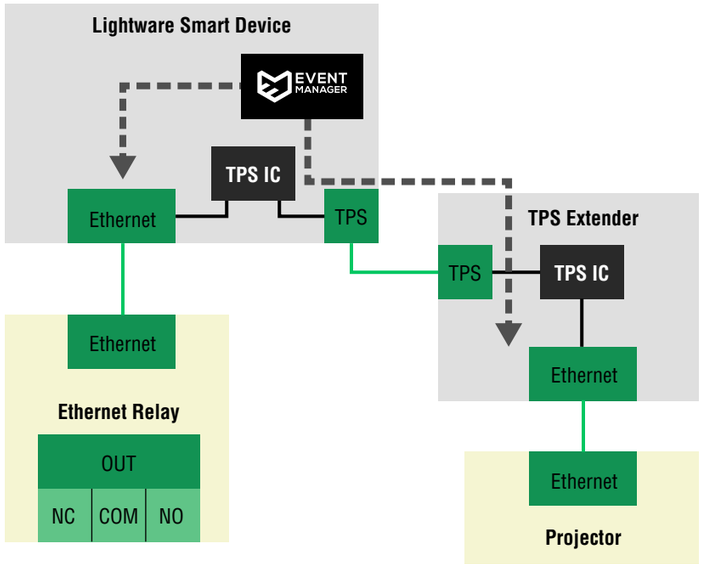
Infra Code Repeat Timeout (ms) parameter

Periodically received identical IR codes are recognized once in a timeout period. Remote controls can send identical IR commands periodically when a button is pressed and held. E.g. an 'ON/OFF' command and a 'Volume +/-' command require different repeat timeout values.

> INFO: Not the entire IR code is stored in the devices, just a generated hash code, which is enough to identify the original code.

5.4. Action: Controlling a Projector/Monitor

5.4.1. Controlling via an Ethernet Port



NO: normally open; **NC**: normally closed

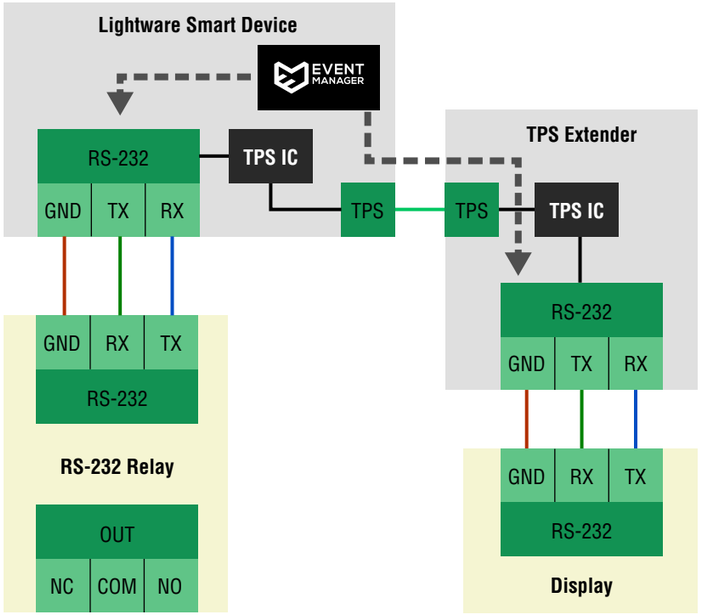
The example above contains two Ethernet devices:

- The **Relay** device connected to the local Ethernet port (P1), and
- The **Display** device connected to the TPS Ethernet port (P3).

The **Smart TPS device** is able to send TCP messages to both devices via the local and the TPS Ethernet ports (Ethernet signal is also transmitted via the TPS connection). The message sending works by using the **Event Manager** as well.

Please see the Sending a Message via a TCP/IP Port section for more information.

5.4.2. Controlling via an RS-232 Serial Port



NO: normally open; **NC**: normally closed

The example above includes two serial devices:

- The **Relay** device connected to the local RS-232 port (P1), and
- The **Display** device connected to the TPS RS-232 port (P2).

The Smart TPS device is able to send messages to both devices via the local and the TPS RS-232 ports (RS-232 signal is also transmitted via the TPS connection).

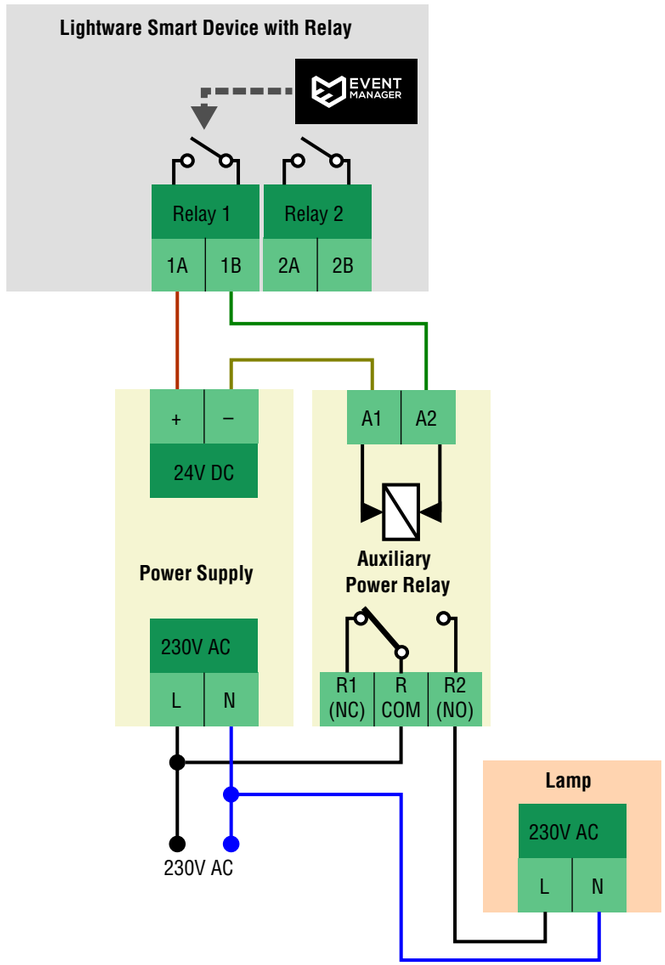
Pay attention to the serial port settings:

- Baud rate, Data bits, Stop bits, Parity
- Control mode
- P1 is the local, P2 is the TPS RS-232 port.

Please see the Sending a Message via an RS-232 Port section for more information.

5.5. Action: Switching a Power Relay

The Relay port of a Lightware device can be used to connect and control a Power Relay. A typical example can be seen in the figure below about how to switch on a lamp:

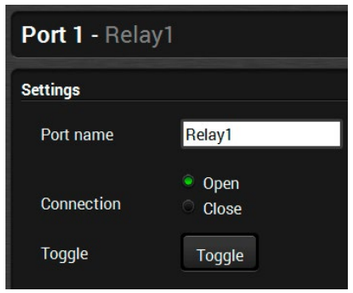


NO: normally open; **NC**: normally closed

Step 1. Setup the circuit as seen in the figure. Connect the wires to the **Relay port**.

Step 2. The default **Connection** of the **Relay ports** is **Open**. When the circuit is **Closed,** the lamp is powered on. This can be used as an **Action**.

The connection state can be also set in LDC:



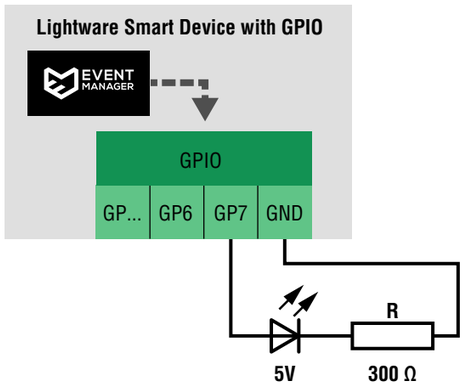
The path of the node that controls the connection state: (C=close, O=open)

- /MEDIA/RELAY/P1.Output

5.6. Action: Supplying a LED Directly

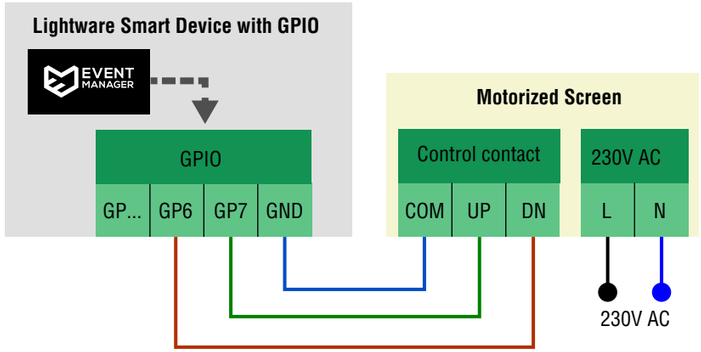
The technical structure of the GPIO port allows supplying simple devices working at low power consumption. In the example below a +5V LED is connected to a GPIO pin.

ATTENTION! A GPIO pin can supply at most 30mA (5V) and the total load of the seven GPIO pins must not be more than 180mA. Always check the technical parameters of the third-party device to avoid overload.



5.7. Action: Controlling a Motorized Screen (Part #1)

The **GPIO port** of a Lightware device can be used to connect and control a Motorized Projection Screen. In this example the GPIO pins will be used as **Outputs**. A typical example can be seen in the figure below:



Rolling the Screen

The Motorized Screen can be controlled via the COM|UP|DN contact pins as follows:

- **The screen rolls up:** if the **UP** and **COM** pins are connected (**DN** is not connected to **COM**).
- **The screen rolls down:** if the **DN** and **COM** pins are connected (**UP** is not connected to **COM**).

Step 1. Setup the circuit as seen in the figure.

Step 2. Set the **Direction** of the GP6 and GP7 pins to **Output**.

Step 3. The default **Output level** of the GPIO pins is **High**. The screen control can be arranged by setting the GP6 and GP7 pins as **Actions** by the **Event Manager,** set as follows:

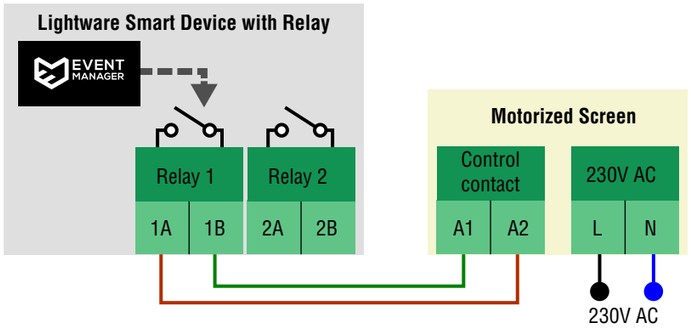
	GP6 level is low	**GP6 state is high**
GP7 level is low	-	screen rolls up
GP7 level is high	screen rolls down	-

The path of the node that sets the output level: (L=low, H=high)

- /MEDIA/GPIO/P7.Output

5.8. Action: Controlling a Motorized Screen (Part #2)

The **Relay port** of a Lightware device can be also used to connect and control a Motorized Projection Screen. A typical example can be seen in the figure below:



Rolling the Screen

The projection screen can be controlled via the Control contact pins as follows:

- **Roll down: A1** and **A2** pins are connected (the **Relay** is **Closed**).
- **Roll up: A1** and **A2** pins are not connected (the **Relay** is **Open**).

Step 1. Set up the system as seen in the figure.

Step 2. Connect the wires to the **Relay port**.

Step 3. The default **Connection** of the **Relay ports** is **Open**. When the **Relay1** port is **Closed,** the screen rolls down. If the **Relay1** port is **Open,** the screen rolls up. These can be used as **Actions**.

6

Typical Applications – Example Descriptions

The first chapter contains two simple applications of how the Event Manager can be used in practice. This chapter contains the details of how to connect and setup the devices:

- Details of Example 1
- Details of Example 2

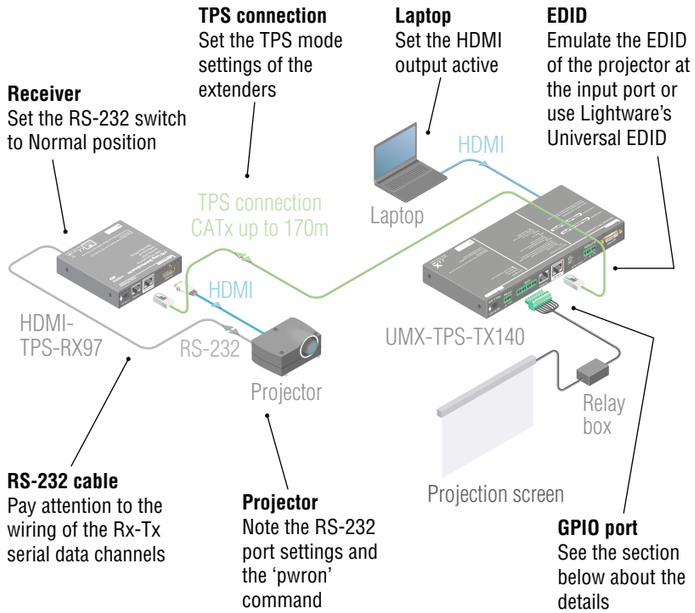
6.1. Details of Example 1

The Desired Working Method

If signal is detected on the HDMI input port of the transmitter, then it

- transmits the signal to the receiver,
- switches on the projector, and
- rolls down the projection screen.

Make sure that the desired ports are unmuted, unlocked and the port parameters are set properly – see also the figure below:



The application above contains a typical example of how to connect a projection screen and control via the GPIO port. For that kind of setup, a relay box is necessary, which is installed between the projection screen and the transmitter.

To roll up/down, the two projection screen pins have to be controlled, that is why GP6 and GP7 are connected to the Relay box. Both have to be in the indicated state to roll the screen.

GPIO Port Wiring and Settings

Pin nr.	1	2	3	4	5	6	7	Ground
Wired to	-	-	-	-	Relay box			
Function					Relay box 5V power	Screen control 1	Screen control 2	Ground
Pin direction					output	output	output	-
Roll up the screen					-	high	low	-
Roll down the screen					-	low	high	-

Conditions and Actions

You do not have to set the **Conditions** at each **Event** separately, only at one **Event**, then just **Link** the **Condition** as described in Linking a Condition or an Action section.

Nr.	Trigger	Condition	Action	What Happens
E1	A/V signal is connected to the HDMI input port	Signal is detected on I2	Set GPIO output state to 'Low' on P6	Screen is rolled down
E2			Set GPIO output state to 'High' on P7	
E3			Switch video input I2 to output O1	Switch HDMI to TPS output
E4			Send RS-232 message 'pwron' on P2	Switch on the Projector
E5	A/V signal is not detected	Video signal is not detected on O1	Set GPIO output state to 'Low' on P7	Screen is rolled up
E6			Set GPIO output state to 'High' on P6	
E7			Send RS-232 message 'pwroff' on P2	Switch off the Projector

Delaying the Action

To avoid an unwanted system switch off, apply the **Delay** option at E5-E7 Events (e.g. **Continuously exist**, 1 minute); see also in the Testing the Action section.

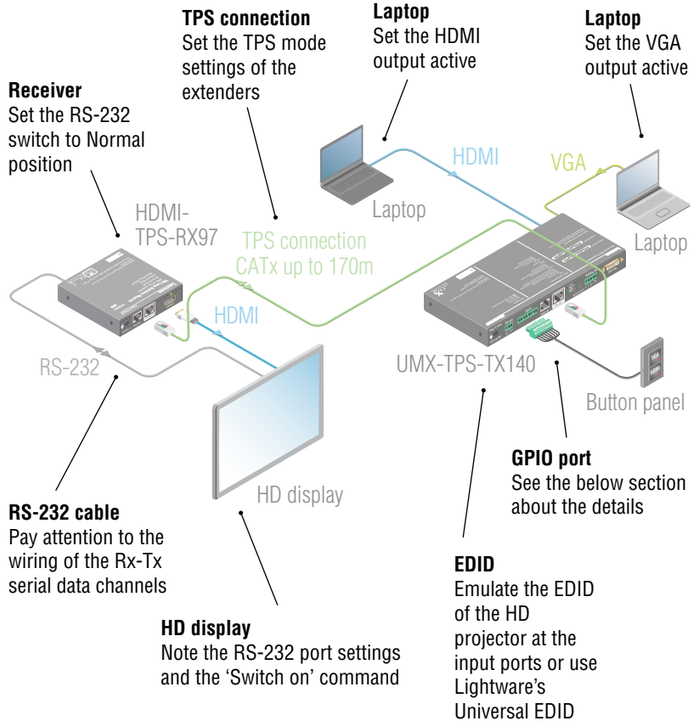
6.2. Details of Example 2

The Desired Working Method

If a button is pressed on the button panel

- transmit the incoming signal to the receiver,
- switch on the given LED on the button panel,
- switch on the HD display.

Make sure that the desired ports are unmuted, unlocked and the port parameters are set properly – see also the figure below:



GPIO Port Wiring and Settings

Pin nr.	1	2	3	4	5	6	7	Ground
Wired to	Button panel				-	-	-	Button panel
Function	VGA LED	VGA button	HDMI LED	HDMI button				Ground
Pin direction	output	input	output	input				-
High	LED lights	default	LED lights	default				-
Low	LED is dark	input select	LED is dark	input select				-

Conditions and Actions

You do not have to set the **Conditions** at each **Event** separately, only at one **Event**, then just **Link** the **Condition** as described in the Linking a Condition or an Action section.

Nr.	Trigger	Condition	Action	What Happens
E1	VGA button is pressed	GPIO state changes to 'Low' on P2	Switch video input I1 to output O1	Switch VGA to TPS output
E2			Set GPIO output state to 'High' on P1	Switch on the VGA LED
E3			Set GPIO output state to 'Low' on P3	Switch off the HDMI LED
E4			Send RS-232 message 'pwron' on P2	Switch on the HD display
E5	HDMI button is pressed	GPIO state changes to 'Low' on P4	Switch video input I2 to output O1	Switch HDMI to TPS output
E6			Set GPIO output state to 'High' on P3	Switch on the HDMI LED
E7			Set GPIO output state to 'Low' on P1	Switch off the VGA LED
E8			Send RS-232 message 'pwron' on P2	Switch on the HD display
E9	No signal is detected	Video signal is not detected on O1	Send RS-232 message 'pwroff' on P2	Switch off the HD display
E10			Set GPIO output state to 'Low' on P1	Switch off the VGA LED
E11			Set GPIO output state to 'Low' on P3	Switch off the HDMI LED

Delaying the Action

To avoid an unwanted system switch off, apply the **Delay** option at E9-E11 Events (e.g. **Continuously exist**, 1 minute); see also in the Testing the Action section.