

Application Notes

Event Manager

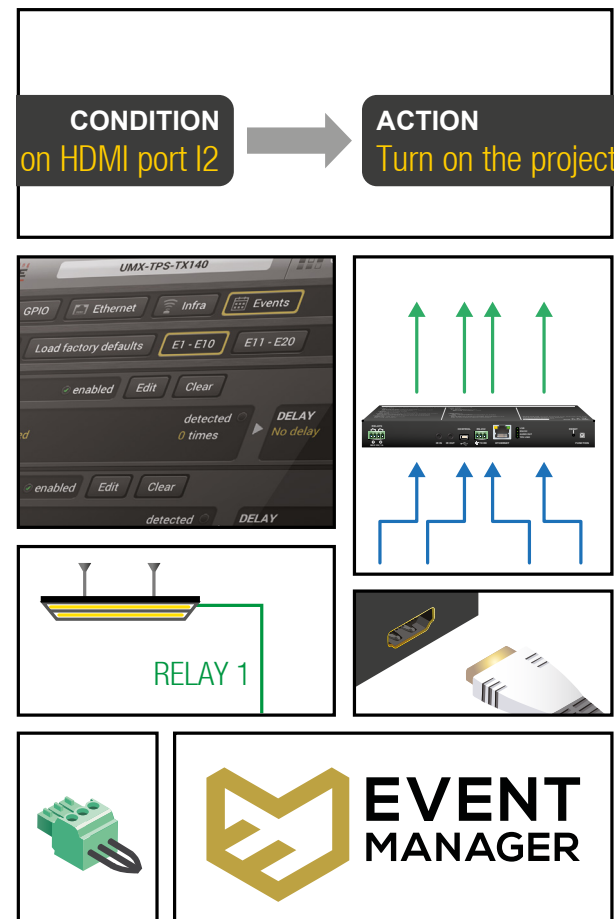


Table of Contents

1. INTRODUCTION	3	5.5. ACTION: SWITCHING A POWER RELAY	19
1.1. DESCRIPTION	3	5.6. ACTION: SUPPLYING A LED DIRECTLY	19
1.2. DEFINITION.....	3	5.7. ACTION: CONTROLLING A MOTORIZED SCREEN (PART #1).....	19
1.3. SUPPORTED DEVICES	3	5.8. ACTION: CONTROLLING A MOTORIZED SCREEN (PART #2).....	20
1.4. TYPICAL APPLICATIONS	4	6. TYPICAL APPLICATIONS – EXAMPLE DESCRIPTIONS	21
2. EVENT MANAGER – WHERE IS IT?.....	5	6.1. DETAILS OF EXAMPLE 1	21
2.1. THE EVENTS TAB	5	6.2. DETAILS OF EXAMPLE 2	22
2.2. ADDING AN EVENT – THE EVENT EDITOR	6		
2.3. USEFUL TOOLS	10		
2.4. FURTHER FEATURES.....	10		
3. THE CONDITION – EXPRESSIONS AND PARAMETERS.....	11		
3.1. GENERAL CATEGORY	11		
3.2. VIDEO CATEGORY	11		
3.3. AUDIO CATEGORY	12		
3.4. INFRA CATEGORY	12		
3.5. GPIO CATEGORY	12		
4. THE ACTION - EXPRESSIONS AND PARAMETERS	13		
4.1. GENERAL CATEGORY	13		
4.2. VIDEO CATEGORY	13		
4.3. AUDIO CATEGORY	14		
4.4. RS-232 CATEGORY.....	14		
4.5. INFRA CATEGORY	15		
4.6. ETHERNET CATEGORY	15		
4.7. GPIO CATEGORY	15		
4.8. EDID CATEGORY	15		
4.9. RELAY CATEGORY.....	15		
4.10. SENDING A MESSAGE VIA AN RS-232 PORT	16		
4.11. SENDING A MESSAGE VIA A TCP/IP PORT	16		
4.12. SENDING A MESSAGE VIA A UDP PORT	16		
5. TYPICAL CONNECTED DEVICES.....	17		
5.1. CONDITION: USING A PUSHBUTTON.....	17		
5.2. CONDITION: USING A MOTION SENSOR	17		
5.3. CONDITION: IR SIGNAL DETECTION	18		
5.4. ACTION: CONTROLLING A PROJECTOR/MONITOR.....	18		

Document Information

Document revision: 1.2

Release date: 21-08-2018

Editor: Laszlo Zsedenyi

Contact Us

sales@lightware.com

+36 1 255 3800

support@lightware.com

+36 1 255 3810

Lightware Visual Engineering LLC.

Peterdy 15, Budapest H-1071, Hungary

www.lightware.com

©2018 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(TM) 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](#)

1.1. 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

Currently ² the following products include Event Manager:

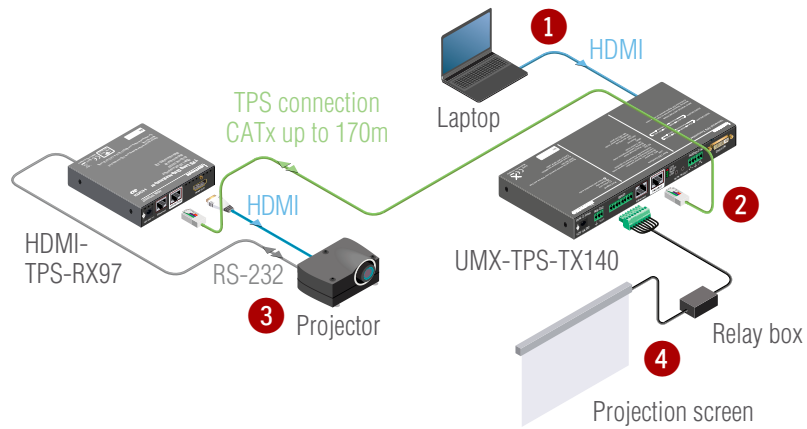
Device Type(s)	Supported Number of the Events
UMX-TPS-TX120/130/140	20
MMX6x2-HT200/210/220	100
MMX4x2-HDMI/HT200	100
UMX-HDMI-140	20
HDMI-TPS-TX210/TX220	20
HDMI-TPS-RX110AY	100
SW4-TPS-TX240	20
SW4-OPT-TX240RAK	20
HDMI-3D-OPT-TX210A/TX210RAK	20
WP-UMX-TPS-TX120-US/130-US	20
DVI-HDCP-TPS-TX210/TX220	20
DP-TPS-TX210/TX220	20
MODEX	32

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

² Currently: on the day of publishing this document.




1.4. Typical Applications

Example 1



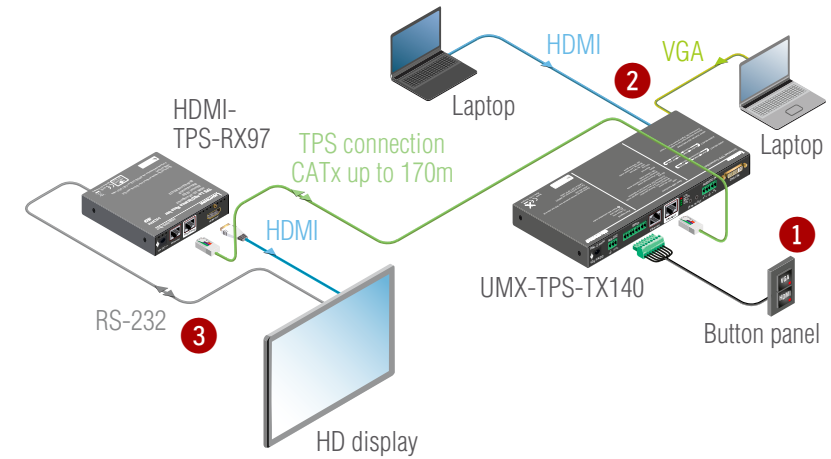
In the first example if a signal is detected on the HDMI input port of the UMX-TPS-TX140, the listed actions are launched automatically:

- 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

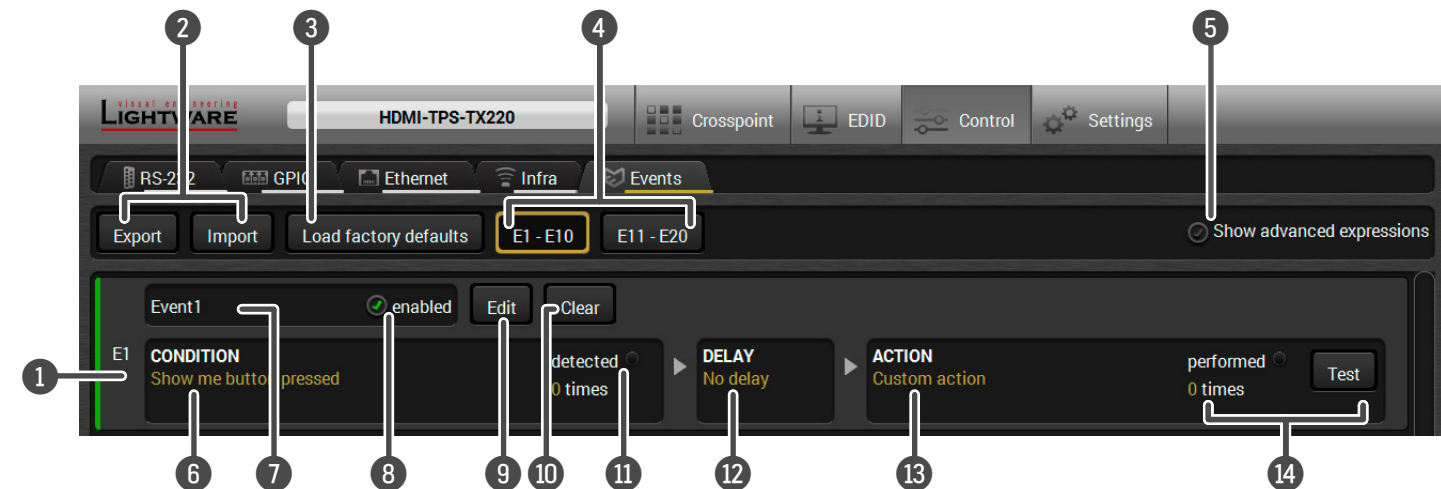
Event Manager – Where is it?

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.

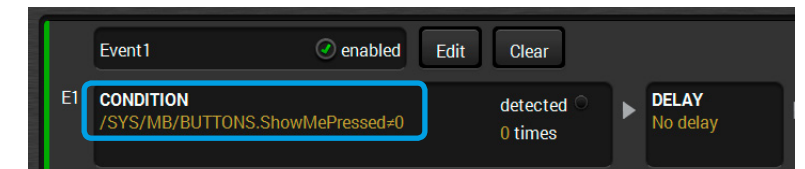
- ▶ [THE EVENTS TAB](#)
- ▶ [ADDING AN EVENT – THE EVENT EDITOR](#)
- ▶ [USEFUL TOOLS](#)
- ▶ [FURTHER FEATURES](#)

2.1. The Events Tab

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



- | | | | |
|---|---|---|---|
| <p>① Event Lines</p> <p>② Export and Import Buttons</p> <p>③ Factory Defaults</p> <p>④ Event Pages</p> <p>⑤ Show Advanced Expressions</p> <p>⑥ Condition</p> <p>⑦ The Name of the Event</p> <p>⑧ Switch</p> | <p>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.</p> <p>The Events (with all their settings) can be saved into a file and can be imported. See more details in the Import / Export section.</p> <p>All the Events can be deleted by pressing this button (a confirmation window will pop up before the deleting).</p> <p>10 pieces of Events are listed at once.</p> <p>Toggle the display mode of the Conditions and Actions shown in the list (see below).</p> <p>Displays the expression shown in Wizard mode or the exact LW3 path and node.</p> <p>It can be edited by the user in the Event Editor. See also The Name of the Event section.</p> <p>The Event can be enabled or disabled.</p> | <p>⑨ Edit Button</p> <p>⑩ Clear Button</p> <p>⑪ Condition Test</p> <p>⑫ Delay Settings</p> <p>⑬ Action</p> <p>⑭ Action Test</p> | <p>Press the button to open the Event Editor and set all the parameters.</p> <p>Delete the settings of the given Event.</p> <p>If the Condition is detected, the green indicator is lit for three seconds and the counter is increased. See also the Testing the Settings section.</p> <p>The Action can be scheduled to follow the Condition after the set time value. See also the Delaying the Action after Detecting the Condition section.</p> <p>Displays the expression shown in Wizard mode or the exact LW3 path and node.</p> <p>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.</p> |
|---|---|---|---|



2.2. Adding an Event – the Event Editor

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

The screenshot shows the Event Editor window with the following components and callouts:

- 1 Event Header:** Shows the event name 'E9', a status 'enabled' with a green checkmark, a 'Clear' button, a text input field containing 'Sig_det', and a 'Set name' button.
- 2 Condition Header:** Displays the condition description 'Video signal is detected on I2' in white text and the LW3 protocol expression '/MEDIA/VIDEO/I2.SignalPresent=1' in yellow text.
- 3 Condition Panel:** Contains three tabs: 'Wizard' (selected), 'Advanced', and 'Link'. Below the tabs are fields for 'Category' (set to 'Video'), 'Expression' (set to 'Signal is detected on a port'), and 'Port' (set to 'I1'). There are 'Clear' and 'Apply' buttons at the bottom.
- 4 Condition Test:** A panel with a 'Counter: 0 times' and a 'Reset' button. Below it, a radio button labeled 'detected' is selected.
- 5 Delay Settings:** A 'DELAY' section with a 'No delay' status, a dropdown set to 'Simple delay', a numeric input '0', and a unit selector set to 'min 30 sec'. An 'Apply' button is at the bottom.
- 6 Action Test:** A panel with a 'Counter: 0 times' and a 'Reset' button. Below it, a radio button labeled 'performed' is selected, and a 'Test action' button is present.
- 7 Action Panel:** Contains three tabs: 'Wizard' (selected), 'Advanced', and 'Link'. Below the tabs are a tree view for 'Node' (showing /MEDIA/VIDEO/XP) and a 'Property' section with 'muteSource()' and a 'Value' input field containing 'I2'. There are 'Clear' and 'Apply' buttons at the bottom.
- 8 Action Header:** Shows the action description 'Custom action' in white text and the LW3 protocol expression '/MEDIA/VIDEO/XP.muteSource=I1' in yellow text.

INFO: The Condition test, 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. Use 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 but only in **Advanced** mode.
- 3 Condition Panel** The **Wizard**, the **Advanced** or the **Link** tool is available to set the condition. The parameters and settings are displayed below the buttons.
- 4 Condition Test** The selected **Condition** can be tested to see if it works in practice.
- 5 Delay Settings** The **Action** can be scheduled to follow the **Condition** after the set time value.
- 6 Action Test** The set **Action** can be tested to see the working method in the practice.
- 7 Action Panel** The **Wizard**, the **Advanced** or the **Link** tool is available to set the **Action**. The parameters and settings are displayed below the buttons.
- 8 Action Header** If the **Action** is set, the description (white colored text) and the exact LW3 protocol expression (yellow colored text) can be seen. **Custom Action** means the entry is not available in **Wizard** mode but only in **Advanced** mode.

2.2.1. The Wizard Interface

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

The screenshot displays the Event Manager Wizard interface, which is divided into several sections for configuring an event:

- Event Editor:** At the top, it shows the event is **ET enabled** with a green checkmark. There are **Clear** and **Set name** buttons, and a text input field containing **A_btn**.
- CONDITION:** The left panel shows the condition settings. The **Wizard** tab is selected. The condition is **Video signal type changes to HDMI on I2 /MEDIA/VIDEO/I2.SignalType=1**. The **Category** is **Video**, the **Expression** is **Signal type changes to HDMI**, and the **Port** is **I1**. There are **Clear** and **Apply** buttons.
- ACTION:** The right panel shows the action settings. The **Wizard** tab is selected. The action is **Send RS-232 message 'PWR0\x0d\x0a' on P1 /MEDIA/UART/P1.sendMessage=PWR0\x0d\x0a**. The **Category** is **RS-232**, the **Expression** is **Send RS-232 message**, the **Port** is **P2**, and the **Message** is **PWR0\x0d\x0a**. There are **Clear** and **Apply** buttons.
- CONDITION TEST:** At the bottom left, it shows a counter of **0 times** and a **Reset** button. The status is **detected** with a radio button.
- DELAY:** In the center, it shows **No delay** and a **Simple delay** dropdown. The delay is set to **0 min 20 sec**. There is an **Apply** button.
- ACTION TEST:** At the bottom right, it shows a counter of **0 times** and a **Reset** button. The status is **performed** with a radio button. There is a **Test action** button.

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.

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.

The screenshot displays the Event Manager interface in Advanced mode. At the top, an event named 'Event_1' is shown as enabled. Below this, the interface is split into two main sections: **CONDITION** and **ACTION**.

CONDITION Panel: Shows the configuration for a condition. The 'Node' is set to '/SYS/MB/BUTTONS' and the 'Property' is 'ShowMePressed'. The 'Operator' is set to 'not equal (≠)' and the 'Value' is '1'. The 'Property' dropdown also shows a description: '[<number of button press events>]'. The 'Node' tree on the left shows a path from /SYS/MB/RS232.

ACTION Panel: Shows the configuration for an action. The 'Node' is set to '/MEDIA/UART/P2' and the 'Property' is 'Rs232Mode'. The 'Value' is set to '1'. The 'Property' dropdown shows a description: '["0" | "1" | "2"] Rs232 operation mode (0=Pass; 1=Control; 2=CommandInjection)'. The 'Node' tree on the left shows a path from /MEDIA/UART/P1.

At the bottom, there are three test panels: **CONDITION TEST** (Counter: 0 times, detected), **DELAY** (No delay, 0 min 0 sec), and **ACTION TEST** (Counter: 0 times, performed). Arrows indicate the flow from the Condition and Delay panels to the Action Test panel.

INFO: The **Wizard** and the **Advanced** modes can be used simultaneously; if the **Condition** is set in **Wizard** mode the **Action** can be set in **Wizard** or **Advanced** mode also 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 many feedbacks 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** which was defined at another **Event** previously, therefore:

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

The screenshot displays the 'Link' interface in the Event Manager. At the top, 'E1' is enabled and named 'Event1'. Below are two main panels: 'CONDITION' and 'ACTION', each with 'Wizard', 'Advanced', and 'Link' tabs. The 'Link' tab is active in both. Below each panel is a table of available events to link to.

CONDITION Panel:

- Condition: Show me button pressed /SYS/MB/BUTTONS.ShowMePressed#0
- Table:

ID	Event name	Condition
E6	Event6	Infra code recognized on S1

ACTION Panel:

- Action: Custom action /MEDIA/UART/P2.Rs232Mode=0
- Table:

ID	Event name	Action
E2	Event2	Send RS-232 message 'PWR_OFF' on P1
E3	Event3	/MEDIA/UART/P2.Rs232Mode=2

At the bottom, there are three test panels: 'CONDITION TEST' (Counter: 0 times, detected), 'DELAY' (No delay, 0 min 0 sec), and 'ACTION TEST' (Counter: 0 times, performed). Arrows indicate the flow from the Condition and Delay panels to the Action Test panel.

This helps a lot when the **Action** or **Condition** which was used multiple time have 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.

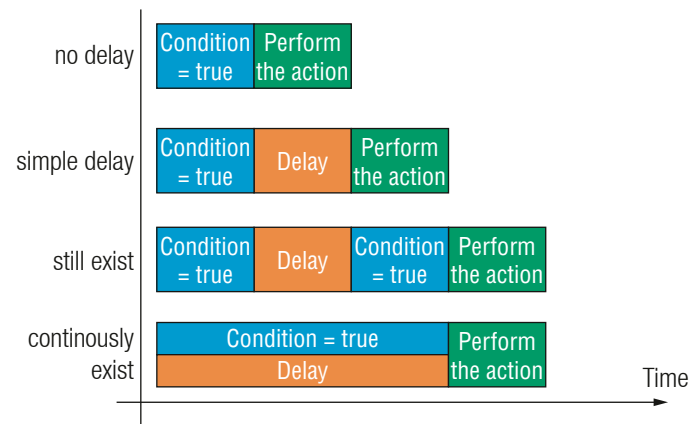
ATTENTION! Do not forget to press the **Apply** button when the linked **Condition/Action** is arranged.

2.3. Useful Tools

2.3.1. Delaying the Action after Detecting the Condition

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 exists:** 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 exists:** After the Condition was detected the Condition is checked continuously through the Delay time. If it is continuously existing the Action is launched.

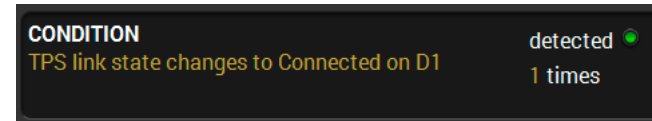


2.3.2. Testing the Settings

If you have created a Condition or an Action you can test them. The feature is available at the list of Events and in the Event Editor window as well.

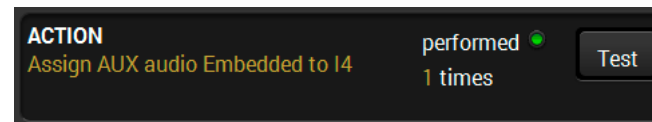
Testing a Condition

The counter displays the number of detections which can be reset to zero. If you trigger the **Condition**, the **detected** indicator is lit for three seconds and the counter is increased.



Testing an Action

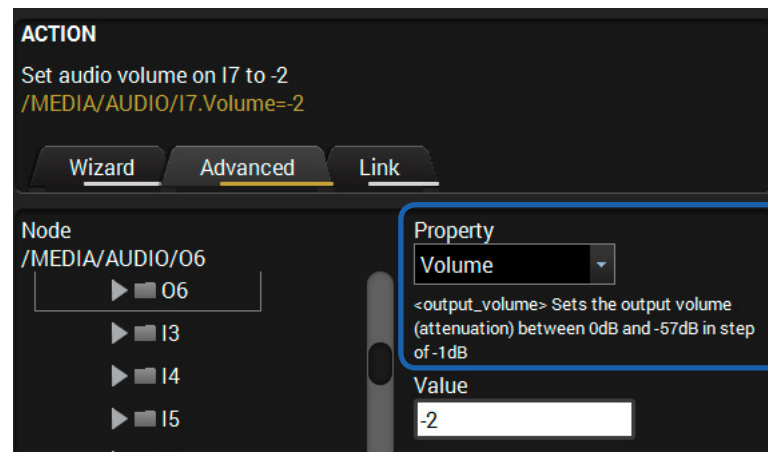
The counter works the same as at the **Condition** test, but in this case the **Action** itself can be tested by pressing the **Test** button.



2.4. Further Features

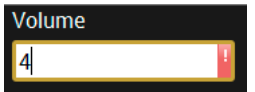
Displaying the Manual of a Property

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):



Incorrect Values

If you typed an incorrect value you will get a notice by displaying an exclamation mark.



The Name of the Event

The name of an Event can be set by typing the new name and clicking 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 ().

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 valid for the software and firmware components. Therefore, the successful working of an exported/imported list of **Events** is guaranteed only in the same type of device and running the same Firmware version. Certain **Events** may be applied in different type of devices but this is not guaranteed.

The import/export feature is available on the **Control/Events** tab:



INFO: The exported file can be edited by a simple text editor e.g. Notepad, but this is recommended only for expert users.


3

The Condition – Expressions and Parameters

This chapter describes the Conditions which 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 some 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

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

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

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

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

NOTE: - 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

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

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

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

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

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

5.) Show me button pressed

PARAMETER: N/A

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

6.) Function button pressed

PARAMETER: N/A

NOTE: - 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

PARAMETER: - Port number (input or output)

NOTES: - Any port type can be selected which can carry a video signal (e.g. HDMI, DVI, TPS, OPT, etc.).
- This **Condition** is about to check 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

PARAMETER: - Port number (input or output)

NOTES: - Any port type can be selected which can carry a video signal (e.g. HDMI, DVI, TPS, OPT, etc.).
- This **Condition** is about to check 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

PARAMETER: - Port number (output)

NOTE: N/A

4.) Signal type changes to HDMI

PARAMETER: - Port number (output)

NOTE: N/A

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

PARAMETER: - Port number (output)

NOTE: 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

PARAMETER: - Port number (input or output)

NOTE: N/A

2.) Signal is not detected on a port

PARAMETER: - Port number (input or output)

NOTE: N/A

3.) Signal type changes to PCM

PARAMETER: - Port number (output)

NOTE: N/A

4.) Signal type changes to Compressed

PARAMETER: - Port number (output)

NOTE: - Check the desired port specifications about the supported audio signal types.

5.) Signal type changes to HBR

PARAMETER: - Port number (output)

NOTE: - Check the desired port specifications about the supported audio signal types.

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

PARAMETER: - Port number (output)

NOTE: 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'

PARAMETER: - Pin number (GPIO port)

NOTES:

- To sense the status of a GPIO port, it has to be set as **Input**. Nevertheless, the state is 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'

PARAMETER: - Pin number (GPIO port)

NOTES:

- To sense the status of a GPIO port, it has to be set as **Input**. However the state is shown when it is configured as output as well.

- 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 which 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 **Conditions**. Some of the **Conditions** are available only at certain devices.

4.1. General Category

1.) Turn off LCD backlight

PARAMETER: N/A

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

2.) Turn on LCD backlight

PARAMETER: N/A

NOTE: N/A

3.) Blink LCD backlight

PARAMETER: N/A

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

PARAMETER: - Duration time (x10 ms)

NOTE: 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 which 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

PARAMETER: 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

PARAMETER: - Port number (output)

NOTE: N/A

4.) Disable autoselect on output

PARAMETER: - Port number (output)

NOTE: N/A

5.) Load crosspoint preset

PARAMETER: - Preset name

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

6.) Mute output

PARAMETER: - Port number (output)

NOTE: N/A

7.) Unmute output

PARAMETER: - Port number (output)

NOTE: N/A

8.) Mute input

PARAMETER: - Port number (input)

NOTE: N/A

9.) Unmute input

PARAMETER: - Port number (input)

NOTE: N/A

4.3. Audio Category

1.) Set volume

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

NOTES: - 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

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

3.) Decrease volume

PARAMETERS: - Port number (output)
- Percent value

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

4.) Mute output

PARAMETER: - Port number (output)

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

5.) Unmute output

PARAMETER: - Port number (output)

NOTE: N/A

6.) Mute input

PARAMETER: - Port number (input)

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

7.) Unmute input

PARAMETER: - Port number (input)

NOTE: N/A

8.) Mute analog output

PARAMETER: - Port number (output)

NOTE: N/A

9.) Unmute analog output

PARAMETER: - Port number (output)

NOTE: N/A

10.) Mute HDMI output embedded audio

PARAMETER: - Port number (output)

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

11.) Unmute HDMI output embedded audio

PARAMETER: - Port number (output)

NOTE: N/A

12.) Change audio assignment

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

NOTE: N/A

13.) Switch input to output

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

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

14.) Switch next input to output

PARAMETER: 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

PARAMETER: - Port number (output)

NOTE: N/A

16.) Disable autoselect on output

PARAMETER: - Port number (output)

NOTE: N/A

4.4. RS-232 Category

1.) Send RS-232 message

PARAMETERS: - Port number
- Message text

NOTES: - Any port type can be selected which 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 [Sending a Message via an RS-232 Port](#) section.

2.) Switch input to output

PARAMETERS: - Port numbers (input and output port)

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

3.) Mute output

PARAMETER: - Port number

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

4.) Unmute output

PARAMETER: - Port number

NOTE: - Any port type can be selected which 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)

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

2.) Mute output

PARAMETER: - Port number

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

3.) Unmute output

PARAMETER: - Port number

NOTE: - Any port type can be selected which 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 [Sending a Message via a TCP/IP Port](#) section.

2.) Send UDP command

PARAMETERS: - Message

- IP address (destination device)
- Port number (destination device)

NOTE: N/A

4.7. GPIO Category

1.) Set output state to 'High'

PARAMETER: - 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

PARAMETER: - 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'

PARAMETER: - 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)

NOTE: N/A

4.9. Relay Category

1.) Open contact on relay 1/2

PARAMETER: N/A

NOTE: N/A

2.) Close contact on relay 1/2

PARAMETER: N/A

NOTE: N/A

3.) Toggle contact on relay 1/2

PARAMETER: N/A

NOTE: N/A

4.10. Sending a Message via an RS-232 Port

ATTENTION! The below mentioned commands 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 characters (e.g. <CR><LF>).

Escaping in the Message

DEFINITION: Indicating (highlighting) **Control characters** (e.g. Carriage return, Line feed) in a command means the **Escaping**. That also helps sending 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 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, the 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 created for sending a binary message in HEX format.

ATTENTION! Control characters (e.g. <CR><LF>) are not processed, the 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 below mentioned commands 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 characters (e.g. <CR><LF>).

Escaping in the Message

DEFINITION: Indicating (highlighting) **Control characters** (e.g. Carriage return, Line feed) in a command means the **Escaping**. That also helps sending 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 created for sending a text message in ASCII-format.

Example

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

ATTENTION! The 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 created for sending a binary message in HEX format.

Example

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

ATTENTION! The escaping is not working in this case.

4.12. Sending a Message via a UDP Port

ATTENTION! The below mentioned commands 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 characters (e.g. <CR><LF>).

Escaping in the Message

DEFINITION: Indicating (highlighting) **Control characters** (e.g. Carriage return, Line feed) in a command means the **Escaping**. That also helps sending 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 created for sending a text message in ASCII-format.

Example

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

ATTENTION! The 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 created for sending a binary message in HEX format.

Example

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

ATTENTION! The escaping is not working in this case.

5

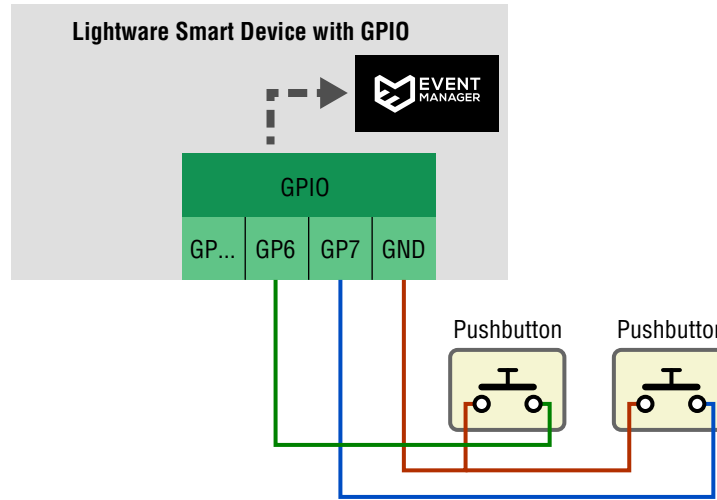
Typical Connected Devices

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 \(PART #2\)](#)

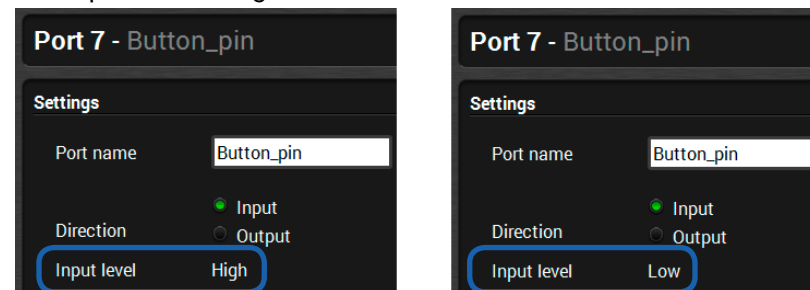
5.1. Condition: Using a Pushbutton

DEFINITION: The below simple pushbutton 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 common for 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 got **Low** state. 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:

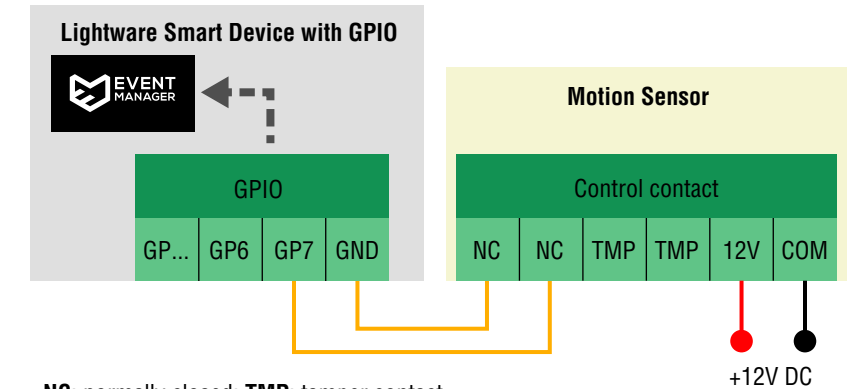


The path of the node that shows the input level:
(L=low, H=high)

> /MEDIA/GPIO/P7.Input

5.2. Condition: Using a Motion Sensor

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

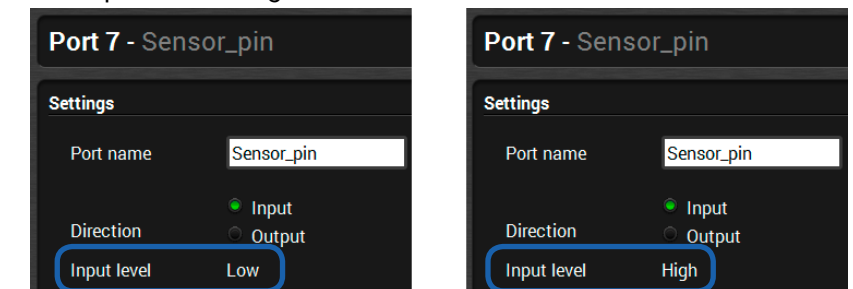


NC: normally closed; TMP: tamper contact

The working method is similar like in the case 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 as default so the pin got **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:

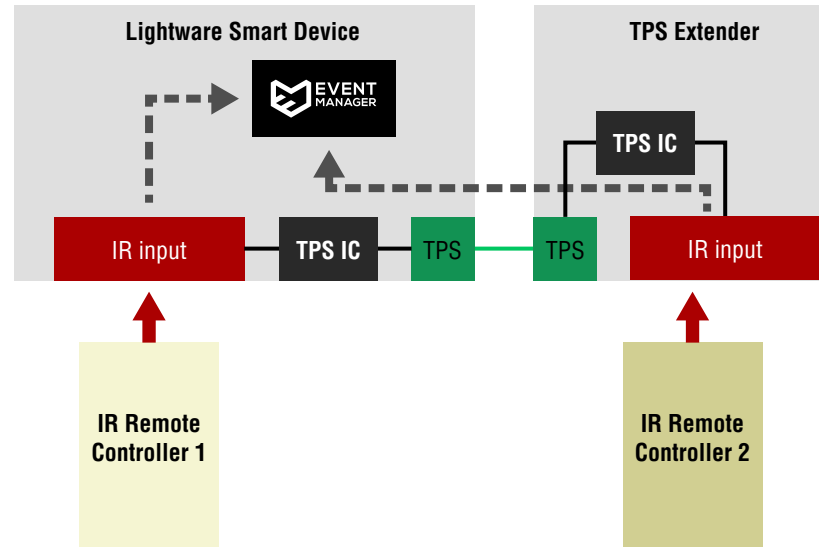


The path of the node that shows the input level: (L=low, H=high)

> /MEDIA/GPIO/P7.Input

5.3. Condition: IR Signal Detection

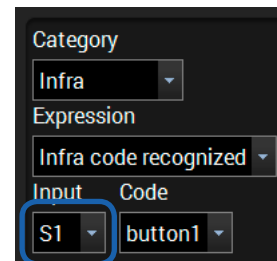
The below example describes two ways of applying an incoming IR signal as a **Condition**. In both cases the **Condition** is processed in the smart TPS device.



- Step 1.** Setup 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 signal, the **Condition** can be set to sense the **local** or the **TPS IR port**. In above example the **RC1** sends IR signal to the **local port** of the **Smart TPS device**, the **RC2** sends IR signal to the IR receiver connected to the IR input port of the **TPS extender**. In last 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.



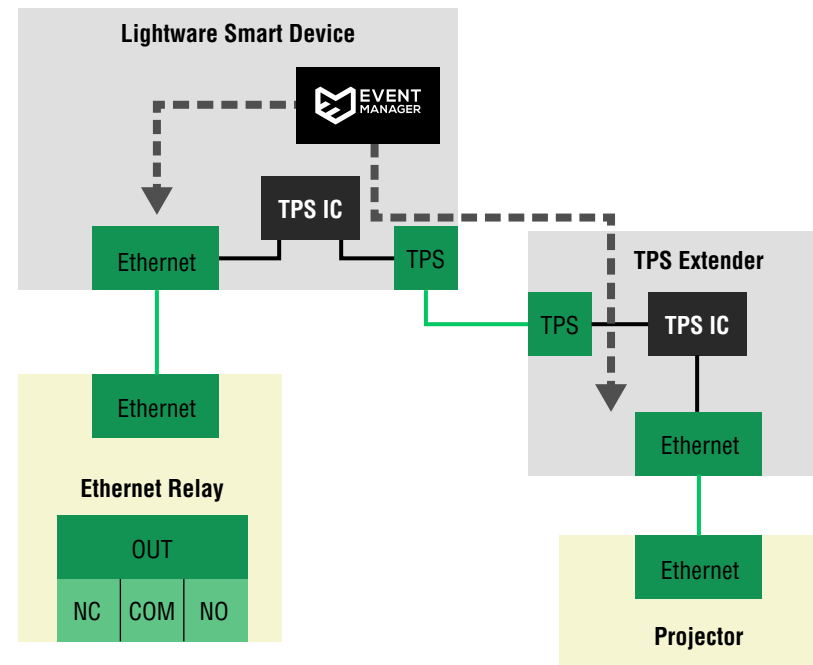
Infra Code Repeat Timeout (ms) parameter

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 whole IR code is stored in the devices but 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

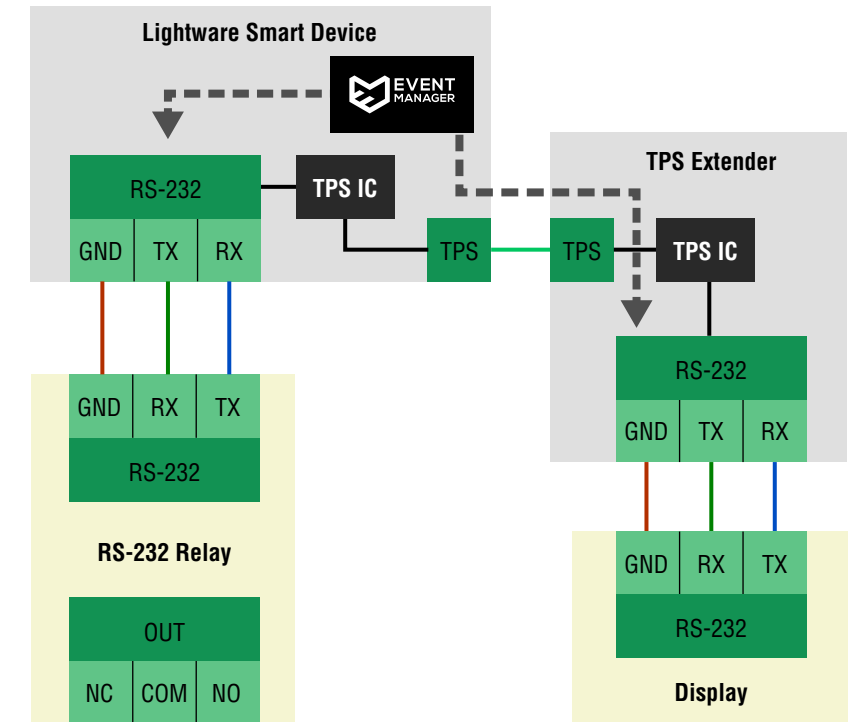
The above example contains two Ethernet devices:

- 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 above example 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).

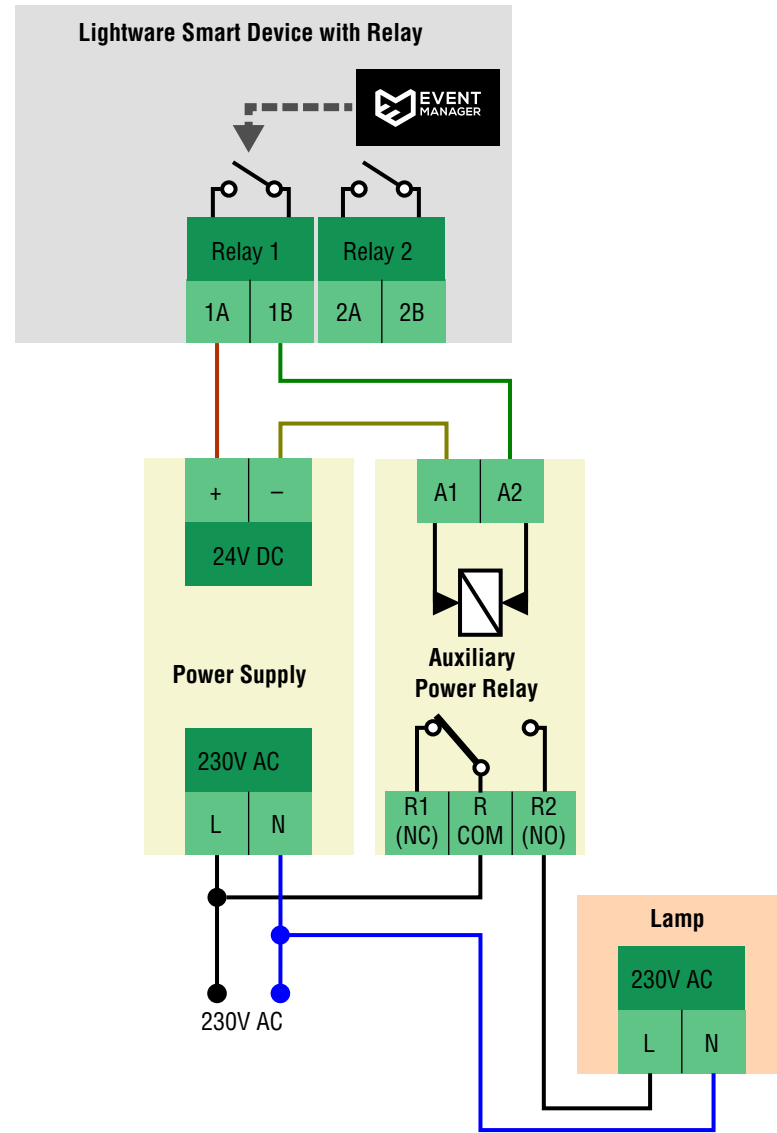
Pay attention to the serial port settings:

- 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

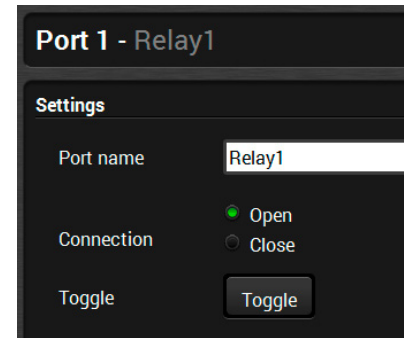
The Relay port of a Lightware device can be used to connect and control a Power Relay. A typical example can be seen in the below figure about how to switch a lamp:



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:



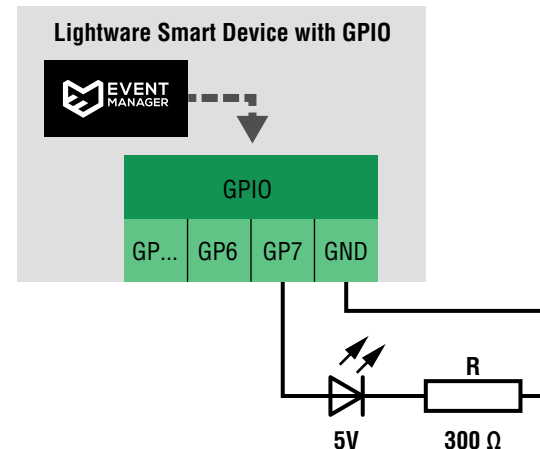
The path of the node that controls the connection state:
(C=close, O=open)

```
> /MEDIA/RELAY/P1.Output
```

5.6. Action: Supplying a LED Directly

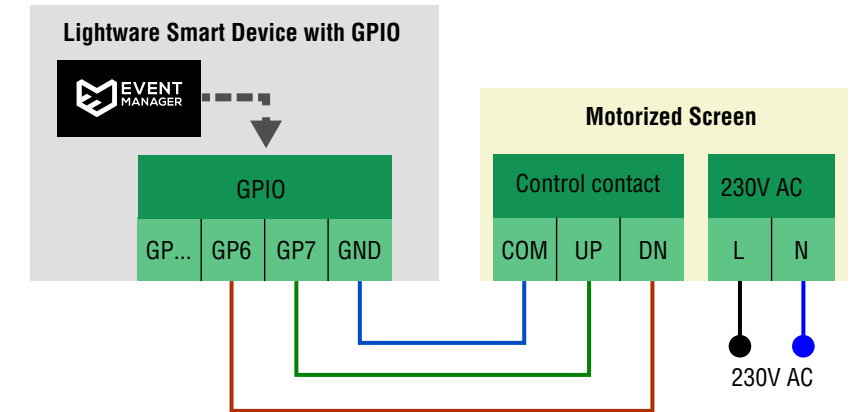
The technical structure of the GPIO port allows supplying simple devices working at low power consumption. In below example a +5V LED is connected to a GPIO pin.

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 below figure:



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:

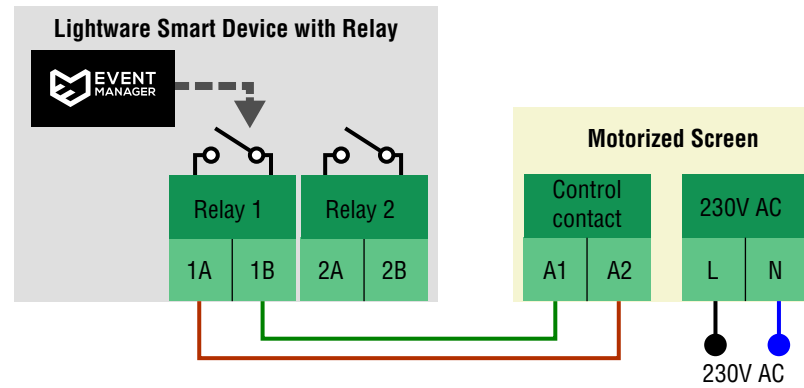
	GP6 level is low	GP6 state is high
GP7 level is low	-	screen rolls up
GP7 level is high	screen rolls down	-

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 below figure:



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. Setup 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](#)

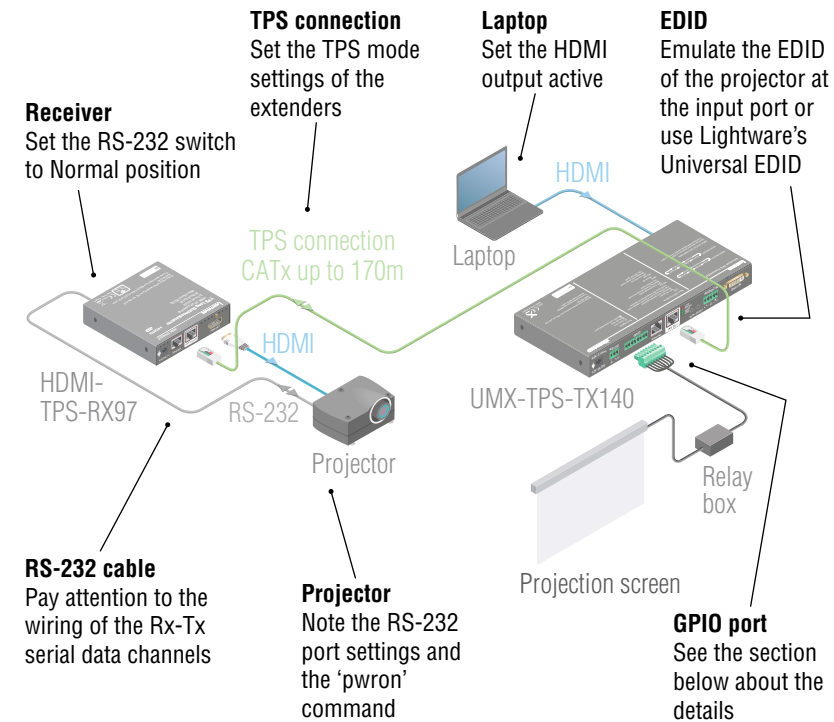
6.1. Details of Example 1

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 above application 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 [Delaying the Action after Detecting the Condition](#) section.

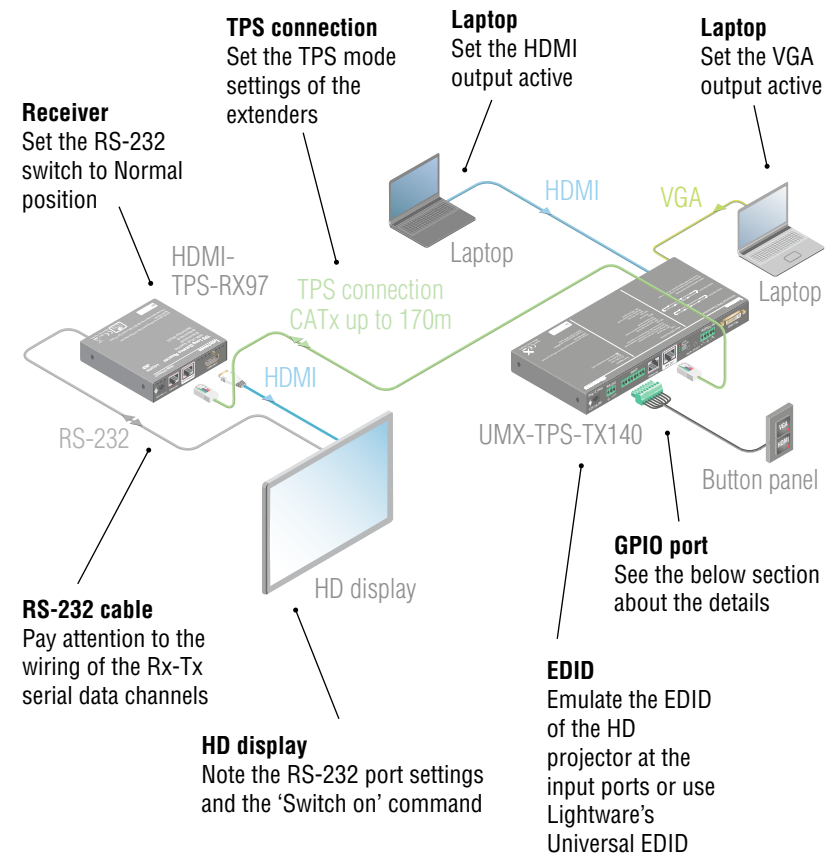
6.2. Details of Example 2

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 [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 [Delaying the Action after Detecting the Condition](#) section.