

Technical note

GPIO channel activation with PlugnCast G2

Version 001A_en *****

Introduction

This document explains how to play a new content using GPIO(s) in PlugnCast G2 environment. A typical use case is to connect buttons to control the player's content.

Prerequisites

- Gekkota 3.12.19+, on a platform which supports gpios (DMB300 for example)
- Plugncast server 2.50.31+
- Plugncast Studio 2.50.41+

GPIOs configuration

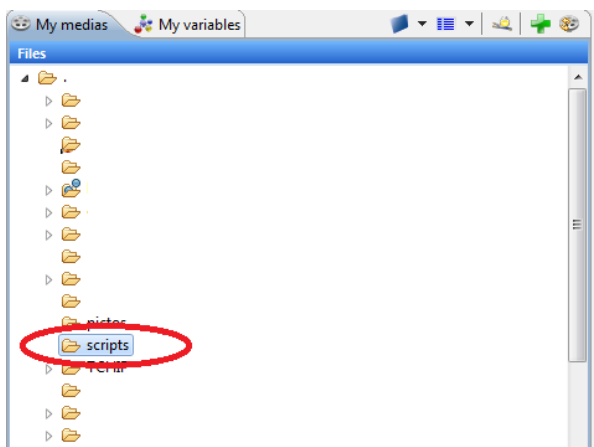
First, refer to document 'Use of GPIOs - application note' to configure the player GPIOs. This permits to configure the proper GPIOs: input/output, debouncing filter (for this use case GPI need to be configured as input). It is advised to check your GPIO configuration firstly with Screen Composer. Note: The GPIOs will be considered as active when their logical level go from 0 to 1 (invert GPIO polarity in your configuration could be needed).

Use with PlugnCast G2

Script

Contrary to ScreenComposer G3, the GPIOs content can not be controlled directly with PlugnCast Studio. That is the reason why you need to use a script 'gpio_channel.js'.

Open PlugnCast Studio and add this script into a folder named 'scripts' to your medias:

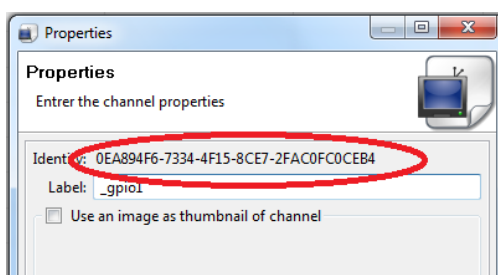


Channels

Each GPIO will permit to replace the current channel(s) with a new one. So you have to create as much new channels as you need.

- In case new gpio channel is an audio channel, the current audio channel will be replaced.
- In case new gpio channel is an audio-visual channel, the current audio-visual channel will be replaced (note: you can stop the current audio channel in the same time, by using the property channel_type, explained after).

The only identifier of each of the new channels is needed to complete the configuration. To get it, go to the 'Channel' tab, select the proper channel, right clic on it, select properties, and copy the channel_id :



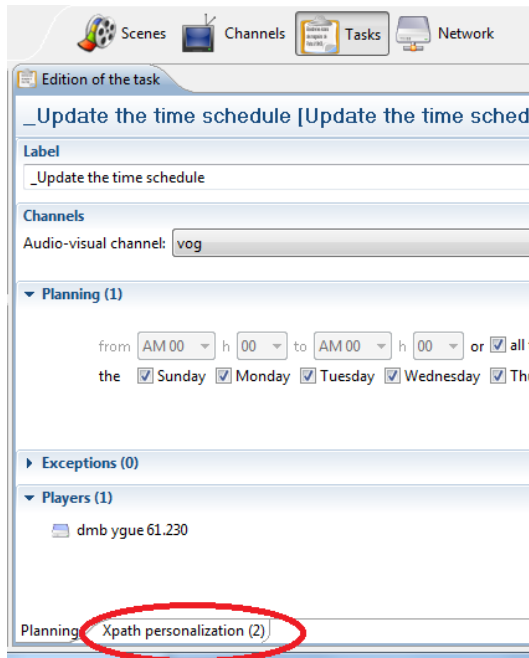
When a gpio channel is selected, the scene (placed inside its calendar) is played. Note: - Do not use the default scene, - Do use only one scene in this channel.

The calendars hours are not taken into account. The only need is to have only one scene available:

18 ⁰⁰							
19 ⁰⁰							
20 ⁰⁰							
21 ⁰⁰	8:23 PM-11:31 PM • _1	8:23 PM-11:31 PM • _1	8:23 PM-11:31 PM • _1	8:23 PM-11:31 PM • _1	8:23 PM-11:31 PM • _1	8:23 PM-11:31 PM • _1	8:23 PM-11:31 PM • _1
22 ⁰⁰							
23 ⁰⁰							

Task configuration

The task 'Update the time schedule' must be tuned for the player to download the script 'gpio_channel.js' and all the new gpio channels. For that, use the tab 'Xpath personalization':



Then you need to add several elements:

First element: script configuration content

```
##

##### //xpfplayer

##

##### <script xmlns="http://www.innes.fr/2007/XPF10/Language">
// Set to true, if a scene launched by a gpio can be interrupted by another (default false)
const interrupt_scene = false;
// The channel id for gpio 1 to 6 to edit (required)
// example : const gpio1_channel_id="3FCC8756-2FB4-42D0-A67D-9F7535779204";
const gpio<1..6>_channel_id="<channel_id>"
// Must be set only if the normal audio channel must be stopped
const gpio<1..6>_channel_type="audio-video";
// The channel id for jack35 gpio to edit (required)
const jack35_channel_id="<channel_id>"
// Must be set only if the normal audio channel must be stopped
const jack35_channel_type="audio-video";
</script>

##
```

You can change the first constant `interrupt_scene` depending on whether a gpio scene can be interrupted by another one.

Then add as much part `const gpio<1..6>_channel_id=` as you need (until 6 gpios). For each, define the identifier of the channel to be played.

Then, add as much part `const gpio<1..6>_channel_type="audio-video"` as you need (until 6 gpios), if the audio channel of the normal source must be stopped.

Note: GPIO jack35 can also be used as well.

Second element: script loading content

```
##

##### //xpfplayer

##
```

```
##### <script src=".medias/scripts/gpio_channel.js"
xmlns="http://www.innes.fr/2007/XPF10/Language"/>

##
```

The part ".medias/scripts/gpio_channel.js" is the relative location where can be found the script.

Element(s) gpio channel loading: audio-visual channels content

```
##

##### //xpfplayer

##

##### <channel height="100%" id="<channel_id>"
left="0px" top="0px" transition="none" transitionDuration="3000"
type="visual" width="100%"
xmlns="http://www.innes.fr/2007/XPF10/Language" zindex="1"/>

##
```

Replace the part the part <channel_id> with the channel identifier of the gpio channel. If you want to activate inter-medias transition, change the transition="none" by transition="opacity", and change transitionDuration if needed (duration in milliseconds, 3000 maximum).

Element(s) gpio channel loading: audio channels content

```
##

##### //xpfplayer

##

##### <channel id="<channel_id>"
type="audio" xmlns="http://www.innes.fr/2007/XPF10/Language" zindex="1"/>

##
```

Replace the part <channel_id> with the channel identifier of the gpio channel.

Each gpio channel must be associated to a script selement.

For example, if you need to use 6 GPIOs + 1 JACK35 GPIO, you will have 9 elements (2 + 6 + 1) inside Xpath Personalization.

WARNING: Ensure that all the elements are in the position 'Add a note in the following location:' else it can work properly.

Example of final rendering:

