

HDIP

created by Just Add  Power



2G+4+

Revised 2017-02-10

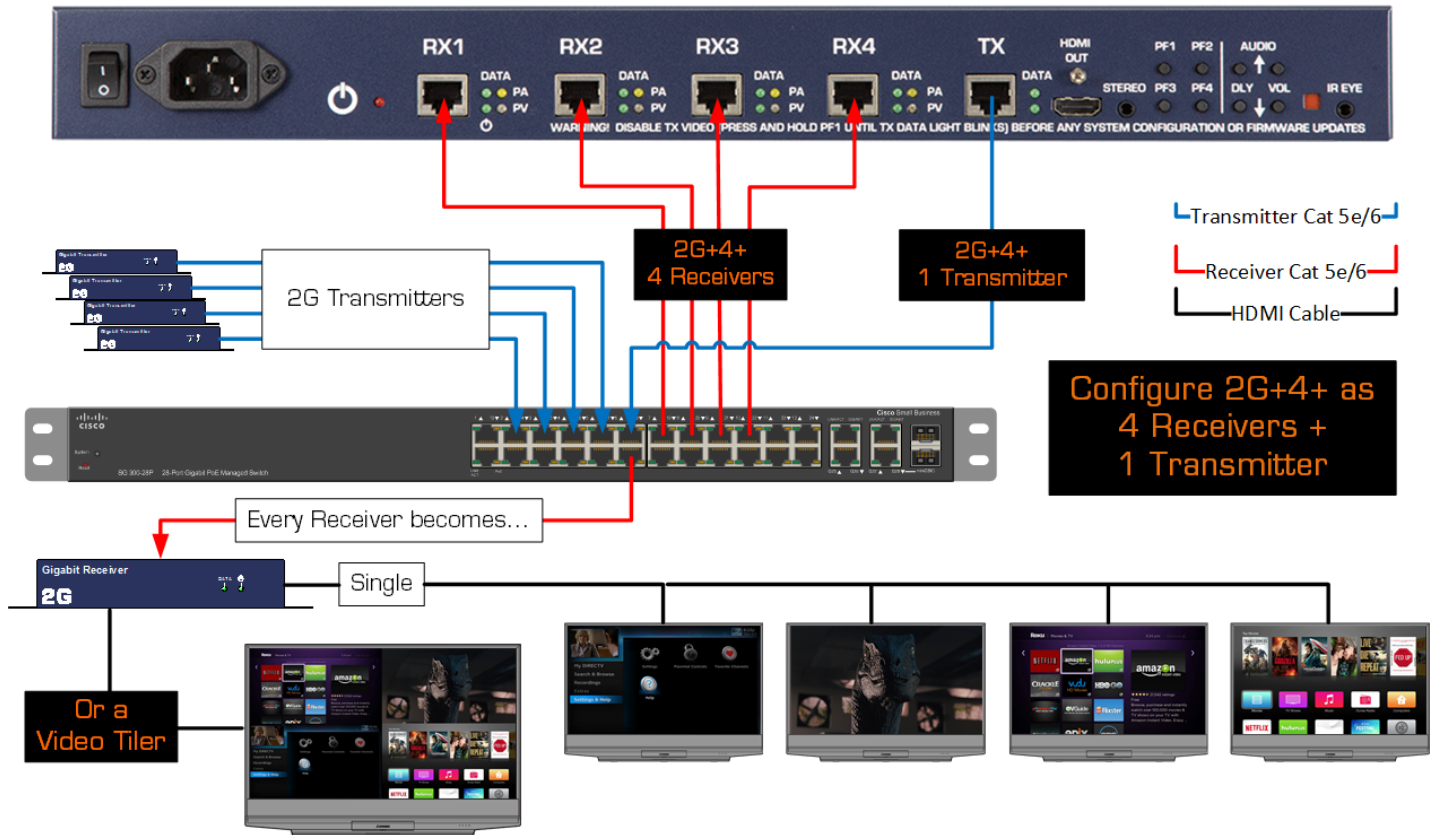
Table of Contents

Introduction.....	3
Video Processing.....	4
Video Formats.....	4
Basic API	5
Functions.....	6
Set Format	6
Set Primary Video	7
Set Primary Audio.....	7
Focus Border Enable/Disable	7
Transmitter Video Disable.....	8
Transmitter Video Enable.....	8
Function Buttons	8
Enhanced API	9
Video Positioning & Size.....	10
Video Fade.....	15

Introduction

2G+4+ Tiling Transmitter

ImagePlay™ by adding Video Tiling to every display

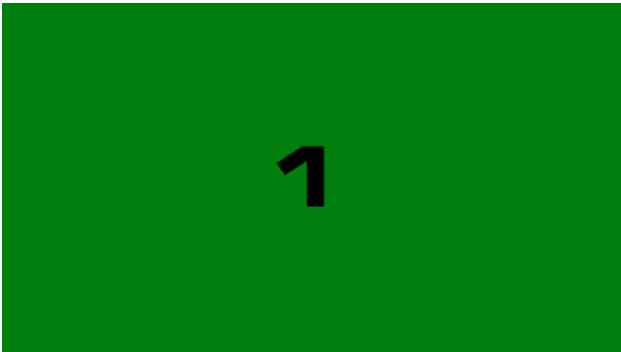


The Just Add Power 2G+4+ Tiling Processor takes 4 network inputs from Just Add Power Transmitters, compiles it into one picture, and outputs that compiled video signal as a unique Transmitter source.

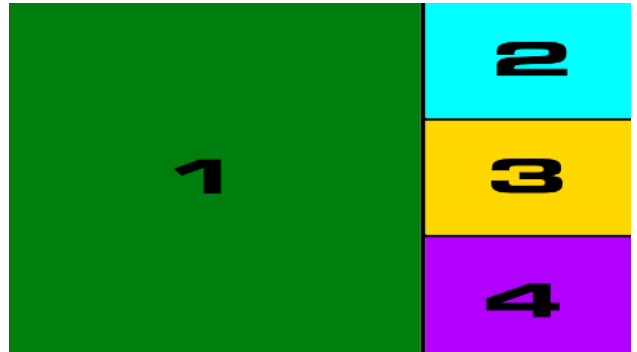
The audio and video of the compiled HDMI signal can be modified by a control system to produce several different combinations of audio/video.

Video Processing

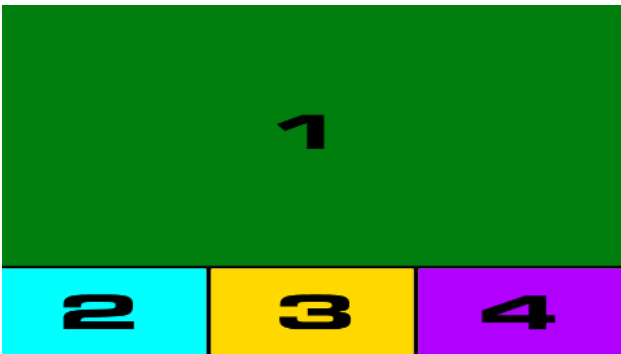
Video Formats



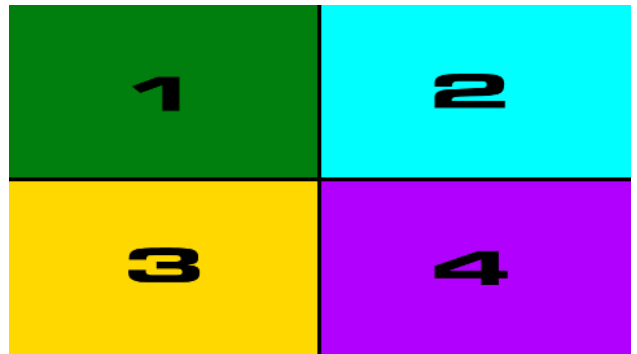
Single-screen



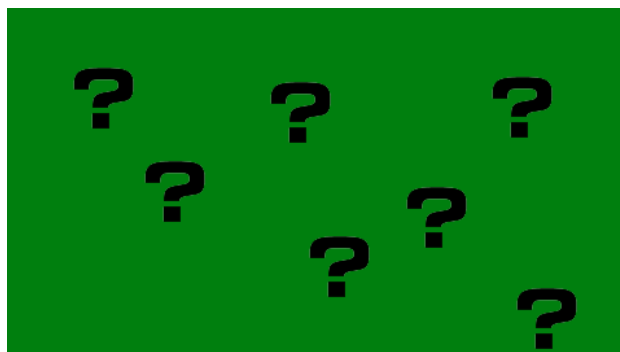
Secondary Video Right



Secondary Video Bottom



Quad-View



Custom mode

Basic API

All commands in the Basic API are available in all 2G+4+ devices regardless of what firmware they are on. The Basic API covers standard layouts, audio selection, and video selection.

[Connect to Tiling Transmitter](#)

The Transmitter provides the interface for controlling the 2G+4+ video format.

To send these commands, telnet into the IP address of the **2G+4+ TRANSMITTER**.

Function List

- [Set Format](#)
- [Set Primary Video](#)
- [Set Primary Audio](#)
- [Focus Border Enable/Disable](#)
- [Transmitter Video Disable](#)
- [Transmitter Video Enable](#)
- [Function Buttons](#)

Functions

Set Format

Set the layout of the video signal output by the Transmitter

Command

`_4in1_layout.sh X`

Variable

X = 1-4 or 10

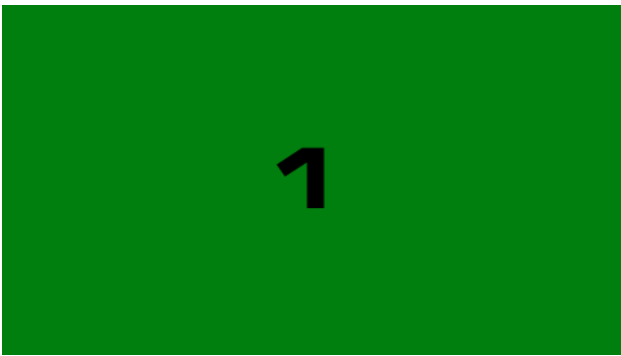
1 = Single-screen

2 = 3 secondaries on right side

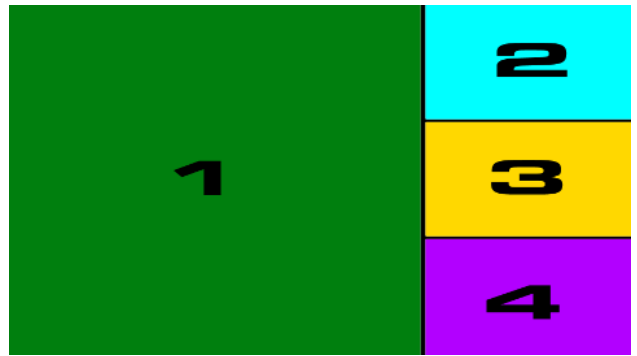
3 = 3 secondaries on bottom

4 = Quad view

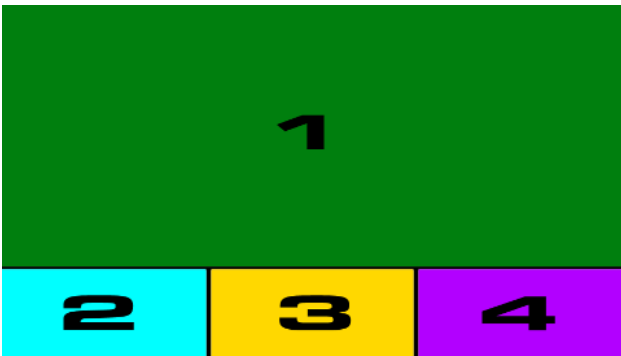
10 = Custom layout based on last changes made with Enhanced API



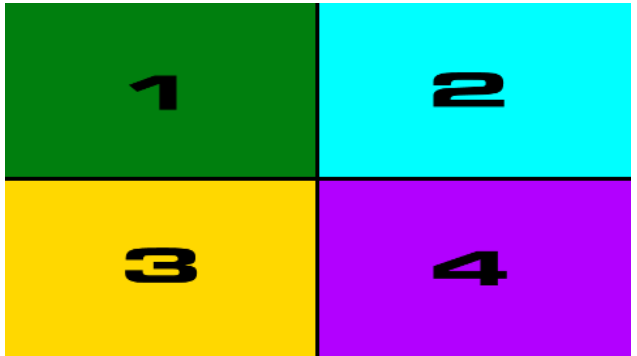
1 = Single-screen



2 = Secondary Video Right



3 = Secondary Video Bottom



4 = Quad-View

Set Primary Video

Set which input Receiver will act as the primary video source

Default is 1

Command

```
_4in1_primary_video.sh X
```

Variable

X = 1-4

Set Primary Audio

Set which position in the layout will act as the audio source →

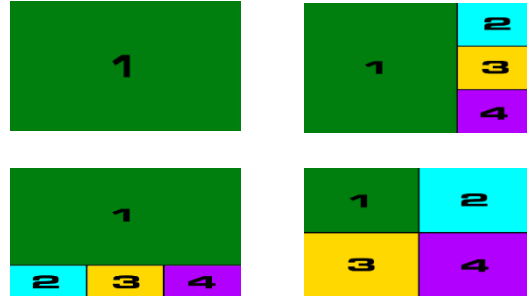
Default is 1

Command

```
_4in1_focus.sh X
```

Variable

X = 1-4 according to the icons to the right →



Focus Border Enable/Disable

Toggle the green audio border on and off.

When set to 'off', any change in audio focus will cause the border to appear for 2 seconds and disappear.

Command

To turn the border on/off on-the-fly (does not persist through reboot):

```
_4in1_focus_onoff.sh X
```

Variable

X = on

Enable the green border

Example: `_4in1_focus_onoff.sh on`

X = off

Disable the green border (default)

Example: `_4in1_focus_onoff`

Command

To set the default on/off state of the green border (persists through reboot):

```
astparam s focus X;astparam save;sleep 1;reboot
```

Variable

X = y

Enable the green border on boot-up

Example: `astparam s focus y;astparam save;sleep 1;reboot`

X = n

Disable the border on boot-up

Example: `astparam s focus n;astparam save;sleep 1;reboot`

Transmitter Video Disable

Stop the video output of the HDMI and Transmitter network port. Must be used for initial setup, as Holding PF1 for 3 seconds will also disable the TX video output.

Command

```
_4in1_stop_tx.sh
```



Transmitter Video Enable

Enable the video output of the HDMI and Transmitter network port. Power-cycling the 2G+4+ will return it to this state.

Command

```
_4in1_restart_tx.sh
```

Function Buttons

- PF1: Tap to disable tiling
Hold to disable HDMI output
- PF2: Tap to cycle primary audio
- PF3: Tap to cycle video format
- PF4: Tap to cycle primary video



Enhanced API

The Enhanced API – activated with firmware A6.4.12 and requiring a separate update to the tiling processor – adds many features to the 2G+4+:

- Video Positioning & Sizing – set the position and size of each video signal
- Fade – transition from one video signal to another
- Chromakey – remove a range of colors from one video and insert another video signal in its place

[Connect to Tiling Transmitter](#)

The Transmitter provides the interface for controlling the 2G+4+ video format.

To send these commands, telnet into the IP address of the **2G+4+ TRANSMITTER**.

Function List

- [Video Positioning & Size](#)
- [Video Fade](#)
- Chromakey – coming soon!

Video Positioning & Size

Description

The `p4p.sh` command has 4 video inputs (referred to as 'channels') available to modify, and many options for modifying those channels.

Channels are not the same as Receiver numbers on the 2G+4+. Channels reference the relation of a specific video input to the primary video input. See the table below for matching:

When primary video is...	...channel 1 is...	...channel 2 is	...channel 3 is...	...channel 4 is...
Receiver 1	Receiver 1	Receiver 2	Receiver 3	Receiver 4
Receiver 2	Receiver 2	Receiver 3	Receiver 4	Receiver 1
Receiver 3	Receiver 3	Receiver 4	Receiver 1	Receiver 2
Receiver 4	Receiver 4	Receiver 1	Receiver 2	Receiver 3

Note: The command to set Primary Audio follows channel numbers in the same way.

Command

```
p4p.sh [layer:ABCD] [chX] [hpos:HPOS] [vpos:VPOS] [width:WIDTH] [height:HEIGHT]
```

Options

1. layer:**ABCD**

- Set the overlapping order for channels 1-4 where they overlap.
- **A,B,C,D** are variables for the channel numbers 1-4
- Examples:
 - i. `p4p.sh layer:1234`
 - Sets the layering order so that channel 1 is always on top with channel 2 underneath channel 1, channel 3 underneath channels 1 & 2, and channel 4 on the bottom
 - This is the layering that must be used when performing Chromakey
 - ii. `p4p.sh layer:3142`
 - Sets the layering order so that channel 3 is always on top with channel 1 underneath channel 3, channel 4 underneath channels 3 & 1, and channel 2 on the bottom

2. ch**X**

- Choose the channel that the subsequent `hpos`, `vpos`, `width`, and `height` commands will modify
- **X** is the variable for the channel number (1-4) to be modified
- The options `hpos`, `vpos`, `width`, and `height` can all be modified in the same command
- Multiple channels can be modified in the same command
- Examples:
 - i. `p4p.sh ch1 hpos:320 vpos:0 width:1280 height:720`
 - Sets channel 1 horizontal starting position to 320 pixels from the left side of the screen, vertical starting position to the top of the screen, with a width of 1280 pixels and height of 720 pixels
 - ii. `p4p.sh ch4 hpos:1280 vpos:720 width:640 height:360`
 - Sets channel 4 horizontal starting position to 1280 pixels from the left side of the screen, vertical starting position to 720 pixels from the top of the screen, with a width of 640 pixels and height of 360 pixels

3. `hpos`: **HPOS**

- Set the horizontal pixel position of the top-left corner of the video channel
- The 2G+4+ tiler outputs 1920 horizontal pixels
 - i. Left side of the screen is pixel 0
 - ii. Right side of the screen is pixel 1920
- Examples:
 - i. `p4p.sh ch1 hpos:320`
 - Sets channel 1 horizontal starting position to 200 pixels from the left side of the screen
 - ii. `p4p.sh ch4 hpos:1280`
 - Sets channel 4 horizontal starting position to 1280 pixels from the left side of the screen
 - iii. `p4p.sh ch1 hpos:320 ch4 hpos:1280`
 - Sets channel 1 horizontal starting position to 200 pixels from the left side of the screen and channel 4 horizontal starting position to 1280 pixels from the left side of the screen in one command

4. `vpos`: **VPOS**

- Set the vertical pixel position of the top-left corner of the video channel
- The 2G+4+ tiler outputs 1080 vertical pixels
 - i. Top of the screen is pixel 0
 - ii. Bottom of the screen is pixel 1080
- Examples:
 - i. `p4p.sh ch1 vpos:0`
 - Sets channel 1 vertical starting position to the top of the screen
 - ii. `p4p.sh ch4 vpos:720`
 - Sets channel 4 vertical starting position to 720 pixels from the top of the screen
 - iii. `p4p.sh ch1 vpos:0 ch4 vpos:720`
 - Sets channel 1 vertical starting position to the top of the screen and channel 4 vertical starting position to 720 pixels from the top of the screen in one command

5. `width`: **WIDTH**

- Set the horizontal pixel width of the video channel
- The 2G+4+ tiler outputs 1920 horizontal pixels
- Examples:
 - i. `p4p.sh ch1 width:1280`
 - Sets channel 1 horizontal width to 1280 pixels
 - ii. `p4p.sh ch4 width:640`
 - Sets channel 4 horizontal width to 640 pixels

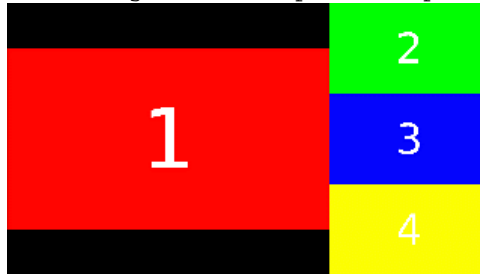
6. `height`: **HEIGHT**

- Set the vertical pixel height of the video channel
- The 2G+4+ tiler outputs 1080 vertical pixels
- Examples:
 - i. `p4p.sh ch2 height:720`
 - Sets channel 2 vertical height to 720 pixels
 - ii. `p4p.sh ch3 height:360`
 - Sets channel 3 vertical height to 360 pixels
 - iii. `p4p.sh ch2 height:720 ch3 height:360`
 - Sets channel 2 vertical height to 720 pixels and channel 3 vertical height to 360 pixels in one command

Examples

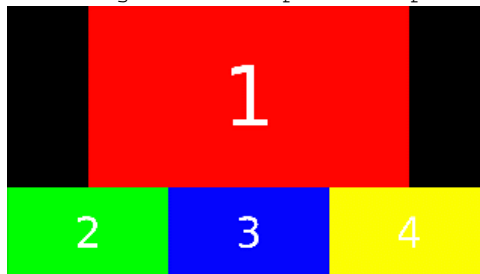
1. Three on the right with proper aspect ratio

- `p4p.sh ch1 hpos:0 vpos:180 width:1280 height:720 ch2 hpos:1280 vpos:0 width:640 height:360 ch3 hpos:1280 vpos:360 width:640 height:360 ch4 hpos:1280 vpos:720 width:640 height:360`



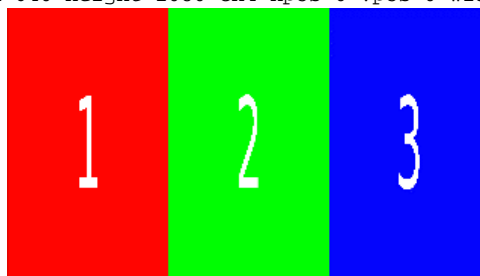
2. Three on bottom with proper aspect ratio

- `p4p.sh ch1 hpos:320 vpos:0 width:1280 height:720 ch2 hpos:0 vpos:720 width:640 height:360 ch3 hpos:640 vpos:720 width:640 height:360 ch4 hpos:1280 vpos:720 width:640 height:360`



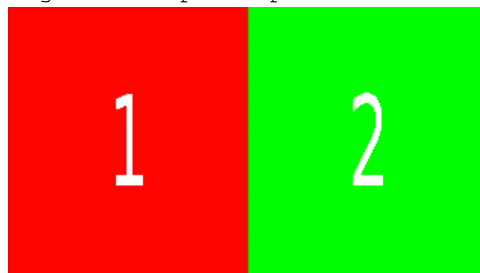
3. Three Columns

- `p4p.sh ch1 hpos:0 vpos:0 width:640 height:1080 ch2 hpos:640 vpos:0 width:640 height:1080 ch3 hpos:1280 vpos:0 width:640 height:1080 ch4 hpos:0 vpos:0 width:0 height:0`



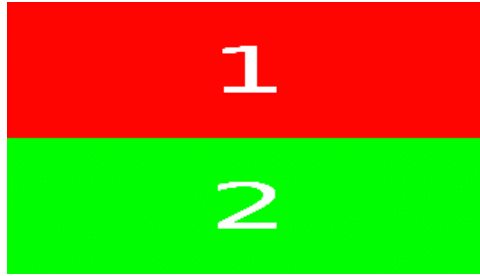
4. Left and Right

- `p4p.sh ch1 hpos:0 vpos:0 width:960 height:1080 ch2 hpos:960 vpos:0 width:960 height:1080 ch3 hpos:0 vpos:0 width:0 height:0 ch4 hpos:0 vpos:0 width:0 height:0`



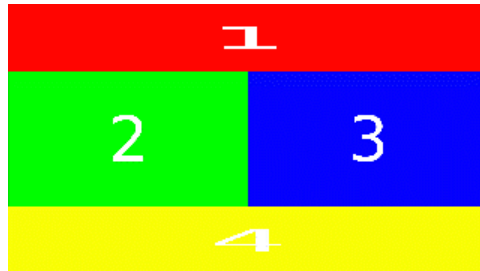
5. Top and Bottom

- `p4p.sh ch1 hpos:0 vpos:0 width:1920 height:540 ch2 hpos:0 vpos:540 width:1920 height:540 ch3 hpos:0 vpos:0 width:0 height:0 ch4 hpos:0 vpos:0 width:0 height:0`



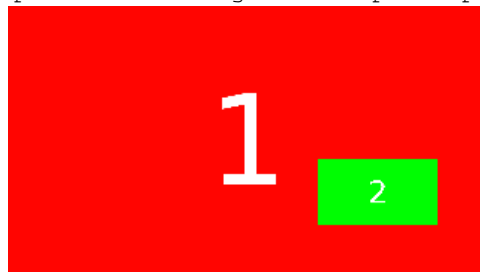
6. Sandwich

- `p4p.sh ch1 hpos:0 vpos:0 width:1920 height:270 ch2 hpos:0 vpos:270 width:960 height:540 ch3 hpos:960 vpos:270 width:960 height:540 ch4 hpos:0 vpos:810 width:1920 height:270`

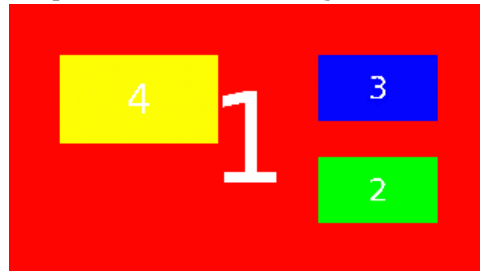


7. Picture-On-Picture

- `p4p.sh layer:4321 ch1 hpos:0 vpos:0 width:1920 height:1080 ch2 hpos:1240 vpos:610 width:480 height:270 ch3 hpos:0 vpos:0 width:0 height:0 ch4 hpos:0 vpos:0 width:0 height:0`

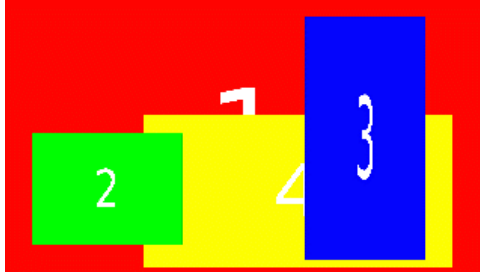


- `p4p.sh layer:2134 ch1 hpos:0 vpos:0 width:1920 height:1080 ch2 hpos:1240 vpos:610 width:480 height:270 ch3 hpos:1240 vpos:200 width:480 height:270 ch4 hpos:200 vpos:200 width:640 height:360`



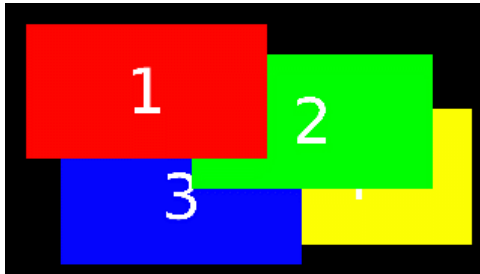
8. Oddball

- `p4p.sh layer:3241 ch1 vpos:0 hpos:0 width:1920 height:1080 ch2 hpos:103 vpos:525 width:600 height:450 ch3 hpos:1194 vpos:67 width:480 height:967 ch4 hpos:549 vpos:453 width:1232 height:610`

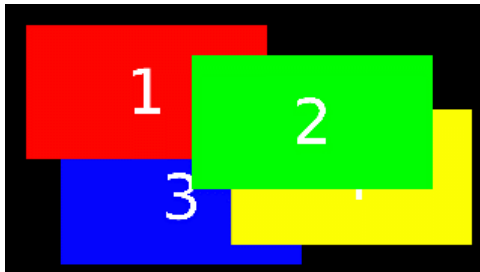


9. Layer Demo

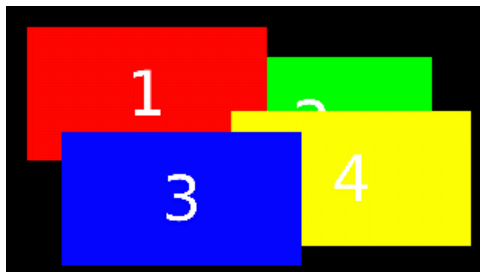
- `p4p.sh ch1 hpos:80 vpos:80 width:960 height:540 ch2 hpos:740 vpos:200 width:960 height:540 ch3 hpos:220 vpos:500 width:960 height:540 ch4 hpos:900 vpos:420 width:960 height:540`
 - i. `p4p.sh layer:1234`



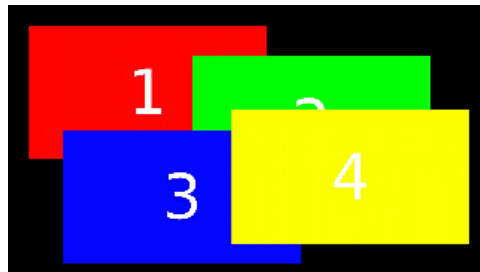
- ii. `p4p.sh layer:2143`



- iii. `p4p.sh layer:3412`



- iv. `p4p.sh layer:4321`



Video Fade

Description

The `p4p.sh` command can fade between the 4 video inputs (referred to as 'channels') but fade only occurs where all four videos are overlapping.

Channels are not the same as Receiver numbers on the 2G+4+. Channels reference the relation of a specific video input to the primary video input. See the table below for matching:

When primary video is...	...channel 1 is...	...channel 2 is	...channel 3 is...	...channel 4 is...
Receiver 1	Receiver 1	Receiver 2	Receiver 3	Receiver 4
Receiver 2	Receiver 2	Receiver 3	Receiver 4	Receiver 1
Receiver 3	Receiver 3	Receiver 4	Receiver 1	Receiver 2
Receiver 4	Receiver 4	Receiver 1	Receiver 2	Receiver 3

Note: The command to set Primary Audio follows channel numbers in the same way.

Commands

- `_4in1_layout.sh 1`
 - In order for fading to work, all channels must completely overlap with each other
 - This command sets all channels to completely overlap with layering order of 1, 2, 3, 4
- `p4p.sh fade:AB#X`
 - A** & **B** are channel numbers. **A** is the channel that begins the fade and **B** is the channel that is being faded to.
 - X** determines the number of tenths of seconds (0.1s) that the fade transition takes to complete
 - Examples:
 - `p4p.sh fade:12#10`
 - Transition from channel 1 to channel 2 with a 1-second fade time
 - If channel 1 is not the top layer, then it will be made the top layer before fading
 - `p4p.sh fade:42#5`
 - Transition from channel 4 to channel 2 with a 0.5-second fade time
 - If channel 4 is not the top layer, then it will be made the top layer before fading