

Raw Panel

SKAARHOJ



————— Your parameter values

————— Your encoder knobs

————— Your colors

————— Your images

————— Your text

————— Your button press

————— =Your panel!

Integrate SKAARHOJ panels with your product!

Cross Over

Frame Shot

Micro Fly

Quick Pad

Inline 10

PTZ Wiz

PTZ Fly

PTZ Pro

PTZ Extreme

Inline 22

XC7

Vision Mixing

XPoint 48

Air Fly Pro

Master Key One

MK48

MKT1A/B

Air Fly

Rack Fusion Live

MKA1

MKA2

MKA3

MKA4

XPoint 24

Live Fly

Mini Fly

RCP & Color Control

RCPv2 - Wheel

RCPv2 - Fader

RCPv2 - Joystick

RCPv2 Pro

Wave Board

Color Fly

Audio

Wave Board

Utility

Rack Fly Uno

Rack Fly Duo

Rack Fly Trio

Rack Control Uno

Rack Control Duo

Rack Control XK1

Rack Control XK2

Rack Control XK3

Frame Shot Uno

Link IO

Tally Box

Tally Lamp

ATEM-TCP Link

SDI-B4 Link

ETH-LANC Link

ETH-SDI Link

ETH-B4 Link

SDI-GPI Link

ETH-GPI Link

Blue Pill

Blue Pill

Raw Panel

- ★ All SKAARHOJ panels supported
- ★ Easy to get started
- ★ IP Networked - TCP server
- ★ Event based triggers
- ★ Feedback for LEDs and displays
- ★ ASCII or binary encoded messages
- ★ Self describing topology
- ★ Discoverable via mDNS/zeroconf
- ★ Panel Emulator
- ★ Helpful tool chain

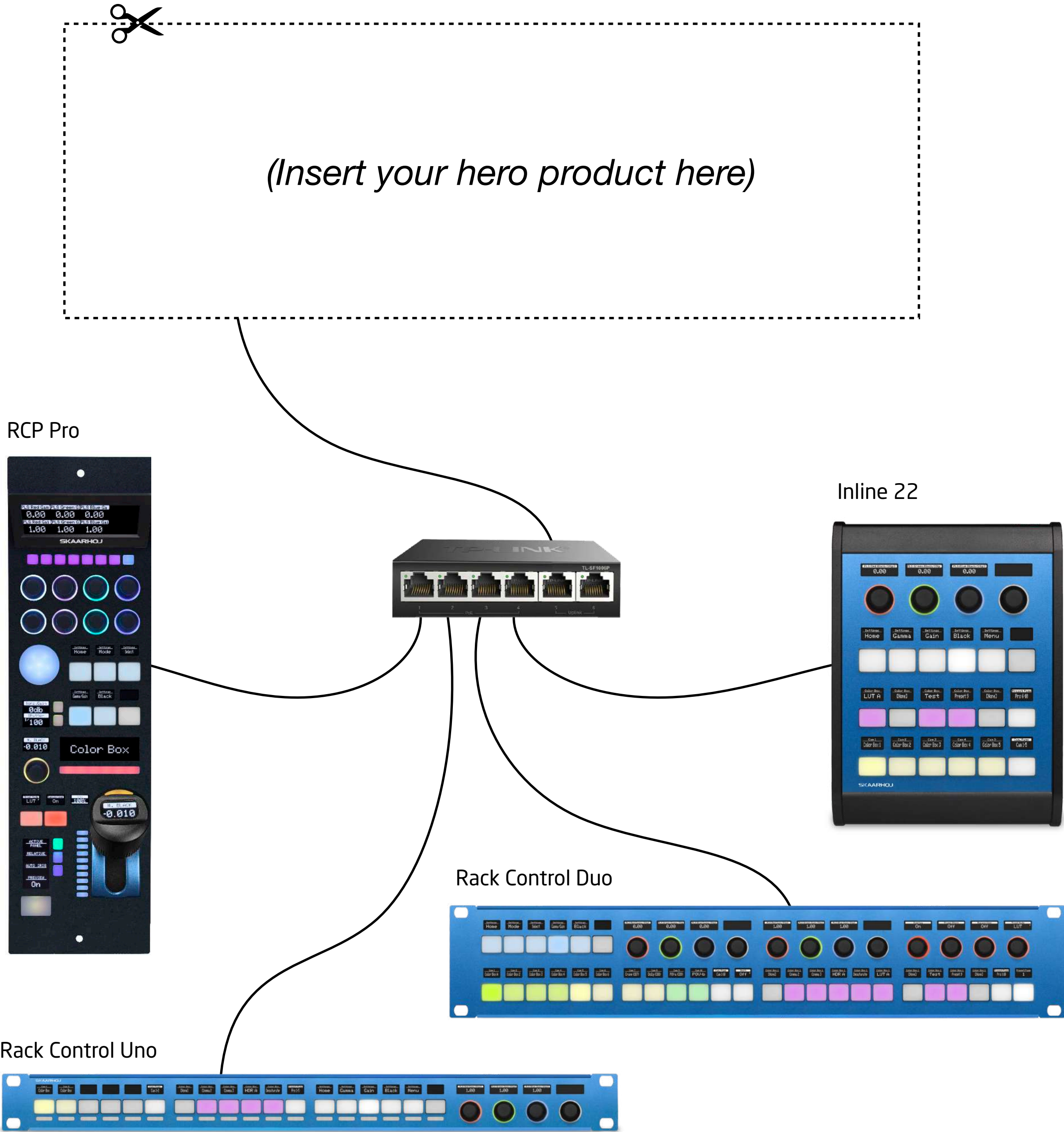
SKAARHOJ panels are designed for broadcast and AV live productions. It's not a toy, not a game pad, not a compromise. We are dedicated to supply every type and form factor of tactile control panels the broadcast market needs. Raw Panel is the internal backbone protocol in any SKAARHOJ panel. It means any SKAARHOJ panel is equally easy to integrate with your software or hardware solution. The panels are 100% IP with Power over Ethernet (PoE) and you simply connect one or more clients via a TCP connection to the panel.

The panel sends triggers back over TCP as the user presses buttons, turns knobs, moves a fader or a joystick. In return your product can send color codes and display text or graphics back to the panel.

The messaging protocol is uniquely capable of operating in ASCII or binary mode. In ASCII mode all messages are human readable and easy to decipher. In the binary encoding your get more efficiency for your software integration. Raw Panel includes a concept for panels to communicate their topology. This makes a panel completely self describing. The topology reveals all relevant details for any hardware component on the panel so a connected system can build configuration interfaces for any current and future Raw Panel compliant device.

Panels are also discoverable with ZeroConf/mDNS and SKAARHOJ provides a powerful and free set of tools to emulate any SKAARHOJ panel and help explore the Raw Panel protocol.

Easy!



```
kasper — nc 192.168.11.196 9923 — nc — nc 192.168.11.196...

Last login: Fri Sep  2 22:22:55 on ttys034
[~ > nc 192.168.11.196 9923
list
_model=SK_INLINE22V2
_serial=999923
_name=Inline 22
_platform=simulator
_bluePillReady=1
_serverModeLockToIP=
HWC#2.4=Down
HWC#2.4=Up
HWC#26=Enc:1
```

Just connect...
Type "list" and enter, the panel will tell you it's identity
Try to trigger a *hardware component* (HWC) like a button or turn an encoder knob - they will let you know immediately :-)

T-Bar:
HWC#50=Abs:0
HWC#50=Abs:22
HWC#50=Abs:39
HWC#50=Abs:66
HWC#50=Abs:92
...
HWC#50=Abs:967
HWC#50=Abs:988
HWC#50=Abs:1000

Triggers



Encoder (Left/Right turns):

HWC#86=Enc:1
HWC#86=Enc:2
HWC#86=Enc:6
HWC#86=Enc:1
HWC#86=Enc:-4
HWC#86=Enc:-3
HWC#86=Enc:-6
HWC#86=Enc:-5

Four-way buttons

HWC#55.8=Down
HWC#55.8=Up
HWC#55.1=Down
HWC#55.1=Up
HWC#55.2=Down
HWC#55.2=Up
HWC#55.4=Down
HWC#55.4=Up

(Digits 1,2,4 and 8 are edge detection)

NKK Broadcast Buttons

HWC#12=Down
HWC#12=Up
HWC#13=Down
HWC#12=Up

(Straight up old school)

Feedback

Fader Position:

HWCx#87=21035

Alternative JSON encoding:
{"HWCIDs":53,"HWCEntended":
{"Interpretation":5,"Value":555}}

Blue as RGB (3*2bit):

HWCC#52=203

Alternative JSON encoding:
{"HWCIDs":52,"HWCColor":{"ColorRGB":
{"Red":52,"Green":213,"Blue":255}}}

"On" and Amber:

HWC#53=4

HWCC#53=136

Alternative JSON encoding:
{"State":4,"HWCColor":{"ColorIndex":{"Index":8}}}

Text

HWCT#54=|||Output 6|1|Feed1|Drone|1

Alternative JSON encoding:
{"HWCIDs":53,"HWCText":
{"Formatting":7,"Title":"Output
6","Textline1":"Feed1","Textline2":"Drone","PairMo
de":1}}



Message Encoding

Raw Panel ASCII Command sequence (v1):

```
HWC#53=4
HWCc#53=136
HWCx#53=21035
HWCt#53=|||Output 6|1|Feed1|Drone||1
```

Raw Panel ASCII JSON (v2):

```
{"HWCIDs": [53], "HWCMODE": {"State": 4}, "HWCCOLOR": {"ColorIndex": {"Index": 8}}, "HWCEXTENDED": {"Interpretation": 5, "Value": 555}, "HWCTEXT": {"Formatting": 7, "Title": "Output 6", "Textline1": "Feed1", "Textline2": "Drone", "PairMode": 1}}
```

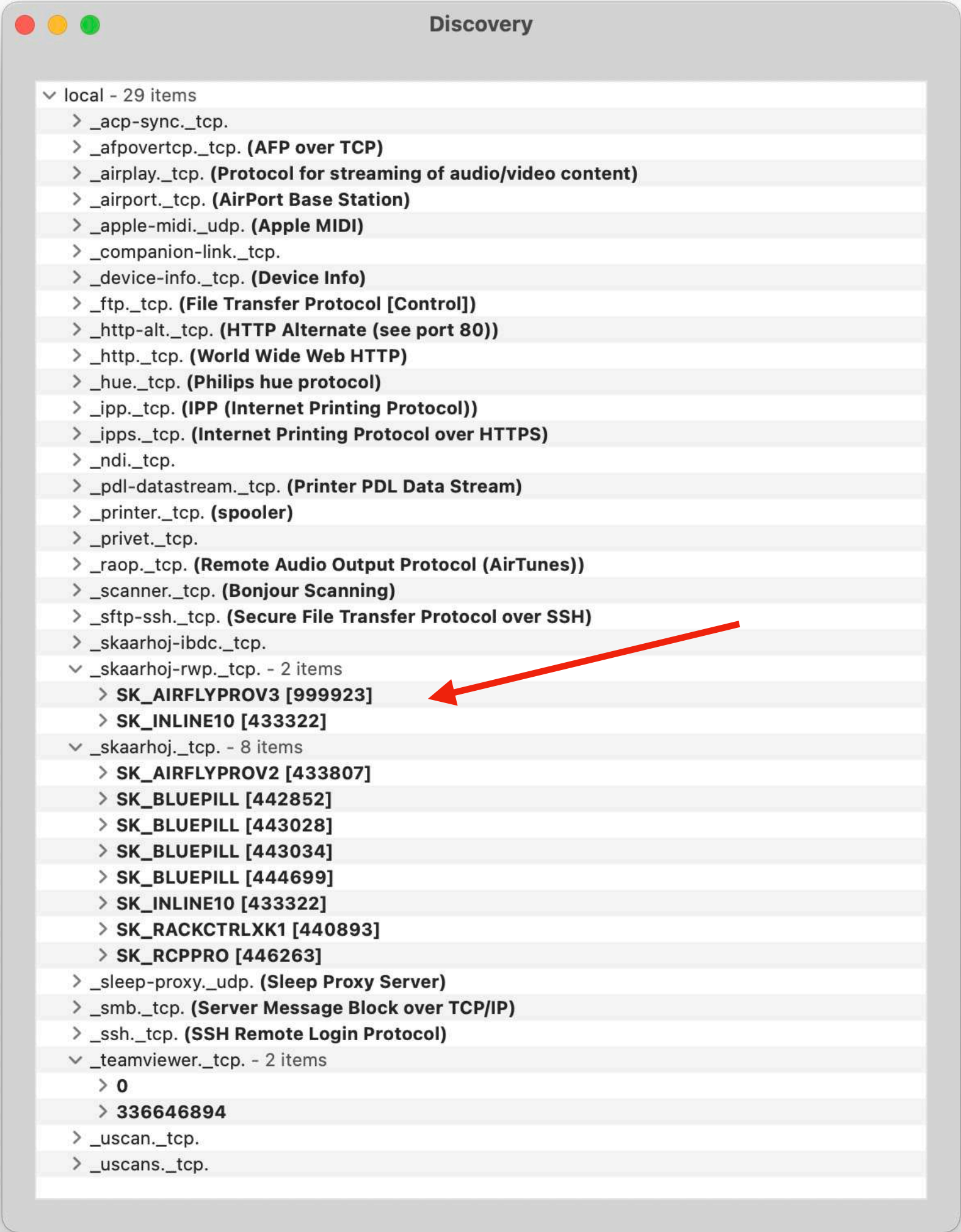
Raw Panel Binary Protobuf command:

```
34 00 00 00 42 32 0a 01 35 12 02 10 04 1a 04 12 02 08 08 22
05 10 05 18 ab 04 2a 1c 18 07 3a 08 4f 75 74 70 75 74 20 36
4a 05 46 65 65 64 31 52 05 44 72 6f 6e 65 60 01
```

Two non-binary and a protobuf met at a bar...

On the side of a Raw Panel client (your product!), Raw Panel can be encoded both in ASCII form and in binary form. The availability will depend on the given panel. UniSketch panels (SKAARHOJs line up until recently) support ASCII V1 only. Blue Pill panels (new platform) supports both ASCII V1, V2 (JSON encoding) as well as binary Protocol Buffer encoded form. ASCII V1 supports all essentials of Raw Panel, it's just encoded in a proprietary way that is sometimes great and at other times clumsy. The example above shows how easy it is to turn a button on (HWC#53=4) while putting content into a display is more easy to decipher when looking at the JSON version (v2). Most third party integrators will get started with ASCII v1 and if they need to, they will jump to ASCII V2 or even the binary encoding. The Binary encoding is used internally in all SKAARHOJ products and basically falls right out of the Google Protocol Buffer libraries which are conveniently integrated, so it's really the far easiest and fastest option for us. It should also be your choice if you intend to integrate Raw Panel with Go (Golang) since we have a lot of free libraries available for all that handling and conversion.

Where's Waldo?



Network Discovery
SKAARHOJ panels are easily discovered on your network using ZeroConf/mDNS look-up.

Self Describing Topology



JSON and SVG

Panels can deliver a complete description of their features via a combination of a SVG basis and a JSON data structure that describes every hardware component.

Components are characterized by features such as input type (binary, pulsed, analog and intensity), their output (rgb led, extended features such as motorized faders and of course associated display resolution and color style (b/w, gray, rgb)

SKAARHOJ uses nothing but the topology SVG and JSON to render the visualization of panels in our applications and emulators. If we can do it, you can do it too!

Same Same

SKAARHOJ provides an emulator for Windows, Mac and Linux which can emulate any SKAARHOJ panel. It will start a TCP server and open up a web browser with a view based on the panels topology. You can interact with the virtual panel - press buttons, turn encoders, drag faders and joysticks around. The emulator will also display text, graphics and LED colors faithfully. The emulator will even emulate button and display brightness commands and the sleep timer!

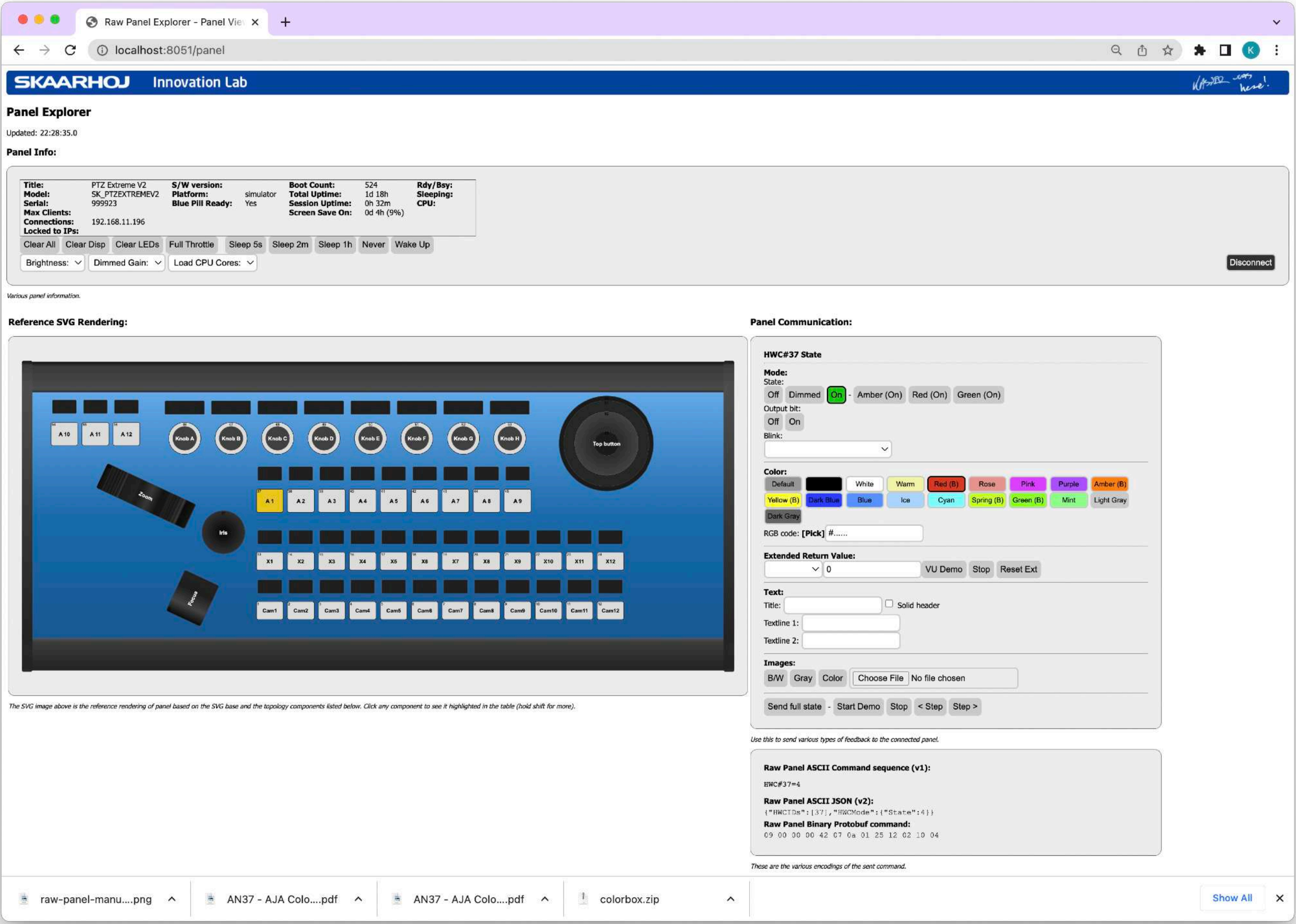


Real World Panel

Emulated Panel



Have Fun Learning



Panel Explorer

We have made an awesome exploration tool to play your way into Raw Panel. With Panel Explorer (Windows/Mac/Linux) you can not only scan your network for available panels, you can also connect to them, try to send over feedback commands for colors and display contents - and you will see the commands shown. It will get you up to speed in no time.

Triggers from the panel are also displayed in Panel Explorer. There is even a "trigger scope" that draws the triggers graphically and analyses various timing aspects.

Oh, did we mention - the source code (written in Go) is MIT licensed and public! There are binaries for download at Github too.

<https://github.com/SKAARHOJ/raw-panel-explorer>

Topology Summary:

HWC id	Text	In	Out	Ext	Display
1	Cam1	Button, Four-way (b4)	RGB LED (rgb)		64x32, Mono
2	Cam2	Button, Four-way (b4)	RGB LED (rgb)		64x32, Mono
3	Cam3	Button, Four-way (b4)	RGB LED (rgb)		64x32, Mono
4	Cam4	Button, Four-way (b4)	RGB LED (rgb)		64x32, Mono
5	Cam5	Button, Four-way (b4)	RGB LED (rgb)		64x32, Mono
6	Cam6	Button, Four-way (b4)	RGB LED (rgb)		64x32, Mono
7	Cam7	Button, Four-way (b4)	RGB LED (rgb)		64x32, Mono
8	Cam8	Button, Four-way (b4)	RGB LED (rgb)		64x32, Mono
9	Cam9	Button, Four-way (b4)	RGB LED (rgb)		64x32, Mono
10	Cam10	Button, Four-way (b4)	RGB LED (rgb)		64x32, Mono
11	Cam11	Button, Four-way (b4)	RGB LED (rgb)		64x32, Mono
12	Cam12	Button, Four-way (b4)	RGB LED (rgb)		64x32, Mono
13	X1	Button, Four-way (b4)	RGB LED (rgb)		64x32, Gray
14	X2	Button, Four-way (b4)	RGB LED (rgb)		64x32, Gray
15	X3	Button, Four-way (b4)	RGB LED (rgb)		64x32, Gray
16	X4	Button, Four-way (b4)	RGB LED (rgb)		64x32, Gray
17	X5	Button, Four-way (b4)	RGB LED (rgb)		64x32, Gray
18	X6	Button, Four-way (b4)	RGB LED (rgb)		64x32, Gray
19	X7	Button, Four-way (b4)	RGB LED (rgb)		64x32, Gray
20	X8	Button, Four-way (b4)	RGB LED (rgb)		64x32, Gray
21	X9	Button, Four-way (b4)	RGB LED (rgb)		64x32, Gray
22	X10	Button, Four-way (b4)	RGB LED (rgb)		64x32, Gray
23	X11	Button, Four-way (b4)	RGB LED (rgb)		64x32, Gray
24	X12	Button, Four-way (b4)	RGB LED (rgb)		64x32, Gray
37	A 1	Button, Four-way (b4)	RGB LED (rgb)		64x32, Mono
38	A 2	Button, Four-way (b4)	RGB LED (rgb)		64x32, Mono
39	A 3	Button, Four-way (b4)	RGB LED (rgb)		64x32, Mono
40	A 4	Button, Four-way (b4)	RGB LED (rgb)		64x32, Mono
41	A 5	Button, Four-way (b4)	RGB LED (rgb)		64x32, Mono
42	A 6	Button, Four-way (b4)	RGB LED (rgb)		64x32, Mono
43	A 7	Button, Four-way (b4)	RGB LED (rgb)		64x32, Mono
44	A 8	Button, Four-way (b4)	RGB LED (rgb)		64x32, Mono
45	A 9	Button, Four-way (b4)	RGB LED (rgb)		64x32, Mono
46	Knob A	Encoder w/button (pb)	RGB LED (rgb)		112x32, Mono
47	Knob B	Encoder w/button (pb)	RGB LED (rgb)		112x32, Mono
48	Knob C	Encoder w/button (pb)	RGB LED (rgb)		112x32, Mono
49	Knob D	Encoder w/button (pb)	RGB LED (rgb)		112x32, Mono
50	Knob E	Encoder w/button (pb)	RGB LED (rgb)		112x32, Mono
51	Knob F	Encoder w/button (pb)	RGB LED (rgb)		112x32, Mono
52	Knob G	Encoder w/button (pb)	RGB LED (rgb)		112x32, Mono
53	Knob H	Encoder w/button (pb)	RGB LED (rgb)		112x32, Mono
54	A 10	Button, Four-way (b4)	RGB LED (rgb)		64x32, Mono
55	A 11	Button, Four-way (b4)	RGB LED (rgb)		64x32, Mono
56	A 12	Button, Four-way (b4)	RGB LED (rgb)		64x32, Mono
57	Iris	Encoder w/button (pb)	-		
58	Zoom	Intensity component, horizontal (ih)	-		
59	Focus	Encoder (pulsed input) (p)	-		
60	LR	Intensity component, horizontal (ih)	-		
61	UD	Intensity component, vertical (iv)	-		
62	Rotate	Intensity component, rotational (ir)	-		
63	Top button	Button (b)	-		

The golden QR code



{ Raw Panel API Manual }