



Spades AI Manual

Installing, configuring and integrating Spades AI at live tables

Release **v3.2.0**

Build v3.2.0-59ec385

Contents

1 Introduction	4
1.1 Supported games and modules	4
1.2 Provided binaries	4
1.3 Feedback to the developers	5
2 Network requirements	6
2.1 Internet access	6
2.2 Network settings of the device	6
2.3 Option A: direct cable (link-local)	7
2.4 Option B: dedicated VLAN (recommended)	7
2.5 Ports	7
2.6 Live video	8
3 Fresh installation on the Seeed Studio reComputer J4011	9
3.1 What you need	9
3.2 Step 1: Install the RTC battery	9
3.3 Step 2: Flash the UEFI firmware	9
3.4 Step 3: Create the installer USB stick	10
3.5 Step 4: Install or update the Spades Vision OS	11
3.6 Step 5: Check the device panel	12
3.7 Troubleshooting the firmware flash	13
4 Licensing system	14
4.1 How licensing works	14
4.2 License types	14
4.3 Get a license file	14
4.4 Upload the license file	15
4.5 When the license is not valid	15
5 Setup and usage guide	17
5.1 Lens focus guidelines	17
5.2 Device panel	17
5.3 Table configuration	18
5.4 Analytics and Excel reports	24
6 Testing procedure	26
6.1 Testing the system as a whole	26
6.2 Testing with a recorded video	27
7 Side bet configuration	29
7.1 The file	29
7.2 How a bet is decided	30
7.3 Conditions	30
7.4 Examples	31
7.5 What a config cannot express	36
7.6 Uploading and managing side bets	36
8 API documentation	39
8.1 MQTT API	39
8.2 Configuration API (HTTP)	42
9 Troubleshooting guide	44
9.1 First checks	44
9.2 Downloading the diagnostics	44
9.3 Reporting a single round	45
9.4 Diagnosing without sending a technician	45
9.5 Replacing a device	45
9.6 Remote access for Spades Vision	45
9.7 The device panel is not reachable	46
9.8 The panel says "Not connected to the Spades Vision server"	46
9.9 License issues	46
9.10 Detection and calibration issues	46

10 Supported environments and requirements 48

10.1 Table environment 48

10.2 Game environment 48

10.3 Playing cards 49

11 Changelog 52

1 Introduction

Spades AI follows a live gaming table through a camera mounted above it. A Jetson device next to the table recognises the cards, chips and bills, follows each round and publishes what happens over MQTT. You configure it in the device panel, a web page the device serves. The **Manual** link at the top right of the panel opens this manual for the software version installed on that device.

1.1 Supported games and modules

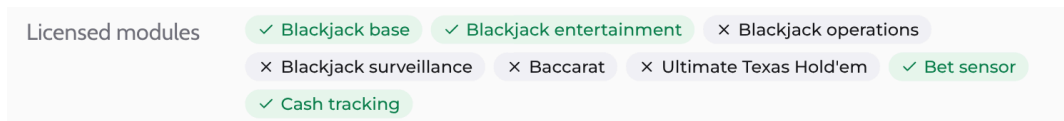
Blackjack is the supported game. The license of a device enables modules for each game and for the trackers that work with any game:

Module	Shown in the panel as
Spades AI Base Module for Blackjack	Blackjack base
Spades AI Entertainment Module for Blackjack	Blackjack entertainment
Spades AI Operations Module for Blackjack	Blackjack operations
Spades AI Surveillance Module for Blackjack	Blackjack surveillance
Spades AI Module for Baccarat	Baccarat
Spades AI Module for Ultimate Texas Hold'em	Ultimate Texas Hold'em
Spades AI Bet Sensor Module	Bet sensor
Spades AI Cash Tracking Module	Cash tracking

The Base module unlocks the Blackjack configuration under “Table Configuration”. Baccarat and Ultimate Texas Hold'em are not supported yet; their modules are reserved for them.

The bet sensor reads the chips on the bet spots, cash tracking reports the bills put down on the table. Both work independently of the game, and each runs only with its module: without it, the panel does not offer its configuration (the bet sensor under “Table Configuration”, the “Cash model” under “System Configuration” → “Models”) and the device does not track it. When the license switches a module off, the Configuration API also refuses to configure it (403). A license issued before a module existed enables that module.

“System Configuration” → “License” shows which modules the installed license enables:



1.2 Provided binaries

Spades Vision publishes every release to a shared Google Drive folder, in a subfolder named after the version, e.g. v3.1.0. It contains:

File	Description
spades-vision-initrd-flash-<device>-<version>	Flashes the UEFI firmware to a Jetson Orin NX. Self-contained, runs on any x86-64 Linux PC.
spades-vision-installer-<device>-<version>.iso	Installs or updates the OS on a Jetson Orin NX from a USB stick.
Spades Vision Jetson Manual_<version>.pdf	This manual.

`<device>` names the device configuration, `<version>` the release and the commit it is built from, e.g. `v3.1.0-32a5d09`. [Chapter 3](#) explains how to use both binaries.

1.3 Feedback to the developers

If something doesn't work as expected, or you have a request for an improvement, email info@spadesvision.com. For a problem with a device, include its machine ID and system version ("System Configuration" → "License") and when the problem happened. [Section 9.2](#) explains how to download the device's diagnostics.

2 Network requirements

Every table has its own device. The camera connects to the device over USB, so no camera video crosses the network. The network carries the device panel, the APIs and the live video shown in the panel. The device uses IPv4.

You can connect a device in two ways: a direct cable to one client PC (Option A), or a dedicated VLAN shared by all devices and client PCs (Option B).

2.1 Internet access

The device needs internet access once, when you commission it: it sets its clock over NTP and downloads its development license (see [Chapter 4](#)). Afterwards it runs without internet access. With internet access it also offers remote support, which you can block. The ports are listed under “Outbound from the device” below.

2.2 Network settings of the device

Each Ethernet port has one of three settings, in the device panel under “System Configuration” → “Networking”:

- **Factory default:** the address from DHCP. This is how the device ships.
- **Automatic (DHCP):** the same, set explicitly.
- **Static IP:** an address, a prefix length and an optional gateway. Only one port can have a gateway. The device has no DNS server setting: with a static IP it resolves names through 9.9.9.9, so enter your MQTT broker as an IP address or allow DNS to 9.9.9.9.

In every setting the port also keeps the link-local address 169.254.1.123. If another device on the same cable or network already has that address, the device picks another free address in 169.254.0.0/16.

To set a static IP:

1. Open the device panel and go to “System Configuration” → “Networking”.
2. In the card of the Ethernet port, select “Static IP”.
3. Enter the address and the prefix length (24 for the netmask 255.255.255.0). Enter a gateway if the device has to reach other networks, for example a broker or the internet.
4. Click “Apply”. The “Now” line of the card shows the addresses the port has. The setting is kept across software updates.

Networking

How each network interface gets its address.

enP8p1s0

Link up

Now 169.254.1.123, 10.0.4.57 via 10.0.4.1

☐ Factory default
 ☐ Automatic (DHCP)
 ☒ Static IP

Address

/

Gateway (optional)

Apply

2.3 Option A: direct cable (link-local)

One Ethernet cable connects the device to the table's client PC. No switch, DHCP server or DNS is involved.

- The device has the address `169.254.1.123`.
- Leave the client PC's Ethernet port on automatic addressing (DHCP). With no DHCP server on the cable, it gives itself an address in `169.254.0.0/16`.
- The client PC opens the device panel at `https://169.254.1.123` and subscribes to the MQTT API on the device's own broker at `169.254.1.123`, port 1883.
- Nothing else reaches the device: configuration and support need the client PC or a laptop at the table.

2.4 Option B: dedicated VLAN (recommended)

All devices and the client and management PCs share one VLAN. You configure and support every table from any PC in the VLAN. We recommend this for every installation with more than one table.

- One 1 GbE access port per device, all in the same VLAN (one L2 broadcast domain).
- A DHCP reservation per MAC address, or a static IP set on each device (see above).
- No client or port isolation inside the VLAN: devices and PCs talk to each other directly.
- Multicast passes inside the VLAN, for mDNS (`224.0.0.251`, UDP 5353). Each device publishes the name `<prefix>-<machine ID>.local`, for example `sv-demo-1a2b3c4d5e6f.local`. We tell you the prefix of your installation. The machine ID is shown in the device panel under "System Configuration" → "License". The device also sends this name to the DHCP server as its hostname. If multicast is blocked, create a DNS A record per device or use its IP address.
- The device's own MQTT broker is not reachable from a VLAN outside `192.168.0.0/16` (see the table below). Run a broker in the VLAN and enter it in the device panel under "System Configuration" → "MQTT API". The device then publishes to that broker and your systems subscribe there.

2.5 Ports

Inbound to the device

Port	Protocol	Purpose
443	TCP	Device panel, Configuration API, diagnostics download and live video signalling (HTTPS and secure WebSocket). The certificate is self-signed.
80	TCP	The same as port 443 over plain HTTP, without a redirect.
50000–50099	UDP, TCP	Live video to the browser (WebRTC media). Accepted from any address.
1024–65535	UDP	Video from a browser's camera to the device (WebRTC). Accepted only from private addresses: <code>10.0.0.0/8</code> , <code>172.16.0.0/12</code> , <code>192.168.0.0/16</code> and <code>169.254.0.0/16</code> .
1883	TCP	The device's own MQTT broker. Accepted only from <code>169.254.0.0/16</code> and <code>192.168.0.0/16</code> .
5353	UDP	mDNS, <code>.local</code> names (multicast).
22	TCP	SSH, for maintenance by Spades Vision. You can block it.
41641	UDP	Remote support, direct path. Optional.

Outbound from the device

Port	Protocol	Destination	Purpose
1883 or 8883	TCP	Your MQTT broker	MQTT API, when you enter a broker in the device panel. 8883 is the usual port with TLS; the device uses the port you enter.
53	UDP, TCP	DNS servers from DHCP; 9.9.9.9 with a static IP	Name resolution.
123	UDP	NTP servers from DHCP, else 0.nixos.pool.ntp.org to 3.nixos.pool.ntp.org	Clock. Needed once at commissioning.
443	TCP	admin.spadesvision.com	Development license download at commissioning.
443	TCP	*.tailscale.com	Remote support (Tailscale). Optional.
3478, 41641	UDP	Internet	Remote support, direct path. Optional; falls back to TCP 443.

If diagnostics upload is enabled for your installation, the device also uploads one diagnostics archive a day to Google Cloud Storage (storage.googleapis.com, TCP 443).

2.6 Live video

The device sends live video straight to the viewer's browser; there is no relay server. The browser opens the panel on TCP 443, which also carries the signalling, and then receives the video on the device's ports 50000–50099, over UDP or, where UDP does not get through, over TCP. Video only flows while a page with a live view is open, and its bitrate adapts to the link.

A viewer outside the devices' VLAN needs a firewall rule from the viewer's network to every device:

- TCP 443: the panel and the live video signalling.
- UDP 50000–50099: the live video.
- TCP 50000–50099: the live video where UDP is not allowed. UDP gives the smoother picture.

The browser connects to the device, so the replies only have to be allowed back (a stateful firewall does that by itself).

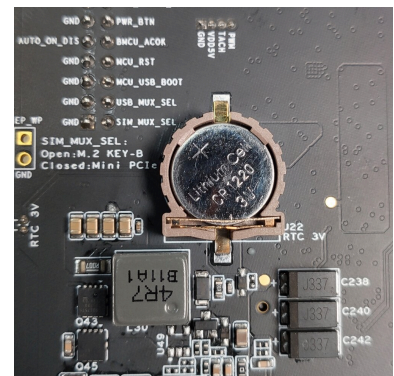
3 Fresh installation on the Seeed Studio reComputer J4011

This chapter takes a new reComputer J4011 (Jetson Orin NX) to a running Spades Vision system:

1. Install the RTC battery.
2. Flash the UEFI firmware. Needed once per device.
3. Write the installer ISO to a USB stick.
4. Boot the device from the USB stick. The same stick also updates devices that are already installed.
5. Check the device panel.

3.1 What you need

- The reComputer J4011
- A 3 V CR1220 coin cell battery
- A jumper wire, or anything metallic to short two pins
- A USB Type-C data cable
- A Linux PC (x86-64) for the firmware flash. Any distribution works.
- A USB stick for the installer. It is overwritten completely.
- An Ethernet connection to a network with internet access
- From the release folder:
 - the firmware flashing script
`spades-vision-initrd-flash-<configuration>-<version>`
 - the installer ISO
`spades-vision-installer-<configuration>-<version>.iso`



The RTC battery in its socket

`<configuration>` names the system build for your devices and `<version>` the software version. The release folder holds exactly one flashing script and one installer ISO.

3.2 Step 1: Install the RTC battery

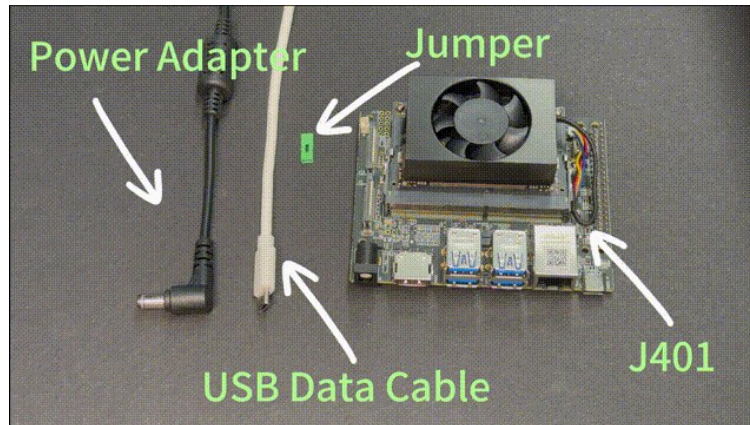
The license check needs the correct time. The RTC battery keeps the clock running while the device has no power; without it the device starts at 1970 after every power cut.

1. Open the reComputer and unscrew the Jetson board to reach the RTC socket.
2. Insert the CR1220 battery as shown above, positive (+) side up.

WARNING An unpowered device drains the battery quickly. Remove the battery from devices that are stored on the shelf.

3.3 Step 2: Flash the UEFI firmware

The firmware Seeed Studio and NVIDIA ship cannot boot the Spades Vision OS, so replace it once per device. Updates later on do not need this step.



Force Recovery Mode, from the Sseed Studio wiki's [reComputer J4012 page](#), which applies to the J4011 too

1. Put the Jetson in Force Recovery Mode.

1. Connect the Jetson to the Linux PC with the USB Type-C data cable.
2. Short the FC REC pin and a GND pin with the jumper wire.
3. Connect the power adapter while the pins are shorted, then remove the jumper.
4. On the Linux PC, check that the Jetson is in recovery mode:

```
lsusb | grep -i nvidia
```

SHELL

An NVIDIA Corp. device is listed.

2. Run the flashing script in the folder you downloaded it to:

```
chmod +x spades-vision-initrd-flash-  
sudo ./spades-vision-initrd-flash-*
```

SHELL

The script is self-contained. On its first run it unpacks its dependencies to `/root/.cache/tmpx-...` without printing anything, which takes up to 5 minutes on older PCs. Do not interrupt it, or the cache is left broken (see [“File not found”](#)).

3. Wait for the script to end with:

```
Device reported flashing succeeded. Device should be rebooting.
```

OUTPUT

The whole flash takes less than 10 minutes.

3.4 Step 3: Create the installer USB stick

Write the installer ISO to the USB stick as a raw image. This erases everything on the stick.

On Linux: find the stick with `lsblk` (usually `/dev/sdb`) and write to the whole disk, not to a partition (`/dev/sdb`, not `/dev/sdb1`):

```
sudo dd if=spades-vision-installer-*.iso of=/dev/sdX \
bs=4M conv=fsync status=progress
```

SHELL

On Windows: use [Rufus](#) in DD Image mode:

1. Start Rufus and choose the USB stick under **Device**.
2. Under **Boot selection**, press **SELECT** and open the installer ISO. Leave the other settings as they are.

3. Press **START**. Rufus asks how to write the image: choose **Write in DD Image mode**, not the recommended ISO Image mode. A stick written in ISO Image mode does not boot on the Jetson.
4. Confirm that the stick is erased. Rufus shows **READY** when it is done.
5. Windows may then ask to format the stick, because it cannot read it. Press **Cancel**.

Other tools that write the image unchanged, such as USBImager or Raspberry Pi Imager, also work.

balenaEtcher works on some PCs, but recent versions often fail with *Error opening source: (0, h.requestMetadata) is not a function*. This is a bug in balenaEtcher, not a broken ISO. Right-click the balenaEtcher program and choose **Run as administrator**, or use Rufus. With installer ISOs older than v2.13.1, balenaEtcher warns about a missing partition table; press Continue.

3.5 Step 4: Install or update the Spades Vision OS

The installer runs on its own as soon as the device boots from the stick; you need no keyboard or display. It checks the device's disk and takes one of two paths:

- **Existing system → update in place**. The system is replaced by the version on the stick. The license, the table configuration, the network settings and the recorded data are kept.
- **Blank disk, or a disk that does not mount → fresh install**. The disk is wiped, partitioned and installed from scratch.

To update devices to a new release, write the new ISO to the stick and repeat this step on each device.

Steps

1. Connect the device's Ethernet port to your network or directly to your PC.
2. Insert the USB stick into any USB port of the Jetson and power the device on, or reboot it.
3. Wait for the install to finish: about 5 minutes on a USB 3 stick, less for an update. You can follow it on the installer page (see below).
4. When the installer reports **Update complete** or **Fresh install complete**, remove the USB stick before the device reboots: 60 seconds after an update and 10 seconds after a fresh install. A stick left in can start the installer again on every boot.
5. Continue with step 5.

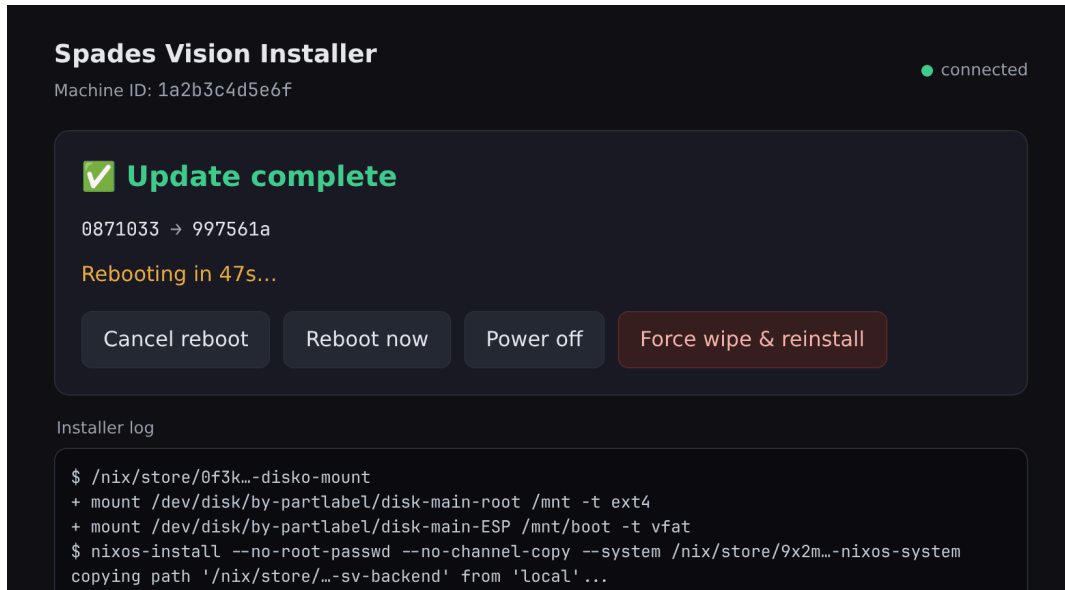
If an install step fails, the installer stops, shows **Installation failed** and does not reboot, so the device never boots a half-installed system. Take a screenshot of the installer page, including its log, and send it to Spades Vision.

The installer page

The installer serves a status page over plain HTTP (not HTTPS) at:

- `http://169.254.1.123` when your PC is connected directly to the device,
- `http://sv-installer-<hardware ID>.local` on networks that pass mDNS, with the hardware ID the display shows (see below),
- the address the DHCP server gave the device, from the router's lease list.

When the installer updates a device that has a static IP configured, it uses that IP as well. A display connected to the HDMI port shows the device's hostname, hardware ID, current time and IP addresses.



The installer page after an update. The line with the arrow shows the installed version and the version on the stick.

The page shows the progress, the installer log and these buttons:

- **Cancel reboot, Reboot now, Power off:** control the automatic reboot after a successful install.
- **Force wipe & reinstall:** erases the disk and installs fresh, also on a device that would only be updated. Use it to start a device clean.

WARNING Force wipe & reinstall erases the license, the table configuration and all recorded data. Download the license and the table configuration from the device panel first if you need them again.

3.6 Step 5: Check the device panel

1. Connect the device to a network with internet access. The device then sets its clock over NTP and stores the time in the RTC. If it has no license yet, it downloads a development license (see [Chapter 4](#)).
2. Find the device's address:
 - Directly connected to your PC: 169.254.1.123. With more than one Spades Vision device on the same link, the others take a random 169.254.x.x address.
 - On a network: the address in the router's DHCP lease list, under the device's hostname. On networks that pass mDNS, the device is also reachable as <hostname>.local.
 - A display connected to the HDMI port shows the hostname and the IP addresses.
3. Open <https://<address>> in a browser. The device uses its own certificate, so the browser warns that the connection is not private; accept to continue. If your installation uses sign-in, sign in with the login Spades Vision gave you.
4. On **System Configuration**, check the **License** section:
 - **Current time** shows the correct date and time.
 - **System version** matches the <version> in the release file names.
 - The license status is **Valid**.
5. For a deployed device, upload its license file as described in [Chapter 4](#).

The device is now ready for the table setup.

3.7 Troubleshooting the firmware flash

USB write timeout

The flash stops early with:

```
[ 0.1949 ] Sending bct_br  
[ 0.1949 ] ERROR: might be timeout in USB write.  
Error: Return value 3  
...  
Reading board information failed.
```

OUTPUT

USB auto-suspend on the Linux PC interrupts the transfer. Disable it until the PC reboots:

```
echo -1 | sudo tee /sys/module/usbcore/parameters/autosuspend
```

SHELL

Then unplug the USB cable, put the Jetson into Force Recovery Mode again and rerun the script. Repeat the command after every reboot of the PC.

File not found

After an interrupted first run, the script fails with:

```
... ./nix/store/vj3svda3d07wbhh5d0hnmvjsfzfnf3q4k-startup: No such file or directory
```

OUTPUT

The dependency cache is incomplete. Delete it and run the script again, which unpacks it anew:

```
sudo rm -r /root/.cache/tmpx-  
sudo ./spades-vision-initrd-flash-
```

SHELL

4 Licensing system

4.1 How licensing works

A device detects and tracks only while it has a valid license file (also called machine file). A license file is bound to one device and is valid for a limited time. Without a valid license, card, chip and cash detection and tracking stop. The device checks the license continuously: a new license takes effect within seconds, without a restart.

Machine ID. The device is identified by its machine ID, a 12-character code derived from the Jetson's serial number. It stays the same when you reinstall the software. You find it:

- in the device panel: at the top left of every page, next to the machine name, and under “System Configuration” → “License” → “Machine ID”,
- on the installer web page (“Machine ID”) and on a screen connected to the device (“Hardware ID”),
- at the end of the device's network name, e.g. sv-...-1a2b3c4d5e6f.local.

Time. A license is valid from its issue date to its expiry date, so the device must know the current time. It sets its clock from the internet (NTP) and keeps it in the real-time clock (RTC), which the coin cell battery powers while the device is off. **Connect a freshly installed device to the internet at least once**, so that it sets its clock. Without the RTC battery, the device loses the time on every power cut and rejects its license.

4.2 License types

Type	Use
Development	Downloaded by the device on its own (see below). Detection and tracking work only during the first hour after each start; restart the device for another hour. For installing and testing only.
Trial	Valid for 3 months. One trial license per machine ID.
Paid	Valid for 1 year. Counts towards the monthly license costs.
Internal	Valid for 1 year, at most 5 devices. For showrooms and trade shows.

Your account can have several licenses of one type, for example one paid license per region to separate the license costs.

Development license. A device without a license file downloads a development license from admin.spadesvision.com as soon as it is online. Accounts with access to that development license see the device under “Recent machine activations” for 20 days, with its machine ID and its local IP address, which helps you find the device on the network.

4.3 Get a license file

You create license files on the admin platform at admin.spadesvision.com. Spades Vision creates your account, sends you its login and enables the licenses you may issue. You can change your password on the platform.

1. Copy the machine ID from the device panel (“System Configuration” → “License”).
2. Log in to the admin platform. The “Licensing” tab opens.
3. Under “License control”, choose the license under “License”. The platform shows how long a file from this license is valid and its type.
4. Enter the **Machine ID** and a **Name** that tells where the device stands, e.g. “Grand Casino BJ 06”. The panel shows this name as “Machine name” and the Excel reports of the device carry it.

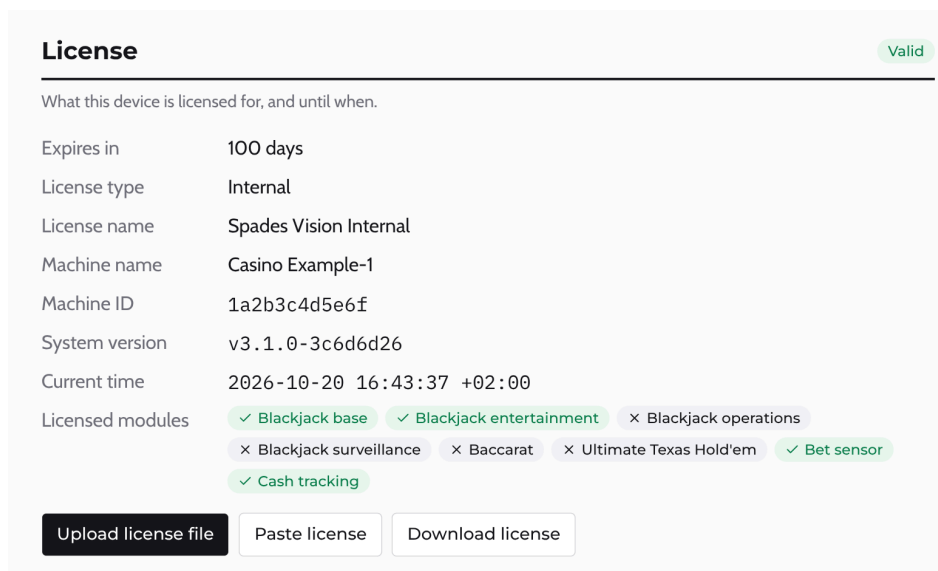
5. Tick the modules the device is licensed for, under “Blackjack”, “Other games” and “Trackers” (see “Supported games and modules” in the introduction):
 - any Blackjack module requires **Base**,
 - **Surveillance** requires **Operations**,
 - a Trial or Development license must enable all modules.
6. Click “Download license file”. Your browser saves `machine_file_<machine ID>_expires_<expiry date>.lic`.

To renew a license before it expires, download a new file for the same machine ID and upload it. A trial license cannot be issued twice for one machine ID.

4.4 Upload the license file

WARNING Uploading replaces the license on the device, even if the new file turns out to be invalid. Click “Download license” first to keep a copy of the current one.

1. Open the device panel and go to “System Configuration” → “License”.
2. Click “Upload license file” and choose the `.lic` file. Alternatively, open the file in a text editor, copy its whole content and click “Paste license”. That button is only there when the panel is opened over https, because browsers allow reading the clipboard only there.
3. Check the result. A working license looks like this:



License Valid

What this device is licensed for, and until when.

Expires in	100 days
License type	Internal
License name	Spades Vision Internal
Machine name	Casino Example-1
Machine ID	1a2b3c4d5e6f
System version	v3.1.0-3c6d6d26
Current time	2026-10-20 16:43:37 +02:00
Licensed modules	✓ Blackjack base ✓ Blackjack entertainment ✗ Blackjack operations ✗ Blackjack surveillance ✗ Baccarat ✗ Ultimate Texas Hold'em ✓ Bet sensor ✓ Cash tracking

The badge next to “License” reads **Valid**. “Expires in”, “License type”, “Machine name” and “Licensed modules” match what you chose on the admin platform, and “Current time” is correct.

“Download license” saves the installed license as `license.lic`. A factory reset and a software update keep the license; reinstalling with a wiped disk removes it, so upload your copy afterwards.

4.5 When the license is not valid

The panel shows the reason under “The license is not usable”:

Message contains	What to do
No license on this machine	Upload a license file.
Fingerprint is invalid	The file was made for another device. Issue a file for this device’s machine ID.

Message contains	What to do
no license key matched	The file is damaged or comes from a license this device does not accept. Download it again; if that does not help, contact Spades Vision.
issued in the future	The device clock is behind. Connect the device to the internet and check the RTC battery.
Expiry is in the past	The license has expired, or the device clock is ahead. Check “Current time”; if it is right, upload a new license.
Development license: valid only for the first hour	Restart the device for another hour, or upload a trial, paid or internal license.

5 Setup and usage guide

This chapter takes you through the device panel and the configuration of a table: camera calibration, blackjack, bet sensor, copying a configuration to other tables, and the Excel reports. Side bets have their own chapter, “Side bet configuration”.

5.1 Lens focus guidelines

The focus has to be even across the whole table, and sharpest at the side farthest from the camera.

1. Focus roughly until the table is clearly visible: in “Live View”, turn on “Camera”.
2. Fine-tune with the “Sharpness” step of the camera calibration (section “Camera calibration” below). It measures the focus on the calibration board wherever you lay it and scores it green, yellow or red.

The camera’s dome cover changes the focus. Focus with the dome mounted, and check again after you turn or replace it.

5.2 Device panel

Open the device’s address in a browser: see “Step 5: Check the device panel” in the installation chapter for how to find it, and [Section 9.7](#) if the panel does not load. Depending on the installation, the panel asks you to sign in; use the login Spades Vision gave you. The device’s own display never needs a login.

Table 12 sv-demo-1a2b3c4d	System Configuration	Table Configuration	Live View	Analytics	Display Examples	Manual	admin
---------------------------	----------------------	---------------------	-----------	-----------	------------------	--------	-------

The bar at the top shows the machine name and ID, a link to this manual, and the panel’s five pages. An admin login sees all five; any other login sees only “Live View”, “Analytics” and “Display Examples”.

Page	What it does
System Configuration	<p>The device itself:</p> <ul style="list-style-type: none"> • Stream: where the frames come from. Tables use “Basler Camera”. “Recorded Video” plays a recording instead, for tests (see Chapter 6); the other types are for demos. • Models: the detection models. Leave them as delivered unless we ask you to change them. • Recording quality: how much detail the round videos keep. Changing it deletes the videos recorded so far. • License: status, machine ID and licensed modules, see Chapter 4. • Networking: the address of each Ethernet port, see Chapter 2. • Timezone: changing it restarts the Spades Vision Server. • MQTT API: the broker the device publishes to, see Chapter 8. • Reboot and reset: a factory reset erases everything the Spades Vision Server has saved (table configurations, rounds, videos and settings) except the license. Diagnostics are kept.
Table Configuration	Everything about the table, see Section 5.3 .
Live View	The live round on a digital table, the camera and detections streams, and the past rounds with their videos. Chapter 6 shows how to use it to check a table.
Analytics	Excel reports of the recorded rounds, see Section 5.4 .
Display Examples	The screens players see at the table, showing this table live. “Open full size” opens one in its own window to put on a screen.

5.3 Table configuration

A table configuration holds everything the system knows about one table: the camera calibration and layout picture, the boxes and bets, the blackjack rules, the bet sensor spots, the side bets and the end of the gaming day. A device can store several, for example one per layout a table is switched between, and exactly one of them is active: the table runs with it.

The “Table Configuration” page works on the active configuration, top to bottom:

- **Table configuration:** the list of stored configurations. Choosing one makes it active at once and restarts the tracking with it, so switch between rounds. “Add new...” creates an empty configuration and makes it active. “Rename or delete” lists them all; the active one cannot be deleted.
- **General:** “Gaming day finishes at”, the time of day at which one gaming day ends and the next begins (default 12:00). Reports group rounds by gaming day.
- **Camera calibration, Blackjack and Bet sensor:** step-by-step wizards. The panel shows only the modules the license includes. A wizard that was completed before opens on its “Finished” step; click any step to go back to it.
- **Backup and restore:** download and upload the configuration and its layout picture, see “Calibrating several tables with the same layout” below.

Configure in this order: camera calibration first, then blackjack and the bet sensor, which place their positions on the calibrated layout picture.

Camera calibration

Calibrate the camera when you install a table and every time the layout cloth changes. The calibration fixes which part of the camera picture the system looks at, and records a picture of the empty layout. The system later finds the camera’s position by matching that picture, so the camera can move, for example when the display it is mounted on is turned.

Before you start, print the calibration board: choose the paper on the “Calibration board” step, click “Download board PDF” and print it at actual size, without scaling. A board printed smaller makes the system zoom in too far.

1. **Start:** click “Start calibration”. The automatic camera tracking pauses until the layout is saved.
2. **Calibration board:** lay the board on the felt next to the dealer’s chip tray (the float), on the players’ side and centred on it, as the drawing under “How to place the board” shows. Place it precisely: it is the reference for every position. Click “Find calibration board”. The view zooms in on the board. If the board is not found, move it closer to the camera and try again, then move it step by step into place. Use “Zoom in”, “Zoom out” and “Whole image” until every betting spot is in the picture, with as little margin as possible. Click “Save camera position”.

Calibration steps

✓ Start

2 Calibration board

3 Camera settings

4 Sharpness

5 Layout

6 Check

7 Finished

Back

Save camera position

Calibration board

Place the calibration board just above the float, find it, and zoom until every betting spot is in the frame.

> How to place the board

Board printed on

A4

Letter

Download board PDF

Print it at actual size, without scaling.

Find calibration board

Whole image

Zoom in

Zoom out

- Camera settings** (Basler camera only): lay a few cards face up where the dealer deals them. Choose the brightness until the pips are crisp and neither the layout nor the cards are washed out; the camera exposes automatically, so let the picture settle after each change. If a lamp washes out shiny cards, choose the preset “Glare”, or “Strong glare”, and fine-tune the brightness from there. “Reset to default” restores the delivered brightness.
- Sharpness**: lay the board flat on the felt, stand clear of it and wait for the live reading to settle. Move it through the whole playing area: every betting position, the dealer’s area, the corners and the far edge. The four tiles keep the reading of each quarter of the picture. Green is good, yellow is marginal, red is too soft for reliable detection. If the whole table is soft, refocus the lens; if only some positions are, turn the dome. Lock the lens and measure again afterwards: tightening the lock ring shifts the focus. “If a position reads yellow or red” in the panel explains each fix.

Sharpness

Back

Sharpness check done

- Place the calibration board flat on the felt, inside the playing area. Keep it flat: a curled or propped-up board measures the board, not the camera.
- Stand clear so you do not shadow the board, and wait for the reading to settle.
- Read the score: green is good, yellow is marginal, red is too soft.
- Move the board to the next position and repeat.

Cover the whole felt: every betting position, the dealer area, and the corners and far edge, not just the middle. The camera enclosure can have soft spots that only affect part of the image.

Live reading

●

—

/ 100

no board in view

● Top left

Not checked

● Top right

Not checked

● Bottom left

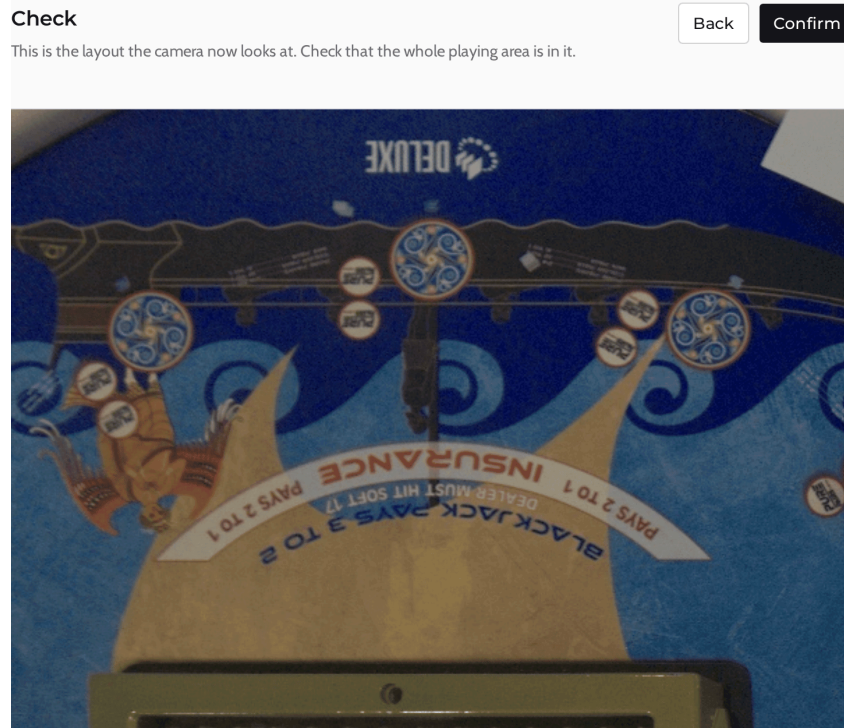
Not checked

● Bottom right

Not checked

> If a position reads yellow or red

- Layout**: remove the board. Check in the picture that the leftmost and rightmost players stay in frame even with splits and many cards, and that the dealer’s cards are fully visible. If not, go back to “Calibration board” and correct the board’s position. Click “Save layout and next”: this saves the picture of the empty layout and resumes the automatic camera tracking.
- Check**: the saved layout picture. Check that the whole playing area is in it and click “Confirm”.



The **Camera position** card above the steps shows which camera position is in use:

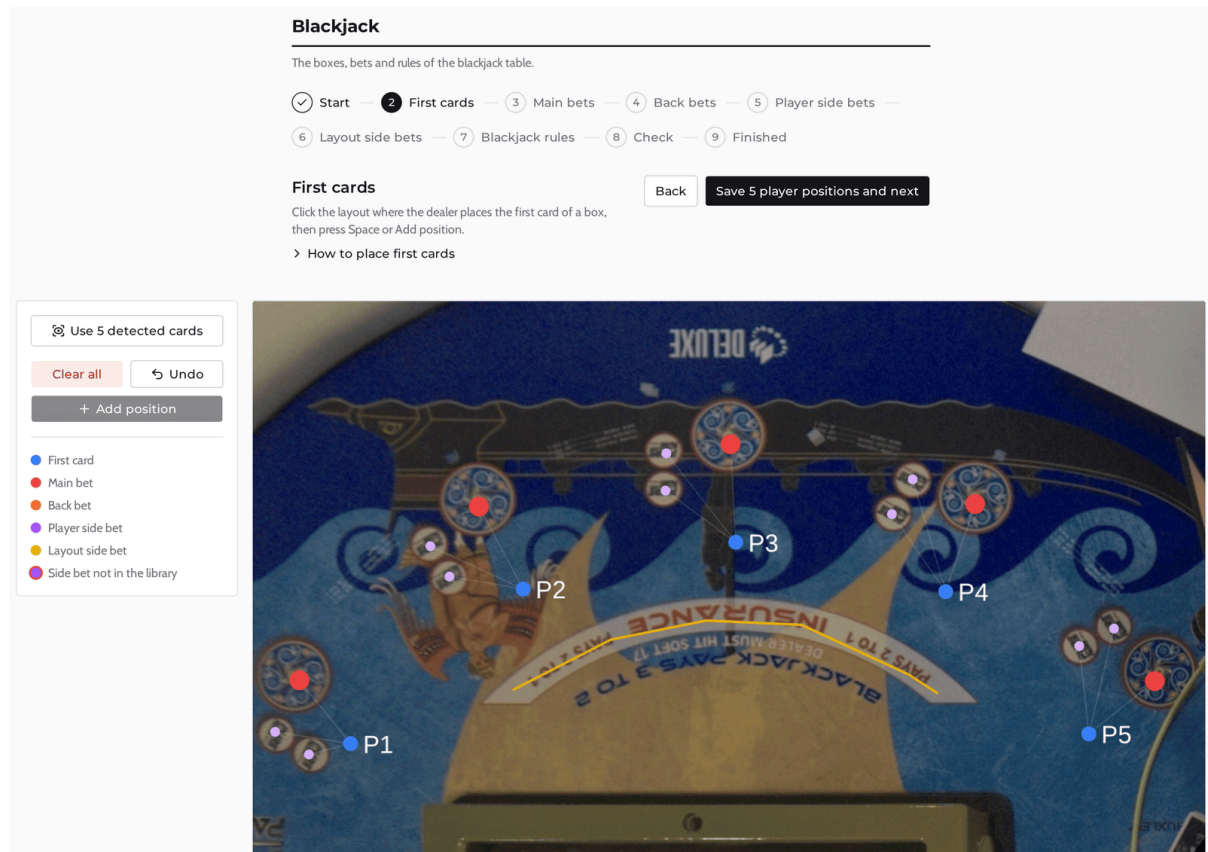
- **Found from the layout:** the automatic tracking found the camera's current position.
- **Saved position:** the position saved during calibration.
- **Saved position (none found that could be used):** tracking is on, but the layout was not found, for example because objects cover it.
- **Not calibrated (whole image):** no calibration is saved.

“Track the camera position automatically” is on by default. The system then looks for the layout every few seconds between rounds, while no cards are on the table, and immediately after a configuration or layout change. “Allowed movement” sets how far the found position may be from the saved one: “Any movement” uses every position found, “Small corrections only” rejects positions far from the saved one, which is almost always the layout matched one seat off. “Default” follows the setting of your installation. Use “Small corrections only” for a camera that never moves. If the tracking still picks wrong positions, turn it off: the saved position is then used. “Find camera position now” runs one search immediately and shows the result.

Blackjack configuration

Shown only if the license includes blackjack. Each step draws the layout picture of the calibration; you mark positions on it with the mouse. Click the layout, then press Space or “Add position”. “Undo” reverts the last placement. “Save ... and next” saves the step to the table at once. For bets, choose the shape: “Point” (a chip-sized spot), “Line” or “Polygon” (click the corners of a printed box). Each step’s “How to ...” explains its details.

1. **Start:** click “Start configuration”.
2. **First cards:** mark where the dealer places the first card of each box. Boxes are numbered from left to right, so P1 is the leftmost player. Place nothing for the dealer. Instead of clicking, you can lay a card on each box and click “Use ... detected cards”. Right-click a first card to remove it.



3. **Main bets:** mark each box's main bet. It goes to the nearest box.
4. **Back bets:** mark the back-bet spots, if the layout has them. Each goes to the box whose main bet or first card is nearest. "Clear player" or a right-click removes all of a player's back bets.
5. **Player side bets:** choose the side bet, then mark its spot at each box. "Manage side bets" adds your own side bets, see [Chapter 7](#).
6. **Layout side bets:** side bets the whole table shares, such as insurance. Choose the side bet and mark its area.
7. **Blackjack rules:** the dealer procedure (US, UK or Europe), the number of decks, surrender, the blackjack payout, dealer peek, hit on soft 17, and the double and split rules. The rules decide how the system settles hands. Click "Save rules and next".

Blackjack rules

Set the blackjack rules of this table, then save them.

[Back](#)
[Save rules and next](#)

General

Dealer procedure

US

Number of decks

6

Surrender

Late

Blackjack pays

3 to 2

☐ Dealer peeks
 ☒ Dealer hits on soft 17

Doubles

Double allowed on

Any card

Double after split

Not on split aces

☒ Double on soft hands

Splits

Split allowed on

Any card

Maximum resplits

4

Split aces

Always

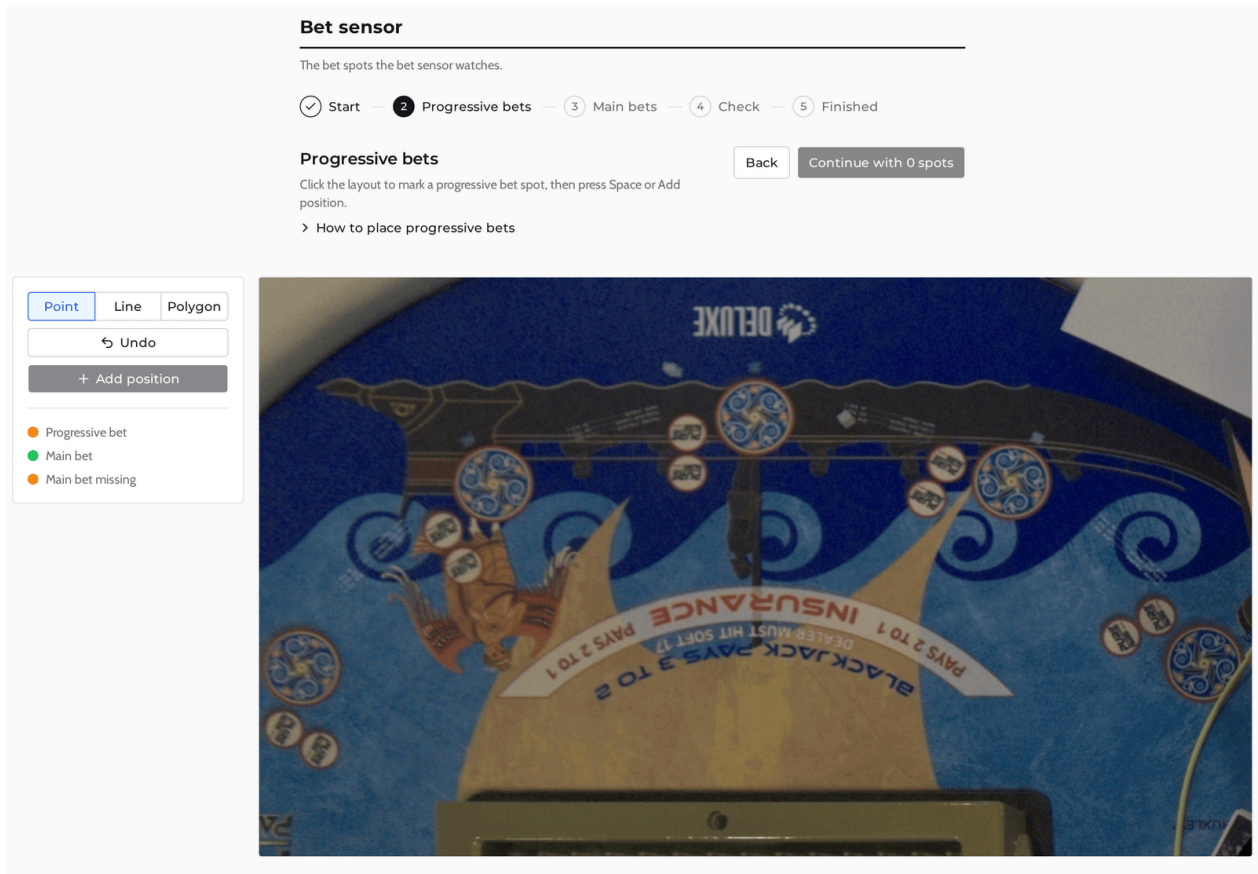
☐ Can hit split aces

- Alerts:** how much each alert counts at this table. “On” shows it with a badge and counts it, “Review only” keeps it on the round for review without a badge or count, and “Off” shows it nowhere but still records it. Click “Save alerts and next”.
- Check:** all boxes, bets and side bets on the layout. Click “Confirm”.

Bet sensor configuration

Shown only if the license includes the bet sensor. Unlike the blackjack wizard, nothing is saved until the last step.

- Start:** click “Start configuration”.
- Progressives:** list the progressives a seat can play, each with a name and its denomination in whole currency units, for example “\$5 jackpot” at 5 and “\$10 jackpot” at 10. Every seat gets one spot for each. The order is the order of the sensor numbers.
- Main bets:** mark each seat’s main bet. Every main bet is a seat, numbered from left to right. Mark a printed main-bet box as a “Polygon” around the box: a “Point” only counts chips right on it. Right-click a main bet to remove its seat.
- Progressive spots:** pick a progressive under “Progressive to place” and mark its spot at every seat. A spot goes to the seat with the nearest main bet. Right-click a spot to remove it.
- Check:** check every seat and its spots. “Sensor numbers” shows the `player_index` each spot has in `side_bet_sensor`: one pass of seats per progressive, so with 7 seats and two progressives, 0 to 6 are the first progressive and 7 to 13 the second. Click “Confirm and save”.



Calibrating several tables with the same layout

Instead of configuring each table from the start, copy a finished configuration to tables with the same layout cloth.

1. On the configured device, open “Table Configuration” and click “Download configuration and layout” under “Backup and restore”. Your browser saves two files: `config.json` and `calibration.jpg`.
2. On each other device, open “Table Configuration”:
 1. Uploading replaces the active configuration. To keep it, first choose “Add new...” in the configuration list and create a new one.
 2. Click “Upload configuration” and choose `config.json`. A configuration from an older software version is converted on upload.
 3. Click “Upload layout” and choose `calibration.jpg`.
 4. Run the “Camera settings” and “Sharpness” steps of the camera calibration: every camera needs its own brightness and focus.
 5. Check the “Camera position” card: with tracking on, it shows “Found from the layout” after a few seconds. If the layout is not found, or the positions do not sit right in the blackjack “Check” step, calibrate the camera on this table and check the positions again.

Recovering a configuration from diagnostics

The device diagnostics contain the table configurations, so you can restore a configuration from them, also onto another device.

1. Download a diagnostics archive as described in [Section 9.2](#).
2. Extract the `.tar` file.
3. The active configuration is in `state/config/`: `config.json` and `calibration.jpg`. The other stored configurations are in `state/table_configs/<id>/`, each with a `meta.json` that names it.

4. Upload the two files as in “Calibrating several tables with the same layout”.

5.4 Analytics and Excel reports

The “Analytics” page downloads the recorded blackjack rounds and cash transactions as an Excel file. A table that only tracks cash gets a report of its transactions.

Excel report

Download the rounds of one or more gaming days, or of any time span, as an Excel file.

Gaming days

Click a day, then **Shift** -click another to take every day between. Keys: arrows move, **Space** selects, **Shift** + arrows extend.

Week of 14 Sep

Thu 17 Sep
56 rounds

Week of 21 Sep

Fri 25 Sep
10 rounds

Sun 27 Sep
18 rounds

Week of 12 Oct

Thu 15 Oct
36 rounds

Selected 15 Oct · 36 rounds

From

To

10/15/2026, 12:49 PM

10/15/2026, 01:06 PM

[Download Excel report](#)

1. Choose the time span:
 - Click a gaming day, or click one and Shift-click another to take every day in between. The page lists the gaming days of the last 30 days that have rounds or transactions, with how many of each. The selection fills in “From” and “To” with the first and last of those days.
 - Or set “From” and “To” yourself, to any time span.
2. Click “Download Excel report”. The file is named SV Report <machine name> <gaming day>.xlsx, or ... <first day> to <last day>.xlsx for several days.

The device keeps the rounds of about the last 90 days of a busy table; older rounds are deleted.

The report has these sheets, the round sheets when there are rounds and the cash sheet when there are transactions:

- **Round Data:** one row per round, including rounds that look incomplete, so they can be looked into. The first two rows name and describe the columns. Per round: start and end time, duration, number of hands and players, system issues detected, the dealer’s hand, and per box the hands, cards, actions, decision strength, payout multiplier, bet and result, followed by each side bet. The bet columns hold placeholder amounts (10 per main bet, 5 per side bet played): the system does not read the bet amounts. Enter the real amounts and the result columns and the dealer’s result recalculate.
- **Analytics - Round Speed:** rounds and hands per hour, table open, playing and idle time, the time spent in each phase of a round, and the same per number of active boxes. Only complete rounds count.
- **Analytics - Side Bets:** how often each side bet was played compared with how often players could have played it, with an explanation of how opportunities are counted.

- **Alerts:** per kind of alert how many were raised and how they were reviewed in the Live View, with the precision; then every alert with its review, note, who reviewed it and when, and every alert marked as missed.
- **Cash Transactions:** every transaction with its notes per denomination and what they were worth, then per gaming day the notes, the amount and the largest transaction.

6 Testing procedure

Test every table after you install and configure it ([Chapter 5](#)), and again after each update: first the whole system with real cards at the table, then, if you want to check the systems that receive our data without dealing, with the recorded testing video.

6.1 Testing the system as a whole

Deal real rounds and check that the system follows them. You need a dealer, cards and chips, and access to the device panel.

1. **License.** In “System Configuration”, the “License” section shows “Valid” and the modules you ordered.
2. **Stream.** In the “Stream” section, “Running” names the table’s camera, e.g. “Basler Camera”.
3. **Camera view.** Open “Live View” and turn on “Camera” above the digital table. Check that:
 - the whole table is in the picture, including the far side and the dealer’s area;
 - the cards at the far side are sharp ([Section 5.1](#));
 - on a display-mounted camera, the table stays in the picture when you turn the display through its whole range.
4. **Detections.** Turn on “Detections”. Lay a card on every seat and in the dealer’s area: each card is marked with its rank and suit (e.g. 10s for the ten of spades), each chip stack with a box labelled s. The number after a label is how sure the model is.
5. **Rounds.** Deal at least 10 rounds that together use every seat and include a hit, a double, a split, a dealer bust and a blackjack, plus every side bet the table offers. During each round, “Live round” follows the table: the cards appear on the digital table and under “Players” with each hand’s total, and the phase at the top right moves from “Dealing” through “Players’ turn” and “Dealer’s turn” to “Payout”.
6. **Results.** Each finished round appears in the list on the left with the dealer’s total and how many hands won and lost. Click it and compare every hand’s cards and result (“Won”, “Lost”, “Push”) with what was dealt.
7. **Integrations.** If a system receives our data over MQTT, the “MQTT API” section in “System Configuration” shows “Connected to” the broker, and its count of published messages rises. Ask the integrator to confirm that:
 - the `heartbeat` topic says `"status": "online"` and its `camera_status` has no issue;
 - exactly one `round_summary` arrives per dealt round, with the results you saw in “Live View”;
 - their system supports the installed release. The “System version” in the “License” section is the commit that [Chapter 11](#) lists next to each release.

Live - current round

1 alert

Dealers turn - 0:57

Thu 15 Oct

13:03	BUST	3 won - 3 hands	2
13:02	BUST	2 won - 2 hands	3
13:01	18	1 won - 3 lost - 4 hands	3
13:01	BUST	2 won - 2 hands	2
13:00	20	4 lost - 4 hands	1
12:59	18	1 won - 1 lost - 2 hands	3
12:59	17	1 lost - 1 hand	
12:59	5	1 won - 1 hand	1
12:59	7	1 lost - 1 hand	1
12:59	BUST	1 won - 1 hand	
12:59	18	2 lost - 2 hands	1
12:58	20	1 won - 1 hand	
12:58	17	1 won - 2 lost - 3 hands	1
12:57	17	1 won - 1 hand	
12:57	17	1 won - 1 hand	2
12:57	19	2 won - 3 lost - 1 push - ...	2
12:56	21	3 lost - 1 push - 4 hands	1
12:55	BUST	3 won - 3 hands	2

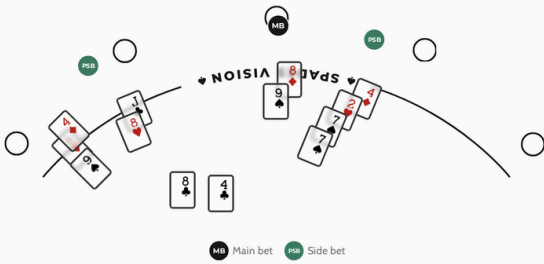
Live round

Round id 01a13f3b-2379-7f31-93e6-117119105f09

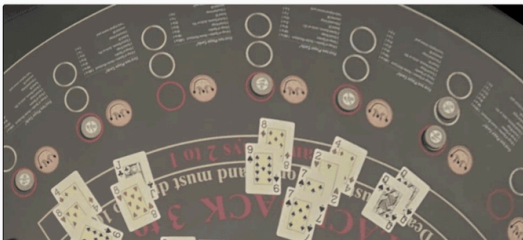
Table

Detections

Camera



Camera



Dealers turn - 0:57

Players

Dealer

8 4

12

Player 1

8 4 9

Strength 100%

21

Played Hit - Stand

Player 2

J 8

Strength 100%

18

Played Stand

Player 3

8 9

Strength 100%

17

Played Stand

Player 4

4 2 7 7

Strength 100%

20

Played Hit - Hit - Stand

Player 5

Live View during the dealer's turn: past rounds on the left, the digital table, the camera stream and every hand under "Players".

Players

Dealer

8 4 9

21

Player 1

8 4 9

Strength 100%

Push 21

Played Hit - Stand

Player 2

J 8

Strength 100%

Lost 18

Played Stand

Player 3

8 9

Strength 100%

Lost 17

Played Stand

Player 4

4 2 7 7

Strength 100%

Lost 20

Played Hit - Hit - Stand

A finished round opened from the list: each hand's cards, total and result.

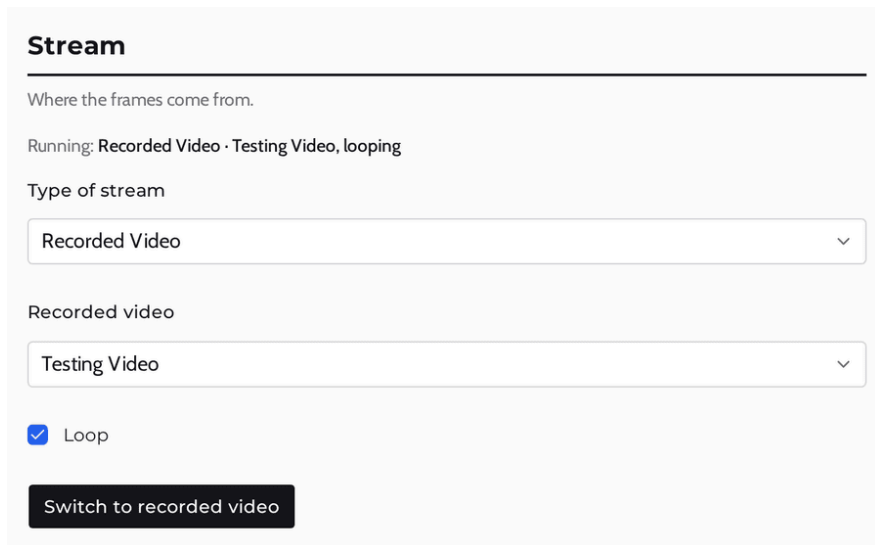
A round with alerts carries a yellow number in the list; open it to read them. An alert on a round that was dealt correctly, such as "Card where none was expected", usually means a seat's position in the table configuration is off: check the calibration. If rounds still come out wrong, report them as described in [Chapter 9](#).

6.2 Testing with a recorded video

Every device ships with a recorded video of Blackjack rounds: "Testing Video", about 13 minutes long. Playing it makes the device behave as if those rounds were dealt at its table, so you can test the systems that receive our data (MQTT clients, table displays) without a dealer.

The video only plays correctly with the table configuration it was recorded with. Spades Vision provides that configuration (a `.json` file and a `.jpg` layout) on request.

1. In “Table Configuration”, open the “Table configuration” list, choose “Add new...” and name it, e.g. “Testing video”. The table runs with the new, empty configuration at once; its own configuration stays in the list.
2. Under “Backup and restore”, click “Upload configuration” and choose the video’s `.json` file, then “Upload layout” and choose its `.jpg` file.
3. In “System Configuration”, under “Stream”, set “Type of stream” to “Recorded Video” and “Recorded video” to “Testing Video”. Leave “Loop” on so the video restarts when it ends. Click “Switch to recorded video”.
4. “Running” now names the video. Open “Live View”: the video’s rounds appear as if dealt at the table, and are published to MQTT like real rounds.



Stream

Where the frames come from.

Running: Recorded Video · Testing Video, looping

Type of stream

Recorded Video

Recorded video

Testing Video

☒ Loop

Switch to recorded video

When you are done, return the table to live play:

1. In “System Configuration”, set “Type of stream” back to “Basler Camera” and click “Switch to Basler camera”.
2. In “Table Configuration”, select the table’s own configuration in the “Table configuration” list. Remove the testing one with “Rename or delete”.

WARNING The video’s rounds are stored like real ones: they stay in the Live View history and appear in Analytics exports covering the time of the test.

7 Side bet configuration

Each player side bet is defined by one small JSON file: the winning hands and what each pays. The system reads the file and settles the bet from the cards on the table. This chapter covers everything you need to write one and to load it onto a device.

7.1 The file

A side bet is a table of rows. Each row is a winning combination and its payout, ordered from the highest-priority win down. This is the built-in Perfect Pairs, exactly as the device lets you download it:

```

{
  "name": "Perfect Pairs",
  "id": "PP",
  "rows": [
    {
      "outcome": "PERFECT_PAIR",
      "payout": 25.0,
      "conditions": [
        "same_rank(PAYER_UPCARDS)",
        "same_suit(PAYER_UPCARDS)"
      ]
    },
    {
      "outcome": "COLOUR_PAIR",
      "payout": 12.0,
      "conditions": [
        "same_rank(PAYER_UPCARDS)",
        "same_color(PAYER_UPCARDS)"
      ]
    },
    {
      "outcome": "MIX_PAIR",
      "payout": 5.0,
      "conditions": [
        "same_rank(PAYER_UPCARDS)"
      ]
    }
  ],
  "progressive": false
}

```

The top-level keys:

- **name**: the bet's display name.
- **id**: 1 to 5 letters and digits (PP, 213, S16). The table configuration and the APIs refer to the bet by this ID.
- **rows**: the winning rows, highest priority first.
- **progressive** (*optional, default false*): true marks a per-seat jackpot bet. Only the main better can place it, not back betters, so the reports count one opportunity per seated box for it instead of one per betting entity.

Each row has:

- **outcome**: your label for this win, conventionally in UPPER_SNAKE_CASE. The round data reports it as the side bet's result. Do not use LOSS or UNKNOWN: the system reports those for a lost bet and for one it does not score.
- **payout**: what the row pays, "to 1" (25.0 means 25:1). A jackpot row uses "GRAND", "MAJOR" or "MINOR" instead; the casino's progressive system pays that tier.

- **conditions:** the conditions that must *all* hold for the row to win.

WARNING Paytables printed “for 1”, common on progressives, include the stake. Enter one less: “25 for 1” is 24.0.

The three rows above pay a perfect pair (same rank and suit), then a colour pair (same rank and colour, different suit), then any other pair.

7.2 How a bet is decided

- A row wins when *every* one of its conditions holds.
- Rows are checked top to bottom and the first row that wins pays. In the example a suited pair meets all three rows; `PERFECT_PAIR` comes first, so it pays 25.
- You never write a losing row: when no row can win any more, the bet is lost.
- A bet settles as soon as the cards make the result certain, often before the round ends. A row that is still open holds back the rows below it: with two picture cards, Get The Picture (below) waits for the dealer’s second card before it pays either row.
- There is no “or” and no “not”. For “A or B”, write two rows with the same outcome and payout. A row only pays when no row above it wins, so the coloured pair row above needs no “not suited”.
- If the system later reads a card differently, it settles the bet again from the corrected cards.

7.3 Conditions

Most conditions look at a set of cards, the **scope**, and ask one question about it.

Scope	Cards it looks at
<code>PLAYER_UPCARDS</code>	the box’s first two cards
<code>PLAYER_FIRST_THREE</code>	the box’s first two cards plus the next card it draws, if it draws one
<code>DEALER_UPCARD</code>	the dealer’s first card
<code>DEALER_FIRST_TWO</code>	the dealer’s first two cards
<code>PLAYER_UPCARDS_AND_DEALER_UPCARD</code>	the box’s first two cards and the dealer’s first card (the 21+3 hand)
<code>PLAYER_FIRST_CARD_AND_DEALER_UPCARD</code>	the box’s first card and the dealer’s first card
<code>DEALER_FINAL_HAND</code>	every card the dealer draws

Write a scope in place of `SCOPE`:

Condition	True when
<code>same_rank(SCOPE)</code>	all the cards share one rank
<code>same_suit(SCOPE)</code>	all the cards share one suit
<code>same_color(SCOPE)</code>	all the cards share one colour (red: <code>H D</code> , black: <code>C S</code>)
<code>straight(SCOPE)</code>	the cards form a run of consecutive ranks, the ace high or low
<code>total(SCOPE, n)</code>	the blackjack total is exactly <code>n</code> ; <code>total(PLAYER_UPCARDS, 21)</code> is a blackjack
<code>total(SCOPE, OP n, ACE)</code>	the total compares to <code>n</code> ; <code>OP</code> is one of <code>= > >= < <=</code> , <code>ACE</code> is <code>BLACKJACK</code> or <code>PIP</code>
<code>busted(SCOPE)</code>	the blackjack total is over 21
<code>num_cards(SCOPE, OP n)</code>	the number of cards compares to <code>n</code>
<code>has_rank(SCOPE, n, {ranks})</code>	at least <code>n</code> of the cards have one of the ranks

Condition	True when
<code>has_suit(SCOPE, n, {suits})</code>	at least <code>n</code> of the cards have one of the suits
<code>has_card(SCOPE, n, {cards})</code>	at least <code>n</code> of the cards are one of the cards, written rank then suit (<code>AS, 10C</code>)
<code>first_hand({outcomes})</code>	the hand holding the box's first card ended one of these ways: <code>WIN BLACKJACK PUSH LOSS SURRENDERED</code>
<code>split()</code>	the box split
<code>unknown()</code>	never decided: a row with it settles as <code>UNKNOWN</code> and pays nothing

Ranks are `A K Q J 10 9 8 7 6 5 4 3 2`. Suits are `H` hearts, `D` diamonds, `C` clubs, `S` spades.

Details:

- `total` counts an ace as 11 unless that busts the hand, and a picture card as 10. The long form compares instead of matching exactly, e.g. `total(PLAYER_UPCARDS, >= 20, BLACKJACK)`, and with PIP counts an ace as 1: `Over 13 is total(PLAYER_UPCARDS, > 13, PIP)`.
- `same_rank`, `same_suit`, `same_color` and `straight` need at least two cards; on `DEALER_UPCARD` they are never true.
- `has_rank`, `has_suit` and `has_card` take a count and a set in braces. `has_rank(PLAYER_UPCARDS, 2, {J,Q,K})` is two picture cards; `has_suit(PLAYER_UPCARDS, 2, {S})` is two spades. The count is over the whole set: `has_card(PLAYER_UPCARDS, 2, {AS,KS})` is two cards that are each the ace or king of spades, so two aces of spades (possible in a multi-deck shoe) meet it. To require two *different* cards, write one condition per card, as Sweet 16 does below.
- `PLAYER_FIRST_THREE` may end with two cards: when the box stands on two, those two are the hand. A condition on it settles at the end of the box's turn at the latest. Blazing Seven uses it for its three-seven rows.
- `DEALER_FINAL_HAND` usually settles only when the dealer's turn ends.
- `first_hand` settles when the hand holding the box's first card is settled. `split()` is false once the box can no longer split.
- `unknown()` is for a bet you want counted but not scored: the built-in Unknown Progressive (P) is three `unknown()` rows.

7.4 Examples

These are built-in bets, each a different shape. Download one in the panel (see "Uploading and managing side bets") and edit it as a starting point.

21+3

Uses the three-card `PLAYER_UPCARDS_AND_DEALER_UPCARD` scope and settles like a poker hand. Every win pays 9, so the row order only decides which outcome is reported. A straight flush is both a straight and a flush, so it comes first.

```
{
  "name": "21+3",
  "id": "213",
  "rows": [
    {
      "outcome": "STRAIGHT_FLUSH",
      "payout": 9.0,
      "conditions": [
        "straight(PLAYER_UPCARDS_AND_DEALER_UPCARD)",
        "same_suit(PLAYER_UPCARDS_AND_DEALER_UPCARD)"
      ]
    }
  ]
}
```

JSON

```

    ]
  },
  {
    "outcome": "THREE_OF_A_KIND",
    "payout": 9.0,
    "conditions": [
      "same_rank(PAYER_UPCARDS_AND_DEALER_UPCARD)"
    ]
  },
  {
    "outcome": "STRAIGHT",
    "payout": 9.0,
    "conditions": [
      "straight(PAYER_UPCARDS_AND_DEALER_UPCARD)"
    ]
  },
  {
    "outcome": "FLUSH",
    "payout": 9.0,
    "conditions": [
      "same_suit(PAYER_UPCARDS_AND_DEALER_UPCARD)"
    ]
  }
],
"progressive": false
}

```

Sweet 16

A two-card total of exactly 16, graded. The top row needs the ace and the five of hearts, one `has_card` condition each. Below it a suited 16 beats a same-colour 16, which beats any 16. Every row requires `total(PAYER_UPCARDS, 16)`, so any other total loses.

```

{
  "name": "Sweet 16",
  "id": "S16",
  "rows": [
    {
      "outcome": "ACE_HEARTS_FIVE_HEARTS",
      "payout": 160.0,
      "conditions": [
        "total(PAYER_UPCARDS, 16)",
        "has_card(PAYER_UPCARDS, 1, {AH})",
        "has_card(PAYER_UPCARDS, 1, {5H})"
      ]
    },
    {
      "outcome": "SUITED_16",
      "payout": 16.0,
      "conditions": [
        "total(PAYER_UPCARDS, 16)",
        "same_suit(PAYER_UPCARDS)"
      ]
    },
    {
      "outcome": "SAME_COLOUR_16",
      "payout": 12.0,
      "conditions": [
        "total(PAYER_UPCARDS, 16)",
        "same_color(PAYER_UPCARDS)"
      ]
    }
  ]
}

```

JSON

```

    },
    {
      "outcome": "MIXED_COLOUR_16",
      "payout": 6.0,
      "conditions": [
        "total(PAYER_UPCARDS, 16)"
      ]
    }
  ],
  "progressive": false
}

```

Get The Picture

Combines player and dealer conditions in one row. Both rows need two picture cards in the box; the 300 row also needs the dealer's first two cards to be the same rank and suit. A qualifying box takes 300 when the dealer has such a pair and 12 otherwise.

```

{
  "name": "Get The Picture",
  "id": "GTP",
  "rows": [
    {
      "outcome": "TWO_PICTURES_DEALER_SUITED_PAIR",
      "payout": 300.0,
      "conditions": [
        "has_rank(PAYER_UPCARDS, 2, {K,Q,J})",
        "same_rank(DEALER_FIRST_TWO)",
        "same_suit(DEALER_FIRST_TWO)"
      ]
    },
    {
      "outcome": "TWO_PICTURES",
      "payout": 12.0,
      "conditions": [
        "has_rank(PAYER_UPCARDS, 2, {K,Q,J})"
      ]
    }
  ],
  "progressive": false
}

```

JSON

Buster

Pays when the dealer busts, more the more cards the bust took, and loses when the dealer does not bust. Every row needs `busted`, and the `num_cards` thresholds are `>=`, so the row order makes the tiers exclusive: an eight-card bust takes the top row, a seven-card bust the next, down to any bust (two cards cannot bust, so `>= 3` is the floor). The bet settles when the dealer's turn ends.

```

{
  "name": "Buster",
  "id": "BST",
  "rows": [
    {
      "outcome": "BUST_8_PLUS",
      "payout": 250.0,
      "conditions": [
        "busted(DEALER_FINAL_HAND)",

```

JSON

```

        "num_cards(DEALER_FINAL_HAND, >= 8)"
    ],
    {
        "outcome": "BUST_7",
        "payout": 50.0,
        "conditions": [
            "busted(DEALER_FINAL_HAND)",
            "num_cards(DEALER_FINAL_HAND, >= 7)"
        ]
    },
    {
        "outcome": "BUST_6",
        "payout": 15.0,
        "conditions": [
            "busted(DEALER_FINAL_HAND)",
            "num_cards(DEALER_FINAL_HAND, >= 6)"
        ]
    },
    {
        "outcome": "BUST_5",
        "payout": 4.0,
        "conditions": [
            "busted(DEALER_FINAL_HAND)",
            "num_cards(DEALER_FINAL_HAND, >= 5)"
        ]
    },
    {
        "outcome": "BUST_3_4",
        "payout": 2.0,
        "conditions": [
            "busted(DEALER_FINAL_HAND)",
            "num_cards(DEALER_FINAL_HAND, >= 3)"
        ]
    }
],
"progressive": false
}

```

Progressive Swisscasinos

A per-seat jackpot bet, so "progressive": true. It pays on a box blackjack. Read from the bottom up: any blackjack; a blackjack with a jack (a two-card 21 always holds an ace, so `has_rank(PAYER_UPCARDS, 1, {J})` makes it exactly ace and jack); that hand in one colour; that hand suited. The two top rows are the jackpot tiers: a suited ace and jack in the box while the dealer's first two cards are also an ace and jack pays "MINOR", and "MAJOR" when the dealer's are suited too. The amounts are the posted "for 1" amounts minus one.

```

{
  "name": "Progressive Swisscasinos",
  "id": "PSC",
  "rows": [
    {
      "outcome": "SUITED_BJ_JACK_PAIR",
      "payout": "MAJOR",
      "conditions": [
        "total(PAYER_UPCARDS, 21)",
        "has_rank(PAYER_UPCARDS, 1, {J})",
        "same_suit(PAYER_UPCARDS)",
        "total(DEALER_FIRST_TWO, 21)",
        "has_rank(DEALER_FIRST_TWO, 1, {J})",

```

JSON

```

        "same_suit(DEALER_FIRST_TWO)"
    ]
},
{
    "outcome": "SUITED_BJ_JACK_DEALER",
    "payout": "MINOR",
    "conditions": [
        "total(PLOYER_UPCARDS, 21)",
        "has_rank(PLOYER_UPCARDS, 1, {J})",
        "same_suit(PLOYER_UPCARDS)",
        "total(DEALER_FIRST_TWO, 21)",
        "has_rank(DEALER_FIRST_TWO, 1, {J})"
    ]
},
{
    "outcome": "SUITED_BJ_JACK",
    "payout": 69.0,
    "conditions": [
        "total(PLOYER_UPCARDS, 21)",
        "has_rank(PLOYER_UPCARDS, 1, {J})",
        "same_suit(PLOYER_UPCARDS)"
    ]
},
{
    "outcome": "COLORED_BJ_JACK",
    "payout": 49.0,
    "conditions": [
        "total(PLOYER_UPCARDS, 21)",
        "has_rank(PLOYER_UPCARDS, 1, {J})",
        "same_color(PLOYER_UPCARDS)"
    ]
},
{
    "outcome": "BJ_JACK",
    "payout": 19.0,
    "conditions": [
        "total(PLOYER_UPCARDS, 21)",
        "has_rank(PLOYER_UPCARDS, 1, {J})"
    ]
},
{
    "outcome": "BLACKJACK",
    "payout": 4.0,
    "conditions": [
        "total(PLOYER_UPCARDS, 21)"
    ]
}
],
"progressive": true
}

```

Twins

A bet on how the box plays, not only on its cards: the box's first card has the rank of the dealer's up card, and the bet pays by how the box's hand ends. This is the built-in Crazy Twins (CT) without its identical-fives bonus rows. A twin pays 22 when the hand holding the box's first card wins (a blackjack included), else 11 when the box split, else 11 on a push; otherwise the bet loses. The bet settles when that hand is settled, at the end of the round.

JSON

```
{
  "name": "Twins",
  "id": "TW",
  "rows": [
    {
      "outcome": "TWIN_WON",
      "payout": 22.0,
      "conditions": [
        "same_rank(PAYER_FIRST_CARD_AND_DEALER_UPCARD)",
        "first_hand({WIN, BLACKJACK})"
      ]
    },
    {
      "outcome": "TWIN_SPLIT",
      "payout": 11.0,
      "conditions": [
        "same_rank(PAYER_FIRST_CARD_AND_DEALER_UPCARD)",
        "split()"
      ]
    },
    {
      "outcome": "TWIN_PUSH",
      "payout": 11.0,
      "conditions": [
        "same_rank(PAYER_FIRST_CARD_AND_DEALER_UPCARD)",
        "first_hand({PUSH})"
      ]
    }
  ]
}
```

7.5 What a config cannot express

A config sees the box's first two or three cards, the dealer's cards, how the hand holding the box's first card ended, and whether the box split. A bet that depends on anything else needs us, for example:

- doubles, or how the hands after a split played out;
- the player's cards beyond the third;
- the amounts bet.

If you are not sure a bet fits, send us the payable. The layout side bets (Insurance, Bust, Swiss Casino Jack) are fixed and are not configured with these files.

7.6 Uploading and managing side bets

Side bets are managed in the device panel under **Table Configuration**, in the **Blackjack** section, on the **Player side bets** step. Open **Manage side bets** there. It lists every side bet the device can settle, with its name, ID, type and whether it is progressive.

▼ Manage side bets

Download a side bet as a template, upload a new or edited one, or remove a custom one.

⬆️ Upload side bet

Name	Id	Type	Progressive	Actions
Perfect Pairs	PP	Built-in		Download
Pure Pairs	PUP	Built-in		Download
Sweet 16	S16	Built-in		Download
21+3	213	Built-in		Download
Get The Picture	GTP	Built-in		Download
Kings Bounty	KB	Built-in		Download
Over 13	O13	Built-in		Download
Under 13	U13	Built-in		Download
Buster	BST	Built-in		Download
Progressive Swisscasinos	PSC	Built-in	✓	Download
Blazing Seven	B7	Built-in	✓	Download
Crazy Twins	CT	Built-in		Download
Unknown Progressive	P	Built-in	✓	Download

Manage side bets, listing the built-in side bets

- **Built-in:** ships with the device. You can download it, not remove it.
- **Custom:** a file you uploaded. You can download and remove it.

The side bet list in the same step, which you use to place side bet positions on the layout, shows the same bets: a bet must be in this list before you can place it.

To add or change a side bet:

1. Click **Download** on the bet closest to what you need. The browser saves it as `<name>.json`.
2. Edit the file in a text editor: `name`, `id`, `rows` and `payouts`.
3. Click **Upload side bet** and select the file.

The device checks the whole file, every condition included, before it stores anything. A malformed file, a missing or wrong key, an invalid ID or a condition that does not parse is rejected, and the panel shows the reason in a message. An accepted file takes effect immediately, with no separate save: tracking restarts with the new table, so a round in progress is not tracked to its end. Upload between rounds.

Custom side bets are stored in the active table configuration. They are part of its backup (**Download configuration and layout**), and another table configuration has its own.

NOTE An uploaded file with the ID of a built-in, for example **PP**, replaces that built-in and the list shows it as *Custom*. This is how you change a built-in's payouts: download it, edit the numbers, upload it under the same ID. **Remove** on that entry deletes your version and the built-in returns. Uploading over one of your own custom bets replaces it.

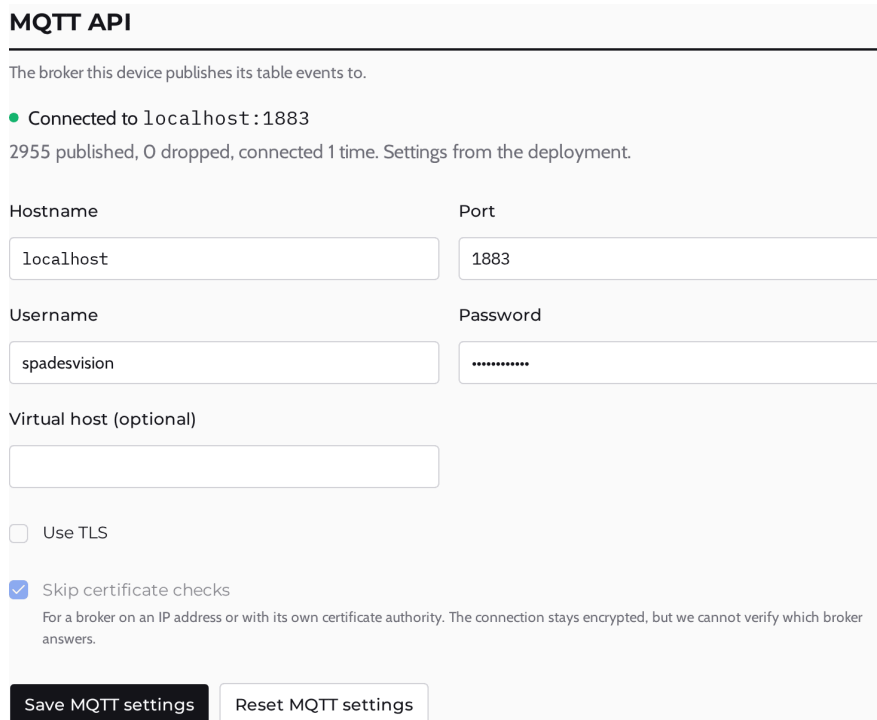
The ID is what placed positions refer to. Renaming a bet is safe; a new ID makes it a different bet. You can remove a bet that placed positions still use: the confirmation says how many. Those positions then show as not in the library, and their bets are counted but settle as `UNKNOWN` until you upload a bet with that ID again or place the positions anew.

8 API documentation

The device has two APIs: MQTT tells you what happens at the table, and the Configuration API (HTTP) changes how the device is set up. Tell us if you need data that neither offers.

8.1 MQTT API

The device publishes table events to an MQTT broker (MQTT 3.1.1). Set the broker in the device panel under **System Configuration → MQTT API**:



MQTT API

The broker this device publishes its table events to.

● Connected to localhost:1883
2955 published, 0 dropped, connected 1 time. Settings from the deployment.

Hostname	Port
<input type="text" value="localhost"/>	<input type="text" value="1883"/>
Username	Password
<input type="text" value="spadesvision"/>	<input type="password" value="*****"/>
Virtual host (optional)	
<input type="text"/>	
☐ Use TLS	
☒ Skip certificate checks <small>For a broker on an IP address or with its own certificate authority. The connection stays encrypted, but we cannot verify which broker answers.</small>	
<input type="button" value="Save MQTT settings"/> <input type="button" value="Reset MQTT settings"/>	

- **Hostname, Port, Username, Password:** the broker and the login the device publishes with.
- **Virtual host:** for a RabbitMQ broker only. The device then logs in as `{virtual host}:{username}`.
- **Use TLS:** connect encrypted (usually port 8883).
- **Skip certificate checks:** for a broker on an IP address or with a certificate from your own certificate authority. The connection stays encrypted, but the device cannot verify which broker answers.

Save MQTT settings reconnects with the new settings. **Reset MQTT settings** returns to the settings the device was installed with. The line at the top shows whether the device is connected, or why not, and how many messages it has published and dropped.

Topics

All topics start with `sv/{machine_id}/`. Everything a specific game defines sits under that game's name (`blackjack/...`). The topics at the root are game-agnostic, so an integration covering a mixed floor only needs `heartbeat` and `round_summary`. The sensor topics are at the root as well: the bet and cash trackers do not know which game is dealt.

Each topic's message schema is documented on our website. Its JSON schema is at the same address followed by `interface_schema.json`, for example https://docs.spadesvision.com/v3_2/mqtt_docs/heartbeat/interface_schema.json.

Game-agnostic

Topic	Documentation
sv/{machine_id}/heartbeat	Message schema
sv/{machine_id}/round_summary	Message schema
sv/{machine_id}/side_bet_sensor	Message schema
sv/{machine_id}/side_bet_sensor_with_denom	Message schema
sv/{machine_id}/cash_sensor	Message schema

Blackjack

Topic	Documentation
sv/{machine_id}/blackjack/summary	Message schema
sv/{machine_id}/blackjack/state_changed	Message schema
sv/{machine_id}/blackjack/round_over	Message schema
sv/{machine_id}/blackjack/side_bets	Message schema
sv/{machine_id}/blackjack/live_update	Message schema
sv/{machine_id}/blackjack/live_dealer_odds	Message schema
sv/{machine_id}/blackjack/card_positions	Message schema

Which topic to subscribe to

`round_summary` and `blackjack/summary` are sent exactly once per finished round, at the same time and with the same `round_id` and `sequence`. Both are already evaluated: main-bet win/loss, participation and timing are computed for you, so you do not have to implement the game rules.

- `round_summary` is the compact, game-agnostic summary, meant to be forwarded to a casino management system. The same shape describes a Blackjack, Baccarat or UTH round. Side bets appear as uptake: how many players could place each one and how many did.
- `blackjack/summary` adds the Blackjack detail: the dealer's hand, every player hand with its cards, value and result, and each side bet with its outcome and payout. `round_summary` is derived from it, so a Blackjack-only integration needs only this one.
- `blackjack/round_over` carries the raw table state (positions, tracking data, chip stacks) at the end of the round and leaves the game rules to you.

No field on either summary is currency: the system detects bet presence and relative size, not chip denominations. Every money field is a multiple of the bet it settles, with the initial bet as 1.0, and is *net* and *player-positive*: -1 a loss, 0 a push, +1 a win, +1.5 a blackjack paid 3:2, -2 a lost doubled hand.

Delivery

- **State:** `heartbeat` is QoS 1 and retained, so a new subscriber learns at once whether the device is up. It is sent every 10 seconds and immediately when a camera issue appears. When the device shuts down or loses the broker, the heartbeat changes to "status": "offline".
- **Round events:** `round_summary`, `blackjack/summary`, `blackjack/state_changed`, `blackjack/round_over`, `blackjack/side_bets` and `blackjack/live_dealer_odds` are QoS 1 and not retained: replaying the last one to a late subscriber would look like a new event.
- **Live samples:** `blackjack/live_update`, `blackjack/card_positions`, `side_bet_sensor`, `side_bet_sensor_with_denom` and `cash_sensor` are QoS 0 and not retained: the next sample replaces a lost

one within a second. `side_bet_sensor_with_denom` is only sent when a chip model reads denominations at this table.

While the device is disconnected from the broker, messages are dropped, not queued, so a reconnect never replays stale table state. Both summary topics carry a `sequence` counter: a gap in it tells you which rounds you missed.

Camera issues

The `camera_status` of every `heartbeat` says whether the device can see the table. Its `issue` is `null` while the camera is fine. Otherwise it names one issue: when several hold at once, the first of the list below.

The device checks the camera on every frame. A heartbeat goes out at once when an issue appears, changes to another kind or clears, and every 10 seconds in between. The issue's `since` is when it appeared; it starts over when the issue clears and comes back. Short issues are reported too, for example a camera that is gone for a second, so alert on `since` if you only care about issues that last.

Below, the *table position* is where the device believes the table is in the camera picture: the one it last found, or the calibrated one. The device searches for the layout between rounds, about every 5 seconds, while “Track the camera position automatically” is on.

1. **NO_FRAMES: the camera delivers no picture.**
 - *Sent:* within about two seconds of the last frame.
 - *Cleared:* with the first frame.
 - *Causes:* the camera is unplugged, not detected, or on a USB port too slow for it. See [Section 9.10](#).
2. **NOT_YET_CALIBRATED: there is nothing to check the camera against.**
 - *Sent:* while the table has no camera calibration or no player positions, and while the calibration is being done.
 - *Cleared:* when the calibration is saved.
3. **CAMERA_MOVED: the camera no longer sees what it has to.**
 - *Sent:* on the first frame on which a configured position (a player's initial card or main bet spot, or a bet sensor spot) lies outside the camera picture at the table position.
 - *Sent:* when two searches in a row find the layout at the same new place and configured positions lie outside the camera picture there: the camera was turned so far that only part of the table is in view. About 5 to 10 seconds after the move.
 - *Sent:* with “Allowed movement” set to “Small corrections only”, when two searches in a row find the layout somewhere a fixed camera cannot have moved to. This one stays until the layout is found in its place again.
 - *Cleared:* as soon as none of these holds.
4. **LAYOUT_NOT_VISIBLE: the device cannot find the table layout.**
 - *Sent:* with no cards on the table since the layout was last found, when two searches in a row miss it: 5 to 10 seconds after it disappeared. On an empty table the layout should be in plain view, so whatever hides it is reported: the camera pointing away, the dealer spreading the decks, an arm across the felt, a covered or dark table.
 - *Sent:* with cards on the table, once the layout has been missing for 60 seconds. Time with cards counts for less, because dealing hides the layout: with 6 cards half, with 12 cards a third.
 - *Not sent* for a layout found in its place but partly covered: that counts as found.
 - *Cleared:* by the next search that finds the layout.
 - *Never sent* while “Track the camera position automatically” is off: the device then works on the calibrated position.

A camera that may move (“Allowed movement”: “Any movement”) is followed: after the layout is found three times in a row at its new place, about 15 seconds between rounds, the device works on and saves the new table position. No issue is sent for the move itself, only for what the camera can no longer see.

8.2 Configuration API (HTTP)

Everything the device panel does goes through this API: table configurations, calibration, rules and side bets, the MQTT broker, network and time zone settings, rebooting. Your systems can use it too.

The API is at `https://{ip of the device}/api/v1/`. Requests and responses are JSON. The device's certificate is self-signed, so tell your HTTP client to accept it (curl: `-k`). A refused request answers with an error status and a JSON body `{"error": "..."}` that says why.

API reference

The API is described by an OpenAPI 3.1 document, which tools such as Postman, Insomnia, Swagger UI or a client generator can load:

- From the device: `https://{ip of the device}/api/v1/openapi.json`. It always matches the software the device runs.
- From our website, for each release: https://docs.spadesvision.com/v3_2/api_docs/openapi.json and a browsable version at https://docs.spadesvision.com/v3_2/api_docs/.

Authentication

If sign-in is enabled for your installation, people reach the device panel and the Configuration API after signing in with their login. Access for your own systems is then set up with Spades Vision on request.

Example: listing, selecting and renaming table configurations

A device stores one or more table configurations, for example one per layout a table is switched between. Exactly one is active: the table runs with its calibration, rules and side bets. The requests below are what the panel's list of table configurations does.

1. List the table configurations

```
curl -k https://{ip of the device}/api/v1/table-configs
```

SHELL

The answer:

```
{
  "active": "3f9a1c2e",
  "entries": [
    {
      "id": "3f9a1c2e",
      "name": "Table configuration",
      "created": "2026-09-01T10:15:00.120384Z",
      "updated": "2026-10-02T08:30:00.554107Z",
      "source": "migrated",
      "testcase": null
    },
    {
      "id": "b7d04e91",
      "name": "BJ 6 seats, 21+3",
      "created": "2026-10-05T14:02:11.905512Z",
      "updated": "2026-10-05T14:40:56.017239Z",
      "source": "created",
      "testcase": null
    }
  ]
}
```

JSON

`active` is the ID of the configuration the table runs with. `entries` lists every stored configuration, oldest first, with times in UTC. The other requests take the ID; the name is for people.

2. Select a table configuration

SHELL

```
curl -k -X PUT https://{ip of the device}/api/v1/table-configs/active \
-H 'Content-Type: application/json' -d '{"id": "b7d04e91"}'
```

The device switches at once: tracking restarts with that configuration's calibration, rules and side bets, so a round in progress is not tracked to its end. Switch between rounds. The answer is the list from step 1, now with `"active": "b7d04e91"`. Selecting the active configuration changes nothing. An unknown ID answers 404.

3. Rename a table configuration

SHELL

```
curl -k -X PUT https://{ip of the device}/api/v1/table-configs/b7d04e91 \
-H 'Content-Type: application/json' -d '{"name": "Table 12 - Blackjack"}'
```

Any configuration can be renamed, active or not; renaming does not change which one is active. A name has 1 to 80 characters. The answer is again the updated list. An unknown ID answers 404, an empty or too long name 400.

The same endpoints also create a new, empty configuration and activate it (POST `/api/v1/table-configs` with `{"name": "..."}), and delete one (DELETE /api/v1/table-configs/{id}; deleting the active configuration answers 409).`

From Python with requests

PYTHON

```
import requests

device = "https://169.254.1.123"

# The device's certificate is self-signed, see above: verify=False.
configs = requests.get(f"{device}/api/v1/table-configs", verify=False).json()
wanted = next(e["id"] for e in configs["entries"] if e["name"] == "BJ 6 seats, 21+3")
if configs["active"] != wanted:
    answer = requests.put(
        f"{device}/api/v1/table-configs/active", json={"id": wanted}, verify=False
    )
    answer.raise_for_status()
```

9 Troubleshooting guide

This chapter is for problems at a live table: first checks, collecting diagnostics for Spades Vision, giving remote access, and fixes by symptom.

NOTE This chapter grows with feedback from the field. Tell us what would make this guide, or the system, easier to work with.

9.1 First checks

1. Open the device panel. With a direct cable to the table PC it is at <https://169.254.1.123>. If it does not load, see [Section 9.7](#).
2. In “System Configuration” → “License”, check that the license is “Valid” and that “Current time” is correct. If not, see [Chapter 4](#).
3. In “Live View”, switch on “Detections”. Check that every player position is in the picture and that cards on the table get boxes with the right labels. If not, see [Section 9.10](#).
4. Deal a blackjack round. The cards appear under “Live round”, assigned to the right players and the dealer.
5. If a check fails and the fixes below do not help, download the diagnostics and send them to info@spadesvision.com with a description of the problem, the machine ID and system version (“System Configuration” → “License”), and when it happened.

9.2 Downloading the diagnostics

The device collects its diagnostics into a `.tar` archive a few minutes after it boots (at most about 20 minutes) and every 24 hours after that. Each archive covers the time since the previous one. A problem you saw a moment ago is only in the next archive: to get one now, reboot the device (“System Configuration” → “Reboot and reset” → “Reboot”) and download the archive that appears about 20 minutes later.

1. Open <https://<device address>/diagnostics/> in a browser, for example <https://169.254.1.123/diagnostics/> with a direct cable. It is only served over https. If your installation uses sign-in, sign in with an admin login.
2. The page lists the archives, named `diagnostics_<hostname>_<date>_<time>.tar`. Download the ones that cover the time of the problem; if unsure, the newest two.
3. Copy them to a USB flash drive if the PC has no internet, and send them to Spades Vision.

An archive contains:

Path	Content
<code>state/logs/</code>	The Spades Vision software’s logs
<code>state/database/</code>	The recorded rounds
<code>state/recordings/</code>	Every round with an alert since the previous archive (up to 150), and a small sample of ordinary rounds, each with its video and detections
<code>state/config/, state/table_configs/</code>	The table configurations and calibration pictures
<code>systemd_logs.txt</code>	The system log of the last 24 hours
<code>network.txt</code>	The network interfaces and their addresses
<code>disk_usage.txt</code>	How full the disk is
<code>revision</code>	The installed software version

Windows 11 opens a .tar file in File Explorer; on older Windows, run `tar -xf <file>` in a command prompt. The device keeps old archives until its disk is 85 % full, then deletes the oldest. A factory reset keeps them, and the license.

9.3 Reporting a single round

For a wrong result in one round, send its round ID; we look up the rest. In “Live View”, the list on the left holds the past rounds; the yellow number marks rounds with alerts. Click a round to see its ID, its alerts with their time in the round’s video, and the video itself. “Open the video in a new tab” lets you save the video.

Round Thu 15 Oct, 13:01 · 18
1 won · 3 lost · 4 hands

13:01:29 – 13:01:56
Round id 01a13f39-6869-7222-9117-c1a098f6359a

Card where none was expected	at 0:06 in the video
A player card is missing	at 0:08 in the video
Payout on a lost hand · Player 5	at 0:26 in the video

9.4 Diagnosing without sending a technician

If the casino’s own technician can help, ask them to do the first checks and download the diagnostics. We analyse the diagnostics within 24 hours. If that does not solve the problem, replace the device.

9.5 Replacing a device

Keep a spare device with the Spades Vision OS installed and no paid license.

1. Swap the spare in, and restore the table configuration on it (“[Calibrating several tables with the same layout](#)”).
2. Create a license for the spare’s machine ID on admin.spadesvision.com and upload it ([Chapter 4](#)).
3. Email Spades Vision to disable the paid license of the old device, so that it no longer counts towards your licenses.
4. Put the old device online in the office and describe the problem to us, so that we can log in and diagnose it. If it is not a hardware fault, reinstall the Spades Vision OS ([Chapter 3](#)) and use it as the next spare.

9.6 Remote access for Spades Vision

The device connects to Spades Vision’s VPN by itself as soon as it has internet through its Ethernet port; there is nothing to set up. It needs outbound TCP 443 and, for a faster direct path, UDP 41641 ([Chapter 2](#)).

Ways to give it internet:

- Ask the casino for an Ethernet cable from a network with internet access to the device.
- If we do not need the live camera picture, take the device to a router with internet access.

If neither is possible but the table PC can reach the internet, for example over the casino’s guest WiFi:

1. Install RustDesk from <https://rustdesk.com> on the table PC.
2. Send us the RustDesk ID and password.
3. We reach the device through the table PC.

9.7 The device panel is not reachable

Find the device's address. `169.254.1.123` only applies to a direct cable between the device and one PC; if several devices share that link, the others take another `169.254.x.x` address. On a network with DHCP the device gets its address from the DHCP server: ask the network administrator for its lease, or open

`https://<hostname>.local`.

Look at the device's HDMI output. Connect a monitor. Above the login prompt the device shows its hostname, hardware ID, current time and IP addresses. The addresses update when a cable is plugged in.

Scan the network for devices that answer on the panel's port, for example with nmap (replace the range with your network's):

```
nmap -p 443 --open 192.168.1.0/24
```

SHELL

If the device was just installed and the panel never appeared, the installation may have stopped with an error: see [Section 3.5](#). Otherwise power-cycle the device, wait 5 minutes and try again; then download the diagnostics.

9.8 The panel says “Not connected to the Spades Vision server”

The panel loads, but a yellow bar under the tabs says “Not connected to the Spades Vision server. Reconnecting...”: the Spades Vision server, which does the tracking and answers the panel, is not running.

- After a boot or a restart it takes a few minutes to load its models. Wait 5 minutes.
- It restarts by itself when it crashes or stops processing frames for 2 minutes. After 10 failed starts within 10 minutes it stays down until the device reboots.
- To reboot, use “System Configuration” → “Reboot and reset” → “Reboot”. If the server does not respond to that, power-cycle the device.

The diagnostics download works while the server is down. If the problem persists, send the diagnostics.

9.9 License issues

If the badge next to “License” in “System Configuration” does not read “Valid”, the panel shows the reason under “The license is not usable”. [Chapter 4](#) lists each reason with its fix. While the license is not valid the device detects and tracks nothing, so the Live View shows no cards. For a reason not listed there, send a screenshot with the diagnostics.

9.10 Detection and calibration issues

In “Detections” the table is rotated, or players on the left or right are missing. The calibration matched the layout the wrong way round, which can happen on very symmetric layouts. Repeat the camera calibration in “Table Configuration” and send us the diagnostics.

The Camera picture is black or frozen. Check the camera's USB cable. The device reconnects to the camera by itself within seconds after it is plugged in again.

The heartbeat reports a camera issue. “[Camera issues](#)” says when each issue is sent and when it clears.

`NO_FRAMES`: check the camera's USB cable. `CAMERA_MOVED`: point the camera back at the table, or calibrate again if it was moved on purpose. `LAYOUT_NOT_VISIBLE`: clear the felt and check that the camera points at the table and the table is lit.

No cards are detected at all. Check the license (see [Section 9.9](#)).

Cards are detected badly.

- Check whether you can read the cards in the Detections picture yourself. If they are blurred, refocus the lens ([Section 5.1](#)).
- Compare the cards with the supported decks in [Section 10.3](#).
- Lighting is rarely the cause: the camera adjusts its exposure and white balance continuously.

Send us the diagnostics in any case.

The Detections picture cuts off part of the table. Check the Camera picture. If the table does not fit, point the camera more towards the table or use a lens with a wider field of view (chapter “Supported environments and requirements”, “Basler camera lens and mount guidelines”).

Live View runs at one frame per second after a quiet period. After an hour without cards on the table the device idles at one frame per second. The first card, or opening the Live View, brings it back to full speed.

10 Supported environments and requirements

10.1 Table environment

The rule of thumb: if you can read the value and suit of every card on the table in the camera stream, the system can too, as long as the cards resemble the supported decks (see [Section 10.3](#)).

Camera, lens and mount

The system uses a 4K Basler dart camera, the daA3840-45uc (3840 × 2160). After calibration it reads out only the part of the sensor that covers the table, so the table does not have to fill the whole picture.

Lens: Choose a lens wide enough that the whole table, including every box, the dealer's cards and the layout side bet spots, stays in view over the full range the display holding the camera can be turned. The 4K sensor leaves enough resolution for that crop. A lens that is much wider than this wastes resolution: the cards on the far side of the table have to pass the **Sharpness** calibration step.

Mount: Mount the camera so that the long side of the sensor runs parallel to the float. Then:

- the display can be turned further before the table leaves the picture, and
- the part of the sensor the system reads out has fewer rows, which raises the frame rate and keeps the camera cooler.

Position: Working installations have the camera 0.55–1.1 m above the table, looking down at the table at 32–50°.

Lighting

The camera sets its exposure itself; you set its target brightness in the **Camera settings** calibration step. Light the table evenly and keep lamps from reflecting off the cards into the camera: glare on a card hides its index.

10.2 Game environment

Dealer procedure

The **Dealer procedure** rule tells the system where the dealer puts their cards. Directions are as seen from the dealer. The player hands are always staggered: each new card overlaps the previous one, offset to the left.

Procedure	How the dealer's cards are placed
Europe	The first card is dealt face up. Each further card goes to the right of the previous one, in a row.
UK	The first card is dealt face up. Further cards are staggered like the player hands.
US	The up card is dealt face up and the hole card face down. When the dealer turns the hole card over, it goes to the left of the up card. Further cards go to the right of the up card, in a row.

Dealing

The system follows the game from the cards on the table. Deal as follows:

1. Deal each box's first card on its first-card position (set in the **First cards** step of the blackjack configuration).
2. Keep the index of every card visible. Never put a card exactly on top of another.

3. Deal a double card turned sideways. The turn tells the system that the box doubled.
4. On a split, move the two cards apart so that they no longer overlap.
5. Take a surrendered hand, or a blackjack paid even money, off the table during the players' turn, before the dealer draws, and do not bring it back. The system reads a surrender only when the **Surrender** rule is Early or Late, and only for two-card hands up to 18.
6. Collect the cards at the end of the round.

Game rules

Set the rules of each table to its house rules in the panel: **Table Configuration** → **Blackjack** → **Blackjack rules**.

General

Dealer procedure

US

Number of decks

6

Surrender

Late

Blackjack pays

3 to 2

☐ Dealer peeks
 ☒ Dealer hits on soft 17

Doubles

Double allowed on

Any card

Double after split

Not on split aces

☒ Double on soft hands

Splits

Split allowed on

Any card

Maximum resplits

4

Split aces

Always

☐ Can hit split aces

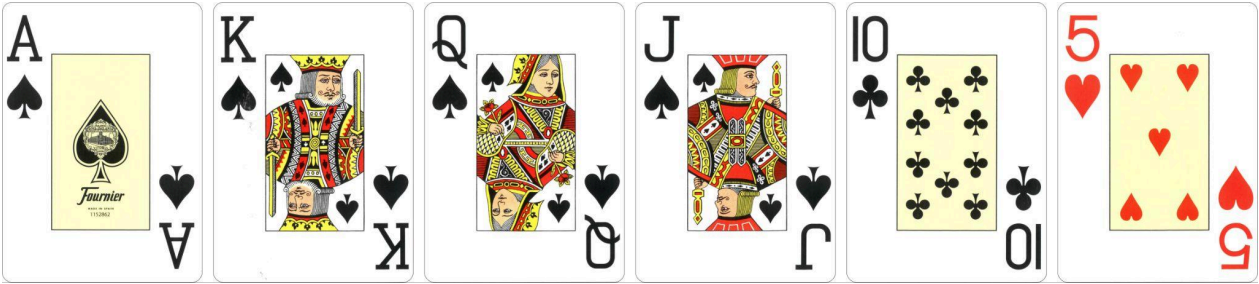
- **Dealer procedure:** Europe, UK or US, see above.
- **Surrender:** Never, Early or Late.
- **Blackjack pays:** 3 to 2, 6 to 5 or 1 to 1.
- **Double allowed on** and **Split allowed on:** any card, never, or all but some cards; for the last, tick the values that may not be doubled or split.
- **Maximum resplits:** the most hands a box can end up with, its first hand included. 4 allows up to 3 splits; 1 or 0 allows none.

10.3 Playing cards

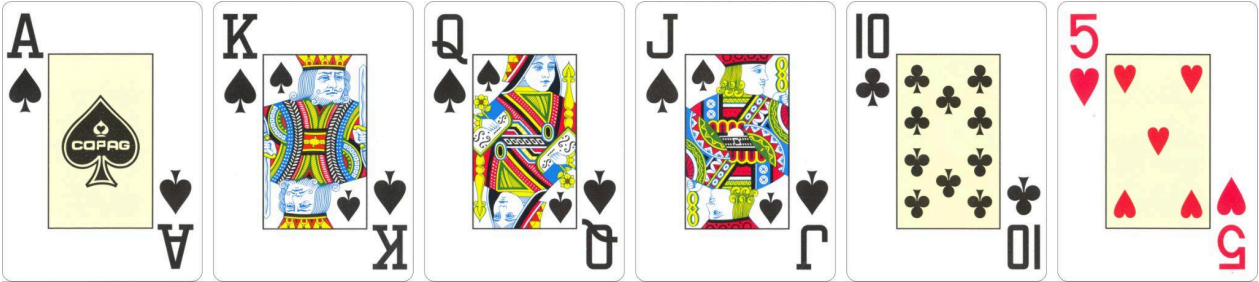
The models are trained on the decks below. They do not need the exact brand or design: decks of a similar style work too, e.g. most common Bee decks.

A deck whose design differs a lot from these may not be recognised reliably. Send Spades Vision scans of its cards so the models can be extended to it. Check the ace of spades in particular: many casinos print their own branding on it. Send us a picture of it before the installation.

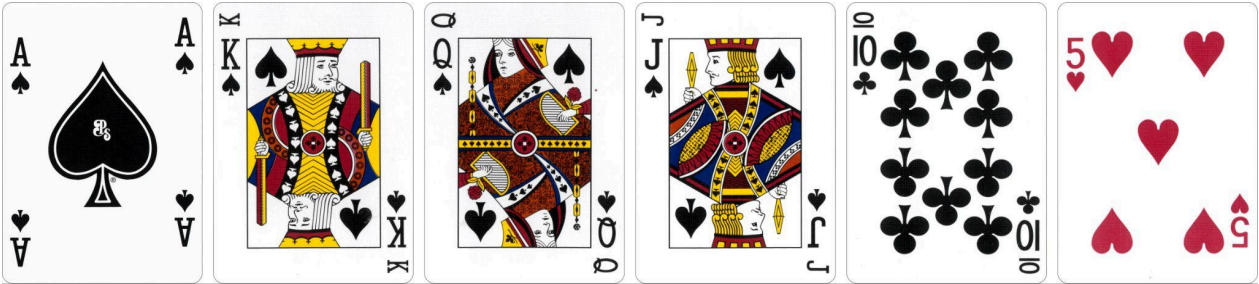
Fournier 818 — Jumbo Index



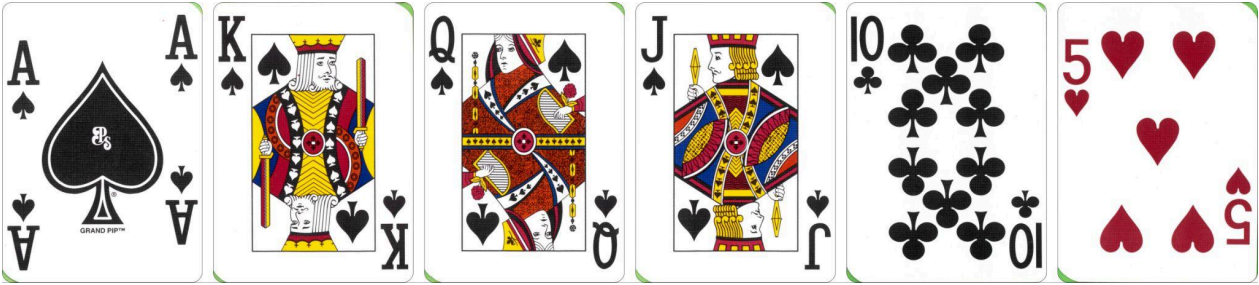
Copag — Jumbo Index



Paulson — Small Index



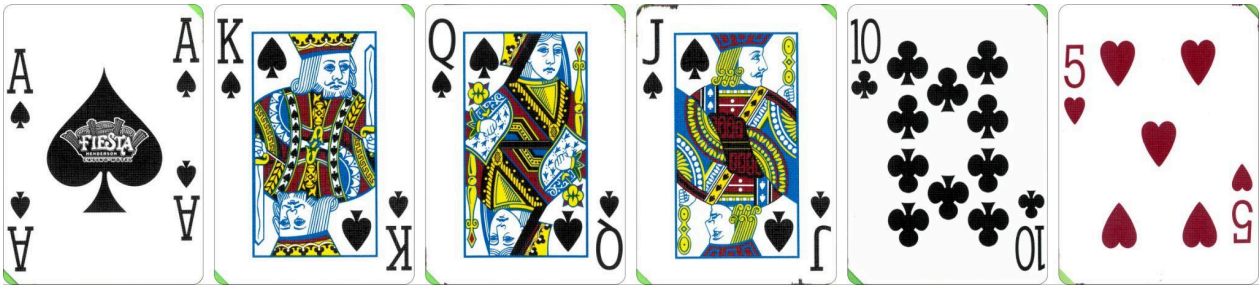
Paulson — Jumbo Index



Gemaco — Small Index



Cartamundi — Jumbo Index



11 Changelog

The newest release first. Each heading gives the release date; the title page names the commit this manual was built from.

Release v3.2.0 (2026-10-21)

- **Live View:** the history searches every round the device has kept, filtered by period (gaming day, week, month), alert kind, review status, bills and amount, and jumps to a given date and time. Past rounds, cash transactions and deck checks play one after the other (autoplay), at 0.5× to 4×, and admins can draw what the device detected over a past round's video.
- **Alert reviews:** each alert is reviewed as **Unclean dealer procedure**, **Wrong alert**, **Correct alert and handled** or **Ignore**, with a note. A reviewer can also mark an alert that should have fired and didn't, and an admin can correct a past round's cards by hand. The Excel report's Alerts sheet counts all of these, with each kind's precision; a cash-only Excel report lists the cash transactions.
- **Alerts:** each kind of alert is **On**, **Review only** (kept on the round to review, without a badge or count) or **Off**, set in the blackjack configuration.
- **Deck checks:** a deck spread out on the table is a deck check, not a round, so it raises no round alerts. The device counts the cards against whole decks and shows the missing and extra cards; a **DeckIncomplete** alert records it, review only by default.
- **Blackjack rules:** the cards are dealt from a shoe or a continuous shuffling machine (CSM). With a CSM the Live View shows no count.
- **Cash tracking:** bills are counted inside a table area you draw in the panel, like the other positions. Existing tables get the area used so far, closed at the top edge of the image.
- **License:** new modules **Cash tracking**, **Baccarat** and **Ultimate Texas Hold'em**. Cash tracking and the bet sensor run only with their module, and the panel offers their configuration only then; the Configuration API refuses to configure them when the license switches them off (403). Licenses issued before keep cash tracking.
- **Bet sensor:** a seat can have several progressives, each with a name and a denomination. The `side_bet_sensor` and `side_bet_sensor_with_denom` messages keep their fields and list one pass of seats per progressive (7 seats, 2 progressives: 0-6 and 7-13), repeating the main bet in each pass. A table with one progressive per seat sends exactly what it sent before.
- **MQTT API:** `LAYOUT_NOT_VISIBLE` is sent after two searches in a row miss the layout on a table without cards, and after 60 seconds or more with cards on the table, instead of after the first search that missed it. `CAMERA_MOVED` is also sent for a fixed camera found somewhere else, and for a camera turned so far that positions fall out of view. "Camera issues" under "MQTT API" lists when each issue is sent. `blackjack/card_positions` no longer jitters by a pixel or two.
- **Configuration API:** new `/table-config/alerts` and `/table-config/cash`, `/rounds/history`, a round's `alert-reviews`, `missed-alerts`, `corrections` and `detections`, and `/card-preps/{id}` and `/cash-transactions/{id}`. The bet sensor configuration lists each seat's progressives.
- **Tracking:** fewer false payout alerts on split boxes, on a bet drawn twice by the detector and on the next round's side bet put down early; a bet hidden by the dealer's arm during the deal keeps its first card. The automatic camera position no longer takes a poor fit for a good one.
- **Live video:** the browser receives it on the device's ports 50000-50099, over UDP or TCP, from any address. A viewer outside the devices' VLAN needs these and TCP 443 open to the devices.
- **Device panel:** a sign-in lasts 30 days.
- **Manual:** published per version at https://docs.spadesvision.com/v3_2/manual.pdf.

Release v3.1.0 (2026-10-15)

- **Installer: Force wipe & reinstall** now asks for confirmation on the installer page itself, which every browser shows, and a failed request is reported on the page.
- **Version:** devices are versioned as `v<release>-<commit>`. The device panel's "System version", the release file names, the device's HDMI output and its boot menu all show it.
- **Tracking:** a card dealt to a hand read as doubled undoes the double, as a doubled hand takes no further card; chips next to a 9, 10 or 11 are less readily read as a double.
- **Configuration API:** `GET /api/v1/rounds/{id}` returns one past round. `GET /api/v1/rounds/pages` no longer takes the `through` parameter.
- **Device panel:** on devices with sign-in, admin logins open every page; user logins open only the Live View, the analytics and the displays. The Configuration API and the diagnostics are for admins only.
- **Live View:** past rounds show their date and are grouped by gaming day. A link to a past round opens it however far back it is.
- **Device panel:** the live connection no longer drops in a background or minimised browser tab.
- **Diagnostics:** the device keeps a full day of rounds with issues (up to 150 per reason), so the daily diagnostics no longer miss any.
- **Reliability:** the device restarts the tracking software by itself when it stops responding.
- **Unchanged:** the MQTT API.

Release v3.0.0 (2026-10-08, 997561a)

- **Device panel:** redesigned, with a new Live View that shows the live table, the past rounds with their round video, and cash transactions.
- **Configuration API:** everything the panel does is also available to your systems over an HTTP API with an OpenAPI description, see "Configuration API (HTTP)".
- **MQTT API:** each hand reports the payout the dealer placed (`payout_stacks`). New alerts flag a payout placed on a lost or pushed hand or on a lost side bet, and side bet alerts carry the bet's name and result. The integrator login can no longer read or write the panel's internal topics.
- **Round videos:** recorded as H.264, so they play in every browser.
- **Cash tracking:** the bills put down on the table are grouped into cash transactions with their highest amount, and listed among the past rounds in the panel.

Release v2.13.1 (2026-10-07, d00c7db)

- **Installer ISO:** you can write the USB stick with balenaEtcher without the "missing partition table" warning, or with Rufus in DD Image mode. The device software is the same as v2.13.0.

Release v2.13.0 (2026-09-29, a144aa0)

- **Camera position:** the automatic camera position is found faster and more reliably on every device. The device panel has one camera position card with its status, the tracking switch, the allowed camera movement and a "Find now" button.
- **Calibration:** the steps run board, camera settings, sharpness, layout. A new camera sharpness check tells you when a betting position is out of focus.
- **Bet sensor:** chips count for a spot by how much of the stack lies on it, stacks follow when slid between spots, and bets are reported faster. Point main bets only accept chips right on the point, so draw them as polygons around the printed box.
- **MQTT API:** `side_bet_sensor` also publishes the moment a bet appears or leaves. Chip stacks in the blackjack topics carry a new optional `position` field with where the stack stands on the felt.
- **Cash tracking:** a new USD cash model, and bills are counted even while their denomination cannot be read yet.

Release v2.12.0 (2026-09-15, 57c980d)

- MQTT API: `blackjack/summary` lists every configured box in `players`, not only the boxes in play. Each entry has a `status` field ("ACTIVE" or "INACTIVE"), `players[i].box_index` is always `i`, and the new `configured_players` and `active_players` fields give the counts. The `schema_version` is 3. Subscribers reading `players` need to check `status`.

Release v2.11.0 (2026-09-15, 004e443)

- MQTT API: the `blackjack/analytics` topic is renamed to `blackjack/live_dealer_odds`, and the dealer odds are one documented field per dealer outcome (`v17` to `v21`, `bj`, `bust`, `dealer_unfinished`) instead of a list of pairs. Subscribers of `blackjack/analytics` need to update.
- MQTT API: new `side_bet_sensor_with_denom` topic, which carries chip counts per denomination once a chip denomination model is deployed. It publishes nothing until then.
- Cash tracking: the model detects a bill separately from reading its value, and the USD cash model is selectable in the device panel.

Release v2.10.1 (2026-09-14, 509fb2f)

- The tracking software no longer restarts over and over while the Basler camera is unavailable; the panel and the MQTT API stay up until the camera is back.

Release v2.10.0 (2026-09-08, 1980beb)

- Better recordings for the diagnostics, without slowing down the tracking.
- Device panel: you set the first cards with the mouse, like the rest of the blackjack configuration. The bet sensor configuration places its spots with the mouse too.
- A device can run only cash tracking or only the bet sensor.

Release v2.9.0 (2026-08-26, 9643ae0)

- MQTT API: new `cash_sensor` topic.
- Flashing script: fixes a potential issue and runs faster (under 4 minutes).
- Card tracking: improvements for the US procedure and more precise surrender detection.

Release v2.8.0 (2026-08-19, 48cc543)

- MQTT API: an updated blackjack summary with evaluated results: who won, lost or pushed, the hand values, the decisions and the strength of each player's decisions.
- MQTT API: a game-independent round summary, the API to expect from every card game module: round start and end times, side bet participation (opportunities and bets played) and the number of active boxes.
- Card tracking: recognises players surrendering or taking even money.

Release v2.7.0 (2026-08-10, c7c68be)

- Side bets are configurable, see "Side Bets".
- MQTT API: reports camera issues.
- Device panel: opening the live view wakes the device from idle. The HDMI screen shows all IPs and the machine ID.
- Device panel: shows the third-party software notice.
- USB installer: updates by default, except on new systems. Its website lets you force a clean wipe.
- Each device publishes the mDNS name `{prefix}-{machine id}.local` and gives it to the DHCP server as its hostname.
- Updated base system.

Release v2.6.2 (2026-07-07, 90fb309)

- Calibration: betting spots are assigned to the closest player automatically, instead of you selecting the player first. Space adds the position to the closest player; right-click removes the positions of the player closest to it. The calibration flow has detailed instructions.
- Excel analytics: much more detail on round speed, player hands per hour, side bet opportunities and utilisation.
- Improved card and chip stack tracking.

Release v2.6.1 (2026-06-25, 3fb00f8)

- The image used for detection is 768 pixels wide, which detects suits much better.
- Calibration: player side bets show their name in a tooltip and use several colours, to help you check the configuration.
- Improved chip stack recognition.

Release v2.6.0 (2026-05-11, 4412150)

- Bet sensor module, with its own MQTT API and configuration steps in the device panel.
- Networking: a static IP disables DHCP, instead of using both as before.
- MQTT API: uses MQTT 3.1.1.
- Optional login for the device panel (off by default).

Release v2.5.0 (2026-01-09, 4ff9f07)

- Networking: setting a static IP, DHCP and the factory default in the device panel is simpler and more robust.
- The HDMI screen shows the IP address.
- Time is shown in local time; you set the time zone in the device panel.
- Licensing supports the blackjack module structure.
- First release with the MQTT API.
- Round videos are served at `https://{ip of the device}/round_videos`, protected by the same credentials as `https://{ip of the device}/diagnostics`.
- System and security updates.

Release v2.4.3 (2025-11-14, 42ab96a)

- Device panel: you set a static IP there, and it shows the current IPs.
- In a DHCP network both the dynamic and the static IP are active.
- The device requests 192.168.182.29 from the DHCP server, so it gets that address in DHCP if it is the only device and the DHCP server supports it.
- Stability fixes.

Release v2.4.1 (2025-11-10, 5ebb287)

- Fixes and lessons from G2E 2025.
- Networking:
 - With DHCP, the device uses only the address the DHCP server gives it.
 - If DHCP fails, it falls back to 192.168.182.29.
- Improved robustness, diagnostics and reproducibility.
- Prepares chip stack and cash tracking.

Release v2.3.3 (2025-10-14, 08fbb44)

- Parameters fine-tuned at G2E.
- Back to 20 fps.

- Better quality of the video streams.

Release v2.3.2 (2025-09-26, a4d2588)

- Static IP 192.168.182.29; DHCP is still accepted.
- Diagnostics are written with hardware acceleration, at much higher quality.
- The device leaves idle instantly (before, it took several seconds).

Release v2.3.1 (2025-09-18, 1be653c)

- Much better calibration instructions in the device panel, with mockups.
- You can repeat the default calibration step by step, which helps in difficult conditions.
- Automatic calibration can be turned off.
- Reboot button.
- About 25% wider view by default (no manual zoom yet).
- Fixes a file handle leak.
- Player regions are masked by default, which improves the automatic calibration.
- Tuned for hardware longevity rather than frame rate.

Release v2.3.0 (2025-09-17, f1ff6e2)

- “Soft” factory reset without the USB stick.
- Better guarantee that the disk does not fill up.
- Diagnostics download at <https://169.254.1.123/diagnostics>, protected by a username and password we share separately. They hold all dynamic state, so we can reproduce and understand most issues:
 - table, layout and game configuration
 - videos of recognised issues, after the transformations
 - videos before the transformations, in 4K at a fifth of the frame rate
 - application and system logs
- Device panel usability improvements.
- Performance improvements and better hardware longevity.

Release v2.2.3 (2025-09-03, 75bef8b)

- More reliable automatic activation of development licenses.
- The device panel is reachable over https only, not http.
- Device panel: better readability, typo fixes.

Release v2.2.0 (2025-08-28, 2445053)

- Better performance: more fps for less CPU and GPU power.
- Includes chip tracking, turned off for now because it needs a model per casino.
- More structured and automated releases.

Release v2.1.0 (2025-08-04, e45c58f)

- Shows the system revision in the device panel.
- Minor update to the base system.
- The hardware clock is set on every time synchronisation, not only on a clean shutdown, so licensing works more reliably.

Release v2.0.0 (2025-07-30, e309452)

- New base system: better performance, security updates and future-proof upgrades. **It is incompatible with the previous bootloader: you need to flash the device.**
- Automatic activation of a development license when online.
- Fixes many edge cases in the card tracking.

Release v1.7.1 (2025-07-14)

- Renews the VPN auth keys of the ISO (they expire after 3 months). The ISO is otherwise identical to v1.7.0.

Release v1.7.0 (2025-04-14, cee3f8)

- Calibration, configuration, licensing (including trial licenses) and operation are fully functional.
- Automatic calibration uses the dealer area, which avoids mismatches (rotation) from repeating patterns at the player positions.
- Updated instructions: connect the device to the internet at least once after the installation, so the time synchronises and licensing works.
- Cards are not detected until a valid license is uploaded.

Release v1.6.0 (2025-03-18, 9c12ada)

- The static IP is 169.254.1.123.
- The device panel shows licensing errors.
- The streams also show on a local network with browser privacy protection turned on.

Release v1.5.0 (2025-03-14, 258ea3a)

- Improved tracking and detection.
- Game rules and dealer procedure style (US, UK and Europe) are configurable.
- Basler camera stream: a high-quality Basler camera feed, selected by default on boot.
- Automatic camera calibration supports the full range of the lens, which helps when you pair the Basler 4K sensor with a wide-angle lens, and handles rotations above 45°.
- Configuration and license backups: on the “Configuration” tab you download the license and the configuration file (TOML) to your PC, for example before updating with the USB installer.
- Calibration and configuration in the device panel are more intuitive. Some settings moved to the “Configuration” tab; the “Stream” tab is about calibration and choosing the stream.

Release v1.2.0 (2025-02-03, d02cc6a)

- Close to the version running at ICE 2025 in Barcelona.
- Remote access for updates and maintenance, when an internet connection is available.
- The device panel is reachable over http and https, for easier display integration.
- WebRTC stream to a separate PC works, at limited quality.