EZF · Integration handbook

Web Master · Modbus IP · API · tools
Edition 1.1 · 6 November 2026

Web Master — setting up the integration

Purpose of the integration

Web Master reads controller and luminaire status through the local HTTP API of the EZ2 adapter, schema ges-v2. This interface provides monitoring data. Starting tests is covered separately under Modbus IP.

Prepare the connection

1. Connect the controller and the integration computer to a network where they can communicate.

2. Note the controller IP address. Use a fixed address or a DHCP reservation.

3. Enable luminaire data sharing in the controller's local API settings. When disabled, the luminaire list and individual diagnostics return HTTP 403.

4. Open the controller address in a browser, then check /data/systeminfo.php and /lumlist.php.

5. Select EZ2 integration in Web Master and enter the controller address according to your Web Master configuration.

6. Import the list and compare the device count and locations with the controller. Add error descriptions from /data/errorcodes.php to the external dictionary.

Replace the example address 192.168.1.50 with your controller address. Keep these services on a trusted monitoring network. This local interface does not require a token; do not expose it directly to the Internet.

Data retrieved by Web Master

GET request	Format	Purpose
/data/systeminfo.php	JSON	System status, counters and data schema
/lumlist.php	TSV, 10 columns	Inventory and luminaire status
/data/errorcodes.php	JSON	Error dictionary
/data/luminaire.php?id=196618	JSON	Individual diagnostics, including NVG

For full examples and HTTP handling, see EZF API.

Identifying devices

The luminaire ID is channel × 65536 + physical address. Channels: 1 = BUS1, 2 = BUS2, 3 = AirBus. Physical address 10 therefore produces 65546, 131082 and 196618. Store the complete adresse value; do not truncate it to 16 bits.

When migrating from an older integration using physical addresses alone, synchronise the inventory again. Remove old records according to the Web Master procedure to avoid duplicates. Moving a luminaire to another channel changes its ID.

Luminaire list columns

Column	Meaning
adresse	Unique channel/address ID
geraetetyp	AW, PIKTO, KOMBI, NVG or UNKNOWN
gruppe	Group number stored in the controller
ueberwachung	1 = monitored; 0 = individually excluded from monitoring
schaltungsart	DS = maintained operation; BS = non-maintained operation; may be empty
betriebszeit	LTM_1H / LTM_2H / LTM_3H / LTM_8H; may be empty
letzte_testdauer	Empty when measured duration of the last test is unavailable
fehlercode	Four hexadecimal characters
ortstext	Location description
aktualisiert_am	Empty when the last update timestamp is unavailable

Do not replace empty fields with zero or the current date. Configured autonomy is not a measurement of the previous test duration. For NVG, the operating mode is reported only when both outputs agree. Mode and autonomy are empty after communication is lost.

Checks before acceptance

• Device counts and IDs match the controller.

• The same physical address on BUS1, BUS2 and AirBus creates three separate records.

• Types and locations are recognised; Fxxx codes have readable descriptions.

• Simulated LED, battery and communication failures are displayed correctly.

• Missing HTTP responses are shown as unavailable current data.

• Tests, inhibition and searches are distinguished from a healthy installation.

The local simulator helps prepare the integration. Acceptance requires a real controller and the intended Web Master. See programs and downloads.

Modbus IP — configuration and status monitoring

Connection parameters

The EZF controller is a Modbus TCP server; the BMS or tester is the client. This guide describes register map v1, available from firmware 10.38.

Parameter	Value
Host	Current controller IP address
Port	502
MBAP Protocol ID	0
Concurrent clients	Maximum 2
Read	FC03 or FC04, 1–125 registers
Write command	FC16, exactly 7 registers starting at 0x0100
Word	16 bits, big endian
uint32	Most significant word first
Addresses in this guide	Zero based, without the 4xxxx prefix

An idle connection may close after 3 seconds; the client must be able to reconnect. Modbus has no authentication. Restrict access to a trusted monitoring network.

Enable it on the controller

1. Open **Settings** → **External communication** and enable Modbus TCP. Appropriate access rights are required; exact menu wording depends on the display version.
2. When using the local web settings, confirm saving with the service PIN.
3. Wait for the scheduled restart and for the controller to return.
4. Enter its IP, port 502 and an appropriate Unit ID in the tester.
5. Read its identity: register 0 should be 0x455A, and register 1 should show map version 1.
6. Read the luminaire list and compare it with the controller. Do not enable remote tests for monitoring alone.

TCP, RTU and remote test permission are disabled by default. In luminaire test results, TCP also denotes the **duration/autonomy test**; distinguish this from the Modbus TCP network transport.

First reads

HEX address (decimal)	Words	Contents
0x0000 (0)	2	Signature and map version
0x0002 (2)	1	Controller status flags
0x0007 (7)	1	Remote tests permitted: 0/1
0x0008 (8)	2	Uptime in seconds
0x0010 (16)	2	Current boot identifier
0x0020 (32)	16	ASCII serial number, 2 characters per word
0x0100 (256)	8	Last job and its state
0x0200 (512)	16	AirBus group A; B=0x0210, C=0x0220, D=0x0230

Register 0x0002 flags: bit 0 fault, 1 communication error, 2 inhibition, 3 search, 4 initialisation/configuration, 5 FRAM memory error. Do not enter register 0 as 40001 unless your client explicitly converts 4xxxx notation to zero-based addresses.

Luminaire map

BUS1 starts at 0x1000 (200 slots), BUS2 at 0x3000 (200) and AirBus at 0x5000 (500). Each slot occupies 16 words. Slot i, counting from 1, starts at base + 16 × (i−1). A slot is a list position, not a physical luminaire address. All words in an empty slot are zero.

Slot offset	Contents
0 / 1 / 2	Channel / physical address / present: 0 or 1
3–7	Status bytes D1–D5
8 / 9	TF / duration test: 0 unknown, 1 running, 2 OK, 3 fault
10	Communication error 0/1
11 / 12	Luminaire type / group
13	NVG: LED1 in bits 0–1, LED2 in bits 8–9; other types 0xFFFF
14	Luminaire inhibition 0/1
15	Confirmed AirBus result: 0 OK; bits 0x20 battery/duration, 0x40 LED/current, 0x80 communication

A result of 0xFFFF means unknown, stale or wired luminaire — not a pass. When a FRAM error is reported, some status fields may be unconfirmed even if the list can still be read. Live reads do not replace archived test reports.

Remote AirBus test

Read first. Then allow remote tests locally on the controller and confirm write access, serial number and the individual command in the tester. This map controls AirBus luminaires and groups. BUS1/BUS2 provide monitoring; attempting to start a wired luminaire test returns exception 01.

FC16 writes exactly seven words starting at 0x0100:

Words	Meaning
0–1	Nonzero, increasing uint32 job ID

Words	Meaning
2	1 = TF, 2 = duration test, 3 = STOP
3	0 = individual AirBus luminaire, 1 = AirBus group
4–5	Luminaire UID or group number 0–3 (A–D)
6	Confirmation marker 0xA55A

Example: TF for AirBus 42, UID=196650, job 1: 0000 0001 0001 0000 0003 002A A55A.

UID = channel × 65536 + address. Check the current UID before sending. A common supervisory system should assign job IDs for both transports. Repeating an identical ID and command does not start a second test; changing the payload for that ID is an error.

After FC16 is accepted, read 0x0100–0x0107. State 1 = queued, 2 = accepted/running, 3 = results collected, 4 = rejected, 5 = STOP accepted. **State 3 does not mean every luminaire passed.** Check individual results and group counters. State 5 does not prove every luminaire received STOP. After a restart, read the boot identifier and current state; do not automatically replay an old command.

Errors and diagnostics

Exception	Meaning	Action
01	Unsupported function	Check FC and supported channel
02	Invalid address	Check zero-based address and range
03	Invalid data	Check all 7 words, ID, UID and marker
04	Device error / unavailable data	Check controller messages; do not display OK
06	Busy	Command was not accepted; retrying the same ID is allowed

For connection problems, check IP, port, TCP enablement, restart, firewall and the client limit. Save TX/RX telegrams and JSON/CSV exports from the tester.

Optional: Modbus RTU

RTU uses the same map: 19200 bit/s, 8E1, address 1–247, default 1. It requires a free BUS1 or BUS2 without stored luminaires. Do not mix Modbus RTU and the EZF luminaire protocol on the same port. Broadcast address 0 executes no commands.

See Windows/macOS testers and local exercises.

EZF API — JSON, TSV and error handling

Scope

This guide covers the local Web Master/EZ2 interface, schema **ges-v2**. All four GET requests are read only. Use Modbus IP to control tests. 192.168.1.50 is an example controller address.

System status in JSON

macOS / Linux:

```
curl --fail --max-time 10 http://192.168.1.50/data/systeminfo.php
```

Windows PowerShell:

```
Invoke-RestMethod -Uri "http://192.168.1.50/data/systeminfo.php" -TimeoutSec 10
```

Selected response fields for one luminaire without faults:


```
{
  "systemtype": "EZ2",
  "status_codes": ["0001"],
  "error_codes": ["0000"],
  "mapping_complete": true,
  "luminaire_count": 1,
  "communication_error_count": 0,
  "fault_count": 0,
  "error_schema": "ges-v2",
  "address_schema": "channel16-address16-decimal",
  "optional_unavailable": ["letzte_testdauer", "aktualisiert_am"]
}
```


This illustrates the format; actual values depend on the installation. The response also includes NVG counters, the error dictionary path and status scope. mapping_complete=true does not mean optional measurements or timestamps are available.

HEX flag	State
0001	Normal state / luminaire snapshot
0002	Global inhibition
0004	General fault
0008	Functional test
0010	Duration test
0100	Search
0200	Initialisation or configuration

Flags may occur together. Evaluate them alongside error_codes. An installation with no configured luminaires is faulty or unconfirmed.

TSV list

```
curl --fail --max-time 10 http://192.168.1.50/lumlist.php -o oprawy.tsv
```

Windows PowerShell:

```
Invoke-WebRequest -Uri "http://192.168.1.50/lumlist.php" -OutFile "oprawy.tsv" -TimeoutSec 10
$oprawy = Import-Csv -Path "oprawy.tsv" -Delimiter "`t"
$oprawy | Format-Table adresse, geraetetyp, fehlercode, ortstext
```

Example: tabs separate columns; preserve empty columns.

```
adresse geraetetyp  gruppe  ueberwachung   schaltungsart   betriebszeit   letzte_testdauer
fehlercode  ortstext   aktualisiert_am
196618  AW  0   1   DS  LTM_3H     0000   Corridor
```

The header and each record contain ten fields. Do not split on arbitrary whitespace, as this loses empty fields. The Web Master guide explains the columns. Do not equate ueberwachung=0 with communication loss.

Individual luminaire diagnostics

Use the ID from the first TSV column:

```
curl --fail --max-time 10 "http://192.168.1.50/data/luminaire.php?id=196618"
```

Example for a standard AirBus luminaire at physical address 10:

```
{"id":196618,"channel":3,"physical_address":10,"error_code":"0000","nvg":null}
```

For NVG, nvg contains led1, led2, mains, input_l1, input_l2, maintained1, maintained2 and communication_error. LED values: 0 off, 1 on, 2 blinking, 3 fault. With communication_error=true, the other NVG values are the last received snapshot, not confirmed live data.

Error dictionary

```
curl --fail --max-time 10 http://192.168.1.50/data/errorcodes.php
```

Keep codes as four-character hexadecimal strings, including leading zeros.

Code	Meaning
0000	No fault in the states read
0400	No communication with luminaire
F001	LED light source fault
F002	Battery missing or disconnected
F003	LED fault and missing battery
F004	Control gear overtemperature
F011 / F012 / F013	NVG: LED1 / LED2 / both outputs faulty
F100	General controller fault or no configured luminaires
FFFE	Unconfirmed state or incomplete diagnostic data

Fxxx codes belong to the EZF adapter. Add their descriptions to the integration system. A single luminaire code cannot describe every simultaneous fault; also use NVG details.

Responses and polling

HTTP	Meaning	Client action
200	Response available	Check format and status codes
400	Invalid ID or channel	Correct the parameter using the TSV list
403	Luminaire API disabled	Enable data sharing on the controller
404	Luminaire not found	Refresh the inventory
503	Data unavailable or response exceeds the limit	Display unavailable current data and check the controller

Use timeouts and limited retries with delays. Set the polling interval during commissioning. Do not run an unlimited request loop. Never display an error, empty response or cached old state as a current OK result.

In this release, the TSV limit is 24576 bytes. Exceeding it returns HTTP 503, not a truncated list with HTTP 200. Reads from separate endpoints are not a single atomic report.

Integration checks

Check successful reads, disabled luminaire API, invalid and nonexistent IDs, and connection loss. Compare fault interpretation with the controller display. Tools and simulators let you prepare the client without a connected device.

Integration tools — Windows, macOS and simulators

Downloads

Files are available directly from the Wiki. No GitHub account is required.

Material	Download
Modbus IP Tester 1.0.2 — Windows	Windows ZIP
Modbus IP Tester 1.0.2 — Mac Apple Silicon	arm64 ZIP
Modbus IP Tester 1.0.2 — Mac Intel	x86_64 ZIP
API — browser tester and Windows/macOS simulators	API client tools
Offline integration handbook — English	HTML · PDF
Data examples and register map	ZIP
Program checksums	SHA-256
Translated handbook checksums	SHA-256

On a Mac, check **Apple menu About This Mac**: an Apple M… chip means Apple Silicon; an Intel processor requires the Intel package. The tester does not install firmware.

Start the Modbus tester

1. Download and extract the complete ZIP.
2. Windows: run EZF-Modbus-Tester.exe from the extracted directory, keeping its companion files.
3. macOS: open EZF-Modbus-Tester.app from the correct package.
4. Select PL, DE or EN.
5. On the Mac, allow local network access when prompted.

Release 1.0.2 has no Developer ID signature or notarisation. If macOS blocks it, check approval for this particular application in Privacy & Security. Do not disable system-wide protections. BUILD.txt identifies the compiled architecture.

Practise without a controller

1. Start the built-in local simulator in the tester.
2. Set host 127.0.0.1, port 1502.
3. Read identity. The expected simulator serial number is EZFDEMO0001.
4. Read BUS1, BUS2 and AirBus lists; inspect UIDs, states and NVG.
5. Check TX/RX telegrams and JSON/CSV exports.
6. For command exercises, enable writes and confirm the simulator serial number.
7. Start TF, observe the job and results, and check STOP.

The simulator models communication and example responses. It does not test physical wiring, radio or battery capacity.

Connect to a real controller

Stop or disconnect the simulator and enter the controller IP and port 502. Start by reading the serial number and comparing the luminaire list. Remote tests require separate permission on the controller and confirmation of each command in the application. The tester does not automatically retry control commands.

API tester and REST simulator

Extract the API client package. It contains an HTML tester and ready-to-run simulators for Windows and both Mac architectures. The original README_KLIENT_PL.html is Polish; full English startup instructions follow: 1. Windows: run Windows/EZF_API_Simulator.exe. Mac: open the app in macOS-AppleSilicon or macOS-Intel. 2. Open http://127.0.0.1:8765/, select a scenario and serial number, and apply. 3. Select **API-Tester öffnen**, then **API prüfen**. The existing API tester and REST simulator have a German interface. 4. Check system status, TSV list, error dictionary and individual luminaire data. 5. Enter the real controller address as the API base and repeat the reads.

Alternatively, open Webmaster_API_Tester_DE.html. If the browser restricts local files, use the tester served over HTTP by the simulator or the curl/PowerShell examples in EZF API. Do not disable global browser protections. Closing the browser does not stop the simulator; use **Simulator beenden**. The programs are unsigned/not notarised. serial_number and simulator=true are simulator additions, not real controller fields.

Controller display training

The controller simulators include classic EZF 10.107, METRO 10.109 and the R6 comparison. They teach display operation and do not provide a Modbus or REST server.

Reporting a problem

Include controller and tester versions, operating system, event time, HTTP/Modbus message, and telegrams or an export. Remove passwords and access credentials. State whether the issue occurs in the simulator, on the real controller or in the intended Web Master.