MIGAL.CO Sensorbox

User Manual

Document type: User Manual
Product: MIGAL.CO Sensorbox – Welding process sensing & data connectivity
Document date: 27 Jan 2026
Version: 1.0

Manufacturer / Contact

MIGAL.CO GmbH
Website: https://www.migal.co
Wattstraße 2, 94405 Landau/Isar, Germany

info@migal.co, https://www.migal.co

Intended use

The Sensorbox is used to capture, pre-process and provide welding process data (e.g., current, voltage, power, gas flow, wire feed speed) and to transmit it via common interfaces (e.g., MQTT, OPC UA, UDP).
It is intended for **qualified welding/industrial personnel**.

Safety information (short)

- Work on electrical equipment may only be carried out by qualified personnel.
- Before installation or maintenance, disconnect power and secure against reconnection.
- Install measurement leads and sensors according to the specifications; incorrect wiring can cause measurement errors or device damage.
- Welding environments have high EMC exposure: observe cable routing and shielding.

Liability & copyright

This manual was prepared with the greatest possible care. Product changes (firmware/hardware) may lead to deviations.
© MIGAL.CO – All rights reserved. Reproduction or distribution only with written permission.

1. Introduction

This user manual describes the **Sensorbox (welding data measurement box)** for capturing, processing, and transmitting process data (e.g., current, voltage, wire feed speed, gas flow, status signals) in the environment of welding systems.

The **general part** applies regardless of the specific variant (hardware options, firmware version, sensor configuration) and forms the basis for installation, commissioning, and safe operation.

> **Note**
> This manual is intended for operators, maintenance, quality assurance, and developers. Basic knowledge of electrical engineering and networking is helpful for commissioning.

---

2. Document notes

2.1 Validity

This manual applies to:

- Product: **Sensorbox / welding data measurement box**
- Firmware: *(v1.x)*
- Hardware revision: *(Rev. A)*

2.2 Applicable documents

Depending on the configuration, the following may also be relevant:

- Installation instructions / wiring diagram of the respective sensors (e.g., wire sensor)
- Network concept / IT approvals (TLS, certificates, firewall)
- Service / calibration records

2.3 Symbols and conventions

Symbol/Format	Meaning
Bold	Controls, menus, important terms
`Code`	Files, parameters, commands, topics
> Note	Useful additional information
> Caution	Risk to device/process
> Warning	Risk to persons

2.4 Version history

Version	Date	Change
1.0	2026-01-27	Initial general part

3. Safety

3.1 Intended use

The Sensorbox is intended for:

- **Capturing measurement and status signals** in the welding process,
- **Time synchronization** and **pre-processing** (e.g., filtering, RMS, features),
- **Storage** (optional) and/or **transmission** to higher-level systems (e.g., InfluxDB, dashboard, quality evaluation).

The Sensorbox does **not** replace any safety functions of the welding system and is **not** part of a safety-related control system.

3.2 Improper use

The following is not permitted, among other things:

- Operation outside specified environmental conditions (temperature/humidity/EMC)
- Hardware modifications (rework, soldering, bypasses) not approved by the manufacturer
- Use as an isolation transformer, power supply, or safety-relevant shut-off device
- Connection to signals/voltages outside the specified input classes

3.3 General warnings

> **Warning – Electrical hazard**
> Work on electrical connections only by qualified personnel.
> Before assembly/disassembly: disconnect power and secure against reconnection.

> **Caution – Measurement connection on the welding process**
> Current/voltage signals may only be connected via the intended measurement adapters/tap points.
> Improper connection can result in personal injury and equipment damage.

> **Caution – EMC and measurement quality**
> Poor grounding, long unshielded cables, or routing parallel to power cables can cause measurement errors and interference.

> **Caution – Data integrity**
> Changes to network settings, time base, or calibration parameters affect data comparability and can distort quality conclusions.

> **Caution – ESD (electrostatic discharge)**
> When opening the housing, electronic components may be damaged by static discharge.
> Use ESD protection (wrist strap) where possible and avoid touching traces/ICs.

3.4 Operator responsibilities

The operator must ensure that:

- the Sensorbox is used only by trained persons,
- applicable accident prevention regulations are followed,
- all cables are mechanically protected and strain-relieved,
- network/IT security (access, certificates, passwords) is implemented according to company policy,
- regular visual inspection and—if intended—calibration checks are performed.

---

4. Product description

4.1 Functional overview

The Sensorbox is a data acquisition and communication module for welding environments:

- **Signal acquisition**
  e.g., current/voltage (analog), wire feed speed (encoder/sensor), gas flow, digital status signals
- **Timestamping & buffering**
  Local time base, ring buffer (depending on configuration)
- **Evaluation (optional)**
  e.g., RMS, peaks, spectrum (FFT), histograms, process fingerprint/anomaly detection
- **Data transmission**
  UDP real-time stream, MQTT (TLS), OPC UA / REST / WebSocket
- **Integration**
  Connection to InfluxDB/Grafana/React dashboards or customer-specific systems

4.2 System boundaries

The Sensorbox:

- provides data for **monitoring/analysis**, not for direct control (unless explicitly approved),
- is **not** a replacement for welding power source log data, but can complement it,
- measures only as well as the installed sensor technology and its setup.

---

5. Scope of delivery

- Sensorbox with microcontroller and connectors for Ethernet, power supply (9–12 V DC) and encoder, 12 V plug-in power supply as well as PoE adapter, 3-pin cable connector for voltage measurement with 5 m twisted measurement lead.
- Optional: wire sensor (encoder) with 1.5 m cable and 5-pin cable connector.
- Optional: gas sensor with 2 m cable and 7-pin cable connector.

> **Note**
> Please check the scope of delivery upon receipt and report transport damage immediately.

6. Nameplate and identification

On the device (or in the web/UI menu) you will typically find:

- Product name / model
- Serial number
- Hardware revision
- Firmware version
- Supply voltage / power consumption

7. Technical data

Parameter	Value
Interface	Ethernet (10/100 Mbit/s)
Data protocols	MQTT (3.1.1 and 5), UDP, OPC UA
Web interface	Configuration and real-time display
Supply voltage	9–12 V DC
Network	DHCP, static IP
Message rate	MQTT: 0.1–5 messages/s · UDP: 50 frames/s × 256 values
Encryption	MQTT: TLS 1.3 · UDP: binary, unencrypted
Dimensions (Sensorbox)	200 × 160 × 60 mm (L × W × H)
Weight (Sensorbox)	1.1 kg
Protection class	IP20

7.1 Measurement ranges & sensors

Parameter	Value
Voltage measurement	−60 to +60 V (AC/DC), ±1% of full scale
Current measurement	−600 to +600 A (AC/DC), ±1% of full scale
Wire diameter range	0.8–1.6 mm (larger on request)
Encoder resolution	0.114 mm, 600 pulses/revolution
Wire feed speed	0–20 m/min
Wire speed accuracy	0.7% at 10 m/min and 0.1 s sampling interval
Encoder weight	0.40 kg
Gas flow	0.5–50 l/min
Gas flow accuracy	±3% of maximum value
Gas flow sensor weight	0.10 kg

8. Environmental and operating conditions (general)

- Temperature: **0…40 °C**
- Relative humidity: **10…90%**, non-condensing
- Protection class: *IP20*
- EMC: compliance ensured by correct installation (shielding, grounding, cable routing)

> **Caution**
> Condensation, metallic dust, and strong vibration can cause failures and measurement errors.

9. Installation principles

Even if individual sensors (e.g., wire sensor) are described separately, the following basic rules apply system-wide:

9.1 Mechanics

- Mount the Sensorbox with low vibration (preferably in an electrical cabinet)
- Route cables with strain relief and appropriate bend radius
- Secure connectors against accidental loosening

9.2 Cable routing / EMC

- Route signal cables separately from power cables (distance, cross at 90°)

- Prefer shielded cables; terminate shield **both ends** or according to the concept
- Define ground/reference potential (star point or defined shield termination)
- For encoder/sensor signals, ensure clean reference and debouncing where needed

9.3 Earthing and equipotential bonding

- Ensure equipotential bonding in the cabinet
- For large potential differences: observe galvanic isolation / isolation concept

---

10. Commissioning – overview (checklist)

10.1 Before powering on

- ☐ Mechanical installation completed
- ☐ All sensor cables correctly connected and strain-relieved
- ☐ Supply voltage correct (polarity, voltage range)
- ☐ Network connection prepared
- ☐ Welding system and Sensorbox have compatible grounding/shielding concepts

10.2 First start

- ☐ Web/UI reachable
- ☐ Time base correct (RTC/SNTP)
- ☐ Data arrives in the target system (e.g., MQTT/UDP)

10.3 Functional check

- ☐ Plausibility check of measurements (idle/welding operation)
- ☐ Wire sensor: wire movement → wire feed speed changes plausibly
- ☐ Current/voltage: stable values, no unrealistic peaks due to interference
- ☐ Test event/job data (if available) is written correctly

---

11. Maintenance, care and service (general)

11.1 Regular visual inspection

- Check housing/connectors/cables for damage
- Tighten/secure loose connections
- Remove contamination (dry, antistatic; no aggressive cleaners)

11.2 Firmware/configuration maintenance

- Update firmware only from approved sources
- After updates: perform a quick measurement plausibility check

11.3 Calibration (principle)

- Sensors that are prone to drift (e.g., analog measurement chains) require regular calibration checks depending on requirements
- Document calibration status (date, reference, deviation, result)

---

12. Disposal

Dispose of electronic components according to local regulations (WEEE).
Dispose of batteries/accumulators (if present) separately.

---

13. Liability and warranty (short text)

- Warranty is void in case of improper use, unauthorized modifications, or operation outside the specification.
- The manufacturer is not liable for data loss due to misconfiguration, missing backups, or IT security violations unless required by law.

---

14. Commissioning (Sensorbox)

Preparation

Ensure the following components are available and operational:

- **Sensorbox**
- **Power supply**
  - either: **9–12 V DC** via power supply
  - or: **PoE** via PoE adapter/switch
- **Ethernet network** with enabled **DHCP server** (router/switch)
- **PC or mobile device** in the same network (for the web interface)
- *(optional, depending on configuration)* connected sensors:
  - wire sensor / encoder
  - current/voltage tap (measurement cable, clamps)
  - gas flow sensor

> **Caution**
> Before connecting current/voltage signals, make sure the signal levels and connection concept match the Sensorbox (measurement adapter, isolation, reference potential).

Connecting the Sensorbox

No.	Connector
1	Wire sensor (5-pin)
2	Gas sensor (7-pin)
3	DC input
4	Ethernet network

No.	Connector
1	Current socket
2	Current socket
3	Welding voltage (3-pin)

Connecting current and voltage signals

Insert the Sensorbox into the welding current circuit via current sockets 1 and 2. Connect the two-core cable from socket 3: **red** to the positive pole and **blue** to the negative pole. The measurement points should be as close to the arc as possible. Use twisted pair wiring and keep the untwisted end as short as possible.

Connecting the sensors

Connect the wire sensor to socket 1 and the gas sensor to socket 2.

Establishing network connection

1. Connect the **Ethernet port** of the Sensorbox to a free port on your **router/switch** or to a **PoE adapter**.
2. Provide power (depending on variant):
    - **Variant A:** Connect power supply to the **DC input (9–12 V DC)**
      or
    - **Variant B:** Provide **PoE** via the Ethernet cable *(IEEE 802.3af, if supported)*

> **Note**
> If PoE is used: ensure a PoE-capable port/injector is actually used (not every switch provides PoE).

Determining the IP address of the Sensorbox

After power-up, the Sensorbox automatically obtains an **IP address via DHCP**. Find it as follows:

Option A: DHCP device list (router/DHCP server)

1. Log in to the UI of your **router** or **DHCP server**.
2. Search the device list for a new entry, typically by:
    - **Hostname** (`mip`)
3. Note the assigned **IP address**.

Option B: Network scan

Use a tool such as:

- **Advanced IP Scanner** (Windows),
- **nmap** (Windows/macOS/Linux),
- or a network scanner app on your smartphone,

to display all active devices on the network. The Sensorbox should appear as a new device.

Accessing the web interface

Enter the determined IP address in a web browser, e.g.:

- `http://192.168.0.42`

Depending on the network configuration, a hostname may also work, e.g.:

- `http://mip`

The Sensorbox web interface will be displayed.

Web interface

After opening the IP address, the **login screen** appears.

Login (default)

Default login:

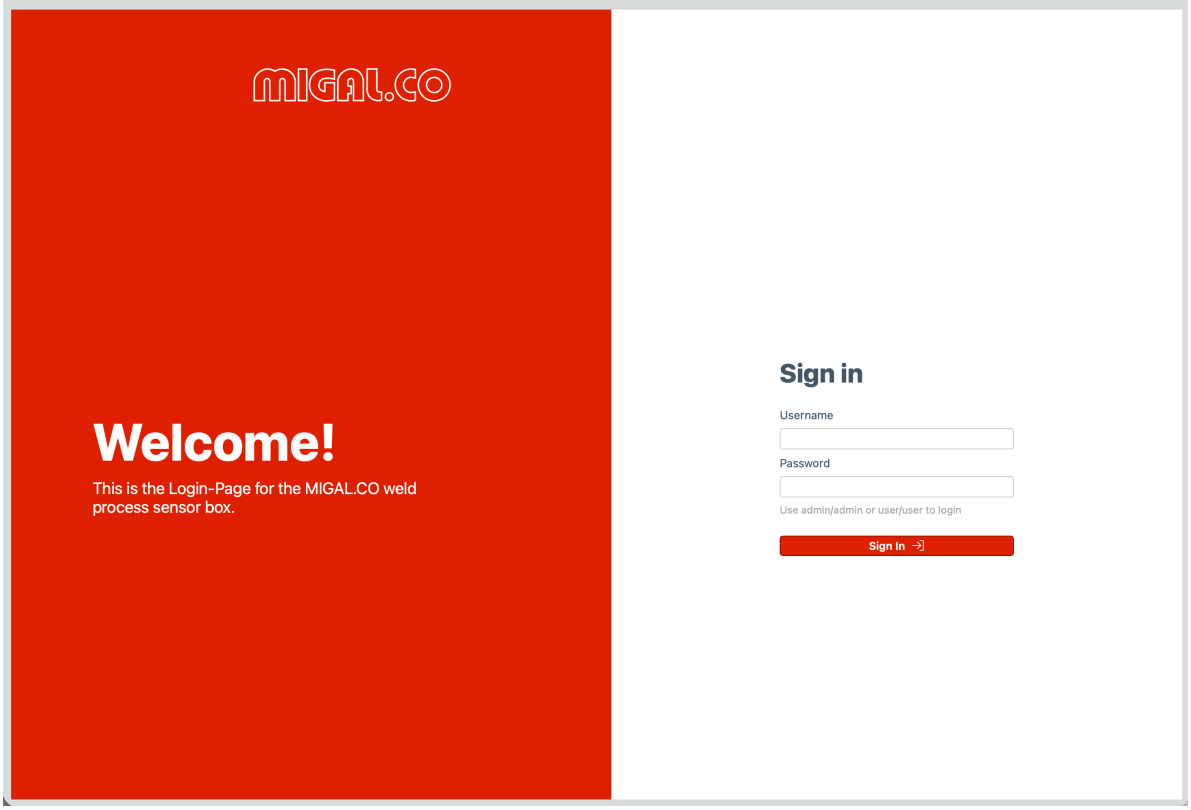
- **Username:** `admin`
- **Password:** `admin`

> **Important**
> For security reasons, change the default password after first commissioning (see “Users & Security”).

15. Web interface (Dashboard & settings)

The Sensorbox has an integrated web interface (Mongoose web server) to manage **live values**, **status**, **network/MQTT/UDP settings**, **certificates**, **calibration**, and **firmware updates**.

Login page



On the login page, sign in with username and password.

- Fields: **Username**, **Password**
- Button: **Sign In**

Navigation & header bar

Left sidebar (menu)

In the left sidebar you will find the main areas:

- **Dashboard** – live view and status
- **Device Settings** – device configuration (network, MQTT, UDP, certificates, NTP, calibration, sensor parameters)

- **Firmware Update** – upload firmware and reboot
- **Admin Password** – change passwords (admin/user)

Header bar (top)

- **Status indicator** (green dot): web UI reachable and session active.
- **Device/page label**: e.g., *Weld process sensor*
- **Logged in user**: e.g., `admin`
- **Logout** (exit icon): ends the session.
- **Hamburger menu** (top left): collapses/expands navigation (depending on screen width).

---

16. Dashboard (live overview)

The dashboard provides a quick assessment of **measurements** and **system status**.

Status panel

In the **Status panel**, key system states are shown as text + indicator dot:

- **OpenAMP OK** – status of internal communication (e.g., between CPUs/tasks)
- **ADC Overrun** – indicates ADC overruns (too much load / timing issues)
- **Encoder Timeout** – wire/encoder signal not available or timeout detected
- **NTP Synced** – time synchronized via NTP/SNTP (important for correct timestamps)
- **Calibration active** – calibration is active (gain/offset applied)

> **Practical tip**
> If measurements look odd: check here first whether **ADC Overrun** or **Encoder Timeout** is present.

Live displays (gauges)

Live values are shown as gauges:

- **Voltage (RMS)** – RMS voltage
- **Amperage (RMS)** – RMS current
- **Power (Active)** – active power
- **Wire feed speed (m/min)** – wire feed speed
- **Gas flow (l/min)** – gas flow

> **Note**
> Scale ranges are fixed in the dashboard and intended for quick orientation.

---

17. Device Settings

Under **Device Settings** you configure communication and conversion/calibration.

MQTT settings

Configure MQTT connectivity here (optional TLS).

- **Enable MQTT**: publish MQTT on/off
- **Server URL**: broker address incl. scheme, e.g. `mqtts://<ip>:8883`
- **MQTT topic**: base topic, e.g. `sensorbox`
- **MQTT Username / MQTT Password**: credentials
- **TLS On/Off**: enable/disable TLS encryption
- **Hostname**: device name (used for identification in some setups)
- **MQTT Version**: selectable (e.g., 5)
- **Client Id**: unique broker client ID, e.g. `sensorbox_001`
- **QoS**: quality of service (e.g., 0)
- **Message frequency**: send interval / message rate

Save changes with **save**.

> **Important**
> For **MQTTS** (TLS enabled), certificates must be correctly uploaded under **Certificates**.

Network settings

- **Enable DHCP**: DHCP on/off
  - **on**: IP obtained automatically
  - **off**: static IP configuration
- **IP address**
- **Gateway address**
- **Network mask**

- **DNS address**

Save via **save**.

> **Note**
> When switching from DHCP to static, ensure IP/gateway/netmask match your network—otherwise the box may become unreachable.

UDP settings

UDP is typically used for **real-time streaming** (e.g., scope/oscilloscope).

- **Enable UDP**: UDP stream on/off
- **Host address**: target IP (receiver, e.g., PC/server)
- **Host port**: target port (e.g., `5005`)
- **UDP mode**: e.g., `binary`
- **Include status**: include status data in the stream
- **Include calibration**: include calibration information in the stream

Save via **save**.

OPC UA settings

OPC UA (Open Platform Communications Unified Architecture) is an industrial standard to integrate process data **in a structured, vendor-neutral way** into SCADA/MES/IIoT systems.

The Sensorbox provides an integrated **OPC UA server** (transport: `opc.tcp`).
Typical clients include **UaExpert**, Node-RED (`node-red-contrib-opcua`), or custom OPC UA clients.

> **Note (scope of use)**
> OPC UA is ideal for **KPIs/derived values** (e.g., RMS, power, gas, wire).
> For **high-frequency raw data** (scope/FFT), continue to use the **UDP stream**.

Settings in the web interface

(Device Settings → OPC UA Settings)

- **Enable OPC UA** – enables/disables the OPC UA server.
- **Port** – TCP port for client connections (default: **4840**), range `1…65535`.
- **Poll interval (ms)** – internal update/server poll interval (e.g., **50 ms**), typical `10…5000 ms`.
  - smaller value → faster updates, higher CPU load
  - larger value → lower load, more sluggish values
- **Endpoint URL** *(read-only)* – automatically built from IP and port, e.g. `opc.tcp://192.168.0.100:4840`

Save via **save**.

> **Important**
> Changes to **Port** or **Enable OPC UA** may trigger an OPC UA server restart.
> If a client is connected, there may be a brief interruption—then reconnect.

Exposed variables (read-only)

Variable	NodeID	Unit	Description
Current	`ns=1;i=6001`	A	RMS current
Voltage	`ns=1;i=6002`	V	RMS voltage
Power	`ns=1;i=6003`	W	Active power (averaged)
Gas flow	`ns=1;i=6004`	l/min	(corrected) volumetric flow
Wire feed speed	`ns=1;i=6005`	m/min	Encoder-based

> **Note**
> Many clients connect by browsing the address space.
> Alternatively, variables can be read directly using the NodeIDs above.

Client connection (example)

1. **Enable OPC UA**: Device Settings → OPC UA Settings → **Enable OPC UA**
2. Read the **Endpoint URL** (e.g., `opc.tcp://192.168.0.100:4840`)
3. Connect in the client:
    - Security: **None** (no encryption/signing)
    - Authentication: **none** (default)

> ⚠️ **Security note**
> With Security **None**, data is unencrypted.
> In production environments, run OPC UA in a **protected network/VLAN** or apply appropriate network security measures.

Troubleshooting (short)

- **Client cannot connect**
    - Correct IP/port? (check Endpoint URL)
    - Firewall/router blocking TCP port **4840** (or configured port)?
    - OPC UA enabled? (**Enable OPC UA**)
    - Same subnet / routing correct?
- **Values do not update**
    - Poll interval set too large?
    - Check dashboard status (e.g., Encoder Timeout, ADC Overrun)
    - Reconnect client after changes (port/enable)

Certificates

Upload TLS files here (e.g., MQTT via `mqtts://`).

- **CA Certificate** (`ca.crt`)

- **CRT Certificate** (client cert, e.g., `client.crt`)
- **Private key** (private key, e.g., `client.key`)

Functions:

- **upload**: upload certificate/key
- **delete** (trash icon): delete file

> **Important**
> Handle private keys securely and upload only over trusted networks/clients.

NTP settings

- **SNTP Server**: time synchronization server address (e.g., `udp://time.google...`)

Save via **save**.

> **Why is NTP important?**
> Correct timestamps are essential for logging, job assignment, and analytics.

Calibration

Set conversion and offset values:

- **Voltage gain / Voltage Offset**
- **Current gain / Current Offset**
- **Gasflow Gain / Gasflow Offset**
- **Encoder Wheel Circumference (mm)**: circumference of the encoder/measuring wheel

Save via **save**.

> **Practical tip**
> Change **Encoder Wheel Circumference** only if the wheel/mechanical setup is different—this directly affects m/min.

Gas flow sensor settings

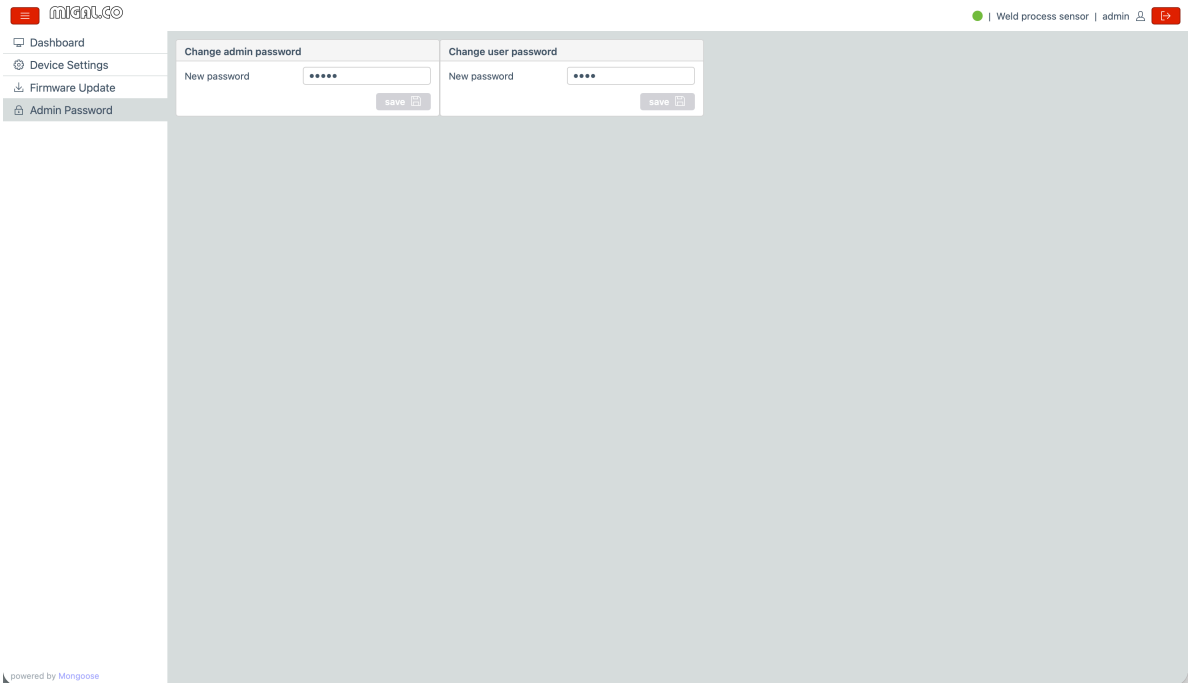
Shielding gas composition (for correcting the flow sensor):

- **Ar**, **He**, **CO2**, **O2**, **N2** (fractions)
- **Correction factor**: calculated correction factor

Save via **save**.

---

18. Admin Password (change passwords)



On this page you can change passwords separately:

- **Change admin password**
  - Field: **New password**
  - Button: **save**
- **Change user password**
  - Field: **New password**
  - Button: **save**

> **Important**
> After commissioning, change the default passwords and document them securely.

User roles (brief)

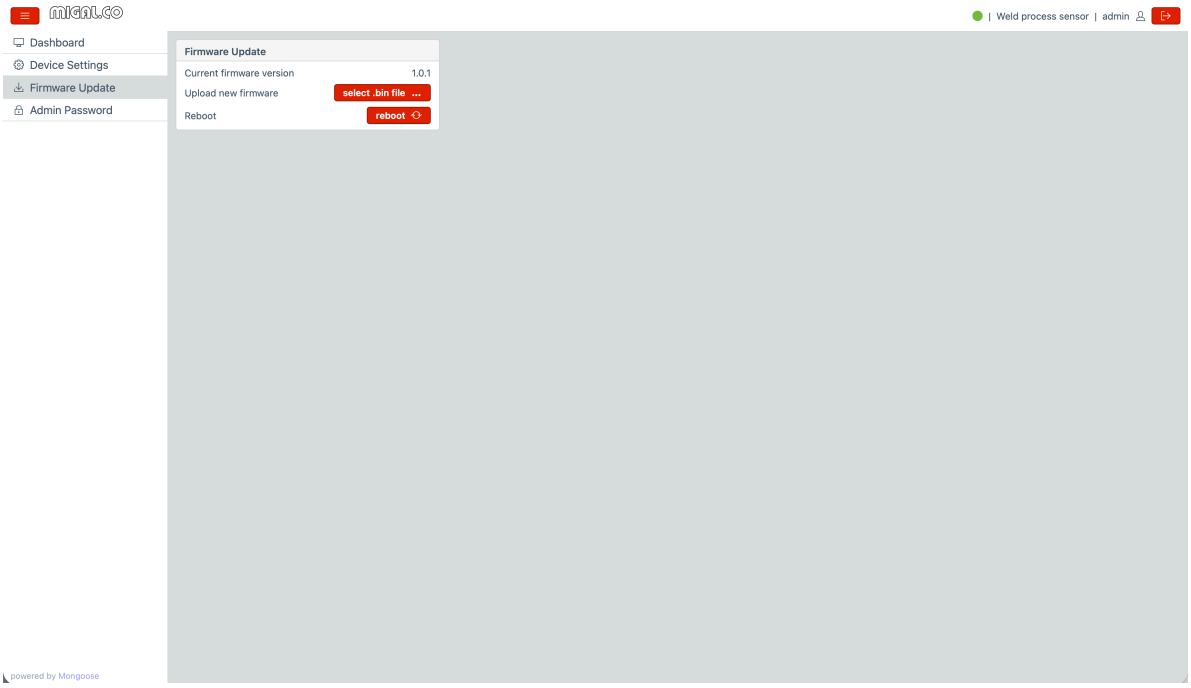
- **admin:** full access to all settings (network, MQTT, certificates, calibration, firmware)
- **user:** access to dashboard and possibly limited settings (depending on firmware)

> **Security recommendation**
> Use strong passwords (min. 12 characters) and change default credentials immediately.

---

19. Firmware Update (notes)

Current firmware information



Under **Current firmware version** the installed firmware version is shown (e.g., **1.0.1**).

Uploading new firmware

1. Click **select .bin file ...**
2. Select the new firmware file in `.bin` format on your computer.
3. Upload starts automatically.

> ⚠️ **Caution**
> Use only **officially released firmware files**. Incorrect or corrupted firmware may cause malfunctions or make the device unreachable.

Rebooting the device

After upload, a reboot is required:

1. Click **reboot** to restart the Sensorbox and activate the new firmware.

After reboot, the Sensorbox is ready for operation.

> 🔐 **Note**
> Device settings usually remain intact during firmware updates. If in doubt, check the **release notes** of the firmware.

Factory reset

To reset the Sensorbox to factory settings:

1. **Disconnect power completely.**
2. **Open the front/cover** to access connectors and buttons (network and power).
3. Press and hold the **blue button (USER)** on the left side of the board.
4. While holding the button, **reconnect power**.
5. Hold the button for **at least 10 seconds**.

6. Release the button.
7. Reinstall the front plate, ensuring **no cables are pinched**.

The device will start with **factory settings**. Stored configurations (network, MQTT, security settings) will be deleted.

> ⚠️ **Note**
> After reset, the Sensorbox is back to **delivery state**. IP address will be obtained via **DHCP** again. Ensure a **DHCP server** is active in the network.

> **Caution – ESD (electrostatic discharge)**
> When opening the housing, electronic components may be damaged by static discharge. Use ESD protection (wrist strap) where possible and avoid touching traces/ICs.

> **Warning – Measurement connection on the welding process**
> Current/voltage signals may only be connected via the intended measurement adapters/tap points.
> Improper connection can result in personal injury and equipment damage.

Note on the MQTT broker

The Sensorbox can transmit measured values via **MQTT** (Message Queuing Telemetry Transport) to an external **MQTT broker**. The broker acts as the central instance for distributing and managing measurement data in the network.

What is an MQTT broker?

An MQTT broker is a server service that mediates between **publishers** (e.g., Sensorbox) and **subscribers** (e.g., visualization, database, dashboard).
It receives messages and distributes them to all authorized recipients subscribing to the respective topic.

Requirements

For MQTT operation, an **MQTT broker must be available** in the network.
The broker is **not** included and is provided by the operator (local or cloud).

Supported MQTT versions

The Sensorbox typically supports:

- **MQTT 3.1.1**
- **MQTT 5.0**

Connections can be:

- **unencrypted (TCP)**, e.g., `mqtt://…`
- **encrypted (TLS)**, e.g., `mqtts://…` *(TLS via certificates in the web UI)*

> **Note**
> For TLS, upload the required certificates under **Device Settings** → **Certificates** (CA, client certificate, private key), if your broker requires client certificates.

Examples of MQTT brokers

Typical broker implementations include:

- **Mosquitto**
- **EMQX**
- **HiveMQ**
- **VerneMQ**

(Selection depends on IT policies, user management, TLS requirements, and scaling.)

Example configuration (Sensorbox)

Example for a local MQTT broker in the company network:

- **Server URL:** `mqtt://192.168.0.10:1883`
  (with TLS: `mqtts://192.168.0.10:8883`)
- **MQTT topic:** `sensorbox/machine1`
- **Username / Password:** configurable by the operator
- **TLS:** optional (upload certificates in the web UI)

> ⚠️ **Caution**
> The MQTT broker must be set up and reachable in the network **before commissioning** so the Sensorbox can send data.

MQTT message (data format)

The Sensorbox sends its measured values as **JSON** (JavaScript Object Notation).
Depending on firmware/active channels, additional fields may be present.

Typical fields (example):

- `timestamp` – ISO-8601 string (UTC)
- `wire_m_min` – float: wire feed speed in **m/min**
- `wire_mm` – float: wire length advanced since previous message in **mm**
- `u_rms` – float: RMS voltage
- `i_rms` – float: RMS current
- `p_active` – float: active power
- `gas_l_min` – float: gas flow in **l/min**
- `status` – object/flags: system status (e.g., NTP synced, ADC overrun, encoder timeout)

> **Note**
> Which fields are actually sent depends on firmware and enabled functions (e.g., “Include status” / “Include calibration”).

Example output

```
```

```
{
  "ts": "2026-01-21T10:15:23.123Z",
  "seq": 4711,
  "u_rms": 23.45,
  "i_rms": 156.78,
  "p_rms": 3680.2,
  "wire_m_min": 8.50,
  "gas_l_min": 12.30,
  "gas_cf": 1.07,
  "status": {
    "openamp_ok": true,
    "adc_overrun": false,
    "encoder_timeout": false,
    "ntp_synced": true
  },
  "fw": {
    "m7": "1.0.3",
    "m4": "1.0.3"
  }
}
```


UDP (real-time streaming)

In addition to MQTT, the Sensorbox can transmit measured data via **UDP**. UDP is particularly suitable for **real-time applications** such as live visualization (scope), fast diagnostics, or recording raw data because it has very low latency and does not require a connection setup.

What is UDP?

UDP (User Datagram Protocol) is a connectionless transport protocol.
The Sensorbox sends data packets (“datagrams”) to a target address (host) and target port—without checking whether the receiver is reachable or whether packets are lost.

Advantages

- Very low latency (ideal for live visualization)
- Low overhead, high data rates possible
- Easy to feed into tools/software (e.g., RoboScope)

Limitations

- No delivery guarantee (packets may be lost)
- No guaranteed ordering
- No built-in encryption/authentication (unlike TLS for MQTT)

> **Note**
> For logging/quality data, MQTT/DB is often used. For live scope/diagnostics, UDP is usually the best choice.

Requirements

- Sensorbox and receiver (PC/server) are in the same network, or routing allows UDP.
- Firewall rules must allow UDP on the used port (example: **5005**).
- The receiver must listen on the specified port (e.g., RoboScope, your own tool, Python script).

UDP settings in the web interface

(Device Settings → UDP Settings)

- **Enable UDP** – enables/disables the UDP stream.
- **Host address** – IP address of the receiver (e.g., PC, server). Example: `192.168.0.74`
- **Host port** – UDP port on the receiver. Example: `5005`
- **UDP mode** – data format selection. Typically **binary** (compact, efficient).
- **Include status** – includes status information in the stream (e.g., NTP synced, ADC overrun, encoder timeout).
- **Include calibration** – includes calibration information (gain/offset, etc.) so the receiver can reproduce the conversion.

Save via **save**.

Typical use cases

Live visualization (scope)

- current/voltage time traces
- detect triggers/peaks/drops
- evaluate stability (spatter, arc outage, short-circuit clustering)

Diagnostics & commissioning

- verify sensors are connected correctly
- check sampling rate, data rate, buffering
- debug “ADC Overrun” or “Encoder Timeout”

Raw data for analysis

- offline analysis (FFT, histogram, feature extraction)
- compare jobs/parameters

Network and security notes

> ⚠️ **Caution (network)**
> UDP can generate high data rates. In sensitive production networks, enable UDP only selectively and for limited time.

> ⚠️ **Caution (security)**
> UDP is not encrypted. Use UDP streaming only in trusted networks (e.g., VLAN, isolated production network) or use secure tunnels/VPN if required.

Troubleshooting (short)

- **No data at receiver**
  - UDP enabled? (**Enable UDP**)
  - Host address correct?
  - Host port correct and receiver listening?
  - Firewall blocking UDP port?
  - Correct netmask/gateway if subnets involved?
- **Stuttering / dropouts**
  - network overloaded (switch/port/VLAN)
  - Wi-Fi instead of Ethernet (not recommended for high UDP rates)
  - receiver PC overloaded (visualization/logging)
- **Status "ADC Overrun"**
  - system load too high / data rate too high
    (Details in "Status messages".)

UDP protocol "OSCI" stream (binary)

The Sensorbox can send measurement data as a **binary UDP stream**. This protocol is optimized for **live visualization** (scope), fast diagnostics, and raw data analysis.

Overview

A UDP packet consists of:

1. **Header (variable length)** – metadata (timestamp, sequence, number of samples, gas/wire, optional status, optional calibration).
2. **Payload (samples)** – `sample_cnt` samples, each **4 bytes** (packed: U_raw + I_raw).

All fields are **little endian**.

Header (base)

Layout (base header, fixed 16 bytes)

Format: `BASE_HDR_FMT = "<IHHII"`
Total length: `BASE_HDR_LEN = 16`

Offset	Size	Type	Field	Description
0	4	`uint32`	`magic`	Magic: `0x3143534F` (ASCII `"OSCI"`)
4	2	`uint16`	`header_len`	Total header length in bytes (incl. optional fields)
6	2	`uint16`	`flags`	Bitflags: status/calibration included
8	4	`uint32`	`seq`	Packet sequence number (incrementing)
12	4	`uint32`	`sample_cnt`	Number of payload samples

Flags

Flag	Value	Meaning
`FLAG_STATUS`	`0x0001`	Status field present in header
`FLAG_CAL`	`0x0002`	Calibration data present in header

Header (mandatory fields after base header)

After the base header, the following 16 bytes always follow:

Timestamp (8 bytes)

Offset	Size	Type	Field	Description
16	8	`uint64`	`ts_ms`	Timestamp in milliseconds

> Note: Timestamp is an **ms counter** (epoch/RTC depends on firmware). For a clean time base, enable NTP/SNTP.

Process metadata (8 bytes)

Offset	Size	Type	Field	Description
24	4	`float32`	`gas_l_min`	Gas flow in **l/min**
28	4	`float32`	`wire_m_min`	Wire feed in **m/min**

Minimum header length:

- `16 (base) + 8 (ts) + 8 (meta) = 32 bytes`

Status bitfield (`status`)

If `FLAG_STATUS (0x0001)` is set in the header flags, the header contains a 32-bit status bitfield (`uint32 status`).

Bits are defined as:

Bit	Mask	Name	Meaning
0	`0x00000001`	`ST_OPENAMP_OK`	Internal communication/subsystem (OpenAMP) OK
1	`0x00000002`	`ST_ADC_OVERRUN`	ADC overrun detected
2	`0x00000004`	`ST_ENCODER_TO`	Encoder timeout
3	`0x00000008`	`ST_NTP_SYNCED`	Time base synchronized via NTP/SNTP
4	`0x00000010`	`ST_CAL_ACTIVE`	Calibration active
5	`0x00000020`	`ST_CFG_APPLIED`	Configuration applied

Interpretation

- Set bit means: **condition active/true**
- Multiple bits can be set at the same time (bitwise OR).

Example (hex → meaning)

Example: `status = 0x00000019`

- `0x01` → OpenAMP OK
- `0x08` → NTP synced
- `0x10` → Calibration active

So **OpenAMP OK**, **NTP synced**, **Calibration active**.

Recommended UI indication

- **Green**: `ST_OPENAMP_OK` set and **no** error bits (`ST_ADC_OVERRUN`, `ST_ENCODER_TO`)
- **Yellow/Orange**: `ST_CFG_APPLIED` missing or NTP not yet synced (startup)
- **Red**: `ST_ADC_OVERRUN` or `ST_ENCODER_TO` set

Optional: Calibration data

If `flags & FLAG_CAL` is set, 16 bytes follow:

Format: `"<ffff"`

Size	Type	Field	Description
4	`float32`	`u_gain`	Voltage gain
4	`float32`	`u_offset`	Voltage offset
4	`float32`	`i_gain`	Current gain
4	`float32`	`i_offset`	Current offset

If no calibration is sent, parser defaults:

- `u_gain = 1.0`, `u_offset = 0.0`
- `i_gain = 1.0`, `i_offset = 0.0`

Payload (samples)

Payload start

Payload always starts at byte offset `header_len`:

- `payload_off = header_len`
- `payload_bytes = len(packet) - header_len`

Sample format (4 bytes per sample)

Each sample is a `uint32` (little endian) containing two 16-bit raw values:

- lower 16 bits: `u_raw`
- upper 16 bits: `i_raw`

```
uint32 w
u_raw = w & 0xFFFF
i_raw = (w >> 16) & 0xFFFF
```

Terminology & measurement method for current, voltage and power

The MIGAL.CO Sensorbox captures **voltage** *u(t)* and **current** *i(t)* as high-frequency raw data and calculates stable KPIs for live display (dashboard) and for logging/analysis.

> **Sampling note**
> Raw data is captured at a fixed sampling rate (e.g., **5 kHz** – depending on firmware/setup).
> For the UDP OSCI stream, a higher effective sample rate can be used (see “UDP protocol”).

Why RMS?

In welding, voltage and current are typically **non-sinusoidal** (e.g., short-circuit phases, pulsed MIG, arc instability).
RMS values provide a robust and physically meaningful quantity describing the “effective” voltage/current—independent of waveform.

How is power calculated?

The Sensorbox calculates **instantaneous power** per sample:

- $p(t) = u(t) \cdot i(t)$

and averages it over a short time window to obtain a stable active power value that reliably reflects real process changes.

These KPIs form the basis for:

- live monitoring (dashboard / alarms)
- trend analysis and process comparison
- quality evaluation
- later AI-based defect detection

Formulas

A time window of **N samples** is assumed:
u[k] and *i[k]* are voltage and current at sample index *k*.

RMS voltage

$$U_{\mathrm{RMS}} = \sqrt{\frac{1}{N}\sum_{k=1}^{N} u[k]^2} \tag{1}$$

RMS current

$$I_{\mathrm{RMS}} = \sqrt{\frac{1}{N}\sum_{k=1}^{N} i[k]^2} \tag{2}$$

Mean active power

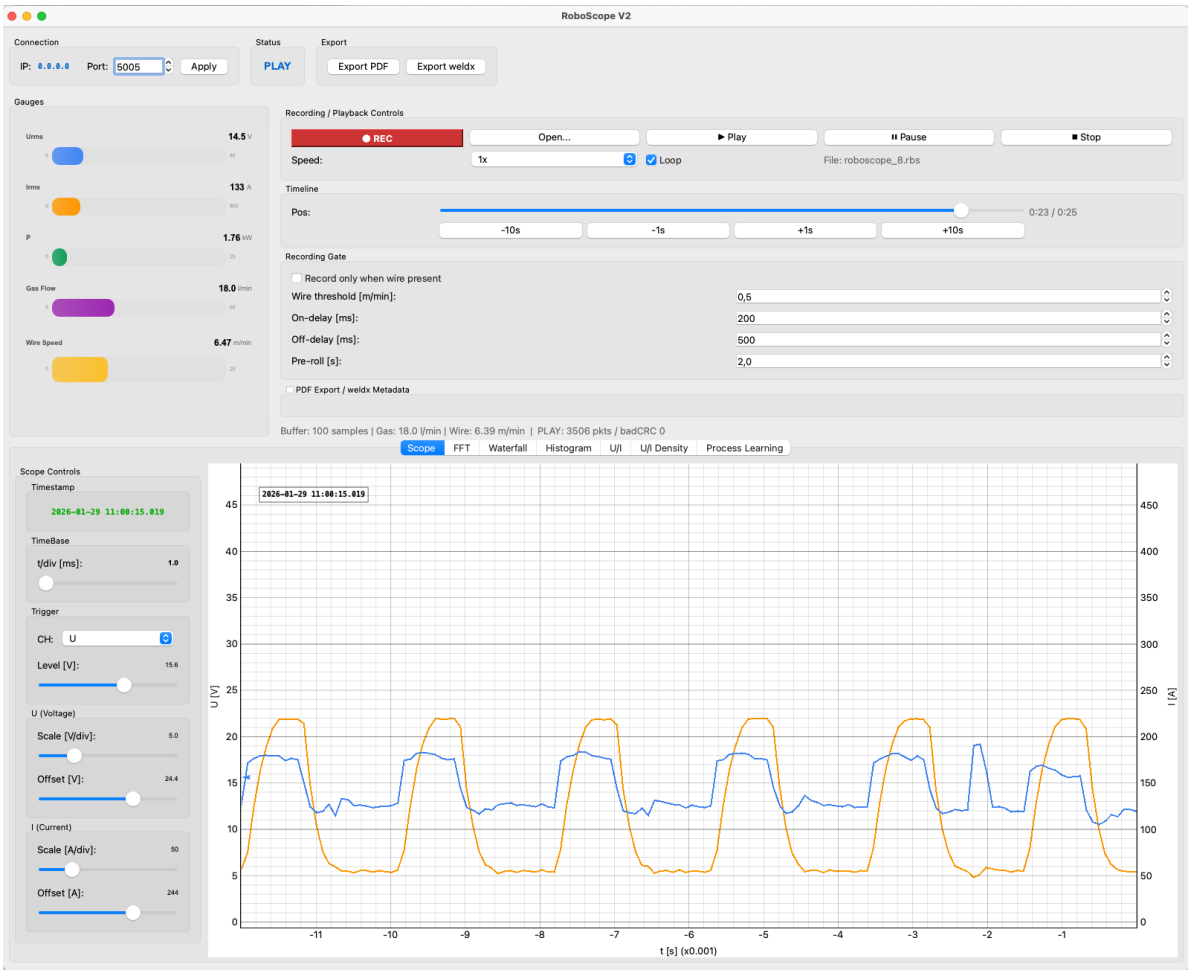
$$P = \frac{1}{N}\sum_{k=1}^{N} \left(u[k] \cdot i[k]\right) \tag{3}$$

> **Note**
> The window size (N) affects display “smoothness”:
>
> - small window → fast but noisy
> - large window → stable but sluggish

---

20. Additional software: RoboScope (optional)

RoboScope is an optional PC software for **live visualization**, **diagnostics**, and **analysis** of welding process data captured by the Sensorbox. For normal operation (e.g., MQTT → database/dashboard), RoboScope is **not required**, but it provides significant added value for **commissioning**, **troubleshooting**, and **process optimization**.



Purpose and application

RoboScope is typically used when you want to:

- quickly verify that **current and voltage measurement** works correctly,
- assess **signal quality** (noise, peaks, drops, interference),
- visualize **short-circuit/arc characteristics**,
- narrow down causes of **instability**, **spatter**, or **power drops**,
- or temporarily obtain a “scope view” of the process without additional measurement equipment.

RoboScope usually receives the Sensorbox **UDP real-time stream** (binary, low latency).

Main functions in RoboScope

1) Live visualization (scope)

RoboScope displays **current and voltage waveforms** in real time, making process events immediately visible, e.g.:

- peaks during ignition
- drops during contact/short-circuit phases
- periodic patterns (e.g., pulsed MIG)

- unstable waveforms (e.g., erratic arc)

2) U–I diagram (process characteristic)

In addition to time plots, RoboScope can display a **U–I diagram** (current over voltage), useful to see how the process moves in the characteristic field:

- compact cloud → stable process
- wide spread → fluctuating process / suboptimal parameters
- outliers → disturbances, contact issues, wire feed problems

3) Spectral analysis (FFT)

The **FFT/spectrum view** shows frequency components in the signals and helps with:

- detecting oscillations (mechanical/electrical)
- interference frequencies due to EMC
- periodic processes (pulse frequencies, clocking)
- comparing “clean” vs. “disturbed” components

4) Histogram / distribution analysis

The histogram view shows the **distribution** of values over time:

- narrow distribution → stable
- broad distribution → high scatter
- multiple peaks → switching states (e.g., different arc phases)

> **Summary**
> RoboScope is an optional tool for a quick, intuitive live view of the welding process. It complements the Sensorbox web interface and is especially useful for commissioning, diagnostics, and optimization.

---

21. Troubleshooting (short list)

Symptom	Possible cause	Remedy
Web interface not reachable	wrong IP / DHCP off / wrong network	determine IP via router/scan, enable DHCP, check subnet
MQTT not connecting	broker unreachable / port blocked / wrong credentials	test broker/port, check user/password, open firewall
MQTT TLS error	CA/client cert/key wrong or missing	check certs, check date/time (NTP!), verify broker config
Timestamps wrong	NTP not synced	set SNTP server, allow UDP 123, check “NTP Synced” status
Encoder timeout	sensor/encoder not connected or signal issue	check wiring/connector/gap, status bit `ST_ENCODER_TO`
ADC overrun	system load too high / data rate too high	review settings, reduce UDP/MQTT rate, diagnose with RoboScope

---

22. EU Declaration of Conformity

Manufacturer:
MIGAL.CO GmbH
Wattstraße 2
94405 Landau a. d. Isar
Germany

Product:
Sensorbox

Type designation:
MSB-1

Intended purpose:
Capturing, pre-processing, and transmitting process data in industrial welding applications (e.g., current, voltage, wire feed, gas flow, status signals). Transmission via Ethernet (incl. MQTT/UDP/OPC UA) to higher-level systems.

We hereby declare that the above product complies with the essential requirements and other relevant provisions of the following EU directives:

- **2014/30/EU** – EMC Directive
- **2011/65/EU** – RoHS Directive

Applied harmonized standards *(examples – final verification required)*

- **EN 61000-6-2:2019** – EMC – Immunity (industrial environment)

- **EN 61000-6-4:2019** – EMC – Emission (industrial environment)
- **EN 62368-1:2020** – Safety of audio/video, information and communication technology equipment *(relevant when using PoE or power supply)*

Additional information

- **Supply voltage:** 9–12 V DC or PoE *(IEEE 802.3af, if supported)*
- **Interfaces:** Ethernet
- **Data transmission:** MQTT v3.1.1 / v5, UDP (binary), OPC UA
- **Encryption:** TLS
- **Protection class:** IP20
- **Environment:** industrial applications

Place, date:
Landau, *27 Jan 2026*

Signature:

Robert Lahnsteiner
Managing Director
MIGAL.CO GmbH