



IO Telemetry Gateway

Technical specification. What the appliance reads, what it emits, and what it runs on.

TECHNICAL SPECIFICATION
INDUSTRIALOBSERVABILITY.COM/GATEWAY

Collection

Field protocols	Modbus TCP, Modbus RTU (RS485 serial), Modbus RTU over TCP
Serial parameters	Baud rate, data bits, parity and stop bits configurable per endpoint
Addressing	Data-model addressing or wire offsets, declared per device definition. The collector refuses a configuration that mixes the two conventions ambiguously.
Data types	int16, uint16, int32, uint32, float32, with big and little word order; per-point scaling factors; enumerated states mapped to names
Counters	Energy and other accumulating registers declare their rollover point, so a wrapped counter yields a correct rate rather than a negative spike
Polling	Per-site default interval with optional jitter; per-device override for slowly changing plant
Topology	Many devices behind one RS485 gateway, or a device with its own Ethernet port as its own endpoint. Requests to a shared bus are serialised with a configurable inter-request delay.

Delivery

Metrics endpoint	Prometheus exposition over HTTP, on a configurable listen address and path. Bind to localhost and scrape locally where the endpoint should not be on the network.
Vendor-neutral export	OpenTelemetry (OTLP) export
Naming	Prometheus naming and unit conventions; site vocabulary applied as labels from the device configuration
Health	Per-device reachability and per-point freshness are exposed as metrics, so a device that stops answering is an alert rather than a flat line nobody noticed

Configuration

Format	YAML. Site configuration in one file; device definitions as separate profile files in a directory beside it.
Portability	Configuration is plain text and is held outside the appliance as well as on it. Moving a site to the cold spare is copying two things.
Validation	The configuration and every referenced device definition are validated before the collector touches any equipment.

Change control	A read-only mode refuses configuration changes made on the appliance, for sites where Ansible or another configuration manager owns the file. A save otherwise keeps a backup beside the config and writes an audit record.
----------------	---

Software

Collector	Written in Go, shipped as a single static binary with no runtime, interpreter or shared-library dependencies
Operation	Runs as an unprivileged systemd service with the standard sandboxing options applied
Local interface	A terminal interface on the appliance for commissioning and fault-finding, showing per-device state and live values
Commissioning tools	Configuration check, resolved-addressing table, single-shot read with machine-readable output and a status code for scripting, and a raw-register decoder
Access	The collector reads. Writing to equipment is not part of what the appliance is for.

Hardware

Platform	Revolution Pi Connect 5
Mounting	DIN rail, 35 mm
Supply	24V DC, typ. 8W
Cooling	Fanless. No moving parts.
Compliance	CE marked
Resilience	Specified as two identical units per site: one live, one preconfigured cold spare

Support

Included	Device definition updates, collector updates and support, covered by the subscription
Register maps	Manufacturer firmware changes to a register map are tracked and the definition reissued
Replacement	A failed unit is replaced, preconfigured for the site

Configuration, in full

A site configuration names the endpoints — the network or serial paths to equipment — and the devices behind them. Each device points at a profile, which is the maintained device definition.

```

version: 1
site: lon1
namespace: io

poll:
  default_interval: 60s
  jitter: 5s

metrics:
  enabled: true
  listen: 127.0.0.1:9464
  path: /metrics

profiles_dir: ./profiles

endpoints:
  # An RS485-to-Ethernet gateway with meters behind it.
  - id: gw_pwr_a
    driver: modbus
    kind: rtu_over_tcp
    address: 10.114.20.11:502
    timeout: 2s
    inter_request_delay: 30ms

  # A UPS with its own Ethernet port is its own endpoint.
  - id: ups_a1
    driver: modbus
    kind: modbus_tcp
    address: 10.20.4.20:502

devices:
  - {id: pdu_a1, endpoint: gw_pwr_a, address: "2", profile: schneider_pdu, tags: {room: dh1, row: A}}
  - {id: pdu_a2, endpoint: gw_pwr_a, address: "3", profile: schneider_pdu, tags: {room: dh1, row: A}}
  - {id: ups_a1, endpoint: ups_a1, profile: apc_ups, tags: {room: dh1, row: A}}

```

Devices behind a gateway carry their unit id, as pdu_a1 and pdu_a2 do here. A device with its own Ethernet port is identified by its endpoint alone.

Commissioning

Before the appliance connects to anything, it will tell you how every point in a device definition resolves onto the wire. This is the table to hold against the equipment's datasheet.

```

$ io --addresses schneider_pdu.yaml

profile schneider_pdu - Schneider floor PDU (driver: modbus)
all points: mode=data_model fc=3 write_fc=-

POINT          UNIT  ACCESS  SOURCE  WIRE          COUNT  TYPE    WORD_ORDER
active_power_total kW    r       40117   116 (0x0074)  1      int16   -
voltage_l1      V     r       40100   99 (0x0063)  2      uint32  big

```

Nothing is contacted to produce it. The same command validates the configuration, so a typo is an error naming a line rather than a device that silently never reports.

A single-shot read then confirms the site is answering, and exits with a status code a commissioning script can act on.

Code	Meaning
0	Every enabled device answered
1	Some answered, some did not
2	None answered, or the configuration could not be used

Identifying an unknown register

Where equipment is not yet mapped, the appliance decodes a raw reading every plausible way, so a type and word order are settled against the equipment's own display rather than guessed. It needs no configuration and no connected device.

```
$ io --decode --scale 0.1 0x0000 0x0961
```

```
raw words (2 registers, as read)
```

```
[0] 0x0000 (0)
```

```
[1] 0x0961 (2401)
```

CANDIDATE	REGISTERS	VALUE	SCALED (x0.1)
uint16	1	0	0
int16	1	0	0
uint32 (big)	2	2401	240.1
uint32 (little)	2	157351936	15735193.6
int32 (big)	2	2401	240.1
int32 (little)	2	157351936	15735193.6
float32 (big)	2	3.364517613e-42	3.364517613e-43
float32 (little)	2	2.708338984e-33	2.708338984e-34

240.1 volts is the only row that is a plausible line voltage. That settles the type.

Not included

The appliance is not DCIM and does not provide an asset database, rack planning, cable management, work orders or capacity planning. It does not provide dashboards, alerting rules or long-term storage — those are your monitoring platform's job, and the gateway is built to feed it rather than replace it.

Talk to us about your site.

Send the equipment list and the number of sites.

industrialobservability.com/gateway/enquire

hello@industrialobservability.com

Industrial Observability is a trading name of **The Melt Cafe Ltd**, registered in England and Wales, company number 09912074.

Manufacturer and product names identify compatible equipment and are trademarks of their respective owners. Their use here does not imply endorsement, affiliation or partnership.