periNODE Modbus

Single Pair Ethernet to Modbus RTU Adapter

Datasheet

Perinet
Seamless IoT Connectivity

Abstract

The *periNODE Modbus* IoT sniffs the Modbus RTU bus and delivers the data to the IT-based infrastructure via Single Pair Ethernet.

It makes Modbus RTU infrastructure CRA-compliant through state-of-the art security measures while the *periNODE Modbus* brings key features like remote firmware updates, automatic device discovery as well as a RESTful API for automated access.

Doc-No.: PRN.100.887 rev: 1 2025-12-12

Document Information

Title	periNODE Modbus
Subtitle	Single Pair Ethernet to Modbus RTU Adapter
Type	Datasheet
Status	Release
Version	1
Date	2025-12-12
Disclosure Restriction	

Contents

1 Overview

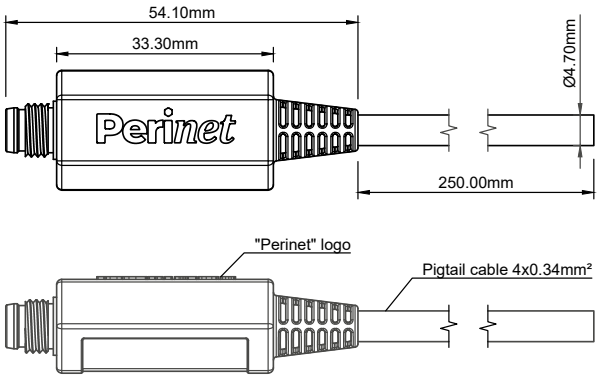
The *periNODE Modbus* IoT sniffs the Modbus RTU bus and delivers the data to the IT-based infrastructure via Single Pair Ethernet.

It makes Modbus RTU infrastructure CRA-compliant through state-of-the art security measures while the *periNODE Modbus* brings key features like remote firmware updates, automatic device discovery as well as a RESTful API for automated access.

Key Features

- Fully operational with One Cable
- Integrated Software Stacks: TCPIPv6, mTLS, HTTPs, RESTful API
- Integrated WebUI & RESTful API
- Based on *sève OS*, an Operating System for Things
- Customizable Firmware
- On device value transformation (Polynomial calculation)

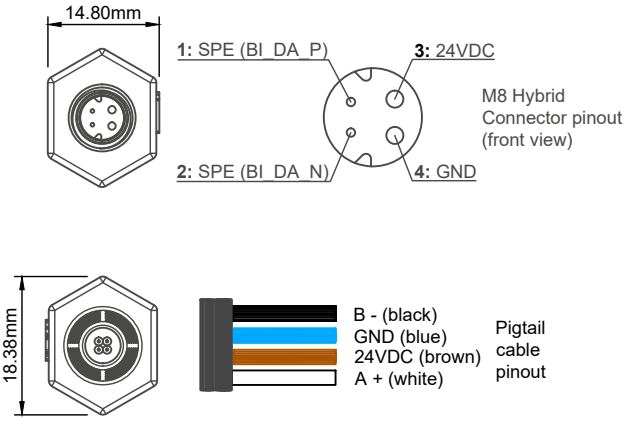
Dimensions & Pinout



periNODE Modbus's dimensions in mm.

Interfaces

- 100BASE-T1 Phy (IEEE 802.3bw, IEC 63717-6)
- Modbus RTU (RS-485)
- Secured Modbus TCP (mTLS)
- HTTPs RESTful API for configuration



periNODE Modbus's connector pinning.

Operational Parameters

Operating voltage:	24VDC input
Supply Voltage:	24V output (2A max)
Temperature range:	-40°C to +70°C
Power consumption:	0.6W
Max Baud Rate:	1Mbaud
Protection Class:	IP 65/67

Compliance

- RoHS 3 (EU 2015/863)
- WEEE (2012/19/EU)
- REACH
- CE

Security

- CRA-ready

- NIST compliant TLS implementation
- Role Based Access Control (RBAC)
- Certificate based client authentication
- AES encryption algorithm
- X.509 certificates & PKIX path validation
- Elliptic Curve Cryptography (ECC)

Software Features

- Integrated HTTP server with TLS
- Mutual TLS (mTLS) based Authentication
- Integrated MQTT-client, with mTLS
- Zero-configuration (mDNS and DNS-SD)
- Full TCP/IPv6 communication stack
- LifeCycle Management, remote firmware update

Typical Application

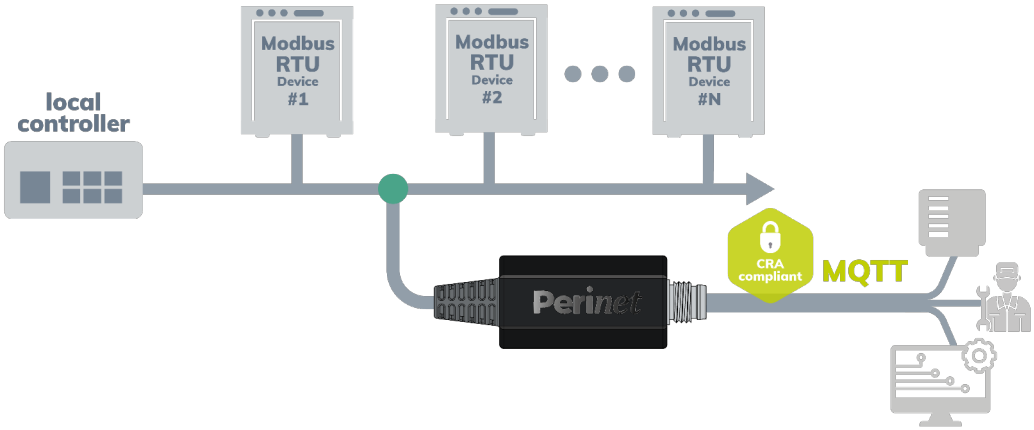


Figure 1: Typical secured installation of a periNODE Modbus.

2 Introduction

2.1 Single Pair Ethernet

Single Pair Ethernet (SPE) describes a single twisted pair of wires that is one aspect of the physical layer for an Ethernet based communication. The periNODE Modbus uses a point-to-point full-duplex communication scheme with 100Mbits/s throughput on the physical layer as specified by the IEEE [17]. Figure 2 shows conventional Ethernet compared to SPE on a simplified overview.

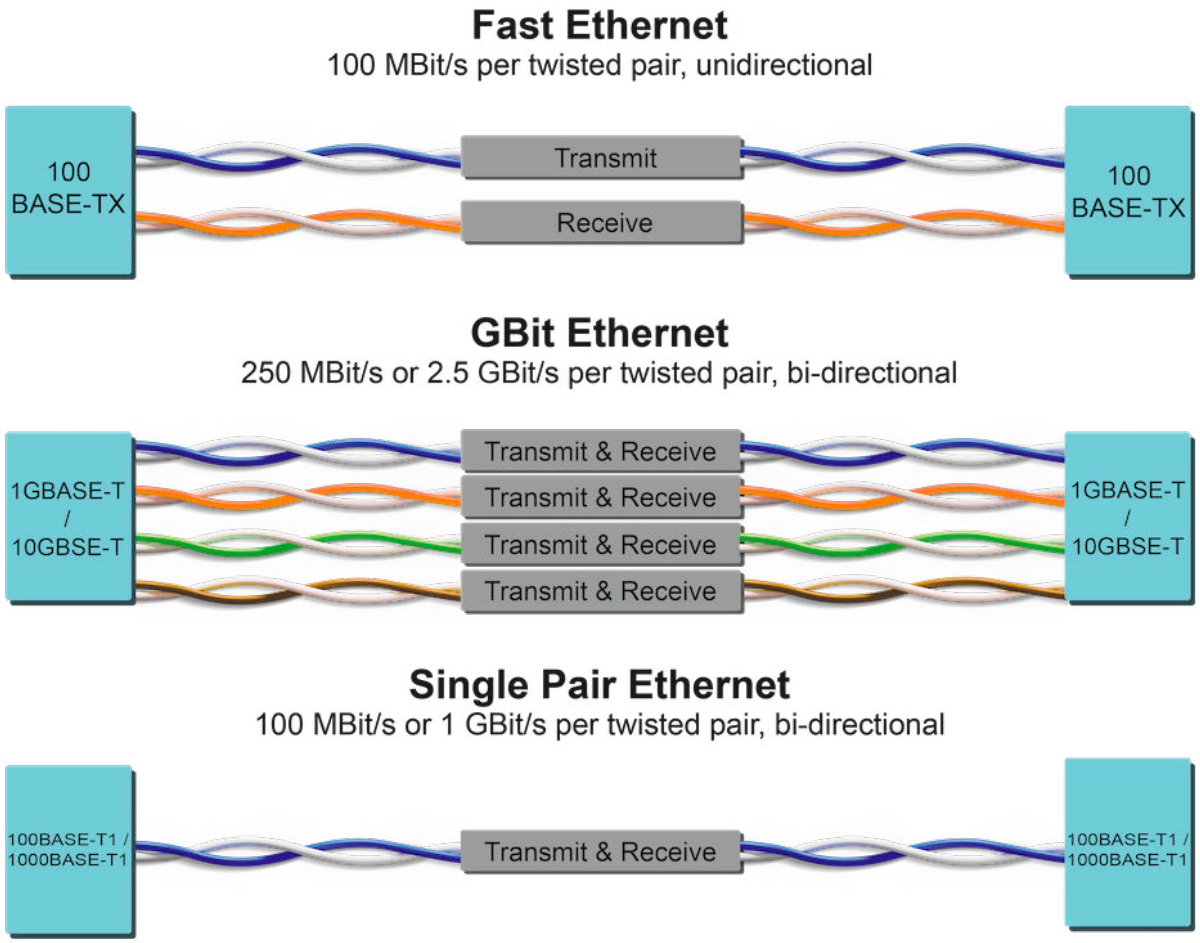


Figure 2: Single Pair Ethernet compared to other Ethernet technologies.

2.2 Modbus

Modbus RTU is a master/slave based, half-duplex, binary data formatted transition communication protocol [19] widely used in industrial automation and control systems. The master can be an HMI interface such as a data logger program software running on a computer, or a SCADA system. The slaves can be sensors, actuators or other control devices such as a PLC. Modbus TCP is the other variation of the Modbus protocol operating over TCP/IP networks

[18]. The *periNODE Modbus* supports the *RS-485* interface on the physical layer for Modbus RTU protocol, as well as the 100BASE-T1 full-duplex interface for the Modbus TCP protocol.

3 Architecture

The periNODE Modbus structure is summarized in the block diagram shown in Figure 3. A periNODE Modbus includes the periCORE SPE communication module and an RS-485 half-duplex transceiver. The former connects to an Ethernet IT-Network with SPE via the Network Interface and the latter implements the Modbus serial line physical interface [19] that connects to any Modbus RTU device via the Peripheral Interface of a periNODE Modbus.

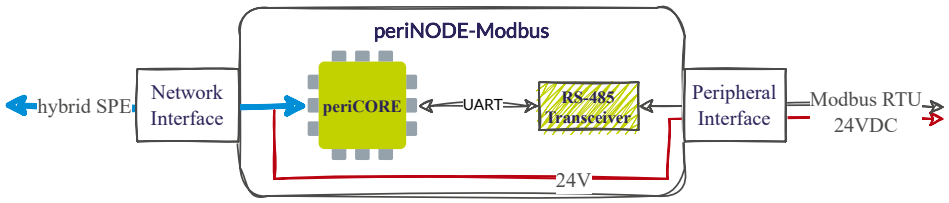


Figure 3: periNODE Modbus hardware block diagram

`Network Interface` is the physical interface which connects to the Ethernet based network infrastructure. It uses hybrid SPE which transports data and power supply on dedicated wire pairs.

`periCORE` is the Single Pair Ethernet communication module [5]. It implements the SPE communication interfaces, the peripheral serial interface and hosts the software for secured communication and the Modbus functionality.

`RS-485 Transceiver` integrates a *RS-485* receiver and transmitter in a single IC. The RS-485 standard (TIA-485(-A) or EIA-485, [1]) defines a half-duplex serial peripheral interface.

`Peripheral Interface` the physical interface, which connects a periNODE Modbus a Modbus RTU compliant device.

3.1 Typical Application

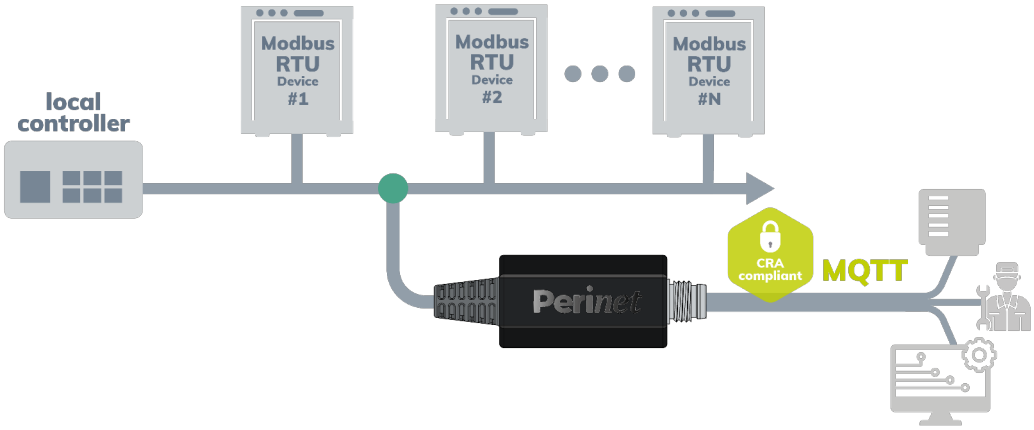


Figure 4: Typical secured installation of a periNODE Modbus.

The periNODE Modbus IoT passively sniffs the Modbus RTU bus and forwards the data traffic to the MQTT broker based on the *Modbus Segment Name*. Communication between the periNODE Modbus and the broker is secured with a TLS connection and can be further strengthened by using mTLS, ensuring mutual authentication. The Figure 4 represents this scenario.

3.1.1 periMICA Modbus IoTizer

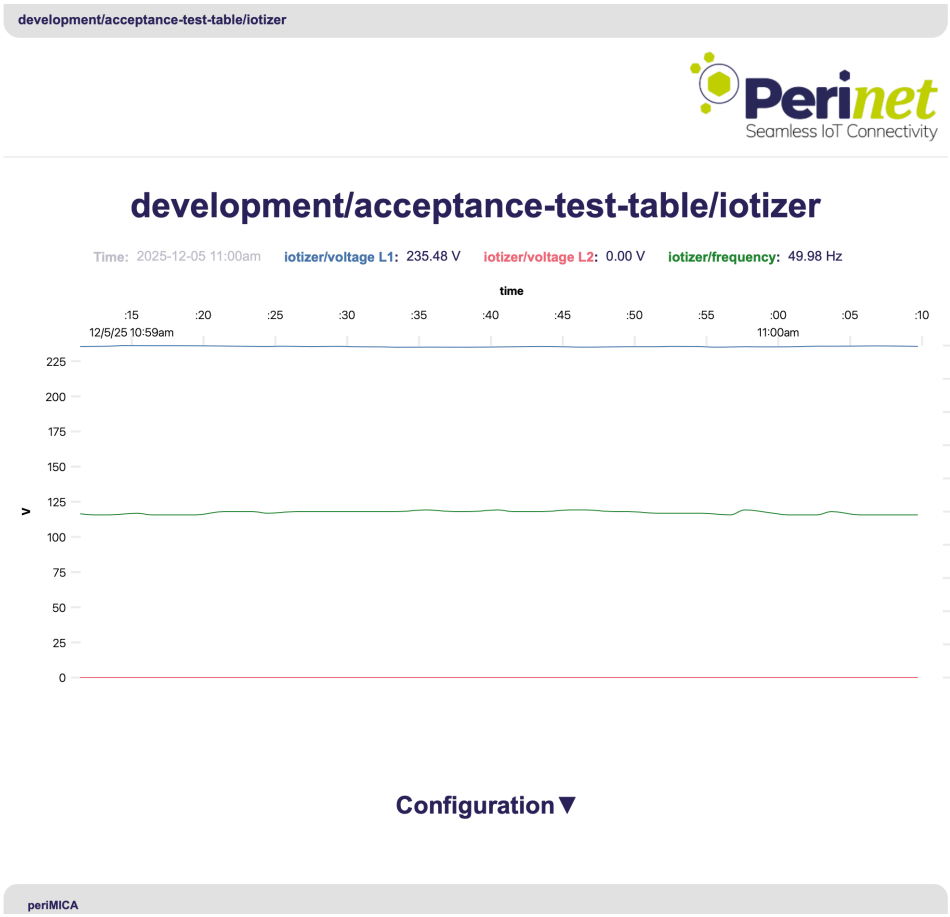


Figure 5: Dashboard periMICA IoTizer Container

The periMICA Modbus IoTizer container is designed to fetch the data trasmitted by one and only one device in the bus, configured by the device address, and translate the data frame into a meaningful telemetry message delivered to the MQTT broker. The data is received via MQTT protocol and processed according to the configuration set in the periMICA IoTizer container web user interface, as shown in Figure 5.

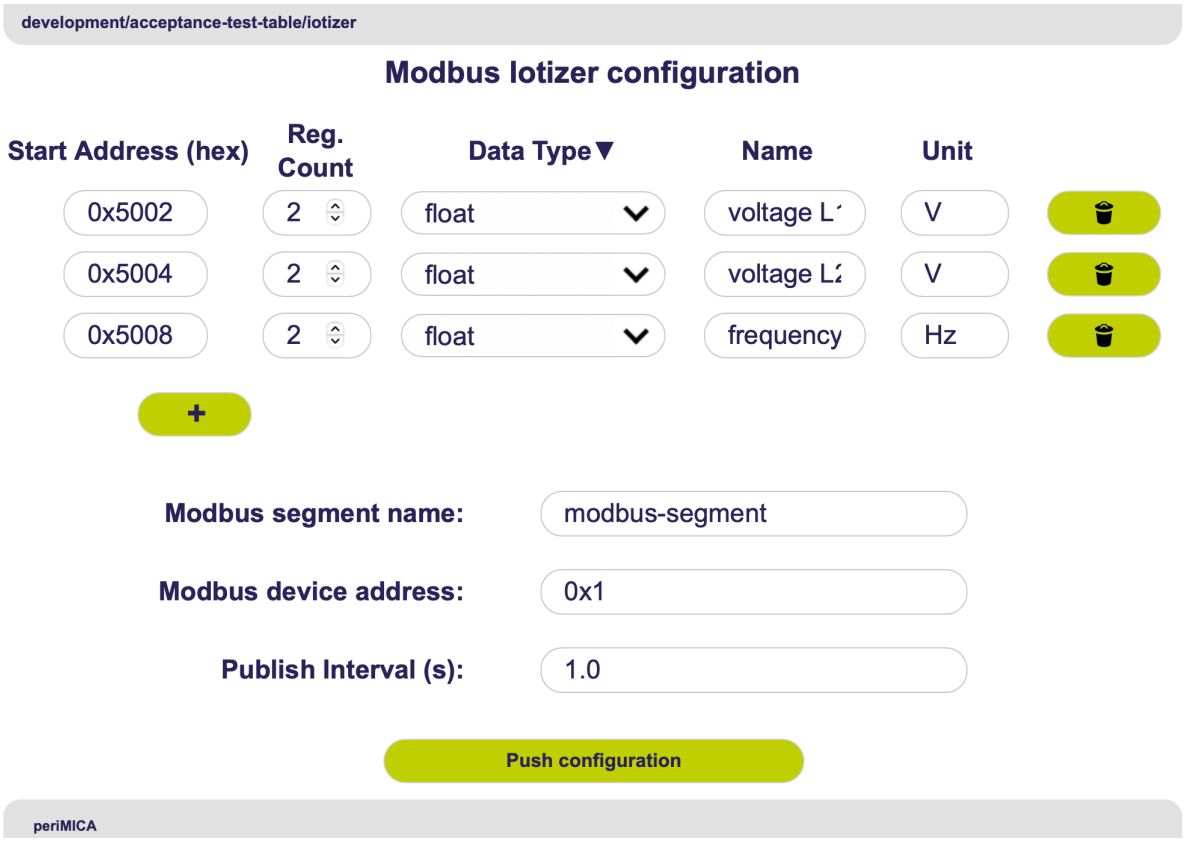


Figure 6: Configuration periMICA IoTizer Container

The communication between the periMICA IoTizer container and the broker is secured with a TLS connection. The periNODE Modbus IoT as well as the periMICA IoTizer follows the Perinet security guidelines and be enforced to use mTLS, ensuring mutual authentication.

3.2 Firmware Variants

The periNODE Modbus is available in two firmware variants:

- *periNODE-Modbus IoT*: This firmware variant implements the Modbus RTU bus sniffer functionality and the Modbus-IoT API, as basically described in the first release of this document.
- *periNODE-Modbus Bridge*: This firmware (version 3) is the production firmware and differs in its functionality. A periNODE Modbus running this version is able to read RTU frames coming from exactly one device and send RTU frames to the connected device, functioning as a bridge between the Modbus RTU and TCP networks.

Note: Performing a reset will always restore the device to the production state and requires a firmware update to switch to the periNODE Modbus IoT firmware which provides the sniffer mode.

4 Electrical Specifications

4.1 Signal Types

Type Name	Description
DC_PWR	Direct current supply
DC_GND	Common Ground supply
SPE_DATA	Digital differential communication
Bus_DATA	RS-485[1] differential pair (A and B)

Table 1: Signal Type Definitions

4.2 Absolute Maximum Ratings

Parameter	Min	Max	Unit
Power supply voltage	0	27	V
Differential voltage between SPE_DATA signals	-16	16	V
Individual or common mode voltage on SPE_DATA signals	-8	8	V
Differential voltage between Bus_DATA signals	-3.5	3.5	V
Individual or common mode voltage on Bus_DATA signals	-27	27	V

Table 2: Absolute maximum ratings of the periNODE Modbus

Warning: Exceeding the specified absolute maximum ratings may damage the periNODE Modbus.

4.3 Common specifications

Parameter	Condition	Min	Typ.	Max	Unit
Supply voltage (U)		21.6	24	26.4	V
Power consumption	Supply voltage U = 24V	-	0.6	-	W

Table 3: Operational conditions

4.4 Bus specification

Parameter	Condition	Min	Typ.	Max	Unit
Internal termination resistance (R_T)			120		Ω
Input voltage, individual or common mode (Vi)		-7		12	V
Differential input voltage		-3		3	V
Output current (I_O)		-60		60	mA
Differential voltage output (V_{OD})	R_L = 60Ω	1.5	2		V

Table 4: Bus specification

5 Mechanical Specifications

The periNODE Modbus provides a network interface to connect to an Ethernet based IT-network (via SPE) and a peripheral interface to connect the Modbus serial line physical layer.

5.1 Dimensions

The following Figure 7 visualizes the physical dimensions of the periNODE Modbus.

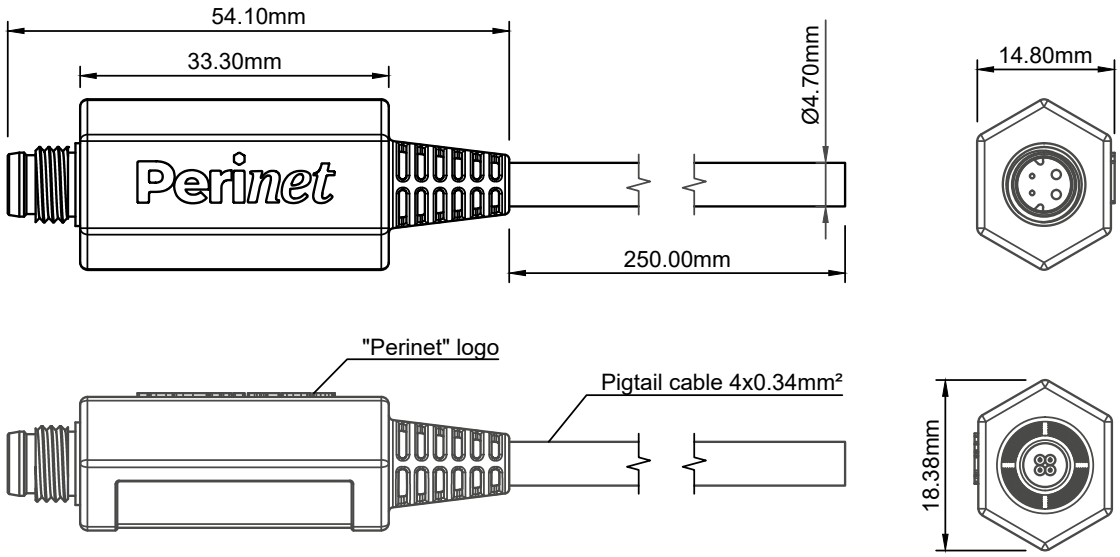


Figure 7: periNODE Modbus's dimensions in mm.

5.2 Network Interface

The Network Interface uses a M8 hybrid male connector. Its style according to the specification IEC 63171-6:2021 [4] is *6P-M8C* (without shield). The interface and therefore its connector combines data (`SPE_DATA`) and power supply (`DC_PWR`, `DC_GND`). (see Figure 8)

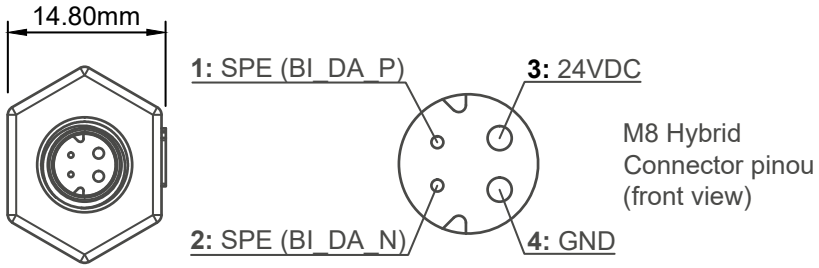


Figure 8: periNODE Modbus Network Interface pinout.

5.3 Peripheral Interface

As peripheral interface a Pigtail with 4 wires is used, see the cable specification at Table 5 Two wires are used for power supply (`DC_PWR`, `DC_GND`) of the peripheral device and the other two as differential pair for the Modbus serial line physical layer (RS-485 [19], `Bus_DATA`). (see Figure 9)

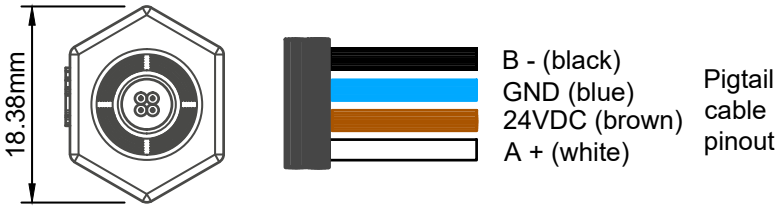


Figure 9: periNODE Modbus Peripheral Interface pinout.

Description	Value	Min	Max	Unit
Dimension (D)	4x0.25			mm^2
Outer diameter (jacket)	3.6			mm
Rated voltage (no power transmission; not for mains)	300			V
Test voltage (A/A)	2000			V
Conductor resistance ($20°$C)			79	Ω/km
Insulation resistance (specific)	7x10^{15}			$\Omega*cm$
Min. bending radius (moving)	10xD		–	
Min. bending radius (fixed)	5xD			–
Temperature range (moving, EN)		-25	80	°C
Temperature range (fixed, EN)		-40	80	°C
Flex cycles (at max. $60°$C)	4000000			$cycles$
Travel length (motion conditions)	10			m
Max. speed (motion conditions)			3	m/s
Max. acceleration (motion conditions)			10	m/s^2

Table 5: periNODE Modbus Cable specification

5.4 Installation

The recommended installation of the periNODE Modbus involves securely mounting the peripheral (e.g. sensor, actuator) and providing strain relief for the cables. A periNODE Modbus is not equipped with mounting features like holes or screw threads. Due to the compact size, a periNODE Modbus can be effectively positioned and supported by its attached peripherals (e.g. sensor, actuator) and cables themselves.

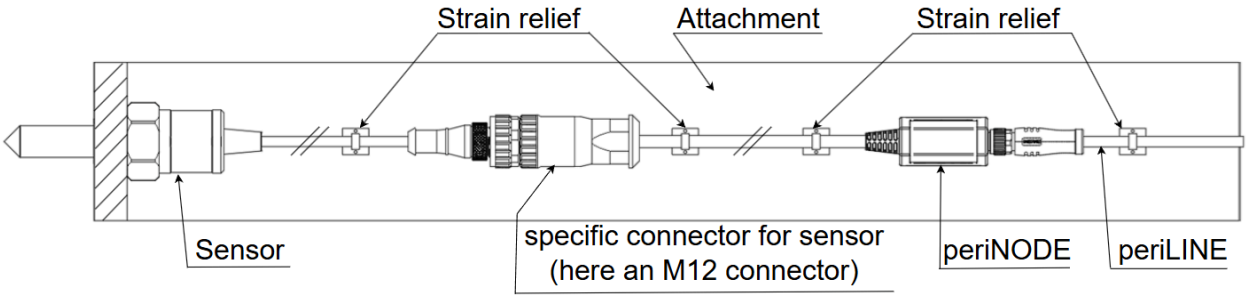


Figure 10: Example installation with support by their and peripheral connector.

Products with connector can be securely attached by plugging them into the mounted sensor (an specific plug needs to be attached to the PeriNODE-cable). To avoid damage from transversal forces, it's important to fix the network cable, e.g. the periLINE, by using a strain relief to a stable structure.

6 API - Application Programming Interface

A periNODE Modbus is a periCORE based product and it features the libperiCORE RESTful API as well as its own periNODE Modbus API. To know more about the periCORE RestFul API, please refer to periCORE Datasheet [5]. In this section the periNODE Modbus RESTful API will be described. Please be advised, that the periCORE based products feature a built-in authentication mechanism implemented in the libperiCORE firmware, it will be explained in the next webUI section 7.2.4 and more details can be found periCORE Datasheet [5].

To connect to the device and access the resources, use the hostname `periNODE-sernm.local`, where *sernm* is a unique identifier based on the product's serial number as explained in Section 9. The API provides various routes, which are listed in Table 6. These routes and their functions will be detailed in the subsequent subsections.

URL Resource	Description
/modbus-iot/config	See Section 6.1
/modbus-iot/sample	
/modbus-iot/rtu/config	See Section 6.2

Table 6: RESTful API routes overview

6.1 Modbus IoT

/modbus-iot/config Return `ModbusIoTConfig` object.

`GET` expects empty body.

`200` Return `ModbusIoTConfig` object.

`401` Unauthorized access due missing authentication, returns empty body

`403` Unauthorized access for the authenticated user, returns empty body

`500` Internal server error on unexpected error, returns empty body.

```
{
  ”operation_mode”: ”SNIFFER”,
  ”modbus_segment_name”: ”modbus-segment”
}
```

Listing 1: ModbusIoTConfig object example

`PATCH` expects a complete or partial `ModbusIoTConfig` object.

`202` The request has been received and accepted, return empty body.

`401` Unauthorized access due missing authentication, returns empty body

`403` Unauthorized access for the authenticated user, returns empty body

`500` Internal server error on unexpected error, returns empty body.

`DELETE` expects empty body.

`202` The request has been received and accepted, return empty body.

`401` Unauthorized access due missing authentication, returns empty body

`403` Unauthorized access for the authenticated user, returns empty body

`500` Internal server error on unexpected error, returns empty body.

/modbus-iot/sample Provides the latest received Modbus IoT RTU PDU.

`GET` expects empty body.

`200` Return `ModbusIoTSnifferTelemetry` object.

`401` Unauthorized access due missing authentication, returns empty body

`403` Unauthorized access for the authenticated user, returns empty body

`500` Internal server error on unexpected error, returns empty body.


```
{
  ”data”: [
    {
    ”timestamp”: {
      ”incarnation”: 17211,
      ”system_time”: 41161000000
    },
    ”data”: {
      ”modbus_frame”: {
      ”value”: [1,3,80,22,0,2,52,207],
      ”unit”: ”RAW”
      }
    }
    }
  ]
}
```


Listing 2: ModbusIoTSnifferTelemetry object example

6.2 Modbus IoT RTU Config

/modbus-iot/rtu/config Provides the Modbus IoT RTU interface config object.

`GET` expects empty body.

`200` Return a `ModbusIoTRTUConfig` object.

`401` Unauthorized access due missing authentication, returns empty body

`403` Unauthorized access for the authenticated user, returns empty body

`500` Internal server error on unexpected error, returns empty body.


```
{
  "baudrate": "B19200",
  "word_length": "WLEN8",
  "stop_bits": "STOP1",
  "parity": "PARITY_NONE",
  "response_timeout": 0.15
}
```


Listing 3: ModbusIoTRTUConfig object example

`PATCH` expects a complete or partial `ModbusIoTRTUConfig` object.

`202` The request has been received and accepted.

`401` Unauthorized access due missing authentication, returns empty body

`403` Unauthorized access for the authenticated user, returns empty body

`500` Internal server error on unexpected error, returns empty body.

`DELETE` expects empty body.

`202` The request has been received and accepted, return empty body.

`401` Unauthorized access due missing authentication, returns empty body

`403` Unauthorized access for the authenticated user, returns empty body

`500` Internal server error on unexpected error, returns empty body.

7 Integrated WebUI

periNODE Modbus offers two main interfaces: a web-based user interface for humans and a RESTful API for machines. These interfaces are available through an integrated HTTP server, which requires TLS for secure access.

Note: Access to the integrated HTTP server is only possible with TLS enabled.

Each periNODE Modbus smart component has a unique hostname, formed by combining the host prefix `periNODE` with its unique serial number. Example: `periNODE-sernm`, where `sernm` is the serial number. This unique identifier is also present on the product's label (refer to Section 9 for details).

By default, periNODE Modbus use the Zeroconf protocol multicast DNS (mDNS) and Link Local Multicast Name Resolution (LLMNR) to enable easy network discovery.

To access the web-based user interface, you can use mDNS-based name resolution. To access a periNODE Modbus device, you would use its mDNS name as follows:

```
https://periNODE-sernm.local/
```

or via an LLMNR based name resolution:

```
https://periNODE-sernm/
```

A RESTful API specification is available on each periNODE via the URI:

```
https://periNODE-sernm.local/api.html
```

7.1 Authenticity and Security Warnings

Users may encounter a security warning in environments where the **Perinet ECC Root CA** trust anchor is not installed. This certificate is essential for verifying periNODE Modbus smart components. It confirms that the component is genuinely made by an authorized manufacturer.

However, the **Perinet ECC Root CA** certificate is not automatically added to systems and requires manual installation. If this certificate is not present in the system, the authenticity of the smart component cannot be verified.

This situation occurs when a web client, like a standard web browser, attempts to establish a secure connection with the smart component. The web client first tries to authenticate the device. If the **Perinet ECC Root CA** certificate is missing, this authentication will fail, resulting in security warnings, as illustrated in Figures 11 (Edge browser) and 12 (Safari browser).

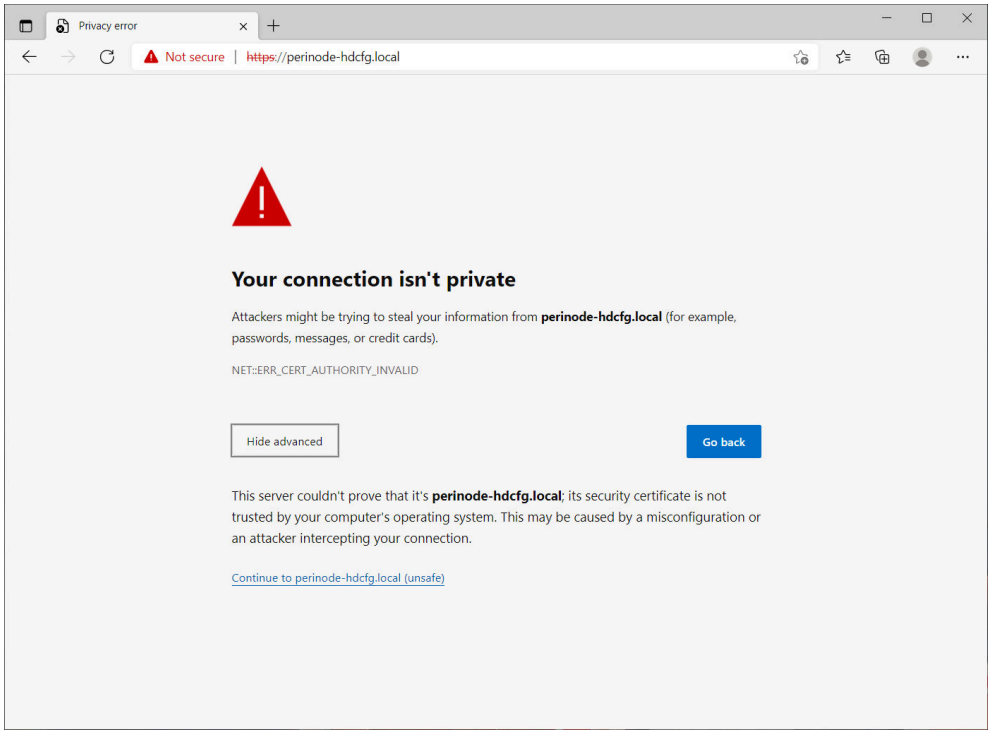


Figure 11: Edge browser security warning example

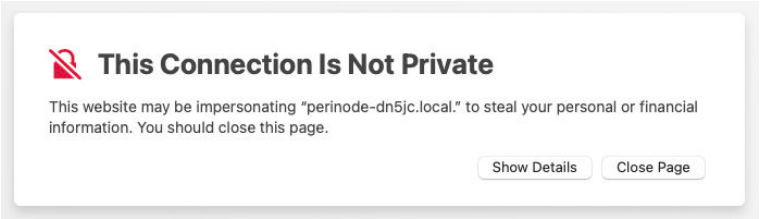


Figure 12: Safari browser security warning example

To address this issue, there are two options: either install the **Perinet ECC Root CA** trust anchor on the system, or choose to ignore the warning. However, bypassing the warning compromises system security.

Note: *Perinet* strongly advises installing the **Perinet ECC Root CA** in the system's list of trusted entities, instead of disregarding the security warning.

7.1.1 Add Perinet as a Trust Anchor

To install the **Perinet ECC Root CA** certificate, download it from https://docs.perinet.io. The installation process is straightforward: double-click the downloaded file and follow the on-screen instructions.

For comprehensive instructions on installing certificates across various operating systems and browsers, refer to the *Security Certificates Installation Guide* [15]. This guide assists in installing the necessary certificates for all *Perinet* Smart Components.

Once the **Perinet ECC Root CA** trust anchor is installed, you should be able to access the web user interface of any smart component without encountering security warnings. If issues persist, please reach out to our support team.

You can verify the installation by clicking on the *locker* icon in your web browser's URL bar. The subsequent figures demonstrate the expected certificate chain for reference.

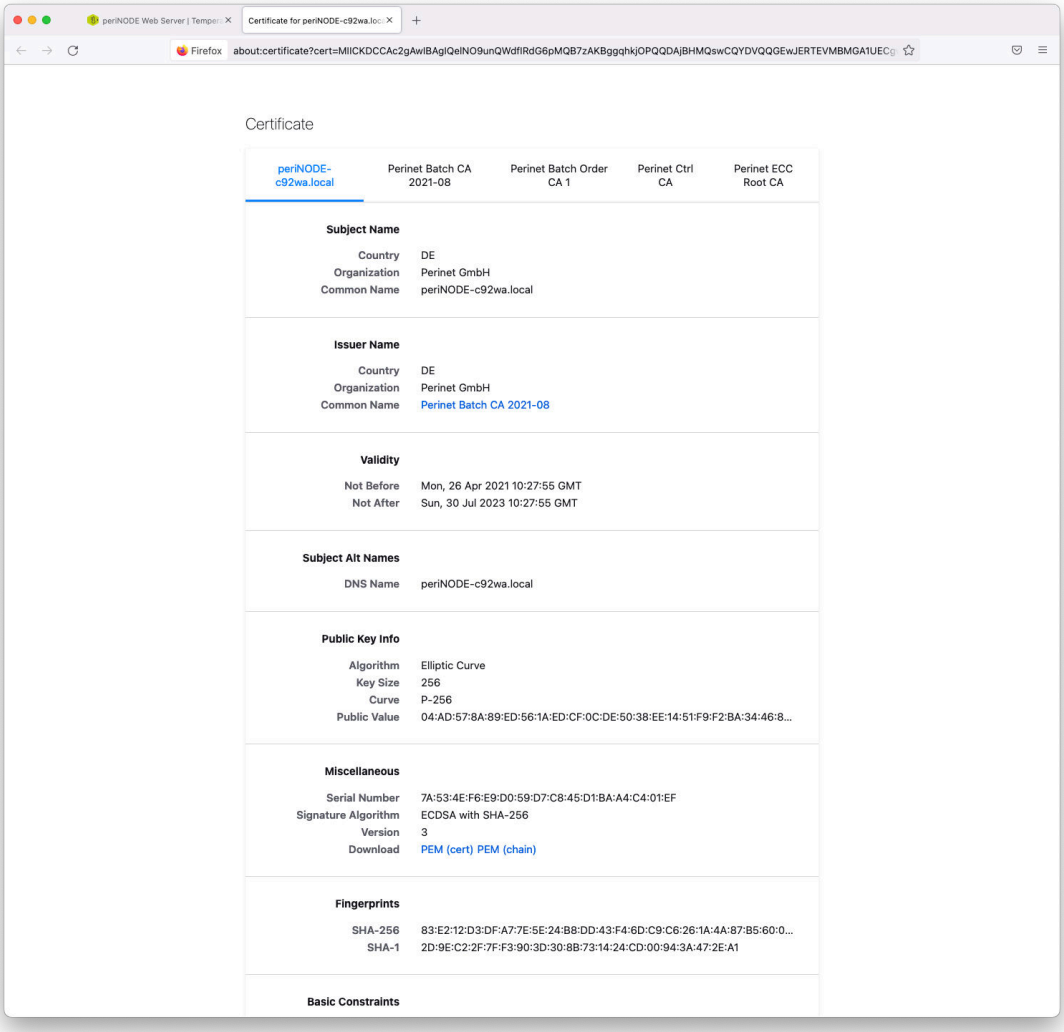


Figure 13: Firefox example security trusted chain

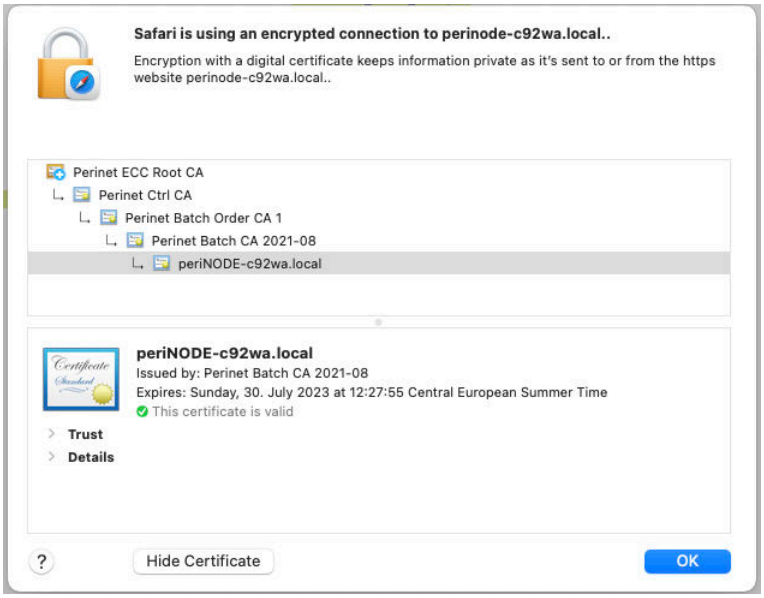


Figure 14: Safari example security trusted chain

Note: When using the RESTful API to access the Smart Component, the absence of the **Perinet ECC Root CA** trust anchor may impact command line tools. In this scenario, it's necessary to install the **Perinet ECC Root CA** trust anchor system-wide, rather than only in the web client (standard browser).

7.1.2 Proceed with Exception

As mentioned earlier, it's possible to bypass the security warnings if you choose not to add *Perinet* as a trusted entity. To do this, click on "Advanced" and confirm your intention to proceed. This action allows encrypted but unauthenticated communication.

When using the RESTful API to access a Smart Component, the absence of the **Perinet ECC Root CA** trust anchor can affect command line tools. However, there are ways to circumvent this. For instance, when using the **curl** tool, the `-k`, `--insecure` option allows proceeding without server authentication.

Note: *Perinet* strongly advises against ignoring security warnings. Without the certificate, the system is vulnerable to certain attacks, such as man-in-the-middle attacks.

7.2 Configuration

The firmware of periNODE Modbus offers two configuration methods: a machine-friendly RESTful API and an HTML-based graphical user interface (GUI). The GUI is designed for interactive use, while the RESTful API is better suited for automated configurations.

All configuration settings are saved persistently and will be reinstated during the startup process.

7.2.1 periNODE Modbus Home

The periNODE Modbus Smart Component features a user-friendly interface for interacting with Modbus RTU devices. The *Home screen* serves as the main page of the web user interface. An example is illustrated in Figure 15. To access the *Home screen*, use the following URL format: `https://periNODE-sernm.local/`.

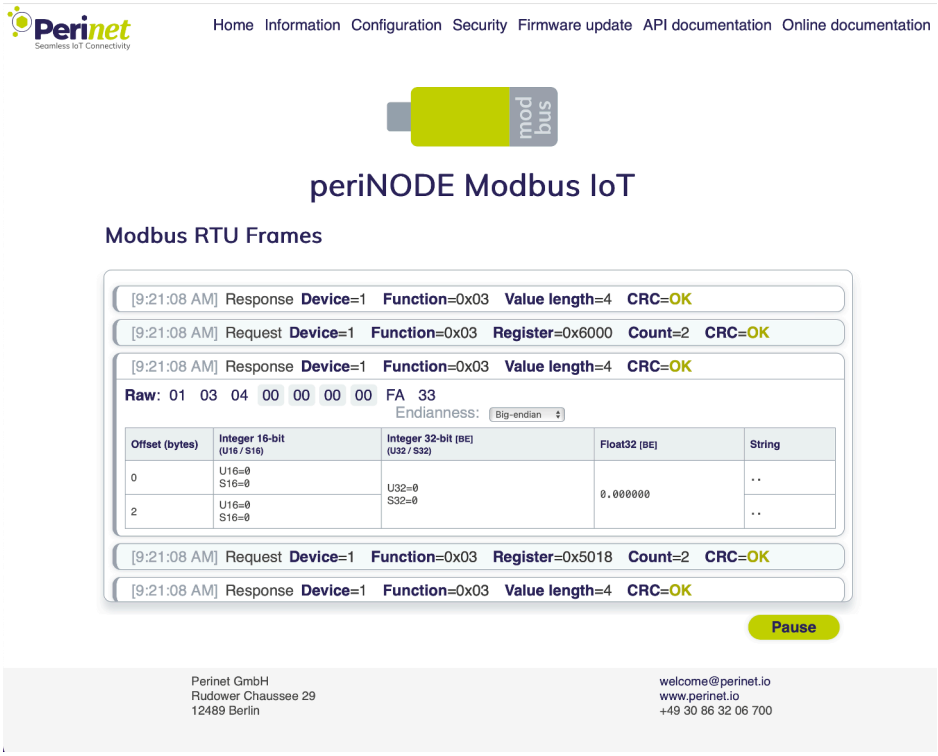


Figure 15: The Web Home Page of periNODE-modbus

7.2.2 periNODE Modbus GUI Configuration

The GUI (Graphical User Interface) of the periNODE Modbus enables configuration of several parameters. It is split into several sections:

Node

Figure 16: periNODE Modbus Node section of Configuration Web Page

The Node section consists of two parameters:

- *Application name*: The (***application_name***) is used to group IoT devices and software components, that together represent an application. It also sets the prefix for MQTT topics.
- *Element name*: The (***element_name***) allows you to label things within applications understandably, getting rid of clumsy structural labelling like IP-addresses or hostnames.

Operation

The Modbus Operation Mode *Sniffer* reads the Modbus RTU traffic and forwards the data to a MQTT Broker under the topic *application_name/modbus_segment_name*. The default Modbus Segment Name is *modbus-segment*.

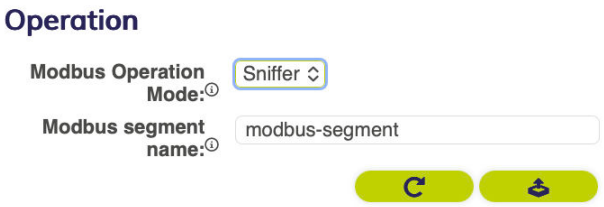


Figure 17: periNODE Modbus Operation section of Configuration Web Page

Modbus-RTU

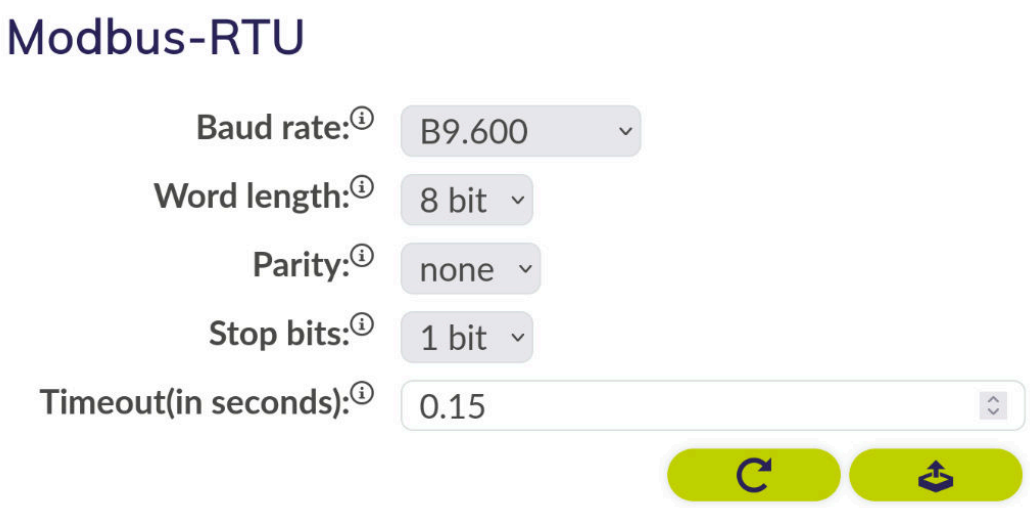


Figure 18: periNODE Modbus Modbus-RTU section of Configuration Web Page

The Peripheral section groups parameters of the Modbus-RTU connection. It consists of the following parameter:

- *Baud rate*: The baud rate of the RTU slave device.
- *Word length*: The number of bits per data word.
- *Parity*: The parity of the transmission.
- *Stop bits*: The number of stop bits.
- *Timeout (in seconds)*: The RTU slave response timeout in seconds.

7.2.3 MQTT Client

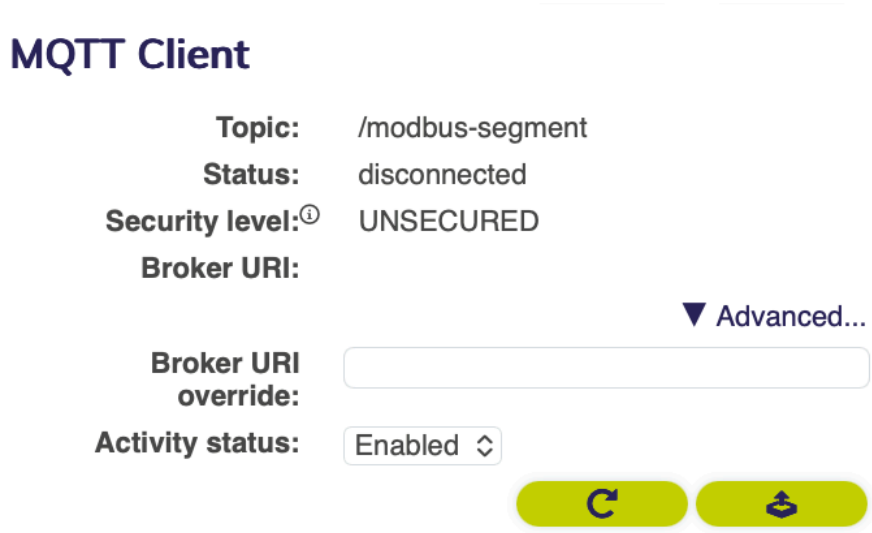


Figure 19: periNODE Modbus MQTT Client section of Configuration Web Page

- *Topic*: The MQTT topic where the Modbus RTU data is published. The topic is formed as *application_name/modbus_segment_name*.
- *Status*: The MQTT client connection status. The status presented can be: *disconnected*, *searching*, *resolving*, *connecting*, *connected* and *failed*.
- *Security Level*: The security level of the MQTT connection. UNSECURED (when disconnect), *ENCRYPTED*, *AUTHENTICATED* and *AUTHORIZED* (mTLS enabled).
- *Broker URI*: The URI of the MQTT broker. By default the periNODE Modbus connects to the MQTT broker within the same application.
- *Advanced/Broker URI override*: The list bring the MQTT brokers available in the network and the user can select one of them to override the Broker URI.
- *Advanced/Activity status*: The activity status (enabled/disabled) can be selected.

The payload message published to the MQTT broker is a JSON object containing the Modbus RTU data as provided by the RESTful API at the path `/modbus-iot/sample`. An example is shown in Listing 2.

Default Settings

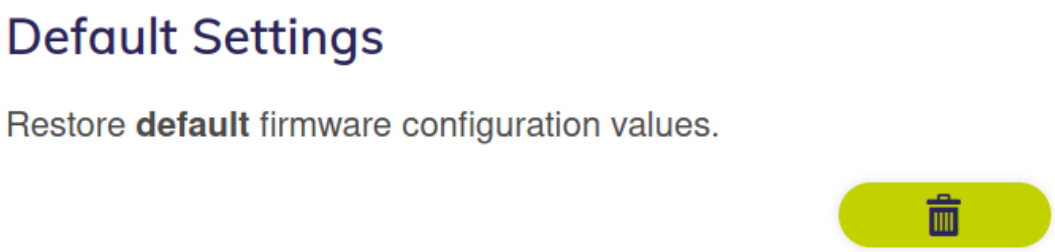


Figure 20: periNODE Modbus Default Settings section of Configuration Web Page

By clicking the wastebasket button all customized parameters are overwritten by its default values.

Reboot Firmware

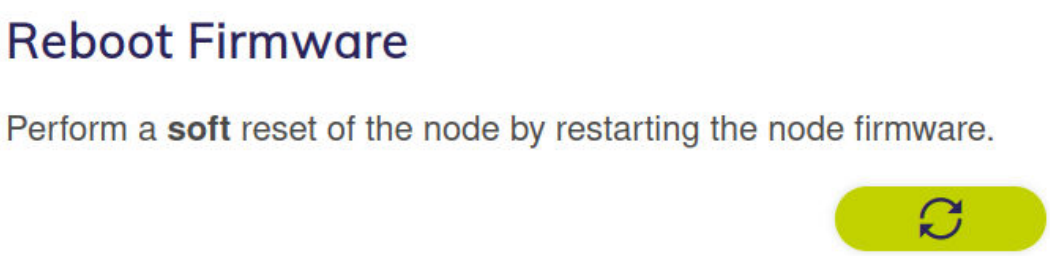


Figure 21: periNODE Modbus Reboot Firmware section of Configuration Web Page

By clicking the Reboot Firmware button a manual reboot of the periNODE Modbus is performed.

7.2.4 Security Configuration

Security settings for a periNODE can be modified using the RESTful API at `/security` or through a web interface accessible at *https://<hostname>.local/security.html*.

There are three certificates for configuring a periNODE's security:

- ***Host Certificate***: A unique certificate that validates the periNODE's identity to external clients like web browsers. It includes the hostname and must be signed by a trusted root CA, ensuring the periNODE's authenticity.
- ***Root Certificate***: Represents the trusted authorities in the periNODE system. It is essential for verifying external clients (e.g. web browsers) during connections, especially when mutual TLS (mTLS) is enabled.
- ***Client Certificate***: Utilized by the periNODE when connecting as a client to external servers, like MQTT brokers. This certificate authenticates the periNODE and defines its access level in these interactions, under the Role Based Access Control (RBAC) system.

For more detailed information on security concepts used for Perinet products, please refer to https://docs.perinet.io.

Mutual TLS and Authorization

A periNODE can be set up to use mutual TLS (mTLS) and Role Based Access Control (RBAC). With mTLS enabled, any remote client must authenticate itself using a client certificate to establish a connection. This certificate is checked against the periNODE's *Root Certificate*.

Note: Remote clients can be authenticated by the periNODE only if the *Root Certificate* is stored in the periNODE and the *Client Certificate* is signed by this same *Root Certificate*.

A user role must be included in the client certificate for the *periNODE*. This role is crucial for the Role Based Access Control (RBAC) system. The *periNODE* supports three roles: ***admin***, ***super***, and ***user***:

- ***admin***: This user has complete read/write access to the *periNODE*, with no limitations.
- ***super***: This user can access and modify all resources, except for */security* and */update*.
- ***user***: This user can only view resources and cannot modify any.

Note: The ***user*** role is not permitted to modify actuator states via the RESTful API.

Mutual TLS (mTLS) is not active by default. It will also be deactivated following a security or factory reset.

Note: By default, role-based access control for both the RESTful API and web user interface is *deactivated*.

For more detailed information on the client certificate, please refer to https://docs.perinet.io.

Web User Interface

The web user interface includes sections for inputting all three types of certificates: the *Host Certificate*, *Root Certificate*, and *Client Certificate*. For illustration, Figure 22 displays the input section specifically for the *Host Certificate*.

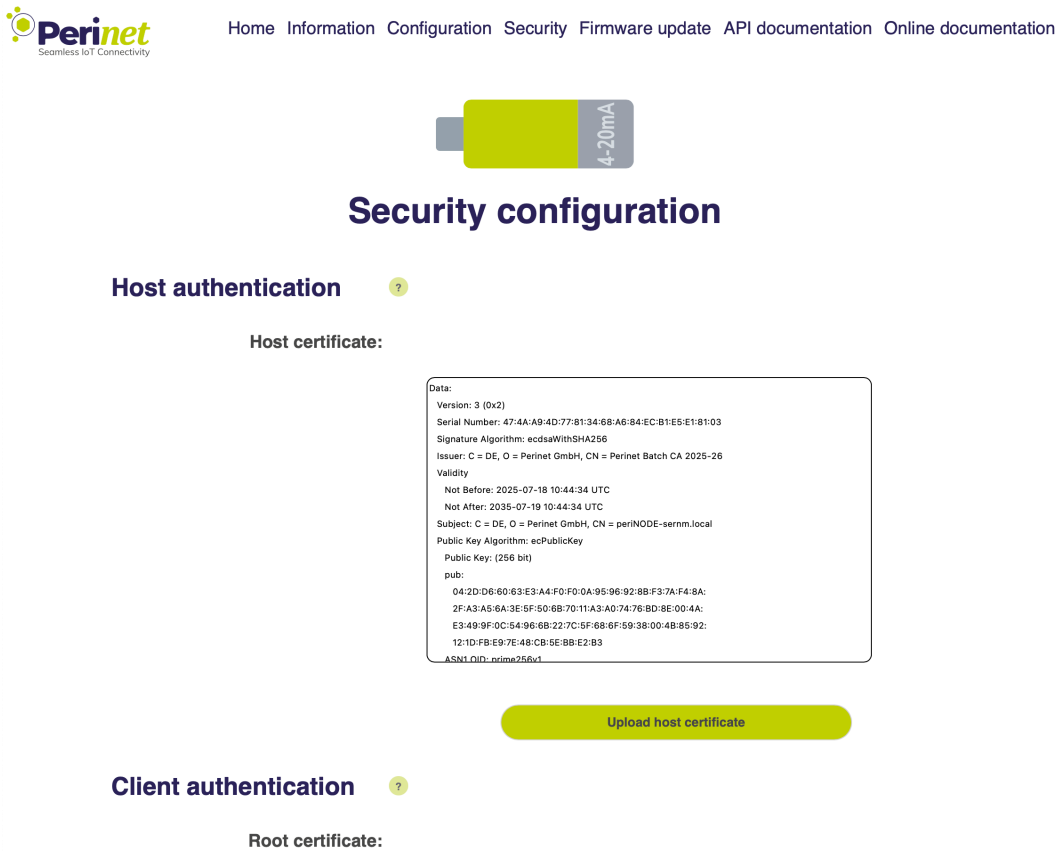


Figure 22: Security web user interface

The certificate displayed in the text area is the current one stored in the *periNODE*. This device accepts X.509 certificates encoded in PEM format (Base64 ASCII). Commonly, files with the extension `.pem` use this encoding, but it's also seen in `.crt`, `.cer`, and `.key` files.

For the *Host Certificate* and the *Client Certificate*, they should be uploaded along with their respective private keys concatenated at the end. Conversely, the *Root Certificate* should be uploaded without its private key.

For detailed guidance on generating certificates, visit https://docs.perinet.io.

Enforce mTLS Access

Activating the mTLS feature mandates that all remote clients authenticate themselves to the *periNODE* using a valid *Client Certificate*. This *Client Certificate* is verified against the *Root Certificate* already stored in the system.

Note: Before activating 'Enforce mTLS Access', ensure that a valid *Root Certificate* is already in place.

RESTful API Certificate Deployment

The RESTful API facilitates the management of all three types of certificates through dedicated resources. These resources are modifiable exclusively by the *admin* role and must be updated using an HTTP `PATCH` request. The specific endpoints for each certificate are as follows:

- *Host Certificate*: Accessed via `/security/host-cert`.
- *Root Certificate*: Accessed via `/security/root-cert`.
- *Client Certificate*: Accessed via `/security/client-cert`.

The subsequent section provides an example of updating the *Root Certificate* using the **curl** command-line tool.

```
curl --data-binary @<certificate-file> \
 -H "Content-Type: application/octet-stream" \
 -X PATCH https://periNODE-sernm.local/security/root-cert
# reboot device
curl -X PATCH https://periNODE-sernm.local/reboot
```

'Enforce mTLS Access' can be activated through the RESTful API, as demonstrated in the upcoming excerpt:

```
curl --data='{"enable_user_role":true}' \
  -X PATCH https://periNODE-sernm.local/security
# reboot device
curl -X PATCH https://periNODE-sernm.local/reboot
```

Reset Security Configuration

The **Reset security** button enables users to revert the security settings to their original factory defaults. This action will replace any user-configured Host and *Root Certificates* with the default ***Perinet ECC Root CA*** certificates. Additionally, any existing *Client Certificate* will be removed.

The procedure to reset the security configuration via the **curl** command-line tool is shown in the following code snippet:

```
curl -X PATCH https://periNODE-sernm.local/security/reset
```

If the *periNODE* becomes inaccessible, such as in cases where the *Client Certificate* is lost, please consult the Factory Reset procedure detailed in Section Section 7.2.6.

7.2.5 Firmware Update

Perinet may provide security and/or feature updates for periNODE Modbus over time. A firmware update shall be performed with the proper firmware update image artifact (e.g. matching variant).

Note: Inter-variant firmware updates are not supported by Perinet and may result in damaging the product.

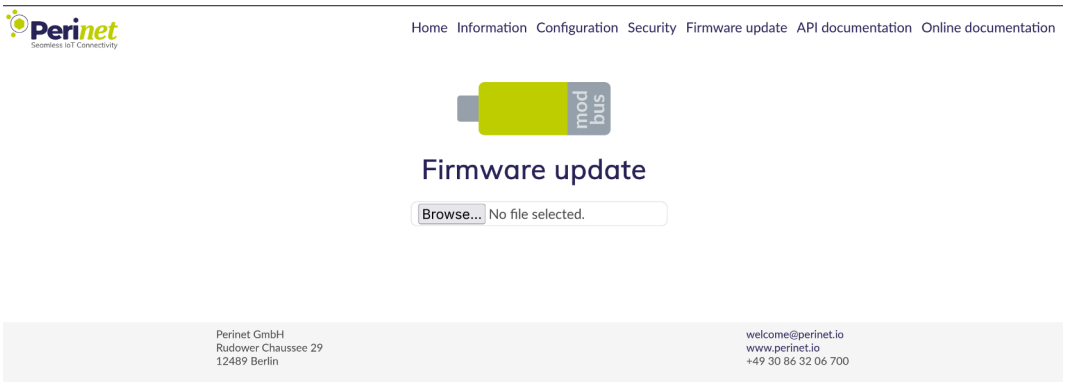


Figure 23: Firmware update

In order to keep the firmware of the device up to date, follow these steps:

1. Download the newest version of periNODE Modbus firmware https://downloads.perinet.io
2. Access the web page of the component through your browser: `https://periNODE-sernm.local/`
3. Click on the **Firmware update** link located in the header of the page.
4. On the update page, click on **Browse...** and select the new firmware image to be uploaded.
5. Finally click on the **Upload file** button to install the new firmware.

Note: After a successful firmware update, a reboot of the component to the newly installed firmware will only be automatically triggered when using the web user interface.

RESTful API

The RESTful API accepts an update image via a *PUT* request on the resource */update*. The following command line shows how to perform an update with the **curl** tool:

```
curl --data-binary @<filename> \
    -H "Content-Type: application/octet-stream" \
    -X PUT https://periNODE-sernm.local/update
# reboot into new firmware
curl -X PATCH https://periNODE-sernm.local/reboot
```

7.2.6 Factory Reset

In order to bring periNODE Modbus back into fresh production state a factory reset performs the following steps:

- Erase host certificate
- Reset all configuration values back to production state
- Reset firmware version to produced firmware version

A factory reset of a periNODE Modbus can be performed via the RESTful API remotely or physically by using a dedicated reset cable. The former solution might not be possible in some cases, e.g. where the client certificate has been lost and the access to the device is not possible anymore.

Factory Reset via Reset Cable

Connect the periNODE Modbus via the reset cable (see Figure 24) for at least 20 seconds to a powered network, e.g. a powered periSTART. Please ensure that the power supply of the network is active before connecting the product.

Figure 24: Perinet Reset Cable for Factory Reset

Factory Reset via RESTful API

When performing the factory reset via the RESTful API, a config as well as a security reset needs to be performed:

```
curl -X PATCH https://periNODE-sernm.local/config/reset
curl -X PATCH https://periNODE-sernm.local/security/reset
```

8 Factory Reset

Performing a Factory Reset on the periNODE Modbus clears the entire *Operational Storage Area*, returning the product to the *OEM Production State*. The reset process affects the following properties:

***Config Data* reset:** All persistent runtime configuration settings are reverted to their default firmware values. This includes network settings like `mqtt_broker_name` or `application_name`.

***Security Data* reset:** All security settings are reset to the default *OEM Security Data*. This disables mTLS and, consequently, client authentication (RBAC). Please refer to 7.2.4 for more detailed information.

***Update Firmware Image* invalidation:** The *Update Firmware Image* stored in the *Operational Storage Area* is invalidated. Upon the next boot, the periNODE Modbus will run the *OEM Firmware Image*, as deployed during production.

A *Factory Reset* can be initiated through the RESTful API (see Section 8.1) or with direct access to the periNODE Modbus as described in Section 8.2. While former method requires authorization the latter can be used without authorization, in cases where for example the admin mTLS certificate is unavailable.

Note: After performing a reset on the device, a firmware/software update is required to ensure proper application functionality. Please download the latest periNODE Modbus firmware available at https://downloads.perinet.io.

8.1 Software Factory Reset

A *Factory Reset* can be initiated through the RESTful API (see Listing 6). To access the API the device requires authorization to perform this action. The factory reset can be performed as demonstrated in the excerpt below (Listing 4).

```
# invoke factory reset
curl -X PATCH https://periNODE-sernm.local/reset

# changes are applied after the reboot of the device
curl -X PATCH https://periNODE-sernm.local/reboot
```

Listing 4: Invoke a *Factory Reset* via the RESTful API.

Note: Any modifications made to the *Security Data* and *Config Data* will only become effective after the device is rebooted. Given that all security settings are reset to their defaults during a factory reset, it is strongly advised to reconfigure the security settings post-reset for optimal security.

8.2 Hardware Factory Reset

A physical reset to *Factory Settings* of the periNODE Modbus can be performed with direct access to the device. This method allows to reset the device without the necessity of authorization from the device.

The foundational *libperiCORE* library facilitates the factory reset process. To physically initiate a factory reset, ensure that the periNODE Modbus is powered on (using `24VDC_IN`) but is not connected to any network through its network ports. The factory reset will automatically occur *20 seconds* after the device is powered up, returning the device to its original factory settings.

9 Product Marking and Packaging

The periNODE Modbus provides all its information in digital form, accessible via the RESTful interface Section 6. Because of size constraints, only limited information are provided with the product label as described below.

Since the periNODE Modbus will be shipped with a package size of 20pcs, a specific logistic label is specified in this section.

9.1 Product Marking

A unique label is added to each periNODE Modbus product during production. It has the dimensions 25mm x 21mm and the corner radius is 0.5mm. The containing information are shown in Figure 25 and explained in Table 7. The included optical code can be scanned from a smartphone for easy access to the configuration of the device.

(a) (b)

Figure 25: Visual marking of a periNODE Modbus (21x25mm r=0.5mm). Original size (left) and scaled (right) for information purposes.

Position	Description
1	Peripheral pinout specification
2	The unique hostname of the product (`https://periNODE-sernm.local`), used to access the RESTful API and integrated UI.
3	Product name
4	Manufacturing Part Number
5	Product batch number
6	CE-mark (conformity for Europe)
7	Two-dimensional Data Matrix (ECC200) code. Encodes the hostname `periNODE-sernm`, where `sernm` is the unique serial number of the product, which is used to generate a unique hostname.

Table 7: Properties encoded in the visual marking of the periNODE Modbus.

9.2 Electronic Product Marking

From the network interface, product identification parameters can be read via the RESTful API (see Section 6):


```
{
 ”manufacturer”: ”Perinet GmbH”,
 ”mpn”: ”PRN.000.018”,
 ”serial_number”: ”sernm”,
 ”product_version”: ”1”,
 ”product_variant”: ”periNODE Modbus”,
 ”batch_number”: ”2024-04”,
 ”product_name”: ”periNODE Modbus”,
 ”hostname”: ”periNODE-sernm”,
 ”mac_address”: ”76:2E:DB:02:3D:8B”,
}
```


Listing 5: The periNODE Modbus production info object received via the resource **/node/production**.

Note: The above listed parameters are provided as an example and do variate for different devices.

9.3 Product Packaging

periNODE Modbus is delivered in boxes (231x111x62 cm) containing 20 devices.

Figure 26: Packaging periNODE Modbus example

9.3.1 Package Labeling

The label is printed in a 2 × 4 inch format. The layout of the label is shown in Figure 27 and the meaning of the encoded information is explained in Table 8.

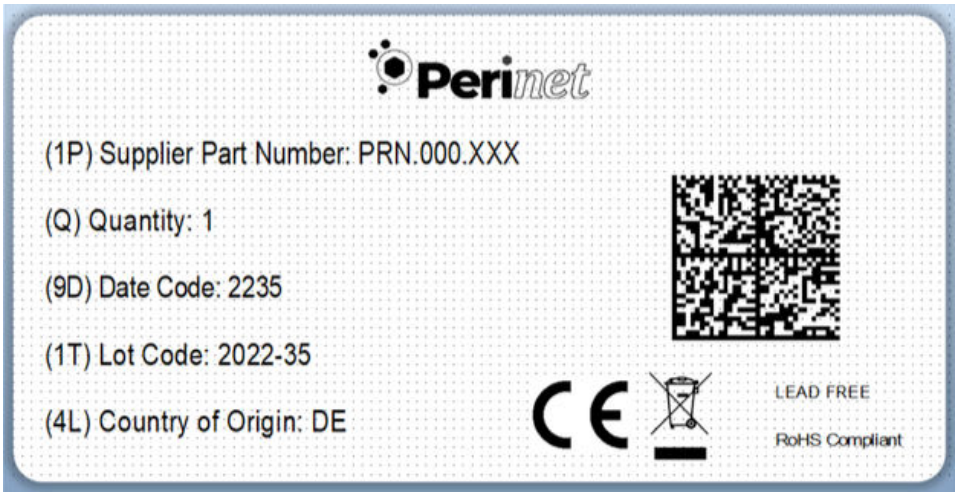


Figure 27: Smart components package layout

Identifier	Description
(1P)	Supplier Part Number: PRN.000.018
(Q)	Quantity
(9D)	Data Code: Format like *YYWW* where *YY* is the year and *WW* the calendar week.
(1T)	Lot Code: Format like YYYY-WW where *YYYY* is the year and *WW* the calendar week.
(4L)	Country of Origin: DE (two-letter uppercase country code)
	Type approval marking: FCC/IC, CE-mark and WEEE conformity for Europe.
	Two-dimensional Data Matrix (ECC200) code. All the parameters are encoded to better machine readability.

Table 8: Properties encoded in the visual marking of the periNODE Modbus.

10 Safety Instruction

This section contains information that you must observe for your personal safety and to avoid damage to property. The instructions for your personal safety are highlighted by a warning triangle, while information on damage to property alone is given without a warning triangle. Depending on the hazard level, the warnings are shown in decreasing order as follows.

Notation	Description
⚠ DANGER	means that death or serious injury will occur if the appropriate precautions are not taken.
⚠ WARNING	means that death or serious injury may occur if the appropriate precautions are not taken.
⚠ CAUTION	with warning triangle, means that minor injury may occur if the appropriate precautions are not taken.
CAUTION	without warning triangle, means that property damage may occur if the appropriate precautions are not taken.
ATTENTION	means that an undesirable result or condition may occur if the corresponding precaution is not taken.
INFO	information on needed measures to guard against system failures.

Table 9: Safety instructions

If several hazard levels occur, the warning notice for the highest level is always used.

If the warning triangle is used in a warning against personal injury, then a warning against property damage can also be added to the same warning.

10.1 Power Supply

10.1.1 ⚠ DANGER

Connection of the 24 V device only via safety extra-low voltage/protective extra-low voltage

The device is designed for operation with a directly connectable safety extra-low voltage (Safety Extra-Low Voltage, SELV) or Protective Extra-Low Voltage (PELV).

Failure to observe this warning may result in electric shock or damage to property.

Therefore, only safety extra-low voltages (SELV) according to IEC 60950-1 / EN 60950-1 / VDE 0805-1 or protective extra-low voltages (PELV) according to EN 50178 may be connected to the supply connections.

10.1.2 ATTENTION

Connecting the 24 V device to a safety extra-low voltage (SELV) turns it into a protective extra-low voltage (PELV). The connection point between ground and PE is established by the device according to the IEC 61851 standard.

10.2 Protection

10.2.1 CAUTION

Property damage

Property damage caused by electrostatic charging.

- Follow the measures necessary for the handling of components prone to electrostatic charging.

10.2.2 ⚠ WARNING

Unwanted heat generation or fire due to insufficient fuse protection

The internal fuses of the device are designed for device protection only. The system installer or operator is responsible for the necessary line protection.

10.3 Repair

Repairs are not permitted. Defective devices must be disposed of in accordance with environmental requirements.

10.3.1 CAUTION

Dangers due to unauthorized opening of the device

Unauthorized opening of the device can cause considerable damage to property or danger to the user.

10.3.2 CAUTION

Voiding of the manufacturer's warranty due to unauthorized modifications to the device

Modifications to the equipment are not permitted. Failure to comply will invalidate the manufacturer's warranty expires.

Only put undamaged components in operation.

10.3.3 INFO

Perinet expressly notes that devices to which this document pertains are unsuitable for use as part of or in connection with medical implants or as important components of life support systems, where failure could lead to severe injuries to people. The components used and the proven level of reliability don't meet the necessary requirements for such applications. In order to avoid damage to devices and equipment as well as injuries to or death of people, the user or application developer must implement suitable, well thought out measures to guard against system failures.

The rights of third parties may be violated through use in the transport market of devices which are described in this document. Perinet can provide no warranty whatsoever that the rights of third parties aren't violated through use of the devices. If you are planning on this sort of use, please contact Perinet in order to clarify any potential questions regarding patent or property rights.

10.4 Qualification of the Personnel

The devices may only be mounted, installed, commissioned and maintained by persons who are familiar with the handling of such devices and who are qualified for the described activities.

This qualification includes:

- Training regarding the installation, connection, earthing, commissioning and maintenance of electrical devices (EN 50 110-1/-2 / VDE 0105 part 100).
- Training regarding the current standards of electrical engineering and safety technology.
- First aid training.

Note: We reserve the right to make technical changes to the products and to the content of this document at any time without prior notification. Perinet does not accept any responsibility for possible errors or incompletions in this document. We reserve all the rights to this document and the topics and illustrations contained within it. Copying, disclosure to third parties or use of its content - even partially - is forbidden without the prior written consent of Perinet.

11 Further Documentation

Document Name	Description
periCORE Product Summary [9]	A product features summary documentation for the product *periCORE*.
periCORE Datasheet [5]	A detailed reference documentation of the product *periCORE*.
periCORE Development Kit Product Summary [6]	A product features summary documentation for the product *periCORE Development Kit*.
periCORE Development Kit Setup Application Note [7]	A setup guide for the product *periCORE Development Kit*. The starting point when you are new to the product which describes how to quickly set up a development environment for firmware development for a periCORE based product.
periCORE Development Kit User Guide [8]	A guide and reference documentation for the product *periCORE Development Kit*.

Document Name	Description
periLINE Product Summary [10]	A product features summary documentation for the product *periLINE*.
periNODE 0-10V Product Summary [11]	A product features summary documentation for the product *periNODE 0-10V*.
periNODE GPIO Product Summary [12]	A product features summary documentation for the product *periNODE GPIO*.
periSWITCH 3-port Product Summary [14]	A product features summary documentation for the product *periSWITCH 3-port*.
periSTART Standard Product Summary [13]	A product features summary documentation for the product *periSTART standard*.
Smart Components Datasheet [16]	A detailed reference documentation of the Smart Components products (*periLINE*, *periNODE 0-10V*, *periNODE Pt100*, *periNODE GPIO*, *periSWITCH 3-port*, *periSTART standard*).

12 Ordering Information

Ordering Code	Product Name	Description
PRN.000.018	periNODE Modbus	Single Pair Ethernet Modbus RTU device adapter
Companion Products		
PRN.000.008	periSWITCH 3-port	3-port hybrid SPE switch
PRN.000.003	periSTART standard	Media converter from SPE to standard Ethernet
PRN.000.017	periLINE 0.2m	Hybrid SPE cable with M8 connectors
PRN.000.025	periMICA	Open modular edge computer
PRN.000.020	periCORE Development Kit	Full featured firmware development setup

13 Contact & Support

For customer support, please call us at **+49 30 863 206 701** or send an e-mail to *support@perinet.io*.

For complete contact information visit us at www.perinet.io

A List of Figures

B List of Listings

C List of Tables

D Glossary

100BASE-T1 A Ethernet Standard where two endpoints are connected by a single twisted pair cable. It is one of the so-called Single Pair Ethernet (SPE) standards. It operates in full-duplex with a data rate of 100 MBit per second. Furthermore, it uses PAM-3 modulation with a voltage level from -1 to +1V, differentially on the two wires. 8

API Application Programming Interface. 17, 20, 40

DNS-SD DNS Service Discovery [2] is a way of using standard DNS programming interfaces, servers and packet formats to browse the network for services. 6

full-duplex In a full-duplex system, both parties can communicate with each other simultaneously. Transmitting or receiving of information is operated independently contrary to a half-duplex scheme.. 7

IoT Internet of Things. 25

LLMNR The Link-Local Multicast Name Resolution is a protocol based on the Domain Name System packet format that allows IPv6 hosts to perform name resolution for hosts on the same local link. 20

mDNS multicast Domain Name Service [3], a protocol that implements a local distributed name resolving mechanism. 6, 20

Modbus Modbus Protocol is a messaging structure, widely used to establish master-slave communication between devices. 7, 14, 15

MQTT Message Queuing Telemetry Transport is a lightweight, publish-subscribe based network protocol that transports messages between devices. 25

mTLS Mutual TLS extends the TLS protocol by requiring clients to pass certificates, allowing to provide authorization mechanisms of Application services. 6, 9, 11, 37

PLC Programmable Logic Controller. 7

RBAC Role Based Access Control. 37

REST REpresentational State Transfer, a web API style. 20, 23, 24, 33, 35, 37

SCADA supervisory control and data acquisition is a control system architecture comprising computers, networked data communications and graphical user interfaces for high-level supervision of machines and processes.. 7

SPE Single Pair Ethernet. 7, 8, 14

TLS Transport Layer Security Protocol. Used by Application Protocols like MQTT or HTTP to allow secure data transfer. 20

E References

[1] Telecommunications Industry Association. *Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint Systems*. TIA TIA-485-A. https://store.accuristech.com/standards/tia-tia-485-a. Telecommunications Industry Association, Mar. 1998. URL: https://store.accuristech.com/standards/tia-tia-485-a.

[2] S. Cheshire and M. Krochmal. *DNS-Based Service Discovery*. RFC 6763. http://www.rfc-editor.org/rfc/rfc6763.txt. RFC Editor, Feb. 2013. URL: http://www.rfc-editor.org/rfc/rfc6763.txt.

[3] S. Cheshire and M. Krochmal. *Multicast DNS*. RFC 6762. http://www.rfc-editor.org/rfc/rfc6762.txt. RFC Editor, Feb. 2013. URL: http://www.rfc-editor.org/rfc/rfc6762.txt.

[4] "Connectors for electrical and electronic components - Product requirements - Part 6: Connectors - Detail specification for 2-way and 4-way (data/power), shielded, free and fixed connectors for transmission capability and power supply capability with frequencies up to 600 MHz". In: *IEC 63171-6* (2021). URL: https://webstore.iec.ch/publication/68051.

[5] Perinet GmbH. periCORE Datasheet. PRN.100.375. https://docs.perinet.io/PRN100375-periCOREDatasheet.pdf.

[6] Perinet GmbH. periCORE Development Kit Product Summary. PRN.100.546. https://docs.perinet.io/PRN100546-periCOREDevelopmentKitProductSummary.pdf.

[7] Perinet GmbH. periCORE Development Kit Setup Application Note. PRN.100.376. https://docs.perinet.io/PRN100376-periCOREDevelopmentKitSetupApplicationNote.pdf.

[8] Perinet GmbH. periCORE Development Kit User Guide. PRN.100.378. https://docs.perinet.io/PRN100378-periCOREDevelopmentKitUserGuide.pdf.

[9] Perinet GmbH. periCORE Product Summary. PRN.100.301. https://docs.perinet.io/PRN100301-periCOREProductSummary.pdf.

[10] Perinet GmbH. periLINE Product Summary. PRN.100.386. https://docs.perinet.io/PRN100386-periLINEProductSummary.pdf.

[11] Perinet GmbH. periNODE 0-10V Product Summary. PRN.100.380. https://docs.perinet.io/PRN100380-periNODE0-10VProductSummary.pdf.

[12] Perinet GmbH. periNODE GPIO Product Summary. PRN.100.382. https://docs.perinet.io/PRN100382-periNODEGPIOProductSummary.pdf.

[13] Perinet GmbH. periSTART Standard Product Summary. PRN.100.383. https://docs.perinet.io/PRN100383-periSTARTstandardProductSummary.pdf.

[14] Perinet GmbH. periSWITCH 3-port Product Summary. PRN.100.385. https://docs.perinet.io/PRN100385-periSWITCH3-portProductSummary.pdf.

[15] Perinet GmbH. Security Certificates Installation Guide. PRN.100.447. https://docs.perinet.io/PRN100447-SecurityCertificatesInstallationGuide.pdf.

[16] Perinet GmbH. Smart Components Datasheet. PRN.100.387. https://docs.perinet.io/PRN100387-SmartComponentsDatasheet.pdf.

[17] "ISO/IEC/IEEE International Standard - Part 3: Standard for Ethernet - Amendment 1: Physical Layer Specifications and Management Parameters for 100 Mb/s Operation over a Single Balanced Twisted Pair Cable (100BASE-T1)". In: *ISO/IEC/IEEE 8802-3:2017/Amd 1:2017(E)* (2018), pp. 1–92. DOI: 10.1109/IEEESTD.2018.8310988.

[18] The Modbus Organization. MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b3. https://www.modbus.org. https : / / www . modbus . org / docs / Modbus _ Application_Protocol_V1_1b3.pdf.
[19] The Modbus Organization. MODBUS over Serial Line - Specification and Implementation Guide v1.02. https://www.modbus.org. https://www.modbus.org/docs/Modbus_over_serial_line_V1_02.pdf.
[20] Karsten Walther. “INTERNET-OF-THINGS MODULE”. European pat. 4038350. Perinet GmbH. Sept. 30, 2019. URL: http://v3.espacenet.com/textdoc?IDX=EP4038350.

F Revision History

Revision	Date	Author(s)	Description
1	2025-12-12	shoe, nsav, mcet	initial release