

Perinet Smart Components

Datasheet

September 14, 2021



Abstract

This technical datasheet describes the Perinet (I)IoT *Smart Components* intended to be used for connecting analogue as well as digital sensors and actuators to the Local Area Network via Single Pair Ethernet (SPE).

Contents

1	Introduction	4
1.1	Product Features	5
2	Perinet Smart Components Portfolio	6
2.1	periNODE Smart Adapter	6
2.1.1	periNODE 0-10V	7
2.1.2	periNODE Pt100	11
2.1.3	periNODE GPIO	16
2.2	periSTART Media Converter	21
2.2.1	periSTART standard	22
2.3	periSWITCH Multi-Port Switch	26
2.3.1	periSWITCH 3-port	26
2.4	periLINE Hybrid SPE Cable	31
2.5	Perinet's Supported Network Topologies	35
2.5.1	Point to Point (PtP) - the simplest use case	35
2.5.2	Point to Multi-Point (PtMP)	36
3	Smart Components Software Interface	37
3.1	Authenticity and Security Warnings	37
3.1.1	Add Perinet as a Trust Anchor	38
3.1.2	Proceed with Exception	40
3.2	Configuration	41
3.2.1	periNODE Home	41
3.2.2	periNODE GUI Configuration	42
3.2.3	periNODE RESTful API Configuration Information	44
3.2.4	periNODE RESTful API Sample Information	46
3.3	Security Configuration	49
3.3.1	Mutual TLS and Authorization	49
3.3.2	Web User Interface	50
3.3.3	RESTful API certificate deployment	52
3.3.4	Reset Security Configuration	53
3.4	MQTT	54
3.5	Secure MQTT	55
3.6	Firmware Update	57
3.6.1	RESTful API	58
3.7	Factory Reset	59
4	Supported Web Browsers	60

5	Qualifications and Approvals	61
5.1	CE Compliance	61
5.1.1	Electromagnetic Compatibility (EMC)	61
5.1.2	Environmental Compatibility	61
5.1.3	RoHS and REACH	62
5.1.4	Waste from Electrical and Electronic Equipment (WEEE)	63
6	Installation	64
7	Labeling and Ordering Information	66
7.1	Product Marking	66
7.2	Data Matrix Code	67
7.3	Ordering Information	68
7.4	Packaging	68
8	Contact & Support	69
9	Safety Instructions	70
9.1	Power Supply	71
9.2	Protection	71
9.3	Repair	72
9.4	Qualification of the Personnel	73
A	Glossary	74
B	Smart Components API Specification	76

1 Introduction

Perinet’s Smart Components allow direct communication between single sensors or actuators and IT systems such as Manufacturing Execution Systems (MES), IoT platforms, factory or company IT and much more. Those IT based systems are communicating already completely over IP protocols and add value in digitized business processes. This seamless communication by IP across all levels is a big benefit, since it simplifies the overall system. The simplification, in turn, leads to a reduction in costs as expensive intermediate systems and experts that take care of installation and maintenance are no longer needed.

Figure 1 shows how the Perinet Smart Components can be integrated into the IIoT environment. Sensors and actuators can be connected to a hybrid Single Pair Ethernet segment using a periNODE. Multiple sensors can be connected to the network using a periSWITCH. Finally the periSTART bridges the Single Pair Ethernet segment with the existing IT infrastructure and is the central power feed to the Hybrid Single Pair Ethernet segment. Thus, Perinet Smart Components can be used to connect all your sensors and actuators to your IT infrastructure, no further intermediate converters or other parts are needed.

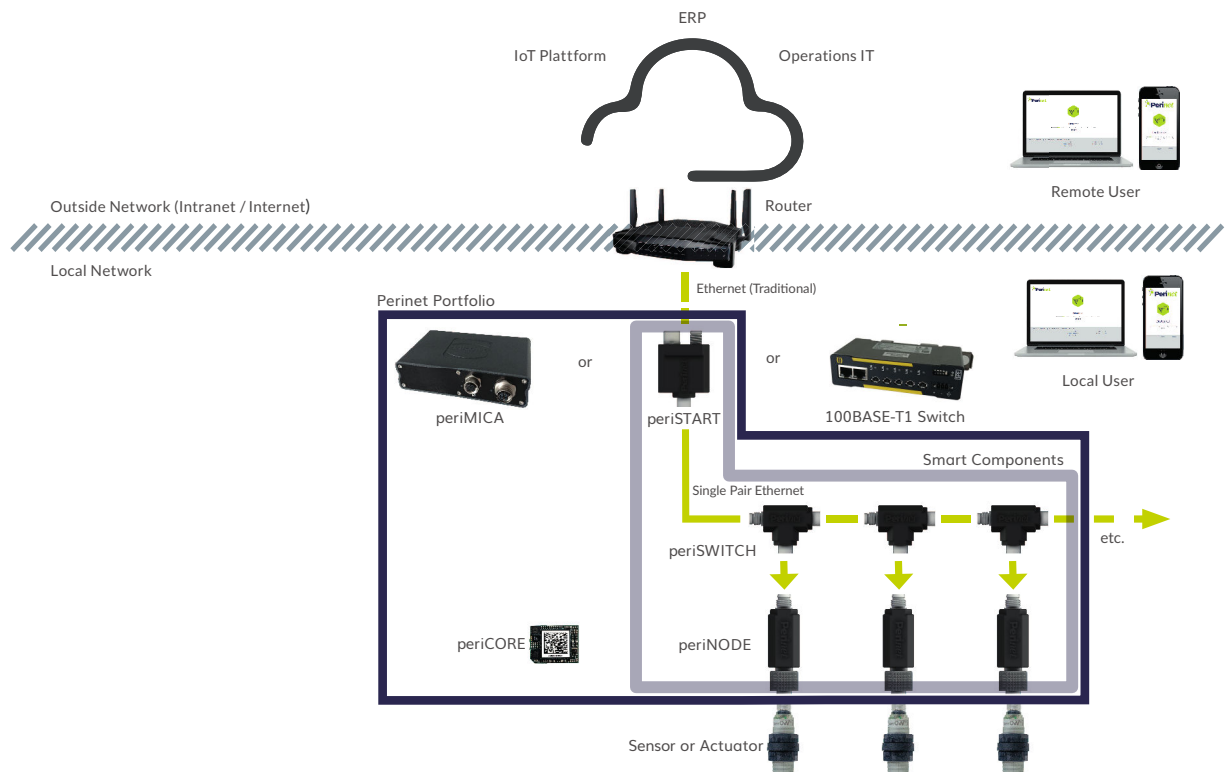


Figure 1: Perinet IP-based communication architecture

A periNODE turns a passive sensor or actuator to an active, intelligent one. Now the sensor provides information instead of raw data, which means unit based compensated and error corrected values instead of raw values are provided. Please note that the fact that the transmission is now digital also means that the quality of the data is no longer affected by the length of the wires. Even more, the data can be enriched with metadata such as timestamp, point of measurement, measurement method and so on. Thus, communicated data is self descriptive and perfect for use in higher level IT systems. Last but not least, the sensor can now send data when an event occurs, instead of being polled, which can significantly reduce the used network bandwidth.

While the ability of ethernet communication for a single sensor is impressive in itself, usability is also a big focus for Perinet. Therefore the Smart Components feature Zeroconf mechanism, which allows them to be used independently from network settings, without any configuration. Also, a state-of-the-art network security mechanism based on certificates is provided. For ease of use, this mechanism can also be managed automatically.

1.1 Product Features

- IPv6 based connectivity
- IT based network protocols TCP/UDP
- Security through end-to-end encryption (E2EE) with TLS
 - Role Based Access Control (RBAC)
 - Incorporable into Public Key Infrastructures (PKI)
 - Confidentiality through TLS 1.2
- HTTPS-Server hosting a web-GUI with sensor values and configuration page
- RESTful API for access to sensor values and configuration
- Publish/Subscribe to MQTT[s]
- Zeroconf networking
 - DNS service discovery
 - multi-cast DNS (mDNS) based unique periNODE-hostname resolution
- A *single* cable for power supply and data lines (hybrid Single Pair Ethernet)
- Extendable firmware architecture

2 Perinet Smart Components Portfolio

2.1 periNODE Smart Adapter

The periNODE smart adapter turns analogue or already digital, passive devices (such as sensors and actuators) into smart, active network participants. Its plug & play functionality aids an easy, fast and flexible deployment in the field, whether in industrial or commercial environments.

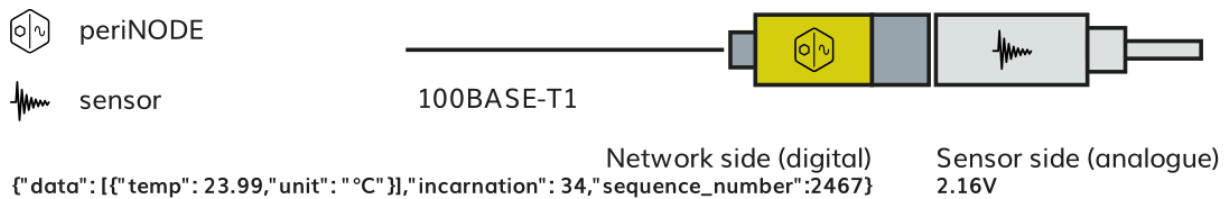


Figure 2: Semantic metrics of the periNODE

A periNODE transforms raw sensor signals into unit-based values and delivers that data including physical unit in digital format via Ethernet to any IT-system or data hub for further processing and analytics (see Figure 2). The smart adapter adds security to the sensor network through authenticated end-to-end data encryption. Furthermore, periNODE permits the configuration of data communication intervals to meet user-specific demand, thereby reducing network traffic and redundant data collection.

Note: *The periNODE smart adapter has not been designed for hard real-time nor safety-critical applications.*

Note: *A periNODE shall be connected to the sensor or actuator directly or with a cable that is less than 3m long.*

2.1.1 periNODE 0-10V



Figure 3: periNODE 0-10V

This periNODE smart adapter connects directly to sensors with 0-10V signal output and M12 A-coded 4-pin connectors:

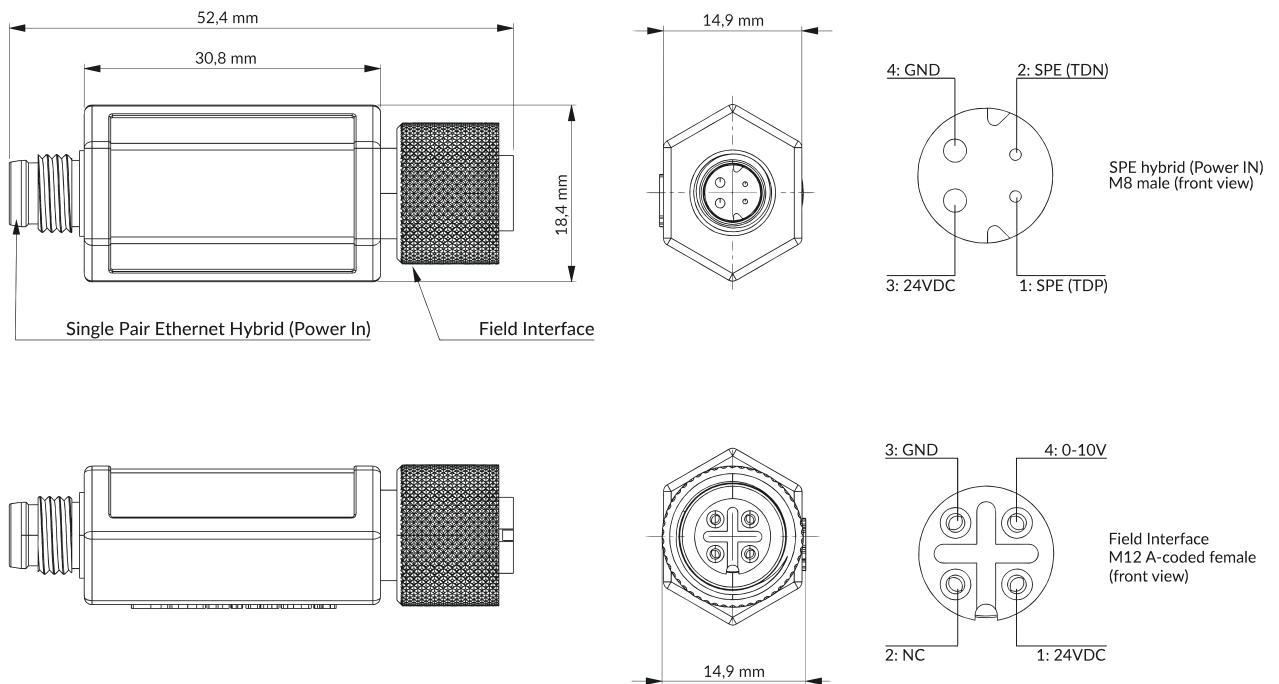


Figure 4: periNODE 0-10V dimensional drawing & pinout

periNODE 0-10V interfaces:

Single Pair Ethernet Hybrid Interface (Network side)	
Connector type	M8 male connector according to IEC 63171-6:2020 (style 6P-M8C without shielding)
Communication	100BASE-T1 Single Pair Ethernet (IEEE 802.3bw)
Power	24VDC input
Field Interface (Sensor side)	
Connector type	M12 A-coded 4-pin female connector with free rotating screw nut for direct sensor attachment
Signal	0-10V analogue input
Power	24VDC output, 100mA max

Table 1: periNODE 0-10V interfaces

The default sample rate for sensor values is 5 SPS (samples per second) and can be configured to a value of maximum 10 SPS.

Supported sensors:

- i.a. Micro Detectors ultrasonic sensor UK6A/H1-0EUL

periNODE 0-10V mechanical specifications:

Mechanical Specification			
Housing	Dimensions	Length	52.4 mm
		Width	16.5 mm
		Depth	14.7 mm
		Weight	19 grams
	Material		Thermoplastic material
	Protection		IP67

Table 2: periNODE 0-10V mechanical specifications

periNODE 0-10V electrical specifications:

Electrical Specifications			
Network Interface	Connector	Type	M8 H coded male
		Standard	Similar to IEC 63171-6:2020 Style 6P-M8C ("Plug")
		Mating	Mates with periLINE and IEC 63171-6:2020 Style 6J-M8C ("Jack")
		Pin 1	TDP
		Pin 2	TDN
		Pin 3	24V DC Input
		Pin 4	GND
	Communication	Type	SPE
		Physical Layer	100BASE-T1
		Standard	IEEE 802.3bw
	Power	Voltage	24VDC
		Range	±10%
	Consumption	Average	0.5 W
	Sensor Interface	Connector	Kind
Type			M12 A coded 4pin female
Signal		Pin 1	24VDC out
		Pin 2	do not connect
		Pin 3	GND
		Pin 4	0..10V input
		Input Current	-10mA ... + 10mA
		Resolution	<1mV
Sample Rate	typical: 5 SPS		

Table 3: periNODE 0-10V electrical specifications

Absolute maximum ratings:

Interface	Rating
Supply voltage	-40V...+40V
100BASE-T1	-40V...+40V
0-10V	-2.7V...+40V

Table 4: periNODE 0-10V absolute maximum ratings

Recommended Environment Conditions:

Operating conditions	-40°C...+70°C
Storage conditions	-45°C...+85°C

Table 5: Recommended environment conditions

Reverse Polarity Protection:

This smart component implements protection against polarity inversion for 24V DC input (Pin 3) and GDN (Pin 4), as well as TDP (Pin 1) and TDN (Pin 2) for the network interface.

Short Circuit limitation:

A protection mechanism against shorting the supply voltage to ground on the sensor or actuator interface might damage the device when current limitation is not applied. The current limitation of the power supply circuits is specified as max. 2A.

2.1.2 periNODE Pt100



Figure 5: periNODE Pt100

This periNODE smart adapter can be directly connected to Pt100 temperature sensors. The Pt100 sensor must have a 3-wire signal output and an M12 A-coded 4-pin connector:

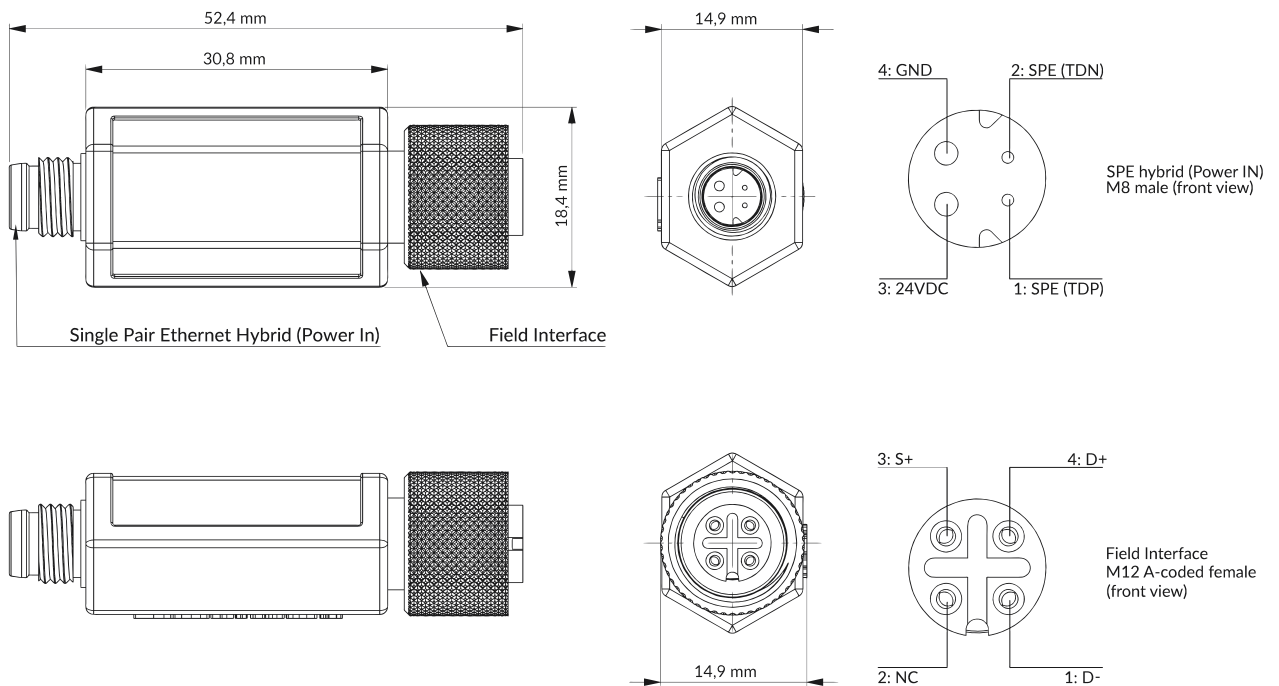


Figure 6: periNODE Pt100 dimensional drawing & pinout

The periNODE Pt100 is designed to measure temperature by applying 3-wire-Pt100 temperature sensor. It uses the ratiometric 3-wire RTD measurement method. This method is robust against failures caused by the connection between temperature detector (the actual Pt100 resistor) and the analogue front end in periNODE Pt100. The digitalization happens via Delta-Sigma-ADC. Afterwards the raw data is adjusted by a 4th order parabola into the final temperature value.

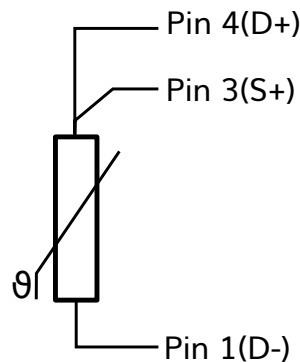


Figure 7: periNODE Pt100 3-wire connection

periNODE Pt100 interfaces:

Single Pair Ethernet Hybrid Interface (Network side)	
Connector type	M8 male connector according to IEC 63171-6:2020 (style 6P-M8C without shielding)
Communication	100BASE-T1 Single Pair Ethernet (IEEE 802.3bw)
Power	24VDC input
Field Interface (Sensor side)	
Connector type	M12 A-coded 4-pin female connector with free rotating screw nut for direct sensor attachment
Signal	Pt100 3-wire
Power	24VDC output

Table 6: periNODE Pt100 interfaces

Supported sensors:

- i.a. WIKA miniature resistance thermometer TR33-Z-P3

periNODE Pt100 mechanical specifications:

Mechanical Specification			
Housing	Dimensions	Length	52.4 mm
		Width	16.6 mm
		Depth	14.7 mm
		Weight	19 grams
	Material		Thermoplastic material
Protection		IP67	

Table 7: periNODE Pt100 mechanical specifications

periNODE Pt100 electrical specifications:

Electrical Specifications			
Network Interface	Connector	Type	M8 H coded male
		Standard	Similar to IEC 63171-6:2020 Style 6P-M8C ("Plug")
		Mating	Mates with periLINE and IEC 63171-6:2020 Style 6J-M8C ("Jack")
		Pin 1	TDP
		Pin 2	TDN
		Pin 3	24V DC Input
		Pin 4	GND
	Communication	Type	SPE
		Physical Layer	100BASE-T1
		Standard	IEEE 802.3bw
	Power	Voltage	24VDC
		Range	±10%
	Consumption	Average	0.5 W

Electrical Specifications			
Sensor Interface	Connector	Kind	Circular connector
		Type	M12 A coded 4pin female
	Signal	Pin 1	D-
		Pin 2	do not connect
		Pin 3	S+
		Pin 4	D+
		Current	10 μ A ... 1.5mA
		Resolution	0.035 K
Sample Rate	typical: 1 SPS (firmware)		

Table 8: periNODE Pt100 electrical specifications

The default sample rate for sensor values is 1 SPS (samples per second) and can be configured to a value of maximum 10 SPS.

Absolute maximum ratings:

Interface	Rating
Supply voltage	-40V...+40V
100BASE-T1	-40V...+40V
Pt100 Interface	-0.3V...+3.60V

Table 9: periNODE Pt100 absolute maximum ratings

Recommended Environment Conditions:

Operating conditions	-40°C...+70°C
Storage conditions	-45°C...+85°C

Table 10: Recommended environment conditions

Reverse Polarity Protection:

This smart component implements protection against polarity inversion for 24V DC input (Pin 3) and GDN (Pin 4), as well as TDP (Pin 1) and TDN (Pin 2) for the network interface.

Short Circuit limitation:

A protection mechanism against shorting the supply voltage to ground on the sensor or actuator interface might damage the device when current limitation is not applied. The current limitation of the power supply circuits is specified as max. 2A.

2.1.3 periNODE GPIO



Figure 8: periNODE GPIO

This periNODE smart adapter has two digital input/output channels with 24 V logic level. It can be connected directly to sensors with a digital 24V output or to actuators with a digital 24V input. periNODE GPIO comes with a pigtail cable outlet on the sensor or actuator side ($4 \times 0.25 \text{ mm}^2$). It can rather be assembled with a connector of choice or be connected to a clamp terminal.

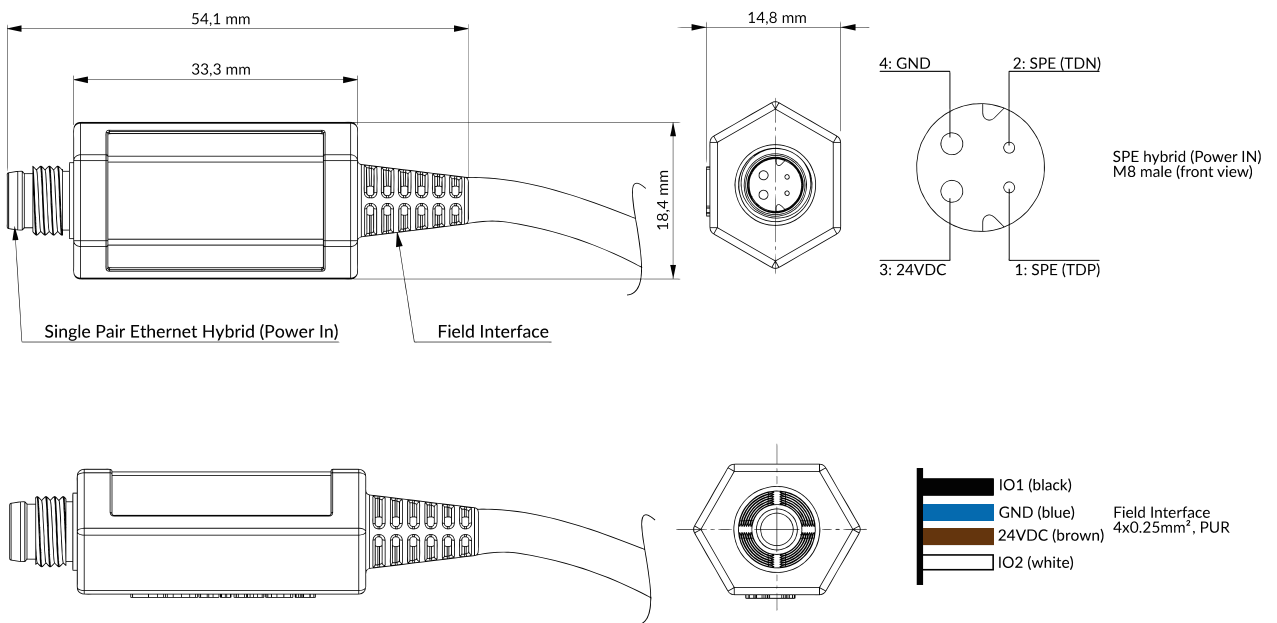


Figure 9: periNODE GPIO dimensional drawing & pinout

periNODE GPIO interfaces:

Single Pair Ethernet Hybrid Interface (Network side)	
Connector type	M8 male connector according to IEC 63171-6:2020 (style 6P-M8C without shielding)
Communication	100BASE-T1 Single Pair Ethernet (IEEE 802.3bw)
Power	24VDC input
Field Interface (Sensor or Actuator side)	
Connector type	4x0.25mm ² , PUR, open-ended cable
Signal	24V digital input or output
Power	24VDC output

Table 11: periNODE GPIO interfaces

The periNODE GPIO generates events only for the ports which have been configured as input.

The default sample rate for a GPIO input port is 0 SPS (samples per second) and represents an event based notification. A maximum of 10 SPS can be configured via the configuration interface.

Please note that an event based notification approach (i.e. when configured as 0 SPS) is generating messages only for changed values. A periodic notification will only be performed when configured greater than 0 SPS.

Supported devices:

- various types of switch contacts or proximity switches in industrial and commercial settings

periNODE GPIO mechanical specifications:

Mechanical Specification			
Housing	Dimensions	Length	54.1 mm + cable = ca. 500 mm
		Width	16.6 mm
		Depth	13.0 mm
		Weight	22 grams
	Material		Thermoplastic material
Protection		IP67	

Table 12: periNODE GPIO mechanical specifications

periNODE GPIO electrical specifications:

Electrical Specifications			
Network Interface	Connector	Type	M8 H coded male
		Standard	Similar to IEC 63171-6:2020 Style 6P-M8C ("Plug")
		Mating	Mates with periLINE and IEC 63171-6:2020 Style 6J-M8C ("Jack")
		Pin 1	TDP
		Pin 2	TDN
		Pin 3	24V DC Input
		Pin 4	GND
	Communication	Type	SPE
		Physical Layer	100BASE-T1
		Standard	IEEE 802.3bw
	Power	Voltage	24VDC
		Range	±10%
	Consumption	Average	0.5 W

Electrical Specifications			
Sensor or Actuator Interface	Connector	Kind	Pigtail cable
		Type	4x0.25mm ² , PUR
	Signal	brown	24VDC out
		white	IO2 (default Input)
		blue	GND
		black	IO1 (default Output)
		Input Logic Levels	logic 0 max. 4.3V, logic 1 min. 18.5V
		Input Current	-400µA ... + 400µA
		Input Type	high impedance
		Output Current	continuous: 100mA, peak <1s: 500mA, reverse current: -90µA ... +300µA
		Output Type	Push Pull
		Output rising timing	3.3µs
		Output falling timing	3.3µs
		Sample Rate	max: 20 SPS (software)
			typical: 0 SPS (event driven)

Table 13: periNODE GPIO electrical specifications

Note: For each channel that has been configured as output, a short circuit to GND of that pin may damage the device due to the extended heat dissipation.

Absolute maximum ratings:

Interface	Rating
Supply voltage	-40V...+40V
100BASE-T1	-40V...+40V
GPIO	continuous: -36V...+36V, peak <100 μ s: -52V...+52V

Table 14: periNODE GPIO absolute maximum ratings

Recommended Environment Conditions:

Operating conditions	-40°C...+70°C
Storage conditions	-45°C...+85°C

Table 15: Recommended environment conditions

Reverse Polarity Protection:

This smart component implements protection against polarity inversion for 24V DC input (Pin 3) and GDN (Pin 4), as well as TDP (Pin 1) and TDN (Pin 2) for the network interface.

Short Circuit limitation:

A protection mechanism against shorting the supply voltage to ground on the sensor or actuator interface might damage the device when current limitation is not applied. The current limitation of the power supply circuits is specified as max. 2A.

2.2 periSTART Media Converter

This media converter is the interface between your existing infrastructure and the Perinet Seamless IoT Connectivity system.

It converts between Fast Ethernet (100BASE-TX) and Single Pair Ethernet (100BASE-T1) and provides a 24VDC feed to attached sensors and actuators.

periSTART bundles data and power supply and transmits both to the IoT Connectivity system via the 4-wire periLINE hybrid SPE cable.

Note: The periSTART media converter has not been designed for hard real-time nor safety-critical applications.

Note: A periSTART media converter shall be connected to a traditional ethernet switch as well as power supply with a cable less than 3m of length.

Key Features:

- Conversion between 100BASE-TX and 100BASE-T1
- 24VDC power feed
- Bundles data and power for reduced cabling and simplified deployment

Perinet provides the following variants of periSTART:

2.2.1 periSTART standard



Figure 10: periSTART standard

periSTART standard is intended for field use and has M8 connectors for infrastructure connectivity:

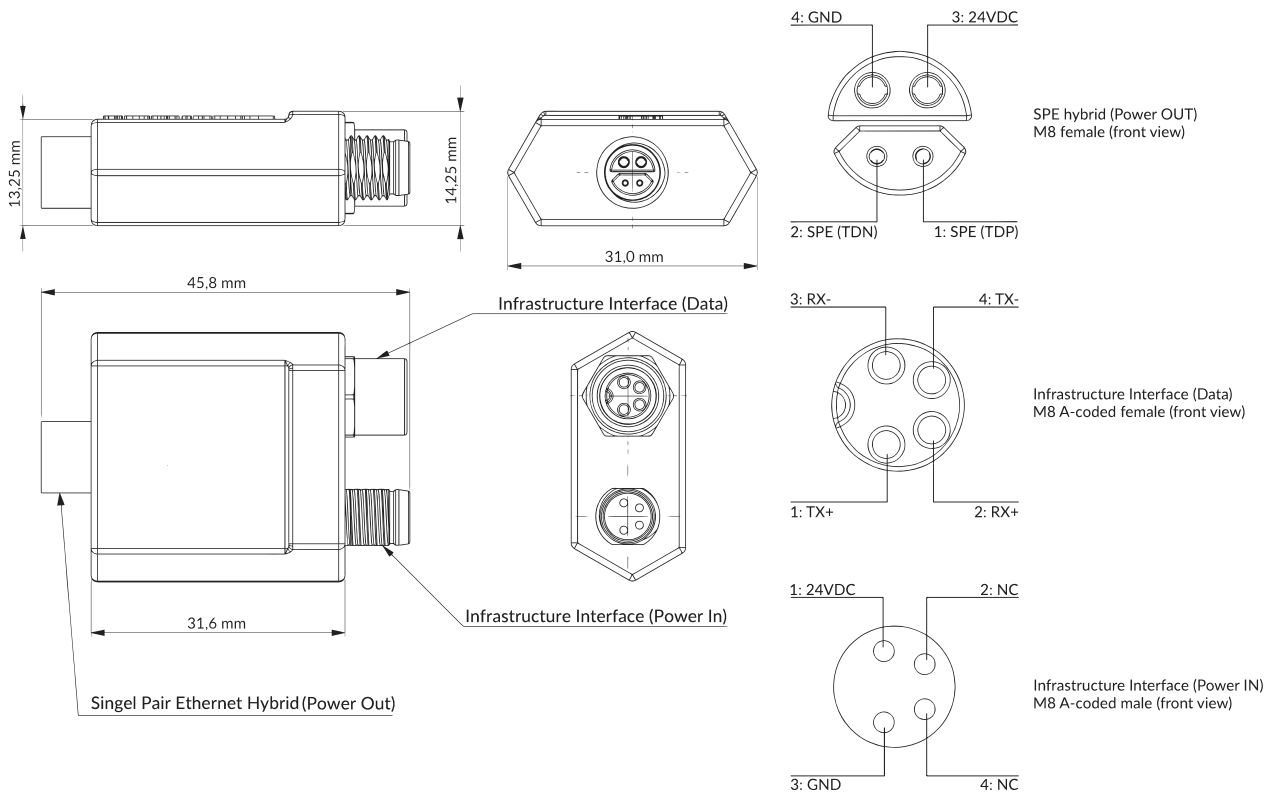


Figure 11: periSTART standard dimensional drawing & pinout

periSTART standard interfaces:

Single Pair Ethernet Hybrid Interface (Power Out)	
Connector type	M8 female connector according to IEC 63171-6:2020 (style 6P-M8C without shielding)
Communication	100BASE-T1 Single Pair Ethernet (IEEE 802.3bw)
Power	24VDC output, continuous 2A max
Infrastructure Interface (Data)	
Connector type	M8 A-coded female connector
Communication	100BASE-TX Fast Ethernet (IEEE 802.3u)
Infrastructure Interface (Power In)	
Connector type	M8 A-coded male connector
Input Voltage	24VDC ($\pm 10\%$)
Consumption	Average 0.5W (max. 2A throughput current)

Table 16: periSTART standard interfaces

periSTART standard mechanical specifications:

Mechanical Specification			
Housing	Dimensions	Length	45.9 mm
		Width	31.1 mm
		Depth	13.9 mm
		Weight	29 grams
	Material		Thermoplastic material
	Protection		IP67

Table 17: periSTART standard mechanical specifications

periSTART standard electrical specifications:

Electrical Specifications			
Network Interface 1	Connector	Type	M8 H coded female
		Standard	Similar to IEC 63171-6:2020 Style 6J-M8C ("Jack")
		Mating	Mates with periLINE and IEC 63171-6:2020 Style 6P-M8C ("Plug")
		Pin 1	TDP
		Pin 2	TDN
		Pin 3	24VDC Input
		Pin 4	GND
	Communication	Type	SPE
		Physical Layer	100BASE-T1
		Standard	IEEE 802.3bw
	Power	Voltage	24VDC
		Direction	Output
	Network Interface 2	Connector	Type
Pin 1			TXP
Pin 2			RXP
Pin 3			RXN
Pin 4		TXN	
Communication		Type	Fast Ethernet
		Physical Layer	100BASE-TX
	Standard	IEEE 802.3u	
Power interface	Connector	Type	M8 A coded male
		Pin 1	24VDC
		Pin 2	NC
		Pin 3	GND

Electrical Specifications			
		Pin 4	NC
	Input voltage	Nominal	24VDC
		Range	+/-10%
	Consumption	Average	0.5 W
	Throughput current	Max.	2A

Table 18: periSTART standard electrical specifications

Absolute maximum ratings:

Interface	Rating
Supply voltage	-40V...+40V
100BASE-T1	-40V...+40V
100BASE-TX	-40V...+40V

Table 19: periSTART standard absolute maximum ratings

Recommended Environment Conditions:

Operating conditions	-40°C...+70°C
Storage conditions	-45°C...+85°C

Table 20: Recommended environment conditions

Reverse Polarity Protection:

This smart component implements protection against polarity inversion for 24V DC input (Pin 3) and GDN (Pin 4), as well as TDP (Pin 1) and TDN (Pin 2) for the network interface.

Short Circuit limitation:

A protection mechanism against shorting the supply voltage to ground on the sensor or actuator interface might damage the device when current limitation is not applied. The current limitation of the power supply circuits is specified as max. 2A.

2.3 periSWITCH Multi-Port Switch

2.3.1 periSWITCH 3-port



Figure 12: periSWITCH 3-port

The periSWITCH 3-port switch facilitates the series bidirectional connection of multiple sensors and/or actuators.

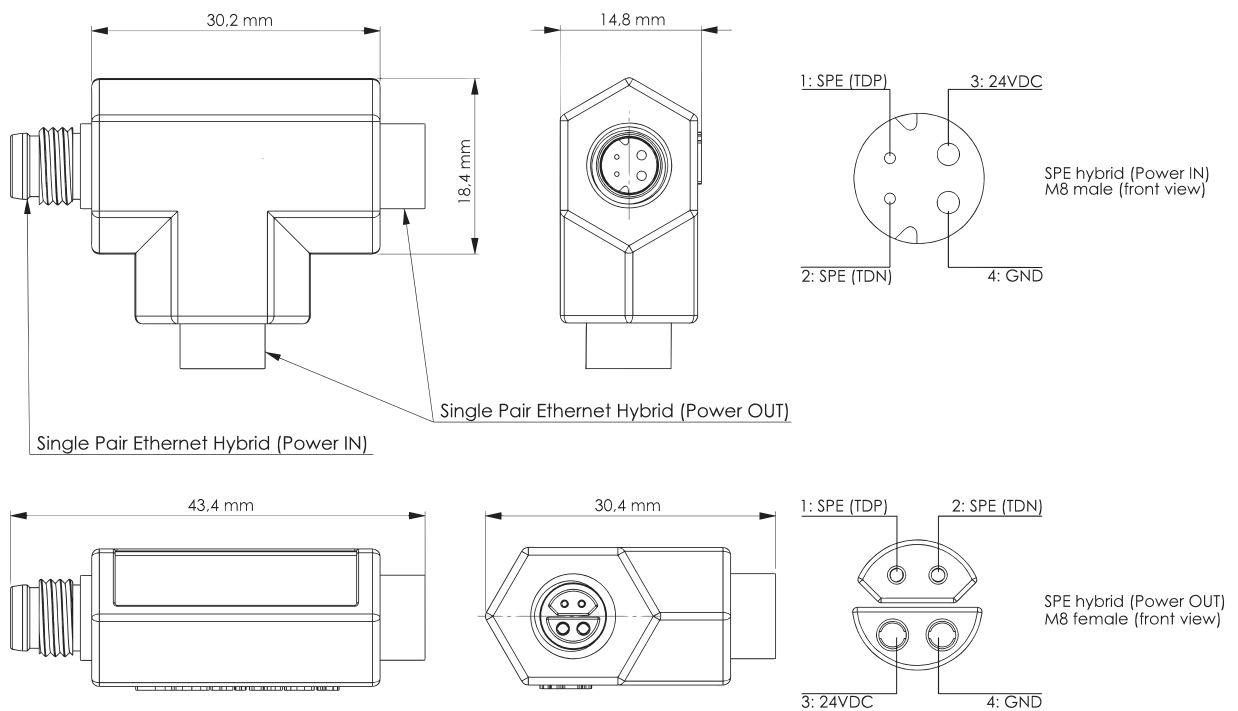


Figure 13: periSWITCH 3-port dimensional drawing & pinout

It, thereby, adds flexibility to the cabling topology (e.g. line, star etc.), enabling easy and straight-forward extension and adaptation to any given use case, setting or environment, from industrial production lines to commercial applications and smart building, to name but a few.

Key Features:

- 1 SPE hybrid IN (power and data)
- 2 SPE hybrid OUT (power and data)
- Facilitates series connection
- Enables field-level subdistribution

periSWITCH 3-port interfaces:

Single Pair Ethernet Hybrid Interface (Power In)	
Connector type	M8 male connector according to IEC 63171-6:2020 (style 6P-M8C without shielding)
Communication	100BASE-T1 Single Pair Ethernet (IEEE 802.3bw)
Power	24VDC input
Input Voltage	24VDC ($\pm 10\%$)
Consumption	Average 0.5W (max. 2A troughput current)
Single Pair Ethernet Hybrid Interfaces (2 Power Out)	
Connector type	M8 female connector according to IEC 63171-6:2020 (style 6P-M8C without shielding)
Communication	100BASE-T1 Single Pair Ethernet (IEEE 802.3bw)
Power	24VDC output

Table 21: periSWITCH 3-port interfaces

periSWITCH 3-port mechanical specifications

Mechanical Specification			
Housing	Dimensions	Length	43.4 mm
		Width	29.4 mm
		Depth	13.0 mm
		Weight	20 grams
	Material		Thermoplastic material
Protection		IP67	

Table 22: periSWITCH 3-port mechanical specifications

periSWITCH 3-port electrical specifications:

Electrical Specifications			
Network Interfaces	Connector	Type	M8 H coded male
		Standard	Similar to IEC 63171-6:2020 Style 6P-M8C ("Plug")
		Mating	Mates with periLINE and IEC 63171-6:2020 Style 6J-M8C ("Jack")
		Pin 1	TDP
		Pin 2	TDN
		Pin 3	24VDC Input
		Pin 4	GND
	Communication	Type	SPE
		Physical Layer	100BASE-T1
		Standard	IEEE 802.3bw
	Power	Voltage	24VDC
		Direction	Input (Interface 1) Output (Interface 2/3)

Electrical Specifications			
Alimentation	Voltage	Nominal	24VDC
		Range	±10%
	Consumption	Average	0.5 W
	Throughput current	Maximum	2A

Table 23: periSWITCH 3-port electrical specifications

Absolute maximum ratings:

Interface	Rating
Supply voltage	-40V...+40V
100BASE-T1	-40V...+40V

Table 24: periSWITCH 3-port absolute maximum ratings

Recommended Environment Conditions:

Operating conditions	-40°C...+70°C
Storage conditions	-45°C...+85°C

Table 25: Recommended environment conditions

Reverse Polarity Protection:

This smart component implements protection against polarity inversion for 24V DC input (Pin 3) and GDN (Pin 4), as well as TDP (Pin 1) and TDN (Pin 2) for the network interface.

Short Circuit limitation:

A protection mechanism against shorting the supply voltage to ground on the sensor or actuator interface might damage the device when current limitation is not applied. The current limitation of the power supply circuits is specified as max. 2A.

Note: The periSWITCH 3-port has not been designed for hard real-time nor safety-critical applications.

Note: A periSWITCH 3-port shall be connected to other components with cables shorter than 15m length.

2.4 periLINE Hybrid SPE Cable



Figure 14: periLINE

The periLINE hybrid Single Pair Ethernet (SPE) cable connects Perinet Smart Components within the Seamless IoT Connectivity system. It carries both data and power supply and, thereby, facilitates easy and straight-forward deployment.

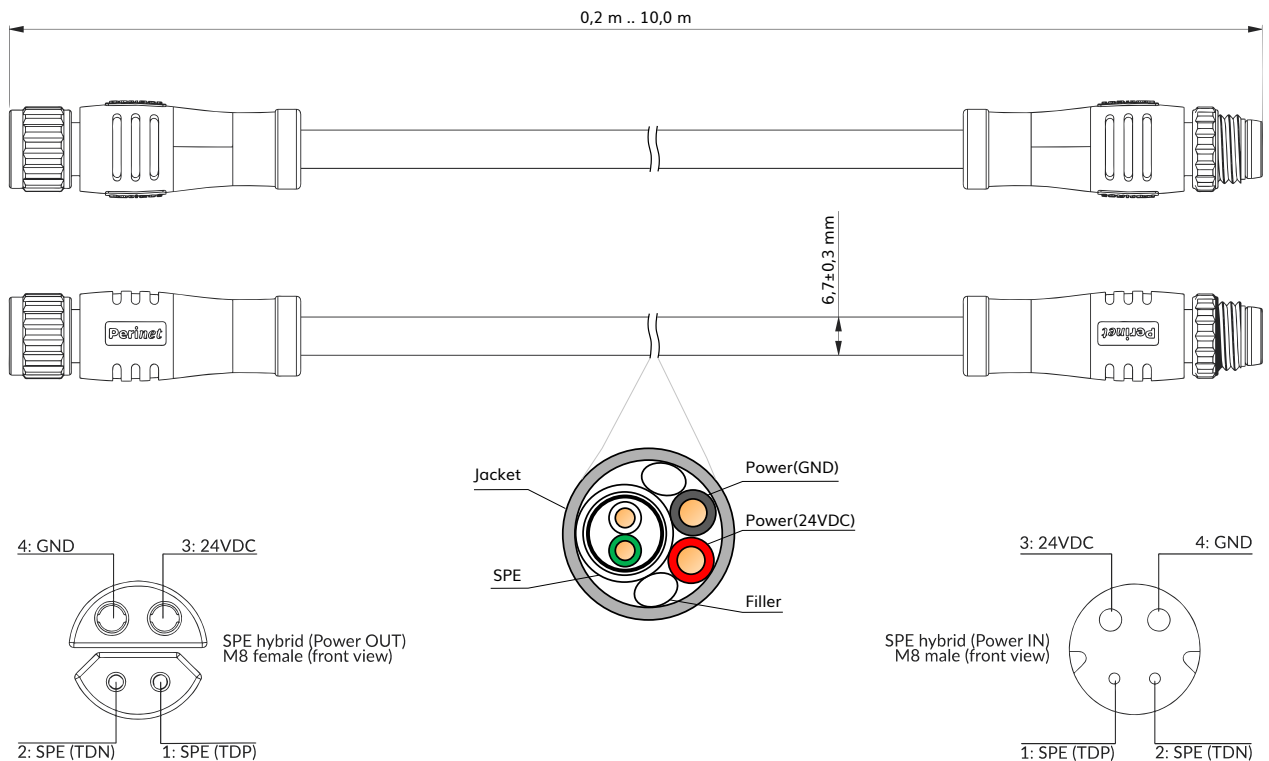


Figure 15: periLINE dimensional drawing & pinout

Connector Specification	
Type	M8 Hybrid SPE connector according to IEC 63171-6:2020 (style 6P-M8C without shielding)
Fixation	with free rotating screw nut
Connector Length	35 mm
Power Pin Diameter	1.0 mm
SPE Pin Diameter	0.5 mm
Contacts Material	Brass with 3μ" Gold Plated
Screw Nut Material	Brass with 120μ" Nickel Plated
Screw Nut Thread	M8 x 1mm (internal for female, external for male)
Insert Material	TPU + GF, black
Minimum Mating Cycles	200
Pin 1	SPE (TDP)
Pin 2	SPE (TDN)
Pin 3	Power (24VDC)
Pin 4	Power (GND)

Table 26: periLINE connector specification

Cable Specification	
Jacket Material	PUR, black
Jacket Outer Diameter	6.7 ±0.3 mm
Jacket Thickness	0.38 mm nominal
SPE Braiding	Tinned Copper, 85% coverage
SPE Al-mylar	100% coverage
SPE Insulation Material	PE (green for TDN, white for TDP)
SPE Insulation Outer Diameter	1.6 ±0.1 mm
SPE Insulation Thickness	0.45 mm nominal
SPE Conductor Material	Stranded Tinned Copper
SPE Conductor Thickness	26AWG
SPE Conductor Resistance	max. 150Ω/km at 20°C
Power Insulation Material	SR-PVC (red for 24VDC, black for GND)
Power Insulation Outer Diameter	1.45 ±0.1 mm
Power Insulation Thickness	0.23 mm
Power Conductor Material	Stranded Tinned Copper
Power Conductor Thickness	20AWG
Power Conductor Resistance	max. 36.7(Ω/km) at 20°C
Insulation Resistance	min. 100(MΩkm) at 20°C
Filler Material	Cotton
Minimum Bending Radius	25mm

Table 27: periLINE cable specification

General Specification	
Overall Length	variable from 0.2m to 15m (incl. connectors)
Degree of Protection	IP67 acc. to IEC 60529
Rated Voltage	300V
Rated Current	2A (for Power)
SPE Impedance	100±15(Ω, 1MHz)
SPE Communication	100BASE-T1 Single Pair Ethernet (IEEE 802.3bw)

Table 28: periLINE general specification

Assorted cable lengths allow for adaptation to any given use case, setting or environment, from industrial production lines to commercial applications, to name but a few.

Note: periLINE has not been designed for hard real-time nor safety-critical applications.

2.5 Perinet's Supported Network Topologies

2.5.1 Point to Point (PtP) - the simplest use case

One media converter and one smart adapter are enough to establish a direct connection of one sensor or actuator to the existing infrastructure:

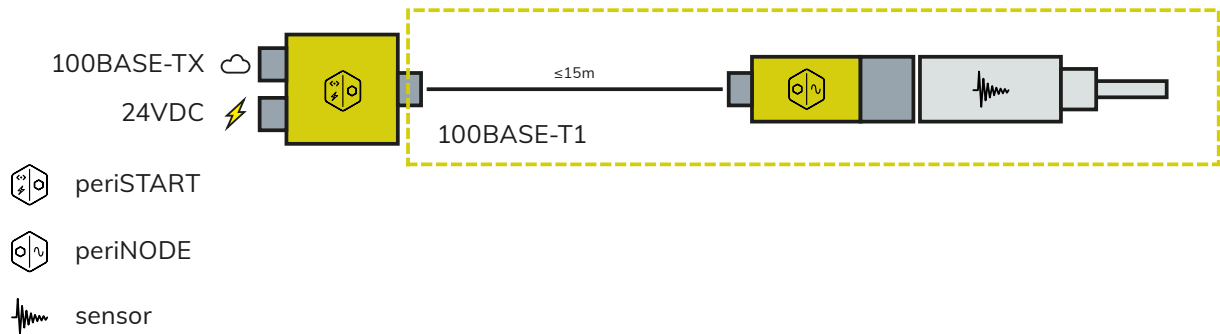


Figure 16: Point to Point Topology

2.5.2 Point to Multi-Point (PtMP)

For more complex scenarios, where multiple sensors and/or actuators are to be connected to the existing infrastructure, Perinet offers different PtMP topology options:

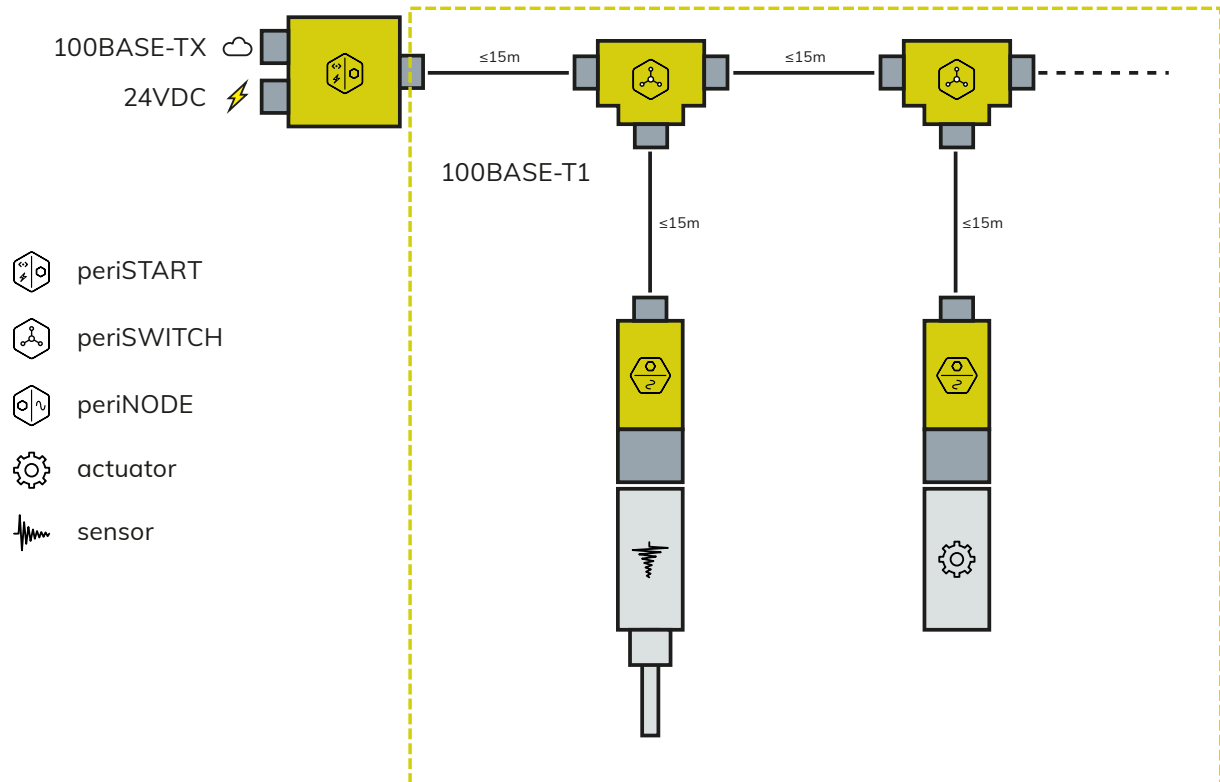


Figure 17: Line Topology

3 Smart Components Software Interface

Perinet Smart Components provide a human readable web based user interface and a machine friendly RESTful API. Both are served via the integrated HTTP server with enabled TLS and can be accessed with well known tools like a web browser.

Note: The integrated HTTP server is only accessible with enabled TLS.

The hostname for each smart component is unique and is a combination of the host prefix and a unique serial number, e.g. `periNODE-serno`, `periSWITCH-serno` or `periSTART-serno` where `serno` represents the serial number. This information is included in the label of the product, as shown in section 7.

Out of the box, a smart component implements the Zeroconf distributed name resolution protocol multicast DNS (mDNS) as well as the Link Local Multicast Name Resolution (LLMNR) protocol.

The web based user interface can be accessed via an mDNS based name resolution (`periNODE` used as example):

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

or via an LLMNR based name resolution:

```
https://periNODE-serno/
```

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

```
https://periNODE-serno.local/doc/api.proto
```

3.1 Authenticity and Security Warnings

A security warning will be triggered on environments where the trust anchor **Perinet ECC Root CA** has not been installed. Smart components produced by Perinet can be verified with the **Perinet ECC Root CA** certificate. This guarantees that the smart component originates from an authentic source, e.g. a manufacturer accredited by Perinet.

However, the **Perinet ECC Root CA** certificate is not deployed automatically and might be missed in the system when not added explicitly.

When a web client (e.g. a standard web browser) tries to verify the authenticity of the remote smart component, it tries to authenticate the remote device before the encrypted connection is established. The authentication of the remote smart component fails (as visualized in Figures 18 (Edge) and 19 (Safari)) when the **Perinet ECC Root CA** certificate is missing.

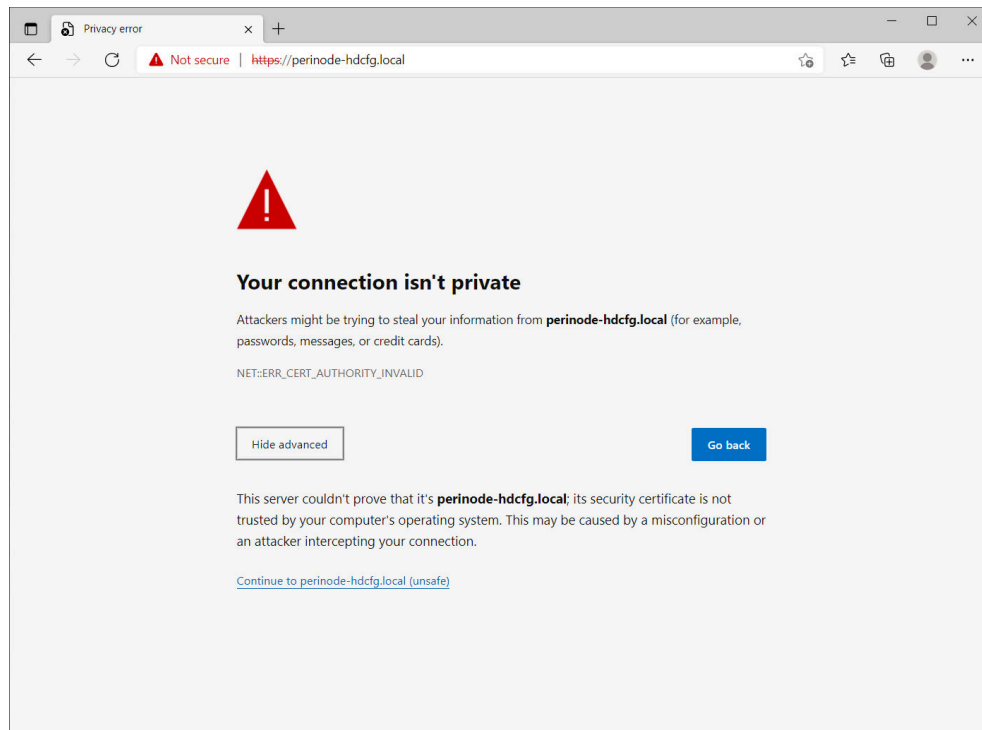


Figure 18: Edge browser security warning example

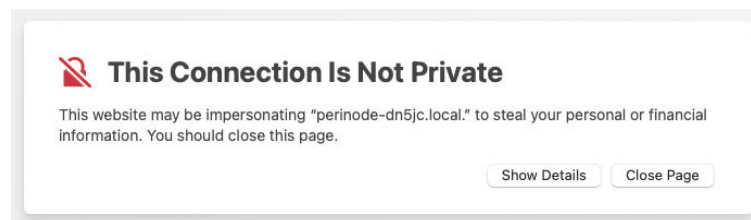


Figure 19: Safari browser security warning example

To proceed, either the trust anchor **Perinet ECC Root CA** has to be added to the system, or the warning has to be ignored. However, ignoring warnings leads to a less secure system.

Note: *Perinet recommends to add the **Perinet ECC Root CA** as a trusted entity to the system rather than proceeding by ignoring the security warning.*

3.1.1 Add Perinet as a Trust Anchor

The **Perinet ECC Root CA** certificate can be downloaded from <https://docs.perinet.io> and can be easily installed by double-clicking on the file and following the installation instructions.

For a more detailed explanation on how to install certificates in different operating systems and browsers, please refer to the *Security Certificates Installation Guide* [3]. This guide supports you in installing all certificates that might be needed for any Perinet Smart Components.

After the installation of the trust anchor **Perinet ECC Root CA**, the access to the web user interface of a smart component is possible without any security warnings. In case you still experience any problems, please contact our support.

Detailed information of the validated certificate can be obtained by clicking on the *locker* in the web browser URL bar. In the the following figures, an example of the expected certificate chain is displayed.

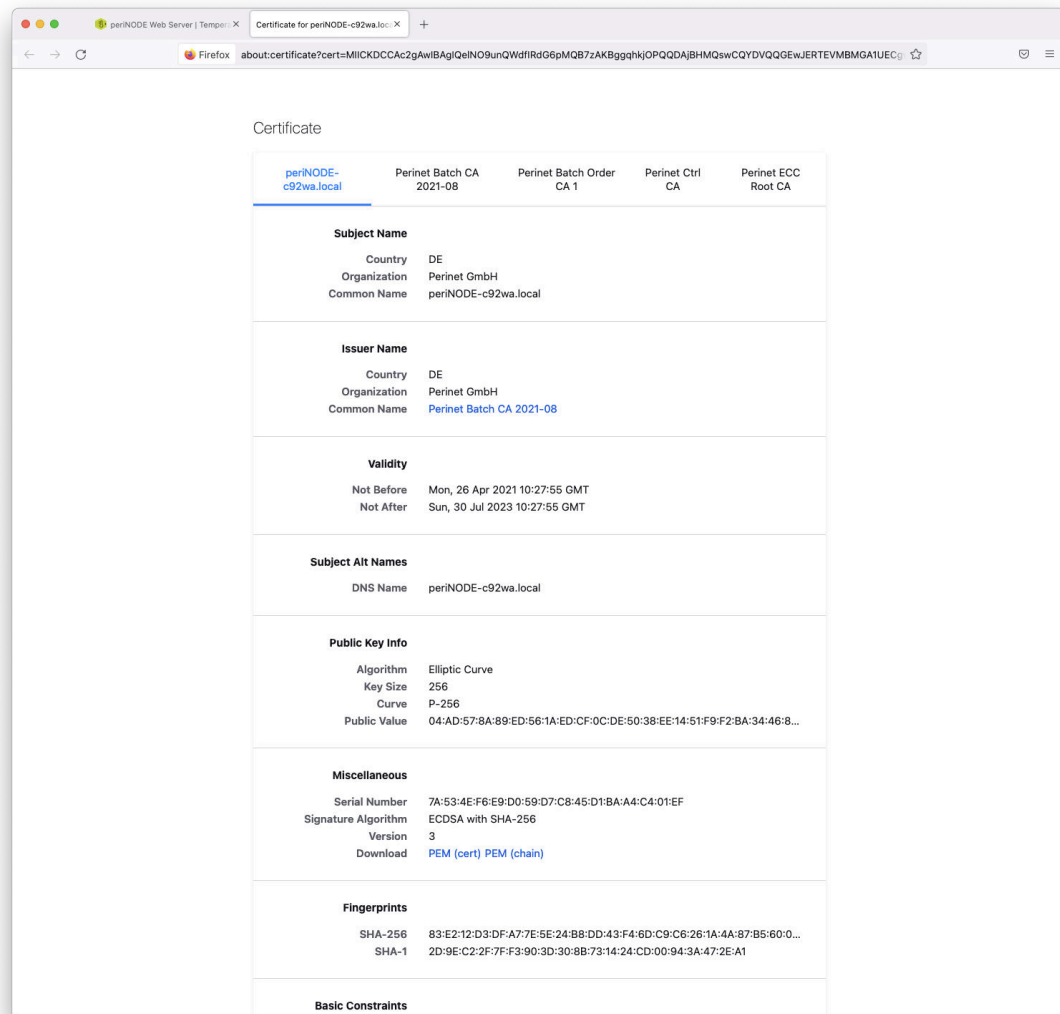


Figure 20: Firefox example security trusted chain

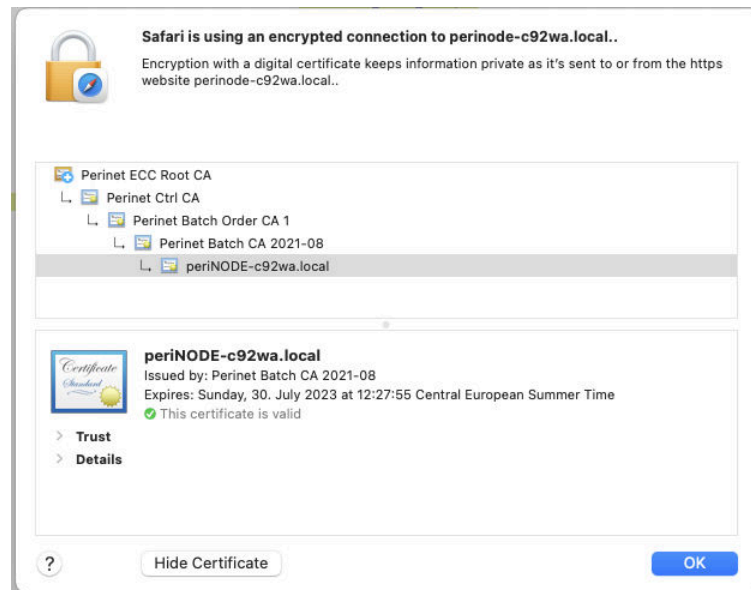


Figure 21: Safari example security trusted chain

Note: When accessing the smart component via the RESTful API, the missing trust anchor **Perinet ECC Root CA** might affect the command line tool. In that case, the trust anchor **Perinet ECC Root CA** needs to be installed system wide, contrary to just installing it inside the web client (standard browser).

3.1.2 Proceed with Exception

As described above, the warnings can also be ignored in case you do not want to add Perinet as a trusted entity for any reason. To do so, just click on "Advanced" and confirm that you want to proceed. The communication will still be encrypted, but not authenticated.

When accessing the smart component via the RESTful API, the missing trust anchor **Perinet ECC Root CA** might affect the command line tool. However, the tools offer ways to get around this, too. For example, if you are using `curl`, the option `-k, -insecure` will enforce to proceed without verifying the authenticity of the remote server.

Note: Perinet does not recommend ignoring security warnings. When the certificate is not installed, certain attacks, e.g. man in the middle attacks, are possible.

3.2 Configuration

The firmware of the periNODE provides two possible configuration options: a machine friendly RESTful API and an HTML based graphical interface. The GUI configuration is relevant for interactive use, while the RESTful API is recommended for automatic configuration. The configuration parameters will be stored persistently and restored during the boot sequence.

Note: For periSTART and periSWITCH, no software configuration is needed from the user.

3.2.1 periNODE Home

A smart component provides an user friendly interface to observe the measurement values of the connected sensor. The *Home screen* is the default page of the web user interface and is shown in Figure 22 for the variant Pt100. The *Home screen* can be reached with the following URI scheme: `https://periNODE-serno.local/`.

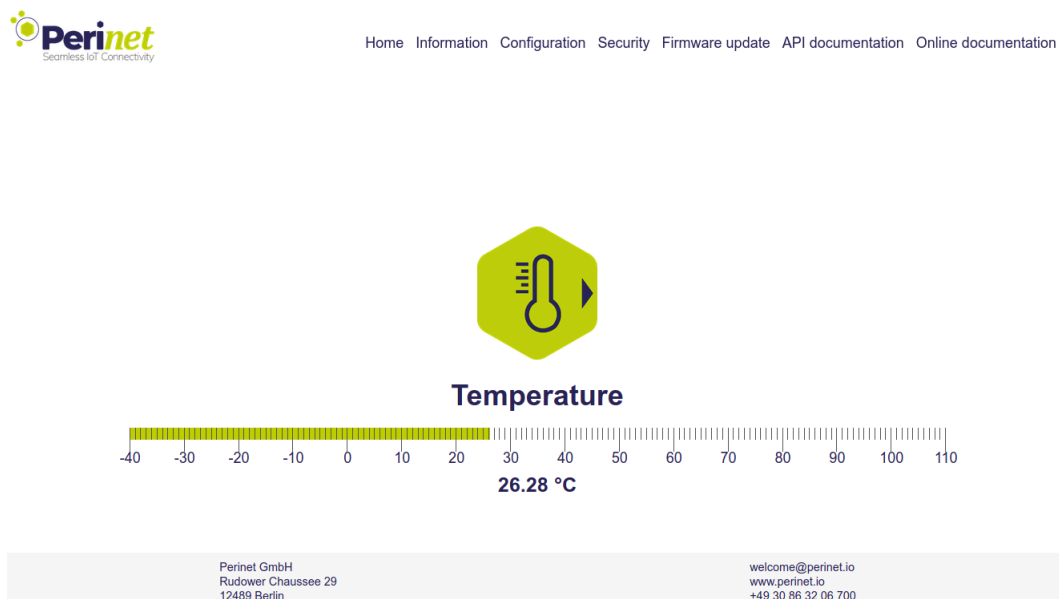
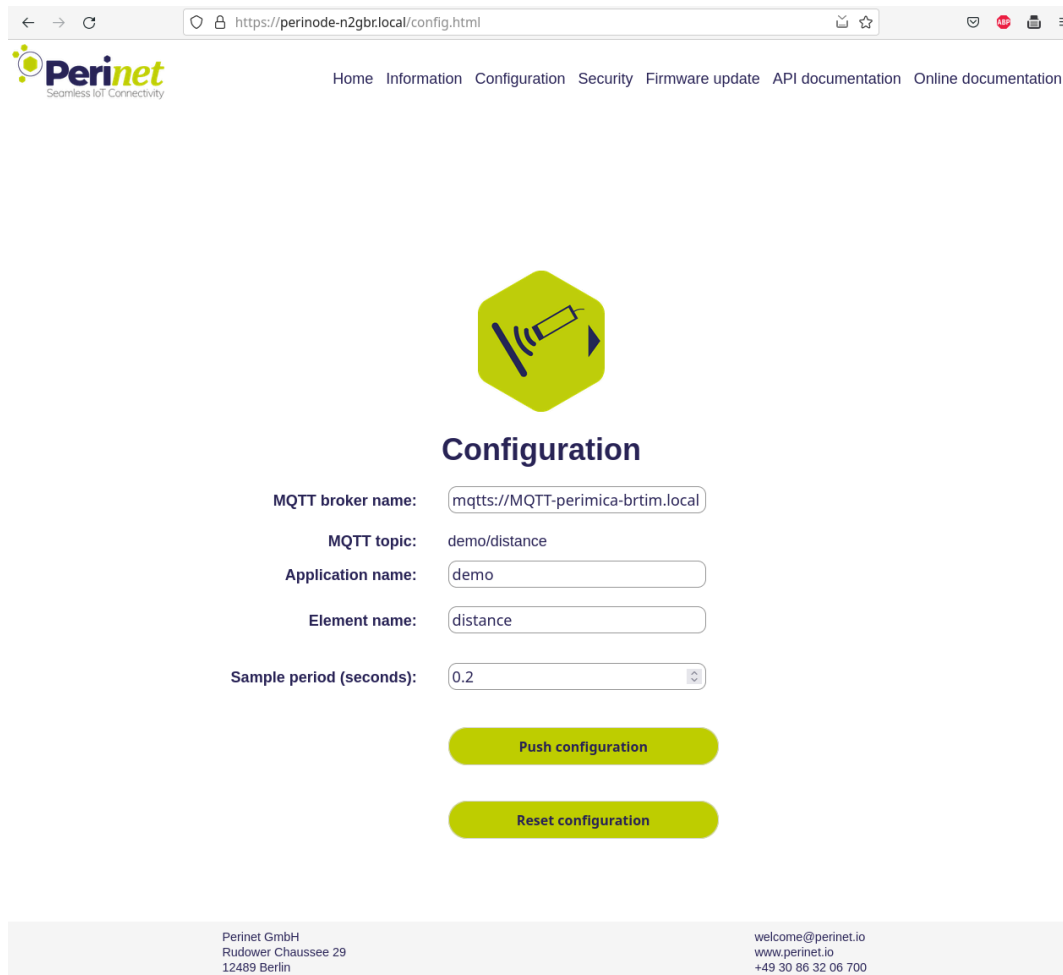


Figure 22: The Web Home Page of a periNODE Pt100 with attached temperature sensor

3.2.2 periNODE GUI Configuration



The screenshot shows a web browser window with the URL `https://perinode-n2gbr.local/config.html`. The page features the Perinet logo and a navigation menu with links: Home, Information, Configuration, Security, Firmware update, API documentation, and Online documentation. A central green hexagonal icon depicts a smart adapter. Below it, the heading "Configuration" is displayed. The form includes the following fields and values:

- MQTT broker name: `mqtt://MQTT-perimica-brtim.local`
- MQTT topic: `demo/distance`
- Application name: `demo`
- Element name: `distance`
- Sample period (seconds): `0.2`

At the bottom of the form are two green buttons: "Push configuration" and "Reset configuration". The footer contains the following information:

Perinet GmbH
Rudower Chaussee 29
12489 Berlin

welcome@perinet.io
www.perinet.io
+49 30 86 32 06 700

Figure 23: periNODE Configuration Web Page

The GUI of the periNODE allows setting up the *MQTT broker name* (***mqtt_broker_name***), the *Application name* (***application_name***), the *Element name* (***element_name***) and the *Sample period (seconds)* (***period_seconds***). A new configuration can be applied by pressing the ***Push configuration*** button. To restore the device to factory default configuration, simply press ***Reset configuration***.

An example configuration is shown in Figure 23. As the name suggests, the ***application name*** thereby defines a subset of IoT devices that belong to one application. It allows to structure elements that belong together and also defines the prefix of the MQTT topics under which periNODE smart adapters publish information. The MQTT protocol allows the use of this

prefix as general topic to subscribe to all sensor data that belongs to the same application (*demo/#* in the example).



Configuration

MQTT broker name:

MQTT topic:

Application name:

Element name (GPIO 1):

Element name (GPIO 2):

GPIO 1:

Sample period GPIO 1(seconds):

GPIO 2:

Sample period GPIO 2(seconds):

Push configuration

Reset configuration

Figure 24: periNODE GPIO Configuration Web Page

The configuration GUI for periNODE GPIO devices differs from other periNODE variants. Besides the MQTT broker and application name, the periNODE GPIO firmware allows setting up the element name for the I/O ports 1 and 2 and configuring which one should be configured as **INPUT** and/or as **OUTPUT** (see Figure 24). In addition to the port configuration, the sample period can be configured for each port. The `sample period` specifies the time between two samples and relates directly to the sample rate. Please refer to section 2.1 for more details on the supported values for the `sample rate` of the product.

3.2.3 periNODE RESTful API Configuration Information

<i>Parameter name</i>	<i>Description</i>
<code>mqtt_broker_name</code>	The URI of the MQTT broker the firmware will connect to. It can be specified as mDNS name (e.g. <code>mqtt-broker.local</code>) or as IPv6 link local address. If set, the periNODE will publish its sensor information or subscribes to topics in the case of an actuator, e.g. for the variant periNODE GPIO. The MQTT topic will be generated according to the scheme <code><application_name>/<element_name></code> .
<code>application_name</code>	Unique identifier defining a set of devices. It represents the prefix of the MQTT topic used for either publishing sensor data (source) and/or subscribing to sensor or actuator events (sink). For detailed information on the concept for setting up an IOT application, please refer to [1].
<code>sink_interface.element_name</code>	Unique identifier defining the MQTT topic suffix under which the <i>periNODE GPIO</i> actuator can be set.
<code>source_interface.element_name</code>	Unique identifier defining the MQTT topic suffix under which sensor data is published.
<code>sink-source_interface.period_seconds</code>	The number of seconds between subsequent published MQTT messages.
<code>sink-source_interface.samples_per_period</code>	The amount of sensor samples published in the interval defined by <i>period_seconds</i> .

Table 29: Configuration parameters

For RESTful API details in regards to the sensor metrics, please refer to section 3.2.4. When using the RESTful API, configuration values can be retrieved via the following URL scheme:

```
https://periNODE-serno.local/config
```

which will return the configuration in a JSON encoded format:

```
{
  "application_name": "periNODE",
  "mqtt_broker_name": "mqtts://mqtt-perimica-serno.local",
  "sink_interface_config": {
    "element_name": "output",
    "period_seconds": 0.0,
    "samples_per_period": 1,
    "type": "DIGITAL_IO_SINK"
  },
  "source_interface_config": {
    "element_name": "input",
    "period_seconds": 0.0,
    "samples_per_period": 1,
    "type": "DIGITAL_IO_SOURCE"
  }
}
```

Figure 25: periNODE configuration object retrieved from the resource /config.

The following excerpt shows how to retrieve configuration data from a periNODE by using the command line tool `curl`.

```
curl -X GET https://periNODE-serno.local/config
```

In order to change configuration values, a http PATCH request to the resource /config is necessary. As body, a valid JSON object needs to be transmitted, but it does not need to be the complete config object as received from the /config resource. Only the provided attributes will be changed in the periNODE configuration. The following excerpt shows how to update the `mqtt_broker_name` of a periNODE by using the command line tool ***curl***:

```
curl --data \
  '{"mqtt_broker_name": "mqtts://mqtt-perimica-serno.local"}' \
  -X PATCH https://periNODE-serno.local/config
```

Note: Make sure to adapt your host network interface and the serial number of the URL with the serial number of the periNODE smart adapter.

For Linux based systems with Debian based distributions, **curl** can be installed by using the following command line:

```
apt install curl
```

For other operating systems, **curl** can be obtained at <https://curl.se/download.html>.

3.2.4 periNODE RESTful API Sample Information

In order to retrieve sensor input information or change actuator output values, sample data can be requested via RESTful API and is delivered as JSON encoded object. The same object can be retrieved from the resource `/sample` of a periNODE or is published to the configured MQTT broker.

Key	Value/data type	Description
incarnation	number/integer	Number that increases with each reset/reboot of the controller and after the overflow of the <code>sequence_number</code>
sequence_number	number/integer	Message counter if sample information has changed or was changed by the user.
interface_type	string	Type of interface which represents the sensor or actuator in use, e.g. TEMPERATURE_SOURCE
data	array	Data segment of the payload, repeating group representing the actual sensor or actuator data.
unit	string	Unit of the <code><value></code>
<code><value></code> e.g. distance	bool, integer, float...	Sensor or actuator value

Table 30: Sample attributes

For RESTful API details in regards to the configuration, please refer to section 3.2.3.

An example on how to retrieve the JSON encoded sample object is shown in the following excerpt:

```
curl -X GET https://periNODE-serno.local/sample
```

```
{  "data": [
    {
      "temp": 22.89,
      "unit": "°C"
    },
    {}
  ],
  "incarnation": 8,
  "interface_type": "TEMPERATURE_SOURCE",
  "sequence_number": 35727 }
```

Figure 26: JSON encoded sample object of a periNODE Pt100

```
{  "data": [
    {
      "port": "PORT1",
      "value": false
    },
    {
      "port": "PORT2",
      "value": false
    },
    {}
  ],
  "incarnation": 28,
  "interface_type": "DIGITAL_IO_SOURCE",
  "sequence_number": 66 }
```

Figure 27: JSON encoded sample object of a periNODE GPIO

The state of an actuator can be changed by sending a JSON object via an HTTP PATCH request to the very same resource. The following excerpt shows how to enable the output of GPIO 2 via the RESTful API:

```
curl -6 -g -k --interface eth0 \  
--data '{"data":[{"value":{"port":"PORT2","value":true}}]}' \  
-X PATCH https://periNODE-serno.local/sample
```

The transmitted partial object sets the GPIO 2 ("port": "PORT2") to the logical value `true`, which is equivalent to enabling 24V on GPIO 2. The value logical `false` is equivalent to 0V.

Note: The JSON encoded sample object transmitted to the periNODE differs from the JSON object received from the periNODE.

3.3 Security Configuration

The security configuration can either be changed via the RESTful API (resource `/security`) or a web-based user interface. The web based user interface can be reached via the address `https://<hostname>.local/security.html`.

The three certificates *Host Certificate*, *Root Certificate* and *Client Certificate* can be configured for a periNODE:

- **Host Certificate:** A unique certificate used by the periNODE to authenticate itself when communicating with a remote client (e.g. a web browser). It holds the host name and, for consistency purposes, it has been issued by a trusted root CA. This certificate proves the originality of the node (the host) itself.
- **Root Certificate:** The certificate that identifies the root CA (Certificate Authority), which is trusted by a periNODE. It is used by the periNODE to authenticate remote clients (e.g. web browser) before they connect to the periNODE, when mutual TLS (mTLS) has been enabled.
- **Client Certificate:** The certificate is used by a periNODE to authenticate itself when connecting as a client to a remote server. It is for example used when the periNODE is connecting to an MQTT broker.

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

3.3.1 Mutual TLS and Authorization

In addition to the configurable certificates, a periNODE can optionally be enabled to enforce mTLS with included Role Based Access Control (RBAC). When enabled, the connecting remote client must authenticate itself by sending its client certificate before the connection can be established. The client certificate will be verified with the stored *Root Certificate*.

Note: *Remote clients can only be successfully authenticated by the periNODE after the Root Certificate has been stored and when the Client Certificate has been signed by the same Root Certificate.*

A user role is expected to be encoded in the client certificate. The user role will be used to implement a Role Based Access Control mechanism (RBAC) for the periNODE. The roles *admin*, *super* and *reader* are supported by a periNODE:

- **admin:** The administrator user has full read/write access to the periNODE with no restrictions.

- **super**: The super user can read and write to any resources with the exception of */security* and */update*.
- **reader**: The read-only user has no write access to any resource.

Note: The role reader has no permission to change actuator states via the RESTful API.

The mTLS functionality is disabled by default and will be disabled after a security and factory reset.

Note: The role based access control for the RESTful API and web user interface is deactivated by default.

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

3.3.2 Web User Interface

The web user interface provides input sections for all three certificates, i.e. *Host Certificate*, *Root Certificate* and *Client Certificate*. Figure 28 shows the input section for the *Host Certificate* as an example.



Security configuration

Host authentication ?

Host certificate:

```

Data:
  Version: 3 (0x2)
  Serial Number: 7A:53:4E:F6:E9:D0:59:D7:C8:45:D1:BA:A4:C4:01:CE
  Signature Algorithm: ecdsaWithSHA256
  Issuer: C = DE, O = Perinet GmbH, CN = Perinet Batch CA 2021-08
  Validity
    Not Before: 2021-04-21 11:18:59 UTC
    Not After: 2023-07-25 11:18:59 UTC
  Subject: C = DE, O = Perinet GmbH, CN = periNODE-d38yj.local
  Public Key Algorithm: ecPublicKey
  Public Key: (256 bit)
  pub:
    04:AA:2C:A9:2E:2E:D9:67:58:B4:9E:4F:59:F0:9F:51:EB:2A:
    1C:A9:C6:F3:42:10:F7:1E:1B:4F:42:DF:C2:62:7F:16:EC:4A:
    06:FF:76:F5:7C:77:8A:8E:8E:83:37:8F:73:R1:08:8C:3A:8C
    
```

Upload host certificate

Perinet GmbH
Rudower Chaussee 29
12489 Berlin

welcome@perinet.io
www.perinet.io
+49 30 86 32 06 700

Figure 28: Security web user interface

The certificate encoded and visible in the text area is the current stored certificate. The periNODE accepts X.509 certificates, which have been encoded with the PEM format (Base64 ASCII). Usually, the encoding scheme is reflected in the extension `.pem` but `.crt`, `.cer` and `.key` have also been observed using this scheme.

The *Host Certificate* as well as the *Client Certificate* are expected to be uploaded with concatenated corresponding private key at the end. A *Root Certificate* is expected to be uploaded without the private key.

For more details on how to generate certificates, please refer to <https://docs.perinet.io>.

Enforce mTLS access

Enabling the mTLS feature forces any remote client to authenticate towards the periNODE with a valid *Client Certificate*. The *Client Certificate* will be validated with the stored *Root Certificate*

Note: Before enabling 'Enforce mTLS access' ensure a valid Root Certificate has been stored.

3.3.3 RESTful API certificate deployment

The RESTful API provides a dedicated resource for all three certificates, which can be written only by the *admin* role and via an HTTP `PATCH` request.

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

The following excerpt shows how to update the *Root Certificate* with the `curl` command line tool.

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

'Enforce mTLS access' can be enabled via the RESTful API as shown in the following excerpt:

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

3.3.4 Reset Security Configuration

The **Reset security** button provides the option to reset the security configuration.

This operation will restore the factory default security configuration. It means that the Host Certificate and the Root Certificate once configured by the user will be replaced with the factory certificates which refer to **Perinet ECC Root CA** and the Client Certificate will be lost.

The following excerpt shows how to reset the security configuration with the **curl** command line tool.

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

When the periNODE is not accessible, e.g. due to the loss of the *Client Certificate*, please refer to the Factory Reset procedure in section 3.7.

3.4 MQTT

The application firmware of the periNODE supports the MQTT protocol to publish sensor values and/or subscribe to an information stream, which can be used to control actuators. The periNODE is compatible to MQTT protocol version 3.1.1.

The periNODE firmware is able to detect MQTT brokers through an mDNS based service discovery protocol (dns-SD). Brokers are discovered by browsing for the **service_type** `_mqtt._tcp` (or `_secure-mqtt._tcp` for MQTT brokers using TLS) and the hostname of the broker e.g. "mqtt-perimica-serno". Discovered brokers can be viewed in the MQTT broker text field drop-down list of the periNODE configuration Web UI. MQTT brokers which do not have the ability to advertise themselves via the mDNS based service discovery protocol can be configured manually via the web user interface or RESTful API.

As the periNODE firmware only supports IPv6, the local network and the broker must also be capable of communicating via IPv6.

The MQTT topic under which the periNODE smart adapter publishes messages or subscribes to is based on the **application_name** and **element_name** provided by the configuration parameters:

```
topic : "<application_name>/<element_name>"
```

The MQTT message payload is in a JSON encoded format. After configuring the periNODE with the desired broker, it is possible to subscribe to periNODE sensor data using any MQTT client, as the client that is provided by the **mosquitto** library.

MQTT Client - Command Line Examples

If the configured **application_name** of the device is **perinode**, the **element_name** is **distance** and the configured MQTT broker is **mqtt.local**, one can subscribe to the periNODE messages (e.g. using the **mosquitto** library):

```
mosquitto_sub -h mqtt.local -t "perinode/distance"
```

which prints sample JSON messages similar to the following:

```
{"incarnation":2, "sequence_number":178, "interface_type":"DISTANCE_SOURCE",  
{"data":[{"unit":"mm","distance":311}]}
```

3.5 Secure MQTT



[Home](#)
[Information](#)
[Configuration](#)
[Security](#)
[Firmware update](#)
[API documentation](#)
[Online documentation](#)



Configuration

MQTT broker name:

MQTT topic:

Application name:

Element name:

Sample period (seconds):

Perinet GmbH
 Rudower Chaussee 29
 12489 Berlin

welcome@perinet.io
 www.perinet.io
 +49 30 86 32 06 700

Figure 29: Configure secure MQTT

The communication between periNODE and MQTT broker can also be protected through TLSv1.2. The periNODE accesses the MQTT broker, verifying the broker certificate against the root certificate configured in the security page. Therefore, the periNODE MQTT client uses the same root certificate as the periNODE HTTPS server to authenticate the remote entity.

Mutual TLS (mTLS), where the remote MQTT broker authenticates the periNODE before continuing with the connection request, is supported as well. The periNODE uses the configured *client certificate* (see section 3.3) in order to authenticate itself towards the MQTT broker.

Secure MQTT on the periNODE smart adapter can be configured implicitly in the configuration page of the periNODE web GUI (Figure 29), by inserting the **mqttp://** prefix in front of the **MQTT broker name**. Leaving out the prefix at all or using **mqtt://** as prefix will disable TLS entirely for the MQTT client connection, which is not recommended by Perinet.

Mosquitto Examples using mTLS:

Subscribing to periNODE sensor data:

```
mosquitto_sub -h mqtt-secure.local -t test/topic -p 8883  
--cafile root-ca.crt --cert client.pem --key client.key
```

Publishing messages:

```
mosquitto_pub -h mqtt-secure.local -t test/topic -p 8883 -m testmessage  
--cafile root-ca.crt --cert client.pem --key client.key
```


3.6 Firmware Update

Perinet may provide security and/or feature updates for the smart components over time. Those updates will be delivered for each periNODE variant as an individual artifact. 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.*

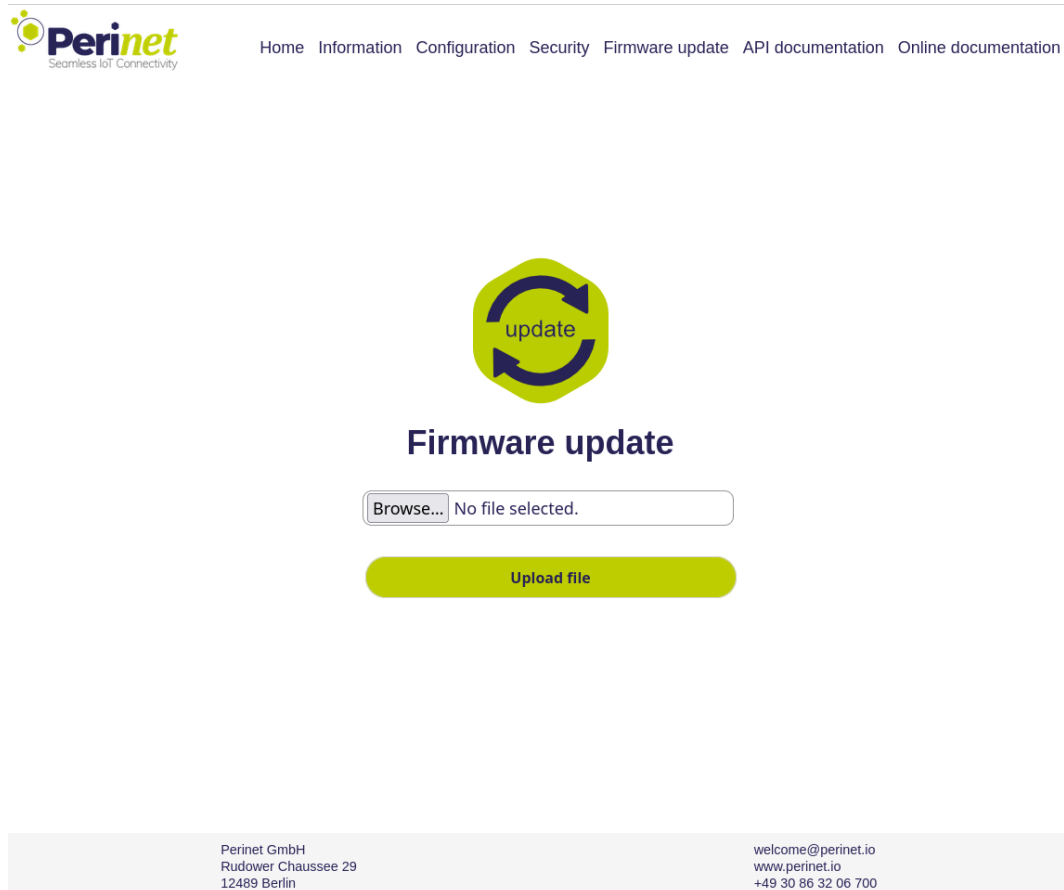


Figure 30: Firmware update

Note: *A firmware update performed with the Firefox browser has been observed to corrupt the firmware image during the upload and result in a missing web user interface. A periNODE would still be functional and accessible via the RESTful API and accept firmware images via the RESTful API.*

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

1. Download the newest version of the firmware matching your product from <https://downloads.perinet.io>
2. Access the web page of the component through your browser:
<https://hostname.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, the reboot of the component to the newly installed firmware will only be automatically triggered when using the web user interface.

3.6.1 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-serno.local/update/  
# reboot into new firmware  
curl -X PATCH https://periNODE-serno.local/reboot
```

3.7 Factory Reset

A factory reset of a smart component 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 smart component via the reset cable (see Figure 31) 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. When resetting a periSWITCH or a periSTART, ensure that only one network port of the product is connected.

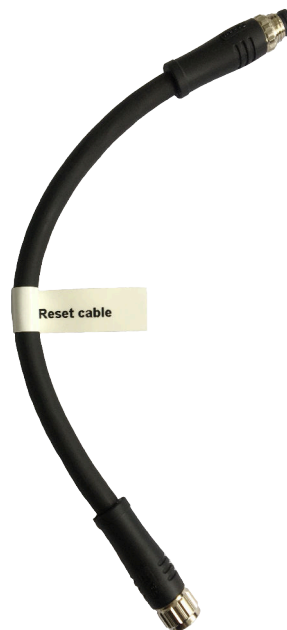


Figure 31: Perinet Reset Cable for Factory Reset

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

4 Supported Web Browsers

A smart component is implementing a simple and small web server to display information and provide an easy way to configure the device as needed. It is recommended to access the web pages through the following browsers:

- Windows Edge
- Safari

Important Notes:

- Firefox is not currently supported by our firmware due to a bug present after a firmware update from Firefox.
- Google Chrome does not support Link Local Address IPv6 without internet connection, therefore it does not support Perinet Smart Components.

5 Qualifications and Approvals

5.1 CE Compliance



Figure 32: CE sign

Perinet products are CE compliant. They meet EU safety, health and environmental protection requirements.

5.1.1 Electromagnetic Compatibility (EMC)

Perinet Smart Components can be used in the electromagnetic environment for which they have been designed without causing disturbances to other equipment or being affected by them (EMC Directive 2014/30/EU).

Perinet products comply to the harmonized Immunity Standard for industrial environments (EN 61000-6-2:2005, EN 61000-6-2:2005/AC:2005), as well as to the harmonized Emission Standard for industrial environments (EN 61000-6-4:2007, EN 61000-6-4:2007/A1:2011).

5.1.2 Environmental Compatibility

Perinet products fulfill all environmental requirements demanded by the IEC 61131-2:2007 standard for programmable controllers.

Operation temperature range is from -40°C to 70°C.

Perinet Smart Components pass the following environmental tests for protection by enclosures (EN 60529:1991 + A1:2000 + A2:2013):

- IP-Code x5: jetting water
- IP-Code x7: temporary immersion
- IP-Code 6x: dust-tight (with underpressure)

With regards to

EN 62368-1:2014/AC:2015 Audio/video, information and communication technology equipment - Part 1: Safety requirements

Perinet Smart Components are not within the scope of this standard.

Perinet devices are only to be used in local installations with common grounding. There are no lines to be connected directly that leave the building where the electronic equipment is installed.

5.1.3 RoHS and REACH



Figure 33: RoHS sign

Perinet products are RoHS and REACH compliant.

This means, they do not contain hazardous substances, such as lead, cadmium, mercury, hexavalent chromium, polybrominated biphenyls (PBB) and polybrominated diphenyl ethers (PBDE), bis(2-ethylhexyl) phthalate (DEHP), butyl benzyl phthalate (BBP), dibutyl phthalate (DBP) and diisobutyl phthalate (DIBP).

All products supplied by Perinet are compliant with the substance-related requirements of REACH.

Our suppliers keep our products free from substances of very high concern (SVHC) dictated by REACH.

Provided that goods obtained from Perinet GmbH are used for normal purposes and in normal circumstances, we can state that we do not supply chemical products.

5.1.4 Waste from Electrical and Electronic Equipment (WEEE)



Figure 34: WEEE sign

Perinet products are WEEE compliant. They respect EU rules on treating waste electrical and electronic equipment, to contribute to sustainable production and consumption.

6 Installation

The installation concept foresees a solid mounting of the sensors, as well as strain reliefs for the cables which are capable to hold the Perinet Smart Components. The smart components are coming without any kind of mounting features, such as holes or screw threads. Their small size allows them to be held in position by the sensors and the cables.

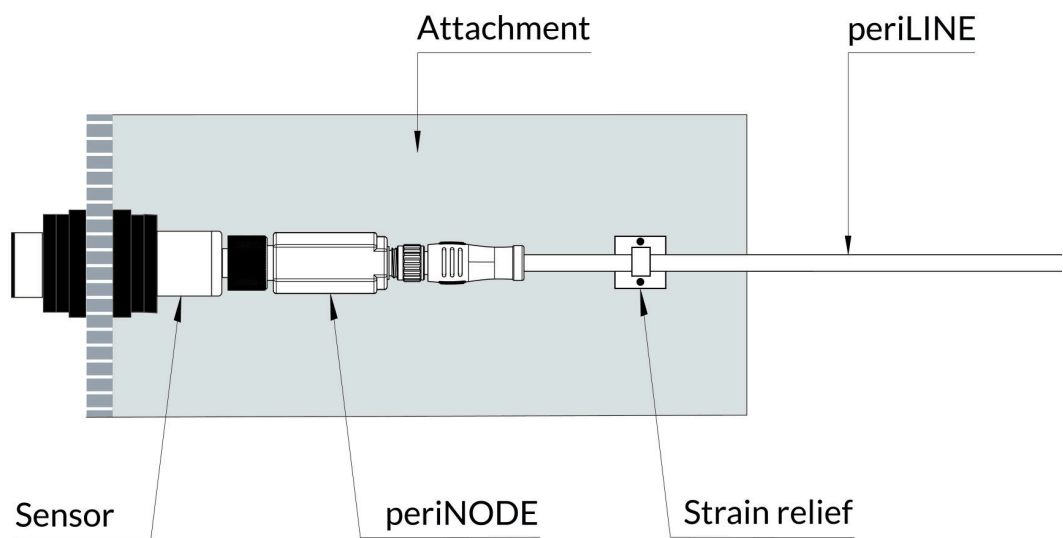


Figure 35: periNODE installation

periNODE 0-10V and periNODE Pt100 can be carried by the mounted sensor by being plugged in with the M12 circular connector. To prevent damaging transversal forces to the setup, the periLINE should be fixed with some kind of strain relief to a solid attachment.

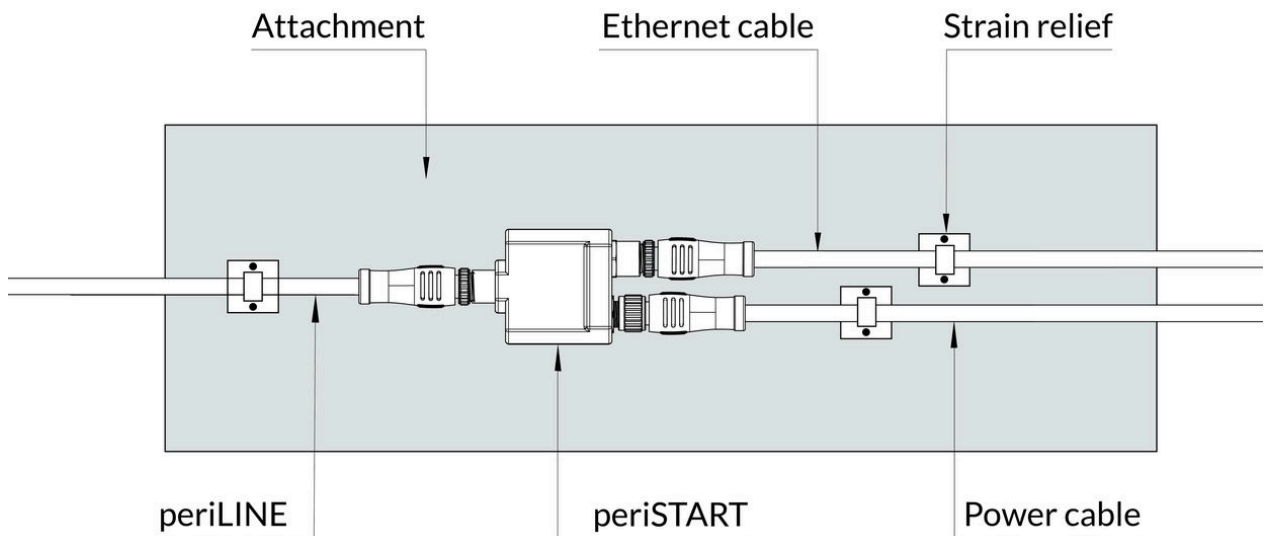


Figure 36: periSTART installation

periSTART standard, periNODE GPIO as well as periSWITCH 3-port are held only by their cables. The cables need to be fixed next to the device with some kind of strain relief to a solid attachment.

Note: The Perinet Smart Component periSTART standard is intended to be used close to a network outlet or to a standard ethernet switch. It is recommended not to exceed the cable length of 3m for the power cable as well as Ethernet cable.

Note: A periNODE shall be connected to its sensor or actuator with a cable that is less than 3m length.

Note: Each periNODE variant, periSWITCH and periSTART shall be connected to other network components via a periLINE shorter than 15m length. (for cables with unshielded data lines, 15m; for cables with shielded data lines 30m).

For your personal safety and to avoid damage of property, please refer as well to our **Safety Instructions**.

7 Labeling and Ordering Information

7.1 Product Marking

The product marking contains the most relevant information regarding the device. The same information is available as well in the machine readable *Data Matrix Code*.

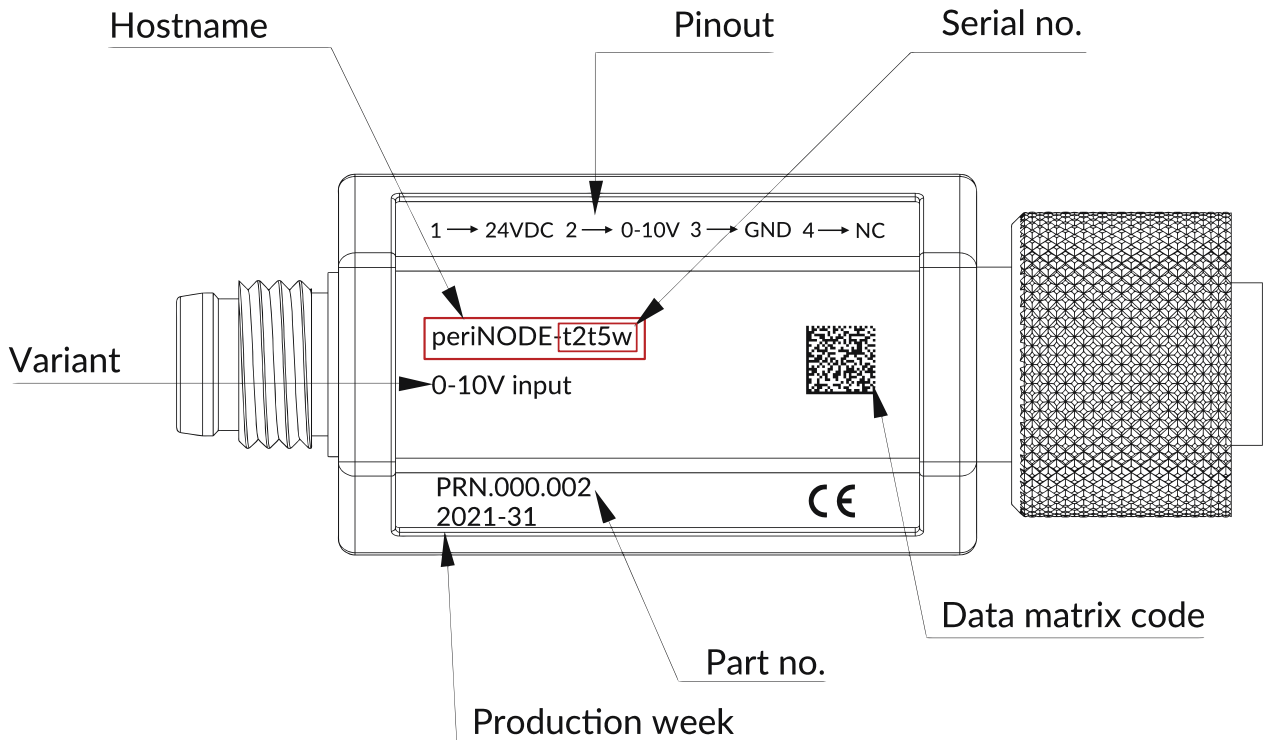


Figure 37: Product marking

- **Pinout** for sensor or actuator interface (periNODE)
- **Hostname:** this is the unique host name of the Perinet Smart Component.
- **URL:** For periNODE, by typing “https://” followed by the host name, followed by “.local/” in your browser, you can reach the device in your Local Area Network. For example, if the device is marked with “periNODE-t2t5w”, the URL is `‘https://periNODE-t2t5w.local/’`
- **Serial no.:** the last five characters of the host name are the unique serial number of a given device. For example, if the device is marked with “periNODE-t2t5w”, the serial number is “t2t5w”.

- **Variant:** for periNODE, the variant specifies the interface signal type, i.e. which signal can be applied on sensor or actuator interface. For example, if the device is marked with “0-10V input”, you can attach a sensor with a 0-10V output signal to the periNODE.
- **Part no.:** the Manufacturer Part Number of this type of devices. This is the precise identifier for this type of devices on the market.
- **Production week:** This tells you when this device was produced. The format is YYYY-CW. For example, if the device is marked with “2020-36”, it means that this device was produced in the 36th calendar week of the year 2020.
- **CE sign:** This sign indicates the CE-compliance of the product (see *Qualifications and Approvals*).
- *Data Matrix Code*

7.2 Data Matrix Code

For better machine-readability, the most important information is available also encoded as a two-dimensional Data Matrix (ECC200) code. To read it, an optical reading device is needed. This can be either an industrial scanner or even a smartphone with an appropriate application running.

The structure of the content is the following:

- Line 1: URL, containing the host name, which includes the serial number
- Line 2: Manufacturer Part Number
- Line 3: Production calendar week

7.3 Ordering Information

Part number	Product Name
PRN.000.002	periNODE 0-10V
PRN.000.006	periNODE Pt100
PRN.000.010	periNODE GPIO
PRN.000.003	periSTART standard
PRN.000.008	periSWITCH 3-port
PRN.000.017	periLINE 0.2m
PRN.000.009	periLINE 1m
PRN.000.011	periLINE 2m
PRN.000.013	periLINE 5m
PRN.000.015	periLINE 10m

Table 31: Ordering information

7.4 Packaging

Perinet Smart Components are delivered in boxes containing 50 devices (for all periNODE variants) and 25 devices (periSWITCH 3-port and periSTART standard).

8 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

9 Safety Instructions

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




 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 32: 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.

9.1 Power Supply

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.

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.

9.2 Protection

CAUTION

Property damage

Property damage caused by electrostatic charging.

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

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.

9.3 Repair

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

WARNING

Dangers due to unauthorized opening of the device

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

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.

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.

9.4 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.*

A Glossary

Abbreviation	Description
AFE	Analog Front End
API	Application Programming Interface
CA	Certification Authority, a trusted entity which is represented by a certificate that is used to verify the signature on a certificate issued by that authority (trust anchor).
COTS	Commercial Off-The-Shelf refers to ready-made merchandise that is available for sale.
dns-sd	DNS Service Discovery is a way of using standard DNS programming interfaces, servers and packet formats to browse the network for services.
EMC	ElectroMagnetic Compatibility
FQDN	Fully Qualified Domain Name, sometimes also referred to as an absolute domain name.
GPIO	General-Purpose Input/Output
I2C	Inter-Integrated Circuit is a synchronous, multi-master, multi-slave, packet switched, single-ended, serial bus.
IC	Integrated Circuit
IIoT	Industrial Internet of Things
IPv6	Internet Protocol Version 6, a communication protocol
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.
MDI	Media Dependent Interface
mDNS	multicast Domain Name Service, a protocol that implements a local distributed name resolving mechanism.
mTLS	Mutual TLS extends the TLS protocol by requiring clients to pass certificates allowing to provide authorization mechanisms of Application services.
MQTT	Message Queuing Telemetry Transport is a lightweight, publish-subscribe based network protocol that transports messages between devices.

OTP	One-Time Programmable memory
SLAAC	Stateless Address Auto Configuration is a mechanism that enables each host on the network to auto-configure a unique IPv6 address without any device keeping track of which address is assigned to which node.
SPE	Single Pair Ethernet
SPI	The Serial Peripheral Interface (SPI) is a synchronous serial communication interface specification used for inter IC communication.
SPS	Samples per second.
TLS	Transport Layer Security Protocol. Used by Application Protocols like MQTT or http to allow secure data transfer.
UART	A Universal Asynchronous Receiver-Transmitter is a computer hardware device for asynchronous serial communication

Table 33: Glossary

B Smart Components API Specification

```

1 //API_VERSION=11
2
3
4 syntax="proto3";
5
6 message FwUpdate {
7     message Header {
8         uint32 length = 1;
9         uint32 chksum = 2;
10    }
11    string signature = 1; // signature
12    Header header = 2; // meta information
13    bytes data = 3; // firmware update binary data
14 }syntax="proto3";
15
16 package perinet.periNODE.api;
17
18 /**
19 *   Message that represents a measured typed sample. see also NodeType
20 *   Example JSON for a APPTYPE=DISTANCE_SOURCE {"incarnation" : 1,"↵
    sequence_number":100,"interface_type":"DISTANCE_SOURCE", "data": [ {"↵
    unit":"mm","distance": 180 } ] }
21 *   Example JSON for a APPTYPE=PH_SOURCE {"incarnation" : 1,"↵
    sequence_number":100,"interface_type":"PH_SOURCE", "data": [ {"unit":"↵
    pH","ph": 0.1},{ "unit":"°C","temp": 22.6 } ] }
22 *   Example JSON for a APPTYPE=DIGITAL_IO_SOURCE {"incarnation" : 1,"↵
    sequence_number":100,"interface_type":"DIGITAL_IO_SOURCE", "data": [ ↵
    {"unit":"","input":{"port":"photoelectric",value:1}} ] }
23 */
24 message Sample {
25     message InterfaceSample {
26         string unit = 1; // base units according to chosen APPTYPE for ↵
            the source interface might be Si based, empty if n.a.
27         oneof sample_values {
28             uint32 raw = 2; // value for Interfaces configured as RawType
29             float temp = 3; // value for Interface sources of type ↵
                TEMPERATURE_SOURCE
30             float pH = 4; // value for Interface sources of type ↵
                PH_SOURCE
31             uint32 distance = 5; // value for Interface sources of type ↵
                DISTANCE_SOURCE
32             IOStatus io = 6; // value for Interface sources of type ↵
                DIGITAL_IO_SOURCE
33             Blind_Control blind_ctrl = 7; // value for Interface sources ↵
                of type BLIND_CONTROL_SOURCE
34             float analog = 8;
35             float acceleration = 9;
36         }
    }
}

```

```

37     }
38
39     /*required*/ uint32 incarnation = 1; // partial unique number of a ←
        message sample, which becomes unique in combination with ←
        sequence_number
40     /*required*/ uint64 sequence_number = 2; // partial unique number of ←
        a message sample, which becomes unique in combination with ←
        incarnation
41     repeated InterfaceSample data = 4; // a collection of samples, which ←
        are defined by the InterfaceType.
42 }
43
44 enum PortNumber {
45     Unassigned = 0;
46     PORT1 = 1;
47     PORT2 = 2;
48 }
49
50 /**
51 * sample type for NodeType of type DIGITAL_IO_SOURCE
52 */
53 message IOStatus {
54     PortNumber port = 1;
55     bool value = 2;
56 }
57
58 enum Blind_Control {
59     STOP = 0;
60     UP = 1;
61     DOWN = 2;
62 }
63 syntax="proto3";
64
65 // package perinet;
66 package perinet.periNODE.api;
67
68 message Certificate {
69     bytes certificate = 1;
70 }
71
72 message CertPKBundle {
73     string cert_pk_bundle = 1;
74 }
75
76 message PrivateKey {
77     bytes private_key = 1;
78 }
79
80 message SecurityConfig {
81     bool enable_user_role = 1;
82     string root_cert = 2; //optional
    
```

```

83     string host_cert = 3; //optional
84     string client_cert = 4; //optional
85 }
86
87 message FwUpdateStream {
88     message Header {
89         uint32 length = 1; //byte count
90         uint32 digest = 2; //sha256 digest over data
91     }
92     string signature = 1; //signed header over sha256
93     Header header = 2;
94     bytes data = 3; // firmware update binary data
95 }
96 /*
97 //header
98 {
99     "length":286720, //cat _build/update_periCORE.img|wc -c
100    "digest":"↵
        U0hBMjU2KF9idWlsZC91cGRhdGVfcGVyaUNPUkUuaW1nKT0gYWI1NjFiOGM2NmJjYTdmNDdhM2E3Z
        =" //openssl dgst -sha256 _build/update_periCORE.img | base64 -w0
101 }
102 */syntax = "proto3";
103
104 package perinet.periNODE.api;
105
106 message MqttBrokerList {
107     repeated string broker = 1;
108 }
109
110 syntax="proto3";
111
112 import public "google/protobuf/empty.proto";
113
114 import "google/api/annotations.proto"; //needs the https://github.com/↵
        googleapis/googleapis repo as include path
115
116 import "Sample.proto";
117 import "NodeConfig.proto";
118 import "NodeInfo.proto";
119 import "MqttBrokerList.proto";
120 import "SecurityConfig.proto";
121 import "FwUpdate.proto";
122
123 package perinet.periNODE.api;
124
125 //needs the https://github.com/googleapis/googleapis repo as include path
126
127 /**
128 * http REST compliant service of a periNODE.
129 */
130 service PeriNodeService {
    
```

```

131 // request general purpose information on the Node
132 rpc GetInfo (google.protobuf.Empty) returns (NodeInfo){
133     option (google.api.http) = {
134         get: "/info"
135     };
136 };
137 // request the configuration information of a periNODE
138 rpc GetConfiguration (google.protobuf.Empty) returns ( NodeConfig ){
139     option (google.api.http) = {
140         get: "/config"
141     };
142 };
143 // request mqtt broker list information of a periNODE
144 rpc GetMqttBrokerList (google.protobuf.Empty) returns ( ←
    MqttBrokerList ){
145     option (google.api.http) = {
146         get: "/mqtt"
147     };
148 };
149 // update the configuration of a periNODE via PATCH or url encoded ←
    GET request
150 rpc SetConfiguration (NodeConfig) returns (NodeConfig){
151     option (google.api.http) = {
152         patch: "/config"
153         body: "*"
154     };
155 };
156
157 // request the last sensor information of a periNODE
158 rpc GetSample (google.protobuf.Empty) returns ( Sample ){
159     option (google.api.http) = {
160         get: "/sample"
161     };
162 };
163
164 // provide the actuator data for a periNODE
165 rpc SetSample (Sample) returns ( Sample ){
166     option (google.api.http) = {
167         patch: "/sample"
168         body: "*"
169     };
170 };
171
172 // request the security configuration of a periNODE
173 rpc GetSecurityConfig (google.protobuf.Empty) returns ( ←
    SecurityConfig ){
174     option (google.api.http) = {
175         get: "/security"
176     };
177 };
178 // request the client certificate for a periNODE
    
```

```

179     rpc GetClientCert (google.protobuf.Empty) returns ( Certificate ){
180         option (google.api.http) = {
181             get: "/security/client-cert"
182         };
183     };
184     // request Perinet root certificate for a periNODE
185     rpc GetRootCert (google.protobuf.Empty) returns ( Certificate ){
186         option (google.api.http) = {
187             get: "/security/root-cert"
188         };
189     };
190     // request Perinet root certificate for a periNODE
191     rpc GetHostCert (google.protobuf.Empty) returns ( Certificate ){
192         option (google.api.http) = {
193             get: "/security/host-cert"
194         };
195     };
196     // provide security configuration for a periNODE
197     rpc SetSecurityConfig (SecurityConfig) returns ( google.protobuf.↵
198         Empty ){
199         option (google.api.http) = {
200             patch: "/security"
201             body: "*"
202         };
203     };
204     // provide client certificate including private key for a periNODE
205     rpc SetClientCertBundle (CertPKBundle) returns ( google.protobuf.↵
206         Empty ){
207         option (google.api.http) = {
208             patch: "/security/client-cert"
209             body: "*"
210         };
211     };
212     // provide user defined root certificate for a periNODE
213     rpc SetRootCert (Certificate) returns ( google.protobuf.Empty ){
214         option (google.api.http) = {
215             patch: "/security/root-cert"
216             body: "*"
217         };
218     };
219     // provide user defined certificate and private key for a periNODE
220     rpc SetHostCertBundle (CertPKBundle) returns ( google.protobuf.Empty ↵
221         ){
222         option (google.api.http) = {
223             patch: "/security/host-cert"
224             body: "*"
225         };
226     };
227     // update periFIRMWARE, does not affecting configuration nor ↵
228     manufacturing data
    
```



```

225     rpc UpdateFirmware (stream FwUpdate) returns ( google.protobuf.Empty ↵
    ) {
226         option (google.api.http) = {
227             put: "/update"
228             body: "*"
229         };
230     };
231     // invoke softreset of a periNODE
232     rpc SetReboot (google.protobuf.Empty) returns (google.protobuf.Empty)↵
    {
233         option (google.api.http) = {
234             patch: "/reboot"
235             body: "*"
236         };
237     }
238     // reset non-volatile configuration settings to its default values.
239     rpc SetConfigReset (google.protobuf.Empty) returns (google.protobuf.↵
    Empty) {
240         option (google.api.http) = {
241             patch: "/config/reset"
242             body: "*"
243         };
244     };
245     // reset security configuration to production state, resets root_cert↵
    , host_cert and client_cert.
246     rpc SetSecurityReset (google.protobuf.Empty) returns (google.protobuf↵
    .Empty) {
247         option (google.api.http) = {
248             patch: "/security/reset"
249             body: "*"
250         };
251     };
252     // reset periNODE into factory defaults.
253     rpc FactoryReset (google.protobuf.Empty) returns (google.protobuf.↵
    Empty) {
254         option (google.api.http) = {
255             patch: "/reset"
256             body: "*"
257         };
258     };
259
260     // mqtt message stream towards the sensor adapter periNODE, which ↵
    acts then as an actuator or data sink, respectively.
261     rpc Publish (stream Sample) returns (google.protobuf.Empty);
262     // mqtt message stream from the sensor adapter periNODE, which acts ↵
    as an data source.
263     rpc Subscribe (google.protobuf.Empty) returns (stream Sample);
264 }
265 syntax="proto3";
266
267 // package perinet;
    
```

```

268 package perinet.periNODE.api;
269
270 // representation of the available applications, implementations of ↵
    sensors respectively.
271 enum InterfaceType {
272     INVALID_APPTYPE = 0; //representation of an invalid sensor interface ↵
        value
273     RAW_SOURCE = 1; // sensor interface of any raw value, debug purpose
274     TEMPERATURE_SOURCE = 2; // sensor interface of a temperature sensor
275     PH_SOURCE = 3; // sensor interface of a ph sensor
276     DISTANCE_SOURCE = 4; // sensor interface of a distance sensor
277     DIGITAL_IO_SOURCE = 5; // sensor interface of a digital io input
278     BLIND_CONTROL_SOURCE = 6; // sensor interface of a switch control ↵
        input
279     DIGITAL_IO_SINK = 7; // sensor interface of a digital io output
280     BLIND_CONTROL_SINK = 8; // sensor interface of a discrete output to ↵
        control shutter or blinds {UP,DOWN,STOP} value range
281     ANALOG_OUT_SINK = 9; // sensor interface of a analogous output ↵
        0.0..1.0 value range
282     ACCELERATION_SOURCE = 10; // sensor interface of acceleration sensor
283 }
284
285 message NodeConfig {
286     message Interface {
287         InterfaceType type = 1; //the type of interface, the particular ↵
            interface has been configured to. Implementation specific.
288         string element_name = 2; // element name of a particular ↵
            interface, like an sensor source or an actuator sink.
289         float period_seconds = 3; // in seconds, 0 means only event based↵
            (triggered) publishing
290         uint32 samples_per_period = 4; //defines how many values shall ↵
            be sampled within a period,
291                                     //the published value will be the↵
            rounded average
292         // repeated Trigger trigger = 5;
293     }
294     string application_name = 1; //
295
296     string mqtt_broker_name = 2; //
297
298     repeated Interface configs = 3; //
299
300     // InterfaceConfig sink_interface_config = 4; // first interface
301     // InterfaceConfig source_interface_config = 5; // second interface
302 }
303 syntax="proto3";
304
305 package perinet.periNODE.api;
306
307 message SwVersion {
308     uint32 api = 1; // API compatibility incarnation
    
```

```

309
310     uint64 build = 2; // build iteration
311
312     uint32 version_number = 3; // periFIRMWARE feature level incarnation
313 }
314
315 message VersionInfo {
316     SwVersion firmware_version = 1; // periFIRMWARE version information
317     string firmware_variant = 4; // product variant, the periFIRMWARE has↔
        been aligned with. e.g. periNODE 0-10V; periSTART; periSWITCH 3-↔
        port
318 }
319
320 //changes to that node might need to be changed to NodeTXTInfo
321 message NodeInfo {
322     VersionInfo version_info = 1; // firmware version information
323     string manufacturer = 2; // manufacturer identifier
324     string hostname = 3; // host network identification name, e.g. ↔
        periNODE-<id>.local
325     string mac_address = 4; // unique mac address
326     string product_charge = 5; // production batch identifier
327     string product_part_number = 6; // product part identifier
328     string product_serial = 7; // serial number of the product
329     string product_name = 8; // calling name of the product
330     string product_version = 9; // version of the product at production ↔
        time
331     string pericore_charge = 10; // batch identifier of the included ↔
        pericore
332     string pericore_part_number = 11; // pericore part identifier
333     string pericore_serial = 12; // pericore serial number
334     string pericore_version = 13; // pericore version identifier
335 }
    
```

List of Figures

1	Perinet IP-based communication architecture	4
2	Semantic metrics of the periNODE	6
3	periNODE 0-10V	7
4	periNODE 0-10V dimensional drawing & pinout	7
5	periNODE Pt100	11
6	periNODE Pt100 dimensional drawing & pinout	11
7	periNODE Pt100 3-wire connection	12
8	periNODE GPIO	16
9	periNODE GPIO dimensional drawing & pinout	16
10	periSTART standard	22
11	periSTART standard dimensional drawing & pinout	22
12	periSWITCH 3-port	26
13	periSWITCH 3-port dimensional drawing & pinout	26
14	periLINE	31
15	periLINE dimensional drawing & pinout	31
16	Point to Point Topology	35
17	Line Topology	36
18	Edge browser security warning example	38
19	Safari browser security warning example	38
20	Firefox example security trusted chain	39
21	Safari example security trusted chain	40
22	The Web Home Page of a periNODE Pt100 with attached temperature sensor	41
23	periNODE Configuration Web Page	42
24	periNODE GPIO Configuration Web Page	43
25	periNODE configuration object retrieved from the resource /config.	45
26	JSON encoded sample object of a periNODE Pt100	47
27	JSON encoded sample object of a periNODE GPIO	47
28	Security web user interface	51
29	Configure secure MQTT	55
30	Firmware update	57
31	Perinet Reset Cable for Factory Reset	59
32	CE sign	61
33	RoHS sign	62
34	WEEE sign	63
35	periNODE installation	64
36	periSTART installation	65
37	Product marking	66

List of Tables

1	periNODE 0-10V interfaces	8
2	periNODE 0-10V mechanical specifications	8
3	periNODE 0-10V electrical specifications	9
4	periNODE 0-10V absolute maximum ratings	10
5	Recommended environment conditions	10
6	periNODE Pt100 interfaces	12
7	periNODE Pt100 mechanical specifications	13
8	periNODE Pt100 electrical specifications	14
9	periNODE Pt100 absolute maximum ratings	14
10	Recommended environment conditions	14
11	periNODE GPIO interfaces	17
12	periNODE GPIO mechanical specifications	18
13	periNODE GPIO electrical specifications	19
14	periNODE GPIO absolute maximum ratings	20
15	Recommended environment conditions	20
16	periSTART standard interfaces	23
17	periSTART standard mechanical specifications	23
18	periSTART standard electrical specifications	25
19	periSTART standard absolute maximum ratings	25
20	Recommended environment conditions	25
21	periSWITCH 3-port interfaces	27
22	periSWITCH 3-port mechanical specifications	28
23	periSWITCH 3-port electrical specifications	29
24	periSWITCH 3-port absolute maximum ratings	29
25	Recommended environment conditions	30
26	periLINE connector specification	32
27	periLINE cable specification	33
28	periLINE general specification	34
29	Configuration parameters	44
30	Sample attributes	46
31	Ordering information	68
32	Safety instructions	70
33	Glossary	75

References

- [1] *Concepts for Setting up an IoT Application*
- [2] *Starter Kit Maxi User Guide*
- [3] *Security Certificates Installation Guide*

Revision History

Revision	Date	Author(s)	Description
1.0	September 14, 2021	Soeren Hoeckner	Initial release