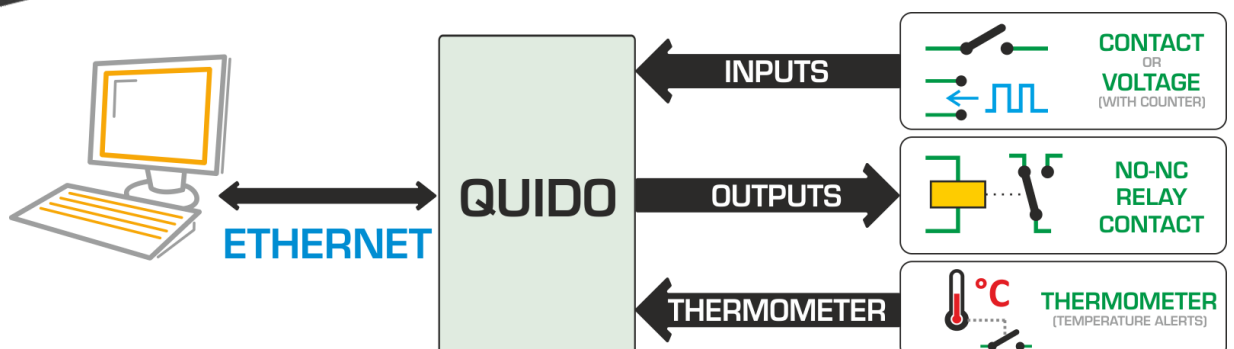
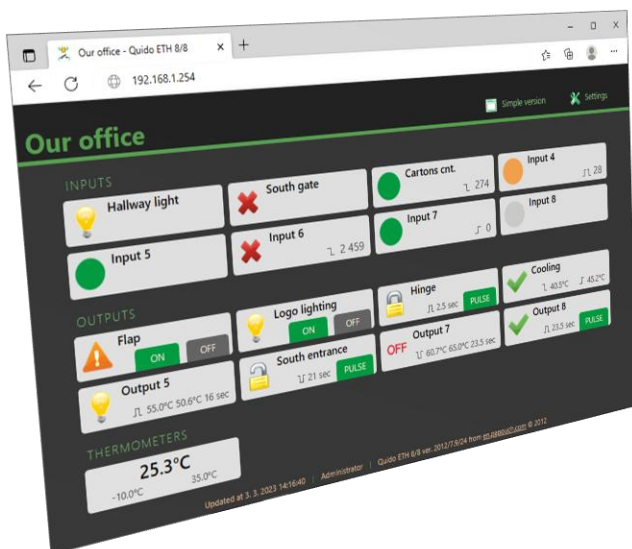




# Quido ETH

Family of I/O modules for output management, input status monitoring and temperature measurement

Communication over Ethernet



# Quido ETH

## Data Sheet

Created: 03/15/2005

Last update: 1/6/2026 13:35

Number of pages: 48

© 2026 Papouch s.r.o.

---

## Papouch s.r.o.

Address:

**Strasnicka 3164  
102 00 Prague 10  
Czech Republic**

Phone:

**+420 267 314 267**

Web:

**en.papouch.com**

Mail:

**info@papouch.com**



**TABLE OF CONTENTS**

Table of Contents.....	3
Firmware changes overview .....	4
About device .....	6
Features .....	7
Basic functions.....	7
Practical applications .....	7
I/O Quido module – control options.....	7
Available designs.....	8
Available accessories .....	10
First Connection.....	11
Input and Output Wiring .....	13
Inputs.....	13
Outputs.....	14
Principle of input status monitoring .....	16
Configuration and Control .....	16
Web interface – Control .....	17
Wix software .....	23
Spinel protocol.....	24
Modbus TCP protocol .....	24
Configuration via Telnet protocol.....	24
Connection .....	24
IP address is not known.....	24
IP address is known.....	25
Telnet main menu.....	25
Server .....	26
Factory Defaults .....	26
Exit without save .....	27
Save and exit .....	27
XML file.....	27
Inputs: din .....	27
Outputs: dout .....	28
Thermometers: temp.....	28
status.....	29
HTTP GET .....	29
Sending the current status to a remote server .....	29
Method of sending the request.....	30
Request parameters .....	30
Quido setup in response to GET .....	31
Quido control by HTTP GET.....	31

Response to HTTP GET .....	33
Sending of e-mails .....	34
Change in the status of input or output.....	34
Exceeded temperature limits.....	34
Sensor error.....	34
Test message to verify the settings.....	34
SNMP .....	35
Temperature watch modes .....	35
FAQ .....	37
Which settings are required for Quido to work in my network? .....	37
How to find out the IP address of Quido? .....	37
Reset .....	38
Indicators .....	39
Technical parameters.....	40
Inputs.....	40
Outputs.....	40
Thermometer .....	40
Control interface.....	41
Connectors .....	42
Other parameters.....	42

## Firmware changes overview

---

*How to find out the firmware version of your device?*

- On the web interface, on the Info tab in the Firmware version line, there are two entries separated by a slash - e.g. **8.7/17**. Here it is the first entry before the slash, i.e. **8.7**.

### Version 8.8 (2/2026)

- [Records for CORS](#) added to XML file headers.

### Version 8.7 (8/2025)

- Unification of Modbus TCP and RTU protocols.

### Version 8.3 (6/2024)

- Fixed http get behaviour: the Content-Length in the get response is now checked. If it is missing, only the header is received and the connection is actively terminated.
- Fix link to the simplified version of the site when accessing the device site for the first time.

### Version 7.9 (3/2023)

- Option to disable Telnet, Firmware Upgrade and SNMP.
- Added configurable TCP timeout - time without communication on TCP, after which the established connection is interrupted.
- [Reset via push button](#) extended with the possibility of obtaining an IP address via DHCP.
- Responsive web interface.
- Increased max length of passwords and clarification of field descriptions in configuration.
- Added documentation of [controlling outputs by measured temperature](#).

**Version 7.8 (8/2022)**

- New http get parameter that can be used to set multiple outputs at once for a certain period of time.

**Version 7.4 (12/2021)**

- SNMP trap on state change is sent even if the state of outputs changes.

**Version 6.9 (5/2020)**

- When communicating with the Spinel protocol via TCP/UDP, it is possible to set the automatic message about the change on the inputs to be sent for each input separately. The function is activated by checkbox on the Network tab.

**ABOUT DEVICE**

Quido ETH is a family of I/O modules with digital inputs, outputs, and a thermometer. Inputs are designed to connect voltage or contacts – can distinguish two states (0 and 1). The outputs are relays with a changeover contact. The thermometer can be connected to a cable of up to 15 meters and is able to measure temperatures in the range from -55 to +125 °C. Quidos connect directly to a LAN (Ethernet) and can be fully controlled and managed through a web interface.

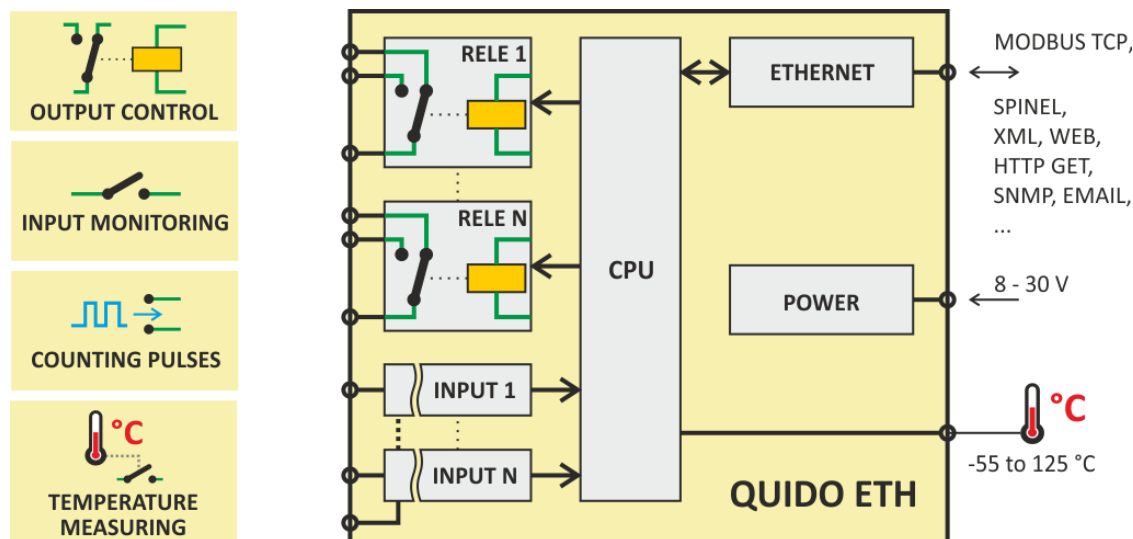


Fig. 1 – Basic functions and block diagram of Quido ETH

Name	Number of inputs To connect contacts, voltage and pulse outputs, etc.	Number of outputs Power relay with one changeover contact	Number of thermometers Sensor for -55 to +125 °C on a cable of up to 15 m
Quido ETH 3/0	3	0	1
Quido ETH 3/0 B <sup>1</sup> (in aluminium case)	3	0	1
Quido ETH 10/1	10	1	1
Quido ETH 4/4	4	4	1
Quido ETH 8/8	8	8	1
Quido ETH 30/3	30	3	1
Quido ETH 60/3	60	3	1
Quido ETH 100/3	100	3	1
Quido ETH 0/2	0	2	1
Quido ETH 2/16	2	16	1
Quido ETH 2/32	2	32	1

Tab. 1 - List of standard variants with Ethernet interface

<sup>1</sup> This type of Quido has a separate documentation.

## Features

---

- Galvanically isolated digital inputs for voltage or contacts.
- SPDT Relay outputs.
- Thermometer with a measuring range of -55 to +125 °C.
- Connects directly to LAN (Ethernet).
- LED indication of power, communication, and the status of inputs and outputs.
- External power supply 8 to 30 V.
- Controlled:
  - by standard protocols [Modbus TCP](#), [SNMP](#), [Http GET](#), [Spinel](#),
  - via a [web interface](#):
    - For all modern web browsers including OS iPhone and Android.
    - Separate access for viewing and for configurations.
    - Optional status symbols of inputs and outputs.
  - by [Wix software](#).

## Basic functions

---

- Reads the current status of inputs
- Automatically sends information (email, HTTP GET, ...) about changes on the inputs
- Counts pulses on the inputs or changes in the status of the input (for first 60 inputs) <sup>2</sup>
- Control of output relays with a changeover contact
- Configuration of the outputs for a defined period
- Thermometer: measuring range of -55 to +125 °C
- Temperature monitoring (the relay closes when temperature changes, closes for a predefined period when a specified temperature has been reached; sends an e-mail or HTTP GET)

## Practical applications

---

- The state of the sensors can be viewed from a PC.
- Computer monitoring of safety sensors and control of various devices.
- Monitoring of the status of door contacts and remote opening of the front door.
- Counting the number of passing people.
- Small-scale automation of objects and manufacturing processes.
- Autonomous control or remote control based on the measured temperature.
- Integration of numerous inputs and outputs in different locations.

## I/O Quido module – control options

---

(For more information about these options see page 16.)

- Via a web interface.
- Standard industrial network protocol [Modbus TCP](#).
- Network protocol [SNMP](#).
- Easy control and sending and notifications of the current status via [Http GET](#) queries.

---

<sup>2</sup> Counters states are not stored when power goes down or after reset. (Tip: [Papago 5HDI 1DO ETH](#) does remember counters even after power failure and has other functions for impulse counting.)

- Access to all information in Quido via [XML](#).
- Via [Spinel protocol](#) (.NET library [Spinel.NET](#) in C# [on GitHub](#), Czech documentation).
- Via [Wix](#) Software.
- Upon your request, we are able to modify the protocol in Quido or implement your own protocol. Feel free to contact us.

## Available designs

### Input voltage levels:

- 4.5 to 9 V
- 7 to 28 V (*standard design*)

### Temperature sensor design:

- Sealed in shrinkable coating (*standard design*)



Fig. 2 – sensor in shrinkable coating

- In metal housing



Fig. 3 – sensor in metal housing

### Sensor cable length:

- 3 m (*standard design*)
- 1 m, 5 m, 10 m, 15 m

### Thermal resistance of the sensor cable:

- -10 to +70 °C (*standard design*)
- -60 to +200 °C – silicone cable; blue color

### Cover and mounting:

- Electronics board only (*standard design*)

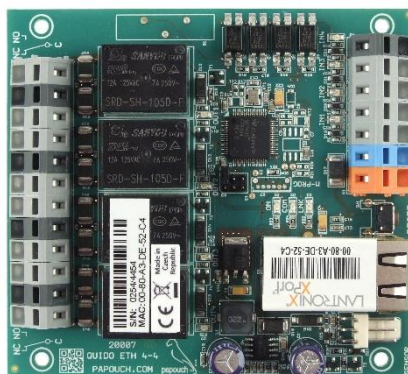
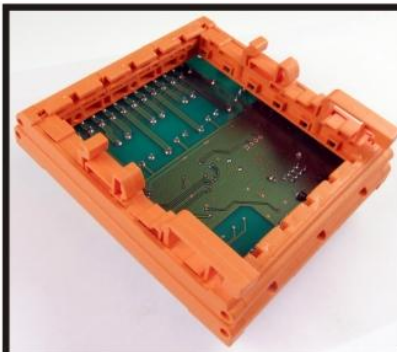


Fig. 4 – example of standard design (here: Quido ETH 4/4)

- All mounting options are illustrated below (here: Quido ETH 4/4):



Board with a DIN rail



Board with a DIN rail (below)



Board with a plexi cover



Board with a plexi and DIN



Board in a box with plexi



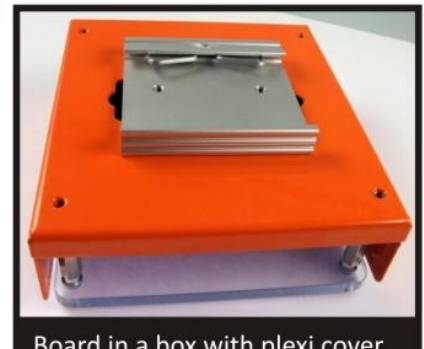
Board in a box



Board in a box (below)



Board in a box with DIN (below)



Board in a box with plexi cover and DIN rail mount (bottom)

### Examples of functions that can be added upon customer's request:

- Outputs will close depending on the preset combination of inputs.
- Two modules connected in synchronous operation. Activation of an input on one module causes the switch on the relay on the other module and vice versa. In this way it is possible to transmit states of multiple signals bidirectionally. (This variant can be ordered as [Quido Duplex ETH.](#))
- Modifications allowing you to use your communication protocol.
- Memory for a greater number of changes in the input state.
- ... we will be happy to add features tailored to your application.

*Do not hesitate to contact us if you have any other requirements concerning the design and functions of Quido ETH.*

**Available accessories**

---

**Power supply unit for Quido – AC adapter**

Switched power supply unit for 100 to 240 V; AC adapter design.

*Available versions: 12V/0,5A; 12V/2A; 24V/1A*



**Power supply unit for Quido – mountable to DIN rail**

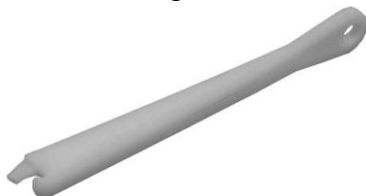
Switched power supply unit for 100 to 240 V; mountable to 35 mm DIN rail.

*Available versions: 12V/0,5A; 12V/2A; 12V/4,5A; 24V/1,5A; 24V/2,5A*



**Lever for Wago 236 terminals – plastic**

a tool enabling easy connection of wires to Wago 236 terminals (standard part of delivery)



**Lever for Wago 236 terminals – metal**

a tool enabling easy connection of wires to Wago 236 terminals (sturdy metal design)



## FIRST CONNECTION

Tip: Quido is supplied with a plastic lever enabling easy connection of wires. The method of using the lever is illustrated in the figure below.<sup>3</sup>

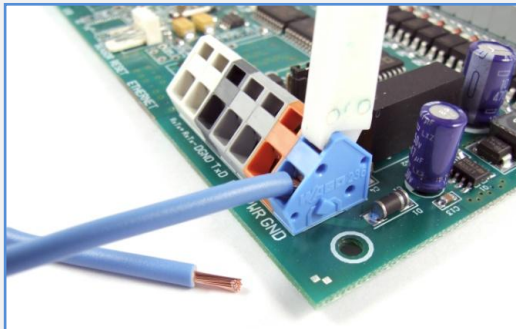


Fig. 5 – how to use the level for easy connection of wires

- 1) Connect direct-current supply voltage within the range of 8 to 30 V to the PWR (+ orange) and GND (– blue) terminals. Quido is equipped with protection against reverse polarity of power supply. After connecting the power supply, the PWR LED on the board lights up.



Fig. 6 – example of terminals on Quido ETH 4/4 (from left): sensor, ethernet, power supply and inputs

- 2) Now it's time to connect Quido to the computer network.
  - a. Use a standard (non-crossover) cable to connect Quido to the net switch.<sup>4</sup>
  - b. If the range of IP addresses in your network is not compatible with the default IP address (**192.168.1.254**) and net mask (**255.255.255.0**) of Quido, set a suitable address by using [Ethernet Configurator](#) or use [Reset button](#) to assign an IP address by DHCP server.

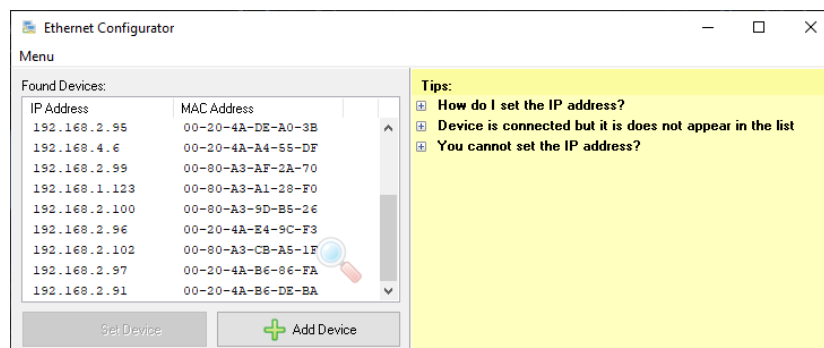


Fig. 7 – Ethernet Configurator for easy configuration of IP address

<sup>3</sup> You can also use a small flat screwdriver.

<sup>4</sup> If you wish to connect Quido directly to a single PC, use a crossover cable.

- c. After setting the address, you can connect to Quido via a web browser. The web interface is available directly at the address of Quido.
- 3) If your Quido enables connection of a thermometer, the board includes a connector (with three terminals - see Fig. 8) labelled SENSOR (or TEMP). Here you can connect the temperature sensor. (The thermometer needs to be connected only if you want to measure temperature or use one of the functions of temperature monitoring.)<sup>5</sup>

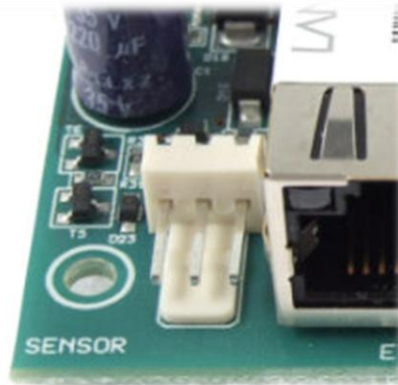


Fig. 8 – connector for the thermometer (here: Quido ETH 4/4)

- 4) Connect the input and output terminals as required. For detailed description of available options see page 13.
- 5) For information about the web interface, control software and communication protocols that can be used to control and communicate with Quido see page 16.

**Configuration jumpers**

The configuration jumpers marked as Duplex, Info and Modbus on Ethernet versions are not used in normal operation.

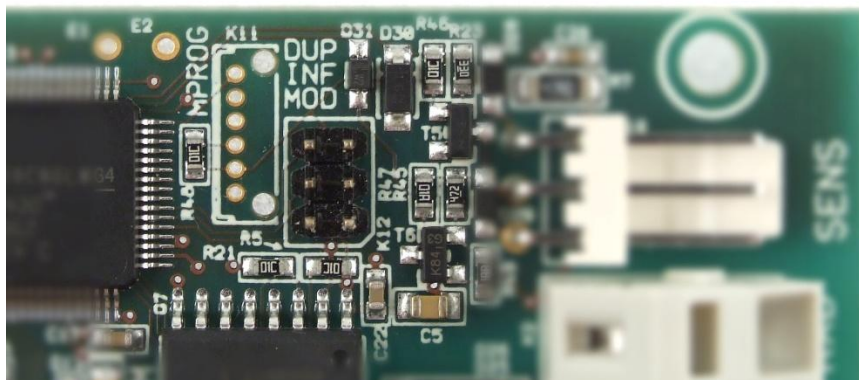


fig. 9 - example of jumpers on Quido

<sup>5</sup> If the electronics board contains the connector but you do not have a temperature sensor, the sensor can be ordered separately.

## INPUT AND OUTPUT WIRING

### Inputs

The inputs can be controlled by applying voltage or connecting a contact.

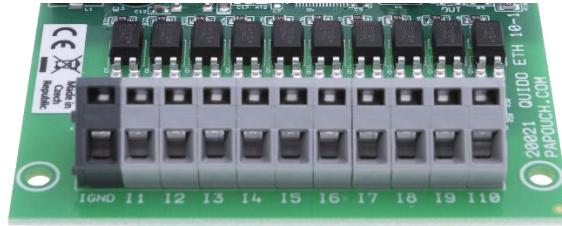


Fig. 10 – input terminal block of Quido ETH 10/1

Each input is connected as shown in Fig. 11. IGND ground is galvanically isolated from device GND.

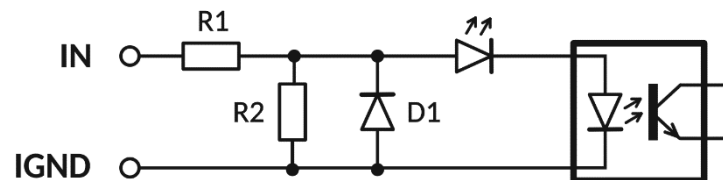


Fig. 11 – wiring of the input circuit

### Input for contact

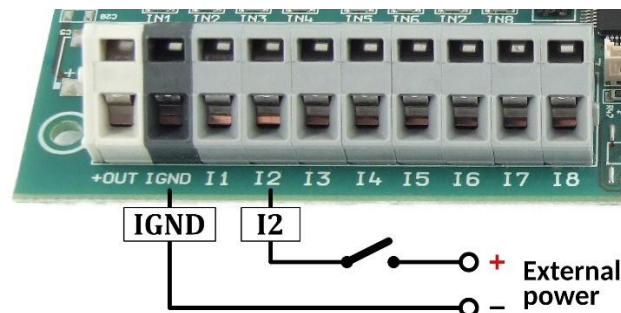


Fig. 12 – contact wiring by connecting voltage from external source (Quido without mounted DC/DC converter – see title +OUT Terminal on page 13)

If same source is used as external power supply for inputs as for device, galvanic isolation of inputs is broken and inputs are then galvanically connected to Quido via source ground!

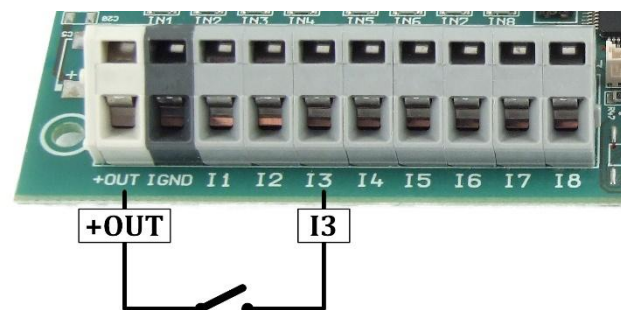


Fig. 13 – contact connection (Quido with mounted DC/DC converter – see title +OUT Terminal on page 13)

### +OUT TERMINAL

The DC/DC converter for supplying galvanically isolated inputs can be mounted on Quidos that have a terminal marked +OUT. In this case, the Quido can be ordered in a variant "Power supply for inputs: from +OUT terminal" and the inputs can be wired as shown on Fig. 13.

Quido in "+OUT power supply" version (and with the DC/DC converter installed) can be ordered on e-shop for these types (updated 10/2022):

- Quido RS 8/8

The version with +OUT terminal and fitted converter can also be ordered as a custom modification on individual request for quotation.

**Voltage input**

Voltage input connections are shown in the following picture.

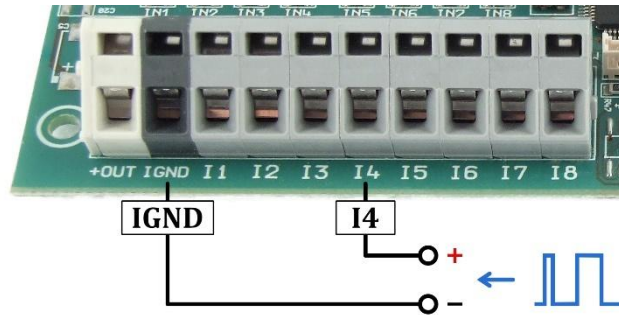


Fig. 14 – Voltage input (to connect pulse output for example)

**Outputs**

Each output has its own relay with switching contact (SPDT) – maximum is 60 V AC or 85 V DC!<sup>6</sup>

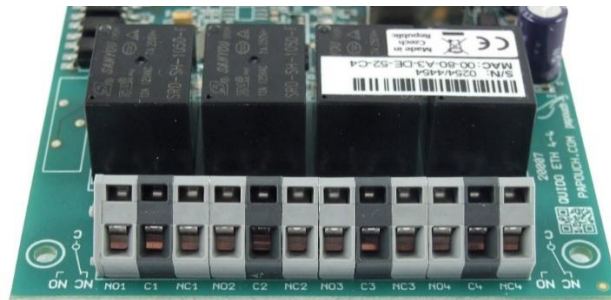


fig. 15 – Relay terminals with switching contacts

Output is connected in idle state as follows:

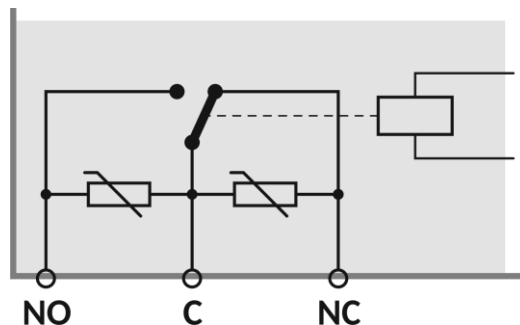


fig. 16 – relay contact connections including protection varistors

<sup>6</sup> Example of switching high voltages is shown on the next page.

**Example of output connection used for higher voltages (e.g. 230 V)**

For higher voltages of e.g. 230 V the Quido outputs need to be strengthened by an external relay or a contactor.<sup>7</sup> An example of wiring is shown in the figure and diagram below.

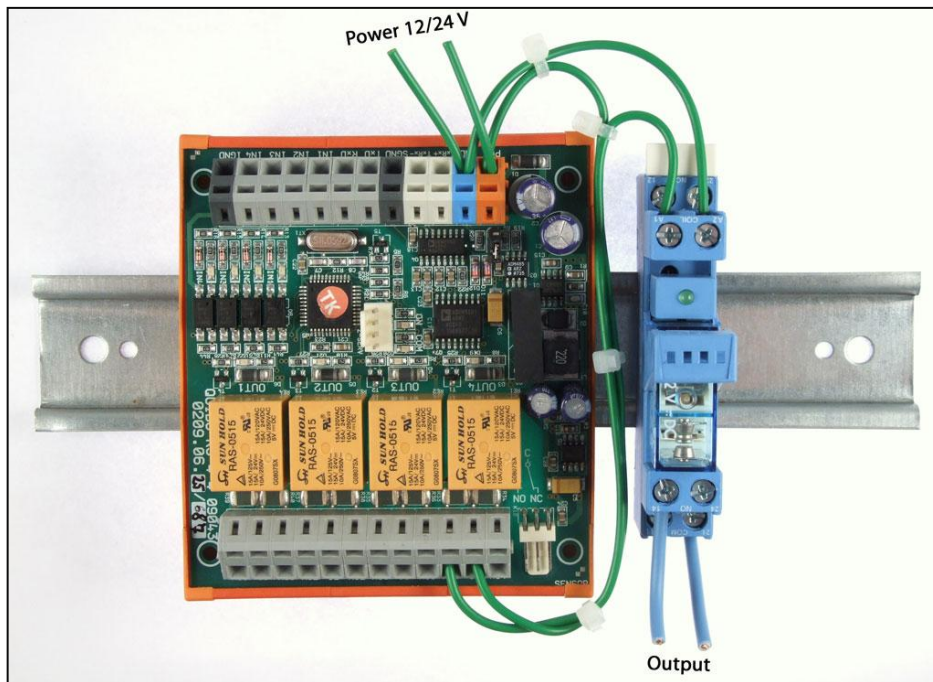


Fig. 17 – example of Quido wiring for 230 V AC

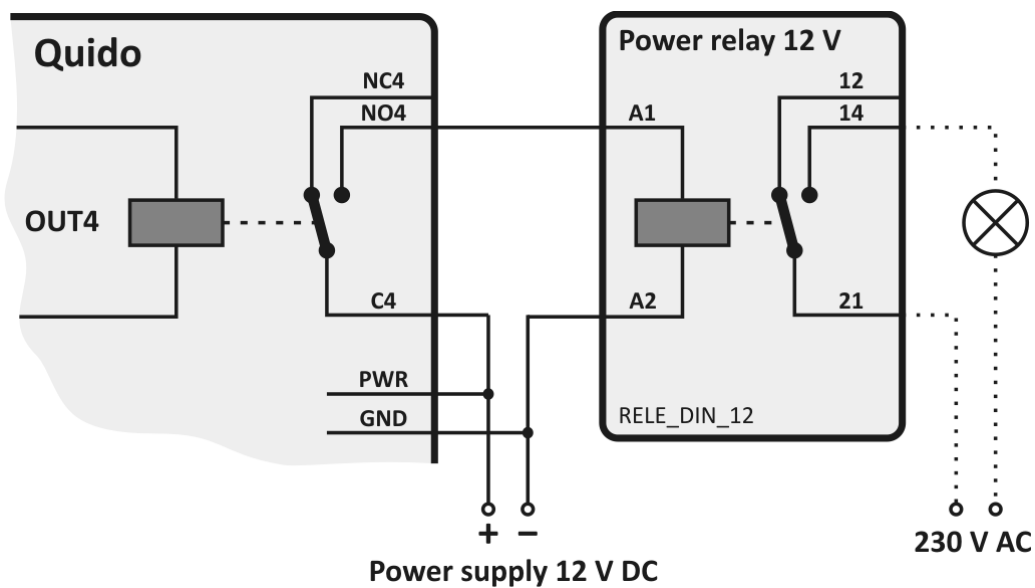


Fig. 18 – block diagram for the wiring shown in Fig. 17 and an example of switching a 230 V bulb

<sup>7</sup> Power relay for up to 300 V AC, illustrated in the figure, can be ordered using the following codes: RELE\_DIN\_5 (for 5 V supply), RELE\_DIN\_12 (12 V), RELE\_DIN\_24 (24 V), RELE\_DIN\_48 (48 V).

**Principle of input status monitoring**

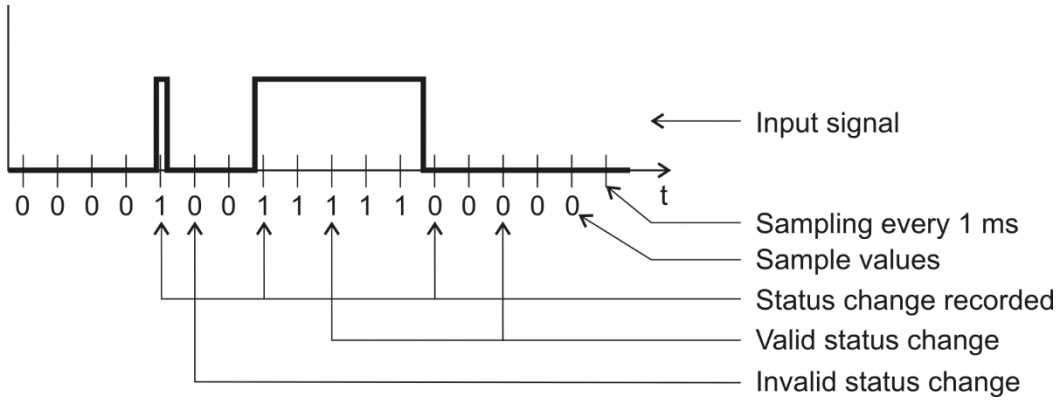


Fig. 19 – principle of input status monitoring

The input is sampled at intervals of 1 ms. The input status is considered valid if the same value has been read several times in a row. The number of times in a row that the same value must be read can be configured via a web interface in the *Inputs* tab, line *Number of samples for status change*. The default value is 20 – i.e. 20 ms. (The value of 20 ms is fixed in Quido ETH 3/0B and cannot be changed.)

When a change in input status occurs, a message is sent automatically (if enabled in the Spinel protocol). If there is a counter is enabled for the input, it is incremented according to its settings.

**CONFIGURATION AND CONTROL**

After being connected, Quido can be controlled and configured by one of the following ways:

	monitoring	control	configuration
Web interface	✓	✓	✓
Wix software	✓	✓	
Ethernet Configurator software			✓
Spinel protocol	✓	✓	✓
MODBUS TCP protocol	✓	✓	
SNMP protocol	✓	✓	
HTTP GET queries	✓	✓	
via XML	✓		
Telnet protocol			✓
E-mail	✓		

## WEB INTERFACE – CONTROL

Web interface allows complete monitoring, control and configuration of Quido. The web interface requires JavaScript to be enabled. The interface is optimized for Mozilla Firefox, Google Chrome and Microsoft Edge (Chromium core).

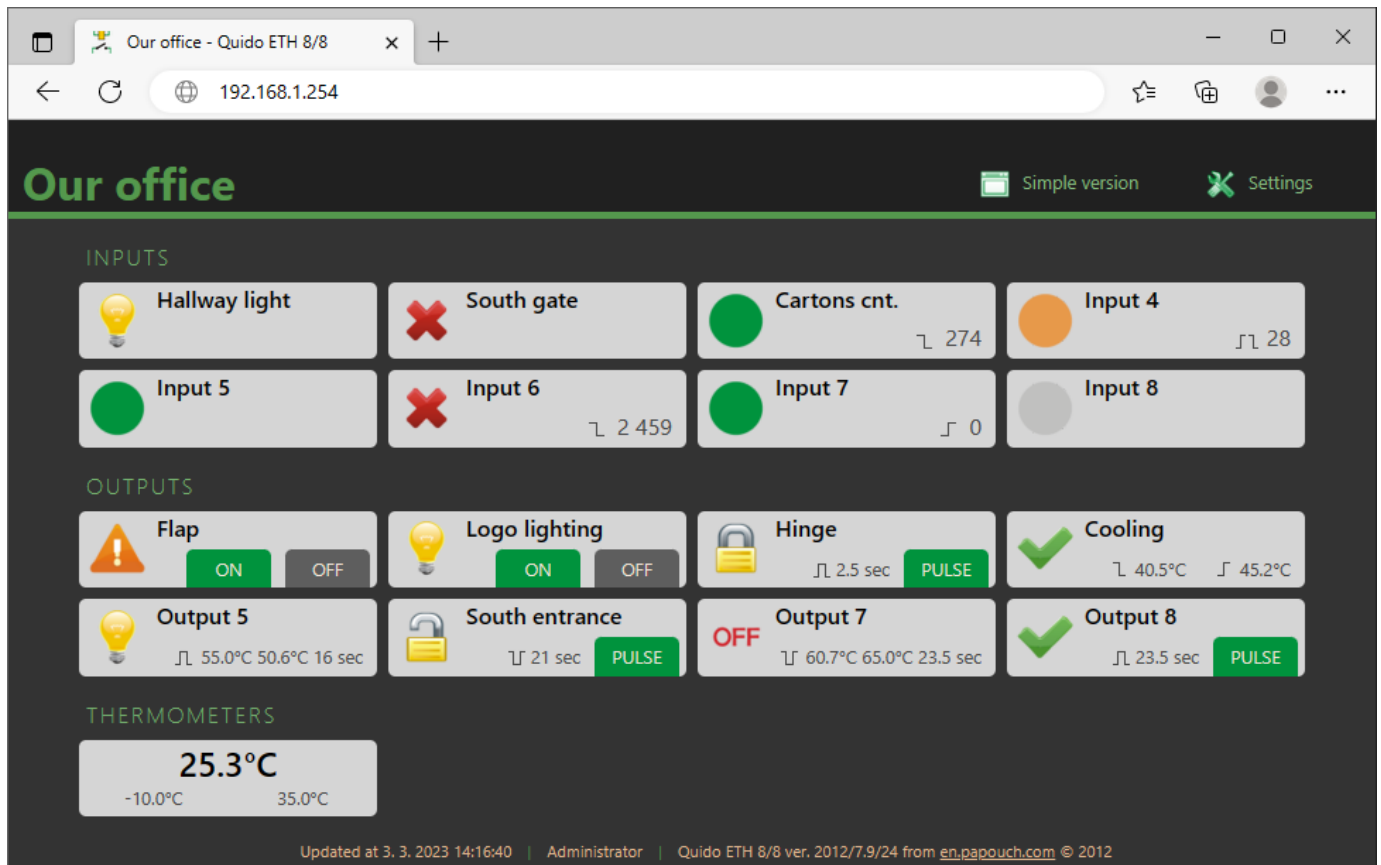


Fig. 20 – Main web page in Quido ETH 8/8

Access to the web interface can be secured with passwords. There are two levels of security: First can be used to access the main page where is available the current status of Quido, outputs can be controlled and counters can be reset. Second is used to enter into configuration.

Help for all setting items is available by hovering the cursor over the setting item.

HTTP GET settings	
WEB server's address	<input type="text" value="iot.example.com"/>
WEB Port	<input type="text" value="80"/>
Path	<input type="text" value="/api/connectors/quido/overheight"/>
Interval of periodic sending	<input type="text" value="60"/>
Send HTTP GET upon changes	<input type="checkbox"/>
Example: HTTP GET from Quido ETH 8/8: www.server.net/script.php?mac=00-20-4A-B4-8D-F7&name=Office&ins=01101010&outs=00100010&tempS=0&	
Tip: If you are sending HTTP GET to a server in different network, you also need to have set Gateway IP address in the Network panel.	

Enter the local path to the receiving script. Max. 100 characters. Example: /api/driver.php  
 Allowed characters: A-Za-z0-9-.\_~:/?#[]@!\$&'()\*+;,%#

fig. 21 - Tooltips

Network	Security	E-mail	SNMP	Sending	Inputs	Outputs	Thermometers	Other	Info	
<b>Network settings</b>										
DHCP <input checked="" type="checkbox"/>										
Device's IP address	<input type="text" value="192.168.1.45"/>									
Netmask	<input type="text" value="255.255.255.0"/>									
Gateway IP address	<input type="text" value="0.0.0.0"/>									
DNS server's IP address	<input type="text" value="0.0.0.0"/>									
WEB port	<input type="text" value="80"/>									
ModBus Port	<input type="text" value="512"/>									
Local port for TCP/UDP connection	<input type="text" value="10001"/>									
TCP timeout	<input type="text" value="60"/>									
<b>Device mode <sup>?</sup></b>										
Communication mode	<input type="text" value="WEB"/>									
Send message about inputs changes individually (TCP/UDP only) <input checked="" type="checkbox"/>										
<b>Remote device for TCP/UDP connection <sup>?</sup></b>										
Remote IP address	<input type="text" value="0.0.0.0"/>									
Remote port	<input type="text" value="0"/>									
									Reset	Save

fig. 22 - Network settings

Network	Security	E-mail	SNMP	Sending	Inputs	Outputs	Thermometers	Other	Info
<b>Security settings</b>									
User password	<input type="text"/>								
Confirm user password	<input type="text"/>								
Administrator's password	<input type="text"/>								
Administrator's password for confirmation	<input type="text"/>								
Current Administrator's password	<input type="text"/>								
Disable Telnet (advanced users only!)	<input type="checkbox"/>								
Disable fw upgrade (advanced users only!)	<input type="checkbox"/>								
									Save

fig. 23 - Security settings

Network	Security	E-mail	SNMP	Sending	Inputs	Outputs	Thermometers	Other	Info
<h2>E-mail settings</h2>									
Activate the e-mails sending globally <input type="checkbox"/>									
SMTP server's name			<input type="text" value="mail.example.com"/>						
SMTP Port			<input type="text" value="587"/>						
Sender's e-mail address			<input type="text" value="quido@example.com"/>						
Recipient's e-mail address			<input type="text" value="it@example.com"/>						
Send e-mails upon changes			<input checked="" type="checkbox"/>						
Send e-mail when the threshold is exceeded			<input checked="" type="checkbox"/>						
<h3>SMTP authorization</h3>									
SMTP server requires verification			<input type="checkbox"/>						
Verification name			<input type="text"/>						
Verification password			<input type="text"/>						
Re-enter the password			<input type="text"/>						
<input type="button" value="Test"/> <input type="button" value="Save"/>									

fig. 24 - Email settings

Network	Security	E-mail	SNMP	Sending	Inputs	Outputs	Thermometers	Other	Info
<h2>SNMP settings</h2>									
Enable SNMP			<input checked="" type="checkbox"/>						
SNMP manager's IP address			<input type="text" value="192.168.1.111"/>						
Read community name			<input type="text" value="public"/>						
Write community name			<input type="text" value="private"/>						
<h3>SNMP traps</h3>									
Enable traps sending			<input type="checkbox"/>						
Send SNMP trap when the threshold is exceeded			<input type="checkbox"/>						
Send SNMP trap upon I/O change			<input checked="" type="checkbox"/>						
<input type="button" value="Save"/>									

fig. 25 - SNMP settings

Network	Security	E-mail	SNMP	Sending	Inputs	Outputs	Thermometers	Other	Info
---------	----------	--------	------	---------	--------	---------	--------------	-------	------

### HTTP GET settings

WEB server's address

WEB Port

Path to script

Interval of periodic sending

Send HTTP GET upon changes

Example: HTTP GET from Quido ETH 8/8:  
 www.server.net/script.php?mac=00-20-4A-B4-8D-F7&name=Office&ins=01101010&outs=00100010&tempS=0&tempV=21.8&cnt2=235&cnt6=126

Tip: If you are sending HTTP GET to a server in different network, you also need to have setGateway IP address in the Network panel.


fig. 26 - Sending settings

Network	Security	E-mail	SNMP	Sending	Inputs	Outputs	Thermometers	Other	Info
---------	----------	--------	------	---------	--------	---------	--------------	-------	------

Choose the input:

### Input settings

Input name

Pictures for ON and OFF states  
 ON:  OFF: 

Hide input

Watch for changes

Counter mode

For all inputs: Number of samples to change the input state (advanced users only)

fig. 27 - Input settings



Network	Security	Email	SNMP	Sending	Inputs	Outputs	Thermometers	Other	Info
Choose the output: ← <span>Output 2: Entrance gate</span> →									
<b>Outputs settings</b>									
Output name	<input type="text" value="Entrance gate"/>								
Pictures for ON and OFF states	ON:  OFF: 								
Hide output	<input type="checkbox"/>								
Watch for changes	<input type="checkbox"/>								
Output mode	<input type="text" value="Pulse - positive pulse"/>								
Output pulse length	<input type="text" value="2 sec"/>								

fig. 28 - Output settings: Pulse mode <sup>8</sup>



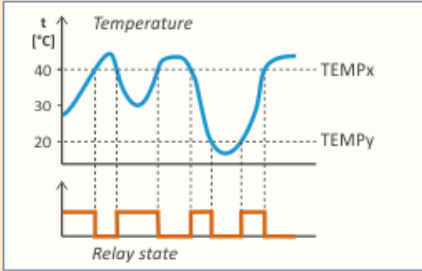
Network	Security	Email	SNMP	Sending	Inputs	Outputs	Thermometers	Other	Info
Choose the output: ← <span>Output 1: Fridge therm.</span> →									
<b>Outputs settings</b>									
Output name	<input type="text" value="Fridge therm."/>								
Pictures for ON and OFF states	ON:  OFF: 								
Hide output	<input checked="" type="checkbox"/>								
Watch for changes	<input checked="" type="checkbox"/>								
Output mode	<input type="text" value="Temperature watch: In limits ON"/>								
TEMPx	<input type="text" value="40"/>								
TEMPy	<input type="text" value="20"/>								
Sensor detached action	<input type="text" value="Keep without change"/>								
									
<input type="button" value="Save"/>									

fig. 29 - Output settings: Temperature watch mode <sup>8</sup>

<sup>8</sup> See also Temperature watch modes on page 34.

Network	Security	E-mail	SNMP	Sending	Inputs	Outputs	Thermometers	Other	Info
<b>Thermometer settings</b>									
Hide thermometer							<input type="checkbox"/>		
Watch temperature thresholds (limits)							<input checked="" type="checkbox"/>		
Upper limit						<input type="text" value="30.2"/>			
Lower limit						<input type="text" value="-6,8"/>			
Hysteresis						<input type="text" value="30"/>			
Periodic sending upon an event						<input type="text" value="60"/>			
<input type="button" value="Save"/>									

fig. 30 - Thermometer settings

Network	Security	E-mail	SNMP	Sending	Inputs	Outputs	Thermometers	Other	Info
<b>Other settings</b>									
Name of the device						<input type="text" value="Our office"/>			
Language						<input type="text" value="English"/> ▼			
Temperature units						<input type="text" value="Fahrenheit [°F]"/> ▼			
<input type="button" value="Save"/>									

fig. 31 - Other settings

Network	Security	E-mail	SNMP	Sending	Inputs	Outputs	Thermometers	Other	Info
<b>Information about the device</b>									
Device type: <i>Quido ETH 8/8</i>									
S/N: <i>0229/0426</i>									
MAC address: <i>00-20-4A-B4-8A-F1</i>									
Firmware version: <i>7.9/24 (Created 08.02.2023 14:16:22)</i>									
Core: <i>Quido ETH 8/8; v0229.02.79; f66 97; t1</i>									
Browser: <i>Firefox 109</i>									
<b>Supplier of the device</b>									
Name: <i>Papouch s.r.o.</i>									
Web site: <a href="http://en.papouch.com">en.papouch.com</a>									
<b>Links</b>									
XML file containing current configuration: <a href="#">settings.xml</a>									

fig. 32 - Device info

## WIX SOFTWARE

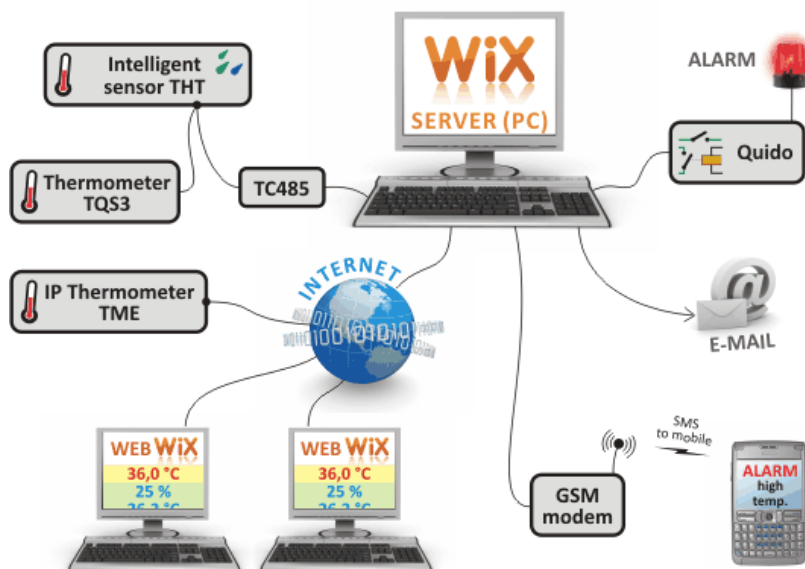
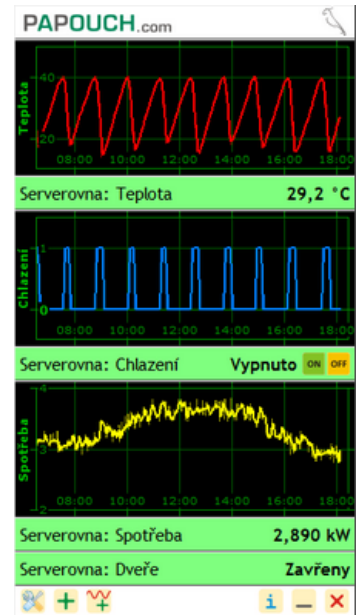


Fig. 33 – universal Wix software



[Wix software](#) is able to work with most of our equipment so Quido can be easily integrated. It can be used to monitor the status of inputs, outputs, counters, and the thermometer. Besides, the program gathers and combines values from all devices connected to it and uses them to perform various actions: to control outputs, send e-mails, display alerts, send SMS via a connected modem, etc.

## SPINEL PROTOCOL

Spinel is the default protocol that Quido uses for communication. It is available both in ASCII and binary versions. It can be used to fully control and configure Quido. All commands are well documented including examples for each instruction. [Spinel Terminal](#) and [online parser](#) is also available for easy debugging.

### Spinel.NET C# library

Spinel protocol can also be implemented using [Spinel.NET](#) library designed for .NET. It is freely available [on GitHub](#). (Czech documentation only.)

Example of turning output OUT3 on for 5 seconds: ([more here](#)):

```
if (MyQuido.CmdSetOutput(3, true, 10))
    Console.WriteLine("Ok");
else
    Console.WriteLine("Error");
```

Example of reading impulses form the IN2 counter: ([more here](#)):

```
if (MyQuido.CmdGetCounter(2, out int counter))
    Console.WriteLine($"Counter value is {counter}.");
```

## MODBUS TCP PROTOCOL

Quido can also communicate via the industry standard protocol Modbus TCP. Complete Modbus TCP documentation is available for download on Quido website at [papouch.com](http://papouch.com).

## CONFIGURATION VIA TELNET PROTOCOL

### Connection

#### IP address is not known

*It is recommended that the IP address should be set using the EthernetConfigurator software (for more information see page 11).*

- 1) Open the window of the cmd command. (In the Windows OS select Start/Run, enter `cmd` in the provided line and click Enter.)
- 2) Make the following entries into the ARP table:
  - a. Type `arp -d` and confirm by Enter. This will delete the current ARP table.
  - b. Use the following command to assign the IP address of 192.168.1.254 to the MAC address of the module:
 

```
arp -s [new_ip_address] [MAC_address_of_device]
```

example: `arp -s 192.168.1.254 00-20-4a-80-65-6e`
- 3) Now open Telnet. (Type in `telnet` and press Enter.<sup>9</sup>)

<sup>9</sup> In OS Windows 10, the client for Telnet is not a standard part of the system. Installation procedure:

a) Enter "windows features" to search bar in Windows 10 and choose *Turn Windows features on or off*.

- 4) Enter `open [new_ip_address] 1` and confirm.
- 5) After a while, the terminal will display an error message saying that connection failed. However, this step is necessary for the module to enter the IP address into its ARP table.
- 6) Connect to the IP address of the module. (Type in `open [IP address in dotted format] 9999` and click Enter.)
- 7) So far you have only entered the configuration mode of the module. The IP address has not yet been set. It must be set in the menu **Server Configuration > IP Address**. If you close the configuration mode without saving the settings and IP address configuration, the whole procedure must be repeated!
- 8) If the entered IP address is valid, the converter displays an introductory text ending with:  
**Press Enter for Setup Mode**  
Press Enter within 3 seconds, otherwise the configuration mode will close.
- 9) The device will display a preview of its settings.
- 10) The preview ends with a paragraph called "Change setup:" which lists the groups of parameters that can be configured. Network parameters can be changed in the "Server" section where you can set a new network address and other parameters.

### IP address is known

- 1) In OS Windows choose Start/Run, enter `telnet` in the provided line and press Enter.<sup>9</sup>
- 2) Connect to the IP address of the module. (Type in `open [IP address in dotted format] 9999` and press Enter.)
- 3) If the entered IP address is valid, the converter displays an introductory text ending with:  
**Press Enter for Setup Mode**  
Press Enter within 3 seconds, otherwise the configuration mode will close.
- 4) The device will display a preview of its settings.
- 5) The preview ends with a paragraph called "Change setup:" which lists the groups of parameters that can be configured. Network parameters can be changed in the "Server" section.

### Telnet main menu

Individual items can be chosen using the numbers written next to them. Choose the required number and press Enter.

The menu structure is as follows:

```
Change Setup:
 0 Server
  ...
 7 Defaults
 8 Exit without save
```

b) Windows features window opens, check Telnet Client and click OK.

9 Save and exit

Your choice ?

## Server

Basic Ethernet settings.

This section contains the following parameters:

**IP Address** : (192) . (168) . (001) . (122)

**Set Gateway IP Address (N) ?**

**Netmask: Number of Bits for Host Part (0=default) (16)**

**Change telnet config password (N) ?**

### IP Address

*(IP address)*

IP address of the module. The digits of the IP address must be entered one by one and separated by Enter.

Default value: 192.168.1.254

### Set Gateway IP Address

*(set the IP address of the gateway)*

#### Gateway IP addr

*(IP address of the gateway)*

In "Set Gateway IP Address" enter "Y" to change the Gateway IP address. The system then prompts you to change the Gateway IP address. The digits of the IP address must be entered one by one and separated by Enter.

### Netmask

*(network mask)*

Here you specify the number of bits of the IP address that make up the network part.

The Netmask is set as a number of bits determining the range of available IP addresses of the local network. If, for example, value 2 is entered, the structure of the Netmask is 255.255.255.252. The entered value specifies the number of bits from the right. The maximum is 32.

Default value: 8

Example:

The mask 255.255.255.0 (binary representation: 11111111 11111111 11111111 00000000) corresponds to number 8.

The mask 255.255.255.252 (binary representation: 11111111 11111111 11111111 11111100) corresponds to number 2.

### Change telnet config password

*(Set the password for Telnet)*

#### Enter new Password

*(Enter the password for Telnet)*

This parameter is used to set a new password which is required prior to any configuration via Telnet.

In item "Change telnet config password" enter "Y" to change the password. The system then prompts you to change the password.

## Factory Defaults

By pressing number 7 the device restores the default settings.

The default setting means that all parameters will return to their initial factory settings. The IP address remains unchanged; the web interface port is set to 80.

## Exit without save

To close the configuration mode without saving the changed parameters

## Save and exit

This option saves the changes. If any parameter has been changed, the device is restarted. The restart takes several tens of seconds.

## XML FILE

It is possible to obtain current values from Quido in the form of a text file in XML format. The file is available at [http://\[IP\\_address\]/fresh.xml](http://[IP_address]/fresh.xml) – i.e. for example at <http://192.168.1.254/fresh.xml> for Quido in the default setting.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<root xmlns="http://www.papouch.com/xml/quido/act">
  <din id="1" name="Svetlo chodba" sts="0" val="0" pic="12" cmo="0" cnt="0" />
  <din id="2" name="Jizni brana" sts="0" val="0" pic="17" cmo="0" cnt="0" />
  <din id="3" name="Pocet krabic" sts="0" val="0" pic="1" cmo="1" cnt="274" />
  <din id="4" name=" " sts="0" val="1" pic="3" cmo="3" cnt="28" />
  <din id="5" name="" sts="0" val="0" pic="1" cmo="0" cnt="0" />
  <din id="6" name="" sts="0" val="1" pic="17" cmo="1" cnt="65535" />
  <din id="7" name="" sts="0" val="0" pic="1" cmo="2" cnt="0" />
  <din id="8" name="" sts="0" val="1" pic="4" cmo="3" cnt="28" />
  <dout id="1" name="Klapka" sts="0" val="0" pic="7" mde="0" pars="" />
  <dout id="2" name="Osvetleni haly" sts="0" val="1" pic="12" mde="0" pars="1;0;300;351;5" />
  <dout id="3" name="Brana" sts="0" val="0" pic="10" mde="1" pars="0;0;0;0;5" />
  <dout id="4" name="Chlazení" sts="0" val="0" pic="16" mde="4" pars="2;0;405;452;6" />
  <dout id="5" name="" sts="0" val="0" pic="12" mde="5" pars="1;1;506;550;32" />
  <dout id="6" name="Jizni vjezd" sts="0" val="1" pic="11" mde="2" pars="0;0;0;0;42" />
  <dout id="7" name="" sts="0" val="0" pic="15" mde="6" pars="1;1;607;650;47" />
  <dout id="8" name="" sts="0" val="0" pic="16" mde="1" pars="3;1;600;653;47" />
  <temp id="1" sts="0" val=" 25,3" th=" 35" tl=" -10" tenb="1" />
  <status location="U Papoucha" unit="C" />
</root>
```

**Note on CORS:** The XML includes headers that allow access to the XML from web applications running on a different domain (Access-Control-Allow-Origin: \* etc.).

The XML file contains four types of nodes: *din*, *dout*, *temp* and *status*

### Inputs: din

#### id

Unique identifier of each digital input of the device. The first input = number 1.

#### name

User-defined name of the input.

#### cnt

The current state of the counter of changes in this input.

**cmo**

Change counter mode:

- 0..... Counter is disabled.
- 1..... For each detected rising edge the counter increases by one.
- 2..... For each detected falling edge the counter increases by one.
- 3..... For each detected edge the counter increases by one.

**val**

The current status of the input (0 or 1).

**sts**

Shows the current status of the input; may have the following values:

- 0..... the value (*val*) is valid and represents the current status of the input
- 1..... waiting for the initial reading of the status
- 4..... an error occurred when reading the status

**Outputs: dout**

---

**id**

Unique identifier of each output of the device. The first output = number 1.

**name**

User-defined name of the output.

**val**

The current status of the output (0 or 1).

**sts**

Shows the current status of the output; may have the following values:

- 0..... the value (*val*) is valid and represents the current status of the output
- 1..... waiting for the initial reading of the status
- 4..... an error occurred when reading the status

**mde**

The current mode of the output; may have the following values:

- 0..... manual control mode
- 1..... time-controlled closing of contact (positive pulse)
- 2..... time-controlled opening of contact (negative pulse)
- 3..... temperature monitoring – closing of contact
- 4..... temperature monitoring – opening of contact
- 5..... temperature monitoring – time-controlled closing of contact (positive pulse)
- 6..... temperature monitoring – time-controlled opening of contact (negative pulse)

**Thermometers: temp**

---

**id**

Unique identifier of individual thermometers; starts from number 1.

**val**

Currently measured temperature in configured units.

**sts**

Shows the current status of the thermometer; may have the following values:

- 0 ..... the value (*val*) is valid and shows the current temperature
- 1 ..... waiting for the first temperature measurement
- 4 ..... sensor error

**tl**

User-defined lower temperature limit.

**th**

User-defined upper temperature limit.

**status****location**

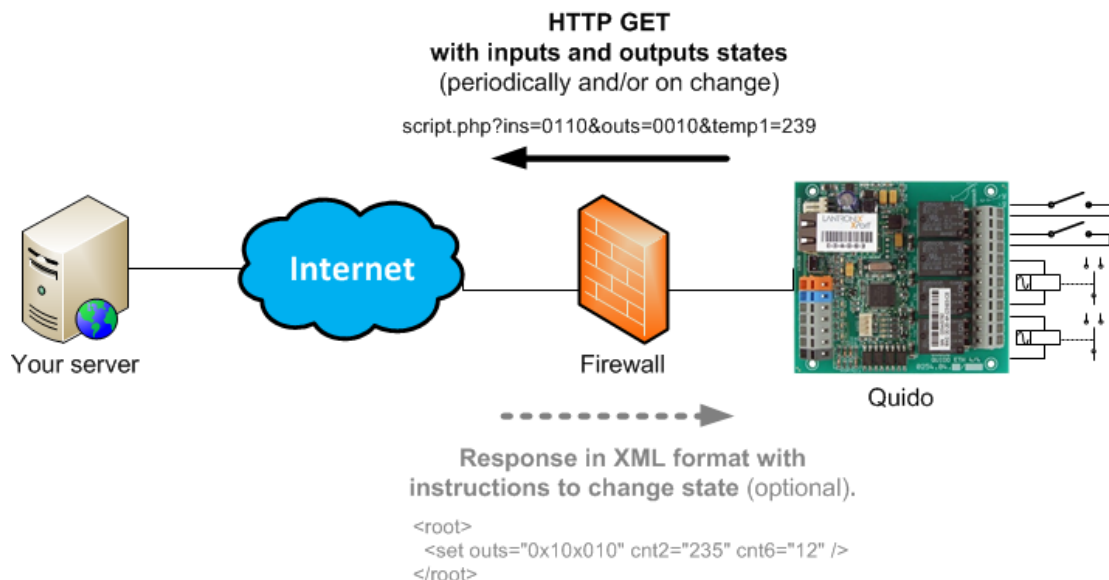
User-defined name of the device.

**unit**

The currently selected temperature unit symbolized by the capital letter C (degrees Celsius), F (Fahrenheit) or K (Kelvin).

**HTTP GET**

HTTP GETs can be used to **send the current status of Quido to a remote server** or to **control the outputs and counters of Quido** (page 31).

**Sending the current status to a remote server**

Data from Quido can be easily sent to a web server that processes the values received from Quido as HTTP GET request parameters. Quido periodically calls a script on the server on the Internet (or intranet) with current data in the HTTP GET parameter. The server can store these values in a database and further work with them – analyse them or just display them on the web for other users.

Quido settings contain the path to the script on the server. Depending on its configuration, Quido sends the current values periodically and/or when there is a change in the status of an input or output.

In addition to current data, Quido's MAC address is also sent to the server in order to uniquely identify Quido in case the server receives data from several devices.

If the response to the HTTP GET contains a setup command, the server can also set the status of outputs and input counters.

## Method of sending the request

---

In this request type, parameters are sent in the message address as standard HTTP GET parameters. An example:

```
www.example.com/script.php?mac=00-20-4A-B4-8D-F7
&name=Office&ins=01101010&outs=00100010&tempS=0&tempV=21.8&cnt2=235&cnt6=126
```

As is evident from the example above, the request has a format that is well-known to web programmers. It is not necessary to learn new programming principles as it is sufficient to use the mechanism known from web form processing (<form name="myform" action=...).

HTTP GET is sent in two cases:

- 1) **Periodically:** If the *Periodical Sending* option is selected in the configuration of the *Sending* tab, HTTP GET is sent every X seconds (with the maximum of 3600 seconds).
- 2) **Immediately after change:** It can be set separately for each input and output whether to monitor its status changes. If a change occurs on one of these inputs/outputs, the HTTP GET is sent immediately.

## Request parameters

---

GET parameters are as follows:

mac..... MAC address to uniquely identify the sender

name..... device name, user-defined

ins..... a string showing the status of individual inputs; for each input of the device, the string contains a character: 0 (inactive), 1 (active), depending on its current status

outs..... a string showing the status of individual outputs; for each output of the device, the string contains a character: 0 (open) or 1 (closed), depending on its current status.

tempS ..... shows the status of the measured temperature; it can have the following values:

0 ....the value is valid and represents the currently measured value

1 ....waiting for the first measurement

2 ....the measured value has exceeded user-defined upper limit

3 ....the measured value has dropped below the user-defined lower limit

4 ....the value is not valid – measuring error or sensor error (damaged sensor or cable)

tempV ..... the measured temperature as a decimal number without units.

cntX..... the number counted by the counter on input X. This parameter is provided for each input. "X" is replaced by the input number (numbering starts from one).

## Quido setup in response to GET

The status of outputs and counters in Quido can be set easily by a string in direct response to HTTP GET. This is especially helpful in cases where it would not be desirable for safety reasons to establish a connection to the local computer network from the Internet. The server in the Internet waits for a regular HTTP GET from Quido and responds by sending a command to change the status of inputs or to reset the counters.

The response is structured in the XML format and looks like this:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<root>
  <set outs="0x10x010" cnt2="235" cnt6="12" />
</root>
```

The *set* node contains individual parameters that can be used to set the status of outputs or change the status of counters. XML should contain no information other than that listed above.

### outs

A string showing the required status of individual outputs. The parameter must show a string whose length is exactly equal to the number of outputs. Each character corresponds to one output. The first character on the left represents the first output. The characters can be:

- 0 ..... Open the output. (It works only if the output is in manual mode.)
- 1 ..... Close the output. (It works only if the output is in manual mode.)
- x..... Leave the output unchanged. (x is the only valid value for outputs controlled by temperature.)

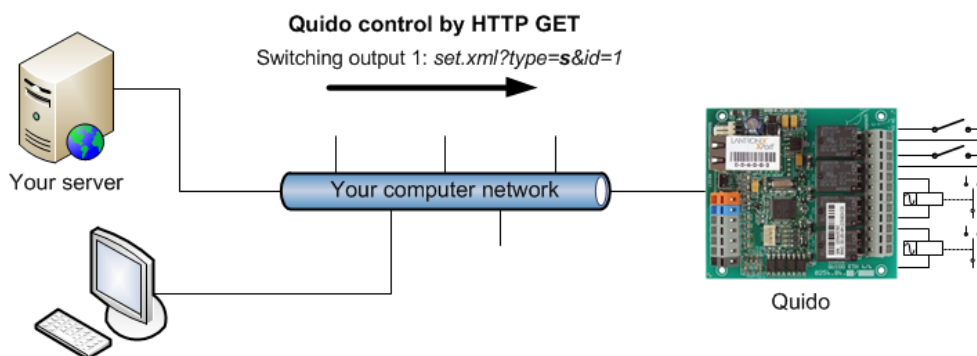
The above example applies to Quido ETH 8/8.

### cntX

These parameters (instead of X, specify the number of the output) can be used to subtract a value from the current state of a counter in Quido. The number you specify is subtracted from the current state of the counter.<sup>10</sup> (The number must be equal to or less than the current counter status.)

The maximum number of cntX parameters in one instruction is ten. Quido is not able to process more than that.

## Quido control by HTTP GET



<sup>10</sup> The counter is not reset because resetting the counter could result in omitting some pulses coming between the last reading of the counter and its resetting.

The following commands can be used to easily control the outputs and counters of Quido. GETs are received by the *set.xml* script. For Quido in the default setting the complete address of the script is: <http://192.168.1.254/set.xml>. This script can only be run **after entering the user's name and password** (if these are set in Quido).

### Close the output

Example: `set.xml?type=s&id=1`

This command closes the output whose number is specified in *id* (OUT1 in this case).

By adding *time* parameter, you can set for how long the output should be ON. You can enter a number from 1 to 255 corresponding with 0.5 to 127.5 seconds ON time.

### Open the output

Example: `set.xml?type=r&id=5`

This command opens the output whose number is specified in *id*.

By adding *time* parameter you can set for how long the output should be OFF. You can enter a number from 1 to 255 corresponding with 0.5 to 127.5 seconds OFF time.

### Invert the output

Example: `set.xml?type=i&id=12`

This command inverts the output whose number is specified in *id*.

### Generate a pulse on the output

Example: `set.xml?type=p&id=2`

This command generates a required pulse on the output whose number is specified in *id*.

### Deduction from a counter

Example: `set.xml?type=c&id=3&cnt=274`

This command can be used to subtract a specified value from the current state of a counter. The number of the counter is specified in *id*. The value to be subtracted is specified in *cnt*. (The *cnt* value must be equal to or less than the current counter status.)<sup>11</sup>

### Reset all counters

Example: `set.xml?type=C`

This command can reset the current value of all counters simultaneously.<sup>11</sup>

### Close all outputs

Example: `set.xml?type=S`

This command can be used to close all outputs.<sup>12</sup>

### Open all outputs

Example: `set.xml?type=R`

This command can be used to open all outputs.<sup>12</sup>

---

<sup>11</sup> This command is not implemented in Quido ETH 3/0B. Can be done via [XML response](#).

<sup>12</sup> Please note that the command works only if all the outputs are in the manual mode!

### Simultaneous setting of multiple outputs

Example: `set.xml?type=A&sts=00110x01`

This can be used to set the status of several outputs simultaneously.<sup>12</sup> The *sts* contains exactly as many characters as is the number of Quido outputs. (The above example applies to Quido ETH 8/8, so the number of characters is 8.) The first character represents OUT1. The characters can be as follows:

- 0 ... to open the output
- 1 ... to close the output
- x ... leave the output unchanged

### Setting multiple outputs for a specific period of time

Example: `set.xml?type=T&sts=10xxxxxx&time=5`

In this way, multiple outputs can be set for a certain period of time at the same time.<sup>12</sup> The *sts* contains exactly as many characters as is the number of Quido outputs. (The above example applies to Quido ETH 8/8, so the number of characters is 8.) The first character represents OUT1. The characters can be as follows:

- 0 ... to open the output for the period *time*
- 1 ... to close the output for the period *time*
- x ... leave the output unchanged

The *time* parameter is a number in the range 1 to 255, corresponding to times 0.5 to 127.5 sec.

### Response to HTTP GET

---

The response to setting commands is in the XML format. Example:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<root>
  <result status="1" outs="6" val="0" pic="7"/>
</root>
```

The meaning of the parameters in XML is as follows:

- The **result** tag is always only one.
- The **status** attribute contains number 1 (command executed) or 0 (command not executed).
- The **outs** attribute contains a number specifying the status of all outputs. The number is decimal and must be converted to binary system to determine the status of each output.  
*Example:* Number 6 in Quido ETH 8/8 means than outputs 2 and 3 are closed. (6 in the decimal system equals to 00000110 in the binary system.)
- The **val** attribute contains 0 (output is open) or 1 (output is closed). This attribute is valid only if the command was for a specific output.

## SENDING OF E-MAILS

Quido can send the following information via email:

### **Change in the status of input or output**

---

If for any reason the status of any input or output changes and this option is enabled<sup>13</sup>, Quido sends an email message. Example:

*Subject:* QUIDO\_info\_Sklad

*Message:* Input 1:(Okno) is ON

The subject shows the user-defined name of Quido. The message body specifies whether the message refers to an input or output, shows its user-defined name and status (ON or OFF).

### **Exceeded temperature limits**

---

If the temperature of the connected thermometer exceeds the preset limits, Quido sends an email message. Depending on the configuration, Quido is able to send messages repeatedly until the temperature returns within the limits. Example:

*Subject:* QUIDO\_info\_Sklad

*Message:* Temperature alert! Upper limit: 26.0 C Current: 28.3 C

The subject shows the user-defined name of Quido. The message body specifies which limit has been exceeded and shows the current temperature.

### **Sensor error**

---

If the function of temperature limits monitoring is enabled and an error occurs in the sensor, Quido send a single message. The error may be caused by a disconnected or defective sensor. Example:

*Subject:* QUIDO\_info\_Sklad

*Message:* Temperature sensor error

The subject shows the user-defined name of Quido.

### **Test message to verify the settings**

---

Press the *Test* button in the *Email* tab in the Quido configuration window to send a test message to verify the settings.

*Subject:* QUIDO\_info\_Sklad

*Message:* Test message

---

<sup>13</sup> It means that changes are monitored for the particular input or output and the function of sending emails is enabled in the relevant tabs of the configuration window.

## SNMP

SNMP protocol contains objects with current states. MIB table, which you can import to your SNMP manager is freely available on papouch.com. Quido uses SNMP version 1.

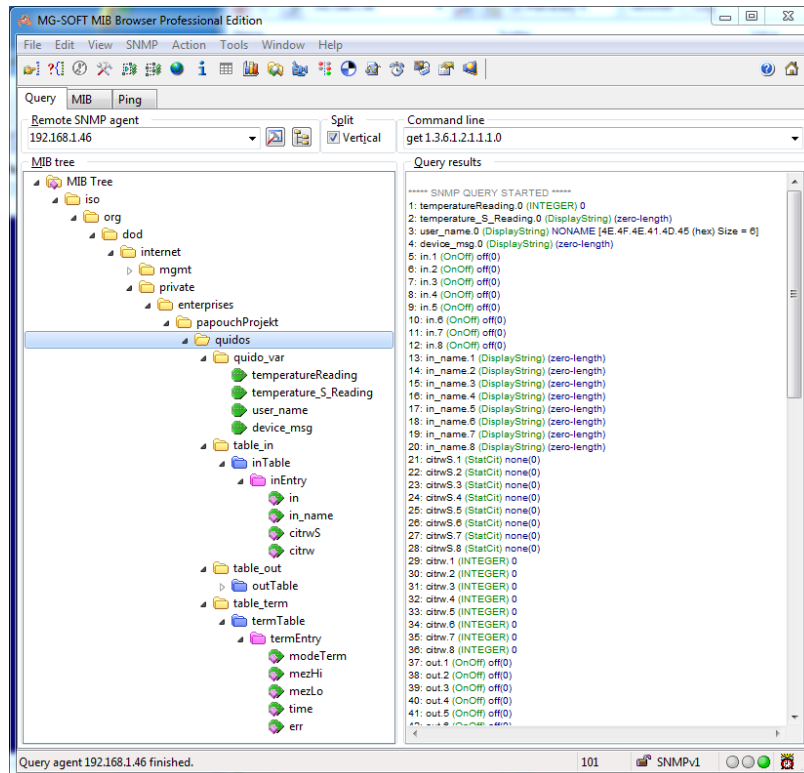


fig. 34 – MIB tree in Quido ETH

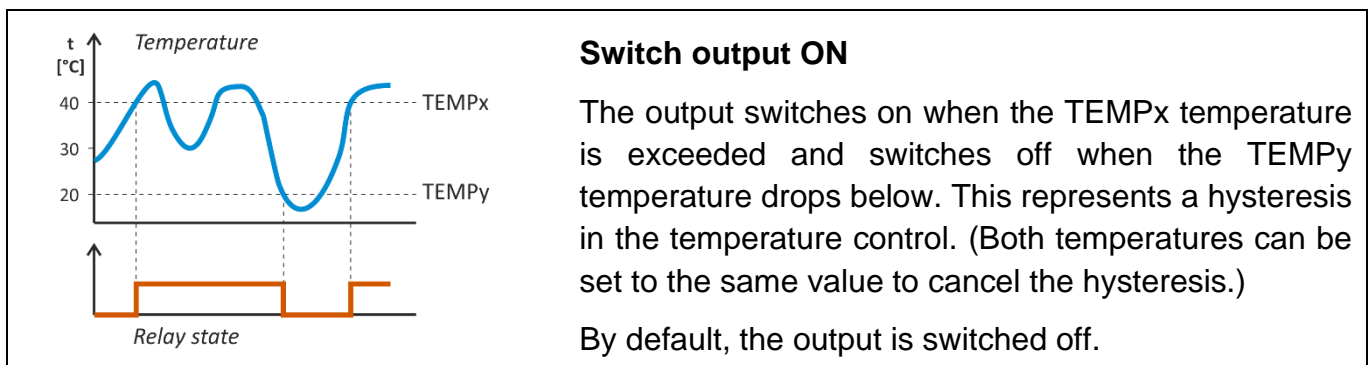
**Tip:** If you want to browse the whole SNMP tree using SNMPWALK utility (Linux), you need to specify exactly where the readout should start after the IP address. For example:

```
snmpwalk -v1 -c public 192.168.1.254 1.3.6.1.4.1.18248
```

If you only rad with IP address, you will get just basic system IOD objects of the device.

## TEMPERATURE WATCH MODES

Following pictures show options for setting temperature monitoring in Quido modules.<sup>14</sup>

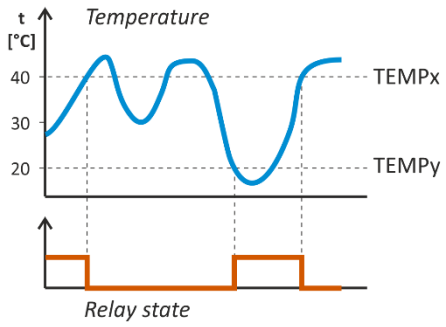
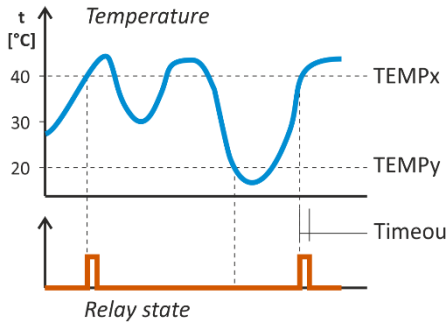
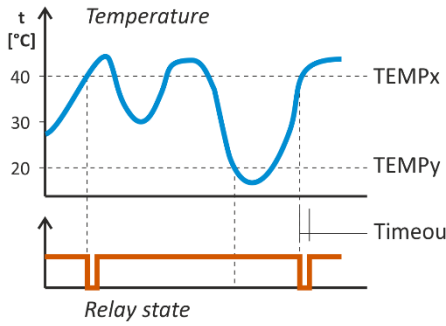
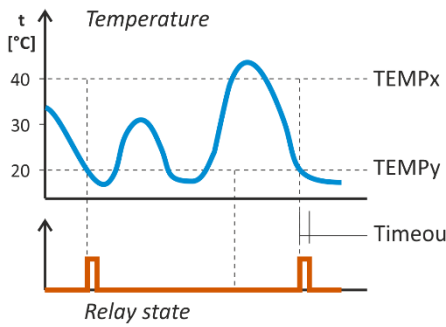
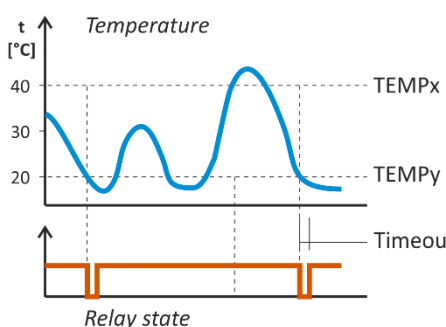


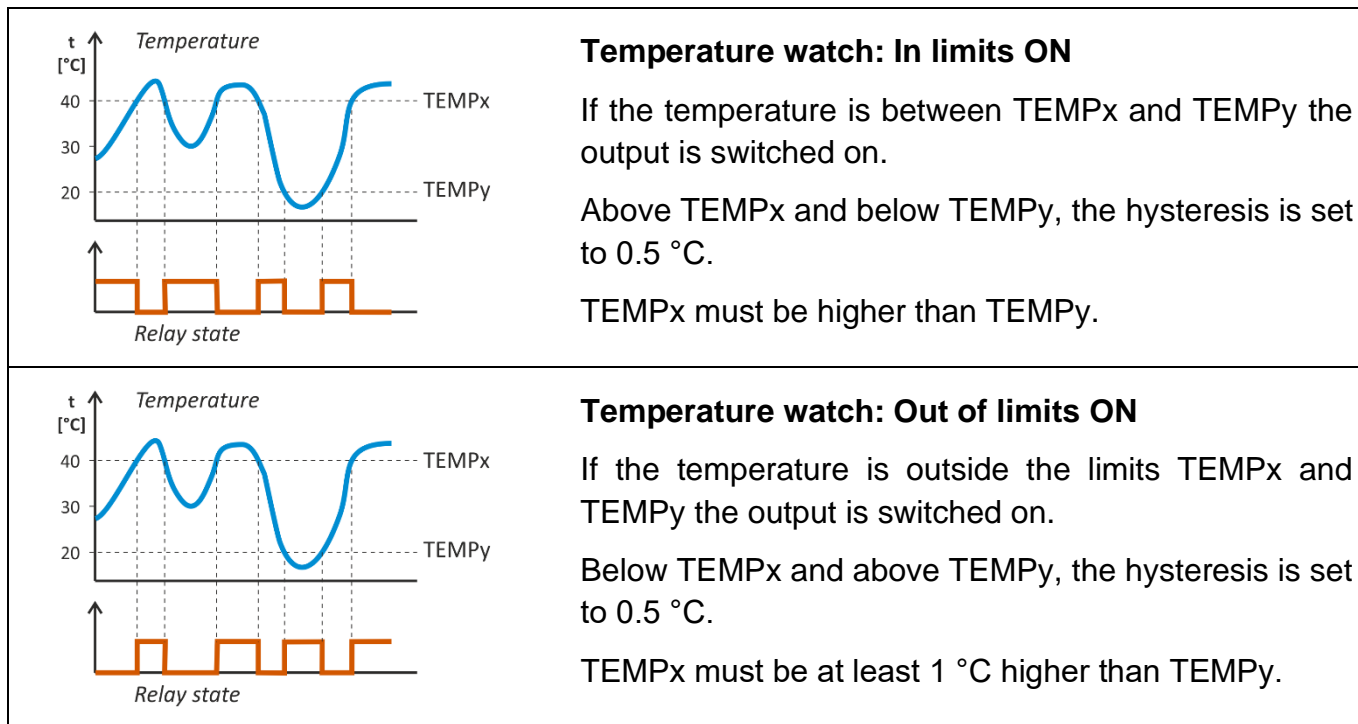
### Switch output ON

The output switches on when the TEMPx temperature is exceeded and switches off when the TEMPy temperature drops below. This represents a hysteresis in the temperature control. (Both temperatures can be set to the same value to cancel the hysteresis.)

By default, the output is switched off.

<sup>14</sup> Temperature monitoring is only available on devices with at least one thermometer and output.

	<p><b>Switch output OFF</b></p> <p>Inverse mode to the previous one.</p> <p>By default, the output is switched on.</p>
	<p><b>Temperature watch: Switch ON for fixed time</b></p> <p>The output switches on for a pre-set time when the TEMPx temperature is <b>exceeded</b>. It can only switch on again when the temperature drops below TEMPy and then rises again to TEMPx.</p> <p>The limits are two to prevent the output from oscillating if the temperature oscillates around one value.</p> <p>By default, the output is switched off.</p>
	<p><b>Temperature watch: Switch OFF for fixed time</b></p> <p>Inverse mode to the previous one.</p> <p>By default, the output is switched on.</p>
	<p><b>Temperature watch: Switch ON for fixed time</b></p> <p>The output switches on for a pre-set time when the temperature <b>drops below</b> TEMPy. It can switch on again only when the temperature rises above TEMPx and then falls below TEMPy again.</p> <p>The limits are two to prevent the output from oscillating if the temperature oscillates around one value.</p> <p>By default, the output is switched off.</p>
	<p><b>Temperature watch: Switch OFF for fixed time</b></p> <p>Inverse mode to the previous one.</p> <p>By default, the output is switched on.</p>



## FAQ

### Which settings are required for Quido to work in my network?

It is only necessary to set the network parameters of Quido to fit your network (the IP address and the netmask, if applicable). The process bellow is described for OS Windows.

- 1) Connect Quido to the network a run [Ethernet Configurator](#) (see Fig. 7).<sup>15</sup>
- 2) Click *Add device* and enter the MAC address of Quido and the required IP address.
- 3) Click *Set*.
- 4) Now you can open the web interface of Quido using your internet browser.

### How to find out the IP address of Quido?

- 1) The default IP address of Quido is 192.168.1.254. If you have changed the address or it is impossible to connect to Quido using the default address, follow the instructions below.
- 2) Run [Ethernet Configurator](#) (see Fig. 7).<sup>15</sup> If Quido is connected to your network and its IP address is compatible, it will be displayed as one of the devices in the *List of detected devices*.
- 3) If Quido is not displayed in the *List*, make sure that it is properly connected and assign it a new IP address by following the instructions above.

<sup>15</sup> This program can also be downloaded from [www.papouch.com](http://www.papouch.com).

## RESET

Use the following procedure to reset the device to "factory settings". Unlike the reset that can be done via the web or [Telnet](#), the IP address will be changed to 192.168.1.254 or assigned by the DHCP server.

- 1) Disconnect the device from the power supply.
- 2) Press and hold the Reset button (may also be marked RST).
- 3) Continue according to how you want to assign the IP address:
  - a. **IP address 192.168.1.254:**
    - i. Turn on the power.
    - ii. Wait approx. 3 sec for the LNK light to flash (approx. 2 Hz).
    - iii. Release the button.
    - iv. IP address is set, device is in "factory setting".
  - b. **IP address assigned by the DHCP server:**
    - i. Turn on the power.
    - ii. Wait approx. 25 sec - during this time the LNK indicator will flash slowly at first (approx. 2 Hz).
    - iii. Once the LNK light starts flashing faster, release the button.
    - iv. The device is in "factory setting". The IP address assigned by the DHCP server can be found in your DHCP server (typically in your router). The section with such assigned addresses has different names - for example, *DHCP Client List*, *DHCP Clients*, etc.

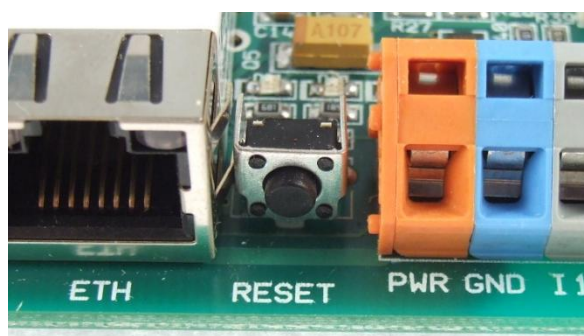


Fig. 35 – reset button on Quido

## INDICATORS

On the device there are indicators for power, communication, status of inputs and outputs.

- **ON** (or **PWR**) light turns on when power is connected.
- **COM** light:
  - In WEB mode permanent light.
  - In other communication modes:
    - Long blink (0.5 sec): When power on, when receiving an instruction with a valid address and when sending an automatic message.
    - A flash (0.1 sec) once every 10 sec indicates that the internal program in Quido running correctly.
- **LNK** light:
  - Lights when TCP connection is established (client and server mode).
  - During reset, it flashes in a specific pattern.
- Input lights (**INx**) are lit when input x is active (i.e. voltage is connected to the input).
- Output lights (**OUTx**) are lit when relay x is switched on.
- **Link** indicator – left indicator light in Ethernet connector:
  - No light .....not connected
  - Yellow.....connected at 10Mbps
  - Green .....connected at 100Mbps
- **Connection type** indicator – right indicator light in the Ethernet connector:
  - No light .....no communication
  - Yellow.....Half-Duplex communication
  - Green .....Full-Duplex communication

## TECHNICAL PARAMETERS

### Inputs

No of digital inputs .....	0 to 100 (depends on type)
Input type .....	for connecting voltage or a NO contact
Galvanic isolation .....	optical
Sampling .....	from 1 ms (by setting; see Fig. 19; default 20 ms)
Maximum no of recordable changes in the mode of counting changes .....	65 535 <sup>16</sup>

#### Version I.

Input voltage for state "1" .....	4,5 – 10 V
Input voltage for state "0" .....	0 – 2,5 V
Input current at 5 V .....	typically 3.2 mA
Input current at 9 V .....	typically 8.9 mA
Maximum input voltage .....	10.0 V

#### Version II.

(This option is the default unless you specify otherwise in your order.)

Input voltage for state "1" .....	7 – 28 V
Input voltage for state "0" .....	0 – 3 V
Input current at 12 V .....	typ. 3.5 mA
Input current at 24 V .....	typ. 7.8 mA
Maximum input voltage .....	28 V

### Outputs

No of digital outputs .....	0 to 32 (depends on type)
Type .....	changeover (SPDT) contact relay
Maximum switching voltage .....	AC voltage: 60 V, DC voltage 85 V
Maximum switched current .....	5 A
Protective varistor .....	$U_{AC} = 60 \text{ V}$ ; $E_{MAX} = 5 \text{ J}$ ; $C = 0,64 \text{ nF}$

### Thermometer

Amount .....	1
Sensor type .....	semiconductor
Measuring range .....	-55 °C to +125 °C
Accuracy .....	$\pm 0.5 \text{ °C}$ in range of -10 °C to +85 °C; otherwise $\pm 2 \text{ °C}$
Temperature drift .....	$\pm 0.2 \text{ °C}$ over 1000 hours at 125 °C

<sup>16</sup> The change counter is automatically reset when Quido is turned on or restarted.

The temperature sensor must not be used for:

- measurements in chemically aggressive environments
- measurements in areas with high electrical interference
- measurement of objects or devices under voltage

#### Sensor in shrinkable rubber coating:

Thermal response time ..... 6 s

#### PVC cable for sensor in rubber coating:

Cable jacket ..... PVC

Length ..... 1, 3, 5, 10 or 15 meters

Working temperature range ..... -10 to +70 °C

Cable diameter ..... max. 4 mm

#### Sensor in metal housing:

Thermal response time .....  $\tau_{50} = 6$  s,  $\tau_{90} = 18$  s

Degree of protection ..... IP 68 (immersion to a depth of up to 1m)

Housing material ..... stainless steel CSN 17240 (DIN 1.4301)

Housing diameter .....  $5.7 \pm 0.1$  mm

Housing length ..... 60 mm

Insulation resistance ..... min 200 M $\Omega$  at 500 V<sub>SS</sub>, at 15 – 35 °C and max 80 % relative humidity

Nominal pressure ..... PN 25

#### Silicone cable for sensor in metal housing:

Cable jacket ..... silicone rubber, blue

Length ..... 1, 3, 5, 10 or 15 meters

Working temperature range – permanent ..... -60 °C to +200 °C

Maximum permissible temperature ..... +220 °C

Cable diameter ..... 4.3 mm ( $\pm 0.1$  mm)

#### Control interface

Type ..... 10/100 Ethernet

Connector ..... RJ45

Communication speed of virtual port ..... 115200 Bd (fixed)

SSL/TLS encryption ..... no

Default IP address ..... 192.168.1.254

Default netmask ..... 255.255.255.0

Default IP address of gateway ..... 0.0.0.0

SMTP ..... without SSL, TLS, etc.

**Connectors**

(Connectors for power supply, inputs, outputs)

- Type ..... Wago 236 terminal block
- Conductor cross section..... 0.08 to 2.5 mm<sup>2</sup>
- The required length of stripping ..... 5 to 6 mm
- Angle of wires to circuit board ..... 45°
- Pitch between terminals ..... 5.08 mm
- Method of attaching wires ..... Wago CAGE CLAMP<sup>®17</sup>

**Other parameters**

- Power supply ..... 8 to 30 V DC
- Reverse polarity protection ..... yes, diode in series
- Operating temperature of electronics.... -20 °C to +60 °C
- Degree of protection..... IP 00 (designed to be built-in)
- Mounting holes – diameter..... 3.2 mm

	No relays closed [mA]		All relay closed [mA]	
	12 V	24 V	12 V	24 V
<b>Quido ETH 3/0</b>	80	44		
<b>Quido ETH 10/1</b>	140	70	175	95
<b>Quido ETH 4/4</b>	140	70	280	150
<b>Quido ETH 8/8</b>	140	75	385	196
<b>Quido ETH 30/3</b>	140	75	250	135
<b>Quido ETH 60/3</b>	140	75	250	135
<b>Quido ETH 100/3</b>	140	75	250	135
<b>Quido ETH 0/2</b>	140	75	200	100
<b>Quido ETH 2/16</b>	140	75	620	320
<b>Quido ETH 2/32</b>	140	75	1135	560

*Tab. 2 – Typical current consumption of I/O Quido ETH modules*

<sup>17</sup> A plastic lever for easy connection of wires to the terminals is included. It is also possible to use a small flat screwdriver.

	Weight [g]
Quido ETH 3/0	40
Quido ETH 10/1	70
Quido ETH 4/4	95
Quido ETH 8/8	180
Quido ETH 30/3	155
Quido ETH 60/3	240
Quido ETH 100/3	350
Quido ETH 0/2	55
Quido ETH 2/16	325
Quido ETH 2/32	590

Tab. 3 – Weight of I/O Quido ETH modules (only electronic board without mounting components)

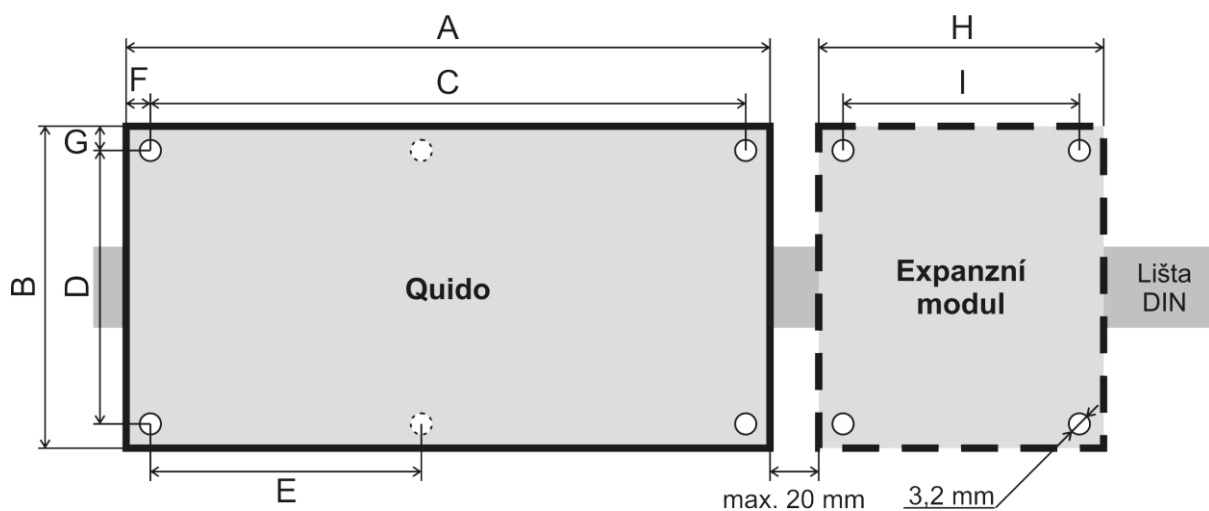


Fig. 36 – Dimensions (specific values are given in the following table)

	A	B	C	D	E	F	G	H	I	Height
Quido ETH 3/0	48	96	47	80		3	8			18
Quido ETH 10/1	74	96	67	80		3	8			20
Quido ETH 4/4	86	96	78	80		4	8			20
Quido ETH 8/8	138	96	130	80		4	8			20
Quido ETH 30/3	176	96	166	66		5	9			20
Quido ETH 60/3 <sup>18</sup>	234	123	225	82	112	4	21			25
Quido ETH 100/3 <sup>18</sup>	234	123	225	82	112	4	21			40
Quido ETH 0/2	54	96	48	80		3	8			20
Quido ETH 2/16	188	123	177	110		6	7			30
Quido ETH 2/32	188	123	177	110		6	7	137	130	30

Tab. 4 – Dimensions in millimeters (electronic boards without mounting components)

<sup>18</sup> There is no upper left mounting hole on the board.

**Board dimensions in the DIN rail holder**

Width (parallel to DIN) ..... Quido + 2 mm<sup>19</sup>

Height (perpendicular to DIN) ..... Quido + 5 mm

Added depth..... approximately 20 mm<sup>20</sup>

---

<sup>19</sup> Please beware that Quido ETH 2/32 is a two-part product. Add the dimension twice to the overall width.

<sup>20</sup> Added depth means the dimension between DIM rail and the PCB. (This does not apply to Quido ETH 60/3.)







# Papouch s.r.o.

Industrial data transmission, line and protocol converters, RS232, RS485, RS422, USB, Bluetooth, Ethernet, LTE, WiFi, measurement modules, smart temperature sensors, I/O modules, custom development and manufacturing.

Address:

**Strasnicka 3164  
102 00 Prague 10  
Czech Republic**

Phone:

**+420 267 314 267**

Web:

**en.papouch.com**

Mail:

**info@papouch.com**

