PAPAGO TH 2DI DO

Environment monitor:

Measures temperature, humidity and calculates dew point

\+ two contact inputs and a relay

Ethernet or WiFi interface

PoE or external power

PAPAGO TH 2DI DO

Datasheet

Created: 14.11.2014

Last update: 3.10 2019 08:39

Number of pages: 48

Papouch s.r.o.

Address:

Strasnicka 3164
102 00 Prague 10
Czech Republic

Tel:

+420 267 314 267

Internet:

www.papouch.com

E-mail:

info@papouch.com

TABLE OF CONTENTS

FIRMWARE VERSIONS

Version 1.05

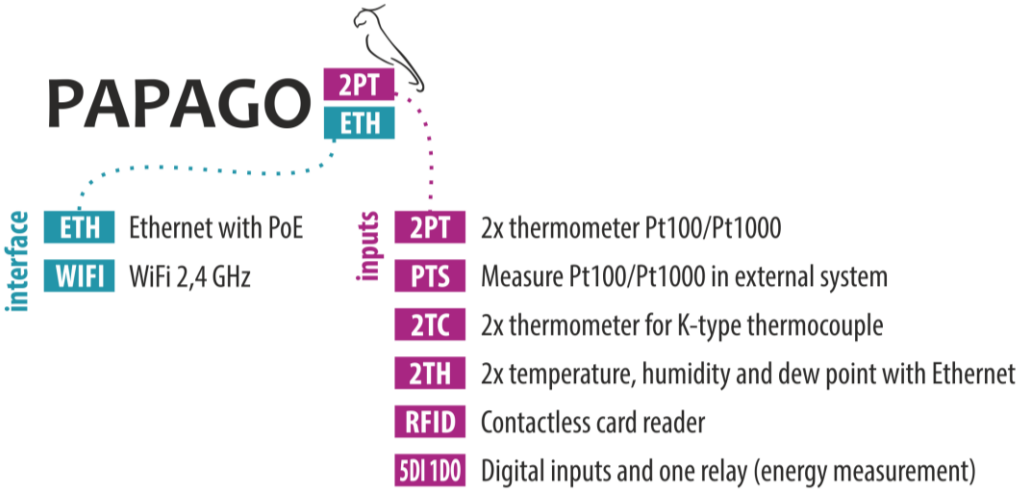
- Added support for new TH3 sensor.

Version 1.0

- First version.

GETTING TO KNOW PAPAGO

PAPAGO is a family of devices with uniform appearance and communication capabilities. It allows to combine communication interfaces on one side and measuring sensors (inputs) on the other side.



Applications

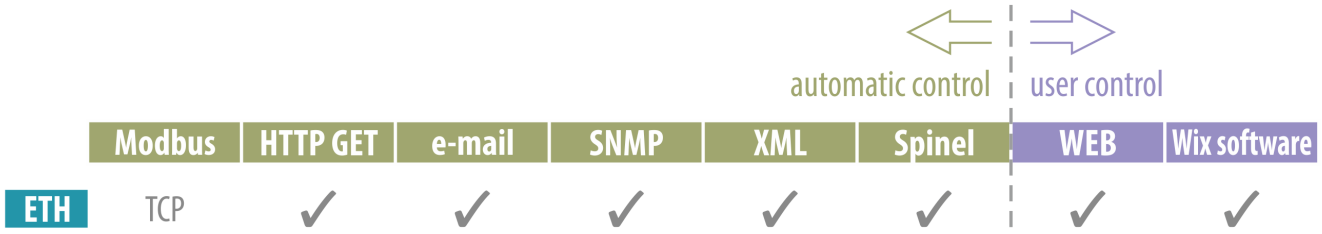
- Temperature and humidity measurement in industry, buildings, server rooms and other environments.
- Measurement of temperature for heating systems.
- Monitoring temperatures in warehouses and archives.
- Monitoring the manufacturing process.
- Monitoring temperature, humidity and reached limits.
- Environmental monitoring via the Internet.
- Measurement for the HACCP system.

Common Features

- Ethernet or WiFi interface to an internal website and many standard communication protocols.
- Ethernet versions with PoE power supply. This eliminates the need to use an external power supply, but the possibility to connect the AC adapter is available.
- Configuration of WiFi parameters via USB interface.
- Internal memory and real-time clock. Measured data including the time of measurement is automatically stored in the memory in the event that communication is lost. The data is automatically sent after the connection is restored.
- Elegant but robust metal box that can be mounted on a DIN rail. The box bears descriptions that allow connection without having to consult the manual. Also LED indicators for all important states help commissioning.
- The possibility to display, store and analyse data in the Wix program.

Communication Options

PAPAGO features different communication options depending on the used interface. PAPAGO can be controlled via a web interface or via software for Windows. Machine data-reading is possible using various standard methods, so PAPAGO can be easily integrated into your existing systems. You can choose the option that is appropriate for your location:



Machine data-reading: Modbus TCP, HTTP GET with encryption, e-mail, SNMP, XML, Spinel

User control: Web interface, Wix software

Properties

PAPAGO TH 2DI DO is used as an environment monitor – it measures temperature, humidity and dew point. It also has two contact inputs to connect either contact or a pulse output and a relay.

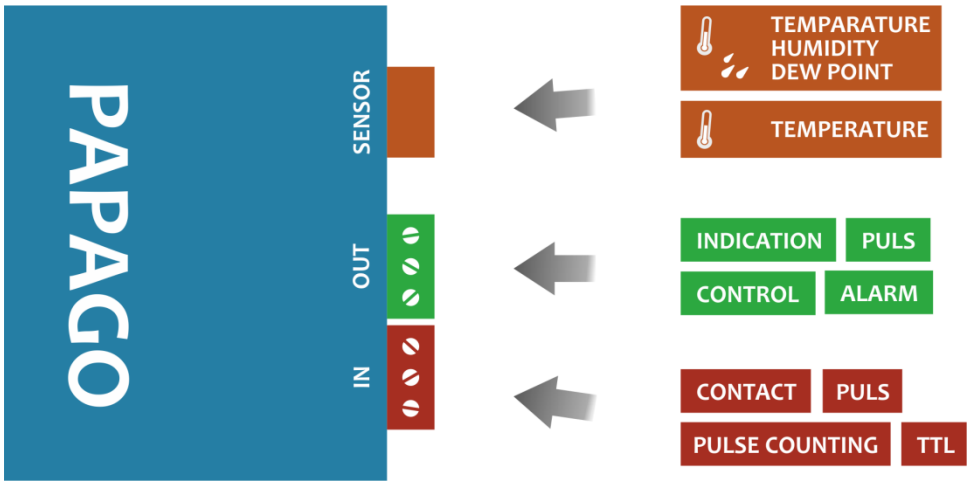


Fig. 1 – use possibilities

Sensor input *(SENSOR)* can accept one of these two types of sensors:

- Sensor A ..... temperature: -40 to 125 °C; humidity: 0 to 100 %
- Sensor B ..... temperature: -55 to 125 °C

Relay *(OUT)* is meant for:

- Switching load with low voltage and power.
- Manual control (on/off or pulse), input mirroring or an alarm for measured value.

Digital inputs *(IN)* can be connected with:

- Contact.
- Pulse output from measuring gauges like electricity, water, gas, heat and other.
- Logic output of 5V TTL levels.

- Family of measuring devices with Ethernet or WiFi interface.
- Modern web interface.
- Data reading via a web interface or Wix software.
- Machine data reading via Modbus, HTTP GET, SNMP, XML, e-mail or Spinel protocol.
- The ability to encrypt data in HTTP GET by 128bit encryption.
- Measurements via external thermometer or combined temperature and humidity sensor. (Sensors are not included.)
- Power supply from PoE (Ethernet versions only) or external source.
- PoE standard according to IEEE 802.3af.
- WiFi 2.4 GHz.
- External DC power supply 11 to 58 V.
- Current consumption typically 72 mA at 24 V.

CONNECTION

1) Ethernet version: Connect the device by a normal uncrossed cable for computer networks to the switch.
2) Ethernet version: If the device cannot be powered by the switch via PoE according to the IEEE 802.3af standard, connect a power adapter to the coaxial connector next to the connector for the Ethernet. DC voltage in the range of 11-58 V is expected. (The positive pole is inside, the input for the power supply has reverse polarity protection.)

   WiFi version: Connect a power adapter to the coaxial connector next to antenna. DC voltage in the range of 11-58 V is expected. (The positive pole is inside, the input for the power supply has reverse polarity protection.)

fig. 2 – Front view with sensor connector and input/output terminals

3) *Sensor* connector: Connect temperature, or combined temperature/humidity sensor (sold separately). Terminal contact and relay connection shown in fig. 2.
4) Ethernet version: Now it is necessary to set the correct IP address of the device. The default IP address is 192.168.1.254 and network mask 255.255.255.0. If your network is

not compatible with this range, set the IP address of the device using Ethernet Configurator.

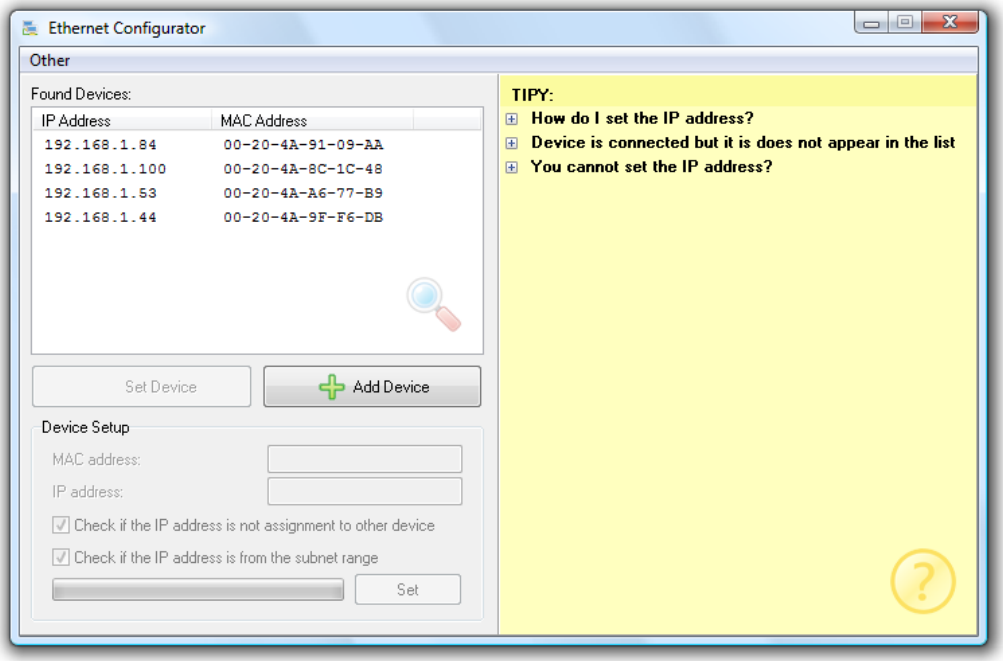


fig. 3 – Ethernet Configurator for setting the IP address

WiFi version: Connect your Papago to a windows PC using the supplied micro USB cable.[1] Run *Papago WiFi Configurator* software, you can download it on papouch.com. Set-up Papago to your WiFi network parameters so you can access it from that network.

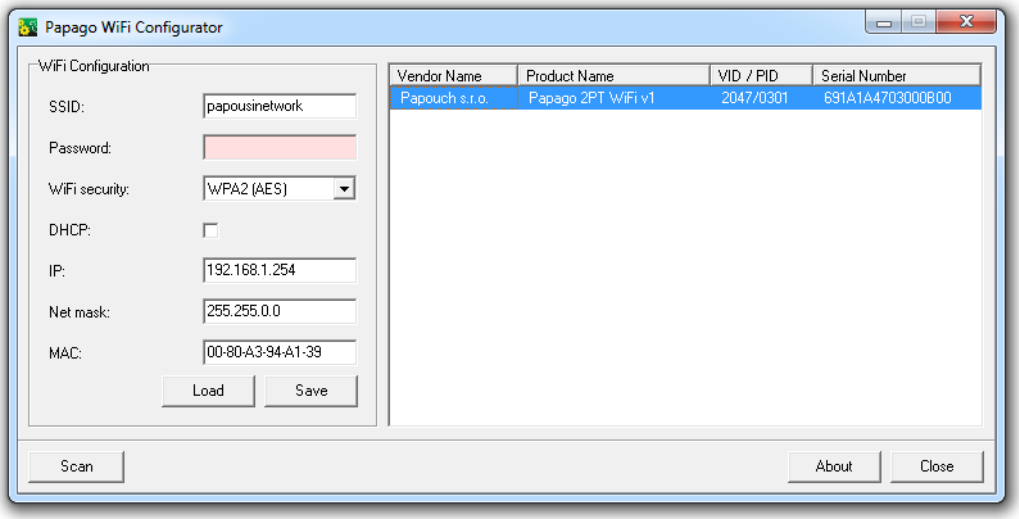


fig. 4 – WiFi configuration via USB

5) After setting the address, you can connect to a Web browser at the address specified as follows: *http://192.168.1.254/* (The example is given for the default IP address.)

[1] In Windows 7 or higher driver will be installed automatically.

CONFIGURATION

Configuration is done via a web interface. The basic network parameters can also be set via Telnet (see page 20). **The web interface** is accessible on the IP address of the device. (The default address is 192.168.1.254.)

After entering the IP address, the main page will appear showing the latest measured values.

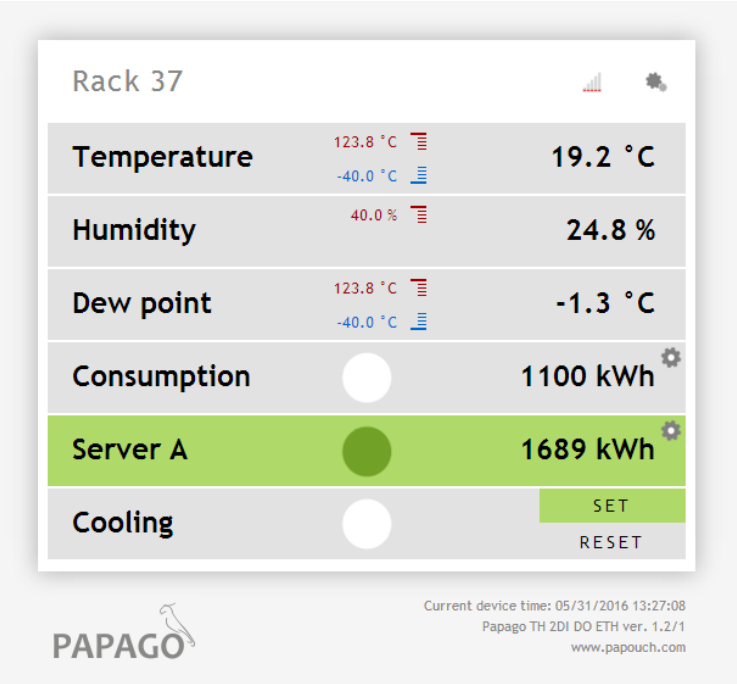


fig. 5 – Web interface with temperature-humidity sensor

The **web interface is secured** with a username and password. You can choose a separate password for the user (who can only display the values on the main page) and for the administrator (who can also change settings).

The web interface is optimized for the following browsers (or later version): Mozilla Firefox 29, Internet Explorer 10, Google Chrome 6, Opera 10.62, Safari 1. The Web interface can also be displayed on mobile phones using OS Android 4.2, iOS 7 and Windows Phone 8.1.

The configuration is displayed when you click the icon of gears in the upper right corner. The configuration is divided into sections according to the types of settings and is available in English and Czech.

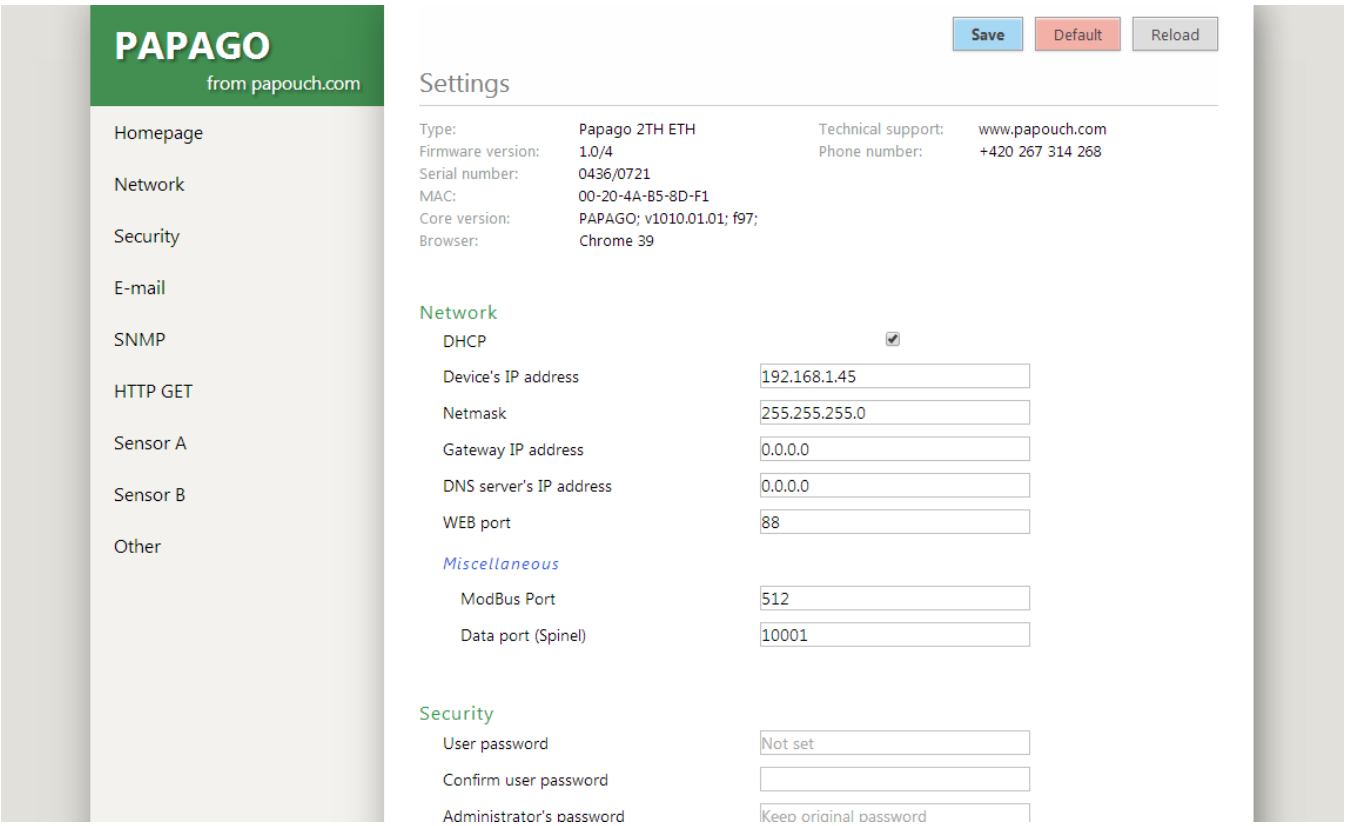


fig. 6 - Configuration of Papago

Network

This section contains the configuration of network parameters.

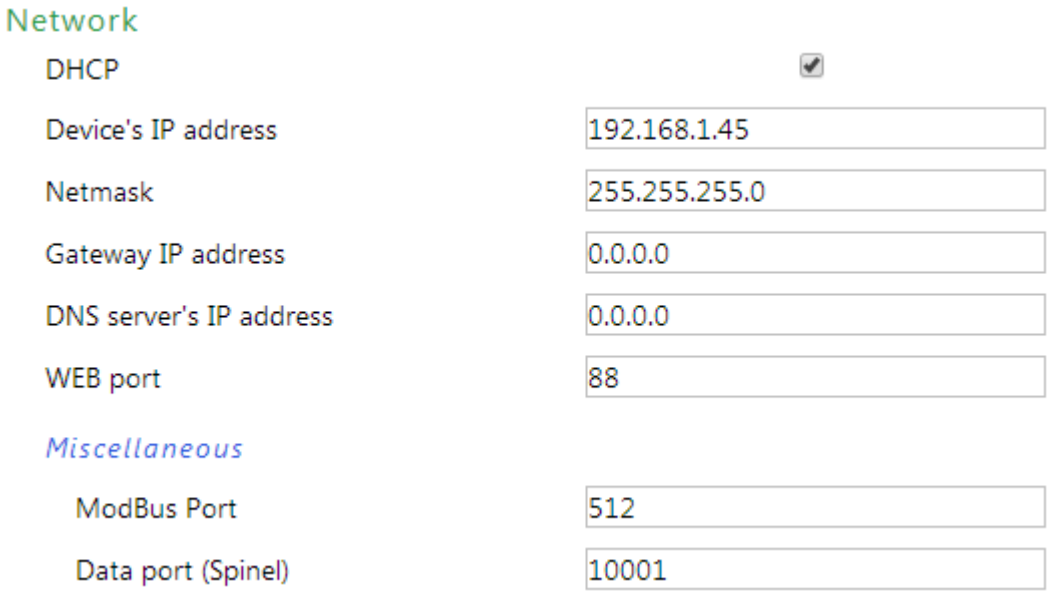


fig. 7 - network configuration

If the box for assigning addresses via DHCP is ticked, the fields for *Device's IP address, Netmask, Gateway IP address* and *DNS server's IP address* are reset and upon reloading the settings they are filled again with data obtained from the DHCP server.

If you have a **version with WiFi interface** in the section *Network* is also following parameters:

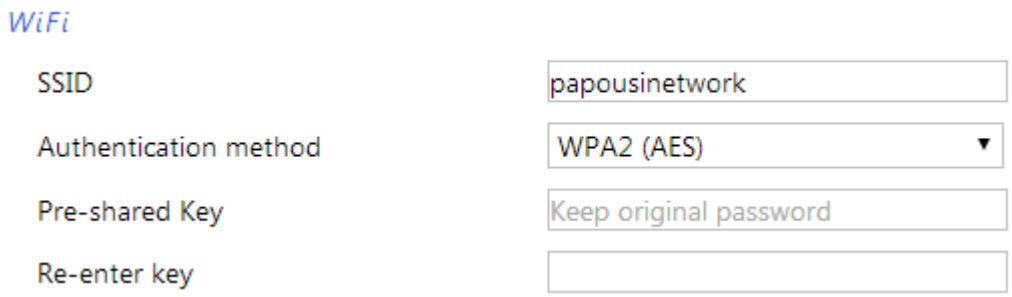


fig. 8 - WiFi network parameters

As *Authentication method* is available this options: *Open, WEP (open), WEP (shared), WPA (TKIP), WPA (AES), WPA2 (TKIP), WPA2 (AES), WPA2 (Mixed)*.

Security

The section for setting the password of the user (can only access the main page) and the administrator (has access to both the main page and the settings).

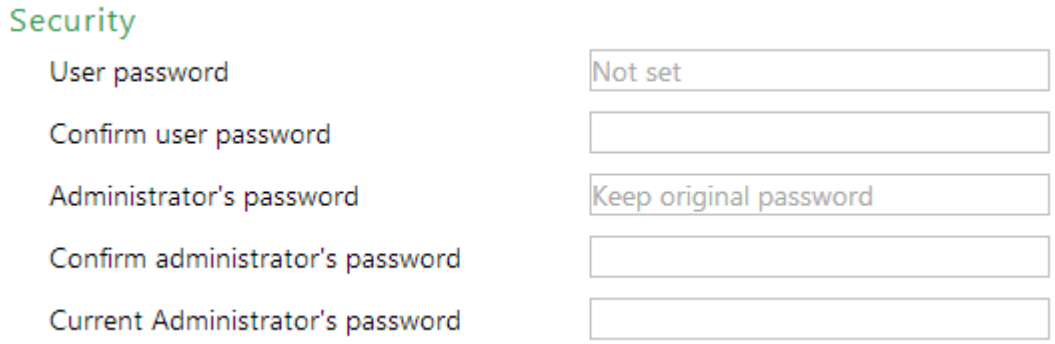


fig. 9 - Access security settings

After saving, the passwords are no longer displayed for security reasons. The fields for entering the password show *Not set*, if the password has not been entered, or *Keep original password*, if the password has been entered but is not to be displayed.

E-mail

The device can send e-mails if one of the thresholds set for any of the measuring channels has been exceeded.

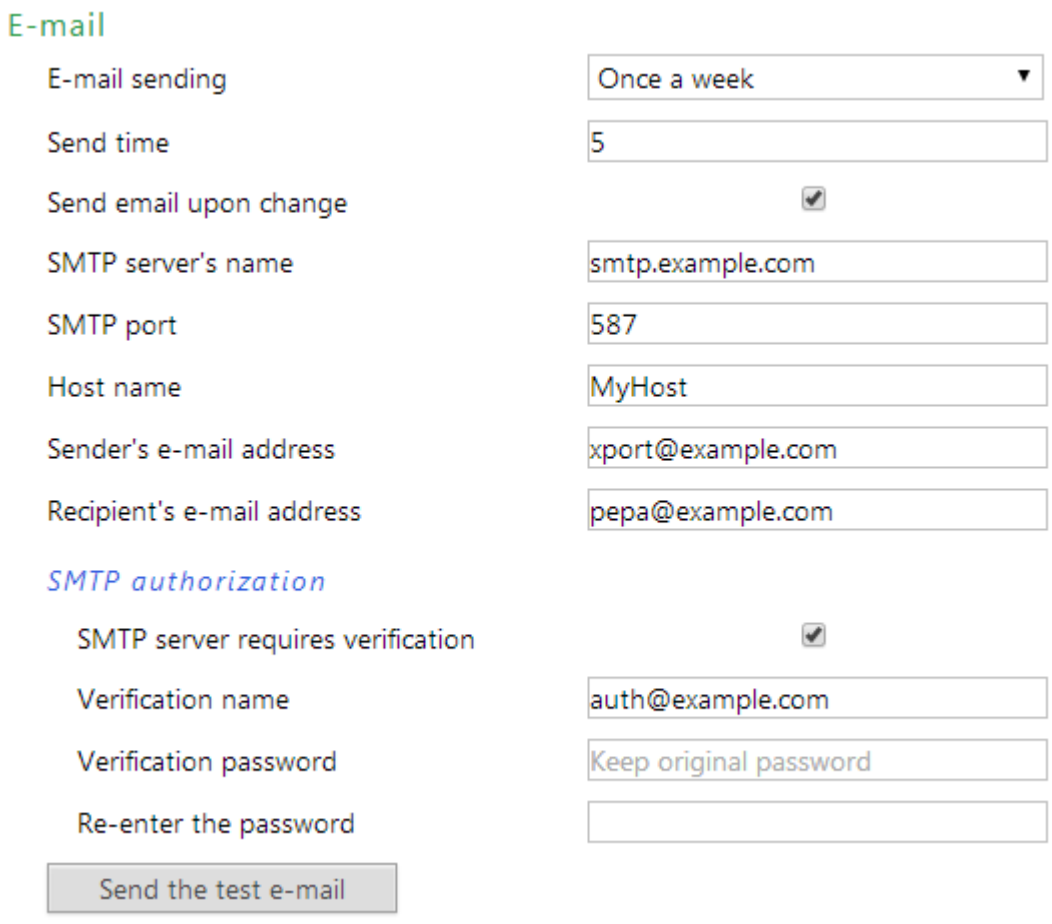


fig. 10 - settings for sending e-mails

E-mail sending

This sets the interval of sending e-mail with current counter states. Available options are once an hour, day, week or month.

- Once an hour: Input 0 to 59 in the Time field to determine which minute of an hour should the e-mail be sent.
- Once a day: Input 0 to 23 in the Time field, e-mail will be sent in the first minute of this hour.
- Once a week: Input 0 to 6 in the Time field (0 is Monday). E-mail will be send in the first minute of this day.
- Once a month: Input 0 to 30 in the Time field. E-mail will be sent in the first minute of the selected day.

E-mail example:

When the limits are exceeded, the device sends an e-mail – format can be seen below:

Input 1 is 29 m3. State is: OFF
Input 2 is 0.049 kWh. State is: OFF
State Output is: OFF
Temperature is in range. Value is 26.7 °C.
Humidity is in range. Value is 61.5 %.
Dewpoint is in range. Value is 18.6 °C.

SNMP

Here you can configure communication via SNMP used for data collection in large networks.

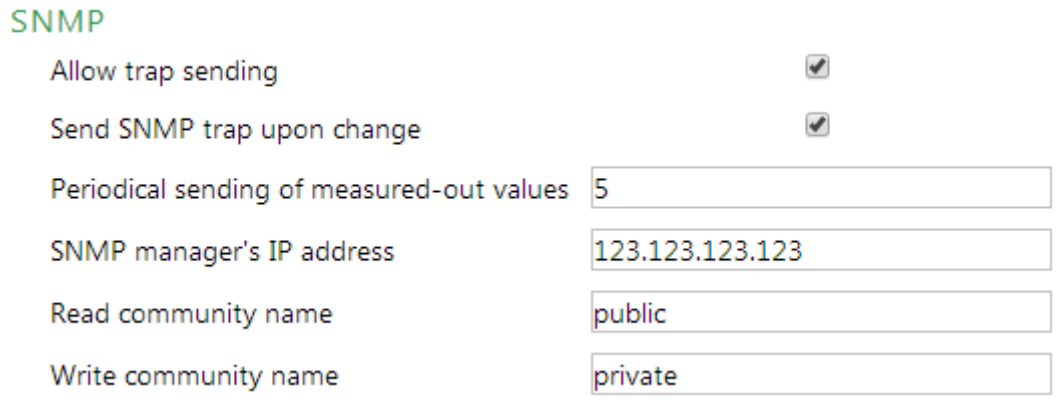


fig. 11 - settings for communication via SNMP

For description of SNMP objects see page 23.

HTTP GET

This section is used to set the sending of measured data to a remote server.

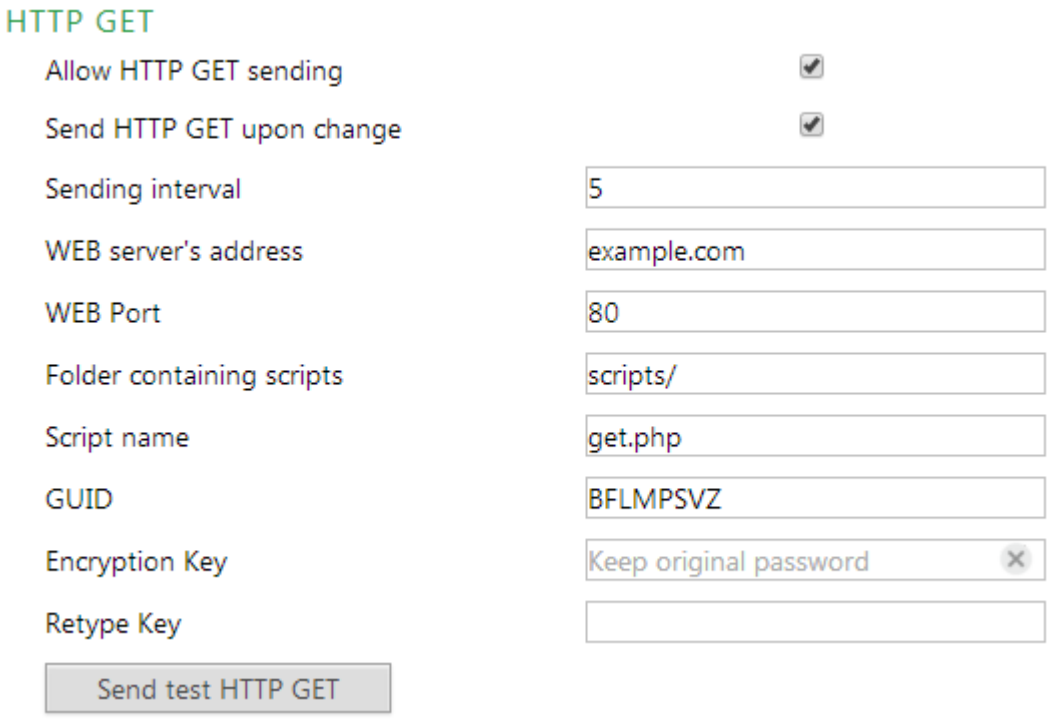


fig. 12 - data sending via HTTP GET

If the sending interval is set to zero, the sending function is turned off. The interval can be set from 0 to 1440 minutes.

If a sensor is set as *Unconnected*, its parameters are not sent in GET.

If you enter an encryption key of 16 characters, the HTTP GET data is encrypted by 128-bit AES cipher (Rijndael), the CFB method.

GET Format

Example of periodic GET:

(GET comes from a PAPAGO unit that has one combined and one temperature sensor. & charracters are deleted for better readability.)

*script.php?**mac**=0080A397EB59 **type**=Papago 1TH 2DI 1DO ETH **guid**= **description**=PER **date_time**=02/12/2016 12:38:40 **in1_name**=Input 1 **in1_state**=0 **in1_conv**=1 **in1_units**=m3 **in1_raw**=1 **in2_name**=Input 2 **in2_state**=0 **in2_conv**=2.000 **in2_units**=kWh **in2_raw**=2000 **out1_name**=Output **out1_state**=0 **T1V1_value**=29.0 **T1V1_units**=°C **T1V1_status**=2 **H1V2_value**=43.2 **H1V2_units**=% **H1V2_status**=0 **D1V3_value**=15.2 **D1V3_units**=°C **D1V3_status**=0*

Example of GET after pressing the button in the settings:

*script.php?**mac**=0080A393A273&**type**=Papago%202PT%20ETH &**guid**=PAPAGO-TEST-GUID&**description**=TEST*

Example of encrypted GET after pressing the button in the settings:

*script.php?**encrypted_data**=%DC%BD%5D%C1%DE%C4%0A%66%8B%69%0C%6D%8D %70%B9%11%EA%8C%19%2A%93%F1%71%87%B7%47%94%77%C7%A2%71%D9%1 A%3D%BA%21%CF%0D%D5%42%1F%01%23%7B%AF%31%C9%6D%D6%EC%87%C4*

%39%E4%76%84%29%A9%C1%31%74%05%31%3F%96%43%13%3C%73%08%D6%8F
%56%F5%6C%A2%77%53%C6%A7%10%8F%47%A5%A7%2D%04%9B%58%A0%94

The following parameters are sent in GET:

OBECNÉ

description..... Indicates a standard GET with measurement (LOG), GET sent when exceeding a limit (WATCH) or a test GET sent when you press the button on the Web (TEST). GET with measurement and GET sent when exceeding a limit contain the same data

mac ..... MAC address of the device.

type ..... Type designation of the device.

guid ..... Unique user-specified text string.

log_index..... The serial number of the record in a circular buffer.[2]

date_time ..... Date and time of recording in the format mm/dd/yyyy hh:mm:ss.

encrypted_data This parameter contains the data of encrypted GET.

INPUTY

inX_name..... Input name.

inX_state ..... Input state, current value 0 (off) or 1 (on).

inXconv ..... Current input counter state converted by the given parameters to a set unit.

inX_units ..... Unit for converted counter state.

inX_raw ..... „Raw“ counter value before conversion.

OUTPUT

out1_name ..... Input name.

out1_state ..... Output state, current value 0 (off) or 1 (on).

SENSOR

The following parameters may be given more than once if there are more measured values from one sensor. The first character may be either T (for temperature), H (for humidity) or D (dew point). Thus, Papago 2PT with two temperature sensors, provides these parameters:

T1V1[3]*_value*.... The first temperature as a decimal number.

T2V1_value..... The second temperature as a decimal number.

T1V1_units..... The unit of the first measured temperature.

T2V1_units..... The unit of the second measured temperature.

[2] This number applies if the network connection to the device has been discontinued for some time. After restoring the network connection, all accumulated GETs are sent in the form of a circular buffer. The buffer has a capacity of 120 entries.

[3] The number after the letter T indicates the serial number of the connector on the device. The number after the letter V indicates the serial number of the parameter from the connected sensor.

T1V1_status..... Status of the first value: the value is OK (0), upper limit exceeded (2), lower limit exceeded (3) invalid value (4).

T2V1_status..... Status of the second value: the value is OK (0), upper limit exceeded (2), lower limit exceeded (3) invalid value (4).

CH1_name..... Channel 1 name.

CH2_name..... Channel 2 name.

Answer to the HTTP GET

If you want to send a command to change output state within the http GET answer, or subtract a value from the counter state, the server should send the answer in XML format. Answer then should contain attributes *out1* and *cnt1* or *cnt2*, which can set the output and / or subtract a value from the counter state. (XML can only contain some of these attributes.) Values shall be sent in this format:

```
<root>
  <set valid="1" out1="1" cnt1="7" cnt2="5.5" />
</root>
```

If the HTTP GET is encrypted, the answer to it has to be encrypted as well and with a following format (the whole length must be no longer than 250 characters):

```
<root>
  <set
  encrypted_data=%DC%BD%5D%C1%DE%C4%0A%66%8B%69%0C%6D%8D%70%B9%11%EA%8C%1
  9%2A%93%F1%71%87%B7%47%94%77%C7%A2%71%D9%1A%3D%BA%21%CF%0D%D5%42%1F%01/
  >
</root>
```

Setting counters and output by an HTTP GET

Using HTTP GET, you can change counters or output states via *set.xml* script. This script accepts non-encrypted messages only. Papago understands these commands:

- **Setting counter to a value**

  set.xml?type=m&id=2&val=156

  Parameter *id* is counter number, counted from 1. Parameter *val* is a new counter value. Papago expects whole or decimal number depending on the number of decimals set for this counter.

- **Subtract value from counter state**

  set.xml?type=n&id=1&val=37.2

  Parameter *id* is counter number, counted from 1. Parameter *val* is a value you want to subtract from the counter state. Papago expects whole or decimal number depending on the number of decimals set for this counter.

- **Turn output on**

  set.xml?type=s&id=1

- **Turn output off**

  set.xml?type=r&id=1

- **Send a pulse to the output**

  set.xml?type=p&id=1

Answer to the sent GET is XML in this format:

```
<root>
  <result status="1" />
</root>
```

Should the attribute *status* be 0, it means the command was not performed because it contains errors or unexpected values.

Inputs and outputs configuration section

Inputs and outputs operation mode configuration.

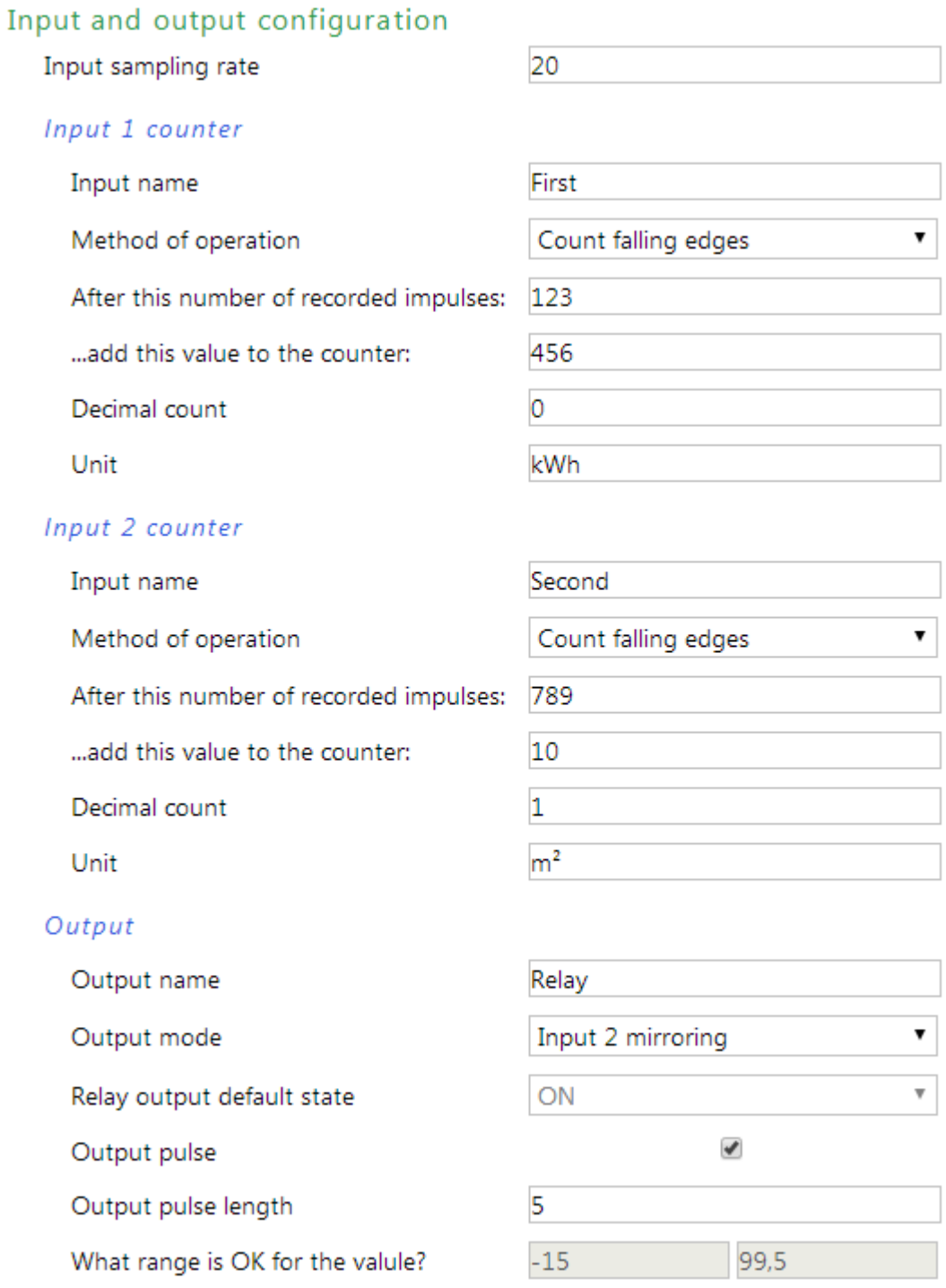


fig. 13 – inputs and outputs settings

Inputs sampling rate is common for both inputs and represents time of how long a value has to be on the input to be considered a valid one.

Operation mode is essentially a way of counting pulses on the input. Counter can be *off* or it can count leading edges, falling edges or both edges. Once the counter is on, conversion settings can be modified to convert the pulses to a real value. (For example, if the connected electricity gauge has a resolution of 100 pulses per kWh, enter this conversion, write kWh to the *Unit* field and the main page will display the consumption in kWh

You can also set **Operation more** on the **output**. One of these can be applied:

- Manual control
- Pulse control
- Input 1 mirroring
- Input 2 mirroring
- Thermostat for temperature
- Thermostat for humidity
- Thermostat for dew point

Once *Manual* or *Pulse control* is set, default *output* relay contact state can also be set. This state will be on after a power up or a reboot of the Papago unit.

In *Pulse mode* the output *pulse length* is set to define how long the relay should be on. Then the pulse of this length can be sent from the main page or via HTTP GET.

In the *Out of limits watching* mode, the two limit fields are used. Once the watched value goes out of these limits, the *output* relay will be turned on.

Sensor Section

Sensor and limits configuration.

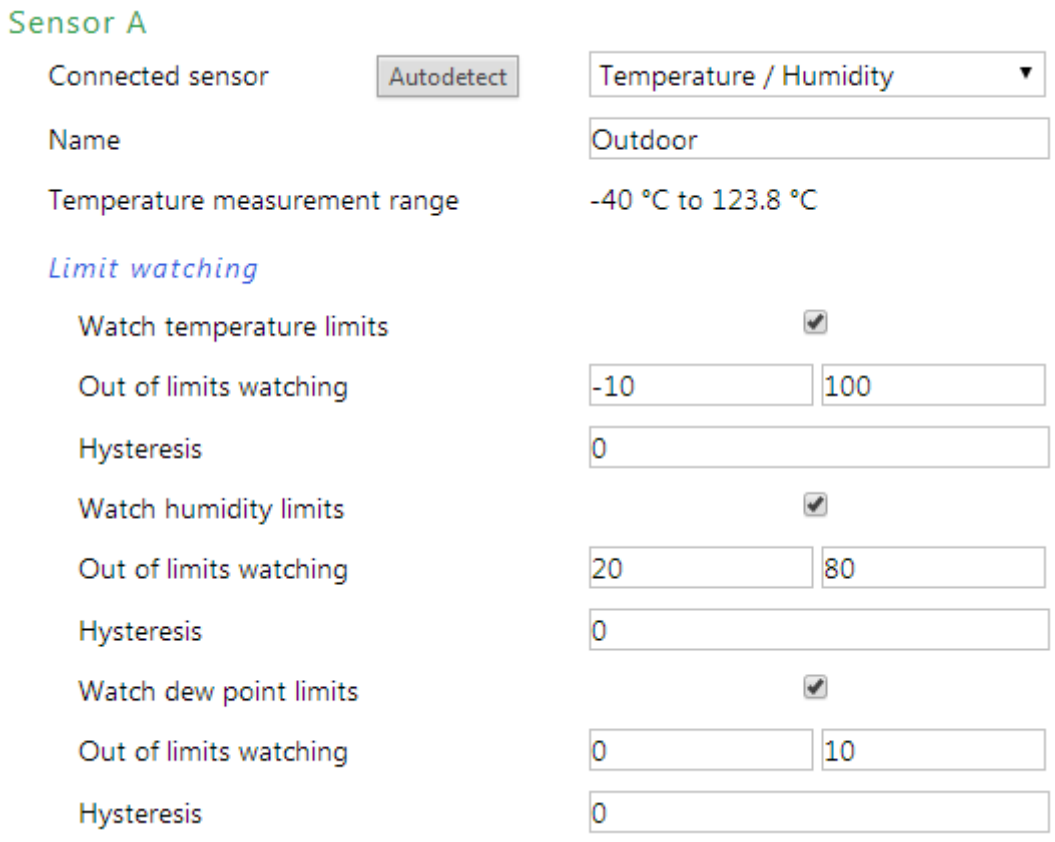


fig. 14 - configuration of one of the sensors

By pressing the *Autodetect* button, all settings for Sensor A and/or B are done automatically according to the currently connected sensor(s), above all the right type of the sensor is entered in the field *Connected sensor*.

Other Settings

This section allows you to set the time, temperature unit, language of the website, etc…

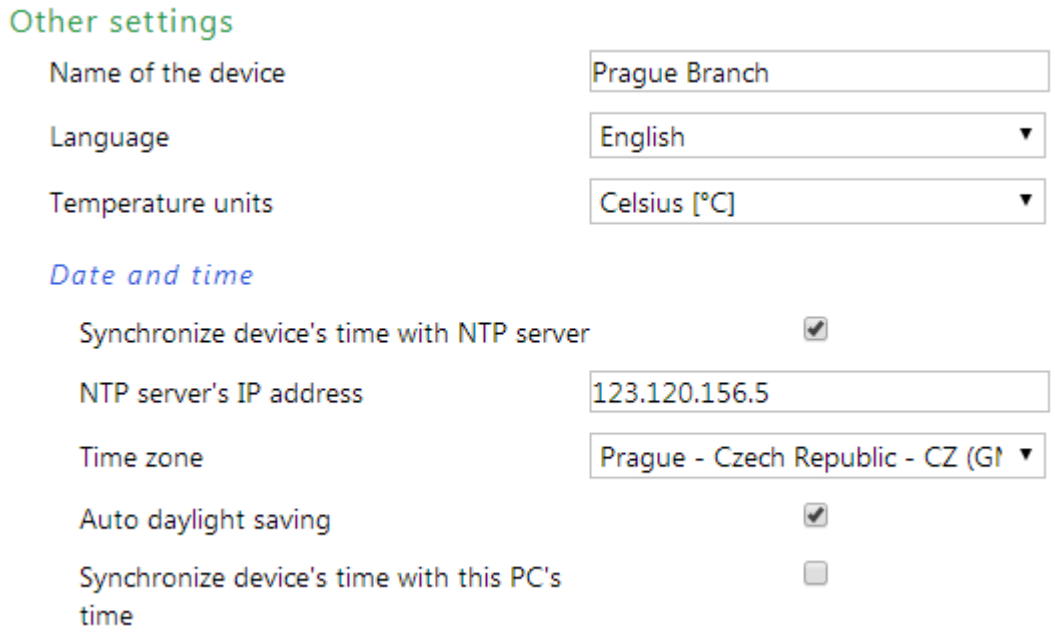


fig. 15 - other settings

The available languages are Czech or English; for temperature units you can choose between degrees Celsius, Fahrenheit or Kelvin.

CONFIGURATION VIA TELNET PROTOCOL

Connection

IP address is not known

It is recommended that the IP address should be set using the Ethernet Configurator software (for more information see page 7).

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 192.168.1.254 to the module MAC address:

      `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 click Enter. [4])
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 device 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.

[4] In OS Windows Vista or higher, the client for Telnet is not a standard part of the system. Install it using the following procedure:

a) Open the “Control Panels/Programs and Features” menu.
b) On the left, click “Enable or disable features of Windows system” (this option requires the administrator to log in).
c) The “Features of Windows system” window displays. Here tick the “Telnet service Client” field and click Ok. The client for Telnet will be installed.

IP address is known

1) In OS Windows choose Start/Run, enter `telnet` in the provided line and press Enter. [4]
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 device 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
  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 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 IP address. The system then prompts you to change the Gateway IP address. The digits 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 form: 11111111 11111111 11111111 00000000) =. number 8.
The mask 255.255.255.252 (binary form: 11111111 11111111 11111111 11111100) = 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 or via WEB interface (admin password).

For 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

It is possible to obtain the last measured values, limits (thresholds) and device name from the device in the form of a text file in the XML format. The file is available at *http:///[IP-adresa]/fresh.xml* – i.e. for example at http://192.168.1.254/fresh.xml for the default settings.

```
<root xmlns="http://www.papouch.com/xml/papago/act">
    <sns id="1" name="Sensor A" type="1" status="0" unit="0" val="19.2" w-min="-40.0" w-max="123.8"
                                type2="2" status2="0" unit2="0" val2="24.8" w-min2="" w-max2="40.0"
                                type3="3" status3="0" unit3="0" val3="-1.3" w-min3="-40.0" w-max3="123.8"/>
    <din id="1" name="Elektromer" bin="0" val="1100 kWh" raw="1100000"/>
    <din id="2" name="Sauna" bin="1" val="1689 kWh" raw="1689"/>
    <dout id="1" name="Rele" bin="0" mode="1" />
    <status location="U Papoucha" signal="0" time="05/20/2016 13:27:08"/>
</root>
```

fig. 16 – example of XML with actual values

The XML file contains a *sns* tag for each measured parameter as well as the *status* tag:

s t a t u s

location

User-defined name of the device.

time

The current system time of the device in the format *mm/dd/yyyy hh:mm:ss*.

s n s

id

The serial number of the measured variable. (The first number is 1.)

name

The name of the sensor.

type, type2, type3

Number 1 (parameters of temperature), 2 (parameters of humidity) or 3 (dew point).

status, status2, status3

It describes the status of the measured value. Can show the following values:

0 ..... the value is valid and shows the currently measured value
2 ..... the measured value exceeds the user-set upper limit
3 ..... the measured value exceeds the user-set lower limit
4 ..... measurement error or sensor error (damaged sensor or cable)

unit, unit2, unit3

The number represents the code the chosen temperature unit. It can have the following values:

0 ..... degrees Celsius
1 ..... degrees Fahrenheit
2 ..... degrees Kelvin

val, val2, val3

The currently measured value as a decimal number, with an accuracy of one- or two-tenths depending on the extent and type of the sensor. (For validity of the value see *status*.)

w-min, w-min2, w-min3, w-max, w-max2, w-max3

The lower (*w-min*) and upper (*w-max*) limit of the value set by the user. The value is shown as a decimal number with an accuracy of one-tenth.

d i n

id

Input number. (Starts with 1.)

name

User input name.

bin

Current input state as a number 0 (off) or 1 (on).

val

Input counter converted based on the set parameters to a set value. String also contains the user specified symbol of the unit.

raw

Current counter state before conversion.

d o u t

id

Output number. (Starts with 1.)

name

User output name.

bin

Current output state as a number 0 (off) or 1 (on).

mode

Output mode based on user settings. Shown as a number from 0 to 6:

0) Manual control
1) Pulse control
2) Input 1 mirroring
3) Input 2 mirroring
4) Thermostat for temperature
5) Thermostat for humidity
6) Thermostat for dew point

SNMP

The SNMP protocol (version 1) contains objects with individual values. For a detailed description of the objects see bellow. The MIB table you can import into your SNMP manager can be downloaded from papouch.com.

papouchProjekt		
papago_1th_2di_1do		
version		
deviceVar		
deviceName		
deviceName.0	DisplayString	NONAME
psAlarmString		
psAlarmString.0	DisplayString	(zero-length)
in_table		
inputsTable		
inputsEntry		
inState		
inState.1	INTEGER	0
inState.2	INTEGER	0
inCounter		
inCounter.1	Counter	1
inCounter.2	Counter	2000
inDecNum		
inDecNum.1	INTEGER	0
inDecNum.2	INTEGER	3
inUnit		
inUnit.1	DisplayString	m3
inUnit.2	DisplayString	kWh
out_table		
outputsTable		
outputsEntry		
outState		
outState.1	INTEGER	1
channel_table		
channelTable		
channelEntry		
inChType		
inChType.1	INTEGER	1
inChType.2	INTEGER	2
inChType.3	INTEGER	3
inChStatus		
inChStatus.1	INTEGER	0
inChStatus.2	INTEGER	0
inChStatus.3	INTEGER	0
inChValue		
inChValue.1	INTEGER	261
inChValue.2	INTEGER	309
inChValue.3	INTEGER	76
inChUnits		
inChUnits.1	INTEGER	0
inChUnits.2	INTEGER	0
inChUnits.3	INTEGER	0
channelEntry.5		
channelEntry.5.1	ChannelEntry	null

fig. 17 – SNMP objects

Objects – variables

Input – State

Name: inState

Object ID: 1.3.6.1.4.1.18248.34.1.2.1.1.1.1

Description: Input state as a number 0 or 1.

Input – Counter value

Name: inCounter

Object ID: 1.3.6.1.4.1.18248.34.1.2.1.1.2.1

Description: Converted counter value as a whole number. (You can subtract set value from the counter state by writing the value here). Following number of decimals has to be applied to the value to get the converted counter value.

Input – Number of decimals

Name: inDecNum

Object ID: 1.3.6.1.4.1.18248.34.1.2.1.1.3.1

Description: Number of decimals as a whole number. This number of decimals has to be applied to the value to get the converted counter value.

Input – Unit

Name: inUnit

Object ID: 1.3.6.1.4.1.18248.34.1.2.1.1.4.1

Description: User defined unit that specifies the converted value.

Output – State

Name: outState

Object ID: 1.3.6.1.4.1.18248.34.1.3.1.1.1.1

Description: Output state as a number 0 (off) or 1 (on).

Type

Name: inChType

Object ID: 1.3.6.1.4.1.18248.34.1.4.1.1.1 to 3 [5]

Description: The type of this value. It can have one of the following values:

0 → Not used.
1 → Temperature.
2 → Humidity.
3 → Dew point.

Status

Name: inChStatus

Object ID: 1.3.6.1.4.1.18248.34.1.4.1.1.2.1 to 3 [5]

Description: The status of this value. It describes the current status of the measured value. It can have one of the following values:

0 → The value is valid and within the limits.
1 → The value has not yet been measured.
2 → The value is valid and exceeds the upper limit.
3 → The value is valid and exceeds the lower limit.
4 → The value is invalid – measurement error.

Measured value

Name: inChValue

Object ID: 1.3.6.1.4.1.18248.34.1.4.1.1.3.1 to 3 [5]

Description: The measured value as an integer. To obtain the real value, divide by ten.

Unit

Name: inChUnits

Object ID: 1.3.6.1.4.1.18248.34.1.4.1.1.4.1 to 3 [5]

Description: Unit of the value. May contain one of the following values:

0 → degrees Celsius.
1 → degrees Fahrenheit.
2 → degrees Kelvin.
3 → percentage (humidity)

SNMP objects – general

The following two objects relate to the entire device.

Device name

Name: deviceName

Object ID: 1.3.6.1.4.1.18248.31.1.1.1.0

Description: User-defined device name.

Alarm text

Name: psAlarmString

Object ID: 1.3.6.1.4.1.18248.31.1.1.2.0

Description: Text of the alarm message sent when a threshold is exceeded.

[5] The ID of the objects shows the values from sensors A and B arranged one after another. First A, then B. The values are arranged in the order of temperature, humidity, dew point, i.e. there are 2 or 6 objects.

Traps

Trap 1 – Value is outside the limits

The trap contains the measured value and the limit that was exceeded.

The trap is only sent when one of the limits has been exceeded. The trap can only be delivered to a properly configured IP address of a PC with the SNMP manager.

Trap 2 – Current measured values

The trap contains all current values as well as the name of the device set by the user.

The trap is sent only if a non-zero frequency of sending has been set.

MODBUS TCP

For the initial configuration of the address, etc. it is recommended to use, for example, the ModbusConfigurator program, which can be downloaded here:

http://www.papouch.com/cz/website/mainmenu/software/modbus-configurator/

Output

Output state reading

To access these values, use *Read Coils* function.

Address	*Access*	*Function*	*Name*
0	read	0x01	**Output 1state** 0 = output is off 1 = output is on

Setting output state

To access these values, use *Write Single Coil* or *Write Multiple Coils* function.

Address	*Access*	*Function*	*Name*
0	write	0x05 0x0F	**Output 1 state** 0 = output is off 1 = output is on

Input state read

To access these values use *Read Discrete Inputs* function.

Address	*Access*	*Function*	*Name*
0 – 1	read	0x02	**Inputs 1 and 2 states** 0 = Input is off 1 = Input is on

Counters

Read counters states

To access these values use *Read Holding Register* function.

Address	*Access*	*Function*	*Name*
Counter 1			
0	read	0x03	**Function** Operation mode as one of these numbers: 0 = not used (turned off in configuration) 1 = counts leading edges 2 = counts falling edges 3 = counts both edges
1, 2	read	0x03	**Date and time** Date and time in format according to the NTP.

Address	*Access*	*Function*	*Name*
3, 4	read	0x03	**Counter value as a whole number** Following register and number of decimals has to be applied to the value to get the converted counter value.
5	read	0x03	**Number of decimals** Number of decimals as a whole number. This number of decimals has to be applied to the previous value to get the converted counter value.
6, 7	read	0x03	**Counter state as a decimal number** State of the counter as a decimal number (32 bit float - IEEE 754).
Counter 2			
from 100	Counter 2 values.		

Counter state setting

To access these values use *Write Multiple Registers* function.

Address	*Access*	*Function*	*Name*
Counter 1			
3, 4	write	0x10	**Counter value as a whole number** Enter counter value as a whole number. Number of decimals will be taken from the decimal settings from WEB configuration.
6, 7	write	0x10	**Counter value as a decimal number** Enter value as a decimal number (32 bit float - IEEE 754).
8, 9	write	0x10	**Value subtraction – as a whole number** Enter counter value as a whole number. This number will be subtracted from the current counter state.**Chyba! Záložka není definována.** Number of decimals will be taken from the decimal settings from WEB configuration.
10, 11	write	0x10	**Value subtraction – as a decimal number** Enter value as a decimal number (32 bit float - IEEE 754).This number will be subtracted from the current counter state.**Chyba! Záložka není definována.**
Counter 2			
From 103	Counter 2 values.		

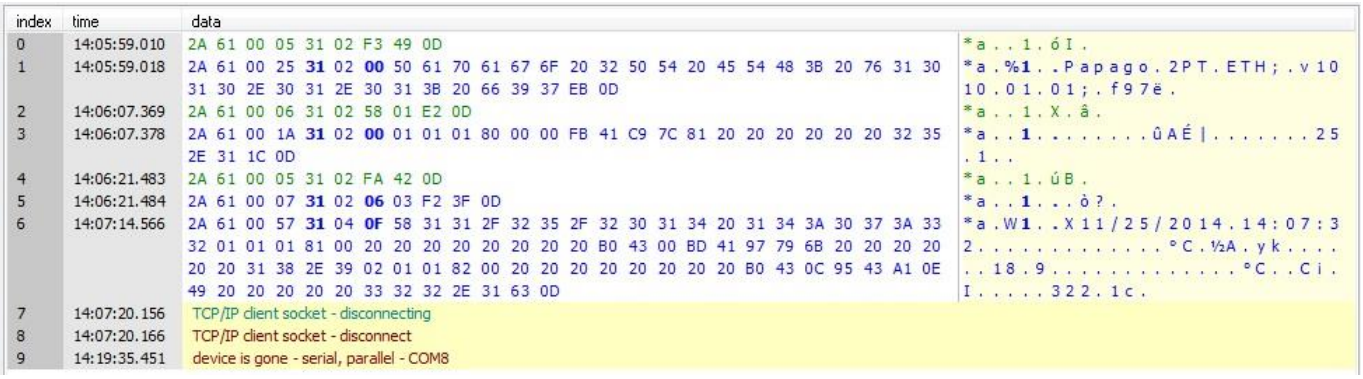
Sensor

To access these values use *Read Input Register* function.

Address	*Access*	*Function*	*Name*
Sensor 1 – head			
0	read	0x04	**Status** Contains the status of the sensor. Possible values: 0 = this sensor is not used (set to Not Connected in the configuration) 1 = this sensor is used for measuring
1, 2	read	0x04	**Date and time** Date and time of the device in the format of NTP.
Sensor 1 – the first parameter (temperature)			
10	read	0x04	**Status of the measured values** Status of the measured values. Options: 0 = the measured value is within the measuring range 2 = exceeded upper limit of the measuring range (overflow) 3 = exceeded lower limit of the measuring range (underflow) 4 = the measured value is invalid
11	read	0x04	**Value in the form of signed integer**
12	read	0x04	**Value in the float format** The upper two bytes.
13	read	0x04	**Value in the float format** The lower two bytes.
14	read	0x04	**Unit** The unit in which information is stored in the previous registries. 0 = °C, or % for humidity 1 = °F 2 = K
Sensor 1 – the second parameter (humidity)			
20 to 24			
Sensor 1 – the third parameter (dew point)			
30 to 34			
Sensor 2			
from 100			

SPINEL

The device contains the standard Spinel protocol (format 97) for communication via the TCP data channel. Spinel Terminal has been designed for comfortable communication via Spinel.



index	time	data	
0	14:05:59.010	2A 61 00 05 31 02 F3 49 0D	*a..1.óI.
1	14:05:59.018	2A 61 00 25 **31** 02 **00** 50 61 70 61 67 6F 20 32 50 54 20 45 54 48 3B 20 76 31 30 31 30 2E 30 31 2E 30 31 3B 20 66 39 37 EB 0D	*a.%**1**..Papago.2PT.ETH;.v1010.01.01;.f97ë.
2	14:06:07.369	2A 61 00 06 31 02 58 01 E2 0D	*a..1.X.â.
3	14:06:07.378	2A 61 00 1A **31** 02 **00** 01 01 01 80 00 00 FB 41 C9 7C 81 20 20 20 20 20 20 32 35 2E 31 1C 0D	*a..**1**.......ûAÉ|.......25.1..
4	14:06:21.483	2A 61 00 05 31 02 FA 42 0D	*a..1.úB.
5	14:06:21.484	2A 61 00 07 **31** 02 **06** 03 F2 3F 0D	*a..**1**...ò?.
6	14:07:14.566	2A 61 00 57 **31** 04 **0F** 58 31 31 2F 32 35 2F 32 30 31 34 20 31 34 3A 30 37 3A 33 32 01 01 01 81 00 20 20 20 20 20 20 20 20 B0 43 00 BD 41 97 79 6B 20 20 20 20 20 20 31 38 2E 39 02 01 01 82 00 20 20 20 20 20 20 20 20 B0 43 0C 95 43 A1 0E 49 20 20 20 20 20 33 32 32 2E 31 63 0D	*a.W**1**..X11/25/2014.14:07:32.............°C.½A.yk.....18.9.............°C..Ci.I.....322.1c.
7	14:07:20.156	TCP/IP client socket - disconnecting	
8	14:07:20.166	TCP/IP client socket - disconnect	
9	14:19:35.451	device is gone - serial, parallel - COM8	

fig. 18 - communication with the device using the Spinel Terminal program

Summary of implemented instructions:

Temperature reading

This instruction reads the current measured values. The values are converted to the currently selected temperature unit. The measured values are returned as a sign integer, as a value in the float format and as an ASCII string.

Request:

Instruction code: 58H

Parameters: (sensor)

sensor	Sensor No.	length: 1 byte
The number of the sensor to be read. It is possible to choose 01H (sensor a) or 02H (sensor b).		

Response:

Acknowledgement code: ACK 00H

Parameters: {(sensor$_1$)(variable$_1$)(type$_1$)(status$_1$)(unit$_1$)(unita$_1$)(value$_1$)} {…}

sensor	Sensor No.	length: 1 byte
This bytes indicates the sensor number and applies to all subsequent bytes until the next *chn* byte. This means that the following bytes belong to the channel with that number. It is numbered from 01H.		

variable	Variable No.	length: 1 byte
The number of the variable from the given sensor. Numbered from 01H.		

type	Variable type	length: 1 byte
The type of the variable can have one of the following values: 00H ..... not defined 01H ..... temperature 02H ..... humidity 03H ..... dew point		

status	Status of the measured value	length: 1 byte
The status of the measured value for the channel with the number given in the previous *chn*.		
bit 0 (LSb)	0 = the **lower limit of the monitored range** was not exceeded	
	1 = the lower limit of the monitored range was exceeded	
bit 1	0 = the **upper limit of the monitored range** was not exceeded	
	1 = the upper limit of the monitored range was exceeded	
bit 2	0 = the **lower limit of the measuring range** was not exceeded	
	1 = the lower limit of the measuring range was exceeded	
bit 3	0 = the **upper limit of the measuring range** was not exceeded	
	1 = the upper limit of the measuring range was exceeded	
bit 7 (MSb)	0 = the measured value is invalid	
	1 = the measured value is valid	

unit	Unit	length: 1 byte
Unit code: 0 for °C, 1 for °F or 2 Kelvin.		

units	Unit in ASCII string	length: 10 bytes
Unit Code as a right-aligned ASCII string. For example °C, °F, etc.		

value	Measured value	length: 16 bytes
The measured value from the channel with the number given in the *chn* byte. The values are sent simultaneously in three different formats. The first is a 16bit sign value (integer in the form of MSB:LSB), followed by two values converted for the current range based on the current setup: in the 32 bit float format according to IEEE 754[6] and in the ASCII format. The values are given in the aforementioned order. *Example:* The value of 9215.85 is expressed as follows: `0AH,58H,46H,0FH,FFH,66H,20H,20H,20H,39H,32H,31H,35H,2EH,38H,35H` INT part: `0AH,58H` (2648) IEEE 754 part: `46H,0FH,FFH,66H` ASCII part: `20H,20H,20H,39H,32H,31H,35H,2EH,38H,35H` (9215.85)		

Examples:

Request – read channel 1:
`2AH,61H,00H,06H,31H,02H,58H,01H,E2H,0DH`
Response:
`2AH,61H,00H,1AH,31H,02H,00H,01H,01H,01H,80H,00H,00H,EEH,41H,BEH,D6H,C3H,20H,20H,20H,20H,20H,32H,33H,2EH,38H,93H,0DH`
The value measured on channel 1 was 21,74. Channel number: `01H` Variable number: `01H` Variable type: `01H` Value status: `80H` Unit: `00H`

[6] The description of the IEEE 754 standard is available here: http://en.wikipedia.org/wiki/IEEE_754

INT part: 00H,EEH (5434)
IEEE 754 part: 41H,BEH,D6H,C3H
ASCII part: 20H,20H,20H,20H,20H,00H,32H,33H,2EH,38H (21.74)

Read input states

Reads current inputs states.

Request:

Instruction code: 31H

Response:

Acknowledge code: ACK 00H

Parameters: (state)

state	Inbut states in bits	length: 1 byte
Bit oriented byte containing Input states where lowest bit (LSb) represents IN1 state, second lowest bit IN2 state.		

Examples:

Request:
2AH,61H,00H,05H,FEH,02H,31H,3EH,0DH
Response:
2AH,61H,00H,06H,31H,02H,00H,02H,39H,0DH

Read output state

Reads current output states.

Request:

Instruction code: 30H

Response:

Acknowledge code: ACK 00H

Parameters: (state)

state	Output state	length: 1 byte
00H = output off 01H = output on		

Examples:

Request:
2AH,61H,00H,05H,FEH,02H,30H,3FH,0DH
Response:
2AH,61H,00H,06H,31H,02H,00H,01H,3AH,0DH

Output settings

Sets output relay to the requested state.

Request:

Instruction code: 20H

Parameters: (state)

State	Output state	length: 1 byte
01H = off 81H = on		

Response:

Acknowledge code: ACK 00H

Examples:

Request:
2AH,61H,00H,06H,FEH,02H,20H,81H,CDH,0DH
Response:
2AH,61H,00H,05H,31H,31H,00H,0DH,0DH

Reading counters

Instruction reads one or more counters.

Request:

Instruction code: 60H

Parameters: (counter)

counter	Sensor number	length: 1 byte
Counter number to read. Enter value 00H for all counters or numbers from 01H to 05H for individual counters.		

Response:

Acknowledge code: ACK 00H

Parameters: {[channel][value][status][int][float][str][unit][decimals][rawint][rawstr]} {…}

channel Input number	id: 00H length: 1 byte
Input number from 1 to 5.	

value Current input state	id: 01H length: 1 byte
Current input state as a value 00H (off) or 01H (on).	

status Counter operation mode	id: 02H length: 1 byte
Can contain these operation mode codes: 00H … no actions	

int Counter value as a whole number	id: 03H length: 4 bytes
Counter number after conversion as a whole number. (Real value can be obtained by getting the number of decimals. Number of decimals is stored in decimals parameter.)	

float Counter value as a decimal number	id: 04H length: 4 bytes
Counter value as a decimal number after conversion (32 bit float - IEEE 754).	

str Counter value as a string	id: 05H length: 10 bytes
Counter value as a string. Decimal separator is a dot. String is right-aligned.	

unit Unit	id: 06H length: 10 bytes
User defined unit. String is right-aligned	

decimals Number of decimals	id: 07H length: 1 byte
Number of decimals to convert the real value by and view it.	

rawint Raw value as a whole number	id: 08H length: 4 bytes
Counter value before conversion as s whole number. (Real value can be obtained by getting the number of decimals. Number of decimals is stored in decimals parameter.)	

rawstr Raw value as a string	id: 09H length: 10 bytes
Counter value before conversion as a string. As a decimal divider a dot is used. String is aligned to the right.	

Examples:

Request – read channel 1:	
`2AH,61H,00H,06H,FEH,01H,60H,01H,0EH,0DH`	
Response:	
`2AH,61H,00H,3DH,31H,01H,00H,`	
`00H,01H,`	- counter number: 0
`01H,00H,`	- input state: 0
`02H,00H,`	- counter state
`03H,00H,00H,00H,D2H,`	- counter value as a whole number
`04H,43H,52H,00H,00H,`	- counter state as a decimal
`05H,20H,20H,20H,20H,20H,20H,20H,32H,31H,30H,`	- counter state as a string
`06H,20H,20H,20H,20H,20H,20H,20H,C2H,B0H,43H,`	- unit as a string
`07H,00H,`	- number of decimals
`08H,00H,00H,00H,D2H,`	- raw value as a whole number

```
09H,20H,20H,20H,20H,20H,20H,20H,32H,31H,30H,    - raw value as a string
23H,0DH
```

Setting or subtracting counter value

Instruction sets given counter or subtracts a value from its current value.

Request:

Instruction code: 65H

Parameters: {[channel][operation][status][int][float][str][unit][decimals][rawint][rawstr]} {…}

channel Input number	id: 00H length: 1 byte
Input number as 1 to 2.	

operation Operation type	id: 01H length: 1 byte
Operation to perform is subtraction (01H) or setting (02H).	

int Value as a whole number	id: 02H length: 4 bytes
Value to set/subtract as a whole number.	

str Value as a string	id: 03H length: 10 bytes
Value to set/subtract as a string. Decimal separator is a dot. String is right-aligned.	

float Value as a decimal number	id: 04H length: 4 bytes
Value as a decimal number (32 bit float - IEEE 754).	

Response:

Acknowledge code: ACK 00H

Reading of name and version

Reads the name of the device, software version and the list of possible communication formats. Set by the manufacturer.

Request:

Instruction code: F3H

Response:

Acknowledgement code: ACK 00H

Parameters: (string)

string	Name and version	length: 1 byte

Papago 1TH 2DI 1DO ETH; v1075.01.03; f97

In addition to the information described above, the string can also contain other information in sections introduced by a semicolon, space and a small letter to determine which information follows.

Examples:

Request:
2AH,61H,00H,05H,31H,02H,F3H,49H,0DH
Response:
2AH,61H,00H,2DH,31H,02H,00H,50H,61H,70H,61H,67H,6FH,20H,31H,54H,48H,20H,32H,44H,49H,20H,31H,44H,4FH,20H,45H,54H,48H,3BH,20H,76H,31H,30H,37H,35H,2H,30H,31H,2EH,30H,33H,3BH,20H,66H,39H,37H,1CH,0DH

Reading of manufacturing data

This instruction reads the manufacturing data of the device.

Request:

Instruction code: FAH

Response:

Acknowledgement code: ACK 00H

Parameters: (product_number)(serial_number)(other)

product_number	length: 2 bytes
Product number. For a device number 0227.00.03/0001 this number is 227.	

serial_number	length: 2 bytes
Serial number. For a device number 0227.00.03/0001 this number is 1.	

other	length: 4 bytes
Other manufacturing information.	

Examples:

Request:
2AH,61H,00H,05H,FEH,02H,FAH,75H,0DH

Automatic message

This response is generated when the preset limits are exceeded or when the measured value exceeds the physical range of the sensor. The message may contain information about one or more channels.

Acknowledgement code: ACK 0FH

Parameters: [event][time] {[sensor][variable][type][status][unit][unitA][value]} {…}

event Number of the event source	length: 1 byte
This byte specifies the event source. It can be used to distinguish the automatic message sent when the limits or measuring range are exceeded from other automated messages from the device. The value of this byte is 30H.	

time time of the event	length: 19 bytes
Time of the event as a string in the format *mm/dd/yyyy hh:mm:ss*	

sensor sensor number	length: 1 byte
The serial number of the sensor the following bytes belong to. Numbering starts from 01H.	

variable variable number	length: 1 byte
The serial number of a variable from one sensor, used to distinguishing between different variables obtained from one sensor, if the sensor provides more than one. Numbering starts from 01H.	

type variable type	length: 1 byte
The type of the variable can have one of the following values: 00H .....not defined 01H .....temperature 02H .....humidity 03H .....dew point	

status Status of the measured value		length: 1 byte
bits 0 to 3 (lower nibble)	0000 = the measured value is within the measuring range	
	0001 = the lower limit of the monitored range was exceeded	
	0010 = the upper limit of the monitored range was exceeded	
	0100 = the lower limit of the physical range of the A/D converter was exceeded	
	1000 = the upper limit of the physical range of the A/D converter was exceeded	
bit 7 (MSb)	0 = the measured value is invalid	
	1 = the measured value is valid	

unit unit ID	length: 1 byte
The numerical designation of the unit: 00H .....°C 01H .....°F 02H .....K	

unitA unit as a string	length: 10 bytes
A right-aligned string designating the selected unit. For example "°C"	

value measured value	length: 16 bytes

The values are sent simultaneously in three different formats. The first is a 16bit sign value (integer in the form of MSB:LSB), followed by two values converted for the current range based on the current setup: in the 32 bit float format according to IEEE 754[7] and in the ASCII format. The values are given in the aforementioned order.

Example:

The value of 9215.85 is expressed as follows:
`0AH,58H,46H,0FH,FFH,66H,20H,20H,20H,39H,32H,31H,35H,2EH,38H,35H`

INT part: `0AH,58H` (2648)

IEEE 754 part: `46H,0FH,FFH,66H`

ASCII part: `20H,20H,20H,39H,32H,31H,35H,2EH,38H,35H` (9215.85)

Example:

Automatic response:

`2AH,61H,00H,57H,31H,04H,0FH,58H,31H,31H,2FH,32H,35H,2FH,32H,30H,31H,34H,20H,31H,34H,3AH,30H,37H,3AH,33H,32H,01H,01H,01H,81H,00H,20H,20H,20H,20H,20H,20H,20H,20H,B0H,43H,00H,BDH,41H,97H,79H,6BH,20H,20H,20H,20H,20H,20H,31H,38H,2EH,39H,02H,01H,01H,82H,00H,20H,20H,20H,20H,20H,20H,20H,20H,B0H,43H,0CH,95H,43H,A1H,0EH,49H,20H,20H,20H,20H,20H,33H,32H,32H,2EH,31H,63H,0DH`

Automatic information about exceeding the lower limit on channel 1 and the upper limit on channel 2. The meaning of the values sent for channel 1:

Instruction No,: `58H`
ASCII time: `31H,31H,2FH,32H,35H,2FH,32H,30H,31H,34H,20H,31H,34H,3AH,30H,37H,3AH,33H,32H`
Channel No.: `01H`
Variable No.: `01H`
Variable type: `01H`
Value status: `81H`
Units numerically: `00H`
Units in ASCII: `20H,20H,20H,20H,20H,20H,20H,20H,B0H,43H`
Current value:
In the form of INT: `00H,BDH`
In the form of float: `41H,97H,79H,6BH`
In the form of ASCII: `20H,20H,20H,20H,20H,20H,31H,3BH,2EH,39H`

[7] The description of the IEEE 754 standard is available here: http://en.wikipedia.org/wiki/IEEE_754

INDICATIONS

Two LEDs integrated in the Ethernet connector:

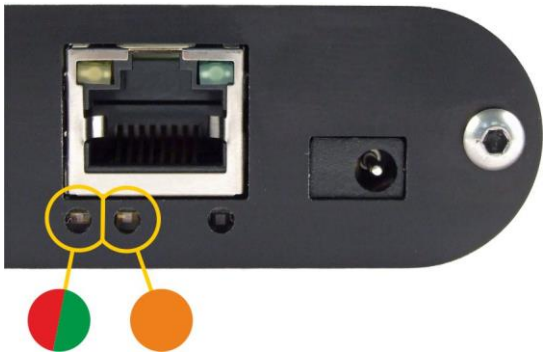
Yellow – LINK: is lit when the device is connected by cable to a switch or PC.

Green – ACT: indicates communication over the Ethernet.

Two LEDs to the left under the Ethernet connector:

Yellow (right): is lit when the connection is established via Spinel or Modbus.

Red-green (left):



- the green light is lit and the red light flashes when the device is working properly and is connected to at least one sensor
- the green and red LEDs are lit when the device works, but is not connected to any sensor
- the red LED is lit to indicate an error

Papago with WiFi connection:

Yellow-blue (right):

- Yellow lights up if Spinel or ModBus connection is established.
- Blue lights up when the Papago is connected to a WiFi network.

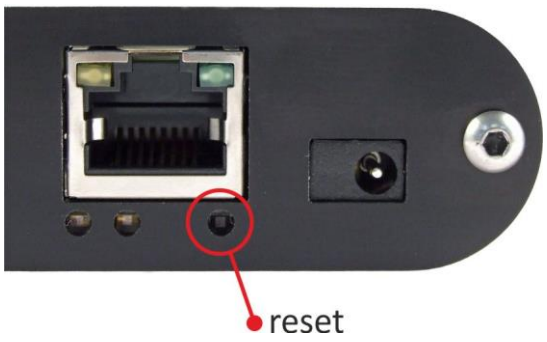
Red-Green (left):

- Green lights up and red flashes if the device is OK and at least one sensor is connected.
- Green and Red light up when the device is OK but no sensor is connected.
- Red lights up in case of device fault

RESET

Follow the instructions below to restore the default configuration set by the manufacturer. Unlike when the reset is performed via the web interface or using the Telnet protocol (see page 22), the IP address is also reset to the default value of 192.168.1.254.

1) Disconnect the device from the power supply.
2) Press the button located in a small hole on the right side under the Ethernet connector.
3) Turn on the power and wait for about 10 seconds until the yellow light below the Ethernet connector flashes 4 times.
4) Release the button.



TECHNICAL PARAMETERS

Integrated temperature and humidity sensor[8]

Important Notice: Polymer sensor is a highly sensitive element that reacts with chemicals. Do not expose even the outer shell of the sensor to chemicals or their vapors (cleaning with alcohol, petrol etc.). Especially organic solvents and compounds can negatively affect the sensor accuracy by as tens of percent RH.

Coverage ..... IP 54

Dimensions ..... 40 × 16 × 10 mm

Material ..... hardened aluminum

Humidity sensor

Humidity range ..... 0 % to 100 % RH

Recommended measurement range ..... 20 – 80 %

Resolution ..... 1% RH

Humidity measurement accuracy ..... see Fig. 19

Sensor element ..... polymer sensor

Sensor mechanical finish ..... inside hardened aluminum block

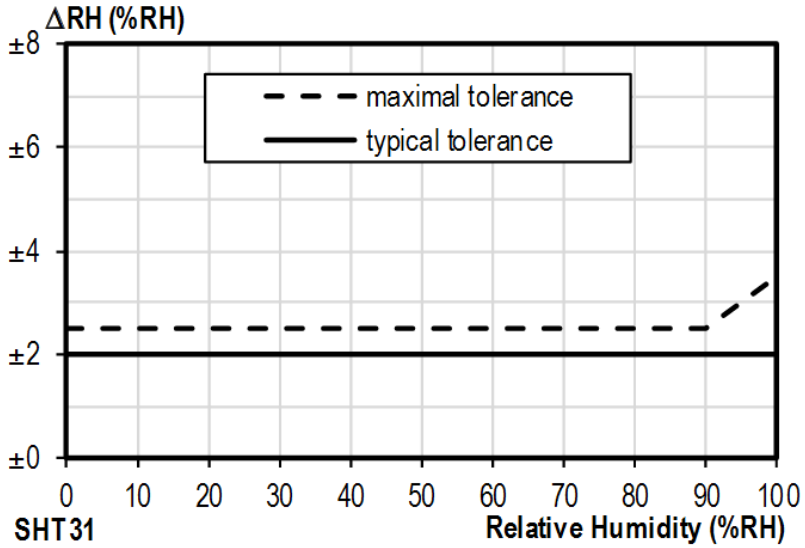


Fig. 19 – Accuracy of humidity measuring

[8] Sensor marked TH3 is supported in firmware including and above version 1.05. If you have an older firmware, you will have to flash the firmware to be able to read from TH3 sensor. Here are the key differences between the old version (Marked as TH2E) and TH3 version:

	TH3 (new sensor)	TH2E (old sensor)
Measurement accuracy within 0 – 10 %	±2 %	±2 to ±4 %
Measurement accuracy within 90 – 100 %	±2 %	±2 to ±4 %
Recommended measurement range	20 – 80 % RH	
Temperature measurement range	-40.0 °C to +125.0 °C	-40.0 °C to +123.8 °C
Temperature measurement accuracy	±0.3 to ±0.5 °C	±0.4 to ±2.0 °C

Operating and Maximum Range of Values

- Sensor is stable in standard range of humidity values. Long-term exposure to conditions outside these values (humidity above 80% in particular) can temporarily shift the measured-out values (by +3% for 60 hours). When the sensor is back to standard ranges, it returns to its pre-calibrated state slowly.[9]
- Long-term exposure to extreme conditions or to chemically aggressive vapor can speed up the aging process of the sensor significantly. It can also shift the measurements.

Temperature sensor

Range.....-40.0 °C to +125 °C

Resolution.....0.1 °C

Sensor element .....semiconductor

Sensor mechanical finish.....inside hardened aluminum block

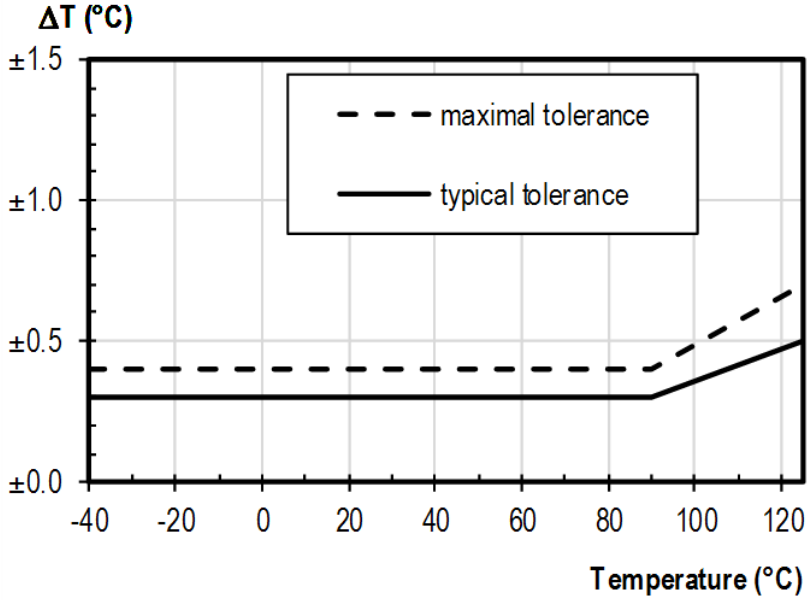


Fig. 20 – Accuracy of temperature measurement

Standalone temperature sensor

Sensor type .....semiconductor

Measuring temperature range .....-55 °C to +125 °C

Accuracy.....±0.5 °C in the range of -10 °C to +85 °C

Temperature drift.....±0.2 °C per 1000 hours at 125 °C

Dimensions.....normalized diameter 6 mm, length 60 mm

Housing material.....hardened alloy

Degree of protection .....IP68 (permanent immersion into 1m max.)

[9] You can speed up this process by doing following:
1) Leave the sensor in environment above 100 to 105 °C and humidity below 5 % for at least 10 hours.
2) Leave the sensor in environment above 20 to 30 °C and humidity approximately 75 % for around 12 hours. (Humidity 75% can be achieved with saturated solution of NaCl.)

Sensor cable

Cable jacket	silicone rubber, blue
Wire insulation	FEP polymer
Length	standard 3 m (optionally up to 20 meters)
Measuring temperature range	-60 °C to +200 °C
Maximum allowable temperature	+220 °C
Cable diameter	4.3 mm (±0.1 mm)

The cable shows excellent resistance to moisture, chemicals and carbohydrates.

Other parameters

Inputs

Type	for contact or TTL levels
Number of inputs	2
Input contact on current	2 mA
Operating voltage	5 V
Maximum sampling frequency	1 kHz
Connector	slip-on terminal

Output

Type	Relay contact, single pole, double throw (SPDT)
Maximum AC voltage	50 V
Maximum DC voltage	85 V
Maximum current	2 A
Maximum power for resistive load	62.5 VA / 60 W
Varistor protection	U_{AC} = 60 V; E_{MAX} = 5 J; C = 0.64 nF
Connector	slip-on terminal

Ethernet interface

Connection	TBase 10/100 Ethernet
Connector	RJ45
GET encryption	128 bit AES; Rijndael; CFB method

WiFi interface

Type	IEEE 802.11 b/g and IEEE 802.11n (single stream), IEEE 802.11 d/h/i/j/k/w/r
Operating frequency	2.4 GHz
Antenna connector	SMA RP

Clock circuit and internal memory

Clock backup method (RTC)	capacitor (not replaceable by the user)

RTC backup time after power outage .....5 days
(if the device was previously connected to a power source for at least three hours without interruption)

Device electronics

PoE power supply.....according to IEEE 802.3af

Power supply from an external source .....11 to 58 V DC (with reverse polarity protection)

Current consumption from ext. source at 15 V ...typically 120 mA

Current consumption from ext. source at 24 V ...typically 72 mA

Current consumption from PoE .....typically 32 mA

Consumption .....typically 1.8 W

Power supply connector .....coaxial 3.8 × 1.3 mm; + inside

Operating temperature range .....-20 to +70 °C

Dimensions (without connectors).....88 × 70 × 25 mm

Housing material.....anodized aluminum

Degree of protection .....IP 30

Other parameters

Weight .....typically 130 g

Default settings of the Ethernet

IP address 192.168.1.254

Netmask .....255.255.255.0 (8 bits; mask C)

IP address of the gateway 0.0.0.0

Available designs

Mountable on 35 mm DIN rail .....optional accessory

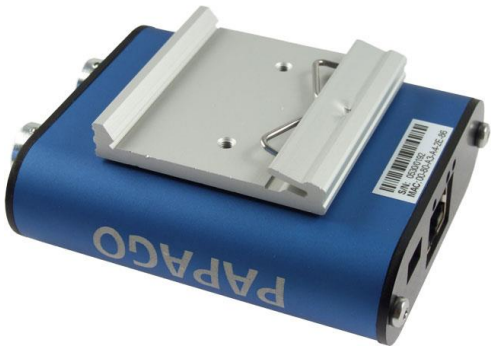


fig. 21 – Papago 2TH ETH with DIN rail holder

Do not hesitate to contact us if you have any other requirements concerning the design and functions of PAPAGO TH 2DI DO.

Papouch s.r.o.

Data transmission in industry, line and protocol conversions, RS232/485/422/USB/Ethernet/GPRS/ WiFi, measurement modules, intelligent temperature sensors, I/O modules, and custom-made electronic applications.

Address:

Strasnicka 3164
102 00 Prague 10
Czech Republic

Tel:

+420 267 314 267

Internet:

www.papouch.com

E-mail:

info@papouch.com