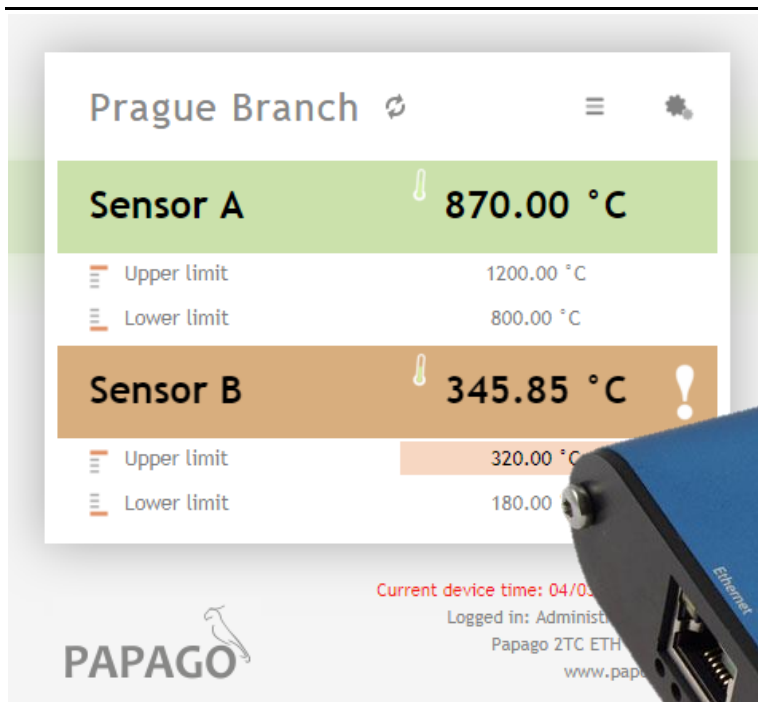# PAPAGO 2TC

Ethernet and WiFi sensors

Measure 2x K type thermocouple

PoE power or external power supply

P A P A G O   2TC

Datasheet

Created: 11/14/2014

Last update: 01/30/2023 01:39

Number of pages: 36

© 2023 Papouch s.r.o.

P a p o u c h   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

FIRMWARE VERSIONS

Version 1.5: SNMP extended by variable type.

Version 1.6

- Information e-mails are sent as a single message with all values at once instead of separate e-mails.
- HTTP GET can be turned on by itself for sending when limits are exceeded.
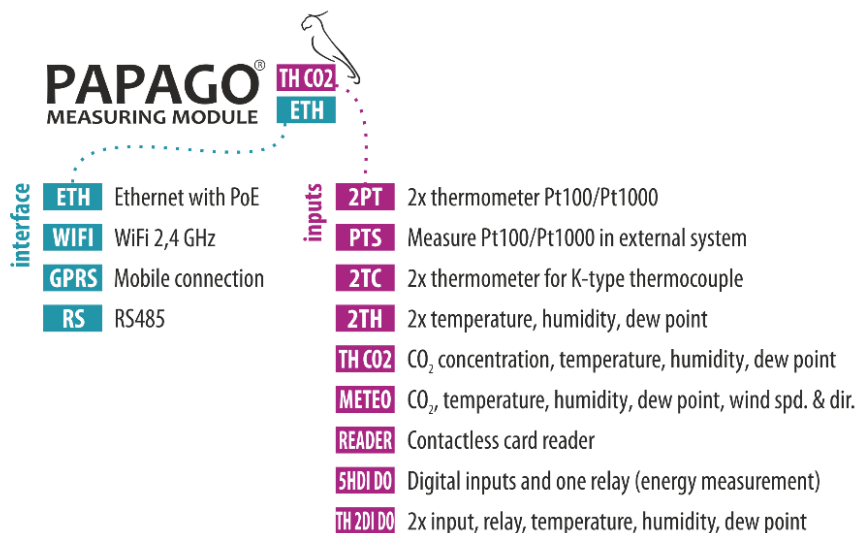
Version 1.7

- *E-mails:* Time in the subject was corrected, temperature state labels corrected as well.
- Fixed: re-directing after log-out from settings.
- Set limits are remembered even if the watching is off.

Version 2.1: Added communication protocol MQTT.

Version 2.3: Added option to disable Telnet, Firmware Update and SNMP.

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.

interface
- **ETH** Ethernet with PoE
- **WIFI** WiFi 2,4 GHz
- **GPRS** Mobile connection
- **RS** RS485

inputs
- **2PT** 2x thermometer Pt100/Pt1000
- **PTS** Measure Pt100/Pt1000 in external system
- **2TC** 2x thermometer for K-type thermocouple
- **2TH** 2x temperature, humidity, dew point
- **TH CO2** CO_2 concentration, temperature, humidity, dew point
- **METEO** CO_2, temperature, humidity, dew point, wind spd. & dir.
- **READER** Contactless card reader
- **5HDI DO** Digital inputs and one relay (energy measurement)
- **TH 2DI DO** 2x input, relay, temperature, humidity, dew point

Applications

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

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 analyze data in the Wix program.

Communication Options

PAPAGO features different communication options depending on the used interface. PAPAGO can be **controlled by user** via a web interface or via software for Windows. **Machine 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:

		MODBUS	HTTP GET	MQTT	EMAIL	SNMP	XML	SPINEL	WEB	WIX
ETH	TCP	✓	✓	✓	✓	✓	✓	✓	✓	✓
WIFI	TCP	✓	✓	✓	✓	✓	✓	✓	✓	✓
GSM		✓		✓						

(← automatic control | user control →)

Machine data-reading: Modbus TCP, MQTT[1], HTTP GET with encryption, e-mail, SNMP, XML, Spinel

User control: Web interface, Wix software

Properties

Papago 2TC on or two external type K thermocouples.

Measuring range .... **-50 to +1350 °C**; resolution 0,1 °C

Measurements per second ... 1, 2 or 5

Connector .... thermocouple connector IEC mini

- Family of measuring devices with Ethernet or WiFi interface.

- Data reading by user via responsive web interface or Wix software.

- Machine data reading via Modbus, HTTP GET, MQTT[1], SNMP, XML, e-mail or Spinel protocol.

- The ability to encrypt data in HTTP GET by 128bit AES encryption.

- Measurements via external Pt100 / Pt1000 thermometer. (Sensors are not included.)

- WiFi 2,4 GHz.

- Power supply from PoE (IEEE 802.3af standard; Ethernet versions only) or external source.

- External DC power supply 11 to 58 V.

- Current consumption typically 72 mA at 24 V.

---

[1] MQTT protocol is available only in version with communication via Ethernet.

CONNECTION

1) <u>Ethernet version:</u> Connect the device by a normal uncrossed cable for computer networks to the switch.

2) <u>Ethernet version:</u> If the device cannot be powered by the switch via PoE according to the IEEE 802.3af standard, connect an 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.)

   <u>WiFi version:</u> Connect an 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.)

3) Connect the K type thermocouple to one or both connectors *sensor a* and *sensor b.*

4) <u>Ethernet version:</u> 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](#).
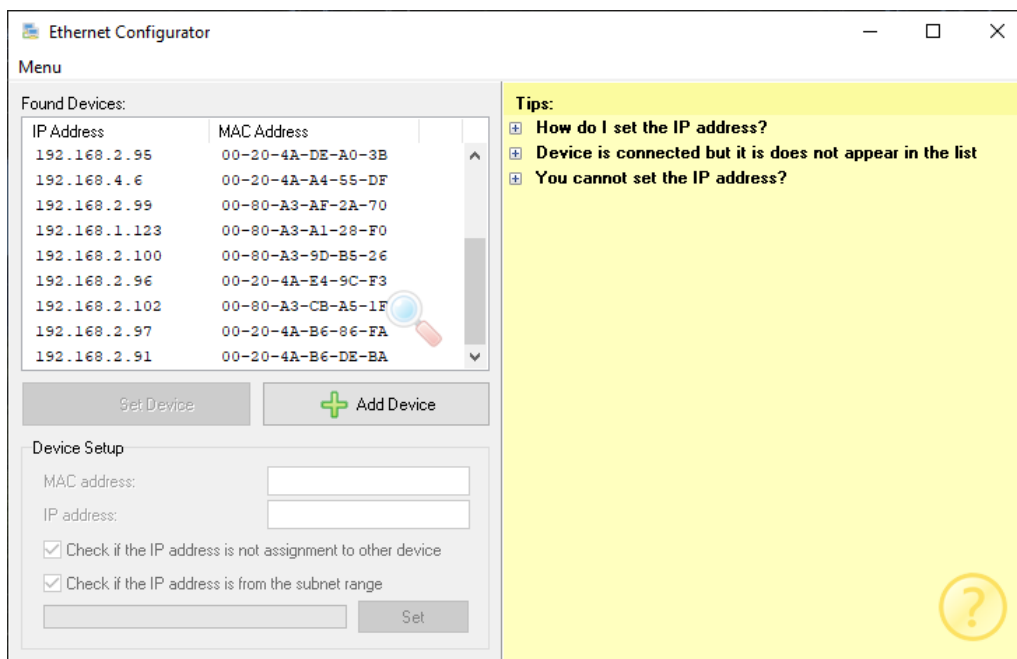

fig. 1 – Ethernet Configurator for setting the IP address

<u>WiFi version:</u> Connect your Papago to a windows PC using the supplied micro USB cable.[2] 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.

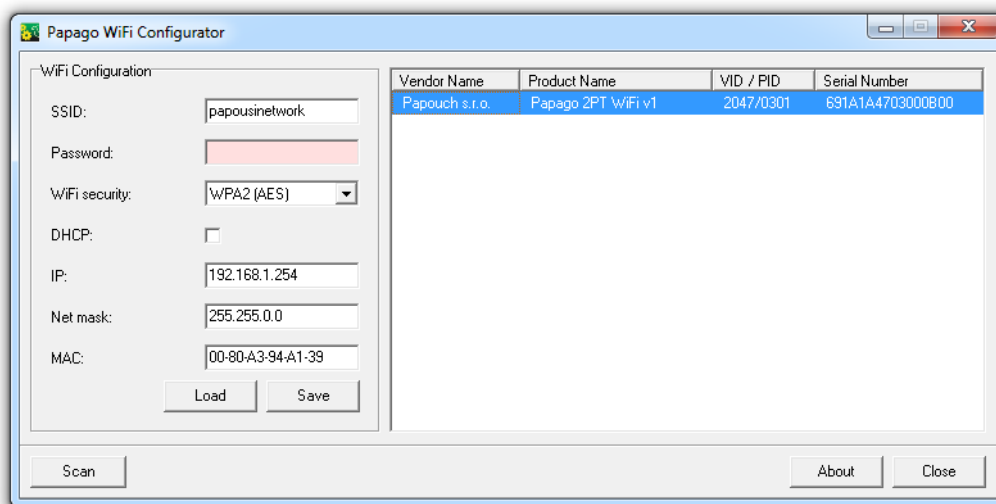[2] In Windows 7 or higher driver will be installed automatically.

fig. 2 – 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.)

CONFIGURATION

Configuration is done via a web interface. The basic network parameters can also be set via Telnet (see page 18). **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.
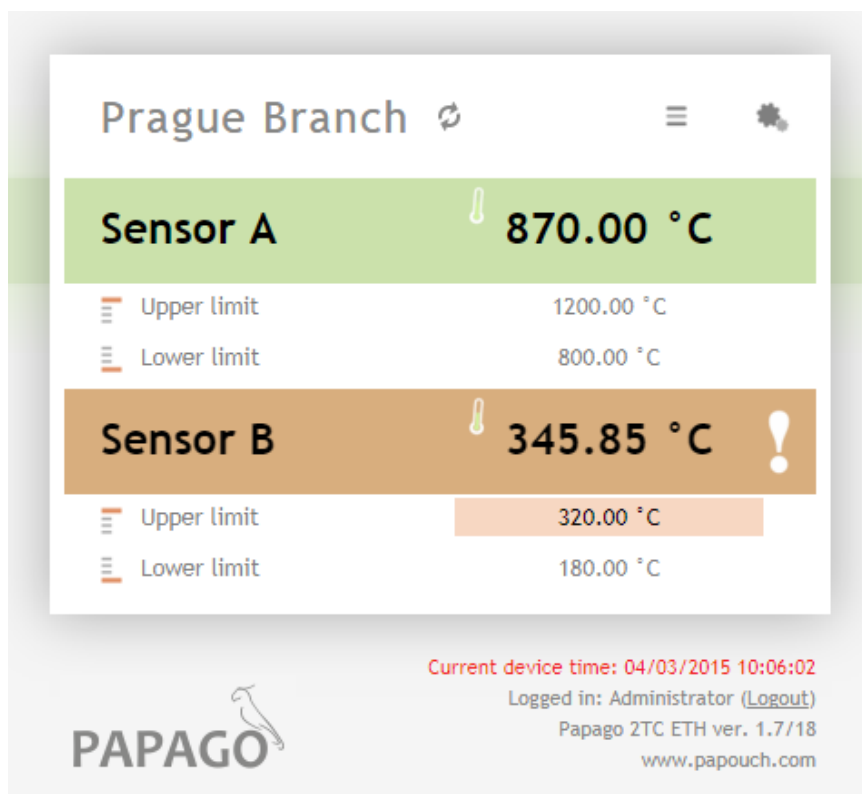

fig. 3 – Readouts from temperature sensors connected to Papago (Sensors can be named)

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 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. 4 – Papago configuration with displayed help for data port setting
(example screen from Papago METEO WiFi)

N e t w o r k

This section contains the configuration of network parameters.

fig. 5 - 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. 6 - 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)*.

Password length is 8 to 30 characters.[3]

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. 7 - 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.

The user name is always 'user', the administrator name is always 'admin'. Passwords have a maximum length of 8 characters.[3]

If user has a password, administrator must also have a password. For security reasons, password is not displayed after saving.

Last two items - *Disable Telnet* and *Disable fw upgrade* - are only available in Ethernet version.

Caution: If you disable Telnet protocol and/or upgrade fw and an error occurs during a firmware upgrade, manufacturer service may be required!

---

[3] Password can contain the following characters:
!#$%()*+,-./0123456789:;=?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[]^_abcdefghijklmnopqrstuvwxyz{|}~

E-mail

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

fig. 8 - settings for sending e-mails

Papago 2TC ETH cannot communicate with servers that require a secure (SSL/TLS) connection.

User password length is max. 10 characters.[3]

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

Example - when limits are exceeded:
    Temperature Senzor A is in range. Value is 27.7 °C.
    Temperature Senzor B exceeded upper limit 27.0 °C. Value is 27.5 °C.

Example – upon returning in the range:
    Temperature Sensor A is in range. Value is 24.3 °C.
    Temperature Sensor B is in range. Value is 24.3 °C.

SNMP

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

fig. 9 - settings for communication via SNMP

For description of SNMP objects see page 21.

HTTP GET / MQTT

Here is settings for sending measured data to remote server – options are *None, MQTT*[1] *and HTTP GET*.

HTTP GET

As *Mode* select HTTP GET.

fig. 10 - 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 (upper- and lower-case letters with no diacritical marks and numbers), the HTTP GET data is encrypted by 128-bit AES cipher (Rijndael), the CFB method.

GET Format

- *Example of periodic GET[4]:*

  ```
  script.php?mac=0080A393A273 type=Papago 2TC ETH guid=PAPAGO-TEST-
  GUID description=LOG log_index=4268 date_time=11/21/2014 14:21:00
     T1V1_value=24.4 T1V1_units=°C T1V1_status=0 T2V1_value=322.1
                    T2V1_units=°C T2V1_status=2
  ```

- *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[5] after pressing the button in the settings:*

  ```
  script.php?encrypted_data=%2C%60%32%08%25%03%44%2E%40%29%63%61%34%0
  8%44%62%67%CF%70%FE%D0%EA%E9%9C%C3%4C%9B%9D%E3%8B%31%18%10%E4%FB%9E
  %59%25%56%A4%60%68%1B%77%CC%EE%23%99%D1%CE%1A%AE%B5%E4%BC%D3%0C%84%
  9E%7C%F4%2B%5F%B1%D4%99%C6%11%F8%75%C7%E5%27%10%93%DC%8D%43%EF%13%7
                    9%37%F1%D2%5B%35%6B
  ```

  Encrypted part above contains following data: `mac=0080A394A139&type=Papago 2TH WIFI&guid=Papago-GUID&description=TEST`

- *Description of get parameters:*

  - *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.

  - *index* ....Sequence number of the get request.

  - *log_index* ....Sequence number of the record in a circular buffer.[6]

  - *date_time* ....Timestamp.

  - *encrypted_data* ....This parameter contains the data of encrypted GET.[5]

  The following parameters may be given more than once if there are more measured values from one sensor. The first character may is T (for temperature). Thus, Papago 2TC with two temperature sensors, provides these parameters:

---

[4] & charracters are deleted for better readability

[5] It is 16 bytes of initialization vector followed by encrypted data as specified in the body of standard get. We have examples of get processing for Node.js and PHP in this article on the web (Czech section of website with comments in English): papouch.com/desifrovani-aes-v-http-getu-z-papaga-p3719/

[6] 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.

- *T1V1[7]_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.

- *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.

MQTT

As *Mode*, select MQTT[1]. Papago works as MQTT Publisher. It sends variables from connected sensors to configured broker.

fig. 11 - MQTT settings[1]

If **sending period** is set to zero, sending is disabled. Period can be set from 0 to 1440 minutes.

If sensor is set as *Unused*, its parameters are not sent.

Topic enter into field *Path / Topic* (publish topic).

QoS can be selected as 0, 1 or 2.

Security: SSL/TLS security is not supported.

Password: Maximum 15 characters.[3]

---

[7] 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.

Payload format:

Message example from Papago 2TH with one temperature and one temperature/humidity sensor in JSON format:

```json
{
    "dev": "Papago 2TH ETH",
    "mac": "0080A3DC7EF4",
    "loc": "U Papoucha",
    "description": "LOG",
    "log_index": 5,
    "time": "06/29/2020 12:38:00",
    "vals": [{
        "t": "temp",
        "v": 28.3,
        "u": 0,
        "io": 1,
        "e": 0
    }, {
        "t": "temp",
        "v": 27.9,
        "u": 0,
        "io": 2,
        "e": 0
    }, {
        "t": "hum",
        "v": 49,
        "u": 0,
        "io": 2,
        "e": 0
    }, {
        "t": "dew",
        "v": 16.2,
        "u": 0,
        "io": 2,
        "e": 0
    }]
}
```

Parameters description:

- **dev:** Device type

- **mac:** MAC address

- **loc:** Location

- **description:** Event type („LOG", „WATCH", „TEST")

  o LOG: Periodically sent message.

  o WATCH: Right now, set limits have been crossed.

  o TEST: Message sent by button in device configuration.

- **log_index:** Sequence number of periodically sent message. This way you can check continuity of sent messages.

- **time:** Message sending time according to Papago's internal clock.

- **vals:** Array with values from connected sensors. Each element of array contains an object with these values:

  - **t**: value type

    - **temp**: temperature

  - **v**: current value

  - **u**: code of physical unit

    - **0** → degrees Celsius

    - **1** → degrees Fahrenheit

    - **2** → Kelvin

  - **io**: connector name from which this variable is read (sensor a = 1, sensor b = 2)

  - **e**: status / error code

    - **0**: everything works

    - **2**: upper limit of measuring range exceeded (overflow)

    - **3**: measured value is less than lower range limit (underflow)

    - **4**: sensor error

Sensor Section

Sensors A and B have their own individual sections with identical settings.

fig. 12 - configuration of one of the sensors

O t h e r   S e t t i n g s

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

Other settings

Name of the device	Chairlift
Language	English
Temperature units	Fahrenheit [°F]

Date and time

Synchronize device's time with NTP server	☑
NTP server's IP address	195.113.144.201
Time zone	(UTC+01:00) Amsterdam, Berlin, Bern, Rome, Stockholm, Vie ⌄
Auto daylight saving	☑
Synchronize device's time with this PC's time	☐

fig. 13 - 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 6).

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. [8])

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.

---

[8] In Windows 10 or higher, Telnet client is not a standard part of system. Install it using 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. [8]

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-address]/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="870.00" w-min="800.00" w-max="1200.00" />
    <sns id="2" name="Sensor B" type="1" status="2" unit="0" val="345.85" w-min="180.00" w-max="320.00" />
    <status msp="1" level="2" signal="" location="My Sensors" time="04/03/2015 10:06:02"/>
</root>
```

fig. 14 – example of XML with actual values

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

status

location

User-defined name of the device.

time

Current timestamp – see example.

sns

id

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

name

The name of the sensor.

type

Number 1 (parameters of temperature)

status

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

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

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-max

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.

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.
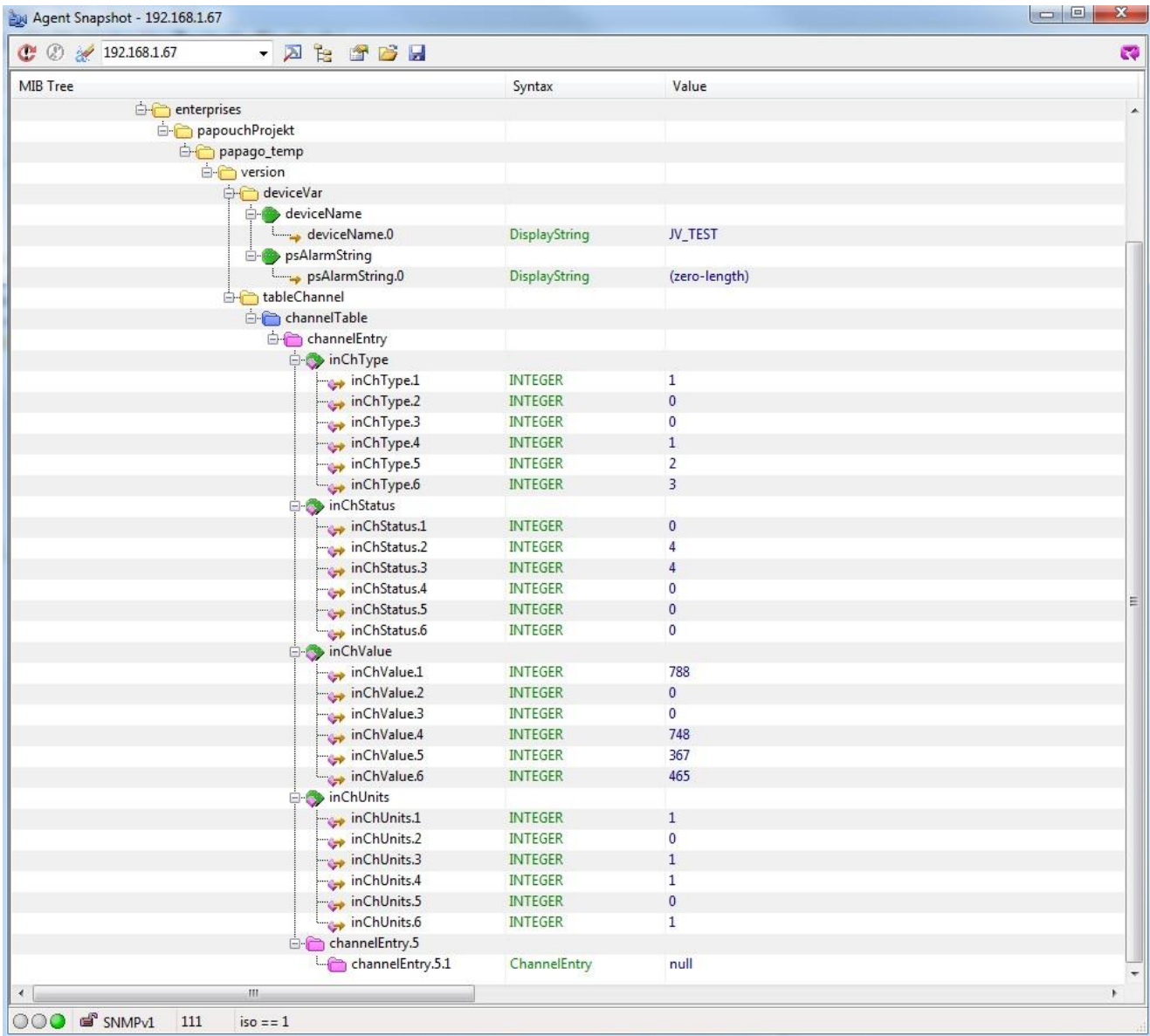
fig. 15 - example from Papago 2TC ETH

If you need to get all the OID objects with the tool SNMPWALK (Linux), then it is necessary to specify the start OID position, for example:

snmpwalk -v1 -c public 192.168.1.254 1.3.6.1.4.1.18248

In case of reading only with the IP address, you will get only basic system OID objects from the ethernet module.

Objects – variables

Type

Name: inChType

Object ID: 1.3.6.1.4.1.18248.31.1.2.1.1.1.1 až 6 [9]

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

      0 → Not used.
      1 → Temperature.

Status

Name: inChStatus

Object ID: 1.3.6.1.4.1.18248.31.1.2.1.1.2.1 to 6 [9]

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: 1.3.6.1.4.1.18248.31.1.2.1.1.3.1 to 6 [9]

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

Unit

Name: inChUnits

Object ID: 1.3.6.1.4.1.18248.31.1.2.1.1.4.1 to 6 [9]

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

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

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.

---

[9] The ID of the objects shows the values from sensors A and B arranged one after another. First A, then B..

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.

T r a p s

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

Input Register

Input Register contains the currently measured data from both sensors in several formats.

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 – 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
			   1 = °F
			   2 = K

Sensor 2			
from 100			

SPINEL

The device contains the standard Spinel protocol (format 97) for communication via the TCP data channel. Application development with this protocol is easy due to Spinel Terminal, Spinel.NET SDK on Github and Spinel online parser.

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;.v10 10.01.01;.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 81 00 20 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 20 B0 43 0C 95 43 A1 0E 49 20 20 20 20 20 33 32 32 2E 31 63 0D	*a.W1..X11/25/2014.14:07:3 2.............°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. 16 - 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

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.		

unita	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[10] 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,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`

---

[10] 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)

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 2TC ETH; v1010.01.01; 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,25H,31H,02H,00H,50H,61H,70H,61H,67H,6FH,20H,32H,50H,54H,20H,45H ,54H,48H,3BH,20H,76H,31H,30H,31H,30H,2EH,30H,31H,2EH,30H,31H,3BH,20H,66H,39 H,37H,EBH,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.	

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	

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[11] 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,20` `H,20H,20H,B0H,43H,00H,BDH,41H,97H,79H,6BH,20H,20H,20H,20H,20H,20H,31H,38H,2` `EH,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`

---

[11] 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
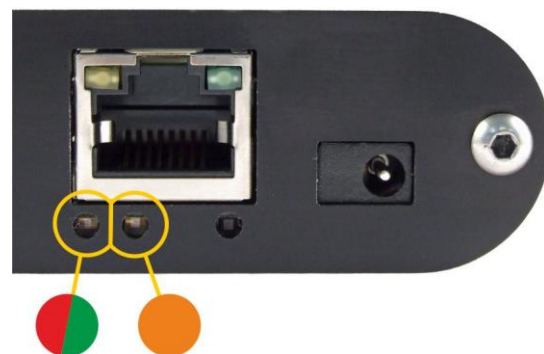
Papago with WiFi connection:

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.

- Greend 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 20), 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.

reset
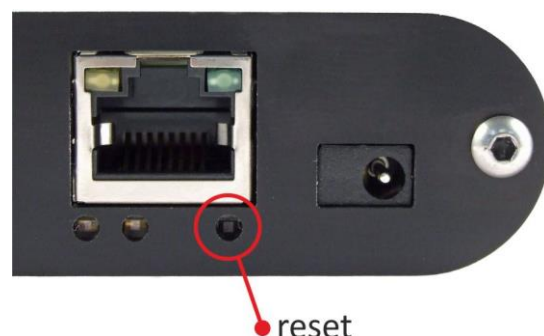
TECHNICAL PARAMETERS

Sensors

Sensor types .... 2 × type K thermocouple (sensor not included)

Measuring range .... -50 to +1350 °C; resolution 0,1 °C

Accuracy .... ±0.5 °C (excluding thermocouple accuracy)

Measurements per second............................. 1, 2 or 5

Connector .... thermocouple connector IEC mini

Ethernet interface

Connection.. TBase 10/100 Ethernet

Connector RJ45

GET encryption .... 128 bit AES; Rijndael; CFB method

Protokol SNMP .... v. 1

Protokol MQTT.. v. 3.1.1

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 aluminium

Degree of protection..................................... IP 30

Weight... typically 130 g

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

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. 17 – Papago 2TC ETH with DIN rail holder

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

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