Inveo Sp. z o.o.

µSCADA

Instruction Manual

Overview



Purpose of the device

The µSCADA module is designed for gathering data via Modbus RTU/TCP and visualizing it on a page accessible via web browser. The webpage utilizes the AJAX mechanism. The built-in FTP server allows the user to update the page contents. Built-in 4GB memory enables creation of extensive visualizations.

For additional security, the module is equipped with two-layer user authentication.

Utilizing the capabilities of webpage technology, it is possible to create impressive graphic visualizations.

Changelog

Table of contents

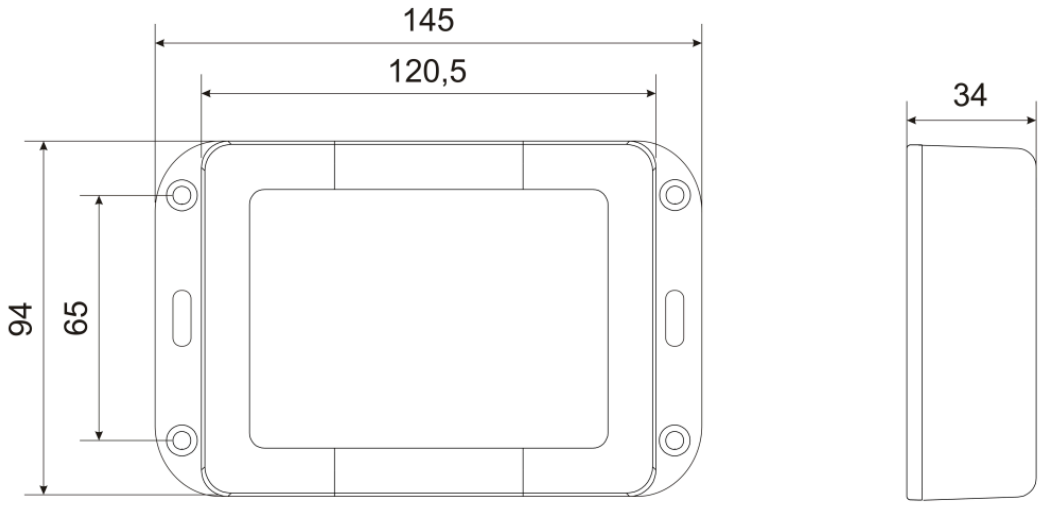
- Overview
  - Purpose of the device
  - Changelog
- Device construction
  - Technical characteristics
- Network configuration
  - LAN and RS-485 configuration
    - FTP server connection
- Creating a visualization
  - Modbus points configuration
  - Catalogue structure
  - Downloading data for visualization
  - Changing the values of Modbus registers
- Examples
  - Connecting a single device
- Warranty and manufacturer's liability
  - Storage, operation and transport conditions
  - Disposal and decommissioning

Device construction

Technical characteristics

Parameter	Description
Power supply	Passive PoE 12V DC
Power consumption	max. 1W
Memory size	4GB
Read throughput	3,2Mbit/s
Write throughput	1,1Mbit/s
Number of Modbus points	16384
Input	**Input type:** Three-way screw terminal
Communication	Ethernet 10Mbps, RJ45

Dimensions



General characteristics

Module communication is carried out via the RS485 port, utilizing the Modbus RTU protocol.

An LED indicating correct device operation (LED blinking) or error condition (LED glowing constantly) is located on the front of the housing.

Network configuration

Computer subnet configuration

After the initial power-up, it is needed to configure the device.

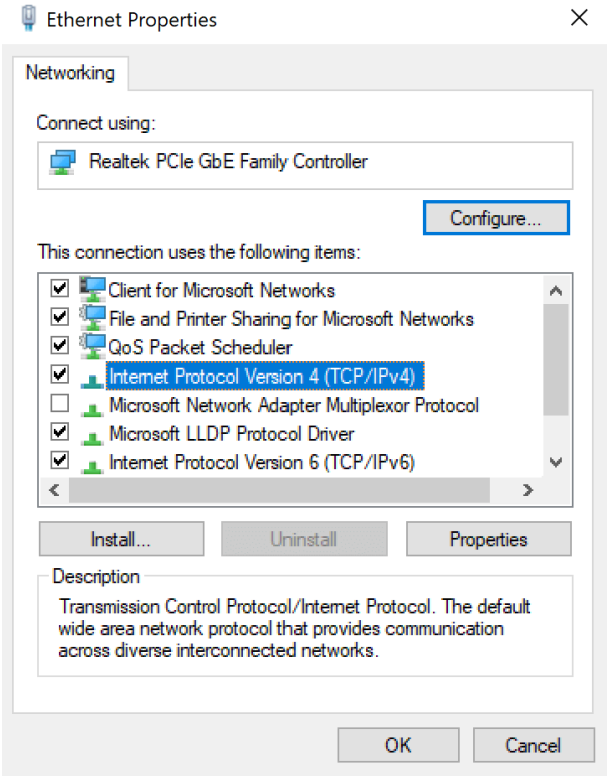
Changing the computer's subnet address for configuration

After connecting the device to the network, it is needed to change the subnet of a computer connected to the same network.

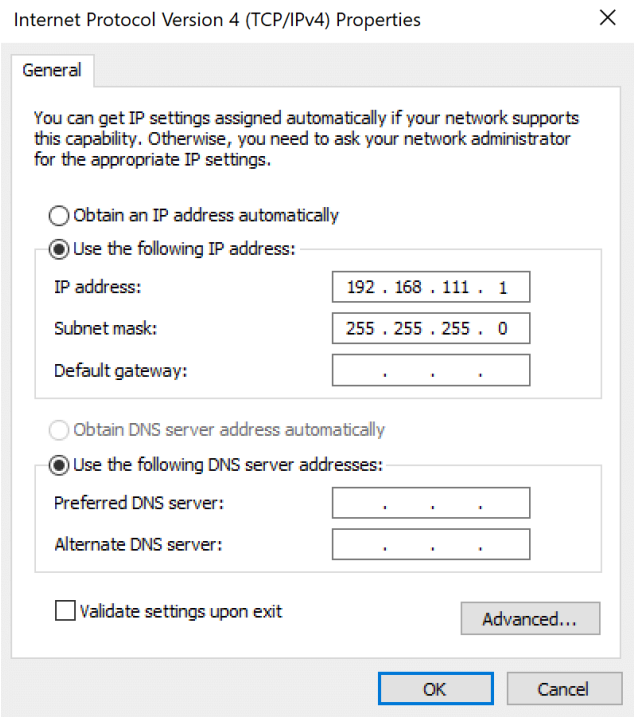
To do that, go to the network configuration: **Start → Control Panel → Network and Internet → Network and sharing center → Change adapter settings.**

Then, select the network connection, right-click it and select **Properties**.

After selecting this option, network connection configuration screen will appear:



Select **Internet protocol version 4 (TCP/IPv4)** and type in the following parameters:



Note

After applying the configuration with OK, start a web browser and type in the following address: 192.168.111.10.

LAN and RS-485 configuration

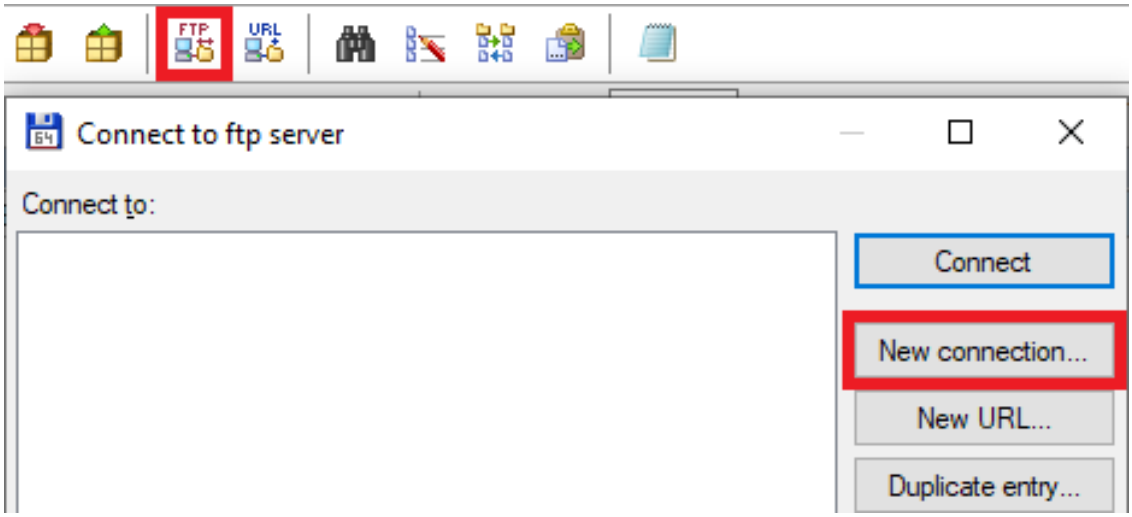
The `config.xml` file can be used to change the device's parameters. Configuration of the devices µScada takes data from, is carried out by editing `points.xml`.

The settings can only be changed via FTP.

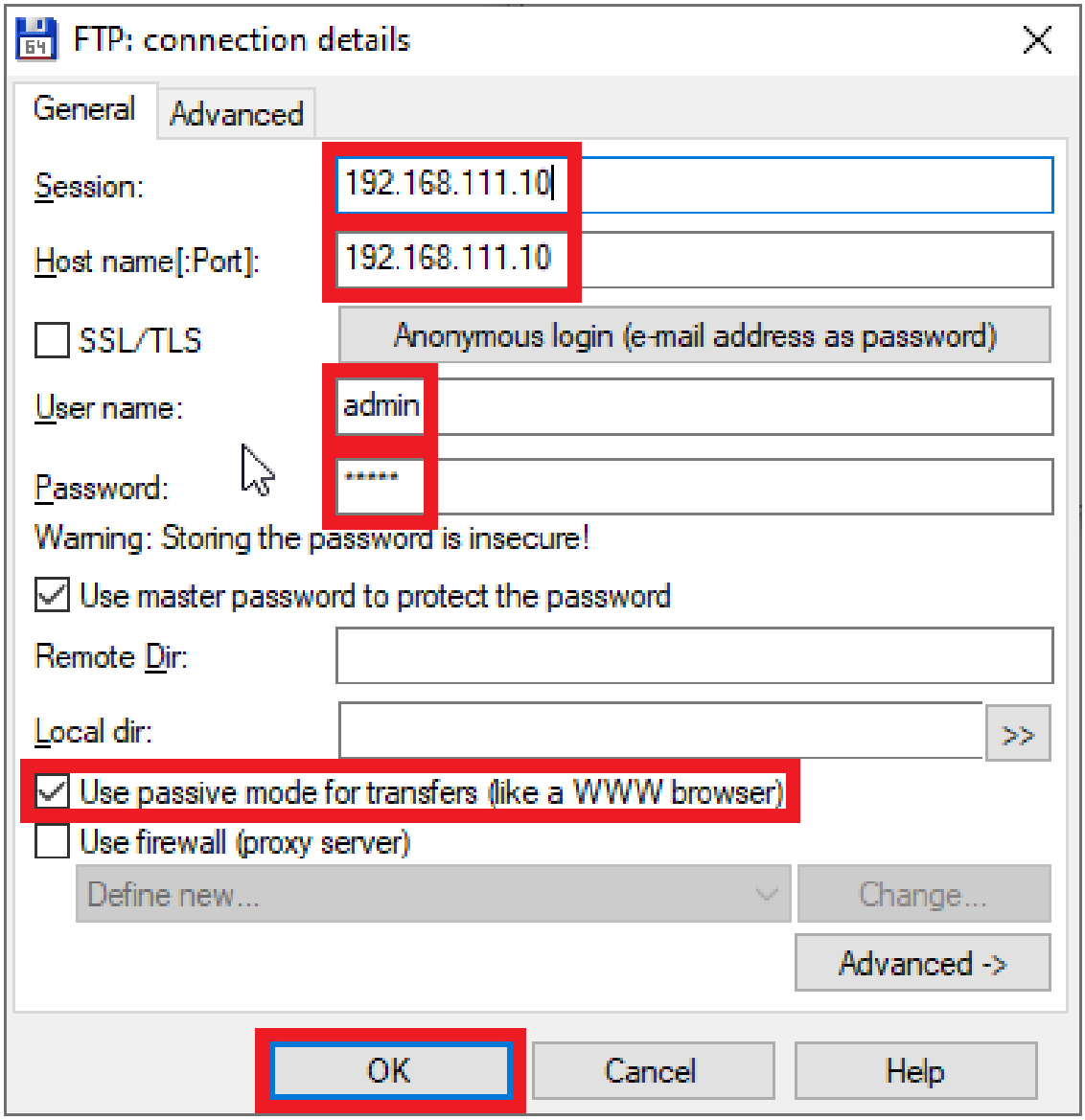
FTP server connection

To establish connection with the FTP server, use any FTP client such as FileZilla, WinSCP, Total Commander etc. Connection parameters:

Example of an FTP server connection with Total Commander:



Firstly, locate and click the **FTP** icon on the upper toolbar. After a **Connect to FTP server** appears, create a **New connection**.



Initial connection configuration

In the **Session** and **Host name** fields, type in the device's IP address (default 192.168.111.10).

Default **User name** and **Password** field contents: user – admin, password – admin.

The server supports both passive and active connection mode.

Click **OK** to save the data and create a new connection.

After changing the configuration parameters in the `config.xml` file, corresponding connection configuration fields should be modified accordingly.

Example `config.xml` file contents:

```
<CONFIG>
    <NET_IP>192.168.111.10</NET_IP>
    <NET_MASK>255.255.255.0</NET_MASK>
    <NET_GW>192.168.111.101</NET_GW>
    <FTP_USER>admin</FTP_USER>
    <FTP_PASS>admin</FTP_PASS>
    <WWW_USER></WWW_USER>
    <WWW_PASS></WWW_PASS>
    <WWW_PROTECT_USER>admin</WWW_PROTECT_USER>
    <WWW_PROTECT_PASS>admin</WWW_PROTECT_PASS>
    <RS485_BAUDRATE>9600</RS485_BAUDRATE>
    <RS485_PARITY>NONE</RS485_PARITY>
    <MB_TIMEOUT>1000</MB_TIMEOUT>
</CONFIG>
```

Warning

If any change was made to points.xml and config.xml files, restart the device to apply the changes.

Explanation of XML tags

µSCADA access:

- `<NET_IP>` – µSCADA's IP address,
- `<NET_MASK>` – Subnet mask,
- `<NET_GW>` – Gateway (used for active FTP mode),
- `<FTP_USER>` – FTP user name,
- `<FTP_PASS>` – FTP user password.

Visualization page access:

- `<WWW_USER>` – Main visualization catalogue user name,
- `<WWW_PASS>` – Main visualization catalogue user password.

Leaving these fields empty will allow visualization access without a password.

- `<WWW_PROTECT_USER>` – Protected visualization catalogue user name,
- `<WWW_PROTECT_PASS>` – Protected visualization catalogue user password.

Modbus RTU configuration:

- `<RS485_BAUDRATE>` – RS485 transmission speed (4800 – 115200),
- `<RS485_PARITY>` – Parity – available options: NONE, EVEN, ODD,
- `<MB_TIMEOUT>` – Maximum Modbus response wait time.

Creating a visualization

The module is equipped with built-in memory for storing visualization contents in the form of a www webpage. The website can be built using HTML + CSS languages. Visualization data is available via AJAX mechanism.

The `points.xml` file can be generated using the uSG application.

Modbus points configuration

Definition of Modbus registers read points is contained within the `points.xml` file.

Example of a single data read group via Modbus RTU:

```
<group>
    <id>node_</id>
    <id_start>5</id_start>
    <len>3</len>
    <protocol>RTU</protocol>
    <dev_addr>1</dev_addr>
    <cmd>MB_MULTIHOLD</cmd>
    <address>4000</address>
    <access>rw</access>
    <poll>50</poll>
</group>
```

XML tags definitions:

- `<group>` – Single group definition,
- `<id>` – Name of the group of points (name member for a single Modbus register of this group) – see section **Downloading data for visualization**,
- `<id_start>` – Start number added to the name defined in `<id>`,
- `<len>` – Amount of points in a group (amount of downloaded Modbus registers),
- `<protocol>` – Specifies the protocol:
  - **RTU** – Modbus RTU via RS485,
  - **TCP** – Modbus TCP via LAN.
- `<dev_addr>` – Specifies the number of Modbus RTU device (1 for TCP),
- `<cmd>` – Sets the Modbus command:
  - `0x01` / `0x05` – Read/Write Single Coil – MB_COIL,
  - `0x02` - Read Discrete Input – MB_INPUTSTAT,
  - `0x03` / `0x10` – Read/Write Holding Registers – MB_MULTIHOLD,
  - `0x04` – Read Input Registers – MB_INPUT.
- `<address>` – Modbus resource address,
- `<access>` – Access mode (r – read, w – write, rw – read and write),
- `<poll>` – Poll time value * 0.1s (value of 50 means 5s).

Additional parameters are required for Modbus TCP:

- `<ip_addr>` – Adress IP of the Modbus device,
- `<ip_port>` – Modbus device port (default 502).

> **Warning**
>
> **If any change was made to points.xml and config.xml files, restart the device to apply the changes.**

Catalogue structure

- `\config.xml` – Network configuration file,
- `\points.xml` – Modbus device points file,
- `\www` – Visualization page catalogue,
- `\www\protect` – Protected page catalogue,
- `\www\comm.xml` – XML file used for data exchange.

Downloading data for visualization

Data from all points is located in the `comm.xml` file.

The `comm.xml` file has the following format:

`<idX>value</idX>`, where:

- `id` is the name defined in the `id` tag (`points.xml` file),
- `X` is the number from `id_start` or a subsequent one (depending on the `len` tag value).

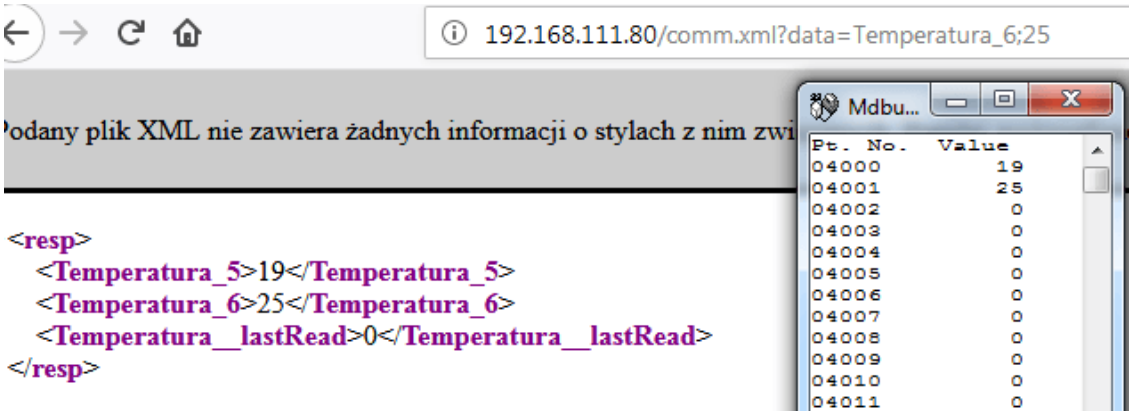
Changing the values of Modbus registers

HTTP GET with the following should be used to send data:

`192.168.111.10/comm.xml?data=idX;value`, where:

- `192.168.111.10` – µScada's IP address,
- `id` – Point name defined in the `id` tag (`points.xml`),
- `X` – is the number from `id_start` or a subsequent one,
- `value` – Value to be sent.

Example `comm.xml` contents:



Addresses and values assigned to them are presented on the right side of the image.

Examples

Any ajax.js or AJAX-compatible (eg. Jquery) library can be used to send and receive data by a www browser.

Connecting a single device

In the following example, a single device assigned to Modbus RTU with a physical address of 1 and register addresses 4000 and 4001 will be used.

Firstly, modify the `points.xml` file:

```
<group>                     //group definition
    <id>Temperature_</id>   //name that will be read
    <id_start>5</id_start>  //number, we start the numbering with
    <len>2</len>            //amount of points in the group
    <protocol>RTU</protocol>
    <dev_addr>1</dev_addr>  //physical address
    <cmd>MB_MULTIHOLD</cmd>
    <address>4000</address> //registers initial address
    <access>rw</access>
    <poll>50</poll>
</group>                    //end group definition
```

In `points.xml`, a single group with two points (len=2) has been defined.

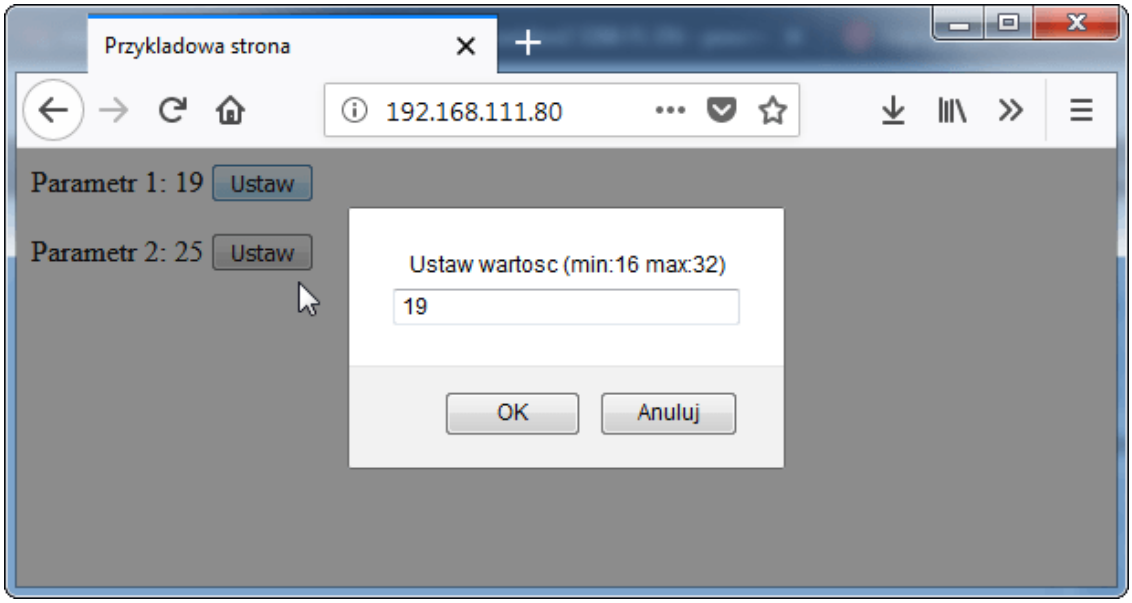
Next, modify the file from the www catalogue, with an example name of `index.htm`:

```
<html>
<head>
    <meta http-equiv="content-type" content="text/html; charset=ISO-8859-2">
        <title>Example page</title> //name of the page
        <script src="/ajax.js" type="text/javascript"></script>
</head>
    <body>
        <p>Parameter 1:
            <span id="data1">?</span>
            <button id="btn1" onclick="setData('Temperature_5',16,32)">Set</button>
        </p>
        <p>Parameter 2:
            <span id="data2">?</span>
            <button id="btn2" onclick="setData('Temperature_6',16,32)">Set</button>
        </p>
        <script type="text/javascript"><!--
        function updateStatus(xmlData)
{
    setTimeout("newAJAXCommand('comm.xml', updateStatus, false)",500);
    if(!xmlData) return;
    document.getElementById('data1').innerHTML = getXMLValue(xmlData, 'Temperature_5');
    document.getElementById('data2').innerHTML = getXMLValue(xmlData, 'Temperature_6');
}
    function setData(id, min, max)
{
    var p=prompt("Set value (min:"+min+" max:"+max+")","0");
    var val = parseInt(p);
    if((val>=min)&&(val<=max))
{
    sendData(id, val);
}
}
    function sendData(id, data)
{
    newAJAXCommand('comm.xml?data=' + id + ';' + data);
}
    setTimeout("newAJAXCommand('comm.xml',updateStatus,false)",2000);
    -->
        </script>
    </body>
</html>
```

Explanation

The code shown above allows for definition of parameters 1 and to from the range of 16-32 (the limit is defined by `"setData('Temperatura_5',16,32)"`)

After clicking the button, registers 4000 and 4001 will be written with the following values:



Warranty and manufacturer's liability

> **Warning**
>
> **The manufacturer provides a two-year warranty for the device and post-warranty service for a period of 10 years from the date of introduction of the device to the market. The warranty covers all material and production defects.**

The manufacturer undertakes to respect the warranty agreement, if the following conditions are met:

- All repairs, changes, expansions and device calibrations are carried out by the manufacturer or an authorized service center,
- The power supply system meets the applicable standards,
- The device is operated in accordance with the suggestions presented in this manual,
- The device is operated in accordance with its intended purpose.

The manufacturer assumes no responsibility for consequences resulting from improper installation, improper use of the device, failure to comply with the instruction manual, and repairs made by unauthorized personnel.

> **Warning**
>
> **The device contains no user serviceable parts inside.**

Storage, operation and transport conditions

The device should be stored in enclosed rooms, where the atmosphere is free from vapours and corrosive substances:

- Environment temperature from -30°C to +60°C (-22°F - 140°F),
- Humidity from 25% to 90% (condensation unacceptable),
- Atmospheric pressure from 700 to 1060 hPa.

The device is intended to operate in the following conditions:

- Environment temperature from -10°C do +55°C (14°F - 131°F),
- Humidity from 30% to 75%,
- Atmospheric pressure from 700 to 1060 hPa.

Recommended transport conditions:

- Environment temperature from -40°C do +85°C (-40°F - 185°F),
- Humidity from 5% to 95%,
- Atmospheric pressure from 700 to 1060 hPa.

Installation and device operation:

- The module should be operated in accordance with recommendations provided later in this manual.

Disposal and decommissioning

In an event the device needs to be decommissioned (eg. after its intended life period is surpassed), it is recommended to contact the manufacturer or his representative, who are responsible to respond appropriately, i.e., to collect the device from the user. The user can alternatively contact companies specializing in electronic device or computer equipment disposal and/or decommissioning. Under no condition should the device be placed with other waste.