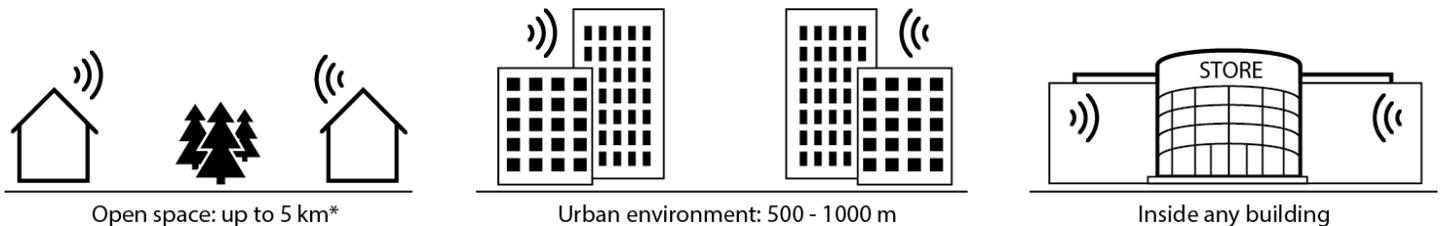




**7Bit Airpoint** is a wireless telemetry acquisition module that transmits data over a radio channel using LoRa technology. In general, the system consists of one or more wireless, autonomous **7Bit Airpoint** radio modules and a **7Bit Airgate** data gateway that receives messages over the radio channel and transmits them to some upper-level system, for example, WebHMI, another SCADA or cloud IoT platforms such as : Amazon AWS, Microsoft Azure, etc., over a local network or over the Internet, using the MQTT protocol.



The main feature of this concept is the long-term autonomous operation of telemetry modules (more than 1 year, or about 50,000 packets, from one set of batteries), with the ability to transmit data over a distance of up to several kilometers in open areas and up to 1000 meters in urban areas \*



## Application examples

- 1. Monitoring of climate and air quality parameters (temperature, humidity, CO<sub>2</sub> content) for office buildings, sports clubs, schools, hospitals, etc.** These types of facilities are characterized by large number of people. In many cases, retrofitting them with additional sensors in the premises allows obtaining objective information about the operation of heating, ventilation and air conditioning systems, correctly balancing their operation, and, as a result, reducing operating costs, while increasing the level of comfort. As a rule, such a task is not solved by standard means of industrial automation, due to the size of these objects, lack of necessary communications, unwillingness to spoil the decoration of premises, etc. In the case of radio modules, the deployment of such a system will take only a few hours, while the sensors can be located exactly there where it is needed, not where it works.
- 2. Collecting data at utilities.** Modern trends in the field of public untidily facilities industry have led to the emergence of tens of thousands of condominium boards and hundreds of management companies. The use of autonomous radio modules allows you to quickly and inexpensively automate the collection of data from metering devices, control engineering networks: the temperature of the coolant supply, leakage detection, operation of

pumps, phase loss, etc., control access to attics and basements. As well as having convenient remote access to this information, incl. through various cloud services.

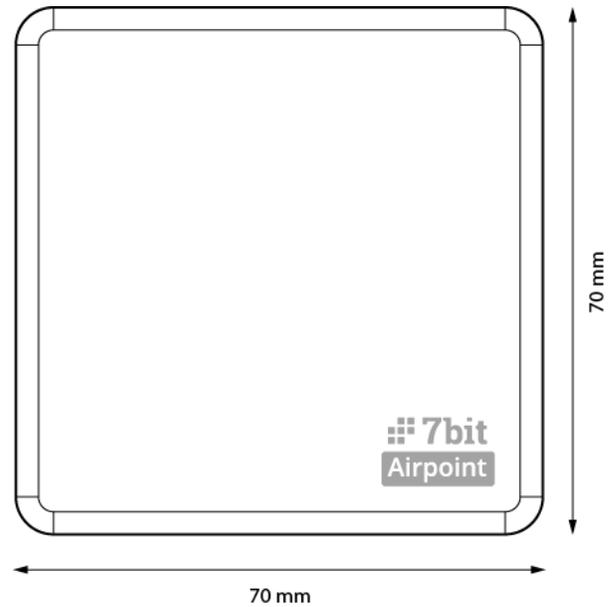
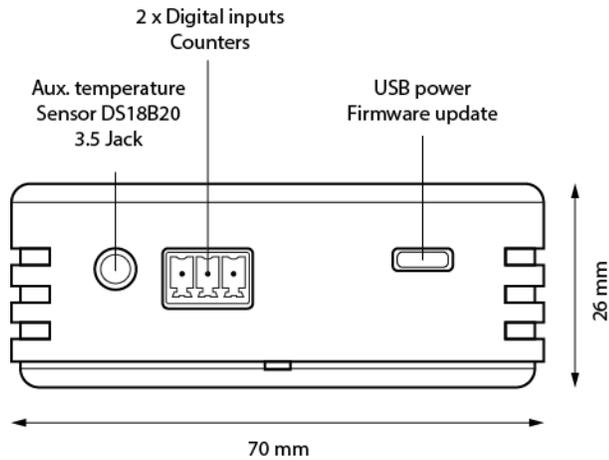
3. **Collecting telemetry data in the greenhouse industry.** Typical features for objects of this type are: long length and lack of communications. With its sensor set and communication capabilities in this environment, Airpoint modules are ideal for greenhouse climate monitoring applications. Standard set of signals: air temperature control at 2 points, humidity, soil temperature, coolant temperature, pressure in the heating system, opening / closing doors. In combination with the capabilities of the WebHMI integration controller, which can also control all engineering systems: pumps and valves for water supply, boiler room, emergency generators, etc., you end up with a comprehensive solution for object automation.
4. **Connecting signals from remote systems and devices.** There are tasks where remote parameters are interconnected with a certain technological unit, which we would like to see together with the data of the unit itself, or provide control of the unit depending on the value of the remote parameter:
  - Water tower, or basins - sedimentation tanks, the pressure / level from which must be transferred to the pumping station
  - Water consumption in the water utility mains with the transmission of a flow signal to remote water treatment stations for the correct dosing of reagents
  - End position sensors for gates, gate valves, etc.

## 1. General specifications - 7Bit Airpoint

Power supply	3 x AAA type elements and/or Micro USB (+5V)
Communication	Radio 868 MHz, 25 mw, LoRa, internal antenna
Temperature	Internal sensor, -40 +125 °C, accuracy +/-0.3°C External sensor DS18B20, Audio jack, - 40 +125 °C, accuracy +/-0.5°C(accessory)
Humidity	Internal sensor, 0 – 100 %, accuracy 3% for 20 – 80% range, 5% for the rest
CO <sub>2</sub>	400 – 8000 ppm (optional)
Digital input	2 x dry contacts, counting inputs up to 10kHz
Extras	Internal accelerometer, thief and unauthorized access protection
Mean time to battery replacement	Up to 60 000 telegrams <sup>1</sup>
Telegram sending rate	Default: once per 5 minutes (can be changed when ordering) With external power:: once per minute In the Alarm mode: upon digital input / accelerometer change
Reach	In the open air – up to до 5 км <sup>2</sup>
Environmental	Indoor, protection class IP 40, protective coating -40 + 85 °C with external supply, -25 + 55 °C with batteries
Dimensions	70 x 70 x 26 мм.

<sup>1</sup> Operating time depends on the operating mode and, accordingly, the frequency of sending packets

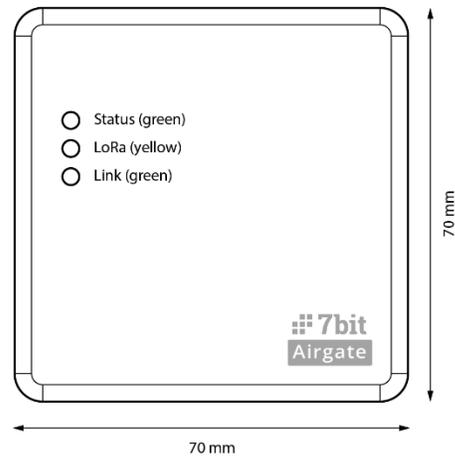
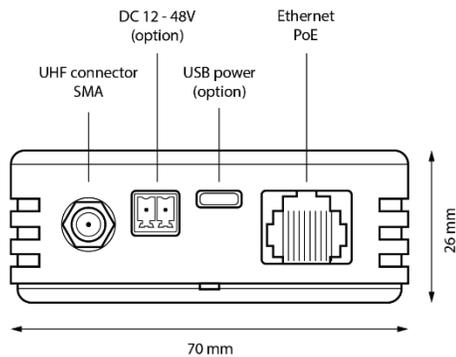
<sup>2</sup> The transmission range is highly dependent on many factors: the location of the receiver and transmitter, line-of-sight, obstacles, interference, etc..



## 2. General specifications - 7Bit Airgate

The gateway is designed to receive messages from telemetry modules over a radio channel and send them to other systems using the MQTT protocol, via a local network or the Internet, in a transparent bridge mode, without intermediate storage, visualization and analysis.

Power supply type	PoE, Micro USB (+5V), 12 – 48 VDC
Communication capabilities	Ethernet, Wi-Fi b/g/n, embedded antenna
Protocol	MQTT
Configuration	Web-interface
Number of sensors	Up to 255
Environmental	Indoor, protection class: IP 40, protective coating, external antenna 2dBi, SMA, 868 MHz
Dimensions	70 x 70 x 26 mm. Antenna - 108 mm



- Status (green)
- LoRa (yellow)
- Link (green)

### 3. MQTT protocol features

This protocol is widely used in the world of IoT (Internet of Things). Its main difference from most industrial protocols is that the initiator of the exchange is a slave device (sensor), and not a data acquisition system. Information transfer is implemented according to the Publisher / Subscriber scheme and occurs through an intermediate server (Broker). The data source (Publisher) sends data to the server (Broker), other systems and devices, being Subscribers, read them from there. This method of interaction is well suited for event-based messaging, when devices periodically communicate themselves. Thus, MQTT is not inherently a "real-time" protocol and is primarily used for data collection rather than control.

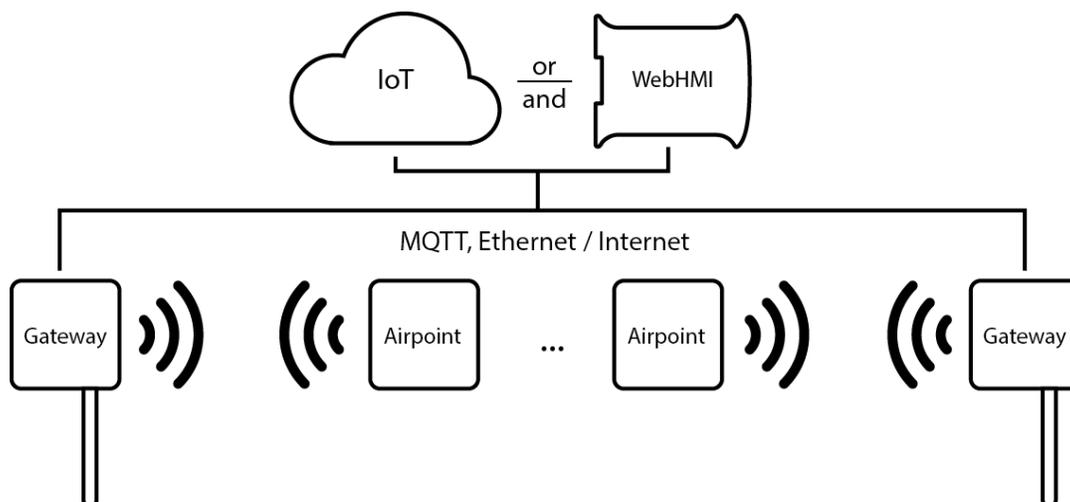
### 4. Tips for choosing the location of the modules

Data between devices is transmitted over a radio channel, therefore, in general, one should be guided by generally known facts about the propagation of radio waves. Whenever possible, choose open places, avoid obstacles, such as blind thick brick and reinforced concrete walls, basements, etc. During the installation of modules, you should monitor such parameters characterizing the quality of communication - RSSI (Received signal strength indicator) - signal level and SNR (Signal-to-noise ratio) - signal-to-noise ratio. These parameters will have the best values near the base station (gateway) and will gradually deteriorate with distance from it. You should also take into account:

- Position the gateway in the center of the object. Provide him with the best possible working conditions if possible.
- Antennas have a specific radiation pattern. The best results are achieved in the case of vertical orientation of the antennas of the modules and the gateway.
- The location of devices near (on the surface) of massive metal structures significantly worsens their working conditions
- To improve the quality of radio signal reception by 7Bit Airgate gateways, you can replace the original antenna with another one with a better gain.
- The quality and range of communication can be influenced by the capacity of the human body, if you hold the device in your hands, as well as by its movement in space.



In large facilities or territories, to improve the quality of coverage, you can use several gateways connected to one server (Broker), which will not affect the configuration of the data collection system, but will only increase its reliability.



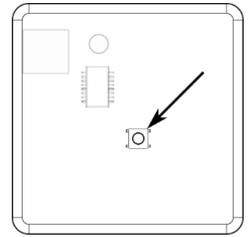
## 5. Setting up 7Bit Airgate

Remove the top cover (you will need it to put the gateway into configuration mode) and apply power in one of the following ways: PoE Y-cable, PoE switch port, micro-USB connector.

To put the gateway into configuration mode, press the button on the gateway circuit board. The green "Status" LED should light up permanently (see the figure above).

After switching to the configuration mode, the gateway creates its own wireless access point in the form of **7bitMQTTGateway** (password **12345678**) with the network address 192.168.4.1.

The setup home page looks like:



### 7BitMQTTGateway

#### WiFiManager

Configure WiFi

Info

Exit

No AP set

Go to the *Configure WiFi* page to configure the gateway. To connect AirGate to the Internet via WiFi, use the list of found networks. When you select the desired network, its name will be substituted in the SSID field. If necessary, provide a password to access this network.

WebHMI_DDS	🔒 📶
TP-LINK	🔒 📶
smarts-ua	🔒 📶
whbox21	📶
Akss	🔒 📶
SM	🔒 📶
M Plus	🔒 📶
DSL-2640U	🔒 📶

SSID

Password

To set up an Internet connection via Ethernet, use the appropriate fields:

Ethernet Settings

IP Addresses

Subnet Masks

Gateways

[Save](#)

[Refresh](#)

No AP set

On the *Info* page, you can quickly reset all settings to factory defaults and view diagnostic information (partially shown):

**Station Subnet**  
0.0.0.0

**DNS Server**  
8.8.8.8

**Hostname**  
espressif

**Station MAC**  
24:A1:60:57:16:00

**Connected**  
No

[Erase WiFi Config](#)

**Available Pages**

---

Page	Function
/	Menu page.

## MQTT broker connection setup:

MQTT Settings

Server

Port

User

Password

Mounting point

If a Mounting point is specified, AirGate will group all messages received from the same AirPoint (as well as its own topics) into one json text message with the Mounting point added as a prefix. This mode is necessary to filter received messages to the desired account in the IoT Level2 cloud server. For example, when specifying a mount point, the package with data from the Airpoint module will look like:

```
"ID111-111-111/AP986EBCOD8A/json  
{ "rssi": "-80", "snr": "12", "uptime": "37", "bat": "414", "v_hard": "9a", "v_soft": "3", "counter_msg": 5",  
"counter_in1": "0", "counter_in2": "0", "temp_int": "283", "hum_int": "-5", "in1": "0", "in2": "0",  
"USB": "0", "gateway": "00165760A12"}"
```

Here:

- **ID111-111-111/AP986EBCOD8A/json** is the address, or topic, the fields as follows:
  - **ID111-111-111** - mounting point
  - **AP986EBCOD8A** - id of the device, from the telegram is send
  - **json** - json format sign
- { ... } – data in the format «key1» -«value1», «key2» - «value2 », ...

Without specifying a mount point, each topic received by the gateway will be sent to the broker as a separate one:

«AP986EBCOD8A/in2 0», «AP986EBCOD8A/counter\_in1 184», etc.

The AirGate has 3 topics of its own, in the following format <[mounting\_point/]id/topic\_name>:

- *uptime* – operating time since the last switch-on (sent every few minutes)
- *v\_hard* – hardware version (sent at the moment of switching on)
- *v\_soft* - firmware version (sent at the moment of switching on)

## Checking connection

There are 3 diagnostic LEDs on the gateway:

- Status (green, upper)
  - Solid light in the configuration mode
  - 1 Hz with the established broker connection
  - 2 Hz without a connection
- Lora (yellow, middle) – blinks at the moment of packet transmission
- Link (green, lower) – blinks with the Ethernet activity

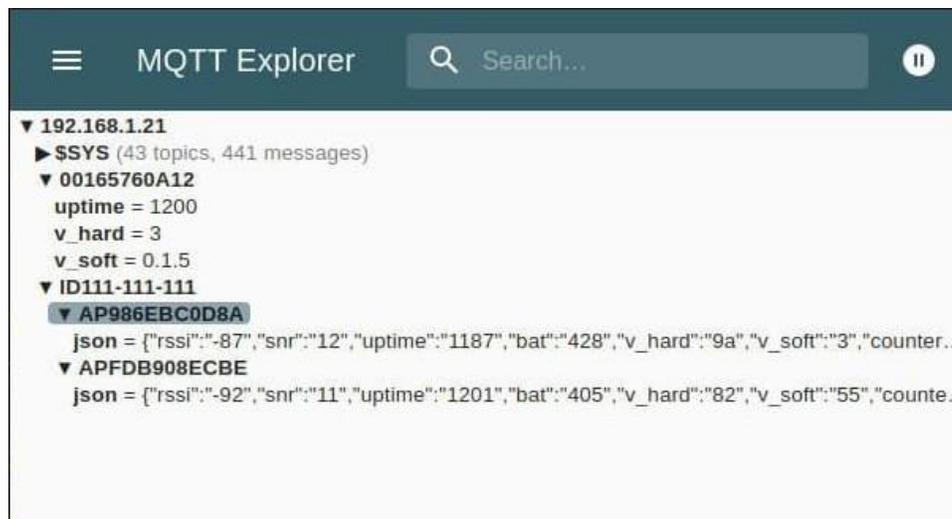


If the Status LED does not indicate the presence of a connection to the broker, first of all, you should check again the settings, the physical connection between the Internet gateway and Airgate (is there an AirGate MAC address in DHCP leases, does the ping command work on the AirGate network address)

It is more convenient to check the connection of the gateway to the data collection system when paired with an AirPoint sensor, since it has a test button for sending a packet, while AirGate sends its topics either when it is turned on or at intervals of 2 minutes. When relaying a package from AirPoint, the yellow (middle) LED on the Airgate should light up.

Then check the correctness of the assignment of topics in the broker, if they are not subtracted.

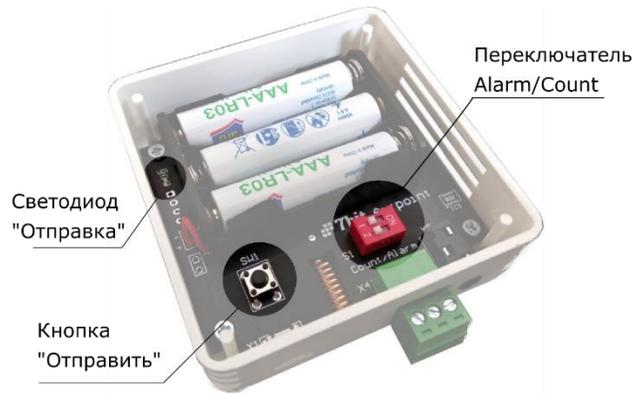
For verification, you can use a third-party utility such as MQTT Explorer.



## 6. Connecting 7Bit Airpoint to the data acquisition system

The module is used in conjunction with the AirGate. Airpoint configuration is partially described in the AirGate configuration section. In the hardware configuration of the module, only the selection of the Alarm / Count mode is available:

- In the *Count* position the data will be sent every 15 minutes
- In the *Alarm* position, data will be sent as soon as any of digital inputs changes its state

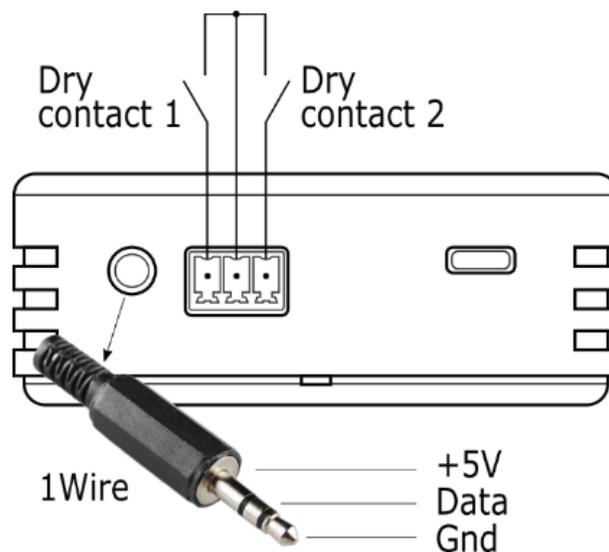


In both modes, if the vibration or shock level is exceeded, Airpoint will send a packet with the current data. This feature can also be used - if the sensor is completely assembled, and you need to check the reception of data from it, just shake the module slightly.

To check the functionality on the module, the following are also provided:

- The button to send a telegram
- LED that signals of telegram sending

### External wiring:



## The parameters available in the AirPoint

Depending on the mounting point set on the AirGate, the parameters from AirPoint will be received either all together in one topic in json format, or separately. The list of parameters is shown in the following table:

Topic	Description	Usage
<u>Data:</u>		
«counter_in1»	Counting input №1	Non-volatile totalizer №1
«counter_in2»	Counting input №2	Non-volatile totalizer №2
«temp_int»	Internal temperature sensor	Temperature sensor 1Wire -40 + 85°C
“hum_int”	Internal humidity sensor	Relative humidity sensor
“temp_ext”	External temperature sensor	Датчик температуры 1Wire -40 + 85°C
«in1»	Digital input #1 state	0/1
«in2»	Digital input #2 state	0/1
“co2”	CO2 level – for the models with a sensor	400 .. 1200
“alarm”	Exceeding vibration limit	It is set if the level of permissible vibration for the sensor has been exceeded. After installation, it resets automatically after 15 minutes.
<u>Service parameters</u>		
“rssi”	Received Signal Strength Indicator	When setting up the connection, it allows you to evaluate the power of the received signal. The boundary values are -115, the more (closer to 0) the better.
“snr”	Signal Noise Ratio	When setting up the connection, it allows you to evaluate the quality of the received signal. The values for reliable reception should be greater than 8.
“bat”	Battery voltage	Estimation of the battery discharge
“v_hard”	Hardware version	For reference
“v_soft”	Software version	For reference
“counter_msg”	Message counter	During setup, it allows you to determine whether the sending of packets continues (by the increment of packets), whether all parcels from the series reach, etc.
«USB»	USB supply indication	0 (batteries) / 1 (USB)
"gateway"	The gateway the message was sent through	
“uptime”	Operating time since the last power up	

When specifying the mount point on the AirGate, the parameters will be packed into one topic in the form “**IDddd-ddd-ddd** / *APhhhhhhhhh* / json with the value

```
{ "rssi": "-80", "snr": "12", "uptime": "37", "bat": "414", "v_hard": "9a", "v_soft": "3", "counter_msg": 5, "counter_in1": "0", "counter_in2": "0", "temp_int": "283", "hum_int": "-5", "in1": "0", "in2": "0", "USB": "0", "gateway": "00165760A12" }
```

where:

- d – decimal figures
- h – hexadecimal figures

## 7. Integrating Airpoints with WebHMI

WebHMI has a MQTT server (broker) which Airgates can be connected to, and its own MQTT client for back read of its own topics or connecting to another broker.

The Airgate has to be configured as follows:

MQTT Settings

Server

Port

User

Password

Mounting point

There is a device template in the WebHMI for adding AirPoints:

The screenshot shows the 'Регистры' (Registers) page in WebHMI. At the top, there is a search bar 'Найти регистр' and a '+ Новый регистр' button. Below this, there are buttons for '+ Новый регистр', '+ Новое соединение', and a dropdown menu 'Инструменты'. A red arrow points to the 'Инструменты' dropdown. The dropdown menu is open, showing options: 'Импорт регистров', 'Экспорт регистров', 'Импорт значений регистров', 'Экспорт значений регистров', 'Сохранить на устройстве', and 'Добавить устройство'. The 'Добавить устройство' option is highlighted with a red box. Below the dropdown, there is a table of registers with columns 'Id', 'Название', 'Lua', and 'Значение'. The table contains several entries, including 'Carel', 'Carel\_clone', 'For mapping', 'Operator', 'Own Modbus', and 'Read Tolstov PC Мос'. At the bottom, there is a 'Групповые действия' (Group actions) button.

You can create a new connection, or specify an existing one, and create a new category for device registers.

Добавить устройство ✕

**Производитель**  
7Bit

**Модель**  
AirPoint

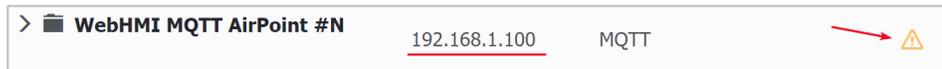
**DeviceID**  
SD235235FDF

**Соединение**  
Создать

Создать категорию для устройства

Отмена Добавить

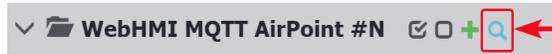
After creation, the connection will be disconnected and marked with a special symbol, you will need to enable it and enter your own WebHMI network address.



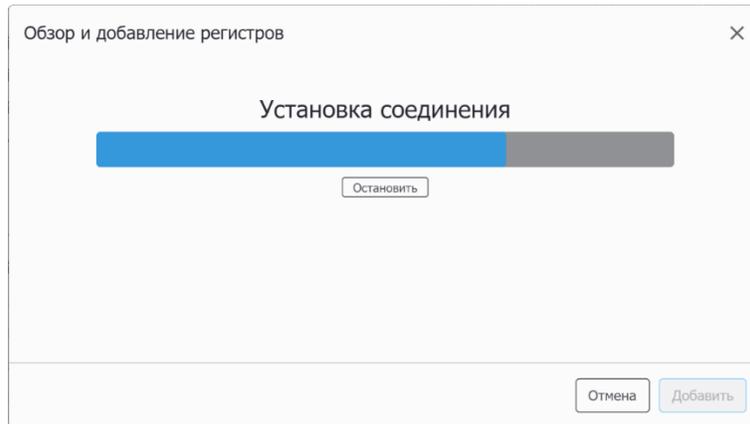
A set of registers will be created inside the connection:

12	WebHMI MQTT AirPoint #N		
<input type="checkbox"/>	119	RSSI	SD235235FDF/rssi
<input type="checkbox"/>	121	SNR	SD235235FDF/snr
<input type="checkbox"/>	123	Battery voltage	SD235235FDF/bat
<input type="checkbox"/>	129	Hardware version	SD235235FDF/v_hard
<input type="checkbox"/>	126	Software version	SD235235FDF/v_soft
<input type="checkbox"/>	127	Message counter	SD235235FDF/counter_msg
<input type="checkbox"/>	130	Input 1 counter	SD235235FDF/counter_in1
<input type="checkbox"/>	120	Input 2 counter	SD235235FDF/counter_in2
<input type="checkbox"/>	133	Temperature	SD235235FDF/temp_int
<input type="checkbox"/>	128	Humidity	SD235235FDF/hum_int
<input type="checkbox"/>	131	Temperature ext	SD235235FDF/temp_ext
<input type="checkbox"/>	132	CO2	SD235235FDF/CO2
<input type="checkbox"/>	122	Input 1	SD235235FDF/in1
<input type="checkbox"/>	124	Input 2	SD235235FDF/in2
<input type="checkbox"/>	125	Motion	SD235235FDF/motion

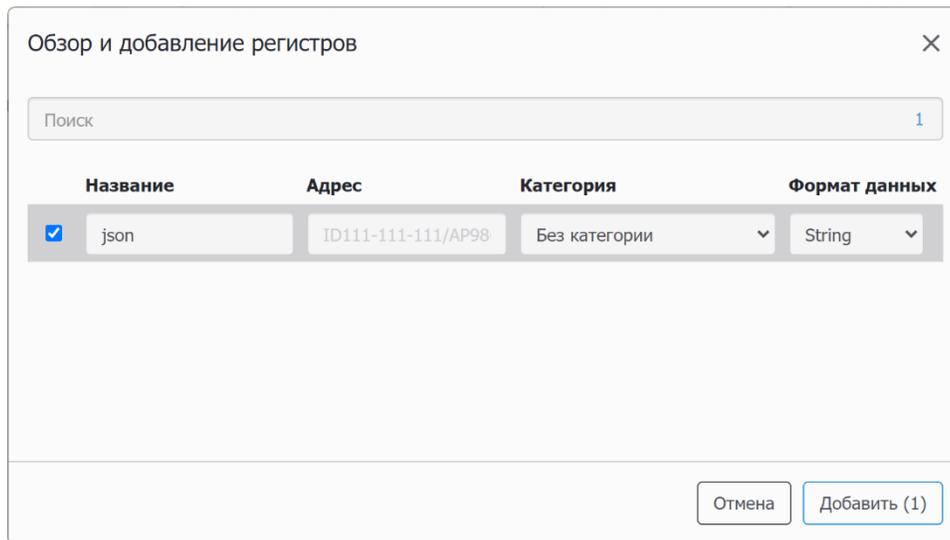
There is a search tool for checking available at the broker sensors, connection and settings check:



After clicking the search button, a search dialog box will open, during the display of which the MQTT WebHMI client will temporarily subscribe to all topics available on the server (topic "#"), and at this time you need to initiate sending data from the sensor.



The topics found will be shown in the list with options to set name, category and data types:



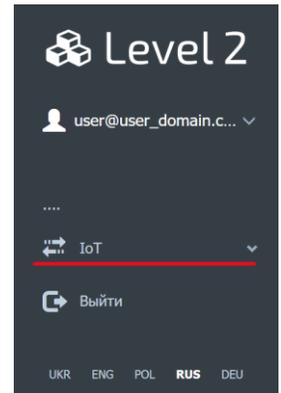
## 8. Integrating Airpoints with Level2 cloud server

The connection of the modules with the cloud server is configured in the Level2 - IoT menu.

Next, two configuration scenarios will be considered - setting up an Airgate (creating a coverage area), and an Airpoint sensor, when there is already coverage at the location of the sensor.

### 1 – Setting up Airgate gateway.

If a mount point has not yet been created in the Level2 office, it must be created on the IoT -> Connection page by clicking the *Create user* button.



☰ Connection

MQTT Integration

A user is required to create a mounting point.

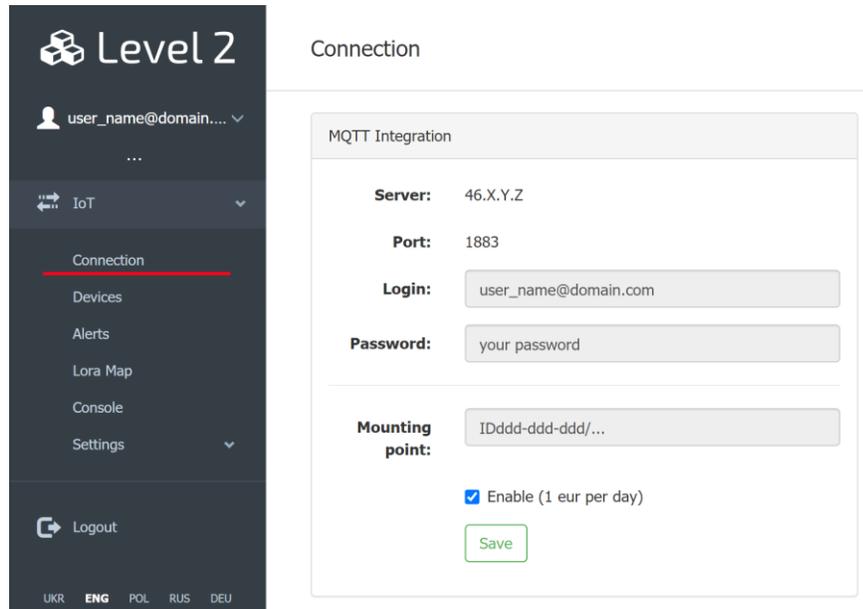
**Server:** 46.101.154.93

**Port:** 1883

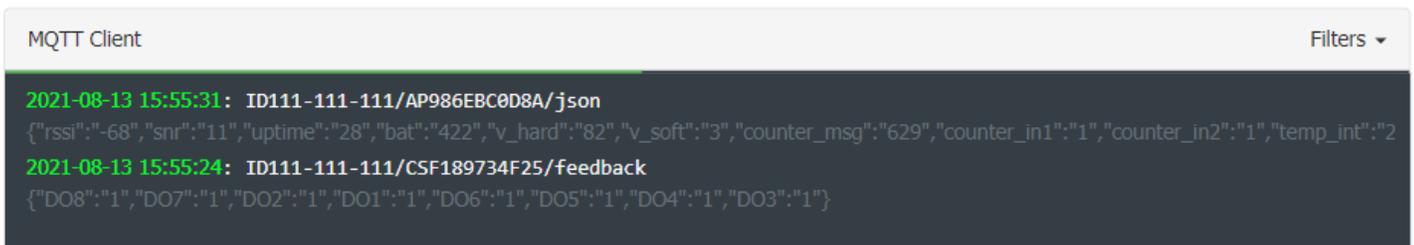
**Login:**

**Password:**

First of all, you need to configure the Airgate according to the MQTT Integration section, specifying in its settings the same mount point number that was allocated for this user (in example *d* - any decimal digit). The option to enable the mount point must be set.

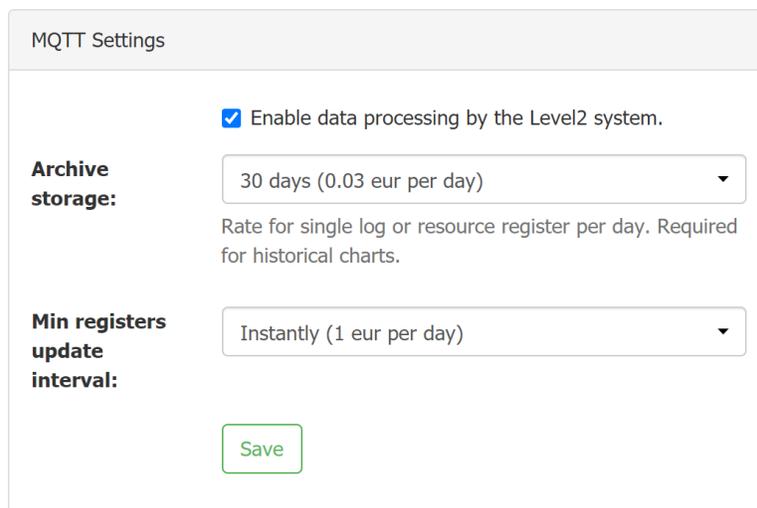


If the AirGate is configured correctly and there is a connection with the sensors, the MQTT debug console should display messages from the sensors or from the gate itself::



## 2 – Setting up Airpoint sensors

In order for the data from the sensors to be used in the Level2 user account, it is necessary to enable the corresponding Level2 services::



These services are paid, the total amount of payments for services from the IoT section consists of the following parts:

- Enabling IoT integration (MQTT broker) – the cost is fixed for the entire cabinet
- Processing current values - the cost is fixed for the entire cabinet
- Archiving registers or resource counters– this tariff is multiplied by the number of registers
- Device activation – this tariff is multiplied by the number of devices

The next step in the cabinet configuration is to add a Device.

The screenshot shows the 'Level 2' IoT management interface. On the left is a dark sidebar with navigation options: Connection, Devices (highlighted), Alerts, Lora Map, and Console. The main area is titled 'Devices' and features a 'New device' button in the top right, indicated by a red arrow. Below the title is a summary table:

Total devices:	Total log registers:	Total meters registers:	Total:
15 x 0.01 EUR = 0.15 EUR	44 x 0.03 EUR = 1.32 EUR	3 x 0.03 EUR = 0.09 EUR	<b>1.56 EUR</b> per day

Below the summary is a search and filter section with fields for Title, Serial, All types, City, and All categories. A table of devices is shown below:

ID	Title	Serial	Type	City	Category	Enable	Registers
100163	My Airpoint	APFDB908ECBE	AirPoint			<input checked="" type="checkbox"/>	4

It is necessary to set the name, type (AirPoint), id, location on the map, and permissions for the users to whom it will be available.

The screenshot shows the configuration page for a device. The page title is 'Devices' and the activation amount is '0.01 eur/day'. The configuration is divided into three tabs: 'Main' (selected), 'Map', and 'Permissions'. The 'Main' tab contains the following fields:

- Title:** My AirPoint
- Type:** AirPoint
- ID Device:** A8CEAE91025 (Used as part of a topic. Example: Mounting point level / **Device level** / Register level)
- Category:** No category
- City:** (empty)
- Address:** (empty)
- Total object area, m<sup>2</sup>:** 1
- Commercial object area, m<sup>2</sup>:** 1

A 'Save' button is located at the bottom of the configuration form.

After clicking Save, the system will offer a list of registers (metrics) that can be immediately created in this device:

Automatic creation of registers. ✕

Register	
RSSI	<input type="checkbox"/>
SNR	<input type="checkbox"/>
Battery voltage	<input type="checkbox"/>
Hardware version	<input type="checkbox"/>
Software version	<input type="checkbox"/>
Message counter	<input type="checkbox"/>
Input 1 counter	<input checked="" type="checkbox"/>
Input 2 counter	<input checked="" type="checkbox"/>
Temperature	<input checked="" type="checkbox"/>
Humidity	<input checked="" type="checkbox"/>
Temperature ext	<input checked="" type="checkbox"/>
CO2	<input checked="" type="checkbox"/>
Input 1	<input checked="" type="checkbox"/>
Input 2	<input checked="" type="checkbox"/>
Alarm	<input type="checkbox"/>
USB	<input type="checkbox"/>
Gateway	<input type="checkbox"/>

The second way to add Airpoint to the system is to be logged into the Level2 office from the phone and scan the QR code printed on the printed circuit board or sensor body. The device entry with the correct id will be automatically added to the device list.



The device list description (see the picture below):

1. Summary about the services (from the IoT division) activated for the devices in the list.
  - a. Total devices = <device number (16)> \* <device activation fee (0,01)> The activated devices are marked with  sign in the **Enable** column.
  - b. Total log register = <register number (45)> \* <one register log fee (0,03)> . The devices with the registers logged are marked with a  sign along with a figure of register count.

c. Total meters registers = <meters number (3)> \* <one register log fee (0.03)>, are marked with a sign  along with meter count figure.

2. Register list filter pane
3. The device has been automatically added (via QR code) . The sign reminds to complete the device setup.
4. The flag for indication and quick activation / de-activation of the device (see p. 1a)
5. The area of indication and quick service activation / de – activation for real-time data process (  sign)logging, meter logging. The  sign shows when the data from a device was read last time (black – no data, blue – more than a day ago, green– data is sent regularly)
6. Buttons for editing the device description, editing its registers, cloning and deleting, respectively.

Total devices:		Total log registers:		Total meters registers:		Total:	
16 x 0.01 EUR = 0.16 EUR		45 x 0.03 EUR = 1.35 EUR 		3 x 0.03 EUR = 0.09 EUR 		<b>1.6 EUR</b> <span style="color: red; font-weight: bold; border: 1px solid red; border-radius: 50%; padding: 2px;">1</span>	
per day							

Title



Serial



AirPoint ▼

City



All categories ▼ 2

Id	Title	Serial	Type	City	Category	Enable	Registers	
100163	Diehl meter	APFDB908ECBE	AirPoint				 <sup>4</sup>  <sup>3</sup>  <sup>2</sup> 	   
100174	AP285DDBD65C office	AP285DDBD65C	AirPoint				 <sup>1</sup>  <sup>1</sup>  	   
100176	AirPoint #2 edited	APFDB908ECBE	AirPoint				   	   
100191	AirPoint #9  Complete the blank <span style="color: red; font-weight: bold; border: 1px solid red; border-radius: 50%; padding: 2px;">3</span>	APEECC859A80	AirPoint				   	   
100194	test for eb	AP1D9F933	AirPoint				 <sup>3</sup>   <sup>1</sup> 	   
100205	Diehl test tty	APFDB908ECBE	AirPoint				 <sup>2</sup>  <sup>2</sup>  	   
100211	Title mine	APEECC859A80	AirPoint				   	   

4

5

6

## Register configuration

For correct processing of data from Airpoint in the system, it is necessary to correctly configure its registers. The register configuration associates it with a specific physical parameter entering the system, normalizes it (sets the offset, scale, etc.) and sets the methods for its processing (storage, notification, etc.)

## “Main” tab of the configuration register

Operation: Readonly  
Title: Input 1  
Topic: ... / json  
Type: JSON  
Key: in1  
Category: No category  
 Include real time

**Operation (Mode)** - either read only or read – write. The "read-only" mode is enabled by default - it protects the user from accidentally changing the parameter in the "opposite direction", in this case, erroneous values from the device may appear, which are actually generated by the user himself.

When the Read / Write mode is set, it becomes possible to select the parameter format - String or Logical. The string parameter, when changed by the user, will be converted from a numeric (or temporary) value to a string, a logical one - by inverting the current state and writing back 0/1, respectively.

Operation: Read/Write  
Input data type: String, String, Logical (0 / 1)

**Title** – register name in the project

**Topic** – for AirPoint should be set to *json*.

**Type** – for AirPoint should be set to *JSON*.

**Key** – the parameter key, which is read from Airpoint for the given register

**Category** – **registers’s category**, it’s used for filtering in the project menus

**Опция include realtime** – turns on real-time data processing for the register

## “Value” tab of the register configuration page

Value = [Value] × [Multiply] + [Shift]

How many digits you want to see after decimal point

kg, ms, A, B, °C, etc

**Value type** – a value “as is”, time or time duration

For time and time duration one can set arbitrary display format:

YYYY-MM-DD HH:mm:ss Help

d[d] h:m:s Help

**Multiply, Shift** – multiplier and offset for normalizing a value

**Precision** – accuracy, decimal point place

**Units of measurement** – this string will be appended to the register value

**Dictionary** – a dictionary of substitutions of the kind: “register value –replacement text”

### “Log” tab of the register configuration

Enables archiving of the register with a specified interval, you can set an arbitrary color to be displayed on the graphs.

The screenshot shows the 'Edit register' dialog box with the 'Log' tab selected. The 'Save data for graphs' checkbox is checked. The 'Time interval' dropdown is set to '1 hour'. The 'Color' field is set to '#337ab7' with a color picker icon. The 'Cancel' and 'Save' buttons are visible at the bottom.

### States tab of the register configuration

The states purposes are:

The screenshot shows the 'Edit register' dialog box with the 'States' tab selected. It displays four states with their respective ranges and colors: 'Disabled state' (From/To fields, color #8B8D91), 'Normal state' (From/To fields, color #5CB85C), 'Warning state' (From/To fields, color #F0AD4E), and 'Alert state' (From/To fields, color #EC4953). The 'Cancel' and 'Save' buttons are visible at the bottom.

- Colorizing the register depending on the range its value falls to: disabled, normal, warning, alert.
- Generating warning / alert messages, upon the register get respective state.

The state is defined according to the following formula:  $(low\ bound) \leq register\ value < (upper\ bound)$

If the register goes into an emergency / warning state, the status line on the Level2 website will turn red / yellow:

The screenshot shows the Level 2 IoT interface. The top navigation bar includes the Level 2 logo, user information (user@user\_dom.com), and a 'New register' button. The main content area displays 'Registers of the device eam\_mqtt\_explorer\_test'. Below this, there is an MQTT Client log showing three entries with timestamps and topics. A table below the log lists registers with columns: Id, Title, Topic, Type, Key, R/W, and Value. The first entry, 'Alert test register', is highlighted in red, and a red arrow points to it. The status bar at the bottom indicates 'Alerts: 2' and 'Warnings: 4'.

Id	Title	Topic	Type	Key	R/W	Value
1	Alert test register	ID111-111-111/CS7CB51D817E/alert_test	Value	—	R/W	3

After the user has viewed the list of alarms, the status bar will turn black again. Alarms will remain in the list even after the value has passed to the normal state, i.e. they "snap" into the system so that the user does not miss them. After the user confirms the alarm, it disappears from the list.

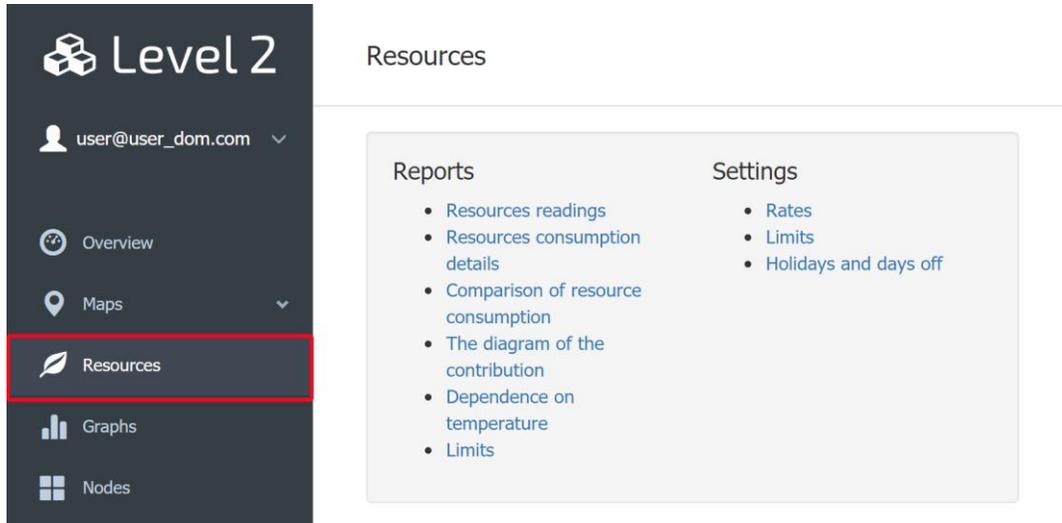
Alerts		
Title	Value	
Alert test register	3	<a href="#">Acknowledge</a>
Alert test register	3	<a href="#">Acknowledge</a>
AC1	0	<a href="#">Acknowledge</a>

### Resources tab of the register configuration

If the register is marked as a resource, its values will be interpreted as readings of the resource counter (water, heat, etc.) and can be used in the Level2 analytics tools

The screenshot shows the 'Edit register' dialog box with the 'Resources' tab selected. The 'Resource type' dropdown menu is open, showing options: None, Electricity, Gas, Heat, Water, Hot water. The 'Rate' field is also visible. The 'Save' button is highlighted in green.

Resource types and their Rates are described in the Level2 Resources section. For more details, see the documentation site docs.webhmi.com.ua



### Mirror tab of the register configuration

The register can be duplicated in the form of a topic in the "hidden" shared broker, to connect to which you need to specify the server level2.webhmi.com.ua, login / password = shared / shared. In this case, another device - a client (subscriber) can read data from another account through the shared area using the topic specified in the **To** field

Retain message option keeps the last value received by the broker, until it re-written with a new value.

### Telegram tab of the register configuration

Allows you to configure sending a message by condition, depending on the register value:

The image shows a screenshot of the 'Edit register' dialog box, specifically the 'Telegram' tab. The dialog has a title bar 'Edit register' with a close button. Below the title bar are several tabs: Main, Value, Log, States, Resources, Mirror, and Telegram (selected). The main content area contains a checkbox labeled 'Send message to Telegram'. Below it is a 'Your chat ID' field with a 'Get it' button. Underneath is a 'Condition' field with a dropdown menu showing '=' and an empty input field. Below that is a 'Message text' text area with a placeholder text 'Add {v} to the text to get the variable.' At the bottom right of the dialog are 'Cancel' and 'Save' buttons.

The "Get it" button allows to move to the chat with @webhmibot and get the chat id, for message exchange.

Edit register
✕

---

Main
Value
Log
States
Resources
Mirror
Telegram

**To** AP58A849C118 / json / rssi ✕

Enable

Retain message

Cancel
Save

Welcome to WebHMI chat Bot!

Your ChatId is 569335646

For help please refer to <http://wiki.webhmi.com.ua> 11:09

### Ordering information

Model	Description
7bit Airgate	Wireless gateway Lora – MQTT (set of gateway and antenna <sup>3</sup> 3dBi)
7bit Airgate_ip65	Wireless gateway Lora – MQTT in the IP65 enclousre (set of gateway and antenna 3dBi)
7bit Airpoint <sup>4</sup>	Wireless telemetry module Airpoint
7bit Airpoint_ip65	Wireless telemetry module Airpoint in the IP65 enclosure
7bit AirComm	Wireless telemetry module with comm. interface <sup>5</sup> for resource meter connection
7bit AirLigt	Wireless dimmer module <sup>6</sup> for street light control

<sup>3</sup> Custom option is an 12dBi external antenna

<sup>4</sup> Custom model is possible with an analog input of voltage/current

<sup>5</sup> Custom model

<sup>6</sup> Custom model