Operating instructions

Gateway

ZB0929

GB

11455705 / 00 04 / 2022

Contents

1 Preliminary note

You will find instructions, technical data, approvals, accessories and further information using the QR code on the unit / packaging or at www.ifm.com.

2 Symbols used

✓	Requirement
▶	Instructions
▷	Reaction, result
[...]	Designation of keys, buttons or indications
→	Cross-reference
	Important note Non-compliance may result in malfunction or interference.
	Information Supplementary note

3 Safety instructions

- The unit described is a subcomponent for integration into a system.
  - The system architect is responsible for the safety of the system.
  - The system architect undertakes to perform a risk assessment and to create documentation in accordance with legal and normative requirements to be provided to the operator and user of the system. This documentation must contain all necessary information and safety instructions for the operator, the user and, if applicable, for any service personnel authorised by the architect of the system.
- Read this document before setting up the product and keep it during the entire service life.
- The product must be suitable for the corresponding applications and environmental conditions without any restrictions.
- Only use the product for its intended purpose (→ Intended use).
- If the operating instructions or the technical data are not adhered to, personal injury and/or damage to property may occur.
- The manufacturer assumes no liability or warranty for any consequences caused by tampering with the product or incorrect use by the operator.
- Installation, electrical connection, set-up, operation and maintenance of the product must be carried out by qualified personnel authorised by the machine operator.
- Protect units and cables against damage.

3.1 Interference with medical devices

The device emits radio waves that may interfere with the operation of electronic devices in the vicinity, including pacemakers, hearing aids and defibrillators.
If you have a pacemaker or other implanted medical product, do not use the device without first consulting your doctor or the manufacturer of your medical product. Keep a safe distance between the device and your medical devices and refrain from further use of the device if you observe permanent impairment of your medical product.

4 Intended use

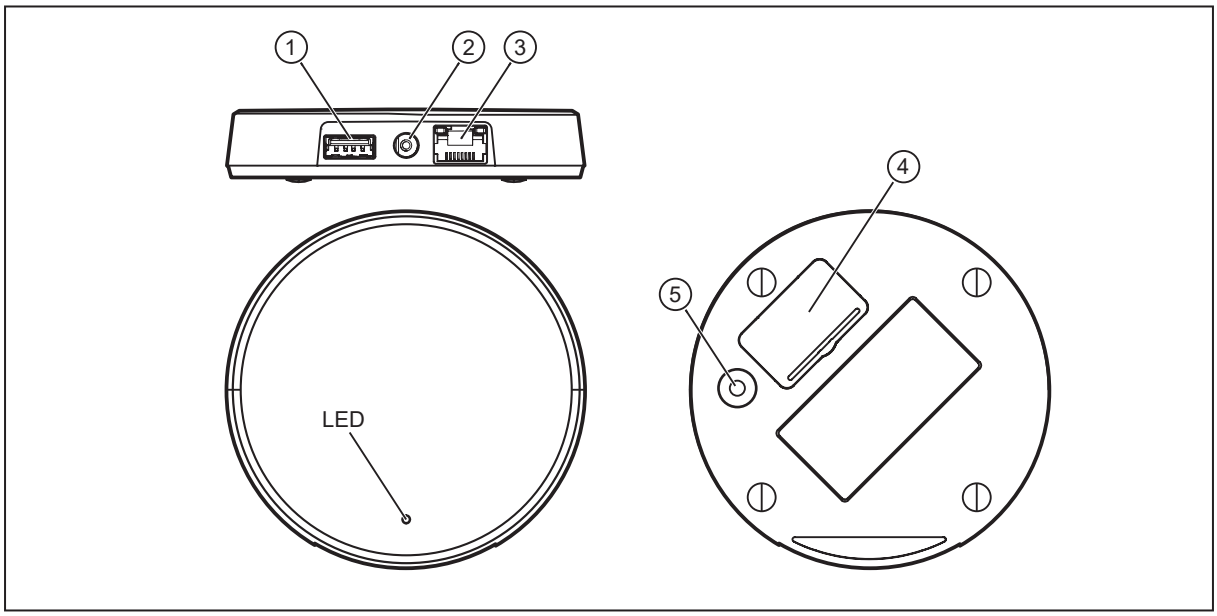
- The gateway, being an MQTT client, transmits sensor data to a backend system via an MQTT broker. In the backend system, the sensor data is made available for visualisation and can be used to carry out evaluations and create alarm rules.
- The gateway is connected to the Internet or network either via a wired Ethernet connection, wirelessly via a mobile network (NB-IoT, 2G or LTE Cat M1) or via a wireless LAN connection.

The device must not be operated within a radius of 20 km around Ny-Ålesund, Spitsbergen, Norway.

5 Installation

- ▶ Position the device close to the sensor.
- ▶ If necessary, use the supplied mounting accessories for installation.

6 Operating and display elements



1: USB connection
2: Mains connection
3: Ethernet connection
4: Micro-SIM card slot
5: Configuration button

7 Electrical connection

▶ Connect the device with the supplied power supply.

8 Set-up

The device switches on automatically when connected to the power supply.

The device's status LED indicates whether the device can access the set PING address.

Recommendation:
Take the address of the backend used (e.g. the MQTT broker) as PING address in order to be able to visually check the connection status.

Status LED	Colour	State	Meaning
	Green	On	The device can access the set ping address and is connected to the Internet/network.
	Blue	On	The device starts up.
		Flashes	The device is in local configuration mode.
	Red	On	The device cannot access the set ping address and is not connected to the Internet/network. ▶ Open the configuration mode and check the settings.

8.1 Gateway configuration

In order to activate the configuration mode, make sure the device start up is completed. Observe the status LED, which is either red or green, for this purpose.

▶ Press the configuration key for at least 10 s until the status LED flashes blue.

▷ The device opens a wireless LAN access point.

▷ The serial number and password for the initial set-up is on the type label on the back of the gateway.

▶ Connect PC, smartphone etc. to the wireless LAN access point. The name of the wireless LAN access point is "ifmgw-[serial number]".

▶ Enter the password.

▷ Windows 10 may ask for a PIN code as primary access point password. Use the option 'password'.

▶ Open the browser on the PC, smartphone, etc.

▶ Open the address 192.168.0.1:3001 in the browser.

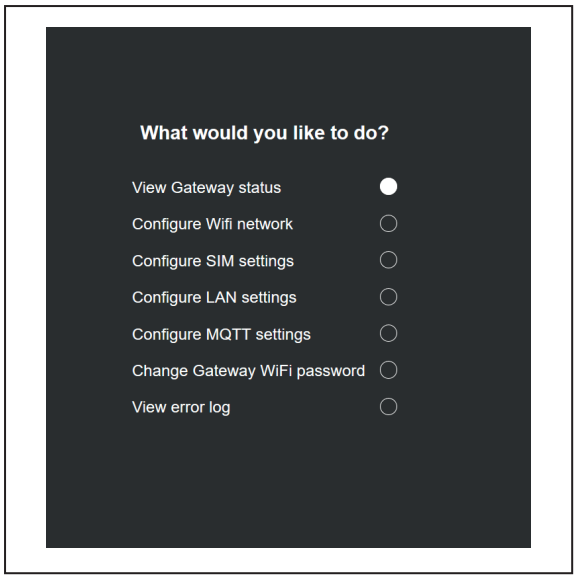
▷ No HTTPS command can be used to open the gateway's web server (http://... is not possible).

▷ As of firmware 6.x, the address 192.168.0.1 must be used without indicating a port.

▷ If the gateway's LAN connection is already configured with a static IP in the address space of 192.168.0.xxx and the gateway is connected to the PC via Ethernet, the configuration interface is accessible at 192.168.1.1(:3001).

▶ Make the necessary configurations.
(E.g. change password for gateway access, check error log, set up connection via wireless LAN, LAN or mobile network or make MQTT broker settings → chapter "Connecting to the Internet / network").

▶ Restart the gateway by pressing [Quit].



8.2 Connecting to the Internet / network

8.2.1 Ethernet connection

▶ The configuration for the Ethernet connection can be found in the configuration menu under "Configure LAN settings"

▶ Use the RJ45 socket for Ethernet connection.

If a Dynamic Host Configuration Protocol (DHCP) server is available, the DHCP client can be activated.

If the DHCP client is deactivated, a static IP address can be configured.

The IP address range of device and PC must match the subnet mask.

	Address	**Address, segment relating to the network address**	**Address, segment relating to the station address**
Subnet mask	255.255.255.0	255.255.255.	0
Wireless gateway ZB0929	e.g. 192.168.0.15	192.168.0.	e.g. 15
PC	e.g. 192.168.0.10	192.168.0.	e.g. 10

▶ Configure a DNS server if necessary.

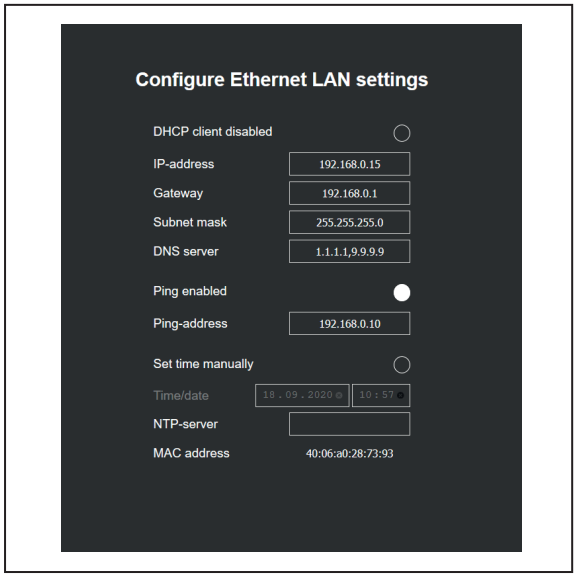
The device can independently send out ping commands,
thus allowing for confirmation that a particular host in a network can be reached. The status LED provides visual information on the connection to the set ping address.

In case of a local connection, the host's IP address, e.g. 192.168.0.10, can be used as ping address. If an internet connection is required (e.g. when using a cloud), any domain can also be used as ping address (e.g. www.ifm.com).

Time synchronisation requires an NTP server. If there is no NTP server, the time can be set manually.

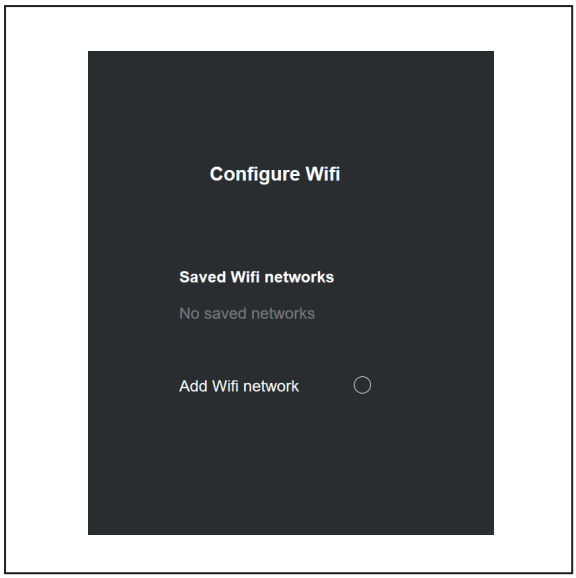
▷ If the gateway is disconnected for a longer period, the time has to be corrected in the configuration menu.

The MAC address of the gateway can be found in the Ethernet setting.



8.2.2 Wireless LAN connection

The gateway can be configured as a wireless LAN client → Configuration menu under [Configure Wifi]



▶ Search for and select a wireless LAN network under [Add Wifi network].

▷ Do not use any special characters (except _-!?) in the name and password for the wireless LAN network.

▷ The wireless LAN network must already be active when the gateway is started up.

8.2.3 Mobile network connection

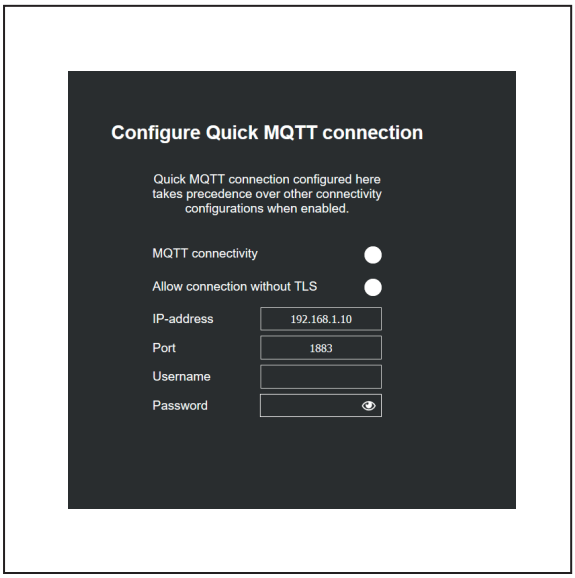
▶ Insert the SIM card before connecting the power supply.

▶ Open the micro-SIM card slot.

▶ Unlock and open the card holder.

- ▶ Insert the micro-SIM card with the contacts facing down and close everything.
- ▶ Supply the gateway with voltage.
- ▶ Open the [Gateway configuration] chapter.
- ▶ Configuration for the mobile communications connection → Configuration menu under [Configure SIM settings].
- ▶ Configure the SIM settings.
- ▶ Restart the gateway by pressing [Quit].

8.2.4 MQTT connection



- ▶ Configuration for the MQTT data connection → Configuration menu under [Configure MQTT settings].
- ▶ Activate MQTT connection
- ▶ Set the broker's IP address and the port.
- ▶ Set the user name and password, if necessary.
- ▶ TLS encryption must be enabled or disabled according to the MQTT broker settings.

▷ The ports [1883] and [8883] are usually reserved for MQTT.

▶ Establish a new rule to the effect that the ports are released for communication in the Windows Defender Firewall.

▷ Some MQTT brokers refuse a connection without username and password in the default settings.

▷ If TLS encryption is not used, it has to be deactivated. Otherwise, no connection can be established.

MQTT data

▷ The device is an MQTT client and always requires an MQTT broker.

The device publishes under the MQTT topic "ifm/VWV/{SID}", with {SID} being a placeholder for the sensor's serial number.

The gateway sends the data to the MQTT broker, deleting it afterwards. In case of disconnection, the data is only retained for a few minutes.

The data is sent as a JSON message and includes the following elements:

JSON information	Meaning
SensorNodeId	Unique serial number of the sensor
GatewayId	Unique gateway number of the gateway
Temperature	Temperature value in °C
Vibration": {"RMS":{"Y":4}}	Effective value of the vibration velocity, division by 100 results in an mm/s value, e.g. 4/100 = 0.04 mm/s
Timestamp	Timestamp in xx
BatteryAlert	Battery status "0" means that there is still sufficient battery capacity, "1" indicates that the battery will still last for about 3 months. The BatteryAlert information is only sent with a "True" value. BatteryVoltage is not recommended as a status indicator.

8.3 Gateway - Changing the access password

Configuration for the gateway password → Configuration menu under [Change Gateway Wifi password].

▶ If necessary, change the factory-set gateway password.

▷ The password cannot be reset to factory settings.

8.4 Sensor setup

▷ Recommendation:
Due to the mesh topology, arrange the sensors around the gateway that is supplied with voltage.

▶ Press the sensor button briefly (switch on the device).

▷ Green sensor LED lights up briefly and goes out again.

▶ Wait until the sensor LED lights up a second time.

▷ Green sensor LED lights up a second time: connection with the gateway established.

▷ Red sensor LED lights up a second time: no connection with the gateway.

▶ Position the sensor closer to the gateway and check the connection again by briefly pressing the sensor button.

▷ Upon initial set-up, the sensor LED may briefly light up in red despite an active connection to the gateway. Check by pressing the button again. If the connection is successful, the sensor LED should then light up green twice.

▷ If the sensor is already switched on, the connection can be checked again by briefly pressing the button.

Switch off the sensor:

▶ Press the button for at least 5 seconds until the LED lights up red for approx. 4 seconds.

You will find detailed information about the sensors at www.ifm.com.

9 Connection to a backend

9.1 Connection to an MQTT broker

The device is an MQTT client and always requires an MQTT broker.

▶ Configure and activate MQTT communication in the gateway.

▷ The device publishes under the MQTT topic "ifm/VWV/{SID}", with {SID} being a placeholder for the sensor’s serial number.

▷ By subscribing to the topic "ifm/VWV/#" all sensors are displayed.

10 Approvals

If approvals are granted, the approval texts of the respective countries shall apply.

10.1 Overview

The overview of the approval status of a device is available on our website at www.ifm.com.

10.2 Europe / EU declaration of conformity

ifm electronic gmbh hereby declares that the device corresponds to the directive 2014/53/EU.

You can find the EU declaration of conformity at the following Internet address:
www.documentation.ifm.com.

11 Maintenance, repair and disposal

The unit is maintenance-free.

- ▶ It is not possible to repair the unit.
- ▶ After use, dispose of the unit in an environmentally friendly way in accordance with the applicable national regulations.

12 ifm IoT Core

12.1 General information

The device has an ifm IoT Core software interface.

The ifm IoT core enables the user to address the device via a REST API from IT networks and to integrate it into Internet of Things applications.

In addition, the ifm IoT Core Visualizer provides a graphical user interface to access data and services of the ifm IoT Core via a browser.

A device description is stored on the device. This device description is a structured, machine-readable data object in JSON format. All current values of parameters, process data, diagnostic data and device information are mapped in this data object. These data values can be read and changed by means of services. Moreover, services for updating the firmware and the ifm IoT Core interface and for triggering on-demand measurements of the connected sensors are offered.

The following pages are intended to provide an overview of communication with an ifm IoT Core.

12.2 Updating the firmware via the IoT Core Visualizer

12.2.1 Starting the ifm IoT Core Visualizer

To start the ifm IoT Core Visualizer:

Requirements:

✓ The device is in normal operating mode.

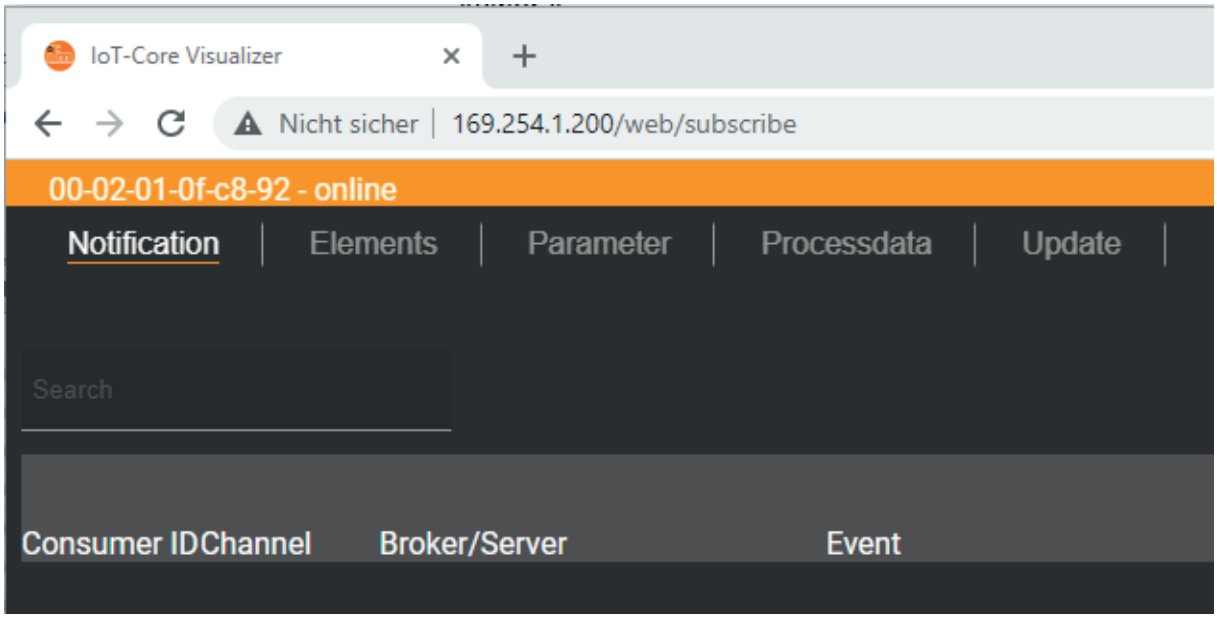
✓ The device is connected to the IT network via the IoT port.

✓ The device is in the same subnet as the PC.

▶ Start the web browser.

▶ Enter the address of the ifm IoT Core Visualizer: http://ipaddress:8001/web/subscribe (e.g. `http://192.168.178.1:8001/web/subscribe`)

▷ The browser shows the ifm IoT Core Visualizer.

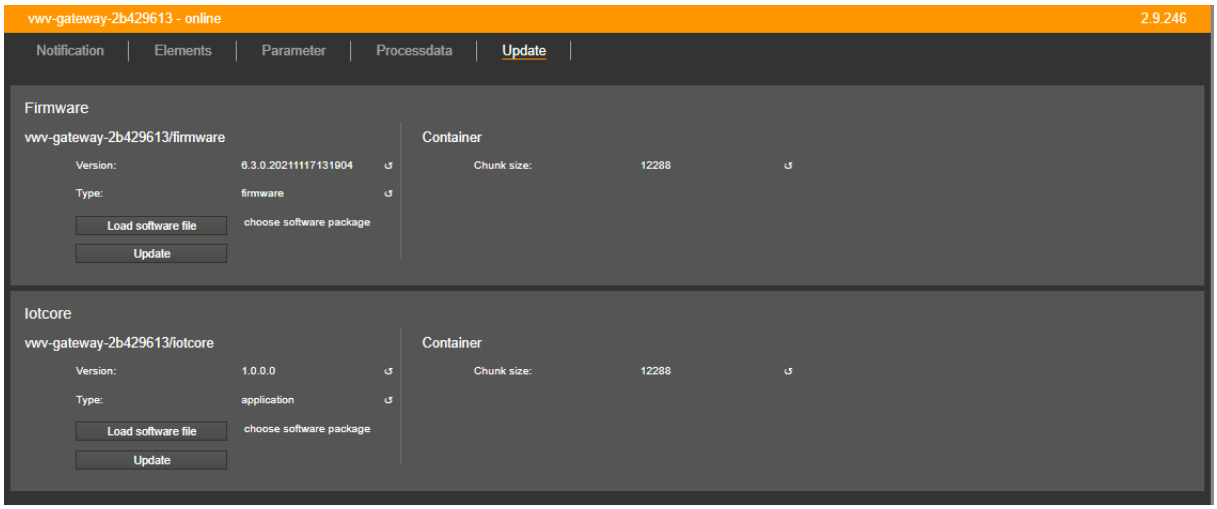


The navigation menu gives the user access to the following functions:

- [Notification]: Create and manage notifications (`subscribe` / `unsubscribe`)
- [Elements]: View the data structure of the IoT core tree and copy URLs for query
- [Parameter]: Read and write sensor parameters
- [Processdata]: Read and write process data
- [Update]: Update the device firmware and IoT core software

12.2.2 Updating the firmware

The [Update] menu page of the ifm IoT Core Visualizer provides information on the current firmware version and the IoT Core version.



▶ During the firmware update, ensure that the device is connected to the supply voltage.

▶ Click on [Load software file] and select new firmware file (*.tar).

▶ Click on [Update] to start the update process.

▷ The firmware of the device is updated.

The firmware update may take several minutes.

▷ If the update process has been successful, the device will restart automatically.

12.3 Accessing the ifm IoT Core using a REST API

Using a REST API, the user can access the ifm IoT Core via HTTP requests.

The following request methods are available.

12.3.1 GET request

Using the GET method the user has read access to a data point.

The syntax of the request to the ifm IoT Core is:

`http://[ip address]:8001/datapoint/service`

Parameter	Description
Ip address	IP address of the IoT interface with port 8001
datapoint	Data point which is to be accessed
service	Service

The response syntax of the IoT Core is:

```
{
"cid":id,
"data":{"value":resp_data},
"code":diag_code
}
```

Parameter	Description
id	Correlation ID for the assignment of request and response
resp_data	Value of the data point; depending on the data type of the data point
diag_code	Diagnostic code

12.3.1.1 Example: GET request

Request via browser to query the product code:

`http://192.168.0.15:8001/deviceinfo/productcode/getdata`

Response:

```
{
"cid": 1,
"data":{"value":"ZB0929"},
"code":200
}
```

Request for a measurement outside the set measurement interval.

`http://192.168.0.15:8001/nodes/vwv-node-ifm12345/processdatanode/on-demand-measurement`

Measurements outside the measurement interval shorten the life of the product.

A command is sent to the VWV00x sensor with the serial number `ifm12345`, triggering an immediate measurement outside the measurement interval. The measured data must then be queried with the `getdata` service under the corresponding data points, e.g. with the request:

`http://192.168.0.15/nodes/vwv-node-ifm12345/processdatanode/vibration-measurement-x/v-rms-x/getdata`

12.3.2 POST request

Using the POST method the user has read and write access to a data point.

The syntax of the request to the IoT Core is:

```
{
"code":"code_id",
"cid":id,
"adr":"/data_point/service",
"data":{req_data}
}
```

Field	Parameter	Description	
code	code_id	Service class	
		• request	Request
		• transaction	Transaction
		• event	Event
cid	id	Correlation ID for the assignment in pairs of request and response; identifier freely assignable by the user	
adr	data_point	Data point of the element tree to be accessed	
	service	Service to be performed	
data	req_data	Data sent to the IoT Core (e.g. new values); syntax depending on the service. Optional: only required for services that send data to the IoT Core (e.g. setdata)	

The response syntax of the IoT Core is:

```
{
"cid":id,
"data":{"value":resp_data},
"code":diag_code
}
```

Field	Parameter	Description
cid	id	Correlation ID for the assignment of request and response
data	resp_data	Values returned by the IoT Core; syntax depending on the service. Optional: only required for services that receive data from the IoT Core (e.g. getdata)
code	diag_code	Diagnostic code

12.3.2.1 Example: POST request to read data

Request:

```
{
"code":"request",
"cid":4711,
"adr":"/deviceinfo/productcode/getdata"
}
```

Response:

```
{
"cid":4711,
"adr": "/deviceinfo/productcode/getdata",
"data":{"value":"ZB0929"},
"code":200
}
```

12.3.2.2 Example: POST request to write data

Request:

```
{
"code":"request",
"cid":4711,
"adr":"/nodes/vwv-node-ifm12345/nodeconfig/measurement-interval/setdata",
"data":{“newvalue“:120}
}
```

12.3.3 Diagnostic codes

Code	Text	Description
200	OK	Request successfully processed
230	OK but needs reboot	Request successfully processed; the IO-Link master must be rebooted
231	OK but block request not finished	Request successfully processed; but block request not finished

Code	Text	Description
232	Data has been accepted, but internally modified	New values have been accepted but adjusted by the device (master cycle time)
400	Bad request	Invalid request
403	Forbidden	Forbidden request
500	Internal Server Error	Internal fault
503	Service Unavailable	Service not available
530	The requested data is invalid	Invalid process data

12.3.4 Getting started

To read the device description and the address tree of the device:

▶ Send the following POST request to the device:
`{"code":"request","cid": 1,"adr":"gettree"}`

▷ The device returns the device description as structured JSON object.

▶ Identify all substructures and the data points contained therein in the tree structure of the JSON object.

▶ Identify the applicable services for the access to substructures and the data points contained therein.

12.3.5 General functions

The device has the element `device`.

The following services can be used on the root element of the type `device`:

Service	Description
../gettree	Provide the complete tree or subtree of the device description (JSON)
../getidentity	Read device information
../getdatamulti	Read several parameter values sequentially
../getsubscriberlist	Print a list of all active notification subscriptions
../querytree	Search device description for specific elements

Depending on the read and write access rights, the following services can be applied to elements of the type `data`:

Service	Description
../getdata	Read the value of the element
../setdata	Write the value of the element

12.3.5.1 Example: Reading several parameter values of the gateway simultaneously

Task

The following current values are to be read by the device: IoT Core version, serial number

Solution

Read the current parameter values using the service `getdatamulti` (data point version: `/iotCore/version`; data point serial number: `/deviceinfo/serialnumber`)

POST request example

Request:

```
{
"code":"request",
"cid":4711,
"adr":"/getdatamulti",
"data":{"datatosend":["/iotCore/version","/deviceinfo/serialnumber"]}
}
```

Response:

```
{
"code": 200,
"cid": 5,
"adr": "/getdatamulti",
"data":
{
"/iotCore/version": { "code": 200, "data": "1.0.0.0" },
"/deviceinfo/serialnumber": { "code": 200, "data": "ifm12345" }
}
}
```

12.3.6 Defined tree structure

The tree structure of the IoT Core interface is divided into various substructures.

These include information about the device, functions for updating the device firmware and the IoT core interface.

The `nodes` structure is offered as an additional substructure. It builds up a separate substructure for each connected VWV00x sensor.

Structure: `[ip address]:8001`

Substructure	Description
`[ip address]:8001`	
../deviceinfo	Information about the device
../firmware	Information and update of the device firmware
../iotCore	Information and update of the IoT Core interface
../nodes	Substructure

12.3.6.1 Gateway information

Substructure: `[ip address]:8001/deviceinfo`

Profile: `deviceinfo`

Available data points:

Data points	Description	Access
`deviceinfo`		
../deviceclass	IoT core class of the device	r
../devicesymbol	Image of the device	r
../hwrevision	Hardware revision of the device	r
../productcode	Product code of the device	r
../productinstanceuri	Biunique URI of the device	r
../producttext	Device name	r
../serialnumber	Serial number	r
../swrevision	Software revision of the device	r
../vendor	Vendor	r
r ... read only		

Applicable services:

Services	Description
../.../getdata	Read the data point

12.3.6.2 IoT Core information and update

Substructure: `[ip address]:8001/iotCore`

Profile: `software/updateablesoftware`

Available data points:

Data points	Description	Access
`iotCore`		
../installstatus	Installation status	r
../container	URI for the new IoT Core application software	rw
../type	Type of software to be updated (= `iotCore`)	r
../version	Current IoT Core application software	r
r ... read only rw ... read and write		

Applicable services:

Services	Description
../…/getdata	Read the data point
../…/setdata	Write the data point
../install	Command to execute the installation/update

The first step to update the software is to write the URI to the data point `container`. Afterwards, the service `install` can be executed.

12.3.6.3 Firmware information and update

Substructure: `[ip address]:8001/firmware`

Profile: `software/updateablesoftware`

Available data points:

Data point	Description	Access
`firmware`		
../installstatus	Installation status	r
../container	URI for the new device firmware	rw
../type	Type of software to be updated (= firmware)	r
../version	Current device firmware	r
r ... read only rw ... read and write		

Applicable services:

Services	Description
../…/getdata	Read the data point
../…/setdata	Write the data point
../install	Command to execute the installation/update

The first step to update the firmware is to write the URI to the data point `container` . Afterwards, the service `install` can be executed.

12.3.6.4 Sensors connected to the gateway

For each connected sensor, an additional substructure with the corresponding data is created in the `nodes` substructure. The fact that further `devices` are included in this structure is indicated via the `devicegroupmanager` profile. The `../nodes/devicestatuschanged/datachange` event can be used to detect changes in the sensor list and thus in the `nodes` subtree.

Substructure: `[ip address]:8001/nodes`

Profile: `devicegroupmanager`

Available data points:

Data points	Description	Access
`nodes`		
../devicestatus_changed	Indicates whether the sensor list has changed	r
../managed_devices_classes	List of devices in the substructures (= VWV001|VWV002) (Important: if no devices are available or if a higher-level system wants to be informed about any new device)	r
r … read only		

Available events:

Events	Description
../devicestatus_changed/datachanged	Event when the sensor list has changed

Value	TextID	Description
1	device_ok	Device response "OK
2	device_new	A new device has been added to the list
3	device_disconnect	A device has been removed from the list (planned)
4	device_reconnect	A known device that did not communicate has been added again
5	device_lost	Due to persistent problems, a device has been removed from the list
6	device_comerror	A device communicates, but there are communication errors that cannot be solved
7	device_timeout	A device does not respond within the time period
90	device_error	The device itself reports an error (unexpected)
91	device_err_cantintegrate	The device refuses onboarding (e.g. because it has already been onboarded to another system)
92	device_err_compat	The device / device manager reports are incomplete

Substructure: `[ip address]:8001/nodes`

Profile: `devicegroupmanager`

Available data points:

Substructure	Description
`nodes`	
../vwv-node-[serial number]	A substructure with type "device" is created for each connected sensor

Applicable services:

Services	Description
../devicestatus/datachanged/subscribe	Event subscription
../devicestatus/datachanged/unsubscribe	Event unsubscription
../devicestatus/datachanged/triggerevent	Trigger in case of event

12.3.6.5 Sensors

Each VWV00x sensor has its own substructure, which in turn contains substructures to organise the data in a better way. As each structure represents a particular sensor, the `device` profile with the corresponding standard services is also assigned to the structure.

Substructure: `[ip address]:8001/nodes/vwv-node-[serial number]`

Profile: `device`

Available data points:

Substructures	Description
`vwv-node-[serial number]`	
../deviceinfo	Sensor information
./processdatanode	Process data of the sensor
../nodeconfig	Sensor configuration

Applicable services:

Services	Description
../gettree	Provide the complete tree or subtree of the device description (JSON)
../getidentity	Read device information

Services	Description
../getdatamulti	Read several parameter values sequentially
../getsubscriberlist	Print a list of all active notification subscriptions
../querytree	Search device description for specific elements

12.3.6.6 Sensor information

Substructure: `[ip address]:8001/nodes/vwv-node-[serial number]/deviceinfo`

Profile: `deviceinfo`

Available data points:

Data points	Description	Access
`deviceinfo`		
../deviceclass	IoT core class of the device	r
../devicesymbol	Image of the device	r
../hwrevision	Hardware revision of the device	r
../productcode	Product code of the device (VWV)	r
../productinstanceuri	Biunique URI of the device	r
../producttext	Device name	r
../serialnumber	Serial number	r
../swrevision	Software revision of the device	r
../vendor	Vendor	r
r ... read only		

Applicable services:

Services	Description
../.../getdata	Read the data point

12.3.6.7 Sensor process data

The sensor data is located in the `processdatanode` substructure. For the purpose of clarity, the latter is listed below with all substructures. Depending on the VWV00x sensor variant, only data of the existing measuring axes is displayed.

The `processdata` profile has been assigned to all process values and the `timestamp` profile has been assigned to all timestamps.

Substructure: `[ip address]:8001/nodes/vwv-node-[serial number]/processdatanode`

Profile: ---

Available data points:

Data point	Description	Access
`processdatanode`		
../vibration-measurement-x/v-rms-x	v-RMS value in the x-axis	r
../vibration-measurement-x/v-rms-x/unit	Unit of the v-RMS value	r
../vibration-measurement-x/timestamp	Time at which the measured value was received in the gateway	r
../vibration-measurement-y/v-rms-y	v-RMS value in the y-axis	r

Data point	Description	Access
../vibration-measurement-y/v-rms-y/unit	Unit of the v-RMS value	r
../vibration-measurement-y/timestamp	Time at which the measured value was received in the gateway	r
../vibration-measurement-z/v-rms-z	v-RMS value in the z-axis	r
../vibration-measurement-z/v-rms-z/unit	Unit of the v-RMS value	r
../vibration-measurement-z/timestamp	Time at which the measured value was received in the gateway	r
../temperature-measurement/temperature	Temperature value	r
../temperature-measurement/temperature/unit	Unit of the temperature value	r
../temperature-measurement/timestamp	Time at which the measured value was received in the gateway	r
r ... read only		

12.3.6.8 Sensor configuration

The data point `measurement-interval` has the `parameter` profile assigned to it.

Substructure: `[ip address]:8001/nodes/vwv-node-[serial number]/nodeconfig`

Profile: ---

Available data points:

Data point	Description	Access
`nodeconfig`		
../measurement-interval	Measurement interval of the sensor (>30 min)	rw
../measurement-interval/unit	Unit of the interval [min]	r
r ... read only rw ... read and write		

Applicable services:

Services	Description
../…/getdata	Read the data point
../…/setdata	Write the data point