

AN-020 IoT Radar Sensor with WiFi Interface

The OPS243 radar sensor with WiFi wireless connectivity is a complete IoT (Internet of Things) solution for traffic and object monitoring. All that is needed to access remote data is power for the sensor and a WiFi network to transport the data to the cloud for visualization, processing, or storage. The sensor is pre-configured to seamlessly pass data through the WiFi network to the cloud. This application note describes the use of the OPS243 network architecture, how to connect the sensor to a WiFi network, and visualizing the sensor data.

Network Architecture

The WiFi interface on OPS243 is designed for simple connection to a WiFi network and viewing of sensor data. Connecting the sensor to a WiFi network can be done over the wired API or using the SoftAP app ([Android](#) or [iOS](#)) from the WiFi module provider. The older WiFi Radar Sensor app for smartphones and tablets ~~is~~ available on the Google Playstore is being retired.

The overall network architecture is shown in Figure 1. The OPS243 uses the local WiFi network to pass data to an MQTT (Message Queuing Telemetry Transport) Broker in the cloud. The MQTT Broker is pre-configured by OmniPreSense to receive data from the sensor. Data from the sensor is published to the MQTT Broker as it becomes available and is passed on by the broker to any device which is a subscriber to that data. The owner of the sensor uses unique MQTT username and password to subscribe to the sensor data. In addition, API commands can be passed back to the sensor to configure it as desired or query its status.

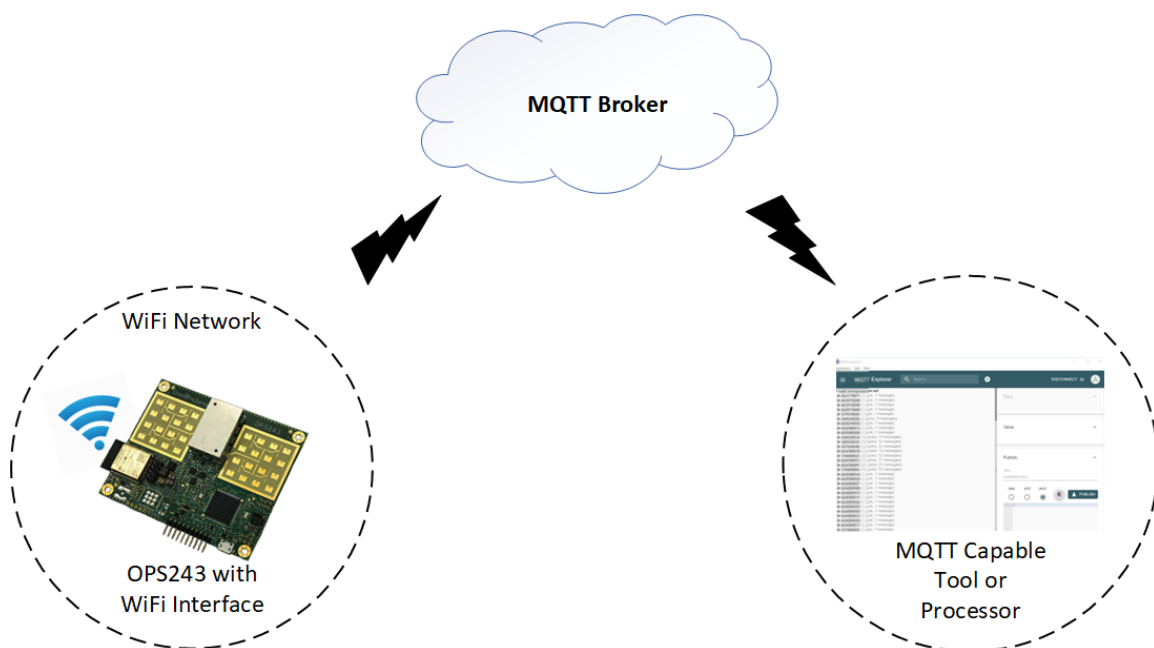


Figure 1. OPS243 Wireless Network Architecture

API WiFi Network Connection

Starting with OPS243-A code v1.2.4 and OPS243-C code v1.2.7, the API has been updated to scan and connect to a WiFi network. Sensors running these code versions or later require WiFi module code v2.0.x or later to make use of the API commands described below. See Table 2 for list of WiFi module code versions and their features supported.

The API provides several commands to enable the WiFi connection, query the WiFi status, or perform an over-the-air (OTA) update. The API commands are summarized in Table 1.

Table 1. WiFi Control API

| Command | Name | R/W | Value |
|------------|--------------------------|-------|--|
| \$V | WiFi Version | Write | Provides the WiFi code version. Response: {"name": "mqtt_blufi_uart", "version": "v2.0.4", "build": "Sep 22 2025 21:46:53"} |
| \$G | WiFi Network | Write | Returns currently set WiFi network ssid and password {"wifi": {"values": "ssid": "OmniPreSense", "password": "*****"}} |
| \$W | WiFi Scan | Write | Returns list of top WiFi networks detected by order of signal strength. {"results": [{"ssid": "network1", "channel": 6, "rssi": -48}, {"ssid": "network2",} |
| \$C=cccccc | WiFi Network SSID Select | Write | Set the desired WiFi network WiFi to join, ccccc=SSID. (max 32 characters) Response: {"values": {"ssid": "network1"}} |
| \$P=pppppp | WiFi Network Password | Write | Set the password of the WiFi network to connect to, ppppp=password. Response: {"values": {"password": "xxxx"}} |
| \$I | WiFi IP Status | Write | Returns the IP address assigned and the MAC address of the WiFi module. Use to confirm connection to WiFi network. Reponse: {"ip": {"MAC": "08A6F7AB8F6C", "address": "10.0.0.97", "netmask": "255.255.255.0", "gateway": "10.0.0.1"}} |
| \$O | OTA Update Check | Write | Returns the check if the current WiFi module code is the latest or an update is available. A response looks like {"ota": "Currently running v2.0.7, OTA is v3.0.14"} if a new version of code is available. |
| \$U | OTA Update | Write | Trigger an OTA update of the WiFi module code. The update may take a little time to download and update the code. The \$V command can be used to monitor the code version and when the update has completed. |

The process to connect the sensor to a WiFi network is as follows:

1. Scan for available WiFi networks with the \$W command
2. Set the SSID for the desired network with the \$C=cccccc command

3. Set the network password if required with the \$P=ppppp command
4. Confirm connection with the \$I command, showing an IP address has been assigned

In addition to these commands, the version of WiFi code on the sensor is returned with the \$V command. Future updates will provide over-the-air (OTA) updates of the WiFi code and ability to change the MQTT broker that data is sent to.

SoftAP WiFi Network Connection

Starting with v3.0.x WiFi module code, the WiFi network connection can be made using the SoftAP app from the WiFi module supplier. This app provides a convenient way to wirelessly make the sensor connection to a WiFi network. The prior WiFi Sensor App from OmniPreSense will no longer work with v3.0.x code.

Using this method, the OPS243 WiFi sensor starts up as a WiFi Access Point (AP). The users' phone or tablet connects to the sensors WiFi network and exchanges WiFi network information with the sensor, allowing it to connect into the desired WiFi network. The phone/tablet and sensor then drop from the sensor WiFi network and join the desired WiFi network.

The steps for the Android and iOS app are the same but may have some differences. Start by powering on your OPS243 WiFi sensor and downloading the SoftAP app ([Android](#) or [iOS](#)) and opening it. On the initial screen (Figure 2), select the Provision button.



Figure 2. SoftAP Opening Screen

The next screen (Figure 3) provides the option to scan a QR code. If a QR code was provided with your sensor, scan it to link to the sensor. If no QR code was provided, select the “I don’t have a QR code” button.

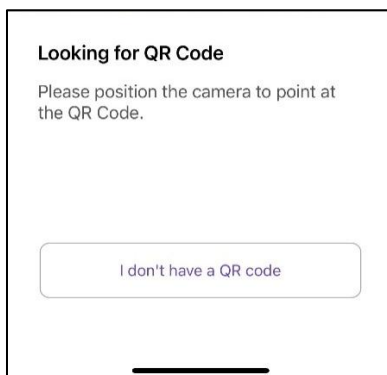


Figure 3. QR Scanner Screen

If the QR code is used, a new popup will request to connect to the sensors WiFi network (Figure 4). Press Join and it will scan for available WiFi networks to connect to. Select the appropriate WiFi network and password if necessary to connect the sensor to WiFi network (Figure 8).

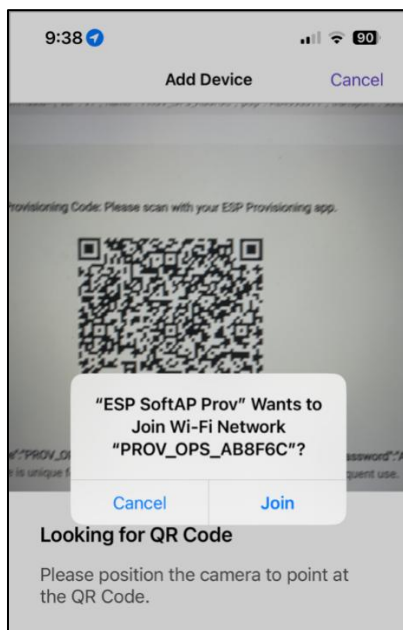


Figure 4. Connecting to Sensor WiFi via QR Code Scan

If no QR code was provided, a new screen appears to Connect to your sensor WiFi (Figure 5). In some cases, iOS may pass through to WiFi Settings screen and you may need to go back to the main WiFi network connection screen. Press Connect to get to the next screen which provides a list of WiFi networks available to connect to (Figure 6). The sensor WiFi network name is PROV_OPS_XXXXXX with the last characters being the last six characters of the sensor MAC address. Connect to the sensor WiFi network by selecting it (Figure 7) and providing the network password. The password is the serial number of the sensor which is provided separately.



Figure 5. Connecting to Sensor AP WiFi Network (no QR code)

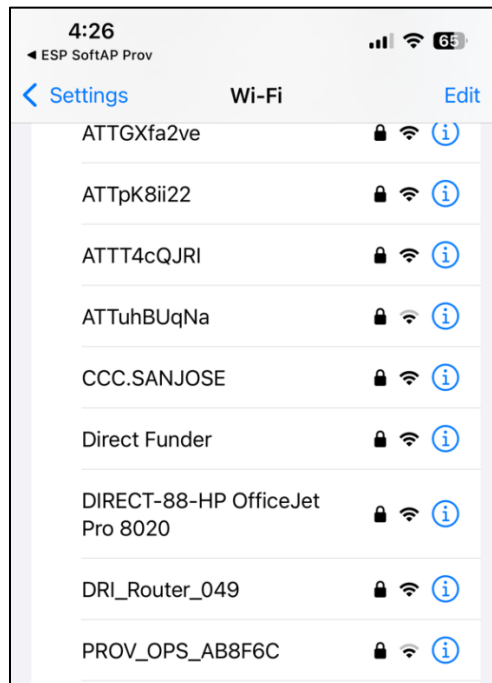


Figure 6. Sensor WiFi Network to Connect (ex. PROV_OPS_AB8F6C)

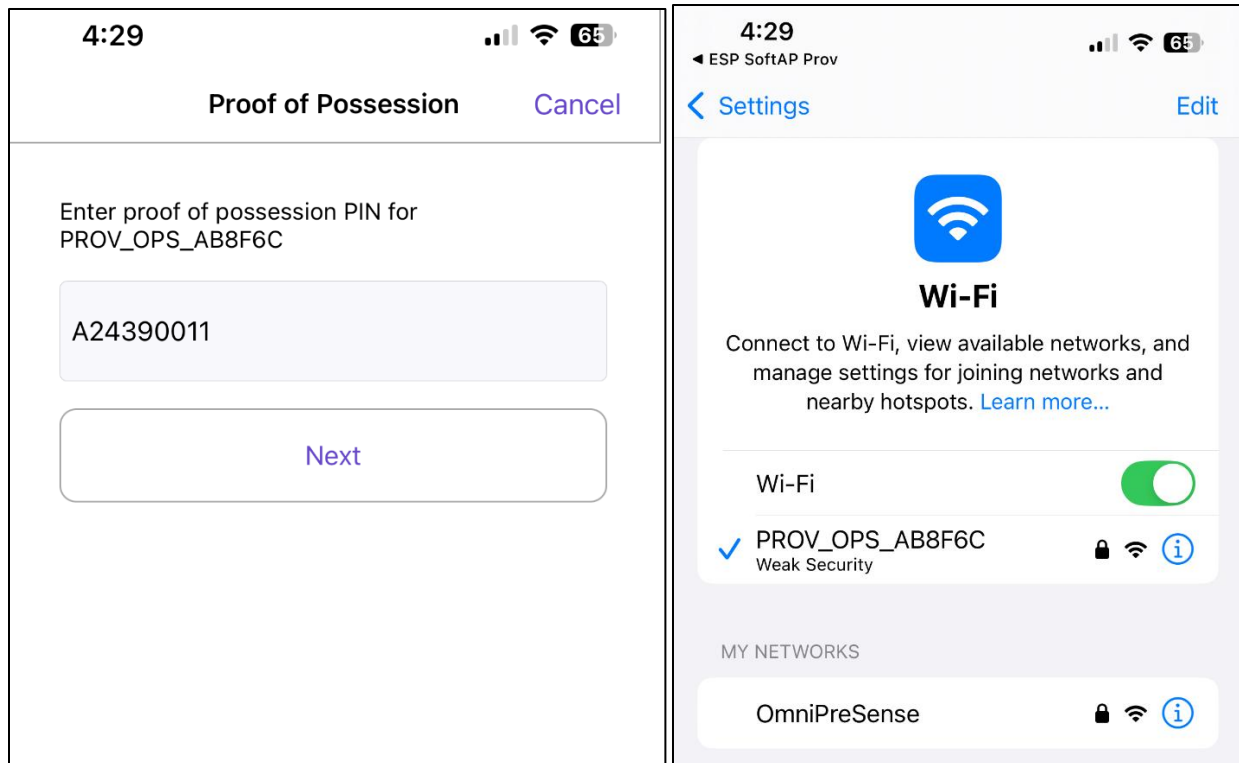


Figure 7. Connected to Sensor WiFi Network with Password

Once connected, the app will provide a new list of networks to connect the sensor to. Select the desired network (Figure 8) and if required, provide the network password.

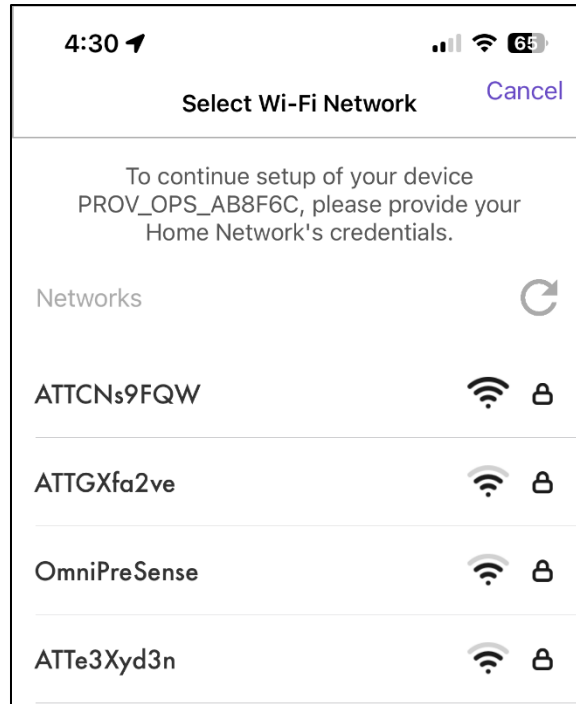


Figure 8. Connected to Sensor to Desired Network (ex. OmniPreSense)

The SoftAP will do the final connection of the sensor to the desired WiFi network and reconnect your phone/tablet to the network it was previously connected to (Figure 9). If a sensor has been provisioned for a WiFi network and needs to be redirected to a new network within range, power cycle the sensor and the above connection sequence is available to use again.

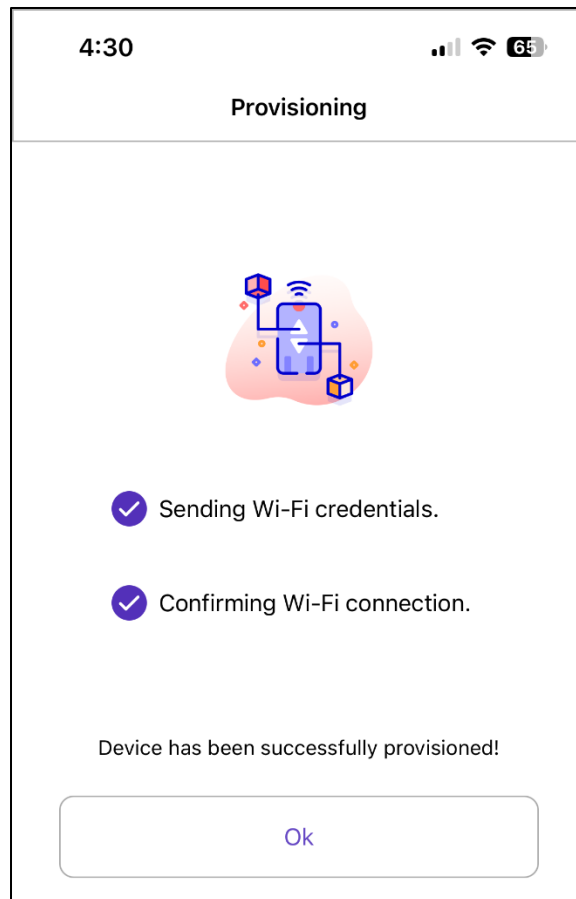


Figure 9. Provisioned Sensor

OTA Update

Starting with v2.4 OPS243-A and v2.7 OPS243-C code, API commands are available to perform OTA code update of the WiFi module. The command \$O is available to check if the WiFi module code is the most recent or available to update. A \$O response message for code available to update looks like the below:

```
{
  "ota": "Currently running v2.0.7, OTA is v3.0.14"
}
```

If the current code is the latest, the following response message appears:

```
{
  "ota": "OTA version matches. not updating.",
  "ota": "Currently running v2.0.7, OTA is v2.0.7"
}
```

To update the code, issue the \$U command. Make sure the sensor stays fully powered-on during the update. The update may take upwards of 60 seconds to complete. Use \$V command to check if the code has completed the updated.

For sensors remotely connected, the WiFi module code can be updated by issuing equivalent API commands over MQTT. On a tool such as MQTT Explorer, under the Publish heading on the right side, set the topic as <serial_number>/command. Select the Raw button and in the box below enter either of the equivalent JSON messages for \$O or \$U. Then hit the Publish button.

\$O: {"ota": "get"}

\$U: {"ota": "update"}

If you use MQTT Explorer, on the left side a WiFi connection message will flash when the update is complete and the sensor reconnects into the network.

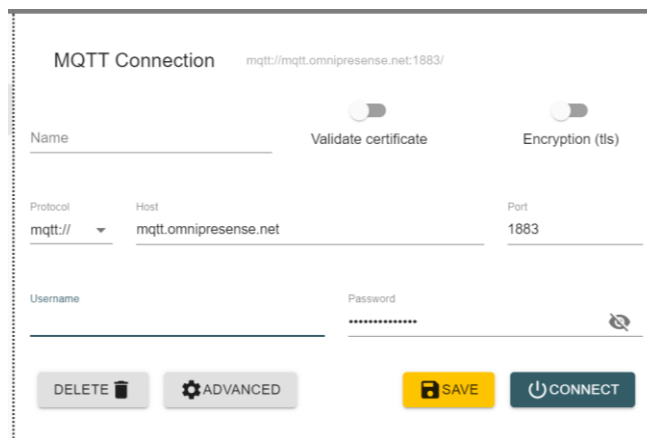
Older sensors with the original WiFi module code (v1.0.1) can update over MQTT in a similar manner. They do not provide any feedback on the process, and the user should wait for a short time to confirm the update worked. Once updated, sending {"system": "version"} over MQTT to topic <serial_number>/command-esp will show the updated code version. An example response using MQTT Explorer is:

```
{ "name": "mqtt_blufi_uart", "version": "v2.0.7", "build": "Oct 3 2025 05:15:37" }
```

Data Visualization

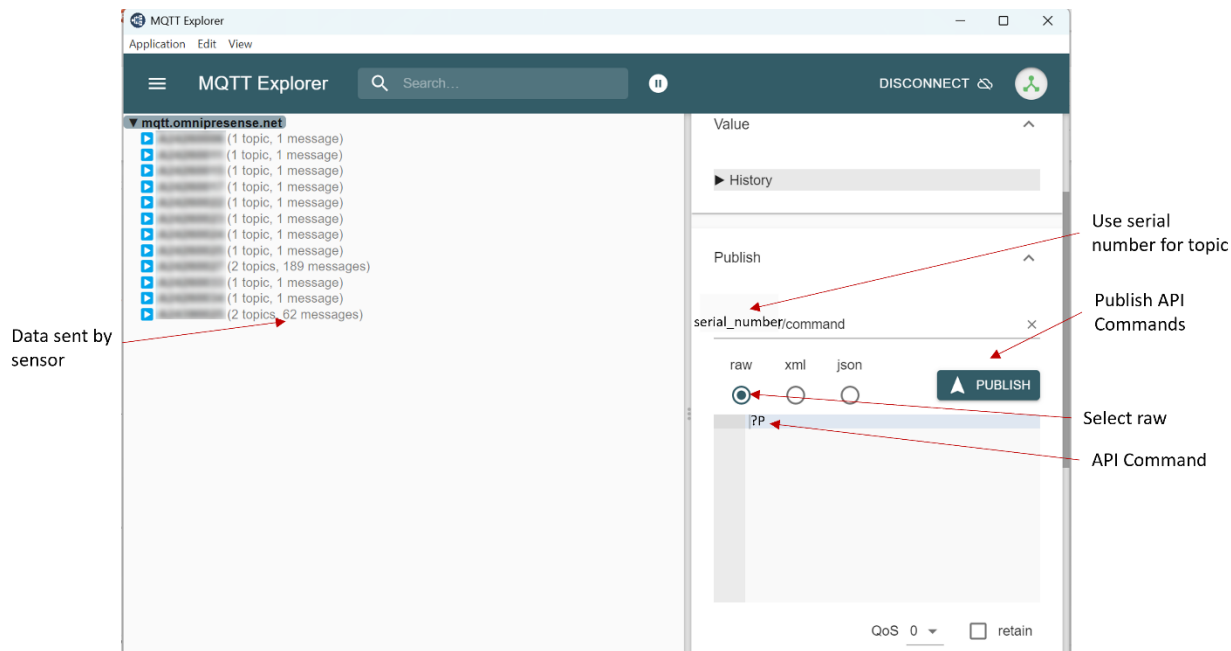
Your OPS243 sensor data can be visualized on any device that can subscribe to MQTT data. A processor board can run code to accept the data for visualization, processing, or storage. A useful PC based tool is [MQTT Explorer](#). The login screen for MQTT Explorer is shown in Figure 10. Contact OmniPreSense

customer service to request the MQTT username and password for your sensor. Once connected, a new screen will show the data sent by the sensor and you're able to send API commands back to the sensor.



The MQTT Connection screen features a title bar with 'MQTT Connection' and the address 'mqtt://mqtt.omnipresense.net:1883/'. Below this are three toggle switches: 'Name' (disabled), 'Validate certificate' (disabled), and 'Encryption (tls)' (disabled). The main form includes fields for 'Protocol' (set to 'mqtt://'), 'Host' (set to 'mqtt.omnipresense.net'), and 'Port' (set to '1883'). There are also fields for 'Username' and 'Password' (masked with dots). At the bottom, there are four buttons: 'DELETE' with a trash icon, 'ADVANCED' with a gear icon, 'SAVE' in yellow, and 'CONNECT' with a power icon.

Figure 10. MQTT Explorer Log-In Screen



The MQTT Explorer main interface is divided into several sections. On the left, a tree view shows the 'mqtt.omnipresense.net' connection with a list of topics and message counts. A red arrow points to this list with the label 'Data sent by sensor'. The top bar includes a search field, a pause button, and a 'DISCONNECT' button. The right panel, titled 'Value', contains a 'History' section and a 'Publish' section. The 'Publish' section has a text input field containing 'serial_number/command', with a red arrow pointing to it labeled 'Use serial number for topic'. Below the input are radio buttons for 'raw' (selected), 'xml', and 'json'. A red arrow points to the 'raw' button labeled 'Select raw'. To the right of the radio buttons is a 'PUBLISH' button, with a red arrow pointing to it labeled 'Publish API Commands'. Below the radio buttons is a text area containing '7P', with a red arrow pointing to it labeled 'API Command'. At the bottom right, there are 'QoS 0' and 'retain' checkboxes.

Figure 11. MQTT Explorer Main Screen

Appendix

Table 2. OPS243 WiFi Module Code Version Matrix

| Feature | V1.0.1 | V2.0.x | V3.0.x | Notes |
|-----------------------------------|--------|--------|--------|---|
| Base WiFi Support | • | • | • | |
| API Connection Command Support | | • | • | Requires v1.2.4 OPS243-A or v1.2.7 OPS243-C code versions or later. |
| SoftAP Support | | | • | |
| OTA Support | | | • | |

Revision History

| Version | Date | Description |
|---------|--------------------|---|
| A | February 3, 2021 | Initial release. |
| B | January 7, 2022 | Added information for connecting to sensor with MQTT X. |
| C | February 15, 2023 | Added information for using iOS BluFi app to connect OPS243 to WiFi network. |
| D | September 29, 2025 | Added WiFi connection method using API in OPS243-A v1.2.4 and OPS243-C v1.2.7 code releases. Added updates on Android app usage. Added viewing using MQTT Explorer, removed MQTT X information. |
| E | October 14, 2025 | Updated WiFi connection sequence with use of SoftAP app Added information about new WiFi connection code v3.0.x and later Added OTA update information |