

TECH NOTE – digiBOX MQTT Application

Version: 2026-03-19

Author: Darren Siantama

Status: HBK: Public

MQTT

digiBOX



UNRESTRICTED

Table of Contents

1. Description	3
2. Hardware Setup	3
3. Broker Installation	4
4. digiBOX MQTT Configuration	4
5. Mosquitto	4
6. VS Code Extension	6
6.1. Installation	6
6.2. Subscribe	9
6.3. Read/Write Data via JSON-RPC	12

1. Description

This document will show you how to set up and use MQTT in the digiBOX system. MQTT works by allowing clients such as sensors, machines or controllers, mobile apps, dashboards, or cloud services to connect to a broker, publish messages to topics, and subscribe to topics to receive updates. This manual also provides 2 methods to work with an MQTT client. They are:

- Using an extension from VS Code
- Using a python library

Note: This guide is for Windows users.

2. Hardware Setup

Before we dive into the MQTT part, we need to set up the digiBOX to make it function properly. The digiBOX application setup that we will be using for this guide looks like this.



1. To connect the digiBOX to the sensor, we will use CON S3003 + cables.
2. For the power supply, CON A2004 + AC/DC15V/550MA are used.
3. Lastly, for the connection between digiBOX and PC, we will use a 4 pin connecting socket CON-S1023 + a RJ45 cable as it will be using an ethernet connection.

3. Broker Installation

Before everything else, we need to first have an MQTT broker. An MQTT broker is a server that receives MQTT messages from clients and delivers them to other clients that subscribe to the corresponding topics. The broker that we will be using is Eclipse Mosquitto.

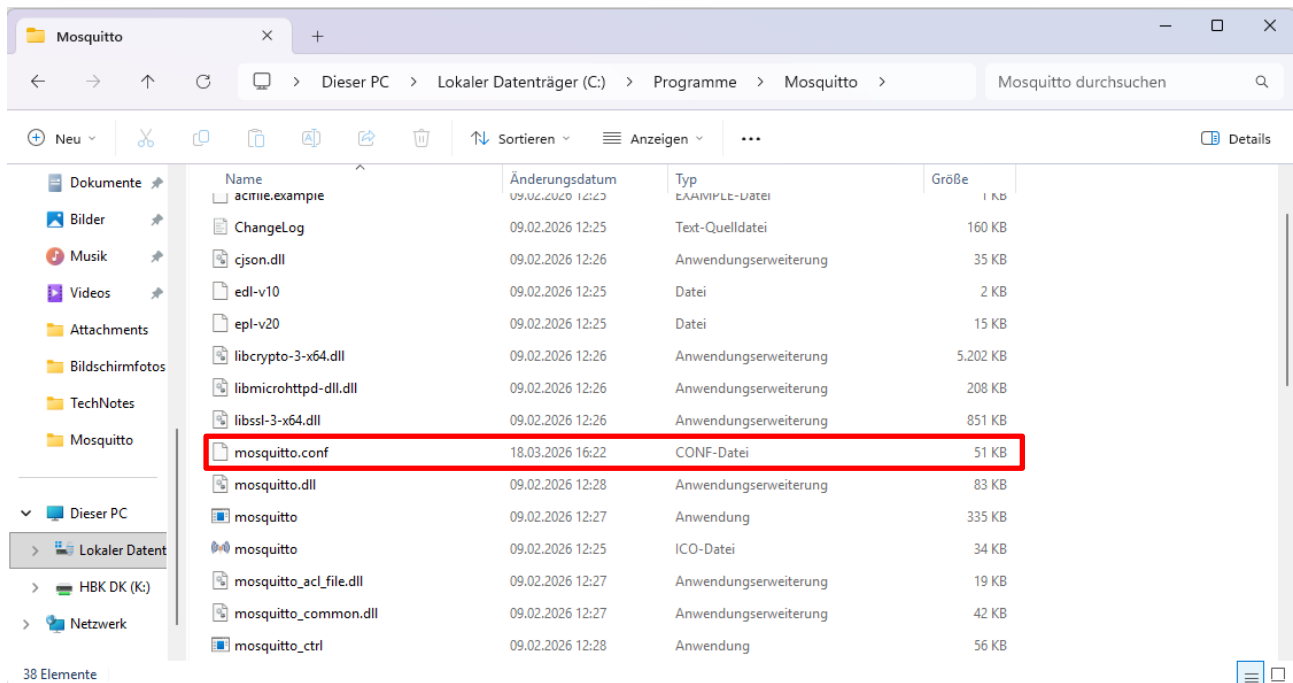
To install Mosquitto, go to the Eclipse Mosquitto website. Then, click on Download and download the .exe file for Windows. Run the file and finish the installation process.

4. digiBOX MQTT Configuration

Now, we need to configure the digiBOX so that it is ready to send and receive payloads to clients via broker. Refer to the document “digiBOX Weighing User Manual” section 9.2 MQTT for this step.

5. Mosquitto

On the file explorer, search for the folder “Mosquitto”. It is usually located in Program Files. Open the config file with Notepad++.



Now we need to configure the listener. Scroll down until you see listener port-number. Then, type listener port-number 0.0.0.0. We use 0.0.0.0 here to listen to all available and supported IP protocol.

```
230 # =====
231 # Listeners
232 # =====
233
234 # Listen on a port/ip address combination. By using this variable
235 # multiple times, mosquitto can listen on more than one port. If
236 # this variable is used and neither bind_address nor port given,
237 # then the default listener will not be started.
238 # The port number to listen on must be given. Optionally, an ip
239 # address or host name may be supplied as a second argument. In
240 # this case, mosquitto will attempt to bind the listener to that
241 # address and so restrict access to the associated network and
242 # interface. By default, mosquitto will listen on all interfaces.
243 # Note that for a websockets listener it is not possible to bind to a host
244 # name.
245 #
246 # On systems that support Unix Domain Sockets, it is also possible
247 # to create a # Unix socket rather than opening a TCP socket. In
248 # this case, the port number should be set to 0 and a unix socket
249 # path must be provided, e.g.
250 # listener 0 /tmp/mosquitto.sock
251 #
252 # listener port-number [ip address/host name/unix socket path]
253 listener 1883 0.0.0.0
```

After that, we need to set `allow_anonymous` to true. This is done to allow clients to connect without any username or password. If you want to have a protected connection, then set this to false. Read the description on the file for further explanation. For this guide, we will be creating an unprotected connection.

```
602 # =====
603 # Security
604 # =====
605
606 # If set, only clients that have a matching prefix on their
607 # clientid will be allowed to connect to the broker. By default,
608 # all clients may connect.
609 # For example, setting "secure-" here would mean a client "secure-
610 # client" could connect but another with clientid "mqtt" couldn't.
611 #clientid_prefixes
612
613 # Boolean value that determines whether clients that connect
614 # without providing a username are allowed to connect. If set to
615 # false then a password file should be created (see the
616 # password_file option) to control authenticated client access.
617 #
618 # Defaults to false, unless there are no listeners defined in the configuration
619 # file, in which case it is set to true, but connections are only allowed from
620 # the local machine.
621 allow_anonymous true
```

Save the config file and run Mosquitto. To run Mosquitto, open command prompt. Navigate to the Mosquitto folder.

```
C:\Windows\System32>cd C:\Program Files\Mosquitto  
C:\Program Files\Mosquitto>
```

Then, run this command.

```
C:\Program Files\Mosquitto>mosquitto -v -c mosquitto.conf
```

If you see these messages on the terminal, it means that you have successfully connected the digiBOX to the broker. It contains the ip address and the client id of the digiBOX on one of the messages.

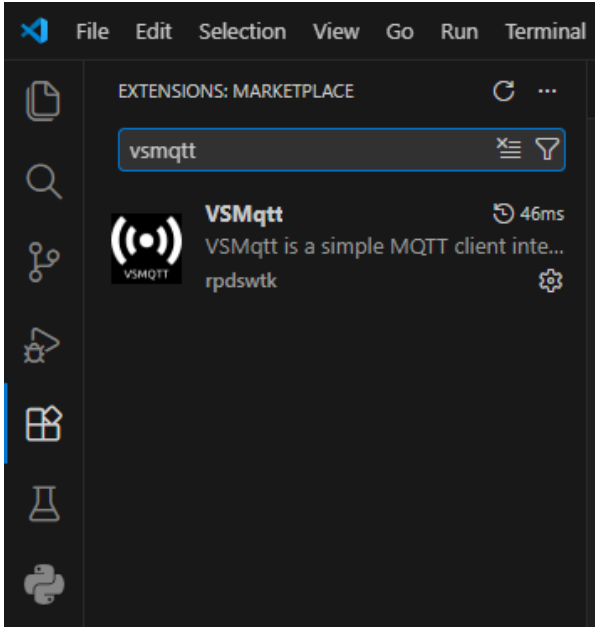
```
C:\Program Files\Mosquitto>mosquitto -v -c mosquitto.conf  
1773922041: mosquitto version 2.1.2 starting  
1773922041: Config loaded from mosquitto.conf.  
1773922041: Bridge support available.  
1773922041: Persistence support available.  
1773922041: TLS support available.  
1773922041: TLS-PSK support available.  
1773922041: Websockets support available.  
1773922041: Opening ipv4 listen socket on port 1883.  
1773922041: mosquitto version 2.1.2 running  
1773922052: New connection from 169.254.245.150:1569 on port 1883.  
1773922052: New client connected from 169.254.245.150:1569 as dgbx (p4, c1, k0).
```

Now we are ready to connect MQTT clients to the broker.

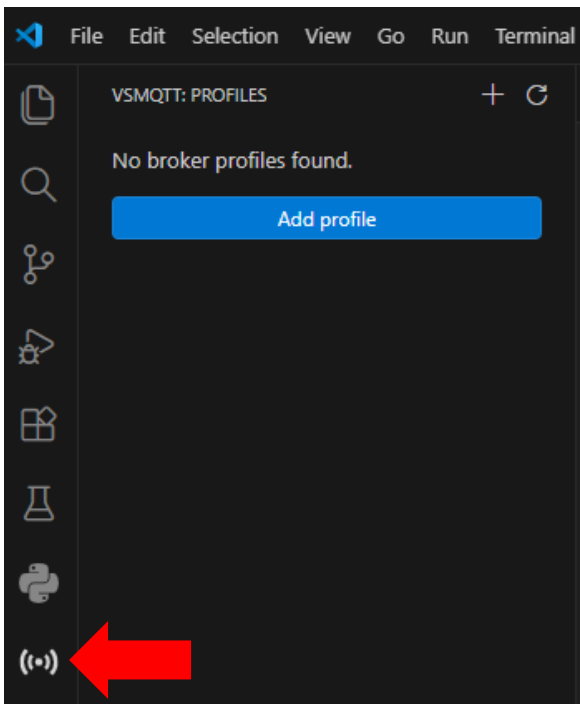
6. VS Code Extension

6.1. Installation

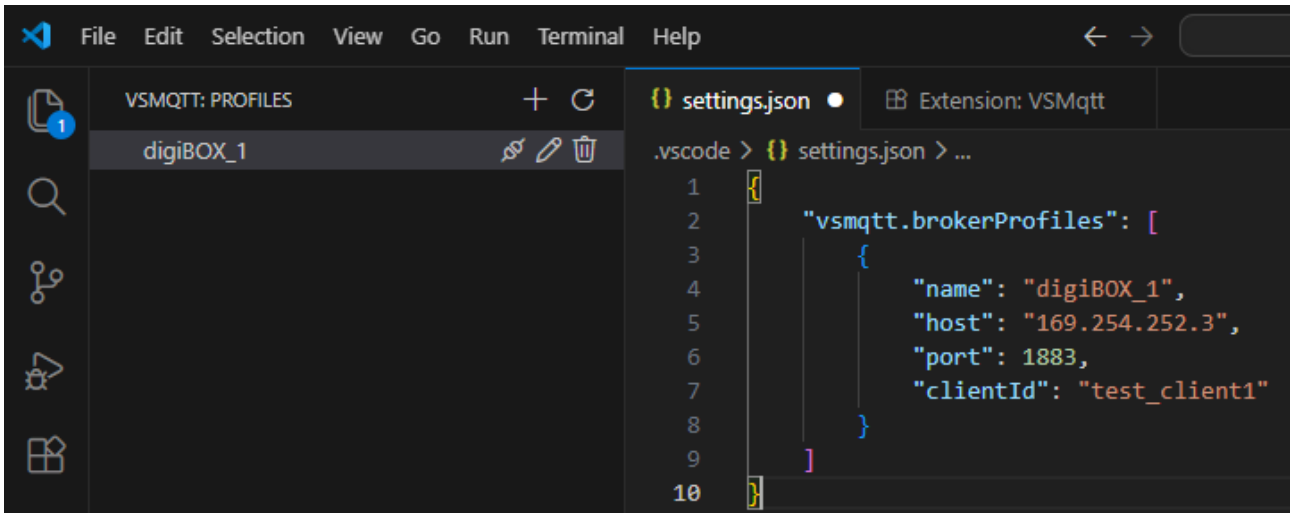
For this method, you would need to install an extension called VSMqtt from rpdswtk in VS Code. You can find the extension by typing the name.



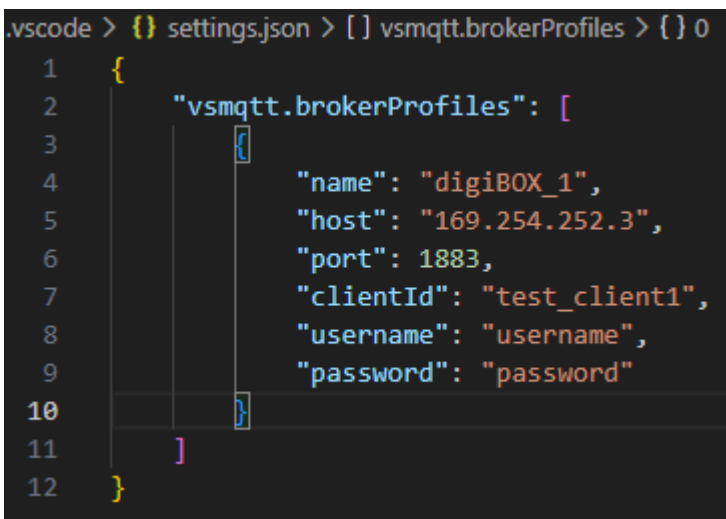
After you successfully installed it, a new icon will appear on the sidebar.



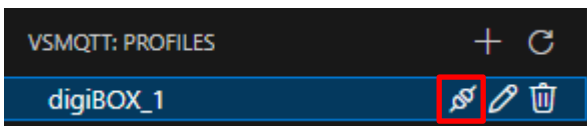
To start, first click on “Add profile” to configure the broker’s connection. Then, you would be ask to type in the broker's host, port and profile name. Then, it will generate a .json file.



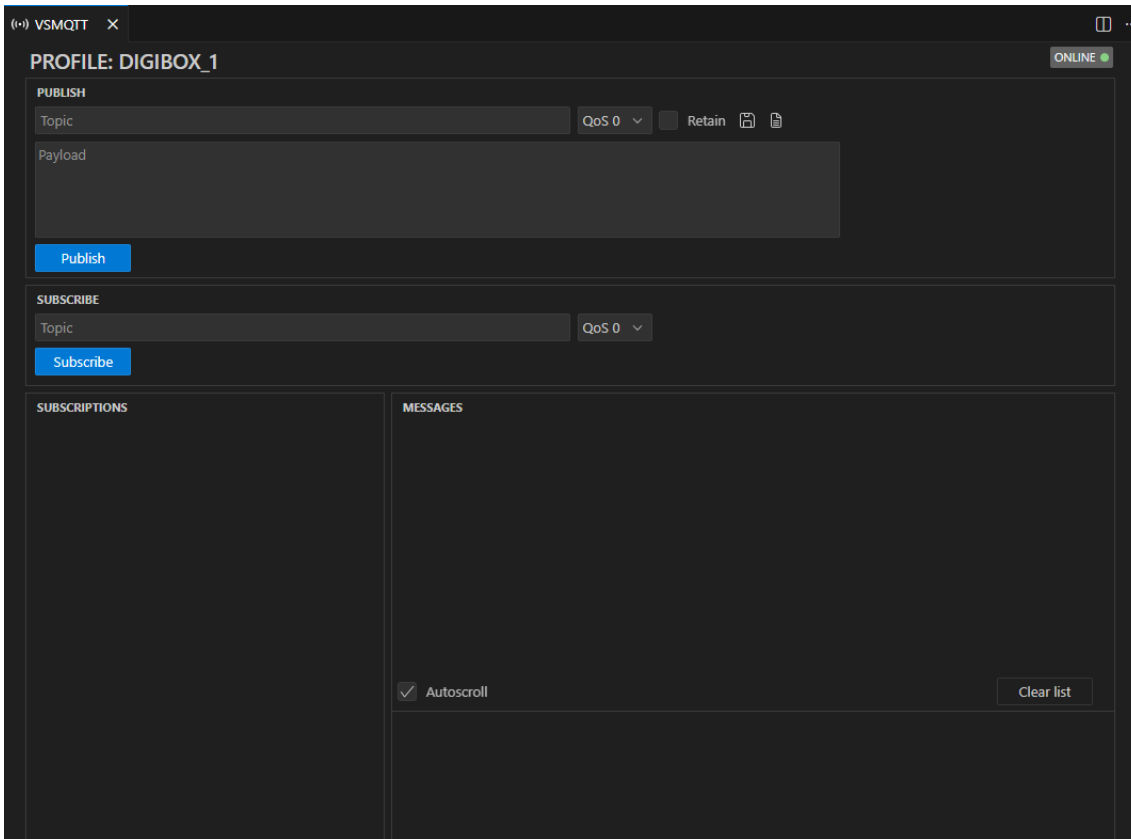
You can change the profile name, host and port on this file if needed. This is an example of a simple unprotected MQTT connection. For a simple protected connection, you can add username and password on this .json file. Make sure this matches the username and password on both the digiBOX web interface and the password file in the Mosquitto folder.



Now click on this button to connect to the broker.

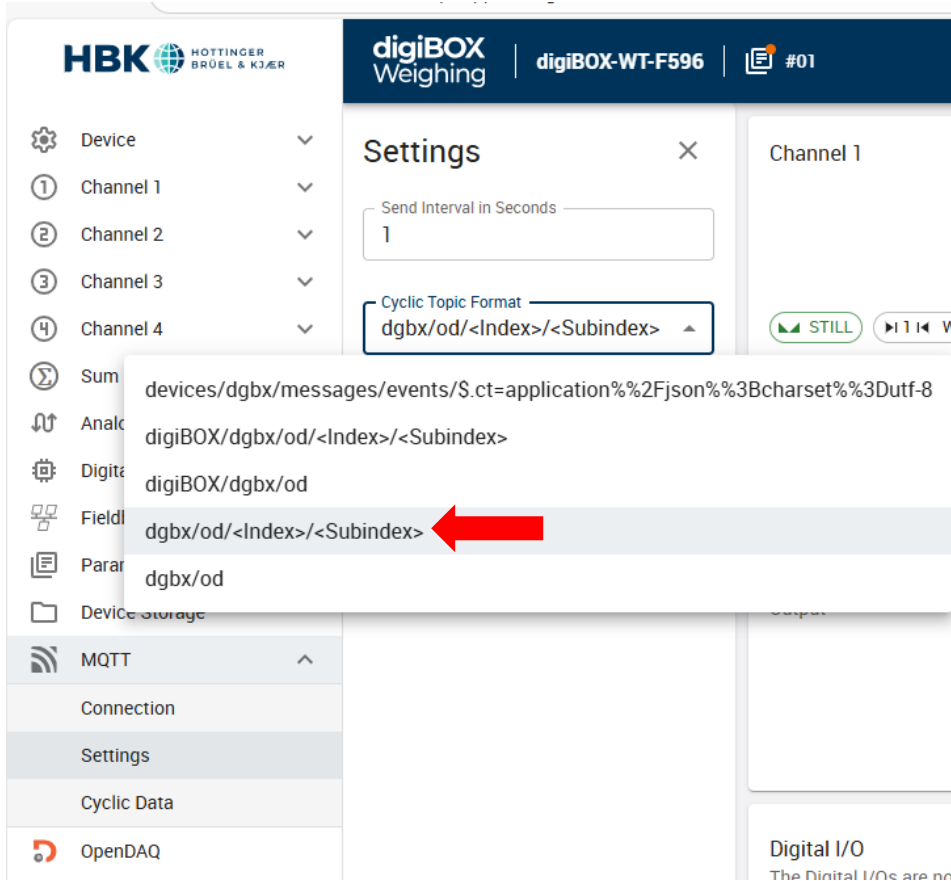


A new window will open and you can see that the status is online on the top right corner of the interface.

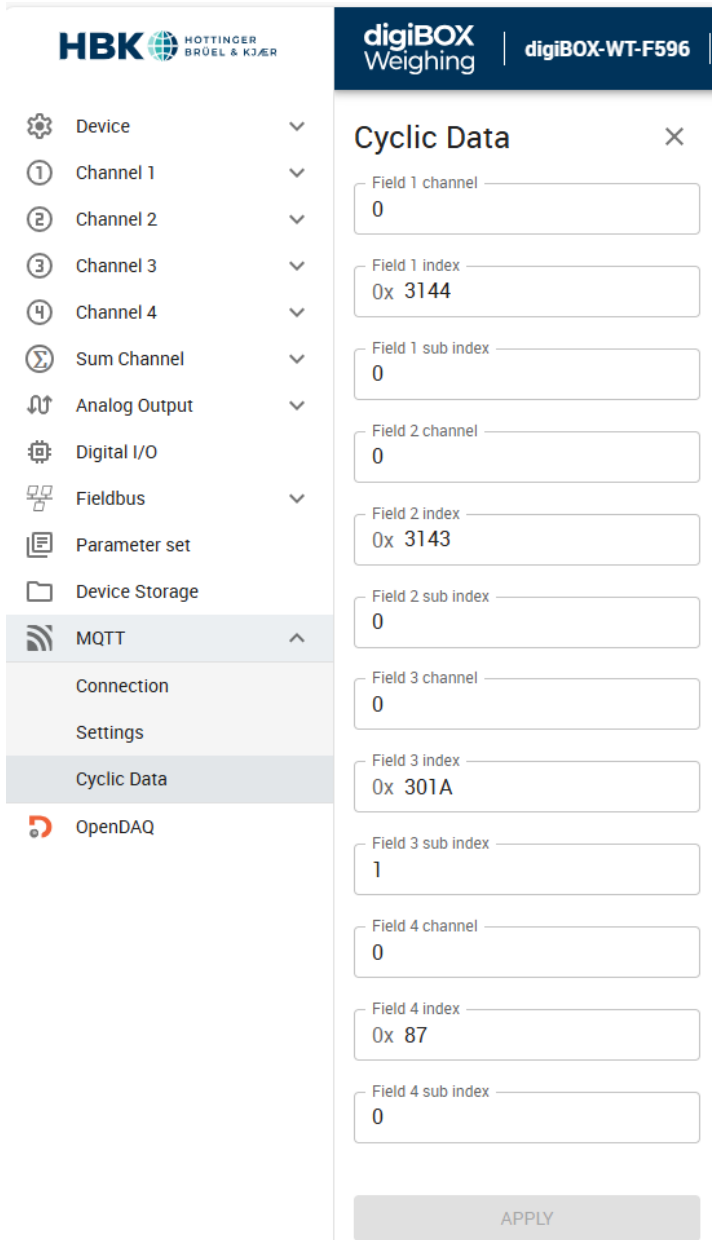


6.2. Subscribe

To subscribe to a topic, there is a specific format provided. You can find the topic format on the digiBOX web interface. There are some options available for the topic format. The most common usage is the format used here. You can also set the send interval in seconds.

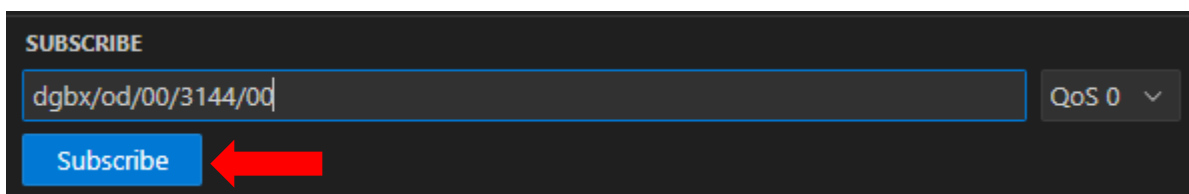


Information about the index and subindex of the payload can be found on the folder device storage. Just a reminder that the indexes and subindexes are all in the form of hexadecimal. To set which value you want to get from digiBOX, go to the tab Cyclic Data. You can then input the index, subindex and the channel number of the payload that you need.

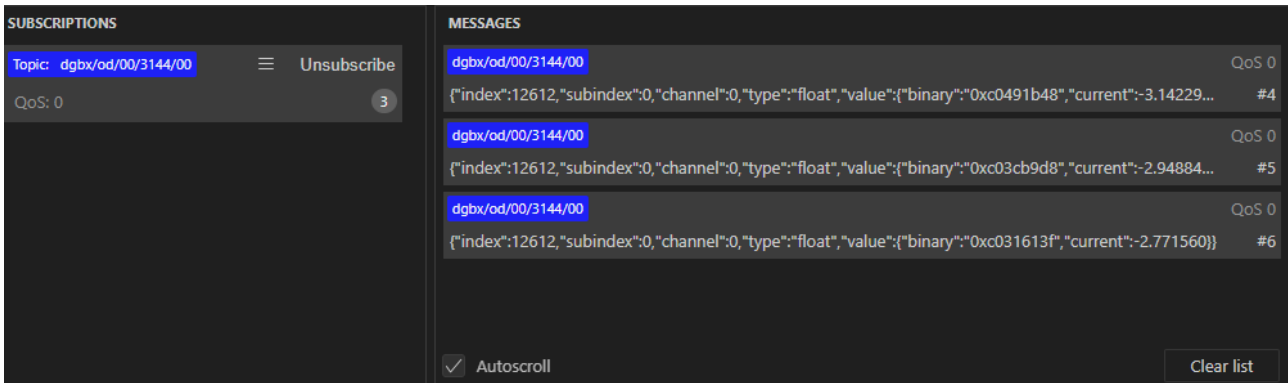


As an example here, the value that I want to get is the weight from channel 1 and it is located at channel number 0, index 0x3144 and subindex 0.

On the VSMQTT tab in VS Code, you can see the subscribe part. There you can input the topic based on the format. Then, click on subscribe and you will see the message from that topic.



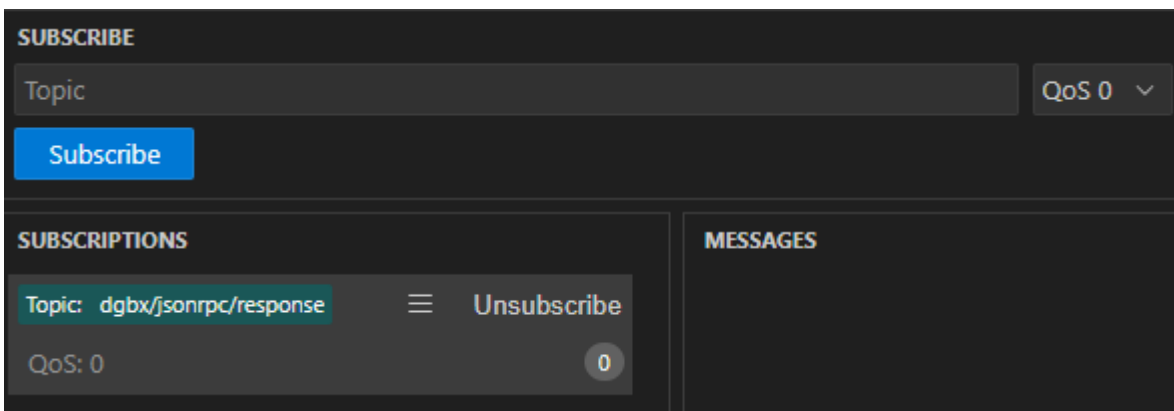
UNRESTRICTED



6.3. Read/Write Data via JSON-RPC

Refer to the document “digiBOX weighing user manual” section 9.2.3 On-demand data access for explanation.

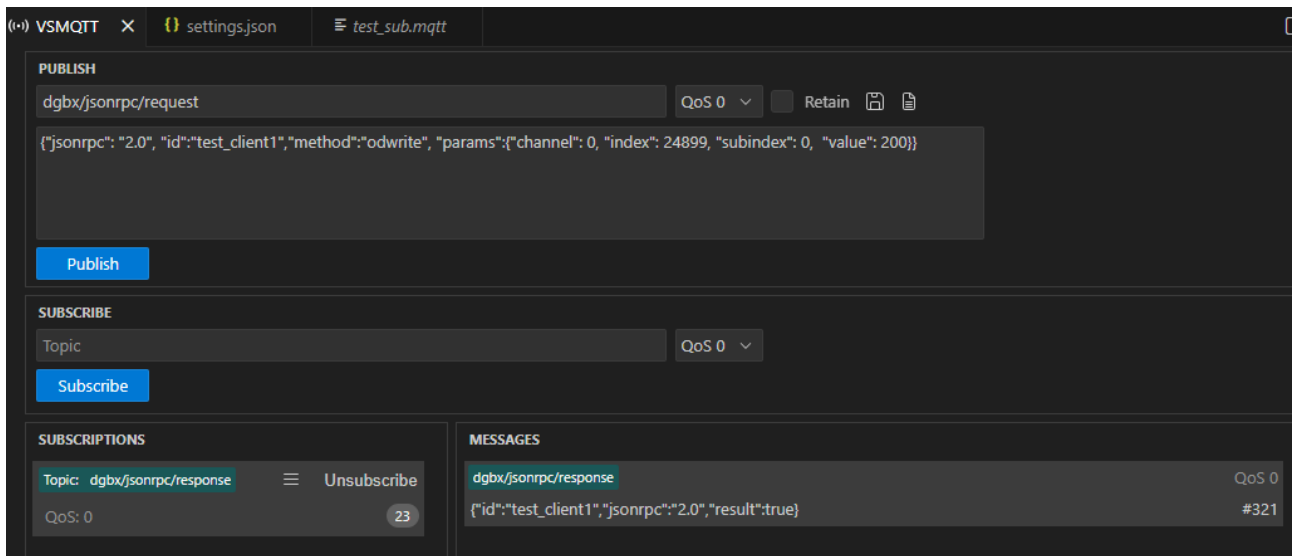
To start, subscribe to the response topic format in VS Code.



Then, send a request to digiBOX by using the request topic format. For example, I wanted to change the tare value. So, the payload will then be:

```
{ "jsonrpc": "2.0", "id": "df", "method": "odwrite", "params": { "channel": 0, "index": 24899, "subindex": 0, "value": 200 }
```

If successful, you will see the following message.



On the digiBOX web interface, you will also see the changes.



Note: always check that the value type corresponds to the type stated on the digiBOX object dictionary.

Disclaimer

These examples are for illustrative purposes only. They cannot be used as the basis for any warranty or liability claims.