



TEKTELIC Communications Inc.  
7657 10th Street NE Calgary, Alberta  
Canada, T2E 8X2

---

# LoRa MQTT Command Line Sensor Configuration

## Instruction Manual

---

<b>Document Type:</b>	Instruction Manual
<b>Document Version:</b>	0.1
<b>Document Status:</b>	Release
<b>Product Name:</b>	TEKTELIC LoRa Sensors
<b>Release Date:</b>	April 25, 2022

---

**PROPRIETARY:**

The information contained in this document is the property of TEKTELIC Communications Inc. Except as specifically authorized in writing by TEKTELIC, the holder of this document shall keep all information contained herein confidential, and shall protect the same in whole or in part from disclosure to all third parties.

© 2022 TEKTELIC Communications Inc., all rights reserved.

All products, names, and services are trademarks and registered trademarks of their respective companies.

**DISCLAIMER:**

Material contained in this document is subject to change without notice. The material herein is solely for information purposes and does not represent a commitment by TEKTELIC or its representatives. TEKTELIC has prepared the information contained in this document solely for use by its employees, agents, and customers. Dissemination of this information and/or concepts to other parties is prohibited without the prior written consent of TEKTELIC. In no event will TEKTELIC be liable for any incidental or consequential damage in connection with the furnishing, performance or use of this material.

TEKTELIC reserves the right to revise this publication in accordance with formal change control procedures defined by TEKTELIC.

## Revision History

Version	Date	Editor	Comments
0.1	April 25, 2022	Carter Mudryk	First draft partially based on Systems Designer Introduction v0.4.

## Table of Contents

Revision History.....	2
Table of Contents .....	3
List of Figures.....	4
List of Acronyms .....	5
1 Overview .....	6
2 Background .....	7
2.1 Information.....	7
2.2 Installing the MQTT Broker .....	7
3 Sending a Single DL .....	9
3.1 Procedure .....	9
3.2 Example .....	10
4 Bulk Sensor Configuration .....	11
4.1 Identical Payloads to Multiple End-Devices .....	11
4.2 Different Payloads to Multiple End-Devices.....	12

## List of Figures

Figure 1: Ensuring the Mosquitto Variable has Been added to the Environment Variables .....	8
Figure 2: Adding Mosquitto to the Path .....	8

## List of Acronyms

<b>DevEUI</b> .....	Device Extended Unique Identifier
<b>DL</b> .....	DownLink
<b>LoRa</b> ..	Long-Range
<b>LoRaWAN</b> .....	LoRa Wide Area Network
<b>MQTT</b> .....	Message Queuing Telemetry Transport
<b>NS</b> .....	Network Server
<b>OS</b> .....	Operating System
<b>OTA</b> .....	Over-The-Air
<b>Reg</b> .....	Register
<b>RH</b> .....	Relative Humidity
<b>SRS</b> .....	Smart Room Sensor
<b>TRM</b> ...	Technical Reference Manual
<b>Tx</b> .....	Transmit / Transmitter
<b>UL</b> .....	UpLink
<b>v</b> .....	version
<b>ver.</b> .....	version

## 1 Overview

This document provides instructions for configuring LoRaWAN end-devices manually via the command line. This document assumes a knowledge of the TEKTELIC Network Server and its interfaces, as well as the procedure for provisioning devices. Details about commissioning information, gateway setup, and NS settings are outside the scope of this document. For help with these other topics, please see the online support Knowledge Base for related support articles.

Sending downlinks (DLs) to end-devices can be done in several ways. The most straightforward way of doing so is usually through the NS *Downlink Queue* tab or through an integrated application, one device at a time. However, in the cases where there is no encoding application or there are many devices that require reconfiguration all at once, this can be cumbersome. By creating a simple script, this task can be semi-automated and made much easier. This document provides the instructions for creating and editing such a script.

## 2 Background

### 2.1 Information

Uplinks (ULs) are data communication streams from the end-device to the NS through the gateway. ULs can contain sensor data, acknowledgement messages, join requests, etc.

Downlinks (DLs) are the opposite; data communication streams from the NS to the end-device through the gateway. DLs can contain configurations and commands that tell the end-device what to do.

Both ULs and DLs are sent via LoRa and are encoded. The messages are sent and handled using an MQTT broker protocol.

Sending DLs can be done through the command line using the MQTT protocol, as described in the following sections.

### 2.2 Installing the MQTT Broker

To use your computer to send DLs via the command line, the MQTT broker must be installed. The following are instructions for doing this on Windows OS.

1. Go to <https://mosquitto.org/download/> to download the appropriate Mosquitto broker.
2. Open the .exe file and install it to your C: drive.
3. Ensure that MOSQUITTO\_DIR has been added to your computer's environment variables by going to Control Panel > System Properties > Advanced > Environment Variables (See Figure 1).

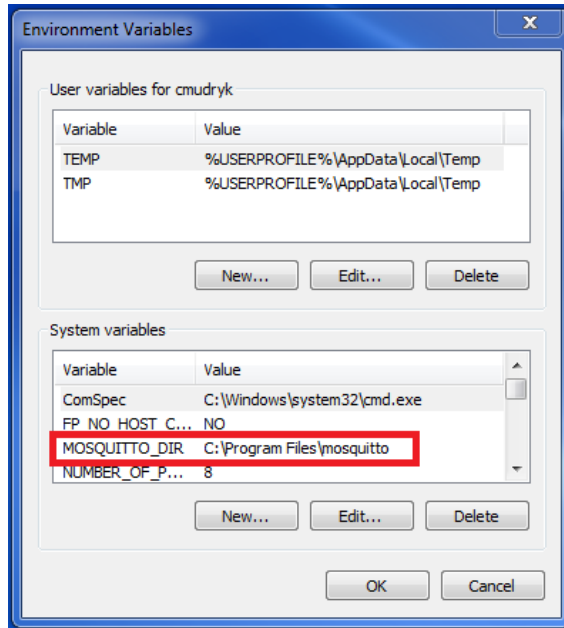


Figure 1: Ensuring the Mosquitto Variable has Been added to the Environment Variables

- Now select *Path* in the *System variables* list and click *Edit*. You must do this as a system administrator. Add “C:\Program Files\mosquitto\” to the end of the string in the *Variable value* field. See Figure 2. Now you should be able to use Command Prompt to send DLs to the device.

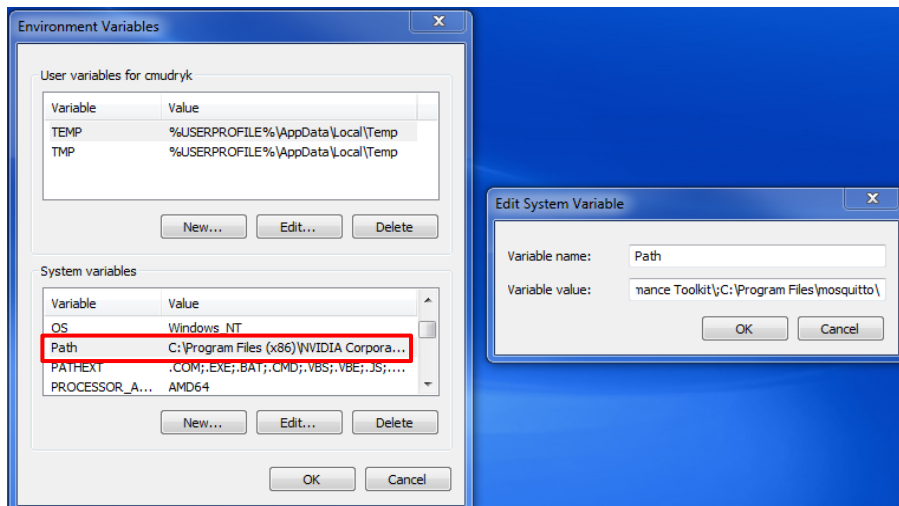


Figure 2: Adding Mosquitto to the Path



## 3 Sending a Single DL

### 3.1 Procedure

To use MQTT to send a single DL to a single device, follow these steps:

1. Provision your device in the NS then let it join the network and begin normal operation. Keep the NS open in your browser.
2. Open Command Prompt as an administrator (hold down ctrl+shift while you click to open).
3. A DL command is structured as follows:

```
mosquitto_pub -h "[Host Name]" -t "app/tx" -u "[App Username]" -P "[App Password]" -d -m "{ \"msgId\": 1, \"devEUI\": \"[DevEUI]\", \"port\": [Port], \"confirmed\": false, \"data\": \"[Base64 Data]\"}"
```

Copy and paste this into the command line in command prompt and replace the field placeholders, including their square brackets, with the following information.

- [Host Name] -> the TEKTELIC NS address. For example, if you are using the NA server, this would be lorawan-ns-na.tektelic.com.
- [App Username] and [App Password] -> The credentials for your NS application that contains the device you are trying to communicate with. To find this information, click on *Applications* in the NS sidebar, select your application, then click the *Manage Credentials* button. These credentials can be left as default or changed to something user-specified.
- [DevEUI] -> The device's EUI, which is the same as when you added it to your Application. To find this number again, click *Manage Devices*, select your device, and your Device EUI should be listed.
- [Port] -> Depending on your device and what you want to tell it to do in the DL, the port is a number that tells the device which virtual docking point the information in the DL is supposed to flow through. The TRM for your device will tell you which port to specify for your purposes.
- [Base64 Data] -> This is the payload of your DL. This data contains the relevant information in the DL.

Hit enter in command prompt once this command is ready to go.

4. Click *Manage Devices* and select the device you're sending the downlink to. Click on the *Downlink Queue* tab and click *Update Downlink Queue*. The DL you just sent should

appear here, waiting to be transmitted to the device at the next scheduled communication interval or UL. Note that if it doesn't appear here, it may have already transmitted to the device.

5. The *Real-Time Packets* tab contains a list of the ULs and DLs transferred from and to your device. You should see your DL appear here when it has transmitted.

## 3.2 Example

Here is an example of encoding a DL to transmit to the Smart Room Sensor (SRS). The SRS, by default, transmits ULs to report the battery voltage, ambient temperature, and relative humidity every hour, because the default number of seconds per core *tick* is set to 3600. Every tick, these values are reported. Let's say you wanted to tell the sensor to report the humidity only after every two ticks. First, open the SRS TRM to *Section 3 – DL Payload Formats*. The tables in this section will tell you which data register correspond to what transducers/settings on the sensor. *Table 3-5* gives the Periodic Transmission Configuration Registers. The ticks per Ambient RH register address is 0x23, and contains 2 bytes. As explained in the TRM, in order to write to register 0x23, the write bit must be set to 1. A short rule-of-thumb for this is to add 0x8 (in hexadecimal) to the left-most digit of the register address. Doing this results in A3 (0x2 + 0x8 = 0xA). The next two bytes are the data, and since we want it to transmit every two ticks, we just enter 0x00 02. The full payload is then:

0xA3 00 02

Translated into base 64, this is:

owAC

The port to configure the home sensor is port 100 (as stated in the TRM). The command prompt command would then be (using an example application and device's credentials):

```
mosquitto_pub -h "lorawan-na.tektelic.com" -t "app/tx" -u "username_123" -  
P "password_456" -d -m "{\"msgId\":1,\"devEUI\":\"647FDA0000001234\",  
\"port\":100,\"confirmed\":false,\"data\":\"owAC\"}"
```

When this DL is sent to the device, it will then send RH readings every two hours.

## 4 Bulk Sensor Configuration

A simple script can be created to automate the sending of DLs to any number of devices. The following sections give the instructions for two cases:

1. Sending the same payload(s) to multiple end-devices.
2. Sending different payloads to multiple end-devices.

### 4.1 Identical Payloads to Multiple End-Devices

1. Open a new .txt file using Notepad or some other program.
2. Copy the following code into the file.

```
for %%G in ([DevEUI1],[DevEUI2],[DevEUI3]) do (
  echo %%G
  mosquito_pub -h "[Host Name]" -t "app/tx" -u "[App Username]" -P "[App
  Password]" -d -m "{ \"msgId\": 1, \"devEUI\": \"%%G\", \"port\": [Port],
  \"confirmed\": false, \"data\": \"[Base64 Data]\"}"
  TIMEOUT /T 1 /NOBREAK
)
```

3. Replace the field placeholders (including the square brackets) with the appropriate real values as described in Section 3.1 Step 3.

#### NOTES:

- Here, the DevEUIs are specified in the first line rather than in the `mosquitto_pub` command block. In this example, only 3 DevEUIs are specified, meaning that this particular DL command will be sent to those 3 devices. Any arbitrary number of devices can be included by adding more DevEUIs to the comma-separated list in the first line.
  - Using this structure, all devices must be provisioned within the same NS application grouping.
4. Save this script as a .bat (batch) file.
  5. To run the script, double click on the newly created batch file. A command window will open and the command will be run. You can verify that the DLs were sent by viewing the devices' Downlink Queue in the NS.

This script format can also be modified to send more than 1 unique payload to each of the devices listed in the first line. This can be done simply by copying the command block as many times is needed for each of the different Base64 Data payloads required, as shown below.

```
for %%G in <[DevEUI1],[DevEUI2],[DevEUI3]> do <
  echo %%G

  mosquitto_pub -h "[Host Name]" -t "app/tx" -u "[App Username]" -P "[App Password]" -d -m "{\"msgId\":1,\"devEUI\": \"%%G\", \"port\": [Port], \"confirmed\": false, \"data\": \"[Base64 Data1]\"}"

  mosquitto_pub -h "[Host Name]" -t "app/tx" -u "[App Username]" -P "[App Password]" -d -m "{\"msgId\":1,\"devEUI\": \"%%G\", \"port\": [Port], \"confirmed\": false, \"data\": \"[Base64 Data2]\"}"

  mosquitto_pub -h "[Host Name]" -t "app/tx" -u "[App Username]" -P "[App Password]" -d -m "{\"msgId\":1,\"devEUI\": \"%%G\", \"port\": [Port], \"confirmed\": false, \"data\": \"[Base64 Data3]\"}"

TIMEOUT /T 1 /NOBREAK
>
```

## 4.2 Different Payloads to Multiple End-Devices

If it is required to send different payloads to different devices, this can still be simplified through the use of scripts. There are 2 ways of doing this:

1. By using the method described in Section 4.1 and saving a different script for each unique group of devices.
2. By putting the individual command blocks in one script. An example is as shown below.

```
mosquitto_pub -h "[Host Name]" -t "app/tx" -u "[App Username1]" -P "[App Password1]" -d -m "{\"msgId\":1,\"devEUI\":\"[DevEUI1]\", \"port\": [Port1], \"confirmed\": false, \"data\": \"[Base64 Data1]\"}"
```

```
mosquitto_pub -h "[Host Name]" -t "app/tx" -u "[App Username2]" -P "[App Password2]" -d -m "{\"msgId\":1,\"devEUI\":\"[DevEUI2]\", \"port\": [Port2], \"confirmed\": false, \"data\": \"[Base64 Data2]\"}"
```

```
mosquitto_pub -h "[Host Name]" -t "app/tx" -u "[App Username3]" -P "[App Password3]" -d -m "{\"msgId\":1,\"devEUI\":\"[DevEUI3]\", \"port\": [Port3], \"confirmed\": false, \"data\": \"[Base64 Data3]\"}"
```

**NOTE:** Using this method allows for a single script to send different to multiple different end-devices, even if they are in different NS applications. To send to multiple applications, ensure that each command block has the appropriate App Username and App Password.