NBIOT PROTOCOL
DEVELOPER GUIDE

Version 1.1
Date 02.08.2017

LIFE IS FOR SHARING.

TABLE OF CONTENT

FOREWORD

This document describes the technical access to the NBIoT-Connector and the use of the NBIoT-Protocol MQTT-SN.

The user equipment is able to send measurements, alarms, and events via MQTT-SN messages. The NBIoT-Connector processes the incoming MQTT-SN messages through the MQTT-SN-Gateway and passes them to the Cloud of Things (CoT).

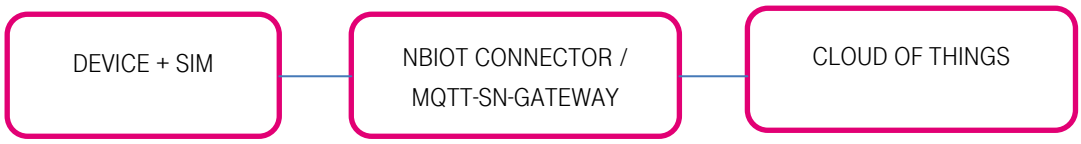

FIGURE 1 SIMPLIFIED VIEW OF NBIOT ARCHITECTURE

List of related links:

- The MQTT-SN specification is available under the link http://mqtt.org/new/wp-content/uploads/2009/06/MQTT-SN_spec_v1.2.pdf.
- Cloud of Things – User Guide: https://www.cumulocity.com/guides/users-guide/overview/

1. DEVICE MANAGEMENT

To gain access to NBIOT via MQTT-SN the following materials and information are provided and required:

- SIM card with unique IMSI (International Mobile Subscriber Identity)
- Associated password to each IMSI number
- Cloud Credentials for web interface

The delivered SIM cards are preregistered with the Cloud and are ready to use for a MQTT-SN connection.

1.1. OVERVIEW OF DEVICES

The overview of preregistered and registered device can be shown in the web interface of the Cloud via the menu entry "Devices" → "All devices" → NBIoT Device <IMSI>.

Preregistered SIM cards are in status "pending". The registration is finished right after the devices first successfully connected via MQTT-SN.

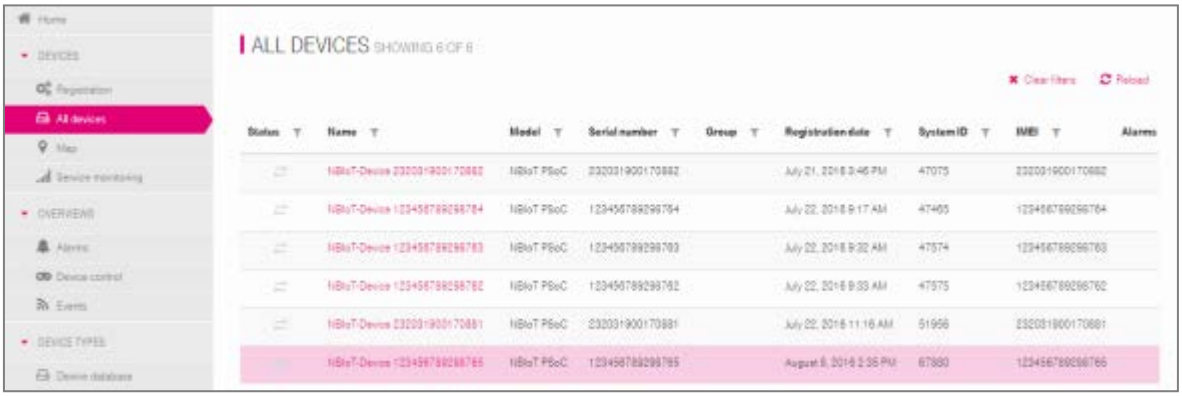

FIGURE 2: ALL DEVICES

1.2. CONNECTING TO MQTT-SN GATEWAY

The combination of IMSI and authentication key (password) results in ClientID. The ClientID is used for the authentication of the devices to the MQTT-SN gateway (NBIoT Connector).

In order to ensure the uniqueness of a device, a key for authentication is provided to each device. The authentication key can be placed at the disposal.

The MQTT-SN "CONNECT" is used for the authentication.[1] The "CONNECT" Message contains the ClientID. The ClientID is a 23 character long string which uniquely identifies the client to the server. The characters of the ClientID consist of the IMSI

[1] http://mqtt.org/new/wp-content/uploads/2009/06/MQTT-SN_spec_v1.2.pdf – clause 5.4.4 "Connect"

and an assigned password. The IMSI consists of 15 characters. The remaining 8 characters of the ClientID are used for the password.

An example of the combination of the ClientID is:

- IMSI: 123456789012345
- Password: SRUkRd5H
- **ClientID: 123456789012345SRUkRd5H**

1.3. IDENTIFICATION OF IMSI

AT commands are instructions used to control a modem. The "AT+CIMI" command queries the IMSI, a 15-digit unique identifier of the SIM card. Please have a look at your modem manual to assert your console interface. User of the Prototyping Hub are able to use the console interface of PSoC2 (Programmable System on Chip). The operation response is visualized in the figure below.

FIGURE 3 IDENTIFICATION OF IMSI

1.4. DELETING DEVICES

The Cloud provides the user a deleting process of the devices. **This feature is not functional for NBIoT yet and shall not be used in the Cloud-GUI.** This feature will be released soon. Please inform your person of contact at Deutsche Telekom if you need to delete devices.

1.5. CHANGE PASSWORD

The Cloud provides the user the password changing function. **This feature is not functional for NBIoT yet and shall not be used in the Cloud-GUI.** This feature will be released soon. Please inform your person of contact at Deutsche Telekom if you need support.

[2] https://de.wikipedia.org/wiki/PSoC

2. CLOUD OF THINGS

In this chapter the different types of Cloud of Things specific messages and their content is described.

2.1. VIEWING DATA VIA CLOUD OF THINGS-GUI

In the following chapters the relevant views of the Cloud of Things-GUI are explained.

2.1.1. DEVICE DATA VIEW

All transmitted device data can be viewed via Cloud Interface (via the menu entry “Devices” → “All devices” → NBIoT Device + IMSI → Measurements). The “Figure 2-9: Measurement” depicts the temperature profile of the “NBIOT-DEVICE 123456789298765”.

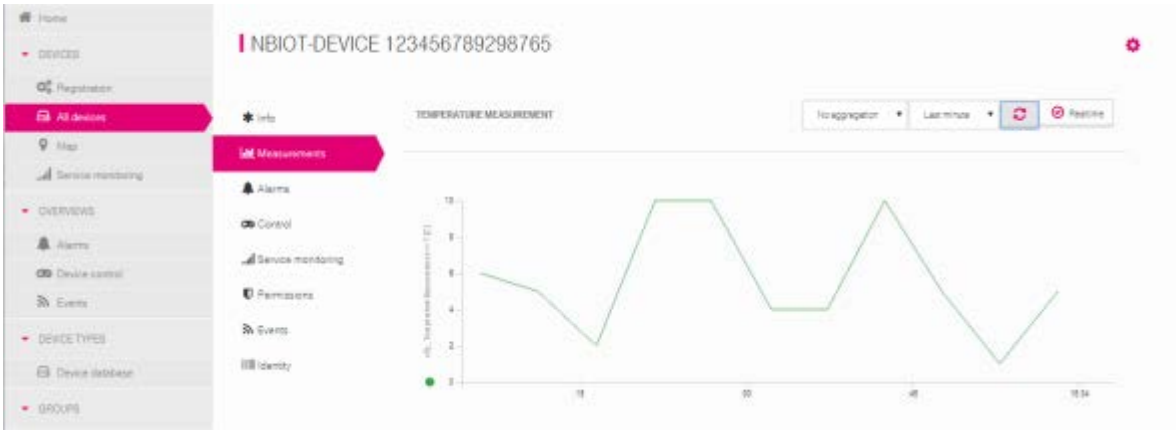

FIGURE 4 MEASUREMENT GUI

2.1.2. ALARM VIEW

Via the menu entry “Devices” → “All devices” → NBIoT Device + IMSI → Alarms, the user can view the generated alarms of the device. The Figure 2-10: Alarm message shows the example alarm.

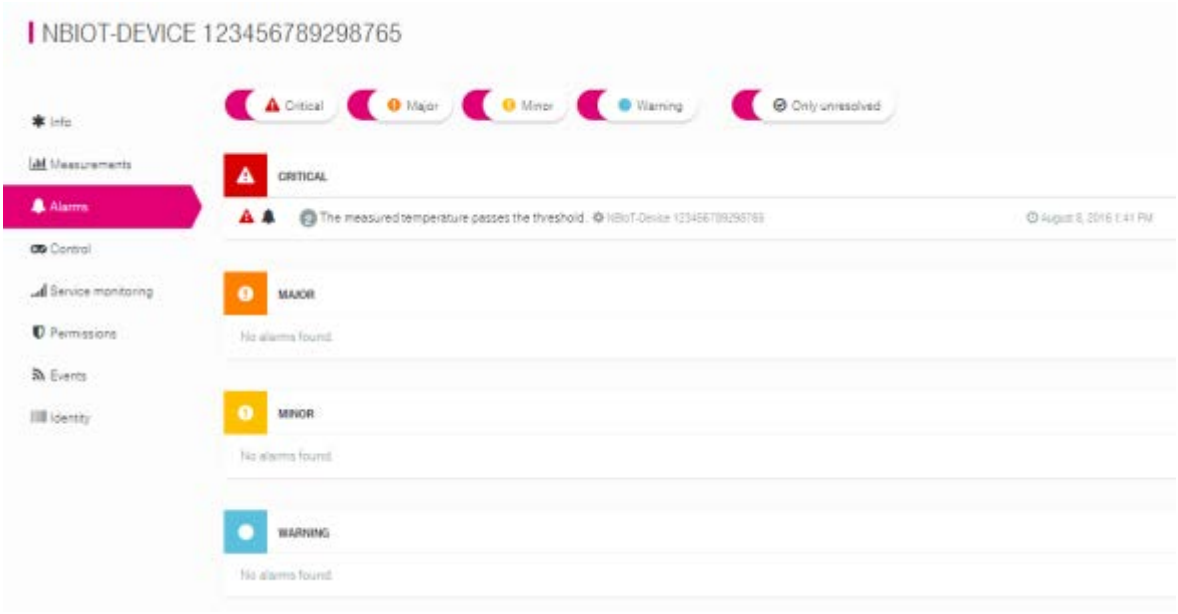

FIGURE 5 ALARM MESSAGE GUI

2.1.3. EVENT VIEW

Via the menu entry “Cockpit” → “Home” → “MAP”, the user can view the transmitted position of the device. The following Figure: Cloud of Things Map - Position shows the position of the devices.

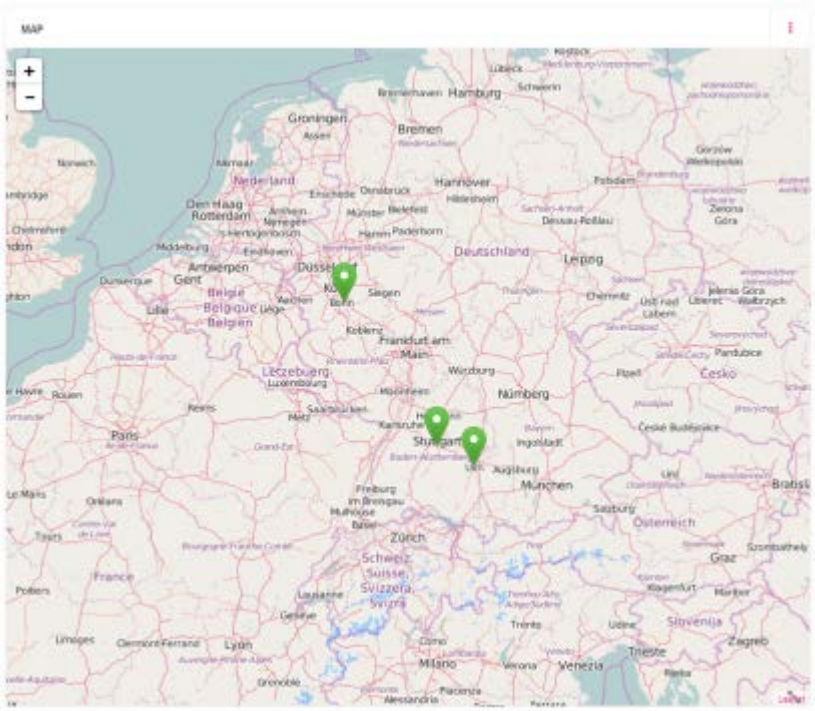

FIGURE 6 CLOUD OF THINGS MAP – POSITION

3. MQTT-SN MESSAGES

3.1. STRUCTURE OF MQTT-SN MESSAGE CONTENT

The general MQTT-SN Specification can be downloaded under the following link: http://mqtt.org/new/wp-content/uploads/2009/06/MQTT-SN_spec_v1.2.pdf

The NBIoT connector requires a predefined MQTT-SN message content structure for processing the data and connection to Cloud of Things. The MQTT-SN message consists of a topic and the payload.

3.1.1. MQTT-SN TOPIC LEVELS

The MQTT-SN topic has a hierarchy of 4 topic levels:

Topic	Definition
NBIoT	NBIoT root node
IMSI	unique device ID (HexString)
MessageType	MES(Measurement), ALM(Alarm), EVT (Event)
PayloadID	ID for the payload contents (HexString)

FIGURE 7 MQTT-SN TOPIC LEVELS

The general structure of a MQTT-SN topic is:

NBIoT/<IMSI>/< MessageType >/<PayloadID>

Example: NBIoT/123456789298765/MES/471A

- “NBIoT” is the specified root node.
- “123456789298765” is the unique IMSI number.
- “MES” defines the MessageType “measurement”.
- “471A”. The PayloadID defines the MQTT-SN payload structure.

3.1.2. MQTT-SN PAYLOAD

The use and the compilation of the MQTT-SN payload depend on the used MQTT-SN topic. The payload consists of a binary byte array. The supported data types are described in Table 2-2: Supported Data Types. This table contains the properties of the data type, the size and a sample.

Data Type	Size in Byte	Example
boolean	1	true (non-zero value) or false (zero value)
int8	1	-128 to 127
uint8	1	0 to 255

short *(big endian byte order)*	2	-32768 to 32767
int32 *(big endian byte order)*	4	−2.147.483.648 to 2.147.483.647
long *(big endian byte order)*	8	−9.223.372.036.854.775.808 to 9.223.372.036.854.775.807
float *(big endian byte order)*	4	1,2E-38 to 3,4E+38
double *(big endian byte order)*	8	2,3E-308 to 1,7E+308
date4 *(big endian byte order)*	4	1472543729 = Unix epoch time (seconds)
date8 *(big endian byte order)*	8	1472543729000 = Unix epoch time (milliseconds)
Char	1	A
string	N	ADCDEF\0 = Zero terminated character array

FIGURE 8 SUPPORTED DATA TYPES

3.2. SEND MEASUREMENT

The chapter “Send Measurement” describes the structure of MQTT-SN message and the supported measurements. The supported measurements are divided into simple and complex structures.

3.2.1. STRUCTURE OF MQTT-SN MESSAGE

The general structure of MQTT-SN message is described in the previous chapter. The structure of the measurement message uses the value “MES” as event type. The topic “MES” is used for all measurement messages. All available measurement topics are described in Table 2-3: Measurement Topic Levels.

The measurement ID corresponds to a predefined set of measurement values. The correspondence between measurement ID and measurement values is defined in the Table 2-4: Simple Measurement.

Topic	Definition
NBIoT	NBIoT root node
IMSI	unique device ID (HexString)
MES	Measurement
PayloadID	ID for the payload contents (HexString)

FIGURE 9 MEASUREMENT TOPIC LEVELS

An example for a general measurement structure with the following topic:

NBIoT/123456789298765/MES/1

The size and the use of the payload depend on the value of the PayloadID. In this example, the PayloadID is 1 and corresponds to measurement value “Temperature”. The payload of the “Temperature” measurement type uses a string data type.

Examples:

- 12 Degrees: “12” (0x31, 0x32)
- -12.5 Degrees: “-12.5” (0x2D, 0x31, 0x32, 0x2E, 0x35)

3.2.2. SIMPLE MEASUREMENT

A simple measurement message sends exactly one measurement value. In the Table 2-4: Simple Measurement, all available measurement types are listed. This table contains the mapping between measurement ID and measurement type, the data type and the measurement unit. The supported measurement types are expandable at any time.

Measurement Type	Measurement ID[3]	Data Type[4]	Measurement Unit
Temperature	1	String[4]	C
Voltage	2	String[4]	mV
Acceleration	3	String[4]	m/s2
Light	4	String[4]	Lux
Humidity	5	String[4]	%RH
Moisture	6	String[4]	%
Distance	7	String[4]	Mm
Current	8	String[4]	A
SignalStrength	9	String[4]	dBm
Pressure	A	String[4]	Db
Volume	B	String[4]	Sone
Weight	C	String[4]	G
Frequency	D	String[4]	Hz

FIGURE 10 SIMPLE MEASUREMENT

3.2.3. COMPLEX MEASUREMENT

In complex measurement the user has the possibility to send more than one measurement value in one MQTT-SN message. In the following chapters, all available complex measurement types are listed.

3.2.3.1. MOTION

PayloadID: 186A1

[3] hexadecimal representation
[4] String without terminating zero

Value	Data Type	Unit	Description
Motion Detected	boolean		Boolean value indicating if motion has been detected (non-zero value) or not (zero value)
Speed	float	km/h	Measured speed towards (+ve) or away (-ve) from the sensor.

FIGURE 11 MOTION DATA TYPE

3.2.3.2. MOTION EXAMPLE

An example for a motion measurement topic structure is:

MQTT-SN topic: NBIoT/123456789298765/MES/186A1

The payload contains the values:

- motionDetected: true (0x01)
- speed:-63.2 (0xC2, 0x7C, 0xCC, 0xCD)

➔ MQTT-SN payload: 0x01, 0xC2, 0x7C, 0xCC, 0xCD

3.2.3.3. SINGLE PHASE ELECTRICITY

PayloadID: 186A2

Value	Data Type	Unit	Description
A+	short	kWh	Total active energy, in
A-	short	kWh	Total active energy, out
P+	short	W	Total active power, out
P-	short	W	Total active power, in

FIGURE 12 SINGLE PHASE ELECTRICITY DATA TYPE

3.3. SEND ALARM

In addition to the measurement messages, the NBIoT connector supports alarm messages. Each device can trigger an alarm in the Cloud of Things over the MQTT-SN message. The construction and the application of the alarm MQTT-SN message are described in the following chapter.

3.3.1. STRUCTURE OF MQTT-SN MESSAGE

The general structure of MQTT-SN message is described in the chapter “2.5 Structure of MQTT-SN Message Content“. The structure of the alarm message uses the value “ALM” as event type. The topic “ALM” is used for all alarm messages. The structure of the alarm topic is described in Table 2-7: Alarm Topic Levels.

Topic	Definition
NBIoT	NBIoT root node
IMSI	unique device ID (HexString)
ALM	Alarm
Severity ID	Severity Level (1 – 4)

FIGURE 13 ALARM TOPIC LEVELS

An example for a general alarm structure is:

NBIoT/123456789298765/ALM/1

The Severity ID supports four levels. These levels are described in chapter “Supported Severity Levels”. The severity ID is a numerical character between 1 and 4.

The alarm MQTT-SN payload is defined as a String.

The MQTT-SN payload contains the alarm message. The MQTT-SN payload is the content of the alarm in the Cloud of Things. An example of an alarm MQTT-SN payload is:

“The measured temperature passes the threshold.”

3.3.2. SUPPORTED SEVERITY LEVELS

The alarm message supports four levels of the severity. This levels are Critical, Major, Minor und Warning. The correspondence between severity ID and severity levels is described in Table 2-8: Severity Levels.

Severity ID	Severity Level
1	Critical
2	Major
3	Minor
4	Warning

FIGURE 14 SEVERITY LEVELS

3.4. SEND EVENT

In addition to the measurement and alarm messages, the NBIoT connector supports event messages. Each device can trigger an event in the Cloud of Things over the MQTT-SN message. The construction and the application of the event MQTT-SN message are described in the following chapter. An example for the event message is sending position information.

3.4.1. STRUCTURE OF MQTT-SN MESSAGE

The general structure of MQTT-SN message is described in the chapter “2.5 Structure of MQTT-SN Message Content“. The event message uses the value “EVT” as event type. The topic “EVT” is used for all event messages. The structure of the event topic is described in Table 2-9: Event Type.

Topic	Definition
NBIoT	NBIoT root node
IMSI	unique device ID (HexString)
EVT	Event
EventID	ID for associating the event of the payload (HexString)

FIGURE 15 EVENT TYPE

In the next sections, the supported types of events are described.

3.4.1.1. POSITION (ALTITUDE, LATITUDE, LONGITUDE) WITH MESSAGE

EventID: 1

Value	Data Type	Unit	Description
Alt	float	M	Altitude
Lat	float	degrees	Latitude
Long	float	degrees	Longitude
Message	String	-	Event Message

FIGURE 16 POSITION (ALT, LAT, LONG) WITH MESSAGE

3.4.1.2. POSITION (LATITUDE, LONGITUDE) WITH MESSAGE

EventID: 2

Value	Data Type	Unit	Description
Lat	float	degrees	Latitude
Long	float	degrees	Longitude
Message	String	-	Event Message

FIGURE 17 POSITION (LAT, LONG) WITH MESSAGE

3.4.1.3. POSITION (LATITUDE, LONGITUDE)

EventID: 3

Value	Data Type	Unit	Description
Lat	float	degrees	Latitude
Long	float	degrees	Longitude

FIGURE 18 POSITION (LAT, LONG)

The Event Message is predefined and is “Location Update”

3.4.1.4. POSITION EXAMPLE

An example for an event position structure is:

MQTT-SN topic: NBIoT/123456789298765/EVT/3

The payload contains the values:

- Lat: 6.15173 (0x40, 0xC4, 0xDA, 0xF9)
- Long: 51.211977 (0x42, 0x4C, 0xD9, 0x10)

➔ MQTT-SN payload: 0x40, 0xC4, 0xDA, 0xF9, 0x42, 0x4C, 0xD9, 0x10

3.5. DOWNLINK MESSAGE

Downlink messages are custom messages sent from CoT to the device. Downlink messages can contain configuration values.

3.5.1. WORKFLOW FOR DOWNLINK MESSAGE

Via the CoT GUI, the user has the possibility to configure a downlink message for the device. The user can specify a subtopic and the payload of the downlink message. The predefined structure of the downlink message is described in the chapter “Structure of Downlink Message”. For downlink communication, the MQTT-SN topic structure has 2 subtopics (CMD and INF). The CMD topic is used to send commands to the device (also known as Operations), the INF topic is used for sending command responses from the device. The responses contain status codes which are displayed on the GUI. (See chapter “Status Codes”).

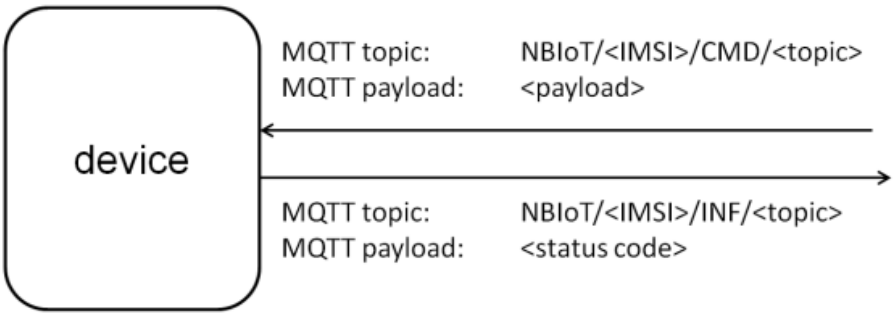

FIGURE 19 COMMUNICATION - DOWNLINK MESSAGE

3.5.2. STRUCTURE OF DOWNLINK MESSAGE

On the CoT GUI, the user must use a predefined input format for commands:

topic:<topic>;data:<payload>

The parameters <topic> and <payload> are freely selectable. The parameter <topic> is a subtopic of the CMD topic. The parameter <payload> is the MQTT-SN payload formatted as a hex dump.

In the next step, an example of usage is described.

Scenario: A timer value shall be sent to the device. The commands topic is “SetTimer” and the payload is any time value (formatted as hex dump).

The complete command that must be entered in the CoTs GUI is:

topic:SetTimer;data:3132

The device receives the following MQTT-SN message:

- MQTT-SN topic: NBIoT/<IMSI>/CMD/SetTimer
- MQTT-SN payload: 0x31, 0x32

If the message is processed successfully, the device sends the following MQTT-SN message to theCoT:

- MQTT-SN topic: NBIoT/<IMSI>/INF/SetTimer
- MQTT-SN payload: 0x30

3.5.3. STATUS CODES

The status codes are used to signal the status of the command. The device sends the status code to the CoT. The CoT processes the code and visualizes the status on the GUI. The code is a unit8 value. All available status codes are described in the Table 2-13: downlink message - status codes.

Value	Status
0 (0x30)	SUCCESFUL
1 (0x31)	FAILED
2 (0x32)	EXECUTING
3 (0x33)	PENDING

FIGURE 20 DOWNLINK MESSAGE - STATUS CODES

3.5.4. SEND DOWNLINK MESSAGE - COT INTERFACE

This chapter shows how to send a downlink message with the CoT GUI. Each device has a submenu for sending the download message. (via the menu entry “DEVICES” → “All devices” → single device → “Shell”).

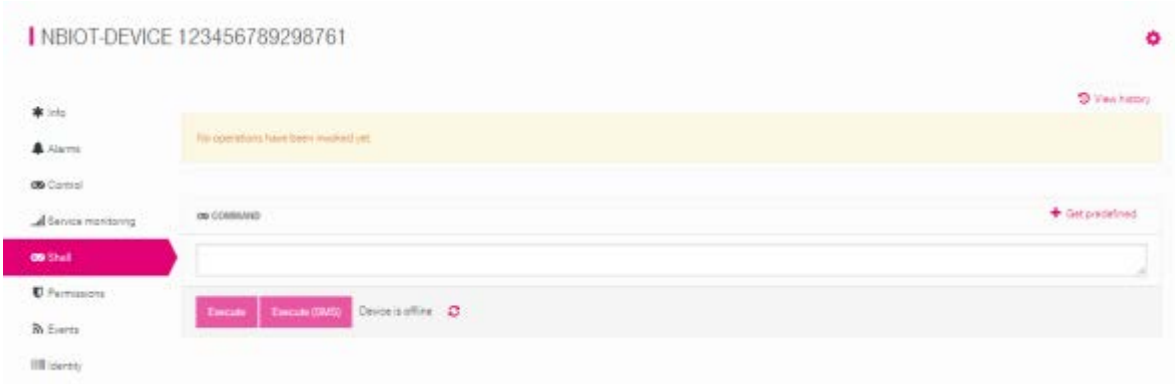

FIGURE 21 DOWNLINK MESSAGE - COT INTERFACE

The downlink message is configured via the command text field. The structure of downlink message is defined in chapter “2.9.2 Structure of Downlink Message”. After pressing the “Execute” button, the downlink message is sent to the device.

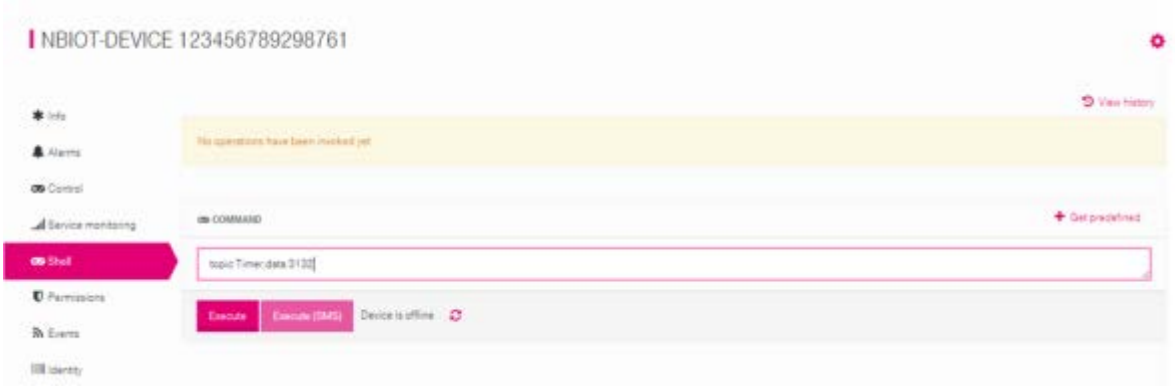

FIGURE 22 EXECUTE DOWNLINK MESSAGE - COT INTERFACE

After sending the downlink message, the status of the message is displayed in the CoT interface. The initial status is pending.

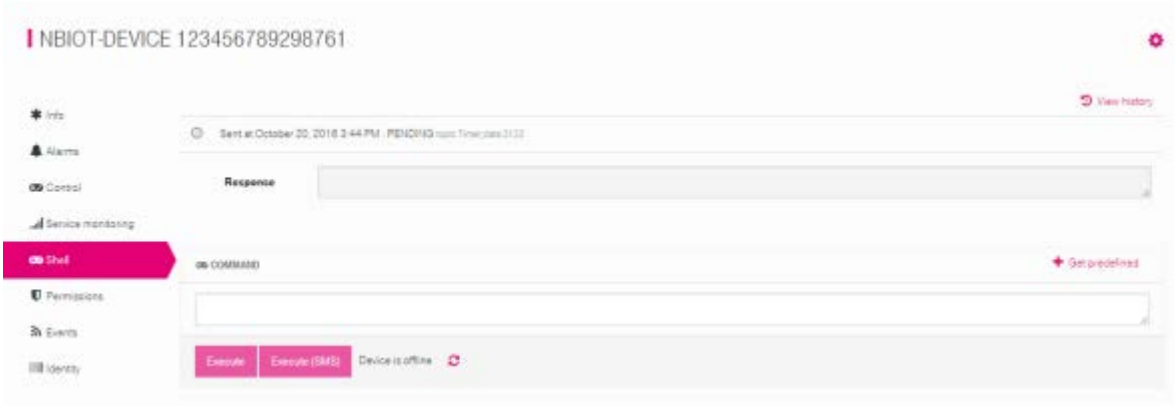

FIGURE 23 STATUS DOWNLINK MESSAGE - COT INTERFACE

After receiving the message, the device sends a response/acknowledgement to the CoT. The response contains the status of the downlink message. The CoT GUI displays this status information. The status can be "SUCCESFULL", "FAILED", "EXECUTING" or "PENDING".

FIGURE 24 STATUS DOWNLINK MESSAGE - COT INTERFACE

This chapter described the procedure for sending the downlink message.

3.5.5. SEND BULK MESSAGES TO GROUPS OF DEVICES

The procedure for creating a device groups is described in the following section.

Via CoT GUI:

- Create a device group:
  - menu "GROUPS"
  - "Add group"-button
  - select a group name

- “Add group”-button
- Add devices:
- menu “GROUPS”

- Created group
  - menu “Sub-assets”
  - “Assign devices”-button
  - select the devices
  - “Assign devices”- button

After a device group was created, the bulk messages can be sent to the devices. To send a bulk message, the user sends a single downlink message to any device. (See chapter “2.9.4 Send Downlink Message - CoT Interface”). The single downlink message serves as basis for sending the bulk message. Via the menu entry “All devices” → single device → “Control”, the user can start sending the bulk message via the “Execute for whole group”-button.

FIGURE 25 DOWNLINK MESSAGE - EXECUTE FOR WHOLE GROUP - COT INTERFACE

In the next step, the user has the possibility to configure the target group, the start time and the delay between the messages.

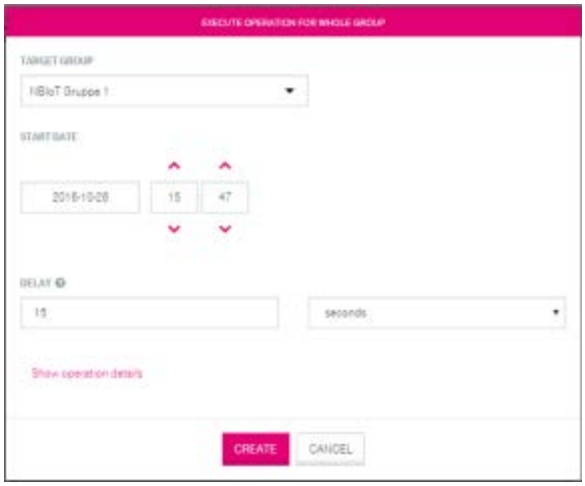

FIGURE 26 CONFIGURATION BULK MESSAGE - COT INTERFACE

After sending the message to the whole group, the status for sending and receiving is accessible via the menu “GROUPS” → device group → “Bulk operations”. The GUI displays status changes for the bulk message.

FIGURE 27 STATUS OF THE BULK MESSAGE - COT INTERFACE

FIGURE 28 STATUS CHANGE OF THE BULK MESSAGE - COT INTERFACE