at home in your wireless world

DECT HAN CMND API

Specification

Version M1.1

July 2018

Headquarters
USA
161 S San Antonio Rd
Suite 10
Los Altos CA 94022
Tel: (408)986-4300
Fax:(408)986-4323

Israel
5 Shenkar Street
Herzelia, Israel 46120
Tel: 972-9-952-9696
Fax: 972-9-954-1234

Europe
Germany
Nelson-Mandela-Platz 18
90459 Nurnberg
Tel: +49-911-2001-0
Fax: +49-911-2001-1210

Scotland
Geddes House
Kirkton North
Livingston EH54 6GU
Tel: +44-(0)1506-407-730

APAC
China
Unit 1 – 4, 11/F
909 Cheung Sha Wan Rd.
Kowloon
Hong Kong
Tel: +852-(3965)-5888

Room 1819, 18/F
Kerry Centre, Renminnan Road,
Shenzhen, China 518001
Tel: +(86 755) 2518 1214

DSPS Solutions
Rm. 24, 3/F, Blk. B,
Proficient Ind. Centre,
6 Wang Chiu Rd.,
Kowloon Bay,
Kowloon, Hong Kong
Tel: +852-(2750)-7325
Fax: +852-(2305)-0640

Ascend Technology Inc. Shenzhen
Room 1303, New World Center,
No.6009
Yitian Road, Futian District
Shenzhen City, China
Tel: +86-755-82024598
Fax: +86-755-23982986

Ascend Technology Inc. Beijing
Rm 607, Hui Huang International
Center,
1st Place, Shangdi 10th Road,
Haidian District, Beijing City, P.R.
China

Japan
1-29-1 Nishi-Gotanda
Shinagawa-Ku Tokyo 141-0031
Tel:+81-(3)-3493-3050

Tomen Electronics
8-27, Kohnan 1 Chome,
Minato-ku, Tokyo
108-8510, Japan
Tel: +81-(3)-5462-9619
Fax: +81-(3)-5462-9686

Korea
Daesung Semiconductors
140-848, RM 401
Wonhyo BD. 46-1
Wonhyo-ro 3ka, Yongsan-gu,
Seoul, Korea
Tel: +82-(2)-3272-7300
Fax: +82-(2)-712-4632-3

Taiwan
Ascendtek Electronics, Inc
11F-7, No. 77, Sec. 1
Hsin Tai Wu Rd.,
Hsi Chih
Taipei Hsien,
Taiwan, R.O.C
Tel: 886-2-26988696
Fax: 886-2-86982138

India
Information Technology Park
Nagawara Village Kasaba Hobli
Bangalore 560045
Tel: +91 80 4024 8399

This document is provided by DSP Group, Inc. and/or one or more of its subsidiaries (“DSP Group”). All information and data contained in this document is for informational purposes only, without any commitment on the part of DSP Group, and is not to be considered as an offer for a contract. DSP Group shall not be liable, in any event, for any claims for damages or any other remedy in any jurisdiction whatsoever, whether in an action in contract, tort (including negligence and strict liability) or any other theory of liability, whether in law or equity including, without limitation, claims for damages or any other remedy in whatever jurisdiction, and shall not assume responsibility for patent infringements or other rights to third parties, arising out of or in connection with this document. Further, DSP Group reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revision changes. These materials are copyrighted and any unauthorized use of these materials may violate copyright, trademark, and other laws. Therefore, no part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of DSP Group. Any new issue of this document invalidates previous issues.

DSP Group reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revision changes.

File: han fun - cmnd api spec m.docx saved 24/07/18 11:11

TABLE OF CHANGES

VERSION	DATE	DESCRIPTION
M1.0	July 2018	First draft
M1.1	July 2018	Formatting changes

TABLE OF CONTENTS

1. Introduction

1.1 Purpose

The purpose of this document is to describe the DECT HAN CMND protocol interface to enable software developers to customize it to meet system requirements.

This document discusses:

Background of Topology and other DECT ULE layers

Link layer description

Application layer description

Flows and sequences

1.2 Definitions, Acronyms and Abbreviations

NAME	DESCRIPTION
BS	Base Station
C → N	Message sent from CMND to Node Host
CMND	Cordless module node
CS	Checksum
DECT	Digital Enhanced Cordless Telecommunications
GW	Gateway
HS	Handset
HAN-FUN	Home Area Network Functional
IE	Information Element
M	Mandatory
MCU	Microcontroller Unit
N→C	Message sent from Node Host to CMND
NODE HOST	This is the microcontroller hardware provided by the customer. It communicates with the CMND either via a simple I/O such as GPIOs or via a more complex protocol over UART/SPI/I2C
O	Optional
RSSI	Received Signal Strength Indicator
ULE	Ultra Low Energy

1.3 References and Bibliography

#	DOCUMENT NAME	VERSION	DATE	LOCATION
[1]	HAN-FUN Protocol	1.4.0	2017-12-15	ULE Alliance

2. Topology

DECT ULE uses a star topology. Each DECT ULE system contains only one base station (BS) and many (up to thousands of) devices.

A repeater may be added for range extension and to act as a pipe; however, this document does not discuss this configuration.

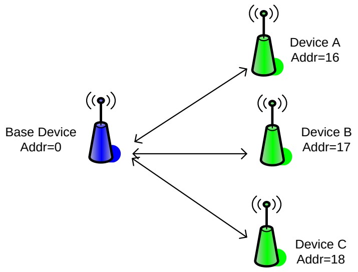


Figure 2-1: Star Topology, No Repeater

2.1 Base station (BS)

All communication passes through the single BS. It may concentrate all activities or, alternately, may pass some or all to 3rd party (user controller, Cloud etc.) The BS address is 0.

2.2 Handset (HS)

A legacy DECT Handset (GAP or CAT-iq) which is registered to same ULE base. Each HS is assigned a specific address by the BS; this address a completely different address from ULE device address ID. For example, there can be HS #1 and ULE device #1.

2.3 Device

The term device is used to describe a DECT Portable Part (PP) that is not a legacy DECT HS. Many (hundreds or thousands of) devices can be supported. Each device is assigned a specific address by the BS, ranging from 16 to 2^{16}.

2.4 Unit

The unit is a functional instance of a certain type (e.g. motion, switch, etc.). A device can host up to 255 units from the same or varied types. A unit is assigned a unit identifier during manufacturing. The combination of the device address and the unit identifier produces a unique unit address.

Unit 0 is reserved for device management, as shown in Figure 2-2. The Unit ID of the first unit is 1.

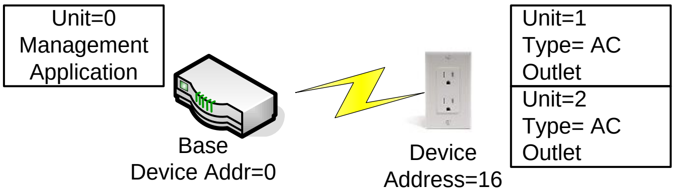


Figure 2-2: Unit Management Scheme

3. Brief Layer Descriptions

3.1 Link Layer

Then CMND and external hardware (for example, an MCU) communicate via UART/SPI/I^{2}C port, depending on the system design. Communication is bidirectional; e.g. sending/receiving events and information. UART protocol settings are shown below. No special settings are required for SPI or I2C communication.

UART protocol settings are as follows:

Baud rate: 115200 bps

Data: 8 bit

Parity bit: None

Stop bit: 1 bit

Network byte order: Big Endian

Flow control: Enable/Disable (at compilation time, optional). Recovery from message loss should be done in higher layers.

3.2 Network Layer

This layer is used for unit management. Each unit should have number ranging from 0 to 255. Unit identifier is provided as part of the CMND API message and defines the source unit ID.

3.3 Data Layer

The UART protocol message structure used for this layer is shown in Table 3-1. The various fields are described in Table 3-2. This protocol should be packed in Big Endian byte order (network order). For example:

16-bit integer: 0x0010 is 00 10 in the message (MSB is sent first).

32-bit integer: 0x12345678 is 12 34 56 78 in the message (MSB is sent first).

Table 3-1: CMND API UART Fields

NAME	Start Code	Length	Cookie	Unit ID	Service ID	Msg ID	Checksum	Data
TYPE	U16	U16	U8	U8	U16	U8	U8	U8 array
VALUES	0xDADA	0 to 0xffff	0 to 0xff	0 to 0xff	0 to 0xffff	0 to 0xff	0 to 0xff	0 to 0xff

Table 3-2: CMND API UART Field Descriptions

FIELD	DESCRIPTION
Start Code	Fixed value defines the beginning of a new message Value: 0xDADA
Length	Total length of the message from Cookie to Data (all inclusive). Length of Data field can be computed by subtracting 6 bytes (Cookie, Unit ID, Service ID, Msg ID and Checksum) from the Length field.
Cookie	Unique message identifier between CMND and Node Host. See section 4.3 for details.
Unit ID	When sent from Node Host to the CMND this value identified the source unit ID (i.e. sending from). When sent from the CMND to the Node Host, this value is the the destination unit ID (i.e. sending to).
Service ID	Defines an ID for a logical group of messages.

FIELD	DESCRIPTION
	See section 4.4 for details.
Msg ID	Message ID under the logical group of messages in the specified Service ID
Checksum	Error checking of the received message. CS = 8 LSBs of byte summation from Length to Data (including Length, not including Checksum field)
Data	Message payload. If message has no data, then this field is empty and Length field is set to 6.

4. Application Layer (CMND API)

4.1 Message Notation

The CMND API service messages use the notations described below:

Requests: These messages are marked with the suffix *xxx_REQ*. A request is message *xxx* sent from element A to element B. Requests can be send from the Node Host to the CMND or vice versa.
Example: Element A (Host) sends a “Turn ON” request to element B. Element B can either be a gateway (GW) or an explicit remote device registered to the same GW. If B is a remote device, then xxx*_REQ* is sent by the GW to the device B.

Confirmations: These messages are marked with the suffix *xxx_CFM*. Confirmation is sent to indicate a previously received request, never as an unsolicited event. A special case of confirmation is a preliminary ACK. This confirmation message means that the request was received by the BS (in terms of OTA) but not yet processed, validated or executed. This message confirms that the message arrived at its destination in terms of a link, and that the request is pending. In some cases, this level of confirmation is enough for the requesting side. For details, see CMND_MSG_GENERAL_LINK_CFM in section 4.10.2.

Example: Element A sends a “Turn ON” request to element B. Element A receives a link confirmation for “Turn ON” if message has arrived to BS.

Responses: These messages are marked with the suffix *xxx_RES*. A response means that the request was executed and a response sent back to the requesting side. The response may contain payload with success/fail status of the execution.
Example: The device has successfully been turned on.

Indications: These messages are marked with the suffix *xxx_IND*. An indication is an unsolicited event.
Example: DECT link is not active.

4.2 Information Elements (IE)

4.2.1 Using IEs

An IE is a self-contained data structure that can be reused within different CMND API messages. Message payloads can contain only IEs. The CMND API defines for each message a list of IEs that can be conveyed within this message.

Examples:

- The CMND_MSG_ALERT_NOTIFY_STATUS_REQ message in “Alert” service contains the CMND_IE_ALERT information element with all the necessary fields for notifying about the alert.
- Most of the confirmation messages reuse the CMND_IE_RESPONSE information element; it contains success/failure status.
- The “CMND_MSG_ONOFF_ON_REQ” request in “On-Off”“ service is sent to the Base; the Base forwards this request to the destination device based on the binding table on base. This request can be extended by explicitly defining the destination device unit and thus skipping the binding table on the BS. This is implemented by adding a CMND_IE_UNIT_ADDR information element to the “Turn On” request.

4.2.2 Information Elements in CMND API Message Payload

The *Data* field of the CMND API message may be either empty or can contain one or more IEs. The IEs are stored sequentially; their order in a message is **not** mandatory. The Data field is structured as shown in Figure 4-1.

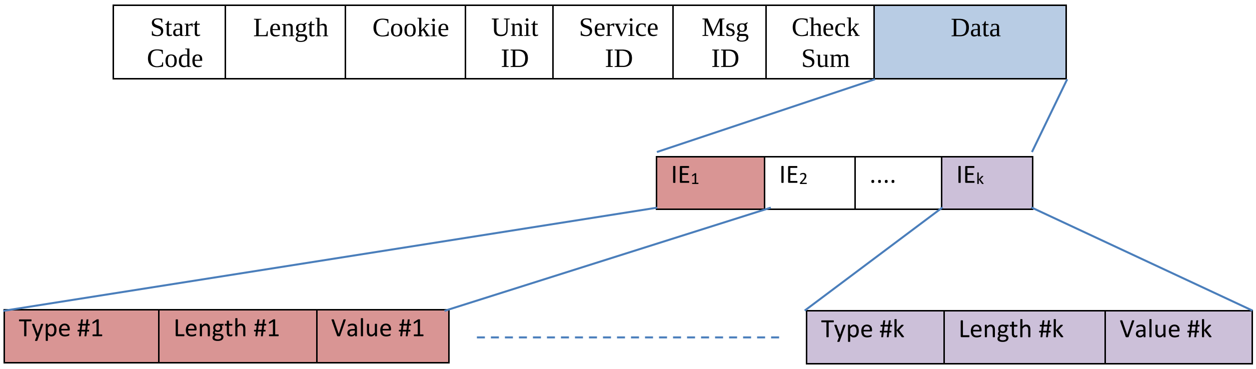


Figure 4-1: Data Field Structure

4.2.3 Structure

Each IE is structured as shown in Table 4-1. Fields are described in Table 4-2. An example of a CMND API message of an Alert service shown in Table 4-3.

Table 4-1: IE Structure Fields

NAME	Type	Length	Value
TYPE	U8	U16	U8 array
VALUE	0 to 0xff	0 to 0xffff	0 to 0xff

Table 4-2: IE Field Descriptions

FIELD	DESCRIPTION
Type	IE unique identifier. See CMND API Information Elements
Length	Length of IE Value field (in bytes)
Value	Information Element payload (Network byte order)

Table 4-3: Alert Service Message Example

FIELD	DATA	BYTE NO
Start Code	da da	1
		2
Length	00 0e	3
		4
Cookie	01	6
Unit ID	01	7
Service ID: Alert	01 00	8
		9
Msg ID: CMND_MSG_TAMPER_ALERT_NOTIFY_STATUS_REQ	04	10

FIELD	DATA	BYTE NO
Checksum	3A	11
IE Type: CMND_IE_TAMPER_ALERT	1D	12
IE Length	00 01	13
		14
IE Value: Tamper alert is ON	00 01	15
IE Type: CMND_IE_RESPONSE_REQUIRED	04	16
IE Length	00 01	17
		18
IE Value: Response is required	01	19

4.2.4 Mandatory/Optional IEs

The CMND API defines for each message which IEs are mandatory (M) and which are optional (O).

Some of IEs may be added to CMND API message payload without being stated in message definition. For example, the CMND_IE_UNIT_ADDR may be added to payload to identify the source or destination address (Device + Unit id). In the table below you will find a list of such IEs:

Table 4-4: Common Optional IEs

INFORMATION ELEMENT	DESCRIPTION
CMND_IE_UNIT_ADDR	Defines Device ID and Unit address. This IE will be added by CMND module in case source address differs from default Device ID = 0x7F and Unit ID = 0x7FFF
CMND_IE_OTA_COOKIE	Contains FUN application sequence fields value if different from zero.

4.3 CMND API Cookie

4.3.1 Overview

The Cookie field is used to identify a request sent from the Node Host to the CMND, and the confirmation sent back from CMND to Host in which a “don’t care” value (0) is set.

4.3.2 Cookie Generation

A Cookie is generated by the Node Host. The same value is later used in confirmation by the CMND.

Table 4-3 defines the Cookie value in different scenarios.

Table 4-5: Cookie Values

SCENARIO	CMND COOKIE VALUE
Node Host sends a request to CMND (REQ)	Node Host generates new value. Another option is to use 0, don’t care.
CMND sends acknowledge for message transmission over DECT (CFM)	CMND uses the same Cookie as provided in the Node Host request
Node Host responds to a request from CMND (RES)	Node Host sets the Cookie to 0, don’t care
The CMND sends a REQ or IND to the Node Host	CMND uses 0, don’t care

4.3.3 Flow Example

Figure 4-2 demonstrates Cookie generation for an “Alarm” event.

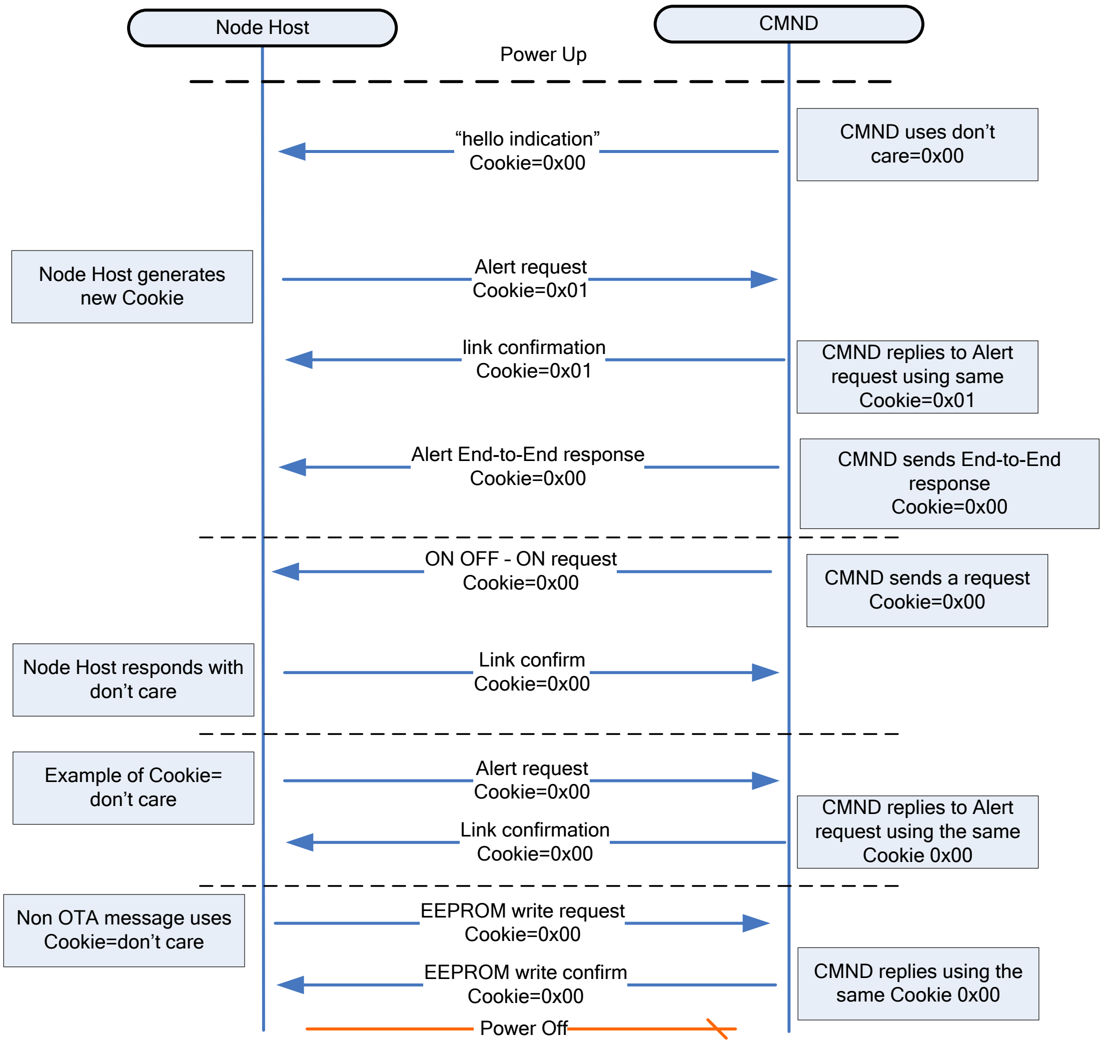

Node Host
CMND
Power Up
“hello indication” Cookie=0x00
CMND uses don’t care=0x00
Node Host generates new Cookie
Alert request Cookie=0x01
link confirmation Cookie=0x01
CMND replies to Alert request using same Cookie=0x01
Alert End-to-End response Cookie=0x00
CMND sends End-to-End response Cookie=0x00
ON OFF - ON request Cookie=0x00
CMND sends a request Cookie=0x00
Node Host responds with don’t care
Link confirm Cookie=0x00
Example of Cookie= don’t care
Alert request Cookie=0x00
Link confirmation Cookie=0x00
CMND replies to Alert request using the same Cookie 0x00
Non OTA message uses Cookie=don’t care
EEPROM write request Cookie=0x00
EEPROM write confirm Cookie=0x00
CMND replies using the same Cookie 0x00
Power Off

Figure 4-2: Cookie Generation for Alarm Event

4.4 CMND API Services

4.4.1 Overview

The CMND API devices are a collection of messages grouped under Service IDs, as listed in Table 4-6. The Service concept is very similar to the object oriented “Class”: Service messages are Class.

Although Service IDs and HAN FUN interfaces are very similar, reference to CMND API should be performed only as explained in this document. One special case is the FUN Service ID, for which the message payload should be constructed with reference to the HAN FUN - OTA API SPEC [1].

Table 4-6: Service IDs

SERVICE ID	NAME	DESCRIPTION
	Management	
0x0001	Device Management	Device registration and de-registration
0x0002	Reserved	
0x0003	Reserved	
0x0004	Identify	Identify Unit/Group in system
0x0005	Reserved	
0x0006	Attribute Reporting	Report attributes by event or periodically
	Generic	
0x0000	General	General-purpose service
0x0100	Alert	Device indicates an alert
0x0101	Tamper Alert	Device indicates a tamper alert
0x0102	Reserved	
0x0103	Reserved	
0x0104	Keep Alive	Device indicates it is alive
0x0105	Reserved	
0x0106	On Off	Turn action on/off
0x0107	Reserved	
0x0108	FUN	Send and receive FUN OTA messages
0x0109	Debug	Internal debug service of Han application
	System	
0x0201	System	System functionality
0x0202	Reserved	
0x0203	Parameters	Access CMND parameters
0x0204	Sleep	Turn DECT module off
0x0205	Reserved	
0x020A	ULE Voice Call	Manage voice call
0x020B	Production	Production mode management
0x020C	SUOTA	Software update over the air service
0x020D	Certification	Certification
0x020E	Remote Control	Remote Control
0x020F	SUOTA Proprietary	SUOTA Proprietary
0x0210	Broadcasting	Broadcasting
0x0211	Media	Media file play and stop service

4.5 Message Confirmation

Message ACK is a Transport Layer confirmation indicating that the message was sent to gateway/base station. It does not indicate the message reached the Application Layer on the destination device. This confirmation is part of the CMND API General Service message, CMND_MSG_GENERAL_LINK_CFM.

4.6 Raw FUN with CMND API

The CMND API protocol defines services for specific purpose such as Alert, Tamper, Device Management, etc. The CMND API also defines a general purpose service - FUN service. This service is used for constructing full FUN message following the FUN standard. The FUN service structure is not bitwise exact to FUN protocol structure, but it contains the exact same fields (i.e. same logical payload). See FUN service for full spec.

The FUN service can be used for following:

- Implement FUN interfaces which are not supported in CMND API. For example:
  - FUN Power interface (0x0110)
  - FUN Time interface (0x0102)
- Implement customer proprietary functionality (FUN proprietary interface):
  - Proprietary customization of the Node Host
  - Proprietary calibration
  - Proprietary reporting
  - Proprietary attributes
  - Other proprietary functionality

The next two paragraphs are providing guidelines and examples on constructing FUN messages for usages defined above.

4.6.1 Using FUN Service to Code FUN Interfaces

One of usages of CMND API FUN service is constructing FUN standard messages. Any FUN interface can be constructed, including those which are already supported by the CMND API services. The Alert Interface can be either created using CMND Alert service or using FUN service and constructing the message from scratch.

Next table presents an example for constructing Low Battery indication for FUN Power interfaces. This interface is not implemented in none of CMND API services. The CMND API content shall be coded as following:

- Service: FUN service (0x0108)
- Message: CMND_MSG_FUN_SEND_REQ (0x1)
- Payload: CMND_IE_FUN (0x5)

Table 4-7: Power interface using FUN service

Field	Data	Byte No
Start Code	da da	1
		2
Length	00 XX[1]	3
		4
Cookie	01	6
Unit Id	01	7
Service Id = FUN	01 08	8
		9
Msg Id: CMND_MSG_FUN_SEND_REQ	01	10
Checksum	ZZ	11
IE type: CMND_IE_FUN	05	12
IE length	00 XX[1]	13

Field	Data	Byte No
		14
IE payload	...	

XX^1 – reperesents an unknown byte value since not all data is presented in this example

The next table presents how to fill CMND_IE_FUN information element for low battery indication:

Table 4-8: CMND_IE_FUN using FUN service

Field	Type	Value	Description
Extended Mode	u8	0x0	0: Non Extended Mode – Default Mode
Source device id	u16	Node host device id	Address of source device. Node Host receives device id in registration confirmation. Node Host can also receive it with Get Status message (general service)
Source unit id	u8	0x0	Source unit Id Power interface shall be used with Unit 0
Address type	u8	0x0	0: Individual Address
Destination device id	u16	0x7FFF	Address of the specific destination device
Destination unit id	u8	0xFF	Destination specific unit identifier
Application reference	u8	0x00	Per use case
Message type	u8	0x01	0x1: Command message
Interface type	u8	0x0	0: Client
Interface Id	u16	0x0110	Power FUN Interface identifier
Interface member	u8	0x03	Command 0x03 –Low Battery voltage alert
Data length	u16	0x0	Length of Data payload (in bytes) 0 – no data
Data payload	u8[]	None	Command payload.

The FUN service can also be used for receiving full FUN messages for FUN interfaces which are not implemented in CMND API. Next example presents CMND API message received on Node Host which contains FUN “Get attribute” message for Power interface for “Battery Voltage” attribute (0x07). The CMND API content shall be:

- Service: FUN service (0x0108)
- Message: CMND_MSG_FUN_RECV_IND (0x2)
- Payload: CMND_IE_FUN (0x5)

Table 4-9: Power interface using FUN service

Field	Data	Byte No
Start Code	da da	1
		2
Length	00 XX^1	3
		4
Cookie	01	6
Unit Id	00	7
Service Id = FUN	0108	8
		9

Field	Data	Byte No
Msg Id: CMND_MSG_FUN_RECV_IND	02	10
Checksum	XX[1]	11
IE type: CMND_IE_FUN	05	12
IE length	00 XX[1]	13
		14

XX[1] – reperesents an unknown byte value since not all data is presented in this example

The content of CMND_IE_FUN information element shall be:

Table 4-10: CMND_IE_FUN using FUN service

Field	Type	Value	Description
Extended Mode	u8	0x0	0: Non Extended Mode – Default Mode
Source device id	u16	Node host device id	Address of source device.
Source unit id	u8	0x0	Source unit Id Power interface shall be used with Unit 0
Address type	u8	0x0	0: Individual Address
Destination device id	u16	Node host device id	Address of the specific destination device
Destination unit id	u8	0x00	Destination specific unit identifier Power is should be implemented in Unit 0
Application reference	u8	0x00	Per use case
Message type	u8	0x01	0x4: Attribute get message
Interface type	u8	0x0	0: Client
Interface Id	u16	0x0110	Power FUN Interface identifier
Interface member	u8	0x03	0x07 – Battery voltage attribute identifier
Data length	u16	0x0	Length of Data payload (in bytes) 0 – no data
Data payload	u8[]	None	Command payload.

4.6.2 Using FUN Service for Proprietary Interfaces

Another usage for the CMND API FUN service is constructing FUN messages for proprietary FUN interfaces.

For example: Node Host uses proprietary interface for calibrating the AC outlet form Base.

Proprietary interfaces shall start from 0x7F00 offset (as defined by FUN standard). Messages and attributes are to be defined by Node Host. The CMND side is not aware of message structure and will not perform any validation on the content transferred from Node Host to Base or vice versa. Message transmission is bidirectional (as with other FUN interfaces), from device to base and from base to device.

Outgoing messages shall be coded as following:

- Service: FUN service (0x0108)
- Message: CMND_MSG_FUN_SEND_REQ (0x1)
- Payload: CMND_IE_FUN (0x5)
-

Incoming message shall be coded as following:

- Service: FUN service (0x0108)
- Message: CMND_MSG_FUN_RECV_IND (0x2)
- Payload: CMND_IE_FUN (0x5)

The CMND_IE_FUN shall be coded as defined in CMND_IE_FUN section. The payload shall be coded according to definition of the proprietary interfaces defined by customer.

4.6.3 Using FUN Service to Get Attribute

In this example GW wants to get FUN attribute value from ULE device. This attribute is stored in the Node host.

- A “Get Attribute” FUN message is sent to CMND node (in this example we use interface 0x7F16)
- A CMND API message is sent to Node host.
- The node host responses to CMND node
- The CMND node translates the message to FUN and sends the response to the GW

The following is the message from CMND node to Node host:

- Service: FUN service (0x0108)
- Message: CMND_MSG_FUN_RECV_IND (0x2)
- Payload: CMND_IE_FUN (0x5)

Table 4-11: Get attribute using FUN service

Field	Data	Byte No
Start Code	da da	1
		2
Length	00 XX[1]	3
		4
Cookie	00	6
Unit Id	01	7
Service Id = FUN	01 08	8
		9
Msg Id: CMND_MSG_FUN_RECV_IND	02	10
Checksum	XX[1]	11
IE type: CMND_IE_FUN	05	12
IE length	00 XX[1]	13
		14
IE payload	…	

XX[1] – represents an unknown byte value since not all data is presented in this example

The content of CMND_IE_FUN information element shall be:

Table 4-12: CMND_IE_FUN using FUN service

Field	Type	Value	Description
Extended Mode	u8	0x0	0: Non Extended Mode – Default Mode
Source device id	u16	Node host device id	Address of source device.
Source unit id	u8	0x1	Source unit Id
Address type	u8	0x0	0: Individual Address
Destination device id	u16	Node host device id	Address of the destination device
Destination unit id	u8	0x01	Destination specific unit identifier The command is sent to Unit 1
Application reference	u8	0x00	Per use case

Field	Type	Value	Description
Message type	u8	0x01	0x4: Attribute get message
Interface type	u8	0x0	0x1: Server
Interface Id	u16	0x7f16	FUN Interface identifier
Interface member	u8	0x01	Attribute ID we want to read
Data length	u16	0x0	Length of Data payload (in bytes) 0 – no data
Data payload	u8[]	None	Command payload.

The following is the response from Node host to CMND node with attribute value

- Service: FUN service (0x0108)
- Message: CMND_MSG_FUN_SEND_REQ (0x1)
- Payload: CMND_IE_FUN (0x5)

Table 4-13: Get attribute response using FUN service

Field	Data	Byte No
Start Code	da da	1
		2
Length	00 XX[1]	3
		4
Cookie	00	6
Unit Id	01	7
Service Id = FUN	01 08	8
		9
Msg Id: CMND_MSG_FUN_SEND_REQ	01	10
Checksum	XX[1]	11
IE type: CMND_IE_FUN	05	12
IE length	00 XX[1]	13
		14
IE payload	…	

XX[1] – represents an unknown byte value since not all data is presented in this example

The content of CMND_IE_FUN information element shall be:

Table 4-14: CMND_IE_FUN using FUN service

Field	Type	Value	Description
Extended Mode	u8	0x0	0: Non Extended Mode – Default Mode
Source device id	u16	Node host device id	Address of source device. Node Host receives device id in registration confirmation. Node Host can also receive it with Get Status message (general service)
Source unit id[1]	u8	0x1	Source unit Id Unit that responses
Address type	u8	0x0	0: Individual Address
Destination device id	u16	0x7FFF	Address of the specific destination device
Destination unit id	u8	0xFF	Destination specific unit identifier
Application reference[2]	u8	0x00	Per use case
Message type	u8	0x05	0x5: FUN Get attribute response
Interface type[3]	u8	0x0	1: Server
Interface Id	u16	0x7f16	FUN Interface identifier
Interface member	u8	0x01	Attribute id 1
Data length[4]	u16	0x1	Length of Data payload (in bytes)
Data payload	u8[]	YY	Attribute value, in this case its 1byte

[1] Source unit id must be equal to Destination unit id in the "get attribute" request

[2] Application reference should comply with [4] FUN protocol application reference field

[3] Interface type – refer to [4] for explanation on values of this field. Setting this field to 0 denotes a client interface, while setting it to 1 denotes a server interface. The interface type is of the destination (receiver) device when message "Message Type" specifies a command or a request. In contrast, in response message the interface type is of the source (sender) device.

This example demonstrates a "get attribute" of an attribute implemented by interface 0x7F16 server side. Therefore, interface type is set to 1 (server) in "Get attribute" request and set to 1(server) again as the response is from server side.

[4] Data length is decided base on the definition of the proprietary interface. In this example the length was randomly chosen as 1 byte.

4.7 Device Management Service

The following abbreviations are used in tables in this section:

N → C: Message can be sent from Node Host to CMND

C → N: Message can be sent from CMND to Node Host

M: Mandatory

O: Optional

4.7.1 Overview

This service is used for registering devices and units with their associated data, such as Unit Type and the list of interfaces they support.

4.7.2 Messages

MESSAGE ID	MESSAGE NAME	MESSAGE DESCRIPTION	N TO C	C TO N
0x01	Reserved			
0x02	Reserved			
0x03	CMND_MSG_DEV_MGNT_REGISTER_DEVICE_REQ	Register device to base	Y	N
0x04	CMND_MSG_DEV_MGNT_REGISTER_DEVICE_CFM	Registration started	N	Y
0x05	CMND_MSG_DEV_MGNT_REGISTER_DEVICE_IND	Success status of registration	N	Y
0x06	CMND_MSG_DEV_MGNT_DEREGISTER_DEVICE_REQ	Local deregistration request of device	Y	N
0x07	CMND_MSG_DEV_MGNT_DEREGISTER_DEVICE_CFM	Deregistration confirmation	N	Y
0x08	CMND_MSG_DEV_MGNT_DEREGISTER_DEVICE_IND	Deregistration indication	N	Y

4.7.2.1 CMND_MSG_DEV_MGNT_REGISTER_DEVICE_REQ

This message is used for registering a device and its units with their associated types on the remote base (gateway). Device unit types and the list of optional FUN interfaces is stored in CMND API parameter Unit info table (0xA). The CMND node will confirm on start of registration with CMND_MSG_DEV_MGNT_REGISTER_DEVICE_CFM.

IE	M/O	Comment
CMND_IE_BASE_WANTED	O	Contains specific BS RFPI. Device will try to register on BS with specified RFPI only. If omitted device will search for any base open for registration

4.7.2.2 CMND_MSG_DEV_MGNT_REGISTER_DEVICE_CFM

This message is a response to a registration request. The CMND informs the Node Host that CMND registration procedure is in progress. The registration result is reflected in CMND_MSG_DEV_MGNT_REGISTER_DEVICE_IND.

IE	M/O	Comment
CMND_IE_RESPONSE	M	0x0 - “start registration” succeeded (does not indicate that registration completed) 0x1 – “start registration” failed

4.7.2.3 CMND_MSG_DEV_MGNT_REGISTER_DEVICE_IND

This indication is sent from the CMND to the Node Host to indicate the registration status between the device and the GW/BS after the registration procedure completes. The message can also be sent by the CMND to Node Host in unsolicited manner if registration was triggered internally by the CMND. When the registration procedure is completed, the CMND sends registration status.

IE	M/O	Comment
CMND_IE_REGISTRATION_RESPONSE	M	Contains registration status, allocated device ID and additional discriminator fields.

4.7.2.4 CMND_MSG_DEV_MGNT_DEREGISTER_DEVICE_REQ

This is a request to de-register the device. The CMND sends a FUN “deregister” command to base, waits for a successful response from base, then indicates back to Node Host of the result. No IE is used. The CMND should respond first with CMND_MSG_DEV_MGNT_DEREGISTER_DEVICE_CFM and later with CMND_MSG_DEV_MGNT_DEREGISTER_DEVICE_IND.

In case there’s no response from base, deregistration will be completed as local deregistration, and will be indicated by CMND_MSG_DEV_MGNT_DEREGISTER_DEVICE_IND anyway to the Node host.

4.7.2.5 CMND_MSG_DEV_MGNT_DEREGISTER_DEVICE_CFM

This message is a confirmation for handling deregistration request. The CMND informs the Node Host that CMND de-registration procedure is in progress. The result is reflected in CMND_MSG_DEV_MGNT_DEREGISTER_DEVICE_IND.

IE	M/O	Comment
CMND_IE_RESPONSE	M	0x00 – Success 0x02 – Not Registered

4.7.2.6 CMND_MSG_DEV_MGNT_DEREGISTER_DEVICE_IND

This message is an indication that device is now deregistered. This indication can arrive due to previously sent de-registration request, or in case device was remotely removed/deleted/deregistered from the BS.

IE	M/O	Comment
CMND_IE_DEREGISTRATION_RESPONSE	M	0x00 – Success 0x08 - Not authorized. 0xFF – Unknown reason.

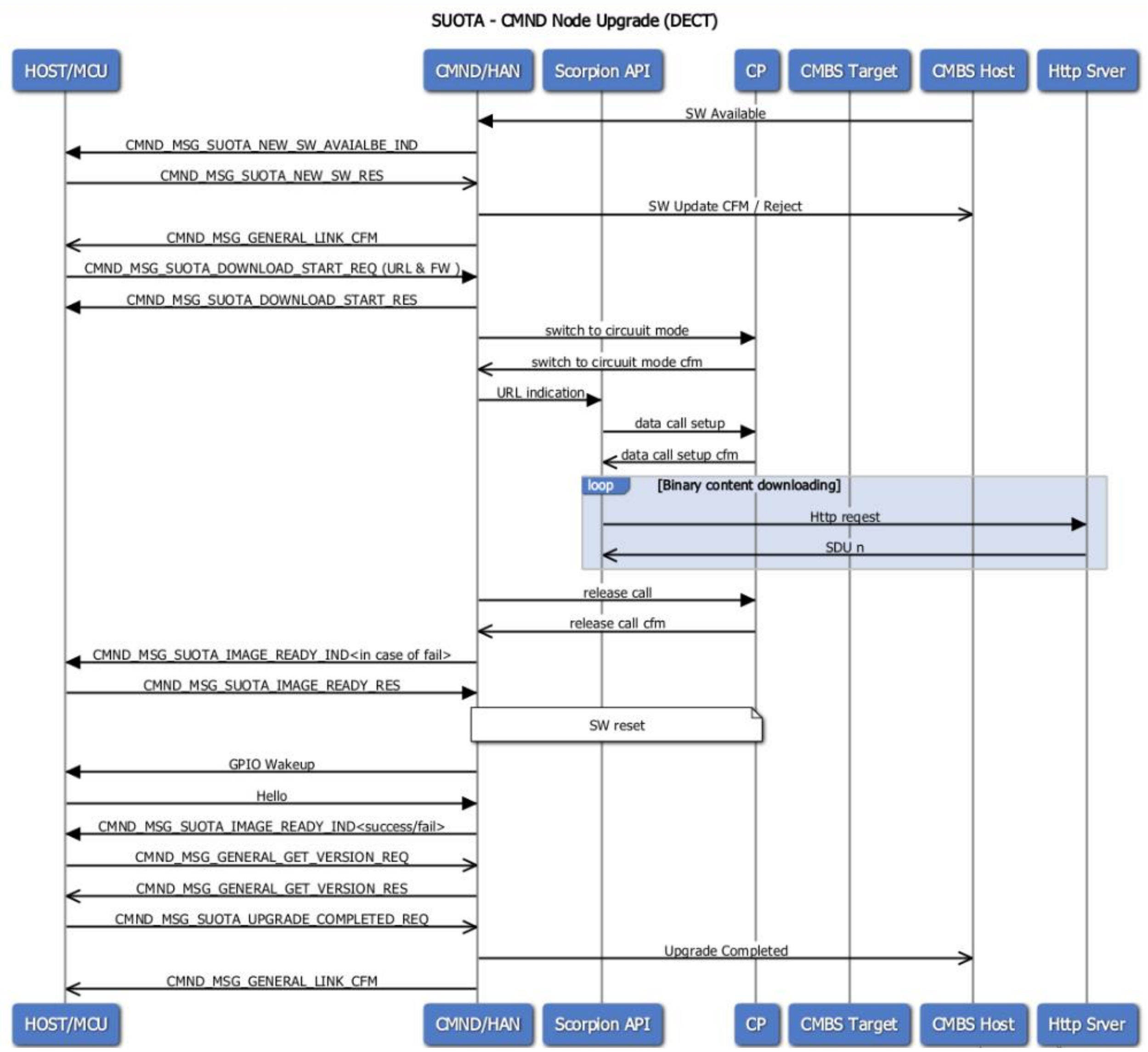
4.8 SUOTA Service

4.8.1 Overview

This service is used for upgrading the SW image of a Node Host (MCU) or of a CMND module (DECT). The upgrade can be initiated either from BS or from Node Host.

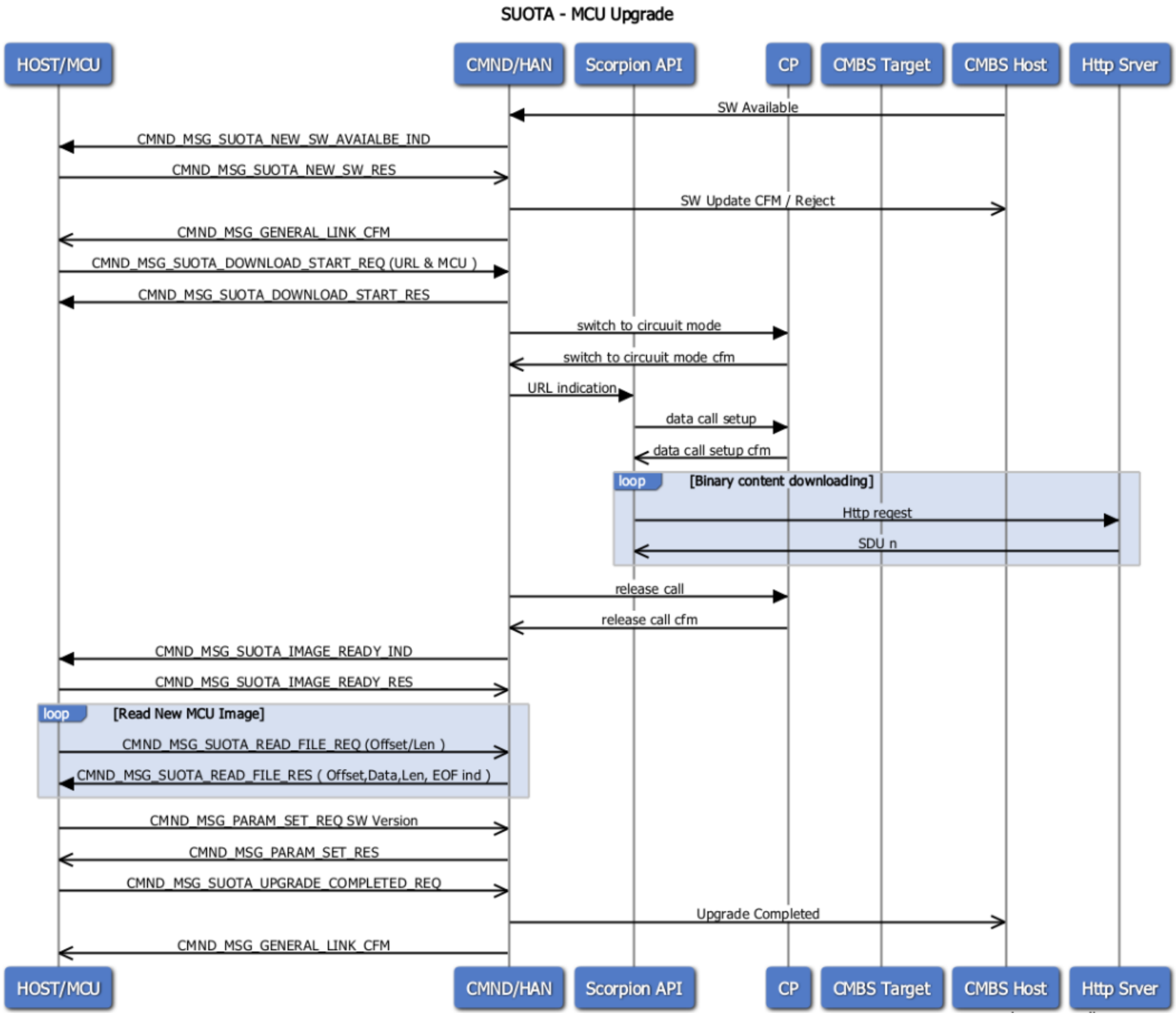
4.8.1.1 SUOTA for CMND node (DECT) initiated by BS

Next figure depicts SW upgrade for CMND module. Base initiates the process by notifying Node Host when new version is available (CMND_MSG_SUOTA_NEW_SW_AVAIALBE_IND). The Node Host replies to Base by accepting or rejecting the SW Upgrade (CMND_MSG_SUOTA_NEW_SW_RES) and then indicates to CMND module to start downloading the new SW image (CMND_MSG_SUOTA_DOWNLOAD_START_REQ). When download completes the CMND modules sends indication with success status if upgrade succeeded or failure status in case of a failure (CMND_MSG_SUOTA_IMAGE_READY_IND). In case of failure the Node Host must reply with CMND_MSG_SUOTA_IMAGE_READY_RES to confirm message is received. In case of success the response should not be sent from Node to CMND. The Node Host then reads the new version from CMND module and updates the Base that SW Upgrade has completed (CMND_MSG_SUOTA_UPGRADE_COMPLETED_REQ). The last message is passed all the way to the base.


SUOTA - CMND Node Upgrade (DECT)
HOST/MCU
CMND/HAN
Scorpion API
CP
CMBS Target
CMBS Host
Http Srver
SW Available
CMND_MSG_SUOTA_NEW_SW_AVAIALBE_IND
CMND_MSG_SUOTA_NEW_SW_RES
SW Update CFM / Reject
CMND_MSG_GENERAL_LINK_CFM
CMND_MSG_SUOTA_DOWNLOAD_START_REQ (URL & FW)
CMND_MSG_SUOTA_DOWNLOAD_START_RES
switch to circuuit mode
switch to circuit mode cfm
URL indication
data call setup
data call setup cfm
loop
[Binary content downloading]
Http reqest
SDU n
release call
release call cfm
CMND_MSG_SUOTA_IMAGE_READY_IND<in case of fail>
CMND_MSG_SUOTA_IMAGE_READY_RES
SW reset
GPIO Wakeup
Hello
CMND_MSG_SUOTA_IMAGE_READY_IND<success/fail>
CMND_MSG_GENERAL_GET_VERSION_REQ
CMND_MSG_GENERAL_GET_VERSION_RES
CMND_MSG_SUOTA_UPGRADE_COMPLETED_REQ
Upgrade Completed
CMND_MSG_GENERAL_LINK_CFM
HOST/MCU
CMND/HAN
Scorpion API
CP
CMBS Target
CMBS Host
Http Srver

4.8.1.2 SUOTA for Node Host (MCU) initiated by BS

Next figure depicts SW Upgrade for Node Host module (MCU upgrade). Base initiates the process and notifies the Node host on new version availability (CMND_MSG_SUOTA_NEW_SW_AVAIALBE_IND). The Node Host accepts SW Upgrade (CMND_MSG_SUOTA_NEW_SW_RES) and then indicates to CMND to start downloading the new image (CMND_MSG_SUOTA_DOWNLOAD_START_REQ). The image is downloaded and stored on the CMND Node Flash. When download is completed the CMND Node indicates that image is ready (CMND_MSG_SUOTA_IMAGE_READY_IND). The Node host should reply with CMND_MSG_SUOTA_IMAGE_READY_RES. Next the Node host reads the new image (in chunks) from the CMND node (CMND_MSG_SUOTA_READ_FILE_REQ). When read is complete the Node host can do SW upgrade. When SW upgrade is complete the Node host sends a confirmation to Base (CMND_MSG_SUOTA_UPGRADE_COMPLETED_REQ). The last message goes all the way to the Base.


SUOTA - MCU Upgrade
HOST/MCU
CMND/HAN
Scorpion API
CP
CMBS Target
CMBS Host
Http Srver
SW Available
CMND_MSG_SUOTA_NEW_SW_AVAIALBE_IND
CMND_MSG_SUOTA_NEW_SW_RES
SW Update CFM / Reject
CMND_MSG_GENERAL_LINK_CFM
CMND_MSG_SUOTA_DOWNLOAD_START_REQ (URL & MCU)
CMND_MSG_SUOTA_DOWNLOAD_START_RES
switch to circuuit mode
switch to circuit mode cfm
URL indication
data call setup
data call setup cfm
loop
[Binary content downloading]
Http reqest
SDU n
release call
release call cfm
CMND_MSG_SUOTA_IMAGE_READY_IND
CMND_MSG_SUOTA_IMAGE_READY_RES
loop
[Read New MCU Image]
CMND_MSG_SUOTA_READ_FILE_REQ (Offset/Len)
CMND_MSG_SUOTA_READ_FILE_RES (Offset,Data,Len, EOF ind)
CMND_MSG_PARAM_SET_REQ SW Version
CMND_MSG_PARAM_SET_RES
CMND_MSG_SUOTA_UPGRADE_COMPLETED_REQ
Upgrade Completed
CMND_MSG_GENERAL_LINK_CFM
HOST/MCU
CMND/HAN
Scorpion API
CP
CMBS Target
CMBS Host
Http Srver

4.8.1.3 SUOTA for CMND node (DECT) initiated by Node Host

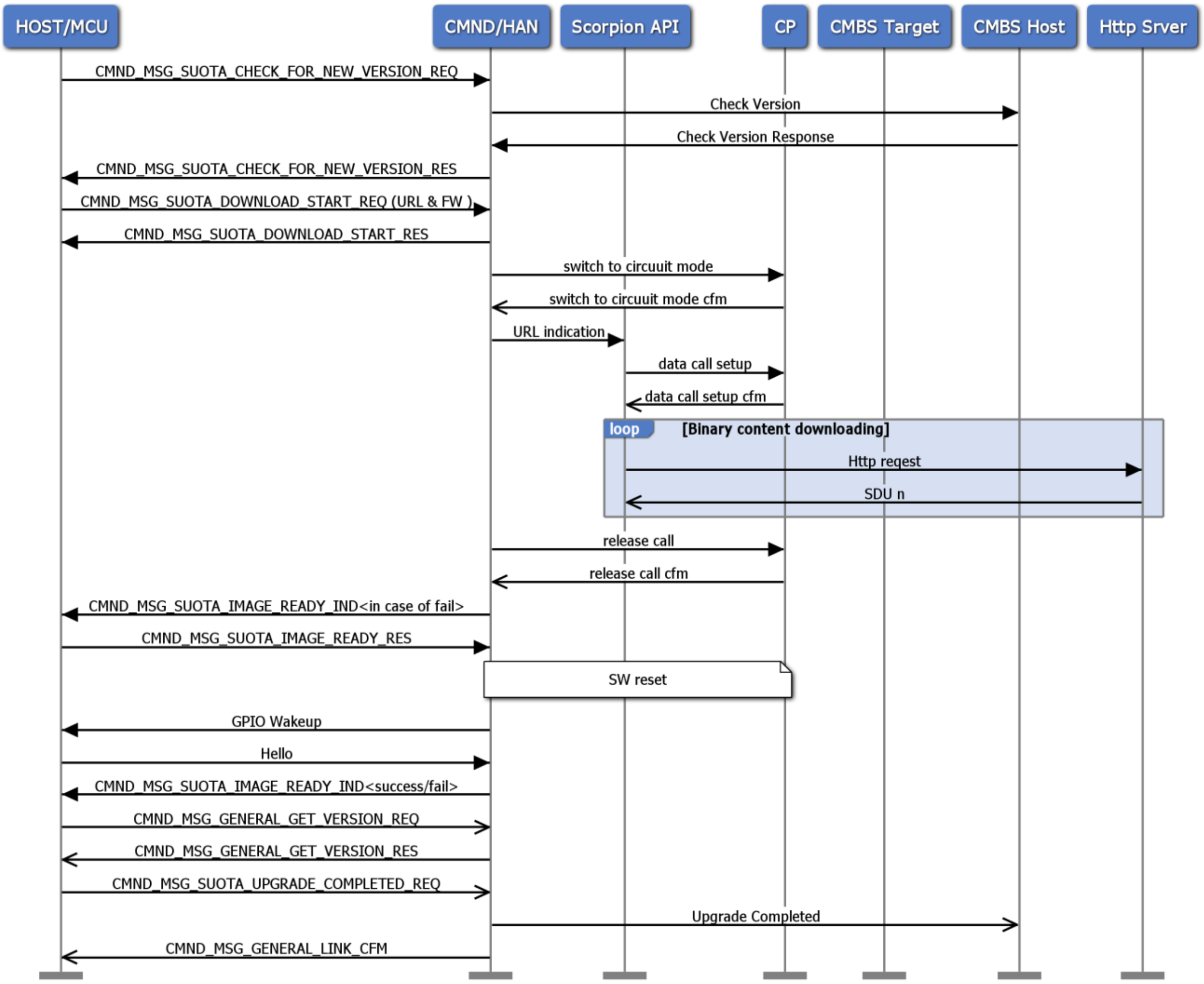
The figure below depicts SW upgrade of CMND Node (DECT module) initiated by Node Host.

Node Host initiates the process by sending CMND_MSG_SUOTA_CHECK_FOR_NEW_VERSION_REQ to check if newer software is available. Note that CMND_IE_CURRENT_SW_INFO information element can be omitted (The CMND node will send current CMND Node version to base in FUN message).

The Node Host receives CMND_MSG_SUOTA_CHECK_FOR_NEW_VERSION_RES with a result of version check. If the result shows that new version is available, then Node Host start downloading the new image by sending CMND_MSG_SUOTA_DOWNLOAD_START_REQ.

The rest of the process is the similar to SUOTA for CMND node (DECT) initiated by BS.

SUOTA - CMND Node Upgrade (DECT) initiated by Node Host


HOST/MCU
CMND/HAN
Scorpion API
CP
CMBS Target
CMBS Host
Http Srver
CMND_MSG_SUOTA_CHECK_FOR_NEW_VERSION_REQ
Check Version
Check Version Response
CMND_MSG_SUOTA_CHECK_FOR_NEW_VERSION_RES
CMND_MSG_SUOTA_DOWNLOAD_START_REQ (URL & FW)
CMND_MSG_SUOTA_DOWNLOAD_START_RES
switch to circuuit mode
switch to circuuit mode cfm
URL indication
data call setup
data call setup cfm
loop
[Binary content downloading]
Http reqest
SDU n
release call
release call cfm
CMND_MSG_SUOTA_IMAGE_READY_IND<in case of fail>
CMND_MSG_SUOTA_IMAGE_READY_RES
SW reset
GPIO Wakeup
Hello
CMND_MSG_SUOTA_IMAGE_READY_IND<success/fail>
CMND_MSG_GENERAL_GET_VERSION_REQ
CMND_MSG_GENERAL_GET_VERSION_RES
CMND_MSG_SUOTA_UPGRADE_COMPLETED_REQ
Upgrade Completed
CMND_MSG_GENERAL_LINK_CFM

4.8.1.4 SUOTA for Node Host (MCU) initiated by Node Host

The figure below depicts SW Upgrade for Node Host module (MCU upgrade) initiated by Node Host.

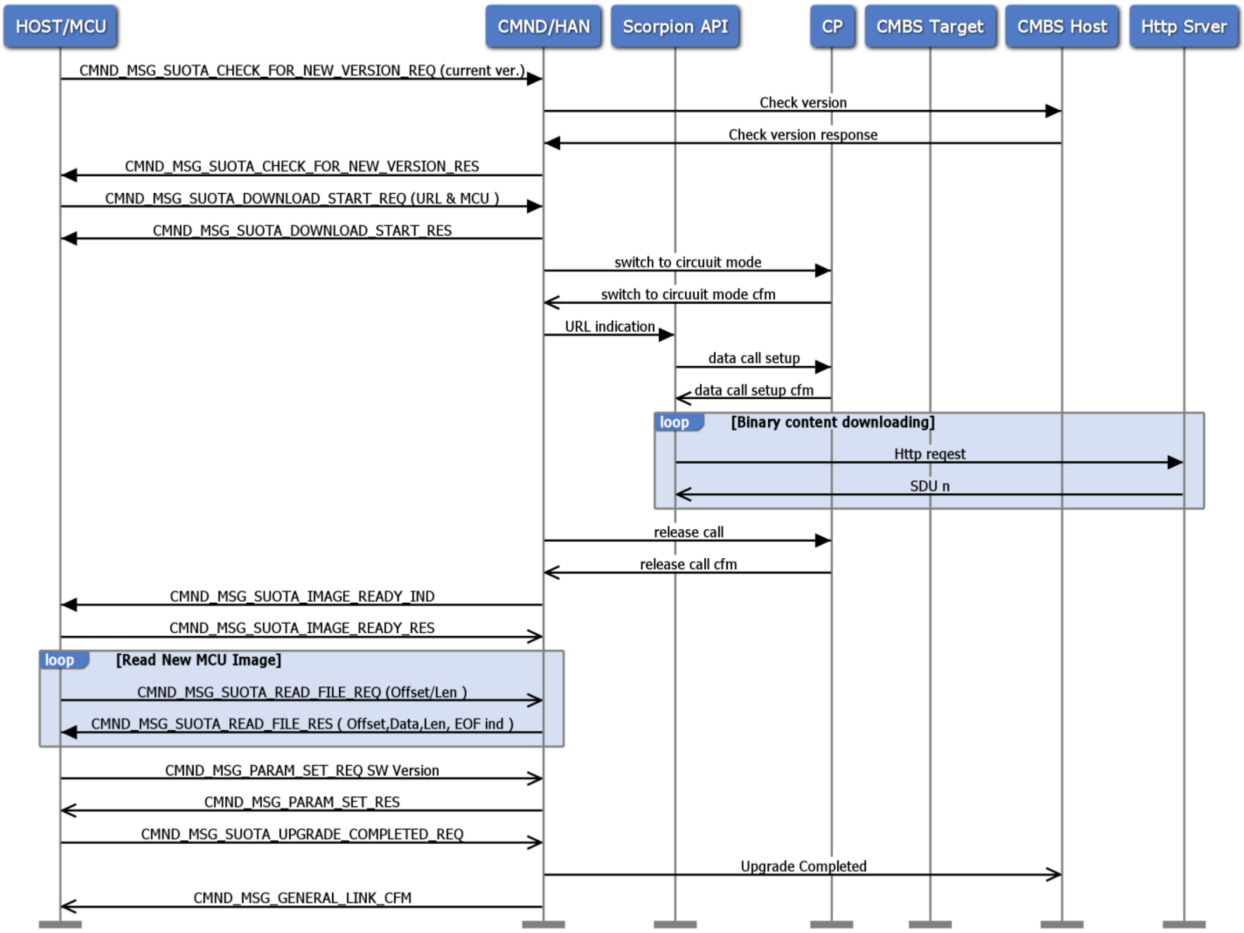
Node Host checks for new version availability using the CMND_MSG_SUOTA_CHECK_FOR_NEW_VERSION_REQ message. The Node Host should set current version in the payload. This information will be used by base when preparing the response.

The Node Host will receive a response CMND_MSG_SUOTA_CHECK_FOR_NEW_VERSION_RES that will indicate availability of new version.

If new version is available, then the Node Host requests to start downloading the new image by sending CMND_MSG_SUOTA_DOWNLOAD_START_REQ.

The rest of the process is the same as SUOTA for CMND node (DECT) initiated by BS.

SUOTA - MCU Upgrade initiated by Node Host


HOST/MCU
CMND/HAN
Scorpion API
CP
CMBS Target
CMBS Host
Http Srver
CMND_MSG_SUOTA_CHECK_FOR_NEW_VERSION_REQ (current ver.)
Check version
Check version response
CMND_MSG_SUOTA_CHECK_FOR_NEW_VERSION_RES
CMND_MSG_SUOTA_DOWNLOAD_START_REQ (URL & MCU)
CMND_MSG_SUOTA_DOWNLOAD_START_RES
switch to circuuit mode
switch to circuuit mode cfm
URL indication
data call setup
data call setup cfm
loop
[Binary content downloading]
Http reqest
SDU n
release call
release call cfm
CMND_MSG_SUOTA_IMAGE_READY_IND
CMND_MSG_SUOTA_IMAGE_READY_RES
loop
[Read New MCU Image]
CMND_MSG_SUOTA_READ_FILE_REQ (Offset/Len)
CMND_MSG_SUOTA_READ_FILE_RES (Offset,Data,Len, EOF ind)
CMND_MSG_PARAM_SET_REQ SW Version
CMND_MSG_PARAM_SET_RES
CMND_MSG_SUOTA_UPGRADE_COMPLETED_REQ
Upgrade Completed
CMND_MSG_GENERAL_LINK_CFM

4.8.2 Messages

MESSAGE ID	MESSAGE NAME	MESSAGE DESCRIPTION	N → C	C → N
0x01	CMND_MSG_SUOTA_NEW_SW_AVAIALBE_IND	New SW available indication sent from base.	N	Y

MESSAGE ID	MESSAGE NAME	MESSAGE DESCRIPTION	N→C	C→N
0x02	CMND_MSG_SUOTA_NEW_SW_RES	Response on New SW available indication. Reject/Confirm the request.	Y	N
0x03	CMND_MSG_SUOTA_DOWNLOAD_START_REQ	Request from CMND to Start the download process.	Y	N
0x04	CMND_MSG_SUOTA_DOWNLOAD_START_RES	Confirm Start request	N	Y
0x05	CMND_MSG_SUOTA_IMAGE_READY_IND	Image ready on flash (success/fail)	N	Y
0x06	CMND_MSG_SUOTA_IMAGE_READY_RES	Confirm for Image ready indication	Y	N
0x08	CMND_MSG_SUOTA_UPGRADE_COMPLETED_REQ	Success/fail on upgrading process, sent to the BASE	Y	N
0x0A	CMND_MSG_SUOTA_READ_FILE_REQ	Read Image File	Y	N
0x0B	CMND_MSG_SUOTA_READ_FILE_RES	Data of Image File	N	Y
0x0C	CMND_MSG_SUOTA_DOWNLOAD_ABORT_REQ	Abort SUOTA process and return to ULE Mode	Y	N
0x0D	CMND_MSG_SUOTA_DOWNLOAD_ABORT_CFM	Abort Operation CFM	N	Y
0x0E	CMND_MSG_SUOTA_CHECK_FOR_NEW_VERSION_REQ	Request from Node Host to check if there is new SW available	Y	N
0x0F	CMND_MSG_SUOTA_CHECK_FOR_NEW_VERSION_RES	Response on check version request with a result.	N	Y

4.8.2.1 CMND_MSG_SUOTA_NEW_SW_AVAIALBE_IND

This message is an indication sent from base to Node Host.

IE	M/O	Comment
CMND_IE_NEW_SW_INFO	M	Holds the sw / hw and URL information

4.8.2.2 CMND_MSG_SUOTA_NEW_SW_RES

This message is a response to new SW available. Host indicate a reject/confirm on new SW. The CMND sends a CMND_MSG_GENERAL_LINK_CFM message of General Service to confirm message arrival to the Base (link confirmation).

IE	M/O	Comment
CMND_IE_RESPONSE	M	Success / Fail status. See Return Codes

4.8.2.2.1 Return Codes

Return Code	Value	Description
CMND_RC_OK	0x0	General success
CMND_RC_FAIL	0x1	General failure
CMND_RC_LOW_BAT	0x2	Low Battery

4.8.2.3 CMND_MSG_SUOTA_DOWNLOAD_START_REQ

This message is a request from the device to start the software download process.

IE	M/O	Comment
CMND_IE_NEW_SW_INFO	M	Holds the sw / hw and URL information
CMND_IE_IMAGE_TYPE	M	Indicates image type FW or Node HOST

4.8.2.4 CMND_MSG_SUOTA_DOWNLOAD_START_RES

This message is a confirmation on START_REQ. This message has no payload.

4.8.2.5 CMND_MSG_SUOTA_IMAGE_READY_IND

This message is an indication on success / fail of downloading the new image

IE	M/O	Comment
CMND_IE_RESPONSE	M	Success / Fail status: 0x00 – Success 0x01 – General Error 0x02 – Session Error 0x03 – Gmep Send Error 0x04 – Write Error 0x05 – Erase Error 0x06 – Checksum Error 0x07 – Flash Data Error 0x08 – Call Ended 0x09 – HTTP Error 0x0A – Version Number Unchanged
CMND_IE_U32	M	Image size in bytes

4.8.2.6 CMND_MSG_SUOTA_IMAGE_READY_RES

This message is response from Node host to confirm receiving of CMND_MSG_SUOTA_IMAGE_READY_IND.

4.8.2.7 CMND_MSG_SUOTA_UPGRADE_COMPLETED_REQ

Success/fail on upgrading process, sent to the BASE

The receiving side of the request sends a CMND_MSG_GENERAL_LINK_CFM message of General Service to confirm message arrival (link confirmation).

IE	M/O	Comment
CMND_IE_SW_VER_INFO	M	Holds SUOTA result and current SW Version.

4.8.2.8 CMND_MSG_SUOTA_READ_FILE_REQ

This message is a request to read from the new image file downloaded.

IE	M/O	Comment
CMND_IE_READ_FILE_DATA_REQ	M	Specifies offset and length to read

4.8.2.9 CMND_MSG_SUOTA_READ_FILE_RES

This message is a response to a read request. End of file is indicated by returned size 0. Another method to detect EOF is by comparing the total number of bytes read so far vs the image size (available in payload of CMND_MSG_SUOTA_IMAGE_READY_IND message)

IE	M/O	Comment
CMND_IE_RESPONSE	M	File Read Success/Failure
CMND_IE_READ_FILE_DATA_RES	M	Data from File

4.8.2.9.1 Return Codes

Return Code	Value	Description
CMND_RC_OK	0x0	General success
CMND_RC_FAIL	0x1	General failure
CMND_RC_OFFSET_WRONG	0x2	Invalid offset
CMND_RC_NO_FILE	0x3	No available file

4.8.2.10 CMND_MSG_SUOTA_DOWNLOAD_ABORT_REQ

This message is a request to abort the file download process and return to ULE Mode. This message has no payload.

4.8.2.11 CMND_MSG_SUOTA_DOWNLOAD_ABORT_CFM

This message confirms that abort request was received and was processed. This message has no payload

4.8.2.12 CMND_MSG_SUOTA_CHECK_FOR_NEW_VERSION_REQ

This message is a request from the device to check the existence of new software.

IE	M/O	Comment
CMND_IE_CURRENT_SW_INFO	O	Holds current sw / hw information. If the element is omitted, then update for current CMND version will be requested.

4.8.2.13 CMND_MSG_SUOTA_CHECK_FOR_NEW_VERSION_RES

This message is a response for CMND_MSG_SUOTA_CHECK_FOR_NEW_VERSION_REQ. The message contains a result of version check and an information about new version.

IE	M/O	Comment
CMND_IE_CHECK_SW_VER_RES	M	Holds version check result, the sw / hw and URL information

4.9 Identify Service

4.9.1 Overview

This service is used to identify physical device/unit or a group of units by blinking or beeping the device, or to identify a device ID in the base by pushing a button on the device and indicating the unit ID on the base. This service is mostly used for installation purposes.

The following abbreviations are used in tables in this section:

N → C: Message can be sent from Node Host to CMND

C → N: Message can be sent from CMND to Node Host

M: Mandatory

O: Optional

4.9.2 Messages

MESSAGE ID	MESSAGE NAME	MESSAGE DESCRIPTION	N → C	C → N
0x01	CMND_MSG_IDENTIFY_DEVICE_REQ	Indication to device to identify itself	Y	Y
0x02	CMND_MSG_IDENTIFY_DEVICE_RES	Response for identification indication	Y	Y
0x03	CMND_MSG_IDENTIFY_SELF_REQ	The unit wishes to identify itself	Y	Y

4.9.2.1 CMND_MSG_IDENTIFY_DEVICE _REQ

This message is a request to the Node Host to identify a specific unit by physical means, such as a blinking LED or beeping. This message should be sent to Unit 0.

IE	M/O	Comment
CMND_IE_UNIT_ADDR	M	Holds the specific device and unit ID to identify
CMND_IE_U8	M	Holds the time interval during which the physical notification should be active.
CMND_IE_RESPONSE_REQUIRED	O	Include this IE if FUN request response is required

4.9.2.2 CMND_MSG_IDENTIFY_DEVICE_ RES

This message is the response from the Node Host to a CMND_MSG_IDENTIFY_DEVICE_REQ request. It should be sent to Unit 0 only if a request response was required.

IE	M/O	Comment
CMND_IE_UNIT_ADDR	M	Holds the specific unit ID to identify (unid ID = 0x0 , directs to device only)
CMND_IE_RESPONSE	M	0x0 - ID identification succeeded 0x1 - ID identification failed

4.9.2.3 CMND_MSG_IDENTIFY_SELF_REQ

This message is a request from the device side to identify itself to the BS or an explicit unit. When this message is targeted to reach a specific unit (destination is not BS), the destination unit should receive the same message. The receiving side of the request sends a CMND_MSG_GENERAL_LINK_CFM message of General Service to confirm message arrival (link confirmation). This message should be sent to Unit 0.

IE	M/O	Comment
CMND_IE_IDENTIFY	M	Holds the identifier of the device to present

4.10 General Service

The following abbreviations are used in tables in this section:

N → C: Message can be sent from Node Host to CMND
C → N: Message can be sent from CMND to Node Host
M: Mandatory
O: Optional

4.10.1 Overview

This service groups general messages and attributes.

4.10.2 Messages

MESSAGE ID	MESSAGE NAME	MESSAGE DESCRIPTION	N → C	C → N
0x01	Reserved			
0x02	Reserved			
0x03	Reserved			
0x04	Reserved			
0x05	CMND_MSG_GENERAL_HELLO_IND	First welcome message	N	Y
0x06	CMND_MSG_GENERAL_ERROR_IND	General error or failure indication	Y	Y
0x07	CMND_MSG_GENERAL_LINK_CFM	General confirmation of request delivery to BS	Y	N
0x08	CMND_MSG_GENERAL_GET_STATUS_REQ	Get the general status of CMND	Y	N
0x09	CMND_MSG_GENERAL_GET_STATUS_RES	Response for general status request	N	Y
0x0A	CMND_MSG_GENERAL_HELLO_REQ	First welcome message	Y	N
0x0B	CMND_MSG_GENERAL_GET_VERSION_REQ	Get CMND version request	Y	N
0x0C	CMND_MSG_GENERAL_GET_VERSION_RES	Response for CMND version request	N	Y
0xD	CMND_MSG_GENERAL_TRANSACTION_START_REQ	Start transaction during which CMND will not enter hibernation	Y	N
0xE	CMND_MSG_GENERAL_TRANSACTION_START_CFM	Start transaction command received by CMND and confirmed	N	Y

MESSAGE ID	MESSAGE NAME	MESSAGE DESCRIPTION	N→C	C→N
0x0F	CMND_MSG_GENERAL_TRANSACTION_END_REQ	End previous transaction, CMND now can enter hibernation when possible	Y	N
0x10	CMND_MSG_GENERAL_TRANSACTION_END_CFM	Confirmation for End transaction request	N	Y
0x11	CMND_MSG_GENERAL_LINK_MAINTAIN_START_REQ	Request to start maintaining link with base	Y	N
0x12	CMND_MSG_GENERAL_LINK_MAINTAIN_START_CFM	Confirmation for request to start maintaining link with base	N	Y
0x13	CMND_MSG_GENERAL_LINK_MAINTAIN_STOP_REQ	Request to stop maintaining link with base	Y	N
0x14	CMND_MSG_GENERAL_LINK_MAINTAIN_STOP_CFM	Confirmation for request to stop maintaining link with base	N	Y
0x15	CMND_MSG_GENERAL_LINK_MAINTAIN_STOPPED_IND	Links maintain stop indication. Timer expired.	N	Y

4.10.2.1 CMND_MSG_GENERAL_HELLO_IND

This message is sent from CMND to Node Host on startup or on wakeup from external event. Message contains CMND API version and the general status of the CMND. The status includes registration status and device ID, production or normal mode and others. This message is addressed to Unit 0.

IE	M/O	Comment
CMND_IE_GENERAL_STATUS	M	Contains general CMND status (registration, EEPROM, startup mode)
CMND_IE_VERSION	M	Contains CMND API version

4.10.2.2 CMND_MSG_GENERAL_ERROR_IND

This message is used to indicate invalid message content such as: wrong checksum, unknown service ID, unknown message ID.

IE	M/O	Comment

CMND_IE_RESPONSE	M	0x00 - Invalid cookie 0x01 - Error parameter 0x02 – Not registered 0x03 - Feature not supported 0x04 - Error checksum 0x05 - Unknown service ID 0x06 - Unknown message ID 0x07 - Invalid message length

4.10.2.3 CMND_MSG_GENERAL_LINK_CFM

This message, sent only from the CMND to the Node Host, returns the status of request delivery to BS. It is a preliminary ACK that does not confirm request execution (by application or any other higher layer on base)

IE	M/O	Comment
CMND_IE_RESPONSE	M	0x00 - Message delivery to BS succeeded 0x01- Message delivery to BS failed 0x12- Failed to lock on base. 0x13 - Failed to transmit message but succeeded to lock. See more values in CMND_IE_RESPONSE IE spec.

4.10.2.4 CMND_MSG_GENERAL_GET_STATUS_REQ

This message is used to receive the general status of the CMND. This message should be sent to Unit 0. No IE is used. This message is available in production mode.

4.10.2.5 CMND_MSG_GENERAL_GET_STATUS_RES

This message is a response for CMND_MSG_GENERAL_GET_STATUS_REQ that conveys the status of the CMND. This message is sent from Unit 0.

IE	M/O	Comment
CMND_IE_GENERAL_STATUS	M	Contains general CMND status (registration, EEPROM, startup mode, device ID)

4.10.2.6 CMND_MSG_GENERAL_HELLO_REQ

This message is an indication from the Node Host to the CMND. It indicates the Node Host is ready to receive CMND messages after power up or wakeup indication from a device. This message should be addressed to Unit 0.

4.10.2.7 CMND_MSG_GENERAL_GET_VERSION_REQ

This message is a request to get CMND version. Version type is specified in message payload. The response is CMND_MSG_GENERAL_GET_VERSION_RES message with requested version. This message should be sent to Unit 0.

IE	M/O	Comment
CMND_IE_U8	M	Define in payload of this IE the version type 0x1 – Software version 0x2 – Hardware version 0x3 – EEPROM 0x4 – CMND API

4.10.2.8 CMND_MSG_GENERAL_GET_VERSION_RES

This message is a response to get 4.11.3.11 CMND_MSG_GENERAL_GET_VERSION_REQ request. This message is sent to Unit 0

IE	M/O	Comment
CMND_IE_VERSION	M	Requested version

4.10.2.9 CMND_MSG_GENERAL_TRANSACTION_START_REQ

This message is a request from MCU to CMND to start a transaction during which the device will not enter hibernation. This message is sent to Unit 0. This message has no payload

4.10.2.10 CMND_MSG_GENERAL_TRANSACTION_START_CFM

This message is a confirmation for CMND_MSG_GENERAL_TRANSACTION_START_REQ request. The CMND confirms that transaction started. This message is sent by Unit 0. The message has no payload

4.10.2.11 CMND_MSG_GENERAL_TRANSACTION_END_REQ

This message is a request from MCU to CMND to end a previously started transaction. This message is sent to Unit 0. This message has no payload

4.10.2.12 CMND_MSG_GENERAL_TRANSACTION_END_CFM

This message is a confirmation for CMND_MSG_GENERAL_TRANSACTION_END_REQ message. The CMND confirms that “end transaction” request was received and handled. After this confirmation the hibernation is no more blocked and CMND may enter hibernation when possible. This message is sent to Unit 0. This message has no payload.

4.10.2.13 CMND_MSG_GENERAL_LINK_MAINTAIN_START_REQ

This message is a request from MCU to CMND to start link maintain with base. This message is to be used when a response is required from base. A general use case is when MCU sends a command to base, and now should wait for a response from base. This requires a periodic message from CMND towards the base to allow the base to respond.
The CMND will transmit empty packet to Base every period (specified in message payload). Link maintain duration is also specified in message payload. This message is sent to Unit 0.
For a sequence flows refer to Using Link Maintain commands and Link Maintain expires.

IE	M/O	Comment
CMND_IE_LINK_MAINTAIN	M	Specified link maintain duration in ping interval. Duration - the total time that CMND will maintain the link. Interval - defines frequency of empty messages sent to base.

4.10.2.14 CMND_MSG_GENERAL_LINK_MAINTAIN_START_CFM

This message is a confirmation for CMND_MSG_GENERAL_LINK_MAINTAIN_START_REQ. The CMND confirms that link maintain has successfully started or failed. This message is sent by Unit 0.

IE	M/O	Comment
CMND_IE_RESPONSE	M	0x00 – Success 0x01 - Failure

4.10.2.15 CMND_MSG_GENERAL_LINK_MAINTAIN_STOP_REQ

This message is a request from MCU to CMND to stop existing link with base. Send this message when link is not required anymore for example when MCU has received the response from base. This message is sent by Unit 0. This message has no payload.

4.10.2.16 CMND_MSG_GENERAL_LINK_MAINTAIN_STOP_CFM

This message is a confirmation for CMND_MSG_GENERAL_LINK_MAINTAIN_STOP_REQ, and indicates that link maintenance has stopped. This message is sent by Unit 0.

IE	M/O	Comment
CMND_IE_RESPONSE	M	0x00 – Success 0x01 - Failure

4.10.2.17 CMND_MSG_GENERAL_LINK_MAINTAIN_STOPPED_IND

This message is a message from CMND to MCU that indicates that link maintain timer expired and not stopped explicitly by sending CMND_MSG_GENERAL_LINK_MAINTAIN_STOP_REQ message. This message is sent by Unit 0. This message has no payload

4.11 Alert Service

The following abbreviations are used in tables in this section:

N → C: Message can be sent from Node Host to CMND
C → N: Message can be sent from CMND to Node Host
M: Mandatory
O: Optional

4.11.1 Overview

This service is used by any device to send alerts.

4.11.2 Messages

MESSAGE ID	MESSAGE NAME	MESSAGE DESCRIPTION	N → C	C → N
0x01	CMND_MSG_ALERT_GET_ATTRIB_REQ	Read service attribute	Y	Y
0x02	CMND_MSG_ALERT_GET_ATTRIB_RES	Response for ‘GET’ attribute request	Y	Y
0x03	CMND_MSG_ALERT_NOTIFY_STATUS_REQ	Indication on Alert	Y	N
0x04	CMND_MSG_ALERT_NOTIFY_STATUS_RES	Response for status notification	N	Y

4.11.2.1 CMND_MSG_ALERT_GET_ATTRIB_REQ

This functionality is not supported.

This message is used to read unit attributes. The receiving side of the request sends a CMND_MSG_GENERAL_LINK_CFM message of General Service to confirm message arrival (link confirmation).

IE	M/O	Comment
CMND_IE_ATTRIBUTE_ID	M	Service attribute to read
CMND_IE_UNIT_ADDR	O	Include this IE when destination device is not BS

4.11.2.2 CMND_MSG_ALERT_GET_ATTRIB_RES

This functionality is not supported.

This is a response to CMND_MSG_ALERT_GET_ATTRIB_REQ.

IE	M/O	Comment
CMND_IE_RESPONSE	M	0x0 - Read succeeded 0x1 - Read failed
CMND_IE_ATTRIBUTE_VALUE	M	Holds attribute value in case of success
CMND_IE_UNIT_ADDR	O	Mandatory if was present in the request

4.11.2.3 CMND_MSG_ALERT_NOTIFY_STATUS_REQ

This message is used to halt an alert. The receiving side of the request sends a CMND_MSG_GENERAL_LINK_CFM message of General Service to confirm message arrival (link confirmation).

IE	M/O	Comment
CMND_IE_ALERT	M	Contains alert source
CMND_IE_RESPONSE_REQUIRED	O	Include this IE if response is required

4.11.2.4 CMND_MSG_ALERT_NOTIFY_STATUS_RES

This message acknowledges a CMND_MSG_ALERT_NOTIFY_STATUS_REQ request. This message should be sent only if CMND_IE_RESPONSE_REQUIRED is included into the request.

IE	M/O	Comment
CMND_IE_RESPONSE	M	0x0 - Response to “Notify Status” request succeeded 0x1 - Response to “Notify Status” request failed

4.12 Tamper Alert Service

The following abbreviations are used in tables in this section:

N → C: Message can be sent from Node Host to CMND
C → N: Message can be sent from CMND to Node Host
M: Mandatory
O: Optional

4.12.1 Overview

This service is used by any device to send tamper alerts.

4.12.2 Messages

MESSAGE ID	MESSAGE NAME	MESSAGE DESCRIPTION	N → C	C → N
0x01	CMND_MSG_TAMPER_ALERT_GET_ATTRIB_REQ	Read service attribute	Y	Y
0x02	CMND_MSG_TAMPER_ALERT_GET_ATTRIB_RES	Response for ‘GET’ attribute request	Y	Y
0x03	CMND_MSG_TAMPER_ALERT_NOTIFY_STATUS_REQ	Indication on tamper alert	Y	N
0x04	CMND_MSG_TAMPER_ALERT_NOTIFY_STATUS_RES	Response for the Notify Status request	N	Y

4.12.2.1 CMND_MSG_TAMPER_ALERT_GET_ATTRIB_REQ

This functionality is not supported.

This message is used to read a unit attribute. The receiving side of the request sends a CMND_MSG_GENERAL_LINK_CFM message of General Service to confirm message arrival (link confirmation).

IE	M/O	Comment
CMND_IE_ATTRIBUTE_ID	M	Service attribute to read
CMND_IE_UNIT_ADDR	O	Include this IE when destination device is not BS

4.12.2.2 CMND_MSG_TAMPER_ALERT_GET_ATTRIB_RES

This functionality is not supported.

This is a response to CMND_MSG_TAMPER_ALERT_GET_ATTRIB_REQ.

IE	M/O	Comment
CMND_IE_RESPONSE	M	0x0 - Read request succeeded 0x1 - Read request failed
CMND_IE_ATTRIBUTE_VALUE	M	Holds attribute value in case of success
CMND_IE_UNIT_ADDR	O	Mandatory if was present in the request

4.12.2.3 CMND_MSG_TAMPER_ALERT_NOTIFY_STATUS_REQ

This message is used for notifying the CMND of a tamper alert. The receiving side of the request sends a CMND_MSG_GENERAL_LINK_CFM message of General Service to confirm message arrival (link confirmation).

IE	M/O	Comment
CMND_IE_TAMPER_ALERT	M	Contains alert source
CMND_IE_RESPONSE_REQUIRED	O	Include this IE if response is required

4.12.2.4 CMND_MSG_TAMPER_ALERT_NOTIFY_STATUS_RES

This message is the required response for a status notification request. The message confirms if the notification reached the BS. This message should be sent only if CMND_IE_RESPONSE_REQUIRED is present in the alert notification message.

IE	M/O	Comment
CMND_IE_RESPONSE	M	0x0 - Response to “Notify Status” request succeeded 0x1 - Response “Notify Status” request failed

4.13 Keep Alive Service

The following abbreviations are used in tables in this section:

N → C: Message can be sent from Node Host to CMND

C → N: Message can be sent from CMND to Node Host

M: Mandatory

O: Optional

4.13.1 Overview

This service is used by any device to send keep alive.

4.13.2 Attributes

ATTRIBUTE ID	ATTRIBUTE NAME	ATTRIBUTE TYPE	ATTRIBUTE VALUES	ATTRIBUTE ACCESS
0x0001	Interval	U32	milliseconds	RW

4.13.3 Messages

MESSAGE ID	MESSAGE NAME	MESSAGE DESCRIPTION	N→C	C→N
0x01	CMND_MSG_KEEP_ALIVE_I_AM_ALIVE_REQ	Indicates that device is alive	Y	N
0x02	Reserved			
0x03	CMND_MSG_KEEP_ALIVE_GET_ATTRIB_REQ	Request from Base to read keep alive attribute	N	Y
0x04	CMND_MSG_KEEP_ALIVE_GET_ATTRIB_RES	Response for ‘GET’ attribute request	Y	N
0x05	CMND_MSG_KEEP_ALIVE_SET_ATTRIB_REQ	Request from Base to set keep alive attribute	Y	N
0x06	CMND_MSG_KEEP_ALIVE_SET_ATTRIB_RES	Response for ‘SET’ attribute request	Y	N

4.13.3.1 CMND_MSG_KEEP_ALIVE_I_AM_ALIVE_REQ

This message is used to indicate that the device is alive. The receiving side of the request sends a CMND_MSG_GENERAL_LINK_CFM message of General Service to confirm message arrival (link confirmation). No IE is used.

4.13.3.2 CMND_MSG_ KEEP_ALIVE_GET_ATTRIB_REQ

This message is indicated from remote base to read the value of a keep alive attribute on node host. Attribute details are specified in message payload. This message should be used only when keep alive is managed by the Node Host and not on the CMND.

IE	M/O	Comment
CMND_IE_ATTRIBUTE_ID	M	ID of the attribute to be read
CMND_IE_UNIT_ADDR	O	Include this IE when destination device is not BS

4.13.3.3 CMND_MSG_ KEEP_ALIVE_GET_ATTRIB_RES

This is a response to CMND_MSG_ KEEP_ALIVE_GET_ATTRIB_REQ. This message should be used only when keep alive is managed by the Node Host and not on the CMND.

IE	M/O	Comment
CMND_IE_ATTRIBUTE_VALUE	M	Must be present to understand which attribute (even in case of failure). In case of failure only attribute ID and result fields are filled.
CMND_IE_UNIT_ADDR	O	Mandatory if present in the request

4.13.3.4 CMND_MSG_KEEP_ALIVE_SET_ATTRIB_REQ

This message is request from remote base to set a keep alive attribute on node host. This message should be used only when keep alive is managed by the Node Host and not on the CMND.

IE	M/O	Comment
CMND_IE_ATTRIBUTE_VALUE	M	Attribute ID and new value
CMND_IE_RESPONSE_REQUIRED	O	Include this IE when end-to-end FUN response is required
CMND_IE_UNIT_ADDR	O	Mandatory if was present in the request

4.13.3.5 CMND_MSG_ KEEP_ALIVE_SET_ATTRIB_RES

This message is a response to CMND_MSG_KEEP_ALIVE_SET_ATTRIB_REQ message. Node host should store the value received in “SET” request and confirm the operation. This message is only mandatory when CMND_IE_RESPONSE_REQUIRED IE is present in the “SET” request. This message should be used only when keep alive is managed by the Node Host and not on the CMND.

IE	M/O	Comment
CMND_IE_RESPONSE	O	0x0 - Request handling succeeded 0x3 - Request Not supported 0x4 – Failed to write to read only attribute 0xFF – Unknown reason Response is Mandatory only if CMND_IE_RESPONSE_REQUIRED is available in the “SET” request
CMND_IE_ATTRIBUTE_ID	O	The ID of the modified attribute. This IE is mandatory if response is required
CMND_IE_UNIT_ADDR	O	Present only if in the request

4.14 On-Off Service

The following abbreviations are used in tables in this section:

N → C: Message can be sent from Node Host to CMND
C → N: Message can be sent from CMND to Node Host
M: Mandatory
O: Optional

4.14.1 Overview

This service is used for switching devices on and off.

4.14.2 Messages

MESSAGE ID	MESSAGE NAME	MESSAGE DESCRIPTION	N → C	C → N
0x01	CMND_MSG_ONOFF_ON_REQ	Switch unit on	Y	Y
0x02	CMND_MSG_ONOFF_ON_RES	Response to Switch On request	Y	Y
0x03	CMND_MSG_ONOFF_OFF_REQ	Set unit off	Y	Y
0x04	CMND_MSG_ONOFF_OFF_RES	Response to Switch Off request	Y	Y
0x05	CMND_MSG_ONOFF_TOGGLE_REQ	Toggle unit status	Y	Y
0x06	CMND_MSG_ONOFF_TOGGLE_RES	Response to Toggle request	Y	Y
0x07	CMND_MSG_ONOFF_GET_ATTRIB_REQ	Get unit attribute	Y	Y
0x08	CMND_MSG_ONOFF_GET_ATTRIB_RES	Response for 'GET' attribute request with attribute data	Y	Y
0x09	CMND_MSG_ONOFF_SET_ATTRIB_REQ	Set attribute request	Y	Y
0x0A	CMND_MSG_ONOFF_SET_ATTRIB_RES	Response for "Set Attribute" request	Y	Y
0x0B	CMND_MSG_ONOFF_SET_ATTRIB_REQ_WITH_RES	Set attribute with response	Y	Y

4.14.2.1 CMND_MSG_ONOFF_ON_REQ

This message is used to switch on the device. This message is available for two directions:

MCU → CMND: Button is pressed to turn on a remote switchable device. The CMND will respond with CMND_MSG_GENERAL_LINK_CFM message of General Service to confirm message arrival (link confirmation).

CMND → MCU: Switchable device receives a command from remote device or base to turn the MCU on.

IE	M/O	Comment
CMND_IE_UNIT_ADDR	O	Include this IE when destination device is not BS
CMND_IE_RESPONSE_REQUIRED	O	Include this IE when end to end FUN response is required

4.14.2.2 CMND_MSG_ONOFF_ON_RES

This message is the response to CMND_MSG_ONOFF_ON_REQ. This message is sent only if the request contained CMND_IE_RESPONSE_REQUIRED. This message is available for two directions:

MCU → CMND: Switchable device responses for previous request to turn on the device

CMND → MCU: Remote device responses for previous request from MCU to turn it on.

IE	M/O	Comment
CMND_IE_RESPONSE	M	0x0 - Request handling succeeded 0x1 - Request handling failed
CMND_IE_UNIT_ADDR	O	Present only if in the request

4.14.2.3 CMND_MSG_ONOFF_OFF_REQ

This message is used to switch off the device. This message is available for two directions:

MCU → CMND: Button is pressed to turn off a remote switchable device. The CMND will respond with CMND_MSG_GENERAL_LINK_CFM message of General Service to confirm message arrival (link confirmation).

CMND → MCU: Switchable device (MCU) receives a command from remote device or base to turn the MCU off.

IE	M/O	Comment
CMND_IE_UNIT_ADDR	O	Include this IE when destination device is not BS
CMND_IE_RESPONSE_REQUIRED	O	Include this IE when end to end FUN response is required

4.14.2.4 CMND_MSG_ONOFF_OFF_RES

This message is the response to CMND_MSG_ONOFF_OFF_REQ. It is sent only if the request contained CMND_IE_RESPONSE_REQUIRED. The message is available for two directions:

MCU → CMND: Switchable device responses for previous request to turn off the device

CMND → MCU: Remote device responses for previous request from MCU to turn it off.

IE	M/O	Comment
CMND_IE_RESPONSE	M	0x0 - Request handling succeeded 0x1 - Request handling failed
CMND_IE_UNIT_ADDR	O	Present only if in the request

4.14.2.5 CMND_MSG_ONOFF_TOGGLE_REQ

This message is used to toggle the state of the device. This message is available for two directions:

MCU → CMND: Button is pressed to toggle the state of a remote switchable device. The CMND will respond with CMND_MSG_GENERAL_LINK_CFM message of General Service to confirm message arrival (link confirmation).

CMND → MCU: Switchable device (MCU) receives a command from remote device or base to toggle MCU state

IE	M/O	Comment
CMND_IE_UNIT_ADDR	O	Include this IE when destination device is not BS
CMND_IE_RESPONSE_REQUIRED	O	Include this IE when end to end FUN response is required

4.14.2.6 CMND_MSG_ONOFF_TOGGLE_RES

This message is the response to CMND_MSG_ONOFF_TOGGLE_REQ. It is sent only if the request contained CMND_IE_RESPONSE_REQUIRED. The message is available for two directions:

MCU → CMND: Switchable device responses for previous request to toggle the state of the device

CMND → MCU: Remote device responses for previous request from MCU to toggle the state.

IE	M/O	Comment
CMND_IE_RESPONSE	M	0x0 - Request handling succeeded 0x1 - Request handling failed
CMND_IE_UNIT_ADDR	O	Present only if in the request

4.14.2.7 CMND_MSG_ONOFF_GET_ATTRIB_REQ

This message is used to read a unit attribute. This message is available for two directions:

MCU → CMND: MCU sends a request to get attribute from remote device. The CMND will respond with CMND_MSG_GENERAL_LINK_CFM message of General Service to confirm message arrival (link confirmation).

CMND → MCU: Remote device (or base) sends a request to MCU to get attribute

IE	M/O	Comment
CMND_IE_ATTRIBUTE_ID	M	Specified the attribute to be read
CMND_IE_UNIT_ADDR	O	Include this IE when destination device is not BS

4.14.2.8 CMND_MSG_ONOFF_GET_ATTRIB_RES

This is a response to CMND_MSG_ONOFF_GET_ATTRIB_REQ. This message is available for two directions:

MCU → CMND: MCU sends a response with attribute value to remote device. The CMND will respond with CMND_MSG_GENERAL_LINK_CFM message of General Service to confirm message arrival (link confirmation).

CMND → MCU: Remote device (or base) sends a response with attribute value to previous request that was sent from MCU to this remote device.

IE	M/O	Comment
CMND_IE_ATTRIBUTE_VALUE	M	Contains attribute value. In case of failure Attribute ID should be equal to the ID in the 'GET' request, attribute value should be ignored. Response Value: 0x0 - Request handling succeeded 0x3 - Fail: Not supported 0xFF – Fail: Unknown reason
CMND_IE_UNIT_ADDR	O	Mandatory if in the request

4.14.2.9 CMND_MSG_ONOFF_SET_ATTRIB_REQ

This message is used to set a unit attribute. This message is available for two directions:

MCU → CMND: MCU sends a request to set attribute on remote device. The CMND will respond with CMND_MSG_GENERAL_LINK_CFM message of General Service to confirm message arrival (link confirmation).

CMND → MCU: Remote device (or base) sends a request to MCU to set attribute on the MCU

IE	M/O	Comment
CMND_IE_SET_ATTRIBUTE_VALUE	M	Specified target attribute and the new value to set
CMND_IE_UNIT_ADDR	O	Include this IE when destination device is not BS

4.14.2.10 CMND_MSG_ONOFF_SET_ATTRIB_RES

This is a response to CMND_MSG_ONOFF_SET_ATTRIB_REQ. This message is available for two directions:

MCU → CMND: MCU sends a response for previously received "Set" attribute message. The CMND will respond with CMND_MSG_GENERAL_LINK_CFM message of General Service to confirm message arrival (link confirmation).

CMND → MCU: Remote device (or base) sends a response for previous request from MCU to set the attribute on remote device

IE	M/O	Comment
CMND_IE_RESPONSE	M	0x0 - Request handling succeeded 0x3 - Fail: Not supported 0x4 – Fail: Read only 0xFF – Fail: Unknown reason
CMND_IE_ATTRIBUTE_ID	M	Identify the attribute which was set
CMND_IE_UNIT_ADDR	O	Mandatory if in the request

4.14.2.11 CMND_MSG_ONOFF_SET_ATTRIB_REQ_WITH_RES

This message is used to set a unit attribute and requires a response from destination (as opposed to CMND_MSG_ONOFF_SET_ATTRIB_REQ where response is optional). The response is CMND_MSG_ONOFF_SET_ATTRIB_RES message.

This message is available for two directions:

MCU → CMND: MCU sends a request to set attribute on remote device. The CMND will respond with CMND_MSG_GENERAL_LINK_CFM message of General Service to confirm message arrival (link confirmation).

CMND → MCU: Remote device (or base) sends a request to MCU to set attribute on the MCU

IE	M/O	Comment
CMND_IE_SET_ATTRIBUTE_VALUE	M	Specified target attribute and the new value to set
CMND_IE_UNIT_ADDR	O	Include this IE when destination device is not BS

4.15 FUN Service

The following abbreviations are used in tables in this section:

N → C: Message can be sent from Node Host to CMND

C → N: Message can be sent from CMND to Node Host

M: Mandatory

O: Optional

4.15.1 Overview

This service enables sending/receiving a full FUN OTA message according to FUN standard [1].

4.15.2 Messages

MESSAGE ID	MESSAGE NAME	MESSAGE DESCRIPTION	N → C	C → N
0x01	CMND_MSG_FUN_SEND_REQ	Send FUN message request	Y	N
0x02	CMND_MSG_FUN_RECV_IND	Indication on received FUN message	N	Y

4.15.2.1 CMND_MSG_FUN_SEND_REQ

This message is used to send a full FUN over-the-air message to a destination device and unit. The *Data* field of CMND API message should contain the exact FUN protocol message based on the FUN OTA protocol specification. The receiving side of the request sends a CMND_MSG_GENERAL_LINK_CFM message of General Service to confirm message arrival (link confirmation).

IE	M/O	Comment
CMND_IE_FUN	M	FUN data container

4.15.2.2 CMND_MSG_FUN_RECV_IND

This message indicates incoming full FUN OTA message from device.

IE	M/O	Comment
CMND_IE_FUN	M	FUN message data container

4.16 System Service

The following abbreviations are used in tables in this section:

N → C: Message can be sent from Node Host to CMND

C → N: Message can be sent from CMND to Node Host

M: Mandatory

O: Optional

4.16.1 Overview

This service provides miscellaneous system functionality, such as battery measurements, accessing specific HW registers, etc.

4.16.2 Messages

MESSAGE ID	MESSAGE NAME	MESSAGE DESCRIPTION	N→C	C→N
0x01	CMND_MSG_SYS_BATTERY_MEASURE_GET_REQ	Get measurements of battery from A2D on CMND	Y	N
0x02	CMND_MSG_SYS_BATTERY_MEASURE_GET_RES	Response for 'GET' parameter request	N	Y
0x03	CMND_MSG_SYS_RSSI_GET_REQ	Get RSSI measurement (link quality)	Y	N
0x04	CMND_MSG_SYS_RSSI_GET_RES	Response for Get RSSI request	N	Y
0x05	CMND_MSG_SYS_BATTERY_IND_ENABLE_REQ	Enable low battery indication	Y	N
0x06	CMND_MSG_SYS_BATTERY_IND_DISABLE_REQ	Disable low battery indication	Y	N
0x07	CMND_MSG_SYS_BATTERY_IND_LOW_IND	Send low battery notification	N	Y
0x08	CMND_MSG_SYS_RESET_REQ	Force watchdog reset CMND node	Y	N
0x09	CMND_MSG_SYS_BATTERY_END_LIFE_IND	Indication on end of battery life	N	Y

4.16.2.1 CMND_MSG_SYS_BATTERY_MEASURE_GET_REQ

Request the CMND to perform battery measurement. Battery is measured during system load mainly during RF transmission. The response is CMND_MSG_SYS_BATTERY_MEASURE_GET_RES with the measured data. This message should be sent to Unit 0.

IE	M/O	Comment
CMND_IE_BATTERY_MEASURE_INFO	M	Define battery measurement info. This command can be used in production mode only if Measurement mode of this IE is set to 2 (measure now)

4.16.2.2 CMND_MSG_SYS_BATTERY_MEASURE_GET_RES

This message returns the battery measurement result. The resolution is 100 mVolts. The battery is measured from A2D during system load. This message should be sent from Unit 0.

IE	M/O	Comment
CMND_IE_BATTERY_LEVEL	M	The payload of the IE is battery measurements in resolution

		of 100 mVolts. If measured voltage is below 1.5V the value 0 is returned

4.16.2.3 CMND_MSG_SYS_RSSI_GET_REQ

Request the CMND to return RSSI measurement. This message should be sent to Unit 0. No IE is used.

4.16.2.4 CMND_MSG_SYS_RSSI_GET_RES

This message is a response to RSSI measurement request. The payload contains RSSI measurement in recent RF activity. This message is sent from Unit 0.

IE	M/O	Comment
CMND_IE_U8	M	The payload of the IE is RSSI value

4.16.2.5 CMND_MSG_SYS_BATTERY_IND_ENABLE_REQ

This message enables special low battery indication using the sensor conditioning IO’s on the DHX91 module. The receiving side of the request sends a CMND_MSG_GENERAL_LINK_CFM message of General Service to confirm message arrival (link confirmation). NOTE – this is a not common battery measurement. Before using this feature please advise with a DSP Group representative.

4.16.2.6 CMND_MSG_SYS_BATTERY_IND_DISABLE_REQ

This message disables the battery indication using the sensor conditioning on DHX91. The receiving side of the request sends a CMND_MSG_GENERAL_LINK_CFM message of General Service to confirm message arrival (link confirmation).

4.16.2.7 CMND_MSG_SYS_BATTERY_IND_LOW_IND

This message is an indication that battery is low using the sensor conditioning IOs on DHX91.

4.16.2.8 CMND_MSG_SYS_RESET_REQ

This message is used to request the CMND to force watchdog reset. CMND node will restart after this command. No response is generated.

4.16.2.9 CMND_MSG_SYS_BATTERY_END_LIFE_IND

This is an indication on battery reaching the end of life. End of life is defined when to conditions are met:

- Battery measurement during load is below 2.0V
- Battery type is Alkaline

4.17 Parameters Service

The following abbreviations are used in tables in this section:

N → C: Message can be sent from Node Host to CMND
C → N: Message can be sent from CMND to Node Host
M: Mandatory
O: Optional

4.17.1 Overview

This service retrieves/sets the CMND API parameters. This service can be used in production or normal mode.

4.17.2 Messages

MESSAGE ID	MESSAGE NAME	MESSAGE DESCRIPTION	N→C	C→N
0x01	CMND_MSG_PARAM_GET_REQ	Read parameter from CMND	N	Y
0x02	CMND_MSG_PARAM_GET_RES	Response for ‘GET’ parameter request	N	Y
0x03	CMND_MSG_PARAM_SET_REQ	Modify CMND parameter	Y	N
0x04	CMND_MSG_PARAM_SET_RES	Response for ‘SET’ parameter request	N	Y
0x05	CMND_MSG_PARAM_GET_DIRECT_REQ	Direct read access to CMND EEPROM/RAM memory	N	Y
0x06	CMND_MSG_PARAM_GET_DIRECT_RES	Response for direct access request	N	Y
0X07	CMND_MSG_PARAM_SET_DIRECT_REQ	Direct write access to CMND EEPROM/RAM memory	Y	N
0x08	CMND_MSG_PARAM_SET_DIRECT_RES	Success/Failure response from write access request	N	Y

4.17.2.1 CMND_MSG_PARAM_GET_REQ

This message is used to read the value of CMND API parameter. The CMND API parameters are stored in CMND EEPROM and accessed by logical id without creating a dependency on the physical location of the parameters in the EEPROM. After the CMND processes this request, it sends a CMND_MSG_PARAM_GET_RES message with requested data. This message should be sent to Unit 0. See the list of CMND API parameters.

IE	M/O	Comment
CMND_IE_PARAMETER	M	Defines the parameter to read. ‘Data length’ and ‘Data‘ are ignored in this request.

4.17.2.2 CMND_MSG_PARAM_GET_RES

This message is the response to CMND_MSG_PARAM_GET_REQ request. If successful, this message contains the requested data in CMND_IE_PARAMETER.

IE	M/O	Comment
CMND_IE_PARAMETER	M	Contains the requested data

CMND_IE_RESPONSE	M	0x0 - 'Get' request succeeded 0x1 – 'Get' request failed

4.17.2.3 CMND_MSG_PARAM_SET_REQ

This message is used to set a CMND parameter. The parameters are stored in CMND EEPROM and accessed by logical id. After the CMND processes this request, it sends a CMND_MSG_PARAM_SET_RES messages with Ok or Fail response. See the list of CMND API parameters.

IE	M/O	Comment
CMND_IE_PARAMETER	M	Defines the parameter to write.

4.17.2.4 CMND_MSG_PARAM_SET_RES

This message is the response to CMND_MSG_PARAM_SET_REQ. The response should contain the ID of the modified parameter without the data.

IE	M/O	Comment
CMND_IE_PARAMETER	M	Only next fields are filled in the response based on the values passed in the "Set". • Parameter Type • Parameter ID Parameter length is set to zero.
CMND_IE_RESPONSE	M	0x0 - 'Set' request succeeded 0x1 - 'Set' request failed

4.17.2.5 CMND_MSG_PARAM_GET_DIRECT_REQ

This message gets direct read access to CMND EEPROM or RAM memory. After the CMND processes this request, it sends a CMND_MSG_PARAM_GET_DIRECT_RES messages with requested data.

IE	M/O	Comment
CMND_IE_PARAMETER_DIRECT	M	Defines memory address and length to read. For EEPROM type, the Offset value defines the offset in EEPROM from which data should be read. For RAM type, the Offset defines the physical address in RAM from which to read. The Data Length field defines number of bytes to get **For DAIF type length must be 4 bytes exactly**.

4.17.2.6 CMND_MSG_PARAM_GET_DIRECT_RES

This message is the response to CMND_MSG_PARAM_GET_DIRECT_REQ request. If successful, this message contains the requested data in CMND_IE_PARAMETER_DIRECT.

IE	M/O	Comment
CMND_IE_PARAMETER_DIRECT	M	Contains the requested data
CMND_IE_RESPONSE	M	0x00 - 'Get' request succeeded 0x01 - 'Get' request failed 0x0A – Invalid size received

4.17.2.7 CMND_MSG_PARAM_SET_DIRECT_REQ

This message is used to write directly to CMND EEPROM or RAM memory. After the CMND processes this request, it sends a CMND_MSG_PARAM_SET_DIRECT_RES messages with Ok or Fail response.

IE	M/O	Comment
CMND_IE_PARAMETER_DIRECT	M	Defines the parameter to write. **For DAIF type, size can be 1, 2 or 4 bytes only**

4.17.2.8 CMND_MSG_PARAM_SET_DIRECT_RES

This message is the response to CMND_MSG_PARAM_SET_DIRECT_RES. The response should contain the identification of the modified parameter without the data.

IE	M/O	Comment
CMND_IE_PARAMETER_DIRECT	M	Only e_AddressType and u32_Offset shall be filled
CMND_IE_RESPONSE	M	0x00 - 'Set' request succeeded 0x01 - 'Set' request failed 0x0A – Invalid size received

4.18 Sleep Service

The following abbreviations are used in tables in this section:

N → C: Message can be sent from Node Host to CMND

C → N: Message can be sent from CMND to Node Host

M: Mandatory

O: Optional

4.18.1 Overview

This service is used for allowing the CMND module to enter low power hibernation mode. The CMND module supports several sleep modes:

1. None – the CMND will never enter hibernation. This relevant to "Always On" devices such as AC plugins.
2. Locally – the CMND handles hibernation internally and will decide on its own when to enter hibernation.
3. MCU control – the CMND will enter hibernation only when MCU sends an explicit sleep request.

These modes are controlled by CMND API parameter "System off mode" (0x16). The Node Host informs the CMND that the CMND can now go to sleep/hibernate/power off. The CMND shall confirm the receiving of this indication.

Notice: in some configuration the CMND may send the CMND_MSG_SLEEP_ENTER_SLEEP_REQ as an indication that CMND is now entering hibernation.

4.18.2 Messages

MESSAGE ID	MESSAGE NAME	MESSAGE DESCRIPTION	N→C	C→N
0x01	CMND_MSG_SLEEP_ENTER_SLEEP_REQ	Enter Sleep mode	Y	Y
0x02	CMND_MSG_SLEEP_ENTER_SLEEP_CFM	Confirmation of Sleep mode request	N	Y

4.18.2.1 CMND_MSG_SLEEP_ENTER_SLEEP_REQ

This message is used to notify the CMND that it is now allowed to enter hibernation (aka, low power mode, sleep mode). The Node Host cannot send the CMND API messages after this message until a proper wake up of CMND is applied. This message should be sent to Unit 0.

4.18.2.2 CMND_MSG_SLEEP_ENTER_SLEEP_CFM

This message is a confirmation for CMND_MSG_SLEEP_ENTER_SLEEP_REQ. The CMND only confirms that the request was received and is being handled. It does not imply if the CMND has entered sleep mode. No IE is used.

4.19 Production Service

The following abbreviations are used in tables in this section:

N → C: Message can be sent from Node Host to CMND

C → N: Message can be sent from CMND to Node Host

M: Mandatory

O: Optional

4.19.1 Overview

This service is used for production settings via CMND API. Requests (xxx_REQ) receive a default response from the CMND Production Service - the CMND_MSG_PROD_CFM message. If this is insufficient, an explicit response is added in the relevant request section.

4.19.2 Messages

MESSAGE ID	MESSAGE NAME	MESSAGE DESCRIPTION	N → C	C → N
0x01	CMND_MSG_PROD_START_REQ	Set Production mode enable – needs a restart	N	Y
0x02	CMND_MSG_PROD_END_REQ	Exit Production mode – needs a restart	Y	N
0x03	CMND_MSG_PROD_CFM	General confirm on a request, applies to all messages	N	Y
0x04	CMND_MSG_PROD_REF_CLK_TUNE_START_REQ	Start 13.824 MHz reference clock tuning	Y	N
0x05	CMND_MSG_PROD_REF_CLK_TUNE_END_REQ	End 13.824 MHz reference clock tuning	Y	N
0x06	CMND_MSG_PROD_REF_CLK_TUNE_END_RES	Confirm on end process with new value returned	N	Y
0x07	CMND_MSG_PROD_REF_CLK_TUNE_ADJ_REQ	Adjust up/down request	Y	N
0x08	CMND_MSG_PROD_BG_REQ	Band gap calibration	Y	N
0x09	CMND_MSG_PROD_BG_RES	Band gap calibration res with ADC and POR returned	N	Y
0x0A	CMND_MSG_PROD_ATE_INIT_REQ	ATE Test initialize request	Y	N
0x0B	CMND_MSG_PROD_ATE_STOP_REQ	ATE Test stop request – for all modes (continuous /Rx/Tx)	Y	N
0x0C	CMND_MSG_PROD_ATE_CONTINUOUS_START_REQ	Continuous request	Y	N
0x0D	CMND_MSG_PROD_ATE_RX_START_REQ	Rx Slot Test request status	Y	N
0x0E	CMND_MSG_PROD_ATE_RX_START_RES	Rx Slot Test request - with value measured, returned multiple	N	Y

MESSAGE ID	MESSAGE NAME	MESSAGE DESCRIPTION	N→C	C→N
		times		
0x0F	CMND_MSG_PROD_ATE_TX_START_REQ	Tx Slot Test request	Y	N
0x10	CMND_MSG_PROD_ATE_GET_BER_FER_REQ	Get BER FER value when Rx Test is ongoing	Y	N
0x11	CMND_MSG_PROD_INIT_EEPROM_DEF_REQ	Initialize EEPROM to default values (select 9 in online menu)	Y	N
0x12	CMND_MSG_PROD_SPECIFIC_PRESET_REQ	Set a specific preset	Y	N
0x13	CMND_MSG_PROD_SLEEP_REQ	Set to Hibernation mode	Y	N
0x14	CMND_MSG_PROD_SET_SIMPLE_GPIO_LOW	Set simple GPIO to low	Y	N
0x15	CMND_MSG_PROD_SET_SIMPLE_GPIO_HIGH	Set simple GPIO to high	Y	N
0x16	CMND_MSG_PROD_GET_SIMPLE_GPIO_STATE	Get simple GPIO's state	Y	N
0x17	CMND_MSG_PROD_GET_SIMPLE_GPIO_STATE_RES	Get simple GPIO's state response	N	Y
0x18	CMND_MSG_PROD_SET_ULE_GPIO_LOW	Set ULE GPIO to low	Y	N
0x19	CMND_MSG_PROD_SET_ULE_GPIO_HIGH	Set ULE GPIO to high	Y	N
0x1A	CMND_MSG_PROD_GET_ULE_GPIO_STATE	Get ULE GPIO's state	Y	N
0x1B	CMND_MSG_PROD_GET_ULE_GPIO_STATE_RES	Get ULE GPIO's state response	N	Y
0x1C	CMND_MSG_PROD_SET_ULE_GPIO_DIR_INPUT_REQ	Configure ULE GPIO to input direction	Y	N
0x1D	CMND_MSG_PROD_RESET_HAN_EEPROM	Reset HAN EEPROM	Y	N
0x1E	CMND_MSG_PROD_FW_UPDATE_REQ	Perform FW update over UART	Y	N
0x1F	CMND_MSG_PROD_GPIO_LOOPBACK_TEST_REQ	GPIOs loopback test	Y	N
0x20	CMND_MSG_PROD_ATE_RX_LOCKING_START_REQ	RX Locking test	Y	N

4.19.2.1 CMND_MSG_PROD_START

Notify the CMND that Production mode should be entered. The device **must be restarted** for this action to take effect. No IE is used.

Response: generic CMND_MSG_PROD_CFM

4.19.2.2 CMND_MSG_PROD_END

Notify the CMND that Production mode should be ended. The device **must be restarted** for this action to take effect.

Response: see CMND_MSG_PROD_CFM. No IE is used.

4.19.2.3 CMND_MSG_PROD_CFM

This is a generic message used to return a response status for last received request.

IE	M/O	Comment
CMND_IE_RESPONSE	M	0x0 - Succeeded 0x1 - Failed

4.19.2.4 CMND_MSG_PROD_REF_CLK_TUNE_START_REQ

Start the 13.824 MHz reference clock tuning process. The command uses a GPIO (not a UART GPIO) for measurement.

IE	M/O	Comment
CMND_IE_U8	M	Specifies GPIO to use (other than UART GPIO)

Response: generic CMND_MSG_PROD_CFM

4.19.2.5 CMND_MSG_PROD_ REF_CLK _TUNE_END_REQ

End the 13.824 MHZ reference clock tuning process and write results to the EEPROM. No IE is used.

Response: CMND_MSG_PROD_RX_TUNE_END_RES

4.19.2.6 CMND_MSG_PROD_ REF_CLK_TUNE_END_RES

Confirm end of 13.824 MHz reference clock tuning process, and to returns the adjusted value.

IE	M/O	Comment
CMND_IE_RESPONSE	M	0x0 - Process succeeded 0x1 - Process failed
CMND_IE_U8	M	Adjusted new value written to EEPROM

4.19.2.7 CMND_MSG_PROD_REF_CLK_TUNE_ADJ_REQ

Change RXTUN value by stepping one up or down. This command should during the 13.824Mhz clock tuning.

IE	M/O	Comment
CMND_IE_U8	M	0x01 – Step Up 0xFF - Step Down

Response: generic CMND_MSG_PROD_CFM

4.19.2.8 CMND_MSG_PROD_BG_REQ

Start band gap calibration procedure. During this procedure the CMND should be powered by a known and calibrated power supplier. The input voltage is used as a reference for the calibration procedure.

IE	M/O	Comment
CMND_IE_BG_REQ	M	MUX Input, Resistor Factor and Supply Volt parameters are used in calibration process.

Response: CMND_MSG_PROD_BG_RES

4.19.2.9 CMND_MSG_PROD_BG_RES

This message is sent when band gap calibration completes.

IE	M/O	Comment
CMND_IE_BG_RES	M	ADC and POR values

CMND_IE_RESPONSE	M	0x0 - Calibration succeeded 0x1 - Calibration failed

4.19.2.10 CMND_MSG_PROD_ATE_INIT_REQ

This message is used to initialize the LMAC ATE test.

IE	M/O	Comment
CMND_IE_U8	M	Sets the packet length of the slot, default should be set to Full slot (320 bit b-field) Slot: Full=1,Double=2 or Long=3

Response: generic CMND_MSG_PROD_CFM

4.19.2.11 CMND_MSG_PROD_ATE_STOP_REQ

This message is used to stop the LMAC ATE test for all modes.

Response: generic CMND_MSG_PROD_CFM

4.19.2.12 CMND_MSG_PROD_ATE_CONT_START_REQ

This message is used to start a continuous ATE test. It requires the following IE.

IE	M/O	Comment
CMND_IE_ATE_CONT_REQ	M	Slot, Tx/Rx, Carrier and Ant passed as parameters to ATE process.

Response: generic CMND_MSG_PROD_CFM

4.19.2.13 CMND_MSG_PROD_ATE_RX_START_REQ

This message is used to start the Rx ATE test.

IE	M/O	Comment
CMND_IE_ATE_RX_REQ	M	Slot type, PP Sync, Slot#, Carrier, Ant, BER-FER and number of frames.

4.19.2.14 CMND_MSG_PROD_ATE_RX_START_RES

This message is used for Rx ATE start test response.

IE	M/O	Comment
CMND_IE_ATE_RX_RES	M	BER measurement, FER Measurement and RSSI
CMND_IE_RESPONSE	M	0x0 - Succeeded 0x1 - Failed

4.19.2.15 CMND_MSG_PROD_ATE_TX_START_REQ

This message is used to start the Tx ATE test.

IE	M/O	Comment
CMND_IE_ATE_TX_REQ	M	Slot, Preamble, Slot#, Carrier, Power Level, Ant and Pattern

Response: generic CMND_MSG_PROD_CFM

4.19.2.16 CMND_MSG_PROD_ATE_GET_BER_FER_REQ

This message is used to get BER-FRE measurement while Rx test is ongoing. No IE is used.

Response: generic CMND_MSG_PROD_ATE_RX_START_RES

4.19.2.17 CMND_MSG_PROD_ INIT_EEPROM_DEF_REQ

This message is used to set the EEPROM to default values by selecting 9 in the online menu. No IE is used.

Response: generic CMND_MSG_PROD_CFM

4.19.2.18 CMND_MSG_PROD_SPECIFIC_PRESET_REQ

This message is used to set the EEPROM to automatic preset - a set of predefined values stored into the EEPROM when selected preset is applied.

IE	M/O	Comment
CMND_IE_U8	M	0x00 – CR GPIO preset 0x01 - CR CMND API preset 0x02 – AC GPIO on DHX91 board 0x03 – Smoke CMND API on DHX91 board 0x04 – Smoke ULE GPIO on DHX91 board 0x05 – ULE voice call standalone on dhx91 0x06 – ULE voice call with CMND API 0x07 – Vendor SPMKT 0x08 – AC Uart 0x09 – Simple Power Metering Uart DHX91 0x0A – Reserved 0x0B – Wakeup from UART on DHX91 board 0x0C – Simple Power Metering DHX91 board 0x0D – Vendor EU Thermostat 0x0E – Vendor EU WallSwitch 0x0F – Vendor EU Window 0x10 – Host Extension 0x11 – Smoke Pageable 0x12 – AC Broadcast 0x13 – AC Broadcast CMND 0x14 – Generic CMND

Response: generic CMND_MSG_PROD_CFM

4.19.2.19 CMND_MSG_PROD_SLEEP_REQ

This message is used to enter hibernation mode. All hw peripherals are disabled (UART stops working), the CMND cannot be woken up from external source. To exit hibernation the CMND must be restarted (it is also advised to discharge the capacitors of the module after reset to restart the module).

Response: generic CMND_MSG_PROD_CFM

4.19.2.20 CMND_MSG_PROD_SET_SIMPLE_GPIO_LOW

This message is used to set a simple GPIO’s state to low. Polarity is ignored.

IE	M/O	Comment
CMND_IE_U8	M	Simple GPO’s number.

Response: generic CMND_MSG_PROD_CFM

4.19.2.21 CMND_MSG_PROD_SET_SIMPLE_GPIO_HIGH

This message is used to set a simple GPIO’s state to high. Polarity is ignored.

IE	M/O	Comment
CMND_IE_U8	M	Simple GPO's number.

Response: generic CMND_MSG_PROD_CFM

4.19.2.22 CMND_MSG_PROD_GET_SIMPLE_GPIO_STATE

This message is used to get the current state of a simple GPIO. Polarity is ignored.

IE	M/O	Comment
CMND_IE_U8	M	Simple GPO identifier.

Response: generic CMND_MSG_PROD_CFM

4.19.2.23 CMND_MSG_PROD_GET_SIMPLE_GPIO_STATE_RES

This message is used as a response to CMND_MSG_PROD_GET_SIMPLE_GPIO_STATE request. Polarity is ignored

IE	M/O	Comment
CMND_IE_RESPONSE	M	Request response status 0x00 – Success 0x01 – Invalid GPIO
CMND_IE_GPIO_STATE	M	GPIO ID and state (low/high)

4.19.2.24 CMND_MSG_PROD_SET_ULE_GPIO_LOW

This message is used to set a ULE GPIO state to low. Polarity is ignored.

IE	M/O	Comment
CMND_IE_U8	M	ULE GPO identifier See ULE GPIO **output** identifiers in CMND_IE_GPIO_STATE

Response: generic CMND_MSG_PROD_CFM

4.19.2.25 CMND_MSG_PROD_SET_ULE_GPIO_HIGH

This message is used to set a ULE GPIO state to high. Polarity is ignored.

IE	M/O	Comment
CMND_IE_U8	M	ULE GPO identifier See ULE GPIO **output** identifiers in CMND_IE_GPIO_STATE

Response: generic CMND_MSG_PROD_CFM

4.19.2.26 CMND_MSG_PROD_GET_ULE_GPIO_STATE

This message is used to get the current state of a ULE GPIO. Polarity is ignored. Must send CMND_MSG_PROD_SET_ULE_GPIO_DIR_INPUT_REQ message before sending this message in order to configure the ULE GPIO direction as input.

IE	M/O	Comment
CMND_IE_U8	M	ULE GPI identifier See ULE GPIO **input** identifiers in CMND_IE_GPIO_STATE

Response: generic CMND_MSG_PROD_CFM

4.19.2.27 CMND_MSG_PROD_GET_ULE_GPIO_STATE_RES

This message is used as a response to CMND_MSG_PROD_GET_ULE_GPIO_STATE request. Polarity is ignored

IE	M/O	Comment
CMND_IE_RESPONSE	M	Request response status 0x00 – Success 0x01 – Invalid GPIO
CMND_IE_GPIO_STATE	M	ULE GPIO ID and state (low/high)

4.19.2.28 CMND_MSG_PROD_SET_ULE_GPIO_DIR_INPUT_REQ

This message is used to configure ULE gpio direction as input.

Response: generic CMND_MSG_PROD_CFM

IE	M/O	Comment
CMND_IE_U8	M	ULE GPI identifier See ULE GPIO **input** identifiers in CMND_IE_GPIO_STATE

4.19.2.29 CMND_MSG_PROD_RESET_EEPROM

This message is used to Reset the HAN/DECT EEPROM Section to its defaults.

IE	M/O	Comment
CMND_IE_U8	M	0x00 – HAN 0x01 -- DECT

4.19.2.30 CMND_MSG_PROD_FW_UPDATE_REQ

This message starts FW upgrade via UART port. The CMND will stop current activity and wait for CMBS FW upgrade API from uart.

IE	M/O	Comment
CMND_IE_U8	M	0x00 – Don't erase factory image 0x01 – Erase factory image

Response: generic CMND_MSG_PROD_CFM, response is sent before FW upgrade starts.

4.19.2.31 CMND_MSG_PROD_GPIO_LOOPBACK_TEST_REQ

This message is used to start GPIO loopback test. The test connects predefined GPIO (output) to ULE GPIO (input), generates a Low or High signal on output GPIO, reads the value from input ULE GPIO and compares the results. If all results are as expected a CMND_RC_OK is returned, otherwise an error is returned

Output	→	Input
GPIO3	→	DCINS
GPIO7	→	LEDSINK2
GPIO7	→	AMP2_OUT
GPIO0	→	AMP1_OUT
GOIO1	→	ANA_IN2
PWM0	→	ANA_IN1

IEs: None

Response: generic CMND_MSG_PROD_CFM

IE	M/O	Comment
CMND_IE_RESPONSE	M	0x00 (CMND_RC_OK) – Tests succeeded 0x01 (CMND_RC_FAIL) – Test failed 0x0B (CMND_RC_FAIL_INVALID_VALUE) – Internal error, invalid ULE GPIO

4.19.2.32 CMND_MSG_PROD_ATE_RX_LOCKING_START_REQ

This message starts ATE RX locking test.

IE	M/O	Comment
CMND_IE_ATE_RX_REQ	M	Slot, Sync pattern, Carrier, Ant, ber-fer, number of frames.

Response: generic CMND_MSG_PROD_CFM

4.20 Attribute Reporting Service

The following abbreviations are used in tables in this section:

N → C: Message can be sent from Node Host to CMND

C → N: Message can be sent from CMND to Node Host

M: Mandatory

O: Optional

4.20.1 Overview

This service is used for periodic or event-driven reporting of miscellaneous attributes.

4.20.2 Add/Delete a Report

To add a report from the base or Node Host, see the flows in Figure 4-3 and Figure 4-4. The flow to delete a report is shown in Figure 4-5.

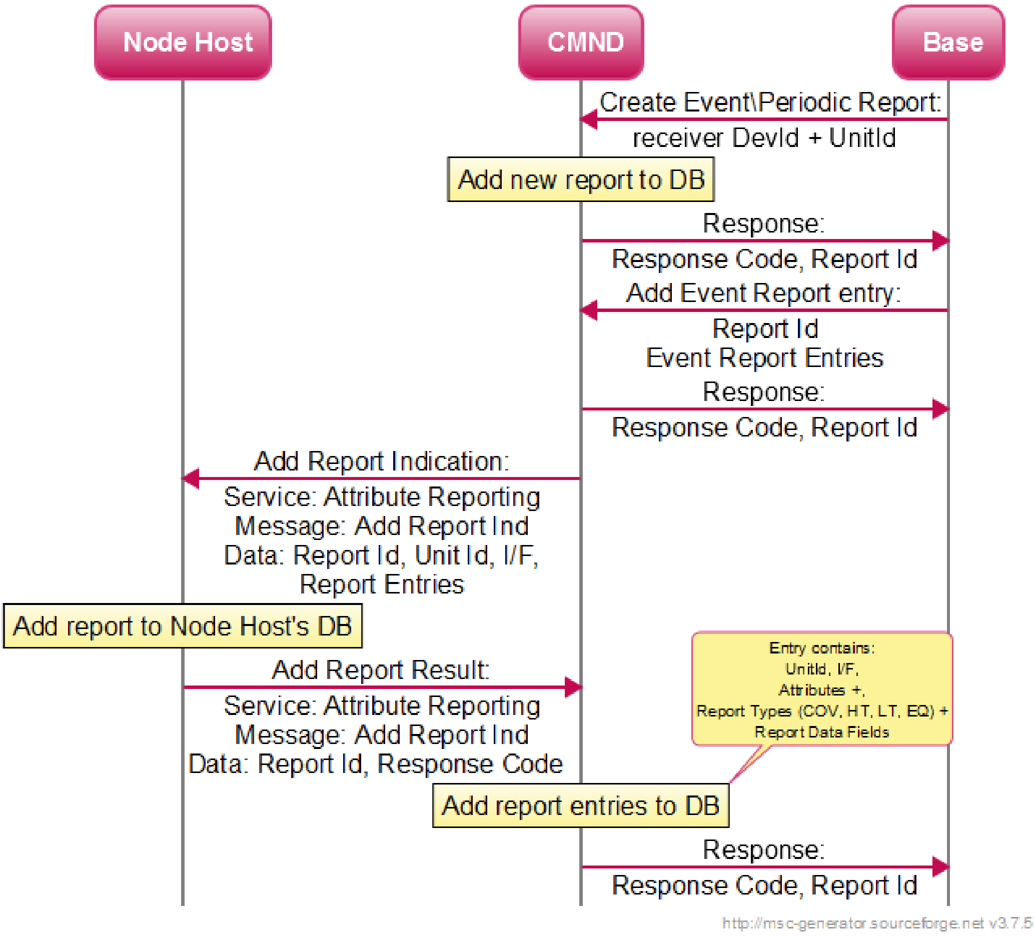


Figure 4-3: Add Report from the Base

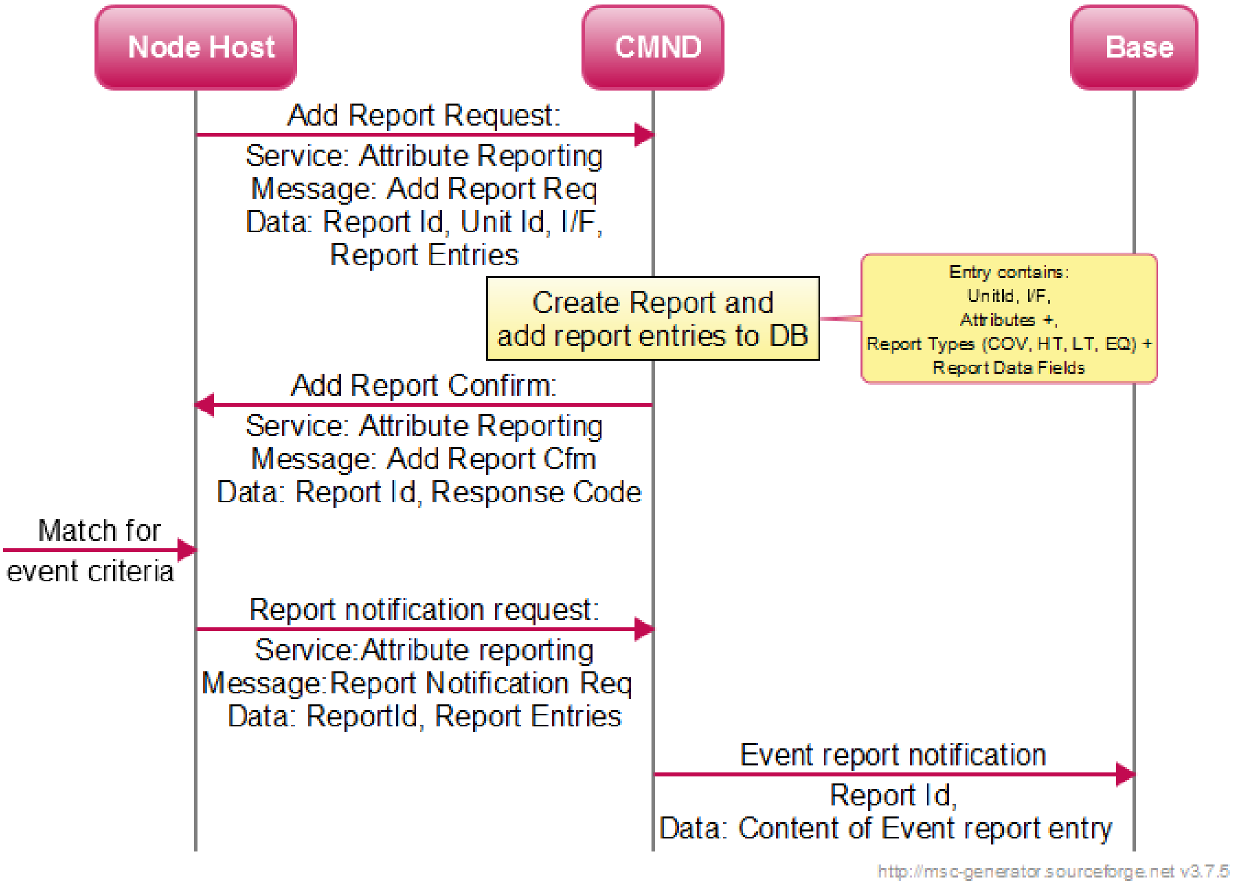

Node Host
CMND
Base
Add Report Request:
Service: Attribute Reporting
Message: Add Report Req
Data: Report Id, Unit Id, I/F,
Report Entries
Create Report and
add report entries to DB
Entry contains:
UnitId, I/F,
Attributes +,
Report Types (COV, HT, LT, EQ) +
Report Data Fields
Add Report Confirm:
Service: Attribute Reporting
Message: Add Report Cfm
Data: Report Id, Response Code
Match for
event criteria
Report notification request:
Service:Attribute reporting
Message:Report Notification Req
Data: ReportId, Report Entries
Event report notification
Report Id,
Data: Content of Event report entry
http://msc-generator.sourceforge.net v3.7.5

Figure 4-4: Add Report from the Node Host

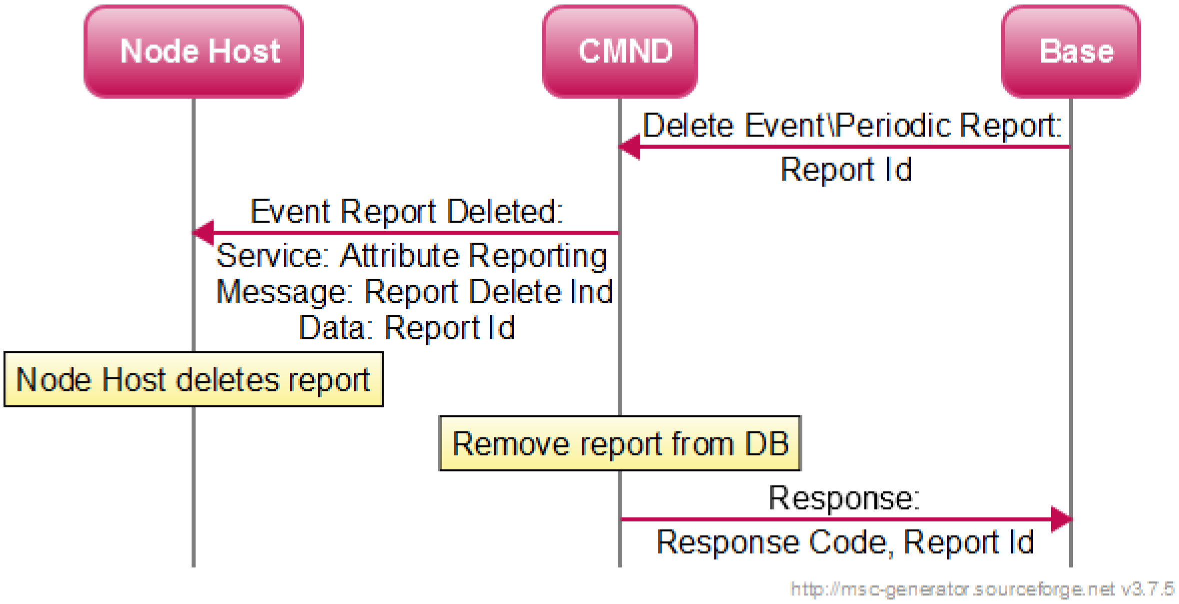


Figure 4-5: Delete Existing Report

4.20.3 Messages

MESSAGE ID	MESSAGE NAME	MESSAGE DESCRIPTION	N→C	C→N
0x11	CMND_MSG_ATTRREP_ADD_REPORT_IND	Add new report from CMND	N	Y
0x12	CMND_MSG_ATTRREP_ADD_REPORT_RES	Response to 'ADD' new report with result	Y	N
0x13	CMND_MSG_ATTRREP_REPORT_NOTIFICATION_REQ	Report to CMND when some condition is met	Y	N
0x14	CMND_MSG_ATTRREP_DELETE_REPORT_IND	Delete report indication	N	Y
0x15	CMND_MSG_ATTRREP_GET_REPORT_VALUES_IND	Get report values	N	Y
0x16	CMND_MSG_ATTRREP_GET_REPORT_VALUES_RES	Response to 'GET' with report values data	Y	N
0x17	CMND_MSG_ATTRREP_ADD_REPORT_REQ	Add report request by Node Host	Y	N
0x18	CMND_MSG_ATTRREP_ADD_REPORT_CFM	Response to 'ADD' new report from Node Host with result	N	Y

4.20.3.1 CMND_MSG_ATTRREP_ADD_REPORT_IND

This message is sent from CMND to the Node Host. This message is used to create a new report or add new report entries to an existing report. The receiving side sends a CMND_MSG_ATTR_REP_ADD_REPORT_RES message to confirm report entries were added successfully.

IE	M/O	Comment
CMND_IE_ATTR_ADD_REPORT_ENTRY	M	Add report information – multiple report entries to add

4.20.3.2 CMND_MSG_ATTRREP_ADD_REPORT_RES

This message is the response to CMND_MSG_ATTR_REP_ADD_REPORT_IND.

IE	M/O	Comment
CMND_IE_RESPONSE	M	0x0 – successful handling 0x1 – failed to handle the request
CMND_IE_REPORT_ID	M	Report type and ID

4.20.3.3 CMND_MSG_ATTRREP_REPORT_NOTIFICATION_REQ

This message is sent when a report condition is met on Node Host (Attribute changed, or a periodic report event) and a report should be generated towards the Base. The message payload conveys attributes with their current values.

IE	M/O	Comment
CMND_IE_REPORT_INFO	M	Report information – multiple attribute values to report

4.20.3.4 CMND_MSG_ATTRREP_DELETE_REPORT_IND

This message is sent when report should be deleted. The receiving side sends a CMND_MSG_DELETE_REPORT_RES message to confirm successful report deletion.

IE	M/O	Comment
CMND_IE_REPORT_ID	M	Report type and ID

4.20.3.5 CMND_MSG_ATTRREP_GET_REPORT_VALUES_IND

This message is sent to receive the current attributes values for a specified report. The receiving side sends a CMND_MSG_GET_REPORT_VALUES_RES message with the response code and report data. This message is commonly used with periodic reports; in this case, attribute values are stored on the Node Host while the periodic timer is managed by the CMND.

IE	M/O	Comment
CMND_IE_REPORT_ID	M	Report type and ID

4.20.3.6 CMND_MSG_ATTRREP_GET_REPORT_VALUES_RES

This message is the response to CMND_MSG_GET_REPORT_VALUES_IND. In contains all report data

IE	M/O	Comment
CMND_IE_RESPONSE	M	0x0 - Handling succeeded 0x1 - Handling failed
CMND_IE_REPORT_INFO	O	Report information – multiple attribute values to report. CMND_IE_REPORT_INFO is mandatory if success; otherwise, it is not present

4.20.3.7 CMND_MSG_ATTRREP_ADD_REPORT_REQ

This message is sent from the Node Host to the CMND. It is used to create a new report or add new report entries to the existing report. The receiving side sends a CMND_MSG_ADD_REPORT_CFM message to confirm the report entries were added successfully. This message is used to create a local report instead of report creation in response to request from a remote Base.

IE	M/O	Comment
CMND_IE_ADD_REPORT_INFO	M	Add report information – multiple report entries to add

4.20.3.8 CMND_MSG_ATTRREP_ADD_REPORT_CFM

This message is the response to CMND_MSG_ADD_REPORT_REQ.

IE	M/O	Comment
CMND_IE_REPORT_ID	M	Report type and ID
CMND_IE_RESPONSE	M	0x0 - Handling succeeded 0x1 - Handling failed

4.21 Debug Service

The following abbreviations are used in tables in this section:

N → C: Message can be sent from Node Host to CMND

C → N: Message can be sent from CMND to Node Host

M: Mandatory

O: Optional

4.21.1 Overview

This service provides debug mechanism, e.g. notification on eeprom write.

4.21.2 Messages

MESSAGE ID	MESSAGE NAME	MESSAGE DESCRIPTION	N→C	C→N
0x04	CMND_MSG_DEBUG_EPROM_WR_NOTIFY_REQ	Start notification on eeprom writes	Y	N
0x05	CMND_MSG_DEBUG_EPROM_WR_EVENT	Notification on eeprom write	N	Y
0x06	CMND_MSG_DEBUG_EPROM_WR_NOTIFY_CFM	Confirm on receive and start notification	N	Y

4.21.2.1 CMND_MSG_DEBUG_EPROM_WR_NOTIFY_REQ

This message is used to request the CMND to perform EEPROM write notifications.

4.21.2.2 CMND_MSG_DEBUG_EPROM_WR_EVENT

IE	M/O	Comment
CMND_IE_32	M	Address Written

4.21.2.3 CMND_MSG_DEBUG_EPROM_WR_NOTIFY_CFM

This message ACK the receive of Debug Notification.

4.22 ULE Voice Call Service

The following abbreviations are used in tables in this section:

N → C: Message can be sent from Node Host to CMND
C → N: Message can be sent from CMND to Node Host
M: Mandatory
O: Optional

4.22.1 Overview

This service provides functionality to manage a voice call with base / other devices.

4.22.2 Messages

MESSAGE ID	MESSAGE NAME	MESSAGE DESCRIPTION	N→C	C→N
0x01	CMND_MSG_ULE_CALL_REQUEST_IND	Indication on voice call request from base	N	Y
0x02	CMND_MSG_ULE_CALL_REQUEST_RES	Response to “call request” indication	Y	N
0x03	CMND_MSG_ULE_CALL_CANCEL_IND	Indication to cancel call request from base	N	Y
0x04	CMND_MSG_ULE_CALL_CANCEL_RES	Response to “cancel call request” indication	Y	Y
0x05	CMND_MSG_ULE_CALL_ACTIVE_REQ	Query if there is an active call	Y	N
0x06	CMND_MSG_ULE_CALL_ACTIVE_RES	Response to “active call” with result	N	Y
0x07	CMND_MSG_ULE_CALL_CODEC_IND	Request which codec to use for voice call	N	Y
0x08	CMND_MSG_ULE_CALL_CODEC_RES	Response to “start call” with result	Y	N
0x09	CMND_MSG_ULE_CALL_START_REQ	Request to start a call	Y	N
0x0A	CMND_MSG_ULE_CALL_START_CFM	Confirmation on start call	N	Y
0x0B	CMND_MSG_ULE_CALL_END_REQ	Request to end active call	Y	N
0x0C	CMND_MSG_ULE_CALL_END_CFM	Confirmation on end call	N	Y
0x0D	CMND_MSG_ULE_CALL_RELEASE_IND	Indication call released remotely	N	Y
0x0E	CMND_MSG_ULE_CALL_SET_VOLUME_REQ	Set voice call volume level	Y	N
0x0F	CMND_MSG_ULE_CALL_SET_VOLUME_CFM	Confirmation for handling set volume request	N	Y
0x10	CMND_MSG_ULE_CALL_VOLUME_UP_REQ	Set volume UP request	Y	N
0x11	CMND_MSG_ULE_CALL_VOLUME_UP_CFM	Confirmation for handling set volume up request	N	Y
0x12	CMND_MSG_ULE_CALL_VOLUME_DOWN_REQ	Set volume down by one step	Y	N

MESSAGE ID	MESSAGE NAME	MESSAGE DESCRIPTION	N→C	C→N
0x13	CMND_MSG_ULE_CALL_VOLUME_DOWN_CFM	Confirmation for volume down request	N	Y
0x14	CMND_MSG_ULE_CALL_CONNECTED_IND	Indication on call connection	N	Y
0x15	CMND_MSG_ULE_CALL_STATUS_UPDATE_REQ	Tell the client about the status of the call request	Y	N

4.22.2.1 CMND_MSG_ULE_CALL_REQUEST_IND

This message is an indication from CMND to Node Host on a request issued by base to start a voice call. Once call is established the base host will route the call according to request / route table. The Node Host should respond with CMND_MSG_ULE_CALL_REQUEST_RES message to approve/reject voice call establishment.

IE	M/O	Comment
CMND_IE_ULE_CALL_SETTING	M	Voice call settings such as codec, dial digits and others.

4.22.2.2 CMND_MSG_ULE_CALL_REQUEST_RES

Response from MCU to CMND which approve/reject and voice call settings. In case of “accept” the CMND will establish a voice call with respect to call settings.

IE	M/O	Comment
CMND_IE_RESPONSE	M	CMND_RC_OK – accept the call CMND_RC_FAIL – reject the call
CMND_IE_ULE_CALL_SETTING	M[1]	Call settings.

[1] Mandatory only if response is CMND_RC_OK

4.22.2.3 CMND_MSG_ULE_CALL_CANCEL_IND

This is an indication to MCU to cancel a voice call request. MCU should respond with CMND_MSG_ULE_CALL_CANCEL_RES and with successful response payload

IEs: none

4.22.2.4 CMND_MSG_ULE_CALL_CANCEL_RES

Response from MCU for end call request.

IE	M/O	Comment
CMND_IE_RESPONSE	M	CMND_RC_OK – accept to cancel call CMND_RC_FAIL – reject to cancel call

4.22.2.5 CMND_MSG_ULE_CALL_ACTIVE_REQ

This message is for MCU to inquire if an ongoing voice call is active.

Not implemented.

4.22.2.6 CMND_MSG_ULE_CALL_ACTIVE_RES

This message ACK the request if a voice calls is active.

Not implemented.

4.22.2.7 CMND_MSG_ULE_CALL_CODEC_IND

This message is used to set the preferred codec for a voice call with base. The MCU should respond with CMND_MSG_ULE_CALL_CODEC_RES message to confirm the codec settings.

Not implemented.

4.22.2.8 CMND_MSG_ULE_CALL_CODEC_RES

This message ACK the request for call codec

Not implemented

4.22.2.9 CMND_MSG_ULE_CALL_START_REQ

Request from Node Host to CMND to start a voice call. The CMND should respond with CMND_MSG_ULE_CALL_START_CFM message to confirm message arrival and starting a voice call.

IE	M/O	Comment
CMND_IE_ULE_CALL_SETTING	M	Voice call settings such as codec, dial digits and others

4.22.2.10 CMND_MSG_ULE_CALL_START_CFM

This message is a confirmation for receiving the CMND_MSG_ULE_CALL_START_REQ and starting a voice call. This message does not confirm whether voice call connected successfully.

IE	M/O	Comment
CMND_IE_RESPONSE	M	CMND_RC_OK – if voice call started. CMND_RC_FAIL – if cannot start voice call at this moment. CMND_RC_FAIL_CALL_ONGOING (0x15) – there's already ongoing call

4.22.2.11 CMND_MSG_ULE_CALL_END_REQ

This message is a request from MCU to CMND to end ongoing voice call. The CMND should respond with CMND_MSG_ULE_CALL_END_CFM message to confirm message handling.

4.22.2.12 CMND_MSG_ULE_CALL_END_CFM

This message is a confirmation for receiving the CMND_MSG_ULE_CALL_END_REQ. The call is actually released when CMND_MSG_ULE_CALL_RELEASE_IND is issued by CMND to Node host.

IE	M/O	Comment
CMND_IE_RESPONSE	M	CMND_RC_OK – there's ongoing voice call and CMND can start the "end call" procedure CMND_RC_FAIL_CALL_NOT_CONNECTED (0x14)– there's no ongoing voice call

4.22.2.13 CMND_MSG_ULE_CALL_RELEASE_IND

This is indication to MCU on call release either unexpectedly by remote side (base) or explicitly by CMND_MSG_ULE_CALL_END_REQ.

IEs: None.

4.22.2.14 CMND_MSG_ULE_CALL_SET_VOLUME_REQ

This is a request from MCU to CMND to set volume of voice call to specified level. The CMND will store this level in EEPROM and if there's ongoing call, the call volume will be affected based on the level passed in this request.

Data:

IE	M/O	Comment
CMND_IE_U8	M	Volume level index: 0..9 This level refers to a 10 level table stored in CMND module EEPROM parameter SD02_EAR_VOL_TAB (DECT part). The values in these table are based volumes for Automatic Controlled Volume (A_ACV) module of voice call data flow. See DSP Table 13 (24dBm)

The following table is taken from [10] Table 13:

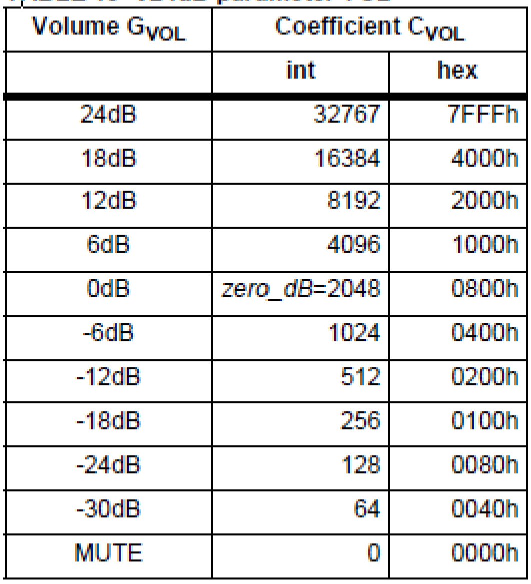


TABLE 13 V24dB parameter *VOL*

Volume G_{VOL}	Coefficient C_{VOL}	
	int	hex
24dB	32767	7FFFh
18dB	16384	4000h
12dB	8192	2000h
6dB	4096	1000h
0dB	*zero_dB*=2048	0800h
-6dB	1024	0400h
-12dB	512	0200h
-18dB	256	0100h
-24dB	128	0080h
-30dB	64	0040h
MUTE	0	0000h

4.22.2.15 CMND_MSG_ULE_CALL_SET_VOLUME_CFM

This message is a confirmation for setting the volume level.

IE	M/O	Comment
CMND_IE_RESPONSE	M	CMND_RC_OK – volume set successfully CMND_RC_FAIL_INVALID_VALUE – invalid volume level

4.22.2.16 CMND_MSG_ULE_CALL_VOLUME_UP_REQ

This is a request from MCU to CMND to increase current volume level up by one. The CMND will store the new level in EEPROM and if there's ongoing call, the call volume will be affected based on the level passed in this request.

Data: None

4.22.2.17 CMND_MSG_ULE_CALL_VOLUME_UP_CFM

This message is a confirmation for setting the volume level up. If current volume is highest, then this command will have not affect. The CMND will not send error in this case

IE	M/O	Comment
CMND_IE_RESPONSE	M	CMND_RC_OK – volume set successfully

4.22.2.18 CMND_MSG_ULE_CALL_VOLUME_DOWN_REQ

This is a request from MCU to CMND to decrease current volume level down by one. The CMND will store the new level in EEPROM and if there's ongoing call, the call volume will be affected based on the level passed in this request.

Data: None

4.22.2.19 CMND_MSG_ULE_CALL_VOLUME_DOWN_CFM

This message is a confirmation for setting the volume level up. If current volume is lowest then this command will have not affect. The CMND will not send error in this case

IE	M/O	Comment
CMND_IE_RESPONSE	M	CMND_RC_OK – volume set successfully

4.22.2.20 CMND_MSG_ULE_CALL_VOLUME_GET_REQ

This is a request from MCU to CMND to read current voice call level.

Data: None

4.22.2.21 CMND_MSG_ULE_CALL_VOLUME_GET_RES

This message is a response with current voice call level.

Data:

IE	M/O	Comment
CMND_IE_U8	M	Volume level: 0..9

4.22.2.22 CMND_MSG_ULE_CALL_CONNECTED_IND

This message is an indication from CMND to Node Host that call got connected.

IE	M/O	Comment

CMND_IE_RESPONSE	M	CMND_RC_OK – call connected CMND_RC_FAIL – Failed to connect the call

4.22.2.23 CMND_MSG_ULE_CALL_STATUS_UPDATE_REQ

Server to inform the Client about the status of the call request.

IE	M/O	Comment
CMND_IE_U8	M	0x01 – In Progress 0x02 - Rejected

4.23 Broadcasting Service

The following abbreviations are used in tables in this section:

N → C: Message can be sent from Node Host to CMND
C → N: Message can be sent from CMND to Node Host
M: Mandatory
O: Optional

4.23.1 Overview

This service provides functionality to manage broadcasting group table with device.

4.23.2 Messages

MESSAGE ID	MESSAGE NAME	MESSAGE DESCRIPTION	N→C	C→N
0x01	CMND_MSG_BROADCAST_CONFIG_DEV_REQ	Allow host to configure device that is supports Broadcast	Y	N
0x02	CMND_MSG_BROADCAST_CONFIG_DEV_RES	Response to broadcast config request	N	Y
0x03	CMND_MSG_BROADCAST_CHANNEL_ALLOCATION_IND	Inform device it was assigned a broadcast channel	N	Y
0x04	CMND_MSG_BROADCAST_GROUP_CAST_MESSAGE	Pass the group-cast message to the host	N	Y
0x05	CMND_MSG_BROADCAST_GROUP_TABLE_ENTRY_IND	Inform device on table entry change	N	Y
0x06	CMND_MSG_BROADCAST_GROUP_TABLE_NUM_ENTRIES_REQ	Query the device for group table Number of Entries	Y	N
0x07	CMND_MSG_BROADCAST_GROUP_TABLE_NUM_ENTRIES_RES	Response to “Number of Entries”	N	Y
0x08	CMND_MSG_BROADCAST_GROUP_TABLE_READ_REQ	Allow host to “Read” group table entries	Y	N
0x09	CMND_MSG_BROADCAST_GROUP_TABLE_READ_RES	Response to “Read” group table entries	N	Y
0x0A	CMND_MSG_BROADCAST_GROUP_TABLE_MAX_NUM_ENTRIES_REQ	Query the device for group table Max Number of Entries	Y	N
0x0B	CMND_MSG_BROADCAST_GROUP_TABLE_MAX_NUM_ENTRIES_RES	Response to “Max Number of Entries”	N	Y

4.23.2.1 CMND_MSG_BROADCAST_CONFIG_DEV_REQ

This message allow host to configure device that it supports Broadcast.

IE’s: None

4.23.2.2 CMND_MSG_BROADCAST_CONFIG_DEV_RES

This is a response to CMND_MSG_BROADCAST_CONFIG_DEV_REQ command.

IE	M/O	Comment
CMND_IE_RESPONSE	M	0x00 Configuration succeeded 0x01 Configuration failed

4.23.2.3 CMND_MSG_BROADCAST_CHANNEL_ALLOCATION_IND

This message is an indication from CMND to Node Host to inform device it was assigned a broadcast channel.

IE	M/O	Comment
CMND_IE_BROADCAST_CHANNEL_ALLOCATION	M	Channel parameters like Channel Periodicity, Paging Id and Multicast Channel Id.

4.23.2.4 CMND_MSG_BROADCAST_GROUP_CAST_MESSAGE

This message is an indication from CMND to Node Host on a group-cast message.

IE’s: None

4.23.2.5 CMND_MSG_BROADCAST_GROUP_TABLE_ENTRY_IND

This message is an indication from CMND to Node Host on any change to the group table.

IE	M/O	Comment
CMND_IE_BROADCAST_GROUP_TABLE_OPERATION	M	Group table operation parameters like Operation Id, Group Id and Unit Id.

4.23.2.6 CMND_MSG_BROADCAST_GROUP_TABLE_NUM_ENTRIES_REQ

This message is a request from Node Host to CMND to get the group table number of entries.

The CMND should respond with CMND_MSG_BROADCAST_GROUP_TABLE_NUM_ENTRIES_RES message.

IE’s: None

4.23.2.7 CMND_MSG_BROADCAST_GROUP_TABLE_NUM_ENTRIES_RES

This message is a response from CMND to Node Host about the group table number of entries.

IE	M/O	Comment
CMND_IE_U8	M	Group table number of entries.

4.23.2.8 CMND_MSG_BROADCAST_GROUP_TABLE_MAX_NUM_ENTRIES_REQ

This message is a request from Node Host to CMND to get the group table max number of entries.

The CMND should respond with CMND_MSG_BROADCAST_GROUP_TABLE_MAX_NUM_ENTRIES_RES message.

IE’s: None

4.23.2.9 CMND_MSG_BROADCAST_GROUP_TABLE_MAX_NUM_ENTRIES_RES

This message is a response from CMND to Node Host about the group table max number of entries.

IE	M/O	Comment
CMND_IE_U8	M	Group table max number of entries.

4.23.2.10 CMND_MSG_BROADCAST_GROUP_TABLE_READ_REQ

This message is a request from Node Host to read group table entries. The CMND should respond with CMND_MSG_BROADCAST_GROUP_TABLE_READ_RES message.

IE	M/O	Comment
CMND_IE_BROADCAST_GROUP_TABLE_RANGE	M	Group Table read parameters like Start Index and Number of Entries to read.

4.23.2.11 CMND_MSG_BROADCAST_GROUP_TABLE_READ_RES

This message is a response from CMND to Node Host on a request for read group table entries.

IE	M/O	Comment
CMND_IE_BROADCAST_GROUP_TABLE_ENTRIES	M	Group Table read response parameters like Start Index, Number of Entries and Table Entry information.

4.24 Media Service

The following abbreviations are used in tables in this section:

N → C: Message can be sent from Node Host to CMND
C → N: Message can be sent from CMND to Node Host
M: Mandatory
O: Optional

4.24.1 Overview

This service provides functionality to play and stop media(audio) file stored on the Node Host.

4.24.2 Messages

MESSAGE ID	MESSAGE NAME	MESSAGE DESCRIPTION	N→C	C→N
0x01	CMND_MSG_MEDIA_PLAY_REQ	Play requested media file once	Y	N
0x02	CMND_MSG_MEDIA_PLAY_CFM	Confirmation for the play request message	N	Y
0x03	CMND_MSG_MEDIA_PLAY_LOOP_REQ	Play requested media file in a loop	Y	N
0x04	CMND_MSG_MEDIA_PLAY_LOOP_CFM	Confirmation for Play media file request	N	Y
0x05	CMND_MSG_MEDIA_STOP_REQ	Request to stop currently played media file	Y	N
0x06	CMND_MSG_MEDIA_STOP_CFM	Confirmation for stop play request	N	Y

4.24.2.1 CMND_MSG_MEDIA_PLAY_REQ

This message is a request to CMND to play media file one time. The file must be located on CMND and is identified using a unique identifier.

IE	M/O	Comment
CMND_IE_U8	M	Defines file identifier, starts from 0.

4.24.2.2 CMND_MSG_MEDIA_PLAY_CFM

This is a response to CMND_MSG_MEDIA_PLAY_REQ message.

IE	M/O	Comment
CMND_IE_RESPONSE	M	0x00 Message is handled successfully 0x16 Invalid file identifier

4.24.2.3 CMND_MSG_MEDIA_PLAY_LOOP_REQ

This message is a request to CMND to play media file in loop until explicitly stopped using the CMND_MSG_MEDIA_STOP_REQ. The file must be located on CMND and is identified via an unique identifier.

IE	M/O	Comment
CMND_IE_U8	M	Defines file identifier, starts from 0.

4.24.2.4 CMND_MSG_MEDIA_PLAY_LOOP_CFM

This is a response to CMND_MSG_MEDIA_PLAY_LOOP_REQ message.

IE	M/O	Comment
CMND_IE_RESPONSE	M	0x00 Message is handled successfully 0x16 Invalid file identifier

4.24.2.5 CMND_MSG_MEDIA_STOP_REQ

This is a request to CMND to stop playing ongoing media.

IE: none

4.24.2.6 CMND_MSG_MEDIA_STOP_CFM

This is a confirmation to CMND_MSG_MEDIA_STOP_REQ message.

IE	M/O	Comment
CMND_IE_RESPONSE	M	0x00 Message is handled successfully 0x01 general error

5. CMND API Information Elements

The following table defines IE identifiers.

Table 5-1: CMND API IE Listing

FIELD	ID	DESCRIPTION
CMND_IE_RESPONSE	0x00	Request return code (ok, fail, …)
CMND_IE_ATTRIBUTE_ID	0x01	Service Attribute ID
CMND_IE_ATTRIBUTE_VALUE	0x02	Service Attribute ID+ value
CMND_IE_UNIT_ADDR	0x03	HAN unique address: device ID + unit ID
CMND_IE_RESPONSE_REQUIRED	0x04	FUN response requirement
CMND_IE_FUN	0x05	Holds FUN message
CMND_IE_ALERT	0x06	Alert status
CMND_IE_SLEEP_INFO	0x07	Holds a request for Safe mode power up
Reserved	0x08	Contains information on device, its units and the supported interfaces
CMND_IE_VERSION	0x09	Contains CMND API version
CMND_IE_BATTERY_LEVEL	0x0A	Contains battery level value
CMND_IE_PARAMETER	0x0B	EEPROM parameter
CMND_IE_PARAMETER_DIRECT	0x0C	Direct access parameter
CMND_IE_GENERAL_STATUS	0x0D	General CMND status
CMND_IE_DEREGISTRATION	0x0E	Deregistration request data
CMND_IE_BIND_ENTRY	0x0F	Holds binding entry details to add or remove from bind table
CMND_IE_GROUP_ID	0x10	Group ID and name
CMND_IE_GROUP_ENTRY	0x11	Content of an entry in group table
CMND_IE_GROUP_INFO	0x12	Group Information - all device-units in the group table
CMND_IE_OTA_COOKIE	0x13	Conveys OTA message sequence/cookie
CMND_IE_CREATE_ATTR_REPORT_RESPONSE	0x16	Response for attribute report creation request
CMND_IE_REGISTRATION_RESPONSE	0x1C	Response for registration request
CMND_IE_TAMPER_ALERT	0x1D	Tamper alert status
CMND_IE_U8	0x1E	General-purpose 8-bit value
CMND_IE_BATTERY_MEASURE_INFO	0x1F	Battery requests information structure
CMND_IE_OTA_CONTROL	0x22	Send Control - Retransmission + Lock
CMND_IE_IDENTIFY	0x23	Contains device ID
CMND_IE_U32	0x24	
CMND_IE_BG_REQ	0x25	Band gap request
CMND_IE_BG_RES	0x26	Band gap response
CMND_IE_ATE_CONT_REQ	0x27	ATE continuous test
CMND_IE_ATE_RX_REQ	0x28	ATE Rx test
CMND_IE_ATE_RX_RES	0x29	ATE Rx test response
CMND_IE_ATE_TX_REQ	0x2A	ATE Tx test
	0x2B	(internal)
CMND_IE_BASE_WANTED	0x2C	Contains the parameters of a requested base
CMND_IE_REPORT_ID	0x2D	Attribute report type and ID

FIELD	ID	DESCRIPTION
CMND_IE_ADD_REPORT_INFO	0x2E	Add report information – multiple report entries to add
CMND_IE_REPORT_INFO	0x2F	Report information – multiple Attribute values to report
CMND_IE_ATTR_ADD_REPORT_ENTRY	0x15	Add entries information – multiple report entries to add
CMND_IE_NEW_SW_INFO	0x32	New SW / Hardware version and URL
CMND_IE_CURRENT_SW_INFO	0x33	Current SW/HW version without URL
CMND_IE_IMAGE_TYPE	0x34	Image type 0x00=FW 0x01=Node Host
CMND_IE_SW_VER_INFO	0x35	Current Software image info SW only
CMND_IE_READ_FILE_DATA_RES	0x36	Read SUOTA file from target response
CMND_IE_READ_FILE_DATA_REQ	0x37	Request to read from file
CMND_IE_U16	0x38	General 16bit value
CMND_IE_PMID	0x39	PMID required to set up a ULE service call
CMND_IE_PORTABLE_IDENTITY	0x3A	Portable Identity is used in setting up ULE service call messages
CMND_IE_SET_ATTRIBUTE_VALUE	0x3C	Service Set Attribute id + value
CMND_IE_DEREGISTRATION_RESPONSE	0x3D	Deregistration response from base
CMND_IE_GPIO_STATE	0x3E	ULE GPIO or Simple gpio identifier and state
CMND_IE_LINK_MAINTAIN	0x3F	Link maintain duration and ping interval
CMND_IE_ULE_CALL_SETTING	0x40	ULE Voice Call setup information
CMND_IE_CHECK_SW_VER_RES	0x42	Result of Check Version and new SW/HW version
CMND_IE_BROADCAST_CHANNEL_ALLOCATION	0x43	Broadcast channel allocation parameters
CMND_IE_BROADCAST_GROUP_TABLE_OPERATION	0x44	Broadcast group table operation parameters
CMND_IE_BROADCAST_GROUP_TABLE_RANGE	0x45	Broadcast group table range parameters
CMND_IE_BROADCAST_GROUP_TABLE_ENTRIES	0x46	Broadcast group table entries parameters

5.1 CMND_IE_RESPONSE

5.1.1 Description

This IE is used to convey the result code of a request. The result may also come in a confirmation or a response messages.

5.1.2 Structure

Field	Type	Description
Result	U8	Defines the result code

5.1.3 Default Return Codes

Return Code	Value	Description
CMND_RC_OK	0x0	General success
CMND_RC_FAIL	0x1	General failure
CMND_RC_NOT_REGISTERED	0x2	CMND message rejected because device is not registered
CMND_RC_FAIL_NOT_SUPPORTED	0x3	Request not supported
CMND_RC_FAIL_READ_ONLY_ATTRIBUTE	0x4	Failed to write to read only attribute

Return Code	Value	Description
CMND_RC_FAIL_SEND_LOCK	0x12	Failed to lock on base.
CMND_RC_FAIL_SEND_TRANSMISSION	0x13	Failed to transmit message but succeeded to lock.
CMND_RC_FAIL_NO_RESOURCES	0xFE	Failure due to non-sufficient resources
CMND_RC_FAIL_UNKNOWN_REASON	0xFF	Unknown reason

5.2 CMND_IE_ATTRIBUTE_ID

5.2.1 Description

This IE identifies service attribute by id

5.2.2 Data

Field	Type	Description
Attribute ID	U8	Service attribute identifier

5.3 CMND_IE_ATTRIBUTE_VALUE

5.3.1 Description

This IE holds attribute ID and value in ‘get attribute’ response or in ‘set attribute’ request.

5.3.2 Data

Field	Type	Description
Attribute ID	U8	Service attribute identifier
Response Value	U8	Operation result value
Data length	U8	Size of attribute value in bytes
Data	U8 array	Attribute Value

5.4 CMND_IE_UNIT_ADDR

5.4.1 Description

This IE contains unique address of a device and unit.

5.4.2 Data

Field	Type	Description
Device ID	U16	Device ID, the value should conform to FUN OTA protocol [1].
Unit ID	U8	Attribute value, the ID should conform to FUN OTA protocol [1].

5.5 CMND_IE_RESPONSE_REQUIRED

5.5.1 Description

This IE defines if a FUN response is required for the requested message.

5.5.2 Data

None

5.6 CMND_IE_FUN

5.6.1 Description

This IE contains FUN message data. Refer to [4] for full description of FUN protocol.

5.6.2 Data

Field	Type	Description
Extended Mode	U8	0 - Non Extended mode – Default 1 - Extended mode Extended mode bit should set to 0 for non-extended mode Extended mode is for future use
Source device ID	U16	Address of the specific source device
Source unit ID	U8	Source unit ID
Address type	U8	0 - Individual address 1 - Group address (not for phase 1 product)
Destination device ID	U16	Address of the specific destination device
Destination unit ID	U8	Specific unit identifier
Application reference	U8	For mapping responses to requests in case several requests were sent before responses arrived. This value will be returned in the response
Message type	U8	0x1 - Command message 0x2 - Command with response required 0x3 - Command response 0x4 - Attribute get message 0x5 - Attribute get response 0x6 - Attribute set message 0x7 - Attribute set with response 0x8 - Attribute set response 0x9 - Get attribute pack 0xA - Get attribute pack response 0xB - Set attribute pack 0xC - Set attribute pack with response 0xD - Set attribute pack response
Interface type	U8	0 - Client 1- Server For commands, set/get attributes – the interface type is of the destination device For response on command or set/get attribute – the interface type is of the device that sends the response. Refer to [4] „Interface type“ for full details.
Interface ID	U16	FUN Interface identifier
Interface member	U8	Interface command or attribute id
Data length	U16	Length of Data payload (in bytes) 0 – no data
Data payload	U8[]	Variable length payload. Limited to 128 bytes

5.7 CMND_IE_ALERT

5.7.1 Description

This IE contains alert status.

5.7.2 Data

Field	Type	Description
Unit Type	U16	FUN unit type: smoke, AC, motion, etc. Values are taken from FUN Standard.
Alert Status	U32	Alert bitmask. 0x0 - All Alerts Idle 0x01- Alert 0 is Alerting 0x21- Alerts 0 & 5 are Alerting, all other Idle

5.8 CMND_IE_SLEEP_INFO

5.8.1 Description

This IE contains sleep information.

5.8.2 Data

Field	Type	Description
Power up mode	U8	Defines the power up mode. 0x00 - Normal mode 0x01 - Safe mode

5.9 CMND_IE_VERSION

5.9.1 Description

This IE contains the CMND API version.

5.9.2 Data

Field	Type	Description
Version length	U8	Length of the version field in bytes.
Version	U8 * N	Version is an array of ASCII bytes without terminating NULL, which defines the version in string format. Length of the version is specified in Length field of the IE.

5.10 CMND_IE_BATTERY_LEVEL

5.10.1 Description

This IE contains the measured battery level.

5.10.2 Data

Field	Type	Description
Battery level	U16	Battery Value measured by CMND

5.11 CMND_IE_PARAMETER

5.11.1 Description

This IE used to access EEPROM or PRODUCTION parameters on CMND.

5.11.2 Data

Field	Type	Description
Parameter type	U8	0x00 – EEPROM

<table>
<tr><td></td><td></td><td>0x01 – Production</td></tr>
<tr><td>Parameter id</td><td>U8</td><td>Parameter ID depends on Parameter type field.

<b><i>EEPROM type (0x00)</i></b> defines logical names for parameters stored in device EEPROM. These parameters are defined in CMND API Parameters chapter.

<b><i>Production type (0x01)</i></b> is used for auto registration procedure. These parameter identifiers are described in table below. The parameter structures are defined in [8].
<table>
<tr><th>Param ID</th><th>Description</th></tr>
<tr><td>0</td><td>DECT Block 1</td></tr>
<tr><td>1</td><td>DECT Block 2</td></tr>
<tr><td>2</td><td>DECT Block 3</td></tr>
<tr><td>3</td><td>ULE</td></tr>
<tr><td>4</td><td>FUN</td></tr>
<tr><td>5</td><td>IPUI</td></tr>
</table></td></tr>
<tr><td>Data length</td><td>U16</td><td>Length of parameter data</td></tr>
<tr><td>Data</td><td>U8 buffer</td><td>Parameter data buffer</td></tr>
</table>

5.12 CMND_IE_PARAMETER_DIRECT

5.12.1 Description

This IE is used for direct access to EEPROM or RAM on CMND.

5.12.2 Data

Field	Type	Description
Address type	U8	0x00 – EEPROM HAN 0x01 – RAM 0x02 – EEPROM DECT 0X03 - DAIF
Offset	U32	For EEPROM HAN type - this is CMND EEPROM offset according to HAN EEPROM map (part of each release). For EEPOM DECT type, this is EEPROM offset in DECT EEPROM map. For RAM address type – this is physical RAM address. For DAIF address type – this is address of register located in Mirrored, WAPU/RAPU, ULE indirect, RF indirect or Analog indirect memory spaces. The address must be flat, for example CP_CTRL (0xA) should have address 0x8A (0x80 is the offset of RF indirect memory space and 0xA the register address)
Data length	U16	Length of parameter data

Field	Type	Description
Data	U8 buffer	Parameter data

5.13 CMND_IE_GENERAL_STATUS

5.13.1 Description

This IE is used to describe the general status of the CMND upon power up.

5.13.2 Data

Field	Type	Description
Power up mode	U8	Defines the power up mode. 0x00 - Normal mode 0x01 - Safe mode 0x02 - Production mode
Registration status	U8	0x00 - Registered 0x01 - Not registered
EEPROM status	U8	0x00 - EEPROM ready 0x01 - Non valid EEPROM
Device ID	U16	Device ID allocated by Base. This value is only relevant if device is registered (Registration status = 0x01)

5.14 CMND_IE_DEREGISTRATION

5.14.1 Description

This IE identified the device ID to be de-registered

5.14.2 Data

Field	Type	Description
Device ID	U16	Device id to be de-registered

5.15 CMND_IE_BIND_ENTRY

5.15.1 Description

This IE contains a binding table entry to allow adding a new entry or removing existing entry from the binding table stored on the GW.

The IE is not implemented.

5.15.2 Data

Field	Type	Description
Source Device ID	U16	0, 0x10 to 0xFFFF
Source Unit ID	U8	
Destination Address Type	U8	0x00 - Individual 0x01 - Group
Destination Device ID	U16	0, 0x10 to 0Xffff
Destination. Unit ID	U8	

Field	Type	Description
Interface ID	U16	FUN Profile interface ID [1]
Direction	U8	0x00 - Client to Server 0x01 - Server to Client

5.16 CMND_IE_GROUP_ID

5.16.1 Description

This IE contains Group ID and group name. This IE can also be used to identify a group by Group ID only. In this case, the group name is ignored and can be empty.

5.16.2 Data

Field	Type	Description
Group ID	U16	0x0001 – 0xfff0
Group Name Length	U8	Group name length
Group Name	U8 array	ASCII non zero terminating string

5.17 CMND_IE_GROUP_ENTRY

5.17.1 Description

This IE contains details of a single entry in group table: device + unit ID.

The IE is not implemented.

5.17.2 Data

Field	Type	Description
Group ID	U16	0x0001 – 0xfff0
Device ID	U16	0, 0x10 to 0xFFFF
Unit ID	U8	

5.18 CMND_IE_GROUP_INFO

5.18.1 Description

This IE contains all information about the device-units assigned to the group.

The IE is not implemented.

5.18.2 Data

Field	Type	Description
Group ID	U16	0x0001 – 0xfff0
Number of entries	U16	Number of device + unit entries in the table
Entry #1 Device ID	U16	0, 0x10 to 0xFFFF
Entry #1 Unit ID	U8	
Entry #2 Device ID	U16	0, 0x10 to 0xFFFF
Entry #2 Unit ID	U8	
…		

5.19 CMND_IE_OTA_COOKIE

5.19.1 Description

This IE contains OTA cookie generated either by the Node Host or received from the GW and passed to the Node Host. The most common usage is containing the FUN “application reference” field. This is an 8bit general purpose field defined by FUN and part of a FUN message.

5.19.2 Data

Field	Type	Description
Cookie	U8	OTA cookie value

5.20 CMND_IE_REGISTRATION_RESPONSE

5.20.1 Description

This IE contains response for registration request.

5.20.2 Data

Field	Type	Description
Response Code	U8	0x00 – Registration succeeded 0x01 – DECT registration failed 0x02 – FUN registration failed (unknown reason) 0x03 – FUN registration failed (not enough resources) 0x04 – ULE capabilities failed 0x05 – DECT registration failed and device fell back to previously known BS 0x06 – ULE capabilities failed and device fell back to previously known BS 0x07 – FUN registration failed, and device fell back to previously known BS 0x08 – FUN registration failed (not authorized) 0x09 – Registration already ongoing
Discriminator Type	U8	Indicates if Base supports proprietary interfaces. 0x00 - Non proprietary 0x01 - Proprietary
Device Address	U16	The device address allocated by D’0:U’0 This field is relevant only if response code is 0x0.
Discriminator Value	U16	If Discriminator Type is 0x01, this field contains the device’s EMC

5.21 CMND_IE_TAMPER_ALERT

5.21.1 Description

This IE contains tamper alert status.

5.21.2 Data

Field	Type	Description
Alert status	U8	0x0 - Tamper off 0x1- Tamper on

5.22 CMND_IE_U8

5.22.1 Description

This IE contains a general-purpose 8-bit value: 0..255

5.22.2 Data

Field	Type	Description
octet	U8	0…255

5.23 CMND_IE_BATTERY_MEASURE_INFO

5.23.1 Description

This IE contains battery measurement request information.

5.23.2 Data

Field	Type	Description
Measurement mode	U8	0x1 - Return latest battery measurement that performed during RF load (Tx/Rx). The result is returned immediately and contains the latest measurement sample. If battery voltage dropped since latest Tx/Rx but there was no RF activity, the measurement remains the same and is updated only on next Tx/Rx. 0x2 - Do battery measurement now. This measurement represents the current state of the battery as measured from A2D

5.24 CMND_IE_OTA_CONTROL

5.24.1 Description

This IE contains optional information for any CMND API message that should be sent to the BS over the air. The payload defines message priority and number of retransmissions.

5.24.2 Data

Field	Type	Description
Priority	U8	0x0 - Urgent priority 0x1 - Normal priority Urgent messages shall be handled before the normal ones.
Retransmission number	U16	Number of retries to re-send the message in case of failure

5.25 CMND_IE_IDENTIFY

5.25.1 Description

This IE contains an identifier to enable easy differentiation between several devices. Length is decided by the length field of the IE.

5.25.2 Data

Field	Type	Description
Identity	U8 buffer	Device identifier

5.26 CMND_IE_BG_REQ

5.26.1 Description

This IE contains parameters needed for band gap calibration process.

5.26.2 Data

Field	Type	Description
MUX	U8	MUX Input 0x0 - DCIN 3 0x1- DCIN 3/2
Resistor	U8	Resistor Factor 0x2 - When MUX is DCIN 3/2 0x1 - When MUX is DCIN 3
Supply Voltage	U16	Supply Voltage in mVolts

5.27 CMND_IE_BG_RES

5.27.1 Description

This IE contains parameters returned from the band gap calibration process.

5.27.2 Data

Field	Type	Description
ADC Value	U32	Voltage measured in auxiliary input. This value represents the battery voltage
POR Value	U32	POR Value - internal measurement

5.28 CMND_IE_ATE_CONT_REQ

5.28.1 Description

This IE contains parameters for ATE continuous test process.

5.28.2 Data

Field	Type	Description
Slot Type	U8	Sets the packet length of the slot, default should be set to Full slot (320 BIT b-field). 0x1 - Full slot 0x2 - Double slot 0x3 - Long slot
Tx/Rx	U8	't' (0x74) - Tx 'r' (0x72) - Rx
Carrier	U8	Sets the frequency of the carrier, the default 0X00 will set the carrier to 1897.344MHz at EU band Carrier, 2 digits
Ant	U8	0

5.29 CMND_IE_ATE_RX_REQ

5.29.1 Description

This IE contains parameters for ATE Rx test process.

5.29.2 Data

Field	Type	Description
Slot Type	U8	Sets the packet length of the slot, default should be set to Full slot (320 BIT b-field) 0x1 - Full slot 0x2 - Double slot 0x3 - Long slot
PP Sync Patter	U8	When measuring BER the S.G sync field (16BIT) pattern should be set according to this parameter setting, the default should be set to FP sync pattern (0XE98A) Use PP Sync Pattern='y', other defaults to FP sync pattern
Slot Number	U8	Slot number
Carrier	U8	Sets the frequency of the carrier, the default 0X00 will set the carrier to 1897.344MHz at EU band Carrier, 2 digits
Ant	U8	Antenna selection out of two optional antennas, for HS/Device the default is `0`
BER-FER Measurement	U8	Enables BER-FER measurement
Number of frames	U8	Number of frames for BER measurement

5.30 CMND_IE_ATE_RX_RES

5.30.1 Description

This IE contains parameters returned from ATE Rx test process.

5.30.2 Data

Field	Type	Description
BER measurement	U16	BER measurement
FER Measurement	U8	FER Measurement
RSSI	U8	RSSI

For a selection of 0x64 frames averaging (100 frames), the expected BER report is 0x20 at sensitivity level; i.e., if 1000PPM BER is the ultimate signal level, both this report and the FER report should be 0x00

The RSSI measurements should comply with the values in Table 5-2.

Table 5-2: DCX RSSI Readings Tabulation

Pin[dBm]	RSSIO Register Report [Hex]
-100	20
-90	38
-80	50
-70	70
-60	88

-50	A8
-40	C0
-30	D8
-20	D8
-10	D8
0	D8
10	D8
-100	20

5.31 CMND_IE_ATE_TX_REQ

5.31.1 Description

This IE contains parameters for ATE Tx test process.

5.31.2 Data

Field	Type	Description
Slot Type	U8	Sets the packet length of the slot, default should be set to Full slot (320 BIT b-field) Slot - Full=1, Double=2 or Long=3
Preamble	U8	Preamble, 'y'=normal - other long. The default should be set to long preamble (32 BIT)
Slot#	U8	Slot#
Carrier	U8	Sets the frequency of the carrier, the default 0X00 will set the carrier to 1897.344MHz at EU band Carrier, 2 digits
Power Level	U8	Power Level, 0..2 the Tx output power can be set to any value below: 0x0 - Full power ~24 dBm 0x1 - Low power ~13 dBm 0x2 - Lowest power ~0 dBm
Ant	U8	Antenna selection out of two optional antennas, for HS/Device the default is 0
Pattern	U8	Determines the content of the B-Field Data Pattern: (0..5: [0,0x22,0xF0,0xDD,Fig41,PRBS])

5.32 CMND_IE_BASE_WANTED

5.32.1 Description

This IE contains Base RFPI to be matched while registering.

5.32.2 Data

Field	Type	Description
RFPI	U8 array	5 bytes of RFPI

5.33 CMND_IE_REPORT_ID

5.33.1 Description

This IE contains Report Type and Report ID.

5.33.2 Data

Field	Type	Description
Report Type	U1	Report type: 0x00 - Periodic 0x01 – Event-driven
Report ID	U7	Report identifier

Byte order and organization of a CMND_IE_REPORT_ID:

8	7	6	5	4	3	2	1	Octet
Report Type	Report ID							1

5.34 CMND_IE_ADD_REPORT_INFO

5.34.1 Description

This IE contains the information about report entries that should be added to the report table. This IE is used in CMND_MSG_ATTRREP_ADD_REPORT_IND and CMND_MSG_ATTRREP_ADD_REPORT_REQ messages. Note that all attribute reporting messages are sent to Unit ID 0 but Unit IDs inside the message are related to units to which the attribute to report, belongs to.

5.34.2 Data

Field	Type	Description
Receiver Device Id	U16	Address of device that should receive the notification.
Receiver Unit Id	U8	Address of unit that should receive the notification.
Periodic Interval	U32	Specifies the time, in seconds, which dictates the sending of a notification.
Report Type	U1	Distinguishes if the report is periodic or event driven. 0x00 – Periodic, * 0x01 - Event
--	U7	This is reserved
Number of Report Entries	U8	Number of add report entries in the message
Entry #1 Unit ID	U8	Identifier of the unit to which the attribute to report, belongs to
Entry #1 Interface ID	U16	Type of the interface to which the attribute to report, belongs to
Entry #1 Pack ID	U8	Attribute Pack Identifier: 0x00 – Mandatory** 0xFE – All*** 0xFF - Dynamic
Entry #1 Type Of Reporting	U8	Reporting Type: 0x00 - Change of value (COV) 0x01 - High Threshold (HT) 0x02 - Low Threshold (LT)

Field	Type	Description
		0x03 – Equal Note: Present only for Report Type = 0x01 - Event
Entry #1 Number of Attributes	U8	Number of Attributes in the entry
Entry #1 Attributes Table	-	See description of Attributes Table below
Entry #2 Unit ID	U8	Identifier of the unit to which the attribute to report, belongs to
Entry #2 Interface ID	U16	Type of the interface to which the attribute to report, belongs to
Entry #2 Pack ID	U8	Attribute Pack Identifier: 0x00 – Mandatory* 0xFE – All** 0xFF - Dynamic
Entry #2 Number of Attributes	U8	Number of Attributes in the entry
Entry #2 Type Of Reporting	U8	Reporting Type: 0x00 - Change of value (COV)*** 0x01 - High Threshold (HT) 0x02 - Low Threshold (LT) 0x03 – Equal Note: Present only for Report Type = 0x01 - Event
Entry #2 Attributes Table	-	See description of Attributes Table below
…		

* In case of Report Type = 0x00 – Periodic “Type Of Reporting” field shall be omitted and the following fields in the Attributes Table (see below) shall be omitted: “Type Of Reporting”, “Attribute Size”, “Attribute Value”.
** In case of Pack ID = 0x00 – Mandatory the “Number of Attributes” field and Attributes Table shall be omitted.
*** In case of Pack ID = 0xFE – All the “Number of Attributes” field and Attributes Table shall be omitted.

The Attributes Table parameter should comply with these values:

Field	Type	Description
Entry #1 Attribute ID	U8	Identifier of an attribute to report
Entry #1 Type Of Reporting	U8	Reporting Type: 0x00 - Change of value (COV)* 0x01 - High Threshold (HT) 0x02 - Low Threshold (LT) 0x03 – Equal Note: Present only for Report Type = 0x01 - Event
Entry #1 Attribute Size	U8	Size of attribute value in bytes Note: Present only for Report Type = 0x01 - Event
Entry #1 Attribute Value	U8 array	The value Note: Present only for Report Type = 0x01 - Event
Entry #2 Attribute ID	U8	Identifier of an attribute to report
Entry #2 Type Of Reporting	U8	Reporting Type: 0x00 - Change of value (COV)* 0x01 - High Threshold (HT) 0x02 - Low Threshold (LT) 0x03 - Equal

Field	Type	Description
Entry #2 Attribute Size	U8	Size of attribute value in bytes
Entry #2 Attribute Value	U8 array	The value
…		

* For the Reporting Type = 0x00 - Change of value (COV) Attribute Size field shall be 0 and no attribute value shall be specified.

Byte order and organization of a CMND_IE_ADD_REPORT_INFO:

8	7	6	5	4	3	2	1	Octet
Receiver Device Id (MSB)								1
Receiver Device Id (LSB)								2
Receiver Unit Id								3
Periodic Interval (1st byte, MSB)								4
Periodic Interval (2nd byte)								5
Periodic Interval (3rd byte)								6
Periodic Interval (4th byte, LSB)								7
Report Type								8
Number of Report Entries								9
Report Entry 1								10
…								⋮
Report Entry n								11 + n

Byte order and organization of a Report Entry:

8	7	6	5	4	3	2	1	Octet
Unit ID								1
Interface ID (MSB)								2
Interface ID (LSB)								3
Attribute Pack ID								4
Number of Attributes								5
Attribute ID								6
Type of Reporting								7
Attribute Size								8
Attribute Value (MSB)								9
…								⋮
Attribute Value (LSB)								10 + n
…								⋮
Attribute ID								11 + n
Type of Reporting								12 + n
Attribute Size								13 + n
Attribute Value (MSB)								14 + n
…								⋮

Attribute Value (LSB)	**15 + n**

5.35 CMND_IE_ATTR_ADD_REPORT_ENTRY

5.35.1 Description

This IE contains the information about report entries that should be added to the report table by Node Host. This IE is used in CMND_MSG_ATTRREP_ADD_REPORT_IND message. Note that all attribute reporting messages are sent to Unit ID 0 but Unit IDs inside message are related to units to which the attribute to report, belongs to.

5.35.2 Data

Field	Type	Description
Report Type	U1	Distinguishes if the report is periodic or event driven. 0x00 – Periodic, * 0x01 - Event
Report Id	U7	Report identifier
Number of Report Entries	U8	Number of add report entries in the message
Entry #1 Unit ID	U8	Identifier of the unit to which the attribute to report, belongs to
Entry #1 Interface ID	U16	Type of the interface to which the attribute to report, belongs to
Entry #1 Pack ID	U8	Attribute Pack Identifier: 0x00 – Mandatory** 0xFE – All*** 0xFF - Dynamic
Entry #1 Type Of Reporting	U8	Reporting Type: 0x00 - Change of value (COV) 0x01 - High Threshold (HT) 0x02 - Low Threshold (LT) 0x03 – Equal Note: Present only for Report Type = 0x01 - Event
Entry #1 Number of Attributes	U8	Number of Attributes in the entry
Entry #1 Attributes Table	-	See description of Attributes Table below
Entry #2 Unit ID	U8	Identifier of the unit to which the attribute to report, belongs to
Entry #2 Interface ID	U16	Type of the interface to which the attribute to report, belongs to
Entry #2 Pack ID	U8	Attribute Pack Identifier: 0x00 – Mandatory* 0xFE – All** 0xFF - Dynamic
Entry #2 Number of Attributes	U8	Number of Attributes in the entry
Entry #2 Type Of Reporting	U8	Reporting Type: 0x00 - Change of value (COV)*** 0x01 - High Threshold (HT) 0x02 - Low Threshold (LT)

Field	Type	Description
		0x03 – Equal Note: Present only for Report Type = 0x01 - Event
Entry #2 Attributes Table	-	See description of Attributes Table below
…		

* In case of Report Type = 0x00 – Periodic "Type Of Reporting" field shall be omitted and the following fields in the Attributes Table (see below) shall be omitted: "Type Of Reporting", "Attribute Size", "Attribute Value".

** In case of Pack ID = 0x00 – Mandatory the "Number of Attributes" field and Attributes Table shall be omitted.

*** In case of Pack ID = 0xFE – All the "Number of Attributes" field and Attributes Table shall be omitted.

The Attributes Table parameter should follow the following table:

Field	Type	Description
Entry #1 Attribute ID	U8	Identifier of an attribute to report
Entry #1 Type Of Reporting	U8	Reporting Type: 0x00 - Change of value (COV)* 0x01 - High Threshold (HT) 0x02 - Low Threshold (LT) 0x03 – Equal Note: Present only for Report Type = 0x01 - Event
Entry #1 Attribute Size	U8	Size of attribute value in bytes Note: Present only for Report Type = 0x01 - Event
Entry #1 Attribute Value	U8 array	The value Note: Present only for Report Type = 0x01 - Event
Entry #2 Attribute ID	U8	Identifier of an attribute to report
Entry #2 Type Of Reporting	U8	Reporting Type: 0x00 - Change of value (COV)* 0x01 - High Threshold (HT) 0x02 - Low Threshold (LT) 0x03 - Equal
Entry #2 Attribute Size	U8	Size of attribute value in bytes
Entry #2 Attribute Value	U8 array	The value
…		

* For the Reporting Type = 0x00 - Change of value (COV) the Attribute Size field and Attribute Value field shall be omitted.

Byte order and organization of a CMND_IE_ADD_REPORT_INFO:

8	7	6	5	4	3	2	1	Octet
Report Type	Report ID							1
Number of Report Entries								2
Report Entry 1								3
…								⋮
Report Entry n								4 + n

Byte order and organization of a Report Entry:

8	7	6	5	4	3	2	1	Octet
Unit ID								1

Interface ID (MSB)	**2**
Interface ID (LSB)	**3**
Attribute Pack ID	**4**
Number of Attributes	**5**
Attribute ID	**6**
Type of Reporting	**7**
Attribute Size	**8**
Attribute Value (MSB)	**9**
…	⋮
Attribute Value (LSB)	**10 + n**
…	⋮
Attribute ID	**11 + n**
Type of Reporting	**12 + n**
Attribute Size	**13 + n**
Attribute Value (MSB)	**14 + n**
…	⋮
Attribute Value (LSB)	**15 + n**

5.36 CMND_IE_REPORT_INFO

5.36.1 Description

This IE contains the information about attribute values that are reported in the CMND_MSG_REPORT_NOTIFICATION_REQ and CMND_MSG_ATTRREP_GET_REPORT_VALUES_RES messages. Note that all attribute reporting messages are sent to Unit ID 0 but Unit IDs inside message are related to units to which the attribute to report, belongs to.

5.36.2 Data

Field	Type	Description
Report Type	U1	Distinguishes if the report is periodic or event driven. 0x00 – Periodic, 0x01 - Event
Report ID	U7	Report identifier
Number of Entries	U8	Number of add report entries in the message
Entry #1 Unit ID	U8	Identifier of the unit to which the attribute to report, belongs to
Entry #1 Interface ID	U16	Type of the interface to which the attribute to report, belongs to
Entry #1 Number of Attributes	U8	Number of Attributes in the entry
Entry #1 Attributes Table	-	See description of Attributes Table below
Entry #2 Unit ID	U8	Identifier of the unit to which the attribute to report, belongs to

Field	Type	Description
Entry #2 Interface ID	U16	Type of the interface to which the attribute to report, belongs to
Entry #2 Number of Attributes	U8	Number of Attributes in the entry
Entry #2 Attributes Table	-	See description of Attributes Table below
…		

The Attributes Table parameter should follow the following table:

Field	Type	Description
Entry #1 Attribute ID	U8	Identifier of an attribute to report
Entry #1 Type Of Reporting	U8	Reporting Type: * 0x00 - Change of value (COV) 0x01 - High Threshold (HT) 0x02 - Low Threshold (LT) 0x03 - Equal
Entry #1 Attribute Size	U8	Size of attribute value in bytes
Entry #1 Attribute Value	U8 array	The value
Entry #2 Attribute ID	U8	Identifier of an attribute to report
Entry #2 Type Of Reporting	U8	Reporting Type: 0x00 - Change of value (COV) 0x01 - High Threshold (HT) 0x02 - Low Threshold (LT) 0x03 - Equal
Entry #2 Attribute Size	U8	Size of attribute value in bytes
Entry #2 Attribute Value	U8 array	The value
…		

* For Periodic report this field should be present but is ignored.

Byte order and organization of a CMND_IE_REPORT_INFO:

8	7	6	5	4	3	2	1	Octet
Report Type	Report ID							1
Number of Report Entries								2
Report Entry 1								3
…								⋮
Report Entry n								4 + n

Byte order and organization of a Report Entry:

8	7	6	5	4	3	2	1	Octet
Unit ID								1
Interface ID (MSB)								2
Interface ID (LSB)								3
Number of Attributes								4
Attribute ID								5
Type of Reporting								6
Attribute Size								7

Attribute Value (MSB)	**8**
…	⋮
Attribute Value (LSB)	**9 + n**
…	⋮
Attribute ID	**10 + n**
Type of Reporting	**11 + n**
Attribute Size	**12 + n**
Attribute Value (MSB)	**13 + n**
…	⋮
Attribute Value (LSB)	**14 + n**

5.37 CMND_IE_NEW_SW_INFO

5.37.1 Description

This IE contains identification of new SW/HW Image and the URL to download it from.
Strings are NOT null terminated.

5.37.2 Data

Field	Type	Description
SW Len	U8	Software Version String Length
SW String	U8 array	15 bytes
HW Len	U8	Hardware Version String Length
HW String	U8 array	15 bytes
URL Len	U8	URL Version String Length
URL String	U8 array	255 bytes

5.38 CMND_IE_CURRENT_SW_INFO

5.38.1 Description

This IE contains identification of current SW/HW Image which is currently running.
Strings are NOT null terminated.

5.38.2 Data

Field	Type	Description
SW Len	U8	Software Version String Length
SW String	U8 array	15 bytes max
HW Len	U8	Hardware Version String Length
HW String	U8 array	15 bytes max

5.39 CMND_IE_IMAGE_TYPE

5.39.1 Description

This IE contains the image type (Node host or CMND Host)

5.39.2 Data

Field	Type	Description

Image Type	U8	DECT= 0x01 (CMND host)
		MCU=0x02 (Node host)

5.40 CMND_IE_SW_VER_INFO

5.40.1 Description

This IE contains identification of the hard-coded device SW Version.

5.40.2 Data

Field	Type	Description
Result	U8	Result of the upgrade 0x00 – Success 0x01 – General Error 0x02 – Session Error 0x03 – Gmep Send Error 0x04 – Write Error 0x05 – Erase Error 0x06 – Checksum Error 0x07 – Flash Data Error 0x08 – Call Ended 0x09 – HTTP Error 0x0A – Version Number Unchanged 0x0B – Not Enough Storage 0x0C– Battery too low 0x0D –Invalid Image Type 0x0E –Upgrade not allowed
SW Len	U8	Software Version String Length, 0x00 - 0x20
SW String	U8 array	String version, characters range 0x20 - 0xFE

5.41 CMND_IE_READ_FILE_DATA_REQ

5.41.1 Description

This IE contains offset of the next chunk to read and its length.

5.41.2 Data

Field	Type	Description
Offset	U32	Start pointer to read file
Length	U16	Number of bytes to read

5.42 CMND_IE_READ_FILE_DATA_RES

5.42.1 Description

This IE contains the data read from the image file.

5.42.2 Data

Field	Type	Description
Offset	U32	Start pointer to read file
Length	U16	Number of bytes to read NOTE: This value may be less than what was requested in read request. This does not indicate on end of file. Zero bytes indicate on end of file.
Data	U8 array	167 bytes

5.43 CMND_IE_CREATE_ATTR_REPORT_RESPONSE

5.43.1 Description

This IE contains the response to CMND_MSG_ATTRREP_ADD_REPORT_IND

5.43.2 Data

Field	Type	Description
Result	U8	Value that indicates the state of the command reception/processing. 0x00 - Ok 0x01 - Fail: Not authorized 0x02 - Fail: Invalid argument 0xFE - Fail: Not enough resources 0xFF - Fail: Unknown reason
Report Type	U1	Report type: 0x00 - Periodic 0x01 – Event-driven
Report ID	U7	Report identifier

5.44 CMND_IE_U32

5.44.1 Description

This IE contains a general-purpose 32-bit value.

5.44.2 Data

Field	Type	Description
data	U32	0x00000000 - 0xFFFFFFFF

5.45 CMND_IE_U16

5.45.1 Description

This IE contains a general-purpose 16-bit value.

5.45.2 Data

Field	Type	Description
data	U16	0x0000 - 0xFFFF

5.46 CMND_IE_PMID

5.46.1 Description

This IE contains Potable Identity (PMID) for certification procedure

5.46.2 Data

Field	Type	Description
PMID1	U8	
PMID2	U8	
PMID3	U8	

5.47 CMND_IE_PORTABLE_IDENTITY

5.47.1 Description

This IE contains Potable Identity (PMID) for certification procedure

5.47.2 Data

Field	Type	Description
UseAltPIFlag	U8	
PMID1	U8	
PMID2	U8	
PMID3	U8	
PMID4	U8	
PMID5	U8	

5.48 CMND_IE_SET_ATTRIBUTE_VALUE

5.48.1 Description

This IE holds attribute ID and value in ‘set attribute’ command or in ‘set attribute’ response.

5.48.2 Data

Field	Type	Description
Attribute ID	U8	Service attribute identifier
Data length	U8	Size of attribute value in bytes
Data	U8 array	Attribute Value

5.49 CMND_IE_DEREGISTRATION_RESPONSE

5.49.1 Description

This IE holds deregistration response code from base

5.49.2 Data

Field	Type	Description
Response Code	U8	0x00 – De-registration succeeded 0x08 – Failed: Not authorized 0xFF – Failed: Unknown reason

5.50 CMND_IE_GPIO_STATE

5.50.1 Description

This IE contains GPIO (ULE or Simple) identifier and state of GPIO (low or high)

5.50.2 Data

<table>
<tr><th>Field</th><th>Type</th><th>Description</th></tr>
<tr><td>GPIO Identifier</td><td>U8</td><td>ULE GPIO identifiers:
<table>
<tr><th>ID</th><th>Pin name</th><th>Input signal</th><th>Output signal</th></tr>
<tr><td>0x01</td><td>ANA_IN1</td><td>DIG1</td><td>ULE_GPO1</td></tr>
<tr><td>0x02</td><td>ANA_IN2</td><td>DIG2</td><td>ULE_GPO2</td></tr>
<tr><td>0x03</td><td>AMP1_OUT</td><td>DIG3</td><td>ULE_GPO3</td></tr>
<tr><td>0x04</td><td>AMP2_OUT</td><td>DIG4</td><td>ULE_GPO4</td></tr>
<tr><td>0x05</td><td>DOUBCAP2P</td><td>DIG5</td><td>ULE_GPO5</td></tr>
<tr><td>0x06</td><td>LEDSINK2</td><td>DIG6</td><td>ULE_GPO6</td></tr>
</table>
NOTE: for “set” API refer to Output signal and for “get” API refer to Input signal
For example:
<ul>
<li>setting ANA_IN1 to high, means setting ULE_GPO1 to high, therefore use ID = 0x01</li>
<li>getting the value of ANA_IN2, means getting value of input signal DIG2, therefore use ID = 0x02</li>
</ul>
Simple GPIO identifiers - Use DHX91 GPIO ids (e.g. 1,2, …)</td></tr>
<tr><td>GPIO State</td><td>U8</td><td>0x00 – Low
0x01 – High
0xFF – Unknown</td></tr>
</table>

5.51 CMND_IE_LINK_MAINTAIN

5.51.1 Description

This IE contains duration and pint interval of link maintain. For example, if duration is 10,000 and ping interval is 300: link maintain will be held for 10seconds, and every 300ms an empty SDU is sent.

5.51.2 Data

Field	Type	Description
Link duration	U16	Total duration of link maintain in Seconds.
Ping interval	U16	Defines the interval for sending empty SDU (milliseconds)

5.52 CMND_IE_ULE_CALL_SETTING

5.52.1 Description

This IE contains the ULE voice call settings. It can contain calling destination / codec.

5.52.2 Defines and constants

5.52.2.1 Voice Call Codecs

Define	Value	Description
CMND_MSG_ULE_CALL_CODEC_NB_BASIC	0x00	Narrow Band voice call codec
CMND_MSG_ULE_CALL_CODEC_WB_G722	0x01	Wide Band voice call codec

5.52.2.2 Voice Call Field Masks

Define	Value	Description
ULE_CALL_IE_PREFERRED_CODEC_MASK	0x00000001	Mask indicating preferred codec is set
ULE_CALL_IE_DIGITS_MASK	0x00000010	Mask indicating destination dial digits is set
ULE_CALL_IE_OTHER_PARTY_TYPE_MASK	0x00000100	Mask indicating destination party type is set
ULE_CALL_IE_OTHER_PARTY_NAME_MASK	0x00001000	Mask indicating destination party name set
ULE_CALL_IE_OTHER_PARTY_ID_MASK	0x00010000	Mask indicating destination party id set
ULE_CALL_IE_CALL_TYPE_MASK	0x00100000	Mask indicating *'Call Type'* field is set

5.52.3 Data

Field	Type	Description
Field mask	u32	Mask indicating which fields are set: Preferred codec - ULE_CALL_IE_PREFERRED_CODEC_MASK Digit - ULE_CALL_IE_DIGITS_MASK Other part type - ULE_CALL_IE_OTHER_PARTY_TYPE_MASK

Field	Type	Description
		Other party name - ULE_CALL_IE_OTHER_PARTY_NAME_MASK Other party Id - ULE_CALL_IE_OTHER_PARTY_ID_MASK
Preferred Codec	u8	Preferred voice codec to use: CMND_MSG_ULE_CALL_CODEC_NB_BASIC CMND_MSG_ULE_CALL_CODEC_WB_G722
Other Party Type	u8	Type of call destination, this is a single character that identifies the other party. Can be 'h' for handset for example or 'd' for device
Digits Length	u8	Number of digits of call destination to dial
Digits	u8[30]	Digits to dial of call destination
Other Party Name Length	u8	Number of characters of call destination name
Other Party Name	u8[30]	Characters of call destination name
Other Party Id Length	u8	Number of digits of call destination id
Other Party Id	u8[30]	Digits of call destination id
Call Type	u8	Bit 0=Call Direction: (0=outgoing, 1=incoming) Bit 1=Should Ring: (0=no,1=yes) Bit 2=Auto Answer: (0=no,1=yes)

The '*Field mask*' field is defining which other field in IE are present. For example, if '*Field mask*' equals 0x00011 then the two fields: '*Preferred Codec*' and '*Digits*' are set in IE payload.

5.53 CMND_IE_CHECK_SW_VER_RES

5.53.1 Description

This IE contains a result of Check Version command and the identification of SW/HW Image currently available for update. Strings are NOT null terminated.

5.53.2 Data

Field	Type	Description
Result	U8	Result of the version check 0x00 – New version available 0x11 – No newer version available 0x12 – Server or Network are down 0xFF – Fail: Unknown Error
SW Len	U8	Software Version String Length
SW String	U8 array	15 bytes max
HW Len	U8	Hardware Version String Length
HW String	U8 array	15 bytes max
URL Len	U8	URL Version String Length
URL String	U8 array	100 bytes max

5.54 CMND_IE_BROADCAST_CHANNEL_ALLOCATION

5.54.1 Description

This IE contains indication on assigned broadcasting channel.

5.54.2 Data

Field	Type	Description
Channel Periodicity	U8	Channel Periodicity
Paging ID	U16	Paging ID
Multicast Channel ID	U16	Multicast Channel ID

5.55 CMND_IE_BROADCAST_GROUP_TABLE_OPERATION

5.55.1 Description

This IE contains indication on any change on the group table (adding or removing entries).

5.55.2 Data

Field	Type	Description
Operation Id	U8	See 5.55.3 Group Table Operation Id
Group Table Entry	Structure	See 5.57.3 Group Table Entry

5.55.3 Group Table Operation Id

Value	Description
0x00	Entry Added
0x01	Entry Deleted
0x02	All Entries deleted

5.56 CMND_IE_BROADCAST_GROUP_TABLE_RANGE

5.56.1 Description

This IE contains group table range, i.e. group table entry start index and number of entries.

5.56.2 Data

Field	Type	Description
Start Index	U8	Start Index
Number Of Entries	U8	Number Of Entries

5.57 CMND_IE_BROADCAST_GROUP_TABLE_ENTRIES

5.57.1 Description

This IE contains client response on node host request for read group table entries.

5.57.2 Data

Field	Type	Description
Start Index	U8	Start Index
Number Of Entries	U8	Number Of Entries
List of Group Table Entries	Structure	See 5.57.3 Group Table Entry

5.57.3 Group Table Entry

Field	Type	Description
Group ID	U16	Group ID

Unit ID	U8	Unit ID

6. CMND API Parameters

This section defines CMND API parameters that are accessed via CMND API Production and Parameter services.

PARAM ID	NAME	TYPE	VALUES	ACCESS
0x16	System off mode	U8	Defines system off mode: 0x10 – hibernation will be executed locally 0x11 – MCU must send sleep request to device to enter hibernation (See CMND API Sleep service) 0x12 – None, device does not hibernate (default)	RW
0x1C	Minimum sleep time	U32	Device information attribute: Paging interval for page-able devices, in milliseconds. Big endian byte order	RW
0x29	Keep alive interval	U32	Keep alive interval, in milliseconds Little endian byte order, For 10sec = 0x2710 use 10 27 00 00	RW
0x2A	Registration status	U8	Registration status 0x05 – Registered 0x06 – Not registered (default value)	RW

7. Handshake

7.1 Overview

The ULE DECT (CMND host) module has several hibernation methods:

1. Non-hibernating device- this is an always ON device which does not hibernate. For example, an AC operated device.
2. Hibernation controlled by MCU/Node Host – this is a hibernating device; the hibernation is triggered by an explicit command from MCU (see Sleep Service CMND_MSG_SLEEP_ENTER_SLEEP_REQ message).
3. Standalone hibernation – this is a hibernating device; entering hibernation is managed by CMND node internally.

This configuration is stored in ULE DECT module (CMND) EEPROM (see CMND API parameter “System off mode”). If the device is a hibernating device (methods 2 and 3) a handshake procedure is necessary between the ULE DECT module and the MCU (Node host).

The goal of handshake is to make sure ULE DECT module is up and running and not in hibernation. Before sending a message from MCU to ULE DECT module, the MCU must first wakeup the DECT module, wait for confirmation and only then send the message.

General approach is creating a signal on ULE GPIO to wake the ULE DECT module. When ULE DECT module is ready it will respond a CMND API General service message CMND_MSG_GENERAL_HELLO_IND. Now it’s safe to communicate with the ULE DECT module.

See the Flow and Sequences Handshake and Sleep Flow chapter for handshake sequences.

7.2 Sanity Reset on Node Host

The following procedure is suggested to be implemented on the Node host (MCU) side. The goal is to reset the CMND in case it does not respond to wakeup event. The node host will toggle an ULE GPIO and wait for hello indication as part of the handshake sequence. It is suggested to reset the CMND if CMND does not respond after N handshake requests.

8. Flow and Sequences

8.1 Auto Triggered Pairing Flow

In the example shown in Figure 8-1 , the registration is triggered automatically by the CMND. Upon power on, the CMND detects if HAN EEPROM is valid (information about device type, unit types, interfaces …) and if so, starts the DECT registration. When registration (DECT + ULE Negotiation + FUN registration) ends, the CMND indicates to the Node Host of registration status.

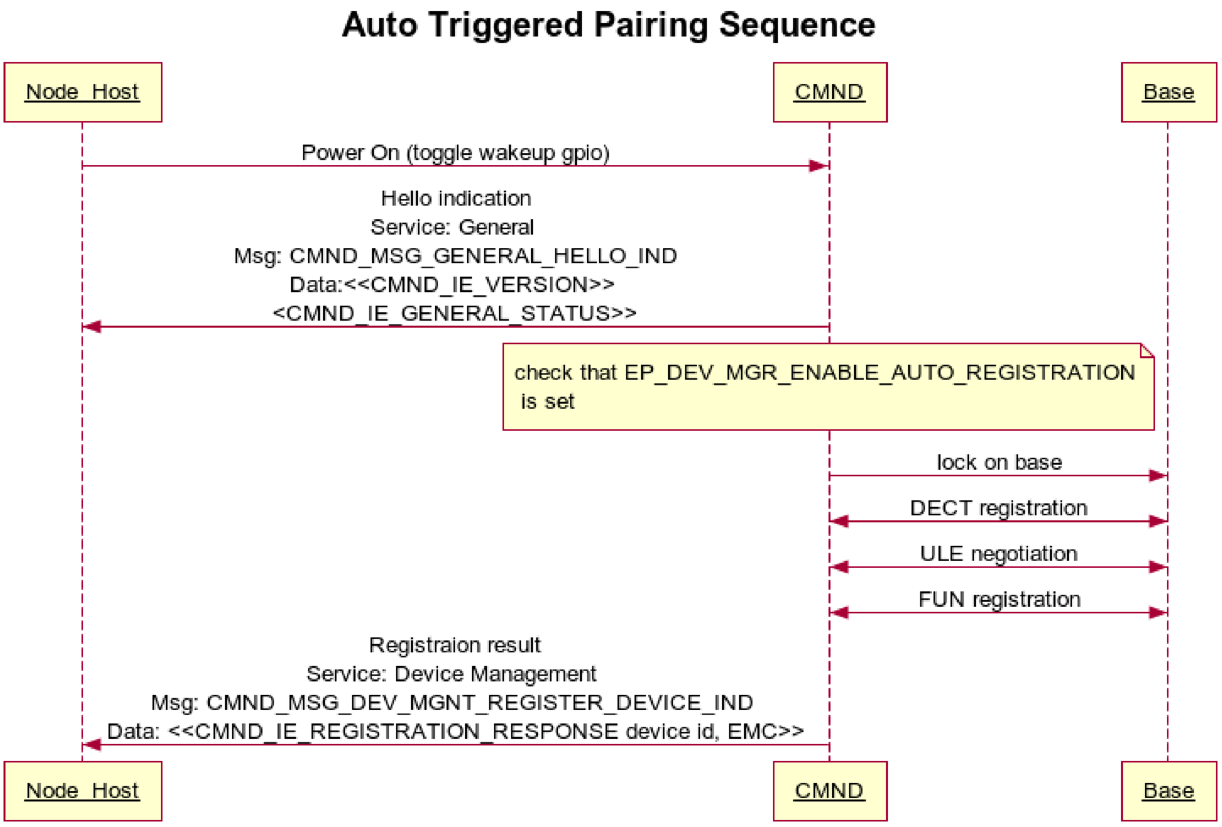

Auto Triggered Pairing Sequence
Node_Host
CMND
Base
Power On (toggle wakeup gpio)
Hello indication
Service: General
Msg: CMND_MSG_GENERAL_HELLO_IND
Data:<<CMND_IE_VERSION>>
<CMND_IE_GENERAL_STATUS>>
check that EP_DEV_MGR_ENABLE_AUTO_REGISTRATION is set
lock on base
DECT registration
ULE negotiation
FUN registration
Registraion result
Service: Device Management
Msg: CMND_MSG_DEV_MGNT_REGISTER_DEVICE_IND
Data: <<CMND_IE_REGISTRATION_RESPONSE device id, EMC>>
Node_Host
CMND
Base

Figure 8-1: Example of Auto Triggered Pairing Flow

8.2 Pairing Triggered by Node Host

In the example shown below the registration is triggered from the Node Host by pressing a button or placing a magnet. Node Host wakes the CMND and sends registration request. The CMND confirms that registration started, performs registration and replies to the Node Host with registration result.

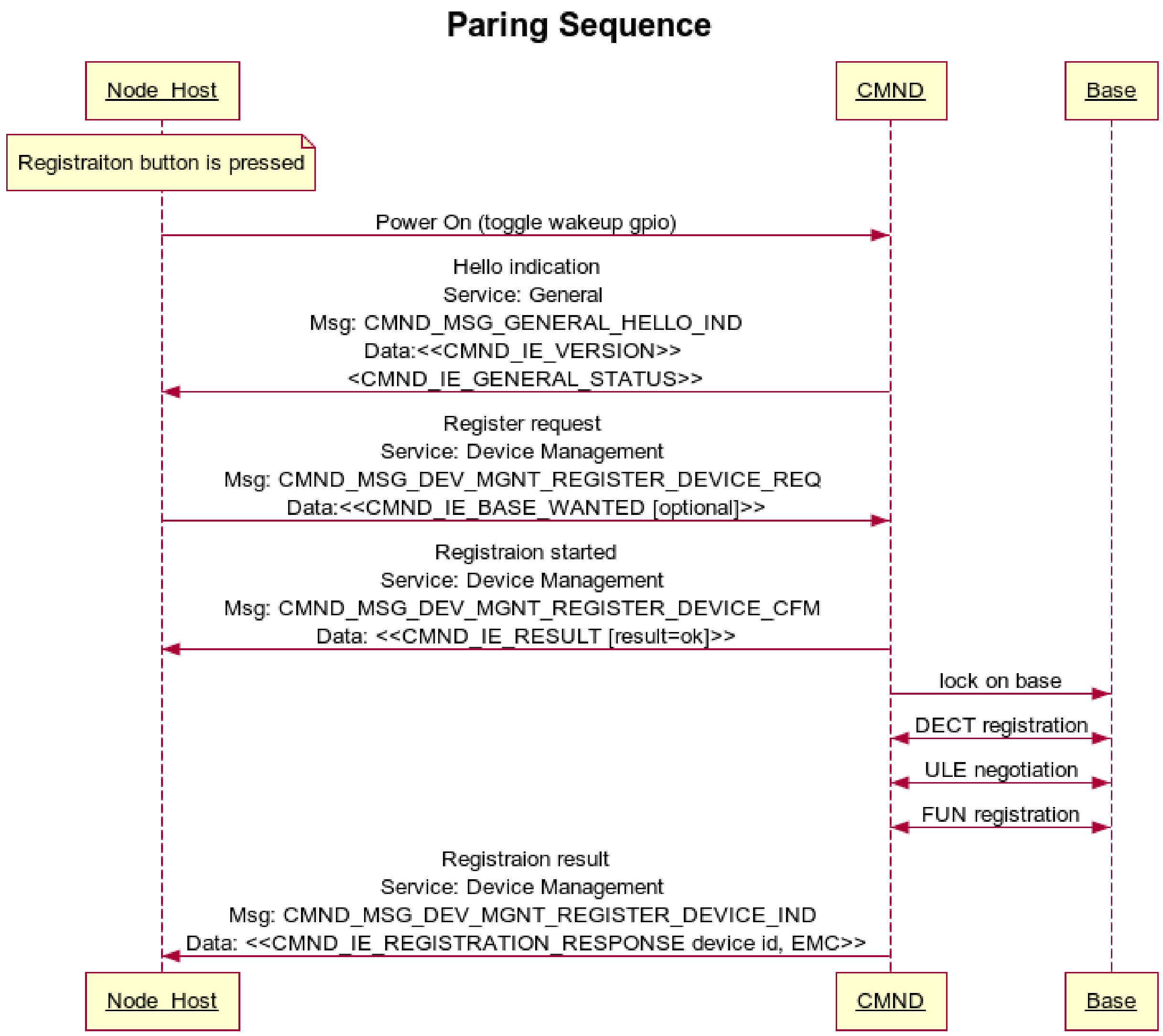

Paring Sequence
Node_Host
CMND
Base
Registraiton button is pressed
Power On (toggle wakeup gpio)
Hello indication
Service: General
Msg: CMND_MSG_GENERAL_HELLO_IND
Data:<<CMND_IE_VERSION>>
<CMND_IE_GENERAL_STATUS>>
Register request
Service: Device Management
Msg: CMND_MSG_DEV_MGNT_REGISTER_DEVICE_REQ
Data:<<CMND_IE_BASE_WANTED [optional]>>
Registraion started
Service: Device Management
Msg: CMND_MSG_DEV_MGNT_REGISTER_DEVICE_CFM
Data: <<CMND_IE_RESULT [result=ok]>>
lock on base
DECT registration
ULE negotiation
FUN registration
Registraion result
Service: Device Management
Msg: CMND_MSG_DEV_MGNT_REGISTER_DEVICE_IND
Data: <<CMND_IE_REGISTRATION_RESPONSE device id, EMC>>
Node_Host
CMND
Base

Figure 8-2: Example of Pairing Triggered by Node Host Flow

8.3 First-**Time** DECT and HAN Configuration Flow

In Figure 8-3, the Node Host sets the CMND parameter on the first activation; i.e., the first time that the battery is inserted.

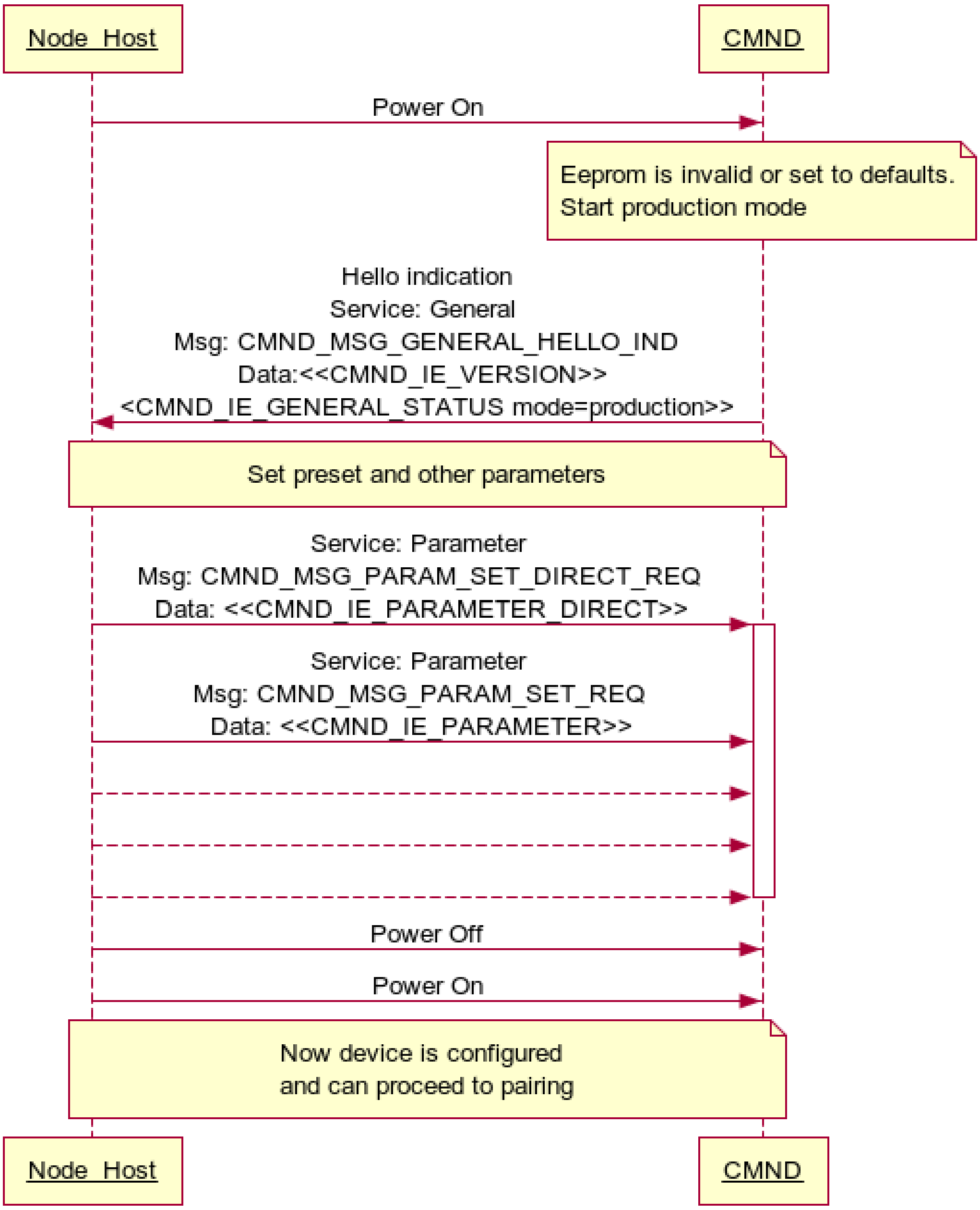

Node_Host
CMND
Power On
Eeprom is invalid or set to defaults.
Start production mode
Hello indication
Service: General
Msg: CMND_MSG_GENERAL_HELLO_IND
Data:<<CMND_IE_VERSION>>
<CMND_IE_GENERAL_STATUS mode=production>>
Set preset and other parameters
Service: Parameter
Msg: CMND_MSG_PARAM_SET_DIRECT_REQ
Data: <<CMND_IE_PARAMETER_DIRECT>>
Service: Parameter
Msg: CMND_MSG_PARAM_SET_REQ
Data: <<CMND_IE_PARAMETER>>
Power Off
Power On
Now device is configured
and can proceed to pairing
Node_Host
CMND

Figure 8-3: First-Time DECT Configuration Flow

8.4 General Event Flow

In Figure 8-4, an Alert event occurs on the Node Host. It wakes up the CMND and sends a message to it. The CMND answers with DECT link confirmation and later a FUN response arrives to confirm that the Alert event was handled.

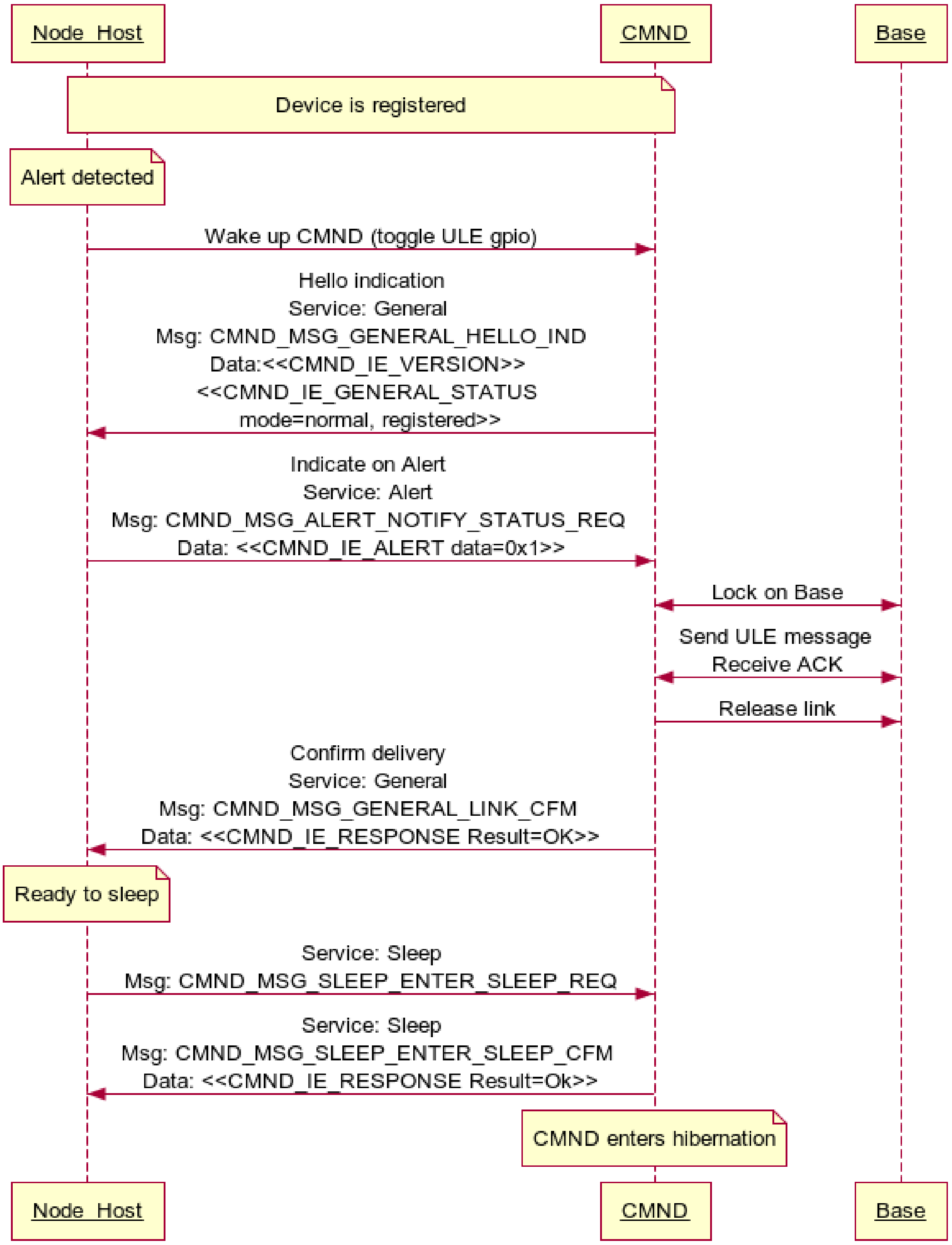

Node_Host
CMND
Base
Device is registered
Alert detected
Wake up CMND (toggle ULE gpio)
Hello indication
Service: General
Msg: CMND_MSG_GENERAL_HELLO_IND
Data:<<CMND_IE_VERSION>>
<<CMND_IE_GENERAL_STATUS
mode=normal, registered>>
Indicate on Alert
Service: Alert
Msg: CMND_MSG_ALERT_NOTIFY_STATUS_REQ
Data: <<CMND_IE_ALERT data=0x1>>
Lock on Base
Send ULE message
Receive ACK
Release link
Confirm delivery
Service: General
Msg: CMND_MSG_GENERAL_LINK_CFM
Data: <<CMND_IE_RESPONSE Result=OK>>
Ready to sleep
Service: Sleep
Msg: CMND_MSG_SLEEP_ENTER_SLEEP_REQ
Service: Sleep
Msg: CMND_MSG_SLEEP_ENTER_SLEEP_CFM
Data: <<CMND_IE_RESPONSE Result=Ok>>
CMND enters hibernation
Node_Host
CMND
Base

Figure 8-4: General Event Flow

8.5 Handshake and Sleep Flow

See Figure 8-5 and Figure 8-6 for flowcharts of wakeup of CMND or the Node Host.

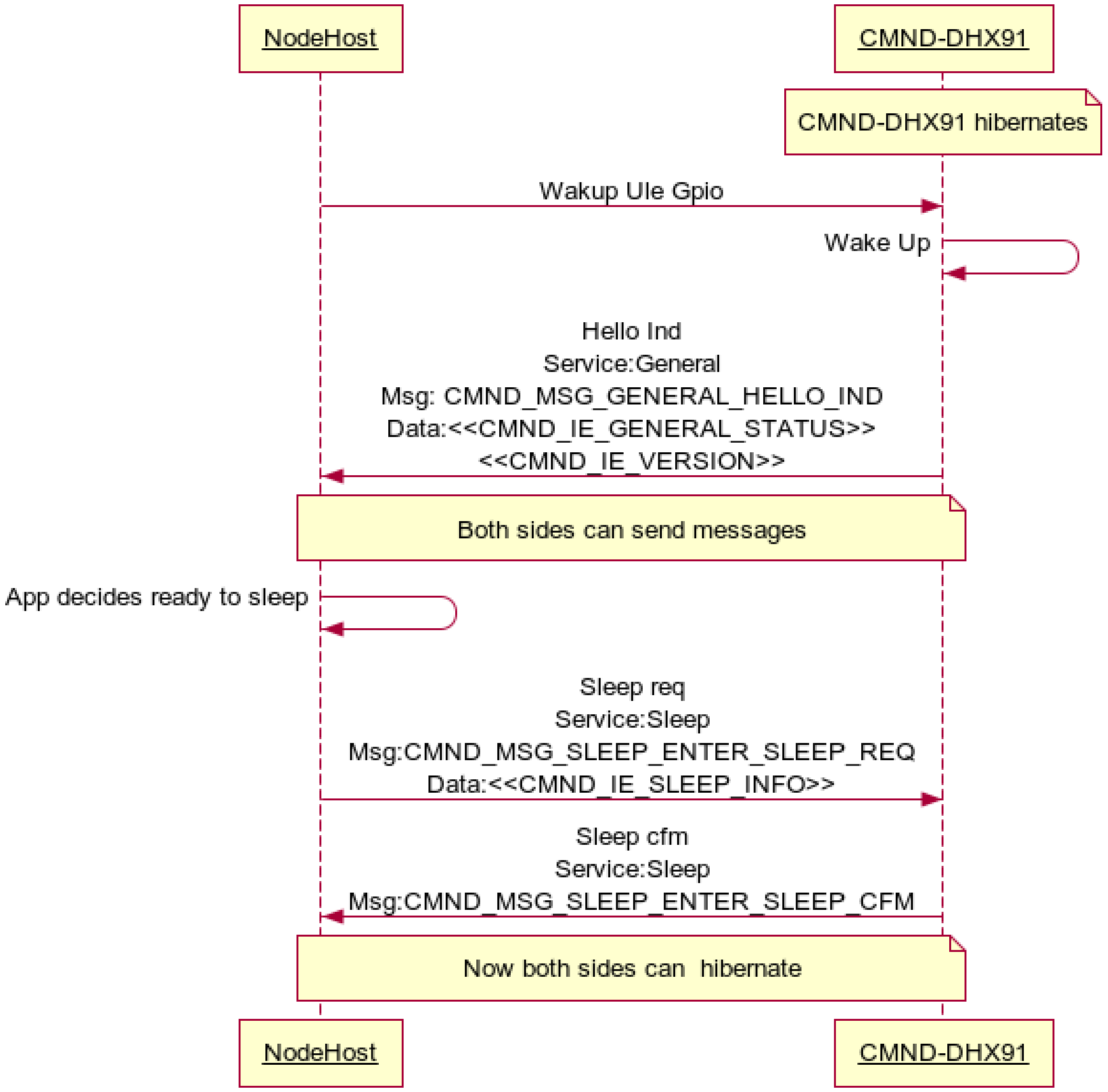

NodeHost
CMND-DHX91
CMND-DHX91 hibernates
Wakup Ule Gpio
Wake Up
Hello Ind
Service:General
Msg: CMND_MSG_GENERAL_HELLO_IND
Data:<<CMND_IE_GENERAL_STATUS>>
<<CMND_IE_VERSION>>
Both sides can send messages
App decides ready to sleep
Sleep req
Service:Sleep
Msg:CMND_MSG_SLEEP_ENTER_SLEEP_REQ
Data:<<CMND_IE_SLEEP_INFO>>
Sleep cfm
Service:Sleep
Msg:CMND_MSG_SLEEP_ENTER_SLEEP_CFM
Now both sides can hibernate
NodeHost
CMND-DHX91

Figure 8-5: Node Host Wakes CMND

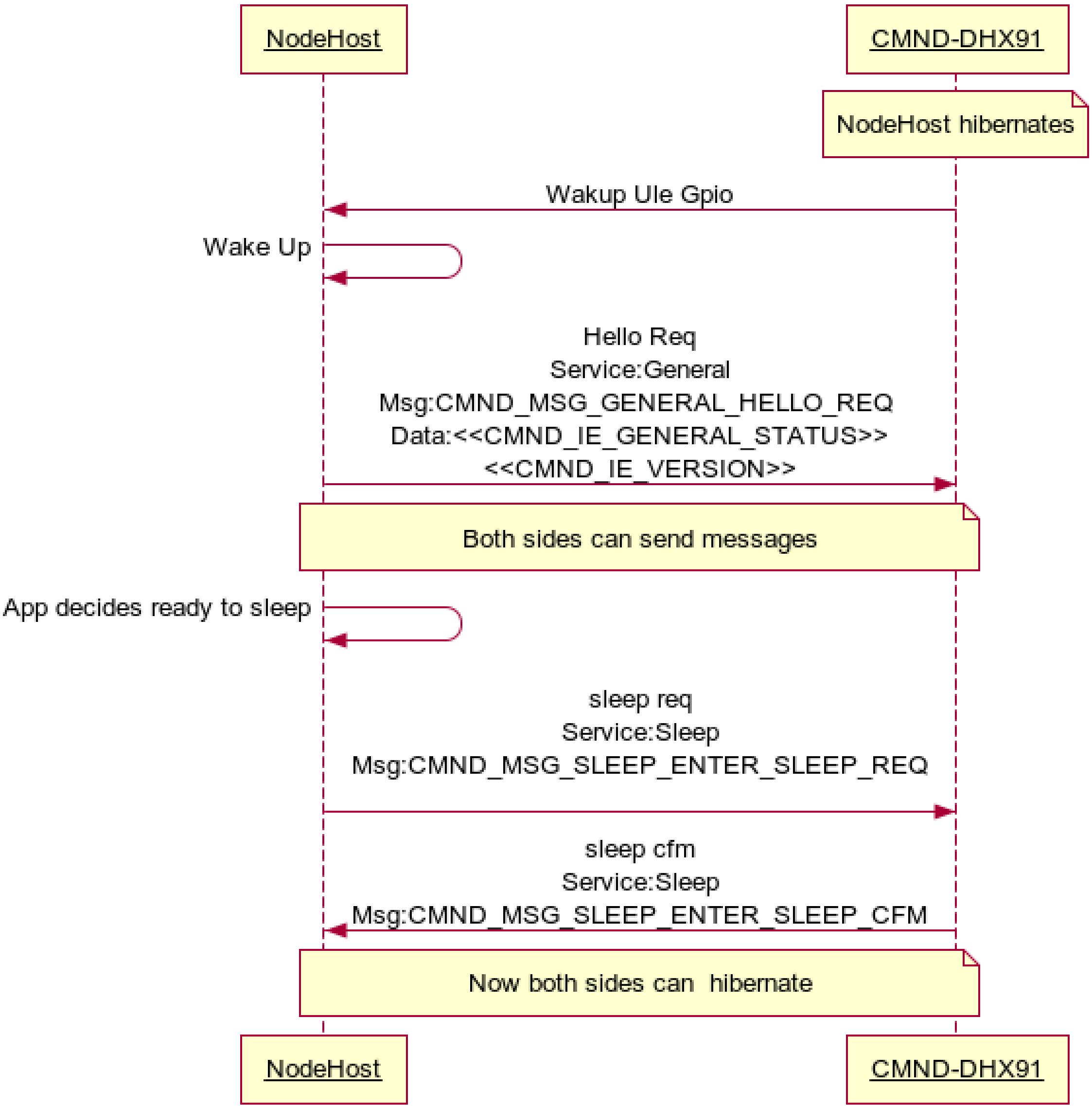

NodeHost
CMND-DHX91
NodeHost hibernates
Wakup Ule Gpio
Wake Up
Hello Req
Service:General
Msg:CMND_MSG_GENERAL_HELLO_REQ
Data:<<CMND_IE_GENERAL_STATUS>>
<<CMND_IE_VERSION>>
Both sides can send messages
App decides ready to sleep
sleep req
Service:Sleep
Msg:CMND_MSG_SLEEP_ENTER_SLEEP_REQ
sleep cfm
Service:Sleep
Msg:CMND_MSG_SLEEP_ENTER_SLEEP_CFM
Now both sides can hibernate
NodeHost
CMND-DHX91

Figure 8-6: CMND Wakes Node Host

8.6 Message Flow for FUN Service

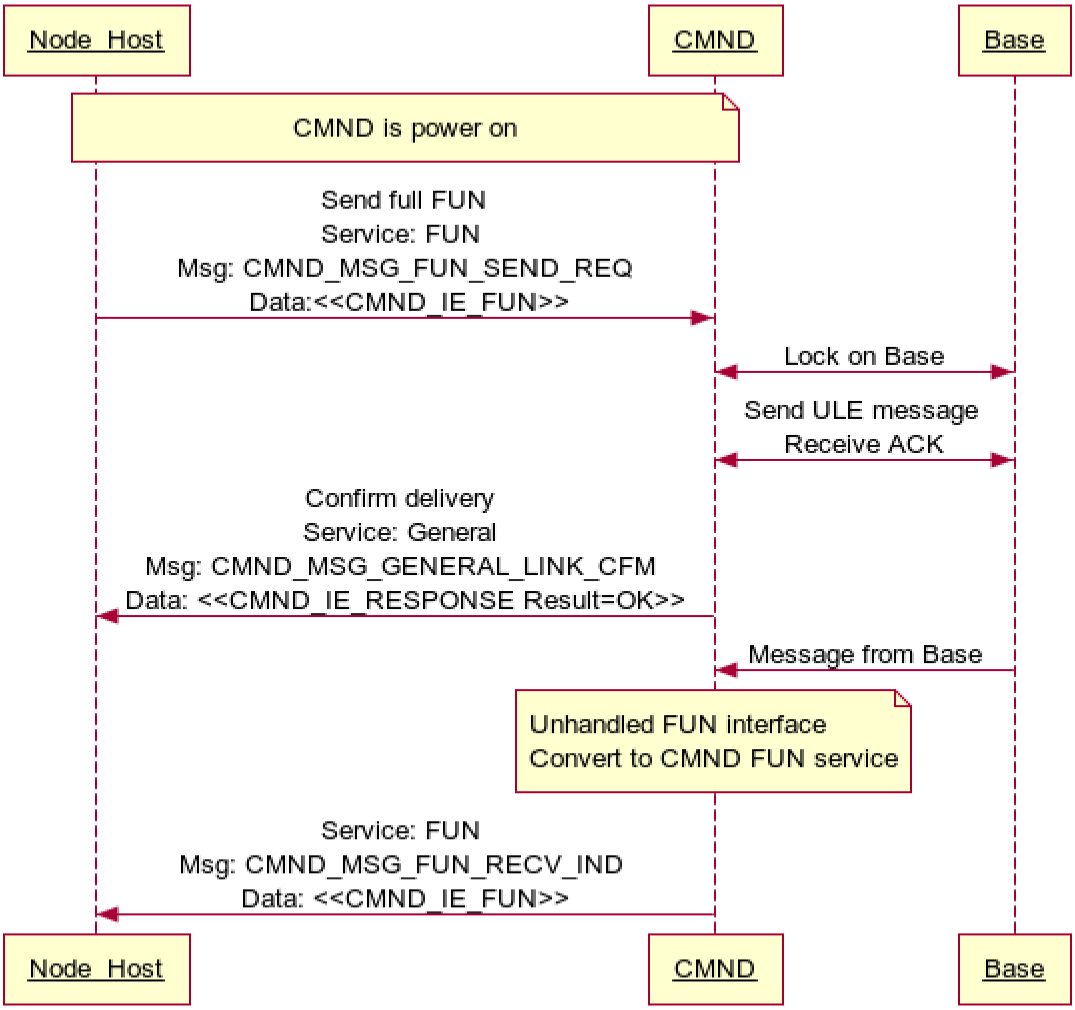


Figure 8-7: Message Flow for FUN Service

8.7 Production Flow

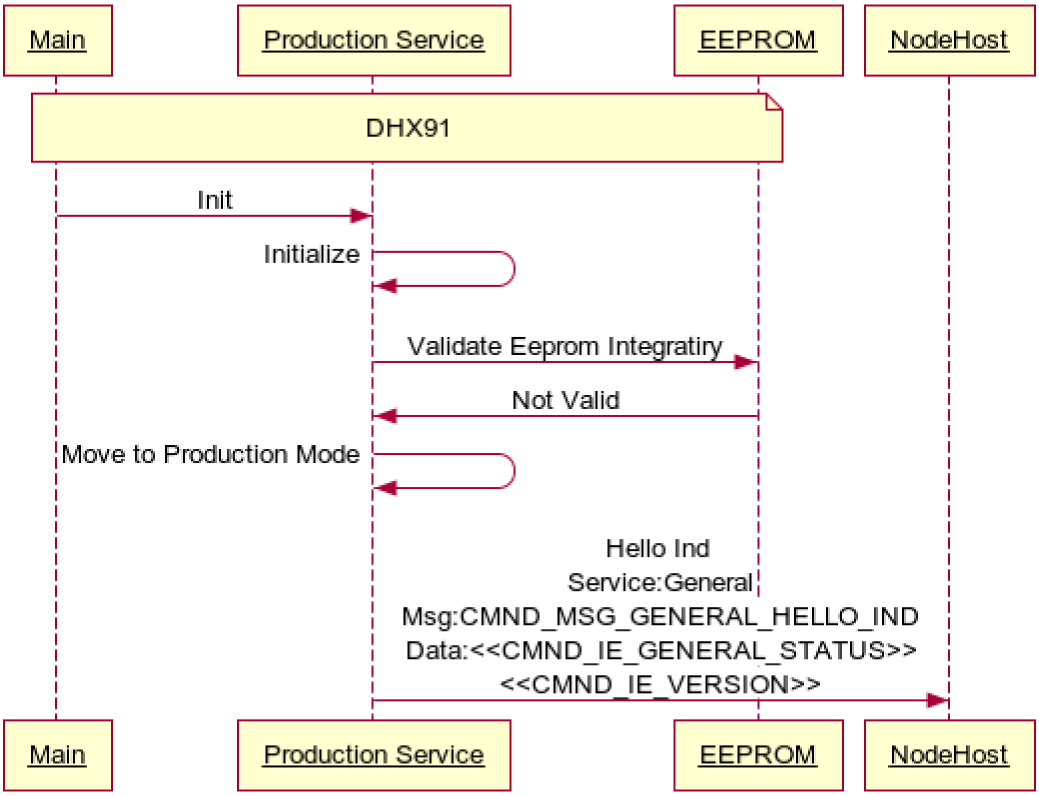


Figure 8-8: Message Flow for FUN Service

8.8 Using Link Maintain commands

This is general use case of using Link Maintain commands. Node Host sends an “On” command to remote device and includes the “Response required” information element to receive a response from base whether the “On” command is properly handled by the remote device. To receive a response from base, the CMND must periodically ping the base. Once a response is ready the base will respond on one of the incoming pings.

After receiving the response the Node Host should stop the link maintain explicitly by sending CMND_MSG_GENERAL_LINK_MAINTAIN_STOP_REQ message.

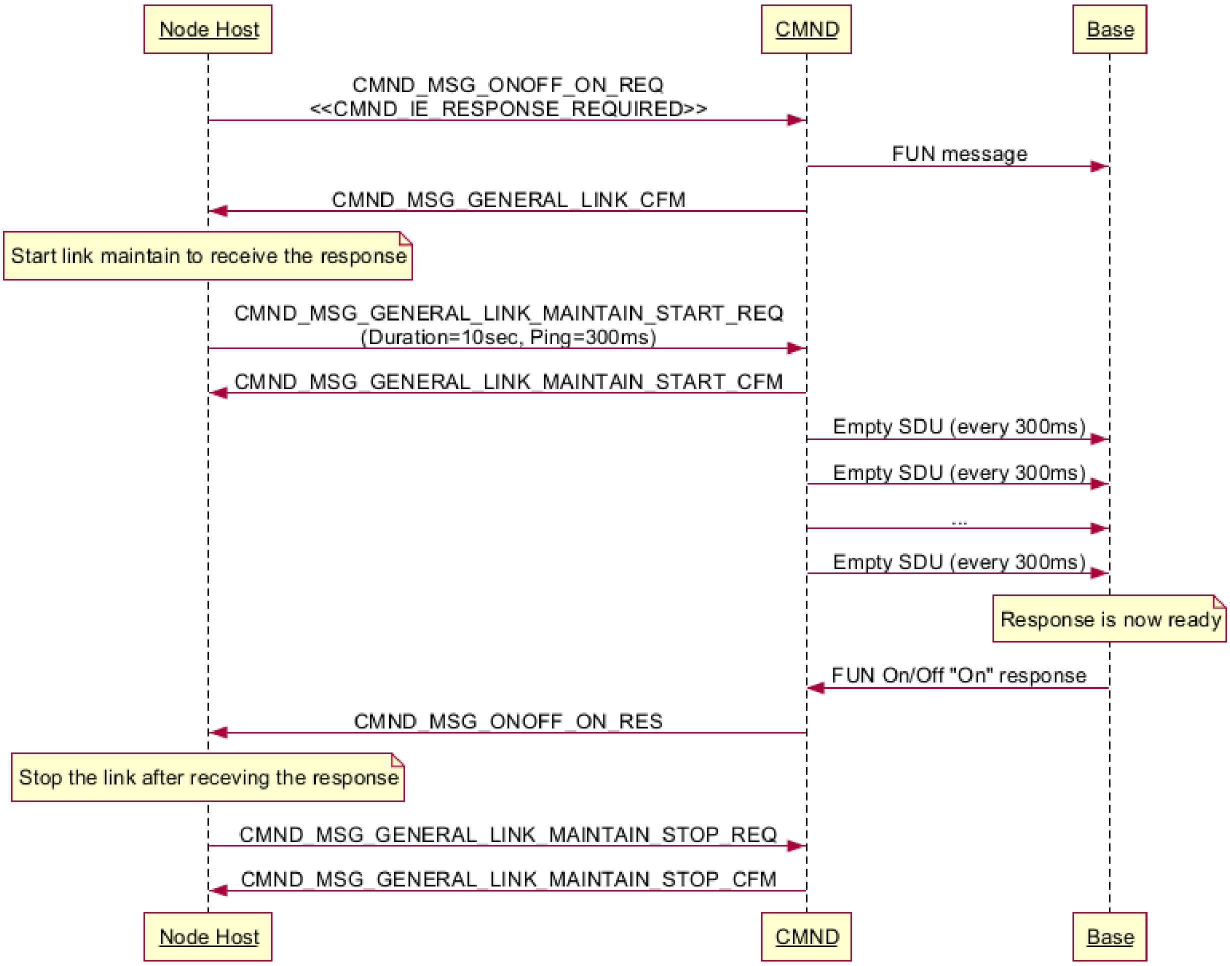

Node Host
CMND
Base
CMND_MSG_ONOFF_ON_REQ
<<CMND_IE_RESPONSE_REQUIRED>>
FUN message
CMND_MSG_GENERAL_LINK_CFM
Start link maintain to receive the response
CMND_MSG_GENERAL_LINK_MAINTAIN_START_REQ
(Duration=10sec, Ping=300ms)
CMND_MSG_GENERAL_LINK_MAINTAIN_START_CFM
Empty SDU (every 300ms)
Empty SDU (every 300ms)
...
Empty SDU (every 300ms)
Response is now ready
FUN On/Off "On" response
CMND_MSG_ONOFF_ON_RES
Stop the link after receving the response
CMND_MSG_GENERAL_LINK_MAINTAIN_STOP_REQ
CMND_MSG_GENERAL_LINK_MAINTAIN_STOP_CFM
Node Host
CMND
Base

Figure 8-9: Message Flow for Link Maintain

8.9 Link Maintain expires

In this sequence flow, Node host maintains the link, however base does not respond with the specified time. When link duration expires, the CMND will send CMND_MSG_GENERAL_LINK_MAINTAIN_STOPPED_IND message to Node Host to indicate the link is not maintained anymore.

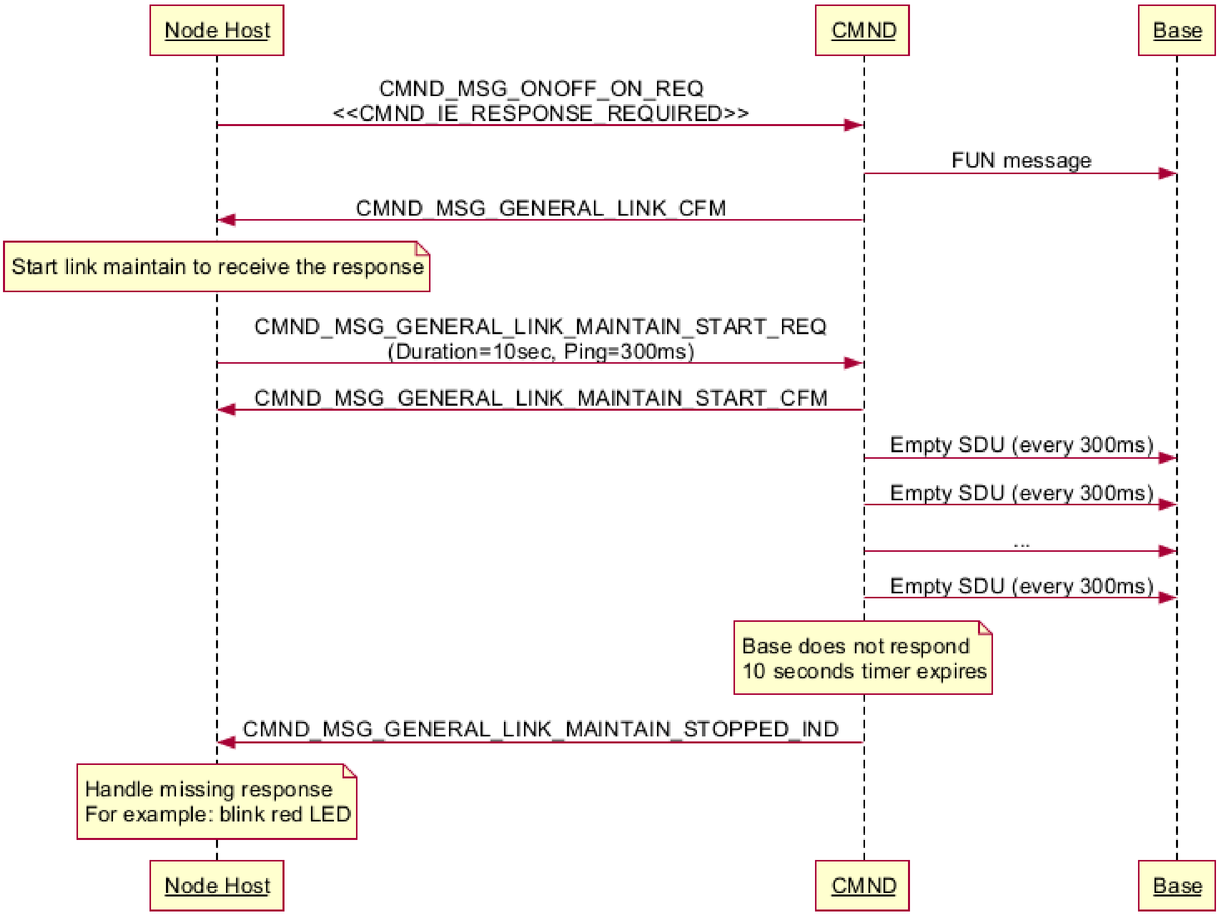

Node Host
CMND
Base
CMND_MSG_ONOFF_ON_REQ
<<CMND_IE_RESPONSE_REQUIRED>>
FUN message
CMND_MSG_GENERAL_LINK_CFM
Start link maintain to receive the response
CMND_MSG_GENERAL_LINK_MAINTAIN_START_REQ
(Duration=10sec, Ping=300ms)
CMND_MSG_GENERAL_LINK_MAINTAIN_START_CFM
Empty SDU (every 300ms)
Empty SDU (every 300ms)
...
Empty SDU (every 300ms)
Base does not respond
10 seconds timer expires
CMND_MSG_GENERAL_LINK_MAINTAIN_STOPPED_IND
Handle missing response
For example: blink red LED
Node Host
CMND
Base

Figure 8-10: Message Flow for Link Maintain expiration

8.10 Start voice call request from base

In this sequence flow base sends a FUN message to device to start voice call. The message is routed to Node Host (MCU) to accept or reject. If Node host accepts the call, the device switches to Normal Mode (circuit mode) and starts a voice call with the base.

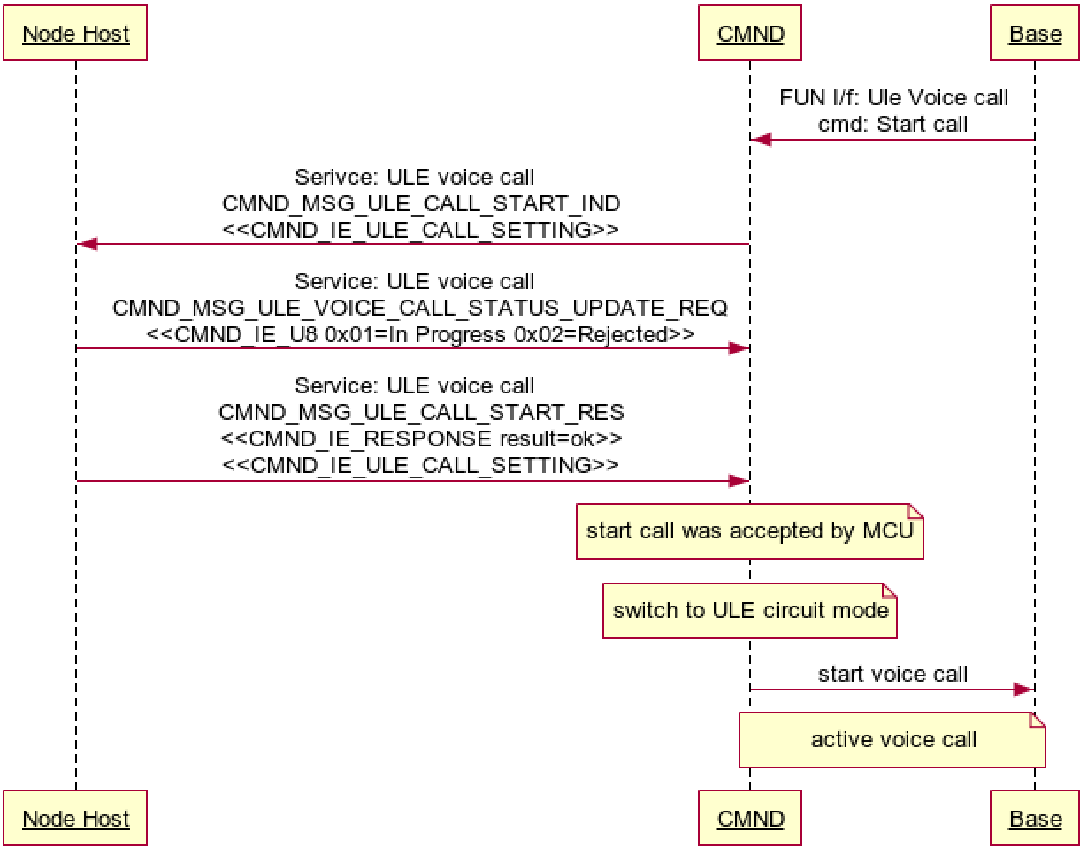

Node Host
CMND
Base
FUN I/f: Ule Voice call
cmd: Start call
Serivce: ULE voice call
CMND_MSG_ULE_CALL_START_IND
<<CMND_IE_ULE_CALL_SETTING>>
Service: ULE voice call
CMND_MSG_ULE_VOICE_CALL_STATUS_UPDATE_REQ
<<CMND_IE_U8 0x01=In Progress 0x02=Rejected>>
Service: ULE voice call
CMND_MSG_ULE_CALL_START_RES
<<CMND_IE_RESPONSE result=ok>>
<<CMND_IE_ULE_CALL_SETTING>>
start call was accepted by MCU
switch to ULE circuit mode
start voice call
active voice call
Node Host
CMND
Base

Figure 8-11: Message Flow for ULE voice call start triggered by base

8.11 Voice call remote release

In this sequence flow base releases the voice call. CMND handles call release and also send a message indication to Node host

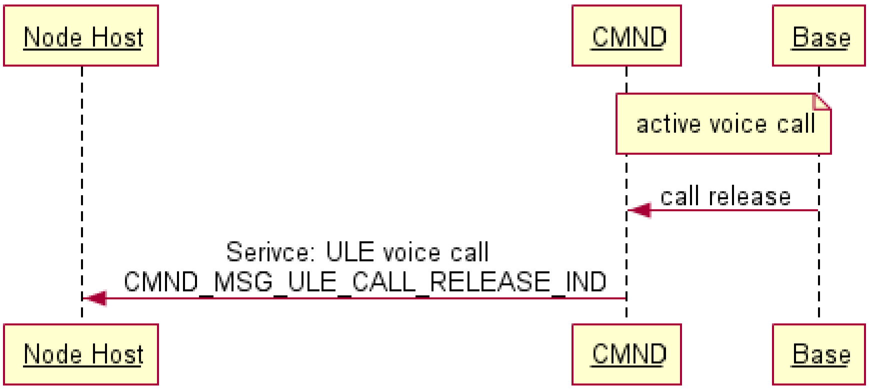


Figure 8-12: Message Flow for ULE voice call remote release

8.12 Start voice call from Node Host

In this sequence flow the Node host sends a request to start a call to the CMND. The CMND confirms receiving this request and establishes a voice call with base.

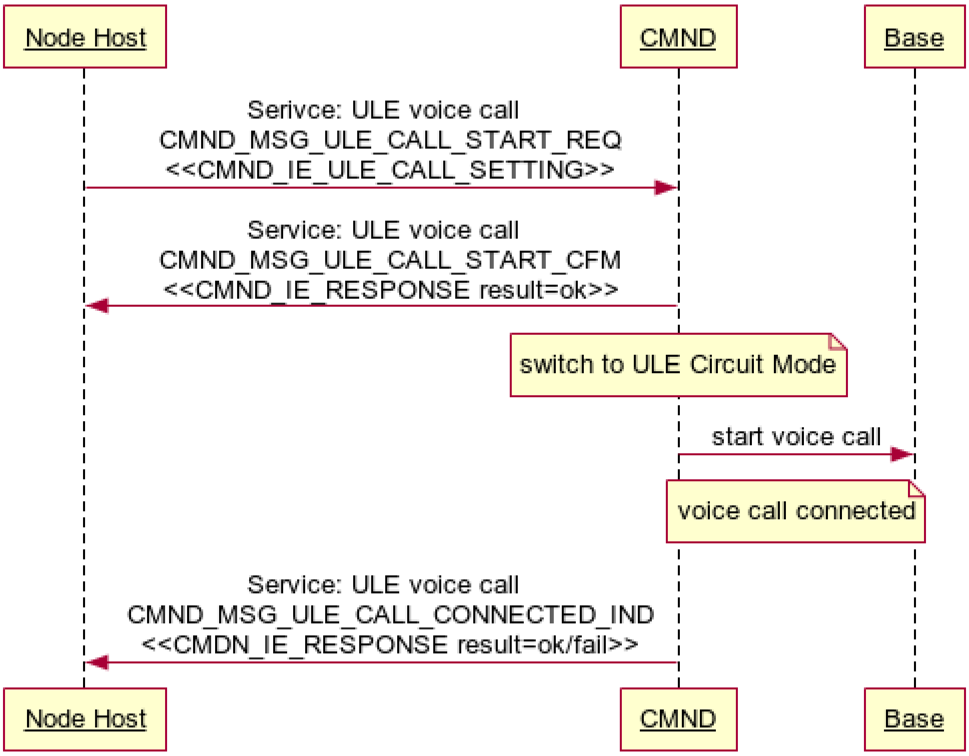

Node Host
CMND
Base
Serivce: ULE voice call
CMND_MSG_ULE_CALL_START_REQ
<<CMND_IE_ULE_CALL_SETTING>>
Service: ULE voice call
CMND_MSG_ULE_CALL_START_CFM
<<CMND_IE_RESPONSE result=ok>>
switch to ULE Circuit Mode
start voice call
voice call connected
Service: ULE voice call
CMND_MSG_ULE_CALL_CONNECTED_IND
<<CMDN_IE_RESPONSE result=ok/fail>>
Node Host
CMND
Base

Figure 8-13: Message Flow for ULE voice call start from Node Host (MCU)

8.13 Broadcast Group Table Indication from CMND

This sequence demonstrates the CMND indications to the Node host on group table entries.

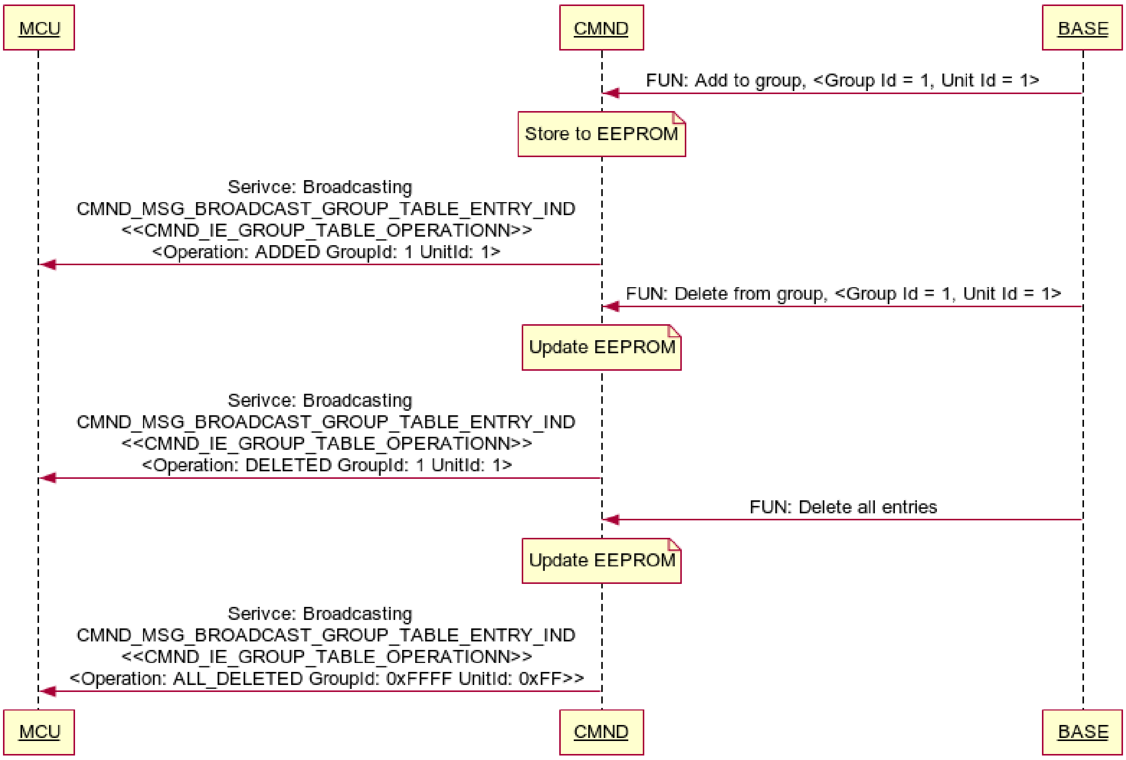

MCU
CMND
BASE
FUN: Add to group, <Group Id = 1, Unit Id = 1>
Store to EEPROM
Serivce: Broadcasting
CMND_MSG_BROADCAST_GROUP_TABLE_ENTRY_IND
<<CMND_IE_GROUP_TABLE_OPERATIONN>>
<Operation: ADDED GroupId: 1 UnitId: 1>
FUN: Delete from group, <Group Id = 1, Unit Id = 1>
Update EEPROM
Serivce: Broadcasting
CMND_MSG_BROADCAST_GROUP_TABLE_ENTRY_IND
<<CMND_IE_GROUP_TABLE_OPERATIONN>>
<Operation: DELETED GroupId: 1 UnitId: 1>
FUN: Delete all entries
Update EEPROM
Serivce: Broadcasting
CMND_MSG_BROADCAST_GROUP_TABLE_ENTRY_IND
<<CMND_IE_GROUP_TABLE_OPERATIONN>>
<Operation: ALL_DELETED GroupId: 0xFFFF UnitId: 0xFF>>
MCU
CMND
BASE

Figure 8-14: Message Flow for a group table indication from CMND

8.14 Broadcast Group Table Indication from CMND

This sequence demonstrates a broadcast message sent by base to 2 registered devices. The broadcast is a single message broadcasted over the air, captured by each device and propagated to the MCU. The broadcast is addressed to group 1 and only devices that have entry of group 1 are handling this message.

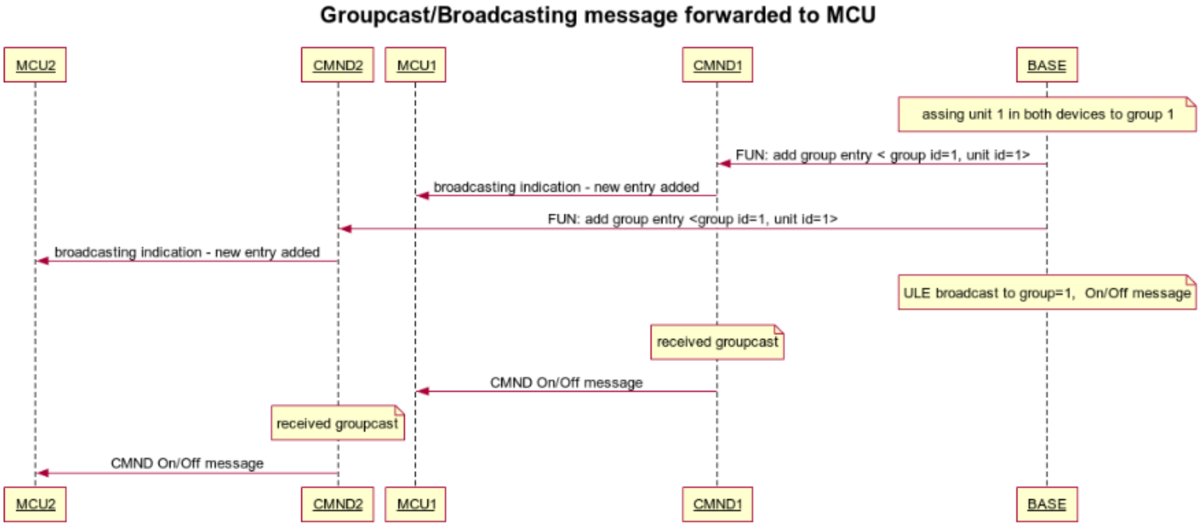

Groupcast/Broadcasting message forwarded to MCU
MCU2
CMND2
MCU1
CMND1
BASE
assing unit 1 in both devices to group 1
FUN: add group entry < group id=1, unit id=1>
broadcasting indication - new entry added
FUN: add group entry <group id=1, unit id=1>
broadcasting indication - new entry added
ULE broadcast to group=1, On/Off message
received groupcast
CMND On/Off message
received groupcast
CMND On/Off message
MCU2
CMND2
MCU1
CMND1
BASE

Figure 8-15: Message Flow for a group table indication from CMND

9. Timeouts

Both the Node Host and the CMND should run watchdog algorithm timeouts, which should be configurable.

9.1 Node Host Timeouts

Table 9-1: Node Host Timer List

NAME	SET AT	STOP AT	EXPIRE ACTION	DURATION
Wakeup	WAKEUP	Any message from CMND	Shut down and restart	TBD

Wait to Pair	Registration request	Registration success or failure	Shut down and restart	DECT = 60sec
				ULE= 13sec
				FUN= 8sec

Wait for CFM of delivery to BS	Host sends CMND a request message (e.g "Alert ON")	CFM message from CMND	Resend message (N repetitions)	TBD
Wait for the required response	Host receives CFM message for request	RES message from CMND	Resend message (N repetitions)	TBD

9.2 CMND Timeouts

Table 9-2: CMND Timer List

NAME	SET AT	STOP AT	EXPIRE ACTION	DURATION
Wait for required response	CMND sends Host message requiring response	RES message from Node Host	Resend message (M repetitions)	TBD
Wait for deregistration response from base	Set when deregistrtion request is receive from Node Host	When Deregistration response is received from base or timeout expires	Send deregistration indication to Node Host	10seconds

10. Appendices

10.1 Appendix 1: Example log of Successful SUOTA for CMND node initiated by BS

MCU: Here is an Alert

```
MCU --> CMND:
Message Length: 19
Raw message: da da 00 0f 68 01 01 00 03 8f 06 00 06 02 04 00 00 00 01
Service ID: CMND_SERVICE_ID_ALERT<01 00>
Message ID: CMND_MSG_ALERT_NOTIFY_STATUS_REQ<03>
Unit ID: 01
Cookie: 68
Raw IE: 00 06 02 04 00 00 00 01
Parsed IE: CMND_IE_ALERT<0x06>  Length<6>  Value<UnitType: FUN_UNIT_TYPE_SMOKE_DETECTOR;
    AlertStatus HAN_CMND_ALERT_STATE_ALERTING >
```

```
              CMND --> MCU:
              Message Length: 14
              Raw message: da da 00 0a 68 01 00 00 07 7b 00 00 01 00
              Service ID: CMND_SERVICE_ID_GENERAL<00 00>
              Message ID: CMND_MSG_GENERAL_LINK_CFM<07>
              Unit ID: 01
              Cookie: 68
              Raw IE: 00 01 00
              Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<CMND_RC_OK>
```

CMND: Ok, I sent it to BS

```
              CMND --> MCU:
              Message Length: 59
              Raw message: da da 00 37 00 00 02 0c 01 1b 32 00 24 04 44 45 43 54 04 64 68 61 6e 19
                  68 74 74 70 3a 2f 2f 31 39 32 2e 31 36 38 2e 31 2e 31 2f 49 54 43 4d 5f 48 03 00
                  03 00 00 02 13 00 01 05
              Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
              Message ID: CMND_MSG_SUOTA_NEW_SW_AVAIALBE_IND<01>
              Unit ID: 00
              Cookie: 00
              Raw IE: 00 24 04 44 45 43 54 04 64 68 61 6e 19 68 74 74 70 3a 2f 2f 31 39 32 2e 31 36
                  38 2e 31 2e 31 2f 49 54 43 4d 5f 48
              Parsed IE: CMND_IE_NEW_SW_INFO<0x32>  Length<36>  Value<SwLen: 4; SwVer: <DECT>;
                  HwLen: 4; HwVer: <dhan>; UrlLen: 25; URL: <http://192.168.1.1/ITCM_H>>
              Raw IE: 00 03 00 00 02
              Parsed IE: CMND_IE_UNIT_ADDR<0x03>  Length<3>  Value<000002>
              Raw IE: 00 01 05
              Parsed IE: CMND_IE_OTA_COOKIE<0x13>  Length<1>  Value<05>
```

CMND: BTW, new software for CMND is available!

MCU: Understood and fine by me.

```
MCU --> CMND:
Message Length: 14
Raw message: da da 00 0a 00 00 02 0c 02 1b 00 00 01 00
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_NEW_SW_RES<02>
Unit ID: 00
Cookie: 00
Raw IE: 00 01 00
```

```
Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<CMND_RC_SUOTA_SUCCESS>

              CMND --> MCU:
              Message Length: 14
              Raw message: da da 00 0a 00 00 00 00 07 12 00 00 01 00
              Service ID: CMND_SERVICE_ID_GENERAL<00 00>
              Message ID: CMND_MSG_GENERAL_LINK_CFM<07>
              Unit ID: 00
              Cookie: 00
              Raw IE: 00 01 00
              Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<CMND_RC_OK>
```

CMND: Ok, I sent to BS that you are informed.

MCU: Let's download and upgrade to that new software.

```
MCU --> CMND:
Message Length: 59
Raw message: da da 00 37 00 00 02 0c 03 5a 32 00 2a 05 73 77 35 35 35 05 68 77 32 32 32 1d 68 74 74
    70 3a 2f 2f 31 37 32 2e 31 39 2e 33 35 2e 33 35 3a 38 30 2f 49 54 43 4d 5f 48 34 00 01 01
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_DOWNLOAD_START_REQ<03>
Unit ID: 00
Cookie: 00
Raw IE: 00 2a 05 73 77 35 35 35 05 68 77 32 32 32 1d 68 74 74 70 3a 2f 2f 31 37 32 2e 31 39 2e 33
    35 2e 33 35 3a 38 30 2f 49 54 43 4d 5f 48
Parsed IE: CMND_IE_NEW_SW_INFO<0x32>  Length<42>  Value<SwLen: 5; SwVer: <sw555>; HwLen: 5; HwVer:
    <hw222>; UrlLen: 29; URL: <http://172.19.35.35:80/ITCM_H>>
Raw IE: 00 01 01
Parsed IE: CMND_IE_IMAGE_TYPE<0x34>  Length<1>  Value<ImageType: DECT<01> >

              CMND --> MCU:
              Message Length: 10
              Raw message: da da 00 06 00 00 02 0c 04 18
              Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
              Message ID: CMND_MSG_SUOTA_DOWNLOAD_START_RES<04>
              Unit ID: 00
              Cookie: 00
```

CMND: Ok

```
              CMND --> MCU:
              Message Length: 26
              Raw message: da da 00 16 00 00 00 00 05 0e 0d 00 05 00 00 00 00 02 09 00 05 04 32 2e
                  39 34
              Service ID: CMND_SERVICE_ID_GENERAL<00 00>
              Message ID: CMND_MSG_GENERAL_HELLO_IND<05>
              Unit ID: 00
              Cookie: 00
              Raw IE: 00 05 00 00 00 00 02
              Parsed IE: CMND_IE_GENERAL_STATUS<0x0D>  Length<5>
                  Value<Mode:Norm,Reg:True,Eeprom:OK,ID:0x0002>
              Raw IE: 00 05 04 32 2e 39 34
              Parsed IE: CMND_IE_VERSION<0x09>  Length<5>  Value<Length: 4; Value: <2.94>>
```

CMND after some time: Hello!

```
              CMND --> MCU:
              Message Length: 21
              Raw message: da da 00 11 01 00 02 0c 05 4e 00 00 01 00 24 00 04 00 00 00 00
              Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
              Message ID: CMND_MSG_SUOTA_IMAGE_READY_IND<05>
              Unit ID: 00
```

CMND: The downloaded and installed image is Ok.

```
              Cookie: 01
              Raw IE: 00 01 00
              Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<CMND_RC_SUOTA_SUCCESS>
              Raw IE: 00 04 00 00 00 00
              Parsed IE: CMND_IE_U32<0x24>  Length<4>  Value<00000000>
```

MCU: So, what's your version now?

```
MCU --> CMND:
Message Length: 14
Raw message: da da 00 0a 00 00 00 00 0b 35 1e 00 01 01
Service ID: CMND_SERVICE_ID_GENERAL<00 00>
Message ID: CMND_MSG_GENERAL_GET_VERSION_REQ<0b>
Unit ID: 00
Cookie: 00
Raw IE: 00 01 01
Parsed IE: CMND_IE_U8<0x1E>  Length<1>  Value<CMND_GENERAL_VERSION_SW>
```

```
              CMND --> MCU:
              Message Length: 20
              Raw message: da da 00 10 00 00 00 00 0c a0 09 00 07 06 33 32 2e 32 36 73
              Service ID: CMND_SERVICE_ID_GENERAL<00 00>
              Message ID: CMND_MSG_GENERAL_GET_VERSION_RES<0c>
              Unit ID: 00
              Cookie: 00
              Raw IE: 00 07 06 33 32 2e 32 36 73
              Parsed IE: CMND_IE_VERSION<0x09>  Length<7>  Value<Length: 6; Value: <32.26s>>
```

CMND: My version is 32.26**s**

MCU: Add my version to yours – set it to 32.26s#1.0.1

```
MCU --> CMND:
Message Length: 30
Raw message: da da 00 19 00 00 02 03 03 ce 0b 00 10 00 07 00 0d 0c 33 32 2e 32 36 73 23 31 2e 30 2e
    31
Service ID: CMND_SERVICE_ID_PARAMETERS<02 03>
Message ID: CMND_MSG_PARAM_SET_REQ<03>
Unit ID: 00
Cookie: 00
Raw IE: 00 10 00 07 00 0d 0c 33 32 2e 32 36 73 23 31 2e 30 2e 31
Parsed IE: CMND_IE_PARAMETER<0x0B>  Length<16>  Value<EepParamType:CMND_PARAM_TYPE_EEPROM<0x00>,
    EepParamId:CMND_PARAM_EEPROM_SW_VERISON<0x07>, ParamLen:13, Value<Length: 12; Value:
    <32.26s#1.0.1>>>
```

```
              CMND --> MCU:
              Message Length: 21
              Raw message: da da 00 11 00 00 02 03 04 31 0b 00 04 00 07 00 00 00 00 01 00
              Service ID: CMND_SERVICE_ID_PARAMETERS<02 03>
              Message ID: CMND_MSG_PARAM_SET_RES<04>
              Unit ID: 00
              Cookie: 00
              Raw IE: 00 04 00 07 00 00
              Parsed IE: CMND_IE_PARAMETER<0x0B>  Length<4>
                  Value<EepParamType:CMND_PARAM_TYPE_EEPROM<0x00>,EepParamId:CMND_PARAM_EEPROM_SW_V
                  ERISON<0x07>,ParamLen:0>
              Raw IE: 00 01 00
              Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<CMND_RC_OK>
```

CMDN: The version was set successfully

MCU: Let's complete the SUOTA with current version 32.26s#1.0.1

```
MCU --> CMND:
```

```
Message Length: 27
Raw message: da da 00 18 00 00 02 0c 08 d8 35 00 10 00 0c 33 32 2e 32 36 73 23 31 2e 30 2e 31
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_UPGRADE_COMPLETED_REQ<08>
Unit ID: 00
Cookie: 00
Raw IE: 00 10 00 07 00 0c 33 32 2e 32 36 73 23 31 2e 30 2e 31
Parsed IE: CMND_IE_SW_VER_INFO<0x35>  Length<16>  Value<Length: 12; Value: <32.26s#1.0.1>>
```

```
            CMND --> MCU:
            Message Length: 14
            Raw message: da da 00 0a 00 00 00 00 07 12 00 00 01 00
            Service ID: CMND_SERVICE_ID_GENERAL<00 00>
            Message ID: CMND_MSG_GENERAL_LINK_CFM<07>
            Unit ID: 00
            Cookie: 00
            Raw IE: 00 01 00
            Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<CMND_RC_OK>
```

CMDN: The BS was informed about SUOTA completion

10.2 Appendix 2: Example log of Failed SUOTA for CMND node initiated by BS

MCU: Here is a Tamper Alert

```
MCU --> CMND:
Message Length: 14
Raw message: da da 00 0a 68 00 01 01 03 95 1d 00 01 00
Service ID: CMND_SERVICE_ID_TAMPER_ALERT<01 01>
Message ID: CMND_MSG_ALERT_NOTIFY_STATUS_REQ<03>
Unit ID: 00
Cookie: 68
Raw IE: 00 01 00
Parsed IE: CMND_IE_TAMPER_ALERT<0x1D>  Length<1>  Value<CMND_TAMPER_ALERT_STATUS_OFF<00>>
```

```
            CMND --> MCU:
            Message Length: 14
            Raw message: da da 00 0a 00 00 00 00 07 12 00 00 01 00
            Service ID: CMND_SERVICE_ID_GENERAL<00 00>
            Message ID: CMND_MSG_GENERAL_LINK_CFM<07>
            Unit ID: 00
            Cookie: 00
            Raw IE: 00 01 00
            Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<CMND_RC_OK>
```

CMND: Ok, I sent it to BS

```
            CMND --> MCU:
            Message Length: 59
            Raw message: da da 00 37 00 00 02 0c 01 1b 32 00 24 04 44 45 43 54 04 64 68 61 6e 19
                68 74 74 70 3a 2f 2f 31 39 32 2e 31 36 38 2e 31 2e 31 2f 49 54 43 4d 5f 48 03 00
                03 00 00 02 13 00 01 05
            Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
            Message ID: CMND_MSG_SUOTA_NEW_SW_AVAIALBE_IND<01>
            Unit ID: 00
            Cookie: 00
            Raw IE: 00 24 04 44 45 43 54 04 64 68 61 6e 19 68 74 74 70 3a 2f 2f 31 39 32 2e 31 36
                38 2e 31 2e 31 2f 49 54 43 4d 5f 48
```

CMND: BTW, new software for CMND is available!

```
              Parsed IE: CMND_IE_NEW_SW_INFO<0x32>  Length<36>  Value<SwLen: 4; SwVer: <DECT>;
                  HwLen: 4; HwVer: <dhan>; UrlLen: 25; URL: <http://192.168.1.1/ITCM_H>>
              Raw IE: 00 03 00 00 02
              Parsed IE: CMND_IE_UNIT_ADDR<0x03>  Length<3>  Value<000002>
              Raw IE: 00 01 05
              Parsed IE: CMND_IE_OTA_COOKIE<0x13>  Length<1>  Value<05>
```

MCU: Understood.

```
MCU --> CMND:
Message Length: 14
Raw message: da da 00 0a 00 00 02 0c 02 1b 00 00 01 00
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_NEW_SW_RES<02>
Unit ID: 00
Cookie: 00
Raw IE: 00 01 00
Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<CMND_RC_SUOTA_SUCCESS>
```

```
              CMND --> MCU:
              Message Length: 14
              Raw message: da da 00 0a 00 00 00 00 07 12 00 00 01 00
              Service ID: CMND_SERVICE_ID_GENERAL<00 00>
              Message ID: CMND_MSG_GENERAL_LINK_CFM<07>
              Unit ID: 00
              Cookie: 00
              Raw IE: 00 01 00
              Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<CMND_RC_OK>
```

CMND: Ok, I sent to BS that you are informed.

MCU: Let's download and upgrade to that new software.

```
MCU --> CMND:
Message Length: 59
Raw message: da da 00 37 00 00 02 0c 03 5a 32 00 2a 05 73 77 35 35 35 05 68 77 32 32 32 1d 68 74 74
    70 3a 2f 2f 31 37 32 2e 31 39 2e 33 35 2e 33 35 3a 38 30 2f 49 54 43 4d 5f 48 34 00 01 01
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_DOWNLOAD_START_REQ<03>
Unit ID: 00
Cookie: 00
Raw IE: 00 2a 05 73 77 35 35 35 05 68 77 32 32 32 1d 68 74 74 70 3a 2f 2f 31 37 32 2e 31 39 2e 33
    35 2e 33 35 3a 38 30 2f 49 54 43 4d 5f 48
Parsed IE: CMND_IE_NEW_SW_INFO<0x32>  Length<42>  Value<SwLen: 5; SwVer: <sw555>; HwLen: 5; HwVer:
    <hw222>; UrlLen: 29; URL: <http://172.19.35.35:80/ITCM_H>>
Raw IE: 00 01 01
Parsed IE: CMND_IE_IMAGE_TYPE<0x34>  Length<1>  Value<ImageType: DECT<01> >
```

```
              CMND --> MCU:
              Message Length: 10
              Raw message: da da 00 06 00 00 02 0c 04 18
              Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
              Message ID: CMND_MSG_SUOTA_DOWNLOAD_START_RES<04>
              Unit ID: 00
              Cookie: 00
```

CMND: Ok...

```
              CMND --> MCU:
              Message Length: 21
              Raw message: da da 00 11 d0 00 02 0c 05 1f 00 00 01 02 24 00 04 00 00 00 00
              Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
```

CMND after some time: Oops, session error has occurred.

```
              Message ID: CMND_MSG_SUOTA_IMAGE_READY_IND<05>
              Unit ID: 00
              Cookie: d0
              Raw IE: 00 01 02
              Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<CMND_RC_SUOTA_ERR_SESSION>
              Raw IE: 00 04 00 00 00 00
              Parsed IE: CMND_IE_U32<0x24>  Length<4>  Value<00000000>
```

MCU: Ok.

```
MCU --> CMND:
Message Length: 10
Raw message: da da 00 06 00 00 02 0c 06 1a
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_IMAGE_READY_RES<06>
Unit ID: 00
Cookie: 00
```

MCU: Let's complete the SUOTA with version 33.06#1.0.1 with failure

```
MCU --> CMND:
Message Length: 26
Raw message: da da 00 16 00 00 02 0c 08 62 35 00 0D 01 0B 33 33 2e 30 36 23 31 2e 30 2e 31
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_UPGRADE_COMPLETED_REQ<08>
Unit ID: 00
Cookie: 00
Raw IE: 00 0D 01 0B 33 33 2e 30 36 23 31 2e 30 2e 31
Parsed IE: CMND_IE_SW_VER_INFO<0x35>  Length<13>  Value<Result: CMND_RC_SUOTA_ERR, SwLen: 11,
    SwVer: <32.06#1.0.1> >
```

CMDN: The BS was informed about SUOTA completion

```
              CMND --> MCU:
              Message Length: 14
              Raw message: da da 00 0a 00 00 00 00 07 12 00 00 01 00
              Service ID: CMND_SERVICE_ID_GENERAL<00 00>
              Message ID: CMND_MSG_GENERAL_LINK_CFM<07>
              Unit ID: 00
              Cookie: 00
              Raw IE: 00 01 00
              Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<CMND_RC_OK>
```

10.3 Appendix 3: Example log of Successful SUOTA for Node Host initiated by BS

MCU: Here is an Alert

```
MCU --> CMND:
Message Length: 19
Raw message: da da 00 0f 68 01 01 00 03 8f 06 00 06 02 04 00 00 00 01
Service ID: CMND_SERVICE_ID_ALERT<01 00>
Message ID: CMND_MSG_ALERT_NOTIFY_STATUS_REQ<03>
Unit ID: 01
Cookie: 68
Raw IE: 00 06 02 04 00 00 00 01
Parsed IE: CMND_IE_ALERT<0x06>  Length<6>  Value<UnitType: FUN_UNIT_TYPE_SMOKE_DETECTOR;
    AlertStatus HAN_CMND_ALERT_STATE_ALERTING >

              CMND --> MCU:
```

```
              Message Length: 14
              Raw message: da da 00 0a 68 01 00 00 07 7b 00 00 01 00
              Service ID: CMND_SERVICE_ID_GENERAL<00 00>
              Message ID: CMND_MSG_GENERAL_LINK_CFM<07>
              Unit ID: 01
              Cookie: 68
              Raw IE: 00 01 00
              Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<CMND_RC_OK>
```

CMND: Ok, I sent it to BS

```
              CMND --> MCU:
              Message Length: 59
              Raw message: da da 00 37 00 00 02 0c 01 39 32 00 24 04 48 4f 53 54 04 64 68 61 6e 19
                  68 74 74 70 3a 2f 2f 31 39 32 2e 31 36 38 2e 31 2e 31 2f 49 54 43 4d 5f 48 03 00
                  03 00 00 02 13 00 01 05
              Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
              Message ID: CMND_MSG_SUOTA_NEW_SW_AVAIALBE_IND<01>
              Unit ID: 00
              Cookie: 00
              Raw IE: 00 24 04 48 4f 53 54 04 64 68 61 6e 19 68 74 74 70 3a 2f 2f 31 39 32 2e 31 36
                  38 2e 31 2e 31 2f 49 54 43 4d 5f 48
              Parsed IE: CMND_IE_NEW_SW_INFO<0x32>  Length<36>  Value<SwLen: 4; SwVer: <HOST>;
                  HwLen: 4; HwVer: <dhan>; UrlLen: 25; URL: <http://192.168.1.1/ITCM_H>>
              Raw IE: 00 03 00 00 02
              Parsed IE: CMND_IE_UNIT_ADDR<0x03>  Length<3>  Value<000002>
              Raw IE: 00 01 05
              Parsed IE: CMND_IE_OTA_COOKIE<0x13>  Length<1>  Value<05>
```

CMND: BTW, new software for MCU is available!

MCU: Understood.

```
MCU --> CMND:
Message Length: 14
Raw message: da da 00 0a 00 00 02 0c 02 1b 00 00 01 00
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_NEW_SW_RES<02>
Unit ID: 00
Cookie: 00
Raw IE: 00 01 00
Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<CMND_RC_SUOTA_SUCCESS>
```

```
              CMND --> MCU:
              Message Length: 14
              Raw message: da da 00 0a 00 00 00 00 07 12 00 00 01 00
              Service ID: CMND_SERVICE_ID_GENERAL<00 00>
              Message ID: CMND_MSG_GENERAL_LINK_CFM<07>
              Unit ID: 00
              Cookie: 00
              Raw IE: 00 01 00
              Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<CMND_RC_OK>
```

CMND: Ok, I sent to BS that you are informed.

MCU: Let's download and upgrade to that new software.

```
MCU --> CMND:
Message Length: 59
Raw message: da da 00 37 00 00 02 0c 03 5b 32 00 2a 05 73 77 35 35 35 05 68 77 32 32 32 1d 68 74 74
    70 3a 2f 2f 31 37 32 2e 31 39 2e 33 35 2e 33 35 3a 38 30 2f 49 54 43 4d 5f 48 34 00 01 02
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_DOWNLOAD_START_REQ<03>
Unit ID: 00
Cookie: 00
```

```
Raw IE: 00 2a 05 73 77 35 35 35 05 68 77 32 32 32 1d 68 74 74 70 3a 2f 2f 31 37 32 2e 31 39 2e 33
    35 2e 33 35 3a 38 30 2f 49 54 43 4d 5f 48
Parsed IE: CMND_IE_NEW_SW_INFO<0x32>  Length<42>  Value<SwLen: 5; SwVer: <sw555>; HwLen: 5; HwVer:
    <hw222>; UrlLen: 29; URL: <http://172.19.35.35:80/ITCM_H>>
Raw IE: 00 01 02
Parsed IE: CMND_IE_IMAGE_TYPE<0x34>  Length<1>  Value<ImageType: MCU<02> >
```

```
              CMND --> MCU:
              Message Length: 10
              Raw message: da da 00 06 00 00 02 0c 04 18
              Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
              Message ID: CMND_MSG_SUOTA_DOWNLOAD_START_RES<04>
              Unit ID: 00
              Cookie: 00
```

CMND: Ok.

```
              CMND --> MCU:
              Message Length: 21
              Raw message: da da 00 11 84 00 02 0c 05 3c 00 00 01 00 24 00 04 00 00 4e 1d
              Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
              Message ID: CMND_MSG_SUOTA_IMAGE_READY_IND<05>
              Unit ID: 00
              Cookie: 84
              Raw IE: 00 01 00
              Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<CMND_RC_SUOTA_SUCCESS>
              Raw IE: 00 04 00 00 4e 1d
              Parsed IE: CMND_IE_U32<0x24>  Length<4>  Value<00004e1d>
```

CMND after some time: The image is ready, length = 19997 bytes

MCU: Ok.

```
MCU --> CMND:
Message Length: 10
Raw message: da da 00 06 00 00 02 0c 06 1a
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_IMAGE_READY_RES<06>
Unit ID: 00
Cookie: 00
```

MCU: Read the image at offset 0, size 100

```
MCU --> CMND:
Message Length: 19
Raw message: da da 00 0f 00 00 02 0c 0a c8 37 00 06 00 00 00 00 00 64
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_READ_FILE_REQ<0a>
Unit ID: 00
Cookie: 00
Raw IE: 00 06 00 00 00 00 00 64
Parsed IE: CMND_IE_READ_FILE_DATA_REQ<0x37>  Length<6>  Value<Offset: 0, Length: 100>
```

```
              CMND --> MCU:
              Message Length: 123
              Raw message: da da 00 77 01 00 02 0c 0b 0f 00 00 01 00 36 00 6a 00 00 00 00 00 64 44
                  53 50 20 47 72 6f 75 70 2d 4c 6f 67 66 69 6c 65 0d 0a 4c 6f 67 20 53 74 61 72 74
                  65 64 20 61 74 3a 20 30 38 2d 31 31 2d 32 30 31 37 20 31 34 3a 30 33 3a 35 35 0d
                  0a 56 65 72 73 69 6f 6e 20 31 2e 31 2e 4b 20 2d 20 42 75 69 6c 64 20 33 30 0d 0a
                  0d 0a 30 38 2d 31 31 2d 32 30 31 37 20 31 34 3a 32 30
              Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
              Message ID: CMND_MSG_SUOTA_READ_FILE_RES<0b>
              Unit ID: 00
```

CMND: Here is the image fragment you requested

```
        Cookie: 01
        Raw IE: 00 01 00
        Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<GENERAL_SUCCESS>
        Raw IE: 00 6a 00 00 00 00 00 64 44 53 50 20 47 72 6f 75 70 2d 4c 6f 67 66 69 6c 65 0d
            0a 4c 6f 67 20 53 74 61 72 74 65 64 20 61 74 3a 20 30 38 2d 31 31 2d 32 30 31 37
            20 31 34 3a 30 33 3a 35 35 0d 0a 56 65 72 73 69 6f 6e 20 31 2e 31 2e 4b 20 2d 20
            42 75 69 6c 64 20 33 30 0d 0a 0d 0a 30 38 2d 31 31 2d 32 30 31 37 20 31 34 3a 32
            30
        Parsed IE: CMND_IE_READ_FILE_DATA_RES<0x36>  Length<106>  Value<Offset: 0,
            Length:100,
            Data:4453502047726f75702d4c6f6766696c650d0a4c6f6720537461727465642061743a2030382d
            31312d323031372031343a30333a35350d0a56657273696f6e20312e312e4b202d204275696c64203
            3300d0a0d0a                                   30>
```

MCU: Read the image at offset 100, length 100

```
MCU --> CMND:
Message Length: 19
Raw message: da da 00 0f 01 00 02 0c 0a 2d 37 00 06 00 00 00 64 00 64
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_READ_FILE_REQ<0a>
Unit ID: 00
Cookie: 01
Raw IE: 00 06 00 00 00 64 00 64
Parsed IE: CMND_IE_READ_FILE_DATA_REQ<0x37>  Length<6>  Value<Offset:100, Length:100>

        CMND --> MCU:
        Message Length: 123
        Raw message: da da 00 77 01 00 02 0c 0b cd 00 00 01 00 36 00 6a 00 00 00 64 00 64 3a
            33 32 0d 0a 43 4d 4e 44 20 2d 2d 3e 20 4d 43 55 3a 0d 0a 4d 65 73 73 61 67 65 20
            4c 65 6e 67 74 68 3a 20 31 34 0d 0a 52 61 77 20 6d 65 73 73 61 67 65 3a 20 64 61
            20 64 61 20 30 30 20 30 61 20 30 30 20 30 30 20 30 30 20 30 30 20 30 37 20 31 32
            20 30 30 20 30 30 20 30 31 20 30 30 20 0d 0a 53 65 72
        Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
        Message ID: CMND_MSG_SUOTA_READ_FILE_RES<0b>
        Unit ID: 00
        Cookie: 01
        Raw IE: 00 01 00
        Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<GENERAL_SUCCESS>
        Raw IE: 00 6a 00 00 00 64 00 64 3a 33 32 0d 0a 43 4d 4e 44 20 2d 2d 3e 20 4d 43 55 3a
            0d 0a 4d 65 73 73 61 67 65 20 4c 65 6e 67 74 68 3a 20 31 34 0d 0a 52 61 77 20 6d
            65 73 73 61 67 65 3a 20 64 61 20 64 61 20 30 30 20 30 61 20 30 30 20 30 30 20 30
            30 20 30 30 20 30 37 20 31 32 20 30 30 20 30 30 20 30 31 20 30 30 20 0d 0a 53 65
            72
        Parsed IE: CMND_IE_READ_FILE_DATA_RES<0x36>  Length<106>  Value<Offset: 100, Length:
            100, Data:
            3a33320d0a434d4e44202d2d3e204d43553a0d0a4d657373616765204c656e6774683a2031340d0a5
            26177206d6573736167653a2064612064612030302030612030302030302030302030302030372031
            32203030203030203031203030200d0a536572>
```

CMND: Here is the image fragment you requested

<Skipped many similar pairs of messages CMND_MSG_SUOTA_READ_FILE_REQ and CMND_MSG_SUOTA_READ_FILE_RES>

MCU: Read the image at offset 19984, size 100

```
MCU --> CMND:
Message Length: 19
Raw message: da da 00 0f c8 00 02 0c 0a ee 37 00 06 00 00 4e 10 00 64
```

```
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_READ_FILE_REQ<0a>
Unit ID: 00
Cookie: c8
Raw IE: 00 06 00 00 4e 10 00 64
Parsed IE: CMND_IE_READ_FILE_DATA_REQ<0x37>  Length<6>  Value<Offset: 19984, Length: 100>

              CMND --> MCU:
              Message Length: 36
              Raw message: da da 00 20 00 00 02 0c 0b 45 00 00 01 00 36 00 13 00 00 4e 10 00 0d 61
                  6c 3c 73 75 63 63 65 73 73 3e 0d 0a
              Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
              Message ID: CMND_MSG_SUOTA_READ_FILE_RES<0b>
              Unit ID: 00
              Cookie: 00
              Raw IE: 00 01 00
              Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<GENERAL_SUCCESS>
              Raw IE: 00 13 00 00 4e 10 00 0d 61 6c 3c 73 75 63 63 65 73 73 3e 0d 0a
              Parsed IE: CMND_IE_READ_FILE_DATA_RES<0x36>  Length<19>  Value<Offset: 19984, Length:
                  13, Data: 616c3c737563636573733e0d0a>
```

CMND: Here is the block you requested, but the size is 13

MCU: Read the image at offset 19997, size 100

```
MCU --> CMND:
Message Length: 19
Raw message: da da 00 0f c9 00 02 0c 0a fc 37 00 06 00 00 4e 1d 00 64
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_READ_FILE_REQ<0a>
Unit ID: 00
Cookie: c9
Raw IE: 00 06 00 00 4e 1d 00 64
Parsed IE: CMND_IE_READ_FILE_DATA_REQ<0x37>  Length<6>  Value<Offset: 19997, Length: 100>

              CMND --> MCU:
              Message Length: 23
              Raw message: da da 00 13 00 00 02 0c 0b d4 00 00 01 00 36 00 06 00 00 4e 1d 00 00
              Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
              Message ID: CMND_MSG_SUOTA_READ_FILE_RES<0b>
              Unit ID: 00
              Cookie: 00
              Raw IE: 00 01 00
              Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<GENERAL_SUCCESS>
              Raw IE: 00 06 00 00 4e 1d 00 00
              Parsed IE: CMND_IE_READ_FILE_DATA_RES<0x36>  Length<6>  Value<Offset: 19997, Length:
                  0>
```

CMND: The requested block is empty

MCU: What's your current version?

```
MCU --> CMND:
Message Length: 14
Raw message: da da 00 0a 00 00 00 00 0b 35 1e 00 01 01
Service ID: CMND_SERVICE_ID_GENERAL<00 00>
Message ID: CMND_MSG_GENERAL_GET_VERSION_REQ<0b>
Unit ID: 00
Cookie: 00
Raw IE: 00 01 01
Parsed IE: CMND_IE_U8<0x1E>  Length<1>  Value<CMND_GENERAL_VERSION_SW>

              CMND --> MCU:
```

```
              Message Length: 20
              Raw message: da da 00 10 00 00 00 00 0c a0 09 00 07 06 33 32 2e 32 36 73
              Service ID: CMND_SERVICE_ID_GENERAL<00 00>
              Message ID: CMND_MSG_GENERAL_GET_VERSION_RES<0c>
              Unit ID: 00
              Cookie: 00
              Raw IE: 00 07 06 33 32 2e 32 36 73
              Parsed IE: CMND_IE_VERSION<0x09>  Length<7>  Value<Length: 6; Value: <32.26s>>
```

CMND: My current version is 32.26s

MCU: Set your current version to 32.26s#1.0.1

```
MCU --> CMND:
Message Length: 30
Raw message: da da 00 1a 00 00 02 03 03 b4 0b 00 0b 00 07 00 07 06 33 32 2e 32 36 73 23 31 2e 30 2e
    31
Service ID: CMND_SERVICE_ID_PARAMETERS<02 03>
Message ID: CMND_MSG_PARAM_SET_REQ<03>
Unit ID: 00
Cookie: 00
Raw IE: 00 0b 00 07 00 0d 0c 33 32 2e 32 36 73 23 31 2e 30 2e 31
Parsed IE: CMND_IE_PARAMETER<0x0B>  Length<17>
    Value<EepParamType:CMND_PARAM_TYPE_EEPROM<0x00>,EepParamId:CMND_PARAM_EEPROM_SW_VERISON<0x07>,P
    aramLen:13,Value<Length: 12; Value: <32.26.s#1.0.1>>>
```

```
              CMND --> MCU:
              Message Length: 21
              Raw message: da da 00 11 00 00 02 03 04 31 0b 00 04 00 07 00 00 00 00 01 00
              Service ID: CMND_SERVICE_ID_PARAMETERS<02 03>
              Message ID: CMND_MSG_PARAM_SET_RES<04>
              Unit ID: 00
              Cookie: 00
              Raw IE: 00 04 00 07 00 00
              Parsed IE: CMND_IE_PARAMETER<0x0B>  Length<4>
                  Value<EepParamType:CMND_PARAM_TYPE_EEPROM<0x00>,EepParamId:CMND_PARAM_EEPROM_SW_V
                  ERISON<0x07>,ParamLen:0>
              Raw IE: 00 01 00
              Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<CMND_RC_OK>
```

CMND: The parameter was set successfully

MCU: Let's complete the SUOTA with current version 32.26s#1.0.1

```
MCU --> CMND:
Message Length: 27
Raw message: da da 00 18 00 00 02 0c 08 d8 35 00 10 00 0c 33 32 2e 32 36 73 23 31 2e 30 2e 31
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_UPGRADE_COMPLETED_REQ<08>
Unit ID: 00
Cookie: 00
Raw IE: 00 10 00 07 00 0c 33 32 2e 32 36 73 23 31 2e 30 2e 31
Parsed IE: CMND_IE_SW_VER_INFO<0x35>  Length<16>  Value<Length: 12; Value: <32.26s#1.0.1>>
```

```
              CMND --> MCU:
              Message Length: 14
              Raw message: da da 00 0a 00 00 00 00 07 12 00 00 01 00
              Service ID: CMND_SERVICE_ID_GENERAL<00 00>
              Message ID: CMND_MSG_GENERAL_LINK_CFM<07>
              Unit ID: 00
              Cookie: 00
```

CMDN: The BS was informed about SUOTA completion

```
              Raw IE: 00 01 00
              Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<CMND_RC_OK>
```

10.4 Appendix 4: Example log of Failed SUOTA for Node Host initiated by BS

MCU: Here is a Tamper Alert

```
MCU --> CMND:
Message Length: 14
Raw message: da da 00 0a 68 00 01 01 03 95 1d 00 01 00
Service ID: CMND_SERVICE_ID_TAMPER_ALERT<01 01>
Message ID: CMND_MSG_ALERT_NOTIFY_STATUS_REQ<03>
Unit ID: 00
Cookie: 68
Raw IE: 00 01 00
Parsed IE: CMND_IE_TAMPER_ALERT<0x1D>  Length<1>  Value<CMND_TAMPER_ALERT_STATUS_OFF<00>>
```

```
              CMND --> MCU:
              Message Length: 14
              Raw message: da da 00 0a 00 00 00 00 07 12 00 00 01 00
              Service ID: CMND_SERVICE_ID_GENERAL<00 00>
              Message ID: CMND_MSG_GENERAL_LINK_CFM<07>
              Unit ID: 00
              Cookie: 00
              Raw IE: 00 01 00
              Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<CMND_RC_OK>
```

CMND: Ok, I sent it to BS

```
              CMND --> MCU:
              Message Length: 59
              Raw message: da da 00 37 00 00 02 0c 01 39 32 00 24 04 48 4f 53 54 04 64 68 61 6e 19
                  68 74 74 70 3a 2f 2f 31 39 32 2e 31 36 38 2e 31 2e 31 2f 49 54 43 4d 5f 48 03 00
                  03 00 00 02 13 00 01 05
              Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
              Message ID: CMND_MSG_SUOTA_NEW_SW_AVAIALBE_IND<01>
              Unit ID: 00
              Cookie: 00
              Raw IE: 00 24 04 48 4f 53 54 04 64 68 61 6e 19 68 74 74 70 3a 2f 2f 31 39 32 2e 31 36
                  38 2e 31 2e 31 2f 49 54 43 4d 5f 48
              Parsed IE: CMND_IE_NEW_SW_INFO<0x32>  Length<36>  Value<SwLen: 4; SwVer: <HOST>;
                  HwLen: 4; HwVer: <dhan>; UrlLen: 25; URL: <http://192.168.1.1/ITCM_H>>
              Raw IE: 00 03 00 00 02
              Parsed IE: CMND_IE_UNIT_ADDR<0x03>  Length<3>  Value<000002>
              Raw IE: 00 01 05
              Parsed IE: CMND_IE_OTA_COOKIE<0x13>  Length<1>  Value<05>
```

CMND: BTW, new software for MCU is available!

MCU: Understood.

```
MCU --> CMND:
Message Length: 14
Raw message: da da 00 0a 00 00 02 0c 02 1b 00 00 01 00
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_NEW_SW_RES<02>
Unit ID: 00
Cookie: 00
Raw IE: 00 01 00
```

```
Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<CMND_RC_SUOTA_SUCCESS>

              CMND --> MCU:
              Message Length: 14
              Raw message: da da 00 0a 00 00 00 00 07 12 00 00 01 00
              Service ID: CMND_SERVICE_ID_GENERAL<00 00>
              Message ID: CMND_MSG_GENERAL_LINK_CFM<07>
              Unit ID: 00
              Cookie: 00
              Raw IE: 00 01 00
              Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<CMND_RC_OK>
```

CMND: Ok, I sent to BS that you are informed.

MCU: Let's download and upgrade to that new software.

```
MCU --> CMND:
Message Length: 59
Raw message: da da 00 37 00 00 02 0c 03 5b 32 00 2a 05 73 77 35 35 35 05 68 77 32 32 32 1d 68 74 74
    70 3a 2f 2f 31 37 32 2e 31 39 2e 33 35 2e 33 35 3a 38 30 2f 49 54 43 4d 5f 48 34 00 01 02
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_DOWNLOAD_START_REQ<03>
Unit ID: 00
Cookie: 00
Raw IE: 00 2a 05 73 77 35 35 35 05 68 77 32 32 32 1d 68 74 74 70 3a 2f 2f 31 37 32 2e 31 39 2e 33
    35 2e 33 35 3a 38 30 2f 49 54 43 4d 5f 48
Parsed IE: CMND_IE_NEW_SW_INFO<0x32>  Length<42>  Value<SwLen: 5; SwVer: <sw555>; HwLen: 5; HwVer:
    <hw222>; UrlLen: 29; URL: <http://172.19.35.35:80/ITCM_H>>
Raw IE: 00 01 02
Parsed IE: CMND_IE_IMAGE_TYPE<0x34>  Length<1>  Value<ImageType: MCU<02> >


              CMND --> MCU:
              Message Length: 10
              Raw message: da da 00 06 00 00 02 0c 04 18
              Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
              Message ID: CMND_MSG_SUOTA_DOWNLOAD_START_RES<04>
              Unit ID: 00
              Cookie: 00
```

CMND: Ok...

```
              CMND --> MCU:
              Message Length: 21
              Raw message: da da 00 11 00 00 02 0c 05 4f 00 00 01 02 24 00 04 00 00 00 00
              Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
              Message ID: CMND_MSG_SUOTA_IMAGE_READY_IND<05>
              Unit ID: 00
              Cookie: 00
              Raw IE: 00 01 02
              Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<CMND_RC_SUOTA_ERR_SESSION>
              Raw IE: 00 04 00 00 00 00
              Parsed IE: CMND_IE_U32<0x24>  Length<4>  Value<00000000>
```

CMND after some time: Oops, session error was occurred.

CMND: Ok.

```
MCU --> CMND:
Message Length: 10
Raw message: da da 00 06 00 00 02 0c 06 1a
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_IMAGE_READY_RES<06>
Unit ID: 00
Cookie: 00
```

MCU: Let's complete the SUOTA with current version 33.06#1.0.1

```
MCU --> CMND:
Message Length: 26
Raw message: da da 00 16 00 00 02 0c 08 61 35 00 06 00 0b 33 33 2e 30 36 23 31 2e 30 2e 31
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_UPGRADE_COMPLETED_REQ<08>
Unit ID: 00
Cookie: 00
Raw IE: 00 06 00 0b 33 33 2e 30 36 23 31 2e 30 2e 31
Parsed IE: CMND_IE_SW_VER_INFO<0x35>  Length<13>  Value<SwLen: 11, SwVer: 33.06#1.0.1>

              CMND --> MCU:
              Message Length: 14
              Raw message: da da 00 0a 00 00 00 00 07 24 00 00 01 12
              Service ID: CMND_SERVICE_ID_GENERAL<00 00>
              Message ID: CMND_MSG_GENERAL_LINK_CFM<07>
              Unit ID: 00
              Cookie: 00
              Raw IE: 00 01 12
              Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<CMND_RC_FAIL_SEND_LOCK>
```

CMND: Failed to inform BS about SUOTA completion

10.5 Appendix 5: Example log of Successful SUOTA for CMND node initiated by Node Host

MCU: Is there any new software for Node Host available?

```
MCU --> CMND:
Message Length: 10
Raw message: da da 00 06 00 00 02 0c 0e 22
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_CHECK_FOR_NEW_VERSION_REQ<0e>
Unit ID: 00
Cookie: 00

              CMND --> MCU:
              Message Length: 14
              Raw message: da da 00 0a 00 00 00 00 07 12 00 00 01 00
              Service ID: CMND_SERVICE_ID_GENERAL<00 00>
              Message ID: CMND_MSG_GENERAL_LINK_CFM<07>
              Unit ID: 00
              Cookie: 00
              Raw IE: 00 01 00
              Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<CMND_RC_OK>
```

CMND: I sent this request to BS

```
              CMND --> MCU:
              Message Length: 62
              Raw message: da da 00 3a 00 00 02 0c 0f 8e 42 00 27 00 06 33 33 2e 30 36 73 04 64 68
                  61 6e 19 68 74 74 70 3a 2f 2f 31 39 32 2e 31 36 38 2e 31 2e 31 2f 49 54 43 4d 5f
                  48 03 00 03 00 00 02 13 00 01 05
              Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
              Message ID: CMND_MSG_SUOTA_CHECK_FOR_NEW_VERSION_RES<0f>
              Unit ID: 00
              Cookie: 00
              Raw IE: 00 27 00 06 33 33 2e 30 36 73 04 64 68 61 6e 19 68 74 74 70 3a 2f 2f 31 39 32
                  2e 31 36 38 2e 31 2e 31 2f 49 54 43 4d 5f 48
```

```
          Parsed IE: CMND_IE_CHECK_SW_VER_RES<0x42>  Length<39>  Value<Result: 0; SwLen: 6;
              SwVer: <33.06s>; HwLen: 4; HwVer: <dhan>; UrlLen: 25; URL:
              <http://192.168.1.1/ITCM_H>>
          Raw IE: 00 03 00 00 02
          Parsed IE: CMND_IE_UNIT_ADDR<0x03>  Length<3>  Value<000002>
          Raw IE: 00 01 05
          Parsed IE: CMND_IE_OTA_COOKIE<0x13>  Length<1>  Value<05>
```

CMND: Yes, there is a new version of CMND FW 33.06s available

MCU: Let's download and upgrade to that new software.

```
MCU --> CMND:
Message Length: 55
Raw message: da da 00 33 00 00 02 0c 03 7f 32 00 26 06 33 33 2e 30 36 73 04 64 68 61 6e 19 68 74 74
    70 3a 2f 2f 31 39 32 2e 31 36 38 2e 31 2e 31 2f 49 54 43 4d 5f 48 34 00 01 01
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_DOWNLOAD_START_REQ<03>
Unit ID: 00
Cookie: 00
Raw IE: 00 26 06 33 33 2e 30 36 73 04 64 68 61 6e 19 68 74 74 70 3a 2f 2f 31 39 32 2e 31 36 38 2e
    31 2e 31 2f 49 54 43 4d 5f 48
Parsed IE: CMND_IE_NEW_SW_INFO<0x32>  Length<38>  Value<SwLen: 6; SwVer: <33.06s>; HwLen: 4; HwVer:
    <dhan>; UrlLen: 25; URL: <http://192.168.1.1/ITCM_H>>
Raw IE: 00 01 01
Parsed IE: CMND_IE_IMAGE_TYPE<0x34>  Length<1>  Value<ImageType: DECT<01> >

          CMND --> MCU:
          Message Length: 10
          Raw message: da da 00 06 00 00 02 0c 04 18
          Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
          Message ID: CMND_MSG_SUOTA_DOWNLOAD_START_RES<04>
          Unit ID: 00
          Cookie: 00
```

CMND: Ok...

```
          CMND --> MCU:
          Message Length: 21
          Raw message: da da 00 11 0e 00 02 0c 05 5b 00 00 01 00 24 00 04 00 00 00 00
          Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
          Message ID: CMND_MSG_SUOTA_IMAGE_READY_IND<05>
          Unit ID: 00
          Cookie: 0e
          Raw IE: 00 01 00
          Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<CMND_RC_SUOTA_SUCCESS>
          Raw IE: 00 04 00 00 00 00
          Parsed IE: CMND_IE_U32<0x24>  Length<4>  Value<00000000>
```

CMND: The image was downloaded successfully

MCU: What's your current version?

```
MCU --> CMND:
Message Length: 14
Raw message: da da 00 0a 00 00 00 00 0b 35 1e 00 01 01
Service ID: CMND_SERVICE_ID_GENERAL<00 00>
Message ID: CMND_MSG_GENERAL_GET_VERSION_REQ<0b>
Unit ID: 00
Cookie: 00
Raw IE: 00 01 01
Parsed IE: CMND_IE_U8<0x1E>  Length<1>  Value<CMND_GENERAL_VERSION_SW>
```

```
              CMND --> MCU:
              Message Length: 20
              Raw message: da da 00 10 00 00 00 00 0c 9f 09 00 07 06 33 33 2e 30 36 73
              Service ID: CMND_SERVICE_ID_GENERAL<00 00>
              Message ID: CMND_MSG_GENERAL_GET_VERSION_RES<0c>
              Unit ID: 00
              Cookie: 00
              Raw IE: 00 07 06 33 33 2e 30 36 73
              Parsed IE: CMND_IE_VERSION<0x09>  Length<7>  Value<Length: 6; Value: <33.06s>>
```

CMND: My current version is 33.06s

MCU: Add my version to yours - set it to 33.06s#1.0.1

```
MCU --> CMND:
Message Length: 30
Raw message: da da 00 1a 00 00 02 03 03 ce 0b 00 10 00 07 00 0d 0c 33 33 2e 30 36 73 23 31 2e 30 2e
    31
Service ID: CMND_SERVICE_ID_PARAMETERS<02 03>
Message ID: CMND_MSG_PARAM_SET_REQ<03>
Unit ID: 00
Cookie: 00
Raw IE: 00 10 00 07 00 0d 0c 33 33 2e 30 36 73 23 31 2e 30 2e 31
Parsed IE: CMND_IE_PARAMETER<0x0B>  Length<16>  Value<EepParamType:CMND_PARAM_TYPE_EEPROM<0x00>,
    EepParamId:CMND_PARAM_EEPROM_SW_VERISON<0x07>, ParamLen:13, Value<Length: 12; Value:
    <33.06s#1.0.1>>>
```

```
              CMND --> MCU:
              Message Length: 21
              Raw message: da da 00 11 00 00 02 03 04 31 0b 00 04 00 07 00 00 00 00 01 00
              Service ID: CMND_SERVICE_ID_PARAMETERS<02 03>
              Message ID: CMND_MSG_PARAM_SET_RES<04>
              Unit ID: 00
              Cookie: 00
              Raw IE: 00 04 00 07 00 00
              Parsed IE: CMND_IE_PARAMETER<0x0B>  Length<4>
                  Value<EepParamType:CMND_PARAM_TYPE_EEPROM<0x00>,EepParamId:CMND_PARAM_EEPROM_SW_V
                  ERISON<0x07>,ParamLen:0>
              Raw IE: 00 01 00
              Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<CMND_RC_OK>
```

CMND: The parameter was set successfully

MCU: Let's complete the SUOTA with current version 33.06s#1.0.1

```
MCU --> CMND:
Message Length: 27
Raw message: da da 00 17 00 00 02 0c 08 d8 35 00 0e 00 0c 33 33 2e 30 36 73 23 31 2e 30 2e 31
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_UPGRADE_COMPLETED_REQ<08>
Unit ID: 00
Cookie: 00
Raw IE: 00 0e 00 0c 33 33 2e 30 36 73 23 31 2e 30 2e 31
Parsed IE: CMND_IE_SW_VER_INFO<0x35>  Length<14>  Value<Length: 12; Value: <33.06s#1.0.1>>
```

```
              CMND --> MCU:
              Message Length: 14
              Raw message: da da 00 0a 00 00 00 00 07 12 00 00 01 00
              Service ID: CMND_SERVICE_ID_GENERAL<00 00>
              Message ID: CMND_MSG_GENERAL_LINK_CFM<07>
              Unit ID: 00
              Cookie: 00
```

CMDN: The BS was informed about SUOTA completion

```
              Raw IE: 00 01 00
              Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<CMND_RC_OK>
```

10.6 Appendix 6: Example log of Failed SUOTA for CMND node initiated by Node Host

MCU: Is there any new software for Node Host available?

```
MCU --> CMND:
Message Length: 10
Raw message: da da 00 06 00 00 02 0c 0e 22
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_CHECK_FOR_NEW_VERSION_REQ<0e>
Unit ID: 00
Cookie: 00

              CMND --> MCU:
              Message Length: 14
              Raw message: da da 00 0a 00 00 00 00 07 12 00 00 01 00
              Service ID: CMND_SERVICE_ID_GENERAL<00 00>
              Message ID: CMND_MSG_GENERAL_LINK_CFM<07>
              Unit ID: 00
              Cookie: 00
              Raw IE: 00 01 00
              Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<CMND_RC_OK>
```

CMND: I sent this request to BS

```
              CMND --> MCU:
              Message Length: 62
              Raw message: da da 00 3a 00 00 02 0c 0f 8e 42 00 27 00 06 33 33 2e 30 36 73 04 64 68
                  61 6e 19 68 74 74 70 3a 2f 2f 31 39 32 2e 31 36 38 2e 31 2e 31 2f 49 54 43 4d 5f
                  48 03 00 03 00 00 02 13 00 01 05
              Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
              Message ID: CMND_MSG_SUOTA_CHECK_FOR_NEW_VERSION_RES<0f>
              Unit ID: 00
              Cookie: 00
              Raw IE: 00 27 00 06 33 33 2e 30 36 73 04 64 68 61 6e 19 68 74 74 70 3a 2f 2f 31 39 32
                  2e 31 36 38 2e 31 2e 31 2f 49 54 43 4d 5f 48
              Parsed IE: CMND_IE_CHECK_SW_VER_RES<0x42>  Length<39>  Value<Result: 0; SwLen: 6;
                  SwVer: <33.06s>; HwLen: 4; HwVer: <dhan>; UrlLen: 25; URL:
                  <http://192.168.1.1/ITCM_H>>
              Raw IE: 00 03 00 00 02
              Parsed IE: CMND_IE_UNIT_ADDR<0x03>  Length<3>  Value<000002>
              Raw IE: 00 01 05
              Parsed IE: CMND_IE_OTA_COOKIE<0x13>  Length<1>  Value<05>
```

CMND: Yes, there is a new version of CMND FW 33.06s available

MCU: Let's download and upgrade to that new software.

```
MCU --> CMND:
Message Length: 55
Raw message: da da 00 33 00 00 02 0c 03 7f 32 00 26 06 33 33 2e 30 36 73 04 64 68 61 6e 19 68 74 74
    70 3a 2f 2f 31 39 32 2e 31 36 38 2e 31 2e 31 2f 49 54 43 4d 5f 48 34 00 01 01
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_DOWNLOAD_START_REQ<03>
```

```
Unit ID: 00
Cookie: 00
Raw IE: 00 26 06 33 33 2e 30 36 73 04 64 68 61 6e 19 68 74 74 70 3a 2f 2f 31 39 32 2e 31 36 38 2e
    31 2e 31 2f 49 54 43 4d 5f 48
Parsed IE: CMND_IE_NEW_SW_INFO<0x32>  Length<38>  Value<SwLen: 6; SwVer: <33.06s>; HwLen: 4; HwVer:
    <dhan>; UrlLen: 25; URL: <http://192.168.1.1/ITCM_H>>
Raw IE: 00 01 01
Parsed IE: CMND_IE_IMAGE_TYPE<0x34>  Length<1>  Value<ImageType: DECT<01> >

              CMND --> MCU:
              Message Length: 10
              Raw message: da da 00 06 00 00 02 0c 04 18
              Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
              Message ID: CMND_MSG_SUOTA_DOWNLOAD_START_RES<04>
              Unit ID: 00
              Cookie: 00
```

CMND: Ok...

```
              CMND --> MCU:
              Message Length: 21
              Raw message: da da 00 11 d0 00 02 0c 05 1f 00 00 01 02 24 00 04 00 00 00 00
              Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
              Message ID: CMND_MSG_SUOTA_IMAGE_READY_IND<05>
              Unit ID: 00
              Cookie: d0
              Raw IE: 00 01 02
              Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<CMND_RC_SUOTA_ERR_SESSION>
              Raw IE: 00 04 00 00 00 00
              Parsed IE: CMND_IE_U32<0x24>  Length<4>  Value<00000000>
```

CMND after some time: Oops, session error was occurred.

CMND: Understood.

```
MCU --> CMND:
Message Length: 10
Raw message: da da 00 06 00 00 02 0c 06 1a
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_IMAGE_READY_RES<06>
Unit ID: 00
Cookie: 00
```

MCU: Let's complete the SUOTA with version 33.06s#1.0.1

```
MCU --> CMND:
Message Length: 27
Raw message: da da 00 18 00 00 02 0c 08 d8 35 00 10 00 0c 33 33 2e 30 36 73 23 31 2e 30 2e 31
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_UPGRADE_COMPLETED_REQ<08>
Unit ID: 00
Cookie: 00
Raw IE: 00 10 00 0c 33 33 2e 30 36 73 23 31 2e 30 2e 31
Parsed IE: CMND_IE_SW_VER_INFO<0x35>  Length<16>  Value<Length: 12; Value: <33.06s#1.0.1>>

              CMND --> MCU:
              Message Length: 14
              Raw message: da da 00 0a 00 00 00 00 07 12 00 00 01 00
              Service ID: CMND_SERVICE_ID_GENERAL<00 00>
              Message ID: CMND_MSG_GENERAL_LINK_CFM<07>
              Unit ID: 00
              Cookie: 00
              Raw IE: 00 01 00
              Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<CMND_RC_OK>
```

CMDN: The BS was informed about SUOTA completion

10.7 Appendix 7: Example log of Successful SUOTA for Node Host initiated by itself

MCU: Is there any new software for Node Host available?

```
MCU --> CMND:
Message Length: 23
Raw message: da da 00 13 00 00                          3 54 04 64 68 61 6e
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_CHECK_FOR_NEW_VERSION_REQ<0e>
Unit ID: 00
Cookie: 00
Raw IE: 00 0a 04 48 4f 53 54 04 64 68 61 6e
Parsed IE: CMND_IE_CURRENT_SW_INFO<0x33>  Length<10>  Value<SwLen: 4; SwVer: <HOST>; HwLen: 4;
    HwVer: <dhan>>
```

```
CMND --> MCU:
Message Length: 14
Raw message: da da 00 0a 00 00 00 00 07 12 00 00 01 00
Service ID: CMND_SERVICE_ID_GENERAL<00 00>
Message ID: CMND_MSG_GENERAL_LINK_CFM<07>
Unit ID: 00
Cookie: 00
Raw IE: 00 01 00
Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<CMND_RC_OK>
```

CMND: I sent this request to BS

```
CMND --> MCU:
Message Length: 60
Raw message: da da 00 38 00 00 02 0c 0f 59 42 00 25 00 04 48 4f 53 54 04 64 68 61 6e
    19 68 74 74 70 3a 2f 2f 31 39 32 2e 31 36 38 2e 31 2e 31 2f 49 54 43 4d 5f 48 03
    00 03 00 00 02 13 00 01 05
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_CHECK_FOR_NEW_VERSION_RES<0f>
Unit ID: 00
Cookie: 00
Raw IE: 00 25 00 04 48 4f 53 54 04 64 68 61 6e 19 68 74 74 70 3a 2f 2f 31 39 32 2e 31
    36 38 2e 31 2e 31 2f 49 54 43 4d 5f 48
Parsed IE: CMND_IE_CHECK_SW_VER_RES<0x42>  Length<37>  Value<Result: 0; SwLen: 4;
    SwVer: <HOST>; HwLen: 4; HwVer: <dhan>; UrlLen: 25; URL:
    <http://192.168.1.1/ITCM_H>>
Raw IE: 00 03 00 00 02
Parsed IE: CMND_IE_UNIT_ADDR<0x03>  Length<3>  Value<000002>
Raw IE: 00 01 05
Parsed IE: CMND_IE_OTA_COOKIE<0x13>  Length<1>  Value<05>
```

CMND: Yes, there is a new version of HOST FW

MCU: Let's download that new software.

```
MCU --> CMND:
Message Length: 53
Raw message: da da 00 31 00 00 02 0c 03 4b 32 00 24 04 48 4f 53 54 04 64 68 61 6e 19 68 74 74 70 3a
    2f 2f 31 39 32 2e 31 36 38 2e 31 2e 31 2f 49 54 43 4d 5f 48 34 00 01 02
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_DOWNLOAD_START_REQ<03>
Unit ID: 00
Cookie: 00
```

```
Raw IE: 00 24 04 48 4f 53 54 04 64 68 61 6e 19 68 74 74 70 3a 2f 2f 31 39 32 2e 31 36 38 2e 31 2e
    31 2f 49 54 43 4d 5f 48
Parsed IE: CMND_IE_NEW_SW_INFO<0x32>  Length<36>  Value<SwLen: 4; SwVer: <HOST>; HwLen: 4; HwVer:
    <dhan>; UrlLen: 25; URL: <http://192.168.1.1/ITCM_H>>
Raw IE: 00 01 02
Parsed IE: CMND_IE_IMAGE_TYPE<0x34>  Length<1>  Value<ImageType: MCU<02> >

              CMND --> MCU:
              Message Length: 10
              Raw message: da da 00 06 00 00 02 0c 04 18
              Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
              Message ID: CMND_MSG_SUOTA_DOWNLOAD_START_RES<04>
              Unit ID: 00
              Cookie: 00
```

CMND: Ok...

```
              CMND --> MCU:
              Message Length: 21
              Raw message: da da 00 11 f2 00 02 0c 05 aa 00 00 01 00 24 00 04 00 00 4e 1d
              Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
              Message ID: CMND_MSG_SUOTA_IMAGE_READY_IND<05>
              Unit ID: 00
              Cookie: f2
              Raw IE: 00 01 00
              Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<CMND_RC_SUOTA_SUCCESS>
              Raw IE: 00 04 00 00 4e 1d
              Parsed IE: CMND_IE_U32<0x24>  Length<4>  Value<00004e1d>
```

CMND: The image was downloaded successfully, it's size is 19997

MCU: Ok...

```
MCU --> CMND:
Message Length: 10
Raw message: da da 00 06 00 00 02 0c 06 1a
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_IMAGE_READY_RES<06>
Unit ID: 00
Cookie: 00
```

MCU: Read the image at offset 0, size 100

```
MCU --> CMND:
Message Length: 19
Raw message: da da 00 0f cb 00 02 0c 0a 93 37 00 06 00 00 00 00 00 64
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_READ_FILE_REQ<0a>
Unit ID: 00
Cookie: cb
Raw IE: 00 06 00 00 00 00 00 64
Parsed IE: CMND_IE_READ_FILE_DATA_REQ<0x37>  Length<6>  Value<Offset: 0, Length: 100>

              CMND --> MCU:
              Message Length: 123
              Raw message: da da 00 77 1f 00 02 0c 0b 2d 00 00 01 00 36 00 6a 00 00 00 00 00 64 44
                  53 50 20 47 72 6f 75 70 2d 4c 6f 67 66 69 6c 65 0d 0a 4c 6f 67 20 53 74 61 72 74
                  65 64 20 61 74 3a 20 30 38 2d 31 31 2d 32 30 31 37 20 31 34 3a 30 33 3a 35 35 0d
                  0a 56 65 72 73 69 6f 6e 20 31 2e 31 2e 4b 20 2d 20 42 75 69 6c 64 20 33 30 0d 0a
                  0d 0a 30 38 2d 31 31 2d 32 30 31 37 20 31 34 3a 32 30
              Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
              Message ID: CMND_MSG_SUOTA_READ_FILE_RES<0b>
              Unit ID: 00
              Cookie: 1f
```

CMND: Here is the image fragment you requested

```
         Raw IE: 00 01 00
         Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<GENERAL_SUCCESS>
         Raw IE: 00 6a 00 00 00 00 00 64 44 53 50 20 47 72 6f 75 70 2d 4c 6f 67 66 69 6c 65 0d
            0a 4c 6f 67 20 53 74 61 72 74 65 64 20 61 74 3a 20 30 38 2d 31 31 2d 32 30 31 37
            20 31 34 3a 30 33 3a 35 35 0d 0a 56 65 72 73 69 6f 6e 20 31 2e 31 2e 4b 20 2d 20
            42 75 69 6c 64 20 33 30 0d 0a 0d 0a 30 38 2d 31 31 2d 32 30 31 37 20 31 34 3a 32
            30
         Parsed IE: CMND_IE_READ_FILE_DATA_RES<0x36>  Length<106>  Value<Offset: 0,
            Length:100,
            Data:4453502047726f75702d4c6f6766696c650d0a4c6f6720537461727465642061743a2030382d
            31312d323031372031343a30333a35350d0a56657273696f6e20312e312e4b202d204275696c64203
            3300d0a0d0a30382d31312d323031372031343a3230>
```

MCU: Read the image at offset 100, size 100

```
MCU --> CMND:
Message Length: 19
Raw message: da da 00 0f cc 00 02 0c 0a f8 37 00 06 00 00 00 64 00 64
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_READ_FILE_REQ<0a>
Unit ID: 00
Cookie: cc
Raw IE: 00 06 00 00 00 64 00 64
Parsed IE: CMND_IE_READ_FILE_DATA_REQ<0x37>  Length<6>  Value<Offset:100, Length:100>

         CMND --> MCU:
         Message Length: 123
         Raw message: da da 00 77 1f 00 02 0c 0b eb 00 00 01 00 36 00 6a 00 00 00 64 00 64 3a
            33 32 0d 0a 43 4d 4e 44 20 2d 2d 3e 20 4d 43 55 3a 0d 0a 4d 65 73 73 61 67 65 20
            4c 65 6e 67 74 68 3a 20 31 34 0d 0a 52 61 77 20 6d 65 73 73 61 67 65 3a 20 64 61
            20 64 61 20 30 30 20 30 61 20 30 30 20 30 30 20 30 30 20 30 30 20 30 37 20 31 32
            20 30 30 20 30 30 20 30 31 20 30 30 20 0d 0a 53 65 72
         Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
         Message ID: CMND_MSG_SUOTA_READ_FILE_RES<0b>
         Unit ID: 00
         Cookie: 1f
         Raw IE: 00 01 00
         Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<GENERAL_SUCCESS>
         Raw IE: 00 6a 00 00 00 64 00 64 3a 33 32 0d 0a 43 4d 4e 44 20 2d 2d 3e 20 4d 43 55 3a
            0d 0a 4d 65 73 73 61 67 65 20 4c 65 6e 67 74 68 3a 20 31 34 0d 0a 52 61 77 20 6d
            65 73 73 61 67 65 3a 20 64 61 20 64 61 20 30 30 20 30 61 20 30 30 20 30 30 20 30
            30 20 30 30 20 30 37 20 31 32 20 30 30 20 30 30 20 30 31 20 30 30 20 0d 0a 53 65
            72
         Parsed IE: CMND_IE_READ_FILE_DATA_RES<0x36>  Length<106>  Value<Offset: 100, Length:
            100, Data:
            3a33320d0a434d4e44202d2d3e204d43553a0d0a4d657373616765204c656e6774683a2031340d0a5
            26177206d6573736167653a2064612064612030302030612030302030302030302030302030372031
            322030302030302030312030302000d0a536572>
```

CMND: Here is the image fragment you requested

<Skipped many similar pairs of messages CMND_MSG_SUOTA_READ_FILE_REQ and CMND_MSG_SUOTA_READ_FILE_RES>

MCU: Read the image at offset 19984, size 100

```
MCU --> CMND:
Message Length: 19
Raw message: da da 00 0f 93 00 02 0c 0a b9 37 00 06 00 00 4e 10 00 64
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
```

```
Message ID: CMND_MSG_SUOTA_READ_FILE_REQ<0a>
Unit ID: 00
Cookie: 93
Raw IE: 00 06 00 00 4e 10 00 64
Parsed IE: CMND_IE_READ_FILE_DATA_REQ<0x37>  Length<6>  Value<Offset: 19984, Length: 100>

              CMND --> MCU:
              Message Length: 36
              Raw message: da da 00 20 00 00 02 0c 0b 45 00 00 01 00 36 00 13 00 00 4e 10 00 0d 61
                  6c 3c 73 75 63 63 65 73 73 3e 0d 0a
              Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
              Message ID: CMND_MSG_SUOTA_READ_FILE_RES<0b>
              Unit ID: 00
              Cookie: 00
              Raw IE: 00 01 00
              Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<GENERAL_SUCCESS>
              Raw IE: 00 13 00 00 4e 10 00 0d 61 6c 3c 73 75 63 63 65 73 73 3e 0d 0a
              Parsed IE: CMND_IE_READ_FILE_DATA_RES<0x36>  Length<19>  Value<Offset: 19984, Length:
                  13, Data: 616c3c737563636573733e0d0a>
```

CMND: Here is the image fragment you requested, but the size is 13 (0x000d)

MCU: Read the image at offset 19997, size 100

```
MCU --> CMND:
Message Length: 19
Raw message: da da 00 0f 94 00 02 0c 0a c7 37 00 06 00 00 4e 1d 00 64
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_READ_FILE_REQ<0a>
Unit ID: 00
Cookie: 94
Raw IE: 00 06 00 00 4e 1d 00 64
Parsed IE: CMND_IE_READ_FILE_DATA_REQ<0x37>  Length<6>  Value<Offset: 19997, Length: 100>

              CMND --> MCU:
              Message Length: 23
              Raw message: da da 00 13 00 00 02 0c 0b d4 00 00 01 00 36 00 06 00 00 4e 1d 00 00
              Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
              Message ID: CMND_MSG_SUOTA_READ_FILE_RES<0b>
              Unit ID: 00
              Cookie: 00
              Raw IE: 00 01 00
              Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<GENERAL_SUCCESS>
              Raw IE: 00 06 00 00 4e 1d 00 00
              Parsed IE: CMND_IE_READ_FILE_DATA_RES<0x36>  Length<6>  Value<Offset: 19997, Length:
                  0>
```

CMND: The requested fragment is empty

MCU: What's your current version?

```
MCU --> CMND:
Message Length: 14
Raw message: da da 00 0a 00 00 00 00 0b 35 1e 00 01 01
Service ID: CMND_SERVICE_ID_GENERAL<00 00>
Message ID: CMND_MSG_GENERAL_GET_VERSION_REQ<0b>
Unit ID: 00
Cookie: 00
Raw IE: 00 01 01
Parsed IE: CMND_IE_U8<0x1E>  Length<1>  Value<CMND_GENERAL_VERSION_SW>

              CMND --> MCU:
              Message Length: 19
              Raw message: da da 00 0f 00 00 00 00 0c 29 09 00 06 05 33 33 2e 30 36
              Service ID: CMND_SERVICE_ID_GENERAL<00 00>
              Message ID: CMND_MSG_GENERAL_GET_VERSION_RES<0c>
              Unit ID: 00
              Cookie: 00
              Raw IE: 00 06 05 33 33 2e 30 36
              Parsed IE: CMND_IE_VERSION<0x09>  Length<6>  Value<Length: 5; Value: <33.06>>
```

CMND: My current version is 33.06

MCU: Add my version to yours - set it to 33.06#1.0.1

```
MCU --> CMND:
Message Length: 29
Raw message: da da 00 19 00 00 02 03 03 ce 0b 00 0f 00 07 00 0c 0b 33 33 2e 30 36 23 31 2e 30 2e 31
Service ID: CMND_SERVICE_ID_PARAMETERS<02 03>
Message ID: CMND_MSG_PARAM_SET_REQ<03>
Unit ID: 00
Cookie: 00
Raw IE: 00 0f 00 07 00 0c 0b 33 33 2e 30 36 23 31 2e 30 2e 31
Parsed IE: CMND_IE_PARAMETER<0x0B>  Length<16>  Value<EepParamType:CMND_PARAM_TYPE_EEPROM<0x00>,
    EepParamId:CMND_PARAM_EEPROM_SW_VERISON<0x07>, ParamLen:13, Value<Length: 12; Value:
    <33.06s#1.0.1>>>
```

```
              CMND --> MCU:
              Message Length: 21
              Raw message: da da 00 11 00 00 02 03 04 31 0b 00 04 00 07 00 00 00 00 01 00
              Service ID: CMND_SERVICE_ID_PARAMETERS<02 03>
              Message ID: CMND_MSG_PARAM_SET_RES<04>
              Unit ID: 00
              Cookie: 00
              Raw IE: 00 04 00 07 00 00
              Parsed IE: CMND_IE_PARAMETER<0x0B>  Length<4>
                  Value<EepParamType:CMND_PARAM_TYPE_EEPROM<0x00>,EepParamId:CMND_PARAM_EEPROM_SW_V
                  ERISON<0x07>,ParamLen:0>
              Raw IE: 00 01 00
              Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<CMND_RC_OK>
```

CMND: The parameter was set successfully

MCU: Let's complete the SUOTA with current version 33.06#1.0.1

```
MCU --> CMND:
Message Length: 26
Raw message: da da 00 16 00 00 02 0c 08 d8 35 00 0f 00 0b 33 33 2e 30 36 23 31 2e 30 2e 31
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_UPGRADE_COMPLETED_REQ<08>
Unit ID: 00
Cookie: 00
Raw IE: 00 0f 00 0b 33 33 2e 30 36 23 31 2e 30 2e 31
Parsed IE: CMND_IE_SW_VER_INFO<0x35>  Length<15>  Value<Length: 11; Value: <33.06#1.0.1>>
```

```
              CMND --> MCU:
              Message Length: 14
              Raw message: da da 00 0a 00 00 00 00 07 12 00 00 01 00
              Service ID: CMND_SERVICE_ID_GENERAL<00 00>
              Message ID: CMND_MSG_GENERAL_LINK_CFM<07>
              Unit ID: 00
              Cookie: 00
              Raw IE: 00 01 00
              Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<CMND_RC_OK>
```

CMDN: The BS was informed about SUOTA completion

10.8 Appendix 8: Example log of Failed SUOTA for Node Host initiated by itself

MCU: Is there any new software for Node Host available?

```
MCU --> CMND:
Message Length: 21
Raw message: da da 00 11 00 00 02 0c 0e 8b 33 00 08 04 48 4f 53 54 02 68 77
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_CHECK_FOR_NEW_VERSION_REQ<0e>
Unit ID: 00
Cookie: 00
Raw IE: 00 08 04 48 4f 53 54 02 68 77
Parsed IE: CMND_IE_CURRENT_SW_INFO<0x33>  Length<8>  Value<SwLen: 4; SwVer: <HOST>; HwLen: 2;
    HwVer: <hw>>
```

```
              CMND --> MCU:
              Message Length: 14
              Raw message: da da 00 0a 00 00 00 00 07 12 00 00 01 00
              Service ID: CMND_SERVICE_ID_GENERAL<00 00>
              Message ID: CMND_MSG_GENERAL_LINK_CFM<07>
              Unit ID: 00
              Cookie: 00
              Raw IE: 00 01 00
              Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<CMND_RC_OK>
```

CMND: I sent this request to BS

```
              CMND --> MCU:
              Message Length: 58
              Raw message: da da 00 36 00 00 02 0c 0f 97 42 00 23 00 04 48 4f 53 54 02 68 77 19 68
                  74 74 70 3a 2f 2f 31 39 32 2e 31 36 38 2e 31 2e 31 2f 49 54 43 4d 5f 48 03 00 03
                  00 00 02 13 00 01 05
              Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
              Message ID: CMND_MSG_SUOTA_CHECK_FOR_NEW_VERSION_RES<0f>
              Unit ID: 00
              Cookie: 00
              Raw IE: 00 23 00 04 48 4f 53 54 02 68 77 19 68 74 74 70 3a 2f 2f 31 39 32 2e 31 36 38
                  2e 31 2e 31 2f 49 54 43 4d 5f 48
              Parsed IE: CMND_IE_CHECK_SW_VER_RES<0x42>  Length<35>  Value<Result: 0; SwLen: 4;
                  SwVer: <HOST>; HwLen: 2; HwVer: <hw>; UrlLen: 25; URL:
                  <http://192.168.1.1/ITCM_H>>
              Raw IE: 00 03 00 00 02
              Parsed IE: CMND_IE_UNIT_ADDR<0x03>  Length<3>  Value<000002>
              Raw IE: 00 01 05
              Parsed IE: CMND_IE_OTA_COOKIE<0x13>  Length<1>  Value<05>
```

CMND: Yes, there is a new version of HOST FW

MCU: Let's download that new software.

```
MCU --> CMND:
Message Length: 51
Raw message: da da 00 2f 00 00 02 0c 03 89 32 00 22 04 48 4f 53 54 02 68 77 19 68 74 74 70 3a 2f 2f
    31 39 32 2e 31 36 38 2e 31 2e 31 2f 49 54 43 4d 5f 48 34 00 01 02
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_DOWNLOAD_START_REQ<03>
Unit ID: 00
```

```
Cookie: 00
Raw IE: 00 22 04 48 4f 53 54 02 68 77 19 68 74 74 70 3a 2f 2f 31 39 32 2e 31 36 38 2e 31 2e 31 2f
    49 54 43 4d 5f 48
Parsed IE: CMND_IE_NEW_SW_INFO<0x32>  Length<34>  Value<SwLen: 4; SwVer: <HOST>; HwLen: 2; HwVer:
    <hw>; UrlLen: 25; URL: <http://192.168.1.1/ITCM_H>>
Raw IE: 00 01 02
Parsed IE: CMND_IE_IMAGE_TYPE<0x34>  Length<1>  Value<ImageType: MCU<02> >

              CMND --> MCU:
              Message Length: 10
              Raw message: da da 00 06 00 00 02 0c 04 18
              Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
              Message ID: CMND_MSG_SUOTA_DOWNLOAD_START_RES<04>
              Unit ID: 00
              Cookie: 00
```

CMND: Ok...

```
              CMND --> MCU:
              Message Length: 21
              Raw message: da da 00 11 00 00 02 0c 05 4f 00 00 01 02 24 00 04 00 00 00 00
              Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
              Message ID: CMND_MSG_SUOTA_IMAGE_READY_IND<05>
              Unit ID: 00
              Cookie: 00
              Raw IE: 00 01 02
              Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<CMND_RC_SUOTA_ERR_SESSION>
              Raw IE: 00 04 00 00 00 00
              Parsed IE: CMND_IE_U32<0x24>  Length<4>  Value<00000000>
```

CMND after some time: Oops, session error was occurred.

CMND: Understood.

```
MCU --> CMND:
Message Length: 10
Raw message: da da 00 06 00 00 02 0c 06 1a
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_IMAGE_READY_RES<06>
Unit ID: 00
Cookie: 00
```

MCU: Let's complete the SUOTA with version 33.06s#1.0.1

```
MCU --> CMND:
Message Length: 27
Raw message: da da 00 17 00 00 02 0c 08 d8 35 00 0e 00 0c 33 33 2e 30 36 73 23 31 2e 30 2e 31
Service ID: CMND_SERVICE_ID_SUOTA<02 0c>
Message ID: CMND_MSG_SUOTA_UPGRADE_COMPLETED_REQ<08>
Unit ID: 00
Cookie: 00
Raw IE: 00 0e 00 0c 33 33 2e 30 36 73 23 31 2e 30 2e 31
Parsed IE: CMND_IE_SW_VER_INFO<0x35>  Length<14>  Value<Length: 12; Value: <33.06s#1.0.1>>

              CMND --> MCU:
              Message Length: 14
              Raw message: da da 00 0a 00 00 00 00 07 24 00 00 01 12
              Service ID: CMND_SERVICE_ID_GENERAL<00 00>
              Message ID: CMND_MSG_GENERAL_LINK_CFM<07>
              Unit ID: 00
              Cookie: 00
              Raw IE: 00 01 12
              Parsed IE: CMND_IE_RESPONSE<0x00>  Length<1>  Value<CMND_RC_FAIL_SEND_LOCK>
```

CMND: Failed to inform BS about SUOTA completion

LIST OF TABLES

LIST OF FIGURES