

Serial Protocol

Document Version: V1.22

2024-09-11

deCONZ

Table of contents

- Table of contents.....1
- Document history 3
- Abbreviations 4
- 1. Overview..... 5
- 2. Requirements 5
 - 2.1 Required Hardware5
 - 2.2 Supported Operating Systems6
- 3. Target Audience..... 6
- 4. Transmission Protocol 6
 - 4.1 16-bit CRC Calculation6
- 5. Application Protocol 7
 - 5.1 Read Firmware Version.....8
 - 5.1.1 Read Firmware Version Request..... 8
 - 5.1.2 Read Firmware Version Response 8
- 6. Configure Network Parameters..... 9
 - 6.1 Read Configuration 10
 - 6.1.1 Read Parameter Request.....10
 - 6.1.2 Read Parameter Response..... 11
 - 6.2 Write Configuration..... 11
 - 6.2.1 Write Parameter Request 11
 - 6.2.2 Write Parameter Response..... 11
 - 6.2.3 Zigbee Install Codes12

7. Control Network State.....	13
7.1 Reading Network State	13
7.1.1 Device State Request	13
7.1.2 Device State Response.....	13
7.2 Create or Join Network	14
7.2.1 Create or Join Network Request	14
7.2.2 Create or Join Network Response.....	14
7.3 Leave Network	15
7.3.1 Leave Network Request	15
7.3.2 Leave Network Response.....	15
7.4 Receiving Data	16
7.4.1 Received Data Notification.....	16
7.4.1 Read Received Data Request.....	16
7.4.3 Read Received Data Response	17
7.4.4 MAC Poll Indication	18
7.4.5 MAC Beacon Indication	18
7.5 Sending Data	19
7.5.1 Enqueue Send Data Request	19
7.5.2 Enqueue Send Data Response.....	20
7.5.3 Query Send Data State Request.....	20
7.5.4 Query Send Data State Response	21

Document history

Date	Version	Description
2017-01-15	1.00	<ul style="list-style-type: none">Initial version
2017-08-04	1.10	<ul style="list-style-type: none">Documented missing CRC16Corrected read parameter request field 'frame length'.Replaced parameter '0x25 device type' with '0x09 APS designed coordinator'.Query send data response: rename third field from 'Status' to 'Reserved'. Correct payload length description.Mark parameter '0x07 NWK address' as read only.Documented command '0x0E status change and corrected related section 'Receiving Data Notification'.
2017-11-28	1.11	<ul style="list-style-type: none">Read received data request: add flag to return only short addresses as source address. Since firmware 0x261b0500.
2018-09-19	1.12	<ul style="list-style-type: none">Correct 'Read/Write Parameter' frame length'Read Received Data Request' add flag to include last hop address in responseDocument parameter 'Protocol Version'
2019-04-11	1.13	<ul style="list-style-type: none">Provide new flag 0x04 in APS_DATA_INDICATION to query both, 16-bit and 64-bit source address (requires protocol version 0x010B)
2019-05-25	1.14	<ul style="list-style-type: none">Document query firmware version command 'VERSION'Document parameter 'Watchdog TTL'Add ConBee II to supported devices
2019-08-05	1.15	<ul style="list-style-type: none">Describe Bootloader Update (work in progress)
2019-08-16	1.16	<ul style="list-style-type: none">Describe 'Link Key Parameter' to configure default TC link key
2019-08-28	1.17	<ul style="list-style-type: none">Describe 'Predefined NWK PANID Parameter'
2020-06-19	1.18	<ul style="list-style-type: none">Describe 'NWK Frame Counter Parameter'Add RaspBee II to supported devices
2020-09-13	1.19	<ul style="list-style-type: none">Describe different frame lengths for 'Device State Response'
2021-01-26	1.20	<ul style="list-style-type: none">Describe regression in 'Read Firmware Version Request'Remove Raspbian Jessie from supports operating systems list since it reached end-of-life.Describe 'MAC Beacon Indication'Describe 'MAC Poll Indication'Describe how the firmware can be rebooted via 'Watchdog TTL' parameterClarify when 'Write Parameter Request' configuration changes become active
2021-07-13	1.21	<ul style="list-style-type: none">Read received data request: add flag to return only short addresses as source address. Since firmware 0x261b0500.
2024-09-10	1.22	<ul style="list-style-type: none">Use new layout and format for this documentAdd section about Zigbee Install Code usageList ConBee /// as supported device

Abbreviations

Abbreviation	Description
APS	Application Support
GUI	Graphical User Interface
IEEE 802.15.4	Standard, applicable to low-rate wireless personal area networks (WPAN)
LQI	Link Quality Indicator
NWK	Network
PANID	Personal Area Network Identifier
RSSI	Received Signal Strength Indication
SLIP	Serial Line Internet Protocol
TC	Trust Center
(W)PAN	(Wireless) Personal Area Network
ZCL	Zigbee Cluster Library
ZLL	Zigbee Light Link Profile
ZHA	Zigbee Home Automation Profile
ZDP	Zigbee Device Profile
Zigbee	Wireless networking standard targeted at low-power applications

1. Overview

Zigbee is a technology which offers a powerful solution to a wide range of low-power, low cost wireless sensor network applications. Some popular application profiles are Home Automation, Smart Energy and Health Care.

In many stages of a product development process it is necessary to interact with the devices in order to verify their correct operation. To achieve this in an efficient way extra PC tools are often built around the related application first for the developer and later for deployment, for operation and for maintenance. The deCONZ application from dresden elektronik is a powerful graphical tool addressing all those stages.

deCONZ provides comprehensive monitoring, control and commissioning capabilities based on the Zigbee PRO specification. The application core is kept completely generic and is therefore not limited to a specific application profile. All Zigbee application specifics like devices, profiles and clusters are described in XML files. Based on this information, deCONZ can generate a fully functional graphical user interface for each device.

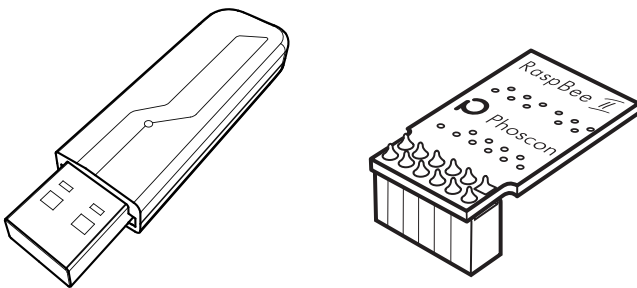
2. Requirements

2.1 Required Hardware

To use the deCONZ application you need appropriate hardware that is capable of communicating with other Zigbee devices. dresden elektronik offers two solutions for that purpose. The ConBee is a Zigbee capable radio USB dongle that turns any PC or MAC into a Zigbee gateway. The RaspBee is a add-on board for the Raspberry Pi GPIO pin header.

The deCONZ Serial Protocol is supported by: RaspBee I and II as well as ConBee I, II and III.

For detailed setup instructions please refer to: <https://phoscon.de>



2.2 Supported Operating Systems

- Microsoft Windows 7, 8, 8.1, 10 and 11
- Debian and Canonical Ubuntu Linux
- Raspberry Pi OS
- Apple macOS

3. Target Audience

This document describes the serial protocol used between the deCONZ application and the radio module. The targeted audience should be familiar with the Zigbee PRO protocol — especially the Application Support Layer (APS), Zigbee Device Profile (ZDP) and Zigbee Cluster Library (ZCL). A deep understanding of these is required to utilize the protocol, since the radio module represents only a basic modem.

Details of the Zigbee protocol and its various standards like Zigbee Light Link (ZLL) and Zigbee Home Automation (ZHA) are described in their respective specifications. These can be obtained from the <https://www.zigbee.org> website (registration required). A good understanding of the Zigbee PRO specification is required to use this protocol.

4. Transmission Protocol

The application protocol frames which are used by the deCONZ application to communicate with the microcontroller are encapsulated in the Serial Line Internet Protocol (SLIP). For a comprehensive documentation and a reference implementation of SLIP, please refer to RFC 1055.

4.1 16-bit CRC Calculation

As extension each frame contains a 16-bit CRC after the content, calculated over the complete frame payload as described in following pseudo code:

```
U16 crc = 0;
for (i = 0; i < payloadLength; i++)
  crc += payload[i];
U8 crc0 = (~crc + 1) & 0xFF;
U8 crc1 = ((~crc + 1) >> 8) & 0xFF;
```

5. Application Protocol

The application protocol describes the messages which are send between the host application and the gateway. In general the procotol follows a request and response pattern with additional notification messages, which are sent from the gateway to the host, without a prior request.

Value	Status Code
0x00	SUCCESS
0x01	FAILURE
0x02	BUSY
0x03	TIMEOUT
0x04	UNSUPPORTED
0x05	ERROR
0x06	NO_NETWORK
0x07	INVALID_VALUE

Value	Network State
0x00	NET_OFFLINE
0x01	NET_JOINING
0x02	NET_CONNECTED
0x03	NET_LEAVING

ID	Command
0x07	DEVICE_STATE
0x08	CHANGE_NETWORK_STATE
0x0A	READ_PARAMETER
0x0B	WRITE_PARAMETER
0x0E	DEVICE_STATE_CHANGED
0x0D	VERSION
0x12	APS_DATA_REQUEST
0x04	APS_DATA_CONFIRM
0x17	APS_DATA_INDICATION
0x1C	MAC_POLL_INDICATION
0x1F	MAC_BEACON_INDICATION
0x21	UPDATE_BOOTLOADER

5.1 Read Firmware Version

The firmware version can be used to check if a fresh enough version is installed and which underlying platform is used. Note that for feature detection the 'Protocol Version' parameter should be considered.

5.1.1 Read Firmware Version Request

Type	Field	Value
U8	Command ID	VERSION (0×0D)
U8	Sequence number	0–255
U8	Reserved	Shall be set to 0
U16	Frame length	9
U32	Reserved	Shall be set to 0

Important: Due a regression in firmware version 0×26660700 the additional empty U32 'Reserved' field needs to be part of the request, which raises the frame length from 5 to 9. Older versions also accept the additional field.

5.1.2 Read Firmware Version Response

Type	Field	Value
U8	Command ID	VERSION (0×0D)
U8	Sequence number	Same as request
U8	Status	SUCCESS
U16	Frame length	9
U32	Version	<p>Example: 0×26330500, where the bytes represent:</p> <p>0×26 — Major version</p> <p>0×33 — Minor version</p> <p>0×05 — Platform</p> <p>0×00 — Reserved</p> <p>Platform:</p> <p>0×05 — ConBee and RaspBee (AVR)</p> <p>0×07 — ConBee II and RaspBee II (ARM/R21)</p> <p>0×09 — ConBee III (MG21)</p>

6. Configure Network Parameters

Various parameters define how the device participates in a Zigbee network. Some of these parameters are read-only and will be set automatically by the stack when the network operation is started.

ID	Name	Type	Description	Mode
0x01	MAC Address	U64	0x0000000000000001 – 0xFFFFFFFFFFFFFFFE	R
0x05	NWK PANID	U16	0x0000 – 0xFFFF	RW
0x07	NWK Address	U16	0x0000 – 0xFFFE	R
0x08	NWK Extended PANID	U64	0x0000000000000000 – 0xFFFFFFFFFFFFFFF	R
0x09	APS Designed Coordinator	U8	0x01 — Coordinator, the node will form a network and let other nodes join. 0x00 — Router, the node will join a network	RW
0x0A	Channel Mask	U32	0x00000000 – 0x7FFF800	RW
0x0B	APS Extended PANID	U64	0x0000000000000000 – 0xFFFFFFFFFFFFFFF	RW
0x0E	Trust Center Address	U64	0x0000000000000000 – 0xFFFFFFFFFFFFFFF	RW
0x10	Security Mode	U8	0x00 — no security 0x01 — preconfigured network key 0x02 — network key from trust center 0x03 — no master but trust center link key	RW
0x15	Predefined NWK PANID	U8	0x00 — not predefined: The NWK PANID will be selected or obtained dynamically. 0x01 — predefined: The value of parameter NWK PANID (0x05) will be used to join or form a network.	RW
0x18	Network Key	U8[16]	Encryption key to secure network traffic	RW
0x19	Link Key	U64 U8[16]	MAC Address Link key to be used during joining. The key is only included in a write request and read response. The read request shall only contain the MAC address.	RW

ID	Name	Type	Description	Mode
0×1C	Current Channel	U8	11–26	R
0×22	Protocol Version	U16	Version of the implemented protocol	R
0×24	NWK Update ID	U8	0–255	RW
0×26	Watchdog TTL	U32	Watchdog timeout in seconds. Must be reset by the application periodically (since protocol version 0×0108) By writing a lower value like 2 seconds, the firmware can be rebooted.	RW
0×27	NWK Frame Counter	U32	Outgoing security frame counter. It shall be only set initially when joining or forming a network.	RW
0×28	App ZDP Response Handling	U16	A bitmap describing which ZDP responses the application wants to handle. The bitmap is not persistent and resets on every power-up of the firmware. Default value is 0×0000. (since firmware version 0×266f0700) Supported flags: 0×0001 Node Descriptor response Reading and writing the bitmap on older firmware results in UNSUPPORTED status code.	RW

6.1 Read Configuration

By reading parameters the current configuration can be obtained. Be aware that this configuration might not reflect the active configuration, since changes to parameters affect the network operation only as soon as it's stopped and started again.

6.1.1 Read Parameter Request

Type	Field	Value
U8	Command ID	READ_PARAMETER (0×0A)
U8	Sequence number	0–255
U8	Reserved	Shall be set to 0
U16	Frame length	7 + Payload length
U16	Payload length	1
U8	Parameter ID	An identifier from Table 6: Parameters

6.1.2 Read Parameter Response

If the response status is SUCCESS the parameter data is included in the response according to its definition in '6. Configure Network Parameters'. If the status is UNSUPPORTED the 'Length' field is 0 and the fields 'Parameter ID' and 'Parameter' aren't included in the response.

Type	Field	Value
U8	Command ID	READ_PARAMETER (0x0A)
U8	Sequence number	Same as request
U8	Status	SUCCESS or UNSUPPORTED
U16	Frame length	7 + Payload length
U16	Payload length	1 + Length of parameter
U8	Parameter ID	Same as request
Variable	Parameter	The parameter

6.2 Write Configuration

6.2.1 Write Parameter Request

Writing parameters updates the network configuration.

Note: Changed network configuration parameters become affective only after sending a 7.3.1 Leave Network Request followed by a 7.2.1 Create or Join Network Request.

Type	Field	Value
U8	Command ID	WRITE_PARAMETER (0x0B)
U8	Sequence number	0-255
U8	Reserved	Shall be set to 0
U16	Frame length	7 + Payload length
U16	Payload length	1 + Length of parameter
U8	Parameter ID	An identifier from '6. Configure Network Parameters'
Variable	Parameter	The parameter

6.2.2 Write Parameter Response

Type	Field	Value
U8	Command ID	WRITE_PARAMETER (0x0B)
U8	Sequence number	Same as request
U8	Status	SUCCESS, UNSUPPORTED or INVALID_VALUE
U16	Frame length	7 + Payload length
U16	Payload length	1
U8	Parameter ID	An identifier from '6. Configure Network Parameters'

6.2.3 Zigbee Install Codes

The Serial Protocol and firmware doesn't support Zigbee Install Codes directly. In order to join a device via Install Code, a device specific Link Key needs to be derived from the Install Code by the host application.

This 16-byte Link Key together with the device MAC address can then be configured by a Write Parameter Request (0x0B) for the Link Key parameter (0x19).

Once configured, during joining process, the device will receive the Network Key encrypted with the Link Key instead of the global Trustee Center Link Key.

A C++ OpenSSL reference implementation for the key derivation can be found at:

<https://github.com/dresden-elektronik/deconz-rest-plugin/blob/master/crypto/mmohash.cpp>

7. Control Network State

Setting the network state is crucial to apply any new network parameters. When new parameters have been written they become only active after leaving and joining (creating) the network.

7.1 Reading Network State

7.1.1 Device State Request

Type	Field	Value
U8	Command ID	DEVICE_STATE (0×07)
U8	Sequence number	0–255
U8	Reserved	Shall be set to 0
U16	Frame length	8
U8	Reserved	Shall be set to 0
U8	Reserved	Shall be set to 0
U8	Reserved	Shall be set to 0

7.1.2 Device State Response

The device state determines if the device is operation in a Zigbee network and if so, various flags provide the state of incoming and outgoing command queues. The 'Network state' field value can be NET_OFFLINE, NET_CONNECTED, NET_JOINING and NET_LEAVING.

Type	Field	Value
U8	Command ID	DEVICE_STATE (0×07)
U8	Sequence number	Same as request
U8	Status	SUCCESS
U16	Frame length	7 or 8
U8	Reserved	0000 0011 — Network state 0000 0100 — APSDE-DATA.confirm flag (0×04) 0000 1000 — APSDE-DATA.indication flag (0×08) 0001 0000 — Configuration changed flag (0×10) 0010 0000 — APSDE-DATA.request free slots flag (0×20)
U8	Reserved	Shall be ignored
U8	Reserved	Shall be ignored, only included when frame length is 8

7.2 Create or Join Network

7.2.1 Create or Join Network Request

The device can create a network when configured as coordinator and trust center, or join a network as a router.

7.2.2 Create or Join Network Response

Type	Field	Value
U8	Command ID	CHANGE_NETWORK_STATE (0×08)
U8	Sequence number	0–255
U8	Reserved	Shall be set to 0
U16	Frame length	6
U8	Network state	NET_CONNECTED (0×02)

A status of SUCCESS means the request will be processed; the network state transitions should be further queried with DEVICE_STATE commands once a second.

Type	Field	Value
U8	Command ID	CHANGE_NETWORK_STATE (0×08)
U8	Sequence number	Same as request
U8	Status	SUCCESS or ERROR
U16	Frame length	6
U8	Network state	NET_CONNECTED (0×02)

The following two behaviors are possible:

1. NET_OFFLINE → NET_JOINING → NET_CONNECTED
2. NET_OFFLINE → NET_JOINING → NET_OFFLINE

The second transition may occur when the device can't join a network, due to invalid parameters or because the network is not opened — which, in Zigbee terms, means no node in the network has its 'Permit Join' flag set.

7.3 Leave Network

7.3.1 Leave Network Request

Type	Field	Value
U8	Command ID	CHANGE_NETWORK_STATE (0×08)
U8	Sequence number	0–255
U8	Reserved	Shall be set to 0
U16	Frame length	6
U8	Network State	NET_OFFLINE (0×00)

7.3.2 Leave Network Response

Type	Field	Value
U8	Command ID	CHANGE_NETWORK_STATE (0×08)
U8	Sequence number	Same as request
U8	Status	SUCCESS or ERROR
U16	Frame length	6
U8	Network state	NET_CONNECTED (0×02)

7.4 Receiving Data

7.4.1 Received Data Notification

When the device receives a data frame an unsolicited DEVICE_STATE_CHANGED command will be send to the application.

Type	Field	Value
U8	Command ID	DEVICE_STATE_CHANGED (0×0E)
U8	Sequence number	0–255
U8	Status	SUCCESS
U16	Frame length	7
U8	Device state	0000 0011 — Network state 0000 0100 — APSDE-DATA.confirm flag (0×04) 0000 1000 — APSDE-DATA.indication flag (0×08) 0001 0000 — Configuration changed flag (0×10) 0010 0000 — APSDE-DATA.request free slots flag (0×20)
U8	Reserved	Shall be ignored

If the APSDE-DATA.indication flag is set, the application can read the received frame from the device by executing an APSDE-Data.indication request.

7.4.1 Read Received Data Request

Type	Field	Value
U8	Command ID	APS_DATA_INDICATION (0×17)
U8	Sequence number	0–255
U8	Reserved	Shall be set to 0
U16	Frame length	7 + Payload length
U16	Payload length	0 / 1
U8	Flags	Only included if payload length is 1 0×01 — always return source address as 16-bit short address 0×02 — put last hop address after ASDU in first two reserved bytes (since protocol version 0×0108) 0×04 — include 16-bit and 64-bit source address (since protocol version 0×010B) source address mode becomes 0×04

7.4.3 Read Received Data Response

Type	Field	Value
U8	Command ID	APS_DATA_INDICATION (0×17)
U8	Sequence number	Same as request
U8	Status	SUCCESS
U16	Frame length	7 + Payload length
U16	Payload length	Variable
U8	Device state	0000 0011 — Network state 0000 0100 — APSDE-DATA.confirm flag (0×04) 0000 1000 — APSDE-DATA.indication flag (0×08) 0001 0000 — Configuration changed flag (0×10) 0010 0000 — APSDE-DATA.request free slots flag (0×20)
U8	Destination address mode	0×01 — Group address 0×02 — NWK address 0×03 — IEEE address
*U16	16-bit destination short address	Only included if destination address mode is 0×01 or 0×02
*U64	64-bit destination extended address	Only included if destination address mode is 0×03
U8	Destination endpoint	0–255
U8	Source address mode	0×02 — NWK address 0×03 — IEEE address 0×04 — NWK and IEEE address (since protocol version 0×010B)
*U16	16-bit source short address	Only included if source address mode is 0×02 or 0×04
*U64	64-bit source extended address	Only included if source address mode is 0×03 or 0×04
U8	Source endpoint	0–255
U16	Profile ID	0×0000–0xFFFF
U16	Cluster ID	0×0000–0xFFFF
U16	ASDU length	0–127 — The APS frame payload length
U8[*]	ASDU	The APS frame payload
U8	Reserved	Shall be ignored
U8	Reserved	Shall be ignored
U8	LQI	0–255 — Link Quality Indication
U8	Reserved	Shall be ignored
U8	Reserved	Shall be ignored
U8	Reserved	Shall be ignored
U8	Reserved	Shall be ignored
I8	RSSI	-100–0 — Received Signal Strength Indication [dBm]

7.4.4 MAC Poll Indication

This frame is sent by the firmware unsolicited when a directly connected child end-device polls the coordinator for data on MAC layer. In this event the device is awake and can receive commands via APS Data Request.

Type	Field	Value
U8	Command ID	MAC_POLL_INDICATION (0×1C)
U8	Sequence number	0–255
U8	Reserved	Shall be set to 0
U16	Frame length	7 + Payload length
U16	Payload length	Variable
U8	Source address mode	0×02 — 16-bit short address 0×03 — 64-bit IEEE address
U16/U64	Source address	16-bit short address if 'Source address mode' = 0×02 64-bit IEEE address if 'Source address mode' = 0×03
U8	LQI	The received LQI value 0–255
S8	RSSI	The received RSSI value in dBm
*U32	Life time	Internal neighbor table state, only included when payload length exceeds the RSSI field.
*U32	Device timeout	Internal neighbor table state. Only included when 'Life time' is field is present.

7.4.5 MAC Beacon Indication

This frame is sent by the firmware unsolicited when an IEEE 802.15.4 beacon is received. Beacon frames are usually sent by surrounding routers and the coordinator when a device joins or rejoins a network. Note that they can belong to different networks.

Type	Field	Value
U8	Command ID	MAC_BEACON_INDICATION (0×1F)
U8	Sequence number	0–255
U8	Reserved	Shall be set to 0
U16	Frame length	7 + Payload length
U16	Payload length	Variable, at least 7
U16	Source address	16-bit short address of a router or coordinator (0×0000)
U16	PANID	The Zigbee network identifier
U8	Channel	The Zigbee channel
U8	Flags	Beacon flags
U8	Update ID	The Zigbee network Update ID
U8[]	Data	Optional additional beacon data

7.5 Sending Data

7.5.1 Enqueue Send Data Request

Type	Field	Value
U8	Command ID	APS_DATA_REQUEST (0x12)
U8	Sequence number	0–255
U8	Reserved	Shall be set to 0
U16	Frame length	7 + Payload length
U16	Payload length	Variable
U8	Request ID	0–255
U8	Flags	0
U8	Destination address mode	0x01 — Group address 0x02 — NWK address 0x03 — IEEE address
*U16	16-bit destination short address	Only included if destination address mode is 0x01 or 0x02
*U64	64-bit destination extended address	Only included if destination address mode is 0x03
*U8	Destination endpoint	0–255 Only included if destination address mode is 0x02 or 0x03
U16	Profile ID	0x0000–0xFFFF
U16	Cluster ID	0x0000–0xFFFF
U8	Source endpoint	0–255
U16	ASDU length	0–127 — The APS frame payload length
U8[*]	ASDU	The APS frame payload
U8	Tx options	0x04 — Use APS ACKs
U8	Radius	The maximum hops that the request will be forwarded. Set to 0 for unlimited hops.

7.5.2 Enqueue Send Data Response

A data response with a status of SUCCESS signals that the request is enqueued and will be processed by the device. Note that the response does not reflect the actual completion of the request, which should be further monitored with an APSDE-DATA.confirm command as soon as the relevant flag is set in the device status fields. The APS Request ID shall be used to match a specific request to its confirmation.

Type	Field	Value
U8	Command ID	APS_DATA_REQUEST (0×12)
U8	Sequence number	Same as request
U8	Status	SUCCESS, NO_NETWORK, ERROR, BUSY or INVALID_VALUE
U16	Frame length	9
U16	Payload length	2
U8	Device state	0000 0011 — Network state 0000 0100 — APSDE-DATA.confirm flag (0×04) 0000 1000 — APSDE-DATA.indication flag (0×08) 0001 0000 — Configuration changed flag (0×10) 0010 0000 — APSDE-DATA.request free slots flag (0×20)
U8	Request ID	Same as request

7.5.3 Query Send Data State Request

Type	Field	Value
U8	Command ID	APS_DATA_CONFIRM (0×04)
U8	Sequence number	0–255
U8	Reserved	Shall be set to 0
U16	Frame length	7
U16	Payload length	0

7.5.4 Query Send Data State Response

Type	Field	Value
U8	Command ID	APS_DATA_CONFIRM (0×04)
U8	Sequence number	0–255
U8	Reserved	Shall be set to 0
U16	Frame length	7
U16	Payload length	11 — For destination address mode 0×01 12 — For destination address mode 0×02 18 — For destination address mode 0×03
U8	Device state	0000 0011 — Network state 0000 0100 — APSDE-DATA.confirm flag (0×04) 0000 1000 — APSDE-DATA.indication flag (0×08) 0001 0000 — Configuration changed flag (0×10) 0010 0000 — APSDE-DATA.request free slots flag (0×20)
U8	Request ID	To match this confirmation to a specific request
U8	Destination address mode	0×01 — Group address 0×02 — NWK address 0×03 — IEEE address
*U16	16-bit destination short address	Only included if destination address mode is 0×01 or 0×02
*U64	64-bit destination extended address	Only included if destination address mode is 0×03
*U8	Destination endpoint	0–255 Only included if destination address mode is 0×02 or 0×03
U8	Source endpoint	0–255
U8	Confirm status	An Zigbee APS, NWK or MAC layer status code
U8	Reserved	Shall be ignored
U8	Reserved	Shall be ignored
U8	Reserved	Shall be ignored
U8	Reserved	Shall be ignored

dresden elektronik ingenieurtechnik gmbh
Enno-Heidebroek-Straße 12
01237 Dresden
Germany

Trademarks and acknowledgements

- Zigbee is a registered trademark of the Zigbee Alliance.
- IEEE 802.15.4 is a trademark of the Institute of Electrical and Electronics Engineers (IEEE).

These trademarks are registered by their respective owners in certain countries only. Other brands and their products are trademarks or registered trademarks of their respective holders and should be noted as such.

Disclaimer

This note is provided as-is and is subject to change without notice. Except to the extent prohibited by law, dresden elektronik ingenieurtechnik gmbh makes no express or implied warranty of any kind with regard to this guide, and specifically disclaims the implied warranties and conditions of merchantability and fitness for a particular purpose. dresden elektronik ingenieurtechnik gmbh shall not be liable for any errors or incidental or consequential damage in connection with the furnishing, performance or use of this guide.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or any means electronic or mechanical, including photocopying and recording, for any purpose other than the purchaser's personal use, without the written permission of dresden elektronik ingenieurtechnik gmbh.

Copyright © 2024 dresden elektronik ingenieurtechnik gmbh. All rights reserved.