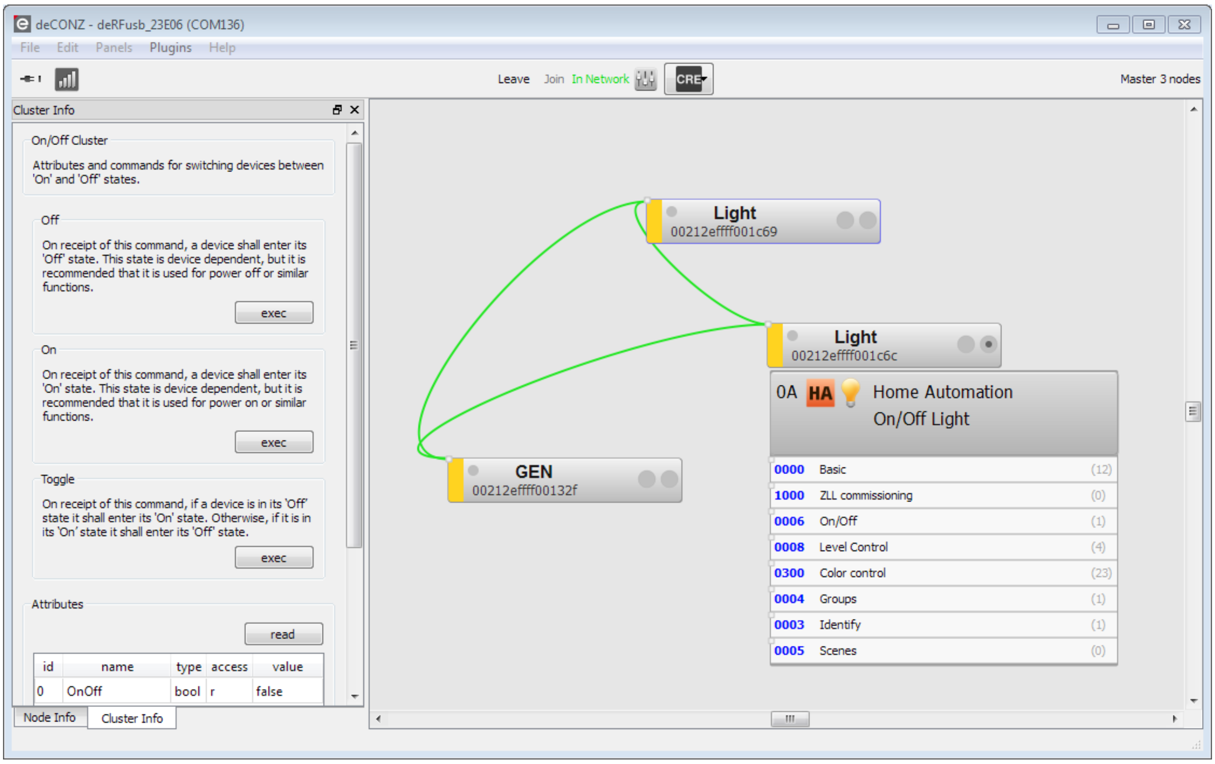
dresden elektronik

deCONZ — Serial Protocol



On/Off Cluster

Attributes and commands for switching devices between 'On' and 'Off' states.

Off

On receipt of this command, a device shall enter its 'Off' state. This state is device dependent, but it is recommended that it is used for power off or similar functions.

exec

On

On receipt of this command, a device shall enter its 'On' state. This state is device dependent, but it is recommended that it is used for power on or similar functions.

exec

Toggle

On receipt of this command, if a device is in its 'Off' state it shall enter its 'On' state. Otherwise, if it is in its 'On' state it shall enter its 'Off' state.

exec

Attributes

read

id	name	type	access	value
0	OnOff	bool	r	false

0000	Basic	(12)
1000	ZLL commissioning	(0)
0006	On/Off	(1)
0008	Level Control	(4)
0300	Color control	(23)
0004	Groups	(1)
0003	Identify	(1)
0005	Scenes	(0)

Document Version V1.14
2019-05-25

Table of contents

Document history

Date	Version	Description
2017-01-15	1.00	Initial version
2017-08-04	1.10	- Documented missing CRC16 - Corrected 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	- Read received data request: add flag to return only short addresses as source address. Since firmware 0x261b0500.
2018-09-19	1.12	- Correct ‘Read/Write Parameter’ frame length - ‘Read Received Data Request’ add flag to include last hop address in response - Document parameter ‘Protocol Version’
2019-04-11	1.13	- 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	- Document query firmware version command ‘VERSION’ - Document parameter ‘Watchdog TTL’ - Add ConBee II to supported devices

Abbreviations

Abbreviation	Description
APS	Application Support
CRE	Control Automatic Discovery
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
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; beside them and other public profiles Zigbee PRO provides the possibility to easily develop special purpose applications.

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. The 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, the deCONZ application can generate a full functional graphical user interface for each device and any application.

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 and ConBee II are Zigbee capable radio USB dongles that turn any PC or MAC with a free USB port into a Zigbee gateway. The RaspBee is an attachment for the Raspberry Pi that uses the Raspberry Pi’s GPIO pins. Before you can use the deCONZ application you have to set up your device and install all required software. A detailed description for this is available for ConBee[1] and RaspBee[2].

ConBee and ConBee II

Raspberry Pi with RaspBee

[1] https://phoscon.de/conbee

[2] https://phoscon.de/raspbee

2.2 Supported Operating Systems

- Microsoft Windows 7, 8, 8.1 and 10
- Canonical Ubuntu Linux 16.04 and 18.04
- Raspberry Pi Raspbian Jessie and Stretch
- Apple Mac OS X 10.11

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 http://www.zigbee.org website (registration required). The very basic specification needed to apply this protocol is the Zigbee PRO Specification.

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 detailed 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

Before running a device inside a network it has to be integrated; at first it has to get connected to the host PC and then it has to be configured to be able to join the network. On Windows, Linux PC or Mac you can use the ConBee or ConBee II USB dongle. On Rasberry Pi you can also use the RasBee shield.

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

Table 1: Status Codes

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

Table 2: Network States

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

Table 3: Commands

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 (0x0D)
U8	Sequence number	0–255
U8	Reserved	Shall be set to 0
U16	Frame length	5

Table 4: Format of Read Firmware Version Request

5.1.2 Read Firmware Version Response

Type	Field	Value
U8	Command ID	VERSION (0x0D)
U8	Sequence number	Same as request
U8	Status	SUCCESS
U16	Frame length	9
U32	Version	Example: 0x26330500, where the bytes represent: 0x26 — Major version 0x33 — Minor version 0x05 — Platform 0x00 — Reserved Platform: 0x05 — ConBee and RaspBee (AVR) 0x07 — ConBee II (ARM/R21)

Table 5: Format of Read Firmware Version Response

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	R
0x07	NWK Address	U16	0x0000–0xFFFE	R
0x08	NWK Extended PANID	U64	0x0000000000000000–0xFFFFFFFFFFFFFFFF	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	0x0000000000000000–0xFFFFFFFFFFFFFFFF	RW
0x0E	Trust Center Address	U64	0x0000000000000000–0xFFFFFFFFFFFFFFFF	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
0x18	Network Key	U8 [16]	Encryption key to secure network traffic	RW
0x1C	Current Channel	U8	11–26	R
0x22	Protocol Version	U16	Version of the implemented protocol	R
0x24	NWK Update ID	U8	0–255	RW
0x26	Watchdog TTL	U32	Watchdog timeout in seconds. Must be reset by the application periodically (since protocol version 0x0108)	RW

Table 6: Parameters

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 (0x0A)
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

Table 7: Format of Read Parameter Request

6.1.2 Read Parameter 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

Table 8: Format of Read Parameter Response

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

6.2 Write Configuration

6.2.1 Write Parameter 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 Table 6: Parameters
Variable	Parameter	The parameter

Table 9: Format of Write Parameter Request

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 Table 6: Parameters

Table 10: Format of Write Parameter Response

7. Control Network State

7.1 Reading Network State

7.1.1 Device State Request

Type	Field	Value
U8	Command ID	DEVICE_STATE (0x07)
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

Type	Field	Value

U8	Command ID	DEVICE_STATE (0x07)
U8	Sequence number	Same as request
U8	Status	SUCCESS
U16	Frame length	8
U8	Reserved	0000 0011 — Network state 0000 0100 — APSDE-DATA.confirm flag (0x04) 0000 1000 — APSDE-DATA.indication flag (0x08) 0001 0000 — Configuration changed flag (0x10) 0010 0000 — APSDE-DATA.request free slots flag (0x20)
U8	Reserved	Shall be ignored
U8	Reserved	Shall be ignored

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.

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.

Type	Field	Value
U8	Command ID	CHANGE_NETWORK_STATE (0x08)
U8	Sequence number	0–255
U8	Reserved	Shall be set to 0
U16	Frame length	6
U8	Network state	NET_CONNECTED (0x02)

Table 11: Format of Create or Join Network Request

7.2.2 Create or Join Network Response

Type	Field	Value
U8	Command ID	CHANGE_NETWORK_STATE (0x08)
U8	Sequence number	Same as request
U8	Status	SUCCESS or ERROR

U16	Frame length	6
U8	Network state	NET_CONNECTED (0x02)

Table 12: Format of Create or Join Network Response

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.

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 (0x08)
U8	Sequence number	0–255
U8	Reserved	Shall be set to 0
U16	Frame length	6
U8	Network State	NET_OFFLINE (0x00)

Table 13: Format of Leave Network Request

7.3.2 Leave Network Response

Type	Field	Value
U8	Command ID	CHANGE_NETWORK_STATE (0x08)
U8	Sequence number	Same as request
U8	Status	SUCCESS or ERROR
U16	Frame length	6
U8	Network state	NET_CONNECTED (0x02)

Table 14: Format of Leave Network Response

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 (0x0E)
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 (0x04) 0000 1000 — APSDE-DATA.indication flag (0x08) 0001 0000 — Configuration changed flag (0x10) 0010 0000 — APSDE-DATA.request free slots flag (0x20)
U8	Reserved	Shall be ignored

Table 15: Format of Unsolicited Device State Command

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.

7.4.2 Read Received Data Request

Type	Field	Value
U8	Command ID	APS_DATA_INDICATION (0x17)
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 0x01 — always return source address as 16-bit short address 0x02 — put last hop address after ASDU in first two reserved bytes (since protocol version 0x0108) 0x04 — include 16-bit and 64-bit source address (since protocol version 0x010B) source address mode becomes 0x04

Table 16: Format of the Read Received Data Request

7.4.3 Read Received Data Response

Type	Field	Value
U8	Command ID	APS_DATA_INDICATION (0x17)
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 (0x04) 0000 1000 — APSDE-DATA.indication flag (0x08) 0001 0000 — Configuration changed flag (0x10) 0010 0000 — APSDE-DATA.request free slots flag (0x20)
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
U8	Source address mode	0x02 — NWK address 0x03 — IEEE address 0x04 — NWK and IEEE address (since protocol version 0x010B)
*U16	16-bit source short address	Only included if source address mode is 0x02 or 0x04
*U64	64-bit source extended address	Only included if source address mode is 0x03 or 0x04
U8	Source endpoint	0–255
U16	Profile ID	0x0000–0xFFFF
U16	Cluster ID	0x0000–0xFFFF
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]

Table 17: Format of the Read Received Data Response

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.

Table 18: Format of the Enqueue Send Data Request

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 (0x12)
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 (0x04) 0000 1000 — APSDE-DATA.indication flag (0x08) 0001 0000 — Configuration changed flag (0x10) 0010 0000 — APSDE-DATA.request free slots flag (0x20)
U8	Request ID	Same as request

Table 19: Format of the Enqueue Send Data Response

7.5.3 Query Send Data State Request

Type	Field	Value
U8	Command ID	APS_DATA_CONFIRM (0x04)
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 (0x04)
U8	Sequence number	0–255
U8	Reserved	Shall be set to 0
U16	Frame length	7
U16	Payload length	11 — For destination address mode 0x01 12 — For destination address mode 0x02 18 — For destination address mode 0x03
U8	Device state	0000 0011 — Network state 0000 0100 — APSDE-DATA.confirm flag (0x04) 0000 1000 — APSDE-DATA.indication flag (0x08) 0001 0000 — Configuration changed flag (0x10) 0010 0000 — APSDE-DATA.request free slots flag (0x20)
U8	Request ID	To match this confirmation to a certain request
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
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

Phone +49 351 - 31850 0

Fax +49 351 - 31850 10
Email wireless@dresden-elektronik.de

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.