ZBOSS NCP: ZBOSS NCP Serial Protocol

Version: 1.5

Prepared by DSR Corporation for ZOI.

ZBOSS Zigbee software protocol stack ZBOSS NCP Serial protocol description document.

DSR Corporation 165 S. Union Blvd, Suite 670 Lakewood, CO 80228, USA Office: +1.720.962.9525 | Fax: +1.720.962.6625 www.dsr-corporation.com | www.ioticity.solutions

1. Introduction

1.1. Executive Summary

That document describes a serial protocol for ZBOSS Network Coprocessor product (NCP).

1.2. Revision History

- v1.0 -- 07/29/2020 (MT) -- Created first version of the protocol description
- v1.1 -- 10/06/2020 (MT) -- Updated list of NCP Protocol commands
- v1.2 -- 02/20/2021 (DD) -- Updated list of NCP Protocol commands
- v1.3 -- 03/25/2022 (ES) -- Updated list of NCP Protocol commands, correction after review
- v1.4 -- 06/16/2022 (ES) -- Updated copyrights
- v1.5 -- 05/12/2023 (EE) -- Updated copyrights

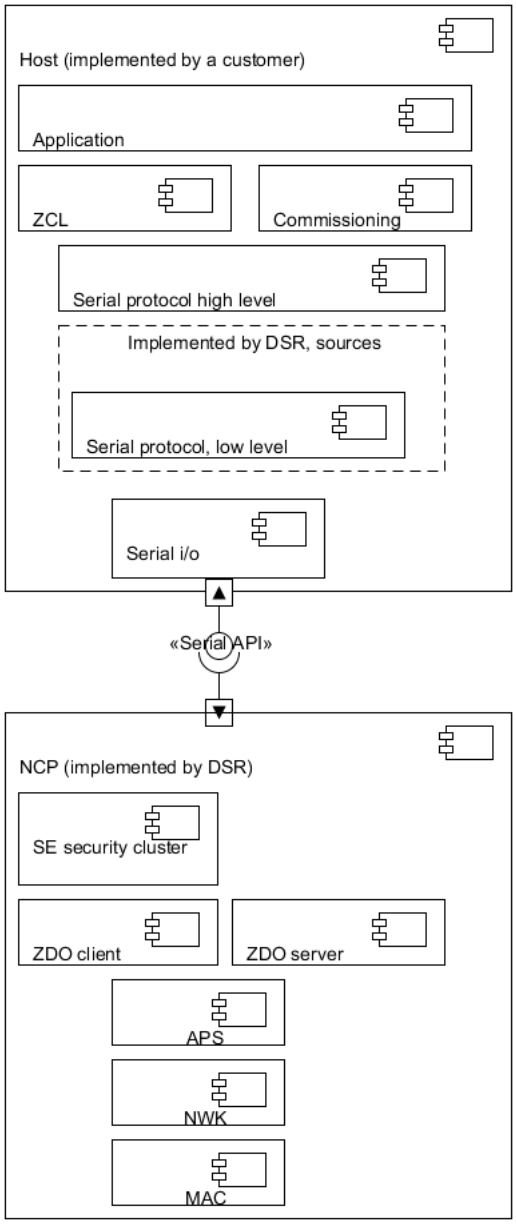
1.3. Table of Contents

2. Overall Description

ZBOSS NCP Serial Protocol (ZNSP) defined in this document is a protocol used by a host application to interact with ZBOSS ZigBee PRO stack running on a ZigBee Platform. ZNSP messages are sent between a host and the Zigbee platform over h/w serial interface, either UART or SPI.

2.1. ZBOSS NCP Architecture

ZNCP solution architecture is described on the picture below.



Key features:

- Serial protocol is the only official API of ZBOSS NCP SP.
- Full featured Zigbee PRO r22 platform is provided:
  - AF
  - ZDO client & server
  - SE security protocol (KE cluster and lower layers)
  - Full functioning APS, NWK and MAC layers.
- Host application implements:
  - Application logic itself
  - Zigbee Clusters
  - SE Commissioning logic
  - High level of the serial protocol
  - Low level i/o for the host (SPI, etc.)
- As a part of ZNCP, "Serial protocol, low level" is provided: routines to support low level communication protocol (not including i/o routines)

3. Communication protocol

Table of updated commands:

Removed:

Command	Description	Call id
AF_SUBGHZ_SUSPEND_IND	Sub-GHz Cluster Received Suspend ZCL Messages Command	0x0105

Communication protocol has 2 levels: lower layer protocol and high-level protocol. High level protocol is the only official NCP API. Communication protocol is same for any serial transport (SPI, UART, USB etc).

3.1. Common types and definitions

NCP API protocols (both levels) are binary protocols. All integer numbers are in Little Endian format.

3.1.1. Elementary types

Type name	Size, bytes	Type
uint8_t	1	unsigned byte
uint16_t	2	unsigned 16-bit integer
uint32_t	4	unsigned 32-but integer
ieee_addr_t	8	uint8_t array of size 8
u8arr_t	variable	array of uint8_t
u16arr_t	variable	array of uint16_t
u8bm_t	1	bitmap of size 1 byte
u16bm_t	2	bitmap of size 2 bytes
u32bm_t	4	bitmap of size 4 bytes
ubit_t	n/8	bitfield

3.2. Low-level protocol

Low level transport protocol is available for the NCP used and DSR copyrighted. Its main features are:

- Resistance to data loss/garbage by using CRC, ACKs and retransmits
- Resistance to TX/TX conflicts in half-duplex environment
- Compatibility with existing ZBOSS debug serial protocols
- Minimal overhead
- Packet fragmentation

Low level protocol packet consists of mandatory header and optional body.

3.2.1. Low level protocol header format

Octets: 2	2	1	1	1
Signature	Packet len	Packet type	Packet flags	Header CRC

3.2.2. Fields of Low level protocol header

Field	Type	Size, bytes	Meaning
Signature	u8arr_t	2	Fixed value 0xde 0xad
Packet len	uint16_t	2	Length of the packet with a header but without a signature
Packet type	uint8_t	1	== 6 for ZBOSS NCP API HL
Packet flags	u8bm_t	1	Flags
Header CRC	uint8_t	1	CRC8 of 4 bytes started from Packet len

If packet len field == 5 (header size), packet body is absent, else packet body is present. Packet body length == packet len - 5.

Description of used CRC8: width=8 poly=0x4d init=0xff refin=true refout=true xorout=0xff check=0xd8 name="CRC-8/KOOP". See this article for details.

3.2.3. Header flags format

Bits: 0	1	2-3	4-5	6	7

Bits: 0	1	2-3	4-5	6	7
isACK	should retransmit	Packet#	Ack#	First fragment	Last fragment

3.2.4. Header flags fields

Field	Size, bits	Meaning
isACK	1	If this flag is set, that packet has ACK or NACK info
should retransmit	1	If this flag is set, this is request for retransmit (NACK)
Packet#	2	Sequential number of packet modulo 4, value 0 is reserved
ACK#	2	Sequential number of ACKed packet modulo 4
First fragment	1	Mark the first fragment of a packet
Last fragment	1	Mark the last fragment of a packet

Packet# value 0 is used for NCP boot indication packet only. Packet numbers 1-3 are used during normal work. So packets numbers sequence is 1 - 2 - 3 - 1 ...

3.2.5. Packet body

Octets: 2	variable
Body CRC	Data

3.2.6. Packet body fields

Field	Type	Size, bytes	Meaning
Body CRC	uint16_t	2	CRC16 of Data field
Data	u8arr_t	variable	Higher level protocol payload

Description of used CRC16: width=16 poly=0x1021 init=0x0000 refin=true refout=true xorout=0x0000 check=0x2189 residue=0x0000 name="CRC-16/KERMIT".

It is supposed that packet may be lost or broken. Every side is able to re-synchronize (skipping broken packet) in that case using Signature and Header CRC fields. Every side must verify body contents using body CRC field. Every side must acknowledge received packet. If no ACK received after TX, the packet must be retransmitted after a timeout. Packet # is used to detect and drop duplicates. In the current version of LL protocol implementation ACK is a separate packet (header without a body), but protocol itself allows including ACK information into data packets in the future.

Typical packets sequence (without retransmissions) is:

- Request (isACK bit not set, should retransmit bit not set).
- ACK (isACK bit set, should retransmit bit not set, packet# is equal to the Request packet #).

To force resending from the peer, ACK is sent with "should retransmit" flag set to 1 (NACK). Resending may be required if packet is corrupted (check it using header and body CRC) or packet # is not as expected.

Packet body is absent if packet length == packet header size; body CRC is also absent in that case.

When passing packet up to High level protocol level, LL protocol removes body CRC.

3.2.7. Packet fragmentation

Low level protocol enables packet fragmentation/reassembly. This feature is useful for big packets that cannot fit into one transport frame. For fragmentation purposes each ACK has information about a fragment such as either the fragment is first or last or both simultaneously. In the last case it means a packet consists of only one fragment. When LL reassembles entire packet it will be passed to user level. Low level is responsible for delivery correct fragments in a right order.

3.3. Hardware Interfaces

It is supposed to use SPI interface to connect NCP to the Host.

Interface lines to be used:

- MOSI
- MISO
- SCLK
- CS
- HOST_INT
- RESET

Host has following SPI settings:

- 8 bits per word
- 1Mbit bitrate

- CPOL = 0
- CPHA = 1
- Default MOSI value is 0

Maximum bit rates supported by NCP hardware, according to TI Simplelink SDK:

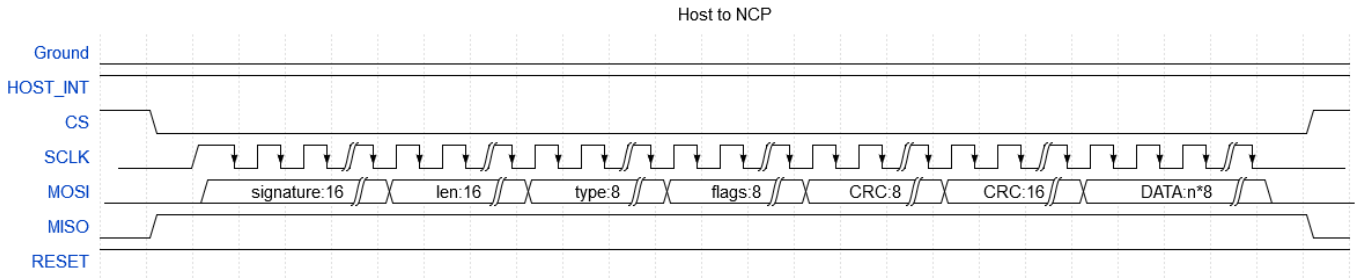
Device Family	Slave Max (MHz)	Master Max (MHz)
CC13XX/CC26XX	4 MHz	12 MHz

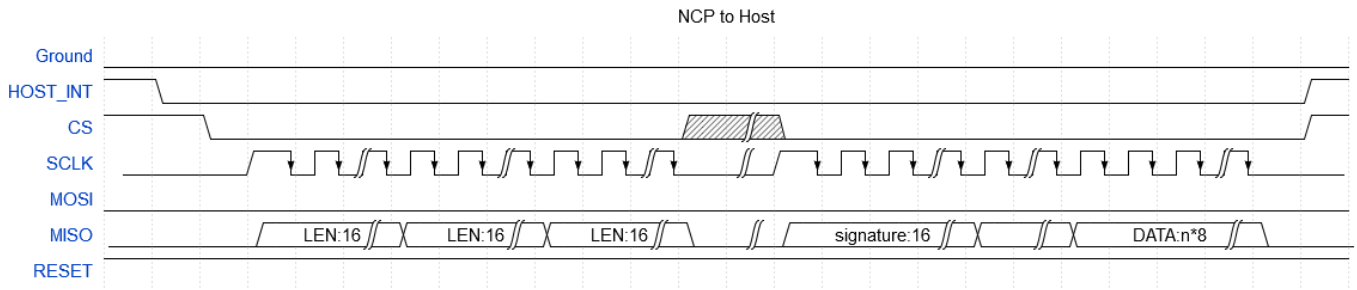
Please note that depending on the specific use case, the driver may not support the hardware's maximum bit rate.

HOST_INT pin is set to PUSH/PULL mode: (PIN_GPIO_OUTPUT_EN | PIN_GPIO_HIGH | PIN_PUSHPULL | PIN_DRVSTR_MED) - configuration from TI SDK.

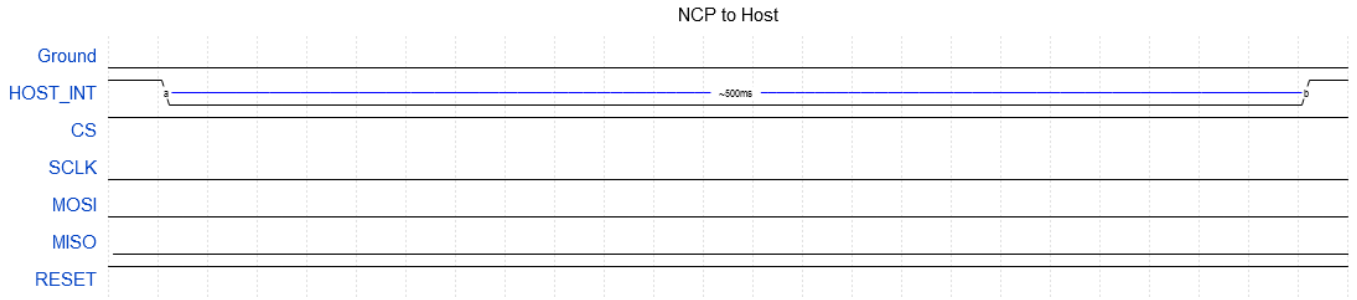
Every packet, including ACK packet, can be send in Host-NCP or NCP-Host direction.

Timing diagrams of hardware interface between Host and NCP is described on the pictures below.





While sending a packet to Host, NCP activates HOST_INT pin to make Host start reading the packet through SPI. For situations when Host doesn't react on HOST_INT pin activation there is a timeout. When the timeout is reached, NCP considers that connection with Host is broken and gets ERROR state with disabled SPI transport to let power saving algorithms start to work. For that moment the timeout is about 500ms. During that time NCP has HOST_INT pin activated and SPI transport ready for communication with Host.



3.3.1. Rx/tx procedures and tx/tx conflict resolution

Low level protocol over SPI supposes that Host-NCP communication is half-duplex: either Host or NCP can send a packet at a given time, but not both simultaneously. Since SPI interface communicates in master-slave model, then Host, being a master, needs to know how many clocks shall to be generated.

When Host wants to send a packet it asserts CS and then starts clocking sending data over MOSI. Slave receives first fixed-size packet header then packet payload. SPI enables CS and sends the packet keeping CS enabled. When transmission complete, SPI disables CS line.

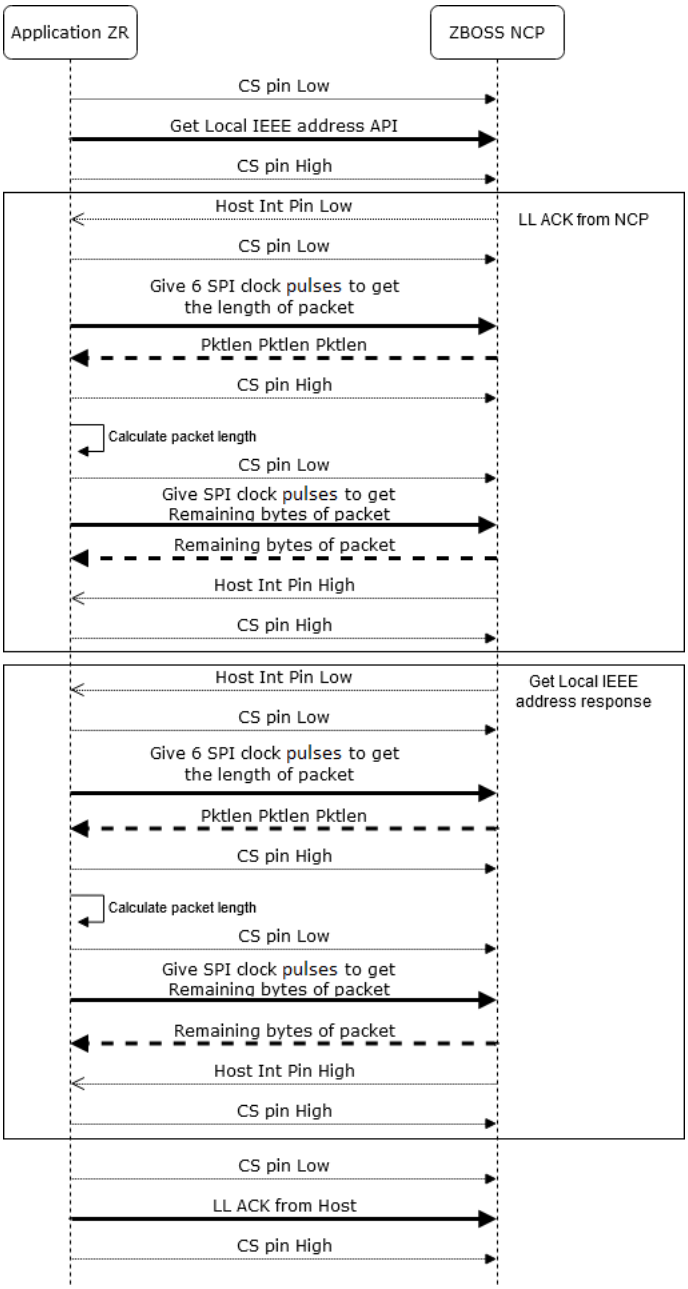
When NCP wants to send a packet, it has to report the length of the packet to the Host to let a proper number of clocks be generated. For this purpose, sequence below is used. NCP sends packet to Host according the sequence:

- NCP prepares a packet by adding a preamble before the packet. The preamble has length of six bytes, each couple of bytes equal to length of the packet.
- NCP enables HOST_INT line.
- Host is interrupted by the HOST_INT line and configures SPI for reading six bytes.
- SPI enables the CS line and reads six bytes from NCP, then disables the CS line.

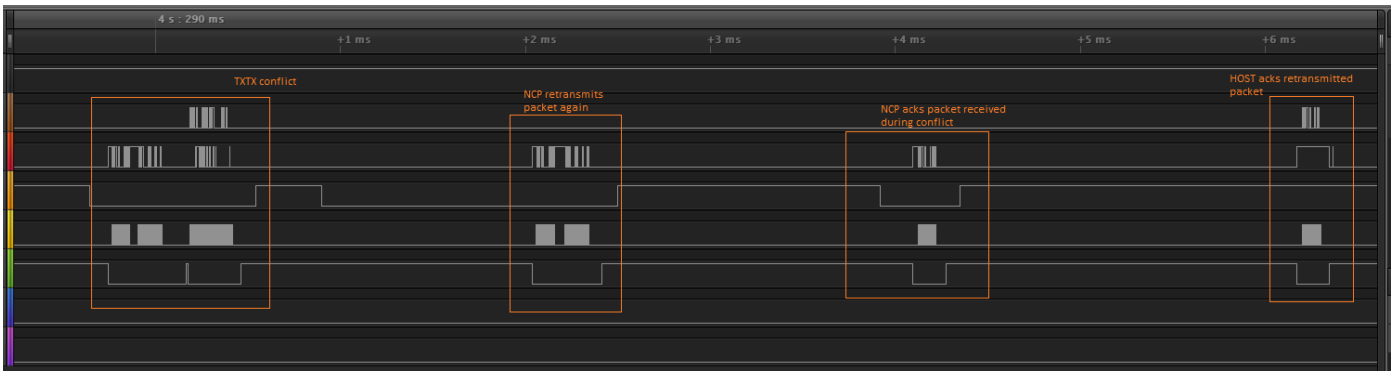
- After receiving the preamble, Host calculates length of the expected packet with majority function. This allows to correct errors on the SPI hardware interface.
- Host configures SPI for receiving the expected amount of bytes, that was computed on the previous step.
- SPI enables the CS line and reads the packet from NCP, then disables the CS line.
- Host passes the received packet to Low Level protocol.

Due to this algorithm the Host is able to receive unsolicited packets with unknown length from the NCP.

The example of basic Low Level communication is shown below



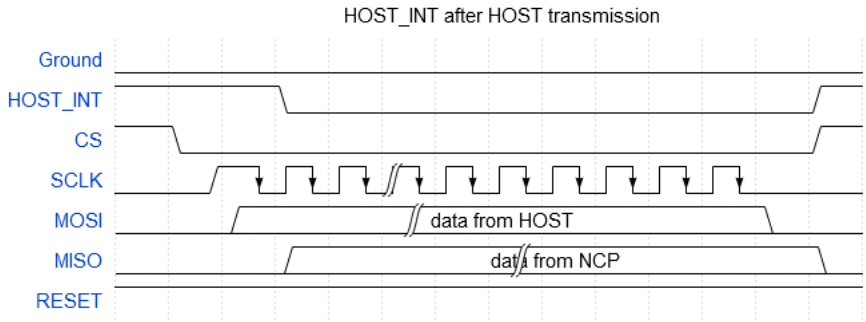
It is possible that NCP asserts HOST_INT just before Host makes a decision to send to NCP and starts transmission. In that case when Host asserts CS and starts clocking those both sides suppose their roles as "transmitter", i.e. not ready to receive data. Data sent in both directions would be lost in that situation. ACKs would not be sent, so data would be retransmitted in both directions. To prevent waiting for such a long time (for Acknowlegment) there was TX/TX conflict resolution procedure introduced. NCP always receives a packet from Host. When transmission transfer is over, NCP compares received data length with the sent one and if there is a mismatch NCP considers that conflict occurred. NCP takes the received data packet and passes it to command parser and notifies Low Level about a failed transmission, so Low Level can retransmit the packet immediately. At the same time received packet is processed and ready for acknowledgement.



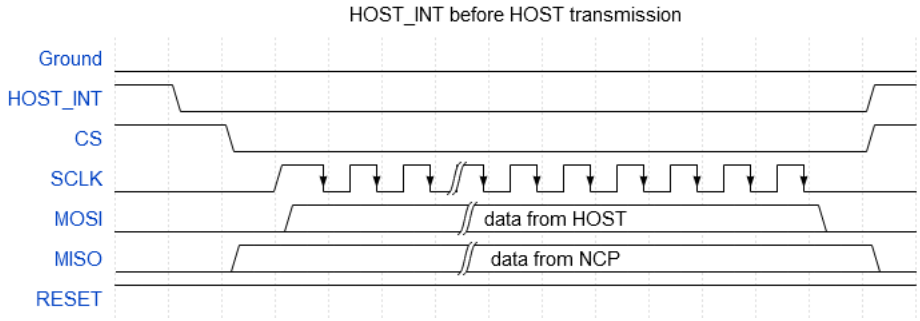
For normal cases, diagrams from chapter 3.3 can be referred to, when NCP wants to send something it asserts HOST_INT pin and waits for SPI clocks to start sending data. Host handles an interruption from HOST_INT pin and generates SPI signals to read data from NCP.

For abnormal cases, when HOST starts transmitting its packet and NCP asserts HOST_INT line to transmit its own packet, TX/TX conflict resolution procedure is performed.

If HOST_INT is asserted after HOST started transmitting its packet, both packets transfer might be failed, because NCP might have I/O transfer uninitialized by that time when HOST started TX. But, it's mostly theoretical case, because the initialization takes pretty short time about hundreds of microseconds and NCP checks the CS line state before any transmission and usually the packets are transferred and processed according to TX/TX conflict resolution procedure.



If HOST_INT is asserted before HOST starts its transmission, TX/TX conflict resolution takes place. NCP will receive HOST packet, because NCP I/O continues receiving until CS line is deactivated, check received packet, recognize conflict, pass RX packet to LL algorithm and retransmit its failed TX packet immediately.



3.3.2. Power management

If Host is a sleeping ZED, it is necessary to allow NCP to go to low power mode when there is no activity over Zigbee and SPI line.

ZBOSS NCP on a sleeping ZED handles radio sleep internally. In Zigbee a sleepy ZED can receive data only in response to Poll packet, so ZBOSS has enough information to put NCP asleep automatically. During the sleep the NCP is put to Standby mode. TI CC1352 in that mode consumes 1uA.

Sleepy ZED NCP can be woken up by the Host or by its internal logic (APS retransmit etc).

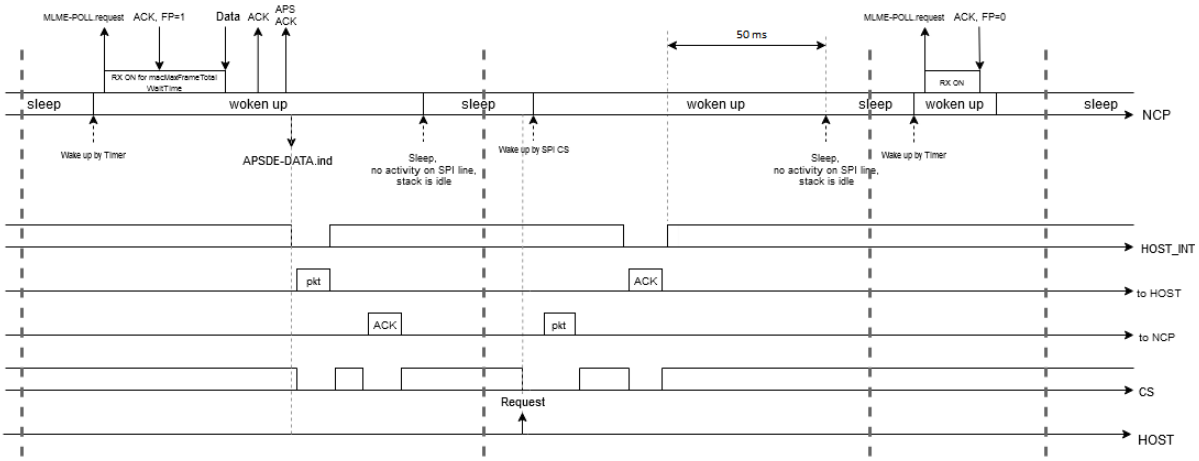
Host is not aware of the NCP sleep. NCP sleep is fully invisible for the Host. Packet loss is handled by low-level ZBOSS NCP protocol. Host behaves as if NCP never sleeps: it just sends a packet to NCP when it is necessary according to the Host's logic, and retransmits on receiving NACK or after ACK timeout expired.

NCP reuses CS line to act as a wake-up line. When NCP is going to sleep, CS line of SPI interface is reprogrammed to GPIO mode, so MCU can be woken by asserting CS.

There are several internal routines in NCP ZBOSS stack which can wake up NCP periodically to update internal state or perform some procedures:

- zb_mac_duty_cycle_periodic - wakes up every ~5min for duty cycle monitoring for SubGHz
- ed_timeout_resp - wakes up to postpone ED Keep Alive response for macResponseWaitTime as per 3.6.10.3 End Device Keepalive
- zdo_poll_parent - wakes up to schedule parent poll

From Host point of view NCP can be woken by sending any packet to it, because any packets sent from Host asserts CS line. There is no special "wake-up" packet. Instead Host just sends a next packet to NCP.



Host introduces some delay between CS line activation and starting packet transmission. During that time NCP finishes its wake-up (if occurred) and preparation for transfer and is able to receive the packet from Host.

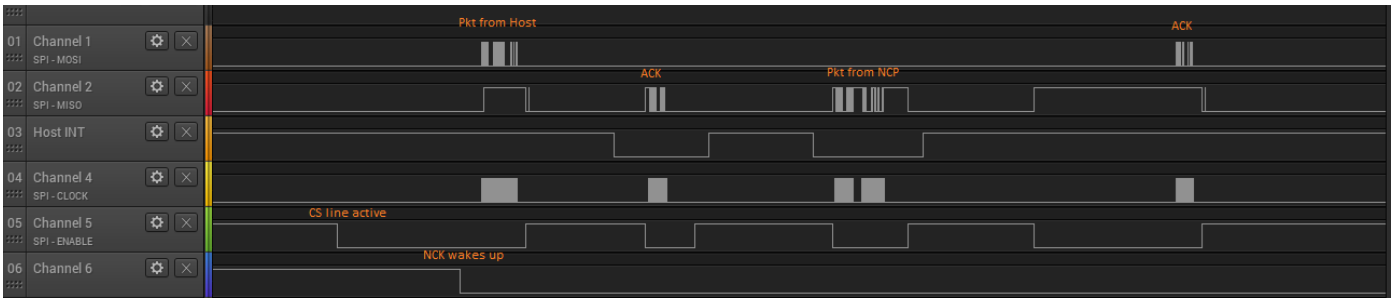


Figure shows a case when NCP awakens by “CS line active”, configures SPI driver and is ready for receiving by the time when Host sends packet - “NCP wakes up” signal. NCP receives packet, sends “ACK”, then responds and Host answers with “ACK” to the response. Disadvantage of the method is necessity to introduce a delay before each packet for Host to NCP direction. The delay is comparable with a whole packet length and affects time for packet transmission. On the other hand, it provides stable approach to awakening NCP without excessive transfers and allows maximum packet length to be transferred to a sleeping NCP without retransmission.

3.3.3. Reset management

Host is able to manage Reset pin of NCP. Note, that Reset is active low, and has duration of 1 us minimum, according to TI CC1352 datasheet.

Timing diagram of Reset line is on the picture below.

When NCP is re-set with pin RESET or another type of external reset sources it sends NCP_RESET_IND with reset reason. If NCP is re-set by software due to NCP_RESET request for example, it answers with NCP_RESET response.

See Device Reset Indication with reset source for available reset reasons.

3.4. High level protocol basics

3.4.1. ZBOSS NCP API subsets

There are the following main API subsets in NCP API:

1. NCP configuration API
2. AF API
3. ZDO API
4. APS API
5. NWK Management API
6. Security API

3.4.2. NCP API high-level protocol message types

High level protocol supports 3 types of messages:

1. Request (from Host to NCP)
2. Confirm: status and/or result of processing Request (NCP to Host)

3. Indication: some information from NCP to Host not initiated by a Request from the Host.

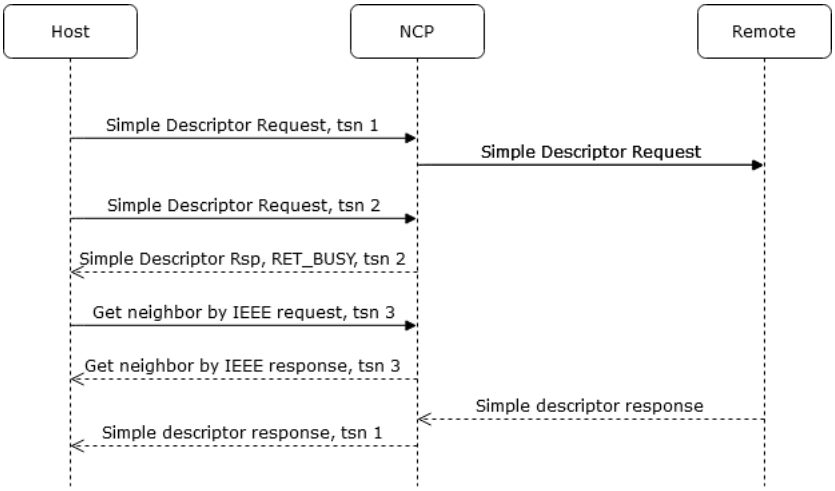
3.4.2.1. Blocking and non-blocking calls

Note that Host not always needs to wait for Request to complete before it can send another Request. For instance, Host can have more than one APSDE_DATA_REQ ongoing.

In many cases Requests still must be sent consequently, in other words when a Response for the previous Request is received, a new Request can be sent. That cases explicitly described as "blocking" operations.

If Host attempts to send to NCP the next "blocking" operation while NCP still have another "blocking" operation ongoing, NCP responds with BUSY error code.

The example demonstrates that ongoing Simple Descriptor Request doesn't let another blocking request to be processed, but non-blocking requests as Get neighbor by IEEE can be handled.



Timeout referred below means maximum time for processing request inside stack including communication with remote node if required, but without HOST-NCP transport overhead, which can be introduced by rate, Host/NCP latency and similar delays.

Blocking API	Timeout
Set Zigbee PAN ID	1ms
Add or Update Simple Descriptor for Endpoint	1ms
ZDO NWK Address Request	5s, 12s for sleepy ZED
ZDO IEEE Address Request	5s, 12s for sleepy ZED
ZDO Power Descriptor Request	5s, 12s for sleepy ZED
ZDO Node Descriptor Request	5s, 12s for sleepy ZED
ZDO Simple Descriptor Request	5s, 12s for sleepy ZED
ZDO Active Endpoint Request	5s, 12s for sleepy ZED
ZDO Bind Request	5s, 12s for sleepy ZED
ZDO Unbind Request	5s, 12s for sleepy ZED
ZDO Management Leave Request	5s, 12s for sleepy ZED
ZDO Management Permit Joining Request	5s, 12s for sleepy ZED
ZDO Management Rejoin Request	Depends on Channel list
NWK Formation	Depends on Scan Duration and Channel list
NWK Discovery	Depends on Scan Duration and Channel list
Join/Rejoin Network	Depends on Scan Duration and Channel list
NWK Permit Joining	1ms
Start Certificate-based Key Establishment	Ephemeral data generation time + 5 seconds or 10 seconds * CryptoSuite numbers if TCLK doesn't answer
Start Partner Link Key Establishment	5s
Send single raw packet	Packet transmission time
Send single poll	Success time 2ms, fail 32ms

3.4.3. High level protocol headers

High level protocol is encapsulated into low level protocol. High level protocol is binary. All integer values are in little endian. High level protocol packet has fixed-size header and variable-size portions.

3.4.3.1. High level Packet contents

Common Header	Specific headers	Parameters	Data
(Mandatory, fixed size)	Optional, depends on API group	(Optional)	(Optional)

3.4.3.2. Common Header structure

Octets: 1	1	2
Version	Control	Call/evt id

3.4.3.3. Common Header fields

Field	Type	Size, bytes	Meaning
Version	uint8_t	1	Protocol version. Currently 0.
Packet type	uint8_t	1	0 - request, 1 - response, 2 - indication
Call/evt id	uint16_t	2	Request or Indication ID

3.4.3.4. Requests packet header

Octets: 4	1
Common header	TSN

3.4.3.5. Requests packet header fields

Field	Type	Size, bytes	Meaning
TSN	uint8_t	1	Request transaction sequence number, 0 - 254

3.4.3.6. TSN

TSN (Transaction Sequence Number) is used to correlate Response with Request. Host side must define unique TSN for every call that is to be processed by NCP.

TSN value 255 (0xff) is reserved and must not be used by Host. It is used by NCP for unsolicited Responses (currently the only existing Unsolicited response is NCP_RESET).

Note that for some ZDO calls (like broadcast Match descriptor req) more than one response can be returned to one request, so all responses will have same TSN.

3.4.3.7. Response packet header

Octets: 4	1	1	1
Common header	TSN	Status category	Status code

3.4.3.8. Response packet fields

Field	Type	Size, bytes	Meaning
TSN	uint8_t	1	TSN same as in corresponding Request
Status category	uint8_t	1	Status category code
Status code	uint8_t	1	Status code inside category

Host application must analyze status code before attempting to analyze call-specific response data fields. For most responses if status is 0 (OK), response contains response header only.

3.4.3.9. Indication packet header

Octets: 4
Common header

Indications have no specific header.

3.4.4. Status codes

Status codes divided by categories.

3.4.4.1. Status codes categories

Category name	Category code	Meaning
GENERIC	0	Internal ZBOSS status codes
MAC	2	MAC layer status codes. Defined in table 78 in MAC 2006 specification
NWK	3	NWK layer status codes. Defined in 3.7 NWK Layer Status Values of Zigbee standard
APS	4	APS layer status codes. Defined in 2.2.9 APS Sub-Layer Status Values of Zigbee standard
ZDO	5	ZDO status codes. Defined in ZigBee Specification 2.4.5 - ZDP Enumeration Description
CBKE	6	CBKE status codes. Defined in 3.4.4.2. Generic ZBOSS Status codes

3.4.4.2. Generic ZBOSS Status codes

Status name	Status Value
OK	0
ERROR	1
BLOCKED	2
EXIT	3
BUSY	4
EOF	5
OUT_OF_RANGE	6
EMPTY	7
CANCELLED	8
INVALID_PARAMETER_1	10
INVALID_PARAMETER_2	11
INVALID_PARAMETER_3	12
INVALID_PARAMETER_4	13
INVALID_PARAMETER_5	14
INVALID_PARAMETER_6	15
INVALID_PARAMETER_7	16
INVALID_PARAMETER_8	17
INVALID_PARAMETER_9	18
INVALID_PARAMETER_10	19
INVALID_PARAMETER_11_OR_MORE	20
PENDING	21
NO_MEMORY	22
INVALID_PARAMETER	23
OPERATION_FAILED	24
BUFFER_TOO_SMALL	25
END_OF_LIST	26
ALREADY_EXISTS	27
NOT_FOUND	28
OVERFLOW	29
TIMEOUT	30
NOT_IMPLEMENTED	31
NO_RESOURCES	32
UNINITIALIZED	33

Status name	Status Value
NO_SERVER	34
INVALID_STATE	35
CONNECTION_FAILED	37
CONNECTION_LOST	38
UNAUTHORIZED	40
CONFLICT	41
INVALID_FORMAT	42
NO_MATCH	43
PROTOCOL_ERROR	44
VERSION	45
MALFORMED_ADDRESS	46
COULD_NOT_READ_FILE	47
FILE_NOT_FOUND	48
DIRECTORY_NOT_FOUND	49
CONVERSION_ERROR	50
INCOMPATIBLE_TYPES	51
FILE_CORRUPTED	56
PAGE_NOT_FOUND	57
ILLEGAL_REQUEST	62
INVALID_GROUP	64
TABLE_FULL	65
IGNORE	69
AGAIN	70
DEVICE_NOT_FOUND	71
OBSOLETE	72

3.4.4.3. CBKE ZBOSS Status codes

Status name	Status Value	Description
UNKNOWN_ISSUER	1	The Issuer field within the key establishment partner's certificate is unknown to the sending device
BAD_KEY_CONFIRM	2	The device could not confirm that it shares the same key with the corresponding device
BAD_MESSAGE	3	The device received a bad message from the corresponding device
NO_RESOURCES	4	The device does not currently have the internal resources necessary to perform key establishment
UNSUPPORTED_SUITE	5	The device does not support the specified key establishment suite in the partner's Initiate Key Establishment message
INVALID_CERTIFICATE	6	The received certificate specifies a type, curve, hash, or other parameter that is either unsupported by the device or invalid
NO_KE_EP	7	Non-standard ZBOSS extension: SE KE endpoint not found

3.5. High level protocol interface

All HL protocol calls are divided by categories. Every category has its own call id codes diapason. interval between categories is 0x100 == 256.

The format of the response packet varies depending on the returned status:

- If status category is GENERIC (0) and status code is OK (0) in the common response header, then the request has completed successfully and a full set of fields (if any) is included into the received packet.
- Any other values mean the request was not successful and only the common response header is present in the received packet.

Application must check the status of the response before accessing any other fields in the packet.

3.5.1. NCP configuration API

This category of the API provides general configuration facilities of the NCP.

Call	code	Implemented?
GET_MODULE_VERSION	0x0001	Yes
NCP_RESET	0x0002	Yes
GET_ZIGBEE_ROLE	0x0004	Yes
SET_ZIGBEE_ROLE	0x0005	Yes
GET_ZIGBEE_CHANNEL_MASK	0x0006	Yes
SET_ZIGBEE_CHANNEL_MASK	0x0007	Yes
GET_ZIGBEE_CHANNEL	0x0008	Yes
GET_PAN_ID	0x0009	Yes
SET_PAN_ID	0x000a	Yes
GET_LOCAL_IEEE_ADDR	0x000b	Yes
SET_LOCAL_IEEE_ADDR	0x000c	Yes
GET_TX_POWER	0x0010	Yes
SET_TX_POWER	0x0011	Yes
GET_RX_ON_WHEN_IDLE	0x0012	Yes
SET_RX_ON_WHEN_IDLE	0x0013	Yes
GET_JOINED	0x0014	Yes
GET_AUTHENTICATED	0x0015	Yes
GET_ED_TIMEOUT	0x0016	Yes
SET_ED_TIMEOUT	0x0017	Yes
SET_NWK_KEY	0x001b	Yes
GET_NWK_KEYS	0x001e	Yes
GET_APS_KEY_BY_IEEE	0x001f	Yes
GET_PARENT_ADDRESS	0x0022	Yes
GET_EXTENDED_PAN_ID	0x0023	Yes
GET_COORDINATOR_VERSION	0x0024	Yes
GET_SHORT_ADDRESS	0x0025	Yes
GET_TRUST_CENTER_ADDRESS	0x0026	Yes
NCP_RESET_IND	0x002b	Yes
NVRAM_WRITE	0x002e	Yes
NVRAM_READ	0x002f	Yes
NVRAM_ERASE	0x0030	Yes
NVRAM_CLEAR	0x0031	Yes
SET_TC_POLICY	0x0032	Yes
SET_EXTENDED_PAN_ID	0x0033	Yes
SET_MAX_CHILDREN	0x0034	Yes
GET_MAX_CHILDREN	0x0035	Yes

3.5.1.1. Get NCP Module Version

GET_MODULE_VERSION

Command id	Parameters	Data	Blocking?	Description
0x0001	No	No	No	Requests firmware, stack and protocol versions from NCP

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7	4	4	4
Common response header	FW Version	Stack Version	Protocol Version

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
FW Version	uint32_t	4	NCP module firmware version - SDK version
Stack Version	uint32_t	4	NCP module stack version - xx.yy.zz.cc version format, where xx - major, yy - minor, zz - revision, cc - commit number
Protocol Version	uint32_t	4	NCP module protocol version - the same as for Serial protocol document version

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success

Major part is increased each time when an external API is changed without keeping back compatibility. Minor part is increased each time when: an internal API is changed keeping backward compatibility or new module/subsystem/tool is introduced or refactored. Revision is increased for each commit in master branch. Commit number - sequential number of a commit for development branch.

3.5.1.2. NCP Module Reset

NCP_RESET

Command id	Parameters	Data	Blocking?	Description
0x0002	Yes	No	No	Force NCP module reboot

NOTE: this is module soft reboot. The command is a software analog of the module reset using RESET HW line.

The 'Erase NVRAM' option is for debugging purposes only, it erases ALL NVRAM contents including NWK packet counter.

The 'Factory reset' option means cleaning of NCP module internal state and removing from NVRAM all information except NWK packets counter. After Factory reset NCP is factory new and not commissioned into Zigbee network. APS counter (1 byte) is not stored in NVRAM according to specification. It is checked only for duplicate rejection and can have any value after NCP reboot. After Factory reset, APS and NWK security counters value are incremented by 1024. APS security counter must be increased for each packet, but it's not checked for holes between packets. Storing NVRAM frequency depends on packet sending rate, for now APS and NWK counters are saved after being incremented by 1024.

The 'Locking debug capabilities' option blocks NCP from reading the NWK and APS keys through GET_NWK_KEYS, GET_APS_KEY_BY_IEEE, SECUR_GET_KEY, WRITE_NVRAM_RESERVED memory access and any debugging routines via JTAG/bootloader and key broadcasting. An only way to unlock is to erase the entire flash content. Reset is required to apply CCFG settings for ROM bootloader and JTAG access settings. Lock status can be obtained via GET_LOCK_STATUS request.

Request packet format:

Octets: 5	1
Common request header	Options

Request parameters:

Req parameter name	Type	Length, bytes	Description
Options	uint8_t	1	Reset options: • 0 - No options • 1 - Erase NVRAM • 2 - Factory reset • 3 - Locking debug capabilities

Response packet format:

Octets: 7
Common response header

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success

Status category	Status code	Description
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	ERROR	Indicates a problem while accessing Flash when Locking debug capabilities option is set

Response has no parameters. Application must check the status of the response - it must be GENERIC.OK (category 0, code 0).

In case of error NCP sends Common Response Header with the status of the error and does not reboot. In case of success NCP reboots right after receiving the command without sending a response. After reboot the NCP sends Unsolicited NCP Reset response with the status of the operation (see next section).

Unsolicited NCP Reset response

When NCP rebooted due to Reset request, after booting successfully it sends to Host an unsolicited NCP reset response. That response has TSN 0xFF. If a reset is caused by some internal reasons Reset Indication is sent.

If application uses HW RESET line during its start, it must wait for Device Reset Indication from NCP before sending any other command to NCP.

If application got unsolicited NCP Reset response during normal work, it can handle NCP reboot.

3.5.1.3. Get Zigbee Role

GET_ZIGBEE_ROLE

Command id	Parameters	Data	Blocking?	Description
0x0004	No	No	No	Requests current Zigbee role of the local device

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7	1
Common response header	Zigbee role

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
Zigbee role	uint8_t	1	Zigbee role code: 0 - ZC, 1 - ZR, 2 - ZED, 3 - NONE

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success

Zigbee role NONE means that no role has been configured.

3.5.1.4. Set Zigbee Role

SET_ZIGBEE_ROLE

Command id	Parameters	Data	Blocking?	Description
0x0005	Yes	No	No	Set Zigbee role of the local device

Request packet format:

Octets: 5	1
Common request header	Zigbee role

Request parameters:

Req parameter name	Type	Length, bytes	Description
Zigbee role	uint8_t	1	Zigbee role code: 0 - ZC, 1 - ZR, 2 - ZED

Response packet format:

Octets: 7

Octets: 7
Common response header

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	INVALID_PARAMETER	Parameter passed with a request is invalid
GENERIC	NOT_IMPLEMENTED	Selected ZB role is not implemented

It is impossible to change NCP role after it has already been commissioned. NCP returns error status in that case.

Note that ZC role is not supported. NOT_IMPLEMENTED will be returned in this case. Default role value is 3 - NONE.

3.5.1.5. Get Zigbee Channel Mask

GET_ZIGBEE_CHANNEL_MASK

Command id	Parameters	Data	Blocking?	Description
0x0006	No	No	No	Get Zigbee channels page and mask of the local device

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7	1	variable
Common response header	Channel List Len	Channel List

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
Channel List Len	uint8_t	1	Number of entries in the following Channel List array.
Channel List	ChannelListEntry array	Channel List Len * ChannelListEntry size	Array of ChannelListEntry structures.

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success

ChannelListEntry Structure

Octets: 1	4
Channel Page	Channels Mask

Member Name	Type	Length, bytes	Description
Channel page	uint8_t	1	Channel page (0 if 2.4GHz)
Channels Mask	u32bm_t	4	Channels bitmask

For 2.4GHz-only NCP build Channel List has only one entry (Channel List Len == 1), Channel page is 0.

3.5.1.6. Set Zigbee Channel Mask

SET_ZIGBEE_CHANNEL_MASK

Command id	Parameters	Data	Blocking?	Description
0x0007	Yes	No	No	Set Zigbee channels mask for a given page at the local device

Request packet format:

Octets: 5	1	4

Octets: 5	1	4
Common request header	page	mask

Request parameters:

Req parameter name	Type	Length, bytes	Description
page	uint8_t	1	Channel page number
mask	u32bm_t	4	Channel mask

Response packet format:

Octets: 7
Common response header

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	INVALID_PARAMETER	Parameter passed with a request is invalid
GENERIC	NOT_IMPLEMENTED	Incorrect combination for ZB role and a band (for e.g. ZR and SubGHz)

Channel mask is applied only for automatic rejoin performed by NCP. The only case when automatic rejoin allowed is processing Leave request with Rejoin bit set. If Host does not know Coordinator channel mask it must set all channels to 1 to be able to find the network after channel changing. Default values are all zeros. Channel pages and masks can be preconfigured using production configuration.

3.5.1.7. Get Zigbee Current Channel

GET_ZIGBEE_CHANNEL

Command id	Parameters	Data	Blocking?	Description
0x0008	No	No	No	Requests current Zigbee channel

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7	1	1
Common response header	Page	Channel

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
Page	uint8_t	1	Channel page number
Channel	uint8_t	1	Channel number

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success

Channel is not a channel mask but a number of channel (for e.g. 11-26 for 2.4). If channel is not set yet (directly or indirectly), then the returned value for the Page and the Channel is 0xff.

3.5.1.8. Get Current Short PAN ID

GET_PAN_ID

Command id	Parameters	Data	Blocking?	Description
0x0009	No	No	No	Requests current short PAN ID

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7	2
Common response header	PAN ID

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
PAN ID	uint16_t	2	Short PAN ID

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success

If PAN ID is not set yet (directly or indirectly), then the returned value is 0xffff.

3.5.1.9. Set Zigbee PAN ID

SET_PAN_ID

Command id	Parameters	Data	Blocking?	Description
0x000a	Yes	No	Yes	Set short PAN ID

Request packet format:

Octets: 5	2
Common request header	PAN ID

Request parameters:

Req parameter name	Type	Length, bytes	Description
PAN ID	uint16_t	2	Short PAN ID

Response packet format:

Octets: 7
Common response header

NOTE: Setting short PAN ID is not always allowed. An example of a forbidden case is setting PAN ID when the device has been already joined, NCP will return status code INVALID_STATE.

3.5.1.10. Get Local IEEE Address

GET_LOCAL_IEEE_ADDR

Command id	Parameters	Data	Blocking?	Description
0x000b	Yes	No	No	Requests local IEEE address

Request packet format:

Octets: 5	1
Common request header	MAC Interface Num

Request parameters:

Req parameter name	Type	Length, bytes	Description
MAC Interface Num	uint8_t	1	The number of MAC interface. Must be always 0.

Response packet format:

Octets: 7	1	8
Common response header	MAC Interface Num	IEEE address

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
MAC Interface Num	uint8_t	1	The number of MAC interface. Always equals for 0.
IEEE address	ieee_addr_t	8	Local IEEE address

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success

3.5.1.11. Set Local IEEE Address

SET_LOCAL_IEEE_ADDR

Command id	Parameters	Data	Blocking?	Description
0x000c	Yes	No	No	Set local IEEE address

Request packet format:

Octets: 5	1	8
Common request header	MAC Interface Num	IEEE address

Request parameters:

Req parameter name	Type	Length, bytes	Description
MAC Interface Num	uint8_t	1	The number of MAC interface. Always equals for 0.
IEEE address	ieee_addr_t	8	Local IEEE address

Response packet format:

Octets: 7
Common response header

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	INVALID_PARAMETER	Parameter passed with a request is invalid

NOTE: This command will override the IEEE address value from production configuration.

3.5.1.12. Get Transmit Power

GET_TX_POWER

Command id	Parameters	Data	Blocking?	Description
0x0010	Yes	No	No	Get Transmit Power

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7	1
Common response header	Current TX power

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
Current TX power	int8_t	1	Current transmit power in dBm

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_STATE	TX power is not initialized yet

3.5.1.13. Set Transmit Power

SET_TX_POWER

Command id	Parameters	Data	Blocking?	Description
0x0011	Yes	No	No	Set Transmit Power

NOTE: In Sub-GHz bands set of power values might be different.

Request packet format:

Octets: 5	1
Common request header	Required TX power

Request parameters:

Req parameter name	Type	Length, bytes	Description
Required TX power	int8_t	1	Required transmitter power. The value depends on specific transmitter implementation.

Response packet format:

Octets: 7	1
Common response header	Resultant TX power

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
Resultant TX power	int8_t	1	If the required TX power is valid, returns the same value. If the required value is invalid, the first valid value which is lower-or-equal than the given one will be set.

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length

3.5.1.14. Get Rx On When Idle PIB Attribute

GET_RX_ON_WHEN_IDLE

Command id	Parameters	Data	Blocking?	Description
0x0012	No	No	No	Requests RxOnWhenIdle PIB attribute

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7	1
Common response header	Rx On When Idle

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
Rx On When Idle	uint8_t	1	Rx On When Idle PIB attribute value: • 0 - FALSE • 1 - TRUE

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success

3.5.1.15. Set Rx On When Idle PIB Attribute

SET_RX_ON_WHEN_IDLE

Command id	Parameters	Data	Blocking?	Description
0x0013	Yes	No	No	Sets Rx On When Idle PIB attribute

Request packet format:

Octets: 5	1
Common request header	Rx On When Idle

Request parameters:

Req parameter name	Type	Length, bytes	Description
Rx On When Idle	uint8_t	1	Rx On When Idle PIB attribute value: • 0 - FALSE • 1 - TRUE

Response packet format:

Octets: 7
Common response header

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success

For NCP Zigbee End Device RX is always OFF when idle.

3.5.1.16. Get Join Status

GET_JOINED

Command id	Parameters	Data	Blocking?	Description
0x0014	No	No	No	Requests current join status of the device

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7	1
Common response header	Joined

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
Joined	uint8_t	1	• bit 0: Device is joined 0 - false, 1 - true • bit 1: Parent is lost 0 - false, 1 - true

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success

Parent lost status is set to True when NCP detects Parent loss through sending corresponding packets.

3.5.1.17. Get Authentication Status

GET_AUTHENTICATED

Command id	Parameters	Data	Blocking?	Description
0x0015	No	No	No	Requests current authentication status of the device

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7	1
Common response header	Authenticated

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
Authenticated	uint8_t	1	• 0 - Device is not authenticated. • 1 - Device is authenticated.

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success

Authenticated status means that a device has received Network key. Once NCP is authenticated (received NWK key), ZBOSS will drop all frames which are not NWK encrypted.

3.5.1.18. Get End Device timeout

GET_ED_TIMEOUT

Command id	Parameters	Data	Blocking?	Description
0x0016	No	No	No	Requests current End Device timeout

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7	1
Common response header	Timeout

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
Timeout	uint8_t	1	Timeout, Index from ED timeout table

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success

ED timeout table

Timeout #	Value, seconds
0	10
1	120
2	240
3	480
4	920
5	1920
6	3840
7	7680
8	15360
9	30720
10	61440
11	122880
12	245760
13	491520
14	983040

3.5.1.19. Set End Device timeout

SET_ED_TIMEOUT

Command id	Parameters	Data	Blocking?	Description
0x0017	Yes	No	No	Sets End Device timeout

Request packet format:

Octets: 5	1
Common request header	Timeout

Request parameters:

Req parameter name	Type	Length, bytes	Description
Timeout	uint8_t	1	Timeout, Index from ED timeout table

Response packet format:

Octets: 7
Common response header

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	INVALID_PARAMETER	Means that ED timeout is greater than 1024 minutes (61440 sec)

Default End Device timeout is 8 see Index from ED timeout table.

3.5.1.20. Set NWK Key

SET_NWK_KEY

Command id	Parameters	Data	Blocking?	Description
0x001b	Yes	No	No	Set NWK Key

Request packet format:

Octets: 5	16	1

Octets: 5	16	1
Common request header	NWK Key	Key number

Request parameters:

Req parameter name	Type	Length, bytes	Description
NWK Key	u8arr_t	16	NWK Key
Key number	uint8_t	1	The number of NWK key.

Response packet format:

Octets: 7
Common response header

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length

NOTE: This command is meaningful only for ZC role. In case of other roles, the call would have no result.

3.5.1.21. Get NWK keys

GET_NWK_KEYS

Command id	Parameters	Data	Blocking?	Description
0x001e	No	No	No	Get list of NWK keys

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7	16	1	16	1	16	1
Common response header	NWK Key	Key number	NWK Key	Key number	NWK Key	Key number

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
NWK Key	u8arr_t	16	NWK Key
Key number	uint8_t	1	Number of NWK key.
NWK Key	u8arr_t	16	NWK Key
Key number	uint8_t	1	Number of NWK key.
NWK Key	u8arr_t	16	NWK Key
Key number	uint8_t	1	Number of NWK key.

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	OPERATION_FAILED	Chip is locked from reading keys

The first NWK key in the response is current network key used by NCP. Other keys can be either next key or previous key. They can be distinguished by comparing their number with a number of current key. If current key number is greater it is previous key, otherwise next one.

3.5.1.22. Get APS key by IEEE

GET_APS_KEY_BY_IEEE

Command id	Parameters	Data	Blocking?	Description
0x001f	Yes	No	No	Get APS key by IEEE

Request packet format:

Octets: 5	8
Common request header	IEEE address

Request parameters:

Req parameter name	Type	Length, bytes	Description
IEEE address	ieee_addr_t	8	IEEE address of remote device

Response packet format:

Octets: 7	16
Common response header	APS Key

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
APS Key	u8arr_t	16	APS Key

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	OPERATION_FAILED	Chip is locked from reading keys
GENERIC	NOT_FOUND	Key is not found for a given parameter

Returns current APS used for encryption APS messages between NCP and device with given IEEE address.

3.5.1.23. Get Parent address

GET_PARENT_ADDRESS

Command id	Parameters	Data	Blocking?	Description
0x0022	No	No	Yes	Get Parent short address

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7	2
Common response header	NWK parent address

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
NWK parent address	uint16_t	2	NWK Parent address

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success

Returns 0xFFFF if there is no parent.

3.5.1.24. Get Extended Pan ID

GET_EXTENDED_PAN_ID

Command id	Parameters	Data	Blocking?	Description
0x0023	No	No	Yes	Get Extended Pan ID

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7	8
Common response header	Extended Pan ID

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
Extended Pan ID	ieee_addr_t	8	Extended Pan ID

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success

Default values are all zeros.

3.5.1.25. Get Coordinator version

GET_COORDINATOR_VERSION

Command id	Parameters	Data	Blocking?	Description
0x0024	No	No	Yes	Get Coordinator version

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7	1
Common response header	Coordinator version

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
Coordinator version	uint8_t	1	Coordinator version

3.5.1.26. Get Short Address of the device

GET_SHORT_ADDRESS

Command id	Parameters	Data	Blocking?	Description
0x0025	No	No	Yes	Get Short Address of the device

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7	2
Common response header	NWK Address

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
NWK Address	uint16_t	2	NWK address of the device

3.5.1.27. Get Trust Center Address

GET_TRUST_CENTER_ADDRESS

Command id	Parameters	Data	Blocking?	Description
0x0026	No	No	Yes	Get Trust Center IEEE Address

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7	16
Common response header	Trust Center IEEE Address

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
Trust Center IEEE Address	ieee_addr_t	16	TC Address

3.5.1.28. Device Reset Indication with reset source

NCP_RESET_IND

Command id	Parameters	Data	Blocking?	Description
0x002b	No	No	No	Device Reset Indication with reset source

Indication packet format:

Octets: 4	1
Common HL header	Reset source

Indication parameters:

Ind parameter name	Type	Length, bytes	Description
Reset source	uint8_t	1	Reset source which triggered reset. • 0 - RESET_SRC_POWER_ON (TI RSTSRC_PWR_ON) • 1 - RESET_SRC_SW_RESET (TI RSTSRC_SYSRESET) • 2 - RESET_SRC_RESET_PIN (TI RSTSRC_PIN_RESET) • 3 - RESET_SRC_BROWN_OUT (TI RSTSRC_VDDS_LOSS or RSTSRC_VDDR_LOSS) • 4 - RESET_SRC_CLOCK_LOSS (TI RSTSRC_CLK_LOSS) • 5 - RESET_SRC_OTHER (TI RSTSRC_WAKEUP_FROM_SHUTDOWN or RSTSRC_WAKEUP_FROM_TCK_NOISE or RSTSRC_WARMRESET)

NCP considers RESET_SRC_SW_RESET as resetting caused by Reset request command (0x0002) and sends Reset response, all other types of reset, like power ON, reset through the pin, clock/power loss are considered as reset by hardware reasons and sends Reset indication (0x002b) with the reason. RESET_SRC_SW_RESET is reset caused by writing to a system register, considered as intentional reset done by NCP and after reboot Reset response will be sent. RESET_SRC_OTHER is unknown reset reason, which cannot be recognized.

3.5.1.29. Write NVRAM datasets

NVRAM_WRITE

Command id	Parameters	Data	Blocking?	Description
0x002e	Yes	Yes	Yes	Writes NVRAM datasets

Request packet format:

Octets: 5	1	variable

Octets: 5	1	variable
Common request header	Number of datasets	data

Request parameters:

Req parameter name	Type	Length, bytes	Description
Number of datasets	uint8_t	1	A number of datasets contained in this request
data	u8arr_t	data len	data bytes array

Response packet format:

Octets: 7
Common response header

3.5.1.30. Read NVRAM dataset

NVRAM_READ

Command id	Parameters	Data	Blocking?	Description
0x002f	Yes	Yes	Yes	Reads an NVRAM dataset

Request packet format:

Octets: 5	1
Common request header	Dataset type

Request parameters:

Req parameter name	Type	Length, bytes	Description
Dataset type	uint8_t	1	A dataset type to read

Response packet format:

Octets: 7	2	2	2	2	variable
Common response header	NVRAM version	Dataset type	Dataset version	Dataset length	data

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
NVRAM version	uint16_t	2	Current NVRAM version
Dataset type	uint16_t	2	Requested dataset type
Dataset version	uint16_t	2	Current dataset version
Dataset length	uint16_t	2	Length of the requested dataset
data	u8arr_t	data len	data bytes array

3.5.1.31. Erase NVRAM

NVRAM_ERASE

Command id	Parameters	Data	Blocking?	Description
0x0030	No	No	Yes	Erases all datasets in NVRAM

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7
Common response header

3.5.1.32. Clear NVRAM

NVRAM_CLEAR

Command id	Parameters	Data	Blocking?	Description
0x0031	No	No	Yes	Erases all datasets in NVRAM except ZB_NVRAM_RESERVED, ZB_IB_COUNTERS and application datasets

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7
Common response header

3.5.1.33. Set TC Policy

SET_TC_POLICY

Command id	Parameters	Data	Blocking?	Description
0x0032	Yes	No	Yes	Sets TC Policy

Request packet format:

Octets: 5	2	1
Common request header	Policy type	Policy value

Request parameters:

Req parameter name	Type	Length, bytes	Description
Policy type	uint16_t	2	A policy type to set
Policy value	uint8_t	1	A policy value to set

Response packet format:

Octets: 7
Common response header

Policy Types

- 0x0000 - TC Link Keys Required
- 0x0001 - IC Required
- 0x0002 - TC Rejoin Enabled
- 0x0003 - Ignore TC Rejoin
- 0x0004 - APS Insecure Join
- 0x0005 - Disable NWK MGMT Channel Update

3.5.1.34. Set extended PAN ID

SET_EXTENDED_PAN_ID

Command id	Parameters	Data	Blocking?	Description
0x0033	Yes	No	Yes	Sets an extended PAN ID

Request packet format:

Octets: 5	8
Common request header	Extended PAN ID

Request parameters:

Req parameter name	Type	Length, bytes	Description
Extended PAN ID	ieee_addr_t	8	An Extended PAN ID value to set

Response packet format:

Octets: 7
Common response header

3.5.1.35. Set maximum number of children

SET_MAX_CHILDREN

Command id	Parameters	Data	Blocking?	Description
0x0034	Yes	No	Yes	Sets the maximum number of children

Request packet format:

Octets: 5	1
Common request header	Number of children

Request parameters:

Req parameter name	Type	Length, bytes	Description
Number of children	uint8_t	1	The number of children to set as a maximum allowed

Response packet format:

Octets: 7
Common response header

3.5.1.36. Get maximum number of children

GET_MAX_CHILDREN

Command id	Parameters	Data	Blocking?	Description
0x0035	No	No	Yes	Gets the maximum number of children

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7	1
Common response header	Number of children

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
Number of children	uint8_t	1	The maximum number of children currently allowed

3.5.2. AF API

This category of the API provides an access to the Application Framework part hosted on the NCP side.

Call	code	Implemented?
AF_SET_SIMPLE_DESC	0x0101	Yes
AF_DEL_SIMPLE_DESC	0x0102	Yes
AF_SET_NODE_DESC	0x0103	Yes
AF_SET_POWER_DESC	0x0104	Yes

3.5.2.1. Add or Update Simple Descriptor for Endpoint

AF_SET_SIMPLE_DESC

Command id	Parameters	Data	Blocking?	Description
0x0101	Yes	No	Yes	Add or update Simple descriptor for a specified endpoint

Request packet format:

Octets: 5	1	2	2	1	1	1	var	var
Common request header	Endpoint	Profile ID	Device ID	Device Version	Input Cluster Count	Output Cluster Count	Input Cluster List	Output Cluster List

Request parameters:

Req parameter name	Type	Length, bytes	Description
Endpoint	uint8_t	1	Endpoint number to add/update simple descriptor for.
Profile ID	uint16_t	2	Profile ID.
Device ID	uint16_t	2	Device ID.
Device Version	uint8_t	1	Device version. See 13-0589-13-ZigBee-Application-Architecture.pdf p.14
Input Cluster Count	uint8_t	1	Count of Input clusters in the simple descriptor.
Output Cluster Count	uint8_t	1	Count of Output clusters in the simple descriptor.
Input Cluster List	u16arr_t	2 * Input Cluster Count	Array of Input cluster IDs.
Output Cluster List	u16arr_t	2 * Output Cluster Count	Array of Output cluster IDs.

Response packet format:

Octets: 7
Common response header

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	INVALID_PARAMETER_1	Parameter passed with a request at the first place is invalid
GENERIC	OVERFLOW	All existing descriptors are already set. Can't allocate more
GENERIC	NO_MEMORY	Can't allocate memory for a descriptor

A storage for 5 simple descriptor records is available. When the request is applied for the same endpoint its Simple descriptor will be updated. In case of NCP, ZBOSS processes Simple descriptor requests from remote devices internally and sends a response back. The Host shouldn't care about descriptor requests from remote device. Simple descriptor can be updated after joining through using the request. Zigbee Role has to be set to add or update Simple descriptor.

3.5.2.2. Delete Simple Descriptor for Endpoint

AF_DEL_SIMPLE_DESC

Command id	Parameters	Data	Blocking?	Description
0x0102	Yes	No	No	Delete Simple Descriptor for a specified endpoint

Request packet format:

Octets: 5	1
Common request header	Endpoint

Request parameters:

Req parameter name	Type	Length, bytes	Description
Endpoint	uint8_t	1	Endpoint number to delete simple descriptor for.

Response packet format:

Octets: 7
Common response header

Status codes:

Status category	Status code	Description

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	INVALID_PARAMETER	Invalid endpoint value
GENERIC	NOT_FOUND	Given endpoint doesn't exist

3.5.2.3. Set Node Descriptor

AF_SET_NODE_DESC

Command id	Parameters	Data	Blocking?	Description
0x0103	Yes	No	No	Set Node Descriptor

Request packet format:

Octets: 5	1	1	2
Common request header	Device Type	MAC Capabilities	Manufacturer Code

Request parameters:

Req parameter name	Type	Length, bytes	Description
Device Type	uint8_t	1	Device type: 0 - ZC, 1 - ZR, 2 - ZED
MAC Capabilities	uint8_t	1	MAC Capabilities bitfield
Manufacturer Code	uint16_t	2	Manufacturer code

Response packet format:

Octets: 7
Common response header

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	INVALID_PARAMETER_1	Parameter passed with a request at the first place is invalid
GENERIC	INVALID_PARAMETER_2	Parameter passed with a request at the second place is invalid

MAC Capabilities bitfield

Bit Number	Bit Name	Bit Description
0	Alternate PAN Coordinator	This field will always have a value of 0.
1	Device Type	1 - device is FFD, 0 - device is RFD.
2	Power Source	1 - device is receiving power from the alternating current mains, otherwise - 0.
3	Receive On When Idle	1 - device does not disable its receiver to save power during idle periods, otherwise - 0.
4	Reserved	This field will always have a value of 0.
5	Reserved	This field will always have a value of 0.
6	Security Capability	This field shall have a value of 0.
7	Allocate Address	1 - device wants the coordinator to allocate 16bit short address as a result of the association procedure, 0 - address is self-selected while using NWK rejoin command.

The Set Node Descriptor request is applicable for configuring Node descriptor parameters only. It doesn't affect NCP settings outside of the Node descriptor fields. ZBOSS initializes part of Node descriptor fields internally basing on its settings. Fields which can be updated using Set Node Descriptor request are following: Device type; Manufacturer code; MAC Capabilities - Power Source, Receive On When Idle, Allocate Address.Corresponding response to Node descriptor request from remote device will be updated.Default values for the fields are below:

Field name	Default value
Logical type	1 (Zigbee Router)
Complex descriptor available	0 (unavailable)
User descriptor available	0 (unavailable)
Reserved	0 (reserved)
APS flags	0 (not supported)
Frequency band	0x18 (if 2.4 + EU SubGHz)
MAC capability flags	0x8e (FFD, Main powered, RX ON, Allocate addr TRUE)
Manufacturer code	0x1234
Maximum buffer size	108
Maximum incoming transfer size	1628
Server mask	0x2c00 (r22)
Maximum outgoing transfer size	1628
Descriptor capability field	0

3.5.2.4. Set Power Descriptor

AF_SET_POWER_DESC

Command id	Parameters	Data	Blocking?	Description
0x0104	Yes	No	No	Set power descriptor for the device

Request packet format:

Octets: 5	1	1	1	1
Common request header	Current Power Mode	Available Power Sources	Current Power Source	Current Power Source Level

Request parameters:

Req parameter name	Type	Length, bytes	Description
Current Power Mode	uint8_t	1	Current power mode value
Available Power Sources	u8bm_t	1	Available power sources bits
Current Power Source	u8bm_t	1	Current power source bit (ONLY one source can be chosen from available power sources)
Current Power Source Level	uint8_t	1	Current power source level value

Response packet format:

Octets: 7
Common response header

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	INVALID_PARAMETER_1	Parameter passed with a request at the first place is invalid
GENERIC	INVALID_PARAMETER_2	Parameter passed with a request at the second place is invalid
GENERIC	INVALID_PARAMETER_3	Parameter passed with a request at the third place is invalid
GENERIC	INVALID_PARAMETER_4	Parameter passed with a request at the fourth place is invalid

Current Power Mode

The Current Power Mode field shall be set to one of the non-reserved values listed in the table below.

Value	Description
0	Receiver synchronized with the receiver on when idle subfield of the node descriptor

Value	Description
1	Receiver comes on periodically as defined by the node power descriptor
2	Receiver comes on when stimulated, for example, by Host side
3-255	Reserved

Power Source Bits

For each power source supported on this node, the corresponding bit of the available power sources field shall be set to 1. All other bits shall be set to 0.

Bit	Description
0	Constant (mains) power
1	Rechargeable battery
2	Disposable battery
3-7	Reserved (shall be 0)

Power Source Levels

Value	Charge Level
0	Critical
4	33%
8	66%
12	100%
All other values	Reserved

By default Power descriptor fields are set to 0. ZBOSS doesn't assign any special values for the descriptor. The Host must set the fields to update Power descriptor. Corresponding response to Power descriptor request from remote device will be updated.

3.5.3. ZDO API

This category of the API provides an access to the Zigbee Device Object hosted on the NCP side.

Call	code	Implemented?
ZDO_NWK_ADDR_REQ	0x0201	Yes
ZDO_IEEE_ADDR_REQ	0x0202	Yes
ZDO_POWER_DESC_REQ	0x0203	Yes
ZDO_NODE_DESC_REQ	0x0204	Yes
ZDO_SIMPLE_DESC_REQ	0x0205	Yes
ZDO_ACTIVE_EP_REQ	0x0206	Yes
ZDO_MATCH_DESC_REQ	0x0207	Yes
ZDO_BIND_REQ	0x0208	Yes
ZDO_UNBIND_REQ	0x0209	Yes
ZDO_MGMT_LEAVE_REQ	0x020a	Yes
ZDO_PERMIT_JOINING_REQ	0x020b	Yes
ZDO_DEV_ANNCE_IND	0x020c	Yes
ZDO_REJOIN	0x020d	Yes
ZDO_SYSTEM_SRV_DISCOVERY_REQ	0x020e	Yes
ZDO_MGMT_BIND_REQ	0x020f	Yes
ZDO_MGMT_LQI_REQ	0x0210	Yes
ZDO_MGMT_NWK_UPDATE_REQ	0x0211	Yes
ZDO_GET_STATS	0x0213	Yes
ZDO_DEV_AUTHORIZED_IND	0x0214	Yes
ZDO_DEV_UPDATE_IND	0x0215	Yes

Call	code	Implemented?
ZDO_SET_NODE_DESC_MANUF_CODE	0x0216	Yes

3.5.3.1. ZDO NWK Address Request

ZDO_NWK_ADDR_REQ

Command id	Parameters	Data	Blocking?	Description
0x0201	Yes	No	Yes	Request for a remote device NWK address

The destination addressing on this command shall be unicast or broadcast to all devices for which RxOnWhenIdle is TRUE.

Request packet format:

Octets: 5	2	8	1	1
Common request header	Dest NWK Address	IEEE Address of Interest	Request Type	Start Index

Request parameters:

Req parameter name	Type	Length, bytes	Description
Dest NWK Address	uint16_t	2	NWK address of the remote device to send request to.
IEEE Address of Interest	ieee_addr_t	8	IEEE address to be matched by the remote device.
Request Type	int8_t	1	Request type: 0 - Single device response, 1 - Extended response, 2-255 - reserved.
Start Index	int8_t	1	Starting index of the returned associated device list. Valid only if the Request type is Extended response.

Response packet format:

Octets: 7	8	2	1/0	1/0	2 * Num Assoc Dev
Common response header	Remote Dev IEEE Addr	Remote Dev NWK Addr	Num Assoc Dev	Start Index	Assoc Dev NWK Addr List

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
Remote Dev IEEE Addr	ieee_addr_t	8	IEEE address of the matched remote device.
Remote Dev NWK Addr	uint16_t	2	NWK address of the matched remote device.
Num Assoc Dev	uint8_t	1 or not present	Number of associated devices in the following address list. Present only if Request type parameter of the request is Extended response.
Start Index	uint8_t	1 or not present	Starting index of the returned associated device list. Present only if the Request type is Extended response and Num Assoc Dev is not 0.
Assoc Dev NWK Addr List	u16arr_t	2 * Num Assoc Dev	Variable-size array of NWK addresses of devices associated with the remote device. Present only if the Request type is Extended response and Num Assoc Dev is not 0.

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	INVALID_PARAMETER	Parameter passed with a request is invalid
GENERIC	BUSY	NCP is busy with processing of another blocking request
GENERIC	NO_MEMORY	There is no memory to process the request
GENERIC	ERROR	Internal ZBOSS error
ZDO	INV_REQUESTTYPE	Refer to ZB specification 2.4.5 ZDP Enumeration Description
ZDO	DEVICE_NOT_FOUND	Refer to ZB specification 2.4.5 ZDP Enumeration Description
ZDO	TIMEOUT	Refer to ZB specification 2.4.5 ZDP Enumeration Description

3.5.3.2. ZDO IEEE Address Request

ZDO_IEEE_ADDR_REQ

Command id	Parameters	Data	Blocking?	Description
0x0202	Yes	No	Yes	Request for a remote device IEEE address

Request packet format:

Octets: 5	2	2	1	1
Common request header	Dest NWK Address	NWK Address of Interest	Request Type	Start Index

Request parameters:

Req parameter name	Type	Length, bytes	Description
Dest NWK Address	uint16_t	2	NWK address of the remote device to send request to.
NWK Address of Interest	uint16_t	2	NWK address to be matched by the remote device.
Request Type	int8_t	1	Request type: 0 - Single device response, 1 - Extended response, 2-255 - reserved.
Start Index	int8_t	1	Starting index of the returned associated device list. Valid only if the Request type is Extended response.

Response packet format:

Octets: 7	8	2	1/0	1/0	2 * Num Assoc Dev
Common response header	Remote Dev IEEE Addr	Remote Dev NWK Addr	Num Assoc Dev	Start Index	Assoc Dev NWK Addr List

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
Remote Dev IEEE Addr	ieee_addr_t	8	IEEE address of the matched remote device.
Remote Dev NWK Addr	uint16_t	2	NWK address of the matched remote device.
Num Assoc Dev	uint8_t	1 or not present	Number of associated devices in the following address list. Present only if Request type parameter of the request is Extended response.
Start Index	uint8_t	1 or not present	Starting index of the returned associated device list. Present only if the Request type is Extended response and Num Assoc Dev is not 0.
Assoc Dev NWK Addr List	u16arr_t	2 * Num Assoc Dev	Variable-size array of NWK addresses of devices associated with the remote device. Present only if the Request type is Extended response and Num Assoc Dev is not 0.

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	INVALID_PARAMETER	Parameter passed with a request is invalid
GENERIC	BUSY	NCP is busy with processing of another blocking request
GENERIC	NO_MEMORY	There is no memory to process the request
GENERIC	ERROR	Internal ZBOSS error
ZDO	INV_REQUESTTYPE	Refer to ZB specification 2.4.5 ZDP Enumeration Description
ZDO	DEVICE_NOT_FOUND	Refer to ZB specification 2.4.5 ZDP Enumeration Description
ZDO	TIMEOUT	Refer to ZB specification 2.4.5 ZDP Enumeration Description

3.5.3.3. ZDO Power Descriptor Request

ZDO_POWER_DESC_REQ

Command id	Parameters	Data	Blocking?	Description
0x0203	Yes	No	Yes	Get the Power Descriptor from a remote device

Request packet format:

Octets: 5	2
Common request header	NWK Address of Interest

Request parameters:

Req parameter name	Type	Length, bytes	Description
NWK Address of Interest	uint16_t	2	NWK address to be matched by the remote device.

Response packet format:

Octets: 7	2	2
Common response header	Power Descriptor	NWK Address of source device

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
Power Descriptor	u16bm_t	2	Power Descriptor Bit Fields: • Bits 0-3 Current power mode value • Bits 4-7 Available power sources bits • Bits 8-11 Current power source bits (ONLY one source can be chosen from available power sources) • Bits 12-15 Current power source level value
NWK Address of source device	uint16_t	2	NWK address of source device.

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	INVALID_PARAMETER	Parameter passed with a request is invalid
GENERIC	BUSY	NCP is busy with processing of another blocking request
GENERIC	NO_MEMORY	There is no memory to process the request
GENERIC	ERROR	Internal ZBOSS error
ZDO	INV_REQUESTTYPE	Refer to ZB specification 2.4.5 ZDP Enumeration Description
ZDO	DEVICE_NOT_FOUND	Refer to ZB specification 2.4.5 ZDP Enumeration Description
ZDO	TIMEOUT	Refer to ZB specification 2.4.5 ZDP Enumeration Description
ZDO	NO_DESCRIPTOR	Refer to ZB specification 2.4.5 ZDP Enumeration Description

In case of NCP, ZBOSS processes Power descriptor requests from remote devices internally and sends a response back. The Host shouldn't care about descriptor requests from remote device.

3.5.3.4. ZDO Node Descriptor Request

ZDO_NODE_DESC_REQ

Command id	Parameters	Data	Blocking?	Description
0x0204	Yes	No	Yes	Get the Node Descriptor from a remote device

ZBOSS always sends Node descriptor Request unicast to the NWK Address Of Interest.

Request packet format:

Octets: 5	2
Common request header	NWKAddrOfInterest

Request parameters:

Req parameter name	Type	Length, bytes	Description
NWKAddrOfInterest	uint16_t	2	NWK address for the request

Response packet format:

Octets: 7	2	1	2	1	2	2	2	1	2
Common response header	Flags	MAC Capabilities	Manufacturer Code	Maximum Buffer Size	Maximum Incoming Transfer Size	Server Mask	Maximum Outgoing Transfer Size	Descriptor Capabilities	NWK Address of source device

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
Flags	u16bm_t	2	• bit0-2 - logical type: 0 - ZC, 1 - ZR, 2 - ZED • bit3 - complex desc available • bit4 - user desc available • bit8-10 - APS flags • bits12-15 - Frequency band
MAC Capabilities	uint8_t	1	MAC Capabilities bitfield
Manufacturer Code	uint16_t	2	manufacturer code that is allocated by the ZigBee Alliance, relating the manufacturer to the device
Maximum Buffer Size	uint8_t	1	maximum size, in octets, of the network sub-layer data unit (NSDU) for this node
Maximum Incoming Transfer Size	uint16_t	2	maximum size, in octets, of the application sub-layer data unit (ASDU) that can be transferred to this node in one single message transfer.
Server Mask	u16bm_t	2	• bit0 - Primary Trust Center • bit1 - Backup Trust Center • bit2 - Primary Binding Table Cache • bit3 - Backup Binding Table Cache • bit4 - Primary Discovery Cache • bit5 - Backup Discovery Cache • bit6 - Network Manager • bits9-15 - Stack Compliance Revision
Maximum Outgoing Transfer Size	uint16_t	2	maximum size, in octets, of the application sub-layer data unit (ASDU) that can be transferred from this node in one single message transfer
Descriptor Capabilities	u8bm_t	1	• bit0 - Extended Active Endpoint List Available • bit1 - Extended Simple Descriptor List Available
NWK Address of source device	uint16_t	2	NWK address of source device.

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	BUSY	NCP is busy with processing of another blocking request
GENERIC	NO_MEMORY	There is no memory to process the request
GENERIC	ERROR	Internal ZBOSS error
ZDO	INV_REQUESTTYPE	Refer to ZB specification 2.4.5 ZDP Enumeration Description
ZDO	DEVICE_NOT_FOUND	Refer to ZB specification 2.4.5 ZDP Enumeration Description
ZDO	TIMEOUT	Refer to ZB specification 2.4.5 ZDP Enumeration Description
ZDO	NO_DESCRIPTOR	Refer to ZB specification 2.4.5 ZDP Enumeration Description

In case of NCP, ZBOSS processes Node descriptor requests from remote devices internally and sends a response back. The Host shouldn't care about descriptor requests from remote device.

3.5.3.5. ZDO Simple Descriptor Request

ZDO_SIMPLE_DESC_REQ

Command id	Parameters	Data	Blocking?	Description
0x0205	Yes	No	Yes	Get the Simple Descriptor from a remote device

Request packet format:

Octets: 5	2	1
Common request header	NWK Address of Interest	Endpoint

Request parameters:

Req parameter name	Type	Length, bytes	Description
NWK Address of Interest	uint16_t	2	NWK address of the remote device.
Endpoint	uint8_t	1	Endpoint number.

Response packet format:

Octets: 7	1	2	2	1	1	1	variable	variable	2
Common response header	Endpoint	App Profile ID	App Device ID	App Device Version	App Input Cluster Count	App Output Cluster Count	App Input Cluster List	App Output Cluster List	NWK Address of source device

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
Endpoint	uint8_t	1	Endpoint number.
App Profile ID	uint16_t	2	Application profile ID.
App Device ID	uint16_t	2	Application device ID.
App Device Version	uint8_t	1	Application device version.
App Input Cluster Count	uint8_t	1	Count of Input clusters supported by the application.
App Output Cluster Count	uint8_t	1	Count of Output clusters supported by the application.
App Input Cluster List	u16arr_t	2 * App Input Cluster Count	Array of Input cluster IDs supported by the application. Omitted if the corresponding Cluster Count is 0.
App Output Cluster List	u16arr_t	2 * App Output Cluster Count	Array of Output cluster IDs supported by the application. Omitted if the corresponding Cluster Count is 0.
NWK Address of source device	uint16_t	2	NWK address of source device.

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	INVALID_PARAMETER_2	Parameter passed with a request at the second place is invalid
GENERIC	BUSY	NCP is busy with processing of another blocking request
GENERIC	NO_MEMORY	There is no memory to process the request
GENERIC	ERROR	Internal ZBOSS error
ZDO	INV_REQUESTTYPE	Refer to ZB specification 2.4.5 ZDP Enumeration Description
ZDO	DEVICE_NOT_FOUND	Refer to ZB specification 2.4.5 ZDP Enumeration Description
ZDO	TIMEOUT	Refer to ZB specification 2.4.5 ZDP Enumeration Description
ZDO	NO_DESCRIPTOR	Refer to ZB specification 2.4.5 ZDP Enumeration Description
ZDO	INVALID_EP	Refer to ZB specification 2.4.5 ZDP Enumeration Description
ZDO	NOT_ACTIVE	Refer to ZB specification 2.4.5 ZDP Enumeration Description

3.5.3.6. ZDO Active Endpoint Request

ZDO_ACTIVE_EP_REQ

Command id	Parameters	Data	Blocking?	Description
0x0206	Yes	No	Yes	Get a list of Active Endpoints from a remote device

Request packet format:

Octets: 5	2
Common request header	NWK Address of Interest

Request parameters:

Req parameter name	Type	Length, bytes	Description
NWK Address of Interest	uint16_t	2	NWK address of the remote device.

Response packet format:

Octets: 7	1	variable	2
Common response header	Endpoint Count	Endpoint List	NWK Address of source device

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
Endpoint Count	uint8_t	1	Count of Active Endpoints in the following list.
Endpoint List	u8arr_t	1 * Active Endpoint Count	List of Active Endpoints. Omitted if the Endpoint Count is 0.
NWK Address of source device	uint16_t	2	NWK address of source device.

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	BUSY	NCP is busy with processing of another blocking request
GENERIC	NO_MEMORY	There is no memory to process the request
GENERIC	ERROR	Internal ZBOSS error
ZDO	INV_REQUESTTYPE	Refer to ZB specification 2.4.5 ZDP Enumeration Description
ZDO	DEVICE_NOT_FOUND	Refer to ZB specification 2.4.5 ZDP Enumeration Description
ZDO	TIMEOUT	Refer to ZB specification 2.4.5 ZDP Enumeration Description
ZDO	NO_DESCRIPTOR	Refer to ZB specification 2.4.5 ZDP Enumeration Description

3.5.3.7. ZDO Match Descriptor Request

ZDO_MATCH_DESC_REQ

Command id	Parameters	Data	Blocking?	Description
0x0207	Yes	No	No	Send Match Descriptor request to a remote device

Request packet format:

Octets: 5	2	2	1	1	variable	variable
Common request header	NWK Address of Interest	Profile ID	Input Cluster Count	Output Cluster Count	Input Cluster List	Output Cluster List

Request parameters:

Req parameter name	Type	Length, bytes	Description
NWK Address of Interest	uint16_t	2	NWK address of the remote device.
Profile ID	uint16_t	2	ID of the profile of interest.
Input Cluster Count	uint8_t	1	Count of Input cluster IDs in the following list.
Output Cluster Count	uint8_t	1	Count of Output cluster IDs in the following list.
Input Cluster List	u16arr_t	2 * Input Cluster Count	Array of Input cluster IDs. Omitted if the corresponding Cluster Count is 0.
Output Cluster List	u16arr_t	2 * Output Cluster Count	Array of Output cluster IDs. Omitted if the corresponding Cluster Count is 0.

Response packet format:

Octets: 7	1	variable	2

Octets: 7	1	variable	2
Common response header	Endpoint Count	Endpoint List	NWK Address of source device

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
Endpoint Count	uint8_t	1	Count of endpoints in the following list.
Endpoint List	u8arr_t	1 * Match Endpoint Count	List of endpoints matched the requested parameters. Omitted if the Endpoint Count is 0.
NWK Address of source device	uint16_t	2	NWK address of source device.

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	BUSY	NCP is busy with processing of another blocking request
GENERIC	NO_MEMORY	There is no memory to process the request
GENERIC	ERROR	Internal ZBOSS error
ZDO	INV_REQUESTTYPE	Refer to ZB specification 2.4.5 ZDP Enumeration Description
ZDO	DEVICE_NOT_FOUND	Refer to ZB specification 2.4.5 ZDP Enumeration Description
ZDO	TIMEOUT	Refer to ZB specification 2.4.5 ZDP Enumeration Description
ZDO	NO_DESCRIPTOR	Refer to ZB specification 2.4.5 ZDP Enumeration Description

3.5.3.8. ZDO Bind Request

ZDO_BIND_REQ

Command id	Parameters	Data	Blocking?	Description
0x0208	Yes	No	Yes	Send Bind request to a remote device

Request packet format:

Octets: 5	2	8	1	2	1	8	1
Common request header	Target NWK Address	Src IEEE Address	Src Endpoint	Cluster ID	Dst Address Mode	Dst Address	Dst Endpoint

Request parameters:

Req parameter name	Type	Length, bytes	Description
Target NWK Address	uint16_t	2	NWK address of the remote device to send the request to.
Src IEEE Address	ieee_addr_t	8	IEEE address of the source device.
Src Endpoint	uint8_t	1	Source endpoint number.
Cluster ID	uint16_t	2	Cluster ID to bind.
Dst Address Mode	uint8_t	1	Destination Address Mode
Dst Address	ieee_addr_t or uint16_t	8	IEEE or NWK address of the destination device depending on the address mode specified. If NWK address is used, remaining bytes are ignored.
Dst Endpoint	uint8_t	1	Destination endpoint number. Shall be set to 0, if Destination Address Mode isn't 0x03.

Response packet format:

Octets: 7
Common response header

Status codes:

Status category	Status code	Description

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	INVALID_PARAMETER	Parameter passed with a request is invalid
GENERIC	BUSY	NCP is busy with processing of another blocking request
GENERIC	NO_MEMORY	There is no memory to process the request
GENERIC	ERROR	Internal ZBOSS error
ZDO	TIMEOUT	Refer to ZB specification 2.4.5 ZDP Enumeration Description
ZDO	NOT_SUPPORTED	Refer to ZB specification 2.4.5 ZDP Enumeration Description
ZDO	INVALID_EP	Refer to ZB specification 2.4.5 ZDP Enumeration Description
ZDO	TABLE_FULL	Refer to ZB specification 2.4.5 ZDP Enumeration Description
ZDO	NOT_AUTHORIZED	Refer to ZB specification 2.4.5 ZDP Enumeration Description

Destination Address Mode

- 0x00 - reserved
- 0x01 - 16-bit group address for Dst Address and Dst Endpoint not present
- 0x02 - reserved
- 0x03 - 64-bit extended address for Dst Address and Dst Endpoint present
- 0x04 - 0xff = reserved

3.5.3.9. ZDO Unbind Request

ZDO_UNBIND_REQ

Command id	Parameters	Data	Blocking?	Description
0x0209	Yes	No	Yes	Send Unbind request to a remote device

Request packet format:

Octets: 5	2	8	1	2	1	8	1
Common request header	Target NWK Address	Src IEEE Address	Src Endpoint	Cluster ID	Dst Address Mode	Dst IEEE/NWK Address	Dst Endpoint

Request parameters:

Req parameter name	Type	Length, bytes	Description
Target NWK Address	uint16_t	2	NWK address of the remote device to send the request to.
Src IEEE Address	ieee_addr_t	8	IEEE address of the source device.
Src Endpoint	uint8_t	1	Source endpoint number.
Cluster ID	uint16_t	2	Cluster ID to unbind.
Dst Address Mode	uint8_t	1	Destination Address Mode
Dst IEEE/NWK Address	ieee_addr_t or uint16_t	8	IEEE or NWK address of the destination device depending on address mode specified. If NWK address is used, remaining bytes are ignored.
Dst Endpoint	uint8_t	1	Destination endpoint number. Shall be set to 0, if Destination Address Mode isn't 0x03.

Response packet format:

Octets: 7
Common response header

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success

Status category	Status code	Description
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	INVALID_PARAMETER	Parameter passed with a request is invalid
GENERIC	BUSY	NCP is busy with processing of another blocking request
GENERIC	NO_MEMORY	There is no memory to process the request
GENERIC	ERROR	Internal ZBOSS error
ZDO	TIMEOUT	Refer to ZB specification 2.4.5 ZDP Enumeration Description
ZDO	NOT_SUPPORTED	Refer to ZB specification 2.4.5 ZDP Enumeration Description
ZDO	INVALID_EP	Refer to ZB specification 2.4.5 ZDP Enumeration Description
ZDO	NO_ENTRY	Refer to ZB specification 2.4.5 ZDP Enumeration Description
ZDO	NOT_AUTHORIZED	Refer to ZB specification 2.4.5 ZDP Enumeration Description

3.5.3.10. ZDO Management Leave Request

ZDO_MGMT_LEAVE_REQ

Command id	Parameters	Data	Blocking?	Description
0x020a	Yes	No	Yes	Request that a Remote Device leave the network

Request packet format:

Octets: 5	2	8	1
Common request header	dst_addr	Device address	leave_flags

Request parameters:

Req parameter name	Type	Length, bytes	Description
dst_addr	uint16_t	2	NWK address of the remote device to send the request to.
Device address	ieee_addr_t	8	IEEE address of the device to be removed from the network.
leave_flags	uint8_t	1	Leave flags bitfield

Response packet format:

Octets: 7
Common response header

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	BUSY	NCP is busy with processing of another blocking request
GENERIC	NO_MEMORY	There is no memory to process the request
GENERIC	ERROR	Internal ZBOSS error
ZDO	TIMEOUT	Refer to ZB specification 2.4.5 ZDP Enumeration Description
ZDO	NOT_SUPPORTED	Refer to ZB specification 2.4.5 ZDP Enumeration Description
ZDO	NOT_AUTHORIZED	Refer to ZB specification 2.4.5 ZDP Enumeration Description

Leave flags bitfield

Bit Number	Bit Name	Bit Description
0-5	Reserved	These fields shall have a value of 0.
6	Remove children	1 - if the device is also being asked to remove its child devices, if any, otherwise - 0.
7	Rejoin	1 - if the device being asked to leave from the current parent is requested to rejoin the network, otherwise - 0.

3.5.3.11. ZDO Management Permit Joining Request

ZDO_PERMIT_JOINING_REQ

Command id	Parameters	Data	Blocking?	Description
0x020b	Yes	No	Yes	Request a remote device or devices to allow or disallow association

Request packet format:

Octets: 5	2	1	1
Common request header	Destination NWK address	Permit Duration	TC Significance

Request parameters:

Req parameter name	Type	Length, bytes	Description
Destination NWK address	uint16_t	2	NWK address of the remote device to send the request to.
Permit Duration	uint8_t	1	The length of time in seconds during which the ZigBee coordinator or router will allow associations. The value 0x00 and 0xff indicate that permission is disabled or enabled, respectively, without a specified time limit.
TC Significance	uint8_t	1	Trust center significance. Must be set to 1 according to ZB specification.

Response packet format:

Octets: 7
Common response header

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	BUSY	NCP is busy with processing of another blocking request
GENERIC	NO_MEMORY	There is no memory to process the request
GENERIC	ERROR	Internal ZBOSS error
ZDO	TIMEOUT	Refer to ZB specification 2.4.5 ZDP Enumeration Description
ZDO	INVALID_REQUEST	Refer to ZB specification 2.4.5 ZDP Enumeration Description
ZDO	NOT_AUTHORIZED	Refer to ZB specification 2.4.5 ZDP Enumeration Description

3.5.3.12. ZDO Device Announce Indication

ZDO_DEV_ANNCE_IND

Command id	Parameters	Data	Blocking?	Description
0x020c	Yes	No	No	Device announce indication

Indication packet format:

Octets: 4	2	8	1
Common HL header	Device NWK Address	Device IEEE Address	MAC Capabilities

Indication parameters:

Ind parameter name	Type	Length, bytes	Description
Device NWK Address	uint16_t	2	Short address of the joined device.
Device IEEE Address	ieee_addr_t	8	IEEE address of the joined device.
MAC Capabilities	uint8_t	1	MAC Capabilities of the joined device.

3.5.3.13. ZDO Management Rejoin Request

ZDO_REJOIN

Command id	Parameters	Data	Blocking?	Description
0x020d	Yes	No	Yes	Rejoin to remote network even if joined already. If joined, clear internal data structures prior to joining. That call is useful for rejoin after parent loss.

Request packet format:

Octets: 5	8	1	variable	1
Common request header	Ext PAN ID	Channel List Len	Channel List	Secure rejoin

Request parameters:

Req parameter name	Type	Length, bytes	Description
Ext PAN ID	ieee_addr_t	8	Extended PAN ID.
Channel List Len	uint8_t	1	Number of entries in the following Channel List array. Must be 1 for 2.4GHz-only build.
Channel List	ChannelListEntry array	Channel List Len * ChannelListEntry size	Array of ChannelListEntry structures.
Secure rejoin	uint8_t	1	If 0 - unsecure rejoin, 1 - secure rejoin. Do secure rejoin (encrypt Rejoin frame by NWK key), else do Unsecure (Trust center rejoin)

Response packet format:

Octets: 7	1
Common response header	Flags

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
Flags	u8bm_t	1	Bit#0 1: Trust Center Swap-out happened, Key Establishment procedure may be required. Bits#1-7 reserved

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	BUSY	NCP is busy with processing of another blocking request
GENERIC	NO_MEMORY	There is no memory to process the request
GENERIC	ERROR	Internal ZBOSS error

Rejoin uses default Scan Duration value 3. To calculate the time spent scanning each channel, need to use the formula specified in NWK_NLME_JOIN call id.

3.5.3.14. ZDO System Server Discovery Request

ZDO_SYSTEM_SRV_DISCOVERY_REQ

Command id	Parameters	Data	Blocking?	Description
0x020e	Yes	No	Yes	Sends a ZDO system server discovery request

Request packet format:

Octets: 5	2
Common request header	Server mask

Request parameters:

Req parameter name	Type	Length, bytes	Description
Server mask	uint16_t	2	Server mask

Response packet format:

Octets: 7
Common response header

3.5.3.15. ZDO Management Bind Request

ZDO_MGMT_BIND_REQ

Command id	Parameters	Data	Blocking?	Description
0x020f	Yes	No	Yes	Sends a ZDO Mgmt Bind request to a remote device

Request packet format:

Octets: 5	2	1
Common request header	Target NWK Address	Start entry index

Request parameters:

Req parameter name	Type	Length, bytes	Description
Target NWK Address	uint16_t	2	the address of the device to send a request to
Start entry index	uint8_t	1	start entry index

Response packet format:

Octets: 7
Common response header

3.5.3.16. ZDO Management LQI Request

ZDO_MGMT_LQI_REQ

Command id	Parameters	Data	Blocking?	Description
0x0210	Yes	No	Yes	Sends a ZDO Mgmt LQI request to a remote device

Request packet format:

Octets: 5	2	1
Common request header	Target NWK Address	Start entry index

Request parameters:

Req parameter name	Type	Length, bytes	Description
Target NWK Address	uint16_t	2	the address of the device to send a request to
Start entry index	uint8_t	1	start entry index

Response packet format:

Octets: 7
Common response header

3.5.3.17. ZDO Management NWK Update Request

ZDO_MGMT_NWK_UPDATE_REQ

Command id	Parameters	Data	Blocking?	Description
0x0211	Yes	No	Yes	Sends a ZDO Mgmt NWK Update Request to a remote device

Request packet format:

Octets: 5	4	1	1	2	2
Common request header	Scan channel mask	Scan duration	Scan count	Manager Address	Destination NWK Address

Request parameters:

Req parameter name	Type	Length, bytes	Description

Req parameter name	Type	Length, bytes	Description
Scan channel mask	u32bm_t	4	Scan channel mask
Scan duration	uint8_t	1	Scan duration
Scan count	uint8_t	1	the number of energy scans to be conducted and reported
Manager Address	uint16_t	2	the NWK address of the network manager
Destination NWK Address	uint16_t	2	the address of the device to send a request to

Response packet format:

Octets: 7
Common response header

3.5.3.18. ZDO Diagnostics request

ZDO_GET_STATS

Command id	Parameters	Data	Blocking?	Description
0x0213	Yes	No	Yes	Require statistics (last message LQI\RSSI, counters, etc.) from the ZDO level

Request packet format:

Octets: 5	1
Common request header	Do cleanup

Request parameters:

Req parameter name	Type	Length, bytes	Description
Do cleanup	uint8_t	1	Cleanup ZDO statistics after requesting if true

Response packet format:

Octets: 7	4	4	4	4	2	2	2	2
Common response header	mac_rx_bcast	mac_tx_bcast	mac_rx_ucast	mac_tx_ucast_total_zcl	mac_tx_ucast_failures_zcl	mac_tx_ucast_retries_zcl	mac_tx_ucast_total	mac_t

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
mac_rx_bcast	uint32_t	4	A counter that is incremented each time the MAC layer receives a broadcast.
mac_tx_bcast	uint32_t	4	A counter that is incremented each time the MAC layer transmits a broadcast.
mac_rx_ucast	uint32_t	4	A counter that is incremented each time the MAC layer receives a unicast.
mac_tx_ucast_total_zcl	uint32_t	4	The same as mac_tx_ucast_total, but non-normalized
mac_tx_ucast_failures_zcl	uint16_t	2	The same as mac_tx_ucast_failures, but non-normalized
mac_tx_ucast_retries_zcl	uint16_t	2	The same as mac_tx_ucast_retries, but non-normalized
mac_tx_ucast_total	uint16_t	2	Total number of Mac Tx Transactions to attempt to send a message (but not counting retries).
mac_tx_ucast_failures	uint16_t	2	Total number of failed Tx Transactions. So if the Mac send a single packet, it will be retried 4 times without ack, that counts as 1 failure
mac_tx_ucast_retries	uint16_t	2	Total number of Mac Retries regardles of whether the transaction resulted in success or failure.
phy_to_mac_que_lim_reached	uint16_t	2	A counter that is incremented each time when MAC RX queue if full.
mac_validate_drop_cnt	uint16_t	2	How many times a packet was dropped on the stage of its validation for length or bad formatting.
phy_cca_fail_count	uint16_t	2	Number of times the PHY layer was unable to transmit due to a failed CCA.
period_of_time	uint8_t	1	Time period the MAC Tx results measured over
last_msg_lqi	uint8_t	1	LQI value of the last received packet
last_msg_rssi	int8_t	1	RSSI value of the last received packet

Rsp parameter name	Type	Length, bytes	Description
number_of_resets	uint16_t	2	An attribute that is incremented each time the device resets.
aps_tx_bcast	uint16_t	2	A counter that is incremented each time the APS layer receives a broadcast.
aps_tx_ucast_success	uint16_t	2	A counter that is incremented each time the APS layer successfully transmits a unicast.
aps_tx_ucast_retry	uint16_t	2	A counter that is incremented each time the APS layer retries sending a unicast.
aps_tx_ucast_fail	uint16_t	2	A counter that is incremented each time the APS layer fails to send a unicast.
route_disc_initiated	uint16_t	2	A counter that is incremented each time the network layer submits a route discovery message to the MAC.
nwk_neighbor_added	uint16_t	2	A counter that is incremented each time an entry is added to the neighbor table.
nwk_neighbor_removed	uint16_t	2	A counter that is incremented each time an entry is removed from the neighbor table.
nwk_neighbor_stale	uint16_t	2	A counter that is incremented each time a neighbor table entry becomes stale because the neighbor has not been heard from.
join_indication	uint16_t	2	A counter that is incremented each time a node joins or rejoins the network via this node.
childs_removed	uint16_t	2	A counter that is incremented each time an entry is removed from the child table.
nwk_fc_failure	uint16_t	2	A counter that is incremented each time a message is dropped at the network layer because the APS frame counter was not higher than the last message seen from that source.
aps_fc_failure	uint16_t	2	A counter that is incremented each time a message is dropped at the APS layer because the APS frame counter was not higher than the last message seen from that source.
aps_unauthorized_key	uint16_t	2	A counter that is incremented each time a message is dropped at the APS layer because it had APS encryption, but the key associated with the sender has not been authenticated, thus the key was not authorized for use in APS data messages. Data messages which are not APS encrypted are not dropped. Cluster id is not analyzed. Information about APS encryption is included into APS Frame FC Security bit. Packets which are not APS encrypted will be passed to the stack, no diagnostic counters change. If packet is APS encrypted but failed to be decrypted, ZBOSS drops it and increases diagnostic counter.
nwk_decrypt_failure	uint16_t	2	A counter that is incremented each time a NWK encrypted message was received, but dropped because decryption failed.
aps_decrypt_failure	uint16_t	2	A counter that is incremented each time a APS encrypted message was received, but dropped because decryption failed.
packet_buffer_allocate_failures	uint16_t	2	A counter that is incremented each time the stack failed to allocate a packet buffer.
average_mac_retry_per_aps_message_sent	uint16_t	2	The average number of MAC retries needed to send an APS message.
nwk_retry_overflow	uint16_t	2	A counter that is incremented on the NWK layer each time when number of attempts to retransmit a packet exceeds a limit. It is a non-standard counter that depends on ENABLE_NWK_RETRANSMIT and will be zero always when the macro is not set.
nwk_bcast_table_full	uint16_t	2	A non-standard counter of the number of times the NWK broadcast was dropped because the broadcast table was full.
status	uint8_t	1	It is zero if a request was done successfully, non-zero otherwise. The status is the same as in HL header. Can be safely skipped.

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	INVALID_PARAMETER	Parameter passed with a request is invalid
GENERIC	BUSY	NCP is busy with processing of another blocking request
GENERIC	NO_MEMORY	There is no memory to process the request
GENERIC	ERROR	Internal ZBOSS error

3.5.3.19. Device Authorized Indication

ZDO_DEV_AUTHORIZED_IND

Command id	Parameters	Data	Blocking?	Description

Command id	Parameters	Data	Blocking?	Description
0x0214	Yes	No	No	Indicates some device in the network was authorized (e.g. received TCLK)

Indication packet format:

Octets: 4	8	2	1	1
Common HL header	IEEE Address	NWK Address	Authorization Type	Authorization Status

Indication parameters:

Ind parameter name	Type	Length, bytes	Description
IEEE Address	ieee_addr_t	8	the IEEE address of the authorized device
NWK Address	uint16_t	2	the NWK address of the authorized device
Authorization Type	uint8_t	1	authorization type
Authorization Status	uint8_t	1	authorization status code

Authorization Types

- 0x00 - legacy
- 0x01 - TCLK

Authorization Status Codes

Status Codes for legacy authorization type.

- 0x00 - Authorization Success
- 0x01 - Authorization Failure

Status Codes for TCLK authorization type.

- 0x00 - Authorization Success
- 0x01 - Authorization Timeout
- 0x02 - Authorization Failure

3.5.3.20. Device Update Indication

ZDO_DEV_UPDATE_IND

Command id	Parameters	Data	Blocking?	Description
0x0215	Yes	No	No	Indicates some device joined the network

Indication packet format:

Octets: 4	8	2	1
Common HL header	IEEE Address	NWK Address	Status

Indication parameters:

Ind parameter name	Type	Length, bytes	Description
IEEE Address	ieee_addr_t	8	the IEEE address of the joined device
NWK Address	uint16_t	2	the NWK address of the joined device
Status	uint8_t	1	Device Update Status Code

Device Update Status Codes

- 0x00 - Standard Device Secured Rejoin
- 0x01 - Standard Device Unsecured Join
- 0x02 - Device Left
- 0x03 - Standard Device Trust Center Rejoin

3.5.3.21. Set Node Descriptor Manufacturer Code

ZDO_SET_NODE_DESC_MANUF_CODE

Command id	Parameters	Data	Blocking?	Description

Command id	Parameters	Data	Blocking?	Description
0x0216	Yes	No	Yes	Sets manufacturer code field in the node descriptor

Request packet format:

Octets: 5	2
Common request header	Manufacturer Code

Request parameters:

Req parameter name	Type	Length, bytes	Description
Manufacturer Code	uint16_t	2	the manufacturer code to set

Response packet format:

Octets: 7
Common response header

3.5.4. APS API

This category of the API provides an access to the Application Support Sub-layer at the NCP.

Call	code	Implemented?
APSDE_DATA_REQ	0x0301	Yes
APSME_BIND	0x0302	Yes
APSME_UNBIND	0x0303	Yes
APSME_ADD_GROUP	0x0304	Yes
APSME_RM_GROUP	0x0305	Yes
APSDE_DATA_IND	0x0306	Yes
APSME_RM_ALL_GROUPS	0x0307	Yes
APS_CHECK_BINDING	0x0308	Yes
APS_GET_GROUP_TABLE	0x0309	Yes
APSME_UNBIND_ALL	0x030a	Yes

3.5.4.1. APSDE Data Request

APSDE_DATA_REQ

Command id	Parameters	Data	Blocking?	Description
0x0301	Yes	Yes	No	APSDE-DATA.request

Note that the request parameters section has fixed size even while some parameters optional of has variable size depending on other parameter values. Such solution simplifies parameters parsing/composing. Note that parameters set is same as defined in Zigbee specification but its order differs.

Request packet format:

Octets: 5	1	2	8	2	2	1	1	1	1	1	1	2	1	variable
Common request header	Param Length	Data Length	Destination IEEE/NWK address	Profile Id	Cluster Id	Destination Endpoint	Source Endpoint	Radius	Dst address mode	TX Options	Use Alias	Alias source address	Alias sequence number	data

Request parameters:

Req parameter name	Type	Length, bytes	Description
Param Length	uint8_t	1	Length of parameters section starting from Destination IEEE/NWK address to Alias sequence number including (fixed as 21 bytes)
Data Length	uint16_t	2	Data section length
Destination IEEE/NWK address	uint16_t or ieee_addr_t or ignored	8	Destination address depending on dst addr mode. Always 8 bytes. 2-bytes short address must be put into first 2 bytes

Req parameter name	Type	Length, bytes	Description
Profile Id	uint16_t	2	Profile id
Cluster Id	uint16_t	2	Cluster id
Destination Endpoint	uint8_t or ignored	1	Destination endpoint (ignored if Dst address mode is other than 0x02 or 0x03)
Source Endpoint	uint8_t	1	Source endpoint
Radius	uint8_t	1	Radius in hops
Dst address mode	uint8_t	1	Destination address mode
TX Options	u8bm_t	1	TX options bitmap
Use Alias	uint8_t	1	0 or 1. If 1, use alias src address, else local short address. Should be 0 for SE.
Alias source address	uint16_t	2	Alias source address. Ignored if use alias is 0.
Alias sequence number	uint8_t	1	Alias sequence number. Ignored if use alias is 0.
data	u8arr_t	data len	Data bytes array. Maximum size of fragmented APS payload is 1550 bytes

Response packet format:

Octets: 7	8	1	1	4	1
Common response header	Destination IEEE/NWK address	Destination Endpoint	Source Endpoint	TX Time	Dst Address Mode

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
Destination IEEE/NWK address	uint16_t or ieee_addr_t or ignored	8	Destination address depending on dst addr mode. Always 8 bytes. 2-bytes short address is put into first 2 bytes
Destination Endpoint	uint8_t or ignored	1	Destination endpoint (ignored if Dst address mode is other than 0x02 or 0x03)
Source Endpoint	uint8_t	1	Source endpoint
TX Time	uint32_t	4	Transmission timestamp, ms
Dst Address Mode	uint8_t	1	Destination address mode

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	NOT_FOUND	Binding entry is not found if address mode is APS_ADDR_MODE_BIND_TBL_ID
GENERIC	NO_MEMORY	There is no memory to process the request
APS	NO_BOUND_DEVICE	Refer to ZB specification 2.2.9 APS Sub-Layer Status Values
APS	NOT_SUPPORTED	Refer to ZB specification 2.2.9 APS Sub-Layer Status Values
APS	NO_SHORT_ADDRESS	Refer to ZB specification 2.2.9 APS Sub-Layer Status Values
APS	SECURITY_FAIL	Refer to ZB specification 2.2.9 APS Sub-Layer Status Values
APS	NO_ACK	Refer to ZB specification 2.2.9 APS Sub-Layer Status Values
APS	ASDU_TOO_LONG	Refer to ZB specification 2.2.9 APS Sub-Layer Status Values

Destination Address Mode

- 0x00 = DstAddress and DstEndpoint not present
- 0x01 = 16-bit group address for DstAddress present, DstEndpoint not present
- 0x02 = 16-bit address for DstAddress present, DstEndpoint present
- 0x03 = 64-bit extended address for DstAddress present, DstEndpoint present
- 0x04 = "destination endpoint" is interpreted as an index in the binding table, all other destination address information is ignored

TX options bitmap

- 0x01 = Security enabled transmission
- 0x02 = Obsolete
- 0x04 = Acknowledged transmission
- 0x08 = Fragmentation permitted
- 0x10 = Include extended nonce in APS security frame.
- 0x20 = Force mesh route discovery for this request.
- 0x40 = Send route record for this request.

Outgoing packet timeouts

Name	Value, s	Description
APS_ACK_WAIT_DURATION_NON_SLEEPY	3	APS ACK time to wait from remote non Sleepy devices (the timeout depends on remote device role)
APS_ACK_WAIT_DURATION_SLEEPY	10	APS ACK time to wait from remote Sleepy devices (the timeout depends on remote device role)
APS_OUT_TRANSACTION_FAILED_NON_SLEEPY	9	Timer for outgoing APS fragment for remote non Sleepy devices (the timeout depends on remote device role)
APS_OUT_TRANSACTION_FAILED_SLEEPY	30	Timer for outgoing APS fragment for remote Sleepy devices (the timeout depends on remote device role)

Retries

Name	Value	Description
MAC_MAX_FRAME_RETRIES	3	The maximum number of retries allowed after a transmission failure. Actual number of all attempts is 4
NWKC_UNICAST_RETRIES	3	The number of network layer retries on unicast messages before reporting the result to the higher layer (4)
N_APS_MAX_FRAME_RETRIES	3	APS maximum of apscMaxFrameRetries times

Maximum size of fragmented APS payload is 1550 bytes. One-fragment packet size: without security 58 bytes, with security (w/o Extended Nonce) 49 bytes, with maximum security header size 41 bytes. A single fragment of multi-fragmented packet size: without security 56 bytes, with security (w/o Extended Nonce) 47 bytes, with maximum security header size 39 bytes. According to SE 1.4 specification, 5.3.6 Concentrator Parameters, ConcentratorRadius is 11 (eleven), that requires 24 bytes for source routing.

3.5.4.2. APSME Bind Request

APSME_BIND

Command id	Parameters	Data	Blocking?	Description
0x0302	Yes	No	No	APSME-BIND.Request

Request packet format:

Octets: 5	8	1	2	1	8	1
Common request header	Source IEEE Address	Source Endpoint	Cluster ID	Destination Address Mode	Destination IEEE/NWK Address	Destination Endpoint

Request parameters:

Req parameter name	Type	Length, bytes	Description
Source IEEE Address	ieee_addr_t	8	IEEE address of the source device.
Source Endpoint	uint8_t	1	Source endpoint number.
Cluster ID	uint16_t	2	Cluster ID to bind.
Destination Address Mode	uint8_t	1	Destination Address Mode
Destination IEEE/NWK Address	ieee_addr_t or uint16_t	8	IEEE or NWK address of the destination device depending on address mode specified. If NWK address is used, remaining bytes are ignored.
Destination Endpoint	uint8_t	1	Destination endpoint number. Shall be set to 0, if Destination Address Mode isn't 0x03.

Response packet format:

Octets: 7	1
Common response header	Index of bind table entry

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
Index of bind table entry	uint8_t	1	Index of bind table entry

Destination Address Mode

- 0x00 - reserved
- 0x01 - 16-bit group address for Dst Address and Dst Endpoint not present
- 0x02 - reserved
- 0x03 - 64-bit extended address for Dst Address and Dst Endpoint present
- 0x04 - 0xff = reserved

3.5.4.3. APSME Unbind Request

APSME_UNBIND

Command id	Parameters	Data	Blocking?	Description
0x0303	Yes	No	No	APSME-UNBIND.request

Request packet format:

Octets: 5	8	1	2	1	8	1
Common request header	Source IEEE Address	Source Endpoint	Cluster ID	Destination Address Mode	Destination IEEE/NWK Address	Destination Endpoint

Request parameters:

Req parameter name	Type	Length, bytes	Description
Source IEEE Address	ieee_addr_t	8	IEEE address of the source device.
Source Endpoint	uint8_t	1	Source endpoint number.
Cluster ID	uint16_t	2	Cluster ID to unbind.
Destination Address Mode	uint8_t	1	Destination Address Mode
Destination IEEE/NWK Address	ieee_addr_t or uint16_t	8	IEEE or NWK address of the destination device depending on address mode specified. If NWK address is used, remaining bytes are ignored.
Destination Endpoint	uint8_t	1	Destination endpoint number. Shall be set to 0, if Destination Address Mode isn't 0x03.

Response packet format:

Octets: 7	1
Common response header	Index of bind table entry

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
Index of bind table entry	uint8_t	1	Index of unbound entry

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	BUSY	NCP is busy with processing of another blocking request
GENERIC	NO_MEMORY	There is no memory to process the request
GENERIC	INVALID_PARAMETER	Parameter passed with a request is invalid
APS	INVALID_PARAMETER	Refer to ZB specification 2.2.9 APS Sub-Layer Status Values
APS	ILLEGAL_REQUEST	Refer to ZB specification 2.2.9 APS Sub-Layer Status Values
APS	INVALID_BINDING	Refer to ZB specification 2.2.9 APS Sub-Layer Status Values

3.5.4.4. APSME Add Group Request

APSME_ADD_GROUP

Command id	Parameters	Data	Blocking?	Description
0x0304	Yes	No	No	APSME-ADD-GROUP.request

Request packet format:

Octets: 5	2	1
Common request header	Group NWK Address	Endpoint

Request parameters:

Req parameter name	Type	Length, bytes	Description
Group NWK Address	uint16_t	2	NWK address of the group.
Endpoint	uint8_t	1	Endpoint number.

Response packet format:

Octets: 7
Common response header

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	INVALID_PARAMETER_1	Parameter passed with a request at the first place is invalid
GENERIC	INVALID_PARAMETER_2	Parameter passed with a request at the second place is invalid
GENERIC	INVALID_PARAMETER	Parameter passed with a request is invalid
APS	INVALID_PARAMETER	Refer to ZB specification 2.2.9 APS Sub-Layer Status Values
APS	TABLE_FULL	Refer to ZB specification 2.2.9 APS Sub-Layer Status Values

3.5.4.5. APSME Remove Group Request

APSME_RM_GROUP

Command id	Parameters	Data	Blocking?	Description
0x0305	Yes	No	No	APSME-REMOVE-GROUP.request

Request packet format:

Octets: 5	2	1
Common request header	Group NWK Address	Endpoint

Request parameters:

Req parameter name	Type	Length, bytes	Description
Group NWK Address	uint16_t	2	NWK address of the group.
Endpoint	uint8_t	1	Endpoint number.

Response packet format:

Octets: 7
Common response header

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success

Status category	Status code	Description
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	INVALID_PARAMETER_1	Parameter passed with a request at the first place is invalid
GENERIC	INVALID_PARAMETER_2	Parameter passed with a request at the second place is invalid
GENERIC	INVALID_PARAMETER	Parameter passed with a request is invalid
APS	INVALID_PARAMETER	Refer to ZB specification 2.2.9 APS Sub-Layer Status Values
APS	TABLE_FULL	Refer to ZB specification 2.2.9 APS Sub-Layer Status Values

3.5.4.6. APSDE Data Indication

APSDE_DATA_IND

Command id	Parameters	Data	Blocking?	Description
0x0306	Yes	Yes	No	APSDE-DATA.indication

Indication packet format:

Octets: 4	1	2	1	2	2	2	1	1	2	2	1	2	2	1	1
Common HL header	Param Length	Data Length	APS frame FC	Source NWK address	Destination NWK address	Group NWK address	Destination Endpoint	Source Endpoint	Cluster ID	Profile ID	APS packet counter	Source MAC header address	Destination MAC header address	LQI	R

Indication parameters:

Ind parameter name	Type	Length, bytes	Description
Param Length	uint8_t	1	Length of parameters section starting from APS frame FC to APS key source & attr including (fixed as 21 bytes)
Data Length	uint16_t	2	Length of data
APS frame FC	uint8_t	1	Received APS frame FC field
Source NWK address	uint16_t	2	Received frame source NWK address
Destination NWK address	uint16_t	2	Received frame destination NWK address
Group NWK address	uint16_t	2	Received frame APS group address (if frame is marked as Group addressed in FC)
Destination Endpoint	uint8_t	1	Destination endpoint
Source Endpoint	uint8_t or ignored	1	Source endpoint
Cluster ID	uint16_t	2	Cluster id
Profile ID	uint16_t	2	Profile id
APS packet counter	uint8_t	1	APS packet counter
Source MAC header address	uint16_t	2	Short source address of device that transmits that packet
Destination MAC header address	uint16_t	2	Next hop short address used for frame transmission
LQI	uint8_t	1	Received frame LQI
RSSI	uint8_t	1	Received frame RSSI
APS key source & attr	u8bm_t	1	Bit #0 - APS key source: • 0 - unknown • 1 - CBKE Bits #1-2 - APS key used: • 0 - provisional TCLK (key derived from install code) • 1 - unverified TCLK (impossible for SE profile) • 2 - verified TCLK (CBKE key) • 3 - application LK (partner link key between two non TC devices)
data	u8arr_t	data len	data bytes array

APS Frame FC

1-byte bitmap. Bits useful for the application:

- bits 2-3 - delivery mode:
  - 0 - Unicast
  - 2 - Broadcast
  - 3 - Group. "Group addr" field is valid.
- bit 5 - Security. 0 - Frame has no APS security, 1 - Frame is APS secured and "APS key source & attr" field is valid.
- bit 6 - APS ACK & retransmit requested

Incoming packet timeouts

Name	Value, s	Description
APS_IN_TRANSACTION_FAILED_NON_SLEEPY	12	Timer for incoming APS fragment for local non Sleepy devices (the timeout depends on NCP device role). The timeout is processed internally in NCP ZBOSS stack and has no meaning for Host application
APS_IN_TRANSACTION_FAILED_SLEEPY	40	Timer for incoming APS fragment for local Sleepy devices (the timeout depends on NCP device role). The timeout is processed internally in NCP ZBOSS stack and has no meaning for Host application

Maximum size of fragmented APS payload is 1550 bytes.

3.5.4.7. APSME Remove All Groups Request

APSME_RM_ALL_GROUPS

Command id	Parameters	Data	Blocking?	Description
0x0307	Yes	No	No	APSME-REMOVE-ALL-GROUPS.request

Request packet format:

Octets: 5	1
Common request header	Endpoint

Request parameters:

Req parameter name	Type	Length, bytes	Description
Endpoint	uint8_t	1	Endpoint number.

Response packet format:

Octets: 7
Common response header

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	INVALID_PARAMETER_1	Parameter passed with a request at the first place is invalid
APS	INVALID_PARAMETER	Refer to ZB specification 2.2.9 APS Sub-Layer Status Values
APS	INVALID_GROUP	Refer to ZB specification 2.2.9 APS Sub-Layer Status Values

3.5.4.8. APS Check Binding

APS_CHECK_BINDING

Command id	Parameters	Data	Blocking?	Description
0x0308	Yes	No	Yes	Checks if there are any bindings for specified endpoint and cluster

Request packet format:

Octets: 5	1	2
Common request header	Endpoint	Cluster ID

Request parameters:

Req parameter name	Type	Length, bytes	Description
Endpoint	uint8_t	1	Endpoint number
Cluster ID	uint16_t	2	Cluster ID

Response packet format:

Octets: 7	1
Common response header	Exists

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
Exists	uint8_t	1	flag indicating whether a binding exists

3.5.4.9. APS Get Group Table

APS_GET_GROUP_TABLE

Command id	Parameters	Data	Blocking?	Description
0x0309	Yes	No	Yes	Gets the APS Group Table

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7	1	variable
Common response header	Number of groups	Group List

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
Number of groups	uint16_t	1	the number of groups in the response (each group is of type uint16_t)
Group List	u16arr_t	2 * number of groups	group list

3.5.4.10. APSME Unbind All

APSME_UNBIND_ALL

Command id	Parameters	Data	Blocking?	Description
0x030a	Yes	No	Yes	Removes all bindings

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7
Common response header

3.5.5. NWK Management API

This category of the API provides Network Layer management command invocation.

Call	code	Implemented?
NWK_FORMATION	0x0401	Yes
NWK_DISCOVERY	0x0402	Yes
NWK_NLME_JOIN	0x0403	Yes

Call	code	Implemented?
NWK_PERMIT_JOINING	0x0404	Yes
NWK_GET_IEEE_BY_SHORT	0x0405	Yes
NWK_GET_SHORT_BY_IEEE	0x0406	Yes
NWK_GET_NEIGHBOR_BY_IEEE	0x0407	Yes
NWK_REJOINED_IND	0x0409	Yes
NWK_REJOIN_FAILED_IND	0x040a	Yes
NWK_LEAVE_IND	0x040b	Yes
PIM_SET_FAST_POLL_INTERVAL	0x040e	Yes
PIM_SET_LONG_POLL_INTERVAL	0x040f	Yes
PIM_START_FAST_POLL	0x0410	Yes
PIM_START_LONG_POLL	0x0411	Yes
PIM_START_POLL	0x0412	Yes
PIM_STOP_FAST_POLL	0x0414	Yes
PIM_STOP_POLL	0x0415	Yes
PIM_ENABLE_TURBO_POLL	0x0416	Yes
PIM_DISABLE_TURBO_POLL	0x0417	Yes
NWK_PAN_ID_CONFLICT_RESOLVE	0x041a	No
NWK_PAN_ID_CONFLICT_IND	0x041b	No
NWK_ADDRESS_UPDATE_IND	0x041c	Yes
NWK_START_WITHOUT_FORMATION	0x041d	Yes
NWK_NLME_ROUTER_START	0x041e	Yes
PARENT_LOST_IND	0x0420	Yes
PIM_START_TURBO_POLL_PACKETS	0x0424	Yes
PIM_START_TURBO_POLL_CONTINUOUS	0x0425	Yes
PIM_TURBO_POLL_CONTINUOUS_LEAVE	0x0426	Yes
PIM_TURBO_POLL_PACKETS_LEAVE	0x0427	Yes
PIM_PERMIT_TURBO_POLL	0x0428	Yes
PIM_SET_FAST_POLL_TIMEOUT	0x0429	Yes
PIM_GET_LONG_POLL_INTERVAL	0x042a	Yes
PIM_GET_IN_FAST_POLL_FLAG	0x042b	Yes
SET_KEEPALIVE_MOVE	0x042c	Yes
START_CONCENTRATOR_MODE	0x042d	Yes
STOP_CONCENTRATOR_MODE	0x042e	Yes
NWK_ENABLE_PAN_ID_CONFLICT_RESOLUTION	0x042f	Yes
NWK_ENABLE_AUTO_PAN_ID_CONFLICT_RESOLUTION	0x0430	Yes
PIM_TURBO_POLL_CANCEL_PACKET	0x0431	Yes

3.5.5.1. NWK Formation

NWK_FORMATION

Command id	Parameters	Data	Blocking?	Description
0x0401	Yes	No	Yes	NLME-NETWORK-FORMATION.request

Request packet format:

Octets: 5	1	variable	1	1	2
Common request header	Channel List Len	Channel List	Scan Duration	Distributed Network Flag	Distributed Network Address

Request parameters:

Req parameter name	Type	Length, bytes	Description
Channel List Len	uint8_t	1	Number of entries in the following Channel List array. Must be 1 for 2.4GHz-only build.
Channel List	ChannelListEntry array	Channel List Len * Channel List Len	Array of ChannelListEntry structures.
Scan Duration	uint8_t	1	The time spent scanning each channel is (aBaseSuperframeDuration * (2^n + 1)) symbols, where n is the value of ScanDuration parameter.
Distributed Network Flag	uint8_1	1	If 0, create a Centralized network, device is ZC. If 1, create a Distributed network, device is ZR
Distributed Network Address	uint16_t	2	The address the device will use when forming a distributed network.

Response packet format:

Octets: 7	1
Common response header	NWK address

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
NWK address	uint16_t	1	

3.5.5.2. NWK Discovery

NWK_DISCOVERY

Command id	Parameters	Data	Blocking?	Description
0x0402	Yes	No	Yes	NLME-NETWORK-DISCOVERY.request

Request packet format:

Octets: 5	1	variable	1
Common request header	Channel List Len	Channel List	Scan Duration

Request parameters:

Req parameter name	Type	Length, bytes	Description
Channel List Len	uint8_t	1	Number of entries in the following Channel List array. Must be 1 for 2.4GHz-only build.
Channel List	ChannelListEntry array	Channel List Len * ChannelListEntry size	Array of ChannelListEntry structures.
Scan Duration	uint8_t	1	The time spent scanning each channel is (aBaseSuperframeDuration * (2^n + 1)) symbols, where n is the value of ScanDuration parameter.

Response packet format:

Octets: 7	1	variable
Common response header	Network Count	Network Descriptors Array

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
Network Count	uint8_t	1	Length of Network descriptors array followed
Network Descriptors Array	Array of Network Descriptors	Network count * 14	Array of Network descriptors

Status codes:

Status category	Status code	Description

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	BUSY	NCP is busy with processing of another blocking request
GENERIC	NO_MEMORY	There is no memory to process the request
GENERIC	INVALID_PARAMETER	Parameter passed with a request is invalid
GENERIC	NOT_IMPLEMENTED	The request is not implemented for the given role and band (for e.g. ZR and SubGhz).
MAC	NO_BEACON	Refer to table 78 in MAC specification
MAC	INVALID_PARAMETER	Refer to table 78 in MAC specification

Network Descriptor

Octets: 8	2	1	1	1	1	1	1
Ext pan id	pan id	NWK Update id	Channel page	Channel	Flags	LQI	RSSI

NWK desc component name	Type	Length, bytes	Description
Ext pan id	ieee_addr_t	8	The 64-bit PAN identifier of the network.
pan id	uint16_t	2	The 16-bit PAN identifier of the network.
NWK Update id	uint8_t	1	The value of the UpdateID from the NIB
Channel page	uint8_t	1	Logical page #. Always 0 for 2.4GHz-only NCP build
Channel	uint8_t	1	The current logical channel occupied by the network.
Flags	uint8_t	1	Packed set of NWK descriptor fields:
			• bit0 - Permit Joining
			• bit1 - Router capacity
			• bit2 - ED capacity
			• bits4-7 - Stack profile
LQI	uint8_t	1	LQI value from the best potential parent in a network.
RSSI	int8_t	1	RSSI value from the best potential parent in a network.

Note that defined by standard ZigBeeVersion, BeaconOrder, SuperframeOrder fields are not included into Network descriptor because that parameters are useless for the modern Zigbee specifications.

3.5.5.3. Join/Rejoin Network

NWK_NLME_JOIN

Command id	Parameters	Data	Blocking?	Description
0x0403	Yes	No	Yes	Join network, do basic post-join actions

NOTE: by executing that request NCP does not only Association/Rejoin, but also initial commissioning actions required by Zigbee specification: receives NWK key, broadcasts Device Announce, starts router and sends Parent Announce if joining device is ZR. NOTE: This call can NOT be used to rejoin from the already joined state. Use ZDO_Rejoin instead.

Request packet format:

Octets: 5	8	1	1	variable	1	1	1
Common request header	Ext PAN ID	Rejoin Network	Channel List Len	Channel List	Scan duration	MAC Capabilities	Security Enable

Request parameters:

Req parameter name	Type	Length, bytes	Description
Ext PAN ID	ieee_addr_t	8	The 64-bit PAN identifier of the network to join/rejoin.
Rejoin Network	uint8_t	1	If 0, Associate with the network. If 2, Rejoin the network. Other values are reserved.

Req parameter name	Type	Length, bytes	Description
Channel List Len	uint8_t	1	Number of entries in the following Channel List array. Must be 1 for 2.4GHz-only build.
Channel List	ChannelListEntry array	Channel List Len * ChannelListEntry size	Array of ChannelListEntry structures.
Scan duration	uint8_t	1	The time spent scanning each channel is (aBaseSuperframeDuration * (2^n + 1)) symbols, where n is the value of ScanDuration parameter.
MAC Capabilities	u8bm_t	1	MAC Capabilities bitfield
Security Enable	uint8_t	1	If 0 - unsecure rejoin, 1 - secure rejoin. If Rejoin Network equals 2, do secure rejoin (encrypt Rejoin frame by NWK key), else do Unsecure (Trust center rejoin)

Response packet format:

Octets: 7	2	8	1	1	1	1
Common response header	short address	Ext pan id	channel page	channel #	enh beacon	MAC interface #

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
short address	uint16_t	2	The 16-bit network address that was allocated to this device.
Ext pan id	ieee_addr_t	8	The 64-bit PAN identifier of the network to join/rejoin.
channel page	uint8_t	1	Channel page of active channel. Always 0 for 2.4GHz-only NCP
channel #	uint8_t	1	Current logical channel
enh beacon	uint8_t	1	If 1 if system uses enhanced beacons
MAC interface #	uint8_t	1	Current MAC interface number. Always 0 for single-MAC NCP (note: single-MAC NCP can still support Sub-GHz band)

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	INVALID_STATE	Device is already joined
GENERIC	INVALID_PARAMETER	Parameter passed with a request is invalid
GENERIC	INVALID_PARAMETER_6	Parameter passed with a request at the sixth place is invalid
ZDO	NOT_AUTHORIZED	Refer to ZB specification 2.4.5 ZDP Enumeration Description
NWK	NOT_PERMITTED	Refer to ZB specification 3.7 NWK Layer Status Values
NWK	NO_NETWORKS	Refer to ZB specification 3.7 NWK Layer Status Values
MAC	NO_BEACON	Refer to table 78 in MAC specification
MAC	INVALID_PARAMETER	Refer to table 78 in MAC specification
MAC	NO_ACK	Refer to table 78 in MAC specification
MAC	NO_DATA	Refer to table 78 in MAC specification

To use the command for rejoining, a device must leave current network at first. NWK rejoin can not be handled from joined state. While leaving a network, the device resets to factory default settings. If a user does not want to leave and reset, ZDO rejoin can be used instead of NWK join/rejoin. Initial joining through NWK rejoin is also possible according to Zigbee specification 3.6.1.4.2 Joining or Rejoining a Network Using NWK Rejoin.

3.5.5.4. NWK Permit Joining

NWK_PERMIT_JOINING

Command id	Parameters	Data	Blocking?	Description
0x0404	Yes	No	Yes	NLME-PERMIT-JOINING.request

Request packet format:

Octets: 5	1
Common request header	Permit Duration

Request parameters:

Req parameter name	Type	Length, bytes	Description
Permit Duration	uint8_t	1	Permit join duration, in seconds. 0 value indicates that permission is disabled. The value 0xff is interpreted internally as 0xfe (see section 2.4.3.3.7 Mgmt_Permit_Joining_req)

Response packet format:

Octets: 7
Common response header

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	BUSY	NCP is busy with processing of another blocking request
GENERIC	NO_MEMORY	There is no memory to process the request

3.5.5.5. Get IEEE address by short address

NWK_GET_IEEE_BY_SHORT

Command id	Parameters	Data	Blocking?	Description
0x0405	Yes	No	No	Get IEEE address by short address from the local address translation table

Request packet format:

Octets: 5	2
Common request header	NWK Address

Request parameters:

Req parameter name	Type	Length, bytes	Description
NWK Address	uint16_t	2	Short address of interest

Response packet format:

Octets: 7	8
Common response header	IEEE address

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
IEEE address	ieee_addr_t	8	IEEE address from the local address translation table.

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	NOT_FOUND	IEEE address is not found for a given short address

If NCP can't find the short address in the local translation table, it responds with only common response header with error code.

If NCP was able to find a device record in the local translation table, it responds with a full response packet.

3.5.5.6. Get short address by IEEE address

NWK_GET_SHORT_BY_IEEE

Command id	Parameters	Data	Blocking?	Description
0x0406	Yes	No	No	Get short address by IEEE address from the local address translation table

Request packet format:

Octets: 5	8
Common request header	IEEE Address

Request parameters:

Req parameter name	Type	Length, bytes	Description
IEEE Address	ieee_addr_t	8	IEEE address of interest.

Response packet format:

Octets: 7	2
Common response header	NWK Address

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
NWK Address	uint16_t	2	Short address from the local address translation table.

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	NOT_FOUND	Short address is not found for a given IEEE address

If NCP can't find an address in the local translation table, it responds with only common response header with error code.

If NCP was able to find a device record in the local translation table, it responds with a full response packet.

3.5.5.7. Get neighbor by IEEE

NWK_GET_NEIGHBOR_BY_IEEE

Command id	Parameters	Data	Blocking?	Description
0x0407	Yes	No	No	Get local neighbor table entry by IEEE address

Request packet format:

Octets: 5	8
Common request header	IEEE Address

Request parameters:

Req parameter name	Type	Length, bytes	Description
IEEE Address	ieee_addr_t	8	IEEE address

Response packet format:

Octets: 7	8	2	1	1	2	4	4	1	1	1	1	1	1	1
Common response header	IEEE Address	NWK Address	Zigbee Role	RX On When Idle	ED Config	Timeout Counter	Device Timeout	Relationship	Transmit Failure Cnt	LQI	Outgoing Cost	Age	Keepalive Received	MAC Interface Index

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
IEEE Address	ieee_addr_t	8	IEEE address.
NWK Address	uint16_t	2	Short address.
Zigbee Role	uint8_t	1	Zigbee role code: 0 - ZC, 1 - ZR, 1 - ZED.
RX On When Idle	uint8_t	1	0 if sleepy ZED, else 1.
ED Config	uint16_t	2	Always 0.
Timeout Counter	uint32_t	4	The current time remaining, in seconds, for the end device. After reaching 0, the end device will be aged out (the field has no significance for NCP ZED or when a neighbor is a router).
Device Timeout	uint32_t	4	Timeout, in seconds, for the end device child. ED child periodically sends a keepalive to their router parent to insure they remain in the router's neighbor table (the field has no significance for NCP ZED or when a neighbor is a router).
Relationship	uint8_t	1	The relationship between the neighbor and the current device: • 0x00=neighbor is the parent • 0x01=neighbor is a child • 0x02=neighbor is a sibling • 0x03=none of the above • 0x04=previous child • 0x05=unauthenticated child. When a router joins a coordinator, initial relationship value that ZC is a parent for ZR, it is given after joining and not cleaned up. There is no harm in such relationship for the router. If a router joins through range extender and if the router and a coordinator can see each other they will be siblings. If a router joins through range extender and if a coordinator can see the router it sends messages directly. ZBOSS can detect a situation when there is asymmetric link and a device shouldn't send messages directly.
Transmit Failure Cnt	uint8_t	1	A value indicating if previous transmissions to the device were successful or not.
LQI	uint8_t	1	The estimated link quality for RF transmissions from this device. LQI is calculated using RSSI provided by PHY layer with the formula: if rssi < -101 LQI = 0; else if (rssi > -40) LQI = 255; else LQI = (255 * (RSSI + 101)) / 61
Outgoing Cost	uint8_t	1	The cost of an outgoing link as measured by the neighbor. A value of 0 indicates no outgoing cost is available (the field has no significance for NCP ZED or when NCP ZR and a neighbor is its child). Formula: 7 - ((LQI >> 5) & 0x07)
Age	uint8_t	1	The number of nwkLink-StatusPeriod intervals since a link status command was received (applicable for routers only)
Keepalive Received	uint8_t	1	At least one keepalive has been received from the end device since the router has rebooted (the field has no significance for NCP ZED).
MAC Interface Index	uint8_t	1	Index into the MAC Interface Table indicating what interface the neighbor or child is bound to. There is only one MAC interface for NCP, so the field has no significance.

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	NOT_FOUND	Neighbor is not found for a given IEEE address

3.5.5.8. Network Rejoin Indication

NWK_REJOINED_IND

Command id	Parameters	Data	Blocking?	Description
0x0409	Yes	No	No	Indicates that network rejoining procedure has completed

Indication packet format:

Octets: 4	2	8	1	1	1	1
Common HL header	NWK Address	Ext PAN ID	Channel page	Channel	Beacon Type	MAC Interface Num

Indication parameters:

Ind parameter name	Type	Length, bytes	Description
NWK Address	uint16_t	2	Assigned short address.
Ext PAN ID	ieee_addr_t	8	Extended PAN ID.
Channel page	uint8_t	1	Current channel page.
Channel	uint8_t	1	Current channel number.
Beacon Type	uint8_t	1	0 - Non-enhanced beacon, 1 - Enhanced Beacon
MAC Interface Num	uint8_t	1	MAC interface number.

The only case when automatic rejoin allowed for NCP is processing Leave request with Rejoin bit set.

3.5.5.9. Network Rejoin Failed Indication

NWK_REJOIN_FAILED_IND

Command id	Parameters	Data	Blocking?	Description
0x040a	Yes	No	No	Indicates that network rejoining procedure has failed

Indication packet format:

Octets: 4	1	1
Common HL header	Status Category	Status Code

Indication parameters:

Ind parameter name	Type	Length, bytes	Description
Status Category	uint8_t	1	Failure status category
Status Code	uint8_t	1	Failure status code

Status codes:

Status category	Status code	Description
ZDO	NOT_AUTHORIZED	Refer to ZB specification 2.4.5 ZDP Enumeration Description
NWK	NOT_PERMITTED	Refer to ZB specification 3.7 NWK Layer Status Values
NWK	NO_NETWORKS	Refer to ZB specification 3.7 NWK Layer Status Values
MAC	NO_BEACON	Refer to table 78 in MAC specification
MAC	INVALID_PARAMETER	Refer to table 78 in MAC specification
MAC	NO_ACK	Refer to table 78 in MAC specification
MAC	NO_DATA	Refer to table 78 in MAC specification

The only case when automatic rejoin allowed for NCP is processing Leave request with Rejoin bit set.

3.5.5.10. Network Leave Indication

NWK_LEAVE_IND

Command id	Parameters	Data	Blocking?	Description
0x040b	Yes	No	No	Network Leave indication

Indication packet format:

Octets: 4	8	1
Common HL header	IEEE Address	Rejoin

Indication parameters:

Ind parameter name	Type	Length, bytes	Description
IEEE Address	ieee_addr_t	8	IEEE address.
Rejoin	uint8_t	1	0 - No rejoin, 1 - Rejoin requested.

3.5.5.11. Set Fast Poll Interval PIM Attribute

PIM_SET_FAST_POLL_INTERVAL

Command id	Parameters	Data	Blocking?	Description
0x040e	Yes	No	No	Set Fast Poll Interval PIM attribute

Request packet format:

Octets: 5	2
Common request header	Fast Poll Interval

Request parameters:

Req parameter name	Type	Length, bytes	Description
Fast Poll Interval	uint16_t	2	Fast poll interval in range 0x01-0xFFFF in quarterseconds.

Response packet format:

Octets: 7
Common response header

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	INVALID_PARAMETER_1	Parameter passed with a request at the first place is invalid

Default Fast Poll Interval is 2 quarterseconds (500ms).

3.5.5.12. Set Long Poll Interval PIM Attribute

PIM_SET_LONG_POLL_INTERVAL

Command id	Parameters	Data	Blocking?	Description
0x040f	Yes	No	No	Set Long Poll Interval PIM attribute

Request packet format:

Octets: 5	4
Common request header	Long Poll Interval

Request parameters:

Req parameter name	Type	Length, bytes	Description
Long Poll Interval	uint32_t	4	Long poll interval in range 0x04-0x6E0000 in quarterseconds.

Response packet format:

Octets: 7
Common response header

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	INVALID_PARAMETER_1	Parameter passed with a request at the first place is invalid

Default Long Poll Interval is 20 quarterseconds (5000ms).

3.5.5.13. Start Fast Poll

PIM_START_FAST_POLL

Command id	Parameters	Data	Blocking?	Description
0x0410	No	No	No	Start poll with the Fast Poll Interval specified by PIM attribute

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7
Common response header

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length

3.5.5.14. Start Long Poll

PIM_START_LONG_POLL

Command id	Parameters	Data	Blocking?	Description
0x0411	No	No	No	Start Long Poll

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7
Common response header

3.5.5.15. Start Poll

PIM_START_POLL

Command id	Parameters	Data	Blocking?	Description
0x0412	No	No	No	Start poll with the Long Poll Interval specified by PIM attribute

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7
Common response header

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	NOT_IMPLEMENTED	Request can not be implemented for ZR role

3.5.5.16. Stop Fast Poll

PIM_STOP_FAST_POLL

Command id	Parameters	Data	Blocking?	Description
0x0414	No	No	No	Stop fast poll and restart it with the Long Poll Interval

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7	1
Common response header	Stop Fast Poll Result

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
Stop Fast Poll Result	uint8_t	1	Operation result: • 0 - Fast poll was not started • 1 - Fast poll was not stopped • 2 - Fast poll was stopped

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length

3.5.5.17. Stop Poll

PIM_STOP_POLL

Command id	Parameters	Data	Blocking?	Description
0x0415	No	No	No	Stop automatic ZBOSS poll

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7
Common response header

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length

NCP stops polling a parent either Fast or Long Poll modes and goes to Single poll mode.

3.5.5.18. Enable turbo poll

PIM_ENABLE_TURBO_POLL

Command id	Parameters	Data	Blocking?	Description
0x0416	Yes	No	No	Enable turbo poll for a given amount of time.

Request packet format:

Octets: 5	4
Common request header	Time

Request parameters:

Req parameter name	Type	Length, bytes	Description
Time	uint32_t	4	Turbo poll active time in milliseconds.

Response packet format:

Octets: 7
Common response header

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length

Turbo poll is a mechanism for increasing polling rate when an End device expects some Zigbee packets in a short time. After enabling the turbo poll interval starts from 100ms, then polling rate increases by 50% each time until it reaches given Turbo poll active time value. When ZED receives a packet the turbo poll rate decreases by 20%. The turbo poll stops when ZED receives a given number of packets for turbo poll (see PIM Start Turbo Poll Packets) or Turbo poll active time is over. When a new packet is added to be polled in turbo mode while it's in progress, polling rate is set back to initial value (100ms). For fragmented APS TX/RX procedures NCP automatically starts turbo poll if it is not prohibited for default value 3000ms. Host must always specify Turbo poll active time for the request. When auto-polling is disabled, NCP polls in turbo mode for APS RX/TX fragments and stack-initiated packets.

3.5.5.19. Disable turbo poll

PIM_DISABLE_TURBO_POLL

Command id	Parameters	Data	Blocking?	Description
0x0417	No	No	No	Disable turbo poll for a given amount of time.

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7
Common response header

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length

3.5.5.20. Pan ID Conflict Resolve request

NWK_PAN_ID_CONFLICT_RESOLVE

Not implemented.

3.5.5.21. Pan ID Conflict Indication

NWK_PAN_ID_CONFLICT_IND

Not implemented.

3.5.5.22. Address Update Indication

NWK_ADDRESS_UPDATE_IND

Command id	Parameters	Data	Blocking?	Description
0x041c	Yes	No	No	Address update indication.

Indication packet format:

Octets: 4	2

Octets: 4	2
Common HL header	NWK Address

Indication parameters:

Ind parameter name	Type	Length, bytes	Description
NWK Address	uint16_t	2	NWK address

3.5.5.23. Start without formation

NWK_START_WITHOUT_FORMATION

Command id	Parameters	Data	Blocking?	Description
0x041d	No	No	No	Start without forming a new network.

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7
Common response header

Status codes:

Status category	Status code	Description
GENERIC	NOT_IMPLEMENTED	Request is not implemented since it's for ZC role

NOTE: This command is meaningful only for ZC role. In case of other roles, the call will return NOT_IMPLEMENTED status.

3.5.5.24. NWK NLME start router request

NWK_NLME_ROUTER_START

Command id	Parameters	Data	Blocking?	Description
0x041e	Yes	No	No	NWK NLME start router request

Request packet format:

Octets: 5	1	1	1
Common request header	Beacon order	Superframe order	Battery life extension

Request parameters:

Req parameter name	Type	Length, bytes	Description
Beacon order	uint8_t	1	
Superframe order	uint8_t	1	
Battery life extension	uint8_t	1	

Response packet format:

Octets: 7
Common response header

3.5.5.25. Parent lost indication

PARENT_LOST_IND

Command id	Parameters	Data	Blocking?	Description
0x0420	Yes	No	No	Indicates that joined device has no parent

Indication packet format:

Octets: 4
Common HL header

This indication occurs every time the number of unsuccessful retry attempts to communicate with the parent reaches a certain limit. The NCP does not take any additional action in this situation, and the Host must make its own decision if to initiate rejoin. If ED does not get any response from parent it will reach PARENT_LOST state. In that state the device remains joined, and it is up to Host whether it should leave or rejoin.

Retries

Name	Value	Description
ZDO_PARENT_LINK_FAILURE_CNT	12	Number of times the device APS to fail sending packets to the parent before initiating rejoin
ZDO_MAX_PARENT_THRESHOLD_RETRY	10	ZDO maximum threshold for parent Data request retries

ZDO_PARENT_LINK_FAILURE_CNT is an upper limit for consecutive APS retries to send a packet. For each packet there are 3 APS transmission attempts, so after 4 APS packets failed to transmit in a row, NCP considers that parent is lost. Actual number of packets on the air depends also on the number of MAC and NWK retries, and for the current configuration there are 4 MAC attempts, 4 NWK attempts, 3 APS attempts. It makes out 48 attempts to send an APS packet and 16*12 = 192 attempts before parent lost indication. ZDO_MAX_PARENT_THRESHOLD_RETRY is an upper limit for consecutively failed NLME-SYNC.requests. After issuing NLME-SYNC.request, ZBOSS stack sends MLME-POLL.request. For each Data request there are 4 MAC transmission attempts, and after 10 failed NLME-SYNC.requests in a row the NCP considers that parent is lost. Actual number of packets on the air depends on number of MAC attempts. So there are 40 Data requests before parent lost indication. From Host application point, NLME-SYNC.request can be considered as a Single Poll request.

3.5.5.26. PIM Start Turbo Poll Packets

PIM_START_TURBO_POLL_PACKETS

Command id	Parameters	Data	Blocking?	Description
0x0424	Yes	No	No	Starts a turbo poll until the specified number of packets are received

Request packet format:

Octets: 5	1
Common request header	Number of packets

Request parameters:

Req parameter name	Type	Length, bytes	Description
Number of packets	uint8_t	1	The number of packets to poll

Response packet format:

Octets: 7
Common response header

3.5.5.27. PIM Start Turbo Poll Continuous

PIM_START_TURBO_POLL_CONTINUOUS

Command id	Parameters	Data	Blocking?	Description
0x0425	Yes	No	No	Starts a turbo poll for specified amount of time

Request packet format:

Octets: 5	4
Common request header	Turbo Poll Timeout

Request parameters:

Req parameter name	Type	Length, bytes	Description
Turbo Poll Timeout	uint32_t	4	The duration of turbo poll in ms

Response packet format:

Octets: 7
Common response header

3.5.5.28. PIM Turbo Poll Continuous Leave

PIM_TURBO_POLL_CONTINUOUS_LEAVE

Command id	Parameters	Data	Blocking?	Description
0x0426	Yes	No	No	Stops the continuous turbo poll

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7
Common response header

3.5.5.29. PIM Turbo Poll Packets Leave

PIM_TURBO_POLL_PACKETS_LEAVE

Command id	Parameters	Data	Blocking?	Description
0x0427	Yes	No	No	Resets a number of packets to turbo poll.

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7
Common response header

3.5.5.30. PIM Permit Turbo Poll

PIM_PERMIT_TURBO_POLL

Command id	Parameters	Data	Blocking?	Description
0x0428	Yes	No	No	Permits Turbo Poll

Request packet format:

Octets: 5	1
Common request header	Permit flag

Request parameters:

Req parameter name	Type	Length, bytes	Description
Permit flag	uint8_t	1	

Response packet format:

Octets: 7
Common response header

3.5.5.31. PIM Set Fast Poll Timeout

PIM_SET_FAST_POLL_TIMEOUT

Command id	Parameters	Data	Blocking?	Description
0x0429	Yes	No	No	Sets the fast poll timeout

Request packet format:

Octets: 5	4

Octets: 5	4
Common request header	Fast Poll Timeout

Request parameters:

Req parameter name	Type	Length, bytes	Description
Fast Poll Timeout	uint32_t	4	the fast poll timeout to set in ms

Response packet format:

Octets: 7
Common response header

3.5.5.32. PIM Get Long Poll Interval

PIM_GET_LONG_POLL_INTERVAL

Command id	Parameters	Data	Blocking?	Description
0x042a	Yes	No	No	Gets the current long poll interval

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7	4
Common response header	Long Poll Interval

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
Long Poll Interval	uint32_t	4	the current long poll interval, in ms

3.5.5.33. PIM Get In Fast Poll Flag

PIM_GET_IN_FAST_POLL_FLAG

Command id	Parameters	Data	Blocking?	Description
0x042b	Yes	No	No	Checks whether the NCP is in fast poll now

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7	1
Common response header	In Fast Poll Flag

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
In Fast Poll Flag	uint8	1	

3.5.5.34. Set Keepalive mode

SET_KEEPALIVE_MOVE

Command id	Parameters	Data	Blocking?	Description
0x042c	Yes	No	No	Sets keepalive mode

Request packet format:

Octets: 5	1
Common request header	Keepalive mode

Request parameters:

Req parameter name	Type	Length, bytes	Description
Keepalive mode	uint8_t	1	keepalive mode

Response packet format:

Octets: 7
Common response header

Keepalive Modes

- 0x00 - Keepalive Disabled
- 0x01 - MAC Data Poll
- 0x02 - ED Timeout Request
- 0x03 - Both Methods

3.5.5.35. Start Concentrator Mode

START_CONCENTRATOR_MODE

Command id	Parameters	Data	Blocking?	Description
0x042d	Yes	No	No	Starts a concentrator mode

Request packet format:

Octets: 5	1	4
Common request header	Radius	Time between discoveries

Request parameters:

Req parameter name	Type	Length, bytes	Description
Radius	uint8_t	1	
Time between discoveries	uint32_t	4	

Response packet format:

Octets: 7
Common response header

3.5.5.36. Stop Concentrator Mode

STOP_CONCENTRATOR_MODE

Command id	Parameters	Data	Blocking?	Description
0x042e	No	No	No	Stops a concentrator mode

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7
Common response header

3.5.5.37. Enable PAN ID Conflict Resolution

NWK_ENABLE_PAN_ID_CONFLICT_RESOLUTION

Command id	Parameters	Data	Blocking?	Description

Command id	Parameters	Data	Blocking?	Description
0x042f	Yes	No	No	Enables or disables PAN ID conflict resolution

Request packet format:

Octets: 5	1
Common request header	Enable flag

Request parameters:

Req parameter name	Type	Length, bytes	Description
Enable flag	uint8_t	1	0 - disable, 1 - enable

Response packet format:

Octets: 7
Common response header

3.5.5.38. Enable Auto PAN ID Conflict Resolution

NWK_ENABLE_AUTO_PAN_ID_CONFLICT_RESOLUTION

Command id	Parameters	Data	Blocking?	Description
0x0430	Yes	No	No	Enables or disables automatic PAN ID conflict resolution

Request packet format:

Octets: 5	1
Common request header	Enable flag

Request parameters:

Req parameter name	Type	Length, bytes	Description
Enable flag	uint8_t	1	0 - disable, 1 - enable

Response packet format:

Octets: 7
Common response header

3.5.5.39. PIM Turbo Poll Cancel Packet

PIM_TURBO_POLL_CANCEL_PACKET

Command id	Parameters	Data	Blocking?	Description
0x0431	No	No	No	Decreases the number of packets to turbo poll by one

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7
Common response header

3.5.6. Security API

Call	code	Implemented?
SECUR_SET_LOCAL_IC	0x0501	Yes
SECUR_ADD_IC	0x0502	Yes
SECUR_DEL_IC	0x0503	Yes

Call	code	Implemented?
SECUR_GET_LOCAL_IC	0x050d	Yes
SECUR_TCLK_IND	0x050e	Yes
SECUR_TCLK_EXCHANGE_FAILED_IND	0x050f	Yes
SECUR_NWK_INITIATE_KEY_SWITCH_PROCEDURE	0x0517	Yes
SECUR_GET_IC_LIST	0x0518	Yes
SECUR_GET_IC_BY_IDX	0x0519	Yes
SECUR_REMOVE_ALL_IC	0x051a	Yes

3.5.6.1. Set Local Device Installcode

SECUR_SET_LOCAL_IC

Command id	Parameters	Data	Blocking?	Description
0x0501	Yes	No	No	Set local device installcode to ZR/ZED

Note: the call is meaningful for joining devices (ZR/ZED) only and should't be done for ZC, since the local Installcode isn't used in a case of coordinator.

Installcode length depends on its type, defined by the Host (NOTE: the length includes 2 bytes of CRC):

- 8 bytes - 48-bit
- 10 bytes - 64-bit
- 14 bytes - 96-bit
- 18 bytes - 128-bit

Request packet format:

Octets: 5	8, 10, 14 or 18
Common request header	Installcode

Request parameters:

Req parameter name	Type	Length, bytes	Description
Installcode	u8arr_t	8, 10, 14 or 18	Installcode, including trailing 2 bytes of CRC

Response packet format:

Octets: 7
Common response header

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	CONVERSION_ERROR	Invalid install code CRC

3.5.6.2. Add Remote Device Installcode

SECUR_ADD_IC

Command id	Parameters	Data	Blocking?	Description
0x0502	Yes	No	No	Set remote device installcode to ZC

The call is meaningful for ZC only.

Request packet format:

Octets: 5	8	8, 10, 14 or 18
Common request header	IEEE Address	Installcode

Request parameters:

Req parameter name	Type	Length, bytes	Description
IEEE Address	ieee_addr_t	8	IEEE address of the remote device
Installcode	u8arr_t	8, 10, 14 or 18	Installcode, including two trailing bytes of CRC

Response packet format:

Octets: 7
Common response header

Status codes:

Status category	Status code	Description
GENERIC	NOT_IMPLEMENTED	The request is not implemented since it's for ZC role

NOTE: This command is meaningful only for ZC role. In case of other roles, the call will return NOT_IMPLEMENTED status.

3.5.6.3. Delete Remote Device Installcode

SECUR_DEL_IC

Command id	Parameters	Data	Blocking?	Description
0x0503	Yes	No	No	Delete remote device installcode from ZC

The call is meaningful for ZC only.

Request packet format:

Octets: 5	8
Common request header	IEEE Address

Request parameters:

Req parameter name	Type	Length, bytes	Description
IEEE Address	ieee_addr_t	8	IEEE address of the remote device

Response packet format:

Octets: 7
Common response header

Status codes:

Status category	Status code	Description
GENERIC	NOT_IMPLEMENTED	The request is not implemented since it's for ZC role

NOTE: This command is meaningful only for ZC role. In case of other roles, the call will return NOT_IMPLEMENTED status.

3.5.6.4. Get local device Installcode

SECUR_GET_LOCAL_IC

Command id	Parameters	Data	Blocking?	Description
0x050d	No	No	No	Get local device Installcode

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7	8, 10, 14 or 18
Common response header	Installcode

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
Installcode	u8arr_t	8, 10, 14 or 18	Installcode, including two trailing bytes of CRC

Status codes:

Status category	Status code	Description
GENERIC	OK	Indicates success
GENERIC	INVALID_FORMAT	Invalid packet format, payload length doesn't correspond to expected parameter's length
GENERIC	NOT_FOUND	There is no install code for the given parameters

3.5.6.5. TCLK Indication

SECUR_TCLK_IND

Command id	Parameters	Data	Blocking?	Description
0x050e	Yes	No	No	TCLK Indication

Indication packet format:

Octets: 4	8	1
Common HL header	TC address	Key type

Indication parameters:

Ind parameter name	Type	Length, bytes	Description
TC address	ieee_adddr_t	8	Trust Center Address
Key type	uint8_t	1	Key type

3.5.6.6. TCLK Exchange Indication Failed

SECUR_TCLK_EXCHANGE_FAILED_IND

Command id	Parameters	Data	Blocking?	Description
0x050f	Yes	No	No	TCLK Exchange Indication Failed

Indication packet format:

Octets: 4	1	1
Common HL header	Status category	Status code

Indication parameters:

Ind parameter name	Type	Length, bytes	Description
Status category	uint8_t	1	Status category
Status code	uint8_t	1	Status code

3.5.6.7. Initiate Key Switch Procedure

SECUR_NWK_INITIATE_KEY_SWITCH_PROCEDURE

Command id	Parameters	Data	Blocking?	Description
0x0517	No	No	No	Initiates a key switch procedure

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7
Common response header

3.5.6.8. Get IC List

SECUR_GET_IC_LIST

Command id	Parameters	Data	Blocking?	Description
0x0518	Yes	No	No	Gets the IC list

Request packet format:

Octets: 5	1
Common request header	Start Index

Request parameters:

Req parameter name	Type	Length, bytes	Description
Start Index	uint8_t	1	SoC will return IC Table entries starting with this index

Response packet format:

Octets: 7	1	1	1	variable
Common response header	IC Table Size	Start Index	Entry Count	IC Table Entries

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
IC Table Size	uint8_t	1	The total number of entries in the IC table
Start Index	uint8_t	1	
Entry Count	uint8_t	1	The number of entries in this response
IC Table Entries	u8arr_t	Entry Count * Entry Size	

IC Table Entry Format

Device Address: ieee_addr_t, 8 bytes

IC Type: uint8_t, 1 byte

IC: u8arr_t, length depends on IC Type: 8, 10, 14 or 18 bytes

3.5.6.9. Get IC by Index

SECUR_GET_IC_BY_IDX

Command id	Parameters	Data	Blocking?	Description
0x0519	Yes	No	No	Get an IC table entry by index

Request packet format:

Octets: 5	1
Common request header	IC Entry Index

Request parameters:

Req parameter name	Type	Length, bytes	Description
IC Entry Index	uint8_t	1	the index of the entry to get

Response packet format:

Octets: 7	variable
Common response header	IC Table entry

Response parameters:

Rsp parameter name	Type	Length, bytes	Description
IC Table entry	u8arr_t	Entry Size	

3.5.6.10. Remove All IC

SECUR_REMOVE_ALL_IC

Command id	Parameters	Data	Blocking?	Description
0x051a	No	No	No	Removes all IC

Request packet format:

Octets: 5
Common request header

Response packet format:

Octets: 7
Common response header