ZigBee PRO Network Processor

Accelerate your ZigBee Development

Applications

- *ZigBee™ systems*
- *Home/Building automation*
- *Industrial control and monitoring*
- *Low power wireless sensor networks*
- *Set-top boxes and remote controls*
- *Automated Meter Reading*

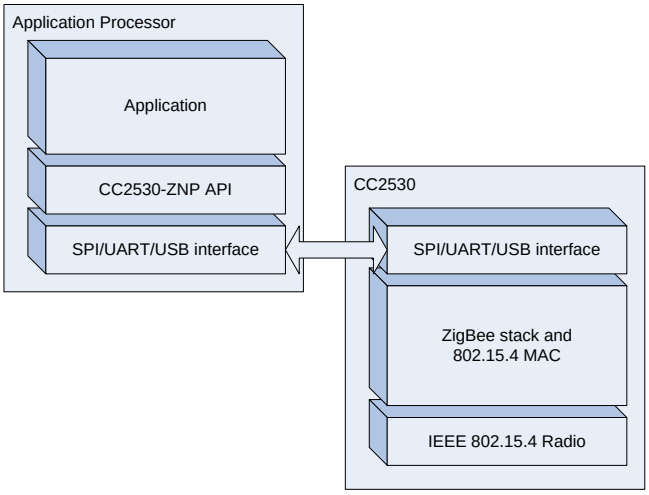
Description

The *CC2530-ZNP* is a cost-effective, low power, ZigBee Processor that provides full ZigBee functionality with a minimal development effort.

In this solution, the ZigBee PRO stack runs on a CC2530 ZigBee SoC and the application runs on an external microcontroller. The *CC2530-ZNP* handles all the ZigBee protocol tasks, and leaves the resources of the application microcontroller free to handle the application.

This makes it easy for users to add ZigBee to new or existing products at the same time as it provides great flexibility in choice of microcontroller.

CC2530-ZNP interfaces to any microcontroller through an SPI, UART or USB interface. For example, it can be combined with an MSP430 or Stellaris ARM Cortex-M3 microcontroller.



Key Features

- *All the powerful features of the ZigBee PRO system-on-chip with a simplified application interface.*
- *SPI, UART or USB interface to application processor with SPI speeds up to 4 MHz.*
- *Designed for low power operation when using SPI interface with maximum time spent in low power mode when using SPI interface.*
- *Access to 12-bit analog-to-digital converter, GPIO pins, non-volatile memory*

References

[R1] CC253x User Guide. http://www.ti.com/litv/pdf/swru191

[R2] CC2530 Datasheet. http://www.ti.com/lit/gpn/cc2530

[R3] CC2531 Datasheet. http://www.ti.com/lit/gpn/cc2531

[R4] CC259x Datasheet. http://www.ti.com/lit/gpn/cc2591

Acronyms

ADC	Analog to Digital Conversion (or Converter)
AF	ZigBee Application Framework
API	Application Programming Interface
AREQ	Asynchronous Request
CTS	Clear To Send
FCS	Frame Check Sequence
GPIO	General Purpose I/O
NV	Non-Volatile
PA/LNA	Power Amplifier / Low Noise Amplifier (CC259x)
POLL	Poll request
RPC	Remote Procedure Call
RTS	Ready To Send
SAPI	Simple API
SoC	System on Chip
SOF	Start Of Frame
SPI	Serial Peripheral Interface bus
SREQ	Synchronous request
SRSP	Synchronous response
UART	Universal Asynchronous Receiver Transmitter
ZDO	ZigBee Device Object
ZNP	ZigBee Network Processor

1 Pin configuration

The figure below shows how an application processor interfaces with the CC2530.

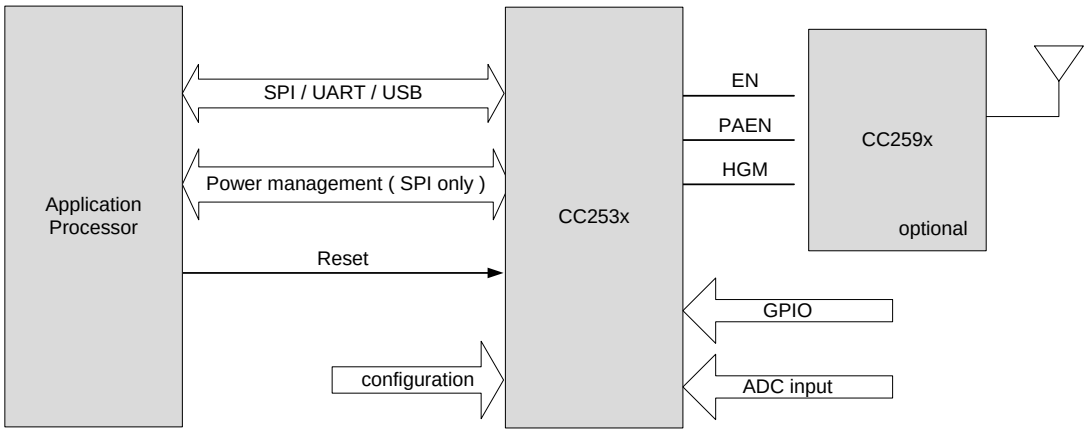


Figure 1 CC2530 Interface

1.1 Network processor signals

The CC2530-ZNP uses the following signals for the hardware interface

- **MI/MO/C/SS for SPI, and RX/TX/RT/CT for UART:** These are the standard signals used for SPI or UART communication. See sections 2.2.3 (for SPI) and 2.3.3 (for UART) for details. This naming convention is consistent with [R1].
- **SRDY:** This signal is asserted by the CC2530 for power management and transaction control when using SPI transport. The application processor can use a regular GPIO pin to poll the status of this signal, or connect it to a GPIO with edge configurable interrupt capability. See section 2.2.3 for details.
- **MRDY:** This signal is asserted by the application processor for power management and transaction control when using SPI transport. This is typically hardwired to the SS pin and does not have to be controlled by a separate GPIO from the application processor. See section 2.2.3 for details.
- **RESET:** This signal is used by the application processor to reset the CC2530.
- **PAEN, EN, HGM:** These signals are used to control the CC259x PA/LNA and should be connected to the appropriate pins on the CC259x. See [R4] for details on the CC259x.
- **CFG0, CFG1:** These two signals are used to configure the CC2530-ZNP. The CC2530-ZNP reads these signals at power up and configures its operation accordingly. See section 2.1.2 for details.
- **GPIO0-3:** These pins can be configured as general purpose I/O or, for some pins, as ADC inputs. See section 4.1.7 for details.

2 CC2530-ZNP physical Interface

The CC2530-ZNP supports SPI, UART, or USB interface to the application processor.

2.1 CC2530-ZNP default configuration

2.1.1 IAR project configuration

The CC2530-ZNP IAR project that is included in the ZStack software package has two project configurations – CC2530-ZNP and CC2531-ZNP. As the name indicates, the configurations are intended for use with the CC2530 and CC2531 (USB) chips.

2.1.2 Configuration pins

The CC2530-ZNP project reads the two hardware configuration pins at powerup and configures itself accordingly.

The CFG0 pin is used to indicate the presence (if pin is high) or absence of the 32kHz crystal connected to the CC2530-ZNP. This is the sleep crystal that is used to maintain accurate timing when the device is in sleep mode. The advantage of using this instead of the internal 32kHz oscillator is that it typically provides faster wakeup time for sleep and a lower power consumption during this time. If this crystal is not populated, then the CC2530 can use the internal RC oscillator.

If the CFG1 pin is high, the CC2530-ZNP will use the SPI transport mode in the main pin configuration listed below. Otherwise, it will use the UART transport mode in the alternate pin configuration listed below. The ZNP Kit pin configuration is used by the ZNP kit target board. The pin-out diagram of the CC2530 can be found in [R2].

2.1.2.1 Main pin configuration

CC2530-ZNP signal	CC2530 PIN	CC2530 NAME	*Direction (on C2530)*
SS / CT	6	P1_4	*In*
C / RT	5	P1_5	*In / Out*
MO / TX	38	P1_6	*In / Out*
MI / RX	37	P1_7	*Out / In*
RESET	20	RESET_N	*In*
MRDY	16	P0_3	*In*
SRDY	15	P0_4	*Out*
PAEN	9	P1_1	*Out*
EN	7	P1_3	*Out*
HGM	12	P0_7	*Out*
CFG0	8	P1_2	*In*
CFG1	36	P2_0	*In*
GPIO0/AIN0	19	P0_0	*Configurable*
GPIO1/AIN1	18	P0_1	*Configurable*
GPIO2	13	P0_6	*Configurable*
GPIO3	*11*	*P1_0*	*Configurable*

2.1.2.2 Alternate pin configuration

CC2530-ZNP signal	CC2530 PIN	CC2530 NAME	*Direction (on C2530)*
SS / CT	15	P0_4	*In*
C / RT	14	P0_5	*In / Out*
MO / TX	16	P0_3	*In / Out*
MI / RX	17	P0_2	*Out / In*
RESET	20	RESET_N	*In*
MRDY	38	P1_6	*In*
SRDY	37	P1_7	*Out*
PAEN	9	P1_1	*Out*
EN	6	P1_4	*Out*
HGM	12	P0_7	*Out*
CFG0	8	P1_2	*In*
CFG1	36	P2_0	*In*
GPIO0/AIN0	19	P0_0	*Configurable*
GPIO1/AIN1	18	P0_1	*Configurable*
GPIO2	13	P0_6	*Configurable*
GPIO3	*11*	*P1_0*	*Configurable*

2.1.2.3 ZNP Kit pin configuration

CC2530-ZNP signal	CC2530 PIN	CC2530 NAME	*Direction (on C2530)*
SS / CT	15	P0_4	*In*
C / RT	14	P0_5	*In / Out*
MO / TX	16	P0_3	*In / Out*
MI / RX	17	P0_2	*Out / In*
RESET	20	RESET_N	*In*
MRDY	36	P2_0	*In*
SRDY	11	P1_0	*Out*
PAEN	9	P1_1	*Out*
EN	6	P1_4	*Out*
HGM	12	P0_7	*Out*
CFG0	19	P0_0	*In*
CFG1	18	P0_1	*In*
GPIO0	13	P0_6	*Configurable*
GPIO1	12	P0_7	*Configurable*
GPIO2	38	P1_6	*Configurable*
GPIO3	*37*	*P1_7*	*Configurable*

2.1.2.4 USB pin configuration

This is only available when used with the CC2531 chip. In this configuration, the CC2530-ZNP will use the USB transport with the alternate pin configuration. The pin-out of the CC2531 can be found in the datasheet [R3]. The USB transport exposes the CDC (communication device class) class USB interface and exposes a virtual COM port to the host. The host processor would then access this device as a regular COM port device and communicate with the ZNP using the UART Transport.

2.2 SPI Transport

2.2.1 Configuration

The following SPI configuration is supported:

- SPI slave.
- Clock speed up to 4 MHz.
- Clock polarity 0 and clock phase 0 on CC2530.
- Bit order MSB first.

2.2.2 Frame Format

SPI transport uses the general frame format described in 2.4.

2.2.3 Signal Description

The following standard SPI signals are used:

- C: Serial clock.
- SS: Slave select.
- MO: Master-output slave-input data.
- MI: Master-input slave-output data.

Two additional signals are required for SPI transaction handling and power management:

- MRDY: Master ready, an active low signal. This signal is set by the application processor when it has data ready to send to the CC2530. This signal can either be controlled independently or it can be hardwired to the slave select signal. The RPC sequence diagrams in this document assume MRDY is hardwired to SS.
- SRDY: Slave ready, a bi-modal signal. This signal is set by the CC2530 when it is ready to receive or send data. When set low, it indicates the CC2530 is ready to receive data. When set high during an SPI POLL or SREQ transaction it indicates the CC2530 is ready to send data. When set high during an SPI AREQ transaction it indicates the CC2530 is done receiving data.

2.2.4 Signal Operation

The signals operate according to the following rules:

1. The application processor initiates a transaction by setting MRDY low and then waits for SRDY to go low.
2. The application processor shall never set MRDY high to end a transaction before all bytes of the frame have been transferred.
3. When receiving a POLL or SREQ, the CC2530 shall set SRDY high when it has data ready for the application processor.
4. When receiving an AREQ, the CC2530 shall set SRDY high when all bytes of the frame have been received.

2.2.5 Protocol Scenarios

2.2.5.1 AREQ Command

The following figure shows an AREQ command sent from the application processor to the CC2530.

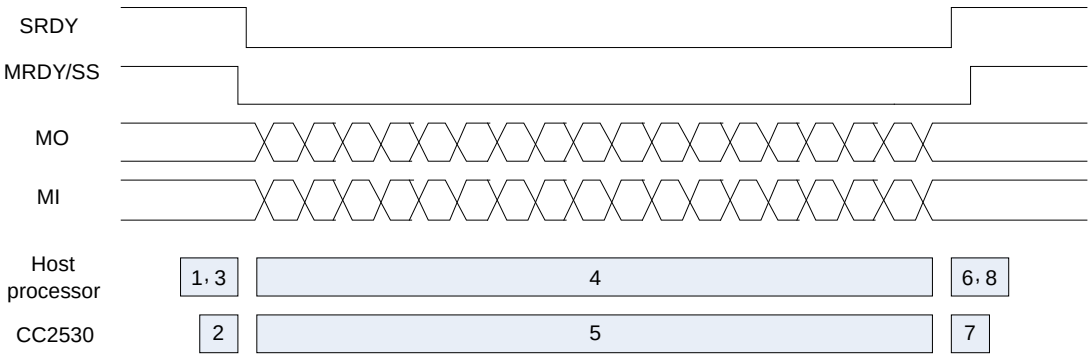


Figure 2 AREQ Command

The following sequence of events occurs on the application processor and CC2530:

1. Application processor has an AREQ frame to send. Set MRDY low and wait for SRDY to go low.
2. CC2530 receives falling edge of MRDY. When ready to receive data set SRDY low.
3. Application processor reads SRDY low. Start data transmission.
4. Application processor transmits data until frame is complete.
5. CC2530 receives data until frame is complete.
6. Application processor waits for SRDY to go high.
7. CC2530 receives complete frame and sets SRDY high.
8. Application processor reads SRDY high. Set MRDY high.

2.2.5.2 POLL Command

The following figure shows a POLL command sent from the application processor to the CC2530-ZNP.

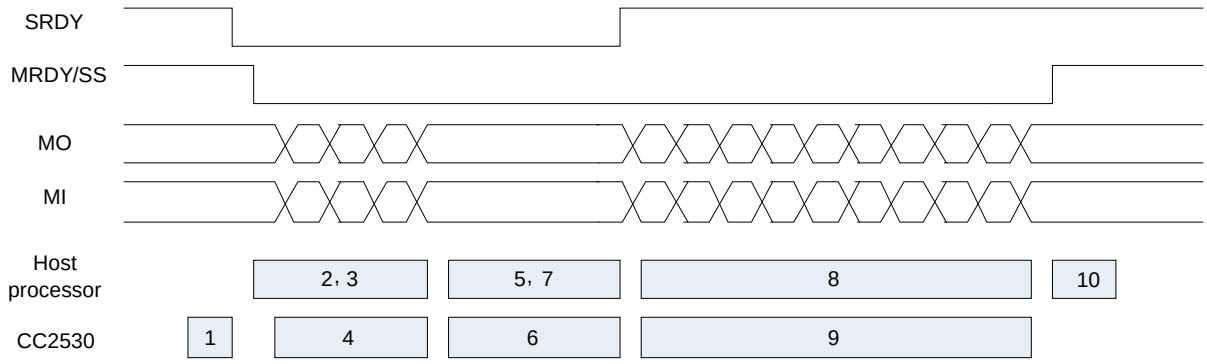


Figure 3 POLL command

The following sequence of events occurs on the application processor and CC2530:

1. CC2530 has an AREQ frame to send. When ready to receive data set SRDY low.
2. Application processor detects SRDY low and sets MRDY low. Prepare POLL command and start data transmission.
3. Application processor transmits data until frame is complete.
4. CC2530 receives data until frame is complete.
5. Application processor waits for SRDY to go high.
6. CC2530 prepares AREQ frame for transmission. When ready to transmit set SRDY high.
7. Application processor reads SRDY high. Start data reception.
8. Application processor receives data until frame is complete.
9. CC2530 transmits data until frame is complete.
10. Application processor receives complete frame. Set MRDY high.

2.2.5.3 SREQ Command

The following figure shows a SREQ command sent from the application processor to the CC2530-ZNP.

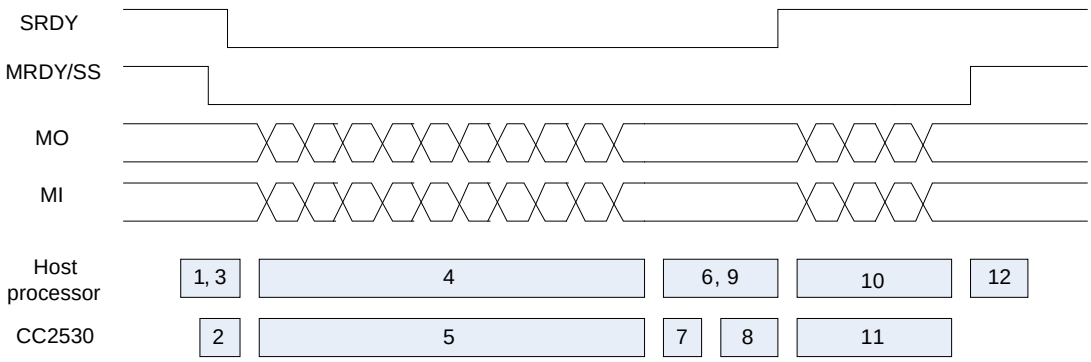


Figure 4 SREQ command

The following sequence of events occurs on the application processor and CC2530-ZNP:

1. Application processor has an SREQ frame to send. Set MRDY low and wait for SRDY to go low.
2. CC2530 receives falling edge of MRDY. When ready to receive data set SRDY low.
3. Application processor reads SRDY low. Start data transmission.
4. Application processor transmits data until frame is complete.
5. CC2530 receives data until frame is complete.
6. Application processor waits for SRDY to go high.
7. CC2530 processes SREQ command and executes function
8. CC2530 prepares SRSP frame. When ready to transmit data set SRDY high.
9. Application processor reads SRDY high. Start data reception.
10. Application processor receives data until frame is complete.
11. CC2530 transmits data until frame is complete.
12. Application processor receives complete frame. Set MRDY high.

2.3 UART Transport

2.3.1 Configuration

The following UART configuration is supported:

- Baud rate: 115200
- Hardware (RTS/CTS) flow control.
- 8-N-1 byte format.

2.3.2 Frame Format

UART transport frame format is shown in the following figure. The left-most field is transmitted first over the wire.

Bytes: 1	3-253	1
SOF	General format frame	FCS

Figure 5 UART Transport Frame Format

SOF: Start of frame indicator. This is always set to 0xFE.

General frame format: This is the general frame format as described in 2.4.

FCS: Frame-check sequence. This field is computed as an XOR of all the bytes in the general format frame fields.

Shown below is a C example for the FCS calculation:

```
unsigned char calcFCS(unsigned char *pMsg, unsigned char len)
{
  unsigned char result = 0;
  while (len--)
  {
    result ^= *pMsg++;
  }
  return result;
}
```

2.3.3 Signal Description

The following standard UART signals are used:

- TX: Transmit data.
- RX: Receive data.
- CT: Clear to send.
- RT: Ready to send.
- The MRDY and SRDY signals are not used with UART transport.

Figure 6 shows the RTS/CTS flow control connections to the host processor. On the CC2530, RT and CT are active-low signals. The RT output is driven low when the receive register is empty and reception is enabled. Transmission of a byte does not occur before the CT input goes low.

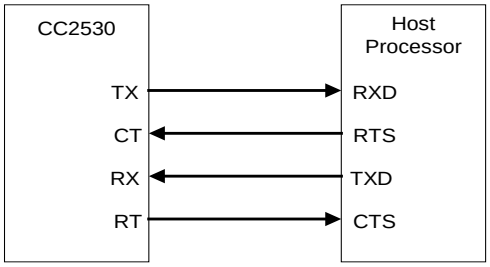


Figure 6 RTS/CTS Flow Control Connections

2.3.4 Signal Operation

UART transport sends and receives data asynchronously. Data can be sent and received simultaneously and the transfer of a frame can be initiated at any time by either the application processor or the CC2530.

2.4 General Frame Format

The general frame format is shown in the following figure. The left-most field is transmitted first over the wire. For multi-byte fields, the lowest order byte is transmitted first.

Bytes: 1	**2**	**0-250**
Length	**Command**	**Data**

Figure 7 General Frame Format

Length: The length of the data field of the frame. The length can range from 0-250.

Command: The command of the frame.

Data: The frame data. This depends on the command field and is described for each command in Section 4.

2.4.1 Command Field

The command field is constructed of two bytes. The bytes are formatted as shown in the following figure. The Cmd0 byte is transmitted first.

Cmd0		**Cmd1**
Bits: 7-5	**4-0**	**7-0**
Type	**Subsystem**	**ID**

Figure 8 Command Field

Type: The command type has one of the following values:

- 0: POLL. A POLL command is used to retrieve queued data. This command is only applicable to SPI transport. For a POLL command the subsystem and ID are set to zero and data length is zero.
- 1: SREQ: A synchronous request that requires an immediate response. For example, a function call with a return value would use an SREQ command.
- 2: AREQ: An asynchronous request. For example, a callback event or a function call with no return value would use an AREQ command.
- 3: SRSP: A synchronous response. This type of command is only sent in response to a SREQ command. For an SRSP command the subsystem and ID are set to the same values as the corresponding SREQ. The length of an SRSP is generally nonzero, so an SRSP with length=0 can be used to indicate an error.
- 4-7: Reserved.

Subsystem: The subsystem of the command. Values are shown below:

Subsystem Value	Subsystem Name
0	RPC Error interface
1	SYS interface
2	Reserved
3	Reserved
4	AF interface
5	ZDO interface
6	Simple API interface
7	UTIL interface
8-32	Reserved

ID: The command ID. The ID maps to a particular interface message. Value range: 0-255.

When the ZNP cannot recognize an SREQ command from the host processor, the following SRSP is returned:

SRSP:

1	1	1	1	1	1
Length = 0x03	Cmd0 = 0x60	Cmd1 = 0x00	ErrorCode	ReqCmd0	ReqCmd1

Attributes:

<table>
<tr><th>Attribute</th><th>Length (byte)</th><th>Description</th></tr>
<tr><td>ErrorCode</td><td>1</td><td>The error code maps to one of the following enumerated values.

<table><tr><th>Value</th><th>Description</th></tr><tr><td>0x01</td><td>Invalid subsystem</td></tr><tr><td>0x02</td><td>Invalid command ID</td></tr><tr><td>0x03</td><td>Invalid parameter</td></tr><tr><td>0x04</td><td>Invalid length</td></tr></table></td></tr>
<tr><td>ReqCmd0</td><td>1</td><td>The Cmd0 value of the processed SREQ</td></tr>
<tr><td>ReqCmd1</td><td>1</td><td>The Cmd1 value of the processed SREQ</td></tr>
</table>

3 Initialization Procedures

3.1 CC2530-ZNP power-up procedure

The recommended power-up procedure is as follows:

1. Application processor and CC2530 power up.
2. Application processor sets CC2530 RESET_N pin low, holding CC2530 in reset.
3. The application processor sets the optional CC2530 CFG0 and CFG1 pins (if these pins are controlled by the application processor).
4. The application processor initializes its UART or SPI interface.
5. The application processor sets CC2530 RESET_N pin high and CC2530 starts operation.
6. Application processor receives the SYS_RESET_IND message using the POLL command. When SPI transport is used CC2530 will set SRDY low to indicate the message is available and the application processor should retrieve the message.
7. The application processor receives the `SYS_RESET_IND` message.

If the CC2530-ZNP device was configured as an end-device (and using SPI transport), it will automatically enter low power state after the application processor retrieves the `SYS_RESET_IND` command from the CC2530.

The CC2530-ZNP can also be reset when the application processor sends a `SYS_RESET_REQ` message. However, resetting CC2530 with the RESET_N pin is recommended because it is faster and more reliable.

3.2 CC2530-ZNP startup procedure

After executing the power-up procedure, the host processor must call some mandatory APIs before executing any APIs that invoke ZigBee over-the-air messaging. Not following this sequence could result in unexpected behaviour. The recommended startup procedure is as follows:

1. The host processor must use the ZB_WRITE_CONFIGURATION command to configure at the minimum the ZCD_NV_LOGICAL_TYPE, ZCD_NV_PAN_ID, and ZCD_NV_CHANLIST configuration items.
2. If the Simple API is used, the ZB_APP_REGISTER_REQUEST command should be sent by the host processor to register the application endpoint.
3. The ZB_START_REQUEST command should be sent by the host processor to either form a network (if the device is a coordinator) or join a network (if the device is a router or end device).
4. The host processor should then wait for the ZB_START_CONFIRM command with a status of ZB_SUCCESS before performing any other API operations.
5. If the Simple API is not used after performing step 1, the AF_REGISTER command should be sent by the host processor to register the application endpoint.
6. The ZDO_STARTUP_FROM_APP command should be sent by the host processor to either form a network (if the device is a coordinator) or join a network (if the device is a router or end device).
7. The host processor should then wait for the ZDO_STATE_CHANGE_IND command with a status of DEV_ZB_COORD, DEV_ROUTER, or DEV_END_DEVICE before performing any other API operations.

4 CC2530-ZNP software command interface

The following subsections describe the CC2530 software command interface. They are sub-divided into the following categories

- The SYS interface provides the application processor with a low level interface to the CC2530 hardware and software. The CC2530 functions that are accessible over this interface include the ADC (analog-to-digital converter), NV memory, GPIO pins and the hardware random number generator.
- The Configuration interface allows the application processor to configure various parameters of the CC2530 device.
- The Simple API interface is a simplified ZigBee interface that can be used to quickly create simple ZigBee compliant networked applications. It allows for easy device configuration, network formation, binding and data transfer. However, a limitation of the Simple API is that it can only be used with one application registered endpoint. Therefore, it is recommended that applications that support multiple endpoints use the AF interface.
- The AF and ZDO interfaces feature the complete ZigBee interface and can be used to create a full range of ZigBee compliant applications. The AF (Application Framework) interface allows the application processor to register its application with the CC2530 and send and receive data. The ZDO (ZigBee Device Object) interface provides various ZigBee management functions like device and service discovery.

In all the message formats shown below, the left-most field is transmitted first over the wire. For multi-byte fields, the lowest order byte is transmitted first.

4.1 SYS interface

4.1.1 SYS_RESET_REQ

4.1.1.1 Description

This command is issued by the application processor to reset the CC2530 device. The reset is achieved through an internal watchdog reset on the CC2530. Note that the hardware reset interface is recommended over using this interface.

4.1.1.2 Usage

AREQ:

1	1	1	*1*
Length = 0x01	*Cmd0 = 0x41*	*Cmd1 = 0x00*	*Type*

Type – 1 byte – This requests a target device reset (0) or serial bootloader reset (1). If the target device does not support serial bootloading, bootloader reset commands are ignored and no response is sent from the target.

4.1.2 SYS_RESET_IND

4.1.2.1 Description

This command is generated by the CC2530 device automatically immediately after a reset.

4.1.2.2 Usage

AREQ:

1	1	1	1	1	*1*
Length = 0x06	*Cmd0 = 0x41*	*Cmd1 = 0x80*	*Reason*	*TransportRev*	*ProductId*

1	1	*1*
MajorRel	*MinorRel*	*HwRev*

Reason – 1 byte – One of the following values indicating the reason for the reset.

Resolution	*Value*
Power-up	*0x00*
External	*0x01*
Watch-dog	*0x02*

TransportRev – 1 byte – Transport protocol revision. This is set to value of 2.

Product – 1 byte – Product ID. This is set to value of 1.

MajorRel – 1 byte – Major release number.

MinorRel – 1 byte – Minor release number.

HwRev – 1 byte – Hardware revision number.

4.1.3 SYS_VERSION

4.1.3.1 Description

This command is issued by the application processor to request for the CC2530 software version information.

4.1.3.2 Usage

SREQ:

1	1	1
Length = 0x00	*Cmd0 = 0x21*	*Cmd1 = 0x02*

SRSP:

1	1	1	1	1	1	1	1
Length = 0x05	*Cmd0 = 0x61*	*Cmd1 = 0x02*	*TransportRev*	*Product*	*MajorRel*	*MinorRel*	*MaintRel*

TransportRev – 1 byte – The transport protocol revision number. This is set to value of 2.

Product – 1 byte – Product ID. This is set to value of 0.

MajorRel – 1 byte – Software major release number.

MinorRel – 1 byte – Software minor release number.

MaintRel – 1 byte – Software maintenance release number.

4.1.4 SYS_OSAL_NV_READ

4.1.4.1 Description

This command is used by the application processor to read data values from an item stored in the CC2530 NV memory.

4.1.4.2 Usage

SREQ:

t	1	1	2	*1*
Length = 0x03	*Cmd0 = 0x21*	*Cmd1 = 0x08*	*Id*	*Offset*

General purpose NV items (pre-defined) that are available for ZNP applications:

NV Item	Size	*Value*
ZNP_NV_APP_ITEM_1	2 bytes	*0x0F01*
ZNP_NV_APP_ITEM_2	2 bytes	*0x0F02*
ZNP_NV_APP_ITEM_3	2 bytes	*0x0F03*
ZNP_NV_APP_ITEM_4	2 bytes	*0x0F04*
ZNP_NV_APP_ITEM_5	16 bytes	*0x0F05*
ZNP_NV_APP_ITEM_6	*16 bytes*	*0x0F06*

Id – 2 bytes – The attribute id of the NV item.

Offset – 1 byte – Number of bytes offset from the beginning of the NV value.

SRSP:

1	1	1	1	1	*0-248*
Length = 0x02-0xFA	*Cmd0 = 0x61*	*Cmd1 = 0x08*	*Status*	*Len*	*Value*

Status – 1 byte – See 4.7 for a listing of the status values.

Len – 1 byte – The length of the NV value (up to 248).

Value – 0-248 bytes – Data values of the NV item.

4.1.5 SYS_OSAL_NV_WRITE

4.1.5.1 Description

This command is used by the application processor to write data values to an item in the CC2530 NV memory..

4.1.5.2 Usage

SREQ:

1	1	1	2	1	1	*0-246*
Length = 0x04-0xFA	*Cmd0 = 0x21*	*Cmd1 = 0x09*	*Id*	*Offset*	*Len*	*Value*

General purpose NV items (pre-defined) that are available for ZNP applications:

NV Item	Size	*Value*
ZNP_NV_APP_ITEM_1	2 bytes	*0x0F01*
ZNP_NV_APP_ITEM_2	2 bytes	*0x0F02*
ZNP_NV_APP_ITEM_3	2 bytes	*0x0F03*
ZNP_NV_APP_ITEM_4	2 bytes	*0x0F04*
ZNP_NV_APP_ITEM_5	16 bytes	*0x0F05*
ZNP_NV_APP_ITEM_6	*16 bytes*	*0x0F06*

Id – 2 bytes – The attribute id of the NV item.

Offset – 1 byte - Number of bytes offset from the beginning of the NV value.

Len – 1 byte – Length of the NV value (up to 246).

Value – 0-246 bytes – Data values of the NV item.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x61	Cmd1 = 0x09	Status

Status – 1 byte – See 4.7 for a listing of the status values.

4.1.6 SYS_OSAL_NV_ITEM_INIT

4.1.6.1 Description

This command is used by the application processor to create and initialize an item in the CC2530 NV memory. The NV item will be created if it does not already exist. The data for the new NV item will be left uninitialized if the *InitLen* parameter is zero. When *InitLen* is non-zero, the data for the NV item will be initialized (starting at offset of zero) with the values from *InitData*. Note that it is not necessary to initialize the entire NV item (*InitLen* < *ItemLen*). It is also possible to create an NV item that is larger than the maximum length *InitData* – use the SYS_OSAL_NV_WRITE command to finish the initialization.

4.1.6.2 Usage

SREQ:

1	1	1	2	2	1	*0-245*
Length = 0x05-0xFA	*Cmd0 = 0x21*	*Cmd1 = 0x07*	*Id*	*ItemLen*	*InitLen*	*InitData*

Id – 2 bytes – The attribute id of the NV item.

ItemLen – 2 bytes - Number of bytes in the NV item.

InitLen – 1 byte – Number of bytes in the initialization data (up to 245).

InitData – 0-245 bytes – Values of the initialization data.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x61	Cmd1 = 0x07	Status

Status – 1 byte – See 4.7 for a listing of the status values.

4.1.7 SYS_OSAL_NV_DELETE

4.1.7.1 Description

This command is used by the application processor to delete an item from the CC2530 NV memory. The *ItemLen* parameter must match the length of the NV item or the command will fail. Use this command with caution – deleted items cannot be recovered.

4.1.7.2 Usage

SREQ:

1	1	1	2	*2*
Length = 0x04	*Cmd0 = 0x21*	*Cmd1 = 0x12*	*Id*	*ItemLen*

Id – 2 bytes – The attribute id of the NV item.

ItemLen – 2 bytes - Number of bytes in the NV item.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x61	Cmd1 = 0x12	Status

Status – 1 byte – See 4.7 for a listing of the status values.

4.1.8 SYS_OSAL_NV_LENGTH

4.1.8.1 Description

This command is used by the application processor to get the length of an item in the CC2530 NV memory. A length of zero is returned if the NV item does not exist.

4.1.8.2 Usage

SREQ:

1	1	1	*2*
Length = 0x02	*Cmd0 = 0x21*	*Cmd1 = 0x13*	*Id*

Id – 2 bytes – The attribute id of the NV item.

SRSP:

1	1	1	2
Length = 0x02	Cmd0 = 0x61	Cmd1 = 0x13	ItemLen

ItemLen – 2 bytes – Length of item in NV memory.

4.1.9 SYS_ADC_READ

4.1.9.1 Description

This command is used by the application processor to read from the CC2530-ZNP ADC (analog-to-digital converter).

4.1.9.2 Usage

SREQ:

1	1	1	1	*1*
Length = 0x02	*Cmd0 = 0x21*	*Cmd1 = 0x0D*	*Channel*	*Resolution*

Channel – 1 byte – The following channels are available.

Channel	*Value*
AIN0	*0x06*
AIN1	*0x07*
AIN0-1 (differential input)	*0x0B*
Temperature Sensor	*0x0E*
Voltage Reading	*0x0F*

Resolution – 1byte – The resolution of the ADC conversion. It can be 7-bit, 9-bit, 10-bit or 12-bit.

Resolution	*Value*
7-bit	*0x00*
9-bit	*0x01*
10-bit	*0x02*
12-bit	*0x03*

SRSP:

1	1	1	*2*
Length = 0x02	*Cmd0 = 0x61*	*Cmd1 = 0x0D*	*Value*

Value – 2 bytes – Value of the ADC conversion based on the specified information. This is a signed value in two's complement representation. Depending on the resolution of the conversion, the appropriate number of lowest order bits should be ignored. For example, for a 7-bit resolution, the lowest 9 bits should be ignored.

4.1.10 SYS_GPIO

4.1.10.1 Description

This command is used by the application processor to configure the accessible GPIO pins on the CC2530-ZNP device. There are four accessible GPIO pins (GPIO0-3) on the CC2530-ZNP device.

4.1.10.2 Usage

SREQ:

1	1	1	1	*1*
Length = 0x02	*Cmd0 = 0x21*	*Cmd1 = 0x0E*	*Operation*	*Value*

Operation - 1 byte - The type of operation to perform on the GPIO pins. It can take following values:

Operation	Value	*Description*
Set direction	0x00	*Configures the direction of the GPIO pins. A value of 0 in a bit position configures the corresponding GPIO pin as an Input while a value of 1 configures it as Output.*
Set Input mode	0x01	*Configures the Input mode of the GPIO pins. A value of 1 in a bit position configures the corresponding GPIO into a tri-state mode. Otherwise, the corresponding bit in the higher-order nibble is examined and a 1 in that position configures the GPIO as a pull-down while a 0 configures it as pull-up.*
Set	0x02	*A value of 1 in a bit position will set the corresponding GPIO pin (writes a 1).*
Clear	0x03	*A value of 1 n a bit position will clear the corresponding GPIO pin (writes a 0).*
Toggle	0x04	*A value of 1 in a bit position will toggle the corresponding GPIO pin.*
Read	*0x05*	*Reads the GPIO pins.*

Value – 1 byte – Each bit position in this field contains the requested value for one of the four GPIO pins. Note that only the four lower order bits are used except when the requested operation is a “set input mode”, in which case the whole byte is used.

SRSP:

1	1	1	*1*
Length = 0x01	*Cmd0 = 0x61*	*Cmd1 = 0x0E*	*Value*

Value – 1 byte – The value after the requested operation is performed.

Note: When the device is in sleep mode, GPIO pins maintain their output value and I/O mode.

4.1.11 SYS_RANDOM

4.1.11.1 Description

This command is used to get a 16-bit random number.

4.1.11.2 Usage

SREQ**:**

1	1	*1*
Length = 0x00	*Cmd0 = 0x21*	*Cmd1 = 0x0C*

SRSP:

1	1	1	*2*
Length = 0x02	*Cmd0 = 0x61*	*Cmd1 = 0x0C*	*Value*

Value – 2 bytes – The random value.

4.1.12 SYS_SET_TIME

4.1.12.1 Description

This command is used by the application processor to set the ZNP device date and time. Time is specified in “seconds since 00:00:00 on January 1, 2000” or in parsed date/time components.

4.1.12.2 Usage

SREQ:

1	1	1	4	1	1	1	1	1	2
Length = 0x0B	*Cmd0 = 0x21*	*Cmd1 = 0x10*	*UTCTime*	*Hour*	*Minute*	*Second*	*Month*	*Day*	*Year*

UTCTime – 4 bytes – Number of seconds since 00:00:00 on January 1, 2000. Set this parameter to zero to use the parsed Hour-Minute-Second-Month-Day-Year parameters.

Hour – 1 byte – Hour of the day (0-23).

Minute – 1 byte – Minute of the hour (0-59).

Second – 1 byte – Second of the minute (0-59).

Month – 1 byte – Month of the year (1-12).

Day – 1 byte – Day of the month (1-31).

Year – 2 bytes – Year (2000 -)

SRSP:

1	1	1	1
Length = 0x01	*Cmd0 = 0x61*	*Cmd1 = 0x10*	*Status*

Status – 1 byte – See section 4.7 for a listing of the status values.

4.1.13 SYS_GET_TIME

4.1.13.1 Description

This command is used by the application processor to get the ZNP system date and time. Time is returned in “seconds since 00:00:00 on January 1, 2000” and parsed date/time components.

4.1.13.2 Usage

SREQ:

1	1	*1*
Length = 0x00	*Cmd0 = 0x21*	*Cmd1 = 0x11*

SRSP:

1	1	1	4	1	1	1	1	1	*2*
Length = 0x0B	*Cmd0 = 0x61*	*Cmd1 = 0x11*	*UTCTime*	*Hour*	*Minute*	*Second*	*Month*	*Day*	*Year*

UTCTime – 4 bytes – Number of seconds since 00:00:00 on January 1, 2000.

Hour – 1 byte – Hour of the day (0-23).

Minute – 1 byte – Minute of the hour (0-59).

Second – 1 byte – Second of the minute (0-59).

Month – 1 byte – Month of the year (1-12).

Day – 1 byte – Day of the month (1-31).

Year – 2 bytes – Year (2000 -)

4.1.14 SYS_SET_TX_POWER

4.1.14.1 Description

This command is used by the application processor to set the ZNP radio transmit power. The returned TX power is the actual setting applied to the radio – nearest characterized value for the specific radio and PA/LNA (if used).

4.1.14.2 Usage

SREQ:

1	1	*1*	*1*
Length = 0x01	*Cmd0 = 0x21*	*Cmd1 = 0x14*	*TX Power*

TX Power – 1 byte – Requested TX power setting, in dBm.

SRSP:

1	1	1	1
Length = 0x01	*Cmd0 = 0x61*	*Cmd1 = 0x14*	*TX Power*

TX Power – 1 byte – Actual TX power setting, in dBm.

4.2 Configuration interface

The CC2530-ZNP device has numerous parameters that can be configured by the application processor. These configuration parameters are stored in non volatile memory on the CC2530-ZNP device and their values persist across a device reset.

The configuration parameters are divided into “network-specific” and “device-specific” parameters. The “network-specific” configuration parameters should be set to the same value for all CC2530-ZNP devices in a ZigBee network to ensure proper network operation. The “device-specific” parameters can be set to different values on each device. These parameters are listed in detail in sections 4.2.3 and 4.2.4.

When the CC2530-ZNP device powers up, it reads two of the configuration parameters immediately. These are the `STARTOPT_CLEAR_CONFIG` bit (part of the `ZCD_NV_STARTUP_OPTION` parameter) and the `ZCD_NV_LOGICAL_TYPE` parameters. Any modification of these parameters will require a CC2530-ZNP device reset before they can take effect.

The rest of the configuration parameters are read when the CC2530-ZNP device starts operation of the ZigBee stack (when the `ZB_START_REQUEST` is issued).

Each of the configuration parameters has a default value that is used if it is not explicitly configured. It is possible to erase all the configuration settings and restore the device to this initial configuration by setting the `STARTOPT_CLEAR_CONFIG` bit option. This is useful if it is necessary to bring the CC2530-ZNP device configuration to a known state.

4.2.1 ZB_READ_CONFIGURATION

4.2.1.1 Description

This command is used to read the value of a configuration parameter from the CC2530-ZNP device.

4.2.1.2 Usage

SREQ:

1	1	1	*1*
Length = 0x01	*Cmd0 = 0x26*	*Cmd1 = 0x04*	*ConfigId*

ConfigId – 1 byte – Specifies the identifier for the configuration property.

SRSP:

1	1	1	1	1	1	*0-128*
Length = 0x03-0x83	*Cmd0 = 0x66*	*Cmd1 = 0x04*	*Status*	*ConfigId*	*Len*	*Value*

Status – 1 byte – See 4.7 for a listing of the status values.

ConfigId – 1 byte – Specifies the identifier for the configuration property.

Len – 1 byte – Specifies the size of the Value buffer in bytes.

Value – 0-128 bytes – A buffer to hold the configuration property.

4.2.2 ZB_WRITE_CONFIGURATION

4.2.2.1 Description

This command is used to write a configuration parameter to the CC2530-ZNP device.

4.2.2.2 Usage

SREQ:

1	1	1	1	1	*1-128*
Length = 0x03-0x83	*Cmd0 = 0x26*	*Cmd1 = 0x05*	*ConfigId*	*Len*	*Value*

ConfigId – 1 byte – The identifier for the configuration property

Len – 1 byte – Specifies the size of the Value buffer in bytes.

Value – 1-128 bytes – The buffer containing the new value of the configuration property

SRSP:

1	1	1	*1*
Length = 0x01	*Cmd0 = 0x66*	*Cmd1 = 0x05*	*Status*

Status – 1 byte – See 4.7 for a listing of the status values.

4.2.3 Device specific configuration parameters

4.2.3.1 ZCD_NV_STARTUP_OPTION

Configuration ID: 0x0003; Size: 1 byte; Default value: 0

This parameter controls the device startup options. This is a bit mask of the following values

Bit position	7-2	1	*0*
Description	*Reserved*	*STARTOPT_CLEAR_STATE*	*STARTOPT_CLEAR_CONFIG*

- STARTOPT_CLEAR_CONFIG – If this option is set, the device will overwrite all the configuration parameters (except this one) with the “default” values that it is programmed with. This is used to erase the existing configuration and bring the device into a known state.

  Note:
  The `STARTOPT_CLEAR_CONFIG` bit is read by the CC2530-ZNP device immediately when it powers up after a reset.
  When the configuration parameters are restored to defaults, the ZCD_NV_STARTUP_OPTION itself is not restored except for clearing the STARTOPT_CLEAR_CONFIG bit.

- STARTOPT_CLEAR_STATE – If this option is set, the device will clear its previous network state (which would exist if the device had been operating on a network prior to the reset). This is typically used during application development. During regular device operation, this flag is typically not set, so that an accidental device reset will not cause loss of network state.

Notes:
The CC2530-ZNP device has two kinds of information stored in non-volatile memory. The configuration parameters (listed in this section) and network state information.
The configuration parameters are configured by the user before start of ZigBee operation.
The network state information is collected by the device after it joins a network and creates bindings etc. (at runtime). This is not set by the application processor. This information is stored so that if the device were to reset accidentally, it can restore itself without going through all the network joining and binding process again.
If the application processor does not wish to continue operating in the previous ZigBee network, it needs to instruct the CC2530-ZNP device to clear the network state information and start again based on the configuration parameters. This is done by setting the STARTOPT_CLEAR_STATE bit in the startup option.

4.2.3.2 ZCD_NV_LOGICAL_TYPE

Configuration ID: 0x0087; Size: 1 byte; Default value: 0x00

This is the logical type of the device in the ZigBee network. This can be set to a COORDINATOR (0x00), ROUTER (0x01) or ENDDEVICE (0x02).

Note:
This parameter is read by the CC2530-ZNP device immediately when it powers up after a reset.

4.2.3.3 ZCD_NV_ZDO_DIRECT_CB

Configuration ID: 0x008F; Size: 1 byte; Default value: FALSE

This configures the manner in which ZDO responses (hereby referred to as callbacks) are issued to the host processor. By default, this item is set to FALSE, which means that the host processor must use the ZDO_MSG_CB_REGISTER command to subscribe to a specific ZDO callback in order to receive it. The ZDO callback is then conveyed as part of the ZDO_MSG_CB_INCOMING command. If ZCD_NV_ZDO_DIRECT_CB is set TRUE, then the host processor will receive the "verbose" response. For example, the host processor would receive the ZDO_IEEE_ADDR_RSP command in response to ZDO_IEEE_ADDR_REQ.

4.2.3.4 ZCD_NV_POLL_RATE

Configuration ID: 0x0024; Size: 2 bytes; Default value: 2000

If this parameter is set to a non-zero value, a CC2530-ZNP device that is configured as an end-device will wake up periodically with this duration to check for data with its parent device. This value is specified in milliseconds and can range from 1 to 65000.

If this parameter is set to zero, the device will not automatically wake up to poll for data. Instead, an external trigger or an internal event (for example, via a software timer event) can be used to wake up the device.

4.2.3.5 ZCD_NV_QUEUED_POLL_RATE

Configuration ID: 0x0025; Size: 2 bytes; Default value: 100

When an end-device polls for data with its parent and finds that it does have data, it can poll again with a shorter duration in case there is more data queued for it at its parent device. This value is specified in milliseconds. This feature can be turned off by setting this value to zero.

4.2.3.6 ZCD_NV_RESPONSE_POLL_RATE

Configuration ID: 0x0026; Size: 2 bytes; Default value: 100

When an end-device sends a data packet, it can poll again with a shorter duration, specified by this parameter, if the application is expecting to receive an application level packet in response. This value is specified in milliseconds. This feature can be turned off by setting the value to zero.

Note: The setting of the queued and response poll rates has to be done with caution if the device is sending and receiving at the same time or if the device is sending data too fast.

If the device is sending data too fast, setting a queued poll rate with a higher duration than the sending rate will cause the poll event to be continuously rescheduled to the future. Then the device will never poll for data with its parent and consequently it may miss any packets destined for it.

4.2.3.7 ZCD_NV_POLL_FAILURE_RETRIES

Configuration ID: 0x0029; Size: 1 byte; Default value: 2.

The number of times an end-device will fail when communicating with its parent before invoking the rejoin mechanism to find and join a new parent.

4.2.3.8 ZCD_NV_INDIRECT_MSG_TIMEOUT

Configuration ID: 0x002B; Size: 1 byte; Default value: 7

The amount of time (in seconds) that a router or coordinator device will buffer messages destined to their end-device child nodes. It is recommended that this is at least greater than the poll rate (ZCD_NV_POLL_RATE) to ensure that end-device will have a chance to wakeup and poll for the data.

4.2.3.9 ZCD_NV_APS_FRAME_RETRIES

Configuration ID: 0x0043; Size: 1 byte; Default value: 3

The number of retransmissions performed on a data packet at the application layer if the packet was transmitted with the end-to-end acknowledgement option enabled.

4.2.3.10 ZCD_NV_APS_ACK_WAIT_DURATION

Configuration ID: 0x0044; Size: 2 bytes; Default value: 3000

The amount of time (in milliseconds) a device will wait before re-transmitting a packet that used the APS acknowledgement option. If the APS acknowledgement is not received by this time, the sending device will assume a failure and attempt a re-transmission.

Note: This is recommended to be set to approximately the expected round trip time for the packet. Note that if the destination (or source) device is an end-device, the round trip time for the packet will include an additional delay up to the poll duration. This is in addition to the delay normally caused by the network.

4.2.3.11 ZCD_NV_BINDING_TIME

Configuration ID: 0x0046; Size: 2 bytes; Default value: 8000

The amount of time (in milliseconds) a device will wait for a response to a binding request.

4.2.3.12 ZCD_NV_USERDESC

Configuration ID: 0x0081; Size: 17 bytes; Default value: "CC2530-ZNP x……" (dots represent the device IEEE address)

An optional user-defined data (up to 16bytes) that can be configured in a CC2530-ZNP device so that it can easily identified or described later. The first byte is the length of the user descriptor data and must not be greater than 16.

4.2.4 Network specific configuration parameters

4.2.4.1 ZCD_NV_PANID

Configuration ID: 0x0083; Size: 2 bytes; Default value: 0xFFFF

This parameter identifies the ZigBee network. This should be set to a value between 0 and 0x3FFF. Networks that exist in the same vicinity must have different values for this parameter. It can be set to a special value of 0xFFFF to indicate *"don't care"*.

4.2.4.2 ZCD_NV_CHANLIST

Configuration ID: 0x0084; Size: 4 bytes; Default value: 0x00000800

This parameter is a bit mask of the channels on which this network can operate (note that multiple channels can be selected). See section 4.5.16 for a table of the bitmap representation that maps to each channel. Multiple networks that exist in the same vicinity are encouraged to have different values.

If multiple channels are selected, the coordinator will pick one of the channels for network operation. First, an energy scan is performed on each channel and those channels with a high energy level are discarded. Then, the coordinator determines the number of existing ZigBee networks on each of the remaining channels and picks the one with the fewest networks. For routers and end-devices, the device will simply scan all the selected channels until it finds the ZigBee network.

4.2.4.3 ZCD_NV_PRECFGKEY

Configuration ID: 0x0062; Size: 16 bytes; Default value: [0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A, 0x0B, 0x0C, 0x0D, 0x0E, 0x0F]

This is used for securing and un-securing packets in the network, if security is enabled for the network.

NOTE: Use of this configuration item requires the ZNP code to be built with the SECURE=1 compile option.

4.2.4.4 ZCD_NV_PRECFGKEYS_ENABLE

Configuration ID: 0x0063; Size: 1 byte; Default value: TRUE

If security functionality is enabled, there are two options to distribute the security key to all devices in the network. If this parameter is true, the same security key is assumed to be pre-configured in all devices in the network.If it is set to false, then the key only needs to be configured on the coordinator device. In this case, the key is distributed to each device upon joining by the coordinator. This key distribution will happen in the "clear" on the last hop of the packet transmission and this constitutes a brief "period of vulnerability" during which a malicious device can capture the key. Hence it is not recommended unless it can be ensured that there are no malicious devices in the vicinity at the time of network formation.

NOTE: Use of this configuration item requires the ZNP code to be built with the SECURE=1 compile option.

4.2.4.5 ZCD_NV_SECURITY_MODE

Configuration ID: 0x0064; Size: 1 byte; Default value: 0

This parameter determines if security is used or not in this network. It can be set to 0 (to turn off NWK security) or 1 (to turn on NWK security).

4.2.4.6 ZCD_NV_USE_DEFAULT_TCLK

Configuration ID: 0x006D; Size: 1 byte; Default value: TRUE

Controls whether a single pre-configured trust center link key is used or whether multiple pre-configured trust center link keys are used, hereby referred to as Single Key Mode and Multiple Key Mode, respectively.

In multiple key mode, unique pre-configured trust center link keys are used between the trust center and each individual device joining the network. Multiple key mode is required by the recommended secure procedure in ZigBee SE profile Specification. In single key mode, all devices are using the same pre-configured trust center link key to join the network. The single key mode provides a simplified alternative procedure to set up the network. It can be used for testing and debugging purpose.

To configure the trust center using Multiple Key Mode:

- Set the value for ZCD_NV_USE_DEFAULT_TCLK to FALSE
- In the ZNP code, set the compile option ZDSECMGR_TC_DEVICE_MAX to the maximum number of devices joining the network. Notice that it has to be no more than 255, as only 255 continuous NV ID space is reserved for preconfigured trust center link keys.
- All preconfigured trust center links keys are stored as separate NV items. The NV item ids are ranging from ZCD_NV_TCLK_TABLE_START to ZCD_NV_TCLK_TABLE_START+ ZDSECMGR_TC_DEVICE_MAX-1. Preconfigured trust center link keys are set by configuring the NV items using OSAL_NV_WRITE for the attributes listed below:

Attribute	Description	Value
Id	NV ID for the trust center link key.	ZCD_NV_TCLK_TABLE_START plus an offset.
Len	Length in bytes of the item.	0x20
Offset	The memory offset into the NV item.	0x0
Value	The data array to be written to the NV item.	Its byte format is listed in the following table. All fields follow little endian first.

Table for byte format of NV item value:

Length	8 Octets	16 Octets	4 Octets	4 Octets
Attribute Field	Extended Address	Key Data	TX Frame Counter	RX Frame Counter
Description	Extended Address of the peer devices which shares the preconfigured tclk	The preconfigured trust center link key data	The tx frame counter of the trust center link key	The rx frame counter of the trust center link key

- To remove a preconfigured trust center link key, simply write all zeros to the NV item.
- It is highly recommended to erase the entire flash before using the multiple key mode to make sure there is no existing NV item for the preconfigured trust center link keys.

To configure the trustcenter using Single Key Mode:

- Set the value for ZCD_NV_USE_DEFAULT_TCLK to TRUE
- The default preconfigured trust center link key is written to NV item ZCD_NV_TCLK_TABLE_START if it has not been initialized yet. To differentiate the default preconfigured trust center link key, the extended address for default preconfigured trust center link key is all 0xFFs. The key data is initialized with defaultTCLinkKey (defined in nwk_globals.c of the ZNP code). The RX and TX frame counters are initialized to all zeros.
- The default preconfigured tclk can be changed by changing the key data, RX and TX frame counter fields in the NV item directly.
- It is highly recommended to erase the entire flash before using the single key mode to make sure there is no existing NV item for the default preconfigured trust center link key.

- To remove the default preconfigured trust center link key, simply write all zeros to that NV item.

Please note that the Single Key Mode and Multiple Key Mode shall be used exclusively.

4.2.4.7 ZCD_NV_BCAST_RETRIES

Configuration ID: 0x002E; Size: 1 byte; Default value: 2.

The maximum number of retransmissions that a device will attempt when trying to transmit a broadcast packet. The typical range is from 1 through 3.

4.2.4.8 ZCD_NV_PASSIVE_ACK_TIMEOUT

Configuration ID: 0x002F; Size: 1 byte; Default value: 5

The amount of time (in units of 100milliseconds) a device will wait before retransmitting a broadcast packet. The retransmission will not happen if the node hears that each of its neighbor nodes have all transmitted that packet.

4.2.4.9 ZCD_NV_BCAST_DELIVERY_TIME

Configuration ID: 0x0030; Size: 1 byte; Default value: 30.

The maximum amount of time (in units of 100ms) that it can take for a broadcast packet to propagate through the entire network. This includes time for all retransmissions.

Note: This parameter must be set with caution. It must be set to a value of at least

(ZCD_NV_BCAST_RETRIES + 1) * ZCD_NV_PASSIVE_ACK_TIMEOUT

To be safe, the actual value should be higher than the above minimum by about 500ms or more.

4.2.4.10 ZCD_NV_ROUTE_EXPIRY_TIME

Configuration ID: 0x002C; Size: 1 byte; Default value: 60.

The amount of time (in seconds) for which a route must be idle (i.e. no packets are transmitted on that route) before that routing entry is marked as expired. An expired entry may be deleted if the table is full and the space is needed for another new routing entry.

This can be set to a special value of 0 to turn off route expiry. In this case, route entries are not expired.

4.2.4.11 ZNP_NV_RF_TEST_PARAMS

Configuration ID: 0x0F07; Size: 4 bytes; Default value: { 0, 0, 0, 0 }.

These parameters are used to initialize characteristics for the RF Test Mode, which is designed to be invoked before/instead of MAC radio initialization. After writing this NV item, the RF Test Mode is started by sending a system reset command, and will run continuously until a physical reset is performed.

Byte	Description	Value
0	RF Test Mode	1 = RX promiscuous mode 2 = TX unmodulated mode 3 = TX modulated mode
1	RF channel	11-26 - selects a 2.4GHz channel for 802.15.4 operation

2	TX power	Sets TXPOWER register for CC2530 (TI document SWRS081A): 0xF5 = +4.5 dBm 0xE5 = +2.5 dBm 0xD5 = +1 dBm 0xC5 = -0.5 dBm 0xB5 = -1.5 dBm 0xA5 = -3 dBm 0x95 = -4 dBm 0x85 = -6 dBm 0x75 = -8 dBm 0x65 = -10 dBm 0x55 = -12 dBm 0x45 = -14dBm 0x35 = -16 dBm 0x25 = -18 dBm 0x15 = -20 dBm 0x05 = -22 dBm Sets TXPOWER register for CC2591 (TI document SWRA308A): 0xE5 = +20 dBm 0xD5 = +19 dBm 0xC5 = +18 dBm 0xB5 = +17 dBm 0xA5 = +16 dBm 0x95 = +14.5 dBm 0x85 = +13 dBm 0x75 = +11.5 dBm 0x65 = +10 dBm
3	TX power tone	0-15 - sets bits 4-7 of the MDMTEST0 register of the CC2530

4.3 Simple API interface

The Simple API interface is intended to present a simplified ZigBee API to the application developer. The complete ZigBee interface is provided via the AF and ZDO interfaces. But since the majority of the applications do not use the full feature set available in ZigBee, this simplified interface is an easy way for the developer to begin ZigBee application development. It contains the necessary interface to commission a ZigBee network, perform bindings between devices and send and receive data.

4.3.1 ZB_APP_REGISTER_REQUEST

4.3.1.1 Description

This command enables the application processor to register its application with the CC2530-ZNP device.

4.3.1.2 Usage

SREQ:

1	1	1	1	2	2	*1*
Length = variable	*Cmd0 = 0x26*	*Cmd1 = 0x0A*	*AppEndPoint*	*AppProfileID*	*DeviceId*	*DeviceVersion*

1	1	2 x Input commands	1	*2 x Output commands*
Unused	*InputCommandsNum*	*InputCommandsList*	*OutputCommandsNum*	*OutputCommandsList*

AppEndPoint – 1 byte – Specifies the endpoint of the device. This should be in the range of 1 through 240 and should be set to same value for all devices in the network.

AppProfileID – 2 bytes – Specifies the profile id of the application. This should be set to same value to all devices in the network. This number is assigned by the ZigBee Alliance.

DeviceId – 2 bytes – This is an application-specific identifier. It identifies the device type within the particular profile id. This is not used by the ZigBee stack in any way other than to identify itself when requested.

DeviceVersion – 1 byte – This is an application-specific identifier. It identifies the version of the device. This is not used by the ZigBee stack in any way other than to identify itself when requested.

Unused – 1 byte – Unused parameter.

InputCommandsNum – 1 byte – Specifies the number of Input commands that this application will process.

InputCommandsList – variable – List of input command identifiers that are processed by this application. Each command identifier is 2 bytes long, LSB first.

OutputCommandsNum – 1 byte – Specifies the number of Output commands that this application will generate.

OutputCommandsList – variable – List of output command identifiers that are generated by this application. Each command identifier is 2 bytes long, LSB first.

Note:

It is mandatory that the application register itself with the CC2530-ZNP device after every reset if it wishes to use the simple API interface. The AppEndPoint and AppProfileId fields should be populated with the same values for all devices in the network.

This list of input and output commands are only used by the ZigBee stack when performing binding. If the binding feature is not used, these may be ignored.

The DeviceId and DeviceVersion are not used by the ZigBee stack and may be ignored.

SRSP:

1	1	1	1
Length = 0x01	*Cmd0 = 0x66*	*Cmd1 = 0x0A*	*Status*

Status – 1 byte – See 4.7 for a listing of the status values.

4.3.2 ZB_START_REQUEST

4.3.2.1 Description

This command starts the ZigBee stack in the CC2530-ZNP device. When the ZigBee stack starts, the device reads the programmed configuration parameters and operates accordingly. After the start request process completes, the device is ready to send, receive, and route network traffic.

NOTE: Calling this function without following the procedure outlined in section 3.2 could result in unexpected behaviour and not result in the ZNP issuing a ZB_START_CONFIRM.

4.3.2.2 Usage

SREQ:

1	1	1
Length = 0x00	*Cmd0 = 0x26*	*Cmd1 = 0x00*

SRSP:

1	1	1
Length = 0x00	*Cmd0 = 0x66*	*Cmd1 = 0x00*

4.3.3 ZB_START_CONFIRM

4.3.3.1 Description

This command is issued by the CC2530-ZNP device to return the results from a ZB_START_REQUEST command.

4.3.3.2 Usage

AREQ:

1	1	1	1
Length = 0x01	*Cmd0 = 0x46*	*Cmd1 = 0x80*	*Status*

Status – 1 byte – Either 0x00 (ZB_SUCCESS) or 0x22 (ZB_INIT). ZB_SUCCESS means that the device started as either a coordinator, router, or end device. This may take anywhere from 500 ms to a few seconds depending on how busy or noisy the channel is. The ZB_INIT status would not be returned if the device attempted to start as a coordinator and failed, due to the channel(s) being too busy.

4.3.4 ZB_PERMIT_JOINING_REQUEST

4.3.4.1 Description

This command is used to control the joining permissions and thus allow or disallow new devices from joining the network. By default, permit joining is always on.

4.3.4.2 Usage

SREQ:

1	1	1	2	*1*
Length = 0x03	*Cmd0 = 0x26*	*Cmd1 = 0x08*	*Destination*	*Timeout*

Destination – 2 bytes – The destination parameter indicates the address of the device for which the joining permissions should be set. This is usually the local device address or the special broadcast address that denotes all routers and coordinator (0xFFFC). This way the joining permissions of a single device or the whole network can be controlled.

Timeout – 1 byte – Indicates the amount of time in seconds for which the joining permissions should be turned on. If timeout is set to 0x00, the device will turn off the joining permissions indefinitely. If it is set to 0xFF, the joining permissions will be turned on indefinitely.

SRSP:

1	1	1	*1*
Length = 0x01	*Cmd0 = 0x66*	*Cmd1 = 0x08*	*Status*

Status – 1 byte – See 4.7 for a listing of the status values.

4.3.5 ZB_BIND_DEVICE

4.3.5.1 Description

This command is used to create or delete a 'binding' to another device on the network. Once bound, an application can send messages to a device by referencing the commandId for the binding. This command can also be issued with a NULL destination address (set to all zeros). In that case, a binding will be established with another device that is in the Allow Bind mode.

NOTE: Calling this function without following the procedure outlines in section 3.2 could result in unexpected behaviour and not result in the ZNP issuing a ZB_BIND_CONFIRM.

4.3.5.2 Usage

SREQ:

1	1	1	1	2	*8*
Length = 0x0B	*Cmd0 = 0x26*	*Cmd1 = 0x01*	*Create*	*CommandId*	*Destination*

Create – 1 byte – TRUE to create a binding, FALSE to remove a binding.

CommandId – 2 bytes – The identifier of the binding.

Destination – 8 bytes – Specifies the 64-bit IEEE address of the device to bind to. Set to NULL if the destination address is unknown and instead the destination device is set to Allow Bind mode.

SRSP:

1	1	*1*
Length = 0x00	*Cmd0 = 0x66*	*Cmd1 = 0x01*

4.3.6 ZB_BIND_CONFIRM

4.3.6.1 Description

This command is issued by the CC2530-ZNP device to return the results from a `ZB_BIND_DEVICE` command.

4.3.6.2 Usage

AREQ:

1	1	1	2	*1*
Length = 0x03	*Cmd0 = 0x46*	*Cmd1 = 0x81*	*CommandId*	*Status*

CommandId – 2 bytes – The command ID of the binding being confirmed.

Status – 1 byte – See 4.7 for a listing of the status values.

4.3.7 ZB_ALLOW_BIND

4.3.7.1 Description

This command puts the device into the Allow Binding Mode for a given period of time. This allows a peer device to establish a binding with this device (in the Allow Binding Mode) by issuing the zb_BindDevice with a destination address of NULL.

NOTE: Calling this function without following the procedure outlined in section 3.2 could result in unexpected behaviour and not result in the ZNP issuing a ZB_ALLOW_BIND_CONFIRM.

4.3.7.2 Usage

SREQ:

1	1	1	*1*
Length = 0x01	*Cmd0 = 0x26*	*Cmd1 = 0x02*	*Timeout*

Timeout – 1 byte – The number of seconds to remain in the allow binding mode. Valid values range from 1 through 65. If 0, the Allow Bind mode will be set false without timeout. If greater than 64, the Allow Bind mode will be true indefinitely.

SRSP:

1	1	*1*
Length = 0x00	*Cmd0 = 0x66*	*Cmd1 = 0x02*

4.3.8 ZB_ALLOW_BIND_CONFIRM

4.3.8.1 Description

This command is issued by the CC2530-ZNP device when it responds to a bind request from a remote device.

4.3.8.2 Usage

AREQ:

1	1	1	*2*

Length = 0x02	*Cmd0 = 0x46*	*Cmd1 = 0x82*	*Source*

Source – 2 bytes – Contains the address of the device attempted to bind to this device.

4.3.9 ZB_SEND_DATA_REQUEST

4.3.9.1 Description

This command initiates transmission of data to another device in the network. This command can only be issued after the application processor has registered its application using the ZB_APP_REGISTER_REQUEST and the device has successfully created or joined a network.

NOTE: Calling this function without following the procedure outlined in section 3.2 could result in unexpected behaviour and not result in the ZNP issuing a ZB_SEND_DATA_CONFIRM.

4.3.9.2 Usage

SREQ:

1	**1**	**1**	**2**	**2**	***1***
Length = 0x08-0x6B	*Cmd0 = 0x26*	*Cmd1 = 0x03*	*Destination*	*CommandId*	*Handle*

1	**1**	**1**	***0-99***
Ack	*Radius*	*Len*	*Data*

Destination – 2 bytes – The destination address of the data packet. It can be one of the following values:

Address	*Description*
0 – 0xFFF7	*16-bit short address of the destination device*
0xFFFC	*Group of all routers and coordinator*
0xFFFD	*Group of all devices with receiver turned on*
0xFFFE	*This is the binding address and should be used when a binding entry has been previously created for this particular CommandId. The destination address will be determined from the binding table by the CC2530-ZNP*
0xFFFF	*Broadcast group of all devices in the network*

CommandId – 2 bytes – The command ID to send with the message. If the binding address is used for destination, this parameter also indicates the binding to use.

Handle – 1 byte – A handle used to identify the send data request. The corresponding ZB_SEND_DATA_CONFIRM will have the same handle value. This can be useful if the application wishes to match up ZB_SEND_DATA_REQUESTs with ZB_SEND_DATA_CONFIRMs.

Ack – 1 byte – TRUE if requesting APS acknowledgement from the destination.

Radius – 1 byte – The max number of hops the packet can travel through before it is dropped.

Len – 1 byte – Specifies the size of the Data buffer in bytes.

Data – 0-99 bytes – Data. Without any security (99 bytes), with NWK security (81 bytes), with NWK and APS security (64 bytes).

SRSP:

1	1	1
Length = 0x00	*Cmd0 = 0x66*	*Cmd1 = 0x03*

4.3.10 ZB_SEND_DATA_CONFIRM

4.3.10.1 Description

This command is issued by the CC2530-ZNP device to return the results from a ZB_SEND_DATA_REQUEST command. For each ZB_SEND_DATA_REQUEST, a ZB_SEND_DATA_CONFIRM is always returned. If APS acknowledgement was used for the ZB_SEND_DATA_REQUEST, the confirm carries the status of whether the APS acknowledgement was received or not (ZApsNoAck – 0xb7). If APS acknowledgement was not used, then the confirm carries the status of whether the MAC acknowledgement ("next hop" acknowledgment) was received or not (ZMacNoACK – 0xe9).

4.3.10.2 Usage

AREQ:

1	1	1	1	1
Length = 0x02	*Cmd0 = 0x46*	*Cmd1 = 0x83*	*Handle*	*Status*

Handle – 1 byte – Specifies the handle.

Status – 1 byte – See 4.7 for a listing of the status values.

4.3.11 ZB_RECEIVE_DATA_INDICATION

4.3.11.1 Description

This callback is called asynchronously by the CC2530-ZNP device when it has received a packet from a remote device.

4.3.11.2 Usage

AREQ:

1	1	1	2	2	2	0-84
Length = 0x06-5A	*Cmd0 = 0x46*	*Cmd1 = 0x87*	*Source*	*Command*	*Len*	*Data*

Source – 2 bytes – Specifies the short address of the peer device that sent the data.

Command – 2 bytes – The command ID associated with the data.

Len – 2 bytes – Specifies the number of bytes in the Data parameter.

Data – Array of bytes – The data sent by the peer device

4.3.12 ZB_GET_DEVICE_INFO

4.3.12.1 Description

This command retrieves a Device Information Property.

4.3.12.2 Usage

SREQ:

1	1	1	*1*
Length = 0x01	*Cmd0 = 0x26*	*Cmd1 = 0x06*	*Param*

Param – 1 byte – The identifier of the device information. It can take one of the following values:

Parameter	Size	*Description*
0	1 byte	*Device state – See 4.5.52*
1	8 bytes	*Device IEEE address*
2	2 bytes	*Device short address*
3	2 bytes	*Short address of the parent device*
4	8 bytes	*IEEE address of the parent device*
5	1 byte	*Channel on which the ZigBee network is operating*
6	2 bytes	*PAN ID of the ZigBee network*
7	*8 bytes*	*Extended PAN Id of the ZigBee network*

SRSP:

1	1	1	1	*8*
Length = 0x09	*Cmd0 = 0x66*	*Cmd1 = 0x06*	*Param*	*Value*

Param – 1 byte – The identifier of the requested device information.

Value – 8 byte – The value of the requested device information, LSB first. This is always 8bytes in length even though the actual value may be smaller in size. The remaining bytes are a don't care.

4.3.13 ZB_FIND_DEVICE_REQUEST

4.3.13.1 Description

This command is used to determine the short address for a device in the network. The device initiating a call to ZB_FIND_DEVICE_REQUEST and the device being discovered must both be a member of the same network. When the search is complete, the ZNP responds with ZB_FIND_DEVICE_CONFIRM.

SREQ:

1	1	1	*8*
Length = 0x08	*Cmd0 = 0x26*	*Cmd1 = 0x07*	*SearchKey*

SearchKey – 8 bytes – Specifies the value to search on.

SRSP:

1	1	*1*
Length = 0x00	*Cmd0 = 0x66*	*Cmd1 = 0x07*

4.3.14 ZB_FIND_DEVICE_CONFIRM

4.3.14.1 Description

This command is issued by the CC2530-ZNP device to return the results from a `ZB_FIND_DEVICE_REQUEST` command.

4.3.14.2 Usage

AREQ:

1	1	1	1	2	*8*
Length = 0x0B	*Cmd0 = 0x46*	*Cmd1 = 0x85*	*SearchType = 0x01*	*SearchKey*	*Result*

SearchType – 1 byte – The type of search that was performed.

SearchKey – 2 bytes – Value that the search was executed on.

Result – 8 bytes – The result of the search.

4.4 AF Interface

This interface allows the host processor to interact with the Application Framework layer (AF).

4.4.1 AF_REGISTER

4.4.1.1 Description

This command enables the host processor to register an application's endpoint description (and its simple descriptor). Multiple endpoints may be registered with the AF by making multiple calls to AF_REGISTER. This could be useful in the case where the device needs to support multiple application profiles, where each AF_REGISTER call would register a unique endpoint description per application profile.

4.4.1.2 Usage

SREQ:

1	1	1	1	2	2
Length = 0x09-0x49	Cmd0 = 0x24	Cmd1 = 0x00	EndPoint	AppProfId	AppDeviceId

1	1	1	0-32	1	0-32
AppDevVer	LatencyReq	AppNumInClusters	AppInClusterList	AppNumOutClusters	AppOutClusterList

Attributes:

Attribute	Length (byte)	Description
EndPoint	1	Specifies the endpoint of this simple descriptor.
AppProfId	2	Specifies the profile id of the application
AppDeviceId	2	Specifies the device description id for this endpoint
AddDevVer	1	Specifies the device version number
LatencyReq	1	Specifies latency. For ZigBee the only applicable value is 0x00. 0x00-No latency 0x01-fast beacons 0x02-slow beacons
AppNumInClusters	1	the number of Input cluster Ids following in the AppInClusterList
AppInClusterList	32	Specifies the list of Input Cluster Ids (2bytes each)
AppNumOutClusters	1	Specifies the number of Output cluster Ids following in the AppOutClusterList
AppOutClusterList	32	Specifies the list of Output Cluster Ids (2bytes each)

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x64	Cmd1 = 0x00	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	Status is either Success (0) or Failure (1).

4.4.2 AF_DATA_REQUEST

4.4.2.1 Description

This command is used by the App processor to build and send a message through AF layer.

4.4.2.2 Usage

SREQ:

1	1	1	2	1
Length = 0x0A-0x6D	Cmd0 = 0x24	Cmd1 = 0x01	DstAddr	DestEndpoint

1	2	1	1	1	1	0-128
SrcEndpoint	ClusterID	TransID	Options	Radius	Len	Data

Attributes:

Attribute	Length (byte)	Description
DstAddr	2	Short address of the destination device
DestEndpoint	1	Endpoint of the destination device
SrcEndpoint	1	Endpoint of the source device
ClusterID	2	Specifies the cluster ID
TransID	1	Specifies the transaction sequence number of the message. The corresponding AF_DATA_CONFIRM will have the same TransID. This can be useful if the application wishes to match up AF_DATA_REQUESTs with AF_DATA_CONFIRMs.
Options	1	The transmit options field is organized as a bitmask. The following enumerates the values for the various supported bitmasks. For example, a value of 0x10 means that bit 4 is set. **Value** / **Description** 0x10 – AF_ACK_REQUEST – set this bit to request APS acknowledgement for this packet 0x20 – AF_DISCV_ROUTE – set this bit to force route discovery if a routing table entry doesn't exist 0x40 – AF_EN_SECURITY – set this bit to enable APS security for this packet.
Radius	1	Specifies the number of hops allowed delivering the message (reference AF_DEFAULT_RADIUS)
Len	1	Length of the data.
Data	0-99	0-99 bytes data. Without any security (99 bytes), with NWK security (81 bytes), with NWK and APS security (64 bytes).

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x64	Cmd1 = 0x01	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	Status is either Success (0) or Failure (1).

4.4.3 AF_DATA_REQUEST_EXT

4.4.3.1 Description

This extended form of the AF_DATA_REQUEST must be used to send an INTER-PAN message. Note that the ZNP code must be compiled with the INTER_PAN flag defined. The INTER_PAN compile option is defined by default. Although the INTER-PAN feature provides a method for message delivery that is non-secured, it offers a low overhead mechanism for message delivery between different networks (and even on different channels), or even devices on the same network. See the ZigBee Smart Energy specification for more details about the INTER-PAN feature.

This extended data request must also be used when making a request with a huge data byte count which is defined to be a size that would cause the RPC request to exceed the maximum allowed size:

```
MT_RPC_DATA_MAX – sizeof(AF_DATA_REQUEST_EXT)
```

Where sizeof(AF_DATA_REQUEST_EXT) counts everything but the data bytes and now stands at 20. MT_RPC_DATA_MAX is set to 250. When making an AF_DATA_REQUEST_EXT with a huge data byte count, the request shall not contain any data bytes. The huge data buffer is sent over separately as a sequence of one or more AF_DATA_STORE requests. Note that the outgoing huge message is timed-out in 15 seconds; thus all AF_DATA_STORE requests must be completed within 15 seconds of an AF_DATA_REQUEST_EXT with a huge data byte count. Any AF_DATA_REQUEST_EXT with a huge data byte count must be completed (or timed-out) before another will be started. The default timeout can be changed by defining the following to other values:

```
#if !defined MT_AF_EXEC_CNT
#define MT_AF_EXEC_CNT  15
#endif
#if !defined MT_AF_EXEC_DLY
#define MT_AF_EXEC_DLY  1000
#endif
```

4.4.3.2 Usage

SREQ:

1	1	1	1	8	1
Length = 0x14-0xFA	Cmd0 = 0x24	Cmd1 = 0x02	DstAddrMode	DstAddr	DstEndpoint

2	1	2	1	1	1	2	0-230
DstPanId	SrcEndpoint	ClusterId	TransId	Options	Radius	Len	Data

Attributes:

<table>
<tr><th>Attribute</th><th>Length (byte)</th><th>Description</th></tr>
<tr><td>DstAddrMode</td><td>1</td><td>A value of 3 (the enumeration value for ‘afAddr64Bit’) indicates 8-byte (64-bit) address mode; otherwise a value of 2 indicates 2-byte (16-bit) address mode, using only the 2 LSB’s of the DstAddr field to form a 2-byte short address.</td></tr>
<tr><td>DstAddr</td><td>8</td><td>LSB to MSB for the long or short address of the destination device (upper 6 bytes are don’t care when short address.)</td></tr>
<tr><td>DstEndpoint</td><td>1</td><td>Endpoint of the destination device (but a don’t care if the DstPanId is non-zero, which indicates an inter-pan message.)</td></tr>
<tr><td>DstPanId</td><td>2</td><td>PanId of the destination device: 0x0000=Intra-Pan; otherwise, Inter-Pan.</td></tr>
<tr><td>SrcEndpoint</td><td>1</td><td>Endpoint of the source device</td></tr>
<tr><td>ClusterID</td><td>2</td><td>Specifies the cluster ID</td></tr>
<tr><td>TransID</td><td>1</td><td>Specifies the transaction sequence number of the message.</td></tr>
<tr><td>Options</td><td>1</td><td>The transmit options field is organized as a bitmask. The following enumerates the values for the various supported bitmasks. For example, a value of 0x10 means that bit 4 is set.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0x10</td><td>AF_ACK_REQUEST – set this bit to request APS acknowledgement for this packet</td></tr>
<tr><td>0x20</td><td>AF_DISCV_ROUTE – set this bit to force route discovery if a routing table entry doesn’t exist</td></tr>
<tr><td>0x40</td><td>AF_EN_SECURITY – set this bit to enable APS security for this packet.</td></tr>
</table></td></tr>
<tr><td>Radius</td><td>1</td><td>Specifies the number of hops allowed delivering the message (reference AF_DEFAULT_RADIUS)</td></tr>
<tr><td>Len</td><td>2</td><td>Length of the data. If a large data length causes the MT command to exceed MT_RPC_DATA_MAX, then zero bytes of the data shall be sent with this request and the data shall be transferred in as many AF_DATA_STORE requests as necessary.</td></tr>
<tr><td>Data</td><td>0-230</td><td>0-230 bytes data</td></tr>
</table>

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x64	Cmd1 = 0x02	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	Status is either Success (0) or Failure (1).

4.4.4 AF_DATA_REQUEST_SRC_RTG

4.4.4.1 Description

This command is used by the App processor to build and send a message through AF layer using source routing.

4.4.4.2 Usage

SREQ:

1	1	1	2	1
Length = 0x0B-0xFF	Cmd0 = 0x24	Cmd1 = 0x02	DstAddr	DestEndpoint

1	2	1	1	1	1	2N	1	0-128
SrcEndpoint	ClusterID	TransID	Options	Radius	Relay Count (N)	RelayList	Len	Data

Attributes:

Attribute	Length (byte)	Description
DstAddr	2	Short address of the destination device
DestEndpoint	1	Endpoint of the destination device
SrcEndpoint	1	Endpoint of the source device
ClusterID	2	Specifies the cluster ID
TransID	1	Specifies the transaction sequence number of the message.
Options	1	The transmit options field is organized as a bitmask. The following enumerates the values for the various supported bitmasks. For example, a value of 0x10 means that bit 4 is set. **Value** / **Description** 0x10 – AF_ACK_REQUEST – set this bit to request APS acknowledgement for this packet 0x20 – AF_DISCV_ROUTE – set this bit to force route discovery if a routing table entry doesn't exist 0x40 – AF_EN_SECURITY – set this bit to enable APS security for this packet.
Radius	1	Specifies the number of hops allowed delivering the message (reference AF_DEFAULT_RADIUS)
Relay Count	1	Specifies the number of devices in the relay list for source routing
Relay List	2N	List of relay devices on the source routing path. For each device, it contains 2 bytes short address for each device.
Len	1	Length of the data.
Data	0-128	0-128 bytes data

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x64	Cmd1 = 0x02	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	Status is either Success (0) or Mem Failure (0x10).

4.4.5 AF_INTER_PAN_CTL

4.4.5.1 Description

This command is used by the host processor for INTER-PAN control.For example, this command is used by the host processor to register an endpoint that will support INTER-PAN communication.

4.4.5.2 Usage

SREQ:

1	1	1	1	0-3
Length = 0x01-0x04	Cmd0 = 0x24	Cmd1 = 0x10	Command	Data

Attributes:

<table>
<tr><th>Attribute</th><th>Length (byte)</th><th>Description</th></tr>
<tr><td>Command</td><td>1</td><td>
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0x00</td><td>InterPanClr – Proxy call to StubAPS_SetIntraPanChannel() to switch back to the NIB-specified channel. Length of data field is 0</td></tr>
<tr><td>0x01</td><td>InterPanSet – Proxy call to StubAPS_SetInterPanChannel() with the 1-byte channel specified. Length of data field is 1.</td></tr>
<tr><td>0x02</td><td>InterPanReg – Specify the endpoint that will be able to send and receive INTER-PAN messages. The endpoint must already be registered by using the AF_REGISTER command. Length of data field is 1</td></tr>
<tr><td>0x03</td><td>InterPanChk - Proxy a call to StubAPS_InterPan() with the 2-byte PanId (LSB:MSB) and 1-byte EndPoint data. Length of the data field is 3. This command is typically used in OSAL based ZAP applications that will have the ZigBee Cluster Library on the host processor.</td></tr>
</table>
</td></tr>
<tr><td>Data</td><td>0-3</td><td>See description of each command for data to pass in</td></tr>
</table>

SRSP:

Byte: 1	1	1	1
Length = 0x01	Cmd0 = 0x64	Cmd1 = 0x10	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	<table><tr><th>Value</th><th>Description</th></tr><tr><td>0x00</td><td>Success</td></tr><tr><td>0x02</td><td>Invalid Parameter</td></tr><tr><td>0x10</td><td>Failure due to channel change already in progress</td></tr><tr><td>0xBA</td><td>ZApsNotAllowed – MAC is not in an idle state</td></tr></table>

4.4.6 AF_DATA_STORE

4.4.6.1 Description

This command is used by the host processor to facilitate the transfer of large packets that use APS fragmentation for over-the-air transmission.

4.4.6.2 Usage

SREQ:

1	1	1	2	1	0-247
Length = 0x03-0xFA	Cmd0 = 0x24	Cmd1 = 0x11	Index	Length	Data

Attributes:

Attribute	Length (byte)	Description
Index	2	Specifies the index into the outgoing data request data buffer to start the storing of this chunk of data
Length	1	Specifies the length of this data chunk to store. A length of zero is special and triggers the actually sending of the data request OTA
Data	0-247	Contains 0 to 247 bytes of data

SRSP:

Byte: 1	1	1	1
Length = 0x01	Cmd0 = 0x64	Cmd1 = 0x11	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	<table><tr><th>Value</th><th>Description</th></tr><tr><td>0x00</td><td>Success</td></tr><tr><td>0x01</td><td>Failure</td></tr><tr><td>0x02</td><td>Invalid Parameter</td></tr><tr><td>0x10</td><td>Mem Fail</td></tr></table> Note that the status is for storing a chunk of data when Length is not zero and the return value of the AF_DataRequest() when it is zero.

4.4.7 AF_DATA_CONFIRM

4.4.7.1 Description

This command is sent by the device to the user after it receives an AF_DATA_REQUEST. For each AF_DATA_REQUEST, a AF_DATA_CONFIRM is always returned. If APS acknowledgement was used for the AF_DATA_REQUEST, the confirm carries the status of whether the APS acknowledgement was received or not (ZApsNoAck – 0xb7). If APS acknowledgement was not used, then the confirm carries the status of whether the MAC acknowledgement ("next hop" acknowledgment) was received or not (ZMacNoACK – 0xe9). This also applies to packets that are sent using AF_DATA_REQUEST_EXT and AF_DATA_STORE. For APS fragmented packets, the value of the configuration item ZCD_NV_APSF_WINDOW_SIZE determines when an AF_DATA_CONFIRM that carries the status of the APS acknowledgement is received.

4.4.7.2 Usage

AREQ:

1	1	1	1	1	1
Length = 0x03	Cmd0 = 0x44	Cmd1 = 0x80	Status	Endpoint	TransID

Attributes:

Attribute	Length (byte)	Description
Status	1	Status is either Success (0) or Failure (1).
Endpoint	1	Endpoint of the device
TransId	1	Specifies the transaction sequence number of the message

4.4.8 AF_INCOMING_MSG

4.4.8.1 Description

This callback message is in response to incoming data to any of the registered endpoints on this device.

4.4.8.2 Usage

AREQ:

1	1	1	2	2	2	1
Length = 0x11-0x74	Cmd0 = 0x44	Cmd1 = 0x81	GroupID	ClusterID	SrcAddr	SrcEndpoint

1	1	1	1	4	1	1	0-99
DestEndpoint	WasBroadcast	LinkQuality	SecurityUse	Timestamp	TransSeqNumber	Len	Data

Attributes:

Attribute	Length (byte)	Description
GroupID	2	Specifies the group ID of the device
ClusterID	2	Specifies the cluster ID
SrcAddr	2	Specifies the ZigBee network address of the source device sending the message.
SrcEndpoint	1	Specifies the source endpoint of the message
DestEndpoint	1	Specifies the destination endpoint of the message
WasBroadcast	1	Specifies if the message was a broadcast or not
LinkQuality	1	Indicates the link quality measured during reception
SecurityUse	1	Specifies if the security is used or not
TimeStamp	4	Specifies the timestamp of the message
TransSeqNumber	1	Specifies transaction sequence number of the message
Len	1	Specifies the length of the data.
Data	0-99	Contains 0 to 99 bytes of data. Without any security (99 bytes), with NWK security (81 bytes), with NWK and APS security (64 bytes).

4.4.9 AF_INCOMING_MSG_EXT

4.4.9.1 Description

This response is triggered by incoming data to any of the registered endpoints on this device when the ZNP code is compiled with the INTER_PAN flag defined.

This extended incoming message indication must also be used when handling an incoming message with a huge data byte count which is defined to be a size that would cause the RPC request to exceed the maximum allowed size:

```
MT_RPC_DATA_MAX – sizeof(AF_INCOMING_MSG_EXT)
```

Where `sizeof(AF_ INCOMING_MSG _EXT)` counts everything but the data bytes and now stands at 27. MT_RPC_DATA_MAX is set to 250. An AF_INCOMING_MSG_EXT with a huge data byte count indication shall not contain any data bytes. The huge data buffer must be retrieved separately as a sequence of one or more AF_DATA_RETRIEVE requests. Note that the incoming

huge message is timed-out in 15 seconds after receiving it; thus all AF_DATA_ RETRIEVE requests must be completed within 15 seconds of an AF_ INCOMING_MSG _EXT with a huge data byte count. Note that multiple AF_INCOMING_MSG_EXT indications with huge data byte counts may be queued, and each will be timed-out separately. The default timeout can be changed by defining the following to other values:

```
#if !defined MT_AF_EXEC_CNT
#define MT_AF_EXEC_CNT  15
#endif
#if !defined MT_AF_EXEC_DLY
#define MT_AF_EXEC_DLY  1000
#endif
```

4.4.9.2 Usage

AREQ:

1	1	1	2	2	1	8	1
Length = 0x1B-0xFA	Cmd0 = 0x44	Cmd1 = 0x82	GroupID	ClusterID	SrcAddrMode	SrcAddr	SrcEndpoint

2	1	1	1	1	4	1	2	0-223
SrcPanId	DstEndpoint	WasBroadcast	LinkQuality	SecurityUse	Timestamp	TransSeqNumber	Len	Data

Attributes:

Attribute	Length (byte)	Description
GroupID	2	Specifies the group ID of the device
ClusterID	2	Specifies the cluster ID
SrcAddrMode	1	A value of 3 (i.e. the enumeration value for 'afAddr64Bit') indicates 8-byte/64-bit address mode; otherwise, only the 2 LSB's of the 8 bytes are used to form a 2-byte short address.
SrcAddr	8	LSB to MSB for the long or short address of the destination device (upper 6 bytes are don't care when short address.)
SrcEndpoint	1	Specifies the source endpoint of the message
SrcPanId	2	Specifies the source PanId of the message
DstEndpoint	1	Specifies the destination endpoint of the message
WasBroadcast	1	Specifies if the message was a broadcast or not
LinkQuality	1	Indicates the link quality measured during reception
SecurityUse	1	Specifies if security is used or not
TimeStamp	4	Specifies the timestamp of the message
TransSeqNumber	1	Specifies transaction sequence number of the message
Len	2	Specifies the length of the data. If a large data length causes the MT command to exceed MT_RPC_DATA_MAX, then zero bytes of the data shall be sent with this request and the host shall retrieve the data with as many MT_AF_DATA_RETRIEVE requests as necessary.
Data	0-223	Contains 0 to 223 bytes of data.

4.4.10 AF_DATA_RETRIEVE

4.4.10.1 Description

This command is used by the host processor for receiving large packets that use APS fragmentation for over-the-air reception.

4.4.10.2 Usage

SREQ:

1	1	1	4	2	1
Length = 0x07	Cmd0 = 0x24	Cmd1 = 0x12	Timestamp	Index	Length

Attributes:

Attribute	Length (byte)	Description
Timestamp	4	The timestamp of the incoming message in order to uniquely Identify it in a queue of incoming huge messages.
Index	2	Specifies the index into the incoming message data buffer to start the retrieving of this chunk of data
Length	1	Specifies the length of this data chunk to retrieve. A length of zero is special and triggers the freeing of the corresponding incoming message

SRSP:

1	1	1	1	1	0-248
Length = 0x02-0xFA	Cmd0 = 0x64	Cmd1 = 0x12	Status	Length	Data

Attributes:

<table>
<tr><th>Attribute</th><th>Length (byte)</th><th>Description</th></tr>
<tr><td>Status</td><td>1</td><td>
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0x00</td><td>Success</td></tr>
<tr><td>0x01</td><td>Failure</td></tr>
<tr><td>0x02</td><td>Invalid Parameter</td></tr>
<tr><td>0x10</td><td>Mem Fail</td></tr>
</table>
</td></tr>
<tr><td>Length</td><td>1</td><td>Specifies the length of this data chunk retrieved</td></tr>
<tr><td>Data</td><td>0-248</td><td>The length of data bytes requested from the specified index into the huge incoming message data buffer</td></tr>
</table>

4.4.11 AF_APSF_CONFIG_SET

4.4.11.1 Description

This command enables the host processor to change the default APS Fragmentation configuration settings for a specific EndPoint. This call only succeeds after first successfully registering the EndPoint with the AF. The window size is the number of fragments that are sent before an APS Fragmentation ACK is expected. Therefore, if the message is broken up into 10 fragments and the max window size is 5 then an ACK will be sent by the receiving device after 5 fragments are received. If one packet of the window size isn't received, the ACK is not sent and all the packets (within that window) are resent. Note that the maximum fragmentation window size that can be set is dictated by the specification as 8 and defined in aps_frag.h as APSF_MAX_WINDOW_SIZE. The frame delay is the delay in milliseconds between fragments within a window. This is used by the sending device.

4.4.11.2 Usage

SREQ:

1	1	1	1	1	1
Length = 0x03	Cmd0 = 0x24	Cmd1 = 0x13	endPoint	frameDelay	windowSize

Attributes:

Attribute	Length (byte)	Description
endPoint	1	Specifies the endpoint for which to set the fragmentation configuration.
frameDelay	1	Specifies the frame delay in milliseconds to set.
windowSize	1	Specifies the window size to set up to APSF_MAX_WINDOW_SIZE.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x64	Cmd1 = 0x13	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	Status is of type afStatus_t, defined in AF.h, and will be one of these 2: afStatus_SUCCESS afStatus_INVALID_PARAMETER

4.5 ZDO Interface

This interface allows the application processor to issue commands to the ZDO layer in the CC2530-ZNP. The result of the command execution will be conveyed to the application processor via the corresponding callback message. See 4.2.3.3 ZCD_NV_ZDO_DIRECT_CB for more details on how to configure the manner in which ZDO callback messages are issued to the host processor.

4.5.1 ZDO_NWK_ADDR_REQ

4.5.1.1 Description

This message will request the device to send a “Network Address Request”. This message sends a broadcast message looking for a 16 bit address with a known 64 bit IEEE address.

4.5.1.2 Usage

SREQ:

1	1	1	8	1	1
Length = 0x0A	Cmd0 = 0x25	Cmd1 = 0x00	IEEEAddress	ReqType	StartIndex

Attributes:

Attribute	Length (byte)	Description
IEEEAddress	8	64 bit IEEE address of the device.
ReqType	1	Value that the search was executed on. Type: Single Device response — Value: 0x00 Type: Extended, include associated devices — Value: 0x01
StartIndex	1	Starting index into the list of children. This is used to get more of the list if the list is too large for one message.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x00	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	Status is either Success (0) or Failure (1).

4.5.2 ZDO_IEEE_ADDR_REQ

4.5.2.1 Description

This command will request a device's IEEE 64-bit address.

4.5.2.2 Usage

SREQ:

1	1	1	2	1	1
Length = 0x04	Cmd0 = 0x25	Cmd1 = 0x01	ShortAddr	ReqType	StartIndex

Attributes:

Attribute	Length (byte)	Description
ShortAddr	2	Specifies the short address of the device.
ReqType	1	Value that the search was executed on. Type: Single Device response — Value: 0x00 Type: Extended, include associated devices — Value: 0x01
StartIndex	1	Starting index into the list of children. This is used to get more of the list if the list is too large for one message.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x01	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	Status is either Success (0) or Failure (1).

4.5.3 ZDO_NODE_DESC_REQ

4.5.3.1 Description

This command is generated to inquire about the Node Descriptor information of the destination device.

4.5.3.2 Usage

SREQ:

1	1	1	2	2
Length = 0x04	Cmd0 = 0x25	Cmd1 = 0x02	DstAddr	NWKAddrOfInterest

Attributes:

Attribute	Length (byte)	Description
DstAddr	2	Specifies NWK address of the device generating the inquiry.
NWKAddrOfInterest	2	Specifies NWK address of the destination device being queried.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x02	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	Status is either Success (0) or Failure (1).

4.5.4 ZDO_POWER_DESC_REQ

4.5.4.1 Description:

This command is generated to inquire about the Power Descriptor information of the destination device.

4.5.4.2 Usage

SREQ:

1	1	1	2	2
Length = 0x04	Cmd0 = 0x25	Cmd1 = 0x03	DstAddr	NWKAddrOfInterest

Attributes:

Attribute	Length (byte)	Description
DstAddr	2	Specifies NWK address of the device generating the inquiry.
NWKAddrOfInterest	2	Specifies NWK address of the destination device being queried.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x03	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	Status is either Success (0) or Failure (1).

4.5.5 ZDO_SIMPLE_DESC_REQ

4.5.5.1 Description

This command is generated to inquire as to the Simple Descriptor of the destination device's Endpoint.

4.5.5.2 Usage

SREQ:

1	1	1	2	2	1
Length = 0x05	Cmd0 = 0x25	Cmd1 = 0x04	DstAddr	NWKAddrOfInterest	Endpoint

Attributes:

Attribute	Length (byte)	Description
DstAddr	2	Specifies NWK address of the device generating the inquiry.
NWKAddrOfInterest	2	Specifies NWK address of the destination device being queried.
Endpoint	1	Specifies the application endpoint the data is from.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x04	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	Status is either Success (0) or Failure (1).

4.5.6 ZDO_ACTIVE_EP_REQ

4.5.6.1 Description

This command is generated to request a list of active endpoints from the destination device.

4.5.6.2 Usage

SREQ:

1	1	1	2	2
Length = 0x04	Cmd0 = 0x25	Cmd1 = 0x05	DstAddr	NWKAddrOfInterest

Attributes:

Attribute	Length (byte)	Description
DstAddr	2	Specifies NWK address of the device generating the inquiry.
NWKAddrOfInterest	2	Specifies NWK address of the destination device being queried.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x05	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	Status is either Success (0) or Failure (1).

4.5.7 ZDO_MATCH_DESC_REQ

4.5.7.1 Description

This command is used to send a match descriptor request, which is used to find devices that match the given criteria. The device that receives this request will first perform a match on the profile ID. If the profile IDs do not match, then the match is unsuccessful and no further matching is performed. If the profile IDs match, a match on the input cluster list is performed. If at least one matching input cluster is found, no further matching is performed. If the device that receives this request is unable to find any matching input clusters, a match on the output cluster list is performed. If the device is unable to find any matching output clusters, the match is unsuccessful.

4.5.7.2 Usage

SREQ:

1	1	1	2	2	2
Length = 0x08-0x48	Cmd0 = 0x25	Cmd1 = 0x06	DstAddr	NwkAddrOfInterest	ProfileID

1	0-32	1	0-32
NumInClusters	InClusterList	NumOutClusters	OutClusterList

Attributes:

Attribute	Length (byte)	Description
DstAddr	2	Specifies NWK address of the device generating the inquiry.
NWKAddrOfInterest	2	Specifies NWK address of the destination device being queried.
ProfileID	2	Specifies the profile ID of the device
NumInClusters	1	Specifies the number of IDs in the InClusterList.
InClusterList	0-32	Contains the input cluster IDs (2bytes each)
NumOutClusters	1	Specifies the number of IDs in the OutClusterList.
OutClusterList	0-32	Contains the output cluster IDs (2bytes each)

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x06	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	Status is either Success (0) or Failure (1).

4.5.8 ZDO_COMPLEX_DESC_REQ

4.5.8.1 Description

This command is generated to request for the destination device's complex descriptor.

4.5.8.2 Usage

SREQ:

1	1	1	2	2
Length = 0x04	Cmd0 = 0x25	Cmd1 = 0x07	DstAddr	NWKAddrOfInterest

Attributes:

Attribute	Length (byte)	Description
DstAddr	2	Specifies NWK address of the device generating the inquiry.
NWKAddrOfInterest	2	Specifies NWK address of the destination device being queried.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x07	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	Status is either Success (0) or Failure (1).

4.5.9 ZDO_USER_DESC_REQ

4.5.9.1 Description

This command is generated to request for the destination device's user descriptor.

4.5.9.2 Usage

SREQ:

1	1	1	2	2
Length = 0x04	Cmd0 = 0x25	Cmd1 = 0x08	DstAddr	NWKAddrOfInterest

Attributes:

Attribute	Length (byte)	Description
DstAddr	2	Specifies destination NWK address.
NWKAddrOfInterest	2	Specifies NWK address of the device the query is intended for.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x08	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	Status is either Success (0) or Failure (1).

4.5.10 ZDO_DEVICE_ANNCE

4.5.10.1 Description

This command will cause the CC2530 ZNP device to issue a “Device announce” broadcast packet to the network. This is typically used by an end-device to announce itself to the network.

4.5.10.2 Usage

SREQ:

1	1	1	2	8	1
Length = 0x0B	Cmd0 = 0x25	Cmd1 = 0x0A	NwkAddr	IEEEAddr	Capabilites

Attributes:

Attribute	Length (byte)	Description
NwkAddr	2	Specifies network address of the device generating the request.
IEEEAddr	8	Specifies the 64 bit IEEE Address of the device being announced.
Capabilites	1	Specifies MAC capabilities Bit: 0 – Alternate PAN Coordinator 1 – Device type: 1- ZigBee Router; 0 – End Device 2 – Power Source: 1 Main powered 3 – Receiver on when idle 4 – Reserved 5 – Reserved 6 – Security capability 7 – Reserved

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x0A	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	Status is either Success (0) or Failure (1).

4.5.11 ZDO_USER_DESC_SET

4.5.11.1 Description

This command is generated to write a User Descriptor value to the targeted device

4.5.11.2 Usage

SREQ:

1	1	1	2	2	1	0-16
Length = 0x05-0x15	Cmd0 = 0x25	Cmd1 = 0x0B	DstAddr	NWKAddrOfInterest	Len	UserDescriptor

Attributes:

Attribute	Length (byte)	Description
DstAddr	2	Specifies network address of the device generating the set request.
NWKAddrOfInterest	2	Specifies NWK address of the destination device being queried.
Len	1	Specifies the length of the user descriptor.
UserDescriptor	0-16	User descriptor array (can be up to 16 bytes).

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x0B	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	Status is either Success (0) or Failure (1).

4.5.12 ZDO_SERVER_DISC_REQ

4.5.12.1 Description

The command is used for local device to discover the location of a particular system server or servers as indicated by the ServerMask parameter. The destination addressing on this request is ‘broadcast to all RxOnWhenIdle devices’.

4.5.12.2 Usage

SREQ:

1	1	1	2
Length = 0x02	Cmd0 = 0x25	Cmd1 = 0x0C	ServerMask

Attributes:

Attribute	Length (byte)	Description
ServerMask	2	Specifies the system server capabilities of the device.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x0C	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	Status is either Success (0) or Failure (1).

4.5.13 ZDO_END_DEVICE_BIND_REQ

4.5.13.1 Description

This command is generated to request an End Device Bind with the destination device.

4.5.13.2 Usage

SREQ:

1	1	1	2	2	1
Length = 0x09-0x49	Cmd0 = 0x25	Cmd1 = 0x20	DstAddr = 0x0000	LocalCoordinator	Endpoint

2	1	0-32	1	0-32
ProfileID	NumInClusters	InClusterList	NumOutClusters	OutClusterList

Attributes:

Attribute	Length (byte)	Description
DstAddr	2	Destination address is always 0x0000.
LocalCoordinator	2	Specifies local coordinator's short address. In the case of source binding, it's the short address of the source address.
IEEE	8	Local coordinator's IEEE address
Endpoint	1	Device's endpoint.
ProfileID	2	Specifies the profile ID of the device.
NumInClusters	1	Specifies the number of IDs in the InClusterList.
InClusterList	0-32	Contains the input cluster IDs (2bytes each)
NumOutClusters	1	Specifies the number of IDs in the OutClusterList.
OutClusterList	0-32	Contains the output cluster IDs (2bytes each)

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x20	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	Status is either Success (0) or Failure (1).

4.5.14 ZDO_BIND_REQ

4.5.14.1 Description

This command is generated to request a Bind.

4.5.14.2 Usage

SREQ:

1	1	1	2	8	1	2
Length = 0x10-0x17	Cmd0 = 0x25	Cmd1 = 0x21	DstAddr	SrcAddress	SrcEndpoint	ClusterID

1	8	1
DstAddrMode	DstAddress	DstEndpoint

Attributes:

Attribute	Length (byte)	Description
DstAddr	2	Specifies the network address of the device to send message.
SrcAddress	8	Specifies the 64 bit binding source IEEE address
SrcEndpoint	1	Specifies the binding source endpoint.
ClusterID	2	Specifies the cluster ID to match in messages.
DstAddrMode	1	Specifies binding destination address mode: Mode / Value / Description ADDRESS_NOT_PRESENT / 0x00 / Address Not Present GROUP_ADDRESS / 0x01 / Group address ADDRESS_16_BIT / 0x02 / Address 16 bit ADDRESS_64_BIT / 0x03 / Address 64 bit BROADCAST / 0xFF / Broadcast
DstAddress	8	Binding destination IEEE address. The field is 8-bytes long. However, depending on the value of the DstAddrMode field, only lowest order 2 bytes could be significant. Not to be confused with DstAddr.
DstEndpoint	1	Specifies the binding destination endpoint.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x21	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	Status is either Success (0) or Failure (1).

4.5.15 ZDO_UNBIND_REQ

4.5.15.1 Description

This command is generated to request an UnBind

4.5.15.2 Usage

SREQ:

1	1	1	2	8	1	2
Length = 0x10-0x17	Cmd0 = 0x25	Cmd1 = 0x22	DstAddr	SrcAddress	SrcEndpoint	ClusterID

1	2/8	0/1
DstAddrMode	DstAddress	DstEndpoint

Attributes:

Attribute	Length (byte)	Description
DstAddr	2	Specifies destination network address of the device to send bind request.
SrcAddress	8	Specifies 64 bit binding source IEEE address.
SrcEndpoint	1	Specifies the binding source endpoint.
ClusterID	2	Specifies cluster ID to match in messages.
DstAddrMode	1	Specifies 64 bit binding destination address mode: Mode / Value / Description ADDRESS_NOT_PRESENT / 0x00 / Address Not Present GROUP_ADDRESS / 0x01 / Group address ADDRESS_16_BIT / 0x02 / Address 16 bit ADDRESS_64_BIT / 0x03 / Address 64 bit BROADCAST / 0xFF / Broadcast
DstAddress	8	Specifies 64 bit binding destination IEEE address. Not to be confused with DstAddr.
DstEndpoint	1	Specifies the binding destination endpoint

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x22	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	Status is either Success (0) or Failure (1).

4.5.16 ZDO_MGMT_NWK_DISC_REQ

4.5.16.1 Description

This command is generated to request the destination device to perform a network discovery.

4.5.16.2 Usage

SREQ:

1	1	1	2	4	1	1
Length = 0x08	Cmd0 = 0x25	Cmd1 = 0x30	DstAddr	ScanChannels	ScanDuration	StartIndex

Attributes:

Attribute	Length (byte)	Description
DstAddr	2	Specifies the network address of the device performing the discovery.
ScanChannels	4	Specifies the Bit Mask for channels to scan: Channel / Value NONE 0x00000000 ALL_CHANNELS 0x07FFF800 CHANNEL 11 0x00000800 CHANNEL 12 0x00001000 CHANNEL 13 0x00002000 CHANNEL 14 0x00004000 CHANNEL 15 0x00008000 CHANNEL 16 0x00010000 CHANNEL 17 0x00020000 CHANNEL 18 0x00040000 CHANNEL 19 0x00080000 CHANNEL 20 0x00100000 CHANNEL 21 0x00200000 CHANNEL 22 0x00400000 CHANNEL 23 0x00800000 CHANNEL 24 0x01000000 CHANNEL 25 0x02000000 CHANNEL 26 0x04000000
ScanDuration	1	Specifies the scanning time. Valid range is 0-14.
StartIndex	1	Specifies where to start in the response array list. The result may contain more entries than can be reported, so this field allows the user to retrieve the responses anywhere in the array list.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x30	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	Status is either Success (0) or Failure (1).

4.5.17 ZDO_MGMT_LQI_REQ

4.5.17.1 Description

This command is generated to request the destination device to return its neighbor table. See section 4.5.44 on ZDO_MGMT_LQI_RSP for the fields contained in each neighbor table entry.

4.5.17.2 Usage

SREQ:

1	1	1	2	1
Length = 0x03	Cmd0 = 0x25	Cmd1 = 0x31	DstAddr	StartIndex

Attributes:

Attribute	Length (byte)	Description
DstAddr	2	Specifies the network address of the device generating the query.
StartIndex	1	Specifies where to start in the response array list. The result may contain more entries than can be reported, so this field allows the user to retrieve the responses anywhere in the array list.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x31	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	Status is either Success (0) or Failure (1).

4.5.18 ZDO_MGMT_RTG_REQ

4.5.18.1 Description

This command is generated to request the Routing Table of the destination device.

4.5.18.2 Usage

SREQ:

1	1	1	2	1
Length = 0x03	Cmd0 = 0x25	Cmd1 = 0x32	DstAddr	StartIndex

Attributes:

Attribute	Length (byte)	Description
DstAddr	2	Specifies the network address of the device generating the query.
StartIndex	1	Specifies where to start in the response array list. The result may contain more entries than can be reported, so this field allows the user to retrieve the responses anywhere in the array list.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x32	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	Status is either Success (0) or Failure (1).

4.5.19 ZDO_MGMT_BIND_REQ

4.5.19.1 Description

This command is generated to request the Binding Table of the destination device.

4.5.19.2 Usage

SREQ:

1	1	1	2	1
Length = 0x03	Cmd0 = 0x25	Cmd1 = 0x33	DstAddr	StartIndex

Attributes:

Attribute	Length (byte)	Description
DstAddr	2	Specifies the network address of the device being queried.
StartIndex	1	Specifies where to start in the response array list. The result may contain more entries than can be reported, so this field allows the user to retrieve the responses anywhere in the array list.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x33	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	Status is either Success (0) or Failure (1).

4.5.20 ZDO_MGMT_LEAVE_REQ

4.5.20.1 Description

This command is generated to request a Management Leave Request for the target device and is used to remove devices from the network.

4.5.20.2 Usage

SREQ:

1	1	1	2	8	1
Length = 0x0B	Cmd0 = 0x25	Cmd1 = 0x34	DstAddr	DeviceAddr	RemoveChildren/Rejoin

Attributes:

Attribute	Length (byte)	Description
DstAddr	2	Specifies the network address of the device generating the request.
DeviceAddress	8	Specifies the 64 bit IEEE Address of the target device you want to leave.
RemoveChildren/Rejoin	1	This field has a value of 1 if the device being asked to leave the network is also being asked to remove its child devices, if any. Otherwise it has a value of 0. Currently, the stack profile of Home Control specifies that this field should always be set to 0.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x34	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	Status is either Success (0) or Failure (1).

4.5.21 ZDO_MGMT_DIRECT_JOIN_REQ

4.5.21.1 Description

This command is generated to request the Management Direct Join Request of a designated device.

4.5.21.2 Usage

SREQ:

1	1	1	2	8	1
Length = 0x0B	Cmd0 = 0x25	Cmd1 = 0x35	DstAddr	DeviceAddr	CapInfo

Attributes:

Attribute	Length (byte)	Description
DstAddr	2	Network address of the device to which the device specified in DeviceAddress is to join.
DeviceAddress	8	The 64 bit IEEE Address of the device you want to be joined to the device at DstAddr.
CapInfo	1	Specifies the operating capabilities of the device being directly joined. Bit weighted values follow: Bit: 0 – Alternate PAN Coordinator 1 – Device type: 1- ZigBee Router; 0 – End Device 2 – Power Source: 1 Main powered 3 – Receiver on when idle 4 – Reserved 5 – Reserved 6 – Security capability 7 – Reserved

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x35	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	Status is either Success (0) or Failure (1).

4.5.22 ZDO_MGMT_PERMIT_JOIN_REQ

4.5.22.1 Description

This command is generated to set the Permit Join for the destination device

4.5.22.2 Usage

SREQ:

1	1	1	2	1	1
Length = 0x04	Cmd0 = 0x25	Cmd1 = 0x36	DstAddr	Duration	TCSignificance

Attributes:

Attribute	Length (byte)	Description
DstAddr	2	Specifies the network address of the destination device whose Permit Join information is to be modified.
Duration	1	Specifies the duration to permit joining. 0 = join disabled. 0xff = join enabled. 0x01-0xfe = number of seconds to permit joining.
TCSignificance	1	Trust Center Significance.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x36	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	Status is either Success (0) or Failure (1).

4.5.23 ZDO_MGMT_NWK_UPDATE_REQ

4.5.23.1 Description

This command is provided to allow updating of network configuration parameters or to request information from devices on network conditions in the local operating environment. Upon receipt, the remote device shall determine from the contents of the ScanDuration parameter whether this request is an update to the *ChannelMask* and *NwkManagerAddr* parameters, a channel change command, or a request to scan channels and report the results.

4.5.23.2 Usage

SREQ:

1	1	1	2	1	4
Length = 0x0B	Cmd0 = 0x25	Cmd1 = 0x37	DstAddr	DstAddrMode	ChannelMask

1	1	2
ScanDuration	ScanCount	NwkManagerAddr

Attributes:

<table>
<tr><th>Attribute</th><th>Length (byte)</th><th>Description</th></tr>
<tr><td>DstAddr</td><td>2</td><td>Short address of the destination device(s). The destination addressing on this primitive can be unicast or broadcast to all devices for which macRxOnWhenIdle=TRUE (i.e., 0xFFFD)</td></tr>
<tr><td>DstAddrMode</td><td>1</td><td>Destination address mode:
<table>
<tr><th>Mode</th><th>Value</th><th>Description</th></tr>
<tr><td>ADDRESS_16_BIT</td><td>0x02</td><td>Address 16 bit</td></tr>
<tr><td>BROADCAST</td><td>0x0F</td><td>Broadcast</td></tr>
</table></td></tr>
<tr><td>ChannelMask</td><td>4</td><td>A bitmap indicating which channels are to be scanned:
<table>
<tr><th>Channel</th><th>Value</th></tr>
<tr><td>NONE</td><td>0x00000000</td></tr>
<tr><td>ALL_CHANNELS</td><td>0x07FFF800</td></tr>
<tr><td>CHANNEL 11</td><td>0x00000800</td></tr>
<tr><td>CHANNEL 12</td><td>0x00001000</td></tr>
<tr><td>CHANNEL 13</td><td>0x00002000</td></tr>
<tr><td>CHANNEL 14</td><td>0x00004000</td></tr>
<tr><td>CHANNEL 15</td><td>0x00008000</td></tr>
<tr><td>CHANNEL 16</td><td>0x00010000</td></tr>
<tr><td>CHANNEL 17</td><td>0x00020000</td></tr>
<tr><td>CHANNEL 18</td><td>0x00040000</td></tr>
<tr><td>CHANNEL 19</td><td>0x00080000</td></tr>
</table></td></tr>
</table>

			CHANNEL 20	0x00100000
			CHANNEL 21	0x00200000
			CHANNEL 22	0x00400000
			CHANNEL 23	0x00800000
			CHANNEL 24	0x01000000
			CHANNEL 25	0x02000000
			CHANNEL 26	0x04000000
ScanDuration	1	Specifies the MAC scan duration. Valid range is 0x00-0x05, 0xFE, or 0xFF. The value of the ScanDuration parameter specifies the type of request. See the ZigBee specification for more details.		
ScanCount	1	This field represents the number of energy scans to be conducted and reported		
NwkManagerAddr	2	Indicates the NWK address for the device with the Network Manager bit set in its Node Descriptor		

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x37	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	Status is either Success (0) or Failure (1).

4.5.24 ZDO_STARTUP_FROM_APP

4.5.24.1 Description

This command starts the device in the network.

4.5.24.2 Usage

SREQ:

1	1	1	2
Length = 0x01	Cmd0 = 0x25	Cmd1 = 0x40	StartDelay

Attributes:

Attribute	Length (byte)	Description
StartDelay	2	Specifies the time delay before the device starts in milliseconds.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x40	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	0x00 – Restored network state 0x01 – New network state 0x02 – Leave and not Started

4.5.25 ZDO_AUTO_FIND_DESTINATION

4.5.25.1 Description

This function will issue a Match Descriptor Request for the requested endpoint. This message will generate a broadcast message. Note that there is no response to the host processor for this message. If there is a successful response to the match descriptor request packet, the binding table on the device will be automatically updated.

4.5.25.2 Usage

AREQ:

1	1	1	1
Length = 0x01	Cmd0 = 0x45	Cmd1 = 0x41	Endpoint

Attributes:

Attribute	Length (byte)	Description
Endpoint	1	Specifies which endpoint to issue the Match Descriptor request for.

4.5.26 ZDO_SET_LINK_KEY

4.5.26.1 Description

This Command sets the application or trust center link key for a given device.

4.5.26.2 Usage

SREQ:

1	1	1	2	8	16
Length = 0x01	Cmd0 = 0x25	Cmd1 = 0x23	ShortAddr	IEEEaddr	LinkKeyData

Attributes:

Attribute	Length (byte)	Description
ShortAddr	2	Specifies the short address of the device.
IEEEaddr	8	Specifies the extended (IEEE) address of the device.
LinkKeyData	16	128 bit link key data of the device.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x23	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	0x00 – Success 0x01 – Fail to add to address manager. 0x11 – Security manager key table full

4.5.27 ZDO_REMOVE_LINK_KEY

4.5.27.1 Description

This command removes the application or trust center link key of a given device.

4.5.27.2 Usage

SREQ:

1	1	1	8
Length = 0x01	Cmd0 = 0x25	Cmd1 = 0x24	IEEEaddr

Attributes:

Attribute	Length (byte)	Description
IEEEaddr	8	Specifies the extended (IEEE) address of the device.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x24	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	0x00 – Success 0xC8 – Unknown device.

4.5.28 ZDO_GET_LINK_KEY

4.5.28.1 Description

This command retrieves the application or trust center link key of a given device.

4.5.28.2 Usage

SREQ:

1	1	1	8
Length = 0x08	Cmd0 = 0x25	Cmd1 = 0x25	IEEEaddr

Attributes:

Attribute	Length (byte)	Description
IEEEaddr	8	Specifies the extended (IEEE) address of the device.

SRSP:

1	1	1	1	8	16
Length = 0x19	Cmd0 = 0x65	Cmd1 = 0x25	Status	IEEEAddr	LinkKeyData

Attributes:

Attribute	Length (byte)	Description
Status	1	0x00 – Success 0xC8 – Unknown device.
IEEEAddr	8	IEEE address of the device
LinkKeyData	16	Link key data of the device.

4.5.29 ZDO_NWK_DISCOVERY_REQ

4.5.29.1 Description

This command is used to initiate a network discovery (active scan).

4.5.29.2 Usage

SREQ:

1	1	1	4	1
Length = 0x05	Cmd0 = 0x25	Cmd1 = 0x26	Scan Channels	Scan Duration

Attributes:

Attribute	Length (byte)	Description
Scan Channels	4	Bit mask for channels to scan. Type: ZIGBEE_CHANNELS. Channel / Value: NONE 0x00000000 ALL_CHANNELS 0x07FFF800 CHANNEL 11 0x00000800 CHANNEL 12 0x00001000 CHANNEL 13 0x00002000 CHANNEL 14 0x00004000 CHANNEL 15 0x00008000 CHANNEL 16 0x00010000 CHANNEL 17 0x00020000 CHANNEL 18 0x00040000 CHANNEL 19 0x00080000 CHANNEL 20 0x00100000 CHANNEL 21 0x00200000 CHANNEL 22 0x00400000 CHANNEL 23 0x00800000 CHANNEL 24 0x01000000 CHANNEL 25 0x02000000 CHANNEL 26 0x04000000
Scan Duration	1	A value used to calculate the length of time to spend scanning each channel

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x26	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	0x00 – Success 0x02 – Invalid Parameter 0xC2 – ZNwkInvalidRequest. if the device is already on a network. User ZDO_MGMT_NWK_DISC_REQ instead. Or leave the network first, then initiate the request. 0xFC – MAC_SCAN_IN_PROGRESS if a channel change is in

		progress. 0x1A – MAC_NO_RESOURCE if the operation could not complete because no memory resource were available

4.5.30 ZDO_JOIN_REQ

4.5.30.1 Description

This command is used to request the device to join itself to a parent device on a network.

4.5.30.2 Usage

SREQ:

1	1	1	1	2
Length = 0x0F	Cmd0 = 0x25	Cmd1 = 0x27	Logical Channel	Pan ID
8	**2**	**1**	**1**	
Extended Pan ID	Chosen Parent	Parent Depth	Stack Profile	

Attributes:

Attribute	Length (byte)	Description
Logical Channel	1	Channel where the PAN is located.
Pan ID	2	Id of PAN to join
Extended Pan ID	8	64-bit extended PAN ID (ver. 1.1 only). If not v1.1 or don't care, use all 0xFF
Chosen Parent	2	Short address of the parent device chosen to join
Parent Depth	1	Depth of the parent
Stack Profile	1	Stack profile of the network to join

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x27	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	0x00 – Success 0xC2 – ZNwkInvalidRequest. if device is already on a network. Leave the network first, then try to join again. 0xC3 – ZNwkNotPermitted if chosen router is not a valid short address.

4.5.31 ZDO_NWK_ADDR_RSP

4.5.31.1 Description

This command is issued by the App processor to return the results from a ZDO_NWK_ADDR_REQ, as long as the ZCD_NV_ZDO_DIRECT_CB configuration item is set to TRUE.

4.5.31.2 Usage

AREQ:

1	1	1	1	8	2
Length = 0x0D-0x53	Cmd0 = 0x45	Cmd1 = 0x80	Status	IEEEAddr	NwkAddr

1	1	0-70
StartIndex	NumAssocDev	AssocDevList

Attributes:

Attribute	Length (byte)	Description
Status	1	This field indicates either SUCCESS or FAILURE.
IEEEAddr	8	Specifies the 64 bit IEEE address of source device.
NwkAddr	2	Specifies the short network address of responding device.
StartIndex	1	Specifies the starting index into the list of associated devices for this report.
NumAssocDev	1	Specifies the number of associated devices.
AssocDevList	0-70	Contains the list of short addresses (2 bytes each) of the associated devices. This list can be a partial list if the entire list doesn't fit into a packet. If it is a partial list, the starting index is StartIndex.

4.5.32 ZDO_IEEE_ADDR_RSP

4.5.32.1 Description

This callback message is in response to the ZDO IEEE Address Request, as long as the ZCD_NV_ZDO_DIRECT_CB configuration item is set to TRUE.

4.5.32.2 Usage

AREQ:

1	1	1	1	8	2
Length = 0x0D-0x53	Cmd0 = 0x45	Cmd1 = 0x81	Status	IEEEAddr	NwkAddr

1	1	0-70
StartIndex	NumAssocDev	AssocDevList

Attributes:

Attribute	Length (byte)	Description
Status	1	This field indicates either SUCCESS or FAILURE.
IEEEAddr	8	Specifies the 64 bit IEEE address of source device.
NwkAddr	2	Specifies the short network address of responding device.
StartIndex	1	Specifies the starting index into the list of associated devices for this report.
NumAssocDev	1	Specifies the number of associated devices.
AssocDevList	0-70	Contains the list of short addresses (2 bytes each) for associated devices. This list can be a partial list if the entire list doesn't fit into a packet. If it is a partial list, the starting index is StartIndex.

4.5.33 ZDO_NODE_DESC_RSP

4.5.33.1 Description

This callback message is in response to the ZDO Node Descriptor Request, as long as the ZCD_NV_ZDO_DIRECT_CB configuration item is set to TRUE. See the ZigBee specification for more detailed descriptions for each of the individual fields.

4.5.33.2 Usage

AREQ:

1	1	1	2	1	2
Length = 0x12	Cmd0 = 0x45	Cmd1 = 0x82	SrcAddr	Status	NwkAddr

1	1	1	2
LogicalType/ ComplexDescAvailable/ UserDescAvailable	APSFlags/ FrequencyBand	MACCapabilityFlags	ManufacturerCode

1	2	2	2	1
MaxBufferSize	MaxTransferSize	ServerMask	MaxOutTransferSize	DescriptorCapabilities

Attributes:

Attribute	Length (byte)	Description
SrcAddr	2	The message's source network address
Status	1	This field indicates either SUCCESS or FAILURE
NWKAddrOfInterest	2	Device's short address of this Node descriptor
LogicalType/ ComplexDescriptorAvailable/ UserDescriptorAvailable	1	Logical Type: Bit 0-2 – Description / Value: ZigBee Coordinator / 0; ZigBee Router / 1; ZigBee End Device / 2 ComplexDescriptorAvailable: Bit 3 – Indicates if complex descriptor is available for the node UserDescriptorAvailable: Bit 4 – Indicates if user descriptor is available for the node Reserved – Bit 5-7 – reserved for future use
APSFlags/FrequencyBand	1	APSFlags – Bit 0-4 – Node Flags assigned for APS NodeFrequencyBand – Bit 5-7 – Identifies node frequency band capabilities

<table>
<tr><td>MacCapabilitiesFlags</td><td>1</td><td>Capability flags stored for the MAC<table><tr><th>Description</th><th>Value</th></tr><tr><td>CAPINFO_DEVICETYPE_RFD</td><td>0x00</td></tr><tr><td>CAPINFO_ALTPANCOORD</td><td>0x01</td></tr><tr><td>CAPINFO_DEVICETYPE_FFD</td><td>0x02</td></tr><tr><td>CAPINFO_POWER_AC</td><td>0x04</td></tr><tr><td>CAPINFO_RCVR_ON_IDLE</td><td>0x08</td></tr><tr><td>CAPINFO_SECURITY_CAPABLE</td><td>0x40</td></tr><tr><td>CAPINFO_ALLOC_ADDR</td><td>0x80</td></tr></table></td></tr>
<tr><td>ManufacturerCode</td><td>2</td><td>Specifies a manufacturer code that is allocated by the ZigBee Alliance, relating to the manufacturer to the device</td></tr>
<tr><td>MaxBufferSize</td><td>1</td><td>Indicates size of maximum NPDU. This field is used as a high level indication for management</td></tr>
<tr><td>MaxInTransferSize</td><td>2</td><td>Indicates maximum size of Transfer up to 0x7fff (This field is reserved in version 1.0 and shall be set to zero)</td></tr>
<tr><td>ServerMask</td><td>2</td><td>Bit 0 - Primary Trust Center
1 - Backup Trust Center
2 - Primary Binding Table Cache
3 - Backup Binding Table Cache
4 - Primary Discovery Cache
5 - Backup Discovery Cache</td></tr>
<tr><td>MaxOutTransferSize</td><td>2</td><td>Indicates maximum size of Transfer up to 0x7fff</td></tr>
<tr><td>DescriptorCapabilities</td><td>1</td><td>Specifies the Descriptor capabilities</td></tr>
</table>

4.5.34 ZDO_POWER_DESC_RSP

4.5.34.1 Description

This callback message is in response to the ZDO Power Descriptor Request, as long as the ZCD_NV_ZDO_DIRECT_CB configuration item is set to TRUE. See the ZigBee specification for more detailed descriptions for each of the individual fields.

4.5.34.2 Usage

AREQ:

1	1	1	2	1	2
Length = 0x07	Cmd0 = 0x45	Cmd1 = 0x83	SrcAddr	Status	NwkAddr

1	1
CurrentPowerMode/AvailablePowerSources	CurrentPowerSource/CurrentPowerSourceLevel

Attributes:

Attribute	Length (byte)	Description
SrcAddr	2	Specifies the message's source network address.
Status	1	This field indicates either SUCCESS or FAILURE.
NWKAddr	2	Device short address that this response describes.
CurrentPowerMode/AvailablePowerSources	1	- CurrentPowerMode: bits 3-0 - AvailablePowerSources: bits 7-4
CurrentPowerSource/CurrentPowerSourceLevel	1	- CurrentPowerSource: bits 3-0 - CurrentPowerSourceLevel: bits 7-4

4.5.35 ZDO_SIMPLE_DESC_RSP

4.5.35.1 Description

This callback message is in response to the ZDO Simple Descriptor Request, as long as the ZCD_NV_ZDO_DIRECT_CB configuration item is set to TRUE.

4.5.35.2 Usage

AREQ:

1	1	1	2	1	2	1
Length = 0x06-4E	Cmd0 = 0x45	Cmd1 = 0x84	SrcAddr	Status	NwkAddr	Len

1	2	2	1
Endpoint	ProfileID	DeviceID	DeviceVersion

1	0-32	1	0-32
NumInClusters	InClusterList	NumOutClusters	OutClusterList

Attributes:

Attribute	Length (byte)	Description
SrcAddr	2	Specifies the message's source network address.
Status	1	This field indicates either SUCCESS or FAILURE.
NWKAddr	2	Specifies Device's short address that this response describes.
Len	1	Specifies the length of the simple descriptor
Endpoint	1	Specifies Endpoint of the device
ProfileId	2	The profile ID for this endpoint.
DeviceID	2	The Device Description ID for this endpoint.
DeviceVersion	1	Defined as the following format 0 – Version 1.00 0x01-0x0F – Reserved.
NumInClusters	1	The number of input clusters in the InClusterList.
InClusterList	0-32	List of input cluster IDs (2 bytes each) supported.
NumOutClusters	1	The number of output clusters in the OutClusterList.
OutClusterList	0-32	List of output cluster IDs (2 bytes each) supported.

4.5.36 ZDO_ACTIVE_EP_RSP

4.5.36.1 Description

This callback message is in response to the ZDO Active Endpoint Request, as long as the ZCD_NV_ZDO_DIRECT_CB configuration item is set to TRUE.

4.5.36.2 Usage

AREQ:

1	1	1	2	1	2	1
Length = 0x06-0x53	Cmd0 = 0x45	Cmd1 = 0x85	SrcAddr	Status	NwkAddr	ActiveEPCount

0-77
ActiveEPList

Attributes:

Attribute	Length (byte)	Description
SrcAddr	2	The message's source network address.
Status	1	This field indicates either SUCCESS or FAILURE.
NWKAddr	2	Device's short address that this response describes.
ActiveEPCount	1	Number of active endpoint in the list
ActiveEPList	0-77	Array of active endpoints (1 byte each) on this device.

4.5.37 ZDO_MATCH_DESC_RSP

4.5.37.1 Description

This callback message is in response to the ZDO Match Descriptor Request, as long as the ZCD_NV_ZDO_DIRECT_CB configuration item is set to TRUE.

4.5.37.2 Usage

AREQ:

1	1	1	2	1	2	1
Length = 0x06-0x53	Cmd0 = 0x45	Cmd1 = 0x86	SrcAddr	Status	NwkAddr	MatchLength

0-77
MatchList

Attributes:

Attribute	Length (byte)	Description
SrcAddr	2	The message's source network address.
Status	1	This field indicates either SUCCESS or FAILURE.
NWKAddr	2	Device's short address that this response describes.
MatchLength	1	The count of endpoints on the remote device that match the request criteria
MatchList	0-77	List of bytes, each represents an 8 bit endpoint

4.5.38 ZDO_COMPLEX_DESC_RSP

4.5.38.1 Description

This callback message is in response to the ZDO Complex Descriptor Request, as long as the ZCD_NV_ZDO_DIRECT_CB configuration item is set to TRUE.

4.5.38.2 Usage

AREQ:

1	1	1	2	1	2	1
Length = 0x06-0x53	Cmd0 = 0x45	Cmd1 = 0x87	SrcAddr	Status	NwkAddr	ComplexLength

0-77
ComplexList

Attributes:

Attribute	Length (byte)	Description
SrcAddr	2	The message's source network address.
Status	1	This field indicates either SUCCESS or FAILURE.
NWKAddr	2	Device's short address that this response describes.
ComplexLength	1	Length of the complex descriptor.
ComplexDescriptor	0-77	Array of bytes contains the complex descriptor.

4.5.39 ZDO_USER_DESC_RSP

4.5.39.1 Description

This callback message is in response to the ZDO User Descriptor Request, as long as the ZCD_NV_ZDO_DIRECT_CB configuration item is set to TRUE.

4.5.39.2 Usage

AREQ:

1	1	1	2	1	2	1	0-77
Length = 0x06-0x16	Cmd0 = 0x45	Cmd1 = 0x88	SrcAddr	Status	NwkAddr	Len	UserDescriptor

Attributes:

Attribute	Length (byte)	Description
SrcAddr	2	The message's source network address.
Status	1	This field indicates either SUCCESS or FAILURE.
NWKAddr	2	Device's short address that this response describes.
UserLength	1	Length of the complex descriptor.
UserDescriptor	0-77	Array of bytes contains user descriptor.

4.5.40 ZDO_USER_DESC_CONF

4.5.40.1 Description

This confirmation notifies the host processor when the device receives a user descriptor.

4.5.40.2 Usage

AREQ:

1	1	1	2	1	2
Length = 0x05	Cmd0 = 0x45	Cmd1 = 0x89	SrcAddr	Status	NwkAddr

Attributes:

Attribute	Length (byte)	Description
SrcAddr	2	The message's source network address.
Status	1	This field indicates either SUCCESS or FAILURE.
NWKAddr	2	Device's short address that this response describes.

4.5.41 ZDO_SERVER_DISC_RSP

4.5.41.1 Description

This callback message is in response to the ZDO System Service Discovery Request, as long as the ZCD_NV_ZDO_DIRECT_CB configuration item is set to TRUE. Upon receiving the request, remote devices shall compare the ServerMask parameter to the Server Mask field in their own Node descriptor. If no bits are found to match, no action is taken.

4.5.41.2 Usage

AREQ:

1	1	1	2	1	2
Length = 0x05	Cmd0 = 0x45	Cmd1 = 0x8A	SrcAddr	Status	ServerMask

Attributes:

Attribute	Length (byte)	Description
SrcAddr	2	The message's source network address.
Status	1	This field indicates either SUCCESS (0) or FAILURE (1).
Server Mask	2	Each bit signifies one system server capability of the node. The bit setting is defined in the following table:

Bit Number	Assignment
0	Primary Trust Center
1	Backup Trust Center
2	Primary Binding Table Cache
3	Backup Binding Table Cache
4	Primary Discovery Cache
5	Backup Discovery Cache
6– 15	Reserved

4.5.42 ZDO_END_DEVICE_BIND_RSP

4.5.42.1 Description

This callback message is in response to the ZDO End Device Bind Request, as long as the ZCD_NV_ZDO_DIRECT_CB configuration item is set to TRUE.

4.5.42.2 Usage

AREQ:

1	1	1	2	1
Length = 0x03	Cmd0 = 0x45	Cmd1 = 0xA0	SrcAddr	Status

Attributes:

Attribute	Length (byte)	Description
SrcAddr	2	The message's source network address.
Status	1	This field indicates either SUCCESS (0) or FAILURE (1).

4.5.43 ZDO_BIND_RSP

4.5.43.1 Description

This callback message is in response to the ZDO Bind Request, as long as the ZCD_NV_ZDO_DIRECT_CB configuration item is set to TRUE.

4.5.43.2 Usage

AREQ:

1	1	1	2	1
Length = 0x03	Cmd0 = 0x45	Cmd1 = 0xA1	SrcAddr	Status

Attributes:

Attribute	Length (byte)	Description
SrcAddr	2	The message's source network address.
Status	1	This field indicates either SUCCESS (0) or FAILURE (1).

4.5.44 ZDO_UNBIND_RSP

4.5.44.1 Description

This callback message is in response to the ZDO Unbind Request, as long as the ZCD_NV_ZDO_DIRECT_CB configuration item is set to TRUE.

4.5.44.2 Usage

AREQ:

1	1	1	2	1
Length = 0x03	Cmd0 = 0x45	Cmd1 = 0xA2	SrcAddr	Status

Attributes:

Attribute	Length (byte)	Description
SrcAddr	2	The message's source network address.
Status	1	This field indicates either SUCCESS (0) or FAILURE (1).

4.5.45 ZDO_MGMT_NWK_DISC_RSP

4.5.45.1 Description

This callback message is in response to the ZDO Management Network Discovery Request, as long as the ZCD_NV_ZDO_DIRECT_CB configuration item is set to TRUE.

4.5.45.2 Usage

AREQ:

1	1	1	2	1	1	1
Length = 0x06-0x4E	Cmd0 = 0x45	Cmd1 = 0xB0	SrcAddr	Status	NetworkCount	StartIndex

1	0-72
NetworkListCount	NetworkList Records

Attributes:

Attribute	Length (byte)	Description
SrcAddr	2	Source address of the message.
Status	1	This field indicates either SUCCESS or FAILURE.
NetworkCount	1	Total number of entries available in the device.
StartIndex	1	Where in the total number of entries this response starts.
NetworkListCount	1	Number of entries in this response.
NetworkList	List	Array of NetworkList items. NetworkListCount contains the number of items in this table:

Name	Size	Description
PAN ID/Extended PAN ID	2 bytes	PAN ID of the neighbor device
Logical Channel	1 byte	The current logical channel occupied by the network.
Stack Profile / ZigBee Version	1 byte	StackProfile: bits 3-0 ZigBeeVersion: bits 7-4 A ZigBee stack profile identifier indicating the stack profile in use in the discovered network. The version of the ZigBee protocol in use in the discovered network.
Beacon Order / Super frame Order	1 byte	BeaconOrder: bits 3-0 SuperframeOrder: bits 7-4
Permit Joining	1 byte	Permit joining flag

4.5.46 ZDO_MGMT_LQI_RSP

4.5.46.1 Description:

This callback message is in response to the ZDO Management LQI Request, as long as the ZCD_NV_ZDO_DIRECT_CB configuration item is set to TRUE.

4.5.46.2 Usage

AREQ:

1	1	1	2	1	1	1
Length = 0x06-0x48	Cmd0 = 0x45	Cmd1 = 0xB1	SrcAddr	Status	NeighborTableEntries	StartIndex

1	0-66
NeighborTableListCount	NeighborTableListRecords

Attributes:

Attribute	Length (byte)	Description
SrcAddr	2	Source address of the message.
Status	1	This field indicates either SUCCESS or FAILURE.
NeighborTableEntries	1	Total number of entries available in the device.
StartIndex	1	Where in the total number of entries this response starts.
NeighborLqiListCount	1	Number of entries in this response.
NeighborLqiList	Variable	Array of NeighborLqiList items. NeighborLQICount contains the number of items in this table.

Name	Size	Description
ExtendedPanID	8 bytes	Extended PAN ID of the neighbor device
ExtendedAddress	8 bytes	Network extended address
NetworkAddress	2 bytes	Device short address
DeviceType/ RxOnWhenIdle/ Relationship	1 byte	DeviceType: bits 1-0 RxOnWhenIdle: bits 3-2 Relationship: bits 6-4
PermitJoining	1 byte	PermitJoining: bits 1-0
Depth	1 byte	
LQI	1 byte	

4.5.47 ZDO_MGMT_RTG_RSP

4.5.47.1 Description

This callback message is in response to the ZDO Management Routing Table Request, as long as the ZCD_NV_ZDO_DIRECT_CB configuration item is set to TRUE.

4.5.47.2 Usage

AREQ:

1	1	1	2	1	1	1
Length = 0x06-0x51	Cmd0 = 0x45	Cmd1 = 0xB2	SrcAddr	Status	RoutingTableEntries	StartIndex

1	0-75
RoutingTableListCount	RoutingTableListRecords

Attributes:

Attribute	Length (byte)	Description
SrcAddr	2	Source address of the message.
Status	1	This field indicates either SUCCESS or FAILURE.
RoutingTableEntries	1	Total number of entries available in the device.
StartIndex	1	Where in the total number of entries this response starts.
RoutingTableListCount	1	Number of entries in this response.
RoutingTableList	variable	Array of RtgList items. RtgListCount contains the number of items in this table.

Name	Size	Description
Destination Address	2 bytes	Network destination address
Status	1 byte	Route status: bits 2-0 0x00 Active 0x01 Discovery Underway 0x02 Discovery Failed 0x03 Inactive 0x04 – 0x07 Reserved
Next Hop	2 bytes	Next hop network address

4.5.48 ZDO_MGMT_BIND_RSP

4.5.48.1 Description

This callback message is in response to the ZDO Management Binding Table Request, as long as the ZCD_NV_ZDO_DIRECT_CB configuration item is set to TRUE.

4.5.48.2 Usage

AREQ:

1	1	1	2	1	1	1
Length = 0x06-0x51	Cmd0 = 0x45	Cmd1 = 0xB3	SrcAddr	Status	BindingTableEntries	StartIndex

1	0-75
BindingTableListCount	BindingTableListRecords

Attributes:

Attribute	Length (byte)	Description
SrcAddr	2	Source address of the message
Status	1	This field indicates either SUCCESS (0) or FAILURE (1).
BindTableEntries	1	Total number of entries available in the device.
StartIndex	1	Where in the total number of entries this response starts.
BindTableListCount	1	Number of entries in this response.
BindTableList	List	An array of BindList items. BindListCount contains the number of items in this table.

Name	Size	Description
SrcAddr	8 bytes	Binding Entry's source IEEE address
SrcEndpoint	1 byte	Binding Entry's source endpoint
ClusterID	1 byte	Message ID in binding table
DstAddrMode	1 byte	Address mode for binding entry's destination address
DstAddr	8 bytes	Binding Entry's destination IEEE address
DstEndpoint	1 byte	Binding Entry's destination endpoint. For V1.1, this field is only present when the DestAddrMode is 64-bits extended address.

4.5.49 ZDO_MGMT_LEAVE_RSP

4.5.49.1 Description

This callback message is in response to the ZDO Management Leave Request, as long as the ZCD_NV_ZDO_DIRECT_CB configuration item is set to TRUE.

4.5.49.2 Usage

AREQ:

1	1	1	2	1
Length = 0x03	Cmd0 = 0x45	Cmd1 = 0xB4	SrcAddr	Status

Attributes:

Attribute	Length (byte)	Description
SrcAddr	2	Source address of the message
Status	1	This field indicates either SUCCESS (0) or FAILURE (1).

4.5.50 ZDO_MGMT_DIRECT_JOIN_RSP

4.5.50.1 Description

This callback message is in response to the ZDO Management Direct Join Request, as long as the ZCD_NV_ZDO_DIRECT_CB configuration item is set to TRUE.

4.5.50.2 Usage

AREQ:

1	1	1	2	1
Length = 0x03	Cmd0 = 0x45	Cmd1 = 0xB5	SrcAddr	Status

Attributes:

Attribute	Length (byte)	Description
SrcAddr	2	Source address of the message
Status	1	This field indicates either SUCCESS (0) or FAILURE (1).

4.5.51 ZDO_MGMT_PERMIT_JOIN_RSP

4.5.51.1 Description

This callback message is in response to the ZDO Management Permit Join Request, as long as the ZCD_NV_ZDO_DIRECT_CB configuration item is set to TRUE.

4.5.51.2 Usage

AREQ:

1	1	1	2	1
Length = 0x03	Cmd0 = 0x45	Cmd1 = 0xB6	SrcAddr	Status

Attributes:

Attribute	Length (byte)	Description
SrcAddr	2	Source address of the message.
Status	1	This field indicates either SUCCESS (0) or FAILURE (1).

4.5.52 ZDO_STATE_CHANGE_IND

4.5.52.1 Description

This callback message indicates the ZDO state change.

4.5.52.2 Usage

AREQ:

1	1	1	1
Length = 0x01	Cmd0 = 0x45	Cmd1 = 0xC0	State

Attributes:

Attribute	Length (byte)	Description
State	1	Specifies the changed ZDO state. An enumerated list starting from 0 (DEV_HOLD).

Name	Description
DEV_HOLD	Initialized - not started automatically
DEV_INIT	Initialized - not connected to anything
DEV_NWK_DISC	Discovering PAN's to join
DEV_NWK_JOINING	Joining a PAN
DEV_NWK_REJOIN	ReJoining a PAN, only for end devices
DEV_END_DEVICE_UNAUTH	Joined but not yet authenticated by trust center
DEV_END_DEVICE	Started as device after authentication
DEV_ROUTER	Device joined, authenticated and is a router
DEV_COORD_STARTING	Starting as Zigbee Coordinator
DEV_ZB_COORD	Started as Zigbee Coordinator
DEV_NWK_ORPHAN	Device has lost information about its parent

4.5.53 ZDO_END_DEVICE_ANNCE_IND

4.5.53.1 Description

This callback indicates the ZDO End Device Announce, as long as the ZCD_NV_ZDO_DIRECT_CB configuration item is set to TRUE. This can be caused by another device sending out the END_DEVICE_ANNCE message to the network.

4.5.53.2 Usage

AREQ:

1	1	1	2	2	8	1
Length = 0x0D	Cmd0 = 0x45	Cmd1 = 0xC1	SrcAddr	NwkAddr	IEEEAddr	Capabilites

Attributes:

Attribute	Length (byte)	Description
SrcAddr	2	Source address of the message.
NwkAddr	2	Specifies the device's short address.
IEEEAddr	8	Specifies the 64 bit IEEE address of source device.
Capabilites	1	Specifies the MAC capabilities of the device. Bit: 0 – Alternate PAN Coordinator 1 – Device type: 1- ZigBee Router; 0 – End Device 2 – Power Source: 1 Main powered 3 – Receiver on when idle 4 – Reserved 5 – Reserved 6 – Security capability 7 – Reserved

4.5.54 ZDO_MATCH_DESC_RSP_SENT

4.5.54.1 Description

This callback indicates that Match Descriptor Response has been sent.

4.5.54.2 Usage

AREQ:

1	1	1	2
Length = 0x04-0x44	Cmd0 = 0x45	Cmd1 = 0xC2	NwkAddr

1	0-32	1	0-32
NumInClusters	InClusterList	NumOutClusters	OutClusterList

Attributes:

Attribute	Length (byte)	Description
NwkAddr	2	Specifies the device's short address
NumInClusters	1	The number of input clusters in the InClusterList.
InClusterList	0-32	List of input cluster IDs (2 bytes each) that matched.
NumOutClusters	1	The number of output clusters in the OutClusterList.
OutClusterList	0-32	List of output cluster IDs (2 bytes each) that matched.

4.5.55 ZDO_STATUS_ERROR_RSP

4.5.55.1 Description

This message is the default message for error status.

4.5.55.2 Usage

AREQ:

1	1	1	2	1
Length = 0x04-0x44	Cmd0 = 0x45	Cmd1 = 0xC3	SrcAddr	Status

Attributes:

Attribute	Length (byte)	Description
SrcAddr	2	Source address of the message
Status	1	This field indicates either SUCCESS (0) or FAILURE (1).

4.5.56 ZDO_SRC_RTG_IND

4.5.56.1 Description

This message is an indication to inform host device of receipt of a source route to a given device.

4.5.56.2 Usage

AREQ:

1	1	1	2	1	2N
Length = 0x04-0x44	Cmd0 = 0x45	Cmd1 = 0xC4	dstAddr	Relay Count (N)	Relay List

Attributes:

Attribute	Length (byte)	Description
DstAddr	2	Destination of the source route
Relay Count	1	This field indicates number of devices in the relay list of the source route.
Relay List	2N	This field contains the list of devices in the relay list of the source route.It includes a two bytes short address for each device.

4.5.57 ZDO_LEAVE_IND

4.5.57.1 Description

This message is an indication to inform the host of a device leaving the network.

4.5.57.2 Usage

AREQ:

1	1	1	2	8	1	1	1
Length=0x0D	Cmd0=0x45	Cmd1=0xC9	SrcAddr	ExtAddr	Request	Remove	Rejoin

Attributes:

Attribute	Length (byte)	Description
SrcAddr	2	Short address (LSB-MSB) of the source of the leave indication.
ExtAddr	8	Extended address (LSB-MSB) of the source of the leave indication.
Request	1	Boolean, TRUE = request, FALSE = indication.
Remove	1	Boolean, TRUE = remove children.
Rejoin	1	Boolean, TRUE = rejoin.

4.5.58 ZDO_ MSG_CB_REGISTER

4.5.58.1 Description

This command registers for a ZDO callback and used in conjunction with the configuration item ZCD_NV_ZDO_DIRECT_CB. It performs a proxy call to the ZDO_RegisterForZDOMsg() function within the ZNP.

4.5.58.2 Usage

SREQ:

1	1	1	2
Length = 0x02	Cmd0 = 0x25	Cmd1 = 0x3E	ClusterID

Attributes:

Attribute	Length (byte)	Description
ClusterID	2	Specifies the ZDO Cluster Id for which to receive a ZDO callback.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x3E	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	Return value of the call to ZDO_RegisterForZDOMsg(). ZSuccess (0x00) if successful, ZMemError (0x10) indicating failure to allocate memory for this callback registration.

4.5.59 ZDO_ MSG_CB_REMOVE

4.5.59.1 Description

This command removes a registration for a ZDO callback and used in conjunction with the configuration item ZCD_NV_ZDO_DIRECT_CB. It performs a proxy call to the ZDO_RemoveRegisteredCB() function within the ZNP.

4.5.59.2 Usage

SREQ:

1	1	1	2
Length = 0x02	Cmd0 = 0x25	Cmd1 = 0x3F	ClusterID

Attributes:

Attribute	Length (byte)	Description
ClusterID	2	Specifies the ZDO Cluster Id for which to receive a ZDO callback.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x65	Cmd1 = 0x3F	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	Return value of the call to ZDO_RemoveRegisteredCB (). ZSuccess (0x00) if successful, ZFailure (0x01) if registered callback does not exist.

4.5.60 ZDO_ MSG_CB_INCOMING

4.5.60.1 Description

This message is a ZDO callback for a cluster ID that the host requested to receive with a ZDO_MSG_CB_REGISTER command. It is used in conjunction with the configuration item ZCD_NV_ZDO_DIRECT_CB. The format of the Data field maps directly to the format of each cluster ID response as specified in the ZigBee specification. It is up to the host processor to parse this data payload and process accordingly.

4.5.60.2 Usage

AREQ:

1	1	1	2	1	2
Length = 0x09-0x6C	Cmd0 = 0x45	Cmd1 = 0xFF	SrcAddr	WasBroadcast	ClusterID

1	1	2	0 - 99
SecurityUse	SeqNum	MacDstAddr	Data

Attributes:

Attribute	Length (byte)	Description
SrcAddr	2	Short address (LSB-MSB) of the source of the ZDO message.
WasBroadcast	1	This field indicates whether or not this ZDO message was broadcast.
ClusterID	2	The ZDO Cluster Id of this message.
SecurityUse	1	N/A – not used.
SeqNum	1	The sequence number of this ZDO message.
MacDstAddr	2	The MAC destination short address (LSB-MSB) of the ZDO message.
Data	0 - 99	The data that corresponds to the Cluster Id of the message. Without any security (99 bytes), with NWK security (81 bytes), with NWK and APS security (64 bytes).

4.6 UTIL Interface

4.6.1 UTIL_DATA_REQ

4.6.1.1 Description

This command is used to send a one shot MAC MLME Poll Request (or data request).

4.6.1.2 Usage

SREQ:

1	1	1	1
Length = 0x01	Cmd0 = 0x27	Cmd1 = 0x11	SecurityUse

Attributes:

Attribute	Length (byte)	Description
SecurityUse	1	TRUE to request MAC security, but not used for now.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x67	Cmd1 = 0x11	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	A MAC status value from ZComDef.h, but only ZMacSuccess for now (0x00).

4.6.2 UTIL_ADDRMGR_EXT_ADDR_LOOKUP

4.6.2.1 Description

This command is a proxy call to the AddrMgrEntryLookupExt() function. This function takes in the Extended address of the device of interest and returns its Network address. The device of interest must either be a parent or child, or have established a binding or application link key.

4.6.2.2 Usage

SREQ:

1	1	1	8
Length = 0x08	Cmd0 = 0x27	Cmd1 = 0x40	ExtAddr

Attributes:

Attribute	Length (byte)	Description
ExtAddr	8	Extended Address (LSB-MSB) of the device for which to lookup the Network Address

SRSP:

1	1	1	2
Length = 0x02	Cmd0 = 0x67	Cmd1 = 0x40	NwkAddr

Attributes:

Attribute	Length (byte)	Description
NwkAddr	2	Network Address (LSB-MSB) of the device that corresponds to the Extended Address sent as a parameter in the request.

4.6.3 UTIL_ADDRMGR_NWK_ADDR_LOOKUP

4.6.3.1 Description

This command is a proxy call to the AddrMgrEntryLookupNwk() function. This function takes in the Network address of the device of interest and returns its Extended address. The device of interest must either be a parent or child, or have established a binding or application link key.

4.6.3.2 Usage

SREQ:

1	1	1	2
Length = 0x02	Cmd0 = 0x27	Cmd1 = 0x41	NwkAddr

Attributes:

Attribute	Length (byte)	Description
NwkAddr	2	Network Address (LSB-MSB) of the device for which to lookup the Extended Address

SRSP:

1	1	1	8
Length = 0x08	Cmd0 = 0x67	Cmd1 = 0x41	ExtAddr

Attributes:

Attribute	Length (byte)	Description
ExtAddr	8	Extended Address (LSB-MSB) of the device that corresponds to the Network Address sent as a parameter in the request.

4.6.4 UTIL_APSME_LINK_KEY_DATA_GET

4.6.4.1 Description

This command retrieves APS link security key, TX, and RX frame counters.

4.6.4.2 Usage

SREQ:

1	1	1	8
Length = 0x08	Cmd0 = 0x27	Cmd1 = 0x44	ExtAddr

Attributes:

Attribute	Length (byte)	Description
ExtAddr	8	The extended address for which to get the application link key data.

SRSP:

1	1	1	1	16	4	4
Length = 0x19	Cmd0 = 0x67	Cmd1 = 0x44	Status	SecKey	TxFrmCntr	RxFrmCntr

Attributes:

<table>
<tr><th>Attribute</th><th>Length (byte)</th><th>Description</th></tr>
<tr><td>Status</td><td>1</td><td>Status returned by the proxy call to APSME_LinkKeyNvIdGet(). Valid status return values are enumerated below:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0x00</td><td>Success - Link key data exists</td></tr>
<tr><td>0xC8</td><td>Unknown device – A device record with the passed-in extended address does not exist</td></tr>
</table></td></tr>
<tr><td>SecKey</td><td>16</td><td>On Success, the link key data looked up; otherwise N/A</td></tr>
<tr><td>TxFrmCntr</td><td>4</td><td>On Success, the APS Tx frame counter; otherwise N/A</td></tr>
<tr><td>RxFrmCntr</td><td>4</td><td>On Success, the APS Rx frame counter; otherwise N/A</td></tr>
</table>

4.6.5 UTIL_APSME_LINK_KEY_NV_ID_GET

4.6.5.1 Description

This command is a proxy call to the APSME_LinkKeyNvIdGet() function. It returns the NV ID code corresponding to a device with the specified extended address.

4.6.5.2 Usage

SREQ:

1	1	1	8
Length = 0x08	Cmd0 = 0x27	Cmd1 = 0x45	ExtAddr

Attributes:

Attribute	Length (byte)	Description
ExtAddr	8	The extended address for which to get the link key NV ID.

SRSP:

1	1	1	1	2
Length = 0x03	Cmd0 = 0x67	Cmd1 = 0x45	Status	LinkKeyNvId

Attributes:

<table>
<tr><th>Attribute</th><th>Length (byte)</th><th>Description</th></tr>
<tr><td>Status</td><td>1</td><td>Status of the proxy call to APSME_LinkKeyNvIdGet(). Valid status return values are enumerated below:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0x00</td><td>Success - Link key data exists</td></tr>
<tr><td>0xC8</td><td>Unknown device – A device record with the passed-in extended address does not exist</td></tr>
</table></td></tr>
<tr><td>LinkKeyNvId</td><td>2</td><td>On Success, the NV ID code for the link key, otherwise 0xFFFF</td></tr>
</table>

4.6.6 UTIL_APSME_REQUEST_KEY_CMD

4.6.6.1 Description

This command is used to send a request key to the Trust Center from an originator device who wants to exchange messages with a partner device.

4.6.6.2 Usage

SREQ:

1	1	1	2
Length = 0x02	Cmd0 = 0x27	Cmd1 = 0x4B	PartnerAddr

Attributes:

Attribute	Length (byte)	Description
PartnerAddr	8	Specifies the extended address of the partner device the originator wants to exchange messages with.

SRSP:

1	1	1	1
Length = 0x01	Cmd0 = 0x67	Cmd1 = 0x4B	Status

Attributes:

Attribute	Length (byte)	Description	
		Value	Description
Status	1	0x00	Success
		0x01	Failure

4.6.7 UTIL_ASSOC_COUNT

4.6.7.1 Description

This command is a proxy call to the AssocCount() function. It returns the number of entries in the associated device list within the parameters values of StartRelation and EndRelation.

4.6.7.2 Usage

SREQ:

1	1	1	1	1
Length = 0x02	Cmd0 = 0x27	Cmd1 = 0x48	StartRelation	EndRelation

Attributes:

<table>
<tr><th>Attribute</th><th>Length (byte)</th><th>Description</th></tr>
<tr><td>StartRelation</td><td>1</td><td>The node relation at which to start counting. Valid node relations are show below
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Parent</td></tr>
<tr><td>1</td><td>Child RFD</td></tr>
<tr><td>2</td><td>Child RFD that has the RxOnWhenIdle MAC capability flag set</td></tr>
<tr><td>3</td><td>Child FFD</td></tr>
<tr><td>4</td><td>Child FFD that has the RxOnWhenIdle MAC capability flag set</td></tr>
<tr><td>5</td><td>Neighbor</td></tr>
<tr><td>6</td><td>Other</td></tr>
</table></td></tr>
<tr><td>EndRelation</td><td>1</td><td>Same as StartRelation, but the node relation at which to stop counting</td></tr>
</table>

SRSP:

1	1	1	2
Length = 0x02	Cmd0 = 0x67	Cmd1 = 0x48	Count

Attributes:

Attribute	Length (byte)	Description
Count	2	Number of entries in the associated device list within the parameters values of StartRelation and EndRelation

4.6.8 UTIL_ASSOC_FIND_DEVICE

4.6.8.1 Description

This command is a proxy call to the AssocFindDevice() function.

4.6.8.2 Usage

SREQ:

1	1	1	1
Length = 0x01	Cmd0 = 0x27	Cmd1 = 0x49	Number

Attributes:

Attribute	Length (byte)	Description
Number	1	Nth active entry in the associated device list

SRSP:

1	1	1	18
Length = 0x12	Cmd0 = 0x67	Cmd1 = 0x49	Device

Attributes:

<table>
<tr><th>Attribute</th><th>Length (byte)</th><th>Description</th></tr>
<tr><td>Device</td><td>18</td><td>The packed (LSB-MSB) associated_devices_t structure returned by the proxy call to AssocFindDevice().The device short address is set to INVALID_NODE_ADDR to indicate failure.
<table>
<tr><th>Name</th><th>Length (byte)</th><th>Description</th></tr>
<tr><td>shortAddr</td><td>2</td><td>Short address of the associated device</td></tr>
<tr><td>addrIdx</td><td>2</td><td>Index from the address manager</td></tr>
<tr><td>nodeRelation</td><td>1</td><td>Relation to this node, see enumeration of values in description for UTIL_ASSOC_COUNT</td></tr>
<tr><td>devStatus</td><td>1</td><td>Device state enumerated in descripton for ZDO_STATE_CHANGE_IND</td></tr>
<tr><td>assocCnt</td><td>1</td><td>Number of times this associated device list entry was added or updated</td></tr>
<tr><td>age</td><td>1</td><td>Number of nwk link status periods since the last link status</td></tr>
<tr><td>txCounter</td><td>1</td><td>Counter of transmission successes/failures</td></tr>
<tr><td>txCost</td><td>1</td><td>Average of sending rssi values if link status is enabled</td></tr>
<tr><td>rxLqi</td><td>1</td><td>Average of received rssi values</td></tr>
<tr><td>inKeySeqNum</td><td>1</td><td>NWK key sequence number</td></tr>
<tr><td>inFrmCntr</td><td>4</td><td>NWK key incoming frame counter</td></tr>
<tr><td>txFailure</td><td>2</td><td>Higher values indicate more failures</td></tr>
</table>
</td></tr>
</table>

4.6.9 UTIL_ZCL_KEY_EST_INIT_EST

4.6.9.1 Description

This command is a proxy call to the zclGeneral_KeyEstablish_InitiateKeyEstablishment(). This command is typically used by ZigBee Smart Energy applications to initiate the Certificate Based Key Establishment (CBKE) procedure in order to establish an application key with a partner device, typically a ZigBee Smart Energy ESI (Energy Service Interface).

4.6.9.2 Usage

SREQ:

1	1	1	1	1	1	1	8
Length = 0x0C	Cmd0 = 0x27	Cmd1 = 0x80	TaskId	SeqNum	EndPoint	AddrMode	Addr

Attributes:

Attribute	Length (byte)	Description
TaskId	1	The OSAL TaskId making the request – write a don't care value for non OSAL based host processor applications
SeqNum	1	The sequence number of the request
EndPoint	1	The destination endpoint on the partner device that implements the key establishment cluster. This is typically discovered by using ZDO_MATCH_DESC_REQ
AddrMode	1	The address mode to the partner Value: 0x02 – Description: Use the 2-byte network address Value: 0x03 – Description: Use the 8-byte extended address
Addr	2 or 8	If AddrMode is afAddr64Bit, the 8-byte extended address of the partner, otherwise the 2-byte network address of the partner

SRSP:

1	1	1	8
Length = 0x01	Cmd0 = 0x67	Cmd1 = 0x80	Status

Attributes:

Attribute	Length (byte)	Description
Status	1	Value: 0x00 – Description: Success – a call to zclGeneral_KeyEstablish_InitiateKeyEstablishment was made Value: 0x01 – Description: Failure – Exceeded maximum number of key establishment entries (default setting is 2)

4.6.10 UTIL_ZCL_KEY_EST_SIGN

4.6.10.1 Description

This command is a proxy call to zclGeneral_KeyEstablishment_ECDSASign(). This function is used to create an ECDSA signature of a message digest.

4.6.10.2 Usage

SREQ:

1	1	1	1	1
Length = 0x0C	Cmd0 = 0x27	Cmd1 = 0x81	InputLen	Input

Attributes:

Attribute	Length (byte)	Description
InputLen	1	The length of the input data
Input	InputLen	The input data

SRSP:

1	1	1	1	42
Length = 0x2B	Cmd0 = 0x67	Cmd1 = 0x81	Status	Key

Attributes:

<table>
<tr><th>Attribute</th><th>Length (byte)</th><th>Description</th></tr>
<tr><td>Status</td><td>1</td><td><table><tr><th>Value</th><th>Description</th></tr><tr><td>0x00</td><td>Success</td></tr><tr><td>0x01</td><td>Failure</td></tr></table></td></tr>
<tr><td>Key</td><td>42</td><td>The output key on success</td></tr>
</table>

4.6.11 UTIL_ZCL_KEY_ESTABLISH_IND

4.6.11.1 Description

This is the indication of the successful completion or premature termination of the key establishment procedure.

4.6.11.2 Usage

AREQ:

1	1	1	1	1	1	1	2
Length = 0x06	Cmd0 = 0x47	Cmd1 = 0xE1	TaskId	Event	Status	WaitTime	Suite

Attributes:

Attribute	Length (byte)	Description
TaskId	1	The OSAL TaskId registered to receive this indication (see UTIL_ZCL_KEY_EST_INT_EST). For non OSAL based host processor applications this is a don't care
Event	1	The OSAL message event. For non OSAL based host processor applications this is a don't care
Status	1	Status according to definitions in the ZigBee Smart Energy specification: Value / Description 0x00 – Success 0x01 – Unknown Issuer 0x02 – Bad Key Confirm 0x03 – Bad Message 0x04 – No Resources 0x05 – Unsupported Suite
WaitTime	1	The wait time – consult the ZigBee Smart Energy specification for details
Suite	2	The key establishment suite – consult the ZigBee Smart Energy specficiation for details

4.6.12 UTIL_TEST_LOOPBACK

4.6.12.1 Description

This command is used by the application processor to test the physical interface to the CC2530-ZNP.

4.6.12.2 Usage

SREQ:

1	1	1	*variable*
Length = variable	*Cmd0 = 0x27*	*Cmd1 = 0x10*	*Test data*

Test data – variable length bytes – this data will be returned by CC2530-ZNP in the response.

SRSP:

1	1	1	*variable*
Length = variable	*Cmd0 = 0x67*	*Cmd1 = 0x10*	*Test data*

Test data – variable length bytes – the Test data from the request is returned.

4.7 Return Values

The status parameter that is returned from the CC2530-ZNP device may take one of the following values:

Name	*Value*
ZSuccess	*0x00*
ZFailure	*0x01*
ZInvalidParameter	*0x02*
NV_ITEM_UNINIT	*0x09*
NV_OPER_FAILED	*0x0a*
NV_BAD_ITEM_LEN	*0x0c*
ZMemError	*0x10*
ZBufferFull	*0x11*
ZUnsupportedMode	*0x12*
ZMacMemError	*0x13*
zdoInvalidRequestType	*0x80*
zdoInvalidEndpoint	*0x82*
zdoUnsupported	*0x84*
zdoTimeout	*0x85*
zdoNoMatch	*0x86*
zdoTableFull	*0x87*
zdoNoBindEntry	*0x88*
ZSecNoKey	*0xa1*
ZSecMaxFrmCount	*0xa3*
ZApsFail	*0xb1*
ZApsTableFull	*0xb2*
ZApsIllegalRequest	*0xb3*
ZApsInvalidBinding	*0xb4*
ZApsUnsupportedAttrib	*0xb5*
ZApsNotSupported	*0xb6*
ZApsNoAck	*0xb7*
ZApsDuplicateEntry	*0xb8*
ZApsNoBoundDevice	*0xb9*
ZNwkInvalidParam	*0xc1*
ZNwkInvalidRequest	*0xc2*
ZNwkNotPermitted	*0xc3*
ZNwkStartupFailure	*0xc4*
ZNwkTableFull	*0xc7*
ZNwkUnknownDevice	*0xc8*
ZNwkUnsupportedAttribute	*0xc9*
ZNwkNoNetworks	*0xca*
ZNwkLeaveUnconfirmed	*0xcb*
ZNwkNoAck	*0xcc*
ZNwkNoRoute	*0xcd*
ZMacNoACK	*0xe9*

5 General Information

5.1 Document History

Table 1: Document History

Revision	Date	Description/Changes
1.0	2010-01-17	Initial version
1.1	2010-07-26	Updated for ZStack 2.3.1 release
1.2	2010-12-01	Removed SYS_RF_TEST command, added section 4.2.4.11
1.3	2011-07-16	Added commands: AF_APSF_CONFIG_SET SYS_GET_TIME SYS_SET_TIME SYS_OSAL_NV_ITEM_INIT SYS_OSAL_NV_ITEM_DELETE SYS_OSAL_NV_ITEM_LENGTH UTIL_APSME_REQUEST_KEY_CMD ZDO_JOIN_REQ ZDO_NWK_DISCOVERY_REQ
1.4	2012-03-10	Added callback: ZDO_LEAVE_IND Added commands: SYS_SET_TX_POWER UTIL_ADDRMGR_EXT_ADDR_LOOKUP

6 Address Information

Texas Instruments Norway AS
Gaustadalléen 21
N-0349 Oslo
NORWAY
Tel: +47 22 95 85 44
Fax: +47 22 95 85 46
Web site: http://www.ti.com/lpw

7 TI Worldwide Technical Support

Internet

TI Semiconductor Product Information Center Home Page: support.ti.com
TI Semiconductor KnowledgeBase Home Page: support.ti.com/sc/knowledgebase

Product Information Centers

Americas

Phone: +1(972) 644-5580
Fax: +1(972) 927-6377
Internet/Email: support.ti.com/sc/pic/americas.htm

Europe, Middle East and Africa

Phone:
Belgium (English) +32 (0) 27 45 54 32
Finland (English) +358 (0) 9 25173948
France +33 (0) 1 30 70 11 64
Germany +49 (0) 8161 80 33 11

Israel (English)	180 949 0107
Italy	800 79 11 37
Netherlands (English)	+31 (0) 546 87 95 45
Russia	+7 (0) 95 363 4824
Spain	+34 902 35 40 28
Sweden (English)	+46 (0) 8587 555 22
United Kingdom	+44 (0) 1604 66 33 99
Fax:	+49 (0) 8161 80 2045
Internet:	support.ti.com/sc/pic/euro.htm

Japan

Fax	International	+81-3-3344-5317
	Domestic	0120-81-0036
Internet/Email	International	support.ti.com/sc/pic/japan.htm
	Domestic	www.tij.co.jp/pic

Asia

Phone	International	+886-2-23786800
	Domestic	Toll-Free Number
	Australia	1-800-999-084
	China	800-820-8682
	Hong Kon	800-96-5941
	India	+91-80-51381665 (Toll)
	Indonesia	001-803-8861-1006
	Korea	080-551-2804
	Malaysia	1-800-80-3973
	New Zealand	0800-446-934
	Philippines	1-800-765-7404
	Singapore	800-886-1028
	Taiwan	0800-006800
	Thailand	001-800-886-0010
Fax		+886-2-2378-6808
Email		tiasia@ti.com or ti-china@ti.com
Internet		support.ti.com/sc/pic/asia.htm

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		Applications	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265