



## **Z-Stack/Z-Tool Serial Port Interface**

Document Number: F8W-2003-0001

**Texas Instruments, Inc.**  
San Diego, California USA  
(919) 542-1200



## TABLE OF CONTENTS

<b>1. INTRODUCTION .....</b>	<b>1</b>
1.1 PURPOSE.....	1
1.2 SCOPE.....	1
1.3 ACRONYMS.....	1
1.4 APPLICABLE DOCUMENTS .....	1
1.5 OVERVIEW.....	1
1.6 INTERFACE MECHANISM.....	2
<b>2. SYSTEM COMMANDS .....</b>	<b>5</b>
2.1 INTRODUCTION .....	5
2.2 SYS_PING.....	5
2.3 SYS_VERSION.....	6
2.4 SYS_SET_EXTENDED_ADDRESS.....	6
2.5 SYS_GET_EXTENDED ADDRESS .....	7
2.6 SYS_RAM_READ .....	7
2.7 SYS_RAM_WRITE .....	8
2.8 SYS_SET_NV .....	8
2.9 SYS_GET_NV .....	9
2.10 SYS_GET_DEVICE_INFO .....	9
2.11 SYS_GET_NV_INFO .....	10
2.12 SYS_SET_PANID.....	11
2.13 SYS_SET_CHANNELS.....	11
2.14 SYS_SET_SECURITY_LEVEL.....	13
2.15 SYS_SET_PRECONFIG_KEY.....	13
2.16 DEBUG MESSAGES.....	14
2.17 SYS_RESET.....	15
2.18 CALLBACK MESSAGES.....	15
2.19 SYS_KEY_EVENT.....	16
2.20 SYS_APP_MSG .....	17
2.21 SYS_LED_CONTROL.....	17
2.22 SYS_GET_TIME_ALIVE.....	18
<b>3. MAC INTERFACE .....</b>	<b>19</b>
3.1 INTRODUCTION .....	19
3.2 ZMAC_INIT_REQUEST .....	19
3.3 ZMAC_DATA_REQUEST .....	19
3.4 ZMAC_ASSOCIATE_REQUEST .....	21
3.5 ZMAC_ASSOCIATE_RESPOND .....	23
3.6 ZMAC_DISASSOCIATE_REQUEST.....	24
3.7 ZMAC_GET_REQUEST .....	26
3.8 ZMAC_ORPHAN_RESPOND .....	27
3.9 ZMAC_POLL_REQUEST .....	28
3.10 ZMAC_RESET_REQUEST.....	30
3.11 ZMAC_RX_ENABLE_REQUEST .....	30
3.12 ZMAC_SCAN_REQUEST .....	31
3.13 ZMAC_SET_REQUEST.....	32
3.14 ZMAC_START_REQUEST .....	34
3.15 ZMAC_SYNCHRONIZE_REQUEST .....	36
3.16 ZMAC_PURGE_REQUEST .....	37

<b>4. MAC CALLBACK INTERFACE.....</b>	<b>38</b>
4.1 INTRODUCTION .....	38
4.2 NWK_ASSOCIATE_INDICATION.....	38
4.3 NWK_ASSOCIATE_CONFIRMATION.....	39
4.4 NWK_DISASSOCIATE_INDICATION .....	40
4.5 NWK_DISASSOCIATE_CONFIRMATION .....	41
4.6 NWK_BEACON_NOTIFY_INDICATION .....	41
4.7 NWK_ORPHAN_INDICATION .....	43
4.8 NWK_SCAN_CONFIRMATION.....	44
4.9 NWK_START_CONFIRMATION.....	45
4.10 NWK_SYNCHRONIZATION_LOSS_INDICATION .....	45
4.11 NWK_POLL_CONFIRMATION .....	46
4.12 NWK_COMM_STATUS_INDICATION.....	46
4.13 NWK_DATA_CONFIRMATION .....	48
4.14 NWK_DATA_INDICATION .....	48
4.15 NWK_RX_ENABLE_CONFIRMATION .....	50
4.16 NWK_PURGE_CONFIRMATION .....	51
<b>5. NWK INTERFACE.....</b>	<b>52</b>
5.1 INTRODUCTION .....	52
5.2 NLDE_DATA_REQUEST .....	52
5.3 NLME_NETWORK_FORMATION_REQUEST.....	53
5.4 NLME_PERMITJOINING_REQUEST .....	54
5.5 NLME_JOIN_REQUEST .....	54
5.6 NLME_LEAVE_REQUEST .....	55
5.7 NLME_RESET_REQUEST .....	55
5.8 NLME_SYNC_REQUEST.....	57
5.9 NLME_GET_REQUEST .....	57
5.10 NLME_SET_REQUEST.....	58
5.11 NLME_NETWORKDISCOVERY_REQUEST.....	58
5.12 NLME_ROUTEDISCOVERY_REQUEST .....	59
5.13 NLME_DIRECTJOIN_REQUEST .....	60
5.14 NLME_ORPHANJOIN_REQUEST .....	60
5.15 NLME_STARTROUTER_REQUEST.....	61
<b>6. NWK CALLBACK INTERFACE .....</b>	<b>62</b>
6.1 INTRODUCTION .....	62
6.2 NLDE_DATA_CONFIRMATION .....	62
6.3 NLDE_DATA_INDICATION .....	62
6.4 NLME_NETWORKFORMATION_CONFIRMATION .....	63
6.5 NLME_JOIN_CONFIRMATION.....	63
6.6 NLME_JOIN_INDICATION.....	63
6.7 NLME_LEAVE_CONFIRMATION.....	64
6.8 NLME_LEAVE_INDICATION.....	64
6.9 NLME_SYNC_INDICATION .....	64
6.10 NLME_NETWORKDISCOVERY_CONFIRMATION .....	65
6.11 NLME_STARTROUTER_CONFIRMATION .....	66
<b>7. AF INTERFACE .....</b>	<b>67</b>
7.1 INTRODUCTION .....	67
7.2 AF_INIT_REQUEST .....	67
7.3 AF_REGISTER_REQUEST .....	67
7.4 AF_SENDMSG_REQUEST .....	68
<b>8. AF CALLBACK INTERFACE.....</b>	<b>70</b>

8.1	AF_INCOMINGDATA_INDICATION.....	70
<b>9.</b>	<b>ZDO INTERFACE .....</b>	<b>71</b>
9.1	INTRODUCTION .....	71
9.2	ZDO_AUTOAPPENDENDDEVICEBIND_REQUEST .....	71
9.3	ZDO_AUTOFINDDESTINATION_REQUEST .....	71
9.4	ZDO_NETWORKADDRESS_REQUEST .....	72
9.5	ZDO_IEEEADDRESS_REQUEST.....	72
9.6	ZDO_NODEDESCRIPTOR_REQUEST .....	73
9.7	ZDO_POWERDESCRIPTOR_REQUEST .....	74
9.8	ZDO_SIMPLEDESCRIPTOR_REQUEST .....	74
9.9	ZDO_ACTIVEENDPOINT_REQUEST.....	75
9.10	ZDO_MATCHDESCRIPTOR_REQUEST.....	75
9.11	ZDO_COMPLEXDESCRIPTOR_REQUEST .....	76
9.12	ZDO_USERDESCRIPTOR_REQUEST .....	77
9.13	ZDO_ENDDEVICEBIND_REQUEST .....	77
9.14	ZDO_BIND_REQUEST .....	78
9.15	ZDO_UNBIND_REQUEST.....	79
9.16	ZDO_MGMT_NWKDISC_REQUEST .....	79
9.17	ZDO_MGMT_LQI_REQUEST .....	80
9.18	ZDO_MGMT_RTG_REQUEST.....	81
9.19	ZDO_MGMT_BIND_REQUEST.....	81
9.20	ZDO_MGMT_DIRECT_JOIN_REQUEST.....	82
9.21	ZDO_USERDESCRIPTOR_SET_REQUEST.....	82
9.22	ZDO_END_DEVICE_ANNC.....	83
9.23	ZDO_MGMT_LEAVE_REQUEST .....	83
9.24	ZDO_MGMT_PERMIT_JOIN_REQUEST.....	84
9.25	ZDO_SYSTEM_SERVER_DISC_REQUEST .....	84
9.26	ZDO_NETWORK_START_REQUEST.....	85
<b>10.</b>	<b>ZDO CALLBACK INTERFACE.....</b>	<b>86</b>
10.1	INTRODUCTION .....	86
10.2	ZDO_NETWORKADDRESS_RESPONSECB .....	86
10.3	ZDO_IEEEADDRESS_RESPONSECB .....	86
10.4	ZDO_NODEDESCRIPTOR_RESPONSECB.....	87
10.5	ZDO_POWERDESCRIPTOR_RESPONSECB.....	88
10.6	ZDO_SIMPLEDESCRIPTOR_RESPONSECB.....	90
10.7	ZDO_ACTIVEENDPOINT_RESPONSECB .....	91
10.8	ZDO_MATCHDESCRIPTOR_RESPONSECB .....	91
10.9	ZDO_USERDESCRIPTOR_RESPONSECB.....	92
10.10	ZDO_ENDDEVICEBIND_RESPONSECB.....	92
10.11	ZDO_BIND_RESPONSECB .....	93
10.12	ZDO_UNBIND_RESPONSECB .....	93
10.13	ZDO_MGMT_NWKDISC_RESPONSECB .....	94
10.14	ZDO_MGMT_LQI_RESPONSECB .....	94
10.15	ZDO_MGMT_RTG_RESPONSECB.....	95
10.16	ZDO_MGMT_BIND_RESPONSECB.....	97
10.17	ZDO_MGMT_DIRECT_JOIN_RESPONSECB.....	97
10.18	ZDO_USERDESCRIPTOR_SET_RESPONSECB .....	98
10.19	ZDO_MGMT_LEAVE_RESPONSECB.....	98
10.20	ZDO_MGMT_PERMIT_JOIN_RESPONSECB .....	99
10.21	ZDO_SYSTEM_SERVER_DISC_RESPONSECB .....	99
<b>11.</b>	<b>APPLICATION INTERFACE .....</b>	<b>100</b>
11.1	INTRODUCTION .....	100

11.2	USERTEST_REQUEST.....	100
------	-----------------------	-----

**LIST OF FIGURES**

FIGURE 1	Z-STACK MESSAGE FORMAT.....	2
----------	-----------------------------	---

## 1. Introduction

### 1.1 Purpose

This document defines the Serial Port Interface (SPI) that is used for communication between the Z-T test program and the target. The Z-T test program issues commands to the target and receives appropriate responses. The communication occurs over a RS-232 serial port.

### 1.2 Scope

This document defines

- The link layer protocols used to exchange packets over the RS-232 serial link
- A set of commands that can be issued by the Z-T test program to the target
- Appropriate responses for each of these commands
- The format of each of the command and the response messages

### 1.3 Acronyms

<b>BB</b>	Baseband
<b>CCA</b>	Clear Channel Assessment
<b>MAC</b>	Media Access Control
<b>NHLE</b>	Next Higher Layer Entity
<b>NWK</b>	Network
<b>PHY</b>	Physical Layer
<b>PIB</b>	Personal Information Base
<b>SPI</b>	Serial Port Interface

### 1.4 Applicable Documents

1. Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), IEEE Standard 802.15.4, May 12, 2003.

### 1.5 Overview

The serial port interface enables the Z-T test program to issue commands to and receive responses from the target. The following functionalities are enabled by these messages.

1. System commands: These messages allow the tester to control device configuration, access memory location, set the debug message filter, control message tracing in each software component, and reset the target.
2. MAC interface commands: The test can issue MAC commands to access the MAC layer functions of the target.

3. NWK interface commands: The tester can issue NWK commands to control NWK layer functionalities of the target.
4. AF interface commands: The tester can issue AF commands to call AF layer functions of the target.
5. ZDO interface commands: The test can issue ZDO commands to access the ZDO functionalities of the target.
6. Application-specific commands: The tester can issue application-specific commands paired with programmed message interfaces from the Test Tool and capabilities within Minitest on the target.

In Z-Stack, each of the above functionalities can be enabled or disabled using compiler flags. One benefit of compiling out unnecessary functionalities is saving the code space which plays a significant role in embedded system development. The details of how to use these compiler flags in preprocessor can be found in the Z-Stack Compiler Options document.

## 1.6 Interface Mechanism

A link layer protocol is necessary so that messages can be exchanged between the tester and target over an RS-232 serial link. The purpose of the link layer protocol is to frame the messages in packets for proper transmission and reception and to ensure message integrity.

The physical transmission uses: no Parity; 8 data bits and 1 stop bits for each byte.

The transmission rate will be 38.4 kbps.

The Z-Tool program must send the message at a time and wait for either the expected response message to a timeout before sending the next message or resending the current message.

Fields that are multi-byte fields are transmitted Most Significant byte first.

There is no provision for retransmission of lost packets

The link layer packet has the following format.

SOP	CMD	LEN	Data	FCS
-----	-----	-----	------	-----

**Figure 1 Z-Stack Message Format**

**SOP (Start of Packet):** This is a one byte field with value equal to 0x02 that defines the start of each link level packet.

**CMD (Command ID):** This is a two byte field (MSByte transmitted first) with a value denoting the Command Identification (ID) for this message. This field is described in detail below.

**LEN (Length):** This one byte field is the number of bytes in the Data field. If the Data field contains no information this LEN field has a value of 0 and the total length of the command is 5 bytes (0 data message).

**Data:** This field contains the actual data to be transmitted. This is a field which varies in size according to the command.

**FCS (Frame check sequence):** This is a one byte field that is used to ensure packet integrity. This field is computed as an XOR of all the bytes in the message starting with CMD and through the last byte of data. The receiver XORs all the received data bytes as indicated above and then XORs the received FCS field. If the sum is not equal to zero, the received packet is in error.

The CMD, LEN and data fields are application message dependant but shown here to fully illustrate the link level protocol. The CMD, LEN and data fields are separately described in the succeeding sections of this document.

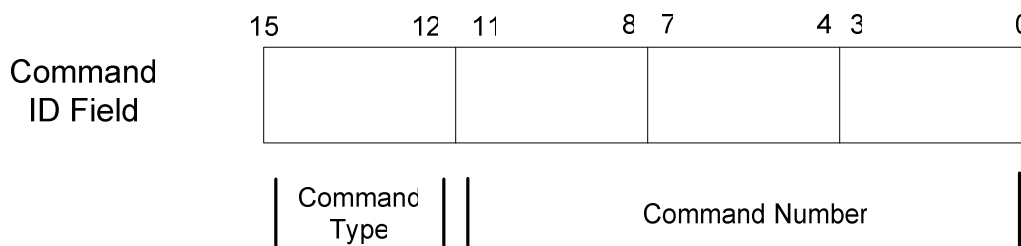
In terms of packet error handling, the procedure is as follows:



- Assuming the start of message indicator is unknown and there is a buffer containing received data bytes to be processed, the first element of the procedure is SOP detection:
  - Parse the buffer until a SOP is detected and return the buffer location of that indicator.
- Next, the SOP must be validated (since other data elements in the buffer may match the SOP). The SOP validation procedure is as follows:
  - The field after the SOP is the CMD and the following is the LEN. Use the LEN field to parse the prospective message to the location of the FCS. Use the FCS to check the potential message and validate the SOP location. If validated, message parsing continues with the next SOP (immediately after the FCS just processed).
  - Otherwise, the SOP is not validated and the procedure continues with SOP detection.
- Once the SOP is detected and validated, message processing continues with successive message sequences as denoted in Figure 1.
- Computation of invalid FCS in a message shall result in invalidation of the SOP detection procedure beginning with the field after the failed FCS since failed FCS could have resulted from an erroneous LEN field.

### 1.6.1 Command ID (CMD)

The Command ID field is a 16-bit field broken up into the following 2 sub fields:



**Command Type** is a most significant 4 bits of the CMD field and is used to route messages. Each bit (3-0) of this sub field is defined as:

- Bit 0 – Response Bit – if set, this bit indicates a response message (each response message is described below).
- Bit 1 – Subscription Bit – if set, this bit indicates that this message was the result of a subscription to a callback service.
- Bit 2 – Debug/Trace Bit – if set, this bit indicates that this message is either a Debug or Telemetry.

0x500 – 0x5FF	SPI Interface Commands
0x600 – 0x6FF	Sequence Interface Commands
0x800 – 0x8FF	APS Interface Commands
0x900 – 0x9FF	AF Interface Commands
0xA00 – 0xAFF	ZDO Interface Commands
0xB00 – 0xBFF	Device Interface Commands

Remember that the *Command Number* and *Command Type* sub fields are combined to make a unique *Command ID*. That means the messages can share *Command Numbers* as long as the *Command Types* are different. For example, a MAC Command, its response, and a MAC subscription (callback) message can all have the same *Command Number*, but the *Command Type* for each is different.

## 2. System Commands

### 2.1 Introduction

These messages are used to issue system related commands.

### 2.2 SYS\_PING

#### 2.2.1 Description

This command issues PING requests to verify if a device is active and check the capability of the device.

#### 2.2.2 Usage

The command message format is as follows

Cmd=0x0007	Len=0x00
------------	----------

#### 2.2.3 SYS\_PING\_RESPONSE

The response to this command has the following format.

Cmd=0x1007	Len=0x02	Capabilities
------------	----------	--------------

Capabilities – 16 bit field. This field represents the interfaces that this device can handle (compiled into the device). Bit weighted and defined as:

MT_CAP_MAC	0x0001
MT_CAP_NWK	0x0002
MT_CAP_AF	0x0004
MT_CAP_ZDO	0x0008
MT_CAP_USER_TEST	0x0010
MT_CAP_SEQ	0x0020
MT_CAP_BOOTLOAD	0x8000

## 2.3 SYS\_VERSION

### 2.3.1 Description

This command is used to request for the device's version string.

### 2.3.2 Usage

The command message format is as follows

Cmd=0x0008	Len=0x00
------------	----------

### 2.3.3 SYS\_VERSION\_RESPONSE

The response to this command has the following format.

Cmd=0x1008	Len	Version
------------	-----	---------

Len – length of VersionString in bytes. This is the number of characters not including the null termination.

Version – device defined ASCII string (NOT null terminated).

## 2.4 SYS\_SET\_EXTENDED\_ADDRESS

### 2.4.1 Description

This command is used to set the 64-bit extended address.

### 2.4.2 Usage

The command message format is as follows

Cmd=0x0010	Len = 8	ExtAddress
------------	---------	------------

ExtAddress – 8 bytes – The device's extended address.

### 2.4.3 SYS\_SET\_EXTENDED\_ADDRESS\_RESPONSE

The response to this command has the following format.

Cmd=0x1010	Len=0x01	Status
------------	----------	--------

The status is a success (0) or fail (1).

## 2.5 SYS\_GET\_EXTENDED ADDRESS

### 2.5.1 Description

This command is used to read the extended address from the device.

### 2.5.2 Usage

The command message format is as follows

Cmd=0x0011	Len = 0x00
------------	------------

### 2.5.3 SYS\_GET\_EXTENDED\_ADDRESS\_RESPONSE

The response to this command has the following format.

Cmd=0x1011	Len=0x08	ExtAddress
------------	----------	------------

ExtAddress – 8 bytes – The device's extended address.

## 2.6 SYS\_RAM\_READ

### 2.6.1 Description

This command is used by the tester to read a single memory location in the target RAM. The command accepts an address value and returns the memory value present in the target RAM at that address.

### 2.6.2 Usage

The command message format is as follows

Cmd=0x0001	Len=0x02	Address
------------	----------	---------

Address is a 2 byte field and contains the address of the memory location that needs to be read.

### 2.6.3 SYS\_RAM\_READ\_RESPONSE

The response to this command has the following format.

Cmd=0x1001	Len=0x03	Status	Value
------------	----------	--------	-------

Status is a one byte field and is either success(0) or fail(1). The fail status is returned if the address value in the command message was not within the valid range.

Value is a byte field that contains the data read from the target RAM.

## 2.7 SYS\_RAM\_WRITE

### 2.7.1 Description

This command is used by the tester to write to a particular location in the target RAM. The command accepts an address location and a memory value. The memory value is written to the address location in the target RAM.

### 2.7.2 Usage

The command message format is as follows

Cmd=0x0002	Len=0x04	Address	Value
------------	----------	---------	-------

Address is a 2 byte field that indicates the address of the memory location that is to be written.

Value is a byte field that contains the data value that is to be written to the memory location.

### 2.7.3 SYS\_RAM\_WRITE\_RESPONSE

The response has the following format.

Cmd=0x1002	Len=0x01	Status
------------	----------	--------

Status is a success (0) or fail (1). The fail status is returned if the address location in the command message was not within the valid range.

## 2.8 SYS\_SET\_NV

### 2.8.1 Description

This command is used by the tester to write to a particular item in non-volatile memory. The command accepts an attribute ID and an attribute value. The attribute value is written to the location specified for the attribute ID in the target.

### 2.8.2 Usage

The command message format is as follows

Cmd=0x0012	Len=variable	Attribute	Value
------------	--------------	-----------	-------

Len is a 1 byte field that indicates the total length of the Attribute and Value fields that follow.

Attribute is a 1 byte field that indicates the ID of the location that is to be written.

Value is a byte array that contains the data value that is to be written.

### 2.8.3 SYS\_SET\_NV\_RESPONSE

The response has the following format.

Cmd=0x1012	Len=0x01	Status
------------	----------	--------

Status is a success (0) or fail (1). The fail status is returned if the address location in the command message was not within the valid range.

## 2.9 SYS\_GET\_NV

### 2.9.1 Description

This command is used by the tester to read a single memory item in the target non-volatile memory. The command accepts an attribute ID value and returns the memory value present in the target for the specified attribute ID.

### 2.9.2 Usage

The command message format is as follows

Cmd=0x0013	Len=0x02	Attribute
------------	----------	-----------

Attribute is a 1byte field and contains the location that needs to be read.

### 2.9.3 SYS\_NV\_GET\_RESPONSE

The response to this command has the following format.

Cmd=0x1013	Len=variable	Status	Attribute	Value
------------	--------------	--------	-----------	-------

Len is a 1 byte field that indicates the total length of the Status, Attribute, and Value fields that follow.

Status is a one byte field that indicates either success(0) or fail(1). Failure status is returned if the attribute ID in the command message was not within the valid range.

Attribute is a 1 byte field that indicates the ID of the data value that is being returned.

Value is a byte array that contains the data returned from the target.

## 2.10 SYS\_GET\_DEVICE\_INFO

### 2.10.1 Description

This command is used by the tester to read a block of active parameters from the target device.

### 2.10.2 Usage

The command message format is as follows

Cmd=0x0014	Len=0x00
------------	----------

### 2.10.3 SYS\_GET\_DEVICE\_INFO\_RESPONSE

The response to this command has the following format.

Cmd=0x1014	Len=0x03	Status	IEEEAddress	ShrtAddress	Device Type	Device State	NumAssociated Devices	Associated Devices List
------------	----------	--------	-------------	-------------	-------------	--------------	-----------------------	-------------------------

Status is a one byte field and is either success(0) or fail(1). The fail status is returned if the address value in the command message was not within the valid range.

IEEEAddress is an 8 byte field.

ShrtAddress is a 2 byte field.

DeviceType is a 1 byte field indicating device type, where bits 0 to 2 indicate the capability of the device to operate as a coordinator, router, or end device, respectively.

DeviceState is a 1 byte field indicating the state of the device with different possible states as shown below:

Description	Value
Initialized - not started automatically	0x00
Initialized - not connected to anything	0x01
Discovering PAN's to join	0x02
Joining a PAN	0x03
ReJoining a PAN, only for end devices	0x04
Joined but not yet authenticated by trust center	0x05
Started as device after authentication	0x06
Device joined, authenticated and is a router	0x07
Starting as Zigbee Coordinator	0x08
Started as Zigbee Coordinator	0x09
Device has lost information about its parent	0x0A

NumAssociatedDevices is a 1 byte field specifying the number of devices being associated to the target device.

AssociatedDevicesList is a variable length array of 16-bits specifying the network address(es) of device(s) associated to the target device.

## 2.11 SYS\_GET\_NV\_INFO

### 2.11.1 Description

This command is used by the tester to read a block of parameters from NVM-Volatile storage of the target device.



## 2.11.2 Usage

The command message format is as follows

Cmd=0x001F	Len=0x00
------------	----------

## 2.11.3 SYS\_GET\_NV\_INFO\_RESPONSE

The response to this command has the following format.

Cmd=0x101F	Len=0x1F	Status	IEEEAddress	ScanChannels	PanID	SecurityLevel	PreConfigKey
------------	----------	--------	-------------	--------------	-------	---------------	--------------

Status is a one byte field. A value of zero indicates success. Failure is indicated by a non-zero value, representing a bit mask of each item that failed to be retrieved from NV memory. Bit0 is used for the first item (IEEEAddress), bit1 for the second item (ScanChannels), and so forth. Data values for failed items are returned as the same bytes of 0xFF, the typical value read from erased NV memory.

IEEEAddress is an 8 byte field. This is the device's unique 64-bit extended address.

ScanChannels is a 4 byte field. This represents a bit-mask of channels to be scanned when starting the device.

PanID is a 1 byte field. Specifies the Pan ID to start joining. Set to 0xFFFF to select a PAN after scanning.

SecurityLevel is a 1 byte field. This specifies the network messaging security level, zero disables security.

PreConfigKey is a 16-byte field. This specifies the pre-configured security key.

## 2.12 SYS\_SET\_PANID

### 2.12.1 Description

Store a PanID value into Non-Volatile memory to be used the next time the target device resets.

### 2.12.2 Usage

The command message format is as follows

Cmd=0x001B	Len = 0x01	PanID
------------	------------	-------

PanID – 1 byte – The device's PanID.

### 2.12.3 SYS\_SET\_PANID\_RESPONSE

The response to this command has the following format.

Cmd=0x101B	Len=0x01	Status
------------	----------	--------

The status is a success (0) or fail (non-zero NV error code).

## 2.13 SYS\_SET\_CHANNELS

### 2.13.1 Description

This command is used to store a channel select bit-mask into NVolatile memory to be used the next time the target device resets.

### 2.13.2 Usage

The command message format is as follows

Cmd=0x001C	Len = 0x04	Channels
------------	------------	----------

Channels – 4 bytes – A bit-mask representing the channel(s) to scan the next time the target device resets.

### 2.13.3 SYS\_SET\_CHANNELS\_RESPONSE

The response to this command has the following format.

Cmd=0x101C	Len=0x01	Status
------------	----------	--------

The status is a success (0) or fail (non-zero NV error code).

## 2.14 SYS\_SET\_SECURITY\_LEVEL

### 2.14.1 Description

This command is used to store a security level value into NVRAM to be used the next time the target device resets.

### 2.14.2 Usage

The command message format is as follows

Cmd=0x001D	Len = 0x01	SecLevel
------------	------------	----------

SecLevel – 1 byte – Security level to use the next time the target device resets. Zero is used to disable security.

### 2.14.3 SYS\_SET\_SECURITY\_LEVEL\_RESPONSE

The response to this command has the following format.

Cmd=0x101D	Len=0x01	Status
------------	----------	--------

The status is a success (0) or fail (non-zero NV error code).

## 2.15 SYS\_SET\_PRECONFIG\_KEY

### 2.15.1 Description

This command is used to store a pre-configured key array into NVRAM to be used the next time the target device resets.

### 2.15.2 Usage

The command message format is as follows

Cmd=0x001E	Len = 0x10	PreCfgKey
------------	------------	-----------

PreCfgKey – 16 bytes – An array representing the pre-configured key to use the next time the target device resets.

### 2.15.3 SYS\_SET\_PRECONFIG\_KEY\_RESPONSE

The response to this command has the following format.

Cmd=0x101E	Len=0x01	Status
------------	----------	--------

The status is a success (0) or fail (non-zero NV error code).

## 2.16 Debug Messages

These messages are used by the tester to set debug threshold level for each software component in the target and receive debug messages from them. The tester uses the Set Debug Threshold command to set the threshold value for debug messages from each software component. The target sends asynchronous debug messages that exceed the threshold value to the tester.

### 2.16.1 SYS\_SET\_DEBUG\_THRESHOLD

#### 2.16.1.1 Description

This command is used by the tester to set the debug threshold level for a particular software component in the target.

#### 2.16.1.2 Usage

The command message format is as follows

Cmd=0x0003	Len=0x02	ComponentID	Threshold
------------	----------	-------------	-----------

ComponentID is a 1 byte field that uniquely identifies a particular software component in the target. The list of software components and their identifiers is listed in Appendix A.

Threshold is a 1 byte field that specifies the threshold value for reporting debug messages by that software component. Its value can be set to CRITICAL (0x01), ERROR(0x02), INFORMATION(0x03) or TRACE(0x04). The software component will report all debug messages whose value exceeds the threshold. The current implementation only supports two levels of differentiation, either ON( >0x00 ) or OFF(0x00)

#### 2.16.1.3 SYS\_SET\_DEBUG\_THRESHOLD\_RESPONSE

The response message has the following format

Cmd=0x1003	Len=0x01	Status
------------	----------	--------

The status field is 1 byte and indicates success(0) or fail(1). The fail status is returned if the component identifier in the command message was not valid.

### 2.16.2 SYS\_DEBUG\_MESSAGE

This message is sent by the tester to the target.

Cmd = 0x4003	Len=5+2n	ComponentID	Value	ParameterCount	Parameter1	Parameter2	Parameter3	Time
--------------	----------	-------------	-------	----------------	------------	------------	------------	------

ComponentID is a 1 byte field that identifies the software component within the target that produced the debug message. The list of components and their identifiers is listed in Appendix A.

Value is a 1 byte field that indicates the severity of the debug message. It takes values of CRITICAL(0x01), ERROR(0x02), INFORMATION(0x03) or TRACE(0x04).

ParameterCount is a 1 byte field and indicates the number of parameters included in the message

TimeStamp is a 2 byte field and indicates the time of occurrence of the debug.

The debug message can have up to 3 parameters. The number of parameters is contained in the ParameterCount field (1 byte, ranges from 0-3) and the parameters themselves are contained (if present) in the parameter1, parameter2 and parameter3 fields (2 bytes each).

The length field has a value of  $5+2n$  where n refers to the number of parameters included in the message.

## 2.17 SYS\_RESET

### 2.17.1 Description

This command is sent by the tester to reset the target device.

### 2.17.2 Usage

The command message format is as follows

Cmd=0x0005	Len=0x01	Type
------------	----------	------

The type field is 1 byte and 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.

### 2.17.3 SYS\_RESET\_RESPONSE

The response has the following format.

Cmd=0x1005	Len=0x01	Type
------------	----------	------

The type field indicates which reset operation is being reported – target device (0) or serial bootloader (1). This response is sent from the target device or serial bootloader after the requested reset is complete.

## 2.18 Callback Messages

Each of the following layer interfaces consists of 2 parts (command and callback). The commands are followed by an immediate response message, but the callbacks are asynchronous and can occur at unexpected times. With this command the tester can subscribe or unsubscribe from these callbacks.

### 2.18.1 SYS\_CALLBACK\_SUBSCRIBE

#### 2.18.1.1 Description

This command subscribes/unsubscribes to layer callbacks.

#### 2.18.1.2 Usage

The command message format is as follows.

Cmd=0x0000	Len=0x03	CommandID	Action
------------	----------	-----------	--------

SubCommandID is a 2 byte field that contains the Command ID field of the expected callback message. Refer to Appendix A for the following callback sections for valid command IDs. A value of 0xFFFF indicates all callbacks.

Action is a 1 byte field that contains the action of the command (0 – unsubscribe, 1 – subscribe).

## 2.18.2 SYS\_CALLBACK\_SUBSCRIBE\_RESPONSE

Cmd=0x1000	Len=0x01	Status
------------	----------	--------

Status is a 1 byte field that indicates SUCCESS or FAILURE (see appendix for failure codes).

## 2.19 SYS\_KEY\_EVENT

### 2.19.1 Description

Sends a key event to the device registered application. The device register application means that the application registered for key events with OnBoard. Not all applications support all keys, so you must know what keys the application supports.

### 2.19.2 Usage

The command message format is as follows.

Cmd=0x0010	Len=0x02	Shift	Key
------------	----------	-------	-----

Shift is a byte field with a Boolean value. True (0x01) means that the shift was enabled.

Key is a byte field with bit values for the keys. The following table has the key types:

Key Event	Bit
Key 1	0
Key 2	1
Key 3	2
Key 4	3
Key 5	4
Key 6	5
Key 7	6
Key 8	7

### 2.19.3 SYS\_KEY\_EVENT\_RESPONSE

Cmd=0x101	Len=0x01	Status
-----------	----------	--------

Status is a 1 byte field that indicates SUCCESS(0) or FAILURE (see appendix for failure codes). Failure means that an application was registered to receive the key event.

## 2.20 SYS\_APP\_MSG

### 2.20.1 Description

Use this message to send raw data to an application. The request message may or may not get a response message. There could also be multiple and/or asynchronous response messages.

### 2.20.2 Usage

The command message format is as follows.

Cmd=0x0018	Len=0x01 + data length	Endpoint	Data
------------	------------------------	----------	------

Endpoint is a byte field – this represents the application endpoint to send the data to. Don't use this field to send ZDO messages (endpoint 0).

Data – a variable length field – raw data packet to send to the application.

### 2.20.3 SYS\_APP\_MSG\_RESPONSE

Cmd=0x1018	Len=var	Endpoint	Data
------------	---------	----------	------

Endpoint is a byte field – this represents the application endpoint the data is from.

Data is a variable length field – raw data packet from an application endpoint.

## 2.21 SYS\_LED\_CONTROL

### 2.21.1 Description

Use this message to control LEDs on the target board.

### 2.21.2 Usage

The command message format is as follows.

Cmd=0x0019	Len=2	LED	Mode
------------	-------	-----	------

LED is a byte field with the following possible values:

Value	Description
1	LED 1
2	LED 2
3	LED 3
4	LED 4
0xFF	All LEDs

Mode is a byte field with the following possible values:

Value	Description
0	LED Off
1	LED On
2	Blink
3	Flash
4	Toggle

### 2.21.3 SYS\_LED\_CONTROL\_RESPONSE

Cmd=0x1019	Len=0x01	Status
------------	----------	--------

Status is a 1 byte field that will be SUCCESS(0) or FAILURE(1).

## 2.22 SYS\_GET\_TIME\_ALIVE

### 2.22.1 Description

This command is used to request the length of the time period since the device's last reset in unit of seconds.

### 2.22.2 Usage

The command message format is as follows.

Cmd=0x001A	Len=0x00
------------	----------

### 2.22.3 SYS\_GET\_TIME\_ALIVE\_RESPONSE

Cmd=0x101A	Len=0x04	Seconds
------------	----------	---------

Seconds is a 4 byte field that will indicate the time duration in seconds.



### 3. MAC Interface

#### 3.1 Introduction

These commands allow the tester to issue commands to the MAC layer in the target and receive responses. Each of these messages has a corresponding message that is returned by the target. The response message only indicates that the command message was received and executed. The result of the command execution will be conveyed to the tester via a callback message interface (defined in the next section), if applicable.

#### 3.2 ZMAC\_INIT\_REQUEST

##### 3.2.1 Description

This command is used to initialize the ZMAC in the current device (on behalf of the next higher layer).

##### 3.2.2 Usage

Cmd	Length
0x0082	0x00

##### 3.2.3 ZMAC\_INIT\_RESPONSE

Cmd	Length	Status
0x1082	0x01	1 byte

Status indicates SUCCESS or FAILURE

#### 3.3 ZMAC\_DATA\_REQUEST

##### 3.3.1 Description

This command is used to send (on behalf of the next higher layer) MAC Data Frame packet.

##### 3.3.2 Usage

Cmd	Length	DestAddressMode	DestAddress	DestPanID	SrcAddressMode
0x0083	0x1C – 0x82	1 byte	8 bytes	2 bytes	1 byte

Handle	TxOptions	LogicalChannel	Power	KeySource	Security Level
1 byte	1 byte	1 byte	1 byte	8 bytes	1 byte

KeyIdMode	KeyIndex	MSDU Length	MSDU
1 byte	1 byte	1 byte	0-102 bytes

**DestAddressMode:** Specifies the format of the destination address.

Description	Value
Address Not Present	0x00
Address 16 bit	0x02
Address 48 bit	0x03
Broadcast	0xFF

**DestAddress:** Address of the destination.

**DestPanID:** PAN ID of the destination.

**SrcAddressMode:** Specifies the format of the source address.

Description	Value
Address Not Present	0x00
Address 16 bit	0x02
Address 48 bit	0x03
Broadcast	0xFF

**Handle:** Handle of the packet.

**TxOption:** Transmitting options

Description	Value
Acknowledged transmission	0x01
GTS transmission (unused)	0x02
Indirect transmission	0x04
Nonretransmit transmission	0x10
Nonconfirm transmission	0x20
Use PIB value MAC_ALT_BE for the minimum backoff exponent transmission	0x40

**LogicalChannel:** Channel that data frame will be transmitted.

**Power:** Power level that data frame will be transmitted.

**KeySource:** Key Source of this data frame.

**SecurityLevel:** Security Level of this data frame.

Description	Value
Nonsecurity is used	0x00

MIC-32 authentication is used	0x01
MIC-64 authentication is used	0x02
MIC-128 authentication is used	0x03
AES encryption is used	0x04
AES encryption and MIC-32 authentication are used	0x05
AES encryption and MIC-64 authentication are used	0x06
AES encryption and MIC-128 authentication are used	0x07

**KeyIdMode:** Key Id Mode of this data frame.

Description	Value
Key is not used / Key is determined implicitly	0x00
Key is determined from the 1-byte key index	0x01
Key is determined from the 4-byte key index	0x02
Key is determined from the 8-byte key index	0x03

**KeyIndex:** Key Index of this data frame.

**MSDULength:** Length of the data

**MSDU:** Actual data.

### 3.3.3 ZMAC\_DATA\_RESPONSE

Cmd	Length	Status
0x1083	1 byte	1 byte

**Status** indicates SUCCESS or FAILURE

## 3.4 ZMAC\_ASSOCIATE\_REQUEST

### 3.4.1 Description

This command is used to request (on behalf of the next higher layer) an association with a coordinator.

### 3.4.2 Usage

Cmd	Length	LogicalChannel	ChannelPage	CoordinatorAddressMode	CoordinatorAddress	CoordinatorPanID
0x0084	0x19	1 byte	1 byte	1 byte	8 bytes	2 bytes

Capability Information	KeySource	SecurityLevel	KeyIdMode	KeyIndex
1 byte	8 bytes	1 bytes	1 bytes	1 byte

**LogicalChannel:** The channel that the device will associate.

**ChannelPage:** The channel page that the device will associate.

**CoordAddressMode:** Specifies the format of the coordinat device address.

Descripti n	Value
Address N t Present	0x00
Address 1 t bit	0x02
Address 4 t bit	0x03
Br adcast	0xFF

**CoordAddress:** Address of the Coordinat

**CoordPanID:** PAN ID of the coordinat.

**CapabilityInformation:** Bit map which specifies the operational capabilities of the device.

**KeySource:** Key Source of this data frame.

**SecurityLevel:** Security Level of this data frame.

Descripti n	Value
N t security is used	0x00
MIC-32 authenticati n is used	0x01
MIC-4 authenticati n is used	0x02
MIC-128 authenticati n is used	0x03
AES encrypti n is used	0x04
AES encrypti n and MIC-32 authenticati n are used	0x05
AES encrypti n and MIC-4 authenticati n are used	0x06
AES encrypti n and MIC-128 authenticati n are used	0x07

**KeyIdMode:** Key Id Mode of this data frame.

Descripti n	Value
Key is n t used / Key is determined implicitly	0x00
Key is determined fr m the 1-byte key index	0x01
Key is determined fr m the 4-byte key index	0x02
Key is determined fr m the 8-byte key index	0x03

**KeyIndex:** Key Index of this data frame.

### 3.4.3 ZMAC\_ASSOCIATE\_RESPONSE

Cmd	Length	Status
0x1084	0x01	1 byte

**Status** indicates SUCCESS or FAILURE

## 3.5 ZMAC\_ASSOCIATE\_RESPOND

### 3.5.1 Description

This command is used to send (on behalf of the next higher layer) an association response message.

### 3.5.2 Usage

Cmd	Length	DeviceExtendedAddress	AssocShortAddress	Status
0x0085	0x14	8 bytes	2 bytes	1 byte

KeySource	SecurityLevel	KeyIdMode	KeyIndex
8 bytes	1 bytes	1 bytes	1 byte

**DeviceExtendedAddress:** Extended Address of the device requesting association.

**AssocShortAddress:** The short address allocated to the device.

**Status:** Indicates success or failure

Description	Value
SUCCESS	0x00
FAILURE	0x01

**KeySource:** Key Source of this data frame.

**SecurityLevel:** Security Level of this data frame.

Description	Value
No security is used	0x00
MIC-32 authentication is used	0x01
MIC-64 authentication is used	0x02
MIC-128 authentication is used	0x03
AES encryption is used	0x04
AES encryption and MIC-32 authentication are used	0x05
AES encryption and MIC-64 authentication are used	0x06

AES encryption and MIC-128 authentication are used	0x07
--	------

**KeyIdMode:** Key Id Mode of this data frame.

Description	Value
Key is not used / Key is determined implicitly	0x00
Key is determined from the 1-byte key index	0x01
Key is determined from the 4-byte key index	0x02
Key is determined from the 8-byte key index	0x03

**KeyIndex:** Key Index of this data frame.

### 3.5.3 ZMAC\_ASSOCIATE\_RESPOND\_RESPONSE

Cmd	Length	Status
0x1085	0x01	1 byte

**Status** indicates SUCCESS or FAILURE

## 3.6 ZMAC\_DISASSOCIATE\_REQUEST

### 3.6.1 Description

This command is used to request (on behalf of the next higher layer) a disassociation of the device from the coordinator.

### 3.6.2 Usage

Cmd	Length	DeviceAddressMode	DeviceAddress	DevicePanID	DisassociateReason	TxIndirect
0x0084	0x18	1 byte	8 bytes	2 bytes	1 byte	1 byte

KeySource	SecurityLevel	KeyIdMode	KeyIndex
8 bytes	1 bytes	1 bytes	1 byte

**DeviceAddressMode:** Specifies the format of the device address.

Description	Value
Address Not Present	0x00

Address 16 bit	0x02
Address 48 bit	0x03
Broadcast	0xFF

**DeviceAddress:** The address of the device (specified by the module above)

**DevicePanID:** Pan ID of the device.

**DisassociateReason:** Reason for disassociation.

**TxIndirect:** Transmit indirect.

**KeySource:** Key Source of this data frame.

**SecurityLevel:** Security Level of this data frame.

Description	Value
No security is used	0x00
MIC-32 authentication is used	0x01
MIC-48 authentication is used	0x02
MIC-128 authentication is used	0x03
AES encryption is used	0x04
AES encryption and MIC-32 authentication are used	0x05
AES encryption and MIC-48 authentication are used	0x06
AES encryption and MIC-128 authentication are used	0x07

**KeyIdMode:** Key Id Mode of this data frame.

Description	Value
Key is not used / Key is determined implicitly	0x00
Key is determined from the 1-byte key index	0x01
Key is determined from the 4-byte key index	0x02
Key is determined from the 8-byte key index	0x03

**KeyIndex:** Key Index of this data frame.

### 3.6.3 ZMAC\_DISASSOCIATE\_RESPONSE

Cmd	Length	Status
0x108	0x01	1 byte

**Status** indicates SUCCESS or FAILURE

### 3.7 ZMAC\_GET\_REQUEST

#### 3.7.1 Description

This command is used to read (on behalf of the next higher layer) a MAC PIB attribute.

#### 3.7.2 Usage

Cmd	Length	Attribute
0x0087	0x01	1 byte

**Attribute:** Specifies the MAC PIB Attributes

Description	Value
ZMacAckWaitDuration	0x40
ZMacAssociatedPermit	0x41
ZMacAutRequest	0x42
ZMacBattLifeExt	0x43
ZMacBattLeftExtPeriods	0x44
ZMacBeaconMSDU	0x45
ZMacBeaconMSDULength	0x46
ZMacBeaconOrder	0x47
ZMacBeaconTxTime	0x48
ZMacBSN	0x49
ZMacCoordExtendedAddress	0x4A
ZMacCoordShortAddress	0x4B
ZMacDSN	0x4C
ZMacGTSPermit	0x4D
ZMacMaxCSMABackoffs	0x4E
ZMacMinBE	0x4F
ZMacPanId	0x50
ZMacPromiscuousMode	0x51
ZMacRxOnIdle	0x52
ZMacShortAddress	0x53
ZMacSuperframeOrder	0x54
ZMacTransactionPersistenceTime	0x55
ZMacAssociatedPanCoord	0x56



ZMacMaxBE	0x57
ZMacMaxFrameTransmitWaitTime	0x58
ZMacMaxFrameRetries	0x59
ZMacResponseWaitTime	0x5A
ZMacSyncSymbolOffset	0x5B
ZMacTimestampSupported	0x5C
ZMacSecurityEnabled	0x5D
ZMacPhyTransmitPower	0xE0
ZMacLogicalChannel	0xE1
ZMacExtendedAddress	0xE2
ZMacAltBe	0xE3

### 3.7.3 ZMAC\_GET\_RESPONSE

Cmd	Length	Status
0x1087	0x01	1byte

**Status** indicates SUCCESS → FAILURE

## 3.8 ZMAC\_ORPHAN\_RESPONSE

### 3.8.1 Description

This command is used to send a response to the Orphan Indication message.

### 3.8.2 Usage

Cmd	Length	OrphanExtendedAddress	OrphanShortAddress	AssociatedMember
0x0089	0x14	8 bytes	2 bytes	1 byte

KeySource	SecurityLevel	KeyIdMle	KeyIndex
8 bytes	1 bytes	1 bytes	1 byte

**OrphanExtendedAddress:** Extended address of the device sending the orphan notification.

**OrphanShortAddress:** Allotted short address of the orphaned device.

**AssociatedMember:** TRUE if the orphaned device is associated with this coordinator.

**KeySource:** Key Source of this frame.

**SecurityLevel:** Security Level of this frame.

Descripti n	Value
No security is used	0x00
MIC-32 authenticati n is used	0x01
MIC-64 authenticati n is used	0x02
MIC-128 authenticati n is used	0x03
AES encrypti n is used	0x04
AES encrypti n and MIC-32 authenticati n are used	0x05
AES encrypti n and MIC-64 authenticati n are used	0x06
AES encrypti n and MIC-128 authenticati n are used	0x07

**KeyIdMode:** Key Id Mode of this frame.

Descripti n	Value
Key is not used / Key is determined implicitly	0x00
Key is determined from the 1-byte key index	0x01
Key is determined from the 4-byte key index	0x02
Key is determined from the 8-byte key index	0x03

**KeyIndex:** Key Index of this frame.

### 3.8.3 ZMAC\_ORPHAN\_RESPOND\_RESPONSE

Cmd	Length	Status
0x1089	0x01	1byte

**Status** indicates SUCCESS or FAILURE

## 3.9 ZMAC\_POLL\_REQUEST

### 3.9.1 Description

This command is used to send a MAC data request packet.

### 3.9.2 Usage

Cmd	Length	CommandAddressMode	CommandAddress	CommandPanID
0x008F	0x10	1 byte	8 bytes	2 bytes

KeySource	SecurityLevel	KeyIdMode	KeyIndex
8 bytes	1 bytes	1 bytes	1 byte

**CoordAddressMode:** Specified the format of the coordinat address

Descripti n	Value
Address N o Present	0x00
Address 16 bit	0x02
Address 48 bit	0x03
Broadcast	0xFF

**CoordAddress:** Coordinat address

**CoordPanID:** PAN Id of the coordinat

**KeySource:** Key Source of this frame.

**SecurityLevel:** Security Level of this frame.

Descripti n	Value
No security is used	0x00
MIC-32 authentication is used	0x01
MIC-48 authentication is used	0x02
MIC-128 authentication is used	0x03
AES encryption is used	0x04
AES encryption and MIC-32 authentication are used	0x05
AES encryption and MIC-48 authentication are used	0x06
AES encryption and MIC-128 authentication are used	0x07

**KeyIdMode:** Key Id Mode of this frame.

Descripti n	Value
Key is not used / Key is determined implicitly	0x00
Key is determined from the 1-byte key index	0x01
Key is determined from the 4-byte key index	0x02
Key is determined from the 8-byte key index	0x03

**KeyIndex:** Key Index of this frame.

### 3.9.3 ZMAC\_POLL\_RESPONSE

Cmd	Length	Status
0x108F	0x01	1 byte

**Status** indicates SUCCESS ↗ FAILURE

## 3.10 ZMAC\_RESET\_REQUEST

### 3.10.1 Description

This command is used to send a MAC Reset command to reset MAC state machine.

### 3.10.2 Usage

Cmd	Length	Set Default
0x0081	0x01	1 byte

**Set Default:**

### 3.10.3 ZMAC\_RESET\_RESPONSE

Cmd	Length	Status
0x1081	0x01	1 byte

**Status** indicates SUCCESS ↗ FAILURE

## 3.11 ZMAC\_RX\_ENABLE\_REQUEST

### 3.11.1 Description

This command contains timing information that tells the device when to enable ↗ disable its receiver, in order to schedule a data transfer between itself and another device. The information is sent from the upper layers directly to the MAC sub-layer.

### 3.11.2 Usage

Cmd	Length	DeferPermit	RxOnTime	RxOnDuration
0x008A	0x09	1 byte	4 bytes	4 bytes

**DeferPermit:**

**RxOnTime:**

**RxOnDuration:**

### 3.11.3 ZMAC\_RX\_ENABLE\_RESPONSE

Cmd	Length	Status
0x108A	0x01	1 byte

**Status** indicates SUCCESS or FAILURE

### 3.12 ZMAC\_SCAN\_REQUEST

#### 3.12.1 Description

This command is used to send a request to the device to perform a network scan.

#### 3.12.2 Usage

Cmd	Length	ScanChannels	ScanType	ScanDuration	ChannelPage	MaxResults
0x008B	0x13	4 bytes	1 byte	1 byte	1 byte	1 byte

KeySource	SecurityLevel	KeyIdMode	KeyIndex
8 bytes	1 bytes	1 bytes	1 byte

**ScanChannels:** Channels to be scanned.

**ScanType:** Type of the scan.

Description	Value
Energy detect scan	0x00
Active scan	0x01
Passive scan	0x02
Orphan Scan	0x03

**ScanDuration:** Duration of the scan.

**ChannelPage:** Channel Page of this frame.

**MaxResults:** Maximum number of PAN descriptor results.

**KeySource:** Key Source of this frame.

**SecurityLevel:** Security Level of this frame.

Description	Value
No security is used	0x00

MIC-32 authentication is used	0x01
MIC-64 authentication is used	0x02
MIC-128 authentication is used	0x03
AES encryption is used	0x04
AES encryption and MIC-32 authentication are used	0x05
AES encryption and MIC-64 authentication are used	0x06
AES encryption and MIC-128 authentication are used	0x07

**KeyIdMode:** Key Id Mode of this frame.

Description	Value
Key is not used / Key is determined implicitly	0x00
Key is determined from the 1-byte key index	0x01
Key is determined from the 4-byte key index	0x02
Key is determined from the 8-byte key index	0x03

**KeyIndex:** Key Index of this frame.

### 3.12.3 ZMAC\_SCAN\_RESPONSE

Cmd	Length	Status
0x108B	0x01	1 byte

**Status** indicates SUCCESS or FAILURE

## 3.13 ZMAC\_SET\_REQUEST

### 3.13.1 Description

This command is used to request the device to write a MAC PIB value.

### 3.13.2 Usage

Cmd	Length	Attribute	AttributeValue
0x008C	0x02-0x10	1 byte	1-15 bytes

**Attribute:** MAC PIB

Description	Value
-------------	-------

ZMacAckWaitDuration	0x40
ZMacAssociatePermit	0x41
ZMacAutorequest	0x42
ZMacBattLifeExt	0x43
ZMacBattLeftExtPeriods	0x44
ZMacBeaconMSDU	0x45
ZMacBeaconMSDULength	0x46
ZMacBeaconOrder	0x47
ZMacBeaconTxTime	0x48
ZMacBSN	0x49
ZMacCoordExtendedAddress	0x4A
ZMacCoordShortAddress	0x4B
ZMacDSN	0x4C
ZMacGTSPermit	0x4D
ZMacMaxCSMABackoffs	0x4E
ZMacMinBE	0x4F
ZMacPanId	0x50
ZMacPromiscuousMode	0x51
ZMacRxOnIdle	0x52
ZMacShortAddress	0x53
ZMacSuperframeOrder	0x54
ZMacTransactionPersistenceTime	0x55
ZMacAssociatePanCoord	0x56
ZMacMaxBE	0x57
ZMacMaxFrameTransmitWaitTime	0x58
ZMacMaxFrameRetries	0x59
ZMacResponseWaitTime	0x5A
ZMacSyncSymbolOffset	0x5B
ZMacTimestampSupported	0x5C
ZMacSecurityEnabled	0x5D
ZMacPhyTransmitPower	0xE0
ZMacLogicalChannel	0xE1
ZMacExtendedAddress	0xE2
ZMacAltBe	0xE3

**AttributeValue:** 1-16 bytes value of the attribute

### 3.13.3 ZMAC\_SET\_RESPONSE

Cmd	Length	Status
0x108C	0x01	1 byte

**Status** indicates SUCCESS or FAILURE

## 3.14 ZMAC\_START\_REQUEST

### 3.14.1 Description

This command is used to request the MAC to transmit beacons and become a coordinator.

### 3.14.2 Usage

Cmd	Length	StartTime	PanID	LogicalChannel	ChannelPage	BeaconOrder
0x008D	0x23	4 bytes	2 bytes	1 byte	1 byte	1 byte

SuperFrameOrder	PanCoordinator	BatteryLifeExt	CoordRealignment
1 byte	1 byte	1 byte	1 byte

RealignKeySource	RealignSecurityLevel	RealignKeyIdMode	RealignKeyIndex
8 bytes	1 bytes	1 bytes	1 byte

BeaconKeySource	BeaconSecurityLevel	BeaconKeyIdMode	BeaconKeyIndex
8 bytes	1 bytes	1 bytes	1 byte

**StartTime:** The time to begin transmitting beacons relative to the received beacon.

**PanID:** The PAN ID to use. This parameter is ignored if Pan Coordinator is FALSE.

**LogicalChannel:** The logical channel to use. This parameter is ignored if Pan Coordinator is FALSE.

**ChannelPage:** The channel page to use. This parameter is ignored if Pan Coordinator is FALSE.

**BeaconOrder:** The exponent used to calculate the beacon interval.

**SuperFrameOrder:** The exponent used to calculate the superframe duration.

**PanCoordinator:** Set to TRUE to start a network as PAN coordinator.

**BatteryLifeExt:** If this value is TRUE, the receiver is disabled after MAC\_BATT\_LIFE\_EXT\_PERIODS full backoff periods following the interframe spacing period of the beacon frame.



**CoordRealignmet:** Set to TRUE to transmit a coordinat to realignment prior to changing the superframe configuration.

**RealignKeySource:** Key Source of this frame.

**RealignSecurityLevel:** Security Level of this frame.

Description	Value
No security is used	0x00
MIC-32 authentication is used	0x01
MIC-64 authentication is used	0x02
MIC-128 authentication is used	0x03
AES encryption is used	0x04
AES encryption and MIC-32 authentication are used	0x05
AES encryption and MIC-64 authentication are used	0x06
AES encryption and MIC-128 authentication are used	0x07

**RealignKeyIdMode:** Key Id Mode of this frame.

Description	Value
Key is not used / Key is determined implicitly	0x00
Key is determined from the 1-byte key index	0x01
Key is determined from the 4-byte key index	0x02
Key is determined from the 8-byte key index	0x03

**RealignKeyIndex:** Key Index of this frame.

**BeaconKeySource:** Key Source of this frame.

**BeaconSecurityLevel:** Security Level of this frame.

Description	Value
No security is used	0x00
MIC-32 authentication is used	0x01
MIC-64 authentication is used	0x02
MIC-128 authentication is used	0x03
AES encryption is used	0x04
AES encryption and MIC-32 authentication are used	0x05

AES encryption and MIC-64 authentication are used	0x00
AES encryption and MIC-128 authentication are used	0x07

**BeaconKeyIdMode:** Key Id Mode of this frame.

Description	Value
Key is not used / Key is determined implicitly	0x00
Key is determined from the 1-byte key index	0x01
Key is determined from the 4-byte key index	0x02
Key is determined from the 8-byte key index	0x03

**BeaconKeyIndex:** Key Index of this data frame.

### 3.14.3 ZMAC\_START\_RESPONSE

Cmd	Length	Status
0x108D	0x01	1 byte

**Status** indicates SUCCESS or FAILURE

## 3.15 ZMAC\_SYNCHRONIZE\_REQUEST

### 3.15.1 Description

This command is used to request synchronization to the current network beacon.

### 3.15.2 Usage

Cmd	Length	LogicalChannel	ChannelPage	TrackBeacon
0x008E	0x03	1 byte	1 byte	1 byte

**LogicalChannel:** The logical channel to use.

**ChannelPage:** The channel page to use.

**TrackBeacon:** Set to TRUE to continue tracking beacons after synchronizing with the first beacon. Set to FALSE to only synchronize with the first beacon.

### 3.15.3 ZMAC\_SYNCHRONIZE\_RESPONSE

Cmd	Length	Status
0x108E	0x01	1byte

**Status** indicates SUCCESS ↗ FAILURE.

### 3.16 ZMAC\_PURGE\_REQUEST

#### 3.16.1 Description

This command is used to send a request to the device to purge a data frame.

#### 3.16.2 Usage

Cmd	Length	msduHandle
0x0090	0x01	1 byte

**msduHandle:** Handle ↗ this frame

#### 3.16.3 ZMAC\_PURGE\_RESPONSE

Cmd	Length	Status
0x1090	0x01	1byte

**Status** indicates SUCCESS ↗ FAILURE

## 4. MAC CALLBACK INTERFACE

### 4.1 Introduction

This message interface enables the target MAC to issue callbacks to the tester. Each message has a response to a message that is sent by the tester to the target. The response message only indicates successful reception of the callback message. These messages are subscription messages. Refer to SYS\_CALLBACK\_SUBSCRIBE for a description of the subscription mechanism.

### 4.2 NWK\_ASSOCIATE\_INDICATION

#### 4.2.1 Description

This function is used to indicate the reception of an association request command.

#### 4.2.2 Usage

Cmd	Length	DeviceAddress	CapabilityInformation
0x2081	0x14	8 bytes	1 byte

KeySource	SecurityLevel	KeyIdMode	KeyIndex
8 bytes	1 bytes	1 bytes	1 byte

**DeviceAddress:** Address of the associated device

**CapabilityInformation:** capability of the associated device

**KeySource:** Key Source of this frame.

**SecurityLevel:** Security Level of this frame.

Description	Value
No security is used	0x00
MIC-32 authentication is used	0x01
MIC-64 authentication is used	0x02
MIC-128 authentication is used	0x03
AES encryption is used	0x04
AES encryption and MIC-32 authentication are used	0x05
AES encryption and MIC-64 authentication are used	0x06
AES encryption and MIC-128 authentication are used	0x07

**KeyIdMode:** Key Id Mode of this frame.

Description	Value
Key is not used / Key is determined implicitly	0x00
Key is determined from the 1-byte key index	0x01
Key is determined from the 4-byte key index	0x02
Key is determined from the 8-byte key index	0x03

**KeyIndex:** Key Index of this frame.

## 4.3 NWK\_ASSOCIATE\_CONFIRMATION

### 4.3.1 Description

This function is used to inform the upper layers of the initiating device whether its request to associate was successful or unsuccessful.

### 4.3.2 Usage

Cmd	Length	Status	AllocatedAddress
0x2082	0x0E	1 byte	2 bytes

KeySource	SecurityLevel	KeyIdMode	KeyIndex
8 bytes	1 bytes	1 bytes	1 byte

**Status:** Status of the association.

**AllocatedAddress:** Allocated address.

**KeySource:** Key Source of this frame.

**SecurityLevel:** Security Level of this frame.

Description	Value
No security is used	0x00
MIC-32 authentication is used	0x01
MIC-64 authentication is used	0x02
MIC-128 authentication is used	0x03
AES encryption is used	0x04
AES encryption and MIC-32 authentication are used	0x05
AES encryption and MIC-64 authentication are used	0x06
AES encryption and MIC-128 authentication are used	0x07

**KeyIdMode:** Key Id Mode of this frame.

Description	Value
Key is not used / Key is determined implicitly	0x00
Key is determined from the 1-byte key index	0x01
Key is determined from the 4-byte key index	0x02
Key is determined from the 8-byte key index	0x03

**KeyIndex:** Key Index of this frame.

## 4.4 NWK\_DISASSOCIATE\_INDICATION

### 4.4.1 Description

This function is used to indicate the reception of a disassociation notification command.

### 4.4.2 Usage

Cmd	Length	DeviceAddress	DisassociateReason
0x208	0x14	8 bytes	1 byte

KeySource	SecurityLevel	KeyIdMode	KeyIndex
8 bytes	1 bytes	1 bytes	1 byte

**DeviceAddress:** Address of the initiating device

**DisassociateReason:** Reason of the disassociation

**KeySource:** Key Source of this frame.

**SecurityLevel:** Security Level of this frame.

Description	Value
No security is used	0x00
MIC-32 authentication is used	0x01
MIC-64 authentication is used	0x02
MIC-128 authentication is used	0x03
AES encryption is used	0x04
AES encryption and MIC-32 authentication are used	0x05
AES encryption and MIC-64 authentication are used	0x06

AES encryption and MIC-128 authentication are used	0x07
--	------

**KeyIdMode:** Key Id Mode of this frame.

Description	Value
Key is not used / Key is determined implicitly	0x00
Key is determined from the 1-byte key index	0x01
Key is determined from the 4-byte key index	0x02
Key is determined from the 8-byte key index	0x03

**KeyIndex:** Key Index of this frame.

## 4.5 NWK\_DISASSOCIATE\_CONFIRMATION

### 4.5.1 Description

This function is sent as the result of a disassociation request.

### 4.5.2 Usage

Cmd	Length	Status	DeviceAddressMode	DeviceAddress	PanID
0x2087	0x0c	1 byte	1 byte	8 bytes	2 byte

**Device Address:**

**Status:** SUCCESS or FAILURE

## 4.6 NWK\_BEACON\_NOTIFY\_INDICATION

### 4.6.1 Description

This function is used to send parameters contained within a beacon frame received by the MAC sub-layer to the next higher layer. The function also sends a measure of the link quality and the time the beacon was received.

### 4.6.2 Usage

Cmd	Length	BSN	TimeStamp	ChildAddressMode	ChildAddress	ChildPanID
0x2083	0x23 – 0x43	1 byte	4 bytes	1 byte	8 bytes	2 bytes

SuperFrameSpec	LogicalChannel	GTSPermit	LinkQuality	SecurityFailure
2 bytes	1 byte	1 byte	1 byte	1 byte

KeySource	SecurityLevel	KeyIdMode	KeyIndex
-----------	---------------	-----------	----------

8 bytes	1 bytes	1 bytes	1 byte
---------	---------	---------	--------

PendingAddrSpec	AddrList	SDULength	SDU
1 byte	0-32 bytes	1 byte	0-32 bytes

**BSN:** Beacon sequence number.

**TimeStamp:** Time stamp when the beacon is received.

**CoordAddressMode:**

Description	Value
Address Not Present	0x00
Address 16 bit	0x02
Address 48 bit	0x03
Broadcast	0xFF

**CoordAddress:** Address of the coordinator.

**CoordPanID:** PAN ID of the coordinator.

**SuperFrameSpec:** The superframe specification of the network.

**LogicalChannel:** The logical channel of the network.

**GTSPermit:** TRUE if coordinator accepts GTS requests.

**LinkQuality:** The link quality of the received beacon.

**SecurityFailure:** Set to TRUE if there was an error in the security processing.

**KeySource:** Key Source of this frame.

**SecurityLevel:** Security Level of this frame.

Description	Value
No security is used	0x00
MIC-32 authentication is used	0x01
MIC-48 authentication is used	0x02
MIC-128 authentication is used	0x03
AES encryption is used	0x04
AES encryption and MIC-32 authentication are used	0x05



AES encryption and MIC-64 authentication are used	0x00
AES encryption and MIC-128 authentication are used	0x07

**KeyIdMode:** Key Id Mode of this frame.

Description	Value
Key is not used / Key is determined implicitly	0x00
Key is determined from the 1-byte key index	0x01
Key is determined from the 4-byte key index	0x02
Key is determined from the 8-byte key index	0x03

**KeyIndex:** Key Index of this frame.

**PendingAddrSpec:** The beacon pending address specification.

**AddrList:** The list of device addresses from which the sender of the beacon has data.

**SDULength:** Length of the SDU data.

**SDU:** Actual SDU data. Can be 0-32 bytes

## 4.7 NWK\_ORPHAN\_INDICATION

### 4.7.1 Description

This function allows the MLME to notify the next higher layer of an orphaned device.

### 4.7.2 Usage

Cmd	Length	OrphanAddress
0x208a	0x13	8 bytes

KeySource	SecurityLevel	KeyIdMode	KeyIndex
8 bytes	1 bytes	1 bytes	1 byte

**OrphanAddress:**

**KeySource:** Key Source of this frame.

**SecurityLevel:** Security Level of this frame.

Description	Value
No security is used	0x00
MIC-32 authentication is used	0x01

MIC-64 authentication is used	0x02
MIC-128 authentication is used	0x03
AES encryption is used	0x04
AES encryption and MIC-32 authentication are used	0x05
AES encryption and MIC-64 authentication are used	0x06
AES encryption and MIC-128 authentication are used	0x07

**KeyIdMode:** Key Id Mode of this frame.

Description	Value
Key is not used / Key is determined implicitly	0x00
Key is determined from the 1-byte key index	0x01
Key is determined from the 4-byte key index	0x02
Key is determined from the 8-byte key index	0x03

**KeyIndex:** Key Index of this frame.

## 4.8 NWK\_SCAN\_CONFIRMATION

### 4.8.1 Description

This function reports the results of a channel scan request.

### 4.8.2 Usage

Cmd	Length	Status	EDMaxEnergy	ScanType
0x208c	0x09 – 0x29	1 byte	1 byte	1 byte

ChannelPage	UnscannedChannels	ResultCount	Results
1 byte	4 bytes	1 byte	0-32 bytes

**Status:** Status of the scan request

**EDMaxEnergy:** Maximum possible energy detect value

**ScanType:** Type of scan request

Description	Value
Energy detect scan	0x00

Active scan	0x01
Passive scan	0x02
Orphan Scan	0x03

**ChannelPage:** The channel page of the scan

**UnscannedChannels:** Bit mask of channels that were not scanned

**ResultListSize:** The number of PAN descriptors returned in the results list

**Results:** List of energy measurements of PAN descriptors (depends on the scan type)

## 4.9 NWK\_START\_CONFIRMATION

### 4.9.1 Description

This function reports the success of the start request.

### 4.9.2 Usage

Cmd	Length	Status
0x208e	0x01	1 byte

**Status:** SUCCESS or FAILURE

## 4.10 NWK\_SYNCHRONIZATION\_LOSS\_INDICATION

### 4.10.1 Description

This function indicates the loss of synchronization of a network beacon.

### 4.10.2 Usage

Cmd	Length	Reason	PANId	LogicalChannel	ChannelPage
0x2080	0x10	1 byte	2bytes	1byte	1byte

KeySource	SecurityLevel	KeyIdMode	KeyIndex
8 bytes	1 bytes	1 bytes	1 byte

**Reason:** Reason that synchronization was lost

**LogicalChannel:** Logical channel of this frame.

**ChannelPage:** Channel page of this frame.

**KeySource:** Key Source of this frame.

**SecurityLevel:** Security Level of this frame.

Description	Value
No security is used	0x00
MIC-32 authentication is used	0x01
MIC-64 authentication is used	0x02
MIC-128 authentication is used	0x03
AES encryption is used	0x04
AES encryption and MIC-32 authentication are used	0x05
AES encryption and MIC-64 authentication are used	0x06
AES encryption and MIC-128 authentication are used	0x07

**KeyIdMode:** Key Id Mode of this frame.

Description	Value
Key is not used / Key is determined implicitly	0x00
Key is determined from the 1-byte key index	0x01
Key is determined from the 4-byte key index	0x02
Key is determined from the 8-byte key index	0x03

**KeyIndex:** Key Index of this frame.

## 4.11 NWK\_POLL\_CONFIRMATION

### 4.11.1 Description

This function reports the results of a polling attempt.

### 4.11.2 Usage

Cmd	Length	Status
0x208b	0x01	1 byte

**Status:** SUCCESS → FAILURE

## 4.12 NWK\_COMM\_STATUS\_INDICATION

### 4.12.1 Description

This function reports a communication status error.

### 4.12.2 Usage

ID	Length	Status	SrcAddressMode	SrcAddress	DestAddressMode	DestAddress	PanID	Reason
0x208d	0x21	1 byte	1 byte	8 bytes	1 byte	8 bytes	2 bytes	1 byte

KeySource	SecurityLevel	KeyIdMode	KeyIndex
8 bytes	1 bytes	1 bytes	1 byte

**Status:** Status of the indication.

**SrcAddressMode:** Indicates the format of the source address

Description	Value
Address Not Present	0x00
Address 16 bit	0x02
Address 64 bit	0x03
Broadcast	0xFF

**SrcAddress:** Address of the initiating device

**DestAddressMode:**

Description	Value
Address Not Present	0x00
Address 16 bit	0x02
Address 64 bit	0x03
Broadcast	0xFF

**DestAddress:** Address of the remote device

**PanID:** PAN ID of the remote device

**Reason:** The reason of the event.

**KeySource:** Key Source of this frame.

**SecurityLevel:** Security Level of this frame.

Description	Value
No security is used	0x00
MIC-32 authentication is used	0x01
MIC-64 authentication is used	0x02

MIC-128 authentication is used	0x03
AES encryption is used	0x04
AES encryption and MIC-32 authentication are used	0x05
AES encryption and MIC-64 authentication are used	0x06
AES encryption and MIC-128 authentication are used	0x07

**KeyIdMode:** Key Id Mode of this frame.

Description	Value
Key is not used / Key is determined implicitly	0x00
Key is determined from the 1-byte key index	0x01
Key is determined from the 4-byte key index	0x02
Key is determined from the 8-byte key index	0x03

**KeyIndex:** Key Index of this frame.

## 4.13 NWK\_DATA\_CONFIRMATION

### 4.13.1 Description

This function is used to send the results of a request to transfer a data SPDU (MSDU) from a local SSCS entity to a single peer SSCS entity, or multiple peer SSCS entities.

### 4.13.2 Usage

Cmd	Length	Status	msduHandle	Timestamp	Timestamp2
0x2084	0x08	1 byte	1 byte	4 bytes	2 bytes

**Status:** Status of the frame

**msduHandle:** Handle of the frame

**Timestamp:** The time, in backoffs, at which the data were received

**Timestamp2:** The time, in internal MAC timer units, at which the data were received

## 4.14 NWK\_DATA\_INDICATION

### 4.14.1 Description

This function indicates the transfer of a data SPDU (MSDU) from the MAC sub-layer to the local SSCS entity.

#### 4.14.2 Usage

Cmd	Length	SrcAddressMode	SrcAddress	DestAddressMode	DestAddress
0x2085	0x2C – 0x92	1 byte	8 bytes	1 byte	8 bytes

Timestamp	Timestamp2	SrcPanID	DestPanID	mpduLinkQuality	Lqi	Rssi	Dsn
4 bytes	2 bytes	2 bytes	2 bytes	1 byte	1 byte	1 byte	1 byte

KeySource	SecurityLevel	KeyIdMode	KeyIndex	MSDULength	MSDU
8 bytes	1 bytes	1 bytes	1 byte	1 byte	0-102 bytes

**SrcAddressMode:**

Description	Value
Address Not Present	0x00
Address 16 bit	0x02
Address 48 bit	0x03
Broadcast	0xFF

**SrcAddress:** Address of the initiating device

**DestAddressMode:**

Description	Value
Address Not Present	0x00
Address 16 bit	0x02
Address 48 bit	0x03
Broadcast	0xFF

**DestAddress:** Address of the remote device

**Timestamp:** The time, in backoffs, at which the data were received

**Timestamp2:** The time, in internal MAC timer units, at which the data were received

**SrcPanID:** Pan Id of the initiating device

**DestPanID:** Pan Id of the remote device

**mpduLinkQuality:**

**Lqi:** Raw LQI of the received data frame.

**Rssi:** Raw RSSI of the received data frame.

**Dsn:**

**KeySource:** Key Source of this frame.

**SecurityLevel:** Security Level of this frame.

Description	Value
No security is used	0x00
MIC-32 authentication is used	0x01
MIC-64 authentication is used	0x02
MIC-128 authentication is used	0x03
AES encryption is used	0x04
AES encryption and MIC-32 authentication are used	0x05
AES encryption and MIC-64 authentication are used	0x06
AES encryption and MIC-128 authentication are used	0x07

**KeyIdMode:** Key Id Mode of this frame.

Description	Value
Key is not used / Key is determined implicitly	0x00
Key is determined from the 1-byte key index	0x01
Key is determined from the 4-byte key index	0x02
Key is determined from the 8-byte key index	0x03

**KeyIndex:** Key Index of this frame.

**MSDULength:** Length of the data

**MSDU:** The actual data

## 4.15 NWK\_RX\_ENABLE\_CONFIRMATION

### 4.15.1 Description

This function reports the results of an RX enable attempt.

### 4.15.2 Usage

Cmd	Length	Status
0x208f	0x01	1 byte

**Status:** SUCCESS or FAILURE



## 4.16 NWK\_PURGE\_CONFIRMATION

### 4.16.1 Description

This function reports the results of a purge attempt.

### 4.16.2 Usage

Cmd	Status	Length	msduHandle
0x2090	1 byte	0x02	1 byte

**msduHandle:** Application-defined handle value associated with the data request.

**Status:** SUCCESS or FAILURE

## 5. NWK Interface

### 5.1 Introduction

These commands allow the tester to issue commands to the NWK layer in the target and receive responses. Each of these messages has a corresponding message that is returned by the target. The response message only indicates that the command message was received and executed. The result of the command execution will be conveyed to the tester via a callback message interface (defined in the next section), if applicable.

### 5.2 NLDE\_DATA\_REQUEST

#### 5.2.1 Description

This command enables the tester to request the transfer of data from the local APS sub-layer to a peer APS sub-layer entity.

#### 5.2.2 Usage

The message format is as follows.

SecurityEnable	DiscoverRoute	RadiusCounter
----------------	---------------	---------------

Status is a 1 byte field that indicates SUCCESS → FAILURE (see appendix f → NLDE status codes).

### 5.3 NLME\_NETWORK\_FORMATION\_REQUEST

#### 5.3.1 Description

This command is used by tester to request (on behalf of the next higher layer) that the device be initiated as a coordinator.

#### 5.3.2 Usage

The message format is as follows

Cmd = 0x0102	Len = 0x03	PanID	ScanChannels	ScanDuration	BeaconOrder	SuperFrameOrder	BatteryLifeExtension
--------------	------------	-------	--------------	--------------	-------------	-----------------	----------------------

PanID is a 2 bytes short address of the intended device.

ScanChannels is a 4 bytes field indicates the list of channels in bit map format.

ScanDuration is a byte field represents the scanning time.

BeaconOrder is a byte field which represents the beacon order type. This parameter doesn't make sense in a non-beacon device. If this value is set to anything other than BEACON\_ORDER\_NO\_BEACONS (0xf) and the device is compiled as a mesh mode device, the response message's status field will be set to ZNwkInvalidParam (0xc1).

SuperFrameOrder is a byte field which specifies the length of the active portion of the superframe. This parameter doesn't make sense in a non-beacon device. If this value is set to anything other than BEACON\_ORDER\_NO\_BEACONS (0xf) and the device is compiled as a mesh mode device, the response message's status field will be set to ZNwkInvalidParam (0xc1).

BatteryLifeExtension is a byte field which indicates whether NLME will request that the ZigBee coordinator is started supporting battery life extension mode.

#### 5.3.3 NLME\_NETWORK\_FORMATION\_RESPONSE

Cmd = 0x1102	Len = 0x01	Status
--------------	------------	--------

Status is a 1 byte field that indicates SUCCESS → FAILURE (see appendix f → NLME status codes).

## 5.4 NLME\_PERMITJOINING\_REQUEST

### 5.4.1 Description

This command is used by the tester to define how the next higher layer of a coordinator device would permit devices to join its network for a fixed period.

### 5.4.2 Usage

Cmd = 0x0103	Len = 0x01	PermitDuration
--------------	------------	----------------

PermitDuration is a byte field which indicates the length of time during which the ZigBee coordinator will be allowing associations. The values 0x00 and 0xff indicate that permission is disabled or enabled, respectively.

### 5.4.3 Response

Cmd = 0x1103	Len = 0x01	Status
--------------	------------	--------

Status is a 1 byte field that indicates SUCCESS or FAILURE (see appendix for NLME status codes).

## 5.5 NLME\_JOIN\_REQUEST

### 5.5.1 Description

This command is used by tester to make a request (on behalf of the next higher layer) to join the device itself or another device to a network.

### 5.5.2 Usage

The message format is as follows.

Cmd = 0x0104	Len = 0x0a	PanID	Channel	CapabilityInfo
--------------	------------	-------	---------	----------------

PanID is a 2 bytes field and indicates the short address of the intended device.

Channel is a byte field and specifies the channel of the intended device.

CapabilityInfo is a byte field which is a bitmap which specifies the operational capabilities of the joining device. The meaning of each bit is specified in the following table.

Meaning	Bit map
Alternate Pan coordinator	0
Device type	1
Power source	2
Receiver on when idle	3

Reserved	4-5
Security capability	6
Alternate address	7

### 5.5.3 NLME\_JOIN\_RESPONSE

Cmd = 0x1104	Len = 0x01	Status
--------------	------------	--------

Status is a 1 byte field that indicates SUCCESS or FAILURE (see appendix f for NLME status codes).

## 5.6 NLME\_LEAVE\_REQUEST

### 5.6.1 Description

This command is used by tester to make a request (on behalf of the next higher layer) that the device itself and one of its child devices leave the network.

### 5.6.2 Usage

The message format is as follows

Cmd = 0x0105	Len = 0x0A	DeviceAddress	RemoveChildren	Reassociate
--------------	------------	---------------	----------------	-------------

DeviceAddress is an 8-byte field that specifies which device is being asked to leave the network. An address of zero is used to indicate the device itself. An 8-byte IEEE extended address is used to indicate one of the device's children.

RemoveChildren (a 1 byte field) indicates whether the device, specified by the DeviceAddress, should remove its children.

Reassociate (a 1 byte field) indicates whether the device, specified by the DeviceAddress, should reassociate after it leaves. It has value of either 1 or 0. If the field has a value of 1, the device being asked to leave from the current parent is requested to rejoin the network again after the leave. Otherwise, it does not reassociate.

### 5.6.3 NLME\_LEAVE\_RESPONSE

Cmd = 0x1105	Len = 0x01	Status
--------------	------------	--------

Status is a 1 byte field that indicates SUCCESS or FAILURE (see appendix f for NLME status codes).

## 5.7 NLME\_RESET\_REQUEST

### 5.7.1 Description

This command is used by tester to make a request (on behalf of the next higher layer) that the NWK layer perform a reset operation.

### 5.7.2 Usage

The message format is as follows

Cmd = 0x0100	Len = 0x00
--------------	------------

### 5.7.3 NLME\_RESET\_RESPONSE

Cmd = 0x1100	Len = 0x01	Status
--------------	------------	--------

Status is a 1 byte field that indicates SUCCESS or FAILURE (see appendix for NLME status codes).

## 5.8 NLME\_SYNC\_REQUEST

### 5.8.1 Usage

The message format is as follows

Cmd = 0x0107	Len = 0x08	Track
--------------	------------	-------

Track is a 1 byte field. 1 = track beacon, = 0 don't track beacon.

### 5.8.2 NLME\_SYNC\_RESPONSE

Cmd = 0x1107	Len = 0x01	Status
--------------	------------	--------

Status is a 1 byte field that indicates SUCCESS or FAILURE (see appendix for NLME status codes).

## 5.9 NLME\_GET\_REQUEST

### 5.9.1 Description

This command is used by tester to make a request (on behalf of the next higher layer) to read the value of an attribute from the NWK information base (NIB).

### 5.9.2 Usage

The message format is as follows

Cmd = 0x0108	Len = 0x02	NIBAttribute	Index	Value
--------------	------------	--------------	-------	-------

NIBAttribute is a byte field which identifies the intended attribute of the NIB.

Index is a byte field and is used to index into table NIB items.

Value is a two-byte field that contains the data value

### 5.9.3 NLME\_GET\_RESPONSE

Cmd = 0x1108	Len = 0x0B	Status	NIBAttributeValue
--------------	------------	--------	-------------------

Status is a 1 byte field that indicates SUCCESS or FAILURE.

NIBAttributeValue is a 10 byte field to store the attribute value. The field is 10 bytes to accommodate the largest attribute.

## 5.10 NLME\_SET\_REQUEST

### 5.10.1 Description

This command is used by tester to make a request (on behalf of the next higher layer) to set the value of an attribute in the NWK information base (NIB).

### 5.10.2 Usage

The message format is as follows

Cmd = 0x0109	Len = 0x0C	NIBAttribute	Index	NIBAttributeValue
--------------	------------	--------------	-------	-------------------

NIBAttribute is a byte field which identifies the intended attribute of the NIB.

Index is a 1 byte field which contains the index into table NIB items.

NIBAttributeValue is a 16 byte field which specifies the value of the NIBAttribute. The field is 16 bytes to accommodate the largest attribute.

### 5.10.3 NLME\_SET\_RESPONSE

Cmd = 0x1109	Len = 0x01	Status
--------------	------------	--------

Status is a 1 byte field that indicates SUCCESS or FAILURE.

## 5.11 NLME\_NETWORKDISCOVERY\_REQUEST

### 5.11.1 Description

This command is used by tester to scan channels for networks.

### 5.11.2 Usage

The message format is as follows

Cmd = 0x010B	Len = 0x5	ScanChannels	ScanDuration
--------------	-----------	--------------	--------------

ScanChannels is a 4 byte field which is the network scan list

ScanDuration is a 1 byte field which scan duration is 2 times the beacon order values.

### 5.11.3 NLME\_NETWORKDISCOVERY\_RESPONSE

Cmd = 0x110B	Len = 0x01	Status
--------------	------------	--------



Status is a 1 byte field that indicates SUCCESS or FAILURE.

## 5.12 NLME\_ROUTEDISCOVERY\_REQUEST

### 5.12.1 Description

This command is used by tester to force a route discovery.

### 5.12.2 Usage

The message format is as follows

Cmd = 0x010C	Len = 0x3	DestAddress	Repair
--------------	-----------	-------------	--------

DestAddress is a 2 byte field which address to discover route for.

Repair is a 1 byte field which specifies a route repair operation if set to TRUE.

### 5.12.3 NLME\_ROUTEDISCOVERY\_RESPONSE

Cmd = 0x110C	Len = 0x01	Status
--------------	------------	--------

Status is a 1 byte field that indicates SUCCESS → FAILURE.

## 5.13 NLME\_DIRECTJOIN\_REQUEST

### 5.13.1 Description

This allows the tester to request to join itself to another device to a network.

### 5.13.2 Usage

The message format is as follows

Cmd = 0x010D	Len = 0x9	ExtendedAddress	CapabilityInfo
--------------	-----------	-----------------	----------------

ExtendedAddress is an 8 byte field – 4 byte address of child device.

CapabilityInfo is a 1 byte field – capability information of the child device.

### 5.13.3 NLME\_DIRECTJOIN\_RESPONSE

Cmd = 0x110D	Len = 0x01	Status
--------------	------------	--------

Status is a 1 byte field that indicates SUCCESS → FAILURE.

## 5.14 NLME\_ORPHANJOIN\_REQUEST

### 5.14.1 Description

This command is used by tester to search for its parent.

### 5.14.2 Usage

The message format is as follows

Cmd = 0x010E	Len = 0x5	ScanChannels	ScanDuration
--------------	-----------	--------------	--------------

ScanChannels is a 4 byte field – Channels in which to discover parent.

ScanDuration is a 1 byte field – Time duration to scan each channel.

### 5.14.3 NLME\_ORPHANJOIN\_RESPONSE

Cmd = 0x110E	Len = 0x01	Status
--------------	------------	--------

Status is a 1 byte field that indicates SUCCESS or FAILURE.

## 5.15 NLME\_STARTROUTER\_REQUEST

### 5.15.1 Description

This command is used by tester to request the device to start functioning as a router.

### 5.15.2 Usage

The message format is as follows

Cmd = 0x010F	Len = 0x3	BeaconOrder	SuperFrameOrder	BatteryLifeExtension
--------------	-----------	-------------	-----------------	----------------------

BeaconOrder is a 1 byte field – beacon order

SuperFrameOrder is a 1 byte field – superframe order

BatteryLifeExtension is a 1 byte field – true if the device is a battery operated device.

### 5.15.3 NLME\_STARTROUTER\_RESPONSE

Cmd = 0x110F	Len = 0x01	Status
--------------	------------	--------

Status is a 1 byte field that indicates SUCCESS or FAILURE.

## 6. NWK Callback Interface

### 6.1 Introduction

This message interface enables the target NWK to issue callbacks to the tester. Each message has a response to a message that is sent by the tester to the target. The response message only indicates successful reception of the callback message. These messages are subscription messages. Refer to SYS\_CALLBACK\_SUBSCRIBE for a description of the subscription mechanism.

### 6.2 NLDE\_DATA\_CONFIRMATION

#### 6.2.1 Description

This message is issued by the target NWK to the tester to report the results of a request to transfer a data PDU from a local APS sub-layer entity to a single peer APS sub-layer entity.

#### 6.2.2 Usage

The message format is as follows

Cmd=0x0180	Len=0x02	NSDUHandle	Status
------------	----------	------------	--------

NSDUHandle is a byte field which indicates the transaction handle (that was created when the data request was issued)

Status is a 1 byte field that indicates SUCCESS or FAILURE (see appendix for NLDE status codes).

### 6.3 NLDE\_DATA\_INDICATION

#### 6.3.1 Description

This message is issued by the target NWK to the tester to indicate the transfer of a data PDU from the NWK layer to the local APS sub-layer entity.

#### 6.3.2 Usage

The packet format is as follows

Cmd=0x0181	Len=var	SrcAddress	NSDULength	NSDU	LinkQuality
------------	---------	------------	------------	------	-------------

SrcAddress is a 2 byte field which is the short address of the source device.

NSDULength is a byte field which indicates the length of the NSDU.

NSDU is the actual network data packet which is of size indicated by NSDULength.

LinkQuality is a byte field which indicates the link quality measured during reception.

## 6.4 NLME\_NETWORKFORMATION\_CONFIRMATION

### 6.4.1 Description

This message is used by the target NWK to inform the tester of the result of a previous association request command.

### 6.4.2 Usage

The message format is as follows

Cmd=0x0182	Len=0x01	Status
------------	----------	--------

Status is a 1 byte field that indicates SUCCESS or FAILURE (see appendix for NLME status codes).

## 6.5 NLME\_JOIN\_CONFIRMATION

This command is issued by the target NWK (to tester) to notify the next higher layer of the results of its request to join itself or another device to a network.

### 6.5.1 Usage

The packet format is as follows.

Cmd=0x0183	Len=0x09	DeviceAddress	PanId	Status
------------	----------	---------------	-------	--------

DeviceAddress is an 8 byte field which specifies the IEEE extended address of the device

PanId is a 2 byte field that indicates the PAN identifier of the network to which the device has joined.

Status is a 1 byte field that indicates SUCCESS or FAILURE (see appendix for NLME status codes).

## 6.6 NLME\_JOIN\_INDICATION

### 6.6.1 Description

This message is sent by the target to notify the next higher layer of a remote join request.

### 6.6.2 Usage

The message format is as follows

Cmd = 0x0184	Len = 0x0b	ShortAddress	ExtendedAddress	CapabilityInfo
--------------	------------	--------------	-----------------	----------------

ShortAddress is a 2 byte field which indicates the network address of an entity that has been added to the network.

ExtendedAddress is a 8-byte field which specifies the IEEE extended address of the same entity.

CapabilityInf is a byte field which is a bitmap which specifies the operational capabilities of the joining device.

## 6.7 NLME\_LEAVE\_CONFIRMATION

### 6.7.1 Description

This message is sent by the target to indicate to the next higher layer that the device itself or another device is leaving the network.

### 6.7.2 Usage

The message format is as follows

Cmd=0x0185	Len=0x01	DeviceAddress	Status
------------	----------	---------------	--------

DeviceAddress is an 8 byte field which specifies the IEEE extended address of the device

Status is a byte field that indicates SUCCESS or FAILURE.

## 6.8 NLME\_LEAVE\_INDICATION

### 6.8.1 Description

This message is sent by the target to indicate a remote leave request to the next higher layer of a coordinator.

### 6.8.2 Usage

The message format is as follows

Cmd=0x0186	Len=0x9	DeviceAddress	Status
------------	---------	---------------	--------

DeviceAddress is an 8 byte field. The 64 bit IEEE address of an entity that has removed itself from the network or NULL in the case that the device issuing the primitive has been removed from the network by its parent.

Status is a 1 byte field that indicates SUCCESS or FAILURE (see appendix for NLME status codes).

## 6.9 NLME\_SYNC\_INDICATION

### 6.9.1 Description

This message is sent by the target to indicate a sync request to the next higher layer of a coordinator.

### 6.9.2 Usage

The message format is as follows

Cmd=0x0189	Len=0x01	Status
------------	----------	--------

Status is a 1 byte field that indicates SUCCESS or FAILURE (see appendix for NLME status codes).

## 6.10 NLME\_NETWORKDISCOVERY\_CONFIRMATION

### 6.10.1 Description

This message is sent by the target to indicate that the network discovery is complete.

### 6.10.2 Usage

The message format is as follows

Cmd=0x018D	Len = 0x29	Number
------------	------------	--------

RouterCapacity_1	DeviceCapacity_1	Version_1	StackProfile_1	SecurityLevel_1
------------------	------------------	-----------	----------------	-----------------

PanID_1	LogicalChannel_1	BeaconOrder_1	SuperFrameOrder_1
---------	------------------	---------------	-------------------

RouterCapacity_2	DeviceCapacity_2	Version_2	StackProfile_2	SecurityLevel_2
------------------	------------------	-----------	----------------	-----------------

PanID_2	LogicalChannel_2	BeaconOrder_2	SuperFrameOrder_2
---------	------------------	---------------	-------------------

RouterCapacity_3	DeviceCapacity_3	Version_3	StackProfile_3	SecurityLevel_3
------------------	------------------	-----------	----------------	-----------------

PanID_3	LogicalChannel_3	BeaconOrder_3	SuperFrameOrder_3
---------	------------------	---------------	-------------------

RouterCapacity_4	DeviceCapacity_4	Version_4	StackProfile_4	SecurityLevel_4
------------------	------------------	-----------	----------------	-----------------

PanID_4	LogicalChannel_4	BeaconOrder_4	SuperFrameOrder_4
---------	------------------	---------------	-------------------

Number is a 1 byte field that contains the number of networks found.

PanID\_x is a 2 byte field PAN Id of network found.

LogicalChannel\_x is a 1 byte field. Network channel.

BeaconOrder\_x is a 1 byte field Beacon Order found for network.

SuperFrameOrder\_x is a 1 byte field Super Frame Order of found network.

RouterCapacity\_x is a 1 byte field – Router capacity of found network (from beacon).

DeviceCapacity\_x is a 1 byte field – Device capacity for function and network (from beacon).

Version\_x is a 1 byte field - network version.

StackProfile\_x is a 1 byte field. Function and network's profile ID.

SecurityLevel\_x is a 1 byte field - Network's security level.

## 6.11 NLME\_STARTROUTER\_CONFIRMATION

### 6.11.1 Description

This message is sent by the target to the next higher layer of the results of its request to start a router.

### 6.11.2 Usage

The message format is as follows

Cmd=0x018F	Len=0x01	Status
------------	----------	--------

Status is a 1 byte field that indicates SUCCESS or FAILURE (see appendix for NLME status codes).



## 7. AF Interface

### 7.1 Introduction

This message interface enables the target AF to issue callbacks to the tester. Each message has a response to a message that is sent by the tester to the target. The response message only indicates successful reception of the callback message. These messages are subscription messages. Refer to SYS\_CALLBACK\_SUBSCRIBE for a description of the subscription mechanism.

### 7.2 AF\_INIT\_REQUEST

#### 7.2.1 Description

This command enables the tester to initialize the target AF.

#### 7.2.2 Usage

The message format is as follows.

Cmd = 0x0900	Len = 0x01
--------------	------------

### 7.3 AF\_REGISTER\_REQUEST

#### 7.3.1 Description

This command enables the tester to register an application's endpoint description.

#### 7.3.2 Usage

The message format is as follows.

Cmd = x0901	Len = var	Endpoint	AppProfileID	AppDeviceID	AppDeviceVersion	AppFlags
AppNumInClusters	AppOutClusterList	AppNumOutClusters	AppOutClusterList	Latency		

Endpoint is a byte field which specifies the endpoint of the device

AppProfileID is a 2 byte field which specifies the profile id of the application

AppDeviceID is a 2 byte field which specifies the device description id for this endpoint

AppDeviceVersion is a byte field which specifies the version of the device description

AppFlags is a byte field which specifies if a complex identifier is available

AppNumInClusters is a byte field which specifies the number of Input cluster Ids following in the AppInClusterList

AppInClusterList is a variable length field of size 'AppNumInClusters' and is the list of Input Cluster Ids.

AppNumOutClusters is a byte field which specifies the number of Output cluster Ids following in the AppOutClusterList

AppOutClusterList is a variable length field of size 'AppNumOutClusters' and is the list of Output Cluster Ids.

Latency is a byte field and has any of the following values 0x00 (no LatencyReqs), 0x01 (fastBeacons) and 0x02 (slowBeacons).

### 7.3.3 AF\_REGISTER\_RESPONSE

Cmd = 0x1901	Len = 0x01	Status
--------------	------------	--------

Status is a 1 byte field that indicates SUCCESS or FAILURE (see appendix for AF status codes).

## 7.4 AF\_SENDMSG\_REQUEST

### 7.4.1 Description

This command is used by tester to build and send a KVP or MSG message. For KVP messages, the code must be built with the following compiler directive: -DAF\_KVP\_SUPPORT=TRUE

### 7.4.2 Usage

The message format is the following:

Cmd = 0x0902	Len = var	FrameType	TxOptions	Radius		
DestAddr	DestEndpoint	SrcEndpoint	ClusterId	AddOrSend	TransId	
CmdType	AttribDataType	AttribId	ErrorCode	TransDataLen	TransData	

FrameType is a byte field which specifies either FRAMETYPE\_KVP or FRAMETYPE\_MSG.

TxOptions is a byte field which consists of the AF Tx Options bit fields; zero if none.

Radius is a byte field representing the number of hops allowed to deliver the message; usually use 7.

DestAddr is the 16-bit ZigBee network address for the message.

DestEndpoint is a byte field which specifies the destination endpoint of the message.

SrcEndpoint is a byte field which specifies the originator endpoint of the message.

ClusterId is a 2-byte field which specifies the cluster ID (only the LSB is used in V1.0 networks.)

AddOrSend is a byte field indicating to ADD\_MESSAGE or SEND\_MESSAGE; MSG set to zero.

TransId is a byte field which specifies the transaction sequence number of the message.

CmdType is a byte field which specifies the KVP command type of the message; MSG set to zero.

AttribDataType is a byte field which specifies the KVP data type of the message; MSG set to zero.

AttributeID is a 2-byte field which specifies the attribute ID; MSG set to zero.

ErrCode is a byte field which specifies the KVP error code, if any; MSG set to zero.

TransDataLen is a byte field which specifies the length of the TransactionData field.

TransData is a variable length field of size TransDataLength and is the transaction data frame.

#### 7.4.3 AF\_SENDMSG\_RESPONSE

Cmd = 0x1902	Len = 0x01	Status
--------------	------------	--------

Status is a 1 byte field that indicates SUCCESS or FAILURE (see appendix for AF status codes).

## 8. AF Callback Interface

### 8.1 AF\_INCOMINGDATA\_INDICATION

#### 8.1.1 Description

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

#### 8.1.2 Usage

The message format is the following:

Cmd = 0x0903	Len = var	FrameType	WasBroadcast	LinkQuality	SecurityUse
SrcAddr	SrcEndpoint	DestEndpoint	ClusterId	TransCnt	TransId
CmdType	AttribDataType	AttribId	ErrorCode	TransDataLen	TransData

FrameType is a byte field which specifies either FRAMETYPE\_KVP or FRAMETYPE\_MSG.

WasBroadcast is a byte field indicating whether the incoming message was broadcast.

LinkQuality is a byte field which indicates the link quality measured during reception.

SecurityUse is a 1 byte field which indicates the type of security used for the transaction.

SrcAddr is the 16-bit ZigBee network address of the source device sending the message.

SrcEndpoint is a byte field which specifies the source endpoint of the message.

DestEndpoint is a byte field which specifies the destination endpoint of the message.

ClusterId is a 2-byte field which specifies the cluster ID (only the LSB is used in V1.0 networks).

TransCnt is a byte field indicating the number of concatenated KVP messages in the transaction data; MSG set to 0.

TransId is a byte field which specifies the transaction sequence number of the message.

CmdType is a byte field which specifies the KVP command type of the message; MSG set to zero.

AttribDataType is a byte field which specifies the KVP data type of the message; MSG set to zero.

AttributeID is a 2-byte field which specifies the attribute ID; MSG set to zero.

ErrorCode is a byte field which specifies the KVP error code, if any; MSG set to zero.

TransDataLen is a byte field which specifies the length of the TransactionData field.

TransData is a variable length field of size TransDataLength and is the transaction data frame.

## 9. ZDO Interface

### 9.1 Introduction

These commands allow the tester to issue commands to the ZDO layer in the target and receive responses. Each of these messages has a corresponding message that is returned by the target. The response message only indicates that the command message was received and executed. The result of the command execution will be conveyed to the tester via a callback message interface (defined in the next section), if applicable.

### 9.2 ZDO\_AUTOAPPENDENDDEVICEBIND\_REQUEST

#### 9.2.1 Description

This command will tell the ZDO to request an End Device Bind. The ZDO will send the message to the ZigBee coordinator based on the information found in the AF Endpoint description of the passed in endpoint.

#### 9.2.2 Usage

The message format is as follows.

Cmd = 0x0A00	Len = 0x01	Endpoint
--------------	------------	----------

Endpoint – byte - Which endpoint to issue the End Device Bind request for.

### 9.2.3 ZDO\_AUTOAPPENDENDDEVICEBIND\_RESPONSE

Cmd = 0x1A00	Len = 0x01	Status
--------------	------------	--------

Status is a 1 byte field that indicates SUCCESS(0) or FAILURE.

### 9.3 ZDO\_AUTOFINDDESTINATION\_REQUEST

#### 9.3.1 Description

This function will issue a Match Description Request for the requested endpoint outputs. This message will generate a broadcast message.

#### 9.3.2 Usage

The message format is as follows.

Cmd = 0x0A01	Len = 0x01	Endpoint
--------------	------------	----------

Endpoint – byte - Which endpoint to issue the End Device Bind request for.

### 9.3.3 ZDO\_AUTOFINDDESTINATION\_RESPONSE

Cmd = 0x1A01	Len = 0x01	Status
--------------	------------	--------

Status is a 1 byte field that indicates SUCCESS(0) → FAILURE.

## 9.4 ZDO\_NETWORKADDRESS\_REQUEST

### 9.4.1 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. You must subscribe to “ZDO Network Address Response” to receive the response to this message. The response message listed below only indicates whether or not the message was received properly.

### 9.4.2 Usage

The message format is as follows.

Cmd = 0x0A02	Len = 0x09	IEEEAddr	ReqType	StartIndex	SecuritySuite
--------------	------------	----------	---------	------------	---------------

IEEEAddr - 8 bytes – 64 bit address bait.

ReqType – byte – following options:

Value	Description
0	Single device response
1	Extended – include associated devices

StartIndex – 1 byte – Starting index into the list of children. This is used to get more of the list if the list is too large for the message.

SecuritySuite - 1 byte –. Security options.<sup>i</sup>

### 9.4.3 ZDO\_NETWORKADDRESS\_RESPONSE

Cmd = 0x1A02	Len = 0x01	Status
--------------	------------	--------

Status is a 1 byte field that indicates SUCCESS → FAILURE (see appendix for APSME status codes).

## 9.5 ZDO\_IEEEADDRESS\_REQUEST

### 9.5.1 Description

This command will request a device's IEEE 64-bit address. You must subscribe to "ZDO IEEE Address Response" to receive the data response to this message. The response message listed below only indicates whether or not the message was received properly.

### 9.5.2 Usage

The message format is as follows.

Cmd = 0x0A03	Len = 0x04	ShortAddr	ReqType	StartIndex	SecuritySuite
--------------	------------	-----------	---------	------------	---------------

ShortAddr – 16 bits – short address of the destination device.

ReqType – byte – following options:

Value	Description
0	Single device response
1	Extended – include associated devices

StartIndex – 1 byte – Starting index into the list of children. This is used to get more of the list if the list is too large for the message.

SecuritySuite - 1 byte -. Security options.

### 9.5.3 ZDO\_IEEEADDRESS\_RESPONSE

Cmd = 0x1A03	Len = 0x01	Status
--------------	------------	--------

Status is a 1 byte field that indicates SUCCESS or FAILURE.

## 9.6 ZDO\_NODEDESCRIPTOR\_REQUEST

### 9.6.1 Description

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

### 9.6.2 Usage

The message format is as follows

Cmd = 0x0A04	Len = 5	DstAddr	NWKAddrOfInterest	SecuritySuite
--------------	---------	---------	-------------------	---------------

DstAddr – 16 bit –NWK address of the device generating the inquiry.

NWKAddrOfInterest – 16 bit - NWK address of the destination device being queried .

SecuritySuite - 1 byte -. Security Options.

### 9.6.3 ZDO\_NODEDESCRIPTOR\_RESPONSE

Cmd = 0x1A04	Len = 0x01	Status
--------------	------------	--------

Status is a 1 byte field that indicates SUCCESS → FAILURE.

## 9.7 ZDO\_POWERDESCRIPTOR\_REQUEST

### 9.7.1 Description

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

### 9.7.2 Usage

The message format is as follows

Cmd = 0x0A05	Len = 5	DstAddr	NWKAddrOfInterest	SecuritySuite
--------------	---------	---------	-------------------	---------------

DstAddr - 16 bit - NWK address of the device generating the inquiry.

NWKAddrOfInterest - 16 bit - NWK address of the destination device being queried.

SecuritySuite - 1 byte -. Security Options.

### 9.7.3 ZDO\_POWERDESCRIPTOR\_RESPONSE

Cmd = 0x1A05	Len = 0x01	Status
--------------	------------	--------

Status is a 1 byte field that indicates SUCCESS → FAILURE.

## 9.8 ZDO\_SIMPLEDESCRIPTOR\_REQUEST

### 9.8.1 Description

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

### 9.8.2 Usage

The message format is as follows

Cmd = 0x0A06	Len = 6	DstAddr	NWKAddrOfInterest	Endpoint	SecuritySuite
--------------	---------	---------	-------------------	----------	---------------

DstAddr - 16 bits - NWK address of the device generating the inquiry.

NWKAddrOfInterest - 16 bit - NWK address of the destination device being queried.



Endpoint – byte – represents the application endpoint the data is from.

SecuritySuite - 1 byte –. Security options.

### 9.8.3 ZDO\_SIMPLEDESCRIPTOR\_RESPONSE

Cmd = 0x1A00	Len = 0x01	Status
--------------	------------	--------

Status is a 1 byte field that indicates SUCCESS or FAILURE.

## 9.9 ZDO\_ACTIVEENDPOINT\_REQUEST

### 9.9.1 Description

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

### 9.9.2 Usage

The message format is as follows

Cmd = 0x0A07	Len = 5	DstAddr	NWKAddrOfInterest	SecuritySuite
--------------	---------	---------	-------------------	---------------

DstAddr – 16 bit – NWK address of the device generating the request..

NWKAddrOfInterest – 16 bit - NWK address of the destination device being queried.

SecuritySuite - 1 byte –. Security options.

### 9.9.3 ZDO\_ACTIVEENDPOINT\_RESPONSE

Cmd = 0x1A07	Len = 0x01	Status
--------------	------------	--------

Status is a 1 byte field that indicates SUCCESS or FAILURE.

## 9.10 ZDO\_MATCHDESCRIPTOR\_REQUEST

### 9.10.1 Description

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

### 9.10.2 Usage

The message format is as follows

Cmd = 0x0A08	Len = var	DstAddr	NWKAddrOfInterest	ProfileID
NumInClusters	InClusterList	NumOutClusters	OutClusterList	SecuritySuite

DstAddr – 16 bit – NWK address of the device generating the request..

NWKAddrOfInterest – 16 bit - NWK address of the destination device being queried.

ProfileID – 16 bit – Profile ID to match.

NumInClusters – byte – Number of ClusterIds in the InClusterList.

InClusterList – byte array – Array of input cluster IDs – (NumInClusters \* 2) long.

NumOutClusters – byte – Number of ClusterIds in the OutClusterList.

OutClusterList – byte array – Array of output cluster IDs – (NumOutClusters \* 2) long.

SecuritySuite - 1 byte -. Security options.

### 9.10.3 ZDO\_MATCHDESCRIPTOR\_RESPONSE

Cmd = 0x1A08	Len = 0x01	Status
--------------	------------	--------

Status is a 1 byte field that indicates SUCCESS or FAILURE.

## 9.11 ZDO\_COMPLEXDESCRIPTOR\_REQUEST

### 9.11.1 Description

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

### 9.11.2 Usage

The message format is as follows

Cmd = 0x0A09	Len = 5	DstAddr	NWKAddrOfInterest	SecuritySuite
--------------	---------	---------	-------------------	---------------

DstAddr – 16 bits – NWK address of the device generating the request .

NWKAddrOfInterest – 16 bit - NWK address of the destination device being queried.

SecuritySuite - 1 byte -. Security options.

### 9.11.3 ZDO\_COMPLEXDESCRIPTOR\_RESPONSE

Cmd = 0x1A09	Len = 0x01	Status
--------------	------------	--------

Status is a 1 byte field that indicates SUCCESS → FAILURE.

## 9.12 ZDO\_USERDESCRIPTOR\_REQUEST

### 9.12.1 Description

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

### 9.12.2 Usage

The message format is as follows

Cmd = 0x0A0A	Len = 5	DstAddr	NWKAddrOfInterest	SecuritySuite
--------------	---------	---------	-------------------	---------------

DstAddr – 16 bits – NWK address of the device generating the request.

NWKAddrOfInterest – 16 bit – NWK address of the destination device being queried.

SecuritySuite – 1 byte – Security options.

### 9.12.3 ZDO\_USERDESCRIPTOR\_RESPONSE

Cmd = 0x1A0A	Len = 0x01	Status
--------------	------------	--------

Status is a 1 byte field that indicates SUCCESS → FAILURE.

## 9.13 ZDO\_ENDDEVICEBIND\_REQUEST

### 9.13.1 Description

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

### 9.13.2 Usage

The message format is as follows

Cmd = 0x0A0B	Len = var	DstAddr	LocalCoordinator	Endpoint	ProfileID
NumInClusters	InClusterList	NumOutClusters	OutClusterList	SecuritySuite	

DstAddr – 16 bits – NWK address of the device generating the request.

LocalCoordinator – 16 bit – Local coordinator's short address. In the case of source binding, it's the short address of the source address.

Endpoint – byte – Device's Endpoint.

ProfileID – 16 bit – Profile ID to match.

NumInClusters – byte – Number of ClusterIds in the InClusterList.

InClusterList – byte array – Array of input cluster IDs – (NumInClusters \* 2) long.

NumOutClusters – byte – Number of ClusterIds in the OutClusterList.

OutClusterList – byte array – Array of output cluster IDs – (NumOutClusters \* 2) long.

SecuritySuite - 1 byte -. Security options.

### 9.13.3 ZDO\_ENDDEVICEBIND\_RESPONSE

Cmd = 0x1A0B	Len = 0x01	Status
--------------	------------	--------

Status is a 1 byte field that indicates SUCCESS or FAILURE.

## 9.14 ZDO\_BIND\_REQUEST

### 9.14.1 Description

This command is generated to request a Bind.

### 9.14.2 Usage

The message format is as follows

Cmd = 0x0A0C	Len = 24	DstAddr	SrcAddress	SrcEndpoint
ClusterID	DstAddrMode	DstAddress	DstEndpoint	SecuritySuite

DstAddr – 16 bits – destination address of the device generating the bind request

SrcAddress – 8 bytes – 64 bit Binding source IEEE address

SrcEndpoint – 8 bits – Binding source endpoint.

ClusterID – 2 byte – Cluster ID to match in messages.

DstAddrMode – 1 byte – Destination address mode: 01 for Group address; 03 for Extended address.

DstAddress – 8 bytes / 2bytes – Binding destination IEEE address. Not to be confused with DstAddr.

DstEndpoint – 8 bits / 0 byte – Binding destination endpoint. It is used only when DstAddrMode is 64 bits extended address

SecuritySuite - 1 byte -. Security options.

### 9.14.3 ZDO\_BIND\_RESPONSE

Cmd = 0x1A0C	Len = 0x01	Status
--------------	------------	--------

Status is a 1 byte field that indicates SUCCESS → FAILURE.

## 9.15 ZDO\_UNBIND\_REQUEST

### 9.15.1 Description

This command is generated to request an UnBind.

### 9.15.2 Usage

The message format is as follows

Cmd = 0x0A0D	Len = 24	DstAddr	SrcAddress	SrcEndPoint
ClusterID	DstAddrMode	DstAddress	DstEndPoint	SecuritySuite

DstAddr – 16 bits – destination address of the device generating the bind request.

SrcAddress – 8 bytes – 64 bit Binding source IEEE address

SrcEndPoint – 8 bits – Binding source endpoint.

ClusterID – 2 byte – Cluster ID to match in messages.

DstAddrMode – 1 byte – Destination address mode: 01 for Group address; 03 for Extended address.

DstAddress – 8 bytes – 64 bit Binding destination IEEE address. Not to be confused with DstAddr.

DstEndPoint – 8 bits – Binding destination endpoint.

SecuritySuite - 1 byte -. Security options.

### 9.15.3 ZDO\_UNBIND\_RESPONSE

Cmd = 0x1A0D	Len = 0x01	Status
--------------	------------	--------

Status is a 1 byte field that indicates SUCCESS → FAILURE.

## 9.16 ZDO\_MGMT\_NWKDISC\_REQUEST

### 9.16.1 Description

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

### 9.16.2 Usage

The message format is as follows

Cmd = 0x0A0E	Len = 0x8	DstAddr	ScanChannels	ScanDuration	StartIndex
--------------	-----------	---------	--------------	--------------	------------

DstAddr – 16 bits – network address of the device performing the discovery.

ScanChannels– 32 bits – Bit Mask for channels to scan

ScanDuration is a byte field represents the scanning time.

StartIndex – 8 bits – 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.

### 9.16.3 ZDO\_MGMT\_NWKDISC\_RESPONSE

Cmd = 0x1A0E	Len = 0x01	Status
--------------	------------	--------

Status is a 1 byte field that indicates SUCCESS or FAILURE.

## 9.17 ZDO\_MGMT\_LQI\_REQUEST

### 9.17.1 Description

This command is generated to request the destination device to perform a LQI query of other devices in the network.

### 9.17.2 Usage

The message format is as follows

Cmd = 0x0A0F	Len = 0x3	DstAddr	StartIndex
--------------	-----------	---------	------------

DstAddr – 16 bits – network address the device generating the query.

StartIndex – 8 bits – 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.

### 9.17.3 ZDO\_MGMT\_LQI\_RESPONSE

Cmd = 0x1A0F	Len = 0x01	Status
--------------	------------	--------

Status is a 1 byte field that indicates SUCCESS or FAILURE.

## 9.18 ZDO\_MGMT\_RTG\_REQUEST

### 9.18.1 Description

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

### 9.18.2 Usage

The message format is as follows

Cmd = 0x0A10	Len = 0x3	DstAddr	StartIndex
--------------	-----------	---------	------------

DstAddr – 16 bits – network address of the device being queried.

StartIndex – 8 bits – 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.

### 9.18.3 ZDO\_MGMT\_RTG\_RESPONSE

Cmd = 0x1A10	Len = 0x01	Status
--------------	------------	--------

Status is a 1 byte field that indicates SUCCESS or FAILURE.

## 9.19 ZDO\_MGMT\_BIND\_REQUEST

### 9.19.1 Description

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

### 9.19.2 Usage

The message format is as follows

Cmd = 0x0A11	Len = 0x3	DstAddr	StartIndex
--------------	-----------	---------	------------

DstAddr – 16 bits – network address of the device being queried.

StartIndex – 8 bits – 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.

### 9.19.3 ZDO\_MGMT\_BIND\_RESPONSE

Cmd = 0x1A11	Len = 0x01	Status
--------------	------------	--------

Status is a 1 byte field that indicates SUCCESS → FAILURE.

## 9.20 ZDO\_MGMT\_DIRECT\_JOIN\_REQUEST

### 9.20.1 Description

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

### 9.20.2 Usage

The message format is as follows

Cmd = 0x0A12	Len = 0xB	DstAddr	DeviceAddress	CapInfo
--------------	-----------	---------	---------------	---------

DstAddr – 16 bits – network address of the device to which the device specified in DeviceAddress is to join.

DeviceAddress – 8 bytes – The 64 bit IEEE Address of the device you want to be joined to the device at DstAddr.

CapInfo – 8 bits – The operating capabilities of the device being directly joined. Bit weighted values follow:

Can you please verify if the table below is accurate? I understand that for a router, the value should be 0x8E, and for an ED, it's 0x80, and I'm not sure if these give with this table?

Bit	Description
0	Alternate PAN Coordinator
1	Device type: 1 – ZigBee Router 0 – End Device
2	Power Source: 1 – Mains powered
3	Receiver on when idle
4-5	Reserved
6	Security capability
7	Reserved

### 9.20.3 ZDO\_MGMT\_DIRECT\_JOIN\_RESPONSE

Cmd = 0x1A12	Len = 0x01	Status
--------------	------------	--------

Status is a 1 byte field that indicates SUCCESS → FAILURE.

## 9.21 ZDO\_USERDESCRIPTOR\_SET\_REQUEST



### 9.21.1 Description

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

### 9.21.2 Usage

The message format is as follows

Cmd = 0x0A13	Len = var	DstAddr	NWKAddrOfInterest	DescLen	Descriptor	SecuritySuite
--------------	-----------	---------	-------------------	---------	------------	---------------

DstAddr – 16 bits – network address of the device generating the set request.

NWKAddrOfInterest – 16 bit - NWK address of the device targeted for the set request.

DescLen – 8 bits – Length, in bytes, of the user descriptor.

Descriptor – byte array – User descriptor array (can be up to 16 bytes).

SecuritySuite - 1 byte – Security options.

### 9.21.3 ZDO\_USERDESCRIPTOR\_SET\_RESPONSE

Cmd = 0x1A13	Len = 0x01	Status
--------------	------------	--------

Status is a 1 byte field that indicates SUCCESS or FAILURE.

## 9.22 ZDO\_END\_DEVICE\_ANNCE

### 9.22.1 Description

This command is generated to request an End Device Announcement.

### 9.22.2 Usage

The message format is as follows

Cmd = 0x0A14	Len = 0xC	DevAddr	DeviceAddress	SecuritySuite	Capabilities
--------------	-----------	---------	---------------	---------------	--------------

DevAddr – 16 bits – network address of the device generating the request.

DeviceAddress – 8 bytes – The 64 bit IEEE Address of the device being announced.

SecuritySuite - 1 byte – Security options.

Capabilities – 1 byte – MAC capabilities – see capabilities field of ZDO\_MGMT\_DIRECT\_JOIN\_REQUEST for bit definitions.

## 9.23 ZDO\_MGMT\_LEAVE\_REQUEST

### 9.23.1 Description

This command is generated to request a Management Leave Request from the target device.

### 9.23.2 Usage

The message format is as follows

Cmd = 0x0A15	Len = 0xA	DstAddr	DeviceAddress
--------------	-----------	---------	---------------

DstAddr – 16 bits – network address of the device generating the request.

DeviceAddress – 8 bytes – The 48 bit IEEE Address of the target device you want to leave.

### 9.23.3 ZDO\_MGMT\_LEAVE\_RESPONSE

Cmd = 0x1A15	Len = 0x01	Status
--------------	------------	--------

Status is a 1 byte field that indicates SUCCESS or FAILURE.

## 9.24 ZDO\_MGMT\_PERMIT\_JOIN\_REQUEST

### 9.24.1 Description

This command is generated to set the Permit Join from the destination device.

### 9.24.2 Usage

The message format is as follows

Cmd = 0x0A16	Len = 0x4	DstAddr	Duration	TC Significance
--------------	-----------	---------	----------	-----------------

DstAddr – 16 bits – network address of the destination device where Permit Join information is to be modified.

Duration – 1 byte – The duration to permit joining. 0 = join disabled. 0xff = join enabled. 0x01-0xfe = number of seconds to permit joining.

TC Significance – 1 byte - Trust Center Significance.

### 9.24.3 ZDO\_MGMT\_PERMIT\_JOIN\_RESPONSE

Cmd = 0x1A16	Len = 0x01	Status
--------------	------------	--------

Status is a 1 byte field that indicates SUCCESS or FAILURE.

## 9.25 ZDO\_SYSTEM\_SERVER\_DISC\_REQUEST

### 9.25.1 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 in this request is 'broadcast to all RxOnWhenIdle devices'.

### 9.25.2 Usage

The message format is as follows

Cmd = 0x0A17	Len = 0x03	Server Mask	Security Suite
--------------	------------	-------------	----------------

Server Mask is a two-byte field with bit settings signifying the system server capabilities of the device.

Security Suite: one byte security option.

### 9.25.3 ZDO\_SYSTEM\_SERVICE\_DISC\_RESPONSE

Cmd = 0x1A17	Len = 0x01	Status
--------------	------------	--------

Status is a 1 byte field that indicates SUCCESS or FAILURE.

## 9.26 ZDO\_NETWORK\_START\_REQUEST

### 9.26.1 Description

In the case where compiler flag HOLD\_AUTO\_START is defined by default, device will start from HOLD state. Issuing this command will trigger the device to leave HOLD state to form or join a network.

### 9.26.2 Usage

The command message format is as follows

Cmd=0x0A18	Len=0x00
------------	----------

There is no parameter for this command.

### 9.26.3 SYS\_NETWORK\_START\_RESPONSE

Cmd=0x1A18	Len=0x01	Status
------------	----------	--------

Status is a 1 byte field that will be SUCCESS(0) or FAILURE(1).

## 10. ZDO Callback Interface

### 10.1 Introduction

These commands allow the tester to issue commands to the ZDO layer in the target and receive responses. Each of these messages has a corresponding message that is returned by the target. The response message only indicates that the command message was received and executed. The result of the command execution will be conveyed to the tester via a callback message interface (defined in the next section), if applicable. These messages are subscription messages. Refer to SYS\_CALLBACK\_SUBSCRIBE for a description of the subscription mechanism.

### 10.2 ZDO\_NETWORKADDRESS\_RESPONSECB

#### 10.2.1 Description

This callback message is in response to the ZDO Network Address Request.

#### 10.2.2 Usage

The message format is as follows

Cmd = 0x0A80	Len = var	SrcAddrMode	SrcAddr	Status	IEEEAddr
nwkAddr	NumAssocDev	StartIndex	AssocDevList		

SrcAddrMode – byte – indicates that the SrcAddr is either 16 bits (2) or 64 bits (3)

SrcAddr – 8 bytes or 16 bits – Source address, size is dependent on SrcAddrMode

Status – byte – this field indicates either SUCCESS or FAILURE.

IEEEAddr – 8 bytes – 64 bit IEEE address of source device

nwkAddr – 16 bit – short network address of responding device

NumAssocDev – byte – the number of associated devices

StartIndex – byte - Starting index into the list of associated devices for this report.

AssocDevList – array of 16 bit short addresses – list of network address of 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.

### 10.3 ZDO\_IEEEADDRESS\_RESPONSECB

#### 10.3.1 Description

This callback message is in response to the ZDO IEEE Address Request.

#### 10.3.2 Usage

The message format is as follows

Cmd = 0x0A81	Len = var	SrcAddrMode	SrcAddr	Status	IEEEAddr
NumAssocDev	StartIndex	AssocDevList			

SrcAddrMode – byte – indicates that the SrcAddr is either 16 bits (2) or 64 bits (3)

SrcAddr – 8 bytes or 64 bits – Source address, size is dependent on SrcAddrMode

Status – byte – this field indicates either SUCCESS or FAILURE.

IEEEAddr – 8 bytes – 64 bit IEEE address of source device

NumAssocDev – byte – the number of associated devices

StartIndex – byte - Starting index into the list of associated devices for this report.

AssocDevList – array of 16 bit short addresses – list of network address of 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.

## 10.4 ZDO\_NODEDESCRIPTOR\_RESPONSECB

### 10.4.1 Description

This callback message is in response to the ZDO Node Descriptor Request.

### 10.4.2 Usage

The message format is as follows

Cmd = 0x0A82	Len = 18	Status	SrcAddr	NWKAddrOfInterest		NodeType		ComplexDescriptor Available	
UserDescriptor Available	APSFlags	Frequency Band	Capabilities	Manufacturer Code		Buffer Size	Transfer Size	Server Mask	

Status – byte – this field indicates either SUCCESS or FAILURE.

SrcAddr – 16 bit – the message's source network address.

NWKAddrOfInterest – 16 bits – Device's short address of this Node descriptor

NodeType – 8 bits – defined as follows:

Bits	Name	Description
7-5	NodeLogicalType	Identifies Node as ZigBee Coordinator or other ZigBee device type
4-0	N/A	Reserved

Node Logical Type	
Type	Value
ZigBee Coordinator	0
ZigBee Router	1
ZigBee End Device	2
Reserved	3-7

Cmd = 0x0A83	Len = 9	Status	SrcAddr	NWKAddrOfInterest	NdeCurrentPowerMode
NdeAvailablePowerSources		NdeCurrentPowerSource	NdeCurrentPowerSourceLevel		

Status – byte – this field indicates either SUCCESS or FAILURE.

SrcAddr – 16 bit – the message's source network address.

NWKAddrOfInterest – 16 bits – Device's short address that this response describes.

NdeCurrentPowerMode – 8 bits – Indicates Power Mode

NdeCurrentPowerMode	
Description	Value
Receiver permanently on or periodically on	0
Receiver periodically comes on	1
Receiver comes on when stimulated	2
Reserved	0x3 – 0xf

NdeAvailablePowerSources – 8 bits – Bit field showing available power sources

NdeAvailablePowerSources	
Description	Value
Constant (Mains) power	0
Rechargeable battery	1
Disposable battery	2
Reserved	0x3 – 0xf

NdeCurrentPowerSource – byte – Indicates current power source. Same values as NdeAvailablePowerSources

NdeCurrentPowerSourceLevel – byte – Indicates charge status of power supply

## 10.6 ZDO\_SIMPLEDESCRIPTOR\_RESPONSECB

### 10.6.1 Description

This callback message is in response to the ZDO Simple Descriptor Request.

### 10.6.2 Usage

The message format is as follows

Cmd = 0x0A84	Len = var	Status	SrcAddr	NWKAddrOfInterest	Endpoint	AppProfileID
AppDevID	AppDevVer	AppFlags	AppInClusterCount		AppInClusterList	
AppOutClusterCount		AppOutClusterList				

Status – byte – this field indicates either SUCCESS or FAILURE.

SrcAddr – 16 bit – the message's source network address.

NWKAddrOfInterest – 16 bits – Device's short address that this response describes.

Endpoint – 8 bits

AppProfileID – 16 bits – The profile ID for this endpoint.

AppDevID – 16 bits – The Device Descriptor ID for this endpoint.

AppDevVer – 8 bits – Defined as the following format

AppDevVer	
Descriptor	Value
Version 1.00	0
Reserved	0x1 – 0xf

AppFlags – 8 bits – Defined as the following format

AppFlags	
Descriptor	bit
ComplexDescriptorAvailableFlag	0
Reserved	1
Reserved	2
Reserved	3

AppInClusterCount – byte – The number of input clusters in the AppInClusterList:



AppInClusterList – byte array – List of input cluster IDs supported

AppOutClusterCount – byte – The number of output clusters in the AppOutClusterList:

AppOutClusterList – byte array – List of output cluster IDs supported

## 10.7 ZDO\_ACTIVEENDPOINT\_RESPONSECB

### 10.7.1 Description

This callback message is in response to the ZDO Active Endpoint Request.

### 10.7.2 Usage

The message format is as follows

Cmd = 0x0A85	Len = var	Status	SrcAddr	NWKAddrOfInterest	ActiveEndpointCount	ActiveEndpointList
--------------	-----------	--------	---------	-------------------	---------------------	--------------------

Status – byte – this field indicates either SUCCESS or FAILURE.

SrcAddr – 16 bit – the message's source network address.

NWKAddrOfInterest – 16 bits – Device's short address that this response describes.

ActiveEndpointCount – 8 bits – Number of active endpoint in the list

ActiveEndpointList – byte array – Array of active endpoints on this device.

## 10.8 ZDO\_MATCHDESCRIPTOR\_RESPONSECB

### 10.8.1 Description

This callback message is in response to the ZDO Match Descriptor Request.

### 10.8.2 Usage

The message format is as follows

Cmd = 0x0A86	Len = var	Status	SrcAddr	NWKAddrOfInterest	MatchCount	MatchEPList
--------------	-----------	--------	---------	-------------------	------------	-------------

Status – byte – this field indicates either SUCCESS or FAILURE.

SrcAddr – 16 bit – the message's source network address.

NWKAddrOfInterest – 16 bits – Device's short address that this response describes.

MatchCount – 8 bits – Number of active endpoints in the list

MatchEndpointList – byte array – Array of active endpoints on this device.

## 10.9 ZDO\_USERDESCRIPTOR\_RESPONSECB

### 10.9.1 Description

This callback message is in response to the ZDO User Descriptor Request.

### 10.9.2 Usage

The message format is as follows

Cmd = 0x0A8F	Len = var	Status	SrcAddr	NWKAddrOfInterest	DescLen	Descriptor
--------------	-----------	--------	---------	-------------------	---------	------------

Status – byte – this field indicates either SUCCESS or FAILURE.

SrcAddr – 16 bit – the message's source network address.

NWKAddrOfInterest – 16 bits – Device's short address that this response describes.

DescLen – 8 bits – Length, in bytes, of the user descriptor

Descriptor – byte array – User descriptor array (can be up to 128 bytes).

## 10.10 ZDO\_ENDDEVICEBIND\_RESPONSECB

### 10.10.1 Description

This callback message is in response to the ZDO End Device Bind Request.

### 10.10.2 Usage

The message format is as follows

Cmd = 0x0A87	Len = 3	Status	SrcAddr
--------------	---------	--------	---------

Status – byte – this field indicates status of the bind request with the following values:

Status	
Descriptor	Value
SUCCESS	0
NOT SUPPORTED	1
TIMEOUT	2
NO MATCH	3
Reserved	0x04 – 0xFF

SrcAddr – 16 bit – the message's source network address.

## 10.11 ZDO\_BIND\_RESPONSECB

### 10.11.1 Description

This callback message is in response to the ZDO Bind Request.

### 10.11.2 Usage

The message format is as follows:

Cmd = 0x0A88	Len = 1	Status	SrcAddr
--------------	---------	--------	---------

Status – byte – this field indicates status of the bind request with the following values:

Status	
Description	Value
SUCCESS	0
NOT SUPPORTED	1
TABLE FULL	2
Reserved	0x03 – 0xFF

SrcAddr – 16 bit – the message's source network address.

## 10.12 ZDO\_UNBIND\_RESPONSECB

### 10.12.1 Description

This callback message is in response to the ZDO UnBind Request.

### 10.12.2 Usage

The message format is as follows

Cmd = 0x0A89	Len = 1	Status	SrcAddr
--------------	---------	--------	---------

Status – byte – this field indicates status of the bind request with the following values:

Status	
Description	Value
SUCCESS	0
NOT SUPPORTED	1

NO ENTRY	2
Reserved	0x03 – 0xFF



RtgCount – 1 byte – Total number of entries available in the device.

StartIndex – 1 byte – Where in the total number of entries this response starts.

RtgListCount – 1 byte – Number of entries in this response.

RtgList – list – an array of RtgList items. RtgListCount contains the number of items in this table.

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

## **10.16 ZDO\_MGMT\_BIND\_RESPONSECB**

### **10.16.1 Description**

This callback message is in response to the ZDO Management Binding Table Request.

### **10.16.2 Usage**

The message format is as follows

### 10.17.2 Usage

The message format is as follows

Cmd = 0x0A8E	Len = 0x03	SrcAddr	Status
-----------------	---------------	---------	--------

SrcAddr – 16 bit – Source address of the message

Status – byte – this field indicates either SUCCESS (0) or FAILURE (1).

## 10.18 ZDO\_USERDESCRIPTOR\_SET\_RESPONSECB

### 10.18.1 Description

This callback message is in response to the ZDO User Descriptor Set Request.

### 10.18.2 Usage

The message format is as follows

Cmd = 0x0A90	Len = 0x03	Status	SrcAddr
-----------------	---------------	--------	---------

Status – byte – this field indicates status of the request with the following values:

SrcAddr – 16 bit – Source address of the message

Status	
Descriptor	Value
SUCCESS	0
NOT SUPPORTED	1
Reserved	0x04 – 0xFF

## 10.19 ZDO\_MGMT\_LEAVE\_RESPONSECB

### 10.19.1 Description

This callback message is in response to the ZDO Management Leave Request.

### 10.19.2 Usage

The message format is as follows



Cmd = 0x0A91	Len = 0x03	Status	SrcAddr
-----------------	---------------	--------	---------

Status – byte – this field indicates either SUCCESS (0) or FAILURE (1).

SrcAddr – 16 bit – Source address of the message

## 10.20 ZDO\_MGMT\_PERMIT\_JOIN\_RESPONSECB

### 10.20.1 Description

This callback message is in response to the ZDO Management Permit Join Request.

### 10.20.2 Usage

The message format is as follows

Cmd = 0x0A92	Len = 0x03	Status	SrcAddr
-----------------	---------------	--------	---------

Status – byte – this field indicates either SUCCESS (0) or FAILURE (1).

SrcAddr – 16 bit – Source address of the message

## 10.21 ZDO\_SYSTEM\_SERVER\_DISC\_RESPONSECB

### 10.21.1 Description

This callback message is in response to the ZDO System Service Discovery Request. 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.

### 10.21.2 Usage

The message format is as follows

Cmd = 0x0A93	Len = 0x03	Status	SrcAddr	Server Mask
--------------	------------	--------	---------	-------------

Status – 1 byte – this field indicates either SUCCESS (0) or FAILURE (1).

SrcAddr – 16 bit – Source address of the address.

Server Mask – 16 bits – each bit signifies the 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

## 11. Application Interface

### 11.1 Introduction

This interface allows the tester to issue application-specific commands to the target and receive responses. The actual commands and responses will depend on the application and will be decided later.

### 11.2 USERTEST\_REQUEST

#### 11.2.1 Description

This message is sent to the target in order to test the functions defined for individual applications (which internally use attributes and cluster IDs from various device descriptions).

#### 11.2.2 Usage

The message format is as follows

Cmd=0x0B51	Len=0x08	Endpoint	Command	Param1	Param2
------------	----------	----------	---------	--------	--------

Endpoint is a byte field that indicates the endpoint of the application

Command is a two-byte field that is used to identify the application command to invoke

Param1 is a two-byte field that is used as the first parameter of the

Param2 is a two-byte field which is used as the second parameter of the command (usually an attribute value). This value is not necessary for some commands.

#### 11.2.3 USERTEST\_RESPONSE

Cmd=0x1B51	Len=0x01	Status
------------	----------	--------

Status is a 1 byte field that indicates SUCCESS or FAILURE (see appendix for failure codes).

## Appendix A Status Codes and Query types

### MAC Callback ID codes

Callback Name	Callback ID
SPI_CB_NWK_SYNC_LOSS_IND	0x2080
SPI_CB_NWK_ASSOCIATE_IND	0x2081
SPI_CB_NWK_ASSOCIATE_CNF	0x2082
SPI_CB_NWK_BEACON_NOTIFY_IND	0x2083
SPI_CB_NWK_DATA_CNF	0x2084
SPI_CB_NWK_DATA_IND	0x2085
SPI_CB_NWK_DISASSOCIATE_IND	0x2086
SPI_CB_NWK_DISASSOCIATE_CNF	0x2087
SPI_CB_NWK_GTS_CNF	0x2088
SPI_CB_NWK_GTS_IND	0x2089
SPI_CB_NWK_ORPHAN_IND	0x208a
SPI_CB_NWK_POLL_CNF	0x208b
SPI_CB_NWK_SCAN_CNF	0x208c
SPI_CB_NWK_SECURITY_ERR_IND	0x208d
SPI_CB_NWK_START_CNF	0x208e

### NWK Callback ID codes

Callback Name	Callback ID
SPI_CB_NLDE_DATA_CNF	0x0180
SPI_CB_NLDE_DATA_IND	0x0181
SPI_CB_NLME_INITCOORD_CNF	0x0182
SPI_CB_NLME_JOIN_CNF	0x0183
SPI_CB_NLME_JOIN_IND	0x0184
SPI_CB_NLME_LEAVE_CNF	0x0185
SPI_CB_NLME_LEAVE_IND	0x0186
SPI_CB_NLME_SYNC_IND	0x0189
SPI_CB_NLME_NWK_DISC_CNF	0x018D
SPI_CB_NLME_START_ROUTER_CNF	0x018F

**AF Callback ID codes**

Callback Name	Callback ID
SPI_CB_AF_DATA_IND	0x0903

**ZDO Callback ID codes**

Callback Name	Callback ID
SPI_CB_ZDO_NWK_ADDR_RSP	0x0A80
SPI_CB_ZDO_IEEE_ADDR_RSP	0x0A81
SPI_CB_ZDO_NODE_DESC_RSP	0x0A82
SPI_CB_ZDO_POWER_DESC_RSP	0x0A83
SPI_CB_ZDO_SIMPLE_DESC_RSP	0x0A84
SPI_CB_ZDO_ACTIVE_EPINT_RSP	0x0A85
SPI_CB_ZDO_MATCH_DESC_RSP	0x0A86
SPI_CB_ZDO_END_DEVICE_BIND_RSP	0x0A87
SPI_CB_ZDO_BIND_RSP	0x0A88
SPI_CB_ZDO_UNBIND_RSP	0x0A89
SPI_CB_ZDO_MGMT_NWKDISC_RSP	0x0A8A
SPI_CB_ZDO_MGMT_LQI_RSP	0x0A8B
SPI_CB_ZDO_MGMT_RTG_RSP	0x0A8C
SPI_CB_ZDO_MGMT_BIND_RSP	0x0A8D
SPI_CB_ZDO_MGMT_DIRECT_JOIN_RSP	0x0A8E
SPI_CB_ZDO_USER_DESC_RSP	0x0A8F

**NLME Status values**

STATUS	VALUE
NLME_SUCCESS	0x00
NLME_FAILURE	0x01
NLME_ILLEGAL_REQUEST	0x02
NLME_UNKNOWN_DEVICE	0x03
NLME_STARTUP_FAILURE	0x04
NLME_NO_SHORT_ADDRESS	0x05
NLME_UNAVAILABLE_KEY	0x06
NLME_FRAME_TOO_LONG	0x07
NLME_FAILED_SECURITY_CHECK	0x08
NLME_INVALID_PARAMETER	0x09
NLME_UNSUPPORTED_ATTRIBUTE	0x0A
NLME_NOT_JOINED	0x0B
NLME_ALREADY_JOINED	0x0C
NLME_NOT_POSSIBLE	0x0D
NLME_TX_FAIL	0x0E
NLME_TX_ERROR	0x0F
NLME_RX_SUCCESS	0x10
NLME_RX_FAILURE	0x11
NLME_ALREADY_PRESENT	0x12
NLME_NO_NETWORKS	0x13
NLME_TABLE_FULL	0x14

**NLME RX\_STATE values**

RX_STATE	VALUE
NLME_RX_CHECK	0x00
NLME_RX_DISABLE	0x01
NLME_RX_ENABLE	0x02
NLME_RX_TRACK	0x03

**NLDE Status values**

STATUS	VALUE
NLDE_SUCCESS	0x00
NLDE_INVALID_TARGET	0x01
NLDE_INVALID_PARAMETER	0x02
NLDE_BUFFER_FULL	0x03
NLDE_MEMORY_ERROR	0x04
NLDE_INVALID_NETWORK_STATE	0x05
NLDE_FAILURE	0x06

**AF Status values**

STATUS	VALUE
afStatus_SUCCESS	0x00
afStatus_FAILED	0x01
afStatus_MEM_FAIL	0x02
afStatus_INVALID_PARAMETER	0x03

**Discovery: Query types and values**

QUERY TYPE	QUERY VALUE
NWK_ADDR_LIST	NULL
IEEE_ADDR_LIST	NULL
SIMPLE_DESC	Endpoint
NODE_DESC	NULL

### Transaction Codes

TRANSACTION CODE	VALUE
TRANSACTIONCODE_OK	0x00
TRANSACTIONCODE_INVALID_ENDPOINT	0x01
TRANSACTIONCODE_INVALID_INTERFACE	0x02
TRANSACTIONCODE_INVALID_ATTRIBUTEID	0x03
TRANSACTIONCODE_INVALID_TRANSTYPE	0x04
TRANSACTIONCODE_INVALID_TRANSDATALENGTH	0x05
TRANSACTIONCODE_INVALID_TRANSDATA	0x06

---

<sup>i</sup> Currently, it is not supported in Z-Stack to enable security for each single ZDO command. The parameter “In order to turn security on, users need to set SECURITY tool in f8wCnfig.cfg. Then devices will form the network with security on and all over the air message are encrypted and authenticated.