Z-Home – Zigbee

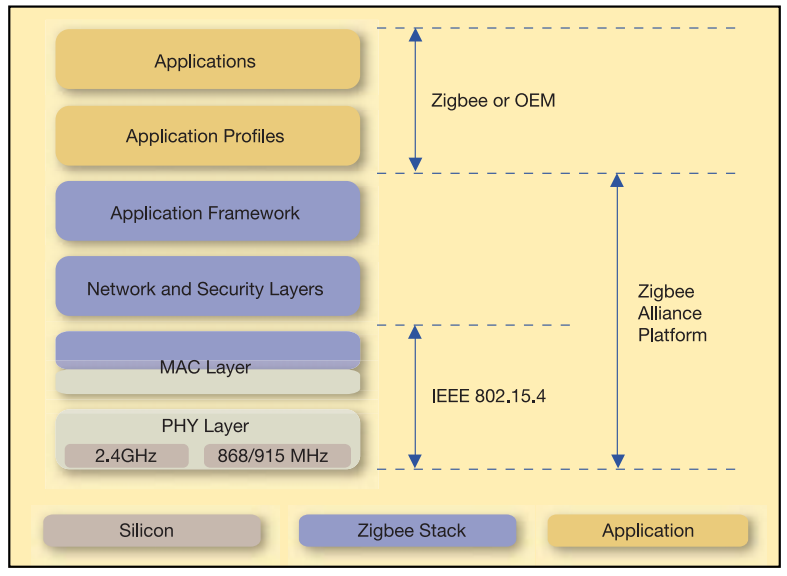
Technical Functional Specification

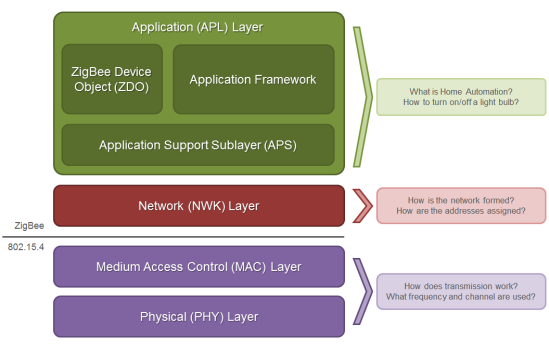
DRAFT

Author: Andrea Ravalli

Date: 6/11/2016

Revision: 0.9





# Definitions

Below are reported the definitions used in this document.

## Addresses

* 64bit address BROADCAST:  
  0x000000000000FFFF
* 16bit address UNKNOWN (BROADCAST):  
  0xFFFE
* DEV\_ADDR\_64: 64 bit address for a known device (already added to the associated device map):  
  0xXXXXXXXXXXXXXXXX
* DEV\_ADDR\_16: 16 bit address for a known device (already added to the associated device map):

0xYYYY

## Other definitions

* Unspecified values or to be calculated are marked by dashes:  
  1 byte to be calculated ‘0x—‘

## Frame Types

### TX\_EXPLICIT

Il *Explicit Addressing Command frame* sarà indicato brevemente come ***tx\_explicit.***

Esso permette di inviare frame di comando esplicitantando anche gli indirizzi di destinazione. Richiede API Output (AO) mode settato ad 1 o superiore.

Per dettagli vedere pag 161 di Xbee\_S2C\_manual\_90002002.pdf.

### RX\_EXPLICIT

Il *Explicit Rx Indicator frame* sarà indicato brevemente come ***rx\_explicit.***

Rappresenta la risposta ad un frame di comando di tipo ***tx\_explicit*** con ***AO>=1***.

Per dettagli vedere pag 176 di Xbee\_S2C\_manual\_90002002.pdf

# Xbee module configuration

## Settings

Zigbee Coordinator (CE) 1

Device Join Permit (JN) FF

Zigbee Stack Profile (ZS) 2

Encryption Enable (EE) 1

Encryption Options (EO) 0

Encryption Key (KY) 5a6967426565416c6c69616e63653039

Network Encryption Key (NK) 0

API Enable (AP) 1

API Output Mode (AO) 3

Sleep Mode (SM) 1

## AT command sequence

Below is reported the list of AT commands ordered to avoid command conflicts/device inconsistent states.

The string ‘ERROR’ is reported on the serial output in case of write failures.

The string ‘OK’ is reported …

+++

ATSM 1

ATCE 1

ATJN FF

ATZS 2

ATEE 1

ATEO 0

ATKY 5a6967426565416c6c69616e63653039

ATNK 0

ATAP 1

ATAO 3

ATWR

ATFR

# General Concepts

## Xbee explicit frames and APS frames

The Xbee firmware will abstract all the Zigbee protocol levels until the APS.



APS frames are encapsulated inside the Xbee explicit frame in the “payload” field. This field is sent as is over the air and since that has to written in “little endian” byte order.



## Reading ZDO message

In order to read ZDO messages the Xbee AO (API Output mode) needs to be set to 3.

ZDO messages needs to be formatted as APS frames including the APS header (frame type and transaction id included). Detail about the APS frames can be found in ZigbeeSpec  cap 2.4.2.8.

# Network Discovery Frames

This chapter will list all frame useful to the construction of the coordinator’s map for the devices belonging to his network.

## Route Record Request

* **Cluster ID: 0x0032**
* **Profile ID: 0x0000**

### Description

Il coordinator al suo avvio controlla quali dispositivi appartengono alla sua rete chiedendo i il contenuto della loro routing table ai dispositivi connessi.

Questa azione può (deve?) essere usata per costrure la tabella dei dispositivi connessi.

Per dettagli vedere pag 12 di APP\_NOTE\_XBee\_ZigBee\_Device\_Profile.pdf

### Pseudo code

zb.send('tx\_explicit',

dest\_addr\_long = BROADCAST,

dest\_addr = UNKNOWN,

src\_endpoint = '\x00',

dest\_endpoint = '\x00',

cluster = '\x00\x32',

profile = '\x00\x00',

data = '\x12'+'\x01'

)

### Frame

Frame details:

|  |  |  |
| --- | --- | --- |
| Frame data fields | Offset | Value |
| Start delimiter | 0 | 0x7e |
| Length | MSB 1 - LSB 2 | 0x0015 |
| Frame Type | 3 | 0x11 |
| Frame Seq ID | 4 | 0xAA |
| 64-bit dest addr | MSB 5 - LSB 12 | BROADCAST |
| 16-bit dest addr | MSB 13 - LSB 14 | UNKNOWN |
| src end point | 15 | 0x00 |
| dest end point | 16 | 0x00 |
| cluster ID | 17-18 | 0x0032 |
| Profile ID | 19-20 | 0x0000 |
| broadcast radius | 21 | 0x00 |
| tx options | 22 | 0x00 |
| Data Payload | 23-24 | 0x12 0x01 |
| Checksum | 25 | -- |

Payload Data description

|  |  |  |
| --- | --- | --- |
| data fields | Offset | Example |
| Transaction ID | 0 | 0x12 |
| Start Index | 1 | 0x01 |

Hexadecimal representation:

0x76001511AA000000000000FFFFFFFE00000032000000001201--

## Route Record Response

* **Cluster ID: 0x8032**
* **Profile ID: 0x0000**

### Description

Risposta da parte di un dispositivo remoto ad una Route Record Request, il cui payload riporta la tabella di rouoting del dispositivo remoto stesso.

Il dispositivo sorgente di questo messaggio come quelli riportati nella sua routing table devono essere aggiunti alla mappa.

Per dettagli vedere pag 12 di APP\_NOTE\_XBee\_ZigBee\_Device\_Profile.pdf

### Pseudo code

N/A.

### Frame

Frame details:

|  |  |  |
| --- | --- | --- |
| Frame data fields | Offset | Value |
| Start delimiter | 0 | 0x7e |
| Length | MSB 1 - LSB 2 | 0x00-- |
| Frame Type | 3 | 0x91 |
| 64-bit dest addr | MSB 4 - LSB 11 | DEV\_ADDR\_64 |
| 16-bit dest addr | MSB 12 - LSB 13 | DEV\_ADDR\_16 |
| src end point | 14 | 0x00 |
| dest end point | 15 | 0x00 |
| cluster ID | 16-17 | 0x8032 |
| Profile ID | 18-19 | 0x0000 |
| rx options | 20 | 0x-- |
| Data Payload | 21-n | -- |
| Checksum | n+1 | -- |

Data Payload description:

|  |  |  |
| --- | --- | --- |
| data fields | Offset | Example |
| Transaction ID | 0 | 0x12 |
| Routing Table | 1 | 0x01 |

Routing Table:

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Size (bytes)** | **Description** |
| Routing Table Entries | 1 |  |
| Routing Table Entries | 1 | The total number of routing table entries |
| Start Index | 1 | The starting point in the routing table |
| Routing Table List Count | 1 | The number of routing table entries in this response |
| Routing Table List | Variable | A list of routing table entries. |

For the Routing Table List details see p.12 di APP\_NOTE\_XBee\_ZigBee\_Device\_Profile.pdf.

Hexadecimal representation: N/A

## Active Endpoint Request

* **Cluster ID: 0x0005**
* **Profile ID: 0x0000**

### Description

Transmission used to discover the active endpoints on a device with a matching 16-bit address.

Questa azione deve essere usata per costrure la mappa degli end\_point disponibili per ciascuno dei dispositivi connessi.

Per dettagli vedere pag 7 di APP\_NOTE\_XBee\_ZigBee\_Device\_Profile.pdf

### Pseudo code

zb.send('tx\_explicit',

dest\_addr\_long = DEV\_ADDR\_64,

dest\_addr = DEV\_ADDR\_16,

src\_endpoint = '\x00',

dest\_endpoint = '\x00',

cluster = '\x00\x05',

profile = '\x00\x00',

data = DEV\_ADDR\_16 [1]+ DEV\_ADDR\_16 [0]

)

### Frame

Frame details:

|  |  |  |
| --- | --- | --- |
| Frame data fields | Offset | Value |
| Start delimiter | 0 | 0x7e |
| Length | MSB 1 - LSB 2 | 0x0015 |
| Frame Type | 3 | 0x11 |
| Frame Seq ID | 4 | 0x-- |
| 64-bit dest addr | MSB 5 - LSB 12 | DEV\_ADDR\_64 |
| 16-bit dest addr | MSB 13 - LSB 14 | DEV\_ADDR\_16 |
| src end point | 15 | 0x00 |
| dest end point | 16 | 0x00 |
| cluster ID | 17-18 | 0x0005 |
| Profile ID | 19-20 | 0x0000 |
| broadcast radius | 21 | 0x00 |
| tx options | 22 | 0x00 |
| Data Payload | MSB 23 - LSB 24 | DEV\_ADDR\_16 |
| Checksum | 25 | -- |

Payload Data description

|  |  |  |
| --- | --- | --- |
| data fields | Offset | Example |
| Transaction ID | 0 | 0x12 |
| Device addres | 1 | DEV\_ADDR\_16 |

Hexadecimal representation:

0x76001511-- XXXXXXXXXXXXXXXXYYYY0000000500000000YYYY--

## Active Endpoint Response

* **Cluster ID: 0x8005**
* **Profile ID: 0x0000**

### Description

Indicates the list of active endpoints supported on the device.

Gli end point riportati devono essere aggiunti alla tabella di mapping del dispositivo al fine di poter essere successivamente utlilizzati.

Per dettagli vedere pag 7 di APP\_NOTE\_XBee\_ZigBee\_Device\_Profile.pdf

### Pseudo code

N/A.

### Frame

Frame details:

|  |  |  |
| --- | --- | --- |
| Frame data fields | Offset | Value |
| Start delimiter | 0 | 0x7e |
| Length | MSB 1 - LSB 2 | 0x00-- |
| Frame Type | 3 | 0x91 |
| 64-bit dest addr | MSB 4 - LSB 11 | DEV\_ADDR\_64 |
| 16-bit dest addr | MSB 12 - LSB 13 | DEV\_ADDR\_16 |
| src end point | 14 | 0x00 |
| dest end point | 15 | 0x00 |
| cluster ID | 16-17 | 0x8005 |
| Profile ID | 18-19 | 0x0000 |
| rx options | 20 | 0x-- |
| Data Payload | 21-n | -- |
| Checksum | n+1 | -- |

Data Payload description

|  |  |  |
| --- | --- | --- |
| data fields | Offset | Example |
| Transaction ID | 0 | 0x12 |
| Active End points | 1 | -- |

Active End points list:

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Size (bytes)** | **Description** |
| Status | 1 |  |
| Network Address | 2 | Indicates the 16-bit address of the responding device |
| Active Endpoint Count | 1 | Number of endpoints in the following endpoint list |
| Active Endpoint List | n | List of endpoints supported on the destination device. One byte per |

Data payload description extracted from APP\_NOTE\_XBee\_ZigBee\_Device\_Profile.pdf p.7.

Hexadecimal representation: N/A

## Simple Descriptor Request

* **Cluster ID: 0x0004**
* **Profile ID: 0x0000**

### Description

Transmission used to discover the active endpoints on a device with a matching 16-bit address.

Questa azione deve essere usata per costrure la mappa degli end\_point disponibili per ciascuno dei dispositivi connessi.

Per dettagli vedere pag 6 di APP\_NOTE\_XBee\_ZigBee\_Device\_Profile.pdf

### Pseudo code

zb.send('tx\_explicit',

dest\_addr\_long = data['source\_addr\_long'],

dest\_addr = data['source\_addr'],

src\_endpoint = '\x00',

dest\_endpoint = '\x00',

cluster = '\x00\x04',

profile = '\x00\x00',

data = '\x13' + data['source\_addr'][1] + data['source\_addr'][0] + '\x01'

)

### Frame

Frame details:

|  |  |  |
| --- | --- | --- |
| Frame data fields | Offset | Value |
| Start delimiter | 0 | 0x7e |
| Length | MSB 1 - LSB 2 | 0x0016 |
| Frame Type | 3 | 0x11 |
| Frame Seq ID | 4 | 0x-- |
| 64-bit dest addr | MSB 5 - LSB 12 | DEV\_ADDR\_64 |
| 16-bit dest addr | MSB 13 - LSB 14 | DEV\_ADDR\_16 |
| src end point | 15 | 0x00 |
| dest end point | 16 | 0x00 |
| cluster ID | 17-18 | 0x0004 |
| Profile ID | 19-20 | 0x0000 |
| broadcast radius | 21 | 0x00 |
| tx options | 22 | 0x00 |
| Data Payload | 23-25 | YYYY01 |
| Checksum | 26 | -- |

Payload Data description

|  |  |  |
| --- | --- | --- |
| data fields | Offset | Example |
| Transaction ID | 0 | -- |
| Device addres | 1 | DEV\_ADDR\_16 |
| Target End Point | 3 | 0x1 |

Hexadecimal representation:

0x76001511--XXXXXXXXXXXXXXXXYYYY0000000400000000YYYY01--

## Simple Descriptor Response

* **Cluster ID: 0x8004**
* **Profile ID: 0x0000**

### Description

Indicates the simple descriptor of the device.

Fornisce il descrittore di del dispositivo associato ad un end point.

Per dettagli vedere pag 6 di APP\_NOTE\_XBee\_ZigBee\_Device\_Profile.pdf

### Pseudo code

N/A.

### Frame

Frame details:

|  |  |  |
| --- | --- | --- |
| Frame data fields | Offset | Value |
| Start delimiter | 0 | 0x7e |
| Length | MSB 1 - LSB 2 | 0x00-- |
| Frame Type | 3 | 0x91 |
| 64-bit dest addr | MSB 4 - LSB 11 | DEV\_ADDR\_64 |
| 16-bit dest addr | MSB 12 - LSB 13 | DEV\_ADDR\_16 |
| src end point | 14 | 0x00 |
| dest end point | 15 | 0x01 |
| cluster ID | 16-17 | 0x8005 |
| Profile ID | 18-19 | 0x0000 |
| rx options | 20 | 0x-- |
| Data Payload | 21-n | -- |
| Checksum | n+1 | -- |

Data Payload description

|  |  |  |
| --- | --- | --- |
| data fields | Offset | Example |
| Transaction ID | 0 | 0x12 |
| Descriptor | 1 | -- |

Descriptor:

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Size (bytes)** | **Description** |
| Status | 1 |  |
| Network Address | 2 | Indicates the 16-bit address of the responding device |
| Length | 1 | Length of the simple descriptor |
| Simple Descriptor | Variable | See simple descriptor below. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | **Size (bits)** | **Description** | |
| Endpoint | | 8 | The endpoint on the node to which this descriptor refers. | |
| Application profile ID | | 16 | The profile ID supported on this endpoint. | |
| Application device ID | | 16 | Specifies the device description identifier supported on the device | |
| Application device version | | 4 | The version of the device description supported on this endpoint. | |
| Reserved | | 4 | |
| Input cluster count | 8 | | The number of input clusters supported on this endpoint. | |
| Input cluster list | Variable | | The list of input clusters supported on this endpoint. Each cluster is 2 bytes in size. This field is not included if the input cluster count is 0. | |
| Output cluster count | 8 | | The number of output clusters supported on this endpoint. | |
| Output cluster list | Variable | | The list of output clusters supported on this endpoint. Each cluster is 2 bytes in size. This field is not included if the output cluster count is 0. | |

Data payload description extracted from APP\_NOTE\_XBee\_ZigBee\_Device\_Profile.pdf p.6.

Hexadecimal representation: N/A

## Device Announce Message

* **Cluster ID: 0x0013**
* **Profile ID: 0x0000**

### Description

Sent by devices announcing them self in the network.

This devices need to be added to the associated device map.

### Pseudo code

N/A.

### Frame

Frame details:

|  |  |  |
| --- | --- | --- |
| Frame data fields | Offset | Value |
| Start delimiter | 0 | 0x7e |
| Length | MSB 1 - LSB 2 | 0x00-- |
| Frame Type | 3 | 0x91 |
| 64-bit dest addr | MSB 4 - LSB 11 | DEV\_ADDR\_64 |
| 16-bit dest addr | MSB 12 - LSB 13 | DEV\_ADDR\_16 |
| src end point | 14 | 0x00 |
| dest end point | 15 | 0x00 |
| cluster ID | 16-17 | 0x0013 |
| Profile ID | 18-19 | 0x0000 |
| rx options | 20 | 0x-- |
| Data Payload | 21 | -- |
| Checksum | n+1 | -- |

Data Payload description

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Size (bytes)** | **Description** |
| Transaction ID | 1 |  |
| Capabilities | 1 |  |

Hexadecimal representation: N/A

# Nodes discovery: startup sequence and persistency

At boot time the device should load from the Xbee persistency the list of associated devices. This data are stored by Xbee module in its persistency and can retrieved by requesting the neighbor table (ZDO cluster 0x0032) or the child table (extended request for ZDO cluster 0x0000 and 0x0001).

The information retrieved should be stored set of suitable data structure or DB.

## Psedocode

if starting

send: neighbor\_table\_req(coordinator)

while: running

if new\_message

goto: parse\_msg

print: dev\_map

parse\_msg:

if neighbor\_table\_res or device\_announce:

add discovered\_device to device\_map

for each dev in device\_map do:

send: active\_endpoint\_req

else if active\_end\_point\_res:

build end\_point\_map

link end\_point\_map to device\_entry in device\_map

for each end\_point in end\_point\_map do:

send: simple\_descriptor\_req

else if simple\_descriptor\_re:

add: descriptor to end\_point in end\_point\_map

## Sequence Diagram

## Data Types (proposal)



# Intruder Alarm System Devices

On a Zigbee network a special regards is dedicated to the security devoted devices and are managed under the IAS, Intrusion and Alarm System, set.

The IAS devices are classified in: Control Indicating Equipment (CIE), Zones device, Warning Device (WD) and Ancillary Control Equipment (ACE).

We will take care only of the CIE and Zones device: the CIE is our network coordinator and the Zones devices are the managed sensors. They are described in chapter 8 of the ZCL and in chapters 7 and 10.7 through 10.9.

## Enrolment Sequence