



OPEN PROTOCOL FOR ELECTRIC NETWORKS



ZigBee®

Control your world

ZigBee OpenWebNet

Version	Document		Modification	Validation	
	Date	Par		Date	Par
4.0	22/11/2016	Berryer	-new specification document	22/11/2016	B. Vulcano

Contents

1	Introduction	5
2	Hardware	6
2.1	USB Interface	6
2.1.1	Dongle based on EM3582	6
2.2	UART Interface	7
2.2.1	Small Daughter Board based on EM3581	7
2.3	Communication	8
2.3.1	Driver for USB dongle based on EM3582	8
3	OpenWebNet	9
3.1	WHO	9
3.2	WHAT	9
3.3	WHERE	9
3.4	DIMENSION	10
3.5	WHEN	10
3.6	Type of frame	10
3.7	ACK	10
3.8	NACK	10
3.9	BUSY NACK	10
3.10	Example of frames	11
4	ZigBee	12
5	Discovering products in the ZigBee Network	13
5.1	ZigBee ID	13
5.2	ZigBee Product is joining the Network	13
5.3	Neighbors table	13
5.4	Scan command	14
6	Product Database Management	15
6.1	Use cases	15
6.1.1	Interface Create a new network	15
6.1.2	Interface Join an existing network	16
6.1.3	Product Join network while interface is powered off	17
6.1.4	Product Leave network while interface is powered off	18
6.1.5	Get Product Information	19
7	Bootload	20
7.1	Ember 358	20
8	Lighting WHO = 1	21
8.1	Use cases	21
8.1.1	Switch light ON	21
8.1.2	Switch light OFF	21
8.1.3	Toggle light to switch product (Supervisor mode disable)	21
8.1.4	Toggle light to switch product (Supervisor mode enable)	21
8.1.5	Movement detector	21
8.2	WHAT table	22
8.3	WHAT id	23
8.3.1	OFF	23
8.3.2	OFF at X speed	24
8.3.3	ON	25
8.3.4	ON at X speed	26
8.3.5	20% to 100%	27
8.3.6	TimeOn x	28
8.3.7	Toggle	30
8.3.8	Movement detected	32
8.3.9	End of movement detected	32
8.4	DIMENSION table	33

8.5	DIMENSION id	33
8.5.1	Setup Level and Speed	33
8.6	REQUEST	35
9	Automation WHO = 2	36
9.1	Use cases	36
9.1.1	Send UP command	36
9.1.2	Send STOP command	36
9.1.3	Send UP command (Supervisor mode enabled)	36
9.1.4	Send Request state	36
9.2	WHAT table	36
9.3	WHAT id	37
9.3.1	STOP	37
9.3.2	UP	38
9.3.3	DOWN	38
9.4	DIMENSION table	39
9.5	DIMENSION id	39
9.5.1	Read position	39
9.5.2	Move to position	39
9.6	REQUEST	40
10	Temperature WHO = 4	41
10.1	Use cases	41
10.1.1	Receive positive temperature value from probe	41
10.1.2	Receive negative temperature value from probe	41
10.2	WHAT table	41
10.3	WHAT id	41
10.4	DIMENSION table	41
10.5	DIMENSION id	41
10.5.1	Temperature level	41
11	Management WHO = 13	42
11.1	Use cases	42
11.1.1	Create ZigBee network	42
11.1.2	Open ZigBee network	42
11.1.3	Product Opens ZigBee network	42
11.1.4	Close ZigBee network	42
11.1.5	Product Closes ZigBee network	42
11.1.6	Join ZigBee network	42
11.1.7	Product Joins ZigBee network	42
11.1.8	Leave ZigBee network	42
11.1.9	Product Leaves ZigBee network	43
11.1.10	Scan network	43
11.1.11	Supervisor	43
11.1.12	Product Information	43
11.1.13	Device MAC address by index	43
11.2	WHAT table	44
11.3	WHAT id	44
11.3.1	Boot mode	44
11.3.2	Reset	44
11.3.3	Create	44
11.3.4	Close	45
11.3.5	Open	45
11.3.6	Join	45
11.3.7	Leave	45
11.3.8	Keep connect	46
11.3.9	Identify	46
11.3.10	Scan	46
11.3.11	Supervisor	46
11.3.12	Supervisor remove	47
11.4	DIMENSION table	48

11.5	DIMENSION id	48
11.5.1	Mac address	48
11.5.2	Firmware version	48
11.5.3	Hardware version	49
11.5.4	Who implemented	49
11.5.5	Product information	49
11.5.6	Get number of product in network	51
11.5.7	ZigBee channel	51
11.5.8	Battery information	51
11.5.9	Device MAC address by index	52
12	Automation WHO = 18	53
12.1	Use cases	53
12.1.1	Get Voltage	53
12.1.2	Get Current	53
12.1.3	Get Frequency	53
12.2	WHAT table	53
12.3	WHAT id	53
12.3.1	Reset	53
12.4	DIMENSION table	54
12.5	DIMENSION id	54
12.5.1	Voltage	54
12.5.2	Current	54
12.5.3	Energy	55
12.5.4	Frequency	55
12.5.5	Active Power	55
12.5.6	Active Power Total	56
12.5.7	Threshold Max Active Power	56
12.5.8	Reactive Power	56
12.5.9	Report Power	57
13	Cen Plus WHO = 25	58
13.1	Use cases	58
13.1.1	Send Binding Request command	58
13.1.2	Received command from bound device	58
13.1.3	Send Unbinding Request command	58
13.2	WHAT table	58
13.3	WHAT id	58
13.3.1	Short Pressure	58
13.3.2	Binding Request	59
13.3.3	Unbinding Request	59
13.3.4	Open Binding	59
13.3.5	Close Binding	59
13.3.6	Cancel Binding	59
13.4	DIMENSION table	60
13.5	DIMENSION id	60

1 Introduction

The OpenWebNet (OWN) interface is the device that allows interacting with the ZigBee Legrand products. An application who integrates port COM management can interact with the interface.

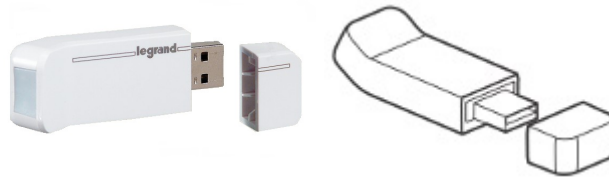
△ Points to be confirmed are highlighted in yellow (Draft document only).

2 Hardware

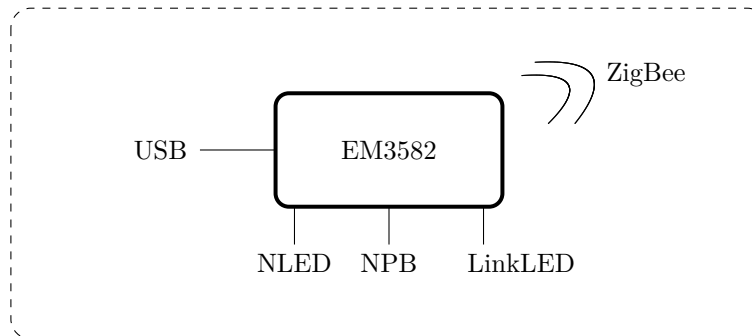
2.1 USB Interface

2.1.1 Dongle based on EM3582

- Mechanic



- Schematic



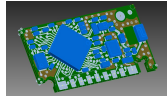
The NPB button is used for actions on the ZigBee network as Open, Close, Create, Leave and Join. There are two LEDs green and orange:

- The orange LED shows to user the network state (Open/Close/Leave/Join/Create).
- The green LED shows to user the network activity.

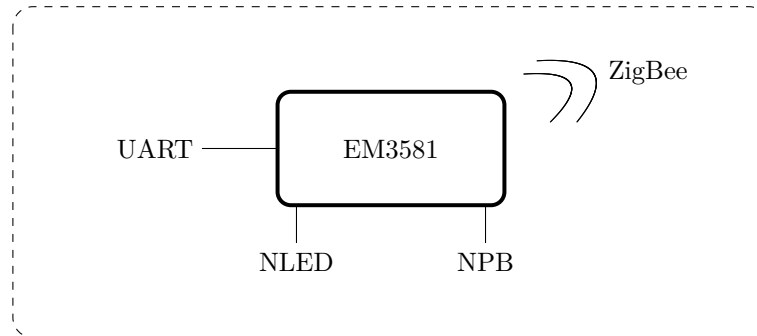
2.2 UART Interface

2.2.1 Small Daughter Board based on EM3581

- Mechanic



- Schematic



The NPB button is used for actions on the ZigBee network as Open, Close, Create, Leave and Join. The orange LED shows to user the network state (Open/Close/Leave/Join/Create).

2.3 Communication

The UART protocol is used for both interfaces in order to communicate with the OpenWebNet protocol :

- 19200 bauds
- 8 bits data
- 1 stop bit
- No parity

△ There is a possibility for USB interface to increase its baudrate up to 115200 bauds.

2.3.1 Driver for USB dongle based on EM3582

The set up is describe in the following document in the section 2 "Host PC Driver" page 2.

<http://www.silabs.com/Support%20Documents/TechnicalDocs/AN740.pdf>

3 OpenWebNet

The string transmitted, regardless of the system, is the structured with successive fields that progressively specify the details of the information contained. The following logic is used:

*WHO*WHAT*WHERE*WHEN##

The OpenWebNet string must start with * and must finish by ##.

3.1 WHO

The field WHO permits to select function such as Lighting, Automation, and Management. The table below show admitted WHO value for ZigBee interface.

WHO	Function
1	Lighting
2	Automation
4	Temperature
13	Management
18	Energy Management
25	CenPlus
1000	Diagnostic

3.2 WHAT

The field WHAT characterizes an action to do or a status. For every WHO there is a specific WHAT table.

3.3 WHERE

It characterizes the set of objects to which the OpenWebNet message is referred. It can be a single product, a specific environment, the entire system, etc. In our case the WHERE field is divided into parameters in order to identify the destination or source and the transmission type command.

WHERE = (TX) ADDR (SYS)

TX:

TX	Description
-	No parameter, transmission is UNICAST , ADDR is the destination address
#	<i>(Not implemented)</i> The transmission is in MULTICAST , ADDR is the destination address
0#	The transmission is in BROADCAST , ADDR is the UNIT you want to control. All products do an action on the selected UNIT. BE CARFEFULL: Do not sent more than 1 broadcast during 1 second. The effect of this case is to overload network and could saturate the ZigBee network, and some product will not receive the data during the next 8 seconds.

ADDR: ADDR is the four last bytes of the MAC address of the product converted in decimal format + the unit you want to control.

The UNIT is **always** on 2 length character;

In broadcast mode if you want to send to all units, you have to send unit equal to 00.

SYS: SYS is the family type, in our case SYS is **always** equal to #9 corresponding to ZigBee network.

3.4 DIMENSION

Is a range of value that characterizes the dimension of the object to which the message is referred. For every WHO (and therefore for every function) there is a specific DIMENSION table. It is possible to require/to read/to write the value of one dimension. Every dimension has a prefixed number of values, described in VALUE field.

3.5 WHEN

Never used.

3.6 Type of frame

- Command / Status Message
*WHO*WHAT*WHERE##
- Status Request Message
*#WHO*WHERE##
- Dimension Message Reques
*#WHO*WHERE*DIMENSION##
- Dimension Message Request with parameters
*#WHO*WHERE*DIMENSION#PARAM1#...# PARAMn##
- Dimension Message Read with parameters
*#WHO*WHERE*DIMENSION#PARAM1#...# PARAMn*VALUE1*...*VALUEn##
- Dimension Message Read
*#WHO*WHERE*DIMENSION*VALUE1*...*VALUEn##
- Dimension Message Write
*#WHO*WHERE*#DIMENSION*VALUE1*...*VALUEn##
- Dimension Message Write with parameters
*#WHO*WHERE*#DIMENSION#PARAM1#...# PARAMn*VALUE1*...*VALUEn##
- Acknowledge Message ACK
*#*1##
- None Acknowledge Message NACK
*#*0##
- Busy None Acknowledge Message BUSY NACK
*#*6##

3.7 ACK

ACK Message will be received when interface execute the incomming message.
There is exactly (unique) one ACK per command.

3.8 NACK

NACK Message will be received if the message is uncorrect or unsupported. When WHERE field does not exist in the network too. The NACK message will be sent after each BUSY NACK received frame.
There is exactly (unique) one NACK per command.

3.9 BUSY NACK

BUSY NACK Message will be received when interface can not temporarily accept more OpenWebNet frames over Zigbee network. When BUSY NACK is received you must wait 500ms before sending the same frame again. BUSY NACK message is always followed by a NACK message.

3.10 Example of frames

- Transmission **unicast** in ZigBee, command **Light ON** and ID = 0x000501F8 (0d328184) and you want to control the **Unit 2**
*1*1*32818402#9##
 - Transmission **unicast** in ZigBee, command **Light ON** and ID = 0x000501F8 (0d328184) and you want to control the **All Units**
*1*1*32818400#9##
 - Transmission **unicast** in ZigBee, command **Automation UP** and ID = 0x00050EC2 (0d331458) and you want to control the **Unit 1**
*2*2*33145801#9##
 - Transmission **unicast** in ZigBee, command **Automation STOP** and ID = 0x00050EC2 (0d0d331458) and you want to control the **Unit 1**
*2*0*33145801#9##
 - Transmission **broadcast** in ZigBee, command **Light OFF** and you want to control the **Unit 1**
*1*0*0#01#9##
 - Transmission **broadcast** in ZigBee, command **Light OFF** and you want to control the all **All Units**
*1*0*0#00#9##
 - Transmission **broadcast** in ZigBee, command **Automation UP** and you want to control the **Unit 1**
*2*2*0#01#9##
 - Transmission **broadcast** in ZigBee, command **Automation STOP** and you want to control the **All Units**
*2*0*0#00#9##
 - ACK message

TX	*13*60*##	}	The command returns ACK .
RX	*##*1##		
 - NACK message

TX	*50*1*32455801#9##	}	The command returns NACK because the WHO = 50 does not exist.
RX	*##*0##		
 - Busy NACK example

TX	*1*1*32455801#9##	}	The command ON will be sent over the ZigBee network.
RX	*##*1##		
TX	*1*0*32455801#9##	}	The command OFF will be sent over the ZigBee network.
RX	*##*1##		
TX	*1*1*32455801#9##	}	The command ON will be sent over the ZigBee network.
RX	*##*1##		
TX	*1*0*32455801#9##	}	BUSY NACK received, the command OFF will not be sent over the ZigBee network.
RX	*##*6##		
RX	*##*0##		
- wait 500ms
- | | | | |
|----|-------------------|---|---|
| TX | *1*0*32455801#9## | } | The command OFF will be sent over the ZigBee network. |
| RX | *##*1## | | |

4 ZigBee

The applicative layer of the interfaces uses the ZigBee MSP (Manufacturing Specific Profile) from Legrand.

⚠ Only the products compatible with the ZigBee network 2.1 can be managed by the interface. This logo below is set on product.

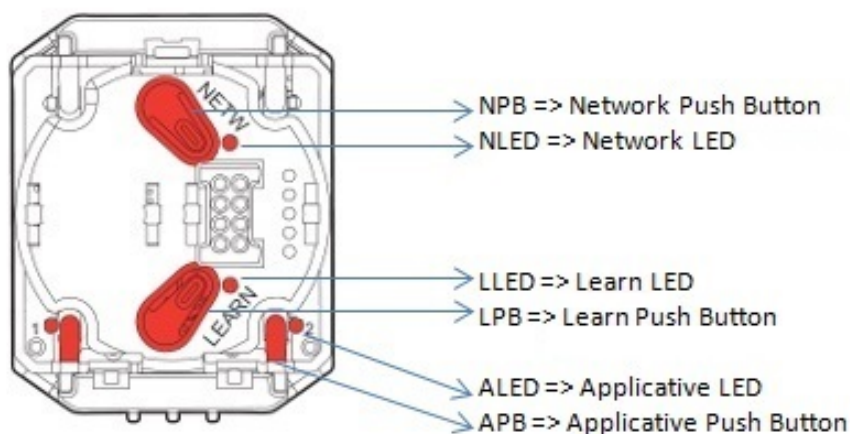


⚠ In order to guarantee maximum ZigBee network performance there must be only one OpenWebNet interface with supervisor mode enabled inside the ZigBee network. If there are more OpenWebNet interfaces, supervisor mode must be disabled in the rest of them. For development purposes it is possible to have more than one interface with supervisor mode enabled considering that higher the number of supervisors lower the performance of ZigBee network.

You will find 2 main kind of product on the ZigBee network :

- ZR(ZigBee Router) = main powered device
- ZED(ZigBee End Device) = battery powered device

In the picture below you will see the main parts on ZigBee products.



5 Discovering products in the ZigBee Network

There are different ways to discover the addresses of the ZigBee products of the network (WHERE field):

5.1 ZigBee ID

You can see this ZigBee ID address in front of the product (example: ID => 0x0005059D must be converted in decimal format => 0d329117).



5.2 ZigBee Product is joining the Network

You can see it when the product enters in the network.

TX	*13*32*##	}	Open the ZigBee network, in order to accept a new product.
RX	*#*1##		
RX	*#13*33*47452500#9##	}	The product is joining the ZigBee network.
TX	*13*31*##		
RX	*#*1##	}	Close the ZigBee network.

5.3 Neighbors table

Discovering can be done by the **neighbors** sequence. This command permits to know the neighbors of each ZR in the ZigBee network.

The first step is to ask to the interface his own neighbors.

TX	*#1000**81##	}	ask to interface his neighbors.
RX	*#1000**81#0*3*46254453##		
RX	*#1000**81#1*4*9088239##	}	answer from interface of all known neighbors.
RX	*#1000**81#2*3*654532##		
RX	*#1000**81#3*5*456416##		
RX	*#1000**81#4*6*9864164##		
RX	*#1000**81#5*3*1876516##		
RX	*#*1##	}	final Ack.

The second step is to ask to each new discovered ZR their neighbors.

TX	*#1000*4625445300#9*81##	}	ask to ZR his neighbors.
RX	*#1000*4625445300#9*81#0*3*46254453##		
RX	*#1000*4625445300#9*81#1*4*9088239##	}	answer from ZR of all known neighbors.
RX	*#1000*4625445300#9*81#2*3*654532##		
RX	*#*1##	}	final Ack.

With this sequence it's possible to get all products of the installation.

5.4 Scan command

Discovering can be done by the **scan** sequence. The scan command permits to send a broadcast message over the ZigBee Network, all active devices (ZR and ZED who are not sleeping) will answer to this broadcast.

TX	*13*65*##	}	The interface sent a broadcast in the ZigBee network.
RX	*#*1##		
			13 seconds after.
RX	*#13**67*3##	}	The interface sends the number of product inside the products database.
TX	*#13**73#0##	}	The user wants to get infos about product index 0 in the products database
RX	*#13*47485500#9*73#0*0##	}	The product sends to the interface each endpoints and each device IDs.
RX	*#*1##	}	ACK at the end
TX	*#13**73#1##	}	The user wants to get infos about product index 1 in the products database
RX	*#13*54672700#9*73#1*0##	}	The product sends to the interface each endpoints and each device IDs.
RX	*#*1##	}	ACK at the end
TX	*#13**73#2##	}	The user wants to get infos about product index 2 in the products database
RX	*#13*45646800#9*73#2*0##	}	The product sends to the interface each endpoints and each device IDs.
RX	*#*1##	}	ACK at the end

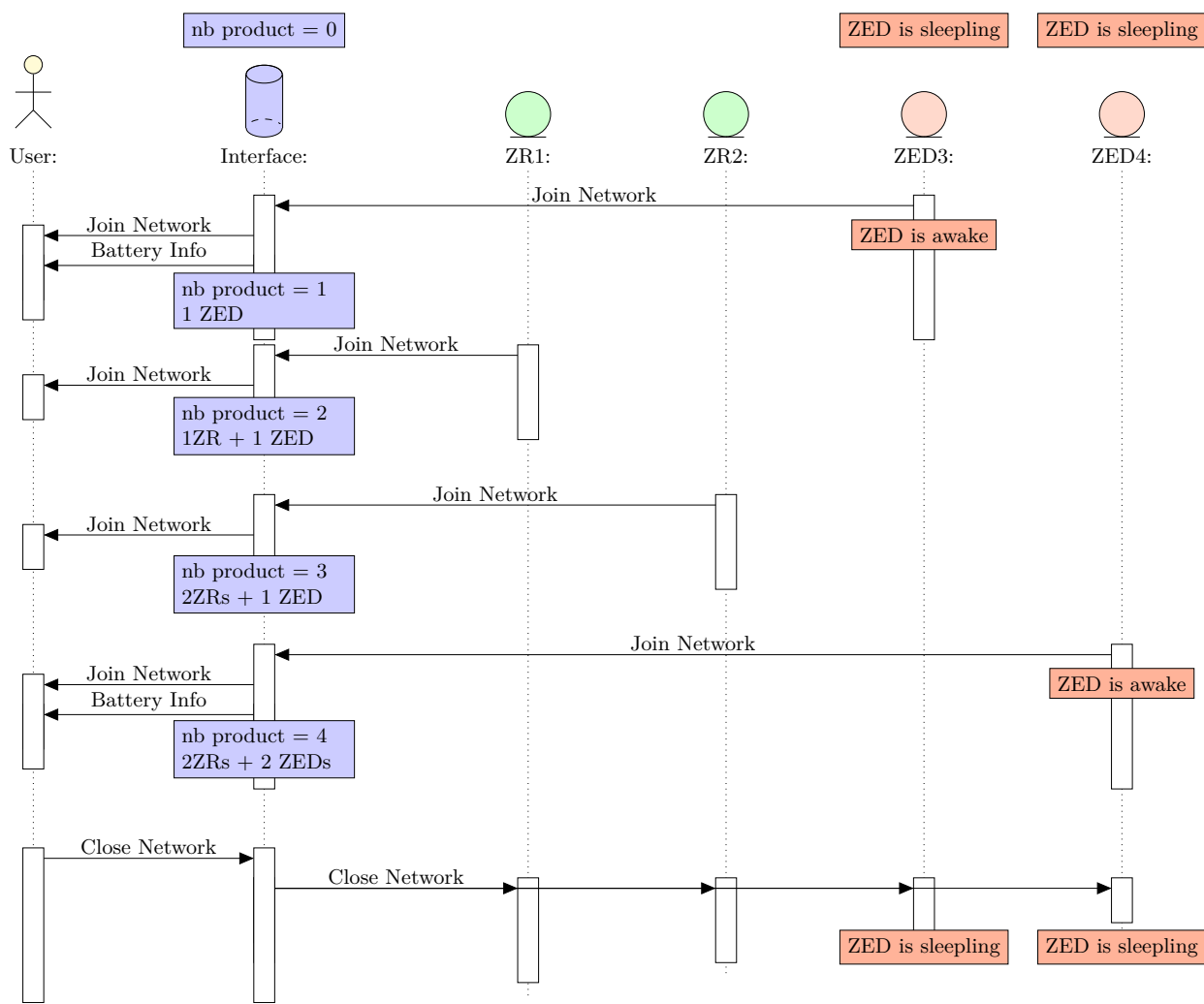
6 Product Database Management

The OpenWebNet interface contains a product database which can contains up to 175 products. This database is managed automatically by the interface. Any new product who is joining the network fill the product database.

6.1 Use cases

6.1.1 Interface Create a new network

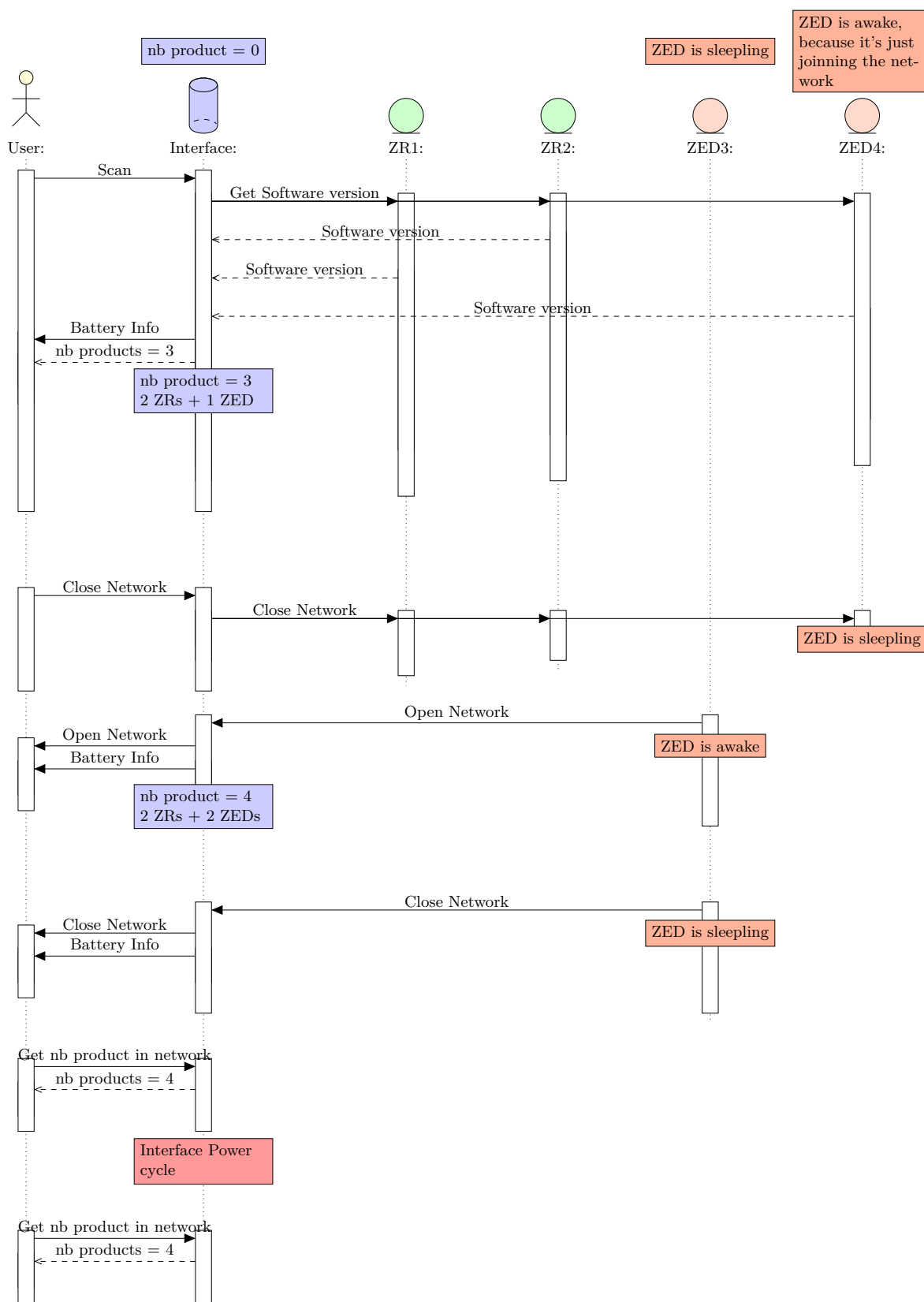
In this case the interface has already created the ZigBee network. The user must push the NPB on product in order to enter them into the ZigBee network.



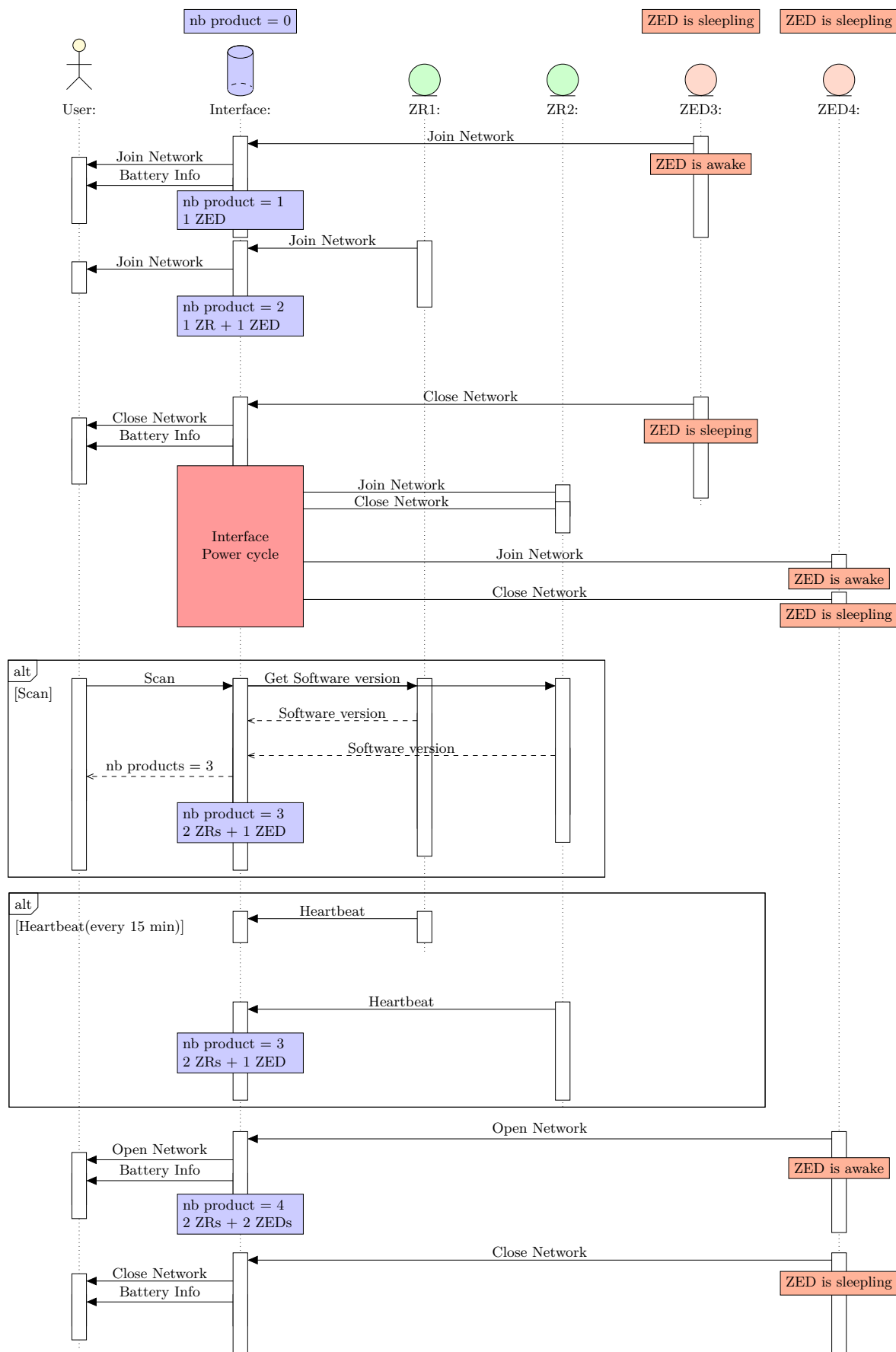
6.1.2 Interface Join an existing network

In this case the interface doesn't know products inside the ZigBee network. The OpenWebNet user has to send a scan command in order to know each active devices. For others, the user must push the NPB in order to fill the database.

After a power cycle on the interface, the number of products inside database doesn't change.

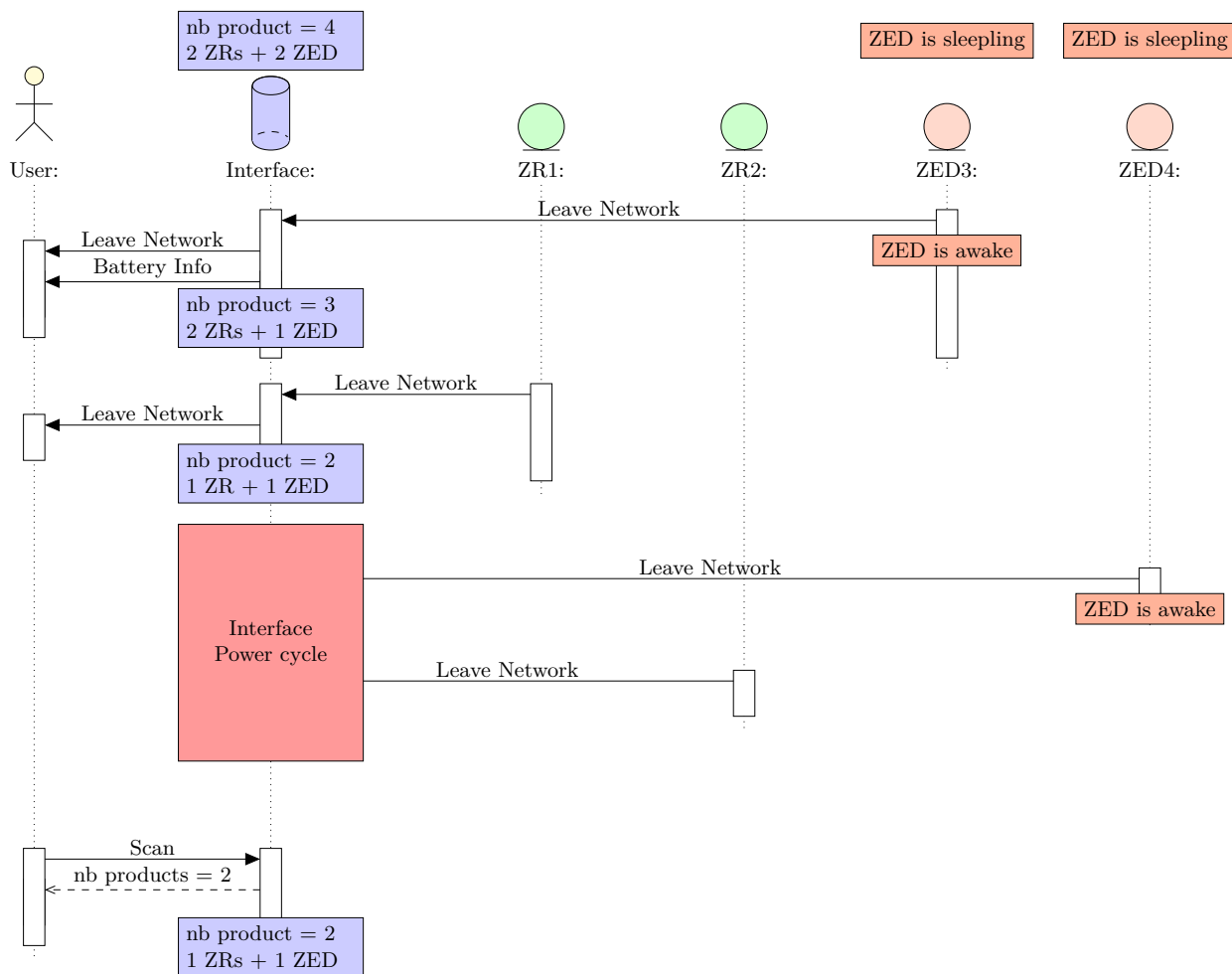


In this case the interface doesn't know the new incoming product. The user must push the NPB in order to fill the battery powered devices inside the database.



6.1.4 Product Leave network while interface is powered off

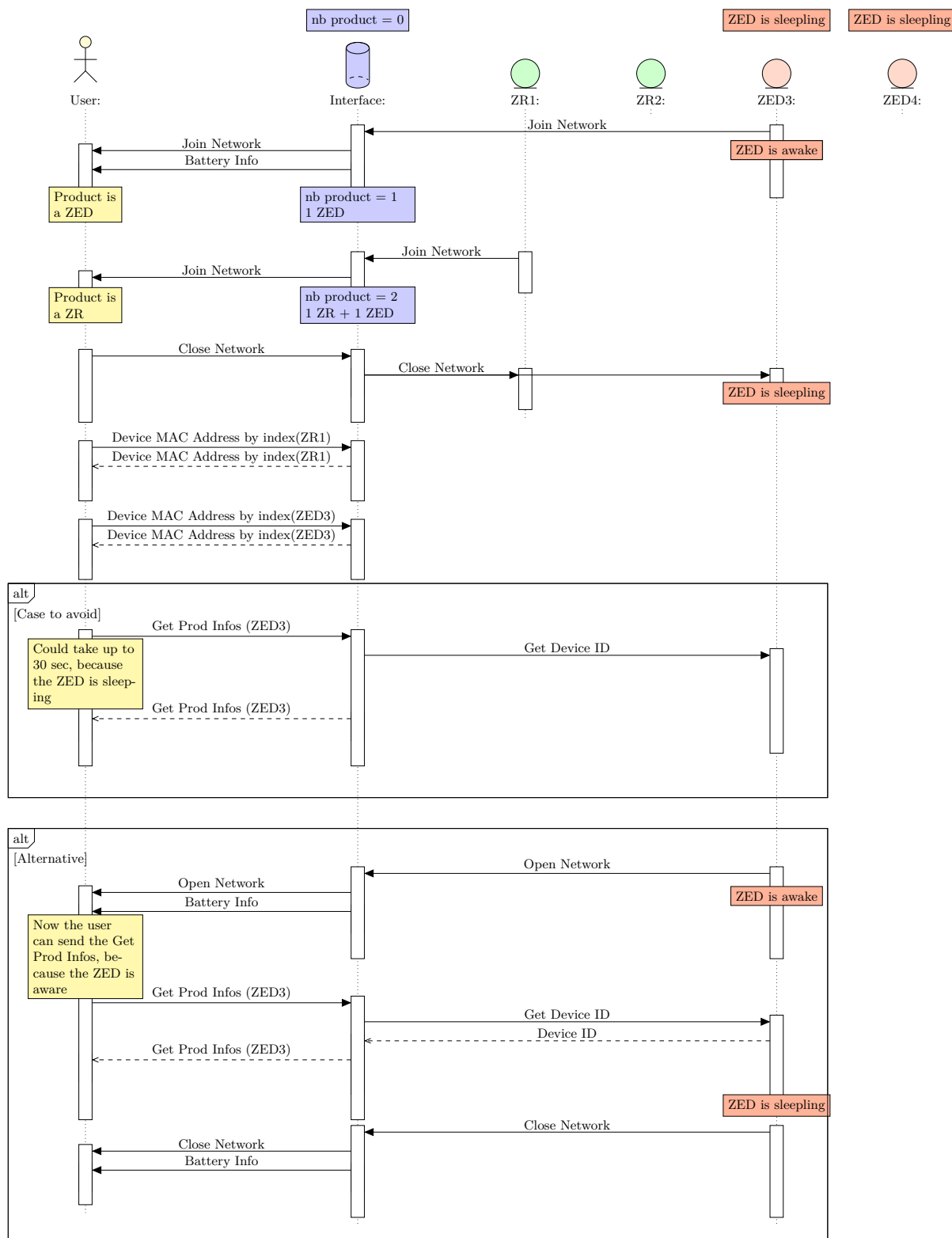
In this case the interface doesn't know the old product who left the ZigBee network. The product who left the network while the interface is powered off will be **always** inside database.



6.1.5 Get Product Information

The command **Get Product Information** sends a ZigBee message in order to get the device ID of an index of a product inside the interface database.

This ZigBee message could take a long time (30 seconds) before receive the OpenWebNet answer in case of no reachable product (battery powered device). The user can know if the product is a battery powered device because after each OpenWebNet frames it receives the **Battery information** dimension.



7 Bootload

7.1 Ember 358

The bootload procedure is describe in the following document in the section "1.1 Serial Upload" page 1.

<https://www.silabs.com/Support%20Documents/TechnicalDocs/AN760-Using-Standalone-Bootloader.pdf>

8 Lighting WHO = 1

This field is used to switch lights ON or OFF on the ZigBee network.

8.1 Use cases

8.1.1 Switch light ON

In this section we will send a command Light ON on unit 2:

```
TX  *1*1*32818402#9##  }
RX  *#*1##             } Send OpenWebNet command Light ON.
```

8.1.2 Switch light OFF

In this section we will send a command Light OFF on all units:

```
TX  *1*0*32818400#9##  }
RX  *#*1##             } Send OpenWebNet command Light OFF.
```

8.1.3 Toggle light to switch product (Supervisor mode disable)

In this section we will send a command Toggle on unit 1:

```
TX  *1*32*32818401#9##  }
RX  *#*1##             } Send OpenWebNet command Toggle.
RX  *1*1*32818401#9##  } Received product state.
```

8.1.4 Toggle light to switch product (Supervisor mode enable)

In this section we will send a command Toggle on unit 1:

```
TX  *1*32*32818401#9##  }
RX  *#*1##             } Send OpenWebNet command Toggle.
RX  *1*1*32818401#9##  } Received product state.
RX  *1*1*32818401#9##  } Received product state (supervisor mode).
```

8.1.5 Movement detector

In this section we will receive a command from movement detector:

△In order to receive this frame the detector must do previously a PnL procedure with OpenWebNet interface (see WHO=25 use cases for more details).

```
RX  *25*35*32818402#9##  }
RX  *1*34*32818402#9##  } When room is occupied.

RX  *1*32*32818402#9##  } Received Toogle at the half of time configured on the detector.
                          } While people are detected, we will received this message from detector.
RX  *1*39*32818402#9##  } Received unoccupied at the end of the time configured on the detector.
```

8.2 WHAT table

ID	Action
0	OFF
0#speed	OFF at X speed
1	ON
1#speed	ON at X speed
2	20%
3	30%
4	40%
5	50%
6	60%
7	70%
8	80%
9	90%
10	100%
11	TimeOn 1 minute
12	TimeOn 2 minutes
13	TimeOn 3 minutes
14	TimeOn 4 minutes
15	TimeOn 5 minutes
16	TimeOn 15 minutes
17	TimeOn 30 seconds
18	TimeOn 0.5 second
32	Toggle
34	Movement detected
39	End of movement detected

8.3 WHAT id

8.3.1 OFF

- switch case :

Direction	OpenWebNet Frame	Description
client → server	*1*0*where#9##	where is the ZigBee product id <ul style="list-style-type: none"> • if unicast : where = MAC address in decimal + unit • if broadcast : where=0# + unit
client ← server	*#*0##	The command has not been sent
client ← server	*#*6##	Wait 500ms before sending the frame again.
client ← server	*#*0##	
client ← server	*#*1##	The command has been sent
client ← server	*1*0*where#9##	If Supervision mode enabled, refer to section WHO = 13, WHAT = 66 where = MAC address in decimal + unit

- dimmer case :

Direction	OpenWebNet Frame	Description
client → server	*1*0*where#9##	where is the ZigBee product id <ul style="list-style-type: none"> • if unicast : where = MAC address in decimal + unit • if broadcast : where=0# + unit
client ← server	*#*0##	The command has not been sent
client ← server	*#*6##	Wait 500ms before sending the frame again.
client ← server	*#*0##	
client ← server	*#*1##	The command has been sent
client ← server	*1*0*where#9##	If Supervision mode enabled, refer to section WHO = 13, WHAT = 66 where = MAC address in decimal + unit
client ← server	*1*0*where#9##	If Supervision mode enabled, refer to section WHO = 13, WHAT = 66 where = MAC address in decimal + unit When dimmer reaches his 0% level state.

8.3.2 OFF at X speed

- dimmer case :

Direction	OpenWebNet Frame	Description
client → server	*1*0#speed*where#9##	<p>where is the ZigBee product id</p> <ul style="list-style-type: none"> • if unicast : where = MAC address in decimal + unit • if broadcast : where=0# + unit <p>speed is the delay to switch Off the light</p> <ul style="list-style-type: none"> • 0 : last speed used • 1-254 : speed value • 255 : default speed
client ← server	*#*0##	The command has not been sent
client ← server	*#*6##	Wait 500ms before sending the frame again.
client ← server	*#*0##	
client ← server	*#*1##	The command has been sent
client ← server	*1*0*where#9##	<p>If Supervision mode enabled, refer to section WHO = 13, WHAT = 66</p> <p>where = MAC address in decimal + unit</p>
client ← server	*1*0*where#9##	<p>If Supervision mode enabled, refer to section WHO = 13, WHAT = 66</p> <p>where = MAC address in decimal + unit</p> <p>When dimmer reaches his 0% level state.</p>

8.3.3 ON

Direction	OpenWebNet Frame	Description
client → server	*1*1*where#9##	where is the ZigBee product id <ul style="list-style-type: none"> • if unicast : where = MAC address in decimal + unit • if broadcast : where=0# + unit
client ← server	*#*0##	The command has not been sent
client ← server	*#*6##	Wait 500ms before sending the frame again.
client ← server	*#*0##	
client ← server	*#*1##	The command has been sent
client ← server	*1*state*where#9##	<p>If Supervision mode enabled, refer to section WHO = 13, WHAT = 66</p> <p>state = 1 if switch</p> <p>state = 2 to 10 if dimmer (depending from last dimming value)</p> <p>where = MAC address in decimal + unit</p>

8.3.4 ON at X speed

- dimmer case :

△ This command will set the dimmer level to 100%.

Direction	OpenWebNet Frame	Description
client → server	*1*1#speed*where#9##	<p>where is the ZigBee product id</p> <ul style="list-style-type: none"> • if unicast : where = MAC address in decimal + unit • if broadcast : where=0# + unit <p>speed is the delay to switch On the light</p> <ul style="list-style-type: none"> • 0 : last speed used • 1-254 : speed value • 255 : default speed
client ← server	*#*0##	The command has not been sent
client ← server	*#*6##	Wait 500ms before sending the frame again.
client ← server	*#*0##	
client ← server	*#*1##	The command has been sent
client ← server	*1*10*where#9##	<p>If Supervision mode enabled, refer to section WHO = 13, WHAT = 66</p> <p>where = MAC address in decimal + unit</p>

8.3.5 20% to 100%

Direction	OpenWebNet Frame	Description
client → server	*1*value*where#9##	<p>where is the ZigBee product id</p> <ul style="list-style-type: none"> • if unicast : where = MAC address in decimal + unit • if broadcast : where=0# + unit <p>value</p> <ul style="list-style-type: none"> • 2 : 20% • 3 : 30% • 4 : 40% • 5 : 50% • 6 : 60% • 7 : 70% • 8 : 80% • 9 : 90% • 10 : 100%
client ← server	*#*0##	The command has not been sent
client ← server	*#*6##	Wait 500ms before sending the frame again.
client ← server	*#*0##	
client ← server	*#*1##	The command has been sent
client ← server	*1*value*where#9##	<p>If Supervision mode enabled, refer to section WHO = 13, WHAT = 66</p> <p>where = MAC address in decimal + unit</p> <p>value</p> <ul style="list-style-type: none"> • 2 : 20% • 3 : 30% • 4 : 40% • 5 : 50% • 6 : 60% • 7 : 70% • 8 : 80% • 9 : 90% • 10 : 100%

8.3.6 TimeOn x

- switch case :

Direction	OpenWebNet Frame	Description
client → server	*1*timeon*where#9##	<p>where is the ZigBee product id</p> <ul style="list-style-type: none"> • if unicast : where = MAC address in decimal + unit • if broadcast : where=0# + unit <p>timeon</p> <ul style="list-style-type: none"> • 11 : 1 minute • 12 : 2 minutes • 13 : 3 minutes • 14 : 4 minutes • 15 : 5 minutes • 16 : 15 minutes • 17 : 30 seconds • 18 : 0.5 second
client ← server	*#*0##	The command has not been sent
client ← server	*#*6##	Wait 500ms before sending the frame again.
client ← server	*#*0##	
client ← server	*#*1##	The command has been sent
client ← server	*1*1*where#9##	<p>If Supervision mode enabled, refer to section WHO = 13, WHAT = 66</p> <p>where = MAC address in decimal + unit</p>
1 minute after sent the command client ← server	*1*0*where#9##	<p>If Supervision mode enabled, refer to section WHO = 13, WHAT = 66</p> <p>where = MAC address in decimal + unit</p>

• dimmer case :

Direction	OpenWebNet Frame	Description
client → server	*1*timeon*where#9##	<p>where is the ZigBee product id</p> <ul style="list-style-type: none"> • if unicast : where = MAC address in decimal + unit • if broadcast : where=0# + unit <p>timeon</p> <ul style="list-style-type: none"> • 11 : 1 minute • 12 : 2 minutes • 13 : 3 minutes • 14 : 4 minutes • 15 : 5 minutes • 16 : 15 minutes • 17 : 30 seconds • 18 : 0.5 second
client ← server	*#*0##	The command has not been sent
client ← server	*#*6##	Wait 500ms before sending the frame again.
client ← server	*#*0##	
client ← server	*#*1##	The command has been sent
client ← server	*1*state*where#9##	<p>If Supervision mode enabled, refer to section WHO = 13, WHAT = 66</p> <p>where = MAC address in decimal + unit</p> <p>state = 2 to 10 (depending from last dimming value)</p>
delay after sent the command		
client ← server	*1*0*where#9##	<p>If Supervision mode enabled, refer to section WHO = 13, WHAT = 66</p> <p>where = MAC address in decimal + unit</p>
client ← server	*1*0*where#9##	<p>If Supervision mode enabled, refer to section WHO = 13, WHAT = 66</p> <p>where = MAC address in decimal + unit</p> <p>When dimmer reaches his 0% level state.</p>

8.3.7 Toggle

- **switch case :**

△ when product toggles it replies with its state (even if the supervisor mode is off). So, when the supervisor is enable, the actuator replies twice.

Direction	OpenWebNet Frame	Description
client → server	*1*32*where#9##	where is the ZigBee product id <ul style="list-style-type: none"> • if unicast : where = MAC address in decimal + unit • if broadcast : where=0# + unit
client ← server	*#*0##	The command has not been sent
client ← server	*#*6##	Wait 500ms before sending the frame again.
client ← server	*#*0##	
client ← server	*#*1##	The command has been sent
client ← server	*1*state*where#9##	where = MAC address in decimal + unit state = 0 or 1
client ← server	*1*state*where#9##	If Supervision mode enabled, refer to section WHO = 13, WHAT = 66 where = MAC address in decimal + unit state = 0 or 1

- **dimmer case :**

△ If the dimmer turns On.

Direction	OpenWebNet Frame	Description
client → server	*1*32*where#9##	where is the ZigBee product id <ul style="list-style-type: none"> • if unicast : where = MAC address in decimal + unit • if broadcast : where=0# + unit
client ← server	*#*0##	The command has not been sent
client ← server	*#*6##	Wait 500ms before sending the frame again.
client ← server	*#*0##	
client ← server	*#*1##	The command has been sent
client ← server	*1*state*where#9##	where = MAC address in decimal + unit state = 2 to 10 (depending from last dimming value)

△ If the dimmer turns Off.

Direction	OpenWebNet Frame	Description
client → server	*1*32*where#9##	where is the ZigBee product id <ul style="list-style-type: none"> • if unicast : where = MAC address in decimal + unit • if broadcast : where=0# + unit
client ← server	*#*0##	The command has not been sent
client ← server	*#*6##	Wait 500ms before sending the frame again.
client ← server	*#*0##	
client ← server	*#*1##	The command has been sent
client ← server	*1*0*where#9##	where = MAC address in decimal + unit
client ← server	*1*0*where#9##	If Supervision mode enabled, refer to section WHO = 13, WHAT = 66 where = MAC address in decimal + unit
client ← server	*1*0*where#9##	If Supervision mode enabled, refer to section WHO = 13, WHAT = 66 When dimmer reaches his 0% level state. where = MAC address in decimal + unit

8.3.8 Movement detected

△ In order to receive this frame the detector must do previously a PnL procedure with OpenWebNet interface (see WHO=25 use cases for more details).

Direction	OpenWebNet Frame	Description
client ← server	*1*34*where#9###	where is the ZigBee product id unicast : where = MAC address in decimal + unit

8.3.9 End of movement detected

△ In order to receive this frame the detector must do previously a PnL procedure with OpenWebNet interface (see WHO=25 use cases for more details).

Direction	OpenWebNet Frame	Description
client ← server	*1*39*where#9###	where is the ZigBee product id unicast : where = MAC address in decimal + unit

8.4 DIMENSION table

ID	Action
1	Setup Level and Speed

8.5 DIMENSION id

8.5.1 Setup Level and Speed

Read Level and Speed :

Direction	OpenWebNet Frame	Description
client → server	*#1*where#9*1##	where is the ZigBee product id <ul style="list-style-type: none"> if unicast : where = MAC address in decimal + unit if broadcast : where=0# + unit
client ← server	*#1*where#9*1*value1*value2##	where is the ZigBee product id <ul style="list-style-type: none"> if unicast : where = MAC address in decimal + unit if broadcast : where=0# + unit value1 : Bright intensity increase of light point, it expressed with percentage value from 101 to 200 value2 : Bright intensity change speed of light point, from 0(immediate) to 255(maximum delay)
client ← server	*#*1##	The command has been sent
client ← server	*#*0##	The command has not been sent
client ← server	*#*6##	Wait 500ms before sending the frame again.
client ← server	*#*0##	

Write Level and Speed :

Direction	OpenWebNet Frame	Description
client → server	*#1*where#9*#1*value1*value2###	<p>where is the ZigBee product id</p> <ul style="list-style-type: none"> • if unicast : where = MAC address in decimal + unit • if broadcast : where=0# + unit <p>value1 : Bright intensity increase of light point, it expressed with percentage value from 101 to 200</p> <p>value2 : Bright intensity change speed of light point, from 0(immediate) to 255(maximum delay)</p>
client ← server	*#*1###	The command has been sent
client ← server	*#*0###	The command has not been sent
client ← server	*#*6###	Wait 500ms before sending the frame again.
client ← server	*#*0###	

8.6 REQUEST

Direction	OpenWebNet Frame	Description
client → server	*#1*where#9##	where is the ZigBee product id where = MAC address in decimal + unit
client ← server	*#*0##	The command has not been sent
client ← server	*#*6##	Wait 500ms before sending the frame again.
client ← server	*#*0##	
client ← server	*1*state*where#9##	where = MAC address in decimal + unit state depending from the state of the product unit state = 0 or 1 for switch state = 0 or level for dimmer
client ← server	*#*1##	The command is received

9 Automation WHO = 2

This field is used to send UP/DOWN/STOP commands on shutters ZigBee products.

9.1 Use cases

9.1.1 Send UP command

In this section we will send a command UP on shutter on unit 1:

```
TX  *2*1*32818401#9##
RX  *#*1##
```

} Send OpenWebNet command UP.

9.1.2 Send STOP command

In this section we will send a command STOP on shutter on unit 1:

```
TX  *2*0*32818401#9##
RX  *#*1##
```

} Send OpenWebNet command STOP.

9.1.3 Send UP command (Supervisor mode enabled)

In this section we will send a command UP on shutter on unit 1, and the shutter reaches its upper limit:

```
TX  *2*1*32818401#9##
RX  *#*1##
RX  *2*1*32818401#9##
RX  *2*0*32818401#9##
RX  *#*2*32818401#9*10*10*100*000*0##
```

} Send OpenWebNet command UP.
 } supervisor mode answer.
 } shutter reaches its upper limit.
 } level status all opened.

9.1.4 Send Request state

In this section we will send a request state command on shutter on unit 1 with his state is STOP:

```
TX  *#*2*32818401#9##
RX  *2*0*32818401#9##
RX  *#*1##
```

} Send OpenWebNet request.
 } ZigBee product answer.

9.2 WHAT table

ID	Action
0	STOP
1	UP
2	DOWN

9.3 WHAT id

9.3.1 STOP

Direction	OpenWebNet Frame	Description
client → server	*2*0*where#9##	where is the ZigBee product id <ul style="list-style-type: none"> • if unicast : where = MAC address in decimal + unit • if broadcast : where=0# + unit
client ← server	*#*0##	The command has not been sent
client ← server	*#*6##	Wait 500ms before sending the frame again.
client ← server	*#*0##	
client ← server	*#*1##	The command has been sent
client ← server	*2*0*where#9##	<p>If Supervision mode enabled, refer to section WHO = 13, WHAT = 66</p> <p>where = MAC address in decimal + unit</p>
client ← server	*#2*where#9*10*status*level*priority*info##	<p>where = MAC address in decimal + unit</p> <p>status : 10 = Stop</p> <p>level :</p> <ul style="list-style-type: none"> 0 = All closed 1-99 = Current position 100 = All opened 255 = Unknown position <p>priority :</p> <ul style="list-style-type: none"> 0 = Always 000 <p>info :</p> <ul style="list-style-type: none"> 0 = Always 0

9.3.2 UP

Direction	OpenWebNet Frame	Description
client → server	*2*1*where#9##	where is the ZigBee product id <ul style="list-style-type: none"> • if unicast : where = MAC address in decimal + unit • if broadcast : where=0# + unit
client ← server	*#*0##	The command has not been sent
client ← server	*#*6##	Wait 500ms before sending the frame again.
client ← server	*#*0##	
client ← server	*#*1##	The command has been sent
client ← server	*2*1*where#9##	If Supervision mode enabled, refer to section WHO = 13, WHAT = 66 where = MAC address in decimal + unit

9.3.3 DOWN

Direction	OpenWebNet Frame	Description
client → server	*2*2*where#9##	where is the ZigBee product id <ul style="list-style-type: none"> • if unicast : where = MAC address in decimal + unit • if broadcast : where=0# + unit
client ← server	*#*0##	The command has not been sent
client ← server	*#*6##	Wait 500ms before sending the frame again.
client ← server	*#*0##	
client ← server	*#*1##	The command has been sent
client ← server	*2*2*where#9##	If Supervision mode enabled, refer to section WHO = 13, WHAT = 66 where = MAC address in decimal + unit

9.4 DIMENSION table

ID	Action
10	Read position
11	Move to position

9.5 DIMENSION id

9.5.1 Read position

⚠ { This dimension can be send when the ZigBee shutter has done a calibration.
 Else the shutter returns 255 => Unknown position in field level

Direction	OpenWebNet Frame	Description
client → server	*#2*where#9*10##	where is the ZigBee product id where = MAC address in decimal + unit
client ← server	*#2*where#9*10*status*level*priority*info##	where = MAC address in decimal + unit status : 10 = Stop 11 = Up 12 = Down level : 0 = All closed 1-99 = Current position 100 = All opened 255 = Unknown position priority : 0 = Always 000 info : 0 = Always 0
client ← server	*#*1##	The command is received
client ← server	*#*0##	The command has not been sent
client ← server	*#*6##	Wait 500ms before sending the frame again.
client ← server	*#*0##	

9.5.2 Move to position

⚠ This dimension can be send if the calibration of ZigBee shutter has been done.

Direction	OpenWebNet Frame	Description
client → server	*#2*where#9*#11*level##	where is the ZigBee product id <ul style="list-style-type: none"> if unicast : where = MAC address in decimal + unit if broadcast : where=0# + unit level is the position of the shutter (0 - 100)
client ← server	*#*0##	The command has not been sent
client ← server	*#*6##	Wait 500ms before sending the frame again.
client ← server	*#*0##	
client ← server	*#*1##	The command has been sent

9.6 REQUEST

Direction	OpenWebNet Frame	Description
client → server	*#2*where#9##	where is the ZigBee product id where = MAC address in decimal + unit
client ← server	*#*0##	The command has not been sent
client ← server	*#*6##	Wait 500ms before sending the frame again.
client ← server	*#*0##	
client ← server	*2*state*where#9##	where = MAC address in decimal + unit state : depending from the state of the product unit 0 = Stop 1 = Up 2 = Down
client ← server	*#*1##	The command is received

10 Temperature WHO = 4

This field is used to receive temperature from the ZigBee probe.

10.1 Use cases

10.1.1 Receive positive temperature value from probe

In this section we will receive temperature from probe:

RX *#4*46085901#9*0*0247## } Receive temperature in celcius (24,7°C)

10.1.2 Receive negative temperature value from probe

In this section we will receive temperature from probe:

RX *#4*46085901#9*0*1032## } Receive temperature in celcius (-3,2°C)

10.2 WHAT table

10.3 WHAT id

10.4 DIMENSION table

ID	Action
0	Temperature level

10.5 DIMENSION id

10.5.1 Temperature level

△ In order to receive this frame the temperature probe must do previously a PnL procedure with OpenWebNet interface (see WHO=25 use cases for more details).

Direction	OpenWebNet Frame	Description
client ← server	*#4*where#9*0*level##	<p>where is the ZigBee product id</p> <p>where = MAC address in decimal + unit</p> <p>level:field is composed from 4 digits($C_1C_2C_3C_4$)</p> <p>$C_1 = 0$ positive temperature</p> <p>$C_1 = 1$ negative temperature</p> <p>C_2C_3 = are the temperature values(tens & units)</p> <p>C_4 = is decimal temperature values(0.1°C steps)</p>

11 Management WHO = 13

This field is used to manage the ZigBee network and products.

11.1 Use cases

11.1.1 Create ZigBee network

In this section the OpenWebNet interface will create a ZigBee network, make sure that the interface has not been in an existing network:

TX	*13*30*##	}	The network is created and opened.
RX	*#*1##		

11.1.2 Open ZigBee network

In this section the OpenWebNet interface will open a ZigBee network, make sure that the interface has been in an existing network:

TX	*13*32*##	}	The network is opened.
RX	*#*1##		

11.1.3 Product Opens ZigBee network

In this section the ZigBee product will open the ZigBee network:

RX	*13*32*47452500#9##	}	The network is opened by ZigBee Product (474525).
----	---------------------	---	---

11.1.4 Close ZigBee network

In this section the OpenWebNet interface will close a ZigBee network, make sure that the interface has been in an existing network:

TX	*13*31*##	}	The network is closed.
RX	*#*1##		

11.1.5 Product Closes ZigBee network

In this section the ZigBee product will close the ZigBee network:

RX	*13*31*47452500#9##	}	The network is closed by ZigBee Product (474525).
----	---------------------	---	---

11.1.6 Join ZigBee network

In this section the OpenWebNet interface will join a ZigBee network, make sure that the interface has not been in an existing network:

TX	*13*33*##	}	The interface has joined the ZigBee network .
RX	*#*1##		
RX	*13*32*47452500#9##		

11.1.7 Product Joins ZigBee network

In this section the ZigBee product will join the ZigBee network:

RX	*13*33*47452500#9##	}	The ZigBee Product (474525) is joining the ZigBee network .
----	---------------------	---	---

11.1.8 Leave ZigBee network

In this section the OpenWebNet interface will leave a ZigBee network, make sure that the interface has been in an existing network:

TX	*13*34*##	}	The interface has left the ZigBee network .
RX	*#*1##		

11.1.9 Product Leaves ZigBee network

In this section the ZigBee product will leave the ZigBee network:

RX *13*34*47452500#9## } The ZigBee product (474525) has left the ZigBee network .

11.1.10 Scan network

In this section the OpenWebNet interface will scan a ZigBee network, make sure that the interface has been in an existing network:

TX *13*65*## }
 RX *#*1## } The interface sent a broadcast in the ZigBee network.
 RX *#13*67*5## } 13 seconds after, the interface sent the number of products discover.

11.1.11 Supervisor

In this section we will send a supervisor frame in the ZigBee network, make sure that the interface has been in an existing network:

△ This command permits to send a ZigBee broadcast frame in order to inform ZigBee products (only active devices) that supervisor is present in network. The product will report to the supervisor their states when they are changing. Once supervisor mode is enabled, supervisor mode frames described in the document will be received whenever there is a status change in related device. Status changes can be generated by related OWN client->server frames or by any other reason (i.e., manual actuation on the device, scenario activation, ...) The OpenWebNet user cannot know if the interface is a supervisor, this mode is enabled only on each product. If a new product joins a network, you have to re-send this frame in order to enable this mode in it.

TX *13*66*## }
 RX *#*1## } The interface sent a supervisor frame in the ZigBee network.

11.1.12 Product Information

In this section we will send a product information command, make sure that the interface has been in an existing network:

The product information command could take 30 seconds if the product is not reachable, for example a battery powered device. This command permits to get the device id of active product (main powered device or battery powered who does not sleeping).

TX *#13*66#2## } The user wants to get infos about product index 2
 RX *#13*47485501#9*66#2*256## }
 RX *#13*47485502#9*66#2*256## } The product sends to the interface each endpoints and each device IDs.
 RX *#*1## } ACK at the end

or

TX *#13*47485500#9*66## } The user wants to get infos about product id "474855"
 RX *#13*47485501#9*66#2*256## }
 RX *#13*47485502#9*66#2*256## } The product sends to the interface each endpoints and each device IDs.
 RX *#*1## } ACK at the end

11.1.13 Device MAC address by index

In this section we will send a get where from index command, make sure that the interface has been in an existing network:

The product information command could take 30 seconds if the product is not reachable, for example a battery powered device, and this command ask to the database the product id and does not send ZigBee frame in order to reach it.

TX *#13*73#4## } The user wants to get the product id of index 4.

RX	*#13*47485500#9*73#4##	} The interface returns the product id of item index 4 .
RX	*#*1##	

11.2 WHAT table

ID	Action
12	Boot mode
22	Reset
30	Create
31	Close
32	Open
33	Join
34	Leave
60	Keep connect
61	Identify
65	Scan
66	Supervisor
67	Supervisor remove

11.3 WHAT id

11.3.1 Boot mode

Direction	OpenWebNet Frame	Description
client → server	*13*12*##	This command permits to enter in boot mode.
client ← server	*#*0##	NACK if interface can not enter in boot mode.
client ← server	*#*1##	ACK if interface enter in boot mode.
client ← server	STX 03600796 ETX	ACK in boot mode (ASCII protocol). The interface will not answer to the next OWN frames, just ASCII protocol frame.

11.3.2 Reset

Direction	OpenWebNet Frame	Description
client → server	*13*22*##	This command permits to reset the interface.
client ← server	*#*0##	NACK if can not be reset.
client ← server	*#*1##	ACK before reset.

11.3.3 Create

Direction	OpenWebNet Frame	Description
client → server	*13*30*##	This command permits to create ZigBee network.
client ← server	*#*1##	ACK if created.
client ← server	*#*0##	NACK if not created.

11.3.4 Close

Direction	OpenWebNet Frame	Description
client → server	*13*31*##	This command permits to close ZigBee network. To interface
client ← server	*##*1##	ACK if closed.
client ← server	*##*0##	NACK if not closed.
client ← server	*13*31*where#9##	From ZigBee product where is the ZigBee product id where = MAC address in decimal + unit

11.3.5 Open

Direction	OpenWebNet Frame	Description
client → server	*13*32*##	This command permits to open ZigBee network. To interface
client ← server	*##*1##	ACK if opened.
client ← server	*##*0##	NACK if not opened. NACK if a binding is in progress.
client ← server	*13*32*where#9##	From ZigBee product. where is the ZigBee product id where = MAC address in decimal + unit

11.3.6 Join

Direction	OpenWebNet Frame	Description
client → server	*13*33*##	This command permits to join ZigBee network. To interface
client ← server	*##*1##	ACK if the interface has joined the network.
client ← server	*##*0##	NACK if the interface has not joined the network.
client ← server	*13*33*where#9##	From ZigBee product. where is the ZigBee product id where = MAC address in decimal + unit

11.3.7 Leave

Direction	OpenWebNet Frame	Description
client → server	*13*34*##	This command permits to leave ZigBee network. To interface
client ← server	*##*1##	ACK if the interface has not left the network.
client ← server	*##*0##	NACK if the interface has not left the network.
client ← server	*13*34*where#9##	From ZigBee product where is the ZigBee product id where = MAC address in decimal + unit

client → server	*13*34*where#9##	To ZigBee product where is the ZigBee product id where = MAC address in decimal + unit
client ← server	*#*1##	ACK if command has been sent.

11.3.8 Keep connect

Direction	OpenWebNet Frame	Description
client → server	*13*60*##	This command permits to know if the interface is ready.
client ← server	*#*1##	ACK if ready.
client ← server	*#*0##	NACK if not ready.

11.3.9 Identify

Direction	OpenWebNet Frame	Description
client → server	*13*61*where#9##	This command permits to identify product. The green led blinking slowly during 5 min. where is the ZigBee product id unicast : where = MAC address in decimal + unit = 00
client ← server	*#*1##	The identify command has been sent
client ← server	*#*0##	The identify command has not been sent
client ← server	*#*6##	Wait 500ms before sending the frame again.
client ← server	*#*0##	

11.3.10 Scan

△ This command returns the number of products filled in the products database of the interface, and no the number of active products (ZR) seen during the scan.

Direction	OpenWebNet Frame	Description
client → server	*13*65*##	This command permits to scan the ZigBee network.
client ← server	*#*1##	The scan command has been sent
13 second after sent the command		
client ← server	*#13*67*value##	value is the number of products discovered in the ZigBee network.
client ← server	*#*0##	The scan command has not been sent

11.3.11 Supervisor

△ If a new product joins a network, you have to re-send this frame in order to enable this mode in it.

△ In order to guarantee maximum Zigbee network performance there must be only one OpenWebNet interface with supervisor mode enabled inside the ZigBee network. If there are more OpenWebNet interfaces, supervisor mode must be disabled in the rest of them. For development purposes it is possible to have more than one interface with supervisor mode enabled considering that higher the number of supervisors lower the performance of Zigbee network.

Direction	OpenWebNet Frame	Description
client → server	*13*66*##	Once supervisor mode is enabled, supervisor mode frames described in the document will be received whenever there is an status change in related device. Status changes can be generated by related OWN client->server frames or by anyother reason (i.e., manual actuation on the device, scenario activation, ...).
client ← server	*13*66*where#9##	From ZigBee product where is the ZigBee product id where = MAC address in decimal + unit
client ← server	*#*1##	The supervisor command has been sent
client ← server	*#*0##	The supervisor command has not been sent

11.3.12 Supervisor remove

Direction	OpenWebNet Frame	Description
client → server	*13*67*##	(Default mode)This command permits to not receive any change of state of products in the ZigBee network.This command is the complement to the supervisor command.
client ← server	*13*67*where#9##	From ZigBee product where is the ZigBee product id where = MAC address in decimal + unit
client ← server	*#*1##	The supervisor remove command has been sent
client ← server	*#*0##	The supervisor remove command has not been sent

11.4 DIMENSION table

ID	Action
12	Mac address
16	Firmware version
17	Hardware version
26	Who implemented
66	Product information
67	Get number of product in network
71	ZigBee channel
72	Battery information
73	Device MAC address by index

11.5 DIMENSION id

11.5.1 Mac address

Direction	OpenWebNet Frame	Description
client → server	*#13**12##	This command permits to have the IEEE address of the interface.
client ← server	*#13**12*value1*value2 *value3*value4*value5 *value6*value7*value8##	value1-8: IEEE address in decimal.
client ← server	*#*1##	ACK
client ← server	*#*0##	The command has not been sent

11.5.2 Firmware version

Direction	OpenWebNet Frame	Description
client → server	*#13*where*16##	where is the ZigBee product id <ul style="list-style-type: none"> if network unicast : MAC address in decimal + unit = 00 + #9 if interface : null
client ← server	*#13*where*16*value1 *value2*value3##	value1: is the firmware version. value2: is the release version. value3: is the build version where is the ZigBee product id <ul style="list-style-type: none"> if network unicast : MAC address in decimal + unit = 00 + #9 if interface : null
client ← server	*#*1##	The command has been sent
client ← server	*#*0##	The command has not been sent
client ← server	*#*6##	Wait 500ms before sending the frame again.
client ← server	*#*0##	

11.5.3 Hardware version

Direction	OpenWebNet Frame	Description
client → server	*#13*where*17##	where is the ZigBee product id <ul style="list-style-type: none"> if network unicast : MAC address in decimal + unit = 00 + #9 if interface : null
client ← server	*#13*where*17*value1 *value2*value3##	value1: is the major version. value2: is the minor version. value3: is the release version where is the ZigBee product id <ul style="list-style-type: none"> if network unicast : MAC address in decimal + unit = 00 + #9 if interface : null
client ← server	*#*1##	The command has been sent
client ← server	*#*0##	The command has not been sent
client ← server	*#*6##	Wait 500ms before sending the frame again.
client ← server	*#*0##	

11.5.4 Who implemented

Direction	OpenWebNet Frame	Description
client → server	*#13*where*26##	where is the ZigBee product id <ul style="list-style-type: none"> if network unicast : MAC address in decimal + unit + #9 (unit must be ≠ to 00) if interface : null
client ← server	*#13*where*26*value *valueN##	valueN: are the who which are implemented in this device. where is the ZigBee product id <ul style="list-style-type: none"> if network unicast : MAC address in decimal + unit + #9 if interface : null
client ← server	*#*1##	The command has been sent
client ← server	*#*0##	The command has not been sent
client ← server	*#*6##	Wait 500ms before sending the frame again.
client ← server	*#*0##	

11.5.5 Product information

The product information command could take 30 seconds if the product is not reachable, for example a battery powered device. This command permits to get the device id of active product (main powered device or battery powered who does not sleeping).

i.e. of using:

TX	*#13**66#2##	}	The user wants to get infos about product index 2
RX	*#13*47485501#9*66#2*256##		
RX	*#13*47485502#9*66#2*256##	}	The product sends to the interface each endpoints and each device IDs.
RX	*#*1##		

} ACK at the end

or

TX	*#13*47485500#9*66##	}	The user wants to get infos about product id "474855"
RX	*#13*47485501#9*66#2*256##		
RX	*#13*47485502#9*66#2*256##	}	The product sends to the interface each endpoints and each device IDs.
RX	*#*1##		

} ACK at the end

Direction	OpenWebNet Frame	Description
client → server	*#13**66#index##	index: the index of the product inside products database. The index started at 0.
	or	
client → server	*#13*where#9*66##	where = MAC address in decimal + unit(always equal to "00")
client ← server	*#13*where#9*66#index*value##	<p>where = MAC address in decimal + unit</p> <p>index: index of scanned product</p> <p>value: is the device ID of the product :</p> <p>Scenario :</p> <p style="margin-left: 40px;">scenario_control 2</p> <p>Lighting :</p> <p style="margin-left: 40px;">on_off_switch 256</p> <p style="margin-left: 40px;">dimmer_control 257</p> <p style="margin-left: 40px;">dimmer_switch 258</p> <p style="margin-left: 40px;">switch_motion_detector 259</p> <p style="margin-left: 40px;">daylight_sensor 260</p> <p style="margin-left: 40px;">scs_on_off_switch 261</p> <p style="margin-left: 40px;">scs_dimmer_control 262</p> <p style="margin-left: 40px;">scs_dimmer_switch 263</p> <p style="margin-left: 40px;">waterproof_1_gang_switch 264</p> <p style="margin-left: 40px;">automatic_dimmer_switch 265</p> <p style="margin-left: 40px;">toggle_control 266</p> <p style="margin-left: 40px;">scs_toggle_control 267</p> <p style="margin-left: 40px;">motion_detector 268</p> <p style="margin-left: 40px;">switch_motion_detector_II 269</p> <p style="margin-left: 40px;">motion_detector_II 270</p> <p style="margin-left: 40px;">auxilliary_toggle_control 271</p> <p style="margin-left: 40px;">scs_auxilliary_toggle_control 272</p> <p style="margin-left: 40px;">multifonction_scenario_control 273</p>

		on_off_control 274 auxiliary_on_off_1_gang_switch 275 Automation : shutter_control 512 shutter_switch 513 scs_shutter_control 514 scs_shutter_switch 515 Interface : scs_1_System_1-4_Gateway 1024 scs_2_System_1-4_Gateway 1025 network_repeater 1029 OpenWebNet interface 1030 Video : video_switcher 1536
client ← server	*#*1##	ACK when all product units are discovered
client ← server	*#13*where#9*66#index*0## *#*1##	The product was not reachable. value field will be equal to 0
client ← server	*#*0##	NACK if the command is not sent over ZigBee or Index is greater than the interface knows.
client ← server	*#*6 ##	Wait 500ms before sending the frame again.
client ← server	*#*0##	

11.5.6 Get number of product in network

△ This command returns the number of products filled in the products database of the interface.

Direction	OpenWebNet Frame	Description
client → server	*#13**67##	This command permits to get how many products are in the ZigBee network.
client ← server	*#13**67*value## *#*1##	value is the number of product discovered
client ← server	*#*0##	NACK if the command is not sent

11.5.7 ZigBee channel

Direction	OpenWebNet Frame	Description
client → server	*#13**71##	This command permits to know the ZigBee network channel.
client ← server	*#13**71*value## *#*1##	value : is the number of the channel. [11 - 26]
client ← server	*#*0##	NACK if the interface is not inside network

11.5.8 Battery information

△ The Battery information frame is visible when you pushed NETW and LEARN buttons on the sleepy end device. If users wants to receive this frame when users pushes the APB button, the device must do previously

a PnL procedure between its APB(s) and OpenWebNet interface (see WHO=25 use cases for more details).

Direction	OpenWebNet Frame	Description
client ← server	*#13*where#9*72*value##	<p>This command permits to know the battery level of battery device. This frame is seen when the battery device send a message to the interface.</p> <p>where = MAC address in decimal + unit</p> <p>value = 0 CRITICAL</p> <p>value = 1 POWER_VALUE_33</p> <p>value = 2 POWER_VALUE_66</p> <p>value = 3 POWER_VALUE_100</p>

11.5.9 Device MAC address by index

Direction	OpenWebNet Frame	Description
client → server	*#13**73#index##	<p>This command permits to get the where field (ZigBee ID) from the index of the products database.</p> <p>index: the index of the product inside products database. The index started at 0.</p>
client ← server	*#13*where#9*73#index*value## *#*1##	<p>where = MAC address in decimal + unit(always equal to "00")</p> <p>index: is the index of the product inside products database.</p> <p>value = 0 UNKNOWN</p> <p>value = 1 MAIN_POWERED_DEVICE</p> <p>value = 2 BATTERY_POWERED_DEVICE</p>
client ← server	*#*0##	NACK if index is unknown

12 Automation WHO = 18

This field is used to get energy management parameters on ZigBee products.

12.1 Use cases

12.1.1 Get Voltage

In this section we will send a dimension Voltage on ZR on unit 2:

TX	*#18*32818402#9*11##	}	get voltage on ZR I:328184.
RX	*#18*32818402#9*11*234##		
RX	*#*1##		

12.1.2 Get Current

In this section we will send a dimension Current on ZR on unit 2:

TX	*#18*32818402#9*17##	}	get current on ZR I:328184.
RX	*#18*32818402#9*17*2##		
RX	*#*1##		

12.1.3 Get Frequency

In this section we will send a dimension Frequency on ZR on unit 2:

TX	*#18*32818402#9*51##	}	get frequency on ZR I:328184.
RX	*#18*32818402#9*51*50##		
RX	*#*1##		

12.2 WHAT table

ID	Action
0	Reset

12.3 WHAT id

12.3.1 Reset

This command permits to reset the energy counter of ZigBee devices.

Direction	OpenWebNet Frame	Description
client → server	*18*75*where#9##	where is the ZigBee product id <ul style="list-style-type: none"> if unicast : where = MAC address in decimal + unit if broadcast : where=0# + unit
client ← server	*#*0##	The command has not been sent
client ← server	*#*6##	Wait 500ms before sending the frame again.
client ← server	*#*0##	
client ← server	*#*1##	The command has been sent

12.4 DIMENSION table

ID	Action
11	Voltage
17	Current
51	Energy
112	Frequency
113	Active Power
114	Active Power Total
115	Threshold Max Active Power
117	Reactive Power
1200	Report Power

12.5 DIMENSION id

12.5.1 Voltage

Direction	OpenWebNet Frame	Description
client → server	*#18*where#9*11##	where is the ZigBee product id unicast : where = MAC address in decimal + unit
client ← server	*#*0##	The command has not been sent
client ← server	*#*6##	Wait 500ms before sending the frame again.
client ← server	*#*0##	
client ← server	*#18*where#9*11*value##	unicast : where = MAC address in decimal + unit value : Voltage in decimal
client ← server	*#*1##	The command has been sent

12.5.2 Current

Direction	OpenWebNet Frame	Description
client → server	*#18*where#9*17##	where is the ZigBee product id unicast : where = MAC address in decimal + unit
client ← server	*#*0##	The command has not been sent
client ← server	*#*6##	Wait 500ms before sending the frame again.
client ← server	*#*0##	
client ← server	*#18*where#9*17*value##	unicast : where = MAC address in decimal + unit value : Current in decimal
client ← server	*#*1##	The command has been sent

12.5.3 Energy

Direction	OpenWebNet Frame	Description
client → server	*#18*where#9*51##	where is the ZigBee product id unicast : where = MAC address in decimal + unit
client ← server	*#*0##	The command has not been sent
client ← server	*#*6##	Wait 500ms before sending the frame again.
client ← server	*#*0##	
client ← server	*#18*where#9*51*value##	unicast : where = MAC address in decimal + unit value : Energy in decimal
client ← server	*#*1##	The command has been sent

12.5.4 Frequency

Direction	OpenWebNet Frame	Description
client → server	*#18*where#9*112##	where is the ZigBee product id unicast : where = MAC address in decimal + unit
client ← server	*#*0##	The command has not been sent
client ← server	*#*6##	Wait 500ms before sending the frame again.
client ← server	*#*0##	
client ← server	*#18*where#9*112*value##	unicast : where = MAC address in decimal + unit value : Frequency in decimal
client ← server	*#*1##	The command has been sent

12.5.5 Active Power

Direction	OpenWebNet Frame	Description
client → server	*#18*where#9*113##	where is the ZigBee product id unicast : where = MAC address in decimal + unit
client ← server	*#*0##	The command has not been sent
client ← server	*#*6##	Wait 500ms before sending the frame again.
client ← server	*#*0##	
client ← server	*#18*where#9*113*value##	unicast : where = MAC address in decimal + unit value : Active Power in decimal
client ← server	*#*1##	The command has been sent

12.5.6 Active Power Total

Direction	OpenWebNet Frame	Description
client → server	*#18*where#9*114##	where is the ZigBee product id unicast : where = MAC address in decimal + unit
client ← server	*#*0##	The command has not been sent
client ← server	*#*6##	Wait 500ms before sending the frame again.
client ← server	*#*0##	
client ← server	*#18*where#9*114*value##	unicast : where = MAC address in decimal + unit value : Active Power total in decimal
client ← server	*#*1##	The command has been sent

12.5.7 Threshold Max Active Power

Direction	OpenWebNet Frame	Description
client → server	*#18*where#9*115##	where is the ZigBee product id unicast : where = MAC address in decimal + unit
client ← server	*#*0##	The command has not been sent
client ← server	*#*6##	Wait 500ms before sending the frame again.
client ← server	*#*0##	
client ← server	*#18*where#9*115*value##	unicast : where = MAC address in decimal + unit value : Threshold Max Active Power in decimal
client ← server	*#*1##	The command has been sent

12.5.8 Reactive Power

Direction	OpenWebNet Frame	Description
client → server	*#18*where#9*117##	where is the ZigBee product id unicast : where = MAC address in decimal + unit
client ← server	*#*0##	The command has not been sent
client ← server	*#*6##	Wait 500ms before sending the frame again.
client ← server	*#*0##	
client ← server	*#18*where#9*117*value##	unicast : where = MAC address in decimal + unit value : Reactive Power in decimal
client ← server	*#*1##	The command has been sent

12.5.9 Report Power

Direction	OpenWebNet Frame	Description
client → server	*#18*where#9*1200#type*time##	where is the ZigBee product id unicast : where = MAC address in decimal + unit type = 1 only active power time : 0 to 255 seconds
client ← server	*#*0##	The command has not been sent
client ← server	*#*6##	Wait 500ms before sending the frame again.
client ← server	*#*0##	
client ← server	*#*1##	The command has been sent

13 Cen Plus WHO = 25

This field is used to know when user push an applicative button on ZigBee Product.

13.1 Use cases

13.1.1 Send Binding Request command

In this section we will do the sequence in order to receive a pushed button from user on ZigBee product on unit 1:

RX	*25*35*683585801#9##	}	User pushes LPB + APB on unit 1 in order to Open binding.
TX	*25*33*683585801#9##		
RX	*#*1##	}	Sends a Request in order to be aware when this button has been pushed.
RX	*25*36*683585801#9##		

13.1.2 Received command from bound device

△ In case of binding with Scenario ZigBee product, you will receive Short Pressure frame

RX	*25*21*683585801#9##	}	User pushes APB on unit 1 on scenario product .
TX	*25*33*683585801#9##		

△ In case of binding with lighting ZigBee product, you will receive Toggle frame

RX	*1*32*45645601#9##	}	User pushes APB on unit 1 on lighting product .
TX	*1*33*45645601#9##		

△ In case of binding with automation ZigBee product, you will receive Up/Down/Stop frames

RX	*2*2*45645601#9##	}	User pushes APB on unit 1 on shutter product .
TX	*2*3*45645601#9##		

13.1.3 Send Unbinding Request command

In this section we will do the sequence in order to stop to receive a pushed button from user on ZigBee product on unit 3:

RX	*25*35*683585803#9##	}	User pushes LPB + APB on unit 1 in order to Open binding.
TX	*25*34*683585803#9##		
RX	*#*1##	}	Sends a Unbinding Request in order to stop notifications of this button.
RX	*25*36*683585803#9##		

13.2 WHAT table

ID	Action
21	Short Pressure
33	Binding Request
34	Unbinding Request
35	Open Binding
36	Close Binding
37	Cancel Binding

13.3 WHAT id

13.3.1 Short Pressure

Direction	OpenWebNet Frame	Description
client ← server	*25*21*where#9##	where = MAC address in decimal + unit

13.3.2 Binding Request

Direction	OpenWebNet Frame	Description
client → server	*25*33*where#9##	where is the ZigBee product id where = MAC address in decimal + unit
client ← server	*#*0##	The command has not been sent
client ← server	*#*6##	Wait 500ms before send the frame again.
client ← server	*#*0##	
client ← server	*#*1##	The command has been sent

13.3.3 Unbinding Request

Direction	OpenWebNet Frame	Description
client → server	*25*34*where#9##	where is the ZigBee product id where = MAC address in decimal + unit
client ← server	*#*0##	The command has not been sent
client ← server	*#*6##	Wait 500ms before sending the frame again.
client ← server	*#*0##	
client ← server	*#*1##	The command has been sent

13.3.4 Open Binding

Direction	OpenWebNet Frame	Description
client ← server	*25*35*where#9##	where is the ZigBee product id where = MAC address in decimal + unit

13.3.5 Close Binding

Direction	OpenWebNet Frame	Description
client ← server	*25*36*where#9##	where is the ZigBee product id where = MAC address in decimal + unit

13.3.6 Cancel Binding

Direction	OpenWebNet Frame	Description
client ← server	*25*37*where#9##	where is the ZigBee product id where = MAC address in decimal + unite when two products are leaders of PnL binding.
client ← server	*25*37*##	when a binding was opened and never closed after 10 minutes

13.4 DIMENSION table

⚠ No Dimension Table for WHO = 25.

13.5 DIMENSION id

⚠ No Dimension ID for WHO = 25.