System Specification

Remote Commissioning

V 1.3

San Ramon, CA, USA, 2016

Executive Summary

Remote Commissioning represents the next part of the open standard of the EnOcean Alliance. This is the service layer definition for interoperable commissioning of devices. It defines the communication format and process between the commissioner and target device, which enables an independent development of these devices, but guarantying the highest possible level of interoperability between them. Additionally, a standardized device description file is introduced. This enables a flexible implementation of features and capabilities per devices, while enabling every commissioner to commission any device that supports Remote Commissioning.

Ver.	Editor	Change	Date
0.1	TM	Document compiled	March 26,2012
0.2	NM	Content reviewed, comments reviewed,	Jul 03, 2012
0.21	NM	Modified chapter 2.2.4; defined telegrams; text of 2.2 and 2.2.2 is NOT modified;	Jul 30, 2012
0.22	NM	Added chapters 2.4.1, 2.4.2; 2.4.3, 2.4.4 and 3; modified chapters 2.2.3 and 2.3;	Aug 16, 2012
0.23	NM	Comments by Isaac on chapters 2.4.3 and 2.4.4 integrated; modifications agreed Sep 19, 2012; editorial changes	Oct 07, 2012
0.24	Team	review	Oct 17, 2012
0.25	BE	Editorial changes – updated commands, security added. to be shared with TWG	
0.26	NM	Results from workshop May 16, 2013, Berlin; 2.2.1 Structure incorporated as per Isaacs input	May 16, 2013
0.27	BE	RPCs updated, added 2.4.5	May 22, 2013

0.28	BE	Heavy editing in configuration section, added rest to defaults RPC (2.4.3), document made more consistent in terminology, additional definitions created.	July, 9, 2013
0.29	BE	Additional clarification, terminology, and consistency updates. Teach RPC split into in and outbound. Removed ability to get config types	July 19th, 2013
0.31	BE	Clarification on Get configuration query response	Dec. 10, 2013
0.32	MH	Major review and content edit / add / remove.	Jan, 17 2014
0.33	MH	Added feedback from TWG NY	Jan, 31 2014
0.34	MH & BE	Added new commands	Feb, 02 2014
0.35	MH	Added diagrams	Feb, 04 2014
0.36	MH	Added: - Product ID beaconing - length to configuration setter and getter - Set commands are broadcast - Acknowledge is only send when message was unicast - Added message bundles - Added direction to link based configuration commands - Defined endianness and stateless communication	Mar, 20 2014
0.37	PT & MH	Added Selective repeating	Mar, 20 2014
0.38	TWG & MH	Major Review by: EnOcean, MSR, Vicos, BSC. Many corrections in text but no essential changes. Added: - Query Status response return codes, added reference to every command, removed explicit list. - Device description file and documentation file will be submitted separate as XSD.	May, 02 2014

1.00	TWG & MH	Adding – radio quality parameter, String parameter, Apply changes only optiona, changed beaconing mode	June, 13 2016
1.01	TWG & TM	RPC- Security addon for LinkTable - GET / SET Link table Commands	Nov, 29 2017
1.02	TM & MH	Adding information about using "SEC_MAN" telegrams for security functionality - GET / SET OWN security RPC - Secure session management - Information about validity of own security parameters	07.12.2017
1.1	MH	Updated figures, review.	5.4.2018
1.2	MH	Added Get query selective.	15.05.2018
1.3	MH	Review of Security addon for LinkTable to be more flexible and simple	13.12.2018

such loss or damage. All Company, brand and product names may be trademarks that are the sole property of their respective owners.

The above notice and this paragraph must be included on all copies of this document that are made.

The EnOcean Alliance Remote Commissioning (RECOM) Specification is available free of charge to companies, individuals and institutions for all non-commercial purposes (including educational research, technical evaluation and development of non-commercial tools or documentation.)

This specification includes intellectual property („IPR“) of the EnOcean Alliance and joint intellectual properties („joint IPR“) with contributing member companies. No part of this specification may be used in development of a product or service for sale without being a participant or promoter member of the EnOcean Alliance and/or joint owner of the appropriate joint IPR. EnOcean Alliance grants no rights to any third party IP, patents or trademarks.

These errata may not have been subjected to an Intellectual Property review, and as such, may contain undeclared Necessary Claims.

EnOcean Alliance Inc.
2400 Camino Ramon, Suite 375
San Ramon, CA 94583
USA
Graham Martin Chairman & CEO EnOcean Alliance

Table of Contents

1. Introduction

The EnOcean Alliance's EnOcean Equipment Profiles (EEP) does an excellent job of promoting interoperability between EnOcean devices during the normal network operation. The scope of EEPs is limited to data transmission and does not account for use cases beyond already configured and operating EnOcean networks.

A core function for the set up of an operating EnOcean network is the commissioning process, the linking and the configuration of devices, connecting sensors, actuators, and gateways to each other in order to achieve the desired functionality. EEPs define the telegrams for linking but not the process for configuring parameters. There are no standard telegrams or process for device configuration. This has led to fragmentation among devices and manufacturers that causes confusion and frustration among installers. Ultimately this can reduce the quality of the customer's experience and increase the capital and operating expenses of an EnOcean network.

Remote commissioning defines an interoperable interface and process to set-up, maintain and troubleshoot EnOcean devices-. Promoting a standard interface will promote ease of use for commissioning and performing maintenance on EnOcean networks. Implementing Remote Commissioning in EnOcean devices will reduce the complexity to install them, remove the need for physical access to configure them, and enable simple replacement of malfunctioning devices.

The commissioning process is currently fragmented among devices and manufacturers. This makes it extremely difficult for higher level software solutions to adequately control and configure devices. With Remote Commissioning's standard interface and processes the complexity of supporting unique EnOcean devices can be automated with software.

The focus of this systems specification is to provide guidance for product developers and managers of target and / or commissioning devices.

1.1. Definitions & references

1.1.1. Definitions

Add / Addition – an EURID and associated data are added into an inbound or outbound link table

Channel – channels are reserved for a single physical device that transmits multiples of the same data type, e.g. a dual channel temperature sensor.

Device Description file - a product specific file which contains the configurable parameters; could be split in two (or more) parts (e.g. file, parameter); optional;

Configuration Parameter - a remotely settable and gettable variable inside a product utilizing an EnOcean interface that affects the devices functionality;

Delete / Deletion – remove an entry (EURID and associated data) from the inbound or outbound link table

Download – from target device (e.g. actuator) to commissioning device (e.g. installation tool)

EEP - **E**nOcean **E**quipment **P**rofile; Specification to define structure of over-the-air data bytes for EnOcean transmitters. Also see Generic Profiles.

EURID – a unique identification number assigned hard coded to every EnOcean module during the production process

Inbound – receiving device which receives e.g. a teach-in request message; Inbound also refers to the incoming communication direction.

Outbound – device which can transmit EnOcean telegrams, e.g. a teach-in request. Outbound also refers to the outbound communication direction.

Link – a logical connection between devices that accept each others communications either unidirectionally or bidirectionally. The Link does not describe the content of the exchanged data. A link expresses only that a data exchange can occur. A link is the result of the learn process.

Link table – a table which describes the logical application links between EnOcean devices .The table lists all IDs, EEP or GPs, channel and other values of entries which are taught in a particular device (inbound) or contains all the devices which that particular device is taught into (outbound); the table resides in a device (e.g. actuator);

Link table entry - a single row in the inbound or outbound link table, representing a logical connection with entered device

Teach process (learn process) – process of creation / deletion of logical connection between EnOcean devices. The result of teach process can be teach-in or teach-out. The process consists of:

- an inbound device entering learn mode and accepting teach requests
- any number of other devices sending teach requests

Teach-in – process to connect logically / link two devices into uni or bi-directional connection; telegram details defined in related specifications (e.g. EEP2.x)

Teach-out – process to disconnect logically / unlink two devices; telegram details defined in related specifications (e.g. EEP2.x)

Local teach - transmitting a teach telegram is performed by pressing the teach button locally on the device (inbound and/or outbound) and then exchanging teach message(s). This is the common way of performing the learn process without any commissioning.

Remote teach - the learn process is performed by exchanging teach messages similar to local teach, but there is no need to press a button on the device. The teach functions is triggered remotely from a commissioning device (inbound and/or outbound).

Direct teach process – the learning process is performed by directly modifying link tables on inbound and outbound target devices by a commissioning device.

Target Device – This is an EnOcean device that is undergoing the commissioning process via remote commissioning. For example, a device that has its link tables and parameters being modified.

Commissioning Device – this is an EnOcean capable device that is transmitting remote commissioning telegrams to the Target Device; e.g., a commissioning tool

Radio Link Test - Testing the system radio link by verifying each radio communication channel quality.

Remote Management – For content and functionality of EnOcean's Remote Management specification refer to [1]. The remote commissioning specification is built on top of this. Remote Management defines the base for interoperable commissioning.

Commissioning - all the operation involved in the management of target devices, setting up nodes for the first time, daily maintenance, troubleshooting, changes in field, etc.

Remote Commissioning - the application of the EnOcean Remote Management functionality by defining new standardized RPCs. The process of commissioning target devices without requiring physical access to the device. This can be accomplished in an interoperable way with the Remote Commissioning interface and process defined in this document.

Triggered Locally – The action is started by acting physically on the device (i.e.: pressing a button) without a commissioning device

Triggered Remotely - The action is performed using Remote Commissioning RPCs.

Upload – from a commissioning device (e.g. installation tool) to a target device (e.g. actuator)

Device Description File – an xml file that details the operational modes, parameters, and compatible application profiles of a Remote Commissioning enabled device.

Product ID – A 6 byte value where the two most significant bytes are the Manufacturer ID and the least four significant bytes are a manufacturer defined product identifier. This is used to link a target device to its device description file.

1.1.2. References

[1] EnOcean Remote Management Specification. https://www.enocean-alliance.org/remote-management

[2] EnOcean Equipment Profiles 2.x (Newest available) http://www.enocean-alliance.org/eep/

[3] EnOcean Wireless Standard https://www.enocean-alliance.org/what-is-enocean/enocean-wireless-standard/

[4] Security of EnOcean Radio Networks, https://www.enocean.com/security-specification

[5] Device Description File and Documentation Structure – XSD and XML Example

1.2. Remote Management Overview

This chapter summarizes the usage of Remote Management (ReMan). The detailed specification of ReMan can be found in [1].

ReMan enables a platform for EnOcean devices to be accessed and configured remotely. Remote Management defines the abstract structure of the commands and some basic rules. Part of Remote Management is also a basic set of standardized and mandatory commands called Remote Management Control Commands (RMCC). Also the abstract structure of a Remote Procedure Call (RPC) is defined. This generic structure can be used to define commands to perform tasks. In this document we focus on standardized commands to commission, configure and maintain remote EnOcean devices. In other words, Remote Management is the underlying specification we are using to enable Remote Commissioning.

ReMan can be implemented for both the EnOcean Radio Protocol (ERP) or for the serial interface (e.g. ESP3) stack.

According to the structure of the OSI reference model, ReMan is implemented on the Transport and Application layer of the ERP stack as shown in Figure 1.

To enhance ReMan and Remote Commissioning for the future, security maintenance telegrams have been introduced. It is highly recommended that all new devices should be based on secured communication and security should be the new default operation mode. Especially since unsecure communication allows it easily for third party to manipulate or misuse configurations. Please refer to chapter 1.3. and document [4] for additional information.

Layer	Services	Data Units
Application	EEP / Generic Profiles, ReMan RPC handling	DATA
Presentation		Raw DATA
Session		
Transport	ReMan Telegram chaining Or Security Maintenance Telegrams	TELEGRAM/ MESSAGE
Network	EnOcean Radio Protocol	TELEGRAM
Data Link Layer	EnOcean Radio Protocol	SUBTELEGRAM
Physical	EnOcean Radio Protocol	BITS / FRAME

Figure 1: ReMan layers in the EnOcean Radio Protocol stack (OSI)

ReMan communication is based on data units called MESSAGES.

The implementation of MESSAGES as data units enables the exchange of long payloads, for instance the transmission of the content of a flash page. The MESSAGE implementation is realized on the transport layer of the radio stack and is composed of chained SYS_EX telegrams as shown in Figure 2.

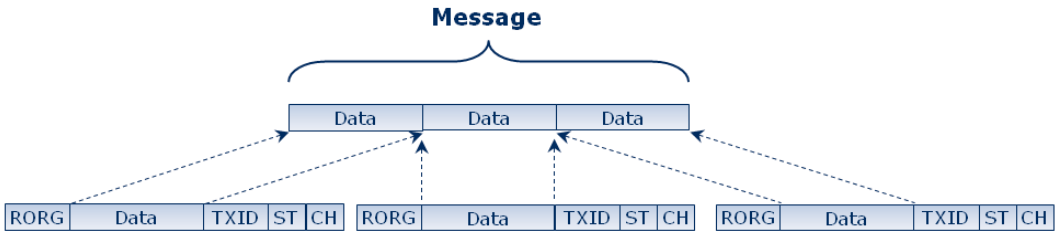

Figure 2: Chaining of SYS_EX telegrams on the transport layer of ERP1

The interpretation of the MESSAGE data content is done in the application layer of the radio protocol. The interpreted payload is called a COMMAND. There are two groups of ReMan COMMANDS:

- Remote Management Control Commands (RMCC) – provide the basic functionality to locate, identify, authenticate, and query the status of a device. RMCCs are mandatory implemented in every ReMan device.

- Remote Procedure Calls (RPC) – provide the definition of a specific function that can be called remotely. These commands are device and manufacturer specific and are optional.

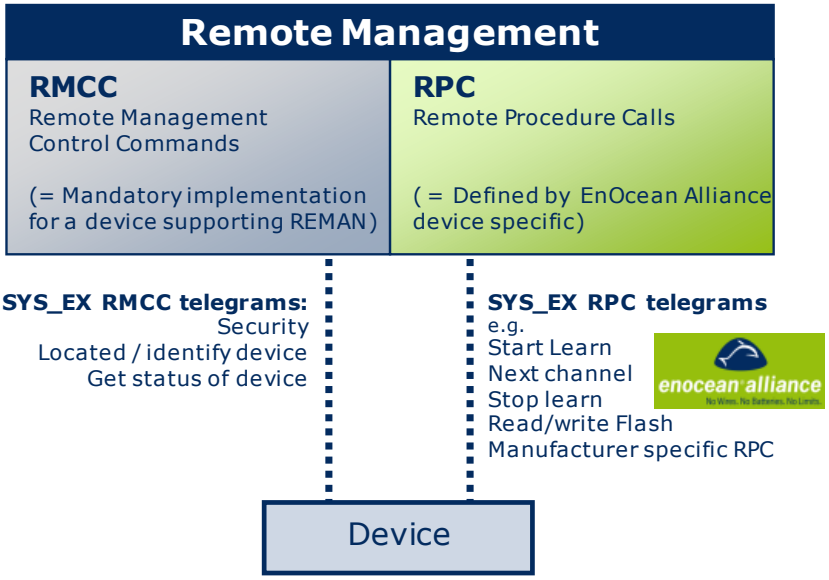

Figure 3: Interpretation of ReMan commands on the application layer

The detailed definition of RMCC can be found in [1]. RPC definition of some commands can be found in [2].

1.2.1. ReMan Functionality defined in RMCC

This chapter summarizes the RMCC functionalities. The detailed definition of RMCC can be found at [1]. According to the use-case the RMCC can be sorted in the following groups:

Security – Not used for ReMan Secure

- LOCK – locks the ReMan interface of the device. Without providing the correct security key via the UNLOCK command there is no response to any RMCC or RPC except PING.
- UNLOCK – unlocks the ReMan interface if the security key for the device is correct
- SETCODE – sets the security key for unlocking the ReMan interface. It is recommended to set the security key to something unique during commissioning.

Session Management – Used for ReMan Secure

- Start Session – opens the session for a ReMan controller.
- Close Session – closes the session for a ReMan controller

Locating/identifying a device

- ACTION – remotely triggers a manufacture specific action on a device, for instance a light controller would flash the light or a shades controller would move the shades up and down. With this function, a remote device with a known ID can be physically located and what it controls can be easily determined.
- PING – tests the radio signal strength to a remote device. The remote device sends a response to the PING command containing the radio signal strength of the received request.
- QUERY ID – tests if a device is functioning and unlocked. If this command is sent as a broadcast all unlocked devices respond with their ID + EEP.

Receiving the status of a device

- Query STATUS – requests the status info of the remote device. The response to this command contains information about the security status, the function code of the last command and the result of the last command.
- QUERY FUNCTION – requests the supported list of RPC. The response to this command contains the list of supported RPC functions by the REMOTE device.

Look at the use case overview for Remote Management and it parts. You can see on the picture, that all functionality besides the RMCC is classified as extended functionality. Remote commissioning is one of these extensions.

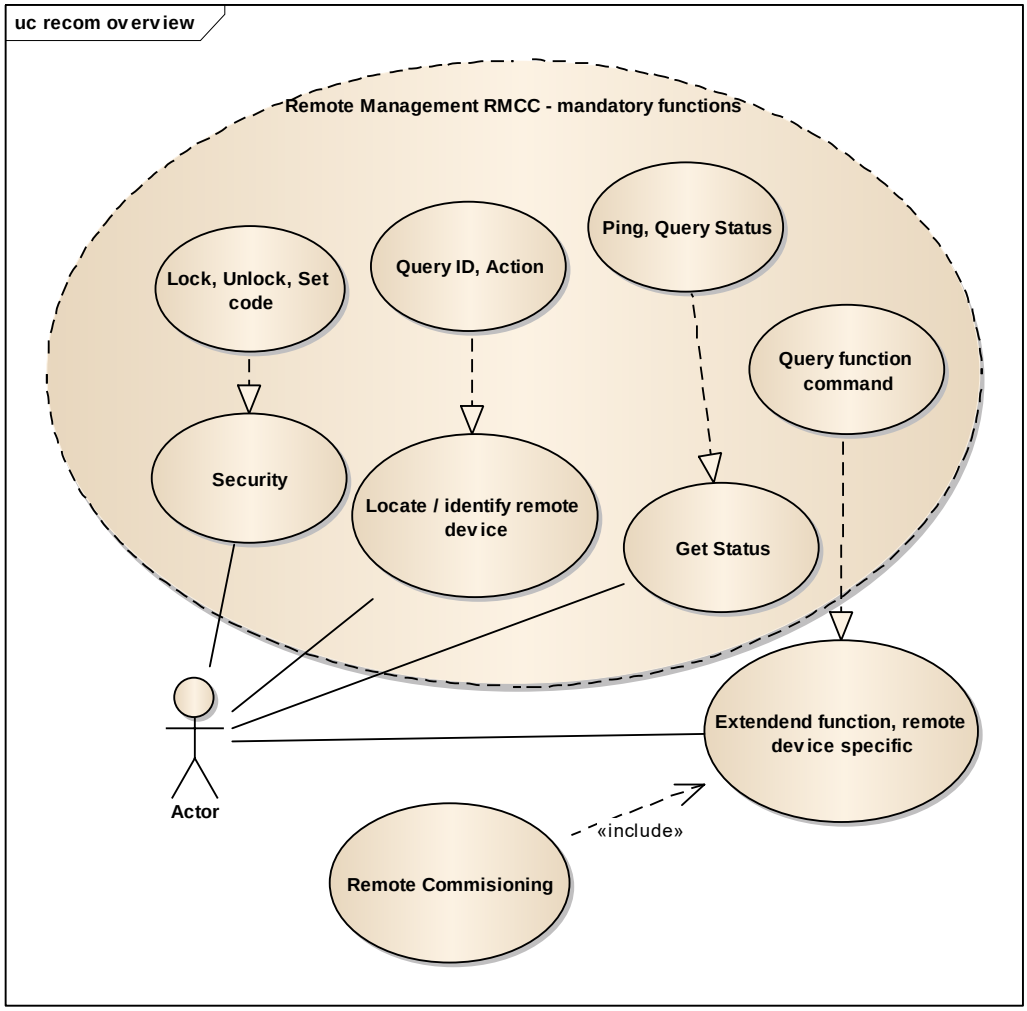

Figure 4 Remote Management overview with Remote Commissioning

1.3. Remote Commissioning and enhanced security

The EnOcean radio standard offers a security layer for encrypting and authentication Data. It is highly recommended to use it for Remote Commissioning and ReMan communication. SEC_MAN (Secure Maintenance) messages are used and allow a finer granularity of access rights. The need for a LOCK and UNLOCK code are not needed and two additional RMCC are defined. These 2 commands and the Message format for secure maintenance telegrams are defined in [4] . Especially for all commands which modify security information, it is necessary that secured messages are used, and the commands which modify secure parameter in a link table or keys of a device should not be used in non secure way.

1.4. Remote Commissioning Overview

Remote Commissioning uses the available Remote Management technology to enable a standardized way of commission EnOcean devices. Its essential part consists of defining the corresponding RPC commands and processes (e.g. message flow, constraints, etc.). This summary describes briefly the Remote Commissioning Specification, which is available in chapter 2.

Remote Commissioning will be used

- During commissioning of newly installed devices, with factory settings, EnOcean networks to perform first set-up of inbound and outbound links and configurations. - *Commissioning a new Installation*
- When modifications are done on functioning EnOcean networks, by adding and removing devices and/or changing devices' configuration parameters. - *Adjustments in a Live Network.*
- When replacing a non-operating device with a pre-commissioned, ready to install one. - *Service in Case of Device Failure.*
- When trouble shooting an operating EnOcean network. - *Maintenance of a Live Network.*
- To document and store the existing installation and device configurations

Each use case may need a different set of processes and functions to be supported by the installed devices. In general we define these steps:

- Common configuration steps, maintenance and service
  - E1 → Upload link table
  - E2 → Teach-in / teach-out of EnOcean devices
  - E3 → Edit link table data (add, delete, modify)
  - E4 → Download link table

- Application specific configuration, maintenance and service
  - A1 → Upload of application specific data
  - A2 → Download of application specific data

- Documentation
  - D1 → Download of link tables
  - D2 → Download of application data

In the figure below you can see the use case summary of Remote Commissioning.

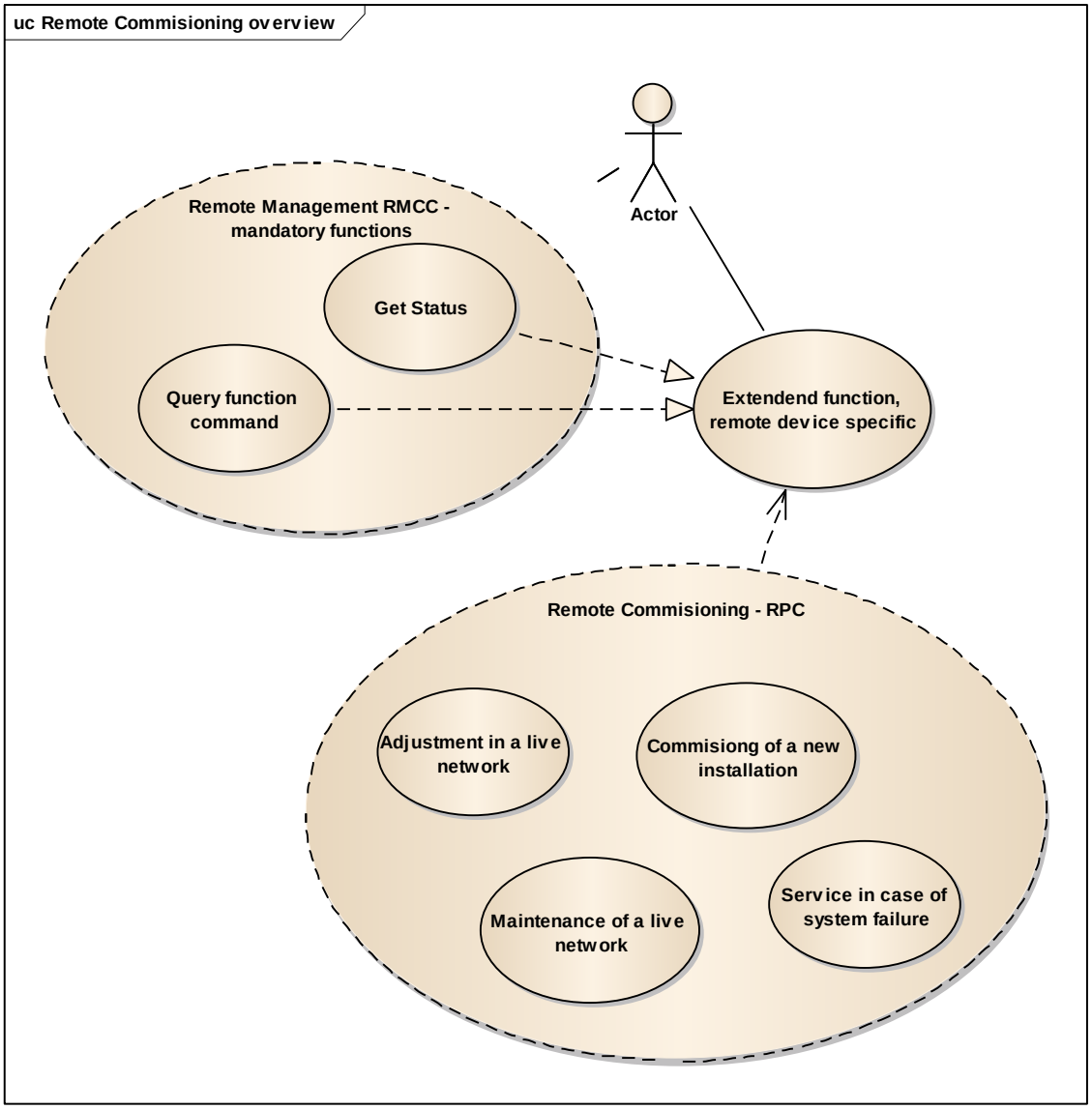

Figure 5 Remote Commissioning overview

Two particular RMCCs are shown in the use case diagram:

- Get Status / Query Status– this RMCC can be executed after every command. The device responses with the return code of the previous executed command. For the list of return codes please refer to A 2 Query Status return codes.
- Query function command – this RMCC is executed at the beginning of commissioning where all supported RPC are returned as response

For detailed information about telegram structures look at the Remote Management Specification [1]. The following chapters describe the requirements and use cases which were used as reference to develop the specification. In each use case is described how the above described steps are applied.

1.4.1. Commissioning a New Installation

Scenario:

Several EnOcean devices are installed and powered, but not commissioned. The installation area may consist of several rooms. Many of these rooms have a similar layout, e.g. in a hotel, resulting in repeating the same commissioning procedure several times with different target devices.

Assumptions:

System design is available and documented (plans, drawings, descriptions, tables or similar). The commissioning team is familiar with the EnOcean devices and knows the systems intended functionality.

EnOcean devices are brand-new, pre-installed, wired or pre-tested as a sub-system. However, the EURIDs and configuration of devices may be unknown.

The commissioning activity can be split into the following steps:

- Pairing of devices for all communication links in question, including bi-directional links (one to one, one to many relations), refer to E1 to E4
- Configuring application related parameters of the device, refer to A1, A2
- Testing the system radio link, by verifying each radio communication link.
- Creating the documentation required as defined in D1 and D2. The full documentation has to fulfill the minimum requirements of a site acceptance test (refer to *3. Documentation of Commissioning Results*).
- Preferably, a device is identified by a label reflecting its EURID.

Output of process:

- Documentation of the system as specified and installed.
- A running system as specified.

This use case also covers major modifications to existing networks, e.g. due to change of building space usage.

1.4.2. Adjustments in a Live Network

Scenario:

The scenario includes changing the control parameters of intelligent actuators, e.g. thresholds and timers in local HVAC equipment. For example the modification can be required due to any adjustments of the application and / or the physical arrangement of the rooms.

This use case may affect the common configuration as well as the configuration of the application.

This use case may apply functionalities E1 to E4, A1, A2, D1 and D2.

1.4.3. Maintenance of a Live EnOcean Network

Scenario:

This use case is dedicated to maintaining the reliability of a live network. The aim is to identify problems that might come up in the network without maintenance. Thus, this is an optional use case.

Please note: This use case is not defined by this specification.

1.4.4. Service in Case of Device Failure

Scenario:

The sole objective of this use case is the replacement of a nonfunctioning EnOcean device. Two different approaches are defined:

1. An EnOcean device which executed outbound teach-in is not functioning.

There are two options:

- update the various devices inbound link tables to reflect this new ID – remote triggered indirect teaching
- teach-in a new device and remove the invalid one – remote triggered direct teaching

In both cases the EEP(s) of the previous device must be used. Additionally the application configuration needs to be restored in the new hardware.

This use case applies functions E2 thru E4, A1 and D1 (see 1.4.1).

2. An EnOcean device which utilizes inbound teach-in is out of operation:

In this case the faulty device is replaced with a new working one that has been commissioned using the documented and previously stored Link Tables and application configuration.

This use case applies functions E1, A1 and D2 (see 1.4.1).

1.5. Core Requirements

From the use cases described in section 1.4. the following requirements are deducted.

REQ 01. Every device should to be identifiable with its unique EURID in the network by radio and label.

REQ 02. Links / connection between devices must be documentable and editable (add/change/delete).

REQ 03. Device configuration parameters must be commissionable and documentable.

REQ 04. A device which replaces a previous device (e.g. because of malfunction) must be configurable by commissioning to match the functionality of the failed device.

REQ 05. A device must be commissionable in field and also pre-commissionable on and off site of the installation.

REQ 06. Installed devices functionality must be configurable after installation.

REQ 07. A Factory Reset to defaults should be available.

REQ 08. The radio link test functionality should be possible by commissioning.

REQ 09. A device supporting remote commissioning must be commissionable without physical access to it.

1.6. Conforming Implementation

The Remote commissioning specification presents an interoperable way of commissioning any device. To ensure interoperability between various target and commissioning devices in the Alliance, Remote Commissioning features can be certified.

The EnOcean Alliance certification specification describes how the functionality will be tested. The tests main aim is not to functionally test a device's implementation of functionalities or other device specific features, but to ensure compatibility in the interoperable Eco system of commissionable targets and commissioning tools. The target device and commissioned device can be developed by different companies and people. This makes certification on the Remote Commissioning interface important to maintain interoperability by EnOcean devices.

Devices fulfilling the certification will carry a certified logo of the remote commissioning certification program. This way it will be presented that this devices is a certified part of the Eco system. Certification details and further description on this topic will be listed in the Remote Commissioning certification specification.

1.7. Design constrains

In this chapter we describe the additional constrains which were considered during design of the protocol.

1.7.1. Endianness

The value coding of the parameters setters and getters are according the Generic Profiles Protocol. Therefore also the endianness from Generic Profiles applies – which is big-endian. The most significant byte comes first.

The endianness of all other telegram fields is according to the EnOcean Radio Specification 1 and 2, which is also big-endian. The most significant byte comes first.

1.7.2. Stateless communication

Communication between commissioning device and target device is stateless. This means actions are defined as set or query action and then an optional response. The response of the next action is not depended to the action before.

Additionally, the response to a query includes complete meta-data description of the provided data. This means for example, if querying a parameter the index and size of the parameter is provided in the response with the actual value, although the commissioning device knows which parameter was requested. This additional mechanism should ensure consistency of data transmitted.

2. Remote Commissioning

This chapter includes the complete interface and process specification. This is structured into these three application areas:

- Remote Learn process
- Link Table functions – process and interface for managing link table
- Configuration – process and interface for configuring parameters

Prerequisites

A target device must be unlocked and currently accepting Remote Management Commands. The Device Description will list the supported features and parameters. Supported commands can be fetched with Query Function command defined in Remote Management.

2.1. Remote Commissioning Acknowledge

Not all the Remote Configuration Messages return data responses. These data replies are considered explicit acknowledges that the message was received and processed successfully. For the messages that do not have explicit responses the Remote Commissioning Acknowledge Response was developed. This response should be sent anytime a command that does not have a paired response executes successfully, e.g. Set Link Table Data, and the request was addressed to the target device. If the Remote Commissioning Acknowledge is not received it is assumed the previous RPC was not executed successfully. The format of this message is below.

Remote Commissioning Acknowledge	
Function code	0x240
Manufacturer Id	0x7FF
Data length	N/A
Addressed	No
Broadcast	Yes

Table 1 Remote Commissioning Acknowledge

In the case where a Remote Commissioning Acknowledge is expected but not received the Query Status Remote Management RMCC can be used to help determine the reason for failure.

2.2. Learn processes with remote commissioning

The traditional learn process consists of two distinct parts. An inbound device must enter into learn mode, a mode where it will accept new links via teach in/out telegrams, and an outbound device must transmit its teach telegram. Without remote commissioning a teach message is always triggered locally, which means the user must have physical access to the target devices

and utilizes a physical interface to transmit a teach message, i.e. a teach button on a sensor. The same is true for putting an inbound device, e.g. an actuator, into the learn mode. The teach process, when triggered locally, is the typical process used to create links in an EnOcean network and is not in the scope of this specification. It is defined in the application layer description of the EnOcean wireless standard [3].

Remote commissioning removes the need for physical device access, and thus on device button presses for the inbound and outbound cases. It is not required for the outbound device to transmit, or even be present, to be learned into an inbound device.

The Teach-in and Teach-out messages are two specific parts of the Learn process, which creates and removes logical radio links between devices on the application layer.

In this specification we will focus on the remote teach process, we differ between:

- Remote learn process -

  This process is similar to the local teach process, the target devices functionality is the same. Teach process is executed by teach message exchange. Remote management provides only the trigger for this action. The telegrams do not differ from the standard local teach messages.

  This process has following applications:

  - Remote teach process inbound (in/out) (e.g. start learn mode on an inbound device)
  - Remote teach process outbound (in/out) (e.g. trigger the transmission of a teach-in request)

  For physical devices this means that learn mode on a device is triggered remotely and the transmission of a teach-in request is also triggered remotely on a second device. The remote teach process on one device (inbound or outbound) can be combined with the local teach process on the other device (inbound or outbound). For example the learn process on an actuator was started by remote commissioning but the teach-in request from the sensor was triggered locally by pressing the teach-in button.
  After performing the remote teach process the link tables on the respected devices are modified according to the result of the teach process.

- Direct learn process

  This process utilizes the direct access to the actual link tables and edits them through a commissioning device. This process has following applications:

  - Direct teach process inbound (in/out). (e.g. add a complete row entry to the inbound link table)
  - Direct teach process outbound (in/out)

2.2.1. Remote teach process

As stated in section 2.2. we discuss two options:

- Remote teach process inbound (in/out)
- Remote teach process outbound (in/out)

Remote teach process inbound (in/out)

An inbound device that supports remote teach-in/-out can receive a teach request telegram from multiple outbound devices and add them into its inbound link table. When set into a teach-in/-out mode, inbound devices shall receive and process teach-in/-out request messages and may answer them using a teach reply message according the defined application layer of the EnOcean standard [3].

With remote teach we define on an inbound device an explicit:

- Learn in mode – only teach-in requests from outbound devices are processed.
- Learn out mode - only teach-out requests from outbound devices are processed.

The inbound device can be only in one mode at a given moment, not both at the same time.

If the teach request message from the outbound device states an explicit teach-in or teach-out, then it must match with the inbound device learn mode, otherwise the message is ignored.

If the teach request message from the outbound device is a toggle message this means, that it is not explicit requesting teach-in or teach-out, then the learn mode of the inbound device defines the outcome of this situation. Example: If a magnet window contact sensor sends an 1BS teach request and the receiver is in learn-in mode, then the result is that the receiver learns-in the sensor. If the sensor sends another teach request nothing happens, because the receiver is still in learn-in mode. To learn out the sensor the receiver must be set to learn-out mode and then the sensor must transmit a teach request.

The remote teach process requires an outbound and inbound device to be functional at commissioning time. In an example case the inbound device is put into learn mode remotely by a managing device and the outbound device is then locally or remotely triggered to send a teach telegram

The following steps are required to complete a remote inbound teach-in work flow:

(1) upload link table to the inbound device (optional)

(2) start explicit learn-in mode for one line item of link table

(3) trigger device that shall be learned-in, either local or remote to send teach request

(4) disable learn-in mode at the inbound device

(5) download link table from the inbound device (optional)

Remote teach process outbound (in/out)

Outbound remote teach process only replaces the action of pressing the teach button on the outbound device. All the following actions and communications are identical to the locally triggered outbound teach process.

To perform an outbound remote teach a device could have an outbound link table which will contain the EURIDs of the devices to which it is linked with. In order to maintain link table integrity the below steps must be followed:

(1) Set the inbound device to learn mode on and in the correct index (local or remote)
(2) Trigger the outbound teach-in remotely.
(3) Verify that the EURID of the inbound device has been added to the outbound link table of the outbound device.
(4) Set the inbound device to learn mode off (local or remote).

Process description

Please find in figure below the complete process description of remote teach process with remote inbound and outbound. In the description also the possibility of local LRN button operation is included.

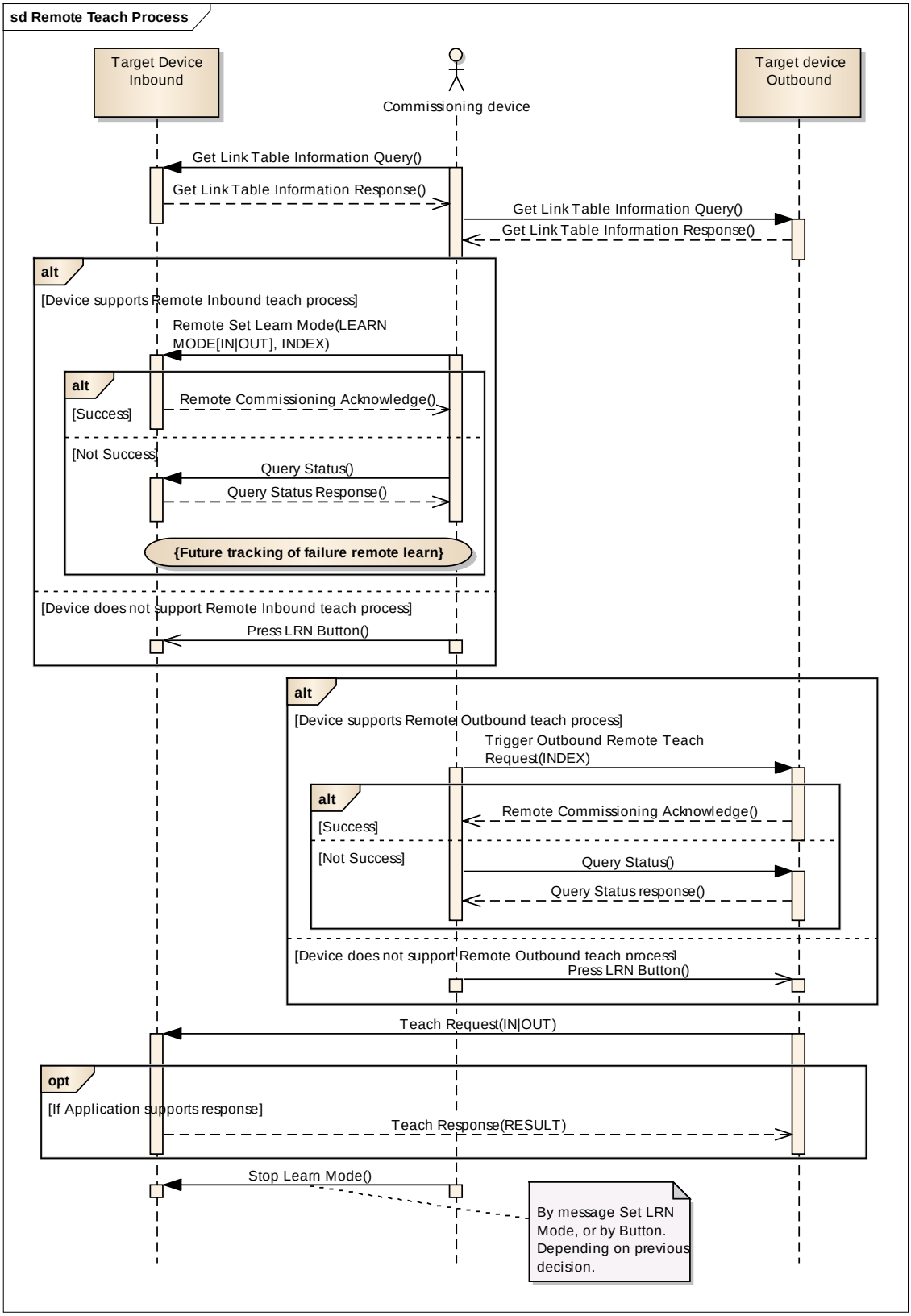

Figure 6 Remote Teach Process

2.2.2. Direct Teach process

Direct Teach is the most efficient method of teach process but requires additional information about the devices to be added to the link tables, e.g. their EEPs and IDs. As stated in section 2.2. we discuss two options:

- Direct teach process inbound (in/out)
- Direct teach process outbound (in/out)

Direct teach process inbound

This method is the simple uploading of link tables to the inbound device from the commissioning device. The outbound device does not have to be accessible or operating to be linked to the inbound device.

Direct teach process outbound

This method is the simple uploading of link tables to the outbound device from the commissioning device.

The EEP and channel of each row of the outbound link table must match with a row of the inbound link table (of an inbound device) where the EURID of the devices is the unique reference to ensure link table integrity and consistency between the linked devices. This means the ID of the outbound device is listed in the inbound table of the inbound device and the inbound devices ID is listed in the outbound table of the outbound device.

When the outbound device is unidirectional and/or the application protocol is unidirectional the outbound device will not update the outbound link table because it is unaware of the ID, EEP, and channel of the inbound device and the outbound device may have no outbound link table at all. Only when linking bidirectional devices within a bidirectional application it is possible to do this automatically.

The Link Table can have optional to EEP an GP Entry. EEP and GP entries are equal in hierarchy and meaning, only the representation is different. Also different commands of handling GP entries are applied. Support of GP entries is optional.

Process description

The Direct teach process consist of two essential parts:

- Getting the current Link tables from the inbound and outbound device– if they are not known
- Modifying the link tables of the Inbound and Outbound devices which are having their link tables modified

The process of getting the complete inbound and outbound Link Table is shown below. It includes all possibilities of the command being applied. Not all work flows will be required.

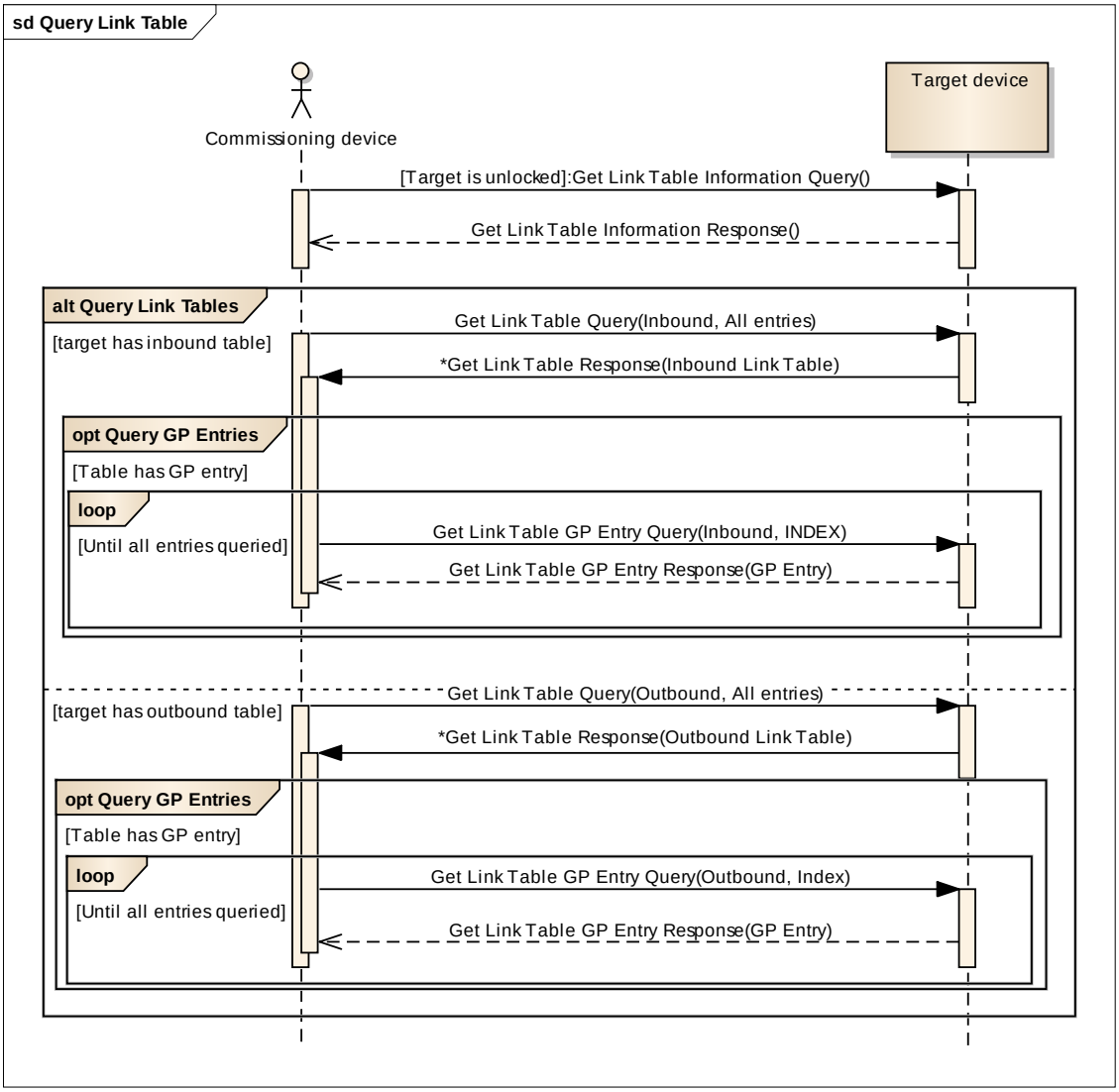

Figure 7 Query Inbound & Outbound Link Tables

If the message payload extends the radio telegram payload then remote management telegram chaining is applied. This is not explicitly included in the process description.

In Figure 8 the Change Link Table process is listed.

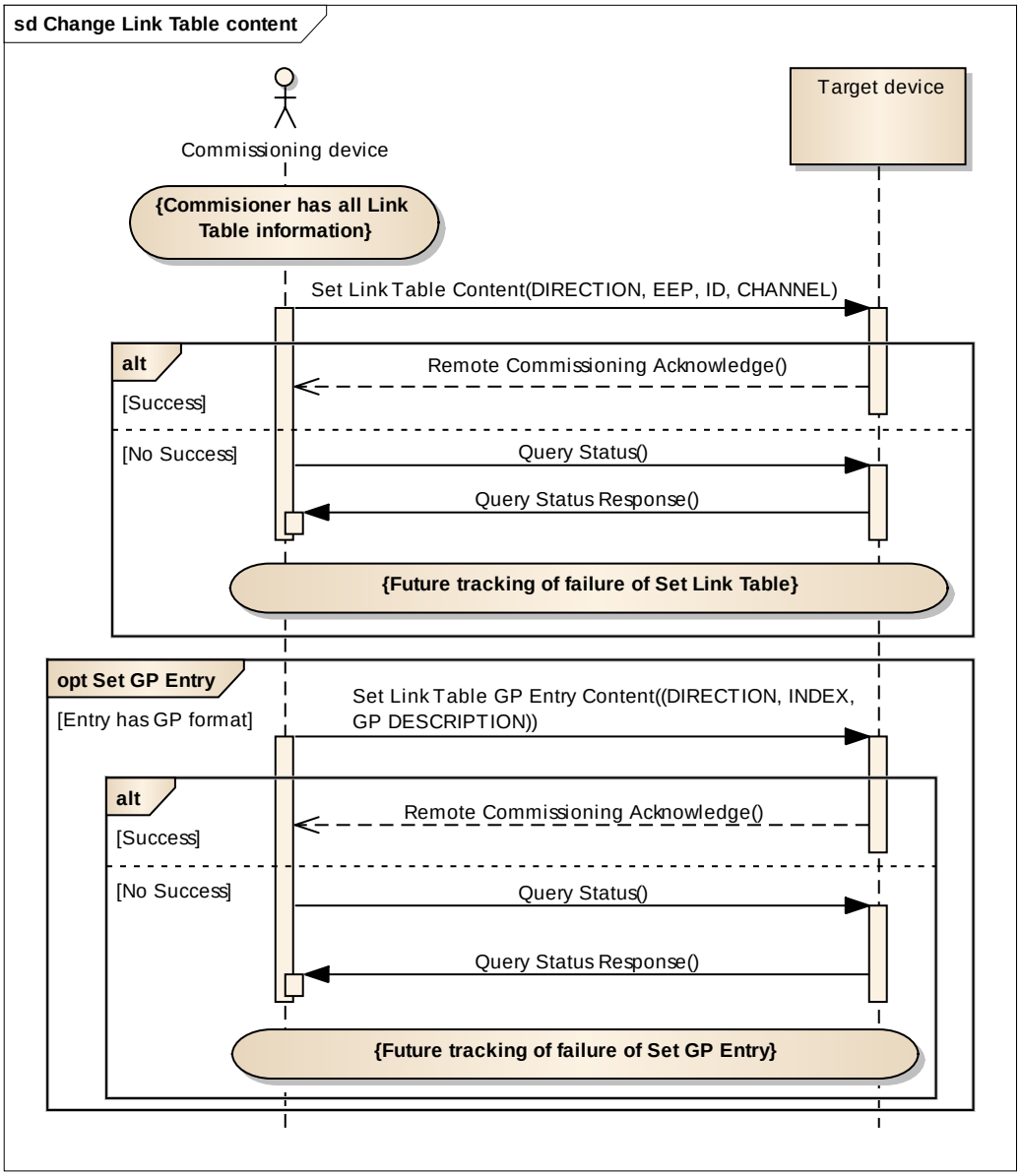

Figure 8 Change Link Table content process

Future tracking of failure includes specific processes which are generic for failures on particular devices and is not related only to the change Link Table process.

In Figure 9 you can find the summary of the whole Direct Teach Process. The above described processes are part of the process of Direct Link Table modification.

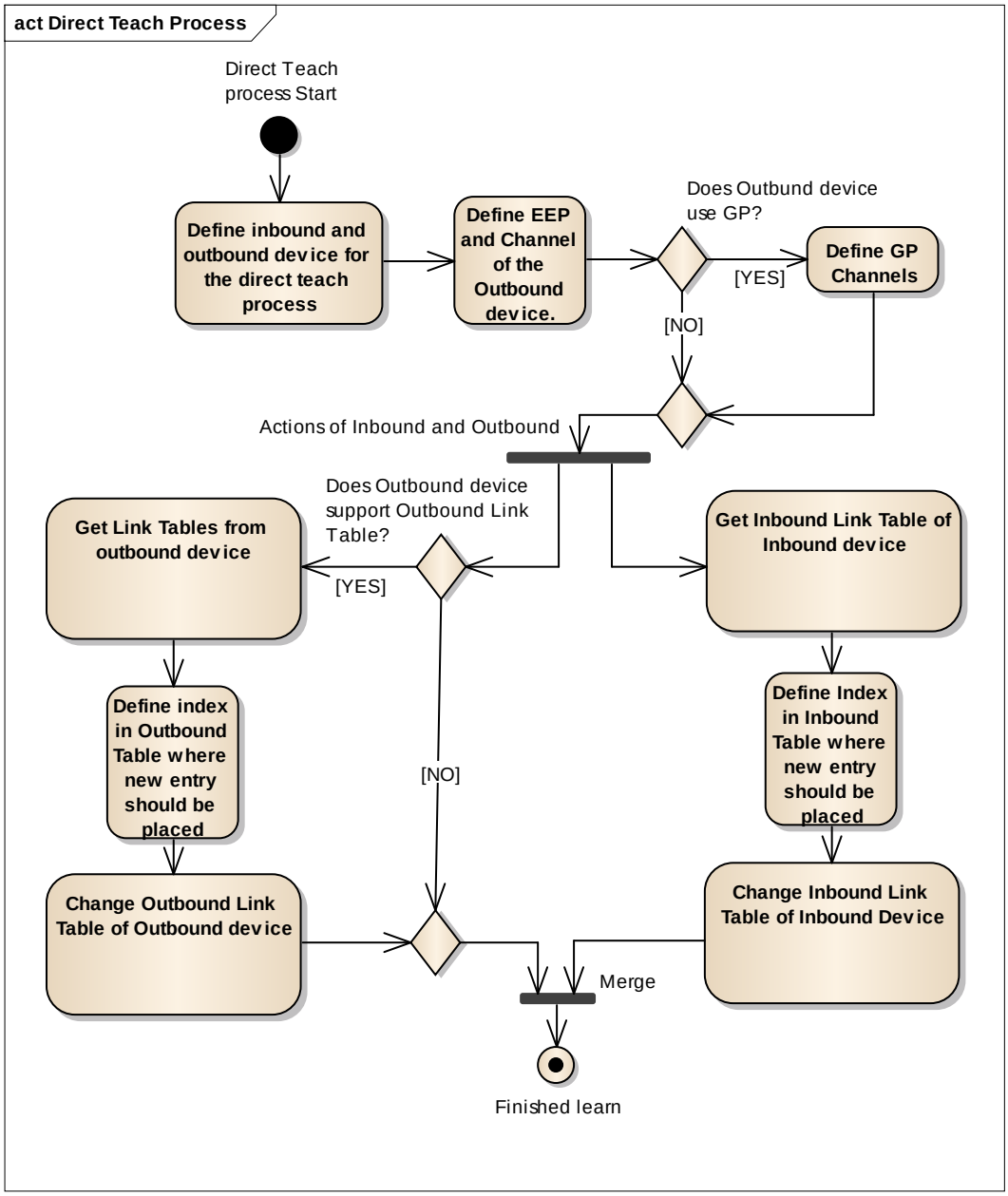

Figure 9 Direct teach process

In case of any of the above actions fails, the whole process should be canceled and returned to original state. This is required to avoid inconsistency of link table across devices.

2.2.3. Link Table definition

Having a standard definition of a link table will enable interoperability among devices that utilize remote commissioning. Devices supporting inbound and outbound teach-in maintain a link table for each direction. The Link Table definition is an abstract structure. It does not have to present as defined in physical memory. Remote management defines interfaces. Devices implementing

Link Table must provide the interface to read and set the link table. The internal implementation of the Link table is completely depending on the developer of the particular device.

EURID	EEP (ORG-FUNC-TYPE)	GP	Channel
0xFFFFFFFF	0xFFFFFF	0xF….F	0xFF
0xFFFFFFFF	0xFFFFFF	0xF….F	0xFF
0xFFFFFFFF	0xFFFFFF	0xF….F	0xFF
…	…	…	…
4 bytes	**3 bytes**	**n bytes**	**1 byte**

Table 1: Link table structure

EURID: The ID of the linked device

EEP: The EEP of the linked device, as specified in EEP [2]. A special purpose EEP, 0xB00000, should be inserted for Generic Profiles. See (GP 4.2.2). If GP is used then the definition of the channel(s) can be fetched by a separate command.

GP: Utilized if device supports GP Profiles in inbound or outbound communication. The entry holds the complete GP channel definition (described during teach in) of the device. The channel identification then refers to a particular channel (referenced by index) or to all by specifying 0xFF.

In the outbound link table case the EEP/GP will be the EEP/GP the outbound device is using to link to the inbound device. The EEP/GP on the inbound devices inbound table must match this.

Channel: A byte to aid in configuration for multi channel devices that specifies the channel to link to. Channels are reserved for a single physical device that transmits multiples of the same data, e.g. a dual channel temperature sensor. If no channel is selected or channel description is not available set to the default 'All' channel, 0xFF. If the default channel is not selected than a single row of the link table will always correspond to a single channel. The following definitions are made:

- Devices using Profile F6-02-01, F6-02-02 [2] have following channel definition:
    - Rocker A = Channel: 0x00

- Rocker B = Channel: 0x01
- All Rockers = 0xFF

- Devices using Profile F6-03-01, F6-03-02 [2] have following channel definition:
  - Rocker A = Channel: 0x00
  - Rocker B = Channel: 0x01
  - Rocker C = Channel: 0x02
  - Rocker D = Channel: 0x03
  - All Rockers = 0xFF
- Any device using Generic Profiles:
  - Generic Profiles Channel index = Channel number
  - If generic profiles device is bidirectional and support outbound tables, then outbound and inbound is used
- Future EEP submissions have to enumerate channel values as part of the EEP.

2.2.4. Security profile definition

Security profiles are assigned to a communication profile to make it encrypted. Security profiles are:

- For general communication – see Chapter 2.6.
- For Remote Management communication (i.e. SYS_EX telegrams itself) - see Chapter 2.8.5. and 2.8.6.

Security Profiles for general communication

These profiles are organized in Outbound and Inbound tables. They can differ based on broadcast / unicast and inbound / outbound communication.

Security profiles for general communication can be assigned to a link table entry by link based parameters. E.g. Outbound Link table entry IDX:5 uses for outbound communication security outbound profile IDX: 8 and inbound communication profile IDX:2. Association between link table entries and security profiles is application specific and is not defined by the Remote commissioning specification. Association can be done be Link based parameters – see Chapter 2.8.

Security profiles support following use cases:

Inbound:

■ Unicast	DESTINATION: CHIP ID,	SOURCE: CHIP ID
■ Broadcast	DESTINATION: NA	SOURCE: CHIP ID

Outbound:

- Unicast DESTINATION: CHIP ID, SOURCE: CHIP ID
- Broadcast DESTINATION: NA, SOURCE: CHIP ID

Above list gives complete overview of possible security profile use cases. The resulting RPCs features will enable the uses cases. An Application can select which use cases it decides to support.

Security profile consists of:

SLF - Security Level Format

RLC - Actual Rolling Code value.

Key - Security key.

SID - Source ID

DID - Destination ID

Considerations:

01 Outbound / inbound security profiles for one link table entry (e.g. device) shall not share one RLC counter and have different AES Keys. The RLC shall be increased by 1 after every message.

02 Outbound / inbound security profiles for one link table entry shall have same SLF definition.

03 Broadcast inbound or outbound profiles shall be market with 0xFF FF FF FF for source or destination ID.

04 Security profiles of inbound devices which have a unicast and a broadcast entry shall have a separate RLC and KEY for each entry.

Security Profiles for Remote Management communication

These profiles are defined in the Remote Management communication [1]. They encrypt the SYS_EX telegrams which are used in Reman and Recom itself. Remote commissioning provides additional RPCs how to read / write the security profiles – Key Index 0x1 – 0xF.

Key Index 0x0 is an operational profile used in operational mode there are no Security Profiles for general communication defined or assigned.

For example a sensor application which has no inbound / outbound security profiles defined will use the operational key security profile.

The application decides if it uses Security Profile organized in Outbound / Inbound tables or only the operation key profile defined in Remote management.

2.3. Application notes references

2.3.1. Physical location of end device aids

The physical location of an end device is stored in the installation documentation of the network. Please refer to chapter 3. for more details. However it is handy to store a reference to physical end location also in the particular end device. For this purpose use the Device configuration parameter with the sting type. In practice it means:

- with the Set Device configuration command (See chapter 2.8.2.) the string type parameter is set to a full text reference of its end location
- with the Get Device configuration command (See chapter 2.8.1.) this string type parameter is returned

2.3.2. RF Link check mechanism with Device Parameters

The radio link quality needs to be checked for every Inbound Link table entry. There shall be a binary flag defined for each entry or byte value. Once the device will receive a telegram from the device defined in the link table with a “good enough dBm” it should be set to TRUE (0b1). A “good enough dBm” value is defined by the device manufacturer (e.g. a reference value could be between -70 dBm to -80 dBm). The variable can be set via the remote configuration interface again to FALSE (0b0) to revalidate the communication link.

- By reading out the value TRUE it can be confirmed that the radio link was confirmed.
- By reading out the value FALSE the radio link was not yet confirmed.

Alternatively the average value in dBm of the past 20 telegrams can be read out as device parameter.

The binary flag can be defined as Link based parameter or as Device Configuration parameters, please see chapter 2.8. . Specific decision depends on the device manufacturer. We use the Get / Set Link based configuration commands or Get / Set Device configuration commands to operate binary radio flags.

2.3.3. RF Link check mechanism with Signal Telegram

Once using Direct Teach Process (See chapter 2.2.2.) it practical to confirm actual radio contact of the links after putting the nodes to final location. Otherwise it cannot be confirmed that the nodes which are installed in end position have good radio link budget. At professional installation process the radio links are checked by additional equipment. The feature proposed here is only a simple way to confirm the functional link.

A device specific parameter exists, which enables the “Link Quality” check. As long as this is active, the system sends Signal telegrams - MID 0x0A RX-channel quality – while active to inform about link quality of linked devices. See

The device specific parameter to enable “Link Quality” check, please see chapter 2.8. Use the Get / Set Device configuration commands to operate the functionality.

2.3.4. Standard secure code

In Reman it has become wide spread to use one standard key for factory shipments. It is a recommended application scenario, to change the default key to a specific one on the first commissioning. To enforce this practice the default key shall be valid only for commands: UNLOCK, SETCODE, LOCK, QUERY ID, GET PRODUCT ID, GET PRODUCT ID SELECTIVE.

Option:

- close recom interface within minutes, hours when standard code after power on
- Do not allow operation without setting specific key
- Accept risk of default code in devices

Once the code was changed and Unlock with new code was done all other commands are available.

2.3.5. Quick check of changed configuration

Recom Managers want to have a quick way to confirm device state (e.g. Paramters & Links) has not changed since the last time when the device was managed. One way is to simply read out the complete status of parameters and compare with stored ones. With growing amount of parameters and increasing count of devices this might take even 10 or more minutes.

Therefore it is recommended to create a configuration parameter device or link (see 2.8.) which is a version or hash representing the device status. By reading out this parameter it can be confirmed with one operation if configuration of a device or link has changed or not and decide what other operation should follow.

In DDF the index of this specific parameters shall be marked.

2.4. Remote commissioning telegram structures overview

Please look at the summary of the defined telegram structures and their use case in Figure 10. The telegram structures have a green background.

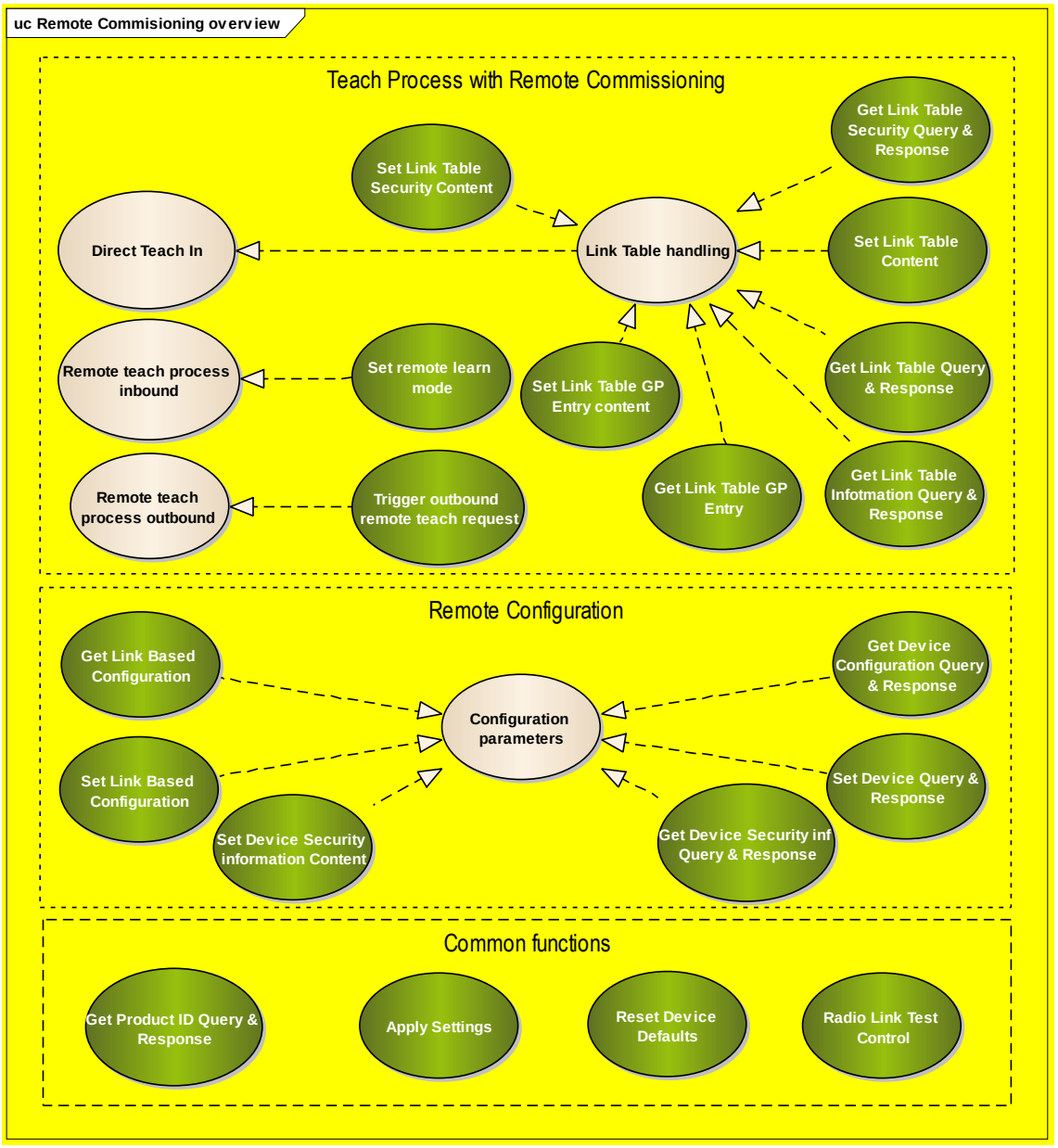

Figure 10 Remote commissioning telegram structures with use cases

2.5. Link Table handling and Remote teach process Telegram Structures

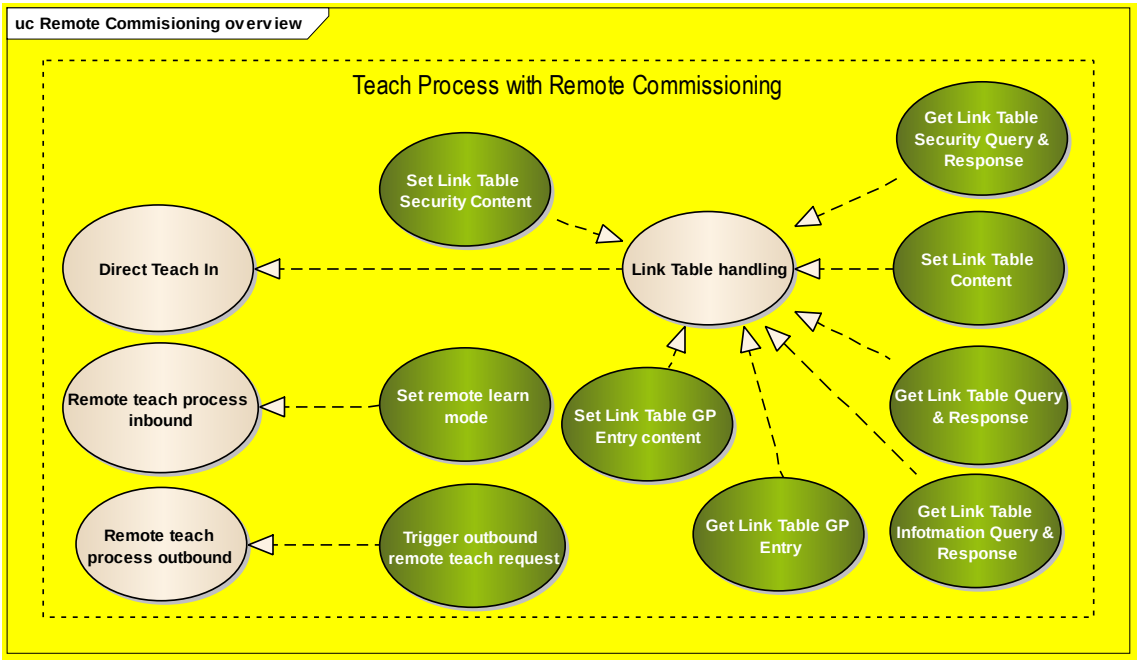

Figure 11 Remote Teach Process with telegram structures and use cases

2.5.1. Get Link Table Metadata Query & Response

Get Link Table Metadata Query	
Function code	0x210
Manufacturer Id	0x7FF
Data length	0 bytes
Addressed	Yes
Broadcast	No
Command has paired response	Yes
Status return code	Please refer to Section A 2. Query Status return codes.

Table 2 Get Link Table Metadata Query

Get Link Table Metadata Response		
Function code		0x810
Manufacturer Id		0x7FF
Data length		5 bytes
Data Content		
	Remote teach outbound supported	1 bit
	Remote teach inbound supported	1 bit
	Outbound Link Table supported	1 bit
	Inbound Link Table supported	1 bit
	Current length of Outbound table	1 byte
	Max size of Outbound table	1 byte
	Current length of Inbound table	1 byte
	Max size of Inbound table	1 byte
Addressed		Yes
Broadcast		No

Table 3 Get Link Table Metadata Response

Data content:

Remote teach outbound supported (1 bit):

- 0b0 – Remote teach-in outbound not supported
- 0b1 – Remote teach-in outbound supported

Remote teach- inbound supported (1 bit):

- 0b0 – Remote teach-in inbound not supported
- 0b1 – Remote teach-in inbound supported

Outbound Link Table used (1 bit):

- 0b0 – Outbound Link Table is not supported
- 0b1 – Outbound Link Table is supported

Inbound Link Table used (1 bit):

- 0b0 – Inbound Link Table is not supported
- 0b1 – Inbound Link Table is supported

Current length of the outbound table (1 byte):

- 0x00 – 0xFF: Number of valid links in the outbound link table

Max Size of the outbound table (1 byte):

- 0x00 - Outbound link table not supported
- 0x01 – 0xFF: Maximum length of the Outbound Table

Current length of the inbound table (1 byte):

- 0x00 – 0xFF: Number of valid links in the inbound link table

Max Size of the inbound table (1 byte):

- 0x00: Inbound link table not supported
- 0x01 – 0xFF: Maximum length of the Inbound Table

Data structure:

	7	6	5	4	3	2	1	0
0	1/0	1/0	1/0	1/0	RESERVED			
1	Current length of Outbound table							
2	Size of Outbound table							
3	Current length of Inbound table							
4	Size of Inbound table							

Table 4 Get Link Table Metadata Response data structure

2.5.2. Get Link Table Query & Response

Get Link Table Query		
Function code		0x211
Manufacturer Id		0x7FF
Data length		3 bytes
Data content		
	Table Direction	1 bit
	Starting Index	1 byte
	Ending Index	1 byte
Addressed		Yes
Broadcast		No
Command has paired response		Yes
Status return code	Please refer to Section A 2. Query Status return codes.	

Table 5 Get Link Table Query

Data content:

Table Direction (1 bit):

- 0b0 – Inbound
- 0b1 - Outbound

Staring Index (1 byte):

- 0x00 – 0xFF Index of the first entry in link table that is requested (zero based)

Ending Index (1 byte):

- 0x00 – 0xFF Index of the last entry in link table that is requested (zero based)

Note:

- If starting and ending index are the same, then one entry is returned
- To receive the whole active link table, perform a Get Link Table Query where Starting Index = 0 and Ending Index =Current Table Size - 1

Data structure:

	7	6	5	4	3	2	1	0
0	Direction	RESERVED						
1	Starting Index							
2	Ending Index							

Table 6 Get Link Table Query data structure

Get Link Table Response	
Function code	0x811
Manufacturer Id	0x7FF
Data length	X*9 bytes + 1 byte
Data content	
Table Direction	1 bit
Link Table Entry	9 byte * X
Addressed	Yes
Broadcast	No

Table 7 Get Link Table Response

Data content:

Table Direction (1 bit):

- 0b0 – Inbound
- 0b1 - Outbound

Link Table Entry (9 byte) * X – For Details see Link Table definition in chapter 2.2.3. :

- Index (1 byte) – 0x00 – 0xFF Index of the Link Table Entry
- ID (4 bytes)
- EEP (3 bytes)
- Channel (1 byte)

X = (Ending Index of request- Starting Index of request) + 1

Data structure:

	7	6	5	4	3	2	1	0
0	Direction	RESERVED						
1	Index							
2	ID							
3								
4								
5								
6	EEP							
7								
8								
9	Channel							

Table 8 Get Link Table Response data structure

2.5.3. Set Link Table Content

Set Link Table Content		
Function code		0x212
Manufacturer Id		0x7FF
Data length		X*9 bytes + 1 byte
Data content		
	Direction	1 bit
	Link Table Entry	9 byte * X
Addressed		Yes
Broadcast		Yes
Command has paired response		No
Status return code	Please refer to Section A 2. Query Status return codes.	

Table 9 Set Link Table Content

Data Content Description:

Table Direction (1 bit):

- 0b0 – Inbound
- 0b1 - Outbound

Link Table Entry (9 byte) * X (for details see Link Table definition please refer to section 2.2.3.):

- Index (1 byte) – 0x00 – 0xFF Index of the Link Table Entry
- ID (4 bytes)
- EEP (3 bytes)
- Channel (1 byte)

Where X is (Ending Index- Starting Index) + 1

Data structure:

	7	6	5	4	3	2	1	0
0	Direction	RESERVED						
1	Index							
2	ID							
3								
4								
5								
6	EEP							
7								
8								
9	Channel							

Table 10 Set Link Table Content data structure

2.5.4. Get Link Table GP Entry Query & Response

Get Link Table GP Entry Query		
Function code		0x213
Manufacturer Id		0x7FF
Data length		2 bytes
Data content		
	Table Direction	1 bit
	Index	1 byte
Addressed		Yes
Broadcast		No
Command has paired response		Yes
Status return code	Please refer to Section A 2. Query Status return codes.	

Table 11 Get Link Table GP entry Query

Data content:

Table Direction (1 bit):

- 0b0 – Inbound
- 0b1 - Outbound

Index (1 byte):

- 0x00 – 0xFF Index of the entry in link table that is requested to send GP channel description (zero based)

Note:

- The command should be used when EEP of an entry was previously returned as 0xB00000.

Data structure:

	7	6	5	4	3	2	1	0
0	Direction	RESERVED						
1	Index							

Table 12 Get Link Table GP entry Query data structure

Get Link Table GP Entry Response		
Function code		0x813
Manufacturer Id		0x7FF
Data length		2 + x bytes
Data content		
	Table Direction	1 bit
	Index	1 byte
	Link Table Entry GP channel desc.	x bytes
Addressed		Yes
Broadcast		No

Table 13 Get Link Table GP entry Response

Data content:

Table Direction (1 bit):

- 0b0 – Inbound
- 0b1 - Outbound

Index (1 byte):

- 0x00 – 0xFF Index of the entry in link table that is requested to send GP channel description (zero based)

Link Table Entry GP description (x bytes):

- The format of the channel description is defined in the Generic Profiles Specification. Please reference [3]. It is equivalent to the payload of the GP teach-in request message.

Data structure:

	7	6	5	4	3	2	1	0
0	Direction	RESERVED						
1	Index							
2	Link Table Entry GP channel desc.							
…								
..								
x+2								

Table 14 Get Link Table Response GP entry data structure

Example

Please find below a Generic Profiles channel description data content of a sensor with two channels, current and occupancy. Total length = 5.5 bytes. Aligned length: 6 bytes

	7	6	5	4	3	2	1	0
0	0b01 – Data type		0b00000110 – Current (A) signal					
1			0b01 – Current vale		0b0101 – 8 bit resolution			
2	0b00000000 – Engineering minimum							
3	0b0001 – 1x Scaling				0b00000101			
4	*Engineering maximum*				0b0001 – 1x scaling			
5	*0b10 – Flag type*		0b00001001 – Occupancy signal					
6			*0b01 – Current value*		Not used			

Table 15 Link Table Entry GP channel description example

2.5.5. Set Link Table GP Entry Content

Set Link Table GP Entry Content		
Function code		0x214
Manufacturer Id		0x7FF
Data length		2 + X bytes
Data content		
	Direction	1 bit
	Index	1 byte
	Link Table Entry	x bytes
Addressed		Yes
Broadcast		Yes
Command has paired response		No
Status return code	Please refer to Section A 2. Query Status return codes.	

Table 16 Set Link Table GP entry Content

Data Content Description:

Table Direction (1 bit):

- 0b0 – Inbound
- 0b1 - Outbound

Index (1 byte):

- 0x00 – 0xFF Index of the entry in link table that is requested to send GP channel description (zero based)

Link Table Entry GP description (x bytes):

- The format of the channel description is defined in the Generic Profiles Specification. Please reference [3]. It is equivalent to the payload of the GP teach-in request message.

Data structure:

	7	6	5	4	3	2	1	0
0	Direction	RESERVED						
1	Index							
2	Link Table Entry GP channel description							
…								
..								
x+2								

Table 17 Set Link Table Content GP entry data structure

2.5.6. Remote Set Learn Mode

Remote Set Learn Mode		
Function code		0x220
Manufacturer Id		0x7FF
Data length		2 bytes
Data content		
	Device LRN Mode	2 bits
	Inbound Index	1 byte
Addressed		Yes
Broadcast		Yes
Command has paired response		No
Status return code	Please refer to Section A 2. Query Status return codes.	

Table 18 Set remote learn mode

If a target device does not utilize explicit inbound indexes 0xFF should be transmitted as the index.

Data content:

Device LRN Mode (2 bits):

- 0b00 – Enter device learn in mode

- 0b01 – Enter device learn out mode
- 0b10 – Exit device learn mode (any)

Inbound Index (1 byte):

- 0x00 – 0xFF - Selected index to enable teach mode on (0xFF if unused on target)

Data structure:

	7	6	5	4	3	2	1	0
0	Device LRN mode		RESERVED					
1	Inbound Index selection							

Table 19 Set remote learn mode data structure

2.5.7. Trigger Outbound Remote Teach Request

Trigger Outbound Remote Teach Request		
Function code		0x221
Manufacturer Id		0x7FF
Data length		1 byte
Data content		
	Channel Selection	1 byte
Addressed		Yes
Broadcast		No
Command has paired response		Yes[1]
Status return code	Please refer to Section A 2. Query Status return codes.	

Table 20 Trigger outbound remote teach request

Depending on the type of teach telegram sent the channel may need to be specified or may be ignored. If a device only has a single channel the default all channels, 0xFF, shall be sent in the trigger message as the channel selection.

Data content:

Channel Selection (1 byte):

- 0x00 – 0xFF – Channel to teach in (0xFF for single channel devices)

Data structure:

	7	6	5	4	3	2	1	0
0	Channel selection							

Table 21 Trigger outbound remote teach request data structure

[1] The outbound teach-in request shall be seen as acknowledge, that the action was successfully executed.

2.6. Security profile outbound and inbound telegram Structures

2.6.1. Get Security Profile Query & Response

Get Security Profile Query		
Function code		0x215
Manufacturer Id		0x7FF
Data length		2 bytes
Data content		
	Direction	1 bit
	Index	1 byte
Addressed		Yes
Broadcast		No
Command has paired response		Yes
Status return code	Please refer to Section A 2. Query Status return codes.	

Table 22 Get Security Profile Query

Data content:

Table Direction (1 bit):

- 0b0 – Inbound
- 0b1 - Outbound

Index (1 byte):

- 0x00 – 0xFF Index of the entry which is requested.

NOTE: Size and availability of security profile shall be communication via DDF referenced by the Product ID.

Data structure:

	7	6	5	4	3	2	1	0
0	Direction	RESERVED						
1	Index							

Table 23 Get Security Profile Query data structure

Get Security Profile Response		
Function code		0x815
Manufacturer Id		0x7FF
Data length		32 Bytes
Data content		
	Table Direction	1 bit
	Index	1 Byte
	Link Table Entry SLF	1 Byte
	Link Table Entry RLC	4 Byte
	Link Table Entry Key	16 Byte
	Destination ID	4 Bytes
	Source ID	4 Bytes
Addressed		Yes
Broadcast		No

Table 24 Get Security Profile Response

Data content:

Table Direction (1 bit):

- 0b0 – Inbound
- 0b1 - Outbound

Index (1 byte)

- 0x00 – 0xFF Index of the Link Table Entry

SLF (1 bytes) – Security Level Format of the linked device

RLC (4 bytes) – Current RLC, real size defined by SLF. If real size is 24 bit, then MSB is 0x00.

Key (16bytes) – AES Key

- All bytes 0 = Key cannot be read (protected)
- All bytes 0xFF = no key has been written
- Other – AES Key

Destination ID (4 bytes)

Source ID (4 bytes)

Data structure:

	7	6	5	4	3	2	1	0
0	Direction	RESERVED						
1	Index							
2	SLF							
3	RLC							
4								
5								
6								
7 - 23	Key							
24	Destination ID							
25								
26								
27								
28	Source ID							
29								
30								
31								

Table 25 Get Security Profile Response data structure

2.6.2. Set Security Profile

Set Security Profile		
Function code		0x216
Manufacturer Id		0x7FF
Data length		23 Byte
Data content		
	Table Direction	1 bit
	Index	1 Byte
	Link Table Entry SLF	1 Byte
	Link Table Entry RLC	4 Byte
	Link Table Entry Key	16 Byte
	Destination ID	4 bytes
	Source ID	4 bytes
Addressed		Yes
Broadcast		Yes
Command has paired response		No
Status return code	Please refer to Section A 2. Query Status return codes.	

Table 26 Set Security Profile

Data content:

Table Direction (1 bit):

- 0b0 – Inbound
- 0b1 - Outbound

Index (1 byte)

- 0x00 – 0xFF Index of the Link Table Entry

NOTE: Count and availability of security profile shall be communication via DDF referenced by the Product ID.

SLF (1 bytes) – Security Level Format of the linked device

RLC (4 bytes) – Current RLC, real size defined by SLF. . If real size is 24 bit, then MSB is 0x00.

Key (16bytes) – Shared AES Key

- Other – AES Key

Destination ID (4 bytes)

Source ID (4 bytes)

Data structure:

	7	6	5	4	3	2	1	0
0	Direction	RESERVED						
1	Index							
2	SLF							
3	RLC							
4								
5								
6								
7 - 23	Key							
24	Destination ID							
25								
26								
27								
28	Source ID							
29								
30								
31								

Table 27 Set Security Profile data structure

2.7. Remote Configuration

The Remote Configuration process is used to configure application specific parameters within a target device. It is mandatory that a device is fully functional without performing the Remote Configuration Process. Thus the device must be provided with a default configuration that is functional. In this way, configuration parameters reflect an extension of default functionality to enable additional applications through configuration.

A good example of a typical configuration parameter is the transmit interval of a line powered energy sensor. The device might come with a default transmit period of three minutes. Using the remote configuration commands, the time can be adjusted so the device transmits every five or ten minutes. Doing so will not influence the basic operation of the device, it will still measure and report an energy even directly out of the box. A device that needs to have the energy measurement transmit configured using remote management in order to transmit at all does not follow this specification.

This Remote Configuration interface comes with the benefit that users (installers) do not need to know low level details of how to configure parameters such as which flash address they have to write. Implementation details are abstracted away and must be handled in firmware. This ensures interoperability between target devices and commissioning devices. An external Device Description file will provide a devices details so it can be fully commissioned.

The interoperable commissioning interface for a device's configuration is represented through RPCs. These RPCs are defined:

- Get Device Configuration
- Set Device Configuration
- Get Link Based Configuration
- Set Link Based Configuration
- Set Device Security Information Content
- Get Device Security Information Content

Two type of parameter set and get interfaces are available. The first is device level parameters. These are addressed with a single index and are parameters that are used to configure a target's functionality, e.g. transmission interval timer. The transmission interval timer has no logical connection to the inbound links

The second interface is for linked parameters. These parameters are referenced with a link table index and a parameter level index. Linked parameters control functionality associated with one or more devices in the link table. A simple example is when an outbound PTM210 based switch is linked to an inbound actuator. The switch function can be configured in:

- a standard rocker mode,
- a toggle mode,
- or a momentary mode

when linked to a target device. Multiple switches may be learned in but still operate in different switching modes. Which mode to use for each inbound switch needs to be defined via the linked parameters. In this case the linked parameter would be linked to the index that the switches are learned into, denoted by the appropriate RPS based EEP. Linked parameters can be thought of as extra columns in the link table that are inbound device specific. Linked parameters can also be used when each inbound link can have a unique value for some parameter regardless of the EEP used. Each link might have a priority parameter. The EEP is not relevant in this case.

A device may have one or neither of these types of parameters depending on the configurability and complexity of the target device.

With these RPCs a commissioning tool can read or modify a target current configuration remotely. The capabilities and format of each parameter is defined in the device description XML file. The manufacturer of a target device must provide a device description file to enable the configuration portion of Remote Commissioning

Configuration parameters are gettable and settable as single values or as a list of values. Each set command deals with a single specific parameter. This enables users (e.g. installers or commissioners) to correct or adjust single values. Configuration parameters are split up into two different types:

- public
- private

Private can be used as a closed-source parameter, meaning that the description and functionality of the parameter is not available to all. If a configuration of this kind of parameter is required by the user, the manufacturer can define the default value in the Device Description file without disclosing its description or type. If the default value is defined, at least the resolution of this private value shall be known. Public parameter is well defined in the Device Description file and described. The description follows the format of the EEP like data types and has its functionality shared with commissioners via the Device Description file.

Process description

Below is the partial description of the parameter configuration process. The key feature is the reference to the Device Description file. For any kind of automated parameter configuration the Device Description file is required and should be supplied by the manufacturer of the target device.

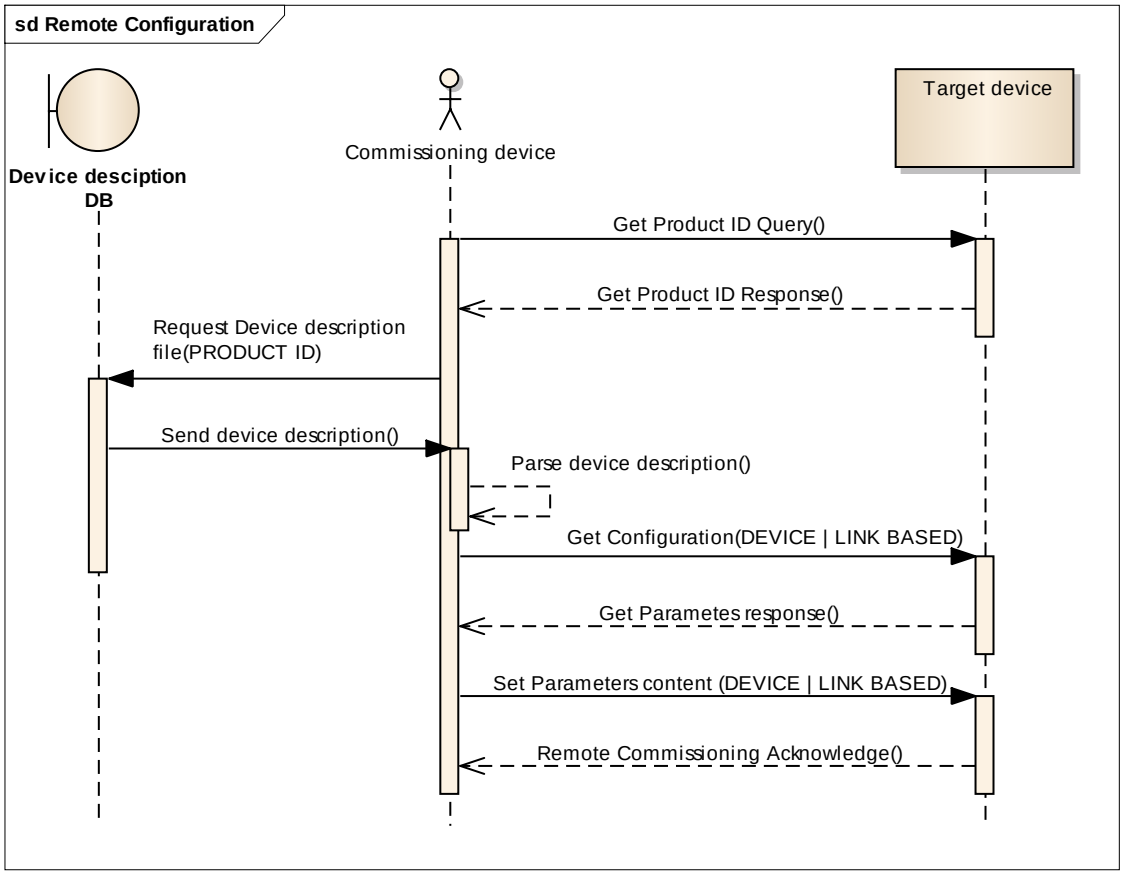

Figure 12 Remote configuration

2.7.1. Configuration parameters description

Configuration parameters are specific for every device and are defined by the manufacturer. The configuration parameters context may not be fetched over the air interface from a particular device. The configuration parameters list must be available from the Device Description file. This should be supplied by the manufacture or a central authority.

The language to describe a target device's parameter is similar to description of the EEPs and the EEP XML. [2]. There are four configuration parameter types:

- Scalable
- Enumerated
- Enumerated with scalable parts
- Text (UTF-8 formating)

Scalable

To describe scalable parameter 5 values have to be supplied:

- Bit range Min and Bit Range Max.

  The 'Bit range' represents the starting-point and the end-point of the respective data inside the telegram coding.

- Value Scale Min and Value Scale Max.
  The 'Value scale', represents the starting-point and the end-point of the respective real value
- Unit.
  Unit represents the physical dimension of the measurement / data. Unit list is defined inside the XSD document.

The linear conversion between the rawValue (transmitted in configuration telegram) and the Device value (real value with physical unit) is following:

Conversion: Valid Range ---> Scale

$$\text{Multiplier} = \frac{\text{Scale}_{MAX} - \text{Scale}_{MIN}}{\text{Range}_{MAX} - \text{Range}_{MIN}}$$

$$\text{Device value} = \text{Multiplier} * (\text{rawValue} - \text{Range}_{MIN}) + \text{Scale}_{MIN}$$

By choosing the correct Bit Range and Value Scale a round Multiplier (aka Step size) can be obtained (e.g. in most cases Multiplier 1, or 0.5, or 0.25 are selected).

Enumerated

To describe an enumerated parameter the enumerated length in bytes and default value is specified. The Enumerated List (list of possible values) must be described with an index and text description.

Enumerated with scalable parts

This field combines the enumerated parameter with a scalable parameter. Both parts need to be described individually.

Text

The text parameter is an UTF-8 formatted parameter. For the text parameter the length in bytes must be specified. The text parameter represents generic text without any specifc reference to its usage.

The configuration parameters list is stored in the XML format similar to the EEP XML representation. Please see the XSD file and XML example for details on the storage details.. The XML files should be fetched before or during the commissioning process. The process of retrieving and maintaining the files from the manufacturer or central authority by a specific token (e.g. Product ID) is out of the scope of this specification. Below is the format of the Device Description file:

XSD file Structure

An XSD defines the structure of the XML documentation file. XSD is required so the stored XML are readable by standardized tools. Please see reference [5] for the XSD.

The XSD is separated from this specification as changes in the structure are expected to take place more frequently then changes in the specification itself.

XML Example

The XML file is representation of both the Link tables and configuration parameters. For the validation and development the XSD file is required. Please see reference [5] for the XML Example.

2.8. Remote Configuration telegrams structures

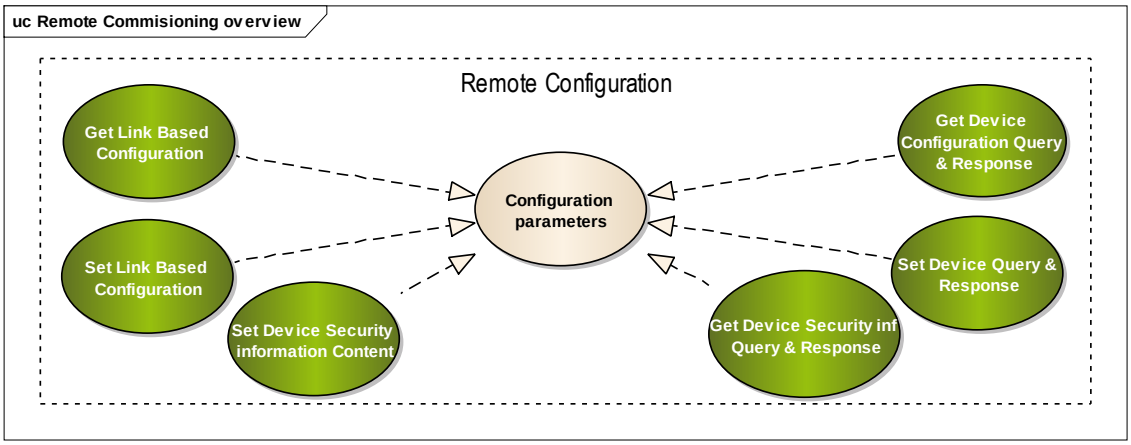

Figure 13 Remote configuration telegram structures with use case

2.8.1. Get Device Configuration Query & Response

Get Device Configuration Query		
Function code		0x230
Manufacturer Id		0x7FF
Data length		5 bytes
Data content		
	Start Index	2 byte
	End Index	2 byte
	Length	1 byte
Addressed		Yes
Broadcast		No
Command has paired response		Yes
Status return code	Please refer to Section A 2. Query Status return codes.	

Table 28 Get common Configuration Query

Data content:

Start Index (2 byte):

- 0x0000 to 0xFFFF - index of starting parameter to request

End Index (2 byte):

- 0x0000 to 0xFFFF - index of end parameter to request

Length (1 byte):

- 0x00 to 0xFF – length of the requested parameters. Length ($\sum_{i=1}^{n}(b_i)$) b_i is the size of the actual configuration parameter in the telegram. The size is always rounded up to next full byte length. E.g 12bits resolution parameter have a 2byte length.

Data structure:

	7	6	5	4	3	2	1	0
0	Start Index							
1								
2	End Index							
3								
4	Length							

Table 29 Get common Configuration Query data structure

Note

If length is set to 0x00, then length indication should be ignored by the target device.

Get Device Configuration Response	
Function code	0x830
Manufacturer Id	0x7FF
Data length	$\sum_{i=1}^{n}(b_i)$+ n *3 bytes
Data content	
Payload	$\sum_{i=1}^{n}(b_i)$+ n *3 bytes
Addressed	Yes
Broadcast	No

Table 30 Get common Configuration Response

Data content:

Payload ($\sum_{i=1}^{n}(b_i)$+ n *3 bytes) *n* is the amount of configuration parameters included, b_i is the size of the actual configuration parameter in the telegram. Each configuration parameter can have a different size – the actual size of the parameter / resolution is defined in the Device Description file. One configuration parameter will have:

- Index (2 bytes)
- Length (1 bytes) – indicates the length of the container in the telegram of value, this value is byte aligned
- Value (x)

Data structure:

The Payload description for the parameter list is below.

A single configuration parameter

	7	6	5	4	3	2	1	0
0	Index							
1								
2	Length							
3	Value							
4								
….								
N								

Table 31 One parameter data structure

Note

If the size of a value is not aligned to a byte (e.g. size of one value is 12 bits) then the field is always aligned to a byte at the least significant bit. The MSBs are then filled with 0b0. The next value starts then on the next byte. The length indication of such parameter would be also 2 bytes. The actual value resolution is indicated in the Device Description file.

	7	6	5	4	3	2	1	0
0	0	0	0	0	Value			
1	Value							

Table 32 Aligned 12 bit Value

2.8.2. Set Device Configuration Query

Set Device Configuration Query		
Function code		0x231
Manufacturer Id		0x7FF
Data length		$\sum_{i=1}^{n}(b_i)$+ n *3 bytes
Data content		
	Payload	$\sum_{i=1}^{n}(b_i)$+ n *3 bytes
Addressed		Yes
Broadcast		Yes
Command has paired response		No
Status return code	Please refer to Section A 2. Query Status return codes.	

Table 33 Set common Configuration Query

Data content:

Payload ($\sum_{i=1}^{n}(b_i)$+ n *3 bytes) *n* is the amount of configuration parameters included, b_i is the size of the actual configuration parameter in the telegram. Each configuration parameter can have different size – the actual size of the parameter / resolution is defined in the Device Description file. One configuration parameter will have:

- Index (2 bytes)
- Length (1 bytes) – indicates the length of the container in the telegram of value, this value is byte aligned
- Value (x)

Data structure:

The Payload description for the parameters is below. The length of each parameter is variable and manufacturer specific. It is defined in the configuration parameters list see section 0

	7	6	5	4	3	2	1	0
0	Index							
1								
2	Length							
3	Value							
4								
….								
N								

Table 34 One parameter data structure

Note

If the size of a value is not aligned to a byte (e.g. size of one value is 12 bits) then the field is always aligned to a byte at the least significant bit. The MSBs are then filled with 0b0. The next value starts then on the next byte. The length indication of such parameter would be also 2 bytes. The actual value resolution is indicated in the Device Description file.

	7	6	5	4	3	2	1	0
0	0	0	0	0	Value			
1	Value							

Aligned 12 bit Value

2.8.3. Get Link Based Configuration Query & Response

Get Link Based Configuration Query		
Function code		0x232
Manufacturer Id		0x7FF
Data length		7 bytes
Data content		
	Direction	1 bit
	Link Table index	1 byte
	Start Index	2 byte
	End Index	2 byte
	Length	1 byte
Addressed		Yes
Broadcast		No
Command has paired response		Yes
Status return code	Please refer to Section A 2. Query Status return codes.	

Table 35 Get link based Configuration Query

Data content:

Direction (1 bit):

- 0b0 –Inbound link table
- 0b1 –Outbound link table

Link Table index (1 byte):

- 0x00 to 0xFF – index of the link table entry which the configuration refers to

Start Index (2 byte):

- 0x00 to 0xFFFF - index of starting parameter to request

End Index (2 byte):

- 0x00 to 0xFFFF - index of end parameter to request

Length (1 byte):

- 0x00 to 0xFF – length of the requested parameters container. Length ($\sum_{i=1}^{n}(b_i)$) b_i is the size of the actual configuration parameter in the telegram. The size is always rounded up to next full byte length. E.g 12bits resolution parameter have a 2byte length.

Data structure:

	7	6	5	4	3	2	1	0
0	Direction	Reserved						
1	Link Table index							
2	Start Index							
3								
4	End Index							
5								
6	Length							

Table 36 Get link entry Configuration Query data structure

Note

If length is set to 0x00, then length indication should be ignored by the target device.

Get Link Based Configuration Response		
Function code		0x832
Manufacturer Id		0x7FF
Data length		2+$\sum_{i=1}^{n}(b_i)$+ n *3 bytes
Data content	Direction	1 bit
	Link Table Index	1 byte
	Payload	$\sum_{i=1}^{n}(b_i)$+ n *3 bytes
Addressed		Yes
Broadcast		No

Table 37 Get link based Configuration Response

Data content:

Direction (1 bit):

- 0b0 –Inbound link table

- 0b1 –Outbound link table

Link Table index (1 byte):

- 0x00 to 0xFF – index of the link table entry which the configuration refers to

Payload ($\sum_{i=1}^{n}(b_i)$+ n *3) *n* is the amount of configuration parameters included, b_i is the size of the actual configuration parameter in the telegram. Each configuration parameters can have different sizes – the actual size of the parameter / resolution is defined in the Device Description files. One configuration parameter will have:

- Index (2 bytes)
- Length (1 bytes) – indicates the length of the container in the telegram of value, this value is byte aligned
- Value (x)

Data structure:

The telegram description for the parameter list is below.

	7	6	5	4	3	2	1	0
0	Direction	Reserved						
1	Link Table index							
2	Payload							
3								
4								
5								
….								
N								

Table 38 Get link based Configuration Query Data structure

A single configuration parameter / payload :

	7	6	5	4	3	2	1	0
2	Index							
3								
4	Length							
5	value							
6								
….								
N								

Table 39 One parameter data structure

Note

If the size of a value is not aligned to a byte (e.g. size of one value is 12 bits) then the field is always aligned to a byte at the least significant bit. The MSBs are then filled with 0b0. The next value starts then on the next byte. The length indication of such parameter would be also 2 bytes. The actual value resolution is indicated in the Device Description file.

	7	6	5	4	3	2	1	0
0	0	0	0	0	Value			
1	Value							

Table 40 Aligned 12 bit Value

2.8.4. Set Link Based Configuration Query

Set Link Based Configuration Query		
Function code		0x233
Manufacturer Id		0x7FF
Data length		2+$\sum_{i=1}^{n}(b_i)$+ n *3 bytes
Data content	Direction	1 bit
	Link Table Index	1 byte
	Payload	$\sum_{i=1}^{n}(b_i)$+ n *3 bytes
Addressed		Yes
Broadcast		Yes
Command has paired response		No
Status return code	Please refer to Section A 2. Query Status return codes.	

Table 41 Set link entry Configuration Query

Data content:

Direction (1 bit):

- 0b0 –Inbound link table
- 0b1 –Outbound link table

Link Table index (1 byte):

- 0x00 to 0xFF – index of the link table entry which the configuration refers to

Payload ($\sum_{i=1}^{n}(b_i)$+ n *3 bytes) *n* is the amount of configuration parameters included, b_i is the size of the actual configuration parameter in the telegram. Each configuration parameter can have different size – the actual size of the parameter / resolution is defined in the Device Description file. One configuration parameter will have:

- Index (2 bytes)
- Length (1 bytes) – indicates the length of the container in the telegram of value, this value is byte aligned
- Value (x)

Data structure:

The description for the parameters is below. The length of each parameter is variable and manufacturer specific. It is defined in the configuration parameters list see section 0

	7	6	5	4	3	2	1	0
0	Direction	Reserved						
1	Link Table index							
2	Payload							
3								
4								
5								
….								
N								

Table 42 Set link based Configuration Query Data structure

One entry / payload is described below:

	7	6	5	4	3	2	1	0
2	Index							
3								
4	Length							
5	value							
6								
….								
N								

Table 43 One parameter data structure

Note

If the size of a value is not aligned to a byte (e.g. size of one value is 12 bits) then the field is always aligned to a byte at the least significant bit. The MSBs are then filled with 0b0. The next value starts then on the next byte. The length indication of such parameter would be also 2 bytes. The actual value resolution is indicated in the Device Description file.

	7	6	5	4	3	2	1	0
0	0	0	0	0	Value			
1	Value							

Table 44 Aligned 12 bit Value

2.8.5. Get Device Security Information Query & Response

Get Device Security Information Query		
Function code		0x234
Manufacturer Id		0x7FF
Data length		
Data content	Security Info Index	1 byte
Addressed		Yes
Broadcast		No
Command has paired response		No
Status return code	Please refer to Section A 2. Query Status return codes.	

Table 45 Get Device Security Information Query

Data content:

Security Info Index (1 byte)

- 0x00 – Reserved
- 0x01 – 0xF Maintenance Key 1 … 15

Note:

- A device may for security reasons not report the key (send 0x0 as key) – application decision, but the RLC and SLF should always be reported. To see that a security links exist and allow a RLC synchronization if needed.
- If a selected key is not supported by the device no response will be send and the error can be queried.

Data structure:

	7	6	5	4	3	2	1	0
0	Security Info Index							

Table 46 Get Device Security Information data structure

Get Device Security Information Response		
Function code		0x834
Manufacturer Id		0x7FF
Data length		22 Byte
Data content		
	Security Info Index	1 Byte
	Link Table Entry SLF	1 Byte
	Link Table Entry RLC	4 Byte
	Link Table Entry Key	16 Byte
Addressed		Yes
Broadcast		No

Table 47 Get Device Security Information Response

Data content:

Security Info index (1 byte)

- 0x00 – Reserved
- 0x01 – 0x0F Maintenance Key 1…15

SLF (1 bytes) – Security Level of the used key. (if Key is from 0x01 – 0x0F Value is 0xEB)

RLC (4 bytes) – Current RLC, real size defined by SLF

Key (16bytes) – Shared AES Key

Note:

SEC_MAN uses predefined SLF (Value: 0xEB). See Reman specification.

Data structure:

	7	6	5	4	3	2	1	0
0	Security Info Index							
1	SLF							
2	RLC							
3								
4								
5								
6 - 21	Key							

Table 48 Get Device Security Information Response data structure

2.8.6. Set Device Security Information Content

Set Device Security Information Content		
Function code		0x235
Manufacturer Id		0x7FF
Data length		22 Byte
Data content		
	Security Info Index	1 Byte
	SLF	1 Byte
	RLC	4 Byte
	Key	16 Byte
Addressed		Yes
Broadcast		Yes
Command has paired response		No
Status return code	Please refer to Section A 2. Query Status return codes.	

Table 49 Set Link Table Content

Data content:

Security Info index (1 byte)

- 0x00 – Reserved
- 0x01 – 0x0F Maintenance Key 1…15

SLF (1 bytes) – Security Level of the used key (if Key is from 0x01 – 0x0F Value is 0xEB)

RLC (4 bytes) – Current RLC, real size defined by SLF

Key (16bytes) – Shared AES Key

Note:

SEC_MAN uses predefined SLF (Value 0xEB). See Reman specification.

Note:

When a new Key or other security parameters are set, it is highly recommended that new settings are active after closing the current ReMan session or after apply changes has been executed. If a device has limited memory capacity, it can immediately change the security information.

Data structure:

	7	6	5	4	3	2	1	0
0	Security Info index							
1	SLF							
2	RLC							
3								
4								
5								
6 - 21	Key							

Table 50 Set Device Security Information Content

2.8.7. Remote Set Learn Mode

Remote Set Learn Mode		
Function code		0x220
Manufacturer Id		0x7FF
Data length		2 bytes
Data content		
	Device LRN Mode	2 bits
	Inbound Index	1 byte
Addressed		Yes
Broadcast		Yes
Command has paired response		No
Status return code	Please refer to Section A 2. Query Status return codes.	

Table 51 Set remote learn mode

If a target device does not utilize explicit inbound indexes 0xFF should be transmitted as the index.

Data content:

Device LRN Mode (2 bits):

- 0b00 – Enter device learn in mode
- 0b01 – Enter device learn out mode
- 0b10 – Exit device learn mode (any)

Inbound Index (1 byte):

- 0x00 – 0xFF - Selected index to enable teach mode on (0xFF if unused on target)

Data structure:

	7	6	5	4	3	2	1	0
0	Device LRN mode		RESERVED					
1	Inbound Index selection							

Table 52 Set remote learn mode data structure

2.9. Common Remote Configuration Functions and telegram structures

The remote configuration process can be used to modify link information and parameters within a target device. A remote configuration process, e.g. loading of configuration data to the device, takes some time; the duration being dependent on the amount of configuration data to be loaded to the device. Without a certain control of the process, the device may operate with inconsistent parameter settings, and therefore show unexpected behavior until the commissioning process is finished. In order to avoid operation with inconsistent configuration data settings, the remote configuration process may be gated with the Apply Settings Command.

In addition to controlling when updated settings may be applied, it may be useful to set a device back to its defaults.

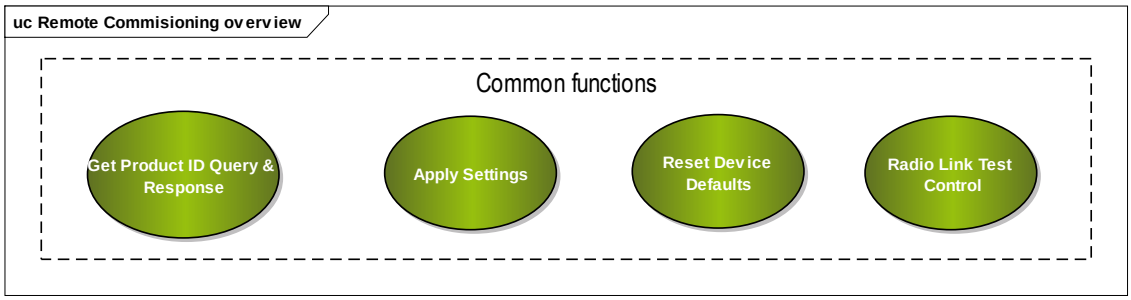

Figure 14 Common function telegram structures

2.9.1. Apply Changes

After loading configuration or link data to a device, it is required to apply the changes to a device with a remote apply changes command. If the device support the apply change command the device shall interpret the apply changes command as information that a commissioning step was finished. Further change can be performed also after the Apply Settings command. An apply changes command may reboot the device depending on its application.

Apply Changes Command		
Function code		0x226
Manufacturer Id		0x7FF
Data length		1 byte
Data content		
	Apply Link Table Changes	1 bit
	Apply Configuration Changes	1 bit
Addressed		Yes
Broadcast		No
Command has paired response		No
Status return code	Please refer to Section A 2. Query Status return codes.	

Table 53 Apply Changes Command

Data content:

Apply Updated Link Table flag (1 bit):

- 0b0 – Do not apply link table changes
- 0b1 – Apply link table changes

Apply Updated Configuration flag (1 bit):

- 0b0 – Do not apply configuration changes
- 0b1 – Apply configuration changes

Data structure:

	7	6	5	4	3	2	1	0
1	1/0	1/0						

Table 54 Apply Changes Command data structure

2.9.2. Reset Device Defaults

The Reset to Defaults command does not need to use the Apply Configuration command. Changes should be effective immediately. Note that the default link table setup may include EEPs and/or IDs and does not infer an empty link table.

Reset to Defaults		
Function code		0x224
Manufacturer Id		0x7FF
Data length		1 byte
Data content		
	Set Configuration Parameters to Defaults	1 bit
	Set Inbound Link Table to Defaults	1 bit
	Set Outbound Link Table to Defaults	1 bit
Addressed		Yes
Broadcast		Yes
Command has paired response		No
Status return code	Please refer to Section A 2. Query Status return codes.	

Table 55 Reset to Defaults

Data content

Set Configuration Parameters to defaults (1 bit)

- 0b0: Do not reset configuration Parameters
- 0b 1: Reset configuration Parameters

Set Inbound Link Table to defaults (1 bit)

- 0b 0: Do not set inbound link table to default inbound link table
- 0b 1: Set inbound link table to default inbound link table

Set Outbound Link Table to defaults (1 bit)

- 0b 0: Do not set outbound link table to default outbound link table
- 0b 1: Set outbound link table to default outbound link table

Data structure

	7	6	5	4	3	2	1	0
0	1/0	1/0	1/0					

Table 56 Reset to Defaults data structure

2.9.3. Radio link test control

This RPC enables or disables the Radio Link Test (RLT) capability of a device. The device under control acts as an RLT Slave (see EEP Specification [2] for RLT details – A5-3F-00). An automatic switch-off secures that a device supporting this RPC does not overload the radio communication channel unintentionally.

Radio Link Test Control		
Function code		0x225
Manufacturer Id		0x7FF
Data length		1 byte
Data content		
	Control	1 byte
Addressed		Yes
Broadcast		No
Command has paired response		Yes[2]
Status return code	Please refer to Section A 2. Query Status return codes.	

Table 57 Radio Link Test Control

Data content:

Enable / Disable (1 bit):

- 0b0 – Disable
- 0b1 – Enable

Number of RLT Slave (7 bits):

- 0x00 – Not Valid
- 0x01 … 0x7F – Number of RLT Slave message clusters (each consisting of 128 individual RLT messages) after which the RLT functionality is disabled automatically.

Data structure:

[2] The RLT telegrams should be considered as acknowledges, confirming that the operation was successful.

	7	6	5	4	3	2	1	0
0	1/0	Number of RLT Slave message clusters						

Table 58 Radio Link Test Control data structure

2.9.4. Get Product ID Query & Response

The Product ID Query returns the Product ID of a device. This manufacture and device specific ID can be used as a key to lookup additional information about an EnOcean device via the Device Description file.

The Product ID is the combination of the Manufacturer ID and a 4 byte Product Reference that is unique per a device's firmware and is managed by the manufacturer. The Product ID is 6 bytes in length.

The Get Product ID Response is transmitted from the target device in a beaconing mode, when the Get Product ID Request is transmitted broadcast. Beaconing mode represents the repeated transmission of a response until the device is acknowledged by the commissioning device with any addressed Remote Management message to the target device. After receiving any addressed Remote Management message the beaconing stops.
The period to repeat the beacon is semirandom specific for every end device 10 times within one minute. At every retransmission of the beacon the period is randomly determined again.

The Get Product ID and Get Product ID Selective may be processed only in locked state if the default factory code is set or not set. Get Product ID Selective shall not be processed in locked status of the managed devices with specific code set [1].

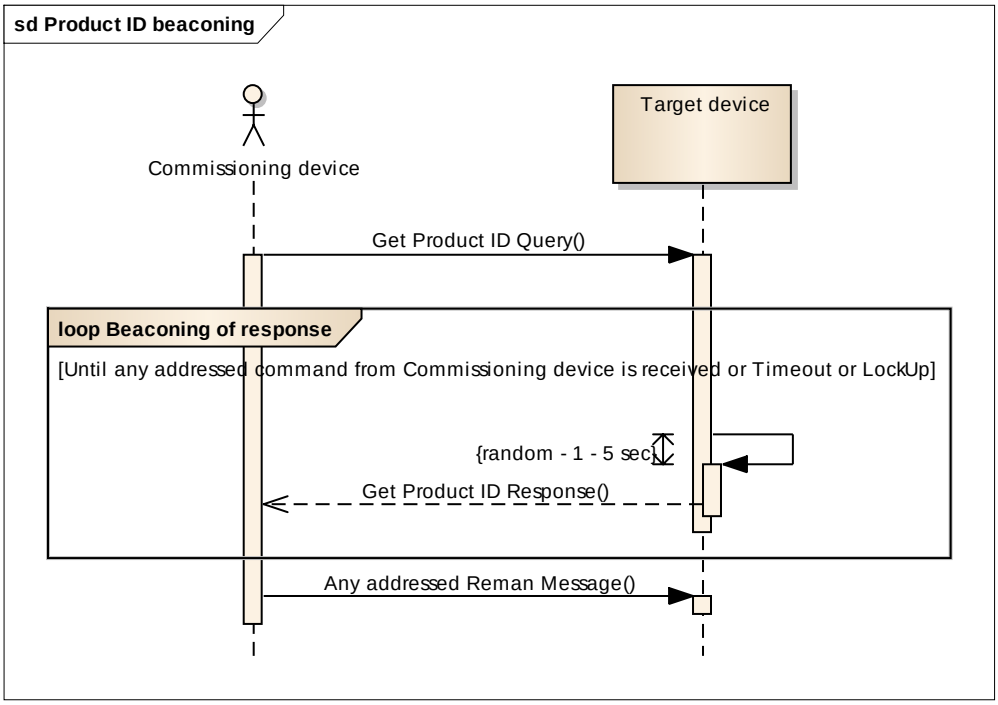

Figure 15 Product ID Beaconing sequence

Get Product ID Query	
Function code	0x227
Manufacturer Id	0x7FF
Data length	0 bytes
Data content	n/a
Addressed	Yes
Broadcast	Yes
Command has paired response	Yes
Status return code	Please refer to Section A 2. Query Status return codes.

Table 59 Get Product ID Query

Get Product ID Response		
Function code		0x827
Manufacturer Id		0x7FF
Data length		6 bytes
Data content		
	Product ID	6 bytes
Addressed		Yes
Broadcast		Yes

Table 60 Get Product ID Response

Data content:

Product ID (6 bytes):

- Manufacturer ID - 2 bytes
- Product Reference ID – 4 bytes

Data structure

	7	6	5	4	3	2	1	0
0	Manufacturer ID							
1								
2	Product Reference							
3								
4								
5								

Table 61 Get Product ID Response data structure

2.9.5. Get Product ID Selective Query & Response

In large environments the amount of devices responding to a broadcast query is too big and difficult to process even with the beaconing process. Therefore additional Get Product ID Selective is defined with a data content narrowing the queried devices down.

Get Product ID Selective Query		
Function code		0x227
Manufacturer Id		0x7FF
Data length		x byte
Data content		Yes
	Type of selection	1 byte
	Selection criteria	X bytes
Addressed		Yes
Broadcast		Yes
Command has paired response		Yes
Status return code	Please refer to Section A 2. Query Status return codes.	

Table 62 Get Product ID Selective Query

Data Content Description:

Type of selection (1 byte):

- 0x00 – dBm selection – only device response if query was received with -80 dBm or better
- 0x01 – dBm selection – only device response if query was received with -70 dBm or better
- 0x02 – dBm selection – only device response if query was received with -50 dBm or better
- 0x03 – Product ID Selection
- 0x04 – EurID modulo 4 Selection
- 0x05 – EurID modulo 8 selection
- 0x06 – EurID modulo 16 selection
- 0x07 – EurID modulo 32 selection

Selection criteria (x bytes):

If dBm selection (Type: 0x00, 0x01, 0x02):

- No additional selection criteria

If Product ID selection (Type: 0x03)

- 6 bytes Product ID– only devices with specified Product ID shall response

If Modulo selection (Type: 0x04, 0x05, 0x06, 0x07)

- 1 byte – Modulo result – only device with Eurid % Modulo Selection = Modulo Result shall response to this query.

  e.g. selection = 4, result = 0x0, device 0x12345678 mod 4 = 0 - device will response

  e.g. selection = 16, result = 0x0, device 0x12345678 mod 16 ≠ 0 - device will NOT response

Data structure:

	7	6	5	4	3	2	1	0
0	Type of Selection							
1	Product ID / Modulo result							
2								
3								
4								
5								
6								

Table 63 Get Product ID Selective data structure

Get Product ID Selective Response	
Function code	0x828
Manufacturer Id	0x7FF
Data length	6 bytes
Data content	
Product ID	6 bytes
Addressed	Yes
Broadcast	Yes

Table 64 Get Product ID Selective Response

Data content:

Product ID (6 bytes):

- Manufacturer ID - 2 bytes
- Product Reference ID – 4 bytes

Data structure

	7	6	5	4	3	2	1	0
0	Manufacturer ID							
1								
2	Product Reference							
3								
4								
5								

Table 65 Get Product ID Selective Response data structure

3. Documentation of EnOcean Networks Utilizing Remote Commissioning

Being able to read a target's link tables (both inbound and outbound) and parameters remotely using remote commissioning enables documentation of an EnOcean installation with ease.

All installations that implement remote commissioning should be documented to allow the execution of all use cases. When documentation is performed, it is mandatory to follow the structure and instructions described in this chapter.

Below is defined a standard method to document link tables and parameters of an EnOcean device.

Information to be documented

All the data stored in each device should be documented. This includes (if relevant):

- - Inbound Link Table
- - Outbound Link Table
- - Configuration Parameters
- - Target's Device Description File

Documentation process

The documentation of an installation needs to be split into two parts:

1. XML file (mandatory): where all the devices, their link tables and configuration parameters' values are included
2. MAP of the installation (optional): Drawing of the installation with the physical location of the devices written down. The following data should accompany each device:
    a. Chip-ID (mandatory)
    b. Base-ID (optional – if utilized)
    c. Model (optional)
    d. Name (optional)
    e. Description (optional)

XSD file Structure

An XSD defines the structure of the XML documentation file. XSD is required so the stored XML are readable by standardized tools. Please see reference [5] for the XSD.

The XSD is separated from this specification as changes in the structure are expected to take place more frequently then changes in the specification itself.

XML Example

The XML file is representation of both the Link tables and configuration parameters. For the validation and development the XSD file is required. Please see reference [5] for the XML Example.

4. Remote Commissioning application types

The EnOcean ecosystem has many different devices and functionalities. Some devices communicate outbound only, some inbound only, and some communicate both in and outbound directions under normal operation. Any of these devices can adopt Remote Commissioning; what portions are adopted depends completely on the device and its functionality.

To comply with the Remote Commissioning standard a minimum set of RPCs must be supported for inbound or outbound communications on either the target or commissioning device regardless of the use case. Beyond this, Remote Commissioning will be utilized based on a target's capabilities and commissioning workflow that best fits the application. We summarize the required commands, based on the use case desired, into bundles of commands

We bundle the groups of Remote Commissioning RPC into functionality below.

4.1. Link table support

In link table first the basic bundle must be supported before the GP or the parameter bundle can be supported.

4.1.1. Basic Commands bundle

- Get Link Table Metadata Query & Response
- Get Link Table Query & Response
- Set Link Table Content

4.1.2. Security bundle

4.1.3. Link table GP Commands bundle

- Get Link Table GP Entry Query & Response
- Set Link Table GP Entry Content

4.1.4. Link table based parameters bundle

- Get Link Based Configuration Query & Response
- Set Link Based Configuration Query

4.2. Remote Learn bundle

- Remote Set Learn Mode
- Trigger Outbound Remote Teach Request

4.3. Configuration parameters bundle

- Get Device Configuration Query & Response
- Set common Configuration Query

4.4. Remote Commissioning Mandatory commands bundle

- Get Product ID Query & Response
- Remote Commissioning Acknowledge

4.5. Remote Commissioning Optional commands bundle

- Reset Device Defaults
- Radio link test control
- Apply Changes
- Get Product ID Selective Query & Response
-

A. Appendix

A 1. Selective Repeating

In the following chapter RPC commands to control selective repeating are defined. These commands control the repeating behavior of nodes capable of repeating and selective (filtered) repeating.

Repeating is the range extension approach in an EnOcean network. You can find more information in the ERP Specification [3]. Selective repeating is repeating performed only with selected telegrams based on predefined conditions. Repeating and selective repeating have a distinct meaning in the EnOcean network, therefore these attributes and parameters are set and controlled with dedicated RPC commands.

The remote repeater configuration process enables the modification of the status, function and level of the repeater functionality of the device. A remote repeater configuration process, e.g. loading of the IDs to be filtered to the device, takes some time; the duration being dependent on the amount of IDs to be loaded to the device. Without a certain control of the process, the repeater of the device may operate with inconsistent parameter settings, and therefore show unexpected behavior until the repeater configuration process is finished. In order to avoid operation with inconsistent repeater settings the repeater functionality should be turned off with Set Repeater Functions Command during the remote repeater configuration process.

A 1.1. Get Repeater Functions Query & Response

Get Repeater Functions Query	
Function code	0x250
Manufacturer Id	0x7FF
Data length	0 byte
Data content	N/A
Addressed	Yes
Broadcast	No
Command has paired response	Yes
Status return code	Please refer to Section A 2. Query Status return codes.

Table 66 Get Repeater Functions Query

Get Repeater Functions Response		
Function code		0x850
Manufacturer Id		0x7FF
Data length		1 byte
Data content		
	Repeaterfunction	2 bit
	Repeaterlevel	2 bit
	Repeater Filter Structure	1 bit
Addressed		Yes
Broadcast		No

Table 67 Get Repeater Functions response

Data content:

Repeater function (2 bit):

- 0b00 – Repeater Off
- 0b01 – Repeater On
- 0b10 – Filtered Repeating On

Repeater level (2 bit):

- 0b01 – Repeater Level 1
- 0b10 – Repeater Level 2

Repeater Filter Structure (1 bit):

- 0b0 – AND for Repeating
- 0b1 – OR for Repeating

Data structure:

	7	6	5	4	3	2	1	0
1	RepFunc		RepLev		RepStruct			

Table 68 Set Repeater Functions Command data structure

A 1.2. Set Repeater Functions Query

This RPC enables or disables the repeater module of a device. The function “Filtered Repeating” enables the selective repeater functionality. This function needs a valid filter list. Therefore, it is mandatory to set repeater filters before activating selective repeating.

Set Repeater Functions Query		
Function code		0x251
Manufacturer Id		0x7FF
Data length		1 byte
Data content		
	Repeater function	2 bit
	Repeater level	2 bit
	Repeater Filter Structure	1 bit
Addressed		Yes
Broadcast		No
Command has paired response		No
Status return code	Please refer to Section A 2. Query Status return codes.	

Table 69 Set Repeater Functions Query

Data content:

Repeater function (2 bit):

- 0b00 – Repeater Off
- 0b01 – Repeater On
- 0b10 – Filtered Repeating On

Repeater level (2 bit):

- 0b01 – Repeater Level 1
- 0b10 – Repeater Level 2

Repeater Filter Structure (1 bit):

- 0b00 – AND for Repeating
- 0b01 – OR for Repeating

Data structure:

	7	6	5	4	3	2	1	0
1	RepFunc		RepLev		RepStruct			

Table 70 Set Repeater Functions Query data structure

A 1.3. Set Repeater Filter Query

This command is used to set or delete single repeater-filter entries for the selective repeating function.

Set Repeater Filter Query		
Function code		0x252
Manufacturer Id		0x7FF
Data length		5 byte
Data content		
	Filter control	4 bits
	Filter type	4 bits
	Filter value	4 bytes
Addressed		Yes
Broadcast		No
Command has paired response		No
Status return code	Please refer to Section A 2. Query Status return codes.	

Table 71 Set Repeater Filter Query

Data content

Filter control (ADD, FILTER KIND or DELETE) (4 bits):

- 0x0: ADD FILTER, BLOCK KIND = BLOCK REPEATING – this expresses a black list type of filtering
- 0x1: ADD FILTER, APPLY KIND = APPLY REPEATING – this expreses a white list type of filtering
- 0x2: DELETE SPECIFIED FILTER
- 0x3: DELETE ALL FILTERS

Filter type (4 bits):

- 0x0: Source ID

- 0x1: RORG
- 0x2: dBm
- 0x3: Destination ID

Filter value (4 bytes):

- Depending on FILTER TYPE: ID (source / destination) / RORG / dBm

Data structure

	7	6	5	4	3	2	1	0
0	FILTER Control				Filter Type			
1	ID3 / 0x00							
2	ID2 / 0x00							
3	ID1 / 0x00							
4	ID0 / Choice / dBm							

Table 72 Set Repeater Filter Query data structure

A 2. Query Status return codes

The RMCC Query Status requests the last return code from a target device. The return codes are defined in the Remote Management specification [1]. Below is the copy of the list, for detailed explanation please refer to the Remote Management Specification.

Status name	Code number (Query Status response)
OK	0x00
Wrong target ID	0x01
Wrong unlock code	0x02
Wrong EEP	0x03
Wrong manufacturer ID	0x04
Wrong data size	0x05
No code set	0x06
Not send	0x07
RPC failed	0x08
Message time out	0x09
Too Long Message	0x0A
Message part already received	0x0B
Message part not received	0x0C
Address out of range	0x0D
Code data size exceeded	0x0E
Wrong data	0x0F

Table 73 Query Status Return codes

A 3. Command list

Remote Commissioning RPC List – in order of appearance

Function Name	RPC Function Code
Remote Commissioning Acknowledge	0x240
Get Link Table Metadata Query	0x210
Get Link Table Metadata Response	0x810
Get Link Table Query	0x211
Get Link Table Response	0x811
Set Link Table Content	0x212
Get Link Table GP Entry Query	0x213
Get Link Table GP Entry Response	0x813
Set Link Table GP Entry Content	0x214
Get Security Profile Query	0x215
Get Security Profile Response	0x815
Set Security Profile	0x216
Remote Set Learn Mode	0x220
Trigger Outbound Remote Teach Request	0x221
Get Device Configuration Query	0x230
Get Device Configuration Response	0x830
Set Device Configuration Query	0x231
Get Link Based Configuration Query	0x232
Get Link Based Configuration Response	0x832
Set Link Based Configuration Query	0x233
Get Device Security Information Query	0x234
Get Device Security Information Response	0x834

Set Device Security Information	0x235
Apply Changes	0x226
Reset Device Defaults	0x224
Radio Link Test Control	0x225
Get Product ID	0x227
Get Product ID Response	0x827